

Navegação autônoma e detecção visual usando uma plataforma robótica de baixo custo

Danilo V. França¹, Leandro N. Couto¹, Jefferson R. Souza¹

¹Universidade Federal de Uberlândia (UFU)
Av. João Naves de Ávila, 2121 - Uberlândia - MG - Brasil

{danilofranca, leandronc, jrsouza}@ufu.br

Abstract. *This work shows a mobile robot platform and application of robot low cost in a task of visual detection and chasing an object, using a smartphone as a sensor and performing the object detection from Computer Vision. Also, a comparison will be made with another simpler technique, showing that the proposed method presents comparable results while offering much greater flexibility.*

Resumo. *Este trabalho apresenta uma plataforma robótica e aplicação de baixo custo de um robô em uma tarefa de detecção visual e perseguição de um objeto, utilizando um smartphone como sensor e realizando a detecção do objeto através de Visão Computacional. Os resultados são comparados com os de uma técnica tradicional da robótica móvel, mostrando que o método proposto mostra resultados comparáveis enquanto oferece flexibilidade muito maior.*

1. Introdução

A robótica móvel vem se desenvolvendo nos últimos anos com o melhoramento da tecnologia dos sensores e da capacidade computacional. Apesar disso, há ainda muitos desafios abertos para serem explorados. Dados visuais são ricos em informações, mas difíceis de interpretar computacionalmente. Os sensores visuais têm se tornado cada vez mais acessíveis, especialmente com os *smartphones*. E as câmeras de alto custo podem oferecer alta resolução de imagens, mas metodologias que tirem proveito de uma simples câmera de um celular são importantes por promover a inclusão digital.

Devido ao aumento do poder computacional dos *smartphones* e a grande adoção pelas pessoas, surge uma possibilidade de avanço na área da robótica, no sentido de se usar o crescente poder computacional dos *smartphones*, obtendo grandes resultados. O objetivo é propor uma plataforma de robô de baixo custo que tira proveito da câmera e capacidades computacionais de um *smartphone*, e demonstrar que a configuração proposta é eficaz em uma tarefa crucial de Robótica Móvel, de detectar e seguir um objeto.

Este trabalho visa promover a acessibilidade de aplicações de robótica com visão computacional, propondo uma arquitetura de baixo custo que pode ser usada inclusive em ambientes educacionais, mas ainda assim versátil para aproveitar o grande poder computacional dos *smartphones*. Ao democratizar o acesso à tecnologia, fomentam-se as oportunidades no avanço do conhecimento na área.

Este artigo está organizado da seguinte forma: na seção de Trabalhos Relacionados serão apresentados trabalhos correlatos da tarefa de robótica. Na seção de Metodologia, apresentaremos a nossa arquitetura de baixo custo. Na seção de Experimentos, descrevemos os experimentos usados para validar a proposta. A seção de Conclusão discute a contribuição do trabalho e oferece sugestões sobre os desafios a serem explorados.

2. Trabalhos Relacionados

[Oros and Jeffrey 2013] mostraram a amplitude de projetos que podem ser desenvolvidos usando um robô com *smartphone* e o sistema operacional Android. Os robôs foram capazes de concluir as suas tarefas com bom desempenho, mostrando que são uma solução viável. [Mendoza 2014] propôs a movimentação de um robô autônomo usando só visão computacional. Esse sistema obteve resultados satisfatórios, capaz de guiar o robô.

[Fusco and Assis 2015] aprimoram os *hardwares* e *softwares* de um robô autônomo que se movimenta com o auxílio de uma câmera de forma semelhante ao sistema humano. O sistema permite a proximidade com um sistema eficiente de sensoramento. E [Moretto and Mello 2020] desenvolveram um algoritmo que permitiu um robô se localizar e navegar de forma autônoma em um ambiente com pisos táteis usando visão computacional. Assim, apresenta uma plataforma robótica para localização e navegação de robôs.

3. Metodologia

A intenção da proposta era promover uma plataforma de baixo custo, que oferecesse funcionalidades de Visão Computacional. A plataforma robótica proposta é baseada em um kit robótico para Arduino, que consiste em uma estrutura de acrílico e metal sustentando quatro rodas motorizadas, e os sensores, periféricos e microcontrolador Arduino. Cada uma das quatro rodas do robô tem seu próprio motor, e cada motor pode ser ativado na polaridade invertida para girar no sentido contrário. Isto torna o robô holonômico, capaz de rotacionar em torno de si mesmo (ativando os motores de cada lado em sentidos opostos) uma característica muito desejável para navegação em ambientes internos, que frequentemente tem pouco espaço disponível para manobras [Romero et al. 2014].

O Arduino é um microcontrolador de baixo custo, de interface simples e com um grande número de periféricos compatíveis. Apesar de suas vantagens, o Arduino é insuficiente para aplicações mais complexas, como as que envolvem processamento de imagens, devido às limitações de memória. Por isso, kits de robótica controlados por Arduino são comuns em aplicações que envolvem outros sensores.

Para solucionar tanto a necessidade de câmera quanto capacidade de processamento, propusemos usar um *smartphone* com sistema operacional Android em conjunto com o Arduino. Propusemos um apoio simples ao telefone celular, que é de fácil remoção, sem conexão cabeada nenhuma com o restante do robô. Isso satisfaz um dos requisitos importantes do sistema, mantendo o custo baixo e qualquer usuário pode utilizar o robô com seu próprio *smartphone*. Esta comunicação é realizada através de um módulo Bluetooth instalado no Arduino e da antena Bluetooth presente em todos os celulares.

O sistema proposto envolve dois softwares distintos: o aplicativo que executa no telefone Android e o software que executa no Arduino. O aplicativo Android usa a biblioteca de visão computacional OpenCV, integrada com o sistema Android através do SDK OpenCV4Android. O OpenCV oferece implementações de diversos métodos de extração de regiões, pontos de interesse ou características descritivas da imagem. Neste trabalho, o algoritmo *Blob detection* (detecção de região de interesse) que segmenta regiões de uma imagem onde *pixels* contíguos sejam similares em valor de cor.

A cada quadro de vídeo capturado, ocorre a execução do algoritmo de detecção de

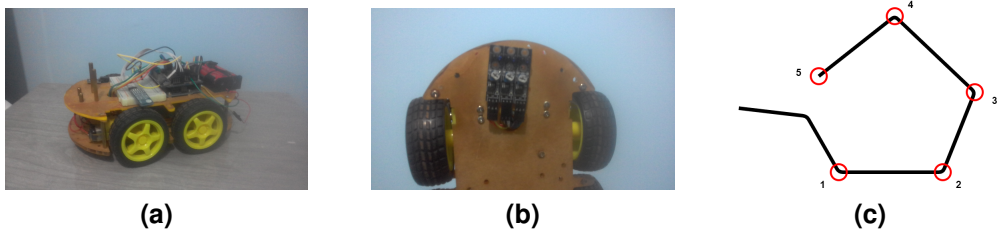


Figura 1. (a) Plataforma robótica móvel utilizada nos experimentos. (b) Módulo sensor infravermelho. (c) Cenário proposto para a realização dos experimentos.

região de interesse da cor como cor do objeto alvo. A região é detectada pelo algoritmo de *Blob detection*, e seu tamanho pode ser medido em *pixels*. Como as dimensões reais do objeto alvo são conhecidas, a partir da altura e largura da região em *pixels*, é possível estimar a distância da câmera ao objeto detectado através de semelhança de triângulos. Pela posição horizontal da região detectada na imagem capturada podemos determinar se o robô está seguindo na direção do objeto. Se a câmera do *smartphone* perde o objeto alvo de vista, o robô deve rotacionar na direção em que o objeto alvo foi avistado pela última vez na imagem. Por fim, o robô pára quando chega a uma certa distância do objeto. Com essa heurística de navegação, e decide-se o comando de navegação a ser enviado ao robô.

O aplicativo envia o comando de navegação para o módulo Bluetooth instalado na placa Arduino. Para realizar essa conexão, usamos o *Serial Bluetooth Terminal* [Morich 2021]. No caso do *software* do Arduino, o programa aguarda os comandos chegarem pela conexão Bluetooth. Os comandos são caracteres correspondentes a “seguir em frente”, “girar para a esquerda”, “girar para a direita” e “parar”. O programa lê o comando, executa e espera um tempo determinado antes de ler o próximo comando. Para contornar essa oscilação, além da implementação do tempo de espera, o aplicativo faz uma média das distâncias obtidas a cada 15 imagens para ter uma medida mais precisa.

4. Experimentos e Resultados

A metodologia proposta de detecção de objetos (OD) foi comparada com uma abordagem de robô seguidor de linha (LF). O seguidor de linha é uma aplicação que gera bons resultados, mas apresenta a limitação de depender de uma linha. Para o experimento com *line follower*, foram usados três sensores de infravermelho, localizados na parte dianteira inferior do robô (**Figura 1b**). Através das leituras dos sensores, o robô pode localizar a linha, e ajustar sua direção de navegação. A tarefa usada na comparação foi o percurso de um circuito em forma de pentágono com cerca de 60 centímetros de aresta (**Figura 1c**). No caso do seguidor de linha (LF), robô só deveria percorrer o pentágono, passando pelos vértices de 1 a 5, enquanto no caso da proposta (OD), o robô deveria localizar objetos em cada um dos 5 *checkpoints* de 1 a 5, aproximando de cada objeto alvo até colidi-lo.

Para cada abordagem foram realizadas 20 testes. Para o LF, foi marcado na tabela o *checkpoint* onde o robô parou, seja por ter chegado ao final do percurso (c5), ou por sair do caminho e não conseguir retornar. O *checkpoint* registrado é o último alcançado, o tempo. Para o OD, foi realizada a marcação dos *checkpoints* onde o robô autônomo colidiu com o objeto em questão. Nos casos onde não ocorreram colisões, não foi feita a marcação. Com o OD, o robô chegou ao final do percurso em todas as vezes, com

Tabela 1. Resultados dos experimentos

LF	Tentativas	Último checkpoint alcançado	Tempo(s)	OD	Tentativas	Colisão com o objeto	Tempo(s)
	1	c3	10,19		1	c3	55,83
	2	c5	10,21		2	c3	75,71
	3	c5	10,05		3	-	106,62
	4	c2	16,45		4	c3	-
	5	c5	10,84		5	-	112,97
	6	c5	11,23		6	c4	78,61
	7	c5	10,98		7	-	68,51
	8	c5	10,32		8	-	78,06
	9	c5	18,41		9	c4	82,8
	10	c4	18,81		10	-	52,4
	11	c5	19,73		11	-	60,86
	12	c5	8,88		12	-	57,22
	13	c5	9,08		13	c3	68,74
	14	c5	9,47		14	c2	61,58
	15	c5	11,31		15	c3	78,64
	16	c5	9,53		16	c5	66,22
	17	c5	9,54		17	-	90,77
	18	c1	12,28		18	-	100,81
	19	c2	30,87		19	c1	71,92
	20	c5	11,3		20	-	73,25

exceção em uma tentativa conforme a (**Tabela 1**), em que o robô autônomo perdeu o objeto de vista e não conseguiu recuperar o rastreamento, e nesse caso o tempo não foi registrado.

5. Conclusão

O método seguidor de linha é notoriamente veloz, e superou o OD em relação ao tempo, porém é um método sem flexibilidade, que restringe a movimentação do robô autônomo a seguir a linha, e que exige uma manipulação prévia do ambiente na demarcação das linhas bem definidas. Somado a isso, falhas no LF são catastróficas; se o robô sai da linha, é muito difícil retomar o percurso, e o LF não foi capaz de chegar ao último *checkpoint* em 5 das 20 tentativas realizadas em nossos experimentos, contra 1 de 20 do OD proposto.

Assim, o OD é mais versátil, mais tolerante a falhas e bastante acessível conforme os experimentos realizados. Um resultado importante deste trabalho foi propor uma arquitetura de baixo custo de visão computacional com um kit de robótica baseado em Arduino e um celular Android, e demonstrar ser possível realizar com ela tarefas complexas de processamento de imagens e navegação robótica. Assim, o custo do kit e a acessibilidade do celular Android usado permite que ele seja utilizado em salas de aula e aplicações multi-robôs, que são uma possibilidade interessante para trabalhos futuros. Já que aplicações multirrobôs apresentam alto custo para serem implementadas. O desempenho das câmeras de diferentes celulares Android na aplicação de detecção e rastreamento de objetos também é uma análise futura com potencial interessante.

Referências

- Fusco, R. J. and Assis, W. O. (2015). Desenvolvimento de um sistema de visão para robô.
- Mendoza, D. (2014). Computer vision applied to autonomous robots using raspberry pi.
- Moretto and Mello (2020). Protótipo de um robô seguidor de piso tátil utilizando visão.
- Morich, K. (2021). Serial bluetooth terminal.
- Oros and Jeffrey (2013). Smartphone based robotics: Powerful, flexible and inexpensive robots for hobbyists, educators, students and researchers. *CECS Tech Report 13-16*.
- Romero, R., Prestes, E., Osório, F., and Wolf, D. (2014). Robótica móvel, editora Itc.