

PROJETO FINAL - SISTEMA DE VISÃO COMPUTACIONAL PARA FUTEBOL DE ROBÔS

SAMUEL VENZI LIMA MONTEIRO DE OLIVEIRA
14/0162241*

Email: samuel.venzi@me.com

1 Abstract

Visando melhorar os resultados da equipe de competição de futebol de robôs *UnBall*, um sistema completo de visão computacional foi desenvolvido com o objetivo de fazer a detecção automática do campo, dos respectivos robôs de cada time e da bola. Trabalhos anteriores da equipe foram utilizados como base para desenvolver a lógica do sistema e além disso, a partir da revisão bibliográfica, diversas técnicas de segmentação e identificação já desenvolvidas foram adotadas. O sistema foi desenvolvido com base no processamento de imagens em OpenCV utilizando C++. Uma base de dados para testes foi criada e parte dela utilizada para as devidas calibrações do algoritmo. Os testes foram bem sucedidos em achar todos os elementos propostos com precisão sob condições de luz esperadas. A partir desse trabalho, pode-se, no futuro, aprimorar o sistema, adaptá-lo e otimizá-lo para outras competições.

2 Introdução

1º parágrafo Este trabalho foi realizado utilizando como base as necessidades da equipe *UnBall*, que compete na categoria IEEE Very Small Size de futebol de robôs. A categoria é disputada no Brasil desde 2003, as principais regras dizem que os robôs devem ser controlados remotamente por um computador que processa a imagem de uma câmera posicionada acima do campo. O principal objetivo das competições de robô é desenvolver-se tecnologicamente de forma que em 2050, seja possível disputar um jogo contra humanos. Esse trabalho tratará do desenvolvimento do sistema de visão.

2º parágrafo A visão computacional é parte fundamental da competição pois a estratégia de jogo depende das saídas do algoritmo da visão, como a posição dos robôs e da bola. A *UnBall*, anteriormente, já havia desenvolvido um sistema utilizando o Kinect, porém o sistema ficou pesado por trabalhar com profundidade. Esse trabalho, utiliza diversas técnicas para realizar toda a detecção dos elementos de interesse.

3 Revisão Bibliográfica

4 Metodologia

4.1 Materiais

Todo o desenvolvimento aconteceu com a utilização do OpenCV e com a linguagem C++. A base de dados foi criada especificamente para esse trabalho, com vídeos gravados acima do campo contendo seis robôs e uma bola.

4.2 Detecção do campo

O primeiro passo do desenvolvimento é realizar a detecção do campo, pois assim é possível criar uma região de interesse para posteriormente detectar os robôs e as bolas. Cada *frame* de imagem durante o início da transmissão do vídeo foi convertido em uma imagem HSV e usado apenas seu canal V, que se aproxima da imagem RGB em escala de cinza, mas com menos presença de ruídos e com valores de intensidade mais significativos. Então aplica-se um filtro de Canny para detecção de bordas com o objetivo de identificar as linhas que demarcam o campo. O filtro de Canny retorna uma imagem binária com as principais bordas encontradas na imagem, e dois parâmetros de *thresh* permitem controlar os limites encontrados. À imagem binarizada é aplicada a função *findcountours()* que por sua vez extrai os conjuntos de pontos que formam um contorno, em cima deste retorno é feito o tratamento de forma que apenas o maior contorno seja extraído, uma vez que este é o dado de interesse nesta etapa.

Na etapa seguinte continua-se a aplicar o filtro de Canny no canal V da imagem convertida em HSV, mas com os contornos obtidos na etapa anterior subtraídos do resultado da segmentação de bordas. Tem-se assim apenas uma imagem binária com os robôs e alguns erros decorrentes do desenho não tão preciso do contorno do campo, mas por formarem linhas e não *blobs* podem ser descartados no tratamento da imagem. Tais *blobs* são considerados robôs e ajusta-se por meio de *trackbar* qual a área mínima para que um *blob* seja considerado um robô. É feito também tratamento de forma a desconsiderar retângulos com a proporção maior que 1:3 ou 3:1, uma vez que os robôs tendem a formar retângulos de razão entre os lados de aproximadamente 1. Por fim, para que fique melhor definidos os *blobs* finais, aplica-se uma erosão

na imagem binarizada final usando um kernel 2x2.

A segmentação da bola é feita de maneira distinta dos robôs, por sua cor ser singular no sistema de visão e assim não aproximar-se das outras cores presentes ou possíveis é viável que se faça segmentação por cor nos *frames* de entrada, para que, assim, a bola seja extraída dos outros elementos da imagem. O primeiro passo para isso é que se converta o *frame* RGB da entrada para o canal HSV, de forma a amenizar a contribuição da luminosidade do ambiente na segmentação. O próximo passo é ajustar os valores dos canais H, S e V para que apenas a variação da cor laranja da bola esteja presente na imagem segmentada, com os valores dos canais desejados, aplica-se a função *inrange* resultando numa imagem com apenas um ponto branco referente ao elemento. É importante ressaltar que as duas segmentações são feitas separadamente em instâncias distintas - a segmentação dos robôs não afeta a segmentação da bola-.

5 Resultados

6 Discussão e Conclusões