

Universidade Federal de Juiz de Fora
Faculdade de Engenharia
Curso de Engenharia Elétrica - Habilitação em Robótica e Automação Industrial

Pedro Henrique Teixeira

Deslocamento de Robô Móvel em um Ambiente Controlado por Visão Global

Juiz de Fora

2016

Pedro Henrique Teixeira

Deslocamento de Robô Móvel em um Ambiente Controlado por Visão Global

Dissertação apresentada ao Curso de Engenharia Elétrica - Habilitação em Robótica e Automação Industrial da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Ana Sophia Cavalcanti Alves Vilas Boas

Juiz de Fora

2016

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Teixeira, Pedro H..

Deslocamento de Robô Móvel em um Ambiente Controlado por Visão
Global / Pedro Henrique Teixeira. – 2016.

52 f. : il.

Orientador: Ana Sophia Cavalcanti Alves Vilas Boas

Dissertação (Graduação) – Universidade Federal de Juiz de Fora, Fa-
culdade de Engenharia. Curso de Engenharia Elétrica - Habilitação em
Robótica e Automação Industrial, 2016.

1. Visão Computacional. 2. Detecção de Objetos. 3. Robótica Móvel.
4. Campos Potenciais. 5. Futebol de Robôs. I. Vilas Boas, Ana Sophia
Cavalcanti Alves, II. Dra. em Engenharia Elétrica III. Elias Ramos Vilas
Boas IV. Me. em Engenharia Elétrica

Pedro Henrique Teixeira

Deslocamento de Robô Móvel em um Ambiente Controlado por Visão Global

Dissertação apresentada ao Curso de Engenharia Elétrica - Habilitação em Robótica e Automação Industrial da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

Prof.^a Dra. Ana Sophia Cavalcanti Alves Vilas Boas
Orientador
Universidade Federal de Juiz de Fora

Me. Elias Ramos Vilas Boas
Coorientador
Universidade Federal de Juiz de Fora

Prof. Me. Exuperry Barros Costa
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Gostaria de prestar meus agradecimentos a todos aqueles que me ajudaram nesta jornada. À minha orientadora Ana Sophia Cavalcanti e aos membros da equipe Rinobot pela dedicação e confiança no projeto. Aos colaboradores Elias Vilas Boas, Lucas Netto e Pedro Alcântara que auxiliaram no desenvolvimento das ferramentas e estratégias principalmente no pós projeto. Aos professores que compartilharam conhecimento até então. A meus pais que me deram todas as ferramentas e possibilidades que me permitiram chegar até este ponto. E por fim aos meus amigos, estes que posso contar nos piores e melhores momentos que suportam minhas lamúrias e me dão o “tapa na cara“ que preciso para voltar a realidade.

“The problem is not the problem,
the problem is your attitude about the problem”,
Captain Jack Sparrow

RESUMO

As principais adversidades encontradas por robôs móveis são: se localizar no espaço e projetar um caminho/trajetória que lhes permitam alcançar determinado ponto objetivo. Ambos os problemas precisam ser tratados simultaneamente a fim de se obter um bom resultado. O presente trabalho foi realizado dentro do cenário da competição de futebol de robôs no qual pode se desenvolver uma gama de técnicas para ambientes dinâmicos e multi agentes em tempo real. Técnicas estas que podem ser repassadas para outras aplicações de lógica similar. Associando estas características a esse ambiente, um robô dotado de odometria visual exteroceptiva com uma câmera de posicionamento global é capaz de vencer o primeiro obstáculo, da localização. Enquanto para o segundo obstáculo um algoritmo de planejamento de caminhos como o de campos potenciais permite ao robô alcançar o objetivo. Algoritmo este que aplicado em tempo real permite ao robô desviar-se de obstáculos mantendo sua integridade física. Este trabalho traz o conjunto de técnicas que levaram a completar o desafio proposto: montar um sistema capaz de identificar a posição dos robôs no ambiente, atuar sobre eles evitando colisões, a fim de, com segurança, alcançar determinado ponto objetivo. Alcançando bons resultados de identificação e processamento em tempo real, a equipe de futebol de robôs RINOBOT da UFJF participou da competição LARC 2016 Recife, utilizando o algoritmo desenvolvido neste trabalho.

Palavras-chave: Visão Computacional. Detecção de Objetos. Robótica Móvel. Campos Potenciais. Futebol de Robôs.

ABSTRACT

For mobile robots the main adversities they might find are to self-locate in space and design a path/trajectory that may allow them to reach a goal point. Both problems need to be treated simultaneously in order to achieve good results. This study has been done considering the scenario of robot soccer competition in which it is possible to develop a variety of techniques for dynamic environments and multi-agents in real time applications. And these techniques expanded to other similar logic applications. Associating these characteristics to this environment, a robot who has exteroceptive visual odometry with a camera global positioned is capable to overcome the first challenge, of localization. While for the second obstacle an algorithm of path planning such as Potential Fields allows the robot to achieve its goal. This algorithm used in real time applications also allows the robot to avoid obstacles keeping its physical integrity. This study presents the sum of techniques that were used to complete the proposed challenge: build a system capable to identify the robots position in the field, act on them in order to avoid collisions, so that, with safety, achieve determined position set. Achieving good results of object identification and real time processing, the team of robot soccer from Rinobot from UFJF participated of LARC 2016 competition in Recife, using the same algorithm developed in this study.

Key-words: Computer Vision. Object Detection. Mobile Robotics. Potential Fields. Robot Soccer.

LISTA DE ILUSTRAÇÕES

Figura 1 – Competição de futebol de robôs	14
Figura 2 – Dimensões do campo, categoria VSS	15
Figura 3 – Sistema geral do ambiente de aplicação da categoria VSS.	16
Figura 4 – Robô de pequeno porte também apresentando sensores IR.	17
Figura 5 – Fluxo de dados da identificação dos componentes e decisão, [10].	18
Figura 6 – Componentes de placa do robô.	20
Figura 7 – Componentes mecânicos do robô. Roda sofreu modificação, Motor de redução	21
Figura 8 – Componentes de placa do robô.	21
Figura 9 – Esquemático do robô utilizado	23
Figura 10 – Protocolo de comunicação, Conceito, Tamanho em Bytes.	24
Figura 11 – Exemplo de envio do protocolo de comunicação.	25
Figura 12 – Exemplo de funcionamento de campos potenciais no espaço.	26
Figura 13 – A esquerda imagem original, no centro atuação do filtro de dilatação aumentando as bordas dos pixels da imagem, e a direita a erosão de pixels.	28
Figura 14 – Cubo representando espectro de cores RGB.	29
Figura 15 – Espectro de cores em HSV.	29
Figura 16 – Camera firewire Utilizada.	30
Figura 17 – Conjunto de imagens para análise de distorção.	31
Figura 18 – Tratamento da malha quadriculada.	32
Figura 19 – Resposta da correção de distorção da imagem na ROI representando o campo.	32
Figura 20 – Esboço da detecção de contorno.	36
Figura 21 – Região de interesse do robô a partir da cor padrão.	37
Figura 22 – Situações de identificação comprometida.	39
Figura 23 – Identificação de margem.	41
Figura 24 – Localização esfera objetivo.	42
Figura 25 – Filtro de cor padrão, amarelo.	43
Figura 26 – Abertura da região de interesse para localização de contorno robô.	43
Figura 27 – Resultado final da localização.	44
Figura 28 – Situação de erro, frame único.	45
Figura 29 – Log de Eventos, seguindo uma trajetória.	46
Figura 30 – Log de eventos, análise movimentação do robô móvel	46
Figura 31 – Log de eventos, obstáculos mais parede.	47
Figura 32 – Obstáculo a frente.	47
Figura 33 – Esfera objetivo sofrendo impulso e se movendo no universo de controle.	48

LISTA DE ABREVIATURAS E SIGLAS

CC	Corrente Contínua
cm	Centímetro
FIRA	<i>Federation of International Robot Soccer Association</i>
fps	Frames por Segundo
HSV	Sistema de cores Matiz, Saturação, Intensidade
ID	Identificação
IR	Infra-vermelho
LARC / CBR	Competição Latino Americana e Brasileira de Robótica
mm	Milímetro
px	Pixel
RGB	Sistema de cores Vermelho, Verde, Azul
ROI	Região de Interesse
SSL	<i>Small Size League</i>
UFJF	Universidade Federal de Juiz de Fora
VSS	<i>Very Small Size</i>
VSSS	<i>Very Small Size Soccer</i>

LISTA DE SÍMBOLOS

F_{att}	Força de Atração
K_{att}	Constante de Atração
F_{rep}	Força de Repulsão
K_{rep}	Constante de Repulsão
F_{tot}	Força Total
Pos.	Posição
Vel.	Velocidade
φ	Velocidade de giro da roda
V	Velocidade Linear
W	Velocidade Angular
r	Raio da roda
L	Distância entre as rodas do robô
ϵ	Tamanho do Raio do Universo de Eventos
D	Distância robô obstáculo
kw	Constante de velocidade angular
vmax	velocidade máxima

SUMÁRIO

1	INTRODUÇÃO	11
1.1	MOTIVAÇÃO	11
1.2	OBJETIVOS	12
1.3	ORGANIZAÇÃO DO TRABALHO	12
2	AMBIENTE DE APLICAÇÃO	14
3	ANÁLISE HISTÓRICA	17
4	FERRAMENTAS E MÉTODOS	20
4.1	ROBÔ MÓVEL	20
4.1.1	Características Construtivas	20
4.1.2	Modelagem	22
4.1.3	Comunicação	23
4.2	PLANEJAMENTO DE CAMINHO	25
4.3	VISÃO COMPUTACIONAL	27
4.3.1	OpenCV	27
4.3.2	Filtros	27
4.3.3	Cores	28
4.3.4	Correção de Distorção da Imagem	29
4.4	INTEGRAÇÃO DAS ETAPAS	32
4.4.1	Captação de Informações	33
4.4.2	Integração	34
5	RESULTADOS	41
5.1	IDENTIFICAÇÃO DE PARÂMETROS E LOCALIZAÇÃO	41
5.1.1	Laterais do Campo	41
5.1.2	Filtro de Cor	42
5.1.3	Identificação dos Robôs	43
5.1.4	Correção de erros	44
5.2	RESPOSTAS DO ROBÔ EM MOVIMENTO	45
6	CONCLUSÃO	49
6.1	TRABALHOS FUTUROS	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

Os robôs estão cada vez mais presentes no cotidiano das pessoas. No século XXI é difícil encontrar alguém que não conhece o termo ou não consegue imaginar a ação de um desses autômatos dentro de uma indústria ou realizando uma exploração espacial.

O cenário de robótica pode ser dividido em dois ramos principais de estudo: os manipuladores robóticos e os robôs móveis. Os manipuladores robóticos são um conjunto de *links* interligados entre si por juntas, resultando em uma cadeia cinemática, tais quais os braços mecânicos. Os robôs móveis, foco de estudo deste trabalho, possuem a capacidade de se locomover no espaço seja por rodas, ou por meio de atuadores. Ambos estão em constante desenvolvimento, aumentando sua confiabilidade, autonomia e reduzindo custos, sendo empregados nas mais diversas áreas.

“Um dos grandes desafios da robótica móvel nos dias atuais é a obtenção de informações sobre o ambiente onde se está inserido, a fim de que, a partir desses dados possa ser realizado um conjunto de tarefas, sem que seja necessária a intervenção humana.” [17].

Ainda nesta colocação de identificação de informações do ambiente “o grau de autonomia normalmente depende do tamanho dos robôs”. [10].

Para que um robô atue corretamente é necessário que esteja sujeito a um controle adequado de acionamento de seus componentes mecânicos e para tal costuma-se utilizar sistemas em malha fechada. Uma gama de sensores proprioceptivos para captação de informações de si mesmo, como os encoders medidores de velocidade nas rodas, e exteroceptivos atuando na identificação de parâmetros externos ao robô se tornam de suma importância para automatização do processo, permitindo ao próprio robô até mesmo tomar decisões.

“A capacidade de calcular aproximadamente a sua posição proporciona a um robô uma maior autonomia e os meios necessários para realizar múltiplas e importantes tarefas.” [16]

1.1 MOTIVAÇÃO

Carros automáticos que não requerem mais um piloto estão em *vogue* na mídia. Sondas espaciais realizam pesquisas em outros planetas, enquanto pequenos robôs aspiradores de pó são utilizados para limpeza em indústrias. Estes são exemplos de robôs móveis que trabalham por meio de uma inteligência artificial, fazendo-se desnecessária a presença de um operador na tomada de decisões.

Para que os sistemas citados acima alcançassem tais feitos, não só sua mecânica sofreu um avanço, como também toda sua eletrônica, em especial a parte de sensoriamento.

É de fundamental importância para profissional que está ingressando no mercado de trabalho ou pesquisador compreender como um sistema deste funciona e como é feita a interligação de diversos conceitos para trazer à tona o potencial da máquina com o que se tem a disposição.

1.2 OBJETIVOS

O objetivo deste trabalho é a montagem de um sistema de visão computacional global, voltado para um jogo de futebol de robôs categoria *Very Small Size* (VSS). Este sistema é capaz de identificar a posição dos robôs no ambiente, atuar sobre eles evitando colisões, a fim de, com segurança, alcançar determinado ponto objetivo.

O objeto de controle é um robô móvel diferencial. Este robô possui duas rodas laterais e outras duas rodas livres para manter o equilíbrio. O ponto objetivo a ser alcançado pelo robô, tendo como base um jogo de futebol de robôs, é a bola do jogo. A bola é uma esfera laranja, a qual para este trabalho o robô deve alcançá-la mesmo que em movimento, com base no sistema em tempo real.

Quando se trata de sistema de visão global, entende-se como sendo uma câmera posicionada externamente (sensor exteroceptivo), capaz de ter visão de toda a região de interesse em que ocorre a ação. Sendo assim, a câmera é posicionada acima de todo o universo de trabalho, para ser capaz de identificar os pontos de interesse.

Por fim, a locomoção é realizada por meio do algoritmo de campos potenciais. Capaz de não só direcionar o robô móvel até o objetivo como também desviar de obstáculos, tendo o conhecimento de suas posições. Fornecidas em tempo real pelo sistema de visão. Os obstáculos serão as paredes e qualquer outro corpo não identificado como o robô móvel ou o objetivo.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido nos seguintes capítulos para expor de maneira clara e objetiva como foi desenvolvido:

O Capítulo 2, Ambiente de Aplicação, descreve brevemente o cenário para qual o trabalho foi desenvolvido, permitindo analisar dentro deste ambiente problemas relevantes ou não. Assim compreender corretamente o problema, permitindo importar e/ou exportar soluções de problemáticas similares.

O Capítulo 3, Análise Histórica, traz de forma sucinta uma visão de trabalhos anteriores sobre o mesmo tema. Descreve metodologias de abordagem sobre a situação e formas de solucionar os desafios propostos, consolidando as estratégias utilizadas neste trabalho.

O Capítulo 4, Ferramentas e Métodos, apresenta os materiais que foram utilizados para montar o sistema. Inicialmente descrevendo a parte física sobre o robô, sua modelagem e como é realizado o planejamento de caminho. Este capítulo também aborda a visão computacional, câmera e correção da imagem e, por fim, a identificação dos objetos na imagem e como interligar todo o processo.

O Capítulo 5, Resultados, por sua vez, aborda a implementação prática, demonstrando todos os casos descritos no capítulo 4 e permite visualizar a resposta no universo real. Avaliando os resultados, por meio de imagens ilustrativas, da resposta apresentada.

2 AMBIENTE DE APLICAÇÃO

“O ano de 1997 é de muita importância para os sistemas de inteligência artificial. Neste mesmo ano o primeiro computador (IBM Deep Blue) derrotou o campeão mundial em xadrez, a missão *pathfinder* da NASA, conseguiu pousar com sucesso utilizando um sistema robótico autônomo, enquanto a RoboCup iniciou o processo de desenvolvimento de um time de futebol de robôs capaz de derrotar um time humano.“ [21]

Após a derrota do campeão de xadrez para um autômato, a ciência necessitava de novas maneiras de se provar e avançar, propondo então novos desafios.

Comparando um jogo de xadrez e um jogo de futebol de robôs, com base na Tabela 1, observa-se uma complexibilidade para jogo de futebol, pois este possui um ambiente dinâmico, operado em tempo real e com limitações na aquisição de dados, como ilustrado na Figura 1. Construindo um ambiente desafiador, favorável para atrair novos avanços em múltiplas áreas e trazer investimentos para o cenário de avanço da robótica.

Tabela 1 – Comparativo entre jogo de xadrez e futebol de robôs.

	Xadrez	Futebol de Robôs
Ambiente	Estático	Dinâmico
Mudança de Estados	Em Turnos	Tempo Real
Acessibilidade à Informação	Completa	Incompleta
Leitura de Sensores	Simbólica	Não Simbólica
Controle	Central	Distribuído

Fonte: <http://www.robocup.org/leagues/6>



Figura 1 – Competição de futebol de robôs

“A primeira competição oficial da RoboCup foi realizada em 1997 com grande sucesso, e um total de 40 times em diversas categorias contando com mais de 5000

espectadores.“ [21]. As competições da RoboCup se dividem entre uma gama de desafios, desde o futebol de robôs até aplicações em casa, resgate e industrial. Ainda com subdivisões entre si dentro de razões de funções, tamanho, ambiente simulado, dentre outros.

A aplicação deste trabalho se encontra voltada dentro do cenário da competição VSSS (Very Small Size Soccer). As regras são simples, similares ao jogo tradicional, tendo um campo e robôs de dimensões previamente determinadas. O Campo, mostrado na Figura 2, é a região de interesse onde todas as ações da visão computacional são tomadas. Os robôs precisam desviar dos obstáculos, outros robôs, e alcançar o objetivo de encontrar a bola.

Fonte: Manual da Competição VSSS

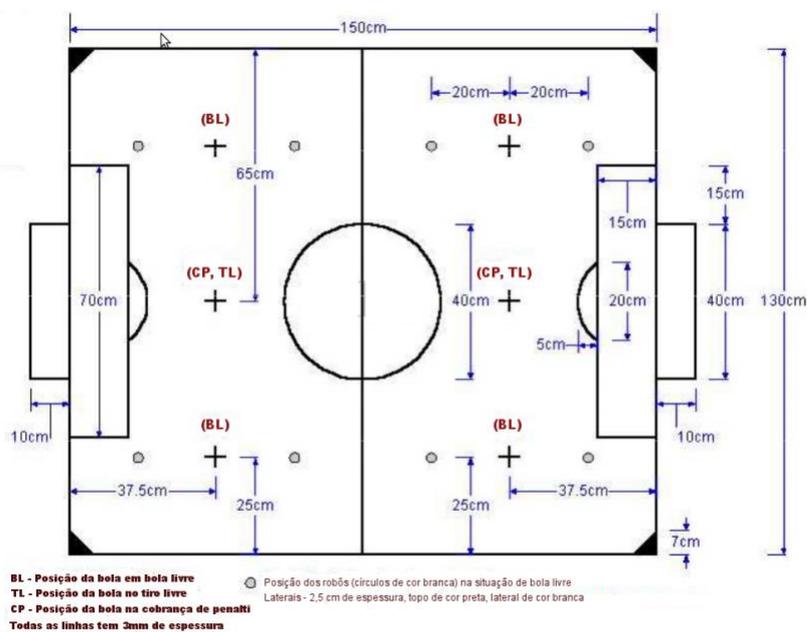


Figura 2 – Dimensões do campo, categoria VSS

O robô desta categoria apresenta duas principais limitações, sendo elas tamanho e identificação. O tamanho não pode exceder um cubo de “7,5 x 7,5 x 7,5 cm” e a identificação acima do robô deve contar com pelo menos uma cor padrão e não pode haver outras cores já reservadas, como branco, preto e laranja.

Sabendo onde atuar, pôde-se limitar o conjunto de ações a serem tomadas visando os aspectos construtivos dos objetos a serem identificados no ambiente. Segue as considerações para o algoritmo de controle:

- Um campo liso e planar de fundo preto com linhas finas (3mm) e bordas brancas.
- Identificação da bola laranja, como objetivo.
- Robô objeto e obstáculo com uma cor padrão de localização.

- Para o robô de controle uma segunda cor para obter orientação.
- Planejar o caminho desviando dos obstáculos, outros robôs e paredes.
- Não existirá nenhum outro corpo no universo de trabalho, e as cores para identificação são de conhecimento prévio.

A Figura 3, ilustra o sistema no qual a competição se desenvolve. Cada computador recebe o sinal da câmera e processa os dados, envia os comandos aos robôs do respectivo time. Esta funcionalidade de “cérebro externo” diminui o tamanho do robô, não sendo necessário processadores tão potentes no corpo do mesmo. Ao passo que o computador permite visualizar todas as situações com uma câmera de ação global tendo a visão de todo universo de trabalho.

Fonte: Manual da Competição VSSS

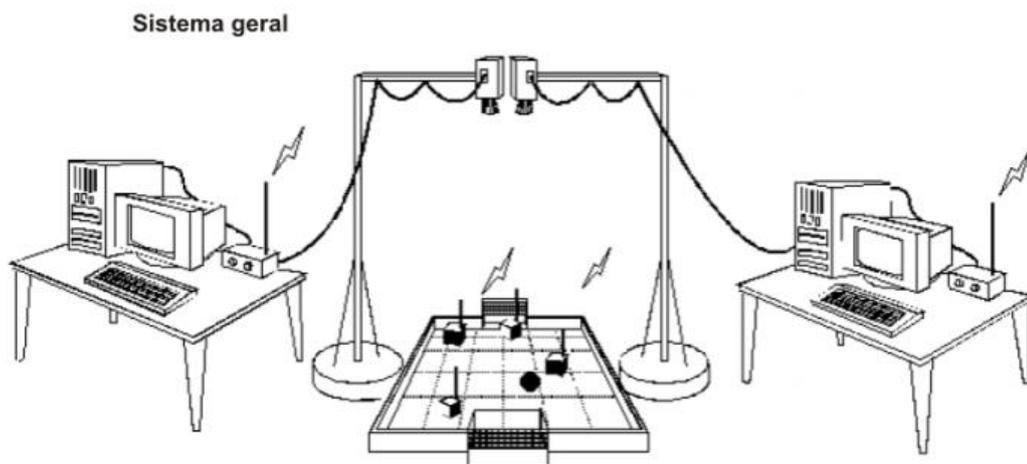


Figura 3 – Sistema geral do ambiente de aplicação da categoria VSS.

Por apresentar um bom suporte para a computação científica optou-se por desenvolver o programa na linguagem de programação Python. Esta linguagem de software aberto e alto nível é utilizada em diversas aplicações. Por ser bem difundida conta com um bom suporte para pesquisas e bibliotecas para tratamento de dados. O Python possui as bibliotecas do OpenCV (visão computacional) e a biblioteca pyserial para comunicação com o arduino, microprocessador utilizado para controle do robô. Outras informações sobre a programação e as bibliotecas utilizadas serão descritas no Capítulo 4.

3 ANÁLISE HISTÓRICA

A problemática de criar um time de futebol de robôs vem sendo tratada desde o início das competições em 1997. E várias equipes de pesquisa em todo o mundo se empenharam em adaptar seus estudos dentro de um contexto de competição para incentivar e testar a criação de novas soluções. Cada categoria tem suas demandas na combinação de robôs e sistema de visão. Estas categorias se dividem entre níveis de autonomia, podendo ser controlados pelo computador e visão centralizada, apresentar visão local e ser completamente autônomo ou até mesmo um *design* intermediário como exemplifica Weiss, N. [10].

Em 1997 Kim, J.H. [12] publica seu estudo de como tratar um ambiente de multi agentes dentro do cenário competitivo da FIRA MiroSot e quais foram as estratégias na tomada de decisões.

“Do ponto de vista de sistema multi agentes, o futebol, que é apenas um dos usuais jogos de time, é um bom exemplo de problemas do mundo real que podem ser moderadamente abstratos”. [12]).

Neste mesmo estudo dividiu-se o cenário para diferentes aspectos de controle para os multi agentes, dentre eles o sistema de complexidade intermediária apresentando um supervisor de visão para controle de posição e velocidade, porém os robôs ainda contavam com outros sensores, como os infravermelhos para evitar obstáculos, como apresentado na Figura 4 .

Fonte: [12]

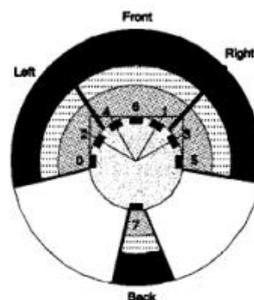


Figura 4 – Robô de pequeno porte também apresentando sensores IR.

Em 2004 Ball, D. [2], realiza um estudo sobre o sistema de visão em tempo real aplicado na categoria SSL (*small size league*). Este artigo deixa claro as dificuldades enfrentadas ao se implementar este tipo de sistema, como lidar com variação de luz aleatória e desconhecida, pouco tempo para calibrar os parâmetros e equipamentos, a não uniformidade das cores em times distintos e identificar corpos que podem se mover a uma velocidade elevada. “Como o sistema de visão global é o sensor primário para os robôs, é crítico que seja robusto para o sucesso na competição.” [2].

Em [11] (Khessal, N. O.), para que o sistema saiba a situação geral em campo, um dos problemas chaves que precisa ser solucionado é a auto localização do robô jogador. Dado o sistema dinâmico do ambiente de jogo do futebol, onde apenas o campo permanece estático, a forma de tratar a imagem afeta diretamente o resultado, como exemplifica [11] Apesar de que podiam obter mais informações, optaram por um sistema mais simples, porém de processamento mais rápido ainda alcançando bons resultados.

Em [10] (Weiss, N.), cita mais pontos sobre como atuar na detecção de corpos, atuando em manchas/conglomerados (*blobs*) de cores específicas, classificando os pixels por cor e relevância. Ao mesmo passo consegue separar os sistemas de visão e controle, Figura 5

Fonte: Traduzido de [10].

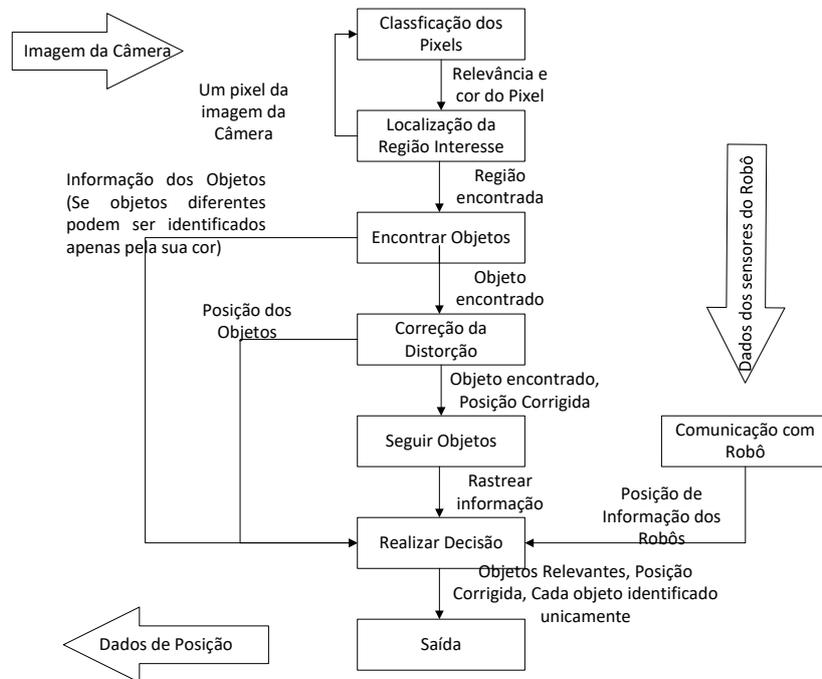


Figura 5 – Fluxo de dados da identificação dos componentes e decisão, [10].

Ainda em [10], a detecção é realizada pelas cores no espaço HSV, será mais detalhado a frente na seção 4.3.3, utilizando todos os três parâmetros calculados individualmente e comparados a valores esperados, sendo essa uma boa forma de localizar os robôs. Já quando trata de detecção:

“Alguns algoritmos identificam os objetos por comparação de uma posição anterior com a posição atual mais próxima. Isso tende a ser mais rápido, porém mais instável especialmente com mais robôs e mais próximos no espaço”. [10]

Soluções como predição de trajetória são propostas, porém outras soluções se bem implementadas apresentam bons resultados, como no caso de [4], que aplicou uma varredura global da imagem frame a frame. Ainda assim é importante frisar a necessidade do sistema apresentar boa resposta e um bom processamento.

“Um cuidado que se deve ter é manter a taxa de frame por segundo sempre superior a 24fps, pois menos que isso, os atrasos entre cada frame, já são perceptíveis à visão humana.” [16].

Ainda nessa dinâmica, muitos dos trabalhos com câmeras utilizam os sistemas de detecção por HSV, dado o fato de ser um sistema mais robusto e que permite uma melhor separação dos valores. Em contrapartida: “Não precisar converter a imagem de RGB para um espaço de cores distinto aumenta dramaticamente a performance do sistema” [2].

Ao se tratar de movimentação, em 1998, Kim, J.H. [13], fala sobre o comportamento em campo dos multi agentes, no qual depende de um processamento rápido que se enquadre na estrutura do robô utilizado. E então utiliza o método de campos potenciais, método este que é geralmente utilizado no controle de robôs móveis por sua simplicidade e funcionalidade (boa resposta) em tempo real. Ainda para reforçar esta visão, como cita Silva, M. O. D. [23] “É um método de baixo custo computacional (pois necessita apenas somar vetores)“. Ambas as abordagens de campos potenciais foram eficazes dentro do universo dinâmico em tempo real para se locomover corretamente e desviar de obstáculos mantendo assim a integridade física do robô.

4 FERRAMENTAS E MÉTODOS

Neste capítulo são discutidas as principais ferramentas utilizadas ao longo do desenvolvimento deste trabalho. Partindo da característica construtiva do robô móvel utilizado e sua modelagem dentro do universo de trabalho, passando pela aplicação do planejamento de caminho e como evitar obstáculos, alcançando ao fim a teoria de visão computacional, utilizando as bibliotecas de *software* aberto pela OPENCV, para detectar os pontos de interesse para a aplicação.

4.1 ROBÔ MÓVEL

Um robô móvel pode ser definido como um autômato capaz de se locomover no espaço modificando sua orientação e posição ao longo do mesmo. Utilizados em diversos ambientes e sendo expostos a várias adversidades, necessitam identificar sua correta localização, além de planejar e executar o caminho que deve ser seguido.

Para o caso específico da competição de futebol de robôs, o ambiente conta com algumas situações pré estabelecidas como: ser planar e contar apenas com objetos conhecidos para identificação. Toda a visão do ambiente é realizada por meio de sensores exteroceptivos e portanto um dos maiores problemas da robótica móvel, “o roubo do robô”, pode ser desconsiderado, o robô nunca se perde no ambiente de controle. Caso ocorra a translocação ao longo do espaço sem ação interna, por meio do usuário por exemplo, o novo caminho é recalculado em tempo real para a nova posição, atualizada a cada *frame* da câmera.

4.1.1 Características Construtivas

O circuito eletrônico é constituído pelos seguintes componentes, Figura 6:

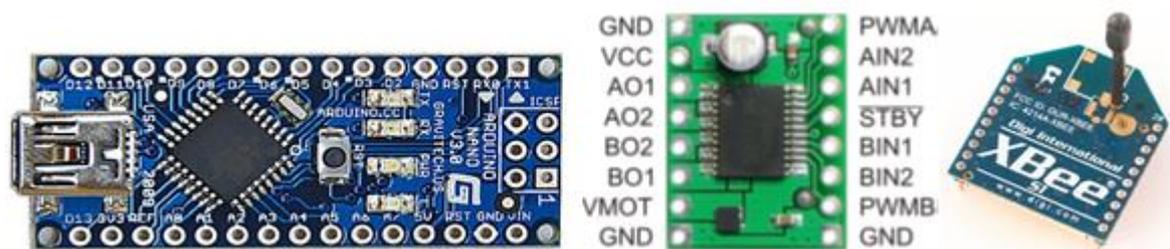


Figura 6 – Componentes de placa do robô.

- Arduino Nano V3, ATmega328, para controlar o robô.
- Ponte H TB6612, para acionar os motores.
- XBee S1, para realizar a comunicação com o computador.

- Dois botões para interrupção da alimentação e da comunicação Arduino/XBee.

O conjunto roda-motor, mostrado na Figura 7, é construído pela empresa *Pololu Robotics and Electronics*. Estas rodas apresentam dimensões de 42x19mm, acionadas por um motor CC, escovado, reduzido 100:1. O conjunto ainda conta com um encoder de quadratura que se encaixa entre o motor e a roda para realizar o controle de velocidade. São duas rodas laterais e mais duas livres para equilibrar o robô. As rodas laterais são motorizadas e estão espaçadas de 60 mm. Na Figura 7, são mostrados os componentes motor e conjunto roda encoder utilizado na parte mecânica do robô diferencial, apesar de a roda ter tido o pneu modificado para a aplicação.

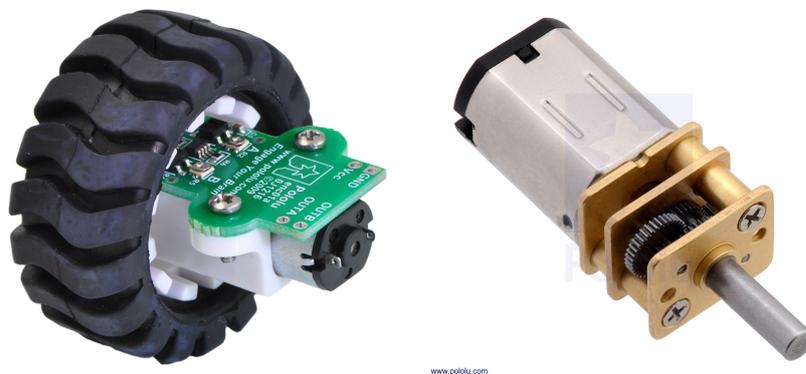


Figura 7 – Componentes mecânicos do robô. Roda sofreu modificação, Motor de redução

Sabendo das necessidades construtivas deste robô: tamanho reduzido e sem necessidade de um microcontrolador mais potente, estes equipamentos são pequenos em dimensão e se adequam a aplicação. Ao todo, a estrutura se mantém dentro do tamanho especificado para competição VSSS, apresentando um formato cúbico com 7,5cm de lado.

A Figura 8 apresenta o robô diferencial utilizado para este estudo. O mesmo robô utilizado pela equipe Rinobot, da UFJF em sua primeira participação em competição de futebol de robôs na liga VSSS.

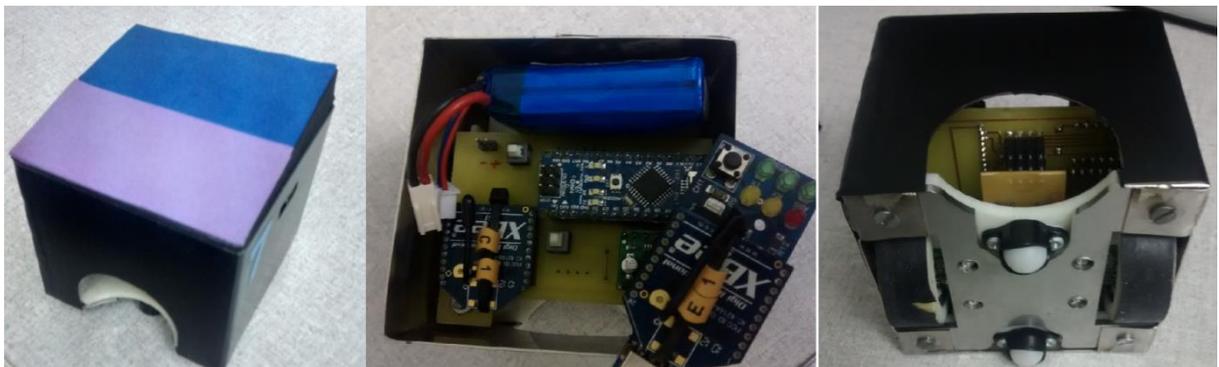


Figura 8 – Componentes de placa do robô.

4.1.2 Modelagem

A cinemática é o ramo da física que estuda a descrição do movimento dos corpos não se preocupando com a dinâmica de forças dos mesmos. Para modelar a cinemática do robô móvel diferencial será levado em conta apenas como pode e deve se mover. Isto é, não foi levado em consideração o atraso de resposta dos motores, o peso do sistema ou a inércia do movimento, por exemplo.

Primeiro, identifica-se as características físicas e espaciais do robô. Possuindo duas rodas laterais, espaçadas em L , motorizadas, apresenta um total de dois graus de liberdade, alcançando qualquer posição e orientação no espaço cartesiano de duas dimensões. O robô diferencial apesar de conseguir alcançar qualquer posição e orientação no ambiente bidimensional ele não é holonômico, não pode se locomover em qualquer direção, apenas para onde suas rodas se orientam.

É necessário identificar a velocidade linear (V) e angular (W) do robô para realizar o controle de posição, a Equação 4.1 apresenta a velocidade linear em função da velocidade angular e o raio (R) do movimento. A velocidade de cada roda pode ser dada pelo deslocamento que ela causa em função de seu raio (r), sendo sua velocidade de giro dado por φ , vide Equação 4.2.

$$V = W \cdot R \quad (4.1)$$

$$vd = r \cdot \varphi_d \quad ve = r \cdot \varphi_e \quad (4.2)$$

Decompondo o vetor de velocidade, a partir da velocidade individual de cada roda, direita (vd) e esquerda (ve), até o ponto central do robô para identificar sua velocidade linear resultante, tem-se a Equação 4.3:

$$V = \frac{(r \cdot (\varphi_d + \varphi_e))}{2} = \frac{(vd + ve)}{2} \quad (4.3)$$

Sabendo a distância entre as rodas do robô (L) e tendo em vista que $L/2$ representa o raio do movimento de cada roda até o centro do robô. Pode-se identificar a velocidade angular do robô partindo da velocidade individual de cada roda. Segue equações 4.4 e 4.5:

$$W = \frac{V}{\frac{L}{2}} \quad (4.4)$$

$$W = \frac{(vd + ve)}{L} \quad (4.5)$$

Cada motor deve ser acionado individualmente pelo microcontrolador. Portanto a velocidade que ao fim deve ser enviada é a velocidade individual de cada roda. Tomando

o robô no espaço como um ponto objetivo é possível caracterizar esta velocidade pela decomposição dos vetores encontrados como resposta do algoritmo de planejamento de caminho, V e W . Pode ser visto o esquemático do robô na Figura 9.

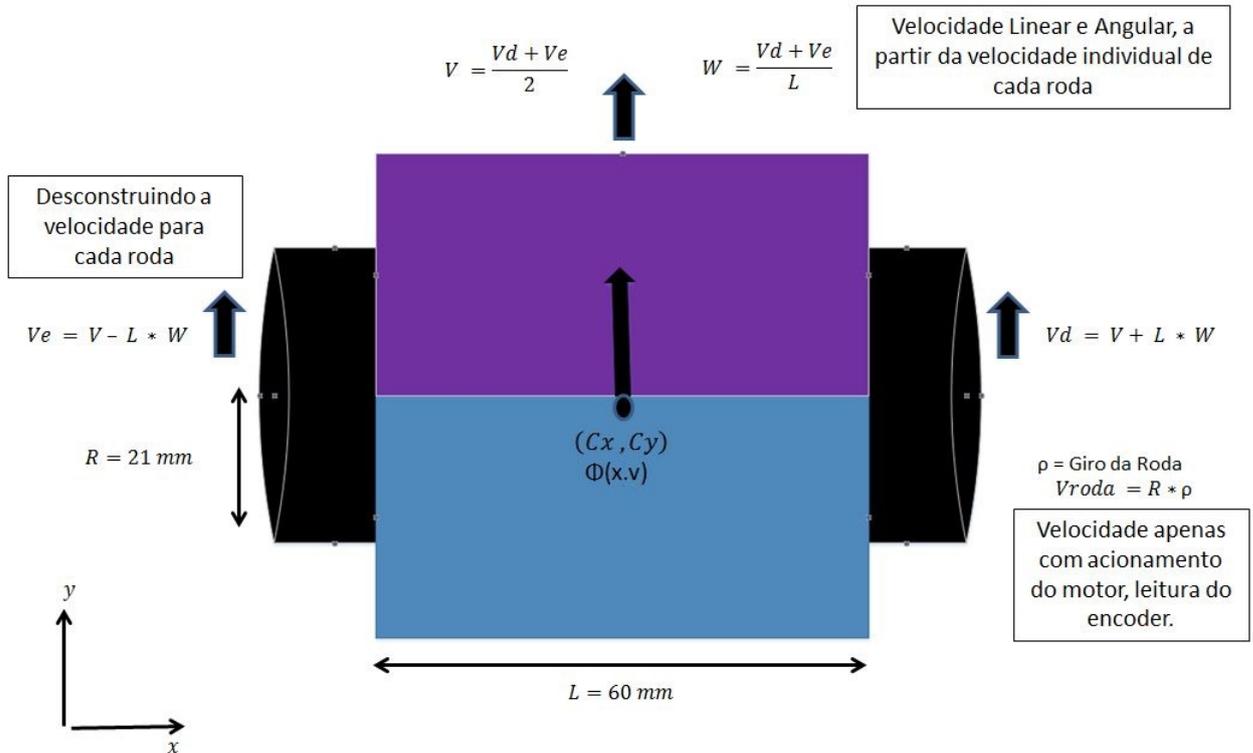


Figura 9 – Esquemático do robô utilizado

Realizando o inverso das Equações 4.3 e 4.5 para obter a velocidade que cada roda deve ter para efetuar o movimento, e seguindo a regra de Fleming (mão direita), sendo o giro para a esquerda apresentando valores positivos, tem-se as Equações para cada roda 4.6 e 4.7:

$$vd = V + L \cdot W \quad (4.6)$$

$$ve = V - L \cdot W \quad (4.7)$$

4.1.3 Comunicação

A comunicação com o robô é feita com processamento externo, sendo realizado todos os cálculos para determinar a ação necessária no computador. Então, um protocolo é enviado por meio de um xbee, módulo RF, para o arduino do robô acionar corretamente os motores e atuar para realizar a função.

Desta forma, apesar do xbee ser um comunicador seguro com diferentes canais de comunicação, o protocolo precisa ser robusto para evitar que dois robôs confundam suas ações ou que algum sinal aleatório prejudique a ação de controle.

O protocolo utilizado apresenta esta segurança, contando com seis parâmetros resultando em um total de treze bytes. Tem seu início com 1 byte padrão que permite ao micro controlador saber que deve iniciar a leitura, segue então a identificação do robô, apresentando também um único byte que traz o valor inteiro que o representa. A velocidade individual de cada roda é enviada em seguida, contando com 4 bytes cada para representar números reais (ponto flutuante) e podendo ser negativos.

Tendo os valores a serem utilizados no controle é necessário realizar a verificação se os dados foram recebidos corretamente. São enviados 2 bytes que representam a soma de todos os anteriores, a mesma soma é feita no arduino e caso o resultado seja o mesmo significa que a leitura foi realizada com sucesso. Por fim, um último byte é enviado para fechar a comunicação.

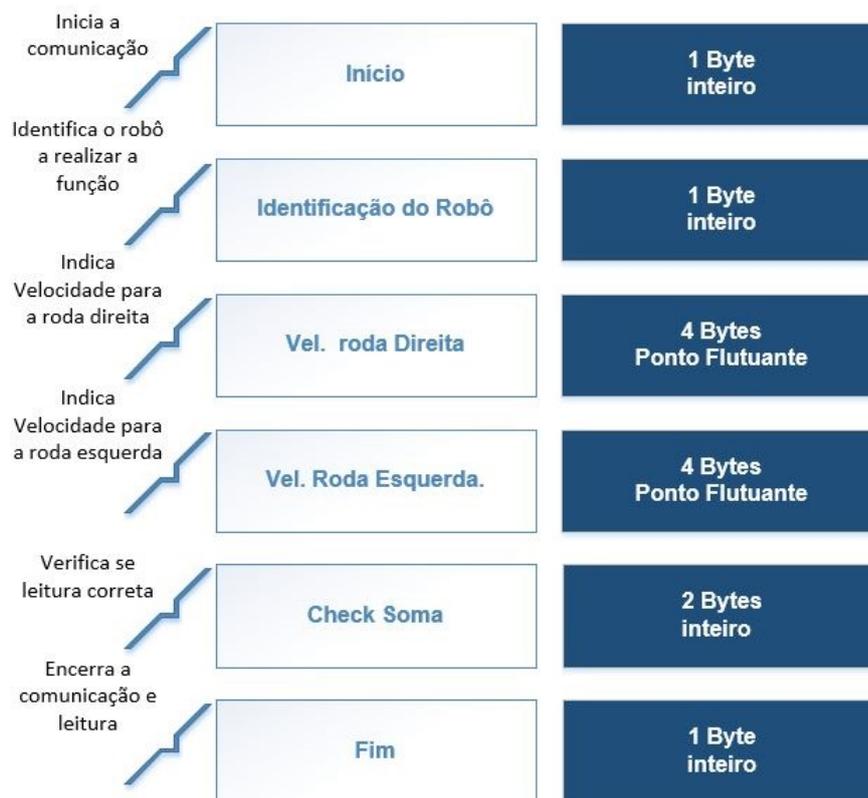


Figura 10 – Protocolo de comunicação, Conceito, Tamanho em Bytes.

Na Figura 10, observam-se as divisões do protocolo, com cada campo apresentando uma certa quantidade de bytes, porém enviados simultaneamente e identificados corretamente no microcontrolador para ser validado. A Figura 11 apresenta um exemplo numérico em hexadecimal de como é realizada a transformação e envio da informação.

O *buffer* é limpo constantemente após a leitura e execução de um protocolo, o mesmo ocorre havendo falha ou quando sobrecarregado. Dessa forma, impede-se que ocorra uma leitura incorreta de um sinal incompleto, um sinal falho que todo acúmulo de treze bytes em diante esteja comprometido, ou desordenado cronologicamente, isto é, sempre lê-se o último valor enviado em tempo real para não realizar o controle baseado em uma informação retroativa.

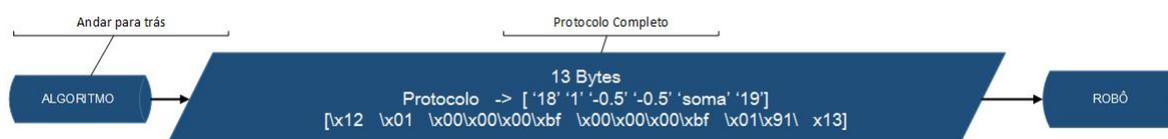


Figura 11 – Exemplo de envio do protocolo de comunicação.

4.2 PLANEJAMENTO DE CAMINHO

Um dos maiores problemas da robótica móvel é a auto localização do robô no espaço. Em virtude da câmera externa tendo visão global do universo, o sistema sempre identifica a posição dos objetos e mesmo casos como o "roubo do robô" podem ser ignorados. O outro grande problema é o robô identificar qual o melhor caminho que deve tomar para alcançar o objetivo. Existem vários algoritmos capazes de traçar uma trajetória do ponto atual até o objetivo, tais quais o algoritmo Bug, A* (A estrela) e o de Campos Potenciais.

Este trabalho utilizou o algoritmo de campos potenciais para o planejamento de caminho. Esse algoritmo foi adaptado para robótica móvel no conceito de evitar colisões por Khatib em 1985. Esta técnica necessita das informações de localização dos obstáculos, robô e objetivo e mesmo em tempo real consegue ser realizada com eficiência, simplicidade e rapidez, reprocessando os dados sem muito esforço computacional mesmo em ambientes dinâmicos.

Esta técnica trata os pontos como se fossem partículas eletricamente carregadas. Imaginando o robô móvel como sendo uma carga positiva e o ponto objetivo como uma carga negativa, corpos de cargas diferentes se atraem, logo uma força de atração (F_{att}) é gerada e tende a ser orientada para o ponto objetivo. Os obstáculos podem ser qualificados com objetos também de carga positiva (a mesma do robô), assim corpos de cargas iguais se repelem gerando uma força de repulsão (F_{rep}). Os vetores são identificados utilizando também teoria do gradiente descendente. A soma dos dois vetores, de repulsão e de atração, indica o melhor caminho a cada iteração que o robô deveria seguir. Composto não só a direção e sentido do movimento, com base na normalização do vetor resultante (F_{tot}), pode-se obter a velocidade com que o robô deve estar. Logo, perto de um obstáculo ou objetivo é ideal que reduza a sua velocidade para atuar com mais precisão.

As equações 4.8 e 4.9 exemplificam as forças de atração e repulsão que se baseiam na teoria de gradiente descendente. A força de atração é inversamente proporcional a distância entre os objetos, quanto mais longe do objetivo, maior a força, assim como um parabolóide tendo o objetivo no mínimo local. Enquanto a força de repulsão quanto mais próxima do obstáculo tende ao infinito, apresentando o formato de um vulcão com assíntotas infinitas. A Figura 12 exemplifica os vetores força ao longo de um espaço e como a soma destes compõem o movimento.

$$F_{att} = K_{att} \cdot [Pos_{objetivo} - Pos_{robo}] \quad (4.8)$$

$$F_{rep} = K_{rep} \cdot \sum_{i=1}^n \left(\frac{1}{e_i^3} \cdot \left(\frac{1}{e_0} - \frac{1}{e_i} \right) * [Pos_{obstaculo} - Pos_{robo}] \right) \quad (4.9)$$

Fonte: (Silva, 2011)

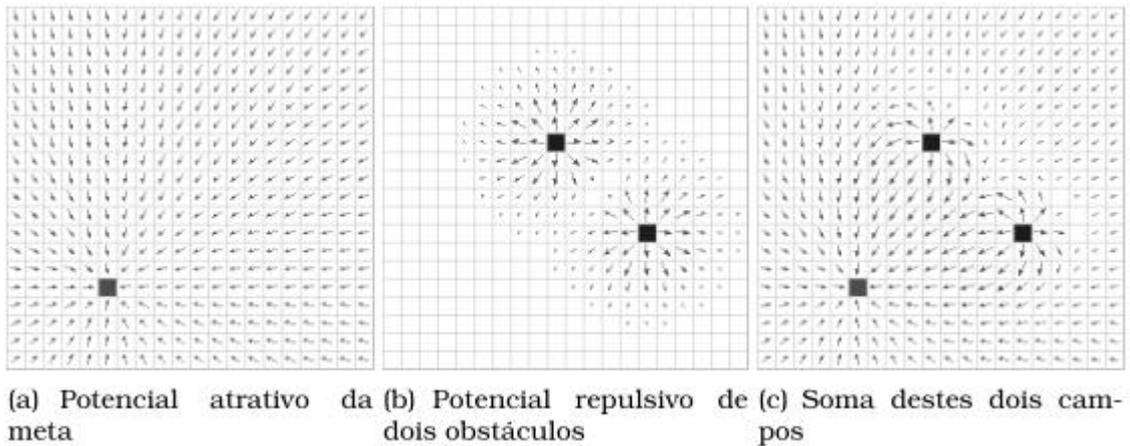


Figura 12 – Exemplo de funcionamento de campos potenciais no espaço.

Este método conta então com a segurança de desviar de obstáculos, garantindo a integridade do robô, e rápido processamento sendo realizada a realimentação de dados do ambiente a cada iteração. Problemas como o de encontrar um mínimo local, um ponto onde ele está livre de forças ou que os vetores se redirecionem para um ponto em que não consegue escapar, como dentro de uma base côncava, por exemplo, impedem que alcance o objetivo.

Ainda assim, dentro do cenário de aplicação, sendo o jogo um ambiente dinâmico, a probabilidade de que um mínimo local se mantenha por muito tempo, é baixa. Não só isso, como pode ser uma vantagem, mantendo o jogador em posição próxima, marcando o objetivo, mesmo que não possa alcançá-lo e sem cometer uma falta.

4.3 VISÃO COMPUTACIONAL

A visão computacional é um conceito de tratamento de imagens que procura emular a visão humana tendo uma interpretação da imagem como um todo. Ao contrário, o processamento de imagens capta uma imagem e interpreta a mesma numericamente. Para tratar a resposta do sensor exteroceptivo, a câmera de visão global do ambiente, são utilizados softwares de computador, a biblioteca OPENCV que permite filtrar ruídos (baixo nível), identificar objetos (nível médio) e então realizar o controle do corpo baseado nesses dados (alto nível).

4.3.1 OpenCV

A biblioteca OpenCV foi desenvolvida com o intuito de criar uma infraestrutura padrão e acessível para visão computacional e aprendizado de máquinas a usuários e programadores nas mais diversas áreas do conhecimento. É uma biblioteca de acesso livre, desenvolvida originalmente em C/C++, com interface nos mais diversos sistemas operacionais e linguagens de programação. [18]

Podem ser encontradas aplicações do OpenCV em pequenas startups e grandes empresas como a Google e Toyota [18]. Sendo também utilizada para situações em tempo real, apresentando mais de 2500 algoritmos otimizados para tratamento de imagens e aprendizado de máquinas.

4.3.2 Filtros

Um filtro é um componente capaz de reduzir distúrbio e/ou permitir apenas a passagem da informação relevante para a aplicação. Para tratar corretamente a imagem alguns filtros devem ser utilizados. “O primeiro passo para processamento de praticamente todo sistema de visão é a redução da quantidade de informação recebida” [10].

O OpenCV conta com várias aplicações para melhorar o entendimento da imagem, filtros para a leitura de contornos, *cannyedges* para detecção de vértices em bordas, a leitura do histograma da imagem para compreender a luminosidade, etc. Dentre os principais filtros utilizados neste trabalho para a localização de um contorno específico foram utilizados os filtros de *blur*, erosão, dilatação e *inrange*.

Os filtros de *blur* embaçam a imagem, apesar de não parecer produtivo a primeira vista, este filtro tem como principal função a de diminuir ruídos. Uma imagem após receber este tratamento se torna mais simples e a detecção de alguma forma ou cor se destaca ao longo do conjunto, se mostrando vantajoso mesmo com a perda de pequenos detalhes na imagem, normalmente estes que poderiam ser ignorados. As linhas do campo e as cores se tornam mais uniformes no espaço.

Os filtros de erosão e dilatação são opostos, como o próprio nome indica eles realizam um aumento ou diminuição dos pixels da estrutura. Desta forma é possível ajustar a imagem, separar corpos, aumentar pontos de referência que devem ser observados, como pode ser visto na Figura 13. Nota-se que o conjunto destes filtros ainda permitem completar um corpo e limpar ruídos.

Fonte: OpenCV Tutorial



Figura 13 – A esquerda imagem original, no centro atuação do filtro de dilatação aumentando as bordas dos pixels da imagem, e a direita a erosão de pixels.

Os filtros de *inrange* são utilizados para detecção de algum parâmetro único dentro da imagem. Procura por alguma cor, utilizando os valores de máximo e mínimo que pode atingir, desconsiderando todos os outros pontos que não correspondem ao intervalo. Pode ser comparado a um filtro passa faixa.

4.3.3 Cores

A biblioteca OpenCV apresenta uma gama de funções para detecção de objetos. É possível treinar um algoritmo para identificar padrões, um formato específico como o de um rosto ou um carro, filtrar contornos e cores. Para a aplicação em competição de futebol de robôs VSSS é sugerido o tratamento de cores.

A imagem em escala de cinza foi utilizada para o tratamento e identificação de formas e contornos. A escala de cinza representa a média dos valores de RGB (cores) da imagem, variando do preto, a ausência total de cor, ao branco, presença de todas as cores. A discrepância entre as mudanças de estado, sendo averiguadas com apenas uma variável e não três, facilitam a identificação dos contornos.

A visão humana e os demais algoritmos trabalham com o sistema de cores RGB, uma mistura de vermelho, verde e azul, que é capaz de alcançar todas as outras cores do espectro visível. O espectro RGB funciona como um cubo, Figura 14, em que cada combinação traz uma cor específica. Apesar de ser possível identificar corretamente as cores, é fácil perder o objeto neste espectro dependendo da variação da iluminação por exemplo.

Fonte: <http://en.academic.ru/dic.nsf/enwiki/157333>

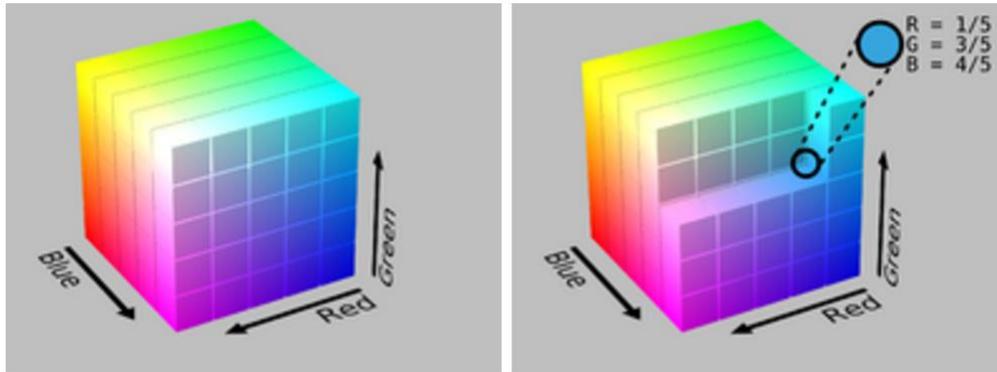


Figura 14 – Cubo representando espectro de cores RGB.

Para identificação de cores específicas é utilizado o espectro em HSV, Figura 15, dividido em: HUE (tonalidade cromática), um valor numérico que distingue cada cor, a saturação, que avaliando a pureza da cor e intensidade (*value*), o grau de luminosidade (associada ao brilho). Este espectro, ao contrário do RGB, consegue separar a cor em mais detalhes, resultando em uma robustez maior para o algoritmo de detecção.

Fonte: <http://en.academic.ru/dic.nsf/enwiki/157333>

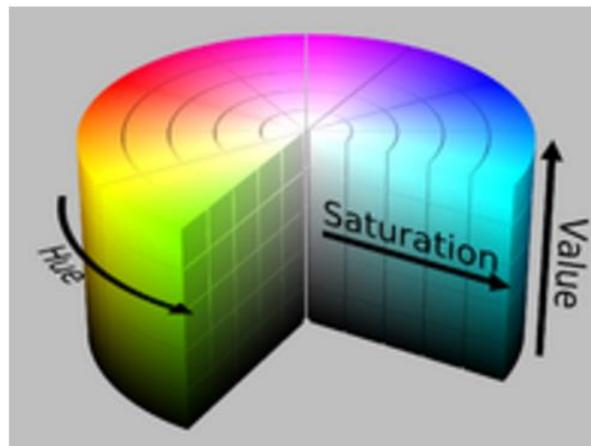


Figura 15 – Espectro de cores em HSV.

4.3.4 Correção de Distorção da Imagem

Para que a imagem seja obtida pelo computador é utilizada uma câmera firewire, BASLER SCA640-120FC, Figura 16, que utiliza o protocolo de comunicação IEEE 1394, com dimensões pequenas, sendo facilmente alocada e portando lentes Fujinom. Capaz de gerar uma imagem nítida em cores de 658x492 px (*pixels*) até uma velocidade de 120fps.

A imagem fornecida do sensor ao computador sofre distorção devido à curvatura da lente e entrada de luz. Portanto, para melhorar a atuação da visão computacional,

Fonte: <http://www.baslerweb.com/en/products/cameras/area-scan-cameras/scout/sca640-120fc>



Figura 16 – Camera firewire Utilizada.

modificar sutilmente estes parâmetros a fim de corrigí-los permitiria um ambiente mais estável e uniforme durante a análise. “A correção da distorção da imagem deve ser realizada devido as limitações do sistema ótico utilizado” [10].

Para realizar esta correção foram necessárias duas etapas. Para diminuição de brilho da imagem é realizado uma correção de *gamma*, junto com a calibração da abertura da lente para entrada de luz. Este filtro consegue minimizar reflexos e melhora a identificação de cores em alguns materiais parcialmente reflexivos.

Em seguida é realizado a correção da distorção intrínseca da lente da câmera, a distorção radial em que linhas retas apresentam uma certa curvatura é encontrada pelas equações 4.10 e 4.11. A distorção tangencial, caso a câmera não esteja corretamente alinhada, onde alguns pontos parecem estar mais próximos que realmente estão apresenta as equações 4.12 e 4.13 para identificação de seus parâmetros.

$$X_{radial} = x * (1 + K1 * r^2 + K2 * r^4 + K3 * r^6) \quad (4.10)$$

$$Y_{radial} = y * (1 + K1 * r^2 + K2 * r^4 + K3 * r^6) \quad (4.11)$$

$$X_{tangencial} = x + [2 * P1 * x * y + P2 * (r^2 + 2 * x^2)] \quad (4.12)$$

$$Y_{tangencial} = y + [P1 * (r^2 + 2 * y^2) + 2 * P1 * x * y] \quad (4.13)$$

$$\text{Fatores de Distorção} = \{K1, K2, K3, P1, P2\} \quad (4.14)$$

A correção de *gamma* é uma função simples para correção do brilho por meio de parâmetros internos de ganhos. Enquanto a correção da distorção envolve um processo de identificação de uma matriz que será utilizada em todo frame para corrigí-lo. Para

identificar esta matriz, um conjunto de imagens devem ser analisadas, apresentando diversas posições e angulações. É sugerido ao menos um conjunto de dez imagens.

A figura comum nessas imagens é uma malha quadriculada, como um tabuleiro de xadrez. O código identifica as linhas que deveriam ser retas e conserta a distorção radial enquanto ajusta a tangencial de acordo com a proximidade, levando em consideração que cada parte deve ter o mesmo tamanho. Dentro das informações da matriz consta as distâncias focais em x e y (f_x, f_y) da lente e onde seria o centro óptico (C_x, C_y).

$$\text{Matriz da Câmera} = \begin{vmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{vmatrix} \quad (4.15)$$

Os parâmetros que precisam ser identificados, vide Matriz 4.15 e Equação 4.14, são encontrados ao tratar o conjunto de imagens visto na Figura 17. A Figura 18 apresenta como a malha quadriculada é analisada, verificando seus vértices e se estão correspondendo ao comportamento que deveriam apresentar.

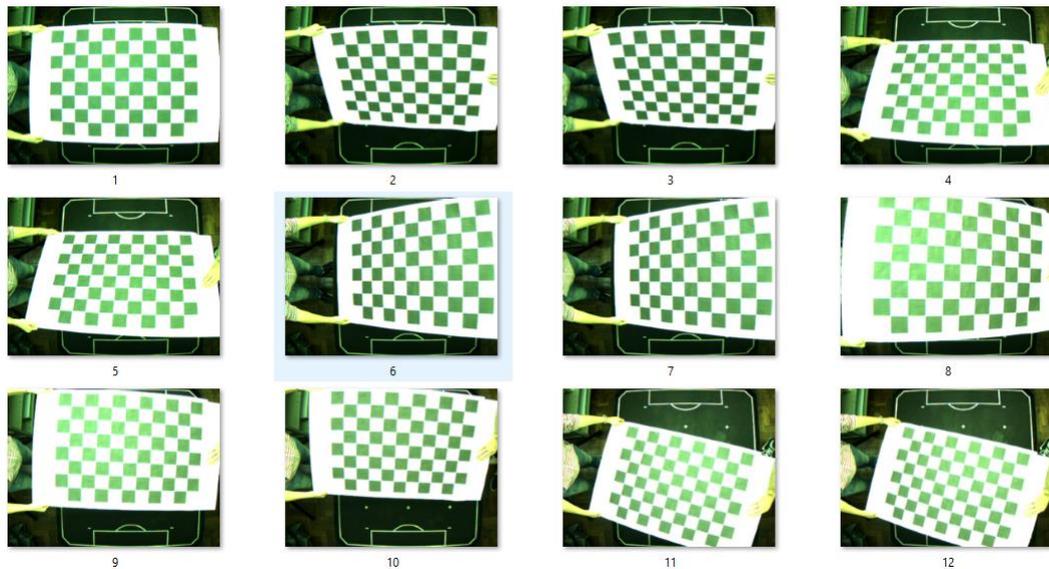


Figura 17 – Conjunto de imagens para análise de distorção.

Na Figura 19, é possível ver o resultado da correção de distorção do campo, a região de interesse do sistema de visão. Na Figura '19 a', a imagem sem tratamento apresenta uma curvatura nas linhas laterais, claramente um lado se encontra maior que outro apesar de serem simétricos e parte do ambiente externo é captado nas bordas da imagem. Após a correção, Figura '19 b', as dimensões se corrigem, não se percebe mais a captura de região externa ao campo e as linhas e proporções agora estão representando corretamente o ambiente real.

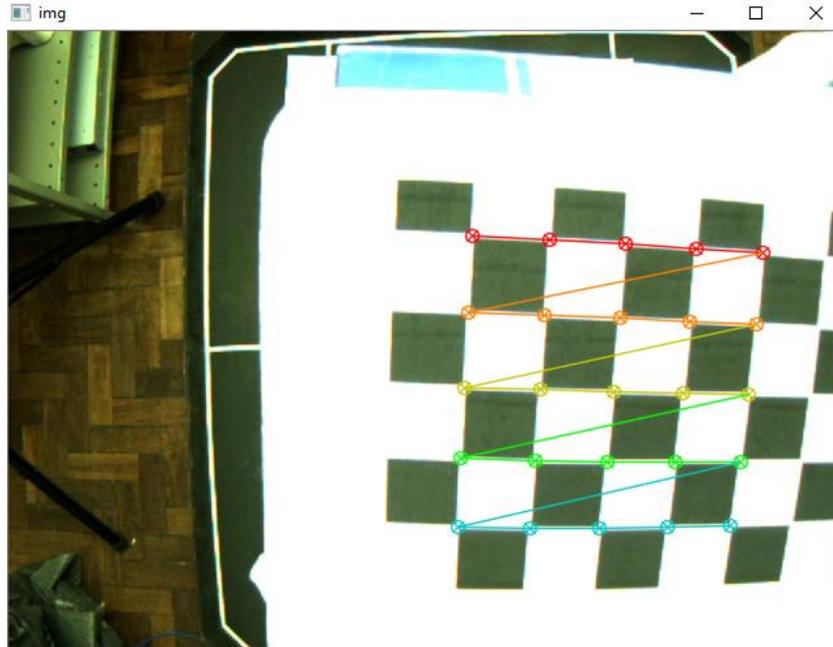
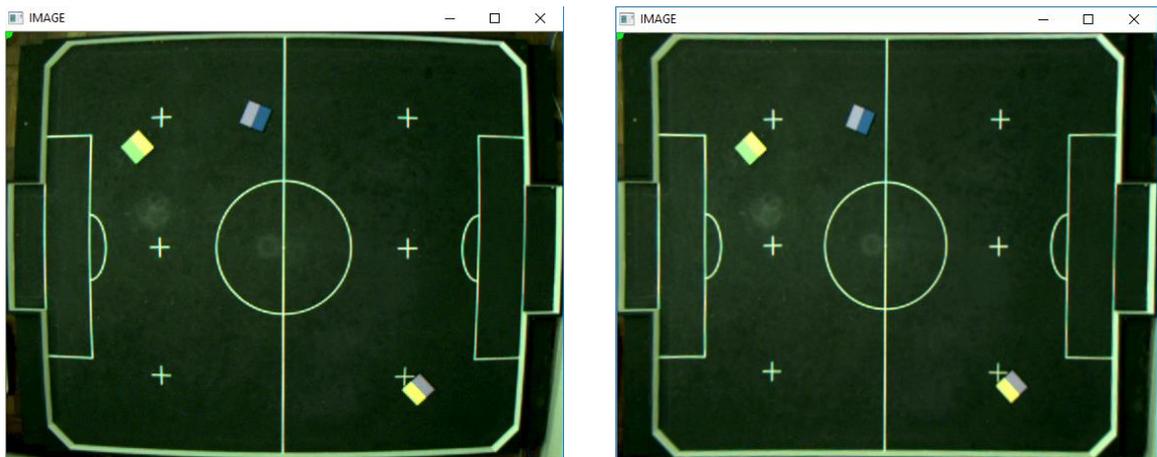


Figura 18 – Tratamento da malha quadriculada.



(a) Campo com distorção.

(b) Imagem pós correção.

Figura 19 – Resposta da correção de distorção da imagem na ROI representando o campo.

4.4 INTEGRAÇÃO DAS ETAPAS

Tendo em vista os elementos, o cenário de aplicação do robô móvel, o campo de futebol e suas laterais, os outros robôs tomados como obstáculos e o objetivo final a bola. Todos estes precisam ser identificados corretamente e suas informações armazenadas para que a ação se conclua como desejado.

O sistema completo possui duas etapas. A primeira etapa consiste na captação de algumas variáveis de controle e então armazená-las em um arquivo de apoio para serem utilizadas no programa principal. A segunda etapa apresenta a identificação dos objetos, fornecendo os parâmetros de localização e orientação no espaço e tratando-os de forma

que seja possível utilizar o algoritmo de campos potenciais para que o robô alcance seu objetivo.

O sistema é robusto e responde de acordo com a velocidade de captação de imagens da câmera, realizando a varredura da imagem *frame a frame*. Aproximadamente 30 iterações por segundo eram realizadas nos piores casos, velocidade superior a visão humana, em torno de 24fps. Logo, mesmo com o objetivo e obstáculos em movimento a resposta do robô demonstrou-se satisfatória. Assim como em [16] dado o posicionamento da câmera ser capaz de, a todo momento, identificar a posição e orientação dos objetos, é feita a captação dos resultados a todo instante, não ocorrendo erros cumulativos.

4.4.1 Captação de Informações

O primeiro algoritmo atua ainda de maneira manual, sendo sua principal função a captação de dados. Escrevendo estes dados em arquivos individuais .txt e .pckl, estes são lidos na primeira iteração do programa principal e armazenados em listas e dicionários para serem tratados corretamente.

A primeira informação é a matriz de distorção, como descrita anteriormente na Seção 4.3.4. Esta matriz corrige a distorção focal provocada pela lente da câmera e alinha a imagem, retirando as bordas arredondadas. Em contrapartida é uma iteração a mais por frame, logo deixa a execução mais lenta.

É importante frisar que apesar de um passo importante, e melhorar a qualidade da imagem e a precisão da captação dos próximos filtros dentro do universo real, este passo não é necessário para a aplicação final, podendo atuar sem a execução do mesmo. Fato é que para os robôs, para a bola e obstáculos o universo de atuação é o mesmo. E mesmo que haja distorção, estará presente para todos, logo se orientam e direcionam corretamente baseado neste universo, alcançando o mesmo resultado.

A imagem para o usuário é vista no padrão de cor RGB, mas toda a verificação é realizada em dois conceitos distintos, a identificação por cor é realizada na imagem em HSV e a detecção de bordas em escala de cinza.

Em Ambiente de Aplicação, Capítulo 2, foi visto as diretrizes que poderiam ser tomadas para o universo de trabalho, direcionando então o estudo para a aplicação específica do futebol de robôs.

As bordas do campo são identificadas pelo maior contorno de algum objeto claro dentro do universo da imagem em escala de cinza. Apesar de simples, essa abordagem se mostra rápida e eficaz. Todas as bordas do campo são brancas e com o fundo do campo e a parte externa mais escuros, o filtro consegue registrar corretamente as extremidades do campo.

Tendo as bordas, o interessante é reduzir a imagem a uma ROI (região de interesse),

desta forma toda a análise a partir deste ponto se concentra dentro do campo e mesmo que apareça algum objeto que possa ser confundido com os de interesse da imagem na extremidade da visão da câmera é descartado por não estar na região de análise.

Sendo realizada a identificação do campo e tendo suas medidas, é feita uma análise espacial dado a quantidade de pixels que possui esta nova ROI. Isto é, um valor que será utilizado para converter pixel em milímetros para comparação espacial do usuário.

A última informação relevante que deve ser passada para o arquivo .txt são as identificações do objetivo, atuador e obstáculos. Dentre as possibilidades de trabalhar em identificação de contornos, área e cor, a cor foi a que demonstrou o melhor resultado em curto prazo, sendo a mais fácil de implementar.

A identificação ocorre em segundo plano com a imagem sendo analisada em HSV. Após selecionar na imagem uma região que apresenta a coloração desejada é feita uma verificação de todos os pixels para encontrar o maior e menor conjunto de parâmetros HSV possíveis. Esta análise ocorre durante cem iterações, com frames distintos para verificar alguma variação de luminosidade que possa interferir no resultado, sendo verificado tanto os valores para hue (cor), saturação e valor (luminosidade). Utilizar os três parâmetros, permite aplicar um filtro mais completo, apesar de mais pesado computacionalmente.

Este método foi acurado na identificação das cores, permitindo mesmo tonalidades próximas dentro do espectro serem diferenciadas corretamente. Um problema é que a iluminação não necessariamente é constante ao longo do campo e portanto estes parâmetros fixos encontram problemas em alguns casos. Para solucionar este efeito é aumentado da resposta padrão de máximo e mínimo um valor extra no qual é possível variar sem se misturar com outras cores.

4.4.2 Integração

Tendo os dados de análise e identificação coletados, no programa principal eles são lidos do arquivo .txt em que foram salvos e armazenados em listas correspondentes para serem utilizados ao longo do algoritmo. A principal vantagem de salvar em um arquivo de parâmetros é não ser necessário modificar o código para alterar as variáveis e se manterem salvas para quantas vezes for necessário rodar a aplicação, sendo fácil modificá-las caso preciso.

Ao realizar a leitura das variáveis o sistema deve ser capaz de reconhecer a região de interesse da imagem e os limites para identificação das cores, fornecendo a cada objeto as informações que serão utilizadas para que o robô possa se locomover no espaço, a posição e orientação em relação à origem. Após identificados, estes valores são tratados e o robô recebe uma ordem para onde deve se movimentar com base nos resultados dos campos potenciais, enviando o protocolo de comunicação.

Pseudo Código – Sistema Completo

1. Leitura e armazenamento de parâmetros do arquivo .txt;
2. Ativar a câmera;
3. Enquanto estiver realizando leitura de imagem da câmera:
 - 3.1. Filtragem da ROI (Campo), cópia para análise e para visualização na tela;
 - 3.2. Se necessário ajuste da imagem;
 - 3.3. Identificação por cor, esfera, ponto objetivo;
 - 3.4. Identificação dos demais objetos:
 - 3.4.1. Robô de Controle:
 - 3.4.1.1. Análise da primeira cor padrão (localização);
 - 3.4.1.2. Análise da segunda cor, dentro de um range (orientação);
 - 3.4.2. Robôs Obstáculos:
 - 3.4.2.1. Análise da cor padrão;
 - 3.4.2.2. Análise de contorno sobre o fundo;
 - 3.5. Encontrar resposta dos campos potenciais para robô se movimentar (V , W);
 - 3.6. Enviar os dados ao robô;

Tendo lido as variáveis, o algoritmo ajusta para a visão do usuário a região de interesse identificada como o campo, já tendo tratado a imagem com a correção caso necessário e permite a visualização da mesma. Simultaneamente realiza uma cópia desta imagem de leitura em HSV e escala de cinza para realizar os próximos passos.

A identificação de cor é padrão para todos os objetos da imagem, assim todas as cores são identificadas da mesma forma. Todavia, para evitar erros na leitura cada objeto sofre verificações secundárias para minimizar ruídos e garantir a aferição correta do mesmo. Um destes processos é a identificação de área, uma área mínima e se necessário uma área máxima na qual deve estar contido o contorno da cor especificada. Ainda dentro deste processo é visto quantos objetos devem ser procurados para trazer todos os pontos de interesse. Em cada iteração, por segurança, toda a imagem é analisada.

Sendo assim, a esfera, que é o ponto objetivo é identificada a partir de sua cor característica e uma área mínima, sendo o maior objeto que corresponde a estes padrões. A imagem é reavaliada por completo a cada frame e a posição atualizada e armazenada. A Figura 20 demonstra o esboço do que ocorre durante a execução da função de identificação de cor. Mesmo havendo vários objetos similares ou ruído, o filtro toma o maior contorno como o resultado correto e ignora todos os outros possíveis, o que obriga que a primeira

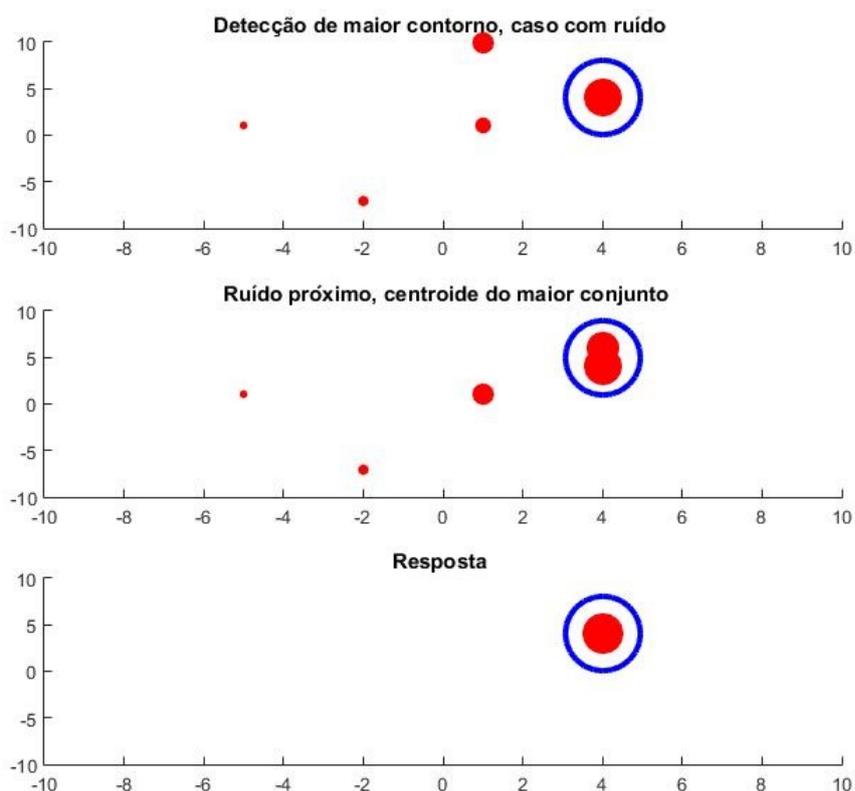


Figura 20 – Esboço da detecção de contorno.

etapa seja robusta e que o ambiente, apesar do dinamismo, se mantenha dentro do previsto para a competição.

Pseudo Código – Identificação de Cor

1. Defina área mínima, quantos contornos e mínimo e máximo das variáveis de cor (HSV);
2. Converter imagem para HSV;
3. Filtros de ruído → `cv2.GaussianBlur(image, blur,0)`;
4. Detectar a cor → `cv2.inRange(mask, lower, upper)`);
5. Filtros de ajuste → `erode, dilate`
6. Encontrar os contornos da cor → `(cv2.findContours)`
7. Se algum contorno encontrado:
 - 7.1. Ajuste em ordem decrescente;
 - 7.2. Para os N primeiros: sendo N o número de contornos esperado definido;

7.3. Análise da área;

7.3.1. Se área > área mínima:

7.3.1.1. Encontrar o centroide do contorno (moments);

7.3.2. Senão:

7.3.2.1. Descartar;

8. Retornar os centróides da cor e a máscara com o filtro do contorno.

Para os robôs obstáculos e de controle é preciso uma verificação secundária diferente para cada um destes. No caso dos robôs obstáculos sabe-se que apresentam uma cor característica, mas não necessariamente cobre toda sua superfície, podendo ser preenchida por uma segunda coloração ou apenas disposta em um formato distinto.

Para estes pontos é identificado a cor padrão e associado o número correspondente de obstáculos encontrados. Mas para saber qual área realmente ocupam e o atuador possa desviar corretamente é feito um filtro na imagem em cinza. Dado o ponto central da cor padrão é estendida uma região de interesse capaz de acomodar o objeto inteiro, Figura 21. Sendo o fundo preto, utiliza-se um filtro de intensidade em escala de cinza para ler qualquer contorno nesta região, acomodando então o obstáculo como o maior contorno encontrado, tendo então um quadrado e seus quatro vértices correspondentes. Quanto as linhas brancas que podem se juntar ao contorno, elas são retiradas da imagem por meio de um filtro de erosão e um filtro de embaçamento (blur).

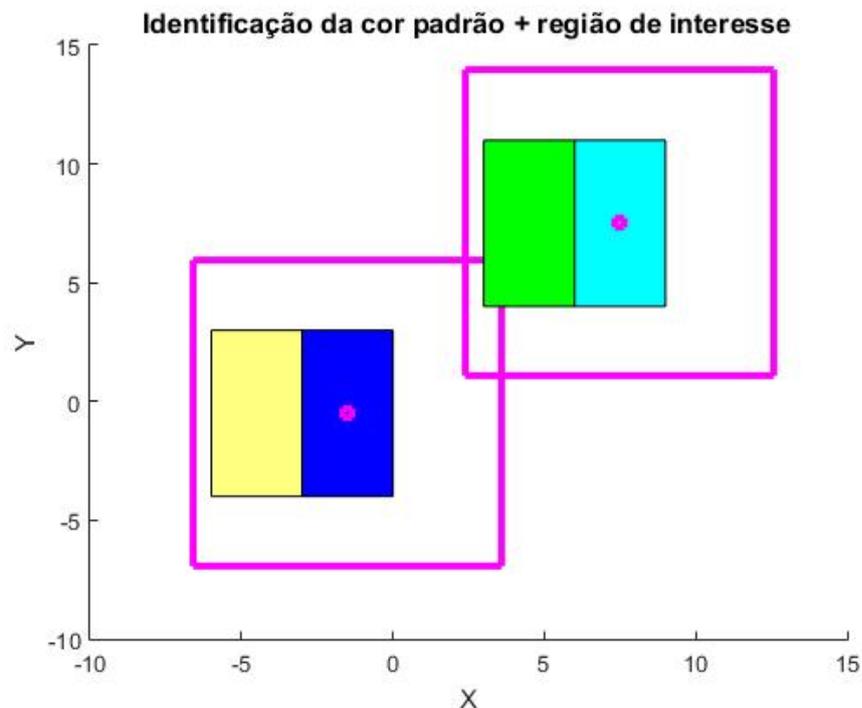


Figura 21 – Região de interesse do robô a partir da cor padrão.

Para o robô diferencial, além da posição, deve ser identificada também a orientação do mesmo. Composto por dois retângulos de cores distintas em sua superfície, Figura 21, é identificado uma cor padrão na parte de trás e uma segunda cor na parte frontal, uma reta é traçada entre os centros para obter a orientação. A localização depende do ponto central desta reta sendo também averiguada a proximidade com o contorno do robô assim como o obstáculo.

Em uma situação específica em que mais de um robô e ou obstáculos se encontram muito próximos, é possível confundir o desenho das cores e a qual realmente pertence. Logo, para isso, a segunda cor do robô de controle só pode ser identificada dentro de um entorno/alcance (*range*) limitado, próximo a cor de identificação padrão. Isto também impede que, caso exista esta cor repetida no cenário, não seja confundida e prejudique a orientação e localização, Figura 21.

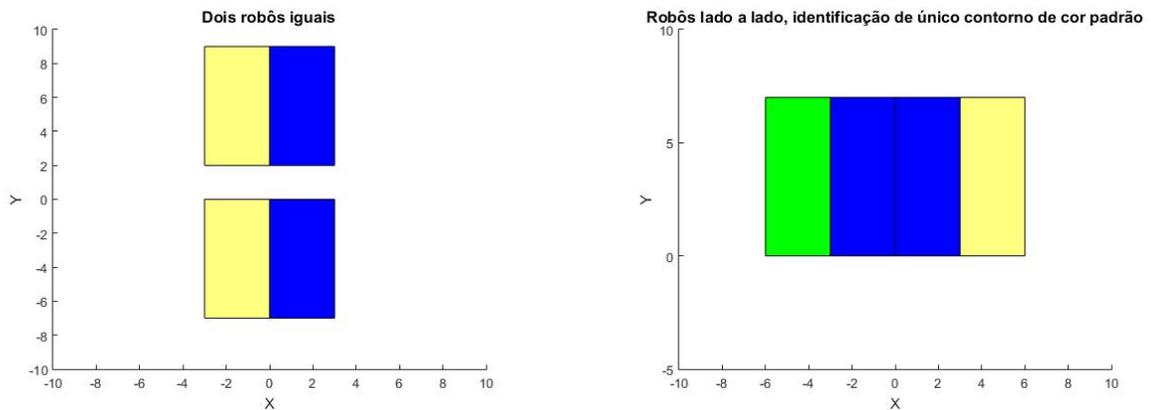
Pseudo Código – Identificação de Objetos

1. Define área mínima, quantos contornos e mínimo e máximo das variáveis de cor (HSV);
2. Para cada objeto:
 - 2.1. Encontra a cor padrão;
 - 2.2. Em um pequeno entorno(*range*) a partir do centro, realizar teste de contorno, para identificar o robô;
 - 2.3. Se obstáculo:
 - 2.3.1. Adquirir centro da caixa de contorno;
 - 2.4. Se objeto de controle:
 - 2.4.1. Procura a segunda cor dentro da região de interesse;
 - 2.4.2. Identifica centro;
 - 2.4.3. Realiza linha entre centros e obtém orientação;

Quando ocorre de interferência na qual não é possível identificar algum objeto previamente identificado ou não se tem certeza sobre alguma informação importante, o dado é descartado e mantém o último valor do frame no qual foi averiguado até que possa ser localizado mais uma vez corretamente.

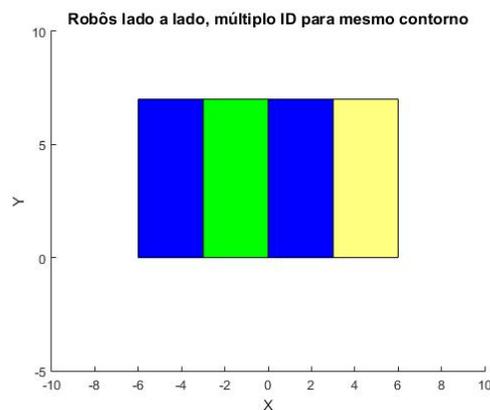
Na Figura 22, segue o caso 1 (a), no qual o mesmo objeto é visto duas vezes, logo será descartado, não identificado. O Caso 2 (b), traz uma área de cor padrão, azul, maior, que pode ser confundido com um único contorno. Quando este fato ocorre e duas identificações secundárias aparecem para o mesmo contorno o valor é descartado. No caso 3 (c), observa-se que o sistema não saberá qual das duas cores pertence ao robô mais a

direita do plano, logo, o sistema identificará o robô como dois objetos e irá descartar esta medida. Com base na velocidade de captação de dados do sistema, imaginando que não tenha havido muito deslocamento por parte dos objetos, robôs, entre um frame e outro da imagem, a medida então não possui seu valor atualizado.



(a) Múltiplos robôs iguais.

(b) Cor padrão em contato.



(c) Robôs lado a lado, múltipla identificação.

Figura 22 – Situações de identificação comprometida.

Em posse de todas estas informações, posição dos objetos e orientação do robô de controle, é possível calcular todos os parâmetros para que os campos potenciais possam atuar. A resposta deste método será a velocidade linear e angular do robô, sendo a câmera a única realimentação do sistema em malha fechada. Para se locomover corretamente os parâmetros de atração e repulsão, tal qual o horizonte de eventos (ϵ), devem ser bem modelados para permitir uma boa resposta. Foram coletados de forma empírica para cada conjunto de motores, baseado em análise visual e uma gama de testes junto ao “log” de eventos para verificar qual seria uma resposta aceitável para a aplicação em cenário de competição.

Pseudo Código – Campos Potenciais

1. Adquire posição e orientação do robô, objeto e obstáculos;

2. Declara constantes de atração, repulsão, horizonte de eventos (ϵ);
3. Enquanto distância robô até objetivo $>$ erro aceitável;

3.1. Realizar cálculo de atração:

$$Fatt = Katt * [(Pos.robô) - (Pos.objetivo)]$$

3.2. Se nenhum Obstáculo (obstáculos, tudo que não é o controle ou objetivo, paredes inclusive):

3.2.1. Força repulsiva inexistente

$$Frep = [0, 0]$$

3.3. Se número de obstáculos $>$ 0:

3.3.1. Para cada obstáculo

$$D = (Pos.robô) - (Pos.obstáculo)$$

$$Frep = Frep + Krep \cdot \frac{1}{D^3} \cdot \left(\frac{1}{\epsilon} - \frac{1}{D}\right) \cdot [(Pos.robô) - (Pos.objetivo)]$$

3.4. Calcula-se a força total orientada pra onde deve ser realizado o movimento:

$$Ftot = Fatt + Frep$$

3.5. Converte este resultado em Velocidade Linear e Angular (V, W)

$$V = \min(\|Ftot\|, v_{max})$$

$$W = Kw * atan2(Ftot_Y, Ftot_X) - \text{Orientação Robo}$$

3.6. Robô atua com V e W.

5 RESULTADOS

Este capítulo apresenta os resultados de diversas situações de jogo, ponderando as considerações vistas nos capítulos anteriores de como identificar os diferentes tipos de objetos, demonstrando o comportamento do algoritmo e respondendo com relativa acurácia.

O sistema de controle do robô móvel, possui apenas odometria visual externa para identificação de sua localização. A imagem proveniente desta câmera é lida por completa a cada iteração, sendo realizado os filtros para identificação de objetos e definição de regiões de interesse do ambiente.

5.1 IDENTIFICAÇÃO DE PARÂMETROS E LOCALIZAÇÃO

Seguindo a lógica descrita na seção 4, inicia-se a leitura da imagem como um todo e adquire da mesma a região de interesse correspondente apenas ao campo, esta etapa é descrita na subseção 5.1.1 Tendo realizado esta primeira detecção, pode-se ignorar toda a região externa captada pela câmera e concentrar os esforços em uma área menor, sabendo que todos os objetos devem estar contidos nesta região.

A partir deste ponto inicia-se o filtro de detecção de cores específicas e pré determinadas, como explicado na subseção 5.1.2. Para então identificar o objetivo e os robôs, tratando quaisquer erros que viessem a acontecer, subseção 5.1.3.

5.1.1 Laterais do Campo

Detecta-se as laterais do campo dentro de toda a imagem da câmera. A Figura 23 apresenta a identificação da margem do campo resultando na região de interesse, a qual todos os objetos relevantes devem estar contidos. Um conjunto de pontos para essa margem é obtido, podendo ser utilizados para traçar as retas que contemplam as linhas do campo.



Figura 23 – Identificação de margem.

Do campo observado na imagem, Figura 23, é possível identificar o centro e as áreas para ataque e defesa. Independente de qual lado a estratégia esteja voltada o ponto (0,0) do campo se mantém o mesmo, a esquerda a baixo da região de interesse que ocupa o campo. Apresentando ao todo 1700 mm de comprimento no eixo X e 1300 mm de largura no eixo Y.

5.1.2 Filtro de Cor

Após obter os resultados dos parâmetros correspondentes a determinada cor de acordo com a escolha do usuário, pode-se realizar a análise de onde ela se encontra no espectro da região em HSV. Observa-se que mesmo cores próximas em RGB podem ser distintas se bem parametrizadas e que ruídos não interferem na localização do ponto objetivo. Mesmo se o corpo se mostrar parcialmente encoberto, o filtro não demonstrou problemas para identificação, contanto que mantenha a área mínima para ser desconsiderado como ruído.

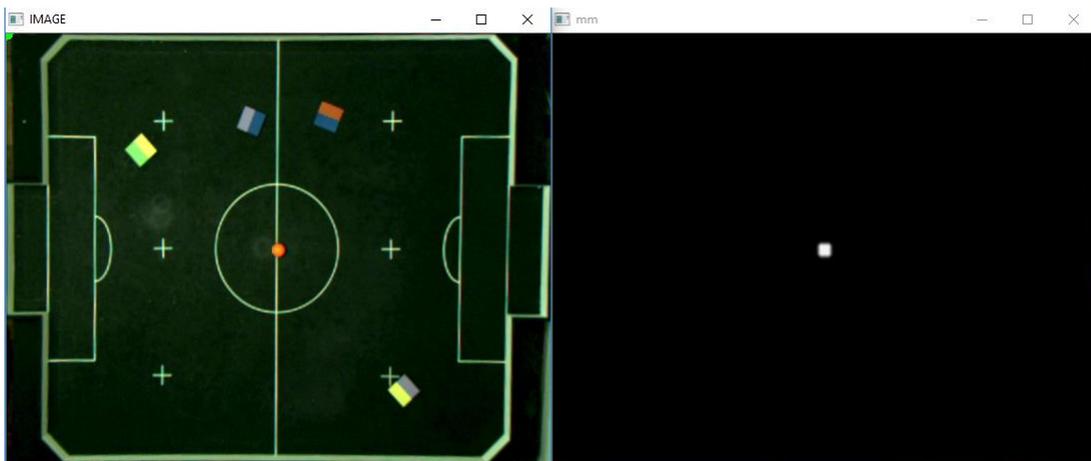


Figura 24 – Localização esfera objetivo.

A Figura 24 traz o resultado da identificação do ponto objetivo, filtrando apenas a bola laranja do jogo de futebol de robôs. Na imagem é possível verificar além da esfera alaranjada um robô que possui uma de suas cores características o vermelho, cores próximas em sua matiz e no espectro RGB. O filtro atuou corretamente se valendo também dos valores de intensidade luminosa e pureza do HSV para realizar esta diferenciação.

Na Figura 25 observa-se a captação de múltiplos objetos de determinada cor padrão, no caso o amarelo, em diferentes pontos de luminosidade do campo, ainda assim obtendo bom resultado. Em ambas as imagens citadas, Figuras 24 e 25, observa-se a máscara que traz apenas o resultado do filtro, excluindo os ruídos e todos os outros corpos que não correspondem aos parâmetros do mesmo.

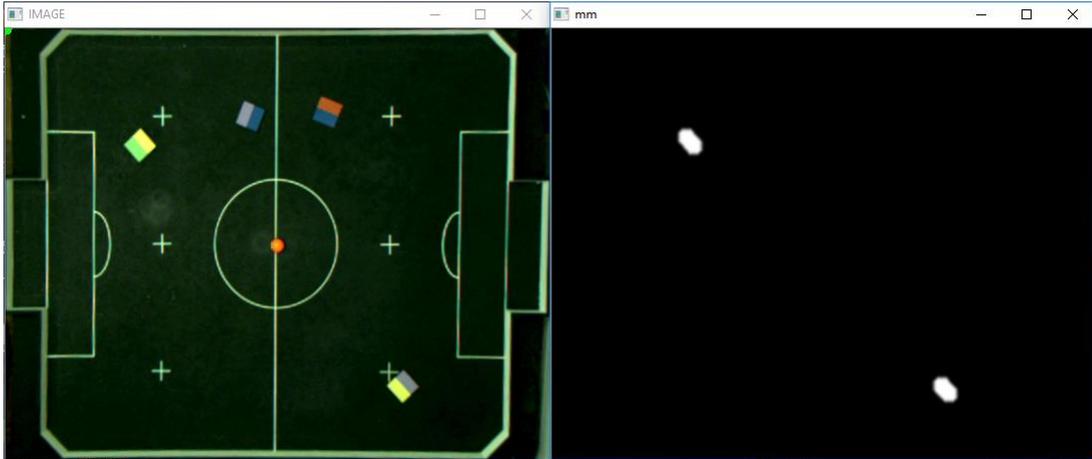


Figura 25 – Filtro de cor padrão, amarelo.

5.1.3 Identificação dos Robôs

Divide-se para o jogo de futebol de robôs dois times, um time portando a cor amarela e outro a coloração azul. No caso dos robôs adversários apenas a cor padrão e seu contorno devem ser identificados. Para o robô do time de controle a orientação também é um parâmetro relevante e é identificado com a segunda cor de superfície, esta cor que também representa a identificação (ID), do robô.

Observando a Figura 26 os robôs são vistos com sua cor característica e um retângulo externo no qual o alcance foi aumentado para verificar o contorno. Esta região de interesse demarcada na imagem por um retângulo magenta, se espalha a partir da cor padrão do objeto, e dentro desta pequena ROI formada todo o robô é englobado. Logo, apenas nessa região é feita a busca da segunda cor para detecção de contorno, orientação e ID.



Figura 26 – Abertura da região de interesse para localização de contorno robô.

Na Figura 27 é possível visualizar o resultado final, com cada objeto sendo encontrado corretamente e sua posição demarcada no mapa em milímetros. Os robôs de cor

amarela, estão sendo considerados como pertencentes ao time adversário. Apresentam apenas a posição representada, não seria possível em um ambiente de competição saber qual tipo de estratégia o adversário utilizaria para a superfície de seus robôs. Já os robôs de controle, cor padrão azul, apresentam além da posição o ID do robô e a orientação.



Figura 27 – Resultado final da localização.

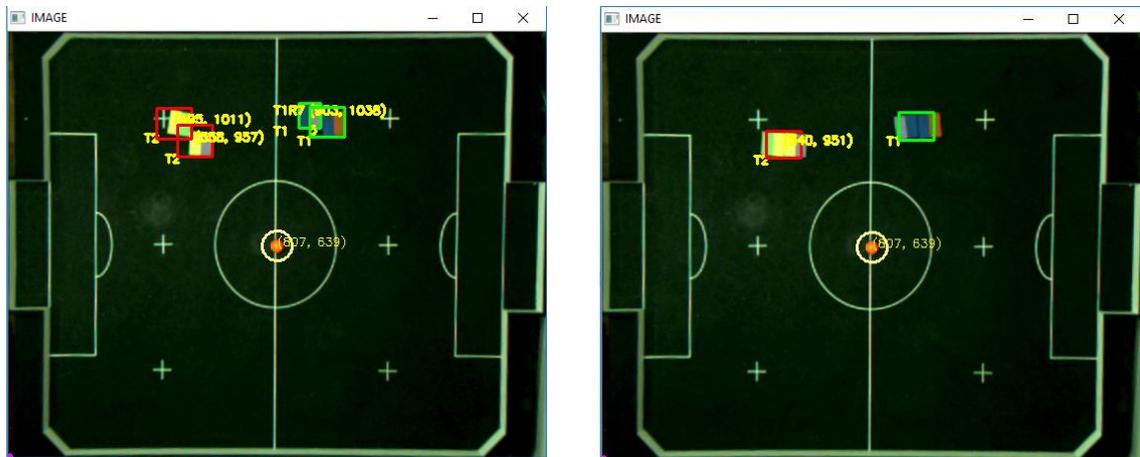
Ainda na Figura 27, a bola marcada por um círculo amarelado e sua respectiva posição para o usuário. Os robôs de cor amarela, time adversário, demarcados por retângulos vermelhos correspondentes a seu contorno e imprimindo sua posição na imagem. Os robôs do time de controle, cor padrão azul para este caso, demarcados por retângulos verdes. Estes robôs apresentam além da posição em milímetros que ocupam no ambiente o respectivo ID que o representa e a angulação em graus em que se situa no plano.

5.1.4 Correção de erros

Como pode ser visto na Figura 28, situações complexas de identificação são apresentadas. Quando objetos próximos e cores distintas em contato, Figura 28 c, o filtro de erosão consegue contornar este caso e permite a identificação correta dos robôs. Apesar que o contorno geral possa parecer que os robôs estão se sobrepondo, algo fisicamente impossível. Pode ocorrer casos em que um obstáculo muito maior foi identificado dado a proximidade com outro objeto, Figura 28 a, ainda assim será visto a ser desviado corretamente, apenas como um obstáculo maior.

Na Figura 28 b, dado a proximidade da mesma cor, um conjunto de área maior foi encontrado, sendo a correta identificação. No caso dos robôs de cor azul, que utilizam também a segunda cor, por acreditar que apenas um robô se encontra naquele intervalo e duas cores de orientação foram encontradas, o resultado é descartado. A última validação

de resultado entre eles é tomada como a verdadeira, mesmo que retroativa, portanto não é visto na imagem os valores de posição.



(a) Proximidade, cor de identificação.

(b) Cor padrão em contato.



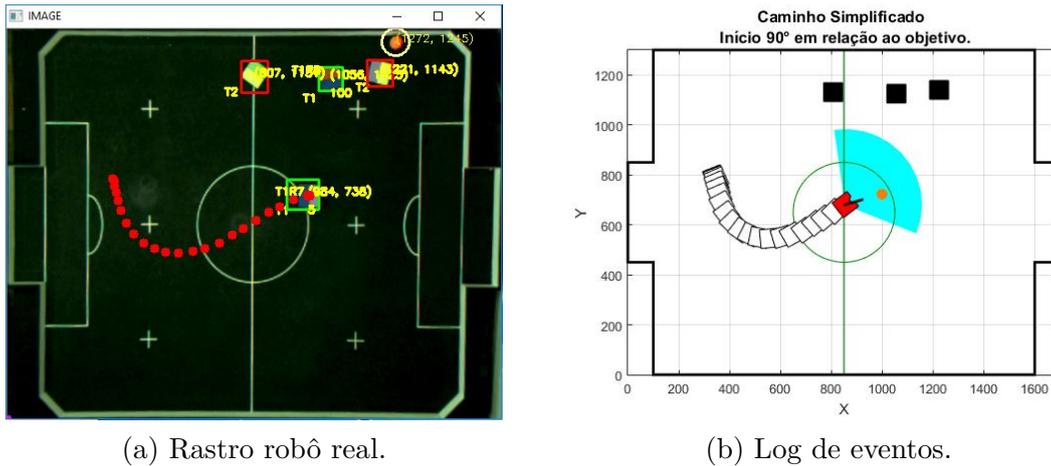
(c) Proximidade cor padrão.

Figura 28 – Situação de erro, frame único.

5.2 RESPOSTAS DO ROBÔ EM MOVIMENTO

Para averiguar como está sendo realizado a dinâmica de forças calculadas do campos potenciais pelo sistema, foi desenvolvido um *log* de variáveis. Salvas em um arquivo “.m” para ser lido no MATLAB [®], realizando a análise frame a frame visualmente e numericamente podendo pausar ou avançar como necessário. A Figura 29 demonstra lado a lado o deslocamento no ambiente real, Figura 29 a, e as variáveis quando desenhadas no ambiente de análise, Figura 29 b.

A Figura 29 b, apresenta a imagem do *log* de variáveis, dentro do cenário do campo e suas dimensões. Em vermelho apresenta a posição atual do robô. É possível ver em azul claro parte do alcance do universo de eventos, apesar de ser uma esfera, é desenhado apenas a parte frontal para aumentar a sensação do que é a frente do robô. Os quadrados sem preenchimento representam por onde o robô passou, enquanto os quadrados pretos são



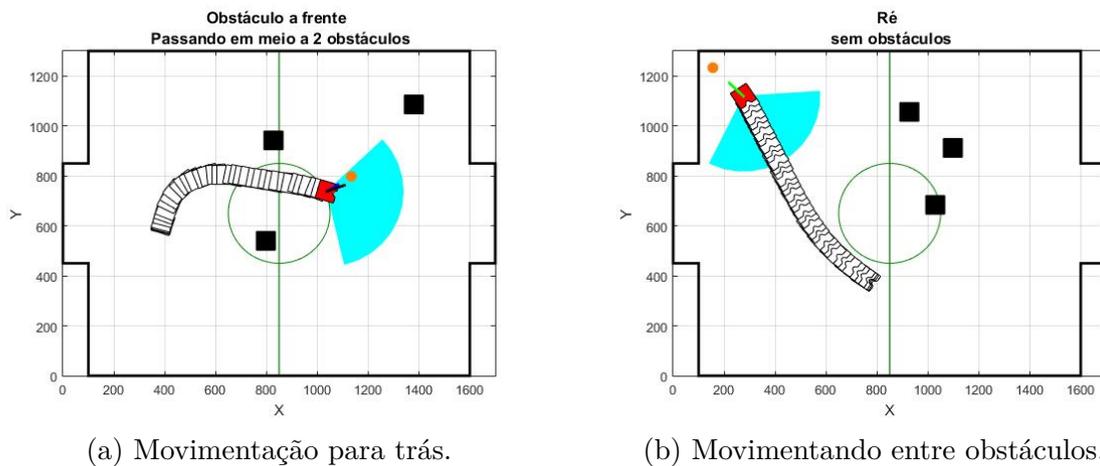
(a) Rastro robô real.

(b) Log de eventos.

Figura 29 – Log de Eventos, seguindo uma trajetória.

os obstáculos. Vale notar que os obstáculo incluem as paredes, os robôs do time adversário e também os robôs do mesmo time, com exceção do que está sendo controlado. O *log* permite visualizar a interação das forças de repulsão, atração e resultante, vermelha, azul e preta respectivamente.

A cinemática bem aplicada resulta em situações simples na qual o robô segue a esfera objetivo, desviando de seus obstáculos no caminho, podendo inclusive andar e seguir se movendo de ré (para trás) caso indique que essa seria a melhor opção. Na Figura 30 por exemplo, observa-se duas situações distintas, Figura 30 a, o robô se movimentando para trás, onde a força de atração indicava seu objetivo. E a Figura 30 b, indo em direção ao objetivo se deslocando corretamente entre os obstáculos.



(a) Movimentação para trás.

(b) Movimentando entre obstáculos.

Figura 30 – Log de eventos, análise movimentação do robô móvel

A Figura 31 mostra um caso em que o ponto objetivo está próximo da parede e obstáculos impedem um deslocamento direto. Percebe-se os vetores das forças de repulsão, atração e resultante ao longo do trajeto do robô. Quando próximo de um obstáculo na primeira imagem é redirecionado para a esquerda, porém na segunda imagem muito

próximo da parede é deslocado com muita velocidade para longe para impedir a colisão e garantir a integridade física do robô.

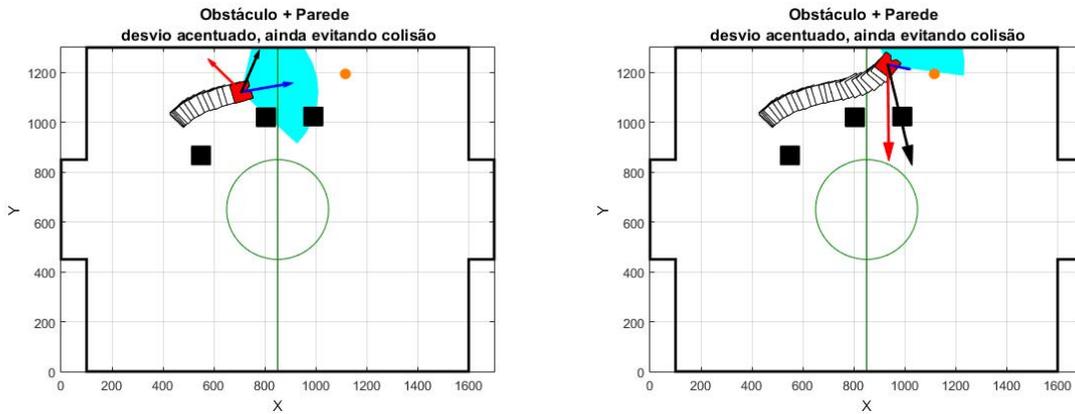


Figura 31 – Log de eventos, obstáculos mais parede.

Para a resposta da Figura 31, realizou a função proposta e manteve a integridade do robô durante todo o trajeto. O algoritmo de campos potenciais atua não só como planejador de caminhos mas também como controlador, mesmo caso ocorresse um atraso de resposta ou alguma falha mecânica nas rodas, a atuação a cada iteração da imagem redirecionaria o robô ao objetivo.

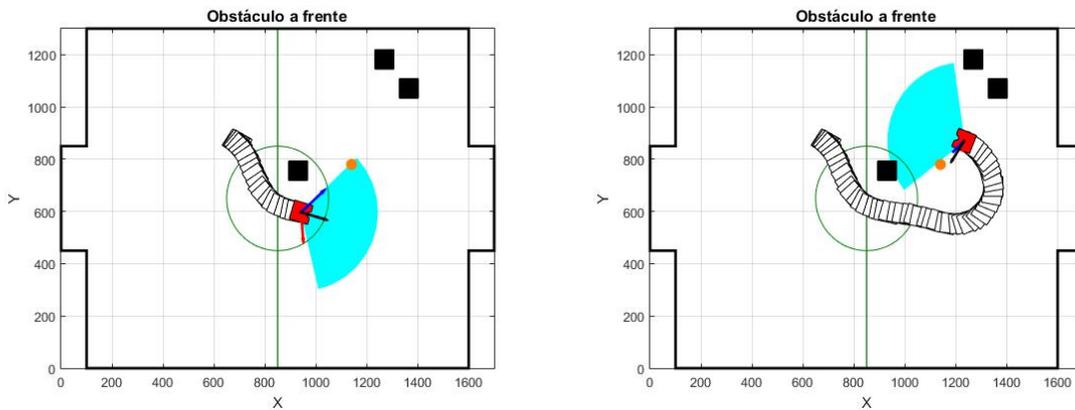


Figura 32 – Obstáculo a frente.

Pode ser observado na Figura 32, com obstáculo a frente, um desvio bem efetuado por parte do robô de controle. Com a ação de ambas as forças atuando, e o objetivo dentro de uma área em que o robô ainda sofre repulsão, uma volta maior foi realizada até alcançar o objetivo. Este é um exemplo da importância de escolher bem os parâmetros dentro do cenário de atuação.

Finalmente na Figura 33 observa-se a principal vantagem do dinamismo do campos potenciais para a aplicação, a cada iteração sendo recalculado, quando o objeto é exposto a um impulso e inicia um movimento o robô vai atualizando sua resposta para alcançá-lo.

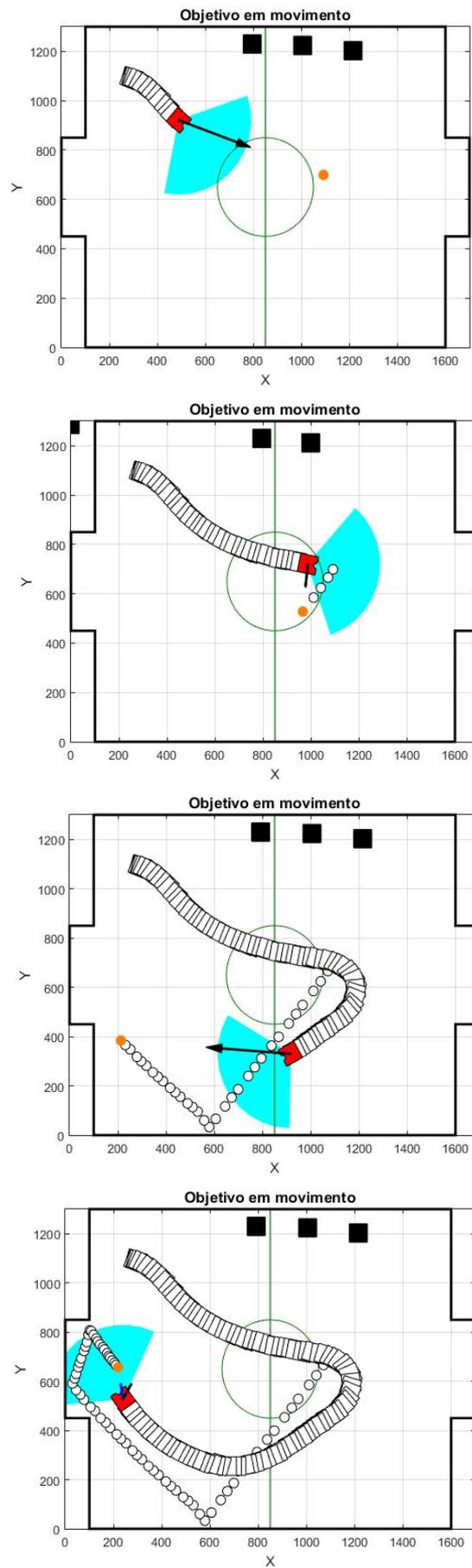


Figura 33 – Esfera objetivo sofrendo impulso e se movendo no universo de controle.

6 CONCLUSÃO

O trabalho teve como objetivos: a construção de um sistema de visão computacional capaz de identificar um robô móvel no espaço, e a implementação da metodologia de campos potenciais para fazê-lo se movimentar sem colidir com outros objetos. Desenvolvido em um ambiente controlado para um cenário de competição de futebol de robôs.

Foi apresentado, no Capítulo 2, as considerações tomadas para tratar o problema de futebol de robôs em ambiente controlado. Desta forma pode-se ter uma noção ampla dos principais focos do estudo: identificar o campo, os robôs e evitar colisão enquanto se movem. Ao mesmo passo que pode-se descartar aspectos mais complexos como corpos desconhecidos ou superfície não planar. Ainda referente ao ambiente de trabalho, no Capítulo 3, foi apresentado como alguns autores abordaram os problemas, e justificando o uso e não uso de algumas das técnicas empregadas.

A partir do Capítulo 4, são expostas as ferramentas utilizadas ao longo do trabalho. A cinemática do robô diferencial, simplificada e sem considerar atrito, atraso, inércia foi o suficiente para obter um bom resultado. O método de campos potenciais para planejamento de caminhos, empregado com eficiência, algoritmo este que também atua como controlador e no dinamismo do ambiente alcançou o objetivo mantendo a integridade física do robô quando bem parametrizado.

Ainda sobre ferramentas, o OpenCV, biblioteca de software aberto para tratamento de imagens, foi de grande valia para o trabalho. A facilidade em se executar as funções, com o auxílio da linguagem de programação Python, permitiu caminhar do baixo nível, realizando os filtros e identificando corpos, ao alto nível, ao realizar o controle de um destes robôs em tempo real.

Dividido em etapas de captação e tratamento de dados, havendo distintas formas de realizá-las, a identificação por cor dentro do espectro HSV e escala de cinza para detecção de bordas foi aplicada fornecendo bons resultados. Apresentando rápida execução e com uma boa acurácia o método se mostrou proveitoso para a aplicação na qual foi desenvolvido. Tendo o universo de trabalho sido aplicado para o futebol de robôs, mais especificamente a categoria VSSS (Very Small Size Soccer) e o algoritmo utilizado pela equipe Rinobot da UFJF pela primeira vez na LARC / CBR (Competição Latino Americana e Brasileira de Robótica) em Recife 2016.

6.1 TRABALHOS FUTUROS

O processo como um todo ainda pode e deve ser melhorado. Principalmente na parte inicial de captação de informações em que ainda está consideravelmente manual, não tendo sido o foco deste trabalho, automatizar esta primeira etapa e focar na identificação

de cor em vários pontos do campo para impedir que possa haver alguma alteração na luminosidade que faça a câmera perder o objeto. Quanto à execução do algoritmo, este ainda pode ser alocado em multitarefas (*threads*) para acelerar a execução do mesmo. Pode ser utilizado um método mais robusto para o deslocamento e não mais tratar os objetos como pontos, baseando em área total de ocupação.

Apesar da motivação, voltada para aplicação em competições de robótica, e o aprendizado que isso constrói como um todo, a visão computacional associada a robôs pode ser utilizada em diversos ambientes. Utilizar uma lógica similar de planejamento de caminho e análise de ambiente permitiria a automatização de diversos processos, sendo o OpenCV um software aberto com bibliotecas prontas de identificação de forma, etc, pode ser utilizado de várias maneiras associando outras ferramentas como um treinamento de aprendizado de máquinas para otimização de casos.

REFERÊNCIAS

- [1] Adorni, G.; Cagnoni, S.; Enderle, S.; Kraetzschmar, G. K.; Mordonini, M.; Plagge, M.; Ritter, M.; Stefan, Sablatnög and Zell, A. Vision-based localization for mobile robots. *Robotics and Autonomous Systems*, pages 103 – 119, 2001.
- [2] Ball, D; Wyeth, G.; and Nuske, S. A global vision system for a robot soccer team. *Australasian Conference on Robotics*, 2004 2004.
- [3] Basso, E. W. Identificação visual de posição e movimento para futebol de robôs. Tese de conclusão de curso, ULBRA, Canoas, 2002.
- [4] Bianchi, R. A. C.; REALI-COSTA, A. H. O sistema de visão computacional do time futeboli de futebol de robôs. *Anais do XIII Congresso Brasileiro de Automática – CBA 2000*, 2000 2000.
- [5] BING-RU, L. ; YUN, X.; YI-MIN, Y.; YI-MIN, X.; and ZHEN-ZHEN Q. A self-localization method with monocular vision for. Technical report, 2005.
- [6] Borges, G. A. Borges; Lima, A. M. N. and Deep, G. S. Controladores cinemáticos de trajetória para robôs móveis. VI Simpósio Brasileiro de Automação Inteligente, Bauru, setembro de 2003.
- [7] Chaves R. A. Uaivision: um sistema de visão para futebol. Tese de conclusão de curso, UFSJ, São João del-Rei, MG, 2015.
- [8] Couto, L. N. Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência. Tese de Mestrado, USP, São Carlos, 2012.
- [9] Dixon, W. E.; Dawson, D. M. and Zergeroglu, E. Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. *IEEE TRANSACTIONS ON CYBERNETICS*, 31:341 – 352, 2001 2001.
- [10] Hildebrand, L.; Weiss, N. An exemplary robot soccer vision system. *CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby*, 2004.
- [11] Khessal, N. O.; Naing, M. Y.; Hwee, E. Ng. B.; Oo, P. S.; Antony, L. H. S. Vision-based autonomous soccer robots. Technical report, 2000.
- [12] Kim, J.H.; Shim, H.S.; Kim, H.S.; Jung, M.J.; Choi, I.H. and Kim, J.O. A cooperative multi-agent system and its real time application to robot soccer. *International Conference on Robotics and Automation*, pages 638–643, 1997 1997.
- [13] Kim, J.H.; Kim, K.-C.; Kim, D.-H.; Ki, Y.-J.; and Vadakkepat, P. Path planning and role selection mechanism for soccer robots. *IEEE International Conference on Robotics & Automation*, 1998.
- [14] Kuo, C.-H.; Yang, C.-M.; and Yang, F.-C. Development of intelligent vision fusion based. *IEEE International Conference on Mechatronics*, 2005.
- [15] Mayer, G.; Utz, H.; and Kraetzschmar, G. Towards autonomous vision self-calibration for soccer robots. *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2002.

- [16] Nascimento, R. C. A. d. Localização de robôs móveis em ambientes. Tese de Mestrado, UFRN, Natal, RN, 2014.
- [17] Oliveira, L. L. R. d. Controle de trajetória baseado em visão. Tese de Mestrado, UFJF, Juiz de Fora, MG, 2013.
- [18] OpenCV, <http://opencv.org/>, Acesso em: Novembro de 2016.
- [19] Penharbel, E. A.; Destro, R. C.; Tonidandel, F.; and Bianchi, R. A. C. Filtro de imagem baseado em matriz rgb de cores. Technical report, São Bernardo, SP.
- [20] Pinto, M. A. G. Posicionamento e calibração de um manipulador robótico submarino com uso de visão computacional. Tese de Mestrado, PUC-Rio, Rio de Janeiro, 2006.
- [21] RoboCup Federation, <http://www.robocup.org/>, Acesso em: Novembro de 2016.
- [22] Secchi, H. A. Una introducción a los robots moviles. Technical report, San Juan, Argentina, 2008.
- [23] Silva, M. O. d. Campos potenciais modificados aplicado ao controle de múltiplos robôs. Tese de Mestrado, USP, São Carlos, 2011.
- [24] Vieira, F. C. Controle dinâmico de robos móveis com acionamento diferencial. Tese de Mestrado, UFRN, Natal, RN, 2005.
- [25] Wyeth G. and Brown B. Robust adaptive vision for robot soccer. Technical report, Brisbane, Australia.