Tracking the trajectory of a swarm of mobile robots with a computer vision system

Seguimiento de la trayectoria de un enjambre de robots móviles con un sistema de visión por computadora

Andrés Jiménez-Mora[1], Kevin Morales-Paz[2], Juan Carlos Brenes-Torres[3], Rebeca Solís-Ortega[4], Cindy Calderón-Arce[5]

Jiménez-Mora, A; Morales-Paz, K; Brenes-Torres, J. C; Solís-Ortega, R; Calderón-Arce, C. Tracking the trajectory of a swarm of mobile robots with a computer vision system. *Tecnología en Marcha*. Vol. 37, special issue. August, 2024. IEEE International Conference on BioInspired Processing. Pag. 44-49.

https://doi.org/10.18845/tm.v37i7.7296

1 Instituto Tecnológico de Costa Rica, Costa Rica.
andjm28@estudiantec.cr.
https://orcid.org/0009-0002-7446-832X
2 Instituto Tecnológico de Costa Rica, Costa Rica.
kevinmoralespaz@gmail.com
https://orcid.org/0009-0006-6833-3479
3 Escuela de Ingeniería en Mecatrónica, Instituto Tecnológico de Costa Rica, Costa Rica. Estudiante Doctorado en Automática, Robótica e Informática Industrial. Universidad Politécnica de Valencia, España.
juanbrenes@tec.ac.cr
https://orcid.org/0000-0001-6323-2173
4 Escuela de Matemática, Instituto Tecnológico de Costa Rica, Costa Rica.
rsolis@tec.ac.cr
https://orcid.org/0000-0002-3065-8386
5 Escuela de Matemática, Instituto Tecnológico de Costa Rica, Costa Rica.
ccalderon@tec.ac.cr
https://orcid.org/0000-0002-0077-225X

Keywords

Swarm robotics; computer-based vision systems; trajectory tracking; OpenCV

Abstract

Swarm robotics research uses a range of tools for evaluating the behaviors and metrics of robot collectives. One crucial tool involves the capability to track each robot's position and orientation at various intervals, enabling the reconstruction of individual robot poses and trajectories. Comprehensive analysis of swarm behavior hinges on the study of the collective trajectories of each robot within the group. This paper demonstrates the implementation of a computer vision system, utilizing a webcam and Python scripts, to effectively track a mobile robot group within a swarm. This shows the feasibility of developing such research tools using commonplace computing equipment. The design and development of the vision system, including a detailed calibration procedure, robot identification methods, and practical examples, are also shown. Furthermore, it offers an exhaustive explanation of the robot tracking process. Experimental trials with three robots validate the system's ability to extract images from video feeds and accurately identify each robot. Subsequently, after image processing, the system generates a dataset encompassing image numbers, robot IDs, x and y positions, and orientations.

Palabras clave

Robótica de enjambres; sistemas de visión por computadora; seguimiento de trayectorias; OpenCV.

Resumen

En el campo de la robótica de enjambres se utilizan una variedad de herramientas para evaluar los comportamientos y las métricas de los colectivos de robots. Una herramienta crucial implica la capacidad de rastrear la posición y orientación de cada robot en varios intervalos, lo que permite la reconstrucción de posturas y trayectorias seguidas por los mismos. El análisis exhaustivo del comportamiento de los enjambres depende del estudio de las trayectorias colectivas de cada robot dentro del grupo. Este artículo demuestra la implementación de un sistema de visión por computadora, que utiliza una cámara web y scripts de Python, para rastrear de manera efectiva un grupo de robots móviles dentro de un enjambre. Esto muestra la viabilidad de desarrollar tales herramientas de investigación utilizando equipos informáticos comunes. Adicionalmente, se muestra el diseño y desarrollo del sistema de visión, incluido un procedimiento de calibración detallado, métodos de identificación de robots y ejemplos prácticos. Además, ofrece una explicación exhaustiva del proceso de seguimiento del robot. Las pruebas experimentales con uno y tres robots validan la capacidad del sistema para extraer imágenes de videos e identificar con precisión cada robot. Posteriormente, después del procesamiento de imágenes, el sistema genera un conjunto de datos que abarca números de imágenes, ID de robots, posiciones (x, y) y orientaciones.

Introduction

In the subject of robotics, swarm robotics draws inspiration from the self-organizing systems found in nature, such as social insects, schools of fish, and flocks of birds. These systems exhibit collective behavior arising from simple local interactions [1]. This paper introduces the

PROE project [2], supported by the Costa Rica Institute of Technology, which focuses on the implementation of a swarm robotics prototype for exploring static scenarios and optimizing routes [3, 4, 5].

The project developed a group of robots called Atta-Bots. They are equipped with sensors to identify obstacles and transmit their position data (Figure 1a). The robots are capable of transmitting data about their position and orientation, which is captured by the data processing system to create a map of the area covered by the swarm. Using this map, optimal routes can be defined within this environment. In the case study, the environment is controlled and consists of a regular surface of granite mosaic, upon which a scenario with reconfigurable MDF walls is mounted (Figure 1b).

(a) Atta-Bots (b) Experimental configuration

Figure 1. Project description.

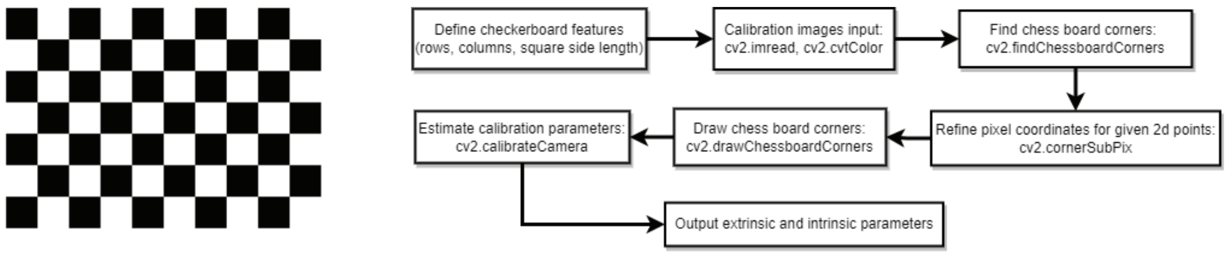
Materials and methods

Software tools

The foundation of the current vision system is the programming language Python and its library for Open Computer Vision (OpenCV), which consists of a collection of code intended for real time artificial vision, containing several functions for image interpretation. The library OpenCV is open source, it was first developed by Intel Corporation, and it is currently on its release version 4.8.0 [6].

Calibration

The purpose of this process is to determine the intrinsic and extrinsic parameters of a camera to accurately translate a three-dimensional point into a two-dimensional projection within the camera coordinate system. In this case, the plane pattern method was applied to perform the camera calibration [7, 8]. The checkerboard shown in figure 2a was used as the reference pattern by placing it in several spots within the experimental scenario and consequently, saving the images for further processing. The flowchart from figure 2b shows in a high level the implementation of the calibration process using OpenCV functions.



(a) Calibration pattern (b) Calibration process

Figure 2. Camera calibration.

Identifier design and detection

The developed implementation tracks the movement of the robots by identifying the displacement between pairs of images taken at the initial moment and the subsequent moment. To locate each robot in every image, an identifier composed of 2 circles, one large and one small, was used (Figure 4a). The designed identifier has a side length of 10 cm, which is the maximum allowed by the robot's dimensions. The size of the two circles was maximized by fitting the smaller circle inside the larger one, and different colors were used for each circle to differentiate them (Figure 4b).

(a) Identifier base design (b) Color identifier

Figure 4. Identifier pattern designed for the application

The chosen colors must stand out from the rest of the objects in the image. Therefore, striking and easily identifiable colors were selected, including blue, cyan, dark green, red, magenta, yellow, and orange. Using these colors, pairs were created, and the modified identifier designed earlier was used. The experimental setup with these identifiers is depicted in Figure 5.

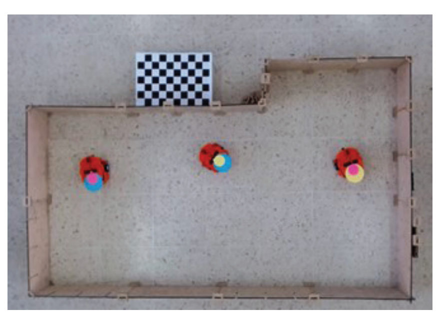


Figure 5. Camera view of the robots with color identifiers.

For the detection of the circles that make up the identifier, the Circular Hough Transform is used, with the utilization of minimum and maximum radius parameters for searching in the image, expressed in pixels.

Robot tracking process

The robot tracking process involves the initial capture of robot images, followed by a comprehensive analysis of these visual data. Through image processing function from OpenCV, the system extracts valuable information from the images, enabling the identification and localization of robots. The process steps are illustrated in the flowchart from figure 6, provide a brief explanation of each step within the robot tracking process. At the end of the processing of all the specified images, the program generates a text file (txt) where it reports the displacement between frames for each robot, and the angle that the robot has with respect to the horizontal axis.

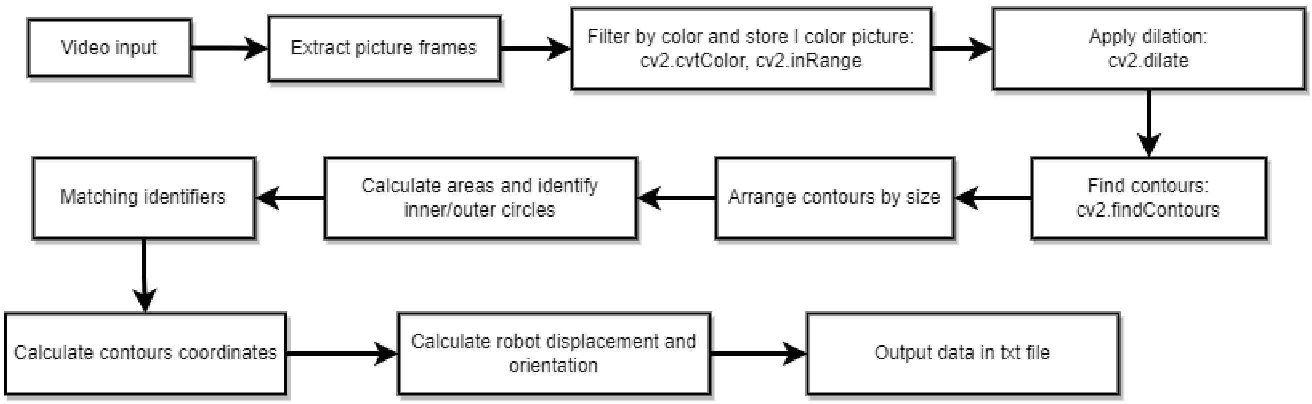
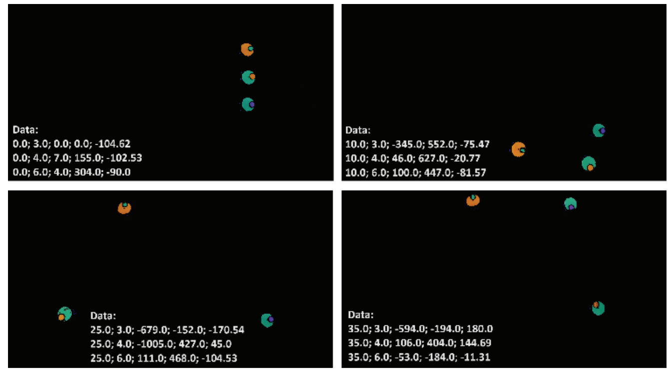


Figure 6. Diagram of the robot tracking process.

Results

The validation methodology was oriented to check if the system is able to detect and calculate the trajectory of the robot. Consequently, robots were placed on a controlled and closed scenario of an approximate dimension of 2x3 meters. Scenario walls were adaptable panels fabricated from medium-density fiberboard (MDF). A digital camera (webcam with a 1080p resolution) was placed on the ceiling, looking downwards. This environment presents abundant artificial light and smooth floor surface with a light granite color. The experiment was done using three robots. The swarm explored the scenario by employing a random walk behavior for about a minute. The present challenge was not only to track a robot and its trajectory, but to distinguish between robots. A video was recorded and a total of 48 pictures were extracted from it (see Figure 8a). Those pictures were successfully processed by employing the computer vision system. Each robot was distinguished and identified correctly by the system, and its trajectory tracked successfully. A sample of the processed pictures and its output data is shown on figure 8b.

(a) Sample of pictures



(b) Processed pictures

Figure 8. Experiment with three robot. Data shown represents: picture number, robot ID, position on x axis (mm), position on y axis (mm), and robot orientation (degrees). Data is over-imposed on picture for explanatory purposes.

Conclusions and future work

The behavior of a robot swarm can be comprehensively analyzed by studying the collective trajectories of each individual robot within the group. This paper has shown the implementation of a computer vision system, using a webcam and a set of Python scripts, to effectively track a group of mobile robots within a robotic swarm, using common computing equipment. Nevertheless, experiments conducted with three robots demonstrated the system's successful extraction of images from a video feed, accurately identifying each robot. Subsequently, after processing all the images, the system generated a dataset comprising image numbers, robot IDs, x and y positions, as well as orientation. Future research endeavors could investigate the application of this tool with a live video feed and real-time calculation of robot trajectories.

Acknowledgements

The authors would like to thank the Research and Outreach Department of the Costa Rica Institute of Technology (VIE, ITCR), for their support of the PROE project (code VIE 5402-1440-5101). These results are part of that project.

References

[1] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.

[2] J. C. Brenes-Torres, "jcbrenes/proe: Proyecto proe: Implementación de un prototipo de enjambre de robots para la digitalización de escenarios estáticos y planificación de rutas óptimas."

https://github.com/jcbrenes/PROE, 2021

[3] C. Calderón-Arce and R. Solís-Ortega, "Swarm robotics and rapidly exploring random graph algorithms applied to environment exploration and path planning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019.

[4] R. Solis-Ortega and C. Calderon-Arce, "Multiobjective problem to find paths through swarm robotics," in *Proceedings of the 2019 3rd International Conference on Automation*, *Control and Robots*, 2019, pp. 12–21.

[5] J. C. Brenes-Torres, F. Blanes, and J. Simo, "Magnetic trails: A novel artificial pheromone for swarm robotics in outdoor environments," *Computation*, vol. 10, no. 6, p. 98, 2022.

[6] Intel, Open-Source Computer Vision Library. Reference Manual, 2001.

[7] W. Qi, F. Li, and L. Zhenzhong, "Review on camera calibration," in *2010 Chinese Control and Decision Conference*, 2010, pp. 3354–3358.

[8] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.