

IDENTIFICAÇÃO E RASTREAMENTO EM TEMPO REAL DE MÚLTIPLOS AGENTES AUTÔNOMOS

Oliver Matias van Kaick
William Robson Schwartz
Murilo Vicente Gonçalves da Silva
Hélio Pedrini

Universidade Federal do Paraná
Departamento de Informática
Curitiba-PR, Brasil, 81531-990
{omvk99,wrs99,mvgs99,helio}@inf.ufpr.br

RESUMO

Futebol de Robôs tem sido adotado internacionalmente como uma atividade científica voltada à educação e pesquisa, visando estimular o desenvolvimento de novas técnicas na área de Inteligência Artificial. O domínio de Futebol de Robôs é bastante dinâmico e complexo, consistindo de múltiplos agentes capazes de realizar ações individuais e colaborativas para a execução de uma tarefa. Este artigo descreve o sistema de visão computacional implementado em nosso time de Futebol de Robôs. A arquitetura básica do time e os algoritmos adotados para a identificação e rastreamento dos objetos são apresentados e discutidos. Um algoritmo simples e eficiente é proposto para a identificação e rastreamento em tempo real de objetos durante o jogo. Diversos experimentos reais são realizados para demonstrar o desempenho de nossos algoritmos.

PALAVRAS-CHAVE: Visão Computacional, Futebol de Robôs, Reconhecimento de Objetos.

1 Introdução

O Futebol de Robôs foi proposto por diversos pesquisadores [3, 5] como uma atividade científica que integra uma grande variedade de tópicos relacionados à educação e pesquisa na área de Inteligência Artificial, tais como visão computacional, princípios de agentes autônomos, colaboração de multi-agentes, raciocínio em tempo real e robótica. O domínio de Futebol de Robôs é bastante dinâmico e complexo, consistindo de múltiplos agentes capazes de realizar ações individuais e colaborativas através de operações e movimentos altamente coordenados para atingir objetivos específicos.

Com a internacionalização da idéia do Futebol de Robôs, surgiu a necessidade de definição de regras de modo a garantir compatibilidade entre as equipes. Como resultado, surgiu a *Federation of International Robot-soccer Association* (FIRA), a qual organiza torneios anuais e

promove o desenvolvimento de sistemas inteligentes e robôs autônomos, contribuindo para o estado da arte em áreas especializadas. Atualmente, a FIRA conta com dez categorias ou ligas, as quais diferenciam entre si com relação ao número, tamanho e forma dos robôs utilizados pelas equipes [2]. Paralelamente aos desenvolvimentos da FIRA, surgiu uma outra organização internacional denominada *Robot World Cup Initiative* (RoboCup). A RoboCup também possui diversas categorias ou ligas, dentre elas a liga de simuladores (softwares), a liga de robôs de pequeno e médio porte, robôs com pernas e a liga humanóide [4].

O projeto de Futebol de Robôs em desenvolvimento na Universidade Federal do Paraná (UFPR) iniciou-se em 1997, contando com a participação dos cursos de Informática e Engenharia Elétrica. Desde então, vários experimentos têm sido realizados visando à construção de um time completo de robôs com adequado grau de integração e robustez a um baixo custo.

Seguindo esse propósito, decidiu-se adotar as regras da RoboCup na categoria de pequenos robôs (*small*), onde cada equipe é composta de cinco robôs cujo formato deve estar contido em um cilindro com 18cm de diâmetro e 22.5cm de altura. Marcas coloridas são utilizadas para identificar os robôs de cada equipe. Em adição a essas marcas, cada robô deve carregar uma bola de ping-pong colorida que é usada como o uniforme ou camiseta do time. O campo é formado por uma superfície plana de cor verde com dimensões de 152.5cm×274cm. Uma câmera instalada a uma altura mínima de 3 metros acima do campo é utilizada para capturar imagens do jogo, que são transmitidas a um computador responsável pela interpretação dessas imagens e decisão das estratégias e táticas de jogos. As ações no domínio são realizadas através de comandos de movimentos enviados aos robôs via comunicação sem fio. Para o jogo, utiliza-se uma bola de golfe com aproximadamente 3cm de diâmetro e de cor laranja. Um esquema básico do sistema é apresentado na figura 1.

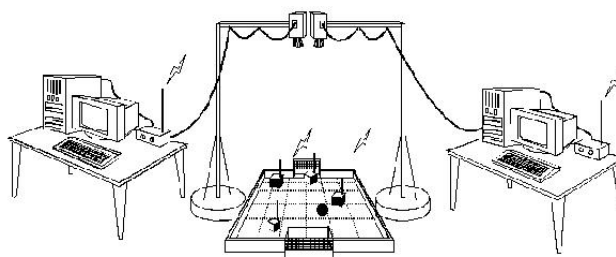


Figura 1: Esquema básico de um sistema de Futebol de Robôs (adaptado de [2]).

O sistema utilizado em nosso projeto está dividido em três módulos principais (figura 2):

- *Visão Computacional*: a partir da imagem capturada pela câmera digital, é responsável pelo reconhecimento dos robôs e da bola no campo de jogo. A determinação da posição e orientação dos robôs é realizada através de marcas coloridas colocadas sobre cada robô;

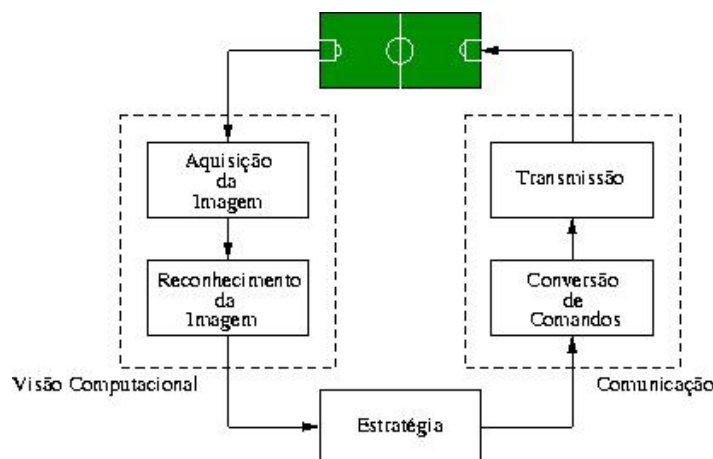


Figura 2: Módulos principais do projeto Futebol de Robôs da UFPR.

- *Estratégia*: através das informações fornecidas pelo módulo de visão computacional, é responsável pelas decisões de jogadas com relação aos movimentos dos robôs, incluindo táticas de jogo, definições das trajetórias dos jogadores e tendências de movimentação da bola. Cada robô deve ser capaz de planificar e refinar seus objetivos de uma maneira eficaz, levando em conta o contexto atual em que ele se encontra;
- *Comunicação*: converte as instruções resultantes da estratégia em comandos de movimentação para os robôs. Tendo em vista a proposta de se ter robôs autônomos, é proibido qualquer tipo de ligação aos robôs com fios.

Este artigo descreve o módulo de visão computacional implementado no time de Futebol de Robôs da UFPR. A seção 2 apresenta os algoritmos propostos para permitir a identificação e localização dos objetos de interesse (robôs e bola) durante o jogo. Algumas otimizações são descritas de modo a melhorar o desempenho do módulo de visão computacional. A seção 3 analisa o desempenho do sistema desenvolvido e apresenta alguns resultados experimentais. Conclusões e sugestões para trabalhos futuros são apresentadas na seção 4.

2 Sistema de Visão Computacional

O principal objetivo do módulo de visão computacional é criar um modelo a partir das informações sobre cada objeto encontrado na imagem, mais especificamente identificando a posição da bola e a posição e orientação dos agentes autônomos

O sistema de visão computacional, utilizando a base de conhecimento, processa a imagem capturada no momento e identifica os objetos e a bola (figura 3). A representação para a situação atual é criada, contendo as posições dos objetos e as orientações dos robôs.

As principais etapas do módulo de visão computacional implementados no time de Futebol de Robôs da UFPR consistem na aquisição da imagem, construção de uma base de conhe-

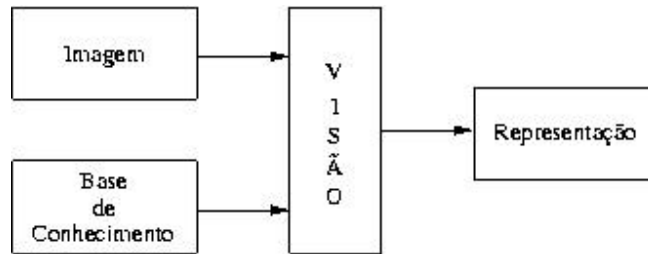


Figura 3: Reconhecimento de objetos.

cimento, reconhecimento das cores dos objetos (robôs e bola) e determinação da posição e orientação dos robôs de nossa equipe no campo de jogo. Essas etapas são descritas nas seções a seguir. imagens

2.1 Construção da Base de Conhecimento

Antes dos jogos, uma base de conhecimento é construída para que o algoritmo possa saber quantos objetos irão haver na cena e quais são as cores e características desses objetos. Essa base de conhecimento é construída atualmente por meio de uma interface gráfica controlada por um operador humano durante o período de calibração do sistema.

Nossa base de conhecimento contém as seguintes informações:

- n : número de robôs de nossa equipe;
- cor_b : cor dos pixels $p(x, y)$ da bola;
- cor_c : cor da camiseta dos robôs (cor única para todos os robôs da nossa equipe);
- $diam_c$: diâmetro da camiseta dos robôs;
- cor_m_i : cor da marca de cada robô r_i ; (uma cor para cada robô)
- $dist_c_m$: distância em pixels entre a marca e a camiseta dos robôs.

2.2 Classificação das cores

O sistema utilizado para atribuir um valor a um pixel é o espaço cromático RGB, o qual é diretamente fornecido pela nossa câmera. Para utilizar o HSI, por exemplo, seria necessário fazer uma conversão de algum outro formato disponível na câmera para este, o que requereria maior tempo de processamento. Outros formatos como o YUV carregam uma quantidade de informação bem menor (8 bits são fornecidos em nossa câmera) comparados com os 24 bits do RGB.

No sistema RGB, a cor de cada pixel é definida por três componentes (R, G, B). Cada componente define a quantidade de uma cor primária presente na cor do pixel. Logo, R define a

quantidade de vermelho, B a quantidade de azul e G a quantidade de verde. No RGB fornecido pela câmera, cada componente é definida por 8 bits, o que totaliza uma boa quantidade de informação sobre uma cor (24 bits).

As imagens utilizadas são compostas por vários objetos dispostos sobre um fundo de cor única. Cada objeto a ser reconhecido também possui apenas uma cor como atributo. Quando as imagens são convertidas para a forma digital, os pixels que definem os objetos acabam sendo de vários tons de uma mesma cor. De forma a verificar se um pixel pertence a uma classe de cor particular, o reconhecimento deve utilizar um espaço de tolerância de uma determinada cor ao invés de um valor absoluto para comparação.

Para classificar um pixel como equivalente a uma determinada cor, cada componente da cor deve pertencer a um intervalo de tolerância. Os três intervalos aos quais as componentes da cor devem pertencer simultaneamente caracterizam um volume no espaço paramétrico do sistema RGB, que é o espaço (intervalo) de tolerância de uma cor.

Para determinação deste espaço de tolerância, o usuário deve inicialmente informar um valor para cada componente do sistema RGB, o qual é feito por meio de uma interface gráfica apresentada durante a etapa de calibração. Este valor é considerado o centro do intervalo de tolerância de uma determinada componente. Os extremos superior e inferior do intervalo são obtidos adicionando-se e subtraindo-se, respectivamente, uma certa tolerância T ao valor inicial. Este valor de tolerância é determinado empiricamente, pois deve ser ajustado segundo as condições de iluminação e contraste das imagens capturadas.

Para verificar se uma determinada componente da cor de um pixel pertence ao intervalo, cujo centro é representado pelo valor cor , utiliza-se a expressão $cor - T < pixel < cor + T$, que otimizada para a computação torna-se $|cor - pixel| < T$. O teste acima é realizado para cada componente (R,G,B) da cor e, se resultar em verdadeiro para as três componentes, $pixel$ é considerado como sendo equivalente a cor , segundo a tolerância T .

De modo a tornar o processamento mais eficiente, é utilizada uma otimização baseada no uso de máscaras [1] para determinar se um valor pertence a um intervalo. Divide-se o intervalo inteiro de uma componente em p partes discretas. Cria-se então um vetor com p posições, cada posição representando um intervalo de t unidades. Determina-se o intervalo de equivalência para uma determinada componente da cor e coloca-se o valor 1 nas posições do vetor que estão dentro do intervalo e o valor 0 nas que estão fora. Para verificar se uma cor pertence ao espaço de tolerância, cada vetor é indexado com a respectiva componente e depois é realizada a operação lógica AND entre esses três elementos dos vetores. Se a operação resultar em verdadeiro, o pixel é equivalente à cor. Essa otimização pode ser realizada com vários sistemas de cores, aqui é ilustrado apenas com o sistema RGB.

Considere a amplitude das componentes

$$A_r = [I_r, S_r]$$

$$A_g = [I_g, S_g]$$

$$A_b = [I_b, S_b]$$

onde I e S são respectivamente os limites inferior e superior do intervalo de amplitude da componente.

Os intervalos de amplitude são divididos em p partes de tamanho t cada um, ou seja

$$t_r = (S_r - I_r)/p$$

$$t_g = (S_g - I_g)/p$$

$$t_b = (S_b - I_b)/p$$

Devemos criar os vetores de componentes V_r , V_g e V_b para uma determinada cor, definida pelos intervalos C_r , C_g e C_b . Os vetores resultantes são

$$V_r = \{r_1, \dots, r_k, \dots, r_p\}$$

$$V_g = \{g_1, \dots, g_k, \dots, g_p\}$$

$$V_b = \{b_1, \dots, b_k, \dots, b_p\}$$

onde r_k equivale a $[I_r + t_r k, I_r + t_r(k + 1) - 1]$, g_k equivale a $[I_g + t_g k, I_g + t_g(k + 1) - 1]$ e b_k equivale a $[I_b + t_b k, I_b + t_b(k + 1) - 1]$.

O valor r_k será 1 se o intervalo representado por ele estiver contido no intervalo da respectiva componente, caso contrário será 0 (se $[I_r + t_r k, I_r + t_r(k + 1) - 1] \subset C_r$ então r_k é 1, senão 0). As outras componentes são determinadas de maneira análoga.

As componentes de um pixel qualquer, sendo $pixel = (r, g, b)$, pertencem aos intervalos C_r , C_g e C_b se $V_r[r/p] = 1$, $V_g[g/p] = 1$ e $V_b[b/p] = 1$.

Se o valor p for uma potência de 2, as divisões r/p , g/p , e b/p podem ser substituídas por divisões na base 2, que são computacionalmente mais rápidas.

2.3 Reconhecimento e Rastreamento dos Objetos

Os objetos a serem identificados na imagem são os robôs e a bola. De modo a permitir a identificação, a localização e a orientação dos robôs de nosso time, algumas marcas são colocadas sobre cada robô. A figura 4 mostra um exemplo de marcas utilizadas na identificação dos robôs de nossa equipe.

No centro da superfície do robô localiza-se sua camiseta. A uma distância d do centro está

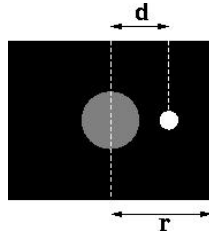


Figura 4: Marcas de identificação dos robôs.

a marca para identificação individual de cada robô e r é a distância do centro do robô até a sua extremidade.

Seja Im o conjunto de todos os pixels $p(x, y)$ da imagem, $dist(p, q)$ uma função que retorna a distância, utilizando a métrica *Manhattan*, do pixel p até o pixel q , $cent(A)$ uma função que retorna o centróide dos pixels pertencentes ao conjunto de pixels A , $orient(p, q)$ uma função que retorna o coeficiente angular do segmento orientado de reta conectando p a q e $cor(p, q)$ uma função que retorna *verdadeiro* se a cor do pixel p é semelhante à cor do pixel q e *falso* caso contrário, sendo que essa semelhança entre pixels é descrita na Seção 2.2.

Definiremos agora os objetos *bola* e *robôs* como:

- *Bola*: conjunto B de pixels da bola.
- *Robôs*: conjunto R com n elementos, onde cada elemento é um robô r_i definido pela tripla (M_i, C_i, pr_i) , tal que M_i é o conjunto dos pixels da marca deste robô, ou seja, os pixels $\{p \in Im \mid cor(p, cor_{m_i}) = verdadeiro\}$, $pr_i(x, y)$ é o primeiro pixel de r_i encontrado na varredura da imagem e C_i é o conjunto de pixels pertencentes à camiseta deste robô, ou seja, os pixels $\{p \in Im \mid cor(p, cor_c) = verdadeiro\}$ e $dist(p, pr_i) < diam_c$. O teste de distância é feito para diferenciação entre cada robô pois todos têm a mesma cor de camiseta.

Nota-se que com essas definições, mais o fato de que $d < \frac{r}{2}$ (conforme figura 4), não ocorre a possibilidade de se associar uma marca errada à camiseta de um robô, mesmo que eles estejam imediatamente adjacentes entre si.

A representação criada a partir da imagem contém:

- *posição de r_i* : denotada por c_{r_i} , esta posição é dada pelo centróide dos pixels pertencentes a camiseta de r_i , ou seja $cent(C_i)$;
- *orientação de r_i* : denotada por o_i ,
dada por $orient(cent(C_i), cent(M_i))$;
- *posição da bola*: denotada por c_b , é dada pelo centróide dos pixels do conjunto B .

A idéia básica do algoritmo é percorrer a imagem e classificar cada pixel segundo sua cor como, pixel da bola, da camiseta, da marca de algum dos robôs ou do fundo, sendo então inserido no respectivo conjunto de pixels (B, C_i, M_i) . Os pixels do fundo são descartados. Em seguida cada conjunto de pixels C_i (camisetas), é associado a um conjunto M_i (marcas) e é calculada a posição da bola e a posição e orientação dos robôs.

Seja n o número de robôs da imagem, C_tmp_i um conjunto temporário de pixels da cor da camiseta e $cont_r$ uma variável que armazena o número de camisetas encontradas até o momento. O pseudocódigo do módulo de reconhecimento de objetos é apresentado a seguir:

```

/* Inicializações */
cont_r ← 0
inicializa todos os conjuntos como vazios
/* Classificação dos pixels */
para todos pixels  $p$  da imagem
    /* Verifica se  $p$  é um pixel da bola */
    se  $cor(p, cor\_b)$ 
         $B \leftarrow B \cup \{p\}$  /* Insere o pixel em  $B$  */
    /* Verifica se  $p$  é da cor da marca de algum robô */
    para  $i$  de 1 até  $n$ 
        se  $cor(p, cor\_m_i)$ 
             $M_i \leftarrow M_i \cup \{p\}$ 
    /* Verifica se  $p$  é da cor da camiseta dos robôs */
    se  $cor(p, cor\_c)$ 
         $j \leftarrow 0$ 
        encontrou ← falso
        /* Para cada camiseta encontrada até o momento */
        enquanto  $j < cont\_r$  e não(encontrou)
            se  $dist(p, pr_i) < diam\_c$ 
                 $C\_tmp_i \leftarrow C\_tmp_i \cup \{p\}$ 
                encontrou ← verdadeiro
        /* Se não pertencer a uma camiseta existente, cria-se uma nova */
        se não(encontrou)
            cont_r ← cont_r + 1
             $pr_{cont\_r} \leftarrow p$ 
eliminar os  $cont\_r - n$  menores conjuntos  $C\_tmp_i$ 
/* Associa marcas com as camisetas temporárias encontradas */
para  $i$  de 1 até  $n$ 

```



```

para  $j$  de 1 até  $n$ 
    se  $dist(cent(M_i), cent(C\_tmp_j)) \leq dist\_c\_m$ 
         $C_i \Leftarrow C\_tmp_j$ ,
/* Encontra centro da bola */
 $c\_b \Leftarrow cent(B)$ 
/* Encontra centro dos robôs */
para  $i$  de 1 até  $n$ 
     $c\_r_i \Leftarrow cent(C_i)$ 
/* Encontra orientação dos robôs */
para  $i$  de 1 até  $n$ 
     $o_i \Leftarrow orient(C_i, M_i)$ 

```

2.4 Otimizações

Com intuito de melhorar a qualidade dos resultados e aumentar a velocidade de processamento pretendemos implementar algumas otimizações que foram estudadas e serão descritas nas seções seguintes.

Para aumento na qualidade da classificação de cores é proposto o uso de redes neurais artificiais. Para obtenção de uma maior velocidade, é proposto um método que utiliza a base de conhecimento para evitar que todos os pixels da imagem sejam percorridos.

Para aumento na qualidade da classificação de cores é proposto o uso de redes neurais artificiais. Para obtenção de uma maior velocidade, são propostos dois métodos, o primeiro usa a base de conhecimento para não percorrer todos os pixels da imagem e o segundo utilizando a representação da imagem anterior para delimitar a máxima região onde o objeto será encontrado na imagem a ser processada.

2.4.1 Classificação das cores Utilizando Redes Neurais Artificiais

Uma possível solução para eliminar a determinação empírica da tolerância utilizada para testar que cor tem um determinado pixel, é a utilização de redes neurais artificiais. Devido a característica de extrapolação, as redes neurais conseguem detectar padrões que não foram utilizados na fase de aprendizado.

A rede neural será um *perceptron* em multicamadas, pela sua capacidade de detecção de padrões. A topologia da rede será composta por três neurônios na camada de entrada, que receberão os tons R, G e B do pixel, uma camada oculta e na camada de saída o número de neurônios será igual ao número de cores diferentes a serem classificadas (Figura 5).

O treinamento da rede será realizado durante a construção da base de conhecimento, utilizando o algoritmo *backpropagation* para correção de erros na determinação dos pesos das sinapses. Na saída, o neurônio que representa esse padrão de entrada terá um valor alto e os

demais neurônios terão valores baixos.

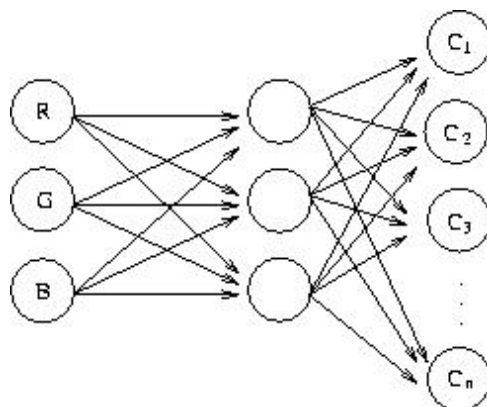


Figura 5: A topologia proposta para rede neural artificial. A entrada recebe os componentes do pixel e a saída mostra qual padrão de cor C_1, C_2, \dots, C_n a entrada representa.

2.4.2 Varredura Parcial da Imagem

Em busca de uma velocidade maior no algoritmo, uma possível solução seria a alteração do método de varredura de imagem. O método consistiria em percorrer parcialmente a área da imagem, o que é possível já que os objetos cobrem uma área que contém um determinado número de pixels. Para detectar o local onde se encontra um objeto é necessário identificar somente a coordenada de um pixel.

Para percorrer parcialmente a imagem é criada uma grade com espaçamento entre cada ponto de tal forma que os objetos sempre estejam no mínimo sobre um ponto da grade. A distância entre dois pontos p_1 e p_2 da grade deve ser menor que o diâmetro d do menor objeto a ser detectado. Define-se a seguinte fórmula para calcular a distância entre cada ponto da grade, $dist(p_1, p_2) < d$, onde $dist(p_1, p_2)$ retorna a distância entre os pontos p_1 e p_2 .

A partir da fórmula acima, é possível calcular o número de pontos que serão percorridos. Supondo uma imagem com altura de h pixels e largura de w pixels, utilizando o método que percorre toda a imagem, o número de pixels percorridos é $w * h$. Utilizando o método de varredura parcial, seriam percorridos $(w / (d - dist(p_1, p_2))) * (h / (d - dist(p_1, p_2)))$ pixels.

3 Resultados Experimentais

Alguns resultados obtidos através da utilização de nosso algoritmo são apresentados nas figuras 6(a)-(f). Para avaliar a robustez do algoritmo, foi utilizado um número variável de robôs, com orientações e posições diferentes nas imagens.

As imagens na coluna à esquerda mostram as imagens originais capturadas pela câmera, enquanto as imagens à direita mostram os resultados gerados pelo algoritmo. As imagens 6(a) e (c) contêm, respectivamente, dois e três robôs voltados para a bola; as imagens 6(b) e (d)

mostram os resultados do processamento das imagens capturadas. O sistema detectou satisfatoriamente todos os objetos, mostrando círculos brancos em torno do centro dos robôs e da bola e um segmento de reta para indicar a orientação de cada robô. Na imagem 6(e), embora os três robôs estejam bastante próximos entre si, o algoritmo identificou corretamente todos os objetos da cena.

Os algoritmos foram implementados em microcomputadores PC Pentium III 450 MHz com 128 Mbytes de RAM utilizando linguagem de programação C com o sistema operacional Linux. As imagens são capturadas por uma câmera de vídeo colorida JVC com saída em VHS e digitalizadas por uma placa PCI da PixelView PV-Bt-878. A imagem digitalizada utiliza resolução de 320×240 pixels e 24 bits de quantização em RGB.

O processamento da imagem utiliza como interface o *video4linux*, possibilitando a aquisição de imagens digitais com grande simplicidade e versatilidade, operando com taxas de 30 quadros por segundo.

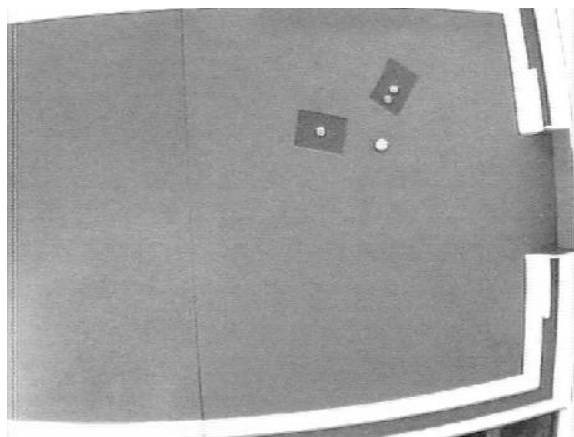
4 Conclusões e Trabalhos Futuros

Nosso sistema de visão computacional mostrou-se robusto às condições de luminosidade, sendo capaz de identificar as cores dos objetos com alta precisão. Ele permite a execução em tempo real, tal que objetos podem ser localizados e rastreados a uma taxa de 30 quadros por segundo.

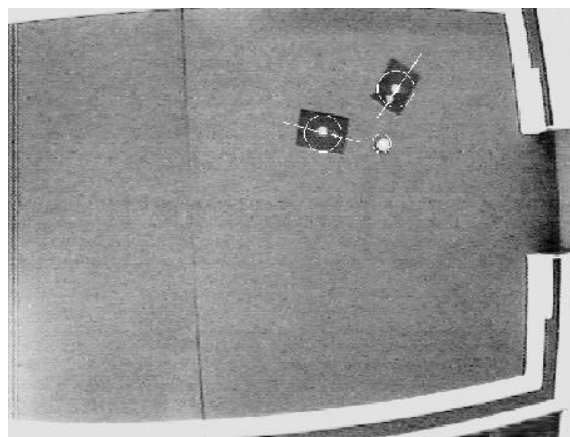
Trabalhos futuros incluem a implementação de todas as otimizações propostas anteriormente de modo a melhorar a qualidade dos resultados e aumentar a velocidade de processamento. O uso de redes neurais para a classificação de cores e localização de objetos, associado a um método de varredura parcial da imagem, devem melhorar a robustez e desempenho do atual sistema de visão computacional.

Referências

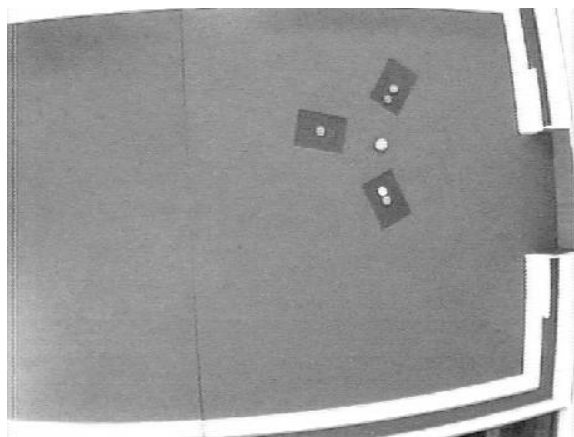
- [1] BRUCE, J., BALCH, T., AND VELOSO, M. Fast and cheap color image segmentation for interactive robots. In *Workshop on Interactive Robotics and Entertainment* (Apr. 2000).
- [2] FIRA. *Federation of International Robot-soccer Association*. <http://www.fira.net>.
- [3] KITANO, H., KUNIYOSHI, Y., NODA, I., ASADA, M., MATSUBARA, H., AND OSAWA, H. Robocup: A challenge problem for AI. *AI Magazine* 18, 1 (1997), 73–85.
- [4] ROBOCUP. *Robot World Cup Initiative*. <http://www.robotcup.org>.
- [5] SANDERSON, A. Micro-Robot World Cup Soccer Tournament (MiroSot). *IEEE Robotics and Automation Magazine* (Dec. 1997).



(a)



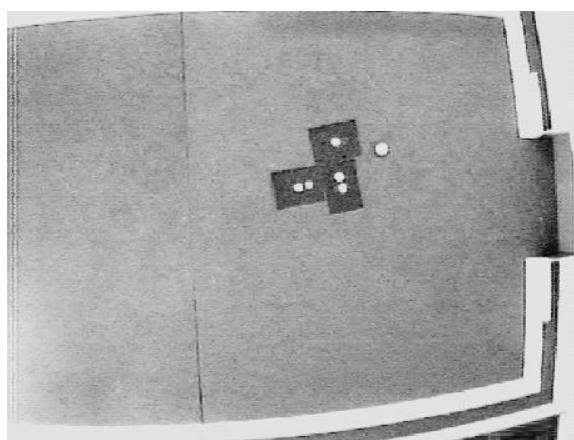
(b)



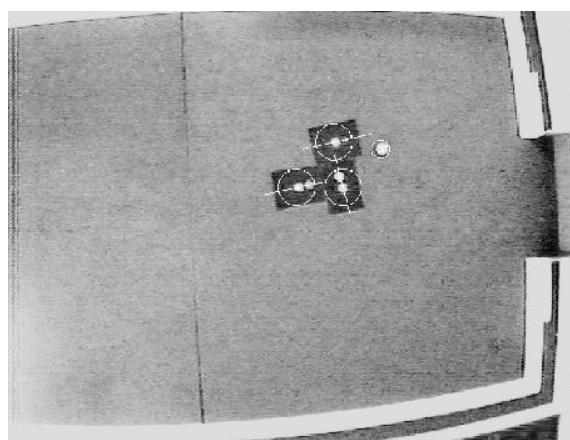
(c)



(d)



(e)



(f)

Figura 6: Exemplos do reconhecimento de objetos. As imagens capturadas pela câmera são mostradas à esquerda, enquanto as imagens à direita mostram o resultado após a detecção dos objetos.