

Tracking de uma bola em um jogo de futebol

Bruno Buss e Lucas Pierezan

4 de julho de 2011

Resumo

Este relatório diz respeito ao trabalho de realizar o *tracking* de uma bola em jogo de futebol. Apresentamos aqui as ideias e métodos utilizados bem como dificuldades encontradas.

1 Introdução

Nesse trabalho foi implementado um algoritmo para realizar o *tracking* de uma bola em um jogo de futebol. Este problema tem diversas aplicações como a detecção automática de eventos do jogo (como gol e impedimento) e geração de dados estatísticos de interesse de emissoras de tv e times de futebol. Na literatura podemos encontrar alguns trabalhos sobre esse assunto [1, 2, 3] indicando um aspecto desafiador devido a fenômenos como oclusão e distorção da bola em alta velocidade. Baseamos nosso trabalho principalmente em ideias presentes em [4]. Outras abordagens como [5] e [6] utilizam como sub-procedimento a detecção de jogadores o que aumentaria consideravelmente a complexidade do trabalho.

Como em [5] utilizamos a Transformada Circular de Hough (CHT) como ferramenta principal. A CHT é um método clássico utilizado para achar padrões circulares em imagens [7]. Existem muitas adaptações da transformada [8] e buscamos incorporar um conjunto de adaptações priorizando os aspectos particulares do nosso problema. Utilizamos a CHT como procedimento para gerar um valor de quão circular é uma região, para posterior filtragem. Na seção 2 explicamos com mais detalhe como isso é feito.

Além da CHT, utilizamos também uma análise de similaridade de histograma para gerar um valor de quão similar é o histograma de uma região comparado com o histograma da bola, que supomos saber. Com os valores de circularidade e similaridade de histograma geramos um *score* final que é então utilizado para localizar a bola. Na seção 3 explicamos como essa análise de histograma é realizada e como é gerado o *score* final.

Na seção 3 explicamos como utilizamos o conceito de Região de Interesse (ROI) para restringir a busca da bola usando informações obtidas em iterações passadas do algoritmo. O comportamento dessa ROI é descrito em detalhes bem como os problemas gerados por essa abordagem.

Vale ressaltar que o algoritmo foi implementado em C++ e utilizamos a biblioteca OpenCV para efetuar alguns sub-procedimentos de processamento de imagens que julgamos secundários.

2 Calculando valor de circularidade

Como já foi dito, para gerar o valor de circularidade, utilizamos a Transformada Circular de Hough (CHT). A CHT trabalha no espaço das arestas, ou seja, buscando padrões circulares no espaço das arestas. Aqui já podemos ver a hipótese da bola estar relativamente bem contrastada na imagem, aparecendo então no espaço das arestas.

Para achar as arestas da imagem I , em RGB, primeiramente convertemos essa para a imagem G_I em escalas de cinza através da função "cvtColor" do openCV. Aplicamos também um borramento gaussiano em G_I para eliminar possíveis ruídos usando a função "GaussianBlur". Finalmente utilizamos a função "Canny".

A CHT busca circunferências no espaço das arestas por um esquema de votação em um espaço acumulador. O espaço acumulador é o espaço dos parâmetros da circunferência que estamos buscando e é da forma *acompletar*. Cada ponto no espaço acumulador representa uma possível circunferência e portanto é da forma (cx, cy, R) . Cada ponto de aresta $P = (x, y)$ e raio R vota, no espaço acumulador, nos pontos de centro que poderiam gerar o ponto P como parte de uma circunferência. De fato, na implementação clássica da CHT, para cada ponto de aresta P e raio R é realizada uma votação no conjunto de pontos $S_{P,R} = ((cx, cy, R) \mid (cx - x)^2 + (cy - y)^2 = R^2)$ no espaço acumulador. O método consiste então em processar todos os pontos de arestas, para os valores R de raio que estão sendo considerados, e acumular votos no espaço dos parâmetros. Os pontos, no espaço dos parâmetros, com maior número de votos são apontados como centros de circunferências.

Em nossa implementação, o espaço acumulador é representado pela *struct acumulador*. Essa *struct* é basicamente uma matriz tridimensional m com as informações do $minR$ e $maxR$. Essa matriz m contém inteiros, que indicam o quantidade de votos recebida por um ponto.

3 Calculando valor de similaridade de histograma

4 Comportamento da Região de Interesse

5 Considerações Finais

Referências