Reconhecimento de Objetos em Tempo Real para Futebol de Robôs

Murilo Fernandes Martins, Flavio Tonidandel e Reinaldo A.C. Bianchi

Centro Universitário da FEI – UniFEI
Av. Humberto de A. Castelo Branco, 3972
09850-901 - São Bernardo do Campo - SP - Brazil
{murilofm, flaviot, rbianchi} @ fei.edu.br

Abstract. *This article presents two techniques to detect and recognize mobile robots in a soccer robot field. One technique is based on model recognition and multiple colors and the other is based on one color and multiple models. The techniques are applied to detect the systems´ own team and the opponent team of robots.*

Resumo. *Este artigo apresenta duas técnicas para detectar objetos em futebol de robôs. Uma técnica é baseada na detecção de uma única forma geométrica independente de cores e a outra é baseada na detecção objetos de uma cor independente da forma. A primeira técnica é usada para detectar os robôs do próprio time e a outra para detecção dos adversários.*

1. Introdução

Desde seu surgimento, o Futebol de Robôs tem sido uma plataforma de pesquisa e desenvolvimento de robótica móvel autônoma e sistemas multi-agentes, cuja intenção é envolver as mais diversas áreas da engenharia e ciência da computação. Esse domínio propõe vários problemas a serem resolvidos, como a construção mecânica, eletrônica e controle de robôs móveis. Mas o principal desafio se encontra nas áreas relacionadas à Inteligência Artificial (I.A.), como sistemas multi-agentes, aprendizado de máquina e visão computacional. Os problemas e desafios citados não são triviais, visto que ambiente do Futebol de Robôs é dinâmico, incerto e probabilístico.

Um sistema de visão computacional para o futebol de robôs deve ser rápido e tolerante à variação de luminosidade, e é desejável que seja capaz de tolerar ruídos e variações de intensidade luminosa. Diversas técnicas podem ser aplicadas para o reconhecimento dos objetos do domínio do Futebol de Robôs, como as descritas por Bianchi e Reali-Costa (2000) e também por Grittani, Gallinelli e Ramírez (2000).

Este trabalho propõe o uso de duas técnicas de segmentação de imagens que detectam os objetos – robôs móveis - por duas maneiras: detecta uma única forma (círculos) independente de cores e detecta uma única cor independente da forma.

Para a detecção de círculos a partir da forma, de qualquer cor, usou-se a Transformada de Hough (HT). Para isso, foi proposta uma nova abordagem para interpretação do espaço de Hough, assim como o método utilizado para reconhecer os objetos, baseado em métodos de satisfação de restrições.

Para a detecção de qualquer formato de uma única cor, foi utilizado um algoritmo baseado em Bianchi e Reali-Costa (2000) estendido para se trabalhar com outros formatos geométricos.

Este artigo está dividido da seguinte maneira: a seção 2 apresenta a detecção de objetos em futebol de robôs. A detecção de uma única forma (círculos) é apresentada na seção 3, detalhando a Transformada de Hough e sua teoria. A detecção de uma única cor independente da forma é descrita na seção 4. Na seção 5 os resultados obtidos são apresentados e a seção 6 conclui o trabalho e apresenta sugestões de trabalhos futuros.

2. Detectando objetos no futebol de robôs

No futebol de robôs, a categoria Mirosot da FIRA (2005) utiliza visão para detectar os objetos, que são robôs móveis, por meio de cores padronizadas para cada equipe que disputa uma partida. Segundo as regras, uma equipe deve ter a cor amarela e a outra equipe deve ter a cor azul no topo dos objetos para que seja reconhecida pelo adversário.

Entretanto, a regra não padroniza a forma geométrica que o robô deve possuir em cima com a cor especificada, desde que a área interna seja um sólido de 12,25 cm^2.

Identificar os objetos pela forma só é possível se a forma for pré-definida ou conhecida a priori. Identificar os objetos por cor pode ter influência da luminosidade e deixar o sistema de visão menos tolerante a variações de luminosidade.

Uma técnica utilizada por Bianchi e Reali-Costa (2000) é a de considerar a cor e a forma. Com a cor e a forma pré-estabelecidas (o círculo), o sistema percorre a imagem pulando pixels. Ao detectar um pixel na cor definida, encontram-se os pixels de mesma cor mais à esquerda e mais à direita e determina-se o tamanho da figura na horizontal. A partir do meio desse segmento horizontal determina-se o segmento vertical, mas encontrando os pixels mais acima e mais abaixo, dessa mesma cor. O centro desse segmento vertical, por se tratar de um objeto circular, também será o centro do objeto em questão. Entretanto, o sistema de visão do time FutePoli (Bianchi e Reali-Costa, 2000) se baseia na cor, que pode sofrer alterações com a variação de luminosidade.

Este artigo tem por objetivo discriminar quais são os robôs móveis que devem ser detectados com maior precisão (quando a informação de ângulo é importante), que serão detectados por formas geométricas e os robôs móveis que podem ser detectados com menos precisão (os adversários) e que serão detectados por cor uma vez que podem conter qualquer forma geométrica não identifica a priori.

3. Detectando uma única forma independente da cor

Para detectar os robôs de seu próprio time, o sistema de visão proposto neste artigo utiliza-se da forma geométrica circular. Para isto, utiliza-se da transformada de Hough (Hough, 1959) e de outras técnicas de visão computacional como detecção de bordas (Canny 1986) e imagem em escala de cinza.

3.1 Detecção de Círculos com a Transformada de Hough

A Transformada de Hough (TH) (Hough, 1959) é uma técnica de segmentação de imagens utilizada para a detecção de objetos a partir do ajuste de modelos. Para detectar objetos em uma imagem a TH tenta casar as bordas encontradas na imagem com o

modelo parametrizado do objeto. A parametrização de uma classe de objetos define a forma desse objeto, portanto, variações de cor na imagem, ou até mesmo nos objetos não afetam o desempenho e a eficiência da TH.

No domínio do futebol de robôs a TH é muito pouco usada e existem sistemas que a usam apenas para localização da bola – mas não dos robôs – como descrito em Gönner, Rous, Kraiss (2000).

Círculos são uma estrutura geométrica muito comum no domínio do Futebol de Robôs, onde todos os objetos podem ser representados por um ou vários círculos. Por exemplo, a bola é uma esfera, cuja projeção na imagem capturada gera um círculo. Os robôs da categoria MiroSot (Fira, 2005) são identificados através de etiquetas em sua parte superior. O modelo de etiqueta utilizado nesse trabalho possui dois círculos dispostos a 45º em relação a frente do robô (Figura 1), onde os círculos possuem o mesmo diâmetro da bola utilizada.

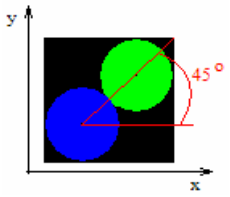

Figura 1. Modelo de etiqueta para diferenciação dos robôs

Círculos são parametrizados pela tripla (x_c,y_c,r), a qual define um conjunto de pontos eqüidistantes em r do ponto central representado pela coordenada (x_c,y_c).

O espaço de parâmetros, também chamado de espaço de Hough, é tridimensional. Como no domínio do Futebol de Robôs os objetos se movimentam apenas em duas dimensões, não existe variação de profundidade dos objetos na imagem, o que possibilita determinar um valor fixo para o raio r. Logo, o espaço de parâmetros torna-se bidimensional e é representado pela tupla (x_c,y_c).

3.2 O espaço de Hough

Detectar círculos de raio fixo em uma imagem que contém apenas pontos de borda de coordenadas (x,y) utilizando a TH consiste em determinar quais desses pontos pertencem à borda do círculo com centro em (x_c,y_c) e raio r. O algoritmo da TH determina para cada ponto de borda (x,y) da imagem um conjunto de possíveis centros (x_c,y_c) no espaço de Hough.

A Figura 2 demonstra a execução do algoritmo para 3 pontos da borda de um círculo da imagem à esquerda, enquanto o respectivo espaço de Hough gerado é mostrado à direita. Os 3 círculos de raio r desenhados no espaço de Hough, a partir de 3 pontos (x,y) de borda do círculo de centro (x_c,y_c) da imagem, se intersectam em apenas um ponto, justamente o ponto central do círculo presente na imagem.

Cada ponto de borda da imagem gera um círculo no espaço de Hough e cada ponto de borda desse círculo no espaço de Hough recebe um voto. Quanto maior o número de votos recebidos por um ponto, maior a probabilidade de esse ponto ser centro de um círculo. Esses pontos de maior probabilidade são os máximos relativos do espaço de Hough e definem os centros dos objetos existentes na imagem.

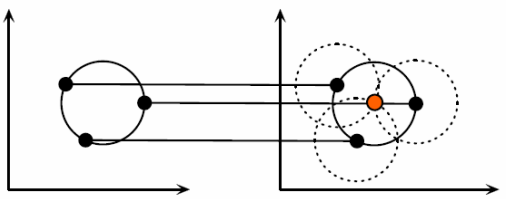

Figura 2. Exemplo de geração do espaço de Hough

3.3 Descrição do sistema de detecção de círculos implementado

O sistema de visão computacional implementado possui basicamente sete etapas, que são elas: aquisição da imagem, subtração de fundo, conversão de cores para escala de cinza, aplicação do filtro de bordas, geração do espaço de Hough, determinação de pontos com alta probabilidade de serem centros de círculos na imagem e reconhecimento dos objetos (robôs e bola).

A aquisição de imagem é feita por uma placa de aquisição de imagens com capacidade de aquisição de 30 quadros por segundo, em imagens de 640x480 pontos.

Como citado anteriormente, apenas as bordas da imagem são relevantes para a TH. Para otimizar o tempo desse algoritmo utilizou-se um método simples de subtração de fundo, descrita por Piccardi (2004), que calcula a diferença entre a imagem capturada e uma imagem do fundo, sem os objetos móveis..

Apesar de simples, esse método é eficaz para essa aplicação, porque o fundo não sofre alterações significativas. O resultado é uma imagem que contém apenas os objetos móveis, que irá resultar apenas em bordas relevantes. Como visto na Figura 3c.

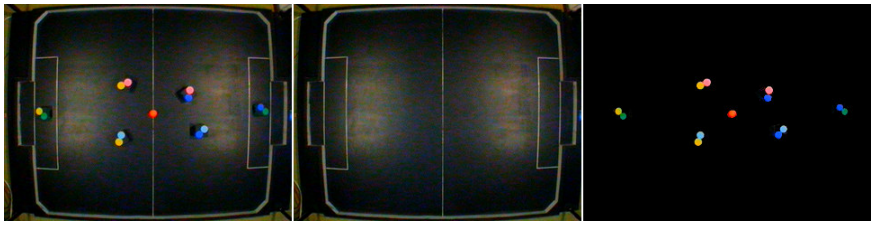

Figura 3. (a) Imagem capturada contendo uma bola e dois times de robôs completos (b) Fundo e (c) Resultado da subtração do fundo.

Após a subtração do fundo, a imagem é transformada de colorida para escala de cinza. Imagens coloridas demandam um maior tempo de processamento, pelo fato de possuírem três canais de informação de cor. Como a TH não necessita de informações de cor, para otimizar o tempo de processamento do sistema a imagem é convertida para escala de cinza. A imagem resultante contém apenas um canal de informação referente à variação de iluminação, a qual é suficiente para a determinação das bordas na imagem.

Existem diversas técnicas capazes de extrair informações de bordas em uma imagem, como as descritas por Forsyth e Ponce (2003, p.165). O presente trabalho utiliza uma técnica muito conhecida para detecção de bordas, o filtro Canny (Canny, 1986) resultando em uma imagem binária. A Figura 4 demonstra o resultado da detecção de bordas com o filtro Canny a partir de uma imagem resultante da subtração do fundo, convertida para escala de cinza.

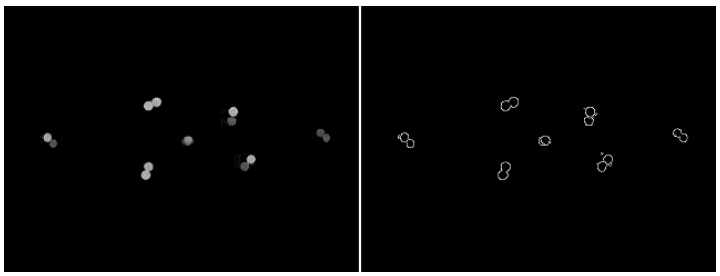

Figura 4. Resultado do filtro Canny

A partir da imagem binária resultante do filtro Canny pode-se gerar o espaço de Hough. A geração do espaço de Hough a partir do algoritmo descrita na seção 3 é eficaz, mas não é eficiente. Para tornar o algoritmo mais eficiente, e eliminar possíveis redundâncias de pixels ao gerar o círculo variando-se apenas o ângulo, utilizou-se um algoritmo de desenho de círculos proposto por Bresenham (1965). Esse algoritmo tira vantagem da simetria de pontos em um círculo, sendo necessário determinar apenas os pontos de um octante, ao invés de determinar 360 pontos referentes à variação do ângulo. Dessa maneira é minimizado o tempo de processamento necessário para a geração do espaço de Hough, permitindo a utilização em aplicações em tempo real.

Com o espaço de Hough gerado, detectar possíveis centros (x_c,y_c) de círculos de raio fixo r em uma imagem se resume a encontrar os pontos mais votados nesse espaço. No algoritmo implementado é verificado se o ponto votado ultrapassou um determinado número de votos mínimo ao mesmo tempo em que o espaço de Hough é gerado. Se um ponto ultrapassar esse limiar, ele é armazenado em um vetor, uma única vez.

Para garantir que todos os pontos que representam centros reais de círculos na imagem sejam inseridos no vetor é definido um baixo número de votos mínimo. Pelo fato de se ter um baixo número de votos mínimo, haverá pontos referentes a falsos máximos relativos presentes nesse vetor. O espaço de Hough gerado para uma imagem de teste pode ser obesrvado na Figura 5. Pode-se verificar, também, a importância da subtração de fundo para evitar que falsos máximos relativos sejam gerados.

O primeiro passo para separar os falsos máximos relativos de pontos de centros de círculo reais é a ordenação dos pontos pelo número de votos recebidos com o algoritmo quicksort. Após a ordenação, o ponto máximo global, presente na primeira posição do vetor, é considerado um centro de círculo. O ponto que possui o segundo valor máximo, presente na segunda posição do vetor, é comparado com o ponto da posição anterior. Para tal comparação é calculada a distância euclidiana entre os pontos de modo a verificar se são círculos sobrepostos (distancia entre os centros < 2*r). Se for menor, o ponto em análise é considerado um falso positivo.

Os pontos que satisfazem essas comparações, considerados como sendo centros de círculos reais, são inseridos em um novo vetor à medida que as comparações são efetuadas. Caso essas restrições sejam satisfeitas, encontrou-se um novo ponto de centro de círculo. O algoritmo continua suas iterações até que seja encontrado o número máximo de centros de círculos determinado manualmente, ou caso todos os pontos do vetor sejam verificados.

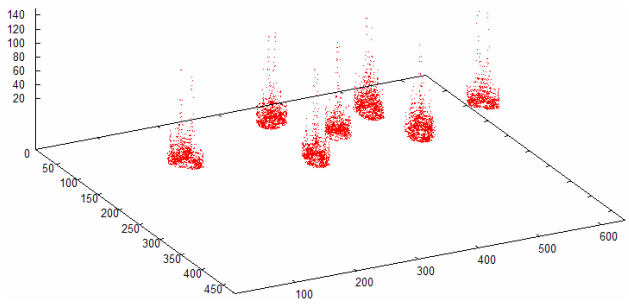

Figura 5. O espaço de Hough gerado

Esse vetor de centros é utilizado para reconhecer os objetos. Essa etapa é a única que considera informações de cor e iluminação da imagem, pois para reconhecer os objetos com sucesso é necessário distingui-los, o que pode ser feito somente a partir das cores principais, pré-determinadas pela regra, diferentes para cada equipe, assim como as cores secundárias, para diferenciar os robôs de uma mesma equipe.

Para se resolver o problema de círculos de dois robôs encostados, utiliza-se uma técnica de satisfação de restrições que resolve o problema analisando as possibilidades de cada círculo pertencer a um robô. O problema pode ser descrito como uma árvore, onde a raiz é um ponto do vetor de centros de cor primária, que representa um time, tendo como nós filhos os outros pontos do vetor de centros. Um nó, ou seja, um ponto de vetor de centros é filho da raiz caso a distância entre esse ponto e o ponto da raiz seja pequena o suficiente para considerar que não há espaço entre as duas circunferências.

Ao percorrer o vetor de centros, é determinada a primeira raiz para se resolver o problema, verificando se esse centro é de um círculo cuja cor seja primária e também se esse centro já pertence a algum objeto reconhecido. Com a raiz determinada, a árvore é construída, determinando seus filhos.

Depois de construída a árvore, é feita uma busca em largura (visto que a árvore tem apenas um nível) para verificar se existem nós filhos válidos, ou seja, com alguma cor secundária, que representa um dos robôs da equipe. A restrição do problema está no fato de que cada raiz pode ter apenas um nó filho válido. Caso a restrição seja satisfeita, é reconhecido o robô de cor primária representada pela raiz e cor secundária representada pelo nó filho, esses pontos passam a representar um objeto já reconhecido.

Tendo as informações de centro de cada círculo da etiqueta e a disposição dos círculos conhecida, o algoritmo determina a posição global (x,y) do robô na imagem e seu ângulo de direção em relação ao eixo x. Entretanto, pode acontecer de robôs estarem próximos, ou até mesmo encostados, fazendo com que uma raiz tenha mais de um nó filho com alguma cor secundária. Nesse caso nenhum robô é reconhecido e os pontos do vetor de centros ainda não representam objetos reconhecidos. O algoritmo continua a percorrer o vetor de centros até encontrar outra raiz. Ao determinar uma nova raiz, outra árvore é construída. Não há critério de parada do algoritmo, ele percorre o vetor de centros por três vezes construindo árvores com raízes e nós filhos que não representam um objeto já reconhecido.

O motivo pelo qual o algoritmo é executado por três vezes é para resolver os problemas em que alguma raiz possua mais de um nó filho de cor secundária. Pela

disposição dos círculos nas etiquetas dos robôs, observou-se que o pior caso seria uma raiz com três nós filhos válidos.

Como a bola é representada por um círculo de cor laranja e essa cor não pode ser utilizada em nenhum outro objeto, todo e qualquer círculo cuja cor seja laranja é considerado uma bola.

4. Detectando uma única cor independente da forma

Para detectar os robôs da equipe adversária, tem-se a informação da cor a priori, mas não se tem informação da forma da etiqueta utilizada pelos robôs. Além disso, pela forma ser livre, cada robô pode ter sua forma especifica. Assim, utiliza-se a técnica de detecção de objetos por cor e de vários formatos.

Esta implementação segue a idéia de Bianchi e Reali-Costa (2000) de traçar segmentos internos aos objetos detectados de uma cor especifica. Em outras palavras, o sistema percorre a imagem e, ao detectar um pixel da cor especificada, começa a traçar segmentos internos para os pixels da mesma cor na horizontal e na vertical. Para percorrer a imagem de maneira eficiente, o sistema salta pixels na horizontal e na vertical em uma imagem que o fundo foi retirado, igual dito na seção 3 anterior.

Os segmentos, por suas vezes, são traçados na horizontal, depois na metade do segmento horizontal é traçado o segmento vertical. O segmento é formado percorrendo a imagem na horizontal desde o ponto do pixel da cor específica encontrado até encontrar outro pixel que não é da cor especificada. O mesmo acontece para o segmento vertical. Esse processo de gerar segmentos na horizontal e na vertical é chamado de processamento em cruz.

Num circulo, apenas um processamento em cruz é necessário para encontrar o centro do objeto, mas em outros formatos geométricos será necessário repetir o processamento em cruz várias vezes. De fato, executa-se o processamento em cruz, recursivamente, n vezes para cada pixel da cor determinada encontrado, onde o processamento m sempre começa no ponto central do segmento vertical do processamento $m-1$ anterior. A figura 6 mostra um exemplo de processamento repetido.

Esse processo em cruz é feito diversas vezes para cada objeto, visto que percorrer a imagem de saltos em saltos permite que esta varredura de pixel caia em vários lugares de um mesmo objeto. Cada processamento em cruz gera um centro provisório do objeto. Isso não é ruim, muito pelo contrário, vários centros provisórios permitem encontrar o centro estimado mais provável do objeto com maior precisão.

Alguns formatos geométricos, como, por exemplo, o formato da figura 6, possuem diversos centros provisórios após a varredura completa da imagem. O centro do objeto será, portanto, estimado pela média dos pontos encontrados.

Essa média é uma média espacial onde cada ponto puxa o centro para próximo de si. Quanto mais pontos uma região do objeto tiver mais próximo a esta região será o centro estimado do objeto. Os pontos considerados para a média são pontos que possuem uma distância euclidiana menor do que um valor pré-estabelecido. A figura 7 mostra como a média aproxima os centros provisórios do centro estimado do objeto.

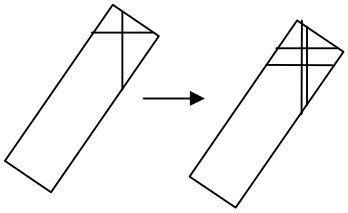

Figura 6. Processamento em cruzes: Segundo processamento em cruz a partir do centro do segmento vertical do primeiro processamento em cruz.

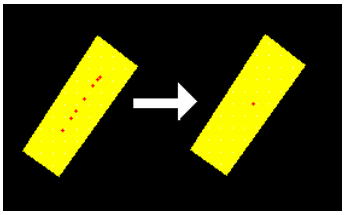

Figura 7. Resultado dos processamentos em cruz e posterior agrupamento de centros provisórios

O agrupamento de centros provisórios permite, inclusive, que imagens ruins ou com ruídos possam ainda ter seus objetos e seus centros estimados sem nenhuma dificuldade. Isso pode ser visto na figura 8b. Pode-se observar que há estimativas de centro ruins quando a imagem está com ruídos, mas o centro continua em um ponto dentro do objeto. Na figura 8b somente o quadrado mais abaixo a esquerda é que teve o centro apontado muito fora do centro real. Os demais os centros estimados foram muito próximos do centro real.

Cabe ressaltar que este método não garante encontrar o centro real de qualquer forma geométrica, ele acha apenas um centro estimado próximo ao centro real, como pode ser visto na figura 8a.

Usou-se na figura 8 a repetição do processamento em cruz 5 vezes, percorrendo a imagem de 10 em 10 pixels na vertical e horizontal e agrupamento de centros provisórios com distância menor que 70 pixels.

5. Resultados obtidos

A implementação dos dois sistemas foi feita em linguagem C++ e utilizou-se a biblioteca de visão computacional OpenCV (Intel, 2005). Os resultados apresentados foram obtidos em um computador equipado com um processador Intel Pentium® 4 HT de 3,2 GHz. O programa foi executado em sistema operacional Windows® XP, configurado para prioridade em tempo real.

Como a intenção é avaliar os algoritmos propostas para uma aplicação em tempo real, a avaliação de desempenho considera como sendo o tempo máximo aceitável para a execução do algoritmo o intervalo de tempo de uma captura de imagem.

O tempo foi medido pela média da execução das etapas em um laço de 500 vezes para assegurar uma medição de tempo mais precisa. Os parâmetros configuráveis, tais quais, limiares de histerese do filtro Canny, raio da circunferência e número de votos mínimo, foram fixados igualmente para todas as imagens testadas.

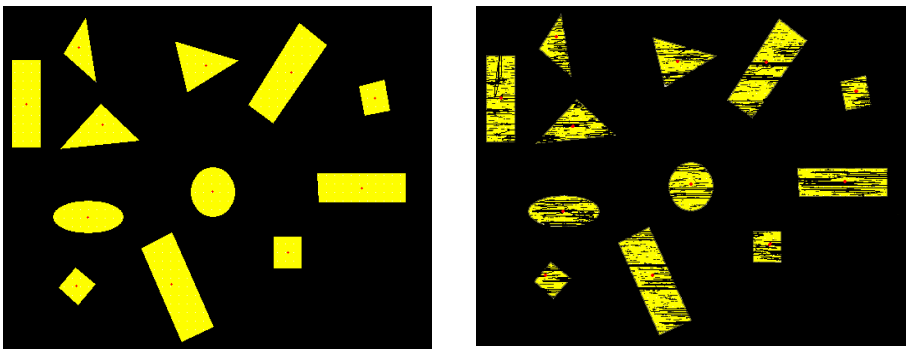

Figura 8. (a) sistema de detecção de centros estimados em qualquer forma. (b) o mesmo sistema aplicado em imagem com ruídos

Os valores apresentados estão em valores aproximados, pois há pequenas variações de processamento cada vez que o teste é executado. O desvio padrão calculado para tais variações é da ordem de 0,3 ms para a imagem de 640x480.

Para efetuar testes do sistema com a transformada de Hough, todos os parâmetros ajustáveis do sistema foram fixados. Para imagens de 640x480, o raio *r* foi fixado em 8 pixels, enquanto o número de votos mínimo fixado em 16. Os limiares de histerese do filtro Canny foram mantidos fixos para todas as imagens em valores de 75 para o limiar baixo e 150 para o limiar alto. Tais valores foram estipulados empiricamente, após algumas observações. O tempo do processamento em cruz é o tempo do pior caso obtido.

Table 1. Tempos de execução

Função	***Tempo de execução 640x480***
Subtração de fundo	*~5ms*
Conversão de cor + filtro Canny	*~8ms*
Geração do espaço de Hough	*~6ms*
Determinação de centros de círculo	*~0ms*
Reconhecimento dos objetos	*~0ms*
Processamento em cruz	*~6ms*
Total	*~25ms*

Todas as funções apresentadas na tabela 1, com exceção do processamento em cruz, são operações seqüenciais. Ou seja, pelo menos 19 ms são gastos necessariamente no processo de detecção de círculos. Já o processo em cruz pode ser processado em paralelo, caso haja necessidade de diminuir o tempo.

A eficiência do sistema em relação à variação de luminosidade foi testada variando-se a intensidade luminosa e verificando quantos círculos eram detectados a cada variação. O sistema detectou todos os 14 círculos nas 22 imagens de teste com diferentes intensidades luminosas, variando de 250 até 1300 lux, em passos de 50 lux.

6. Conclusão e trabalhos futuros

A utilização da TH mostrou-se eficiente e funcional na detecção de círculos em imagens do domínio do Futebol de Robôs. O sistema implementado mostrou-se tolerante a ruídos, além de tolerante a variação de luminosidade, pois apenas considera a forma dos objetos, utilizando informações de cor apenas para distinguir os mesmos. Assim pode-se definir as cores em um intervalo de valores muito mais aberto, em relação a sistemas de reconhecimento baseados em cores.

Já o método de detecção de objetos por cor independente da forma mostrou-se eficaz na determinação dos centros estimados de várias figuras geométricas, mesmo em imagens com ruídos.

Os tempos medidos, juntamente com os testes de reconhecimento dos objetos, demonstram que é possível utilizar o sistema descrito em tempo real, pois supera as necessidades de tempo, precisão e tolerância à variação de iluminação exigidas em um ambiente como o do Futebol de Robôs.

Como trabalho futuro fica a utilização de regiões de interesse que pode resultar em um desempenho ainda melhor do algoritmo, visto que não seria necessário processar uma imagem inteira.

Referências bibliográficas

BIANCHI, R. A. C.; REALI-COSTA, A. H. O Sistema de Visão Computacional do Time FutePOLI de Futebol de Robôs. In: CBA 2000 - CONGRESSO BRASILEIRO DE AUTOMÁTICA, 13., Florianópolis, 2000. Anais. Florianópolis, Sociedade Brasileira de Automática, 2000. p.2156-2161.

BRESENHAM, J. E. Algorithm for Computer Control of A Digital Plotter. **IBM Systems Journal,** v. 4, n. 1, p.25-30, 1965.

CANNY, J. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, v.8, n. 6, p.679-698, nov. 1986.

FIRA. FIRA Small League MiroSot Game Rules. Disponível em: <http://www.fira.net/soccer/mirosot/rules_slm.html>. Acesso em: 21 nov. 2005.

FORSYTH, D. A.; PONCE, J. Computer Vision: A Modern Approach. New Jersey: Prentice Hall, 2003.

GÖNNER, C.; ROUS, M.; KRAISS, K. Real-Time Adaptative Colour Segmentation for the Robocup Middle Size League. **Lecture notes in Artificial Inteligence**, v. 1856, p.243-256, 2000.

GRITTANI, G.; GALLINELLI, G.; RAMÍREZ, J. FutBot: A Vision System for Robotic Soccer. **Lecture notes in Artificial Intelligence**, v. 1952, p.350-358, nov.2000.

Hough, P.V.C. Machine Analysis of Bubble Chamber Pictures, International Conference on High Energy Accelerators and Instrumentation, CERN, 1959.

INTEL. OpenCV Reference Manual. Disponível em: <http://www.intel.com/technology/computing/opencv/index.htm>. Acesso: 21 nov. 2005