

Importância do Pré-Processamento no Reconhecimento de Padrões para IEEE Very Small Size Soccer (VSSS)

Endy K. F. Neumann¹, Waldomiro S. Yuan¹

¹ Universidade Federal do Paraná, Curitiba, Brasil
endy@ufpr.br, wsyuan@ufpr.br

Resumo— Este artigo descreve o desenvolvimento de um algoritmo simples de reconhecimento de padrões utilizando bibliotecas em código aberto no Python, para uso em torneios de competição de futebol de robô autônomo que utilizam a regra do IEEE Very Small Size Soccer (VSSS) Série B, uma competição de robótica em pequena escala. As imagens foram geradas através de um aplicativo de modelagem 3D de situações hipotéticas de uma partida. A partir da imagem da câmera, inicialmente, o programa faz um pré-processamento dessa imagem e, posteriormente, utiliza técnicas para identificar a posição dos robôs jogadores e da bola. Assim, este trabalho demonstra que o pré-processamento pode facilitar e melhorar os resultados do algoritmo de reconhecimento de padrão.

I. INTRODUÇÃO

A *Very Small Soccer* (VSSS) é uma categoria de torneio de robótica estabelecido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE). Além de ser uma competição, é um projeto educacional no qual equipes de estudantes e pesquisadores projetam e constroem robôs autônomos, onde dois times de 3 robôs com no máximo de 8,0 cm³ disputam uma partida de futebol.

As regras do jogo são semelhantes ao futebol convencional, embora tenham sido ajustadas para se adequar ao contexto e escala do VSSS. Todos os robôs de uma equipe específica têm a capacidade de mudar de posição ao longo da partida, podendo atuar como atacante, defensor ou goleiro. O objetivo da partida é o mesmo: marcar gols no time adversário ao conduzir a bola pelo campo [1].

Neste artigo apresentamos o desenvolvimento de um algoritmo para reconhecimento de padrões em Python, utilizando bibliotecas de código aberto, como o OpenCV, numpy e Matplotlib, projetado para ser aplicado no VSSS. O algoritmo emprega técnicas de identificação de padrão para possibilitar a localização e rastreamento de objetos-chave, como o campo, a bola e os adversários, desenvolvendo assim uma ‘percepção’ do ambiente para

posteriormente os robôs poderem executar jogadas durante a partida.

II. REGRAS DO JOGO

A. Campo e Bola

O campo é formado por uma chapa plana e rígida de madeira com dimensões de 150 cm x 130 cm de cor preta fosca e laterais com medidas de 5 cm de altura por 2,5 cm de largura. A parte superior das laterais possuem a mesma cor do campo e as paredes são na cor branca. Nos cantos há quatro triângulos isósceles sólidos de 7 cm x 7cm para a bola não ficar presa nesses locais. A bola deve ser de cor laranja com diâmetro de aproximadamente 42,7 mm de diâmetro e 46 g de massa [2].

No campo também há diversas marcações de linha na cor branca para delimitar o jogo e fornecer referências visuais para os robôs, e pequenos círculos para posicionar a bola em casos de falta para efetuar tiros livres.

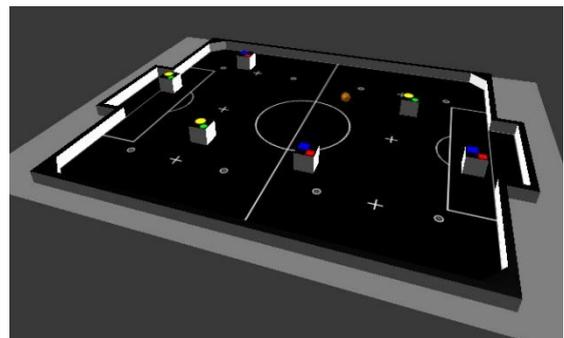


Fig. 1. Vista em perspectiva do IEEE VSSS – série B (Modelo 3D – Blender) (Fonte: Autores).

B. Robôs

Os robôs devem possuir tamanho total com capa limitado a 80 mm x 80 mm x 80 mm, desconsiderando a altura de suas antenas de radiofrequência. Sua parte

superior não deve conter as cores laranja, branco ou cinza e não deve ser colorida com mais de duas cores diferentes da cor preta e da cor do time.

C. A partida

O jogo ocorre em tempos regulamentados, com equipes tentando marcar mais gols do que a equipe adversária para vencer. Uma partida é disputada por dois times de no máximo 3 robôs, podendo um deles ser o goleiro. Permite-se a presença de apenas de três membros de cada time no campo de competição e utiliza-se um computador por equipe dedicado ao processamento visual e para reconhecimento das posições.

III. PROCESSO DE RECONHECIMENTO DE PADRÃO

A imagem é capturada por uma câmera a 2 m de altura do centro do campo. O campo, os jogadores e a bola foram modelados em um ambiente virtual 3D através do aplicativo *Blender*[4]. Nesta modelagem, foi configurado também duas luzes e a posição da câmera: 2 m de altura e voltada para baixo. Assim foi capturada uma imagem pela câmera de uma situação de jogo.

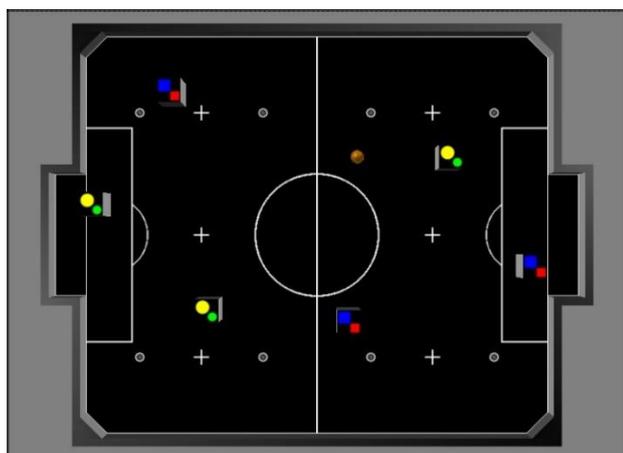


Fig. 2. Vista da câmera de uma situação de jogos (Fonte: Autores).

Os jogadores de cada time devem ter uma cor obrigatória para diferenciá-los. Assim, um time deve possuir uma figura amarela e o outro time deve possuir uma figura azul. O time A apresenta um círculo amarelo e um círculo verde e o time B , um quadrado azul e um quadrado vermelho.

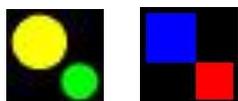


Fig. 3. Time A - Elemento estruturante que é a imagem para identificar padrão e o Time B (Fonte: Autores)

Assim, o processo de identificação do padrão será aplicado na imagem da situação de jogo (Fig. 2) com o padrão do time A (Fig. 3). Esse processo é composto de duas etapas: pré-processamento e processamento.

A. Pré-processamento

O objetivo do pré-processamento é melhorar o desempenho do algoritmo que busca o padrão.

Para poder analisar o efeito deste pré-processamento foram considerados três casos de estudo: sem pré-processamento, pré-processamento I e pré-processamento II.

A1) Sem Pré-processamento

Nesse primeiro caso, efetivamente só há conversão da imagem de uma situação da partida de cores RGB para escala de cinza.



Fig. 4. A figura sem pré-processamento (Fonte: Autores).

A2) Pré-processamento I

Antes de iniciar esse pré-processamento 1, a imagem do campo sem os jogadores de ambos os times e sem a bola deve ser capturada como mostrado na Figura 5.

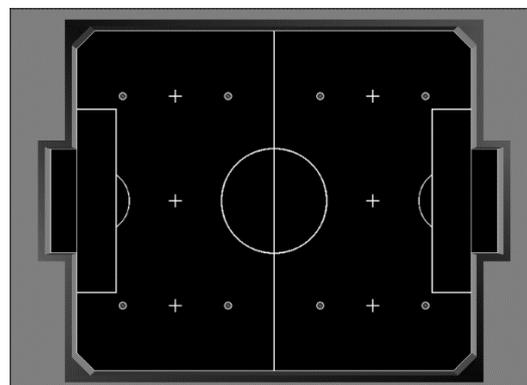


Fig. 5. O campo sem jogadores e bola.

O pré-processamento I converte a imagem de uma situação da partida em escala de cinza e subtrai a imagem capturada no início (campo sem jogadores e nem bola). A ideia seria eliminar as linhas de marcação do campo, todavia se houver deslocamento entre a imagem inicialmente capturada e a imagem recém capturada, não será possível eliminar essas linhas de campo como mostrado na figura 6 (deslocamento introduzido propositalmente).

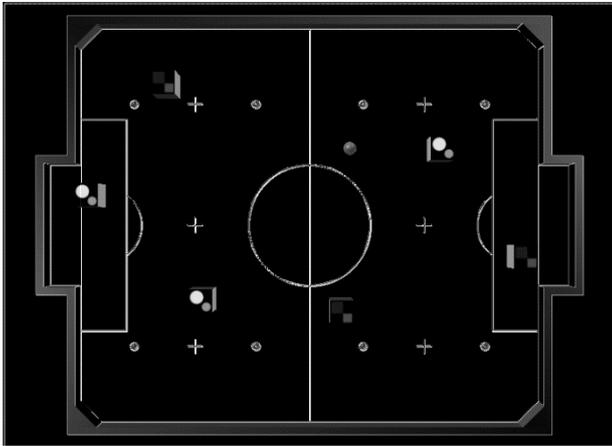


Fig. 6. Figura resultante do pré-processamento I. (Fonte: Autores).

A3) Pré-processamento II

Nesse terceiro caso, o pré-processamento II elimina todos os pixels de cor cinza ($R = G = B$) transformando-os na cor preta ($=$ zero), deixando somente os pixels coloridos: jogadores e bola. Como apresentado na Fig. 7

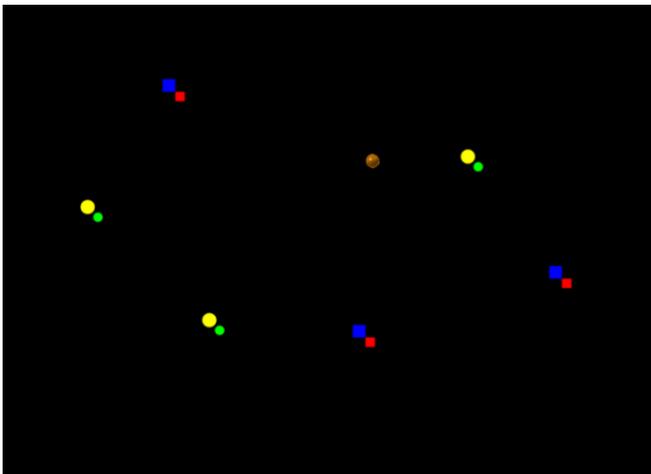


Fig. 7. Imagem obtida após o pré-processamento 2 (Fonte: Autores).

Este ainda é convertido para o cinza antes de acabar o procedimento



Fig. 8. Imagem após o pré-processamento convertida para escala de cinza (Fonte: Autores).

B. Algoritmo I

O algoritmo utiliza a imagem resultante da etapa anterior (pré-processamento: três casos) como base para identificar a posição do padrão (também em escala de cinza). Para isso, ele realiza uma varredura sobre a imagem base em busca desse padrão (convolução bidimensional). Assim, para cada ponto da imagem será gerado um ponto na matriz erro que é a soma dos erros entre os pixels do elemento estruturante e os pixels da região na imagem correspondente ao elemento estruturante.

Portanto busca-se pontos na matriz erro onde o erro é mínimo, indicando a localização mais provável do padrão (valores mais próximos do zero são pretos e valores próximos do 255 são brancos).

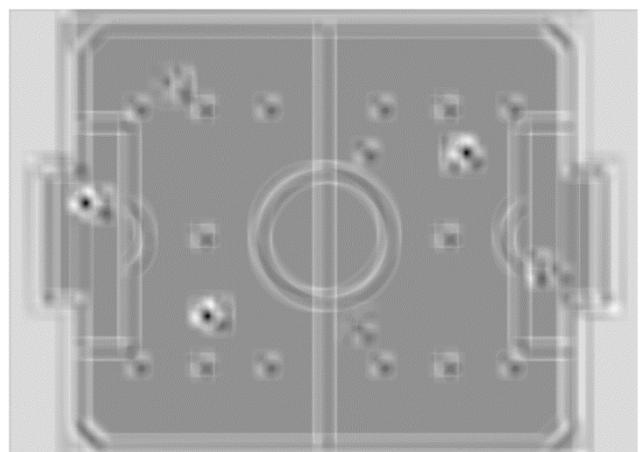


Fig. 9. Imagem após o pré-processamento convertida para escala de cinza (Fonte: Autores).

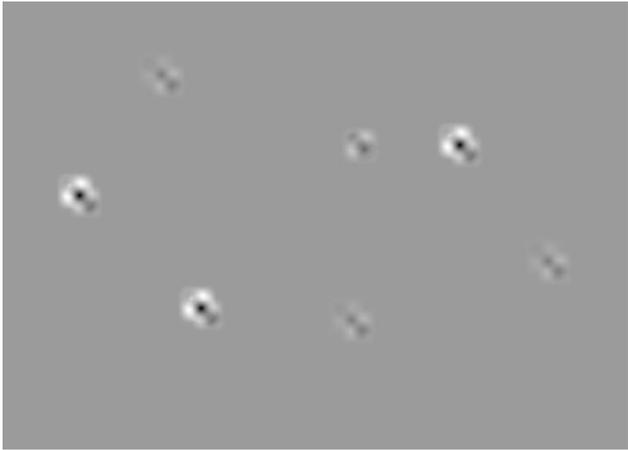


Fig. 10. Imagem da matriz erro resultante da aplicação do algoritmo sobre a resultante do pré-processamento II (Fonte: Autores).

IV. ANÁLISE DE RESULTADOS

Foi executado o pré-processamento e o algoritmo para os três casos.

Para o primeiro caso, a matriz erro resultante está apresentada na Fig. 6 onde pode-se identificar a posição dos jogadores. Todavia não é possível identificar com clareza a posição dos adversários e da bola, além disso a imagem está contaminada com as linhas do campo.

No segundo caso a matriz erro resultante está apresentado na Fig. 9, como no caso anterior, é possível identificar a posição dos jogadores, mas não é possível identificar a posição dos adversários e da bola, porém percebe-se menor variação de valores fora da área de identificação de jogadores.

No terceiro caso a matriz erro resultante está na Fig. 10. onde é possível identificar os jogadores do time e as posições dos jogadores adversários e da bola, porém sem diferenciar os adversários e a bola, seria necessário mais um processamento para diferenciá-los.

V. CONCLUSÃO

Nos três casos foi possível identificar a posição dos jogadores da equipe, porém somente no último caso foi possível identificar a posição dos jogadores adversários e/ou da bola, sendo necessário processamento adicional para diferenciar a bola dos jogadores adversários.

Percebe-se claramente que o pré-processamento pode ou não melhorar o desempenho do algoritmo como foi demonstrado. O resultado do caso I e do caso II foi bem semelhante, embora um tenha tido pré-processamento e o outro não. O pré-processamento II do caso III melhorou consideravelmente o resultado do algoritmo comparado com os casos I e II. Assim podemos concluir que o pré-processamento poderá ajudar se for realizado da maneira correta.

Este é um estudo inicial, onde se preocupou somente com a identificação das posições dos padrões. A continuação deste trabalho será a identificação das orientações dos padrões (rotação dos padrões), além de aplicar estratégia de jogo.

REFERÊNCIAS

- [1] De Oliveira, R. V. "Protótipo de robô para a categoria IEE Very Small Size Soccer: Design mecânico, eletrônica embarcada e simulação". Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia, Belo Horizonte, 2022.
- [2] Pinto, A. H. M. "Regras IEEE Very Small Size Soccer (VSSS)", 2021. Disponível em: <<https://www.cbrobotica.org/wp-content/uploads/2021/05/vssRules3x321.pdf>>. Acesso em: 09 de outubro de 2023.
- [3] Baldin, R., Yuan, W. S. "Reconhecimento Estrutural de Imagem de um Sistema Pervasivo". 26º EVINCI, Curitiba, 2018.
- [4] Community BO. "Blender - a 3D modelling and rendering package". Stichting Blender Foundation, Amsterdam; 2018. Disponível em: <<http://www.blender.org>>.