

# VI SIMPÓSIO DE ENGENHARIAS E TECNOLOGIA

## TECNOLOGIA COMO VETOR DE SUSTENTABILIDADE

### CONTROLE DE POSICIONAMENTO DE ROBÔS POR VISÃO COMPUTACIONAL

*ROBOT POSITIONING CONTROL BY COMPUTER VISION*

*CONTROL DE POSICIONAMIENTO DE ROBOTS POR MEDIO DE VISIÓN COMPUTARIZADA*

Andrei Schiavon Nonato dos Santos<sup>1</sup>  
Edilson Cesar Mello da Silva<sup>2</sup>  
Eduardo Gregório Soares<sup>3</sup>  
Charles Way Hun Fung<sup>4</sup>

**Grupo de trabalho:** Grupo de Inovação, Tecnologia e Sustentabilidade da Uninter

#### RESUMO

Este projeto consiste em controlar o posicionamento de um robô em uma determinada área conhecida, aplicando-se visão computacional para detectar e posicionar o robô. Através de uma câmera convencional conectada ao computador via porta USB, obtém-se a posição do robô por meio da utilização da biblioteca do OpenCV e da linguagem de programação Python. A câmera está fixada acima da área de atuação do robô; a partir da calibração da câmera pelo método do tabuleiro de xadrez, obtém-se parâmetros como: a matriz intrínseca da câmera, o coeficiente de distorção e duas matrizes de rotação e translação, que têm a finalidade de corrigir as distorções radial e tangencial em imagens. É possível reduzir os erros provenientes das imagens capturadas por meio da segmentação de imagens baseada na cor, que consiste em um processo que divide a imagem em regiões que a compõem; atribui-se um padrão de cores RGB específico para identificar o robô, na área conhecida. O robô presente neste estudo está construído com os seguintes componentes: um ESP32, uma ponte H L298N, um suporte para pilhas do tipo AA com plug P4, dois motores DC 3-6V com caixa de redução e eixo duplo, um chassi feito em impressora 3D, duas rodas de borracha e uma roda com rodízio giratório. O robô recebe comandos do computador via comunicação sem fio, através do protocolo de comunicação Esp-Now, transmite o comando para outro ESP32 que está no robô, responsável em executar o comando recebido. Ao integrar a visão computacional e a comunicação sem fio obtém-se a navegação do robô. Como resultado, ao utilizar segmentação de cor para identificar o robô, deve-se ter um ambiente com boa iluminação; a utilização do protocolo ESP-NOW fornece liberdade de movimento ao robô e uma comunicação direta entre computador e robô; observa-se que o processamento digital de imagens necessita de computador com mínimo de recursos de hardware, neste caso, uma placa de vídeo para um melhor desempenho no processamento digital de imagens; sendo assim, deve-se limitar a velocidade do robô para evitar que o programa venha a parar a execução, durante o processamento de imagens.

**Palavras-chave:** comunicação sem fio; visão computacional; robô; computador; OpenCV; ESP-NOW; calibração de câmera; wi-fi.

<sup>1</sup> Estudante do curso de Engenharia Elétrica do Centro Universitário Internacional — UNINTER.

<sup>2</sup> Estudante do curso de Engenharia Elétrica do Centro Universitário Internacional — UNINTER.

<sup>3</sup> Estudante do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário Internacional — UNINTER.

<sup>4</sup> Professor da UNINTER (orientador).

# VI SIMPÓSIO DE ENGENHARIAS E TECNOLOGIA

## TECNOLOGIA COMO VETOR DE SUSTENTABILIDADE

### ABSTRACT

This project consists in controlling a robot's positioning in a certain known area, by applying computer vision to detect and position the robot. Through a conventional camera connected to the computer via a USB port, the position of the robot is obtained by using the OpenCV library and the Python programming language. The camera is fixed above the robot's operation area; from the camera calibration by the checkerboard method parameters such as: the camera's intrinsic matrix, the distortion coefficient, and two rotation and translation matrices are obtained, which have the purpose of correcting radial and tangential distortions in images. It is possible to reduce the errors coming from the captured images by means of color-based image segmentation, which consists of a process that divides the image into regions that compose it; a specific RGB color pattern is assigned to identify the robot, in the known area. The robot in this study is built with the following components: an ESP32, an L298N H-bridge, an AA battery holder with a P4 plug, two DC 3-6V motors with gearbox and double axle, a chassis made in a 3D printer, two rubber wheels, and a caster wheel. The robot receives commands from the computer via wireless communication, through the Esp-Now communication protocol, and transmits the command to another ESP32 on the robot, responsible for executing the received command. By integrating computer vision and wireless communication, the navigation of the robot is obtained. As a result, when using color segmentation to identify the robot, one must have an environment with good lighting; the use of the ESP-NOW protocol provides freedom of movement to the robot and direct communication between the computer and robot; it is observed that the digital image processing needs a computer with minimum hardware resources, in this case, a video board for better performance in the digital image processing; therefore, one must limit the robot speed to prevent the program from stopping the execution during the image processing.

**Keywords:** wireless communication; computer vision; robot; computer; OpenCV; ESP-NOW; camera calibration; wi-fi.

### RESUMEN

Este proyecto consiste en controlar el posicionamiento de un robot en un área conocida determinada, con la aplicación de visión computarizada para detectar y posicionar el robot. A través de una cámara convencional conectada al computador por puerto USB, se obtiene la posición del robot por medio de la utilización de la biblioteca del OpenCV y del lenguaje de programación Python. La cámara está fijada sobre el área de acción del robot; a partir de la calibración de la cámara por el método del tablero de ajedrez, se obtienen parámetros como: la matriz intrínseca de la cámara, el coeficiente de distorsión y dos matrices de rotación y traslación, que tienen la finalidad de corregir la distorsión radial y tangencial en imágenes. Es posible reducir los errores provenientes de las imágenes captadas, por medio de la segmentación de imágenes basada en el color, que consiste en un proceso que divide la imagen en las regiones que la componen; se define un patrón de colores RGB específico para identificar el robot, en el área conocida. El robot utilizado en este estudio está construido con los siguientes componentes: un ESP32, un puente H L298N, un soporte para pilas AA con conector tipo P4, dos motores DC 3-6V con caja de cambios de doble eje, un chasis hecho en impresora 3D, dos ruedas de goma y una rueda de vástago giratorio. El robot recibe comandos del computador por comunicación inalámbrica, a través del protocolo de comunicación Esp-Now, transmite el comando para otro ESP32 ubicado en el robot, responsable por ejecutar el comando recibido. Cuando se integra la visión computarizada y la comunicación sin cable, se logra la navegación del robot. Como resultado, cuando se haga la segmentación del color para identificar el robot, se debe contar con un ambiente con buena iluminación; la utilización del protocolo ESP-NOW ofrece libertad de movimiento al robot y una comunicación directa entre computador y robot; se observa que el procesamiento digital de imágenes necesita de un computador con un mínimo de recursos de hardware, en ese caso una placa de video para un mejor desempeño en el procesamiento digital de imágenes; por otro lado, se debe limitar la velocidad del robot para evitar que el programa llegue a interrumpir la ejecución durante el procesamiento de imágenes.

# VI SIMPÓSIO DE ENGENHARIAS E TECNOLOGIA

## TECNOLOGIA COMO VETOR DE SUSTENTABILIDADE

**Palabras-clave:** comunicação inalámbrica; visão computarizada; robot; computador; OpenCV; ESP-NOW; calibración de cámara; wifi.

### INTRODUÇÃO

Quando o assunto é robôs, imediatamente pensamos em máquinas com características físicas humanas, com ações próprias e independentes (RUSSEL; NORVIG, 2013). Mas, para isso, um robô precisa saber onde está localizado no mundo real, perceber objetos à sua volta, para assim traçar uma trajetória rumo ao seu objetivo (CRAIG, 2012).

Uma das formas de fazer um robô enxergar o mundo como nós é através da visão computacional. Com a utilização de uma câmera com vista superior, neste caso os olhos do robô, ele pode capturar as imagens em tempo real e, através da visão computacional, processar estas imagens, reconhecendo padrões de cores e formas, definir a sua localização no plano cartesiano em um espaço delimitado e traçar o melhor caminho para realizar o seu percurso (BARELLI, 2018).

Este projeto tem como objetivo utilizar a visão computacional para determinar o posicionamento de um robô no plano cartesiano dentro de um espaço determinado e fazer com que ele se mova de forma autônoma até um determinado ponto de referência.

### METODOLOGIA

Para realizar a calibração da câmera utilizou-se uma imagem na forma de tabuleiro de xadrez contendo 8 colunas e 10 linhas, com quadrados no tamanho de 20 x 20 mm. Deste tabuleiro foram tiradas 32 fotos em posições diferentes. Através de um algoritmo, foram detectados os cantos internos nas imagens do tabuleiro, de modo a obter os dados da matriz de calibração da câmera, a fim de minimizar as distorções geométricas da lente da câmera, que são geradas na captura da imagem (OPENCV, 2021a, b, c).

Um pequeno robô, construído com peças confeccionadas em uma impressora 3D, foi utilizado como modelo neste projeto. Ele possui dois motores DC com caixa de redução, um driver para controle dos motores e uma placa de desenvolvimento ESP32. Sobre o robô foram colocados dois círculos coloridos, um de cor verde — que representa a parte de trás do robô — e um laranja — a parte da frente. Outro círculo de cor azul foi utilizado como referencial, de modo a obter a posição do robô em relação a este ponto e poder movimentá-lo até ele.

Uma webcam foi posicionada a aproximadamente um metro de altura, voltada para o chão, onde as limitações da imagem capturada por ela delimitam a área de atuação do robô. A imagem capturada em tempo real pela câmera é corrigida utilizando a matriz de calibração e depois segmentada, de forma a reconhecer apenas

# VI SIMPÓSIO DE ENGENHARIAS E TECNOLOGIA

## TECNOLOGIA COMO VETOR DE SUSTENTABILIDADE

as cores azul, verde e laranja. Com a segmentação dos três círculos coloridos, pudemos determinar o centróide de cada um deles, obtendo assim suas respectivas coordenadas  $x$  e  $y$ . Deste modo foi possível determinar a posição do robô em relação ao referencial no plano cartesiano (GONZALEZ; WOODS, 2010).

Posteriormente, passamos a considerar a parte de trás do robô, círculo verde, como sendo o referencial. Separamos a área capturada pela câmera em 4 quadrantes, sendo o primeiro o canto superior direito da imagem; o segundo, o canto superior esquerdo; o terceiro, o canto inferior esquerdo; e o quarto quadrante, o canto inferior direito. Nos testes realizados, o círculo azul foi posicionado no segundo quadrante e toda a programação de movimentação do robô foi feita para esta situação.

Para determinar a orientação do robô, ou seja, em qual quadrante está localizada a sua frente, que é o círculo laranja, fizemos uma comparação entre as coordenadas  $x_v$  e  $y_v$  do centróide do círculo verde e as coordenadas  $x_l$  e  $y_l$  dos centróides do círculo laranja:

- 1º Quadrante:  $x_l \geq x_v$  e  $y_l \leq y_v$ .
- 2º Quadrante:  $x_l \leq x_v$  e  $y_l \leq y_v$ .
- 3º Quadrante:  $x_l \leq x_v$  e  $y_l \geq y_v$ .
- 4º Quadrante:  $x_l \geq x_v$  e  $y_l \geq y_v$ .

Sabendo em qual quadrante a frente do robô se localiza, podemos determinar o sentido de rotação do robô, horário ou anti-horário. Caso esteja no 1º quadrante, o movimento de rotação será no sentido anti-horário. Caso esteja no 3º, será no sentido horário. Porém, para quando a frente do robô está localizada no 2º, foi necessário calcular o ângulo formado entre a reta gerada entre os pontos dos centróides do círculo verde e do círculo laranja (reta 1) e o eixo  $x$ , que chamamos de  $\theta_1$ ; e o ângulo entre o prolongamento da reta gerada entre os pontos dos centróides do círculo azul e do círculo verde (reta 2) e o eixo  $x$ , que chamamos de  $\theta_2$ . Quando  $\theta_1 \geq \theta_2$  a rotação será no sentido horário, caso contrário, será no sentido anti-horário.

Se a frente do robô está no 4º quadrante realizamos os mesmos cálculos que para o 2º quadrante, porém, quando  $\theta_1 \leq \theta_2$  a rotação será no sentido horário, caso contrário, será no anti-horário. Com a reta 1 e a reta 2 calculamos o ângulo  $\theta_r$  formado entre estas duas retas, para enviar o comando de rotação do robô de modo a alinhar os três círculos azul, laranja e verde respectivamente.

Enquanto o ângulo  $\theta_r$  for maior que  $5^\circ$ , o comando de rotação será enviado até que o ângulo seja menor que  $5^\circ$ . Assim que isso ocorrer, será enviado o comando para que o robô se movimente para frente até chegar no círculo, onde para o movimento (SZELISKI, 2011).

Os comandos para a movimentação do robô são enviados através da porta serial do computador para uma placa de desenvolvimento chamada ESP32. Esta placa possui o recurso de Wi-Fi e, utilizando o protocolo de comunicação Esp-Now,

# VI SIMPÓSIO DE ENGENHARIAS E TECNOLOGIA

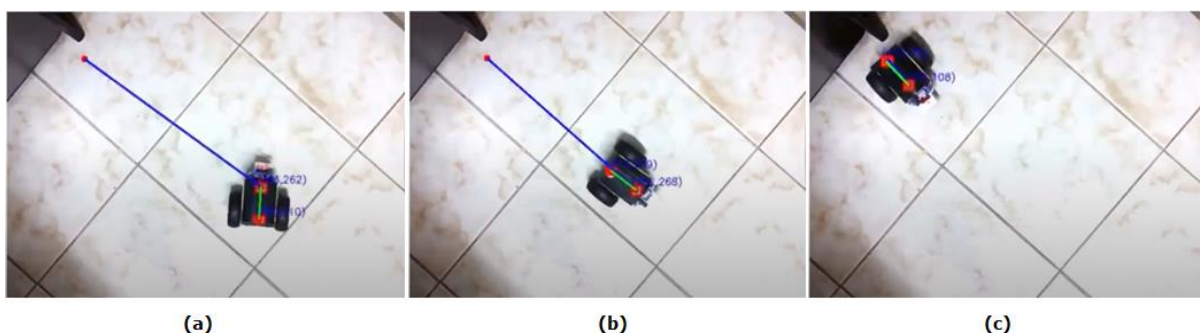
## TECNOLOGIA COMO VETOR DE SUSTENTABILIDADE

transmitimos o comando para outro ESP32 que está no robô, responsável por executar o comando recebido (ESPRESSIF, 2016).

### RESULTADOS E DISCUSSÃO

A Figura 1 apresenta um exemplo de uma sequência de ações do robô em função do objetivo de chegar a um ponto específico a partir do 4º quadrante.

**Figura 1:** navegação do robô até um ponto especificado. Em (a) o robô faz um movimento anti-horário para se fazer com que a linha de orientação do robô (linha verde), fique na mesma direção que a linha de trajetória (linha azul). Em (b) as linhas estão na mesma direção. Em (c) o robô está atingindo o objetivo



Utilizando o método de calibração da câmera, pudemos corrigir as distorções geométricas geradas pela lente, fazendo com que as medições tivessem um nível de erro tolerável, não influenciando de forma significativa os cálculos realizados para a movimentação do robô, bem como os cálculos de distância.

A utilização do ESP32 com o protocolo Esp-Now forneceu liberdade de movimento para o robô e uma comunicação direta entre o computador e o robô, o que fez com que fosse dispensado o uso de redes wireless, roteadores e complicadas configurações de redes.

Encontramos dificuldades na segmentação de cores devido à luminosidade do ambiente e no processamento da imagem, pois para um melhor desempenho se faz necessária uma placa de vídeo da qual não dispúnhamos no momento dos testes.

Quanto à movimentação do robô, se fez necessário o controle de velocidade pois, quando ele se movia de forma rápida, devido ao processamento da imagem necessitar de muito recurso de hardware, o programa parava a execução.

# VI SIMPÓSIO DE ENGENHARIAS E TECNOLOGIA

## TECNOLOGIA COMO VETOR DE SUSTENTABILIDADE

### CONSIDERAÇÕES FINAIS

Para implementações futuras iremos considerar que o ponto de referência para a movimentação do robô esteja em qualquer quadrante e não apenas no 2º, apresentado neste trabalho.

Também pretendemos fazer uma interação homem-robô, onde será possível, através de um click do mouse, determinar o ponto para onde o robô deverá ir, na sequência em que eles forem sendo colocados na tela.

Outra melhoria futura será a implementação de sensores para desviar obstáculos que se encontram na trajetória do robô e a implementação de mais robôs no mesmo espaço.

### REFERÊNCIAS

BARELLI, F. **Introdução à visão computacional**: Uma abordagem prática com Python e OpenCV. São Paulo: Editora Casa do Código, 2018.

CRAIG, JOHN J. **Robótica**. São Paulo: Pearson Education do Brasil, 2012.

ESPRESSIF. Esp-Now User Guide Version 1.0 (2016). Documentação de uso do protocolo. Disponível em: <https://www.espressif.com/en/products/software/esp-now/resources>. Acesso em: 08 ago. 2021.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. São Paulo: Pearson Education do Brasil, 2010.

HOWSE, J.; JOSHI, P.; BEYELER, M. **OpenCV**: Computer Vision Projects with Python. Birmingham: Packt Publishing, 2016.

OpenCV - Open Source Computer Vision, "Camera Calibration". 2021a. Disponível em: [https://docs.opencv.org/master/dc/dbb/tutorial\\_py\\_calibration.html](https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html). Acesso em: 07 ago. 2021.

OpenCV - Open Source Computer Vision, "Pose Estimation". 2021b. Disponível em: [https://docs.opencv.org/master/d7/d53/tutorial\\_py\\_pose.html](https://docs.opencv.org/master/d7/d53/tutorial_py_pose.html). Acesso em: 08 ago. 2021.

OpenCV - Open Source Computer Vision, "Camera Calibration and 3D Reconstruction". 2021c. Disponível em: [https://docs.opencv.org/3.4.15/d9/d0c/group\\_\\_calib3d.html](https://docs.opencv.org/3.4.15/d9/d0c/group__calib3d.html). Acesso em: 13 nov. 2021.

RUSSEL, S.; NORVIG, P. **Inteligência artificial**. Rio de Janeiro: Elsevier Editora, 2013.

SZELISKI, R. **Computer vision**: algorithms and applications. Washington: Springer-Verlag London, 2011.