Revista ELECTRO, Vol. 46, 2024, pp. 374 - 380
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

SURVEILLANCE TWO-WHEEL MOBILE ROBOT FOR PEOPLE AND OBSTACLE DETECTION IN RESTRICTED AREAS

Technical area: IoT
José Ignacio Vega Luna[1], José Francisco Cosme Aceves[1], Gerardo Salgado Guzmán[1], Francisco Javier Sánchez Rangel[1], Víctor Noé Tapia Vargas[1], Víctor Gonzalo Rodríguez Tapia[1], Edgar Alejandro Andrade González[2]
[1]Área de Sistemas Digitales, [2]Área de Comunicaciones, Departamento de Electrónica, Universidad Autónoma Metropolitana-Azcapotzalco.
Av. San Pablo 420, Col. Nueva El Rosario, Alcaldía Azcapotzalco, C.P. 02128, Ciudad de México.
vlji@azc.uam.mx

RESUMEN.
Se presenta el desarrollo de un robot que sigue una línea, marcada en el perímetro de un laboratorio, cuyo objetivo fue que, al encontrar un obstáculo en el recorrido, capture la imagen frontal, marque en ella el objeto detectado y la envíe a una plataforma de Internet de las Cosas, así como al teléfono móvil del responsable del laboratorio. El funcionamiento del robot está controlado por un sistema embebido Raspberry Pi 5. Cuenta con un sistema de sensores infrarrojos que le permiten seguir la línea y un sensor ultrasónico a través del cual determina la distancia al obstáculo. Incorpora una cámara de vídeo de 12.3 megapíxeles y un puente H para controlar los motores de las ruedas. La programación se realizó en Python con la biblioteca OpenCV. Las pruebas realizadas demostraron que el robot tiene una precisión del 98.4% en la detección de objetos y personas.
Palabras Clave: Cámara de video, Internet de las Cosas, OpenCV, Python, Raspberry Pi, robot

ABSTRACT.
This study presents the development of a robot that follows a line, marked on the perimeter of a laboratory, whose objective was that, when encountering an obstacle on the route, it captures the frontal image, marks the detected object on it and sends it to an Internet of Things platform, as well as the mobile phone of the person in charge of the laboratory. The operation of the robot is controlled by a Raspberry Pi 5 embedded system. It has a system of infrared sensors that allow it to follow the line and an ultrasonic sensor through which it determines the distance to the obstacle. It incorporates a 12.3-megapixel video camera and an H-bridge to control the wheel motors. Programming was done in Python with OpenCV library. The tests carried out showed that the robot has an accuracy of 98,4% in detecting objects and people.
Keywords: Internet of Things, OpenCV, Python, Raspberry Pi, robot, video camera

1. INTRODUCTION

Many facilities such as laboratories, hospitals, data centers, bank vaults, flammable substance warehouses, and distribution and logistics centers, among others, have reserved areas with restricted access. These areas store equipment, high-value goods, and hazardous materials, and can only be accessed by service, maintenance, and security personnel using one or more security techniques and devices controlled by the site manager. The most important characteristic of these spaces is that not everyone can access them [1]. There are few people working in these areas, and in some of these areas, access is only allowed during certain hours of the day. Commonly, the furniture, shelves, and cabinets in these areas are installed in lines or rows and strategically distributed in such a way that there are corridors that must be free of obstacles for safe access [2]. Normally, these types of facilities have systems that permanently monitor the presence of people. Some of these systems are based on presence and motion sensors and CCTV, which in addition to alerting the presence of objects and people, activate lighting devices and alarms [3]. The types of sensors used are vibration, passive infrared (PIR), sound, and temperature. Most presence detectors, regardless of the type of technology used, have drawbacks that sometimes cause false positives and negatives, a limited range, dependence on technology, sensitivity, interference, maintenance, installation, and cost. It is important that presence and motion detection systems function optimally, efficiently, and permanently, regardless of their type, as it is necessary to always be prepared for any eventuality or alert situation to safeguard access to the facilities [4].

Sometimes, traditional presence detection systems do not cover the entire area of these facilities, resulting in blind spots where an object or person may be found inappropriately or that may go unnoticed by surveillance personnel during the periodic tours they carry out. In recent years, the use of mobile robots that move around facilities to detect objects and people using image processing techniques has emerged as a technological alternative [5].

Mobile robots emerged during the 1990s. However, in recent years, a large amount of research and robotics has been developed, with diverse characteristics and functions that incorporate recent technologies to solve needs in many fields of human life and industrial processes [6]. There are three important fields in which the development of robots has focused: obstacle detection, navigation algorithms, and applications. Objects and people are detected using two main techniques. The first uses electronic devices, including sensors, video cameras, or thermal cameras, that activate an actuator or record the event. Some commonly used sensors are PIR, ultrasonic, laser, radio signal emitters, and light detection and ranging (LIDAR) devices. A disadvantage of this type of sensor is that it cannot detect certain types of objects and is affected by the reflective surfaces. The second technology combines the

use of a device, such as those indicated above, with a computer vision algorithm or method [7], [8].
Robots using the second technology use video cameras to obtain images of the environment, which can be used to recognize colors, textures, motion, and detect obstacles [9].
The objective of this study presented here was to create a low-cost mobile autonomous robot that travels around the interior perimeter of a computer laboratory to detect the presence of objects or people. The path followed by the robot is fixed and marked by a dark line on the floor. Detection was performed by processing images captured using a video camera installed in front of the robot. Upon detecting presence, the robot marks the object or person with a box on the image. This is sent to an Internet of Things (IoT) server and, together with an alert text message, is transmitted to the laboratory administrator's mobile phone. It is necessary that there are no objects in the robot's path; otherwise, they are considered obstacles. In other words, the path of the robot should normally be free of obstructions.
The mechanical structure of a previously built two-wheeled robot was used, in which a Raspberry Pi 5 embedded system, a group of infrared sensors, an ultrasonic sensor, an H-bridge and a high-resolution video camera were installed. The Raspberry system controls the elements used to track the line and process the captured images. Image processing consisted of a combination of frame differencing and contour-based motion detection algorithms. Programming was performed using Python and OpenCV.
The advantages and contributions of this work are as follows: 1) the use of the robot is not intrusive, since the laboratory infrastructure was not modified, only the route to follow was marked on the floor; 2) there is currently no commercial solution for the detection of objects and people; most do it using laser, infrared, or ultrasonic sensors whose cost is higher or have a shorter range than the solution presented here; 3) it allows the laboratory administrator or installation where it is used, react in a timely manner to the presence of a person not authorized to access the laboratory or an object or obstacle that is in the route; and 4) the technology used is recently created, the software tools used are open source, which reduces the complexity and cost of the system.
In a review of the literature and state-of-the-art studies carried out in recent years, regarding the topic of wheeled mobile robots (WMRs), most have focused on the growing number of applications in different fields, including teleoperation, control, overcoming obstacles and displacement outdoors, off-road, and terrains with different characteristics [10]]. Some of these studies have developed a multi-degrees-of-freedom two-wheeled inverted pendulum robot for the off-road transportation of long objects, pipes, or transferring patients [11]. Others have dedicated themselves to the navigation aspect and have improved the accuracy and robustness of robot inertial navigation systems [12], proposed adaptive control strategies to track a given trajectory [13], and created control schemes that stabilize the attitude and regulate the position of two-wheeled mobile balancing robots [14]. Likewise, studies have been carried out to design multi-wheeled mobile robots based on dissociation by degrees-of-freedom [15], projection-based algorithms to adjust the robot posture [16], distance- and velocity-based simultaneous obstacle avoidance and target tracking (DV-SOATT) methods for the trajectory tracking problem of multiple wheeled mobile robots [17], climbing performance of robots in complex environments [18], methods to generate a trajectory for two-wheeled mobile robots moving in an uncertain environment, where only a few waypoints are available to reach a nearby target state [19], and robotic systems applications to explore places that would otherwise be unsafe or unreachable to humans, such as volcanic and seismic areas, disaster and fire sites or unknown areas on other planets [20].
However, in recent years, interest in surveillance and disaster response scenarios has grown in the studies developed. Multi-robot seamless coverage systems have been created using the concepts of propagation [21], multi-robot systems with connectivity constraints [22], [23], multi-robot systems for perimeter surveillance [24], [25], proximity surveillance and data collection in Industrial IoT [26], robots for crowd surveillance [27], heterogeneous human–robot teams for exploration and patrolling [28], and collaborative multi-robot systems for search and rescue [29].

2. MATERIALS AND METHODS

The development of the application was carried out in two phases: hardware, implemented based on the Raspberry Pi embedded system, and system software.

2.1. System hardware.

The robot hardware integrates six components: Raspberry Pi embedded system, video camera, infrared sensors, H-bridge, gear motors, and ultrasonic sensor. Figure 1 shows a block diagram of the robot hardware.

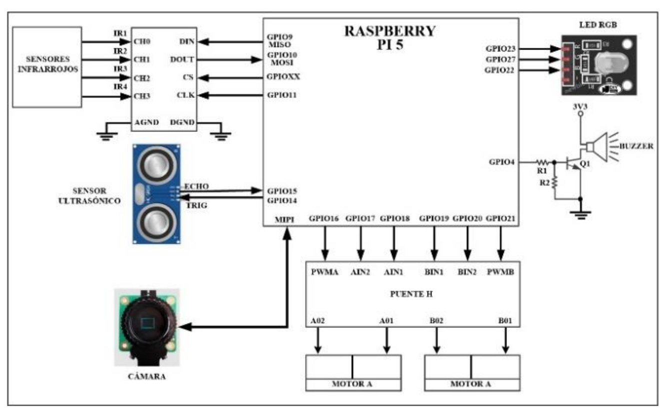

Fig. 1. Robot hardware block diagram

2.2. Raspberry Pi embedded system.

A Raspberry Pi 5 embedded system was used because it integrates the hardware elements necessary to implement the robot functions. The Raspberry Pi OS based on the Debian Bookworm was installed. This software is free access and is optimized for the Raspberry Pi hardware. It integrates Python 3 supporting more than 35,000 Debian packages and a considerable number of open-source function libraries for managing and processing images, including OpenCV. The

Raspberry Pi 5 system has the following main features: a 2.4 GHz 64-bit quad-core ARM Cortex-A76 CPU, 8 GB of SDRAM, two HDMI ports, dual-band 802.11ac and Gigabit Wi-Fi communication interfaces Ethernet, two USB 2.0 ports, two USB 3.0, a real-time clock, two Mobile Industry Processor Interface (MIPI) connectors for video cameras, and a 40-pin connector that includes general-purpose inputs and outputs (GPIO).

2.3. Video camera.

The camera integrated into the robot is a Raspberry Pi high-quality camera. This device has a 12.3-megapixel Sony IMX477 sensor with adjustable back focus, delivers a 7.9 mm diagonal sensor size, and provides images in RAW12/10/8 and COMP8 formats. The camera was connected to one of the Raspberry Pi MIPI ports.

2.4. Infrared sensors, H-bridge and gear motors.

The robot integrated four ITR20001/T infrared sensors to detect the dark line marked on the route to follow. These devices are composed of an infrared emitting diode and an NPN silicon phototransistor, which reflect the light emitted by the diode along the path line. When infrared light is reflected, the phototransistors that receive it provide an analog voltage signal. The voltage from the phototransistors is converted to a digital signal, so that it can be supplied to the Raspberry Pi 5 system. To achieve this, a four-input channel analog–digital converter was used. The converter was connected to the Serial Peripheral Interface (SPI) port of the Raspberry Pi5 system through the Master Output Slave Input (MOSI), MISO (Master Input Slave Output), Chip Select (CS), and clock (CLK) terminals. Infrared sensors, together with the H-bridge and gear motors, control motion and advancement of the robot.

The robot has five motion functions implemented in the software. These functions manage two N20 Mini Micro Metal Gear Motor Reducers, which are powered by 12 VDC and operate at a speed of 16 rpm. The motors were controlled by an H-bridge connected to six GPIO output terminals on the Raspberry Pi 5 system. Table I lists the logic levels supplied to the AIN and BIN input terminals of the H-bridge to drive the motors based on the motion. In the *forward* function, the values of both motors remain constant. To correct the route, one motor is stopped, and the opposite motor is advanced to resume the correct route based on the infrared sensor that detects the line. On *right*, the left motor was moved forward, and the right motor was reversed for 100 µs. For *left*, the right motor was advanced, and the left motor was reversed for 100 µs. For the 180° turn in the reverse function, this process was repeated for 200 µs.

Table I. Logic levels of the GPIO terminals connected to the H-bridge

Function	AIN terminal level for right motor		BIN terminal level for left motor	
forward	0	1	0	1
back	1	0	1	0
stop	0	0	0	0
right	0	1	1	0
left	1	0	0	1

The robot moves along the route at a speed of 10 cm/s and captures an image of the front every 500 ms. This is processed by the software and if an obstacle, an object or a person, is detected, the stop function is executed.

2.5. Ultrasonic sensor.

The robot has an HC-SR04 ultrasonic range sensor that allows it to determine the distance to an obstacle when it detects an object or person. This device offers a good cost-benefit ratio because it is economical, compact, easy to use, has a range of 2–400 cm, and an accuracy of 3 mm with a measuring angle of 15 °. It is composed of an ultrasonic transmitter, a receiver, and a control circuit. It has two control terminals; the TRIG input is used to tell the sensor to send eight 40 KHz pulses and wait for them to return when an object is found. If the pulses return, the sensor provides a pulse through the ECHO output, the duration of which allows the distance of the object to be determined. If no objects were detected, the echo pulse length is 36 ms. Both terminals were connected to the two GPIO terminals of the Raspberry Pi 5 embedded system.

2.6. System software.

The system software was constructed using Python 3. An important feature of the design is that it uses recent functions built into the OpenCV library to handle images focused on motion detection. Normally, the robot moves forward and activates the RGB LED connected to the GPIO22, 23, and 27 terminals of the Raspberry Pi 5 system in green. When it detects motion, it stops moving, activates the RGB LED in red, and the buzzer is connected to the GPIO4 terminal and sends the image showing the detected objects, marked with a rectangle, to the IoT server of the Wia Dashboard platform for registration and the WhatsApp alert text message to the mobile phone of the laboratory administrator.

To detect motion, a methodology was used that consisted of continuously capturing the video signal in front of the robot, reading two sequential frames of the video, comparing the previous frames, the first of which was considered the background, determining the difference between the two frames, obtaining the contour of the objects in the resulting image, excluding the small outlines, those whose area did not exceed an established size and were considered not to represent motion, and marked with a rectangle with a size greater than the established one. It should be noted that the robot moves at a moderate speed to obtain better results when comparing two successive frames in the video.

To implement the actions indicated above, the main program performs the following tasks configures the GPIO terminals and the Wi-Fi interface to connect to the Internet, starts video capture through the *cv2.VideoCapture* function, and enters a cycle in which it invokes the advance and detection routines. The advance routine controls the motion and stopping of the robot, as well as the state of the RGB LED, based on the five motion functions explained above. Figure 2 shows the flowchart of the main program.

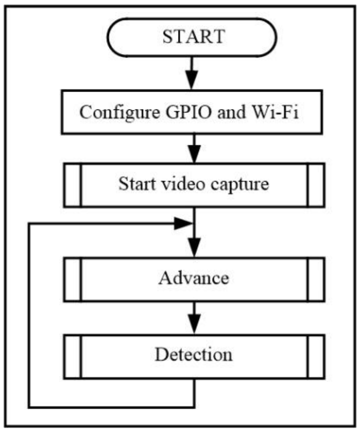

Fig. 2. Main program flowchart

The tasks performed by the detection routine were as following: a) read a frame of the captured video signal using the *video.read* function; b) transform BGR to grayscale, and reduce image noise by applying a bilateral filter using the functions *cv2.bilateralFilter* and *cv2.cvtColor*. The bilateral filter is a nonlinear filter that reduces noise by smoothing the images while preserving the edges and keeping them sharp. This filter uses the weighted average of the intensity of each pixel obtained from nearby pixels, instead of the intensity of each pixel. This type of filter is considered an edge-preserving filter because it does not average across edges. For this reason, the bilateral filter is close to the Gaussian filter; c) repeat the two previous tasks; d) perform the frame differencing, obtaining the difference between the two captured images, using the *cv2.absdif*f function; e) obtain the contours of the objects in the image resulting from the subtraction using the *cv2.findContours* and *cv2.drawContours* functions; f) exclude from the image objects whose area marked with the outline does not exceed a minimum size; g) marks on the image, with a rectangle, the objects detected using the *cv2.rectangle* function; h) invoke the routine that activates the ultrasonic sensor to determine the distance to the detected object or person; i) invokes the routine to send the image, the alert message, and the distance to the object or person; and j) invokes the 500 ms delay routine. Figure 3 shows the flowchart used to perform the detection routine.

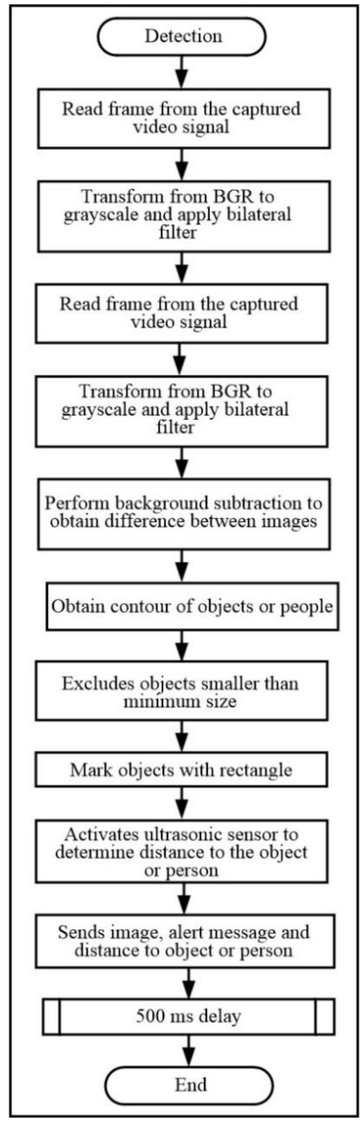

Fig. 3. Detection routine flowchart

Wia Dashboard is an IoT platform in the cloud that allows developing IoT applications and solutions. Through this platform, designers do not have the responsibility of managing the server, storage, and replication of the information necessary for the application, and the information is permanently available to users. Wia enables the control of remote devices, event collection, and information analysis and supports communication with Raspberry Pi embedded systems. For the robot to transmit the images to the Wia platform, a free-to-use software development kit (SDK) for Python was used, which is available to programmers through a set of APIs or functions to publish events to the platform and view them in real-time. In this application, the Python library for Wia's API was installed on the Raspberry Pi 5 system and the *sendWiaConnection* and *wia.Event.publish* functions were used to connect to the platform and send images, respectively.

The Twilio IoT platform was used to send WhatsApp alert messages. Twilio is a cloud service that provides, through the

Revista ELECTRO, Vol. 46, 2024, pp. 374 - 380
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

Twilio Sandbox for WhatsApp, a set of REST APIs that can be invoked in a Python program to transmit SMS and WhatsApp messages and create IoT applications. REST APIs work as an interface between the platform and the programming of remote systems, such as Raspberry Pi, for data exchange using the HTTP protocol. The data can be plain text or in the XML and JSON formats. REST APIs have an advantage over other IoT protocols because they are independent of the type of platform and programming language in which they are invoked. REST is a logical, efficient, and common standard for the creation and use of APIs in Internet services. The use of Twilio Sandbox for WhatsApp allowed the use of a shared phone number without waiting for a dedicated number to be approved by WhatsApp. Both the Twilio and Wia Dashboard offer different types of licenses according to the amount of information and SMS to transmit. For small quantities such as those used in this study, the service is free of charge.

3. RESULTS AND DISCUSSION

Two groups of tests were performed. The first objective was to determine the best value of the minimum size of an object or person detected in the sixth task of the detection routine. This makes it possible to eliminate small areas that could create false positives. To carry out these tests, a program was developed that performs the following tasks: captures an unobstructed image of the front of the robot, captures the second image with three objects and people in front, subtracts the two images, obtains the contour, discriminates the small objects, and mark the large areas with a rectangle. Twenty tests were run. In each test, the previous program was executed and the minimum value of the area of the objects or people was modified to determine the optimal value for the particular environment of the laboratory where the tests were carried out. In the first test, the minimum area was set to 8,000 pixels. In subsequent tests, this value was increased from 500 to 17,500 pixels. During the first two tests, ten false positives were detected, a number that decreased until there were no false positives, which occurred when the minimum surface value was 15,500 pixels, as indicated in the graph in Figure 4. Based on these results, we decided to use the last value in the detection routine.

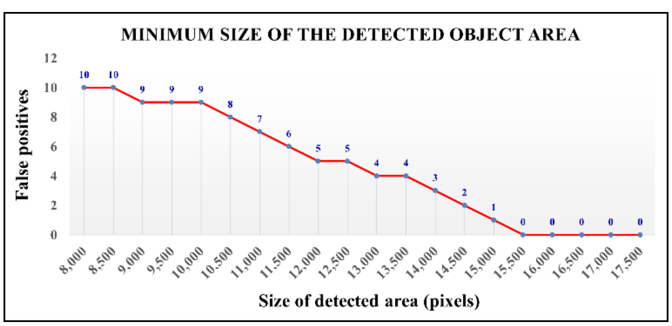

Fig. 4. Minimum size of the area of detected objects to send false positives

The second group of tests aimed to eliminate false|-negatives. Normally, there should not be more than five objects or people in the laboratory corridors, since access is restricted. Considering the above, ten sets of tests were performed. Each set consisted of five tests. In the first test, a person entered the corridor, and the robot detected him correctly. In the following four tests, two, three, four and five more objects or people, respectively, were placed in the corridor in front of the robot. Table II shows the results obtained, which includes the accuracy of each test and the average accuracy (98,4%) of the system in the detection. Figure 5 shows an image of a person detected using the contour marked by the robot software.

Table II. Logic levels of the GPIO terminals connected to the H-bridge

Set	Objects or people present	Objects or people detected	Accuracy (%)
1	1	1	100
	2	2	100
	3	3	100
	4	4	100
	5	4	80
2	1	1	100
	2	2	100
	3	3	100
	4	4	100
	5	5	100
3	1	1	100
	2	2	100
	3	3	100
	4	4	100
	5	5	100
4	1	1	100
	2	2	100
	3	3	100
	4	4	100
	5	4	80
5	1	1	100
	2	2	100
	3	3	100
	4	4	100
	5	5	100
Average accuracy ($\Sigma_{accuracy}$ / number of tests)			98,4

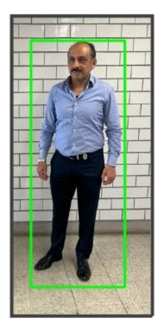

Fig. 5. Image of a person detected with the contour marked

4. CONCLUSION

A two-wheeled robot was developed. The robot moves inside a laboratory at a speed of 10 cm/s was obtained. The robot moves following a dark line marked on the route followed by the corridors of the laboratory. As it moves forward, the robot captures frontal images every 500 ms and uses frame differencing to detect the presence of objects or people. The robot integrates an embedded system and high-resolution video camera. To process the images captured by the camera, programming was developed in Python with OpenCV library. When the robot detects an object or a person, it marks them in the image with a rectangle and sends it to the Wia Dashboard IoT platform server along with a WhatsApp alert message to the laboratory administrator's mobile phone. This will allow you to react in a timely manner to an event. The robot represents a measure complementary to the security mechanisms of a laboratory.

The tests carried out made it possible to adjust the programming to eliminate false positives and negatives and determined that, when detecting objects or people, the robot has an accuracy of 96%. Both the speed at which the robot moves and the period between capturing the frontal image can be adjusted according to the type and characteristics of the facility in which the robot is used.

The robot follows a line marked on the floor and is therefore a static route. In the developed application, it was sufficient to use a robot with two wheels, since it performed stable motions on a flat path. However, it does not have the ability to overcome obstacles on rough or uneven terrain. This study began to integrate two functions. The first is an autonomous navigation module that allows users to follow a dynamic route according to the obstacles or situations encountered in their way. This is planned to be performed using the image processing that was already started in this work. Processing will be carried out using an edge TPU coprocessor and the TensorFlow platform to perform this task more quickly and efficiently, freeing the Raspberry Pi 5 embedded system from this activity. The second function is facial recognition. When the robot detects a person, it attempts to recognize and identify the face using a database of known people to indicate their identity to the laboratory or facility administrator.

REFERENCES

[1] L. -H. Shen, A. -H. Hsiao, F. -Y. Chu and K. -T. Feng, "Time-Selective RNN for Device-Free Multiroom Human Presence Detection Using WiFi CSI," in IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-17, 2024, Art no. 2505817, doi: 10.1109/TIM.2023.3348887.

[2] J. -E. Kim, J. -H. Choi and K. -T. Kim, "Robust Detection of Presence of Individuals in an Indoor Environment Using IR-UWB Radar," in IEEE Access, vol. 8, pp. 108133-108147, 2020, doi: 10.1109/ACCESS.2020.3000796.

[3] L. Wu, F. Gou, S. -T. Wu and Y. Wang, "SLEEPIR: Synchronized Low-Energy Electronically Chopped PIR Sensor for True Presence Detection," in IEEE Sensors Letters, vol. 4, no. 3, pp. 1-4, March 2020, Art no. 2500204, doi: 10.1109/LSENS.2020.2976801.

[4] L. Wu, F. Gou, S. -T. Wu and Y. Wang, "SLEEPIR: Synchronized Low-Energy Electronically Chopped PIR Sensor for True Presence Detection," in IEEE Sensors Letters, vol. 4, no. 3, pp. 1-4, March 2020, Art no. 2500204, doi: 10.1109/LSENS.2020.2976801.

[5] X. Liang et al., "Purely Image-Based Pose Stabilization of Nonholonomic Mobile Robots With a Truly Uncalibrated Overhead Camera," in IEEE Transactions on Robotics, vol. 36, no. 3, pp. 724-742, June 2020, doi: 10.1109/TRO.2019.2961052.

[6] A. Stefek, T. V. Pham, V. Krivanek and K. L. Pham, "Energy Comparison of Controllers Used for a Differential Drive Wheeled Mobile Robot," in IEEE Access, vol. 8, pp. 170915-170927, 2020, doi: 10.1109/ACCESS.2020.3023345.

[7] Y. Zhang, L. Zhang, W. Wang, Y. Li and Q. Zhang, "Design and Implementation of a Two-Wheel and Hopping Robot With a Linkage Mechanism," in IEEE Access, vol. 6, pp. 42422-42430, 2018, doi: 10.1109/ACCESS.2018.2859840.

[8] Y. Huang and J. Su, "Visual Servoing of Nonholonomic Mobile Robots: A Review and a Novel Perspective," in IEEE Access, vol. 7, pp. 134968-134977, 2019, doi: 10.1109/ACCESS.2019.2941962.

[9] X. Deng, J. Liu, H. Gong, H. Gong and J. Huang, "A Human–Robot Collaboration Method Using a Pose Estimation Network for Robot Learning of Assembly Manipulation Trajectories From Demonstration Videos," in IEEE Transactions on Industrial Informatics, vol. 19, no. 5, pp. 7160-7168, May 2023, doi: 10.1109/TII.2022.3224966.

[10] W. Li, L. Ding, H. Gao and M. Tavakoli, "Haptic Tele-Driving of Wheeled Mobile Robots Under Nonideal Wheel Rolling, Kinematic Control and Communication Time Delay," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 336-347, Jan. 2020, doi: 10.1109/TSMC.2017.2738670.

[11] S. Lee et al., "Design and Implementation of a Two-Wheeled Inverted Pendulum Robot With a Sliding Mechanism for Off-Road Transportation," in IEEE Robotics and Automation Letters, vol. 8, no. 7, pp. 4004-4011, July 2023, doi: 10.1109/LRA.2023.3279585.

[12] X. Niu, Y. Wu and J. Kuang, "Wheel-INS: A Wheel-Mounted MEMS IMU-Based Dead Reckoning System," in IEEE Transactions on Vehicular Technology, vol. 70, no. 10, pp. 9814-9825, Oct. 2021, doi: 10.1109/TVT.2021.3108008.

[13] M. Cui, "Observer-Based Adaptive Tracking Control of Wheeled Mobile Robots With Unknown Slipping Parameters," in IEEE Access, vol. 7, pp. 169646-169655, 2019, doi: 10.1109/ACCESS.2019.2955887.

[14] S. -K. Kim, C. K. Ahn and R. K. Agarwal, "Observer-Based Proportional-Type Controller for Two-Wheeled Mobile Robots via Simple Coordinate Transformation Technique," in IEEE Transactions on Vehicular Technology, vol. 70, no. 11, pp. 11458-11468, Nov. 2021, doi: 10.1109/TVT.2021.3116296.

[15] H. Qi et al., "Control Strategy for the Pseudo-Driven Wheels of Multi-Wheeled Mobile Robots Based on Dissociation by Degrees-of-Freedom," in IEEE Access, vol. 8, pp. 155477-155491, 2020, doi: 10.1109/ACCESS.2020.3019342.

[16] T. Liu, C. Zhang, J. Wang, S. Song and M. Q. -H. Meng, "Towards Terrain Adaptablity: In Situ Transformation of Wheel-Biped Robots," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3819-3826, April 2022, doi: 10.1109/LRA.2022.3148486.

[17] X. Li, Z. Xu, Z. Su, H. Wang and S. Li, "Distance- and Velocity-Based Simultaneous Obstacle Avoidance and Target Tracking for Multiple Wheeled Mobile Robots," in IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 2, pp. 1736-1748, Feb. 2024, doi: 10.1109/TITS.2023.3312373.

[18] Y. Sun et al., "Design and Experimental Evaluation of a Multi-Mode Mobile Robot Based on Eccentric Paddle Mechanism," in IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 8607-8614, Oct. 2021, doi: 10.1109/LRA.2021.3112757.

[19] J. Kim, "Trajectory Generation of a Two-Wheeled Mobile Robot in an Uncertain Environment," in IEEE Transactions on Industrial Electronics, vol. 67, no. 7, pp. 5586-5594, July 2020, doi: 10.1109/TIE.2019.2931506.

[20] J. V. Salazar Luces, S. Matsuzaki and Y. Hirata, "RoVaLL: Design and Development of a Multi-Terrain Towed Robot With Variable Lug-Length Wheels," in IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 6017-6024, Oct. 2020, doi: 10.1109/LRA.2020.3010495.

[21] S. Lee, "An Efficient Coverage Area Re-Assignment Strategy for Multi-Robot Long-Term Surveillance," in IEEE Access, vol. 11, pp. 33757-33767, 2023, doi: 10.1109/ACCESS.2023.3260622.

[22] J. Scherer and B. Rinner, "Multi-Robot Persistent Surveillance With Connectivity Constraints," in IEEE Access, vol. 8, pp. 15093-15109, 2020, doi: 10.1109/ACCESS.2020.2967650.

[23] J. Scherer, A. P. Schoellig and B. Rinner, "Min-Max Vertex Cycle Covers With Connectivity Constraints for Multi-Robot Patrolling," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 10152-10159, Oct. 2022, doi: 10.1109/LRA.2022.3193242.

[24] L. Guerrero-Bonilla and D. V. Dimarogonas, "Perimeter Surveillance Based on Set-Invariance," in IEEE Robotics and Automation Letters, vol. 6, no. 1, pp. 9-16, Jan. 2021, doi: 10.1109/LRA.2020.3028055.

[25] X. Duan, D. Paccagnan and F. Bullo, "Stochastic Strategies for Robotic Surveillance as Stackelberg Games," in IEEE Transactions on Control of Network Systems, vol. 8, no. 2, pp. 769-780, June 2021, doi: 10.1109/TCNS.2021.3058932.

[26] W. Hou, Z. Wei, X. Zhu, J. Cao and Y. Jiang, "Toward Proximity Surveillance and Data Collection in Industrial IoT: A Multi-Stage Statistical Optimization Design," in IEEE Wireless Communications Letters, vol. 13, no. 6, pp. 1536-1540, June 2024, doi: 10.1109/LWC.2024.3376774.

[27] S. C. Sethuraman, P. Kompally and S. Reddy, "VISU: A 3-D Printed Functional Robot for Crowd Surveillance," in IEEE Consumer Electronics Magazine, vol. 10, no. 1, pp. 17-23, 1 Jan. 2021, doi: 10.1109/MCE.2020.3029769.

[28] M. Aggravi, G. Sirignano, P. R. Giordano and C. Pacchierotti, "Decentralized Control of a Heterogeneous Human–Robot Team for Exploration and Patrolling," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 4, pp. 3109-3125, Oct. 2022, doi: 10.1109/TASE.2021.3106386.

[29] J. P. Queralta et al., "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision," in IEEE Access, vol. 8, pp. 191617-191643, 2020, doi: 10.1109/ACCESS.2020.3030190.