

Métodos Computacionais para Competições de Futebol de Robôs

Computational Methods for Robot Soccer Competitions

Lucca Rawlyk Holosbach

rh.lucca@gmail.com

Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

João Alberto Fabro

fabro@utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

RESUMO

A categoria VSSS da LARC propõe o desafio de montar um time de robôs, cujo tamanho é bem restrito, que jogue autonomamente num campo também muito reduzido. Na competição um aparador é posto sobre o campo de uma forma que é possível posicionar uma câmera que capture a imagem do campo inteiro. Assim deve haver um sistema que transforme a imagem da câmera em comandos para cada robô. É trivial abstrair que existem dois processos relativamente independentes envolvidos, o processamento da imagem para obter as informações do campo e a transformação dessas nos comandos aos robôs por uma inteligência artificial. Após estudos de ferramentas, foi montado e implementado uma estrutura do sistema escrito em C++ que possui duas características notáveis. A primeira sendo permitir a facilidade no entendimento e calibração dos processos executados por meio de uma interface gráfica, escrita com a biblioteca Qt. E a segunda é a modularidade do sistema, que disponibiliza um protocolo de comunicação entre programas usando memória compartilhada com o módulo Interprocess da biblioteca Boost. O módulo de Visão desenvolvido utiliza a biblioteca de visão computacional OpenCV. Há uma perspectiva de se criar dois módulos de IA, um programado puramente em C++ no POO, e outro no PON, visto que os dois se baseariam em regras o segundo promete melhorias no desempenho e na facilidade de manutenção do código.

PALAVRAS-CHAVE: visão computacional, inteligência artificial, futebol de robôs

ABSTRACT

The VSSS category at LARC proposes the challenge of building a robot team with great size restrictions that is able to autonomously play on an also small sized field. A support is placed during the competition so that it is possible to position a camera capturing the whole field's image. This way there needs to be a system that transforms the camera image into commands for each robot. It is trivial to see that there are two relatively independent processes involved in the transformation, the image processing to obtain the field's information and the transformation of those into the commands using an artificial intelligence. After studying tools, a model was implemented for a C++ system that takes care of the duty, this system has two notable characteristics. The first is a graphic interface, written with the Qt library, which makes the system calibration and debugging greatly easier. And the second one is the system's modularity, rising from the availability of a shared memory based communicating protocol between programs using the Interprocess module of the Boost libraries. The Vision module uses the computer vision library OpenCV, which gives access to efficient function for the image processing in order to detect the wanted colors in the image. There is a prospective to create two AI modules, one programmed using pure C++ with OOP, and the other using NOP, having both based on rules but with the second method promising to have performance and code maintenance benefits.

KEYWORDS: computer vision, artificial intelligence, robot soccer

Recebido: 31 de agosto de 2018

Aprovado: 4 de outubro de 2018

Direito autorial:



INTRODUÇÃO

A LARC (Latin American Robotics Competition) é uma competição de robótica anual, onde equipes disputam em categorias que testam a capacidade de seus robôs de executar tarefas (por exemplo jogar futebol). Esta competição, assim, incentiva o desenvolvimento de tecnologias inovadoras. No futebol de robôs as equipes desenvolvem sistemas robóticos que jogam de forma totalmente autônoma, isto é, sem nenhuma interferência humana em seu funcionamento, e visam ter êxito em vencer partidas de alguma variante do futebol.

A categoria VSSS (Very Small Size Soccer - Futebol de Robôs Muito Pequenos) é uma categoria onde 6 robôs (três de cada time) se enfrentam numa arena que mede 150cm x 130cm, e seus tamanhos são restringidos a um cubo de lado medindo 8cm. Portanto, como já se poderia inferir pelo nome da categoria, se trata de robôs pequenos jogando em um campo pequeno. Isso limita bastante o hardware e a mecânica dos robôs, como pode ser exemplificado pelo fato de nenhuma equipe ter conseguido fazer robôs com chutadores (impulsionadores da bola, mecanismo que é crucial em outras categorias) pela falta de espaço.

A dimensão reduzida não só limita a construção mecânica e eletrônica do robô em si, mas também gera uma exigência de desempenho do sistema, já que os robôs estão constantemente interagindo no jogo, sendo assim mais estritamente de tempo real. Isso obriga o uso dos recursos computacionais com cuidado, visto que atrasos não são tolerados.

A forma global usada de sensoriamento do campo para a categoria é uma câmera posicionada num aparador que a deixa suspensa numa certa altura (em torno de 2 metros). Diversas equipes utilizam a câmera Logitech C920, câmera também usada por nossa equipe, que consegue enquadrar o campo inteiro na imagem.

Para facilitar a tarefa de reconhecimento dos objetos no campo (denominada tarefa de "Visão") é definido um padrão de cores para cada elemento dos jogos. O campo tem a cor preta com base fosca, para possibilitar a captura de imagens o mais nítidas possíveis, já que evita a saturação da imagem pela reflexão. A bola tem a cor laranja (é usada uma bola padrão de golfe), portanto os robôs de nenhum time podem utilizar essa cor. Em um jogo um time adota a cor azul e o outro amarelo, assim cada robô dos times é obrigado a ter um identificador (algo disposto em cima do robô que dispõe uma cor para a câmera) da cor de seu time. Além disso cada robô também pode ter identificadores auxiliares de uma cor diferente das citadas anteriormente para a detecção da sua orientação, como pode se ver na figura 1.

DENSOVIMENTO

No começo do projeto foi definido que ele seria desenvolvido para sistemas operacionais Linux, e como existia um histórico no laboratório de uso de C++, o sistema seria desenvolvido nessa linguagem.

O desenvolvimento do sistema começou poucos meses antes da realização da competição de 2017, e como seria de interesse ter algo funcional para adquirir experiência, as primeiras implementações do projeto começaram sem muito

planejamento. Porém desde o começo percebeu-se uma necessidade de haver programas diferentes para cada função, então foi feita uma pesquisa sobre formas de comunicação entre programas.

Figura 1 – Robôs da equipe na competição



Fonte: Autoria Própria.

A forma que se sobressaiu mais na concepção do projeto era a de memória compartilhada. Dentro de um sistema operacional existe um controle de acesso a áreas de memória, de uma forma que um programa não consegue acessar uma área que não lhe pertence, porém a memória compartilhada abre esse caminho, permitindo processos paralelos trabalharem com os mesmos dados.

Então foi projetado a primeira estrutura do sistema, que parecia interessante, porém como veremos tem seus defeitos. Depois de modelada a estrutura o próximo ponto a ser estudado para o desenvolvimento era como fazer o processamento de imagem. Após pesquisas visando biblioteca eficientes e populares de visão computacional entre as equipes da competição e em projetos com objetivos similares constatou-se que a que mais se destacava era a biblioteca OpenCV (OPENCV-TEAM, 2018), uma biblioteca de código aberto de visão computacional disponível para C++. Assim foi decido estudar e implementar o processamento de imagem usando essa biblioteca.

Para a detecção de cores a biblioteca OpenCV disponibiliza o seguinte método: é criado um filtro, que possui limites calibrados de alguma forma, para binarizar a imagem ressaltando a cor desejada. Assim localizam-se os pixels que pertencem à cor em branco e o resto em preto. Daí podemos usar um algoritmo presente no OpenCV que encontra contornos, isso é, para cada área branca contínua ele encontra os pixels pertencentes ao seu contorno, assim pode-se atribuir cada contorno a um objeto com a cor desejada. Depois, após a remoção de ruídos, obtém-se uma lista de objetos com a cor, que após feito o processo para cada cor procurada dá espaço para o algoritmo de reconhecimento de padrões localizar todas as entidades no campo. A desvantagem desse método é que é necessário calibrar o filtro de cada cor usada no jogo para se adequar a iluminação presente no campo.



Após isso foi desenvolvida uma IA simples em C++, que usa regras e álgebra linear para decidir o comportamento adequado dos robôs. Como por exemplo: se a bola está no campo aliado, o robô definido como defensor se movimenta buscando posicionar-se no ponto médio do segmento de reta que liga a posição da bola ao ponto médio do gol. É importante citar que a equipe é a precursora da utilização de robôs omnidirecionais na categoria, assim há uma dificuldade menor em comandar os robôs ao longo do plano por parte da IA.

Essa modelagem proporcionou um entendimento maior de parte dos desafios do projeto, porém ela apresenta falhas severas. A separação entre os programas proporciona um menor interdependência no código, porém isso também espalha o código de uma forma que ao executar o sistema, não é possível abstrair nada que está acontecendo. Fora isso, a criação de várias memórias compartilhadas não é uma prática muito interessante por motivos similares.

Para solucionar esses problemas e ter a possibilidade de ter um sistema robusto, foi feita uma nova modelagem, dessa vez contendo um programa que tem a funcionalidade de um sistema central. Esse programa é basicamente uma interface gráfica e um módulo de controle da memória compartilhada. Porém ele também cria uma biblioteca que define variáveis necessárias para o acesso da memória compartilhada e define um protocolo de comunicação para os módulos, que são os programas que trocarão informações com esse sistema central, existindo o módulo de Visão, IA e Comunicação (esse último não necessariamente faz parte do escopo do projeto em questão, porém precisa receber as informações do Sistema de Controle, portando necessita de um protocolo). Essa modelagem pode ser vista na figura 2.

A interface gráfica foi feita usando a biblioteca Qt (THE-QT-COMPANY, 2018), que é uma biblioteca que possui um módulo bem completo de desenvolvimento de *Widgets*, como pode ser visto na figura. Essa biblioteca também estende a linguagem C++ introduzindo o conceito de *signals* e *slots*. Esses, em conjunto com a existência de um módulo de Threads da biblioteca, o QThread, levaram a possibilidade de existirem Threads nos módulos, melhorando o desempenho do sistema. Essa interface pode ser vista na figura 3.

Há o prospecto também da construção de uma IA em PON (SIMAO TACLA, 2012), que utiliza os mesmos conceitos da IA apresentados anteriormente (regras e álgebra linear), porém possui certas facilidades na linguagem. No PON a programação não é feita na forma padrão onde o código segue um certo fluxo de operações subsequentes, em vez disso são criados os chamados elementos da base de fatos que são comparáveis às classes no POO, possuindo atributos e métodos, porém em vez de criarmos uma sequência de linhas a serem executadas, são somente criadas regras, que definem quando um método é chamado baseado em atributos. Essa forma de programar permite ao compilador otimizar o programa, uma vez que evitamos testes constantes de variáveis, já que só precisamos realizar uma tarefa quando é detectada uma mudança em um atributo. Outro benefício, que é relevante neste projeto, é a facilidade da manutenção do código, que pode ser necessária em caso de mudança de regras

da categoria, uma vez que só precisamos conhecer as regras para realizar tal manutenção, e não o fluxo inteiro do código.

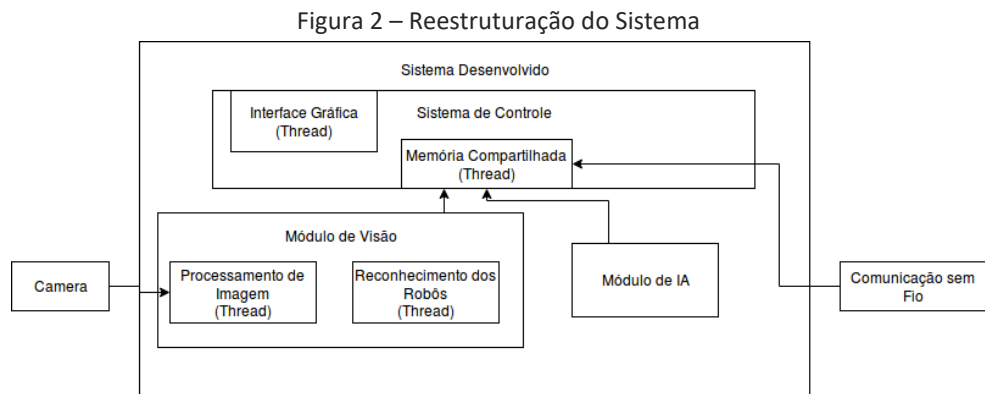
CONCLUSÃO

A reestruturação em primeira instância pareceu ter uma dificuldade menor que a presenciada, o grande número de funcionalidades de bibliotecas diferentes se mostrou custoso no requisito de tempo para um desenvolvedor apenas. Não só isso como também há uma grande dificuldade implícita em C++ por ser uma linguagem de nível baixo comparada a outras como Python.

Essas dificuldades acarretaram numa pouca produtividade do tempo, tendo em vista que o ainda existem muitos `\emph{bugs}` na reestruturação do código que precisam ser resolvidos até a data da competição.

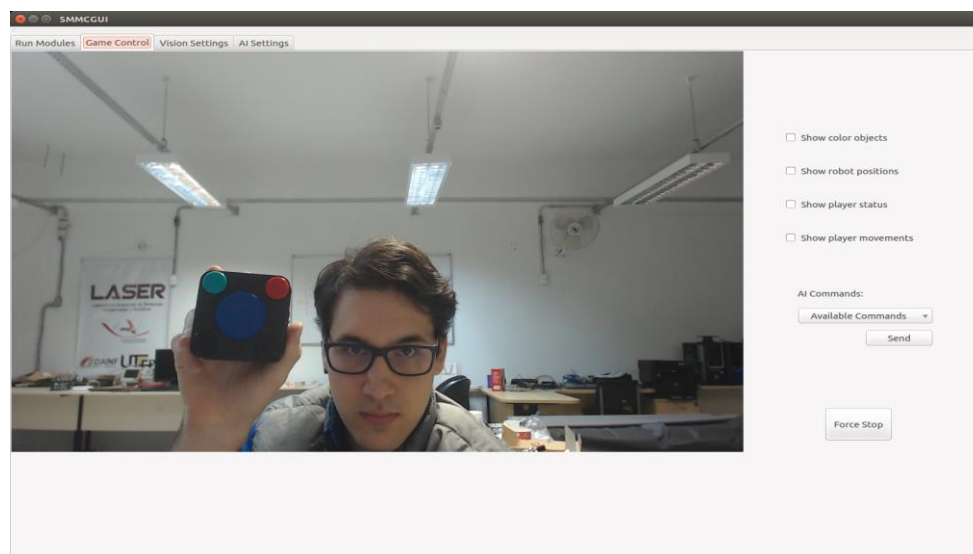
Os ganhos dos métodos usados para a criação desse sistema provavelmente serão mais visíveis ao longo do tempo. Porém já é possível se notar a flexibilidade do sistema ao longo de teste e sessões de correção do código.

O desempenho do sistema na competição ainda é especulativo já que o time dos robôs também está em desenvolvimento, porém em teste foi possível constatar que o tempo computacional do sistema com as otimizações presente é mais que suficiente para o controle dos robôs.



Fonte: Autoria Própria.

Figura 3 – Interface Gráfica do Sistema



Fonte: Autoria Própria.

CITAÇÕES E REFERÊNCIAS

BOOST-C++-LIBRARIES. Boost. 2018. Disponível em: <<https://www.boost.org/>>. Acesso em: 28 de agosto de 2018.

KITWARE. CMake. 2018. Disponível em: <<https://cmake.org/>>. Acesso em: 28 de agosto de 2018.

OPENCV-TEAM. OpenCV. 2018. Disponível em: <<https://opencv.org/>>. Acesso em: 28 de agosto de 2018.

SIMAO TACLA, S. B. Notification Oriented Paradigm (NOP) and Imperative Paradigm: A Comparative Study. 2012. Disponível em: <<https://pdfs.semanticscholar.org/3dcd/d8508c2702db7ff832d46abcb9cf50d33e02.pdf>>. Acesso em: 28 de agosto de 2018.

THE-QT-COMPANY. Qt. 2018. Disponível em: <<https://www.qt.io/>>. Acesso em: 28 de agosto de 2018.

AGRADECIMENTOS

Gostaria de agradecer a todos que compõem a equipe do LASER ou que já contribuíram no projeto, e aos professores que coordenam o laboratório, especialmente ao meu coordenador João Fabro.

Agradeço também aos organizadores e competidores da categoria VSSS na LARC que estão sempre contribuindo para o desenvolvimento coletivo de todos os times visando um maior nível em toda a competição.

E finalmente agradeço a Fundação Araucária, que possibilitou o desenvolvimento desse projeto ao proporcionar a bolsa de incentivo a Iniciação Científica.