

# 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

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, e nela seis robôs, três de cada time, disputam a partida. 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. Este não será somente útil no curto-prazo, a competição, será útil para trabalhos futuros já que muitas técnicas e lógicas utilizadas aqui podem ser reaproveitadas para outras finalidades.

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 a falta de poder computacional para se trabalhar com imagens com profundidade atrapalhou o funcionamento do algoritmo. Esse

trabalho, utiliza diversas técnicas para realizar toda a detecção dos elementos de interesse.

O estudo de métodos a aplicação no sistema se mostrou importante, apesar de não ter sido profundo. Técnicas como a criada por John F. Canny para detecção de bordas e o cálculo de histogramas foram adotadas como base para o algoritmo elaborado. Essas técnicas foram utilizadas com o objetivo de extrair as posições dos seis robôs presentes na partida além da bola e do campo.

A técnica de Canny se baseia nas variações de intensidades da imagem, onde variações abruptas, que são quantificadas por meio de derivadas da imagem, são consideradas bordas. O filtro de Canny apresenta uma boa precisão, boa detecção e sua característica de resposta mínima elimina ambiguidade na marcação das bordas.

Para análises de cor, a imagem no sistema de cores HSV (*Hue*, *Saturation* e *Value*) é indicada por sua robustez em *v*. No sistema HSV, diferentemente do sistema RGB, a saturação não afeta todos os canais uniformemente, tornando fácil identificar cores pela matiz (*hue*).

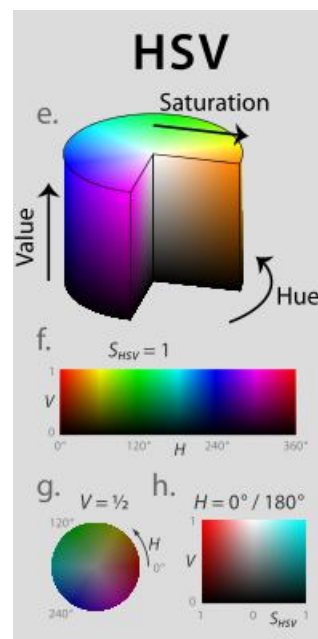


Figure 1: Sistema HSV

### 3 Metodologia

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 de cima de um campo contendo seis robôs e uma bola.

#### 3.1 Detecção do campo

O primeiro passo do desenvolvimento foi realizar a detecção do campo, pois assim é possível criar uma região de interesse para posteriormente detectar os robôs e as bolas. Para cada frame de imagem durante o início da transmissão do vídeo foi aplicado 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 um parâmetro de *thresh* permite controlar os limites encontrado. A partir disso é possível encontrar os contornos e suprimir as linhas da imagem para que na região de interesse permaneçam somente os robôs.

### 4 Resultados

### 5 Discussão e Conclusões