



Exacta

ISSN: 1678-5428

exacta@uninove.br

Universidade Nove de Julho
Brasil

Alves de Araújo, Sidnei; Henriques Librantz, André Felipe
Navegação autônoma de robôs
Exacta, vol. 4, núm. Esp, novembro-special, 2006, pp. 81-83
Universidade Nove de Julho
São Paulo, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=81009913>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica
Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

Navegação autônoma de robôs

Sidnei Alves de Araújo, André Felipe Henriques Librantz

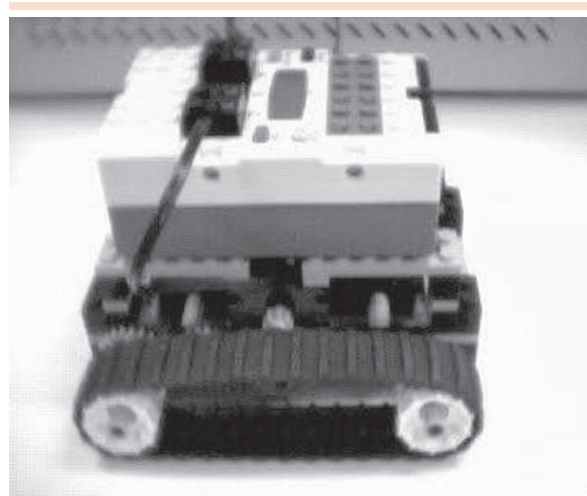
Uninove, Departamento de Ciências Exatas. São Paulo – SP [Brasil]
saraujo@uninove.br

Neste trabalho, é apresentado um sistema de navegação autônoma, baseado em mecanismos de visão computacional que descrevem o ambiente para que um robô, construído com o *kit* Lego Mindstorms, possa tomar decisões. A partir das informações recebidas sobre sua localização, a do alvo e a dos obstáculos, o robô deve definir o trajeto a percorrer para atingir o objetivo.

Robôs autônomos geralmente são objetos de grande admiração, dada a sua “inteligência” para tomar decisões de maneira independente. Entretanto, vale lembrar que os robôs móveis, mesmo com toda a tecnologia, ainda apresentam muitas limitações quanto à sua capacidade de navegação. O estudo e o desenvolvimento de um projeto de robôs autônomos podem estar ligados a diversas aplicações práticas e têm suscitado grandes desafios, visto que as dificuldades se multiplicam à medida que o ambiente no qual o robô deve movimentar-se vai-se tornando mais imprevisível e diversificado. Nesse contexto, inúmeros mecanismos para o controle de robôs móveis têm sido utilizados, tais como sonares, *lasers* e visão (QUILES; ROMERO, 2004; POMERLEAU, 1995).

O objetivo deste trabalho é explorar o desenvolvimento de um sistema de navegação autônoma de robôs, utilizando o *kit* Lego Mindstorms, baseado em mecanismos de visão computacional. A partir do sistema de visão, informações do ambiente são transmitidas ao robô para que ele possa tomar decisões. As informações recebidas pelo robô referem-se à sua localização, à do alvo e à dos obstáculos dispostos aleatoriamente no ambiente.

Com tais informações, o trajeto é calculado por um algoritmo de busca que, a partir de funções heurísticas, tenta escolher o melhor caminho, considerando as particularidades do ambiente. Para possibilitar a navegação autônoma do robô, construído com o *kit* Lego Mindstorms (Fotografia 1), uma câmera de vídeo é posicionada sobre o ambiente a aproximadamente 2,6 metros (m) acima do cenário (Ilustração 1), proporcionando uma visão panorâmica.



Fotografia 1: Foto do protótipo do robô montado. Seu formato e cor facilitam distingui-lo dos demais objetos do cenário

Fonte: Os autores.

A altura escolhida para a captura das imagens mostrou-se satisfatória, pois todos os componentes do cenário ficaram bem definidos. A partir das imagens capturadas pela câmera, o sistema computacional deve identificar, usando coordenadas cartesianas, a posição dos obstáculos, do robô e o local de destino.

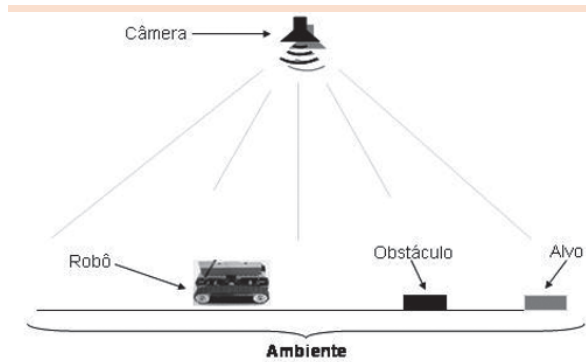


Ilustração 1: Arranjo esquemático do cenário constituído. O robô de cor amarela, o alvo de cor azul e obstáculos de cor preta estão dispostos em um piso de tonalidade cinza

Fonte: Os autores.

Quando o sistema de visão é acionado, uma imagem colorida, denotada por C^y , com resolução de $320 \times 240 \text{ pixels}$, é capturada e armazenada. A partir de C^y , primeiro se detectam as posições do robô e do alvo, dada a facilidade de localizá-los a partir dos componentes RGB (do inglês *red, green, blue*) dos *pixels* que os compõem (QUILES; ROMERO, 2004; HACHOEN; COHEN, 2002a, 2002b). Na etapa seguinte, é feita a localização dos demais objetos (obstáculos). Para isso, a imagem em tons de cinza relativa ao componente R da imagem C^y , denotada por G_R^y , é extraída (Ilustra-

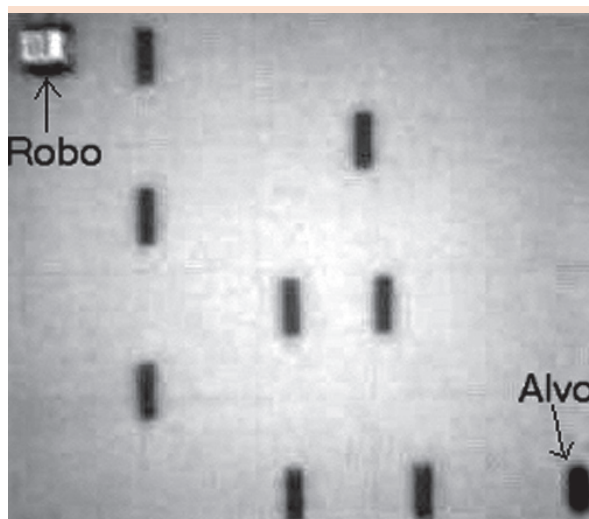
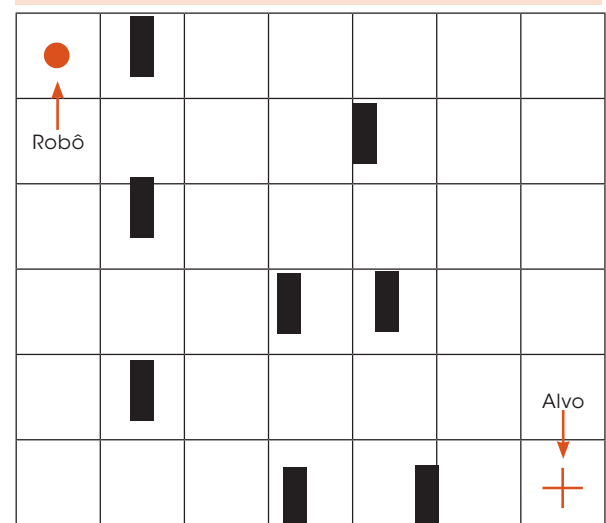


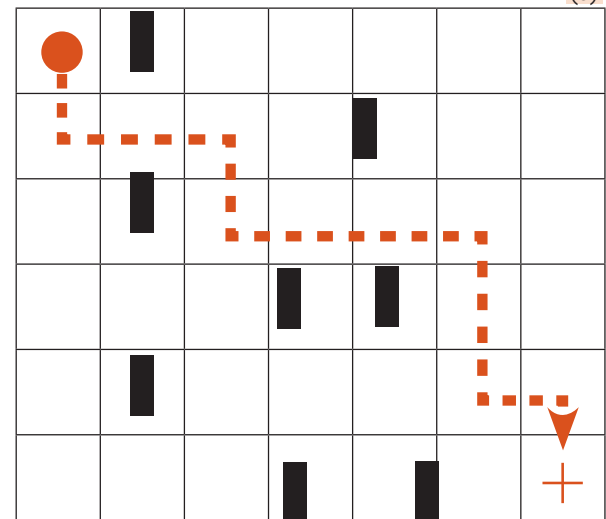
Ilustração 2: Imagem com variações de cinza G_R^y

Fonte: Os autores.

ção 2). A imagem G_R^y é filtrada (filtro gaussiano) e binarizada, utilizando um limiar (*threshold*) 80 e originando a imagem B^y (Ilustrações 3a e 3b).



(a)



(b)

Ilustração 3: Ambiente binarizado: imagem binarizada com os componentes do cenário (a); trajeto do roteamento do robô que será descrito posteriormente (b)

Fonte: Os autores.

A próxima etapa do processamento é a transformação da imagem binária, denotada por B^y , em uma matriz $M^{(i,j)}$. Para cada célula desta matriz é atribuído um valor inteiro V , que assume os valores 0 (livre), 1 (obstáculo), 2 (robô) e 3 (alvo).

Essa matriz é, posteriormente, utilizada pelo algoritmo de roteamento na determinação do trajeto do robô, que é calculado por meio de um algoritmo de busca no espaço de estados, conhecido como busca pela melhor estimativa que emprega a função heurística $f(n) = h(n)$. Utilizamos $h(n)$ para três diferentes medidas de distância, conhecidas como *city-block*, *chessboard* e distância euclidiana (GONZALEZ; WOODS, 2002), as quais são dadas, respectivamente, pelas equações 1, 2 e 3.

$$d_b(p,q) = |x - u| + |y - v| \quad (1)$$

$$d_c(p,q) = \max(|x - u|, |y - v|) \quad (2)$$

$$d_e(p,q) = \sqrt{(x - u)^2 + (y - v)^2} \quad (3)$$

A rota do robô é então calculada pelo algoritmo de busca e transformada em pseudocódigo, que é transmitido, por infravermelho, para acionar o robô. Verificou-se que as três heurísticas apresentaram resultados muito próximos; no entanto, as baseadas nas distâncias *city-block* e euclidiana, em geral, convergiram para soluções melhores (no cálculo da rota mostrada na Ilustração 3b, utilizou-se a distância *city-block*). O tempo de respos-

ta do sistema (aproximadamente 1 segundo [s]) mostrou-se adequado para a aplicação, visto que sistemas de navegação autônoma de robôs requerem rapidez na tomada de decisões. Em trabalhos futuros, pretende-se implementar um algoritmo de roteamento híbrido para melhorar a qualidade das rotas, usando-se os algoritmos de busca tradicionais e os genéticos.

Referências

- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. 2. ed. Nova Jersey: Prentice Hall, 2002.
- HACOHEN, A.; COHEN, Y. *Vision based pursuing of moving vehicle from bird's view – part 1*. Haifa: Israel Institute of Technology, 2002a.
- HACOHEN, A.; COHEN, Y. *Vision based pursuing of moving vehicle from bird's view – part 2*. Haifa: Israel Institute of Technology, 2002b.
- POMERLEAU, D. Neural network vision for robot driving. In: ARBIB, M. A. *The handbook of brain theory and neural networks*. 1. ed. MIT Press, 1995.
- QUILES, M. G.; ROMERO, R. A. F. Um sistema de visão computacional baseado em cores aplicado ao controle de um robô móvel. In: CONGRESSO BRASILEIRO DE COMPUTAÇÃO, 4., 2004, Itajaí. *Anais...* Itajaí: 2004, p. 379-383.

Para referenciar este texto

ARAÚJO, S. A. de; LIBRANTZ, A. F. H. Navegação autônoma de robôs. *Exacta*, São Paulo, v. 4, n. especial, p. 81-83, 25 nov. 2006.