2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) | 979-8-3503-2297-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICECCME57830.2023.10252410

Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023)
19-21 July 2023, Tenerife, Canary Islands, Spain

An AI-based Object Detection Approach for Robotic Competitions

Leonardo Pilarski
UTFPR
Toledo, Brazil
pilarski@alunos.utfpr.edu.br

Luiz E. Luiz
UTFPR
Toledo, Brazil
lluiz@alunos.utfpr.edu.br

João Braun
Research Centre in Digitalization and Intelligent Robotics
Polytechnic Institute of Bragança
Bragança, Portugal
jbneto@ipb.pt

Alberto Y. Nakano
UTFPR
Toledo, Brazil
nakano@utfpr.edu.br

Vítor Pinto
SYSTEC (DIGI2)
FEUP
Porto, Portugal
vitorpinto@fe.up.pt

Paulo Costa
INESC TEC
FEUP
Porto, Portugal
paco@fe.up.pt

José Lima
Research Centre in Digitalization and Intelligent Robotics
Polytechnic Institute of Bragança
Bragança, Portugal
jllima@ipb.pt

***Abstract*—Artificial Intelligence has been introduced in many applications, namely in artificial vision-based systems with object detection tasks. This paper presents an object localization system with a motivation to use it in autonomous mobile robots at robotics competitions. The system aims to allow robots to accomplish their tasks more efficiently. Object detection is performed using a camera and artificial intelligence based on the YOLOv4 Tiny detection model. An algorithm was developed that uses the data from the system to estimate the parameters of location, distance, and orientation based on the pinhole camera model and trigonometric modelling. It can be used in smart identification procedures of objects. Practical tests and results are presented, constantly locating the objects and with errors between 0.16 and 3.8 cm, concluding that the object localization system is adequate for autonomous mobile robots.**

***Index Terms*—Robotic competitions, Autonomous Mobile Robots, Object Detection, Artificial Intelligence.**

I. Introduction

Artificial intelligence seeks to create techniques that mimic human intelligence. One of its aspects is computer vision [1], which can be understood as a set of mathematical and computational techniques to process images and extract relevant information. This artificial vision plays a crucial role in robotics [2], particularly in mobile robotics, involving data acquisition and its use to perform tasks [3]. Advances in this area have allowed autonomous mobile robots to perceive and understand the environment around them, using cameras and other sensors to collect visual information about the terrain [4] [5].

An essential scenario for technological development is robotics competitions. These can be considered the starting point of research and development in many areas, such as science and technology, by evoking breakthrough ideas to solve competition problems [6]. RobotAtFactory [1] is one of these competitions, which seeks to simulate a factory environment. Over the years, it has evolved regarding technology development and complexity. Its most recent version, RobotAtFactory 4.0 [2] brought changes regarding the data available to guide the robot, replacing continuous lines with spaced markers printed on the floor. However, its principle of transporting boxes from an incoming warehouse to the outgoing one within the manufacturing floor remains unchanged.

In seeking to transport objects, locating and identifying them previously to perform specific tasks based on that information is essential [7] [8]. Object detection aims to identify all the target elements in the image, classify, and locate them. Several methods can be employed to obtain and provide the data [9] to applications to estimate the distance between the camera and those objects detected in the image [10].

This work presents the development of a mobile system for detecting and localizing objects through monocular vision. Thus, the system allows the possibility of being attached to a robot, which will use the system estimated distance and the angle to reach the objects of interest.

This paper is structured as follows: Section 2 presents works on state of art in object detection and artificial vision for object distance estimation. Section 3 is divided into two parts: Subsection A describes the system's components. Subsection B describes the development, including the correlation among the components of this work; section IV presents the results; and section V presents the conclusions and future works.

[1]https://robotica2012.dei.uminho.pt/12/index.php-option=com_content&view=category&layout=blog&id=75&Itemid=99.html

[2]https://www.festivalnacionalrobotica.pt/2022/competicoes/robotfactory-4.0-11.

II. State of the art

Object detection in mobile robotics is an important research area [11] [12], and the YOLO (You only look once) neural network has been widely used. YOLO allows real-time object detection with a single image as input, it is suitable for mobile robot applications [13] due to the limited processing usually used in such applications.

Monocular cameras are widely used in mobile robotics for their simplicity and low cost. However, the lack of depth information can limit 3D object detection [3]. Using previously known points of interest can overcome this, relying on obtained size and coordinates to triangulate the distance. This technique allows scene reconstruction in 3D and the objects' detection at different depths, improving the accuracy of the data of interest [14].

There is also research involving stereoscopic cameras capable of estimating depth from the difference between two cameras spaced at a well-calibrated value [15].

Regarding the monocular camera algorithm to estimate the distance, there are different mathematical approaches to finding the data of interest, such as [14], which has a camera in a rectilinear uniform motion, and [10] using this object detection to develop a robot to play football.

III. Methodology

Here, the system description is presented considering hardware, software, and system development.

A. System description

1) Hardware: The system hardware has as main components: a Raspberry Pi 4 model B (4 GB) and a camera module Raspberry Pi Cam Rev 1.3. These interconnected components use a flat cable and a 3D printing box for connection, protection, and support of the main components, as shown in Figure 1.

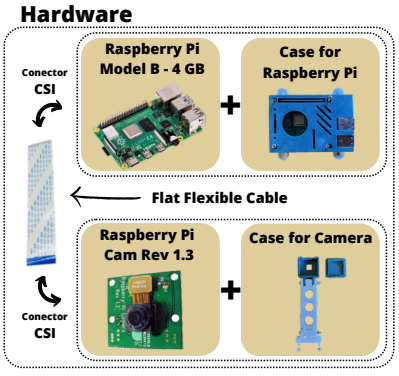


Fig. 1. Hardware Components.

2) Software: The system software is divided into two parts: data production and data usage.

For the first part, the process is shown in Figure 2, being:

- Image Acquisition with comercial camera (100 images);
- Dataset built in ROBOFLOW[3] passing to 237 images;
- Code execution in Google Colab [4];
- Further use of functionalities:
  - Python;
  - OpenCV[5];
  - Darknet[6];
  - YOLOv4 (Tiny)[7].

For training the YOLOv4 Tiny model, pre-trained weights from the 29th layer were used, based on the "COCO Names[8]" dataset. The data was divided into 90% for training and 10% for testing, based on the results obtained by the Roboflow platform."

The second part will use the data from the first part as training weights for the algorithm. In this phase, the OpenCV tools and configuration files (names, weights and configurations) will be used again to assist in developing the algorithm. Three functions are created: object detection, distance estimation, and orientation estimation. The development of this code is presented in section III-B.

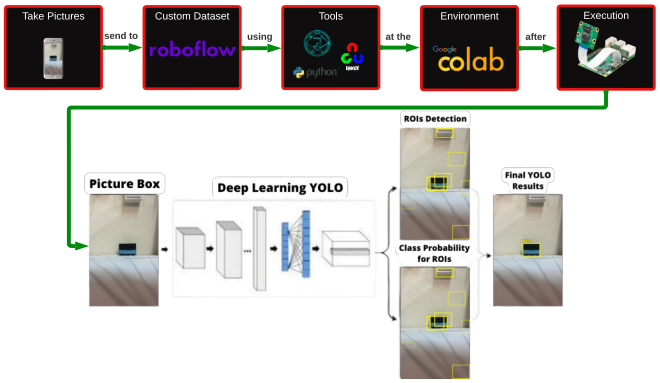


Fig. 2. Detection model training and execution with structure YOLOv4 Tiny.

B. Development

The object detection function, shown in figure 3, receives three parameters: the camera frame, the object accuracy threshold, and the non-maximum suppression value. It creates a list to store the names of the objects to be detected and aims to identify the object bounding boxes in the image.

The detection model is invoked, and if an object is detected, its information is stored in the respective variables. Otherwise, the variables are left empty. If the object list has no names yet, the class names are given to it. In addition, a list is created to store the detection information of the found objects.

[3]https://roboflow.com/
[4]https://colab.research.google.com/
[5]https://opencv.org/
[6]https://github.com/roboflow/darknet
[7]https://colab.research.google.com/drive/1PWOwg038EOGNddf6SXDG5AsC8PIcAe-G?ref=blog.roboflow.com#scrollTo=HQEktcfj9y9O
[8]https://github.com/pjreddie/darknet/blob/master/data/coco.names

Three lists are created, the object identification list, the accuracy list, and the bounding boxes list. Then it is verified to which class the object belongs, and if it belongs to the searched class, the information is stored in the variable created for such, and the function returns the frame and the object information. Otherwise, the function returns the frame and the value zero referring to the information.

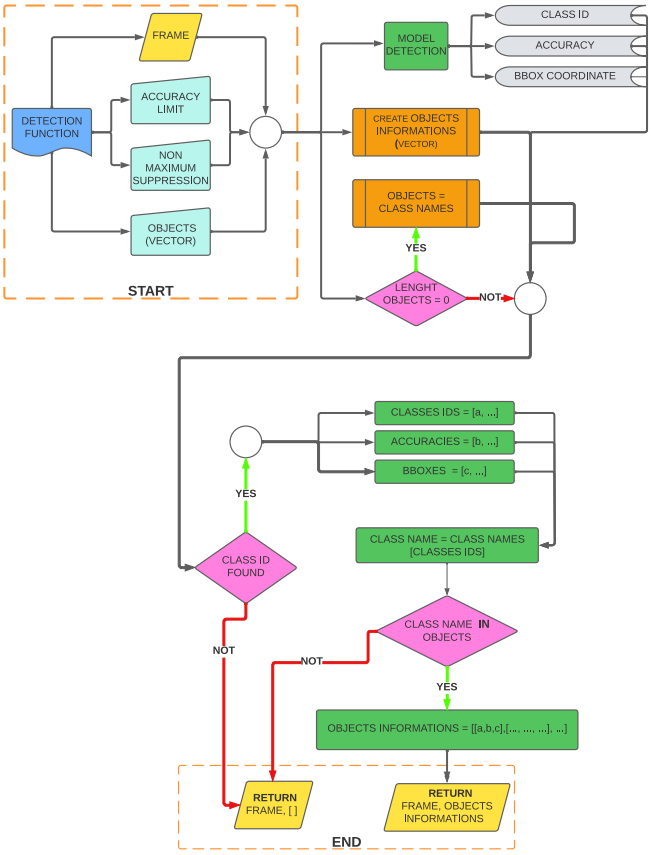


Fig. 3. Flowchart of the detection function.

The parameters for estimating the location, distance, and angle are presented in Figure 4, defined in Figure 5 and 6. Most parameters remain constant , where most of the parameters remain constant, such as the maximum image size and the camera angle of view. The only variable is the distance between the object and the camera, which affects the object's size in the image. Based on this information, the distance and the orientation angle will be calculated.

The distance and angle between the object and module are calculated using camera parameters such as focal length, hFoV (Horizontal Field of View), and image resolution following the equation in Figure 5 considering the parameters used as: 'a' - width in pixels on the image, 'b' - image resolution in pixels, 'c' - actual object width in cm, and 'd' - hFoV.

After distance estimation, the result is used to calculate an offset and fix the alignment angle between the module and the object of interest.

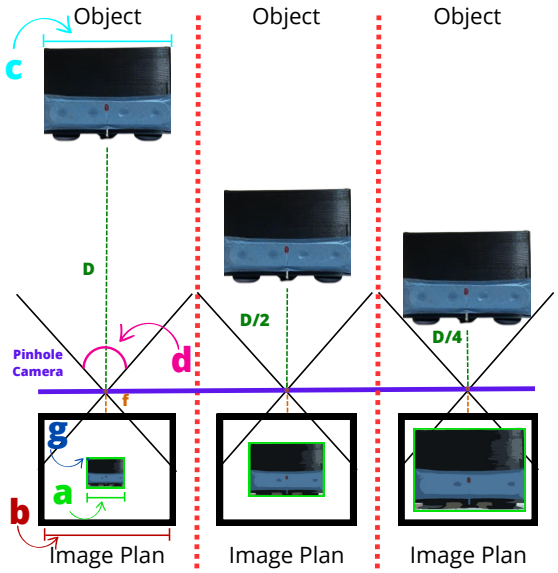


Fig. 4. Distance and angle variation based on the Pinhole model.

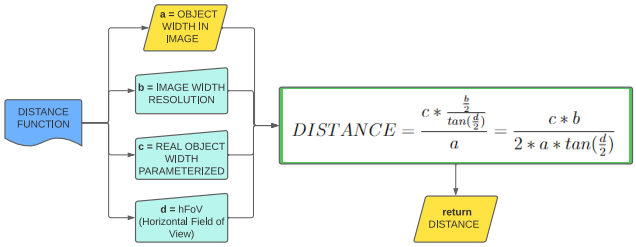


Fig. 5. Flowchart of the distance calculation function.

There are three possibilities of alignment: aligned, to the right, or the left, being this identification important for possible future implementations. The angle is calculated using trigonometric relations, considering that the object is always aligned with the module to the y-axis.

User provided information is utilized to compute the angle between an object and the module, such as the image resolution and the object's actual size. In addition, values provided by the object detection function are used, shown in Figure 6 as: 'g' - the initial position on the x-axis of the detection box, 'a' - its width, and 'D' - the previously estimated distance.

To exemplify the calculation of the angle of a right triangle, the 4 is used as an example, considering the sides D (adjacent) and C/2 (opposite). Firstly, it is necessary to convert the calculated distance in centimeters to pixels, considering the real size of the object and the image resolution, to obtain the adjacent side. Then, if the object is displaced relative to the center of the image, it is necessary to calculate the opposite side by finding the distance between the center of the image and the value of the center of the object with respect to the x-axis. To do this, the initial position of the detection box and

its width divided by 2 are used, taking into account the already identified rotation direction. Finally, to calculate the angle, it is necessary to apply the arctangent function of the division between the opposite and adjacent sides.

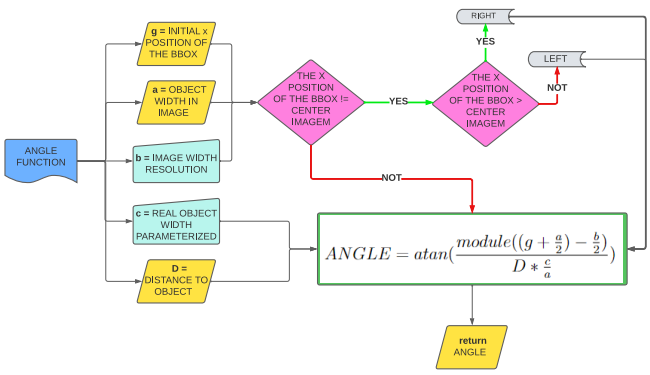


Fig. 6. Flowchart of the angle calculation function.

IV. Results

The training results obtained from the Google Colab script regarding the detection of objects using the YOLOv4 Tiny model, had great error reduction over the decades of training, achieving an accuracy (mAP) of 100% by the end of training.

The tests were made hands-on, using real-time image acquisition with a refresh rate close to 1 fps. Therefore, the module presented in section III-A1 was used, changing the image acquisition settings to a resolution of 640×480 pixels. The system was tested to detect and estimate the object distance and orientation through the acquired images.

Firstly, the applicability of the identification algorithm was verified. The bounding box is obtained with the box positioned at an arbitrary distance and using a smaller and generic training. An example of the detected object is presented in Figure 7, where the green bounding box represents the model's response and, in blue, the expected.

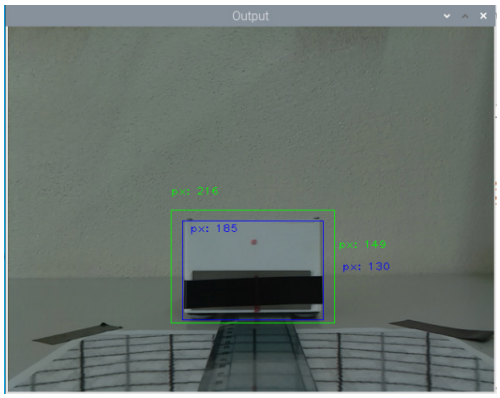


Fig. 7. Comparison of actual bounding box and detection model.

The bounding box formed to frame the area of interest was broader than expected, causing an error in estimating the localization parameters since the bounding box width was crucial. Thus, the error was parameterized for this specific example case. The difference between the actual contour and the one produced by the detection algorithm is calculated by

$$factor(X) = \frac{Object\_Width}{BBox\_width} = \frac{185}{216} \cong 0.85, \qquad (1)$$

which is the factor the algorithm uses to reduce the detection box on the image of its original size. Thus, even if the detection model does not fit the object in the detection box but has a good detection capacity, correction factors can be applied.

This kind of error occurs due to the way objects are marked in the training dataset. If they are displaced or turned differently, they are still identified, but not in the appropriate manner. Therefore, the object annotation and pre-processing phase is quite important and should be conducted as faithfully as possible with respect to their expected shapes and proportions.

The following experiments employed the data from a more specific training focused on detecting our area of interest: the larger frontal side face of the boxes. From these data, we tested the application of algorithms for object detection and distance calculation. Two experiments were conducted to validate the effectiveness of the system.

The first experiment intended to determine the system's range of operation regarding the distance.

The second experiment evaluate the system's effectiveness regarding the position of the object, taking different angles and distances, to identify possible patterns in the data obtained from these variations.

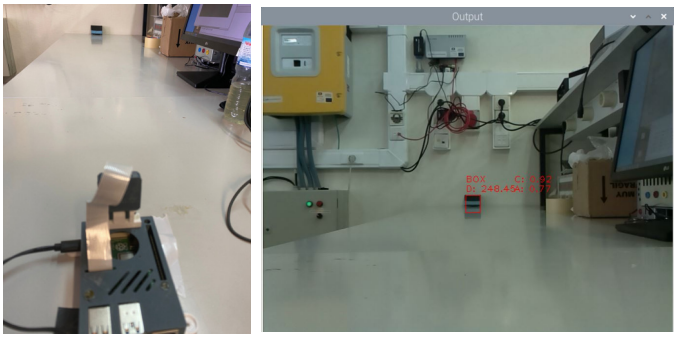
A. Distance range

Through practical tests, the detection range of the module was obtained where the box and the detection module were positioned face to face. Initially, the detection module camera and the box were center-aligned, almost touching one another. Then, the module was moved away in one centimeter increments, maintaining the alignment until the module detected the box. This detection refers to the minimum distance necessary for detecting the object. In the next step, to determine the maximum distance the module could detect objects, the module was moved away until the object was no longer detectable.

The test for measuring the maximum detection distance is shown in Figure 10. Figure 8 was shot to visually identify the module location, the box, and their distance. On the other hand, figure 8a displays the image shot by the module. In Figure 10(b), the information that appears are the object class, confidence, distance, and angle. The values obtained regarding the distance analysis are seen in table I.

TABLE I. Distance Analysis Calculation.

Distance	Actual	Average	Absolute Error
Maximum	218.5	250.7	32.2
Minimum	8.99	9.4	0.4



(a) (b)

Fig. 8. Maximum operating distance measurement. (a) Actual image of maximum distance (b) Image from module with calculated maximum distance.

(a) (b)

Fig. 9. Parameters at 30 cm and 10° at right. (a) Box and Module Positioning and (b) Result obtained by the detection module.

B. Position

The generalization of the model was obtained through the test that evaluated the performance of the detection module with respect to the distance and angle of the object's. It was performed with the module fixed on a polar graph sheet, and the camera aligned with the center of the graph. The experiment parameters use specifications from the Raspberry camera[9], whose hFoV is 53.5 degrees. Distances from 20 to 60 centimeters with steps of 10 centimeters were selected to perform measurements. The angles were defined from -20 to 20 degrees, with steps of 10 degrees, where 0 degrees is the central reference axis. For convenience, it was assumed that the direction of rotation had already been previously verified.

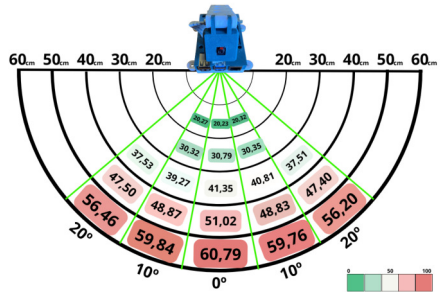
The tests were performed by changing only the position of the box. Ten measurements were made for each position, pointing the camera to the target box's frontal face. The measurements started from the center, 0 degrees, without any angle variation. Subsequently, measurements were made with the box positioned to the left and right, changing the distance and the positioning angle. An example of the positioning and the result obtained can be seen respectively in Figures 9a and 9b.

The first image in the Figures 10 and 11, represents the average of the 10 values obtained for distance and angle, respectively, for each position. The results presented show that the values can be lower or higher than expected for distance between object and modulus. Regarding the second image, the presented values represent the absolute error, in modulus, between the measured and the expected value. The smallest and largest values obtained were 0.16 and 3.8 cm for distance and 0 and 0.64 degrees for angle, respectively.

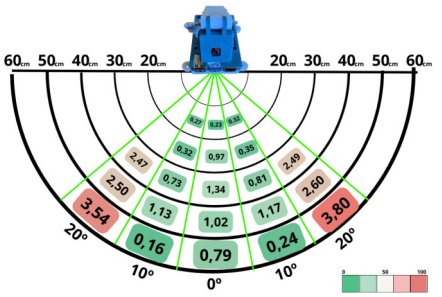
As only one class is to be detected, any object found will be classified as a box. In all cases, 100% of accuracy was obtained. The mean and the absolute error were assumed as metrics which are shown in Figures 10 and 11.

Results highlight the variations due to the change in the object's position. Regarding the average value of the samples

[9]https://www.raspberrypi.com/documentation/accessories/camera.html



(a)

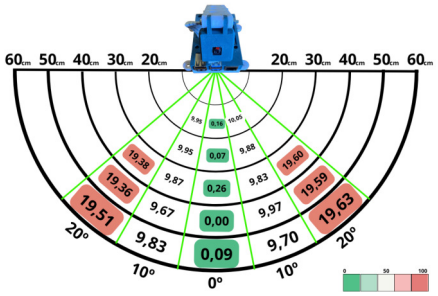


(b)

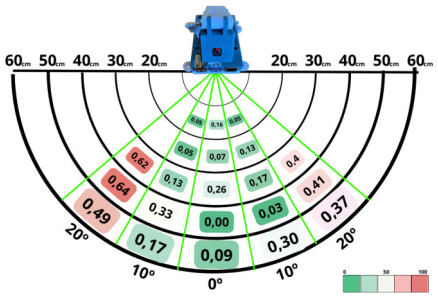
Fig. 10. Relative distance measurements based on 10 samples for each position of interest. (a) Mean (cm) and (b) Absolute Error.

in Figure 10a, the absolute error is presented in Figure 10b. The higher the distance, the higher the error, and near the central axis, the lower the error.

The results between items located at the same distance and angle but on opposite sides did not show equal values due to the irregularities and physical errors introduced, such as differences in lighting, imperfections in the module support and the measurement environment, and possible imperfections in the camera, among other factors. However, the values obtained were within the expected since the more distant the objects are, the more difficult it will be to visualize them, and,



(a)



(b)

Fig. 11. Relative angle measurements, based on 10 samples for each position of interest. (a) Mean (º) and (b) Absolute Error.

consequently, it will not be possible to assertively locate the boundary used as a parameter, resulting in errors regarding the estimated and actual values.

Based on the average angle values shown in Figure 11a, it was possible to calculate the absolute error, represented in Figure 11b. It was observed that an increase in the absolute error as the angular variation increased, being smaller near the center and more significant at the extremities in most of the values.

V. Conclusion

The completion of the project yielded a functional module that demonstrated reliability and accuracy based on the measurements taken. The distance and angle measurements resulted in maximum error percentages of 6.33% and 3.2%, respectively, with a minimum of 0.26% and 0%. The system operates effectively within a range of 9 to 218.5 cm and showed potential for error correction based on established standards. Additionally, the system's implementation utilized the cost-effective and easily-implementable pinhole model technique.

In future works, it is suggested to test the system in a robotic platform to evaluate its effectiveness in real conditions of use, such as the RobotAtFactory Lite competition, where it already has a simple, low-cost robot platform that can be adapted to carry the module [16]. This step is essential to enhance the system and allow for its application in several areas, contributing to developing efficient and accessible technological solutions.

Acknowledgment

The authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020). The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434). The fellowship code is LCF/BQ/DI20/11780028. João Braun is a PhD Student at the Faculty of Engineering, University of Porto (FEUP) supervised by Prof. Paulo Costa.

References

[1] J. L. Crowley, H. I. Christensen, and A. Chehikian, *Vision as process: basic research on computer vision systems*. Springer, 1995.

[2] R. A. Jarvis, "A perspective on range finding techniques for computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 122–139, 1983.

[3] M. H. F. M. Fauadi, S. Akmal, M. M. Ali, N. I. Anuar, S. Ramlan, A. Z. M. Noor, and N. Awang, "Sistema de navegação baseado em visão inteligente para robô móvel: uma revisão tecnológica," *Periódicos de Engenharia e Ciências Naturais*, vol. 6, no. 2, pp. 47–57, 2018.

[4] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, Part B, pp. 6019–6039, 2022.

[5] A. Elmquist and D. Negrut, "Câmeras de modelagem para veículos autônomos e simulação de robôs: uma visão geral," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25547–25560, 2021.

[6] L. Brancalião, J. Gonçalves, M. Á. Conde, and P. Costa, "Systematic mapping literature review of mobile robotics competitions," *Sensors*, vol. 22, no. 6, p. 2160, 2022.

[7] Z. Li, Y. Du, M. Zhu, S. Zhou, and L. Zhang, "Uma pesquisa de algoritmos de detecção de objetos 3d para desenvolvimento de veículos inteligentes," *Vida Artificial e Robótica*, pp. 1–8, 2022.

[8] L. Pérez, . I. n. Rodríguez, N. Rodríguez, R. Usamentiaga, and D. F. García, "Orientação de robôs usando técnicas de visão de máquina em ambientes industriais: uma análise comparativa," *Sensores*, vol. 16, no. 3, p. 335, 2016.

[9] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[10] B. B. e. o. Nair, "Detecção de objeto baseada em câmera, identificação e estimativa de distância," in *2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, 2018.

[11] A. Desouza, GN e Kak, "Visão para navegação de robôs móveis: uma pesquisa," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 2002.

[12] G. Adorni, S. Cagnoni, S. Enderle, G. K. Kraetzschmar, M. Mordonini, M. Plagge, M. Ritter, S. Sablatnög, and A. Zell, "Vision-based localization for mobile robots," *Robotics and Autonomous Systems*, vol. 36, no. 2, pp. 103–119, 2001. Viewing, Sensing, Coordination and Learning in EuroRoboCup 2000.

[13] M. Loureiro, F. S. Machado, R. Mello, and A. Frizera, "Desenvolvimento de uma estratégia de contagem e localizaçao de objetos utilizando uma câmera rgb-d e um robô móvel,"

[14] N. L. Werneck, A. H. R. Costa, and F. S. Truzzi, "Mediçao de distância e altura de bordas horizontais com visao monocular linear para robôs móveis," in *Anais do V Workshop de Visao Computacional*, 2009.

[15] G. R. Esteves, M. FEITOSA, and B. J. T. Fernandes, "Estereoscopia no cálculo de distância e controle de plataforma robótica," in *SIBGRAPI-Conference on Graphics, Patterns and Images*, pp. 65–70, 2011.

[16] L. E. Luiz, L. Pilarski, K. Baidi, J. Braun, A. Oliveira, J. Lima, and P. Costa, "Robot at factory lite-a step-by-step educational approach to the robot assembly," in *ROBOT2022: Fifth Iberian Robotics Conference: Advances in Robotics, Volume 1*, pp. 550–561, Springer, 2022.