

12º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2021

Processamento de Imagem no Reconhecimento de Percursos por Robô Móvel Autônomo

GUILHERME FORTUNATO MIRANDA¹, MASAMORI KASHIWAGI², VERA LÚCIA DA SILVA³

¹ Aluno do Curso Técnico Integrado ao Ensino Médio em Automação Industrial, IFSP, Campus Suzano, ebfmiranda5@gmail.com.

² Docente EBTT, IFSP, Campus Suzano, masamori@ifsp.edu.br.

³ Docente EBTT, IFSP, Campus Suzano, verals@ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 9.16.00.00-6 Engenharia Mecatrônica.

RESUMO: Este trabalho tem como o objetivo utilizar o processamento de imagem para a realização dos desafios da Olimpíada Brasileira de Robótica (OBR) que envolvem os caminhos a serem seguidos pelo robô. O projeto tem o intuito de criar um método alternativo, utilizando uma câmera, com melhor ou mesma performance, comparado com os métodos que utilizam sensores de cores e refletância. Os componentes utilizados para a realização do projeto foram uma Raspberry Pi com um módulo de câmera acoplado, utilizada como processador central para as tarefas envolvendo imagem, e um Arduino Nano para o controle de motores e sensores. Na programação foram utilizadas a biblioteca OpenCV para o processamento de imagem e a Linguagem de Programação C++.

PALAVRAS-CHAVE: linguagem de programação C++; visão computacional; robótica; robô seguidor de linha; OBR.

Path recognition using image processing in mobile autonomous robot

ABSTRACT: The objective of this work is to use image processing to accomplish challenges from the *Brazilian Olympiad of Robotics* (OBR) related to paths followed by the robot. This project has the idea to make an alternative method, with better or equivalent performance, compared to methods that use color or reflective sensors. The components that were used in the project are a Raspberry Pi with an attached camera module, used as the core processor for image processing tasks and an Arduino Nano used for the control of actuators and sensors. On the programming it was used the OpenCV library for image processing and the programming language C++.

KEYWORDS: programming language C++; computer vision; robotics; line follower robot; OBR.

INTRODUÇÃO

Na Olimpíada Brasileira de Robótica (OBR) robôs autônomos devem executar diversas tarefas e passar por obstáculos, simulando um ambiente hostil no qual um humano não tem acesso. Dentre os desafios propostos existe um caminho a ser seguido, envolvendo as habilidades dos robôs de seguir uma linha na cor preta, superar curvas e encruzilhadas e reconhecer sinalizações verdes nas curvas, que indicam a direção correta que o robô deve seguir no percurso. Para a resolução desses desafios, muitos competidores utilizam sensores de cores ou refletância, que têm inúmeros pontos cegos devido ao seu campo de atuação ser focado em um único ponto no solo, gerando incerteza e pouca precisão.

Para a resolução desses desafios pode-se utilizar vias alternativas que tenham o mesmo propósito, como o processamento de imagem, onde o robô segue uma linha por meio das imagens obtidas por uma câmera.

Foi apercebido o desenvolvimento de um software para realização do reconhecimento do caminho e suas sinalizações (MIRANDA et. al, 2020), todavia os testes realizados de maneira móvel ocorreram por meio de um protótipo artesanal muito primitivo, que apresentava algumas desvantagens estruturais como a falta de precisão na conformidade de suas medidas.

Este trabalho propõe a utilização do processamento de imagem na realização dos desafios presentes na área de percurso e rampa, que envolvam seguir um caminho, presentes na OBR, utilizando um minicomputador Raspberry Pi com uma câmera acoplada como o dispositivo de processamento de imagem e um Arduino Nano para o controle dos atuadores e sensores em um robô impresso por meio de uma impressora 3D. O software de controle desenvolvido além de realizar o processamento de imagens, também implementou o controlador Proporcional, Integral e Derivativo (PID).

MATERIAIS E MÉTODOS

Na implementação do programa de controle de robô utilizou-se a biblioteca OpenCV como suporte para a aplicação do processamento de imagem (OPENCV TEAM, 2021). A Linguagem de Programação C++ foi utilizada devido a sua alta performance na execução dos processos (SAVITCH, 2004).

Os componentes utilizados no protótipo do robô são: um par de motores, um par de servo motores, um computador de placa única Raspberry Pi (OLIVEIRA, 2017), um microcontrolador Arduino Nano (MCROBERTS, 2011), uma ponte H, um sensor ultrassônico, um sensor acelerômetro, um módulo de câmera para Raspberry Pi, um Power Bank utilizado na alimentação dos componentes, uma arquitetura modelada em 3D e um *led* para a iluminação da pista.

A arquitetura do robô foi projetada em 3D, utilizando o software SolidWorks, e impressa em uma impressora 3D. O robô foi montado de forma apropriada com os componentes escolhidos e o sistema de controle foi desenvolvido e embarcado no robô. Tanto a arquitetura como a programação do robô tiveram sucessivas modificações até o projeto final. Testes foram realizados para verificar a performance do robô.

RESULTADOS E DISCUSSÃO

A Figura 1 apresenta o protótipo do robô móvel desenvolvido, unindo a arquitetura impressa em 3D e os componentes eletrônicos escolhidos.

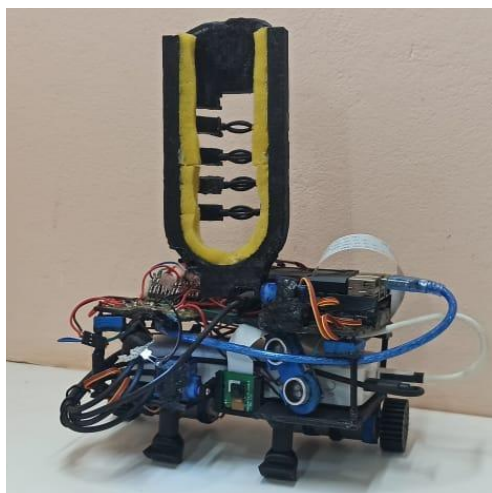


FIGURA 1: Foto do protótipo desenvolvido.

A Figura 2 exemplifica um percurso possível no qual o robô pode se deparar em uma competição.

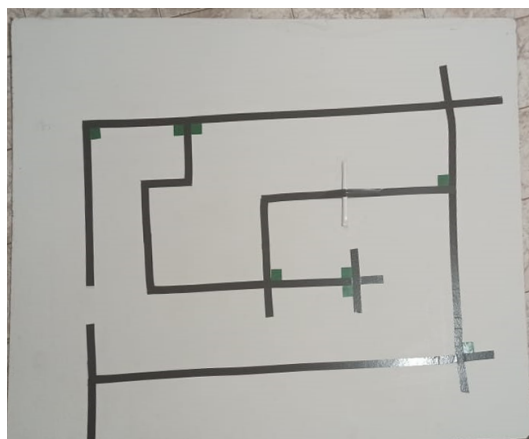


FIGURA 2: Exemplo de percurso

O software embarcado no robô foi desenvolvido na Linguagem de Programação C++ e utilizou a biblioteca OpenCV, permitindo o robô seguir o percurso por meio do processamento de imagens.

No início da execução do programa de controle do robô é estabelecida uma comunicação por meio de porta serial entre o Raspberry Pi e o microcontrolador Arduino Nano. Esta comunicação tem a finalidade de enviar instruções referentes aos atuadores e a transmissão de dados obtidos pelos outros sensores.

As imagens processadas são obtidas por meio de *frames* capturados pelo módulo de câmera conectado na Raspberry Pi. As imagens são analisadas para se extrair a posição central da linha no eixo X, que é relacionada de acordo com a distância do centro do robô.

Após a aquisição realiza-se o recorte da imagem para uma área de alta relevância, para evitar o processamento de áreas com informações irrelevantes e a filtragem de ruídos da imagem, que é feita com base na aplicação de um Kernel de convolução mediano, que tem a finalidade de substituir cada *pixel* da imagem pela média dos outros pixels na redondeza, juntamente com o filtro bilateral que substitui cada pixel da imagem pela média ponderada dos seus pixels vizinhos com base na intensidade de cada um. Dessa maneira eliminando a maioria dos ruídos de larga escala enquanto suaviza a imagem, conforme Figura 3.

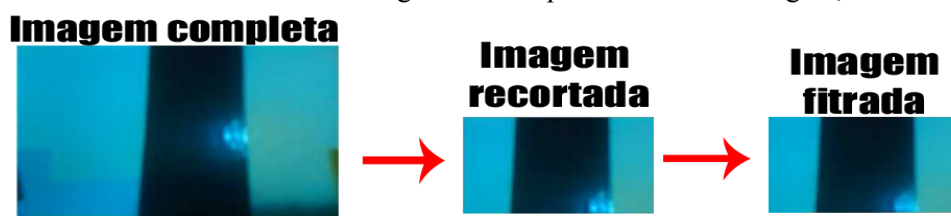


FIGURA 3: Processo de filtragem da imagem.

Após a filtragem da imagem é delimitado o contorno da linha baseado na localização de todos os pixels dentro de uma faixa de cores *RGB* predeterminada como a cor do caminho, no caso atual o preto. Depois da averiguação de todos os pixels pretos são localizados os seus contornos e somente é considerado o maior deles, que se pressupõe que seja a linha preta. Em seguida, é utilizado o algoritmo de invólucro convexo para transformar o contorno atual em uma convexa, e o ponto central é extraído, conforme ilustra a Figura 4.

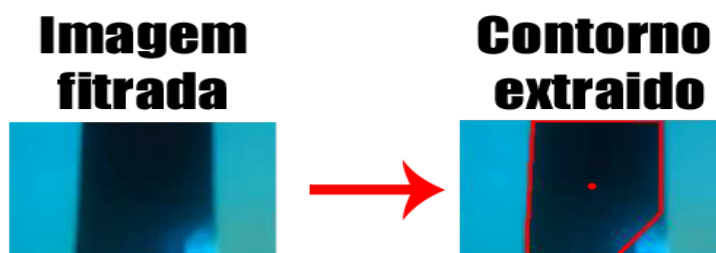


FIGURA 4: Extração do ponto central do caminho.

Após a análise das imagens é transmitido para o microcontrolador a diferença entre o centro do robô e a linha, que será utilizada como principal fator para o controle. Essa diferença será o erro utilizado no controlador Proporcional, Integral e Derivativo (PID). O controlador PID utiliza o erro medido em uma equação que o correlaciona com valores que podem ser utilizados nos atuadores (motores) para o controle do robô. O PID pode ser descrito pela seguinte fórmula:

$$ValorPID = K_p E_{(t)} + K_i \sum_0^t E_{(t)} + K_d [E_{(t)} - E_{(t-1)}] \quad (1)$$

Em que:

Kd – Constante Derivativa

E – Erro

Kp – Constante Proporcional

ValorPID – Valor resultante que pode ser utilizado no controle

Ki – Constante Integral

As constantes são escolhidas com base em testes experimentais realizados com o robô. O resultado da fórmula é utilizado para alteração das velocidades dos motores, sendo adicionada com sentidos opostos em motores de lados diferentes.

Para a interpretação de curvas é extraído a posição central no eixo X das sinalizações em verde na pista (caso encontradas) e os pontos extremos do contorno da linha. As posições das sinalizações são comparadas em relação ao ponto central da linha, para a detecção do lado em que a curva está sendo indicada e, em seguida é classificado se o ponto central está abaixo do ponto extremo da linha pelo lado correspondente, para assim o considerar como válido. Da mesma maneira é realizada a análise para o caso em que o robô tenha que dar meia volta, no qual é demarcado com dois verdes válidos em lados opostos da linha.

Na detecção de curvas sem as sinalizações considera-se a existência de somente um ponto extremo no canto esquerdo ou direito e, se o ponto extremo mais alto tiver uma pequena diferença de distância no eixo Y em relação ao ponto extremo oposto já encontrado.

O algoritmo de envólucro convexo utilizado anteriormente, quando deparado a uma curva, transforma o contorno em convexo assim intensificando muito o erro, indicando uma possível curva, conforme Figura 5.

Nessa situação o programa inicia duas rotinas simultâneas para poder seguir a linha com precisão e detectar curvas, nessa rotina a imagem é recortada em duas fatias, uma maior, que apresenta todas as características da imagem, e uma menor, que ignora a área da localização das curvas, que causariam ruído no momento da aquisição do erro da posição do robô em relação à curva, conforme Figura 6.

invólucro convexo em curva

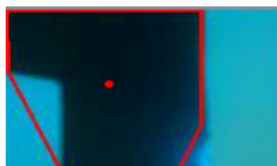


FIGURA 5: Invólucro convexo em curva.

- Visão geral
- Controle do robô
- Detecção de curvas

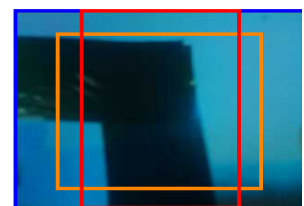


FIGURA 6: Campos de visão selecionados.

A detecção de curvas com ou sem a sinalização verde funciona de maneira similar, ambas detectam os pontos extremos do contorno e verificam se existe um ponto extremo perto dos cantos da imagem. É possível detectar isso com as informações da posição dos pontos no eixo X e pelas dimensões da imagem. No caso de uma curva sem sinalização verde, o sinal positivo só é identificado caso haja um ponto extremo em um dos lados da imagem, já que, caso haja uma bifurcação, o protocolo é seguir reto. Entretanto, com sinalizações verdes é possível haver bifurcações e curvas, já que a direção da curva é concebida pela posição do verde em

relação a linha, que é detectada por meio da diferença do centro do verde e o centro da linha no eixo X. Caso o valor seja positivo a sinalização está à direita e caso seja negativa ela estará à esquerda.

As sinalizações que são utilizadas no processo só são válidas se a sua posição central estiver abaixo dos pontos extremos, detectado por meio da diferença entre a posição no eixo Y dos pontos extremos com o centro da sinalização. Caso haja uma sinalização válida para cada lado, o robô deve dar meia volta. A Figura 7 exibe um possível caso de curva com sinalização verde e a Figura 8 demonstra uma situação de meia volta.

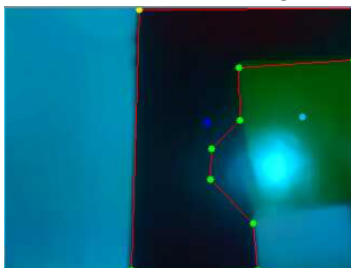


FIGURA 7: Exemplo de caso com sinalização verde com pontos extremos e centro do verde explícito.

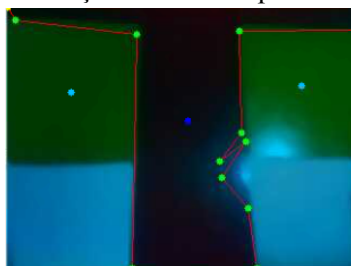


FIGURA 8: Exemplo de caso com duas sinalizações verde, indicando meia volta, com pontos extremos e centro do verde explícito.

Nos testes realizados o robô foi capaz reconhecer e realizar de maneira correta os casos propostos desde que exista um caminho reto após cada obstáculo para que o robô tenha um tempo para a normalização de sua resposta.

CONCLUSÕES

A utilização do processamento de imagem no reconhecimento de caminhos por um robô móvel autônomo apresenta bons resultados, sendo capaz de realizar os desafios propostos, com o empecilho da necessidade de um tempo para a normalização do erro após curvas, que pode ser diminuído com base em protocolos de finalização de curvas que podem ser implementados em futuros projetos.

Como plano para trabalhos futuros pretende-se ampliar a utilização do processamento de imagem para identificar objetos e assim também ser utilizado nos desafios zona de resgate da OBR.

AGRADECIMENTOS

Agradeço a Deus e a meus pais pelo suporte no meu caminhar do áspero caminho da vida, aos meus professores, especialmente aos meus orientadores Vera Lúcia e Masamori Kashiwagi pelo auxílio próximo no desenvolvimento do projeto. Também agradeço pelo auxílio da bolsa fornecida pela MNR - ICJ CNPq/MNR durante o desenvolvimento do projeto.

REFERÊNCIAS

- MCROBERTS, Michael. *Arduíno básico*. São Paulo: Novatec, 2011. 453 p. ISBN9788575222744.
- MIRANDA, G. F.; PEREIRA, N. Q. (2020). *O Uso do Processamento de Imagem em Robô Móvel para Busca e Resgate*, Mostra Nacional de Robótica, 2020.
- OPENCV TEAM. *OpenCV Tutorials*, c2021. Disponível em: <<https://opencv.org/>>. Acesso em: 20 mai. 2021.
- OLIVEIRA, SÉRGIO DE. *Internet das Coisas com ESP8266, Arduino e Raspberry Pi*. São Paulo: Novatec, 2017, 240 p. ISBN: 9788575225820.
- SAVITCH, W. J. *C++ Absoluto*. 1ª edição. Brasil: ERJ Composição Editorial e Artes Gráficas Ltda, 2004.