

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228724805>

Um Sistema de Visão Computacional para Robôs Móveis

Article · January 2006

CITATIONS

0

READS

371

3 authors, including:



Eduardo Oliveira Freire

Universidade Federal de Sergipe

68 PUBLICATIONS 441 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



A new mobile robot control architecture via control output fusion [View project](#)



Automatic code generation of SIMUROSOT game strategies: an approach based on finite state machines [View project](#)

Um Sistema de Visão Computacional para Robôs Móveis

Beatriz Trinchão Andrade¹, Leonardo Nogueira Matos², Eduardo Oliveira Freire³

Grupo de Pesquisa em Inteligência Artificial do Curso de Engenharia Elétrica

^{1,2}Departamento de Ciência da Computação e Estatística

³Departamento de Física – Núcleo de Engenharia Elétrica

Universidade Federal de Sergipe – Cidade Universitária “Professor Aloísio de Campos”, Avenida Marechal Rondon, s/nº, São Cristóvão – SE, 49100-000

{beatriz, leonardo}@dcce.ufs.br efreire@fisica.ufs.br

Abstract. *Vision is one of human beings' most powerful senses. A great scientific challenge is to implement it in robots due to the complexity of the image interpretation process. This article describes the development, implementation and testing of a computer vision system for mobile robots. The aim of this system is to provide to the mobile robot controller sufficient visual information for its locomotion in dynamic environments.*

Keywords: *Computer Vision, Image Processing, Mobile Robots.*

1. Introdução

A visão é um dos sentidos mais poderosos do ser humano. Além da percepção de imagens, ela possibilita a extração de informações sobre o que é visto pela retina. Devido à complexidade que envolve a interpretação de informações visuais, reproduzir a visão em robôs tem sido um grande desafio para a ciência. É nesse contexto que surge a Visão Computacional, área que estuda a simulação da visão humana na tarefa de extrair características de informações visuais.

Neste projeto foi desenvolvido um sistema de visão computacional capaz de extrair imagens do ambiente de operação de um robô móvel a partir de duas câmeras e processá-las, a fim de obter informações sobre o robô, tais como sua posição, orientação e detecção de obstáculos. O sistema foi implementado na linguagem C++, utilizando o ambiente C++ Builder 5 [Borland 2006].

Vale ressaltar alguns trabalhos semelhantes a este, porém que utilizam técnicas de segmentação diferentes. Souza (2003) propõe um sistema de visão computacional baseado na limiarização das cores dos componentes da imagem através das proporções de seus componentes vermelho, verde e azul. Já Penharbel *et al* (2004) utilizam os componentes vermelho, verde e azul dos pixels para obter o tom, saturação, e a intensidade das cores, a fim de efetuar a limiarização sobre este último espaço de valores. Em ambos os trabalhos, o robô, o fundo e os obstáculos têm cores pré-definidas. No sistema de visão desenvolvido neste projeto, as cores são definidas automaticamente, durante a execução do sistema.

Este artigo está organizado como descrito a seguir. A Seção 2 apresenta o sistema de visão computacional, sua estrutura, implementação e teste. Os resultados

obtidos são mostrados na Seção 3. Na Seção 4 são apresentadas as conclusões deste trabalho e apresentada a continuação deste projeto.

2. O Sistema de Visão Computacional

Um das propriedades mais relevantes deste sistema é o fato de que ele opera extraindo imagens a partir de duas câmeras. Uma delas é posicionada perpendicularmente ao plano de operação do robô (câmera 1); e outra de forma inclinada (câmera 2). O interesse em uma segunda câmera com posição inclinada é, além de tornar a análise da cena mais robusta, tornar possível a detecção de obstáculos finos e alongados verticalmente, o que poderia ser tratado como ruído e passar despercebido no sistema de visão da câmera 1. Devido ao posicionamento da câmera 2, as imagens capturadas por ela apresentam distorção, e para corrigi-la foi criada uma rotina de correção que é aplicada sobre todas as imagens advindas desta câmera.

Devido a limitações do sistema operacional Windows, onde não é permitida a captura simultânea de imagens a partir de duas câmeras, foi concebida uma divisão do sistema em três partes: o sistema de visão da câmera 1, o sistema de visão da câmera 2 e o sistema receptor. Os sistemas de visão extraem as informações das imagens, e estas são guardadas em uma estrutura de dados e enviadas via *sockets* para o computador onde o sistema receptor é executado. Futuramente o sistema de controle será inserido neste último sistema, que pode rodar em qualquer uma das duas máquinas ou em uma terceira. A configuração utilizada pelo projeto é ilustrada na Figura 1.

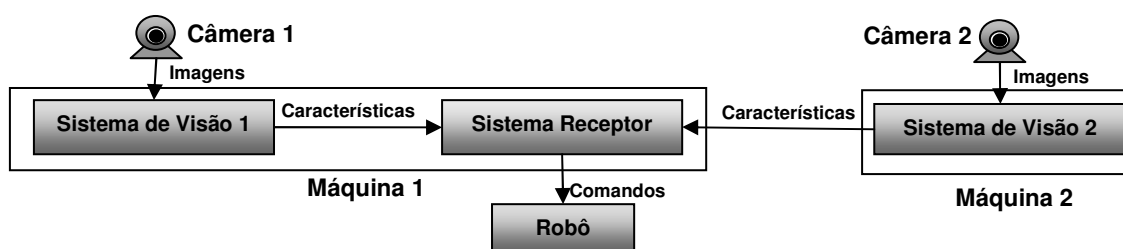


Figura 1. Representação do funcionamento do sistema com duas câmeras.

Com exceção da rotina de correção, o processamento efetuado nas imagens das duas câmeras é semelhante (Figura 2): Inicialmente a imagem capturada pela câmera é pré-processada pelo sistema, visando eliminar seu fundo. Para isso foi utilizada a técnica de limiarização global [Gonzalez e Woods 1992]. No pré-processamento das imagens seguintes, a cor a ser tida como padrão durante a rotulação dos pixels é definida como a média temporal de todas as cores obtidas até então. Essa medida evita que mudanças bruscas no ambiente afetem o funcionamento do sistema.

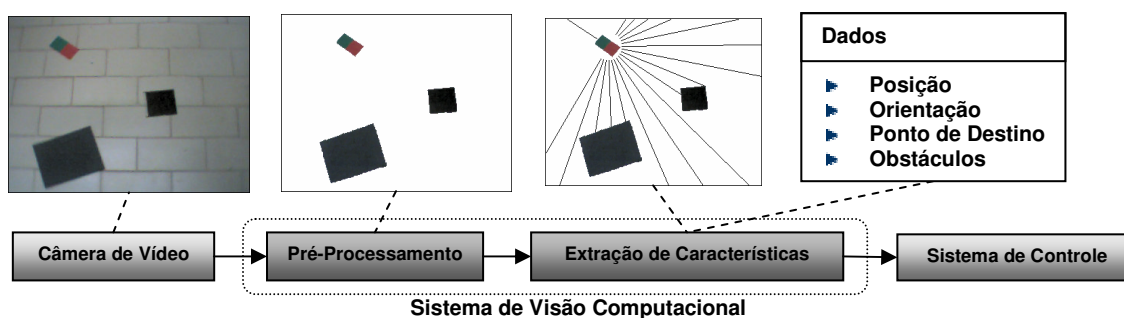


Figura 2. Processo de extração de características.

A segmentação do robô também utiliza a limiarização global. São segmentados os pixels cuja distância no espaço tricromático à cor de frente ou à de fundo do robô (definidas pelo usuário na inicialização do sistema) é inferior a um limiar.

Com o robô segmentado, é possível obter suas características. A área do robô é calculada pelo somatório dos seus pixels; sua posição é definida como sendo as coordenadas de seu centro de área [Horn 1986]; e sua orientação é definida como o ângulo formado pela semi-reta que tem como origem o centro de área da região do fundo do robô e passa pelo centro de área da região de sua frente.

O processo utilizado na verificação de obstáculos é baseado em um sonar visual, onde são utilizados 18 feixes que saem do centro de área do robô em diferentes ângulos e vão percorrendo a imagem pixel a pixel. A varredura de cada feixe termina quando ele encontra um obstáculo ou quando atinge um dos limites da imagem. As coordenadas dos pontos de parada de cada feixe são guardadas em um vetor, que é passado para o sistema de controle juntamente com as características do robô.

3. Resultados Obtidos

O sistema de visão foi testado em robôs apenas com a câmera 1 ativa, pois o sistema de controle ainda não realiza o tratamento de sinais de visão binocular. O estudo sobre a experiência pode ser encontrado em [Carvalho *et al* 2005].

Por ser executado em tempo real, o sistema depende do tempo de execução de suas iterações. Cada iteração é a sequência de ações que abrange desde a aquisição da imagem até o movimento do robô, e deve ser executada em no máximo 100 ms. O tempo médio gasto pelo sistema de visão durante cada iteração é de em média 35 ms sem a rotina de correção de imagens para câmera inclinada, e 45 ms com a rotina de correção.

Nas tabelas a seguir são feitas comparações entre os dados reais da cena e os obtidos pelo sistema de visão computacional em ambientes com luminosidade boa e ruim. Para quantificar a luminosidade, foi capturada uma imagem da cena vazia. A imagem foi então convertida para tons de cinza e foi calculado o valor do centro de área do histograma da imagem. O histograma foi normalizado para ter um y máximo de 255. Os valores próximos a (0,0) indicam uma imagem mais escura, e valores próximos a (255,255), uma imagem mais clara. Assim, com iluminação ruim, o centro de área do histograma foi o ponto (88,72.58), e com a iluminação boa, o ponto (138,68.34).

Foram medidos pelo sistema os valores do centro de massa e da orientação do robô sob os dois tipos de luminosidade, e comparados com os valores reais da cena. A Tabela 1 mostra os resultados destas comparações, que são a média das distâncias Euclidianas entre os centros de massa calculados e os reais; e a média das diferenças angulares (valor absoluto), no caso da orientação.

Tabela 1. Avaliação das características do robô sob diferentes condições de iluminação.

Condições de iluminação	Características do robô	
	Diferença média entre o centro de massa real e o calculado pelo sistema	Diferença média entre a orientação real e a calculada pelo sistema (valor absoluto)
Boa	$\cong 1$ pixel	$\cong 0,91^\circ$
Ruim	$\cong 2$ pixels	$\cong 1,62^\circ$

Foi avaliada também a eficiência do sistema na percepção de obstáculos. Para isso, foram definidas três circunferências de raios diferentes na imagem, onde o centro de massa do robô figurava como o centro. Foram testados três objetos diferentes, dispostos em diversas posições ao longo das circunferências, a fim de verificar os números máximo e mínimo de feixes que interceptavam os objetos em cada circunferência. As diferentes condições de iluminação não interferiram na detecção de obstáculos, mantendo-se idênticas nos testes com iluminação boa e ruim. A Tabela 2 mostra os resultados desta experiência, e pode ser interpretada como o resultado com imagem boa e com imagem ruim, pois foram semelhantes.

Tabela 2. Avaliação da eficiência do sistema de visão na percepção de obstáculos, semelhante nas diferentes condições de luminosidade.

Objeto \ Distância	$d_1 = 0,3m$	$d_2 = 0,6m$	$d_3 = 0,9 m$
Objeto 1 (13 X 13 cm)	Máx=2 Min=2	Máx=2 Min=1	Máx=1 Min=0
Objeto 2 (22 X 22 cm)	Máx=4 Min=3	Máx=2 Min=2	Máx=2 Min=1
Objeto 3 (28 X 28 cm)	Máx=5 Min=4	Máx=4 Min=3	Máx=2 Min=1

4. Conclusões

Neste trabalho foi implementado um sistema de visão para robôs móveis, onde as características do ambiente são obtidas através do processamento de imagens extraídas a partir de duas câmeras.

O sistema se adapta bem a diferentes intensidades luminosas, a mudanças bruscas no ambiente, e a robôs com outras cores e formatos, desde que eles possuam cores de frente e fundo diferentes.

Está sendo feita a migração do sistema de visão computacional para a plataforma Linux, com o objetivo de atingir um tempo de execução menor para o sistema de visão.

Referências Bibliográficas

- Borland, “Borland IDE: C++ Builder”. [on-line] Disponível na Internet via WWW. URL: <http://www.borland.com/br/products/cbuilder/index.html>. Última visita em Março de 2006.
- Carvalho, E. A. N, Andrade, B. T., Costa, B.B., Oliveira, E.O., Matos, L. N. (2005) “Desenvolvimento de um Sistema de Controle para Navegação de um Robô Móvel Baseado em Fusão de Sinais de Controle e Realimentação Visual”. In: VII Simpósio Brasileiro de Automação Inteligente (SBAI), São Luiz.
- Gonzalez, R. C., Woods, R. E. (1992) “Digital Image Processing”. Addison-Wesley Publishing Company.
- Horn, B. K. P. (1986) “Robot Vision”. McGraw-Hill.
- Penharbel, E. A., Destro, R. C., Tonidandel, F., Bianchi, R. A. C. (2004) “Filtro de Imagem Baseado em Matriz RGB de Cores-Padrão para Futebol de Robôs”. In: XXIV Congresso da Sociedade Brasileira de Computação, vol. 1, p.1765. Anais. SBC, Salvador.
- Souza, C.A. (2003) “Desenvolvimento de um Sistema de Visão Computacional para Aplicações de Controle Servo-Visual de Robôs Móveis”. Monografia. Universidade Tiradentes.