

IMPLEMENTAÇÃO DE UM ROBÔ PARA COMPETIÇÃO BASEADO EM VISÃO COMPUTACIONAL

Junior Aguilar de Amorim, Vitória Maciel de Carvalho, Arthur de Souza Silva Mayerhofer - Técnico em Automação Industrial Integrado ao Ensino Médio¹,

Matheus Kuster Rosa - Técnico em Informática para a Internet Integrado ao Ensino Médio¹,

Eduardo Max Amaro Amaral¹

eduardomax@ifes.edu.br

¹ INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO ESPÍRITO SANTO
Serra – ES

Categoria: ARTIGO BÁSICO

Resumo: As olimpíadas de robótica aguçam a curiosidade de alunos, promovendo um conhecimento mais aprofundado na área tecnológica, sendo de extrema importância para a formação acadêmica e profissional dos jovens participantes. Por outro lado, o uso de robôs baseados em realimentação por visão computacional é cada vez mais comum em várias aplicações. Este artigo apresenta a implementação de um robô autônomo seguidor de linha baseado em visão computacional. O sistema de visão proposto é capaz de identificar, em tempo real, linha e objeto. O objetivo é simular a realização de um resgate por meio de uma garra acoplada ao robô, situação típica em competições de robótica. Utilizando o sistema proposto foi possível identificar o objeto e as linhas durante a simulação de uma competição, com boa precisão.

Palavras Chaves: Robótica, Educação, Visão computacional, Competição.

Abstract: Robotics olympiads sharpen students' curiosity, promoting a deeper knowledge in the technological area, being extremely important for the academic and professional training of the young participants. On the other hand, the use of robots based on feedback by computational vision is increasingly common in several applications. This article presents the implementation of an autonomous robot line follower based on computer vision. The objective is to simulate the accomplishment of a rescue by means of a claw coupled to the robot, typical situation in robotics competitions. Using the proposed system it was possible to identify the object and the lines during the simulation of a competition, with good precision.

Keywords: Robotics, Education, Computer Vision, Competition.

1 INTRODUÇÃO

A visão computacional vem sendo apresentada como uma área de pesquisa em grande desenvolvimento, já que imagens estão em todo o lugar e a capacidade de reconhecer objetos, paisagens, rostos, sinais e gestos torna as máquinas muito mais úteis [Antonello, 2018].

De acordo com a BMVA [2018], os seres humanos usam seus olhos e seus cérebros para ver e sentir visualmente o mundo à sua volta, assim a visão computacional possui aplicação em

várias áreas, podendo ser usada em robôs industriais, veículos autônomos, detecção de eventos aplicados ao esporte e em diversas aplicações médicas (reconhecimento de doenças, reconhecimento de instrumento cirúrgico para ser usado como parte de uma enfermeira robótica, entre outras) [Malamas et. al, 2003], [Skinner e Johnson-Roberson, 2016], [Kachore, 2017], [Zhou e Wachs, 2017], além de gerar uma proximidade ao ser humano por permitir sensações parecidas as experimentadas por um de seus sentidos mais importantes, a visão.

Em Ferreira et. al [2016], os autores demonstram um sistema de perseguição ao alvo baseado em visão computacional desenvolvido em um projeto de robótica educacional. O presente trabalho apresenta uma abordagem semelhante, onde o robô desenvolvido possui duas webcams, uma na parte inferior que detecta o objeto e outra na parte superior que realiza a detecção da linha para que o robô saiba o caminho que deverá percorrer. O intuito principal deste projeto foi a criação de um robô autônomo que se locomove para localizar um alvo (vítima) e realizar o seu resgate, através de visão computacional. Para isso, foram utilizados dois sistemas embarcados, Arduino e Raspberry PI 3, famosos por sua praticidade e simplicidade no modo de programar. Este robô possui um sistema seguidor de linha e um sistema de detecção e captura de um objeto (resgate de uma vítima) o que possibilita a simulação de um resgate, desafio comum na OBR (Olimpíada Brasileira de Robótica – <http://www.obr.org.br/>).

Os resultados, após simulações em laboratório, demonstram que o sistema proposto foi capaz de identificar o objeto e as linhas durante a simulação de uma competição, com boa precisão.

Este artigo encontra-se organizado da seguinte forma: a seção 2 apresenta a plataforma robótica. A seção 3 descreve o sistema proposto de detecção e resgate de uma vítima para um robô autônomo seguidor de linha baseado em visão computacional. Os experimentos e resultados são apresentados na seção 4. A seção 5 descreve uma breve discussão sobre os resultados, e as conclusões são apresentadas na seção 6.

2 PLATAFORMA ROBÓTICA

O chassi utilizado neste trabalho foi desenvolvida por alunos do LARSE (Laboratório de Robótica e Sistemas Embarcados), IFES, Campus Serra.

2.1 Arquitetura física

A arquitetura física define os componentes que formam o robô e como estão interligados. O chassi é formado por acrílico servindo de suporte para os componentes robóticos. A estrutura é composta por quatro servomotores, sendo dois de rotação contínua, na parte frontal sendo dois ligados as rodas, um à garra e outro ligado ao braço, uma roda boba na parte traseira e duas câmeras (*webcams*) USB. Uma plataforma Arduino Mega foi utilizada para o controle dos servomotores. Também foi adotada a plataforma Raspberry PI 3 que foi utilizada para o processamento da imagem das duas *webcams*, além de estar conectada ao Arduino. A conexão entre essas duas plataformas, Arduino e Raspberry PI 3, é necessária para que seja estabelecido um sistema de comunicação através de troca de mensagens pela porta serial com as indicações necessárias para o sistema de movimentação do robô. Além disso, uma bateria de lipo na parte inferior está ligada a Raspberry, aos motores e ao Arduino fornecendo 12,5V e regulada para alimentar a placa com 5V pelos reguladores de tensão LM7805. A Figura 1 demonstra a plataforma robótica utilizada neste trabalho. A Figura 1(a) demonstra uma vista frontal da plataforma robótica. A Figura 1(b) mostra uma vista lateral da plataforma robótica.

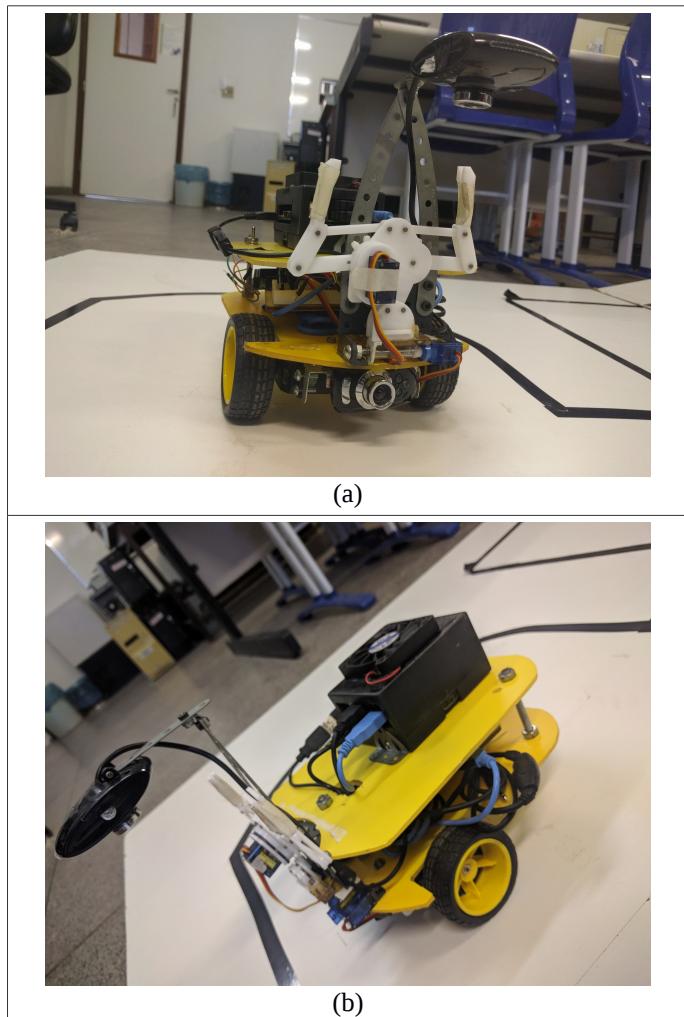


Figura 1 – Plataforma robótica utilizada neste trabalho. (a) demonstra uma vista frontal da plataforma robótica. (b) mostra uma vista lateral da plataforma robótica.

2.2 Controle básico de movimentos e sensoriamento

As funções básicas responsáveis e disponíveis para a movimentação do robô são: “irParaFrente” (movimento para

frente), “irParaTrás” (movimento para trás), “paraMotor” (para o robô), “virarEsquerda” (movimento para esquerda), “virarDireita” (movimento para direita). Estas funções foram embarcadas na placa micro controladora (Arduino ArduinoMega). O sensoriamento básico é realizado pelas câmeras.

3 SISTEMA DE DETECÇÃO DE LINHA E DE DETECÇÃO DE OBJETO

Este trabalho utiliza a visão computacional com intuito de permitir a mobilidade do robô e a detecção de objeto. A Figura 2 mostra o diagrama da arquitetura do sistema do robô e seu funcionamento de forma teórica.

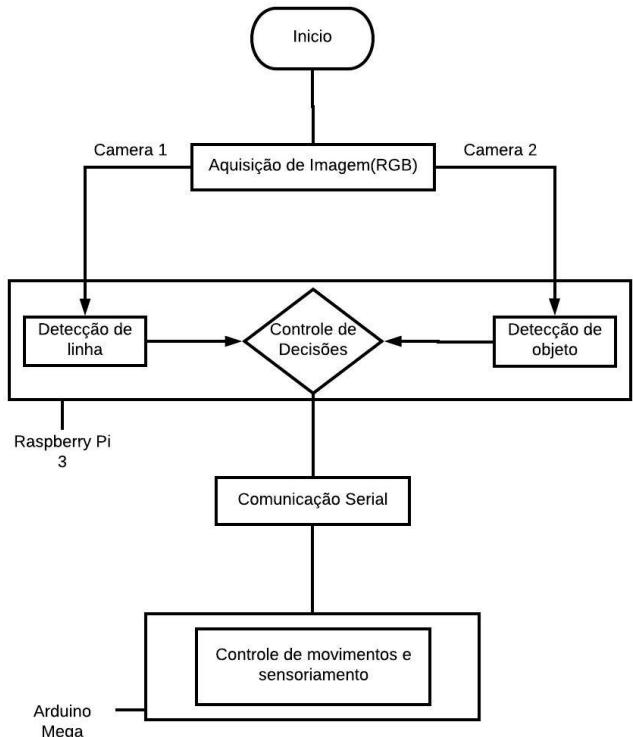


Figura 2 – Arquitetura lógica do Sistema.

Após a obtenção das imagens RGB pela câmera, o algoritmo de processamento é executado na Raspberry Pi 3. De acordo com a missão do robô é ativado o sistema de detecção de linha ou o sistema de detecção de objeto. Após a execução de um desses sistemas (dependendo da missão) comandos são enviados através da porta serial ao Arduino, que possui o sistema responsável pelo controle. Assim, o Arduino encaminha aos motores a definição da direção que o robô deve percorrer, seja para seguir a rota (linha) ou ir em direção ao objeto.

A análise das imagens são realizadas por um algoritmo baseado na biblioteca OpenCV (*Open Source Computer Vision Library*). OpenCV é uma biblioteca, de código aberto, desenvolvida inicialmente pela Intel. Ela implementa ferramentas de interpretação de imagens, indo desde operações simples como um filtro de ruído, até operações complexas, tais como a análise de movimentos, reconhecimento de padrões e reconstrução em 3D [OpenCV, 2018].

3.1 Detecção de linha

O sistema de detecção de linha exerce a função de reconhecer o trajeto por meio de uma linha na cor preta. Para seguir a linha, o centro de massa e os pixels mais externos são usados para

enviar ao robô qual direção ele deve seguir para continuar em cima da linha (direita, centro, esquerda) e informar se a curva é uma curva fechada ou não (curva fechada a direita, curva fechada a esquerda).

3.2 Detecção de objeto

Na detecção de objeto esse trabalho utiliza uma abordagem por espaço de cores HSV/HSI (*Hue, Saturation and Value / (Hue, Saturation and Intensity)*) e descritores.

As imagens (*frames*) capturadas em RGB são enviadas para a Raspberry Pi 3. Desta forma, o algoritmo trata toda a informação fornecida pela câmera guardando-a em um dicionário, que tem os *pixels* como índice. O RGB de cada *pixel* é convertido para HSV e salvo em outro dicionário com o nome “HSV”.

Em seguida, somente a posição dos *pixels* que tem a cor predeterminada são salvos em um terceiro dicionário, chamado de “mask”. Este terceiro dicionário consiste em um mapa binário em que se torna verdadeiro a posição desses *pixels*.

São realizados alguns tratamentos de imagens com intuito de eliminar quaisquer ruídos na imagem e suavizar a forma do que foi salvo em mask. Esse passo é essencial, pois muitos *pixels* podem ser da cor predeterminada, mas não fazem parte do objeto alvo ou os *pixels* agrupados que indicam um objeto podem não ser encontrados por completo. Com os tratamentos de imagem aplicados esses problemas podem ser resolvidos. As funções utilizadas podem ser definidas como:

- *cvtColor*: converte os *pixels* RGB da imagem em HSV, usando COLOR_BGR2HSV;
- *inRange*: procura os *pixels* da cor predeterminada;
- *bitwise_or*: junta duas máscaras caso seja passado mais de um intervalo RGB;
- *mophologyEx*: extrai as informações relativo às características de uma imagem e compara com um outro conjunto menor e de forma já conhecida;
- *medianBlur*: suaviza a imagem usando um filtro mediano.

Após a detecção, o robô é deslocado até o objeto, aciona a garra e resgata o objeto. Em seguida, um sistema de detecção do local a ser deixado o objeto é executado e o robô se desloca até o ponto de

4 EXPERIMENTOS E RESULTADO

Para validar o modelo proposto, foram feitos testes no laboratório, em ambiente controlado, conforme vídeo: <https://youtu.be/UbmaNRNIFUo>

Para validar o modelo apresentado foi montada uma arena similar a utilizada na OBR (Olimpíada Brasileira de Robótica), conforme Figura 3. O ambiente é composto por uma área de percurso anterior à rampa, uma rampa e uma sala de resgate. Na área de percurso anterior à rampa Foi feito um circuito com fita isolante preta, onde o robô tinha a tarefa de seguir a linha até um determinado local (sala de resgate), e após, localizar o objeto (vítima), resgatá-lo e levá-lo à uma região pré determinada (área de resgate). A área de resgate é um triângulo retângulo com lados de 30 cm x 30 cm.

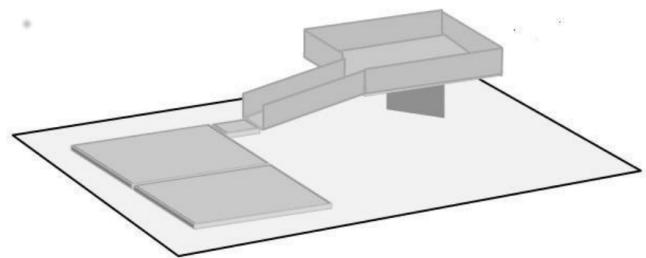
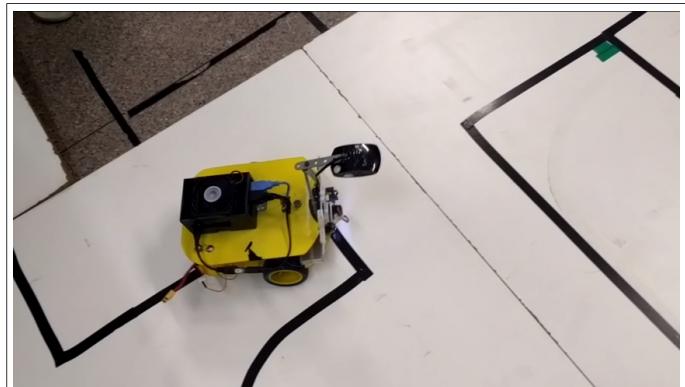


Figura 3 – Ilustração da arena utilizada nos testes.

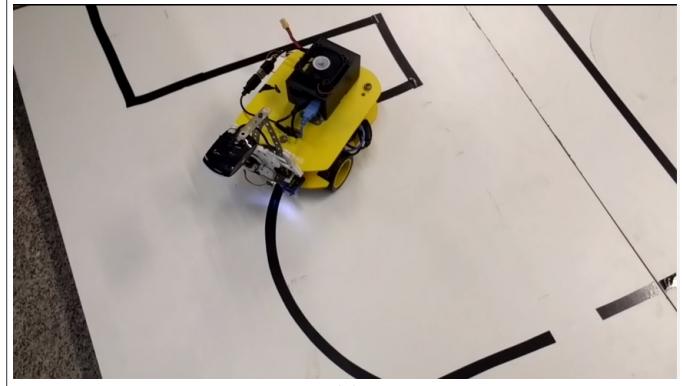
A Figura 4 demonstra uma sequência de funcionamento do sistema detectando e seguindo a linha. Na Figura 4(a) demonstra o posicionamento do robô, no tempo t. A Figura 4(b) mostra o robô no tempo t + 1. A Figura 4(c) demonstra o posicionamento do robô no tempo t + 2. E por fim, a Figura 4(d) mostra o posição do robô no tempo t + 3.



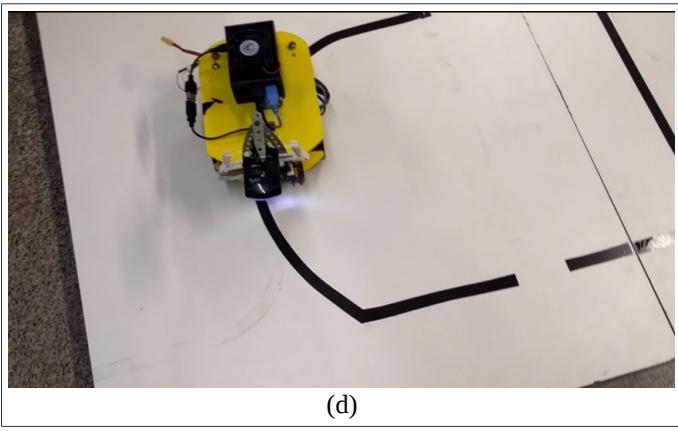
(a)



(b)



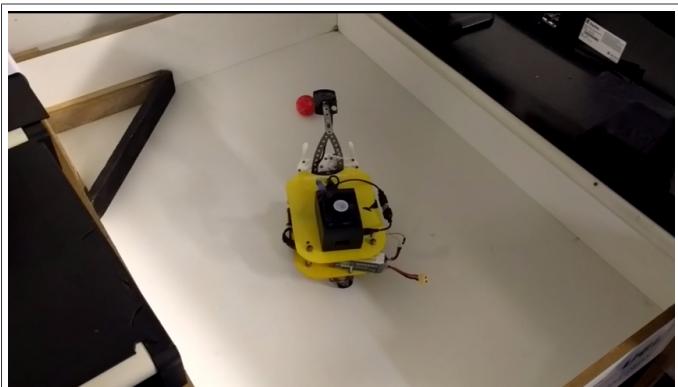
(c)



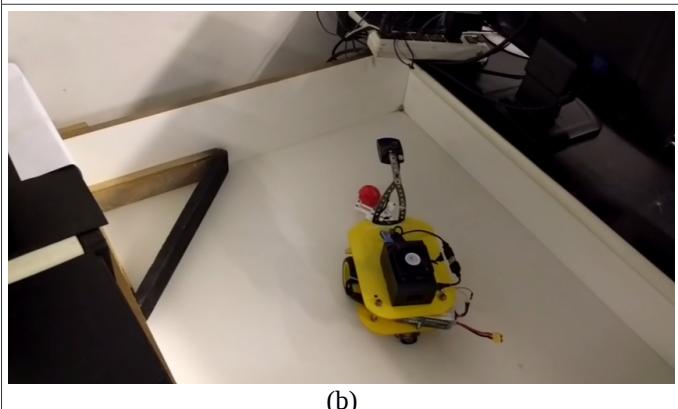
(d)

Figura 4 – Demonstra uma sequência de funcionamento do sistema detectando e seguindo a linha. (a) demonstra o posicionamento do robô, no tempo t. (b) mostra o robô no tempo t + 1. (c) demonstra o posicionamento do robô no tempo t + 2. E por fim, (d) mostra o posição do robô no tempo t + 3.

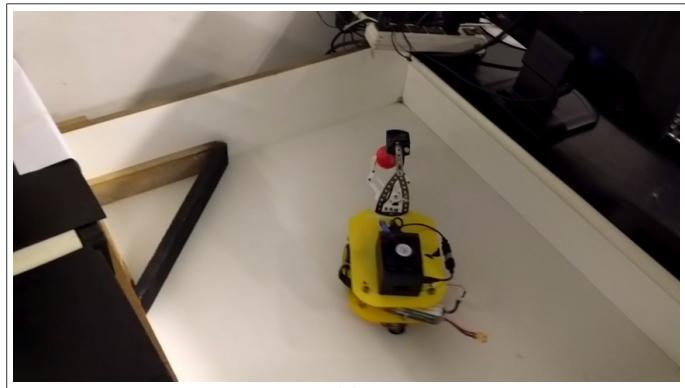
A Figura 5 demonstra uma sequência de funcionamento do sistema detectando o objeto alvo (vítima), posicionando o robô, executando o resgate à vítima e colocando-a na área de resgate. A Figura 5(a) demonstra o posicionamento do robô, no tempo t, após uma busca e detecção, onde o robô se posiciona frontalmente ao objeto alvo. A Figura 5(b) mostra o posicionamento do robô, no tempo t + 1, próximo ao objeto alvo. A Figura 5(c) é demonstrado o momento de captura do objeto alvo pela garra, no tempo t + 2. A Figura 5(d) no tempo t + 3, mostra a posição da garra, com o objeto alvo, na parte superior do robô, se deslocando para a área de resgate. A Figura 5(e) desmonstra o robô colocando o objeto na área de resgate, no tempo t + 4.



(a)



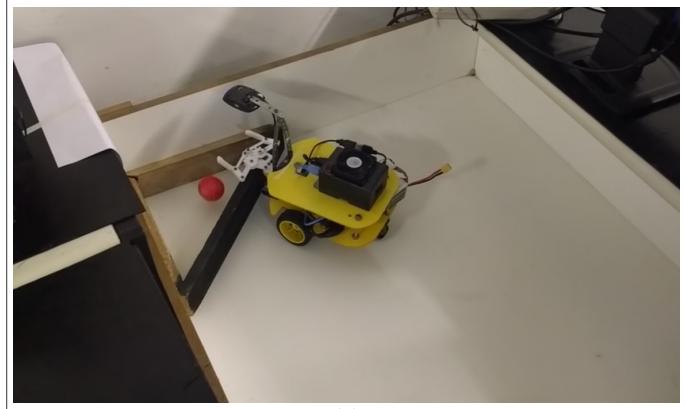
(b)



(c)



(d)



(e)

Figura 5 – Demonstra uma sequência de funcionamento do sistema detectando o objeto alvo (vítima), posicionando o robô, executando o resgate à vítima e colocando-a na área de resgate. (a) demonstra o posicionamento do robô, no tempo t, após uma busca e detecção, onde o robô se posiciona frontalmente ao objeto alvo. (b) mostra o posicionamento do robô, no tempo t + 1, próximo ao objeto alvo. (c) é demonstrado o momento de captura do objeto alvo pela garra, no tempo t + 2. (d) no tempo t + 3, mostra a posição da garra, com o objeto alvo, na parte superior do robô, se deslocando para a área de resgate. (e) desmonstra o robô colocando o objeto na área de resgate, no tempo t + 4.

5 DISCUSSÃO

Os sistemas propostos foram capazes de usufruir de excelentes ferramentas tecnológicas, como a própria robótica, a visão computacional e o OpenCV. Neste contexto, o robô demonstrou eficácia em realizar a tarefa de detectar e seguir a linha preta por toda a parte necessária na arena de competição (simulador). Além disso, na sala de resgate, foi capaz de detectar um objeto alvo, executar as manobras necessárias e posicionar corretamente o robô, realizar os movimentos da

garra, baseado totalmente em sensoriamento visual e resgatar e levar o objeto até um ponto pré determinado.

Uma deficiência do sistema proposto é que ele detecta apenas um tipo de objeto (forma e cor). Outro ponto é que o robô por vezes apresenta problemas de aderência à pista, gerando derrapagens, principalmente na rampa. É possível que uma nova distribuição de peso no plataforma robótica precise ser feita. Outra possível solução para o problema de aderência seria a mudança das rodas do robô.

6 CONCLUSÕES

Neste trabalho foi implementado um robô de competição baseado em visão computacional. Para isso foi desenvolvido um sistema de detecção de linha e um sistema de detecção de objeto a serem utilizados no resgate de um objeto alvo (vítima) para um robô autônomo seguidor de linha utilizando câmeras. A solução aqui proposta demonstrou ser um bom caminho na construção e implementação de robôs utilizando sensoriamento visual por câmera.

O desenvolvimento dos sistemas propostos mostrou o quanto a visão computacional pode ser útil e importante para o mapeamento de uma área, trazendo bons resultados finais. De um modo geral, o robô foi capaz de detectar e seguir uma linha na arena até um ponto determinado (entrada da sala de resgate), e após, reconhecer a vítima (objeto alvo) em tempo real, dirigir-se ao encontro dela e a resgatar com sucesso, levando-a à área de resgate.

O sistema proposto abre direções para trabalhos futuros que poderão superar suas deficiências e aperfeiçoar suas capacidades. Uma possibilidade é fazer com que o robô reconheça outras cores e formas. É importante também que ele consiga identificar mais que uma vítima para um melhor desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

Antonello, R. (2018). Introdução a Visão Computacional com Python e OpenCV. Disponível em: <http://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2017/02/Livro-Introdução-a-Visão-Computacional-com-Python-e-OpenCV.pdf>. Acesso em: agosto de 2018.

BMVA (2018). The British Machine Vision Association and Society for Pattern Recognition. Disponível em: <http://www.bmva.org/visionoverview>. Acesso em: agosto de 2018.

Ferreira, M. F. G. ; Araujo, L. G. ; Macedo, M. S. ; Souza, P. F. ; Salles, F. ; Amaral, E. M. A. (2016). Sistema de controle por visão em primeira pessoa e sistema de detecção e rastreamento de um alvo baseados em visão computacional utilizando kinect. In: VI Mostra Nacional de Robótica, Recife, PE, 2016.

Kachore, R. N. Detections of salient region by using fast pixelwise image saliency aggregation (F-PISA) (2017). International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). DOI: 10.1109/I-SMAC.2017.8058255. Publisher: IEEE. Palladam, India.

Malamas, E. N., Petrakis, E. G., Zervakis, M., Petit, L., and Legat, J.-D. (2003). A survey on industrial vision systems, applications and tools. *Image and vision computing*, 21(2):171–188.

OpenCV (2018). Disponível em: <https://opencv.org/>. Acesso em: agosto de 2018.

Skinner, K. A.; Johnson-Roberson, M. Towards real-time underwater 3D reconstruction with plenoptic cameras (2016). IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon Convention Center, Daejeon, Korea.

Zhou, T.; Wachs, J. P. Finding a Needle in a Haystack: Recognizing Surgical Instruments through Vision and Manipulation (2017). Electronic Imaging, Intelligent Robotics and Industrial Applications using Computer Vision, pp. 37-45(9). Publisher: Society for Imaging Science and Technology.