

Projeto de futebol de robôs na categoria IEEE *Very Small Size Soccer* da Equipe GER

Henrique K. Miyamoto, Uglybe P. Fernandes, Wesley T. S. T. Ide*

Resumo— Este TDP apresenta as contribuições da equipe GER no desenvolvimento do seu projeto para participar da categoria IEEE *Very Small Size Soccer* na Competição Latino-Americana e Brasileira de Robótica (LARC/CBR) 2015. O documento apresenta os subsistemas gerais usados na construção e programação dos robôs, tais como processamento de imagens, localização, estratégia, comunicação por radiofrequência, motores, além de uma possível técnica de controle para melhorar o desempenho dos robôs.

I. INTRODUÇÃO

A equipe GER de futebol de robôs é formada por membros do Grupo de Estudos em Robótica (GER) da Universidade Estadual de Campinas (Unicamp) para participar da Competição Latino-Americana e Brasileira de Robótica (LARC/CBR), na categoria IEEE *Very Small Size Soccer*.

Nessa categoria, cada equipe deve desenvolver três robôs que, com o auxílio de uma câmera instalada sobre o campo e de um computador para processamento das imagens e das estratégias a serem adotadas, sejam capazes de agir de modo autônomo em uma partida de futebol simplificada.

Nesse trabalho serão expostos alguns dos tópicos abordados pela equipe no desenvolvimento do projeto.

II. MÉTODOS E TÉCNICAS

No projeto de um sistema multiagente, como o do futebol de robôs, existem desafios em vários aspectos e apenas com uma abordagem pormenorizada em diversas frentes é possível construir um sistema competitivo. Esses aspectos podem ser de teor físico, como a estrutura dos robôs e o tipo de interface utilizada para comunicação com a câmera e o computador, ou de teor estratégico, como quais formações e jogadas deverão ser mais utilizadas em campo. É interessante notar que esses aspectos não podem ser abordados de todo separadamente: as estratégias adotadas não são independentes do hardware utilizado. Ainda assim, citamos alguns tópicos que merecem destaque no projeto.

A. Arquitetura do sistema da equipe GER

O projeto do GER, baseado na estrutura da competição *Very Small Size Soccer*, compreende um sistema de visão

computacional baseado em uma câmera de vídeo sobre o campo de jogo; um computador controlador, responsável pelo tratamento de imagens, localização dos robôs e planejamento de estratégias; uma rede de comunicação de radiofrequência entre o computador e os robôs; e finalmente o sistema robótico móvel, composto pelos robôs jogadores em campo.

Essa arquitetura foi adotada por definir de forma direta os elementos que deverão ser trabalhados e desenvolvidos durante toda a execução do projeto, priorizando partes essenciais.

Os principais elementos dessa estrutura são representados na Fig. 1. Nela vemos que o computador e os robôs compreendem alguns subsistemas, descritos abaixo:

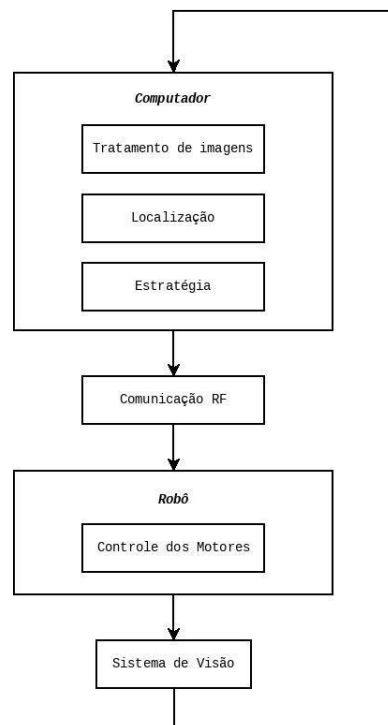


Figura 1. Arquitetura do sistema da equipe GER

1) Tratamento de imagens

Antes mesmo de tratar qualquer imagem, é necessário ajustar a calibragem da câmera, pois isso influencia diretamente nos resultados que ela irá apresentar. Assim, para que a calibragem seja ajustada da melhor forma possível, é muito importante ter conhecimento dos fatores que cercam o ambiente. O principal fator a ser considerado é a iluminação, já que diferentes iluminações causam diferentes tons das cores nas imagens, e mudanças como essa, durante o processamento, podem atrapalhar toda a parametrização realizada para um intervalo de iluminação aceitável. Uma

*H. K. Miyamoto é da Faculdade de Engenharia Elétrica e de Computação (FEEC) da Unicamp, Avenida Albert Einstein, 400, CEP 13803-970, Cidade Universitária, Barão Geraldo, Campinas/SP (e-mail: miyamotohk@gmail.com).

U. P. Fernandes é da Faculdade de Engenharia Mecânica (FEM) da Unicamp, Rua Mendeleev, 200, CEP 13083-860, Cidade Universitária, Barão Geraldo, Campinas/SP (e-mail: uglybe@gmail.com).

W. T. S. T. Ide é do Instituto de Computação (IC) da Unicamp, Avenida Albert Einstein, 1251, CEP 13083-853, Cidade Universitária, Barão Geraldo, Campinas/SP (e-mail: wesley36@gmail.com).

forma de diminuir esse problema é garantir que a luz no local da competição será a mesma durante todo o período do jogo.

O módulo de tratamento de imagens é responsável por analisar as imagens vindas da câmera, extraindo informações como os limites da área de jogo e identificando os objetos em campo (robôs e bola). Esse processo será melhor detalhado na seção D – Processamento de Imagens.

2) Localização

Com as informações obtidas através do tratamento de imagens é possível determinar a pose (posição e orientação), em um sistema de coordenadas, de todos os agentes em campo (robôs da equipe, adversários e bola).

É muito importante determinar a localização de todos os agentes, pois ela influencia diretamente na estratégia que será adotada. Então, quanto mais precisa for a localização, melhor será a execução da estratégia.

3) Estratégia

Conhecendo a situação atual da partida, fornecida pelo sistema de localização, é possível planejar uma estratégia para atingir o objetivo (seja marcar um gol, seja impedir que o oponente o faça). Nesse ponto, diversas técnicas e algoritmos podem ser utilizados para melhorar a eficiência dos robôs em campo, como a forma com que muda de direção, velocidade que deve utilizar, entre outros.

Esse módulo, após analisar o jogo, deve tomar uma decisão e enviá-la aos robôs agentes em campo, através do sistema de comunicação RF.

4) Controle dos motores

Trata-se de um módulo embarcado no sistema robótico móvel, responsável por assegurar a execução das ordens recebidas do computador controlador. Foi montado um sistema de controle em malha fechada, englobando os os motores do robô e os encoders em suas rodas, garantindo que o jogador mantenha a velocidade e direção desejadas.

B. Controlador PID no deslocamento dos robôs

Controlador PID (proporcional integrativo derivativo) é uma técnica de controle utilizada em muitas áreas. Ela calcula um “erro”, que servirá como ajuste, entre a posição atual e a posição pretendida, de tal forma que esse ajuste suavize a mudança de posição, tornando-a mais rápida e eficiente [1].

O PID é composto de 3 termos de correção: proporcional, integrativo e derivativo.

O termo proporcional (P) faz uma correção proporcional ao erro. Este fator é calculado da seguinte forma:

$$P_{out} = K_p e(t) \quad (1)$$

K_p = constante de ganho proporcional

$e(t)$ = erro medido

O termo integrativo (I) é o erro acumulado em um determinado período de tempo. Este controle melhora o desempenho quando o valor é bastante consistente, mas não é tão satisfatório com valores em constante mudança. Este fator é calculado da seguinte forma: seguinte forma:

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (2)$$

$e(T)$ = erro no instante T

K_i = constante de ganho integrativo

Por fim, há o termo derivativo (D), que é a taxa de variação de erros, ou seja, ele só tem efeito se existir tal variação. Este fator atua no sentido de atenuar variações, evitando, por exemplo, mudanças bruscas. Ele é calculado da seguinte forma:

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (3)$$

K_d = constante de ganho derivativo

$e(t)$ = erro medido

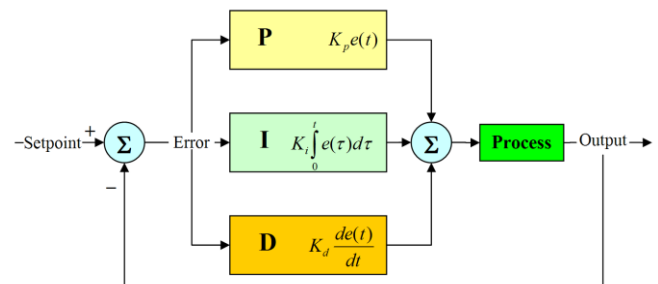
Juntando tudo, temos:

$$\text{erro} = P + I + D, \quad (4)$$

em que esse erro deve ser adicionado ao valor que está sendo usado. A Fig. 2 mostra o fluxo para calcular o PID.

Essa técnica pode ser empregada para o cálculo da potência que deve ser passada ao motor em situações de movimento parcialmente obstruído, como numa disputa de bola, de modo que o robô se movimenta de modo mais similar ao que faria em uma situação de caminho livre.

Figura 2. Fluxo de funcionamento do cálculo do PID [2]



C. OpenCV

OpenCV (Open Source Computer Vision Library) é uma biblioteca *open source* voltada para visão computacional e aprendizado de máquina. Ela tem diversos algoritmos otimizados que podem ser usados para identificar objetos, acompanhar os movimentos de câmera, rastrear objetos em movimento, encontrar imagens similares de um banco de imagens, reconhecer cenário e estabelecer marcadores para cobri-la com a realidade aumentada, etc. [3] OpenCV foi pensada para ter uma grande eficiência computacional, com foco em aplicações em tempo real, como é o nosso caso. Ela é escrita em linguagem C, e ainda pode ter vantagens com múltiplos processadores [4].

A biblioteca está dividida em cinco grupos de funções: processamento de imagens, análise estrutural, análise de movimento e rastreamento de objetos, reconhecimento de padrões e calibração de câmera. Esse modelo pode ser visto na Fig. 3.

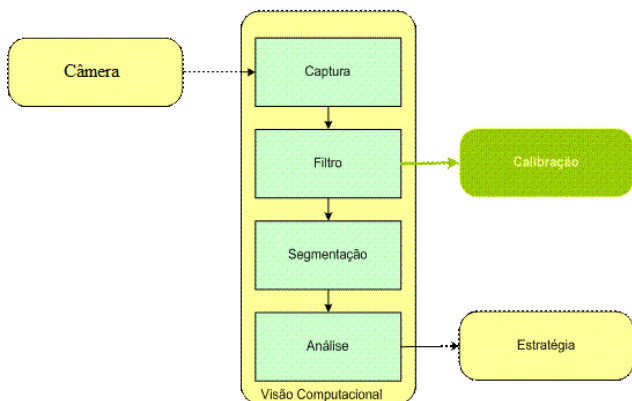


Figura 3. Modelo de processamento de imagens da biblioteca OpenCV

D. Processamento de imagens

A partir do modelo da biblioteca OpenCV foi feita uma adaptação desses grupos para formar uma variação desse processo, no qual o processamento de imagens foi dividido em cinco grupos de funções: aquisição, pré-processamento, segmentação, representação e descrição, reconhecimento de padrões e interpretação [5]. Esse modelo pode ser visto na Fig. 4.

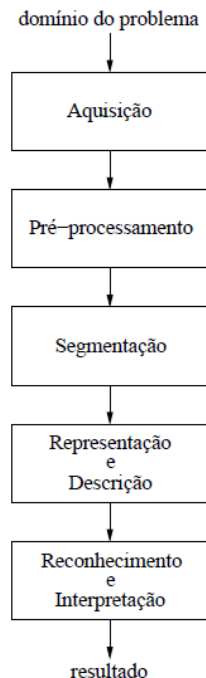


Figura 4. Modelo de processamento de imagens da equipe GER

1) Aquisição

As imagens do jogo serão capturadas pela câmera sobre o campo e convertidas em uma representação adequada para o processamento digital. Tal representação é o RGB (*red, green, blue*). Esse modelo de cores é baseado em um sistema de coordenadas cartesianas, em que o espaço de cores é um cubo, como mostrado na Fig. 5.

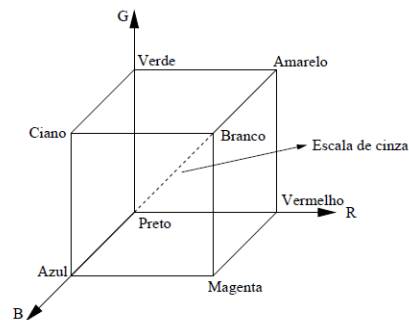


Figura 5. Cubo de cores do modelo RGB

Para os casos em que serão trabalhados fatores com brilho e contraste, é utilizada a representação HSL (*hue, saturation, lightness*), que é definido pelos parâmetros matiz, saturação e luminosidade. A representação gráfica é mostrada na Fig. 6.

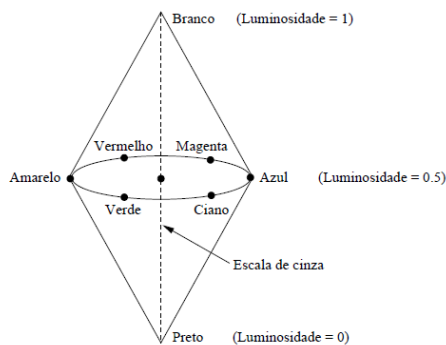


Figura 6. Cones do modelo HSL

2) Pré-processamento

A imagem digital oriunda da câmera apresenta ruídos, que são interferências no sinal de captura da imagem, e que podem atrapalhar a interpretação ou o reconhecimento de objetos na imagem. Ruídos podem vir de diversas fontes, como o tipo de câmera, de sensor, da iluminação, da posição relativa da câmera e do objeto, entre outros.

Então se utilizam algumas técnicas de atenuação de ruído, como a aplicação de filtros passa-baixa. Esse tipo de filtro tende a atenuar transições abruptas da imagem, como é o caso dos ruídos. Por outro lado, o filtro causa um leve borramento na imagem, dependendo da máscara que é aplicada nele. Por isso são utilizados tamanhos de máscara de 3x3 ou 5x5 pixels, para que seja suficiente para atenuar a maior parte dos ruídos, sem perder grandes detalhes da imagem.

Outro filtro que pode ser utilizado é o filtro Gaussiano, que suaviza a imagem. Porém apenas esse filtro não faz diferenciação entre bordas, podendo confundir onde realmente estão as bordas da imagem que são importantes, como os limites do campo. Para corrigir isso, basta utilizar uma filtragem de preservação de bordas, que utiliza o filtro Gaussiano, mas com máscaras específicas. Existem vários tipos de máscaras, como a de Kuwahara, de Tomita e Tsuji, de Nagao e Matsuyama, de Soomboonkaew, entre outras. A que está sendo utilizada pela equipe GER, é a de Soomboonkaew.

3) Segmentação

As técnicas utilizadas nessa parte envolvem encontrar descontinuidades e similaridades na imagem.

Encontrar descontinuidades significa encontrar os locais da imagem onde há mudança do padrão de cor. Para isso são utilizados operadores de gradiente, que são máscaras aplicadas na imagem, porém os valores da máscara são calculados a partir de funções. Existem várias funções que buscam melhores resultados, como de Roberts e de Prewitt. No caso da equipe GER, está sendo utilizado a de Prewitt. Outras opções ainda são utilizar o operador Laplaciano do Gaussiano, de Canny, ou de Boie e Cox.

Encontrar similaridades significa, a partir de um determinado padrão encontrado em uma parte da imagem, seguir esse padrão em todas as direções a partir desse ponto até onde não tenha mais esse padrão. Dessa forma tem-se toda a região procurada. Uma aplicação disso para a competição é, a partir da localização de um ponto que indique que há um robô ali, expandir essa similaridade para saber todo o espaço ocupado pelo robô. Uma técnica para fazer isto é a Limiarização. Esta pode ser global (forma genérica) ou usar variações que foram criadas buscando melhores resultados, como de Bernsen, de Niblack ou de Sauvola e Pietaksinen. A equipe GER utilizou a forma global.

Após serem aplicadas as técnicas para encontrar descontinuidades e similaridades, os resultados são combinados para dividir a imagem em regiões ou objetos, utilizando características como cor e proximidade para determinar tal divisão. Nessa etapa realizamos a identificação de áreas de interesse contidas na imagem, como os limites e linhas do campo, os robôs no campo e a bola.

4) Representação e descrição

O processo de representação é a forma como são identificadas as estruturas extraídas da imagem, para posterior manipulação. As estruturas que são identificadas são: campo, robôs do time, robôs adversários e a bola. Uma estrutura pode ser representada em termos de suas características externas (bordas) ou internas (pixels que compõem o objeto).

Uma forma de representar as bordas é o código da cadeia, que, ao invés de armazenar as coordenadas absolutas dos pixels, utiliza a posição relativa entre pixels consecutivos da borda. Dado um ponto inicial pertencente à borda, o código é definido por uma sequência formada pelas direções entre cada pixel e seu vizinho, até que todos os pixels da borda sejam considerados. Essa é a técnica utilizada, porém existem outras, como aproximações poligonais, assinatura e esqueleto do objeto.

No processo de descrição, as características e propriedades dessas estruturas são extraídas para que possam ser utilizadas posteriormente. Após essa extração, são gerados descritores de regiões, no qual esses passam a representar aquela forma de região. Dessa forma, em processamentos seguintes, pode-se usar esse descritor para utilizar as propriedades da região.

5) Reconhecimento de padrões e interpretação

Baseado em todas as informações e dados coletados até o momento, é possível fazer um reconhecimento dos padrões esperados para os resultados dos processamentos anteriores.

No caso do futebol de robôs, é esperado que se mantenha o padrão do campo, dos robôs e da bola durante todas as leituras da câmera para o mesmo jogo, e dessa forma, identificar cada elemento e relacionar com as imagens processadas antes.

Com isso em mãos, é possível interpretar essas informações para, por exemplo, determinar para onde a bola está indo, qual a direção que ela poderá estar alguns segundos depois, observar o que os robôs adversários estão fazendo para poder, por exemplo, bloquear, roubar a bola, driblar, entre outras ações. Essa parte de interpretação está relacionada com o aprendizado de máquina e inteligência artificial, pois os robôs podem ir aprendendo com os acontecimentos no campo para tomar melhores decisões em jogadas seguintes. Esta combinação possibilita resultados mais rápidos e eficientes.

A equipe GER não utiliza essa estratégia de aprendizado de máquina, mas é uma ideia para futuros trabalhos.

III. CONSIDERAÇÕES FINAIS

O trabalho está sendo desenvolvido de forma simples, mas concisa, utilizando conceitos fundamentais para obter o melhor desempenho dos robôs.

A partir do sistema visão é feita toda captura das imagens. São utilizadas técnicas de programação para fazer o processamento de forma mais eficiente. E, a partir do processamento e da análise dessas imagens, é possível traçar estratégias para serem aplicadas.

Com a transmissão por radiofrequência, os robôs recebem a estratégia adotada, para assim executá-la.

AGRADECIMENTOS

Agradecemos ao apoio dado pela FEEC (Faculdade de Engenharia Elétrica e de Computação), pela FEM (Faculdade de Engenharia Mecânica) e pelo IC (Instituto de Computação), além da ajuda e suporte do Professor Ely Paiva (Unicamp).

REFERÊNCIAS

- [1] K. Aström and T. Häggglund, "PID Controllers: Theory, Design and Tuning", 2nd edition, 1995.
- [2] Wikimedia Commons. File:Pid-feedback-nct-int-correct.png. Disponível em: <<https://commons.wikimedia.org/wiki/File:Pid-feedback-nct-int-correct.png#file>>. Acesso em: 12 jul. 2015.
- [3] OpenCV. About. Disponível em: <<http://opencv.org/about.html>>. Acesso em 4 jul. 2015.
- [4] G. Bradski and A. Kaehler, "Learning OpenCV, O'Reilly", 1st edition, 2008.
- [5] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice – Knowledge Engineering Review", 1994.
- [6] H. Pedrini, W. R. Schwartz, "Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações". Editora Thomson Learning, 2007.