

Etapas de visão estereoscópica para robótica móvel⁽¹⁾

Vilson Heck Junior⁽²⁾; Alan Lanceloth Rodrigues Silva⁽³⁾;
Eneidir Guimarães de Oliveira Junior⁽³⁾; Maycon Kevin Ribeiro⁽³⁾.

Resumo Expandido

⁽¹⁾ Trabalho executado com recursos do Edital PIBIC-EM 2013/2014, do CNPq e da Pró-Reitoria de Pesquisa do IFSC

⁽²⁾ Professor, Instituto Federal de Santa Catarina, Lages, SC, vilson.junior@ifsc.edu.br;

⁽³⁾ Estudante, Instituto Federal de Santa Catarina, Lages, SC.

RESUMO: Neste trabalho são apresentados resultados na continuação de um projeto de pesquisa desenvolvido por alunos do Campus Lages do IFSC. Assim como no ano anterior, este trabalho é dividido em dois módulos individuais: processamento de imagens estereoscópicas e navegação robótica. Para cada um dos módulos são apresentados conceitos e alguns resultados obtidos. Por fim, são apresentadas sugestões para a continuação deste trabalho.

Palavra Chave: Visão Estéreo, Robótica Móvel e Controle Remoto.

INTRODUÇÃO

A Robótica Móvel envolve diferentes áreas tecnológicas, para Pio et al. (2006) ela é definida como a ligação inteligente entre a percepção e a ação. Sua importância pode ser percebida com base nas suas aplicações que, segundo Gaspar (1994), atualmente vão desde as associadas à substituição do homem em tarefas repetitivas, não interessantes, susceptíveis de erro, degradantes ou subutilizadores das suas capacidades e sensibilidade, até as associadas à assistência a pessoas inválidas ou idosas, e vigilância ou operação em ambientes hostis (ex.: ambientes radioativos, zonas profundas dos mares, espaço, etc).

A visão estereo é uma fonte de sensoriamento através de câmeras digitais. Duas câmeras arranjadas lado a lado, visualizando uma mesma cena. A pequena variação espacial na percepção faz com que a maioria dos pontos que foram capturados por uma câmera, também sejam capturados pela outra câmera (BRADSKI, 2008). Apesar da maioria dos pontos estarem presentes em ambas imagens das câmeras, assumindo as imagens como planos cartesianos, os pontos estarão em posições ligeiramente diferentes, em relação à origem. Esta distância cartesiana, entre os pontos, é inversamente proporcional à distância em que o ponto se encontra fisicamente do par de câmeras.

Dentro do contexto da robótica móvel e dando continuidade ao trabalho iniciado em um ano anterior, nosso trabalho pretende implementar um robô móvel com navegação autônoma. Para tomar suas decisões o robô deverá reconhecer o ambiente com base nas informações provenientes da visão

estéreo. Na etapa atual, estamos ainda fazendo implementações intermediárias envolvendo processamento de imagens e testes de controle remoto.

METODOLOGIA

Para atingir o objetivo, dividimos o projeto em duas partes, cada uma desenvolvida por uma equipe: visão estereo e controle remoto. A equipe de controle remoto trabalhou com aperfeiçoamento de um *software* cliente Android e a reescrita de um servidor (para rodar no robô móvel) na linguagem C++. O trabalho focou em melhorias de usabilidade e correção de falhas no cliente Android, para oferecer melhores opções de controle sobre o robô móvel. A tradução do *software* servidor da linguagem Java para C++ se deve à futura integração com funções e bibliotecas de processamento de imagens implementadas nesta linguagem.

No processamento de imagens foi escolhido um algoritmo para correspondência de pontos homólogos em imagens estereo. Este algoritmo foi testado e aprimorado para futura integração em robôs móveis. O algoritmo de correspondência escolhido é apresentado por Oliveira e Wazlavick (2005) e trabalha com indexação de regiões, apresentando resultados em tempo linear de computação.

RESULTADOS E DISCUSSÃO

Os resultados e discussão são apresentados conforme os grupos de atuação, mencionados anteriormente.

Controle Remoto

Tivemos a oportunidade de trabalhar um aplicativo para o sistema operacional Android, que tem como objetivo fazer a comunicação entre o celular e um servidor em C++.

O sistema operacional Android foi escolhido por ser o sistema operacional para dispositivos móveis muito difundido, além de ter uma plataforma livre para o desenvolvedor, sem muitas restrições e custos. O principal objetivo do aplicativo é disponibilizar a interação entre o usuário e o aplicativo, onde o usuário não tenha dificuldade para operar ou configurar o aplicativo.

Na Figura 1, podemos observar botões para seis operações básicas na tela de controle. As operações: *start*, *stop* (botão azul no meio da tela), para frente, para trás, girar à direita e girar à esquerda (botões na cor laranja) são a base para o controle de movimentação de um robô simples. O quadro preto define a área para exibição da câmera embarcada no robô móvel.

Figura 1 – Tela do cliente Android para controle remoto.

Para efetuar e manter a conexão com o servidor, foi necessário criar duas *Threads*, uma para enviar informações para o servidor e a outra para receber o fluxo de vídeo. As bibliotecas para Android oferecem várias formas de trabalhar com *Thread*, sendo optado, para esse projeto, em utilizar a classe *Runnable*, e criação de uma interface de controle chamada de *IOConnection*, com as operações *Start*, *Stop*, *Update* e *Send*. Destas quatro operações, a operação *Start* tem a função de estabelecer a conexão com o servidor, assim

iniciando um *socket*; a operação *Update* tem a função de buscar atualizações das informações enviadas do servidor para a aplicativo; a operação *Stop* tem a tarefa de encerrar a conexão entre o aplicativo e o servidor; e a operação *Send* tem a funcionalidade de enviar informações para o servidor. A Figura 2 demonstra a interface *IOConnection*.



Figura 2 – A interface *IOConnection*.

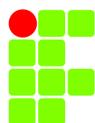
Visão Estéreo

A partir do método de correspondência proposto por Oliveira e Wazlavick (2005), trouxemos sugestões de melhorias para o método de correspondência em tempo linear de computação. Publicamos o trabalho com mais detalhes em Heck Junior e Stemmer (2013).

O método original trabalha construindo um vetor que armazena posições de regiões da imagem direita, onde o índice é construído com base em informações extraídas da imagem, ou seja, o índice é uma descrição da região. Ao mesmo tempo em que se percorre a imagem direita, é percorrida a imagem esquerda. Para cada região visitada na imagem esquerda, é gerado um índice usando o mesmo modelo da imagem direita. Quando um determinado índice da imagem esquerda coincidir com um índice de uma região já observada na imagem direita, estas regiões serão ditas homólogas.

Em nosso trabalho, transformamos o vetor de regiões em uma espécie de tabela de espalhamento. Nesta tabela de espalhamento, temos um vetor (semelhante com o do trabalho original) que em invés de armazenar a posição de uma região, armazena uma fila de posições. Esta sugestão partiu do problema onde duas (ou mais) regiões de uma mesma imagem poderiam gerar o mesmo índice, antes de serem correlacionadas com alguma região da outra imagem.

A comparação de acertos entre a proposta original com nossa proposta é apresentada na Tabela 1. Esta comparação foi feita com a



metodologia e imagens disponibilizadas pelo trabalho Scharstein e Szeliski (2002).

Tabela 1 – Resultados comparativos: percentual de pontos homólogos corretamente encontrados

Número de Imagens	Método Original	Método Proposto
27	15,09%	18,26%

CONCLUSÕES

A conexão e controle remoto entre um dispositivo computacional e um dispositivo Android pode ser feita através de *sockets* desde que ambos tenham conectividade através de redes. Este controle traz a possibilidade de monitorar o robô móvel, mesmo que ele esteja trabalhando em modo autônomo.

O algoritmo de correspondência estéreo proposto efetua a detecção de pontos homólogos acertando em média 3% a mais pontos que a proposta original e em tempo viável para implementação em um projeto de tempo real (para a navegação do robô).

As próximas etapas sugeridas são de estudo sobre métodos de navegação e mapeamento para robôs móveis, bem como emprego dos conhecimentos já estudados por este trabalho.

AGRADECIMENTOS

Os autores agradecem ao Instituto Federal de Santa Catarina, especialmente ao Campus Lages, pelo suporte às atividades e também ao CNPQ apoio concedido.

REFERÊNCIAS

BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer Vision with the OpenCV Library**. O'Reilly Media, 2008.

GASPAR, J. **Visão para robótica móvel: Detecção de obstáculos sobre pavimento plano**. 1994. Tese de Doutorado. Master thesis in Engenharia Electrotécnica e de Computadores, IST.

HECK JUNIOR, V.; STEMMER, M. R. **Linear-Time Computation of Segment-Based Stereo Correspondence**. In: Computing and Automation for Offshore Shipbuilding (NAVCOMP), 2013 Symposium on. 2013, p. 63-68.

PIO, José Luiz de Souza; CASTRO, Thais Helena Chaves de; JÚNIOR, Alberto Nogueira de Castro. **A robótica móvel como instrumento de apoio à aprendizagem de**

computação. In: Anais do Simpósio Brasileiro de Informática na Educação. 2006. p. 497-506.

OLIVEIRA, M. F. A. de; WAZLAVICK, R. S. **Linear complexity stereo matching based on region indexing**. In: Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI'05. 2005. p. 181-188.

SCHARSTEIN, D.; SZELISKI, R. **A taxonomy and evaluation of dense two-frame stereo correspondence algorithms**. International Journal of Computer Vision, IEEE Computer Society, v. 47, p. 7 #42, 2002.