Implementation of perspective-n-point techniques and YOLOv5 algorithm based on surveillance camera for localization

Siripong Pawako[1], Nopparut Khaewnak[2], Jiraphon Srisertpol[1,2]
[1]Mechatronics Engineering Program, School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand
[2]School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand

Article Info

Article history:

Received May 2, 2024
Revised Sep 12, 2024
Accepted Oct 7, 2024

Keywords:

Object detection
Positioning
Scale factor
Surveillance camera
YOLOv5

ABSTRACT

The technology of processors has advanced significantly, resulting in smaller and more powerful devices with much processing capability. Particularly, camera technology has witnessed extensive research in utilizing images for various applications. Currently, surveillance cameras are widely used for security purposes when abnormal events occur. In this research, the benefits of utilizing data from surveillance cameras are explored to assist in determining the position of a moving robot using the perspective-n-point (PnP) technique. the scale factor, which varies, has been improved by integrating checks with the YOLOv5 algorithm. This algorithm employs a custom model to specifically detect the robot of interest, enabling the determination of its real-world position using multiple surveillance cameras. These cameras have different perspectives within the same area. Considering the deviation caused by determining the position from a single viewpoint, multiple cameras are employed to mitigate this issue.

Corresponding Author:

Jiraphon Srisertpol
School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology
30000, Nakhon Ratchasima, Thailand
Email: jiraphon@sut.ac.th

1. INTRODUCTION

Recent advancements in processor, sensor, and computer technologies have accelerated the use of automated robots across various industries, particularly in the transportation sector. These robots are designed to perform a wide range of tasks autonomously without human intervention. However, a significant challenge in the development of autonomous robots is maintaining system stability and enhancing positional accuracy, which are crucial for safe and efficient application.

Although LiDAR-SLAM technology [1] has been used for precise indoor positioning, it faces limitations in detecting obstacles and navigating complex environments, including long-term mechanical errors during movement [2]. Additionally, the use of cameras in conjunction with Visual-SLAM techniques has gained popularity, but still faces challenges in environments with varying light conditions. If we compare the images from the camera at the pixel level to determine the location, [3] and [4] explain the principles of calculating and calibrating the position of an object using the camera. This is known as the perspective-n-point (PnP) technique, which [5] developed to improve measurement and positioning accuracy by calculating pixel-level comparisons. This technique has been applied in various applications, such as traffic camera calibration and visual speed measurement [6]. Other approaches include the use of ArUco codes with a single camera positioned above the robot, typically mounted on the ceiling and parallel to the floor, to calculate object positions [7], or LED-based visual positioning for AGV navigation [8].

Researchers have also explored and developed multi-sensor fusion techniques, combining data from multiple sensors to enhance positioning, navigation, and mapping efficiency, as depicted in Figure 1, which shows the fusion sensor system used in hybrid SLAM [9].

By integrating such data, LiDAR odometry technology combined with IMU enhances the accuracy of predicting and processing vehicle positions and orientations [10]. The use of RGB-D techniques with ROS [11], along with the development of artificial intelligence techniques such as YOLOv5 and R-CNN for real-time image interpretation and object detection in industrial environments [12]–[18] continues to advance. Research has also examined the installation of cameras in various positions, such as ceiling-mounted cameras in warehouses to detect multiple robot tags, and the use of surveillance cameras in industrial settings to detect AGVs [19], [20]. Key technical challenges include managing real-time data and computational complexity, which have been addressed through mobile edge computing combined with 5G technology [21], [22].

To address the aforementioned limitations and to promote the integration of sensor data and fusion sensor technology with artificial intelligence from existing work, this research introduces a novel approach that integrates the YOLOv5 artificial intelligence technique with PnP for real-time robot positioning using surveillance cameras. Testing in a 36-square-meter area with four cameras installed at different angles will help verify and assess positional accuracy. This study aims to generate precise positional data, reduce positioning errors, and offer a cost-effective solution for using commonly available surveillance cameras to solve these challenges.

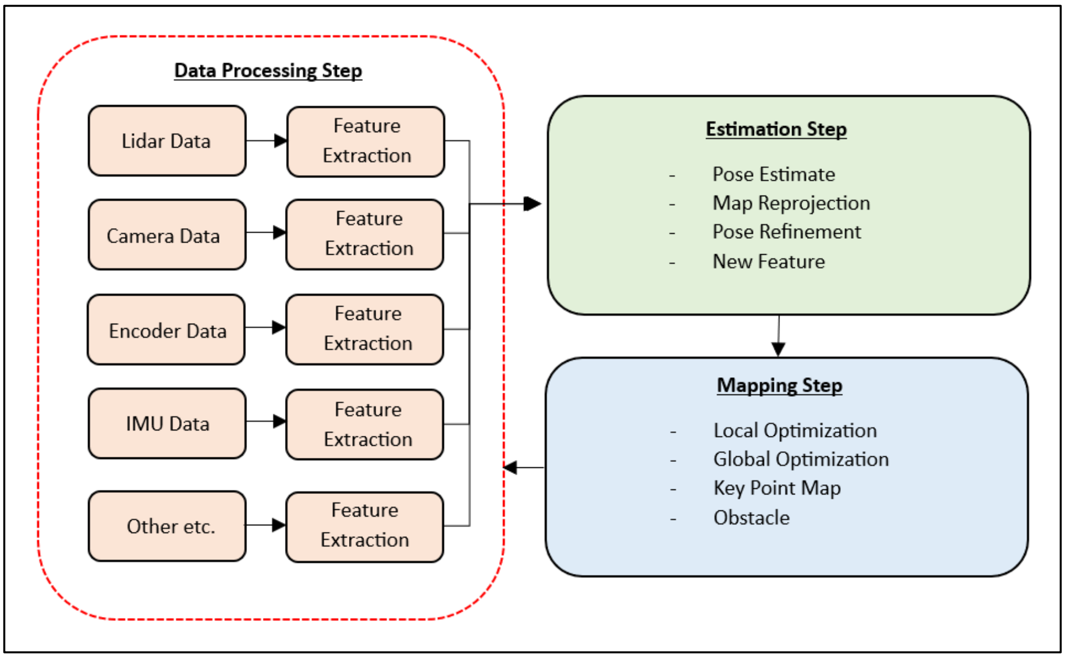


Figure 1. Fusion sensor for hybridized SLAM

2. METHOD

In this study, we utilized surveillance cameras installed within the building to monitor and track the position of the robot, along with calculating its real-time movement. This was achieved using the PnP technique and object detection through YOLOv5. Figure 2 illustrates the concept and experimental setup, which employed the building's surveillance cameras to collect data and accurately update the robot's position.

2.1. Surveillance camera

In this study, the Xiaomi C200 surveillance camera, as shown in Figure 3, was selected due to its 1920 x 1080 resolution, 360-degree rotation capability, and wireless data connection. These features make it well-suited for monitoring and tracking the robot's movements within the experimental setup. The camera was strategically positioned to ensure comprehensive coverage of the area under observation. The specifications of the camera are detailed in Table 1.

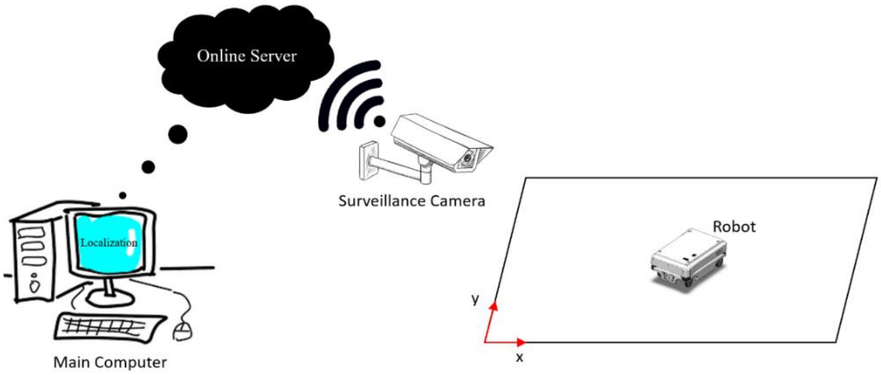


Figure 2. Concept for this research

Figure 3. Smart camera C200

Table 1. General information of smart camera C200

Feature	Value
Resolution	1920 x 1080 (2MP)
Power Supply	5V/2A
Rotation	360 Degree
Wi-Fi	IEEE802.11 2.4 GHz
Feature	Value

2.2. Robot

The Rosmaster X1 robot [23], as shown in Figure 4, was selected for this research due to its advanced control capabilities and a variety of sensors suitable for future exploratory missions. The robot is controlled by an Nvidia Jetson Nano board via the robot operating system (ROS) and is equipped with RPLidar and a stereo camera. These features make it highly adaptable and effective for the tasks required in this study.

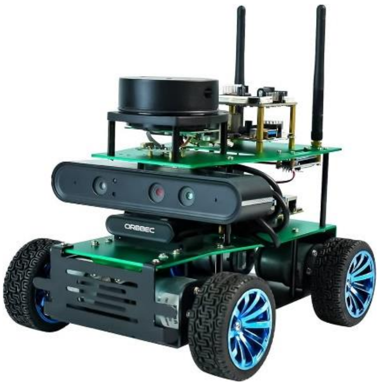


Figure 4. Robot Rosmaster X1

Upon receiving surveillance images, these are processed to localize the robot. The localization process utilizes YOLO object detection, which identifies the robot's position within the image and extracts the relevant pixels for further analysis.

2.3. Perspective-n-point (PnP)

The perspective-n-point (PnP) technique is crucial for determining the 3D location of an object based on 2D images captured by cameras. This technique is particularly important when the object and the camera are positioned at different distances. The pinhole camera model is used to represent the positional relationship between the object's real-world coordinates and its projection onto the camera's image plane, as illustrated in (1).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \{R\} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \{T\} \tag{1}$$

Here, R and T represent the rotation matrix and translation matrix, respectively. The coordinate system is expressed in a 3D format, where X_c, Y_c, Z_c are the coordinates in the camera's coordinate system, and X, Y, Z are the real-world coordinates. Meanwhile, the image data is typically in a 2D pixel format, defined by the image's width and height. Figure 5 demonstrates the transformation process.

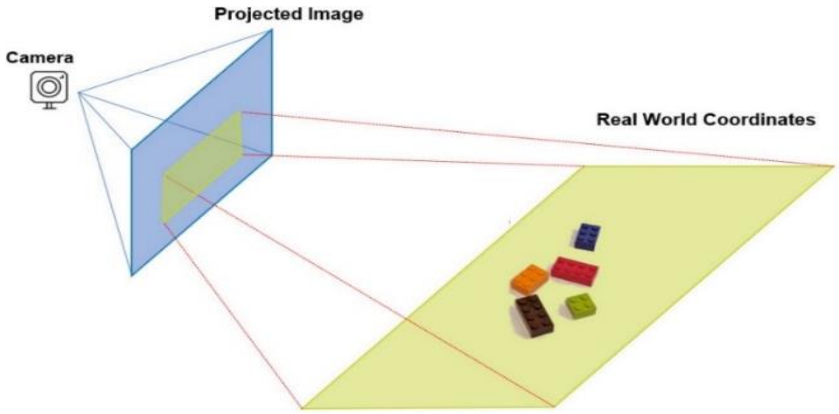


Figure 5. Transformation from 2D image coordinates to 3D world coordinates [4]

To convert a 2D image into a 3D coordinate system, it's necessary to establish the relationship between the image coordinates and the real-world coordinates. This conversion is performed using (2).

$$s \begin{bmatrix} w \\ h \\ 1 \end{bmatrix} = \{A\}\{R|T\} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

In this equation, w and h represent the horizontal and vertical pixel sequences, respectively, from the captured image. The scale factor, s, is crucial for converting the image coordinates into real-world positions based on the camera's perspective. The intrinsic matrix, A, is a key parameter used to identify the camera and compute the coordinates from both the original and undistorted images.

Finally, the accuracy of the position is evaluated using the error calculation formula, which determines the distance between the actual and the ideal position coordinates, as shown in (3).

$$E_m = \sqrt{(\mathrm{X}_r - X_m)^2 + (\mathrm{Y}_r - Y_m)^2} \tag{3}$$

Here, E_m is the resulting positional error, X_r and Y_r are the actual positions in the X and Y coordinates, respectively, and X_m and Y_m are the measured positions from the system in the X and Y coordinates, respectively.

2.4. YOLO object detection

Object detection is a critical component of our research, enabling the precise identification and localization of the robot within the captured images. YOLO, short for "You Only Look Once," [24] is a state-of-the-art, real-time object detection algorithm that has been specifically chosen for this task due to its efficiency and accuracy. Pre-trained on the comprehensive COCO dataset, YOLO distinguishes itself by employing a single neural network to process an entire image at once. This method involves partitioning the image into distinct regions, where the algorithm predicts probabilities and bounding boxes for each region.

For this research, YOLOv5 is utilized, which employs CSP-Darknet 53 in Figure 6 as its backbone architecture. CSP-Darknet 53 builds upon the convolutional network Darknet53, initially used in YOLOv3, and enhances it by integrating the cross-stage partial (CSP) network strategy. This improvement significantly boosts the model's ability to detect objects quickly and accurately, making it ideal for real-time applications in dynamic environments.

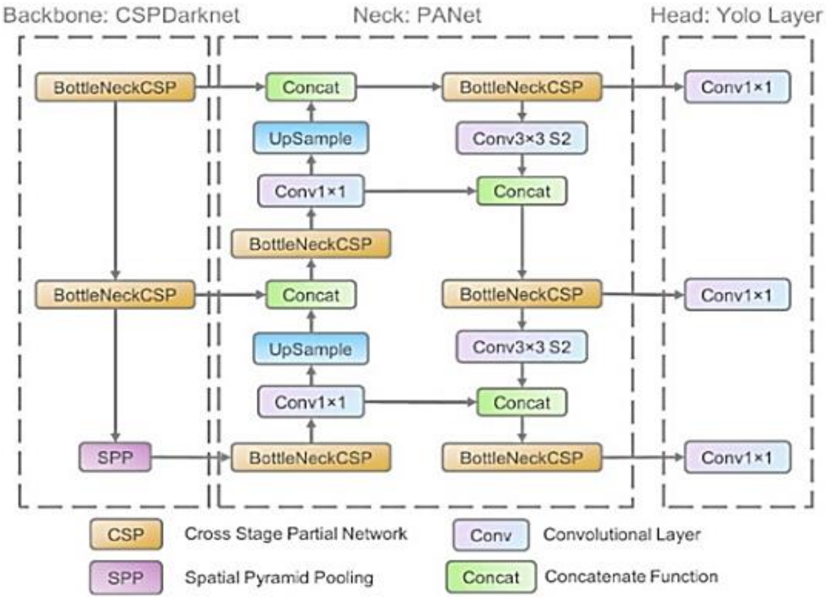


Figure 6. CSP-darknet 53 backbone architecture [25]

2.5. Our system

To address the issue of robot localization, this research presents a method of using surveillance cameras combined with the YOLOv5 artificial intelligence technique for real-time localization. The system workflow is depicted in Figure 7, showing the process from data acquisition to real-time localization.

Our system is developed using Python, where the surveillance cameras send data through an online server to an application called Xiaomi Camera Viewer. The system we created then extracts images from all four cameras using the real-time Snapshot method and combines the images using the OpenCV library. These combined images are then processed through the YOLOv5 model built with the PyTorch library. If a robot is detected within the field of view of any camera, the system extracts the pixel from the center of the bounding box and calculates the position using the PnP technique. The result is displayed as 2D coordinates using the Pygame library. The current position is identified based on which camera detected the robot, and this data can be utilized in future applications. The work presented in this study is divided into four parts

2.5.1. Training the object detection model

Our system is developed to detect and localize robots accurately using the YOLOv5 object detection technique. We created a dataset consisting of 49 images, as shown in Figure 8, which was expanded through a brightness generation process [26], increasing and decreasing the brightness by 5% to make the image data more flexible. This augmentation process increased the dataset to 147 images. Each image was meticulously labeled using the labeling algorithm [27], a popular algorithm for interpreting image content. The model's performance was evaluated using mean average precision (mAP) and loss values to ensure accuracy and robustness.

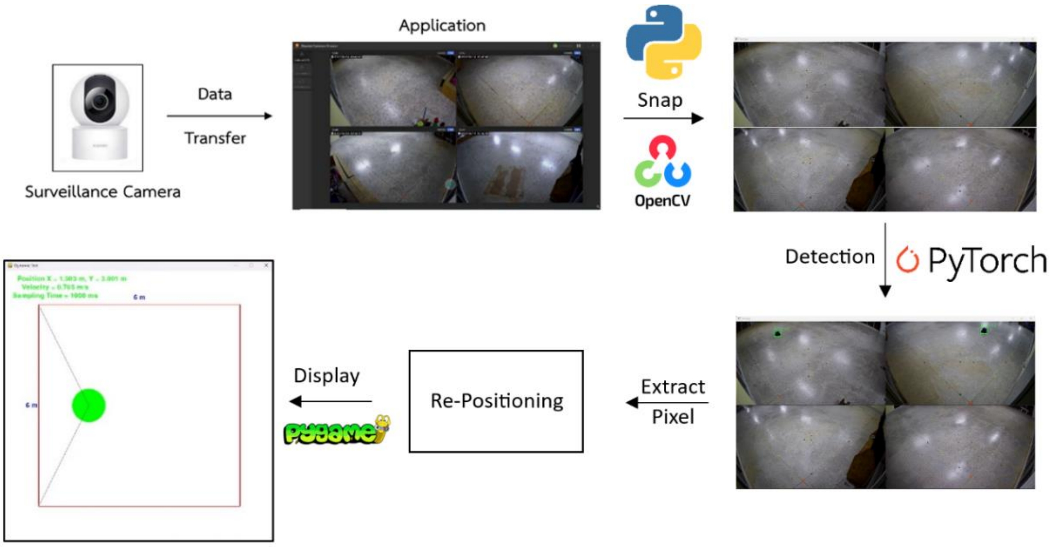


Figure 7. Workflow of our system

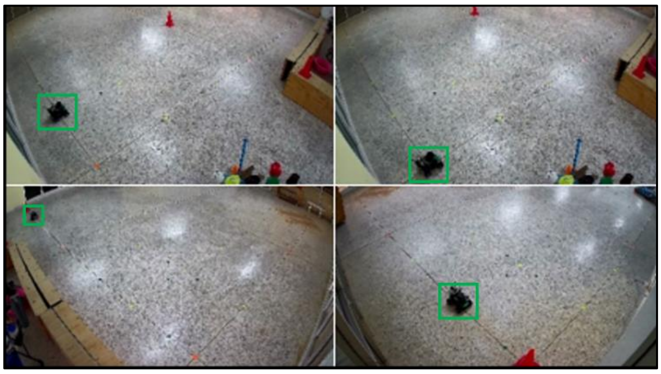
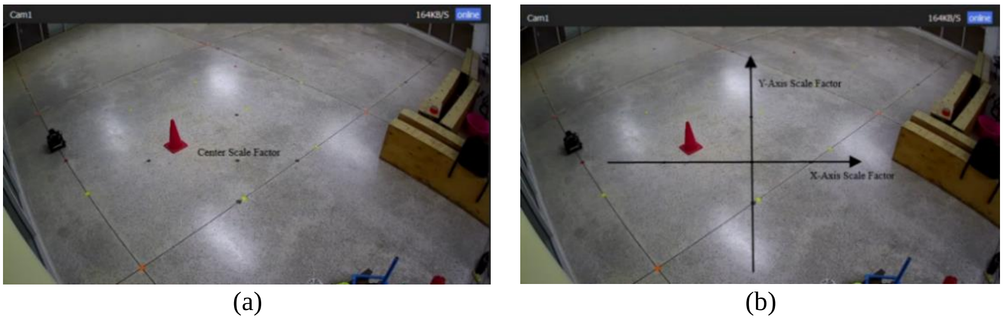


Figure 8. Sample dataset labeled using the labeling algorithm

2.5.2. Camera verification

To enable the cameras to accurately determine positions, it is necessary to calibrate them using chessboard calibration [28] to obtain the distortion factor and to align the camera's perspective with the test area (perspective calibration). Since the viewable area of each camera may differ, we performed a sub-view calibration for a 3x3 square meter area to ensure that each camera covers an equal area. The calibration used 16 reference points marked in orange and green. This process helps to derive the area coefficient for each camera by comparing parameters with the PnP equation to determine the scale factor at various points.



(a)

(b)

Figure 9. Scale factor consideration (a) center point and (b) axis of scale factor calculation

When calculating these values, it was found that the scale factor changes according to the distance from the camera, as will be shown in the experimental results section. As the camera moves further into distant areas, the scale factor increases. The calculation of the Scale Factor for each axis, x and y, is shown in Figure 9, with the resulting equation presented as (4).

$$s = \mathrm{S}_{center} + \mathrm{S}_{x-axis} + \mathrm{S}_{y-axis} \tag{4}$$

Here, from the three equations above, S_{center} represents the scale factor from the center of the area as shown in Figure 9(a), while S_{x-axis} and S_{y-axis} are functions of the scale factor along the X and Y axes of the area, respectively, as shown in Figure 9(b).

The additional equations for each axis of the scale factor are determined using a hybrid solution that combines polynomial and logarithmic equations to reduce errors at longer distances [29]. This method is beneficial for adjusting pixels both horizontally and vertically for position determination within the area. After calibrating all four cameras, separate localization tests were conducted for each camera to assess their performance in determining positions.

2.5.3. Static test

In the practical test for robot position detection, we used four surveillance cameras to cover an area of 6x6 square meters to test the detection and localization of stationary objects. The results will show the robot's positions at various locations, as depicted in Figure 10. The accuracy of each camera will be evaluated based on the detection and localization of the robot within the area, using the root mean square (RMS) Error to measure deviations. Single and overlapping positions detected by more than one camera as shown in Figure 11. Figures 11(a) and 11(b) respectively, will also be considered, and the results will be presented in the dynamic test.

2.5.4. Dynamic test

The dynamic test involved the robot moving along two predefined paths, as shown in Figure 12. The first path in Figure 12(a) moves from the reference point X=0, Y=0 to the opposite edge at position X=6, Y=6, while the second path in Figure 12(b) moves from X=0, Y=6 to the opposite edge at position X=6, Y=0. The performance evaluation was based on the position detection by the cameras, conducted separately for each camera and then combining the results from all camera positions.

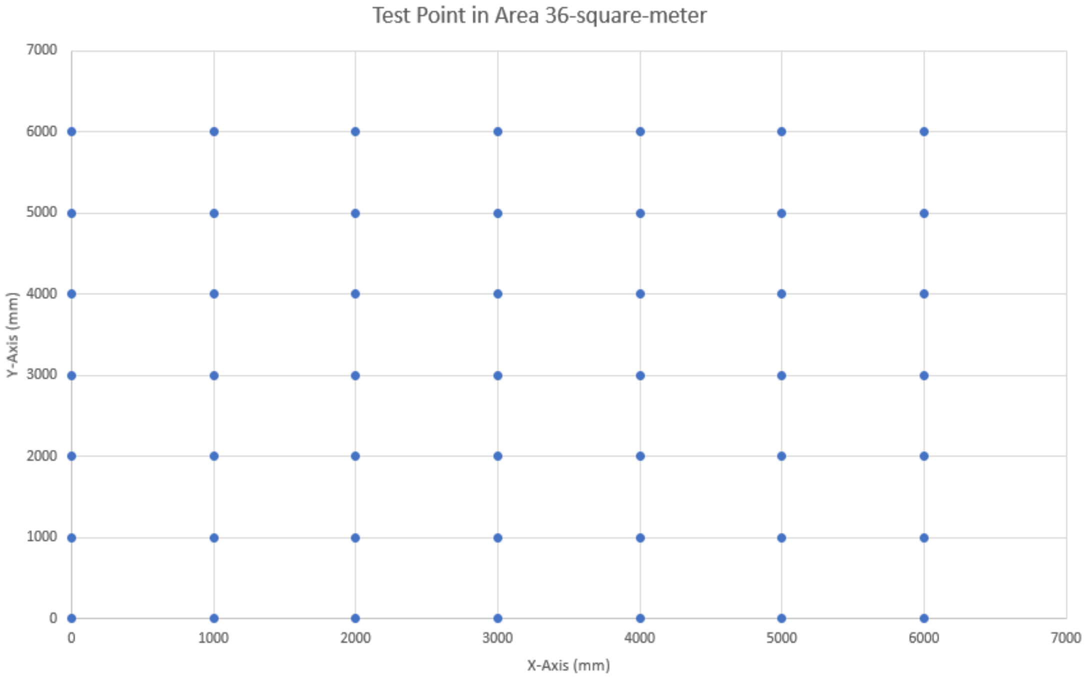
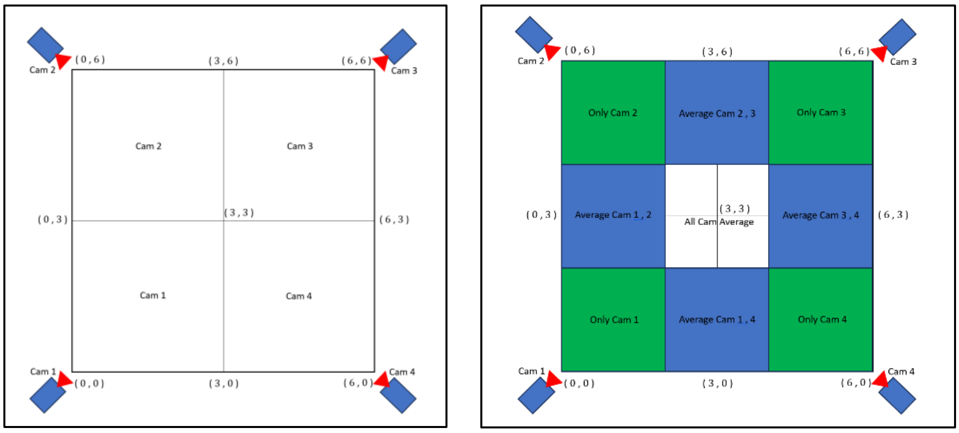
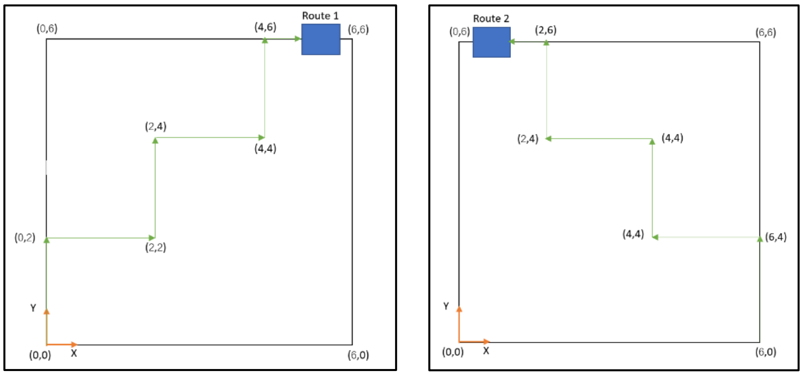


Figure 10. Points used to assess the position from all 4 surveillance cameras



(a) (b)

Figure 11. Determining the robot position from the camera (a) separate area with single camera and (b) combine method with multiple cameras



(a) (b)

Figure 12. Robot movement paths (a) path 1 and (b) path 2

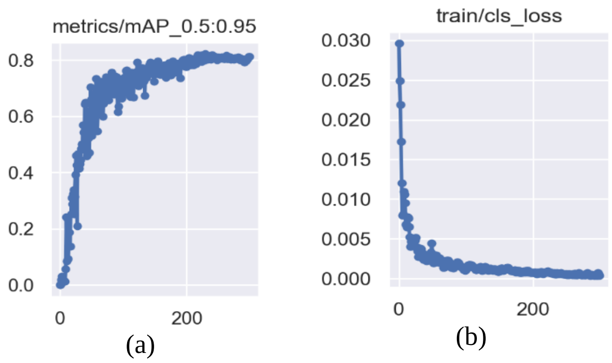
3. RESULTS AND DISCUSSION

3.1. Accuracy model

In training this model, we divided the dataset into 75% for training and 25% for validation, configuring the model to detect only a single class (Class 1) and setting the number of epochs at 300. The training results, shown in Figure 13, indicate a mean average precision (mAP) of 81.3% in Figure 13(a) and a decreasing loss throughout the training process Figure 13(b). This model is designed to be used with surveillance cameras to locate and identify the robot in the area, as will be demonstrated in the experimental section. The use of YOLOv5 for single-class detection, as discussed in this case, demonstrates a relatively high level of accuracy, with a mAP of 81.3% and a consistent reduction in loss during training. These results suggest that the model is effective in identifying and locating the target object within the dataset, making it a reliable tool for this specific application.

However, these results should be considered in the context of real-world application. While an mAP of 81.3% is good, there is still room for improvement, particularly if the operational environment is highly variable or if detection errors could have significant consequences. When using this model in conjunction with surveillance cameras to monitor and locate the robot in confined spaces, it is crucial to consider various challenges, such as lighting conditions, object occlusion, and the physical state of the robot, which could impact detection accuracy.

To improve the model's performance in the future, the dataset could be expanded to cover a wider range of scenarios, the model's parameters could be fine-tuned, or more advanced versions of the YOLO family, such as YOLOv8, could be tested for potentially better accuracy or speed. Additionally, post-processing techniques, such as adjusting the non-maximum suppression (NMS) thresholds, could help reduce detection errors or missed detections.

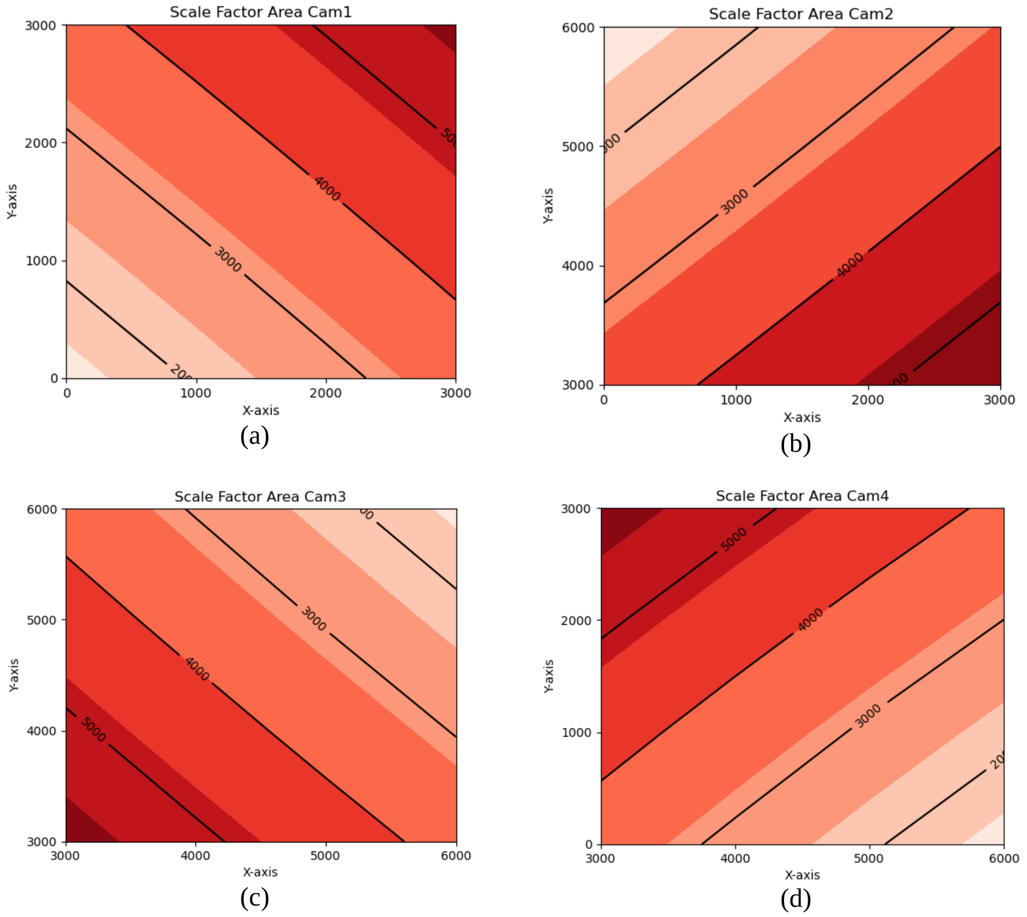


(a) (b)

Figure 13. The result of learning (a) the mAP of the custom model and (b) loss value

3.2. Camera verification

During the process of perspective calibration, it was observed that the scale factor changes relative to the distance from the camera, as shown in Figure 14 from all four cameras Figures 14(a)-14(d) respectively. As the distance from the camera increases, the scale factor correspondingly increases. The inconsistency in the scale factor, while generally increasing, is significant and therefore justifies the approach of calculating the scale factor separately for each axis in this position verification task.



(a) (b) (c) (d)

Figure 14. Scale factor value from back calculation: (a) scale factor from camera 1, (b) scale factor from camera 2, (c) scale factor from camera 3, and (d) scale factor from camera 4

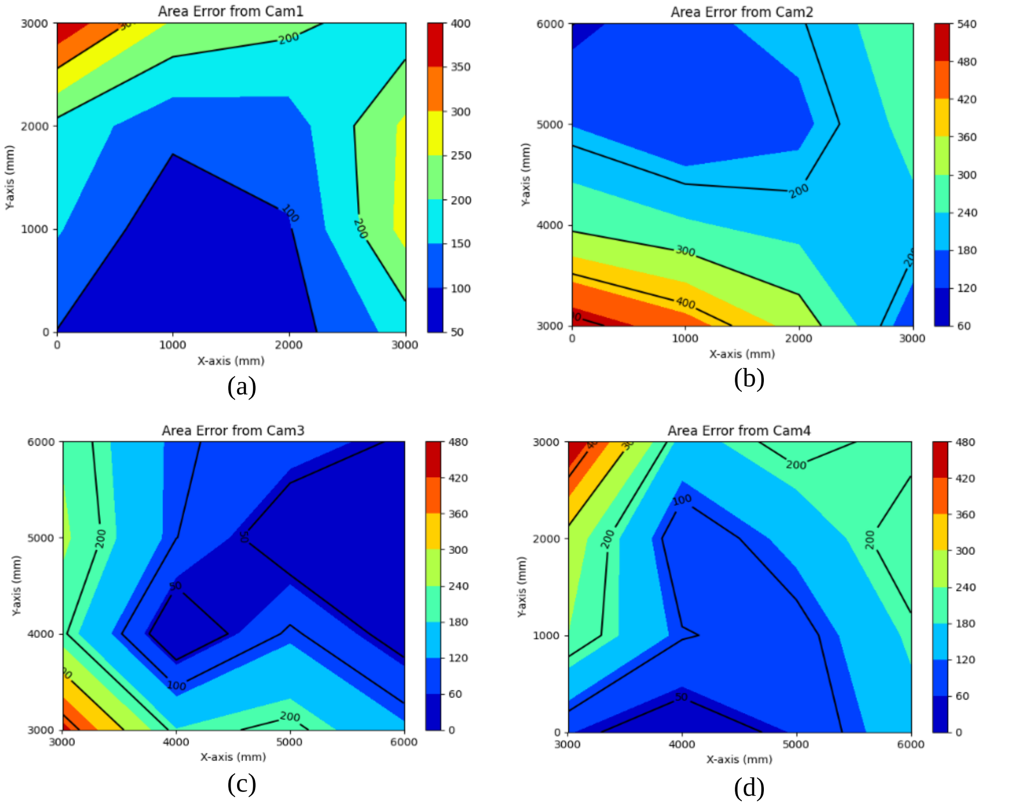
The observed variation in the scale factor during perspective calibration is critical for ensuring accurate position verification. The scale factor's dependence on distance highlights a key aspect of how perspective distortion affects measurements in a real-world setting. As the distance from the camera increases, objects appear smaller, and the corresponding scale factor must be adjusted to maintain accuracy in position detection.

The non-uniform increase in the scale factor across different distances is not merely a minor fluctuation; it has a significant impact on the accuracy of the calibration. This observation underlines the importance of considering the scale factor on a per-axis basis rather than assuming uniformity across the entire field of view. By tailoring the scale factor calculation to each axis, the model can account for these variations more precisely, thereby improving the overall accuracy of position verification.

This approach also mitigates potential errors that could arise from using a single, averaged scale factor across all axes. Such errors could lead to inaccurate position detection, especially in areas where the distance from the camera varies significantly. Therefore, the decision to compute the scale factor separately for each axis is not only justified but also crucial for enhancing the reliability and precision of the verification process.

3.3. Static test

The static test results reveal critical insights into the effectiveness of the surveillance camera setup for accurate robot position detection in Figure 15. The test demonstrated that minimal deviations occurred when the robot was positioned near the corners of the area for each camera in Figures 15(a)-15(d) respectively, which were closer to the cameras. This finding suggests that proximity to the camera is a significant factor in reducing localization errors, as cameras positioned at these vantage points provided more precise data, resulting in the smallest observed error, approximately 100 millimeters.



(a)

(b)

(c)

(d)

Figure 15. Positioning evaluation after calibration

However, as the distance between the robot and the cameras increased, particularly along the X-axis at 4 meters and the Y-axis at 3 meters, the localization errors became more pronounced, with the highest error recorded at 420 millimeters in Figure 16. This increase in error with distance indicates a limitation in the camera system's ability to maintain accuracy across the entire monitored area, especially in regions farthest from the cameras.

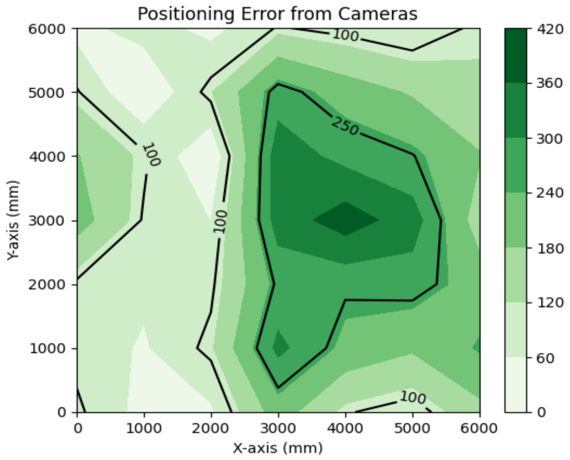


Figure 16. Performance of localization area by all surveillance cameras

The presence of overlapping camera coverage was intended to enhance accuracy by providing multiple viewpoints for triangulating the robot's position. However, the test results indicate that despite this overlap, the accuracy was still compromised at greater distances. This outcome suggests that the calibration of the cameras may not have been optimal, particularly for distant objects, leading to less accurate calculations of the robot's position based on the PnP technique.

The influence of camera calibration on the accuracy of the PnP technique highlights the importance of precise calibration in minimizing positional errors. Inaccurate calibration can distort the relationship between the camera's perspective and the actual position of the robot, resulting in errors that could affect the reliability of the system in practical applications.

To improve the system's performance in future tests, several strategies could be considered. Firstly, enhancing the calibration process to ensure that it is more accurate across different distances and angles could reduce the observed errors. Additionally, adjusting the camera placements or using higher-resolution cameras might help mitigate the increase in error as the robot moves further from the cameras. Finally, implementing advanced post-processing techniques to refine the position estimates, particularly in areas with greater distances from the cameras, could further enhance the accuracy of the system.

3.4. Dynamic test

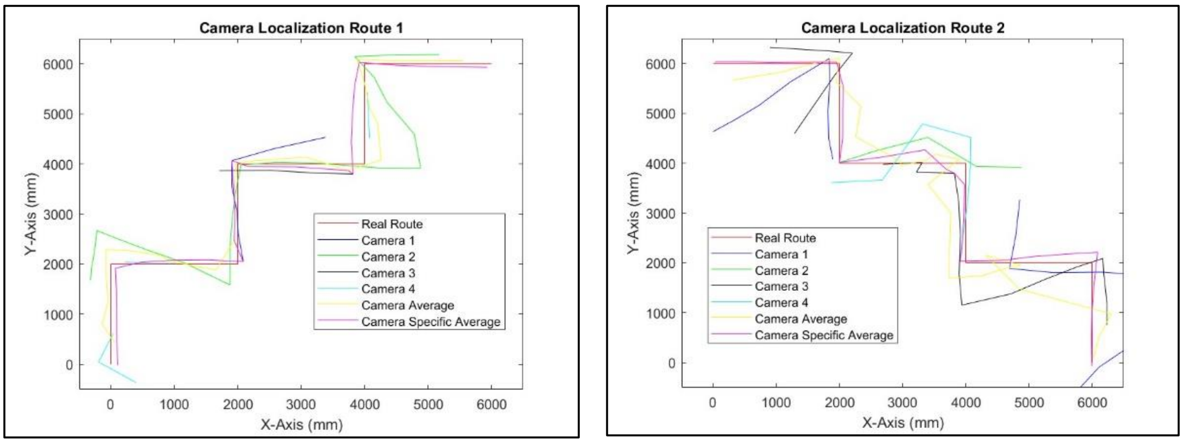
The dynamic test results provide valuable insights into the performance and limitations of the camera-based positioning system. The data indicates that while individual cameras can offer accurate position tracking, errors increase significantly as the robot moves farther away from the cameras. This observation aligns with the findings from the static test, reinforcing the importance of proximity in achieving high positional accuracy.

When the robot moves beyond 2 meters from any single camera, the use of the Combine Method, which averages data from multiple cameras in specific area, was intended to mitigate the increase in error. However, while this method provides continuous position tracking along the entire path, it also introduces higher errors, particularly in areas where the cameras are furthest from the robot. This increased error is most notable in the X-axis between 4 and 6 meters and the Y-axis around 4 meters, where the positional inaccuracies were consistent with those observed in the static test.

The main issue with the Combine Method lies in its averaging process, which includes data from cameras with higher errors. While this method maintains continuous tracking, the inclusion of less accurate data from distant cameras dilutes the overall accuracy, leading to higher positional errors. This is evident in the significant errors observed when the robot is farthest from the cameras, particularly in the X and Y coordinates.

Figure 17 illustrates the results of the robot's movement test, comparing the results of separate camera movement and averaged movement, including the combine method used to determine the average area from the side cameras. This method provided the best results compared to all assessments. In the first path, as shown in Figure 17(a), the movement passed through the aforementioned critical point, leading to high positional errors. In the second path, as shown in Figure 17(b), where the robot moved from X = 0 meters, Y = 6 meters to X = 6 meters, Y = 0 meters, the results were similar to the first path. When the robot reached the 4-meter mark along both axes, positional errors from the combine method increased, particularly in areas distant from the cameras. Camera 4 exhibited high errors at X = 6 meters and Y = 2 meters, contributing to the observed inaccuracies in the averaged positions.

To improve the accuracy of this method, future work could focus on refining the averaging process, perhaps by weighting the data from each camera according to its reliability or proximity to the robot. Additionally, enhancing the calibration of the cameras or employing more advanced algorithms for position estimation could help reduce errors, particularly in areas where the robot is distant from the cameras.



(a) (b)

Figure 17. Dynamic moving result (a) route 1 and (b) route 2

4. CONCLUSION

The advancements in accurately determining the position of autonomous robots using external data sources and integrating surveillance cameras in indoor environments represent a significant leap forward in technology. This research stands at the forefront of technological innovation by introducing a novel method that leverages artificial intelligence techniques, particularly the integration of the YOLOv5 algorithm with the PnP method. This combination enables real-time robot localization as it navigates through meticulously monitored environments, utilizing strategically placed surveillance cameras.

The experiments conducted within a 36-square-meter area, monitored by four cameras, involved the Rosmaster X1 robot navigating the space. The robot's position was determined using calibrated data from the four cameras, with a hybrid approach employed to minimize localization errors by adjusting the pixel scale and using a hybrid solution. The experiments were carried out in a meticulously calibrated environment, demonstrating the capability of the system to accurately track the robot's position in real-time as it moved through the monitored area.

The findings of this research were divided into four key sections. The first section involved collecting visual data of the robot to create a dataset for training a model developed using the YOLOv5 algorithm. The dataset was enhanced with brightness adjustments to increase data volume and suitability for varying lighting conditions. The trained model was then utilized in the second section, focusing on camera verification. All four cameras were calibrated using chessboard calibration and perspective calibration techniques, resulting in internal matrices that defined the relationship of translation and rotation vectors. The accuracy of robot localization within a 3x3 square meter sub-area was evaluated using the model from the previous section. The experiments highlighted errors, particularly with inconsistent scale factors during robot detection, and applied hybrid solutions to address this issue, reducing detection errors when the robot was close to the camera but increasing with distance.

The third section involved static testing by placing the robot at predetermined positions while recording the detected positions. The performance evaluation of the calibrated surveillance cameras revealed that localization errors increased as the robot moved further away from each camera. The maximum deviation, occurring at the center of the map at X=4 and Y=3, was 420 millimeters. The calculated positional errors, assessed through RMS, prompted improvements in the overall localization process by integrating data from all four cameras. This combined data was used in the fourth section, where the robot moved along two predefined paths. The performance of single-camera localization was compared to the combined data from all four cameras. The combined method demonstrated superior performance, particularly in scenarios where the robot was farther away, compared to the path-averaging method, which provided the most accurate positioning.

The outcomes of this research underscore a proactive approach to localization using widely available and cost-effective surveillance cameras, as opposed to relying on expensive sensors. This method not only demonstrates a cost-effective approach but also lays the groundwork for future engineering applications. The integration of external processing not only enhances localization accuracy but also represents a significant step forward in incorporating advanced technology within the domain of autonomous robots or automated guided vehicles (AGVs). This approach not only reduces the financial burden associated with specialized sensors but also paves the way for further technological advancements in efficient and cost-effective engineering solutions, particularly in autonomous navigation.

One of the most significant discoveries of this research is the validation of using external cameras for robot localization, which offers a stable and non-accumulative error-prone alternative compared to previous methods. By addressing and overcoming the challenges of traditional sensor-based localization, this method provides a robust solution that can be further developed and applied in a wide range of industrial and commercial applications. The potential for future development includes refining the hybrid method, improving camera calibration techniques, and expanding the application of this approach to more complex environments.

In summary, this research not only contributes to the field of autonomous robot localization but also opens new avenues for the application of affordable and scalable technology in precision positioning systems. The findings have the potential to influence future research and development, leading to more reliable and efficient solutions in the rapidly evolving field of autonomous systems.

ACKNOWLEDGEMENTS

We thank all contributors for their support. This project is funded by National Research Council of Thailand (NRCT) and Suranaree Medical Equipment Company Limited (SME), grant number N41A650405.

REFERENCES

[1] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, “A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6907–6921, Jul. 2022, doi: 10.1109/TITS.2021.3063477.

[2] J. Zhang and S. Singh, “Low-drift and real-time lidar odometry and mapping,” *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017, doi: 10.1007/s10514-016-9548-2.

[3] “A generic camera calibration method using object space error.” Accessed: May 15, 2022.[Online].Available: http://learn.gistda.or.th/wp-content/uploads/2017/06/GISTDA-Research-GI-2557-generic-camera-calibration-method.pdf.

[4] C. Vision, “Calculate X , Y , Z Real World Coordinates from Image Coordinates using OpenCV,” www.fdxlabs.com. Accessed: May 15, 2022.[Online].Available: https://www.fdxlabs.com/calculate-x-y-z-real-world-coordinates-from-a-single-camera-using-opencv/

[5] X. X. Lu, “A review of solutions for perspective-n-point problem in camera pose estimation,” *Journal of Physics: Conference Series*, vol. 1087, no. 5, p. 052009, Sep. 2018, doi: 10.1088/1742-6596/1087/5/052009.

[6] J. Sochor *et al.*, “Comprehensive data set for automatic single camera visual speed measurement,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1633–1643, May 2019, doi: 10.1109/TITS.2018.2825609.

[7] X. Hu, Z. Luo, and W. Jiang, “AGV localization system based on ultra-wideband and vision guidance,” *Electronics*, vol. 9, no. 3, p. 448, Mar. 2020, doi: 10.3390/electronics9030448.

[8] X. Liu, G. Wang, and K. Chen, “High-precision vision localization system for autonomous guided vehicles in dusty industrial environments,” *Navigation, Journal of the Institute of Navigation*, vol. 69, no. 1, p. navi.502, 2022, doi: 10.33012/navi.502.

[9] D. Vivet, A. Debord, and G. Pagès, “PAVO: a Parallax based Bi-Monocular VO Approach For Autonomous Navigation In Various Environments,” in *The International Conference on Digital Image & Signal Processing (DISP’19)*,

[10] H. Xue, H. Fu, and B. Dai, “IMU-aided high-frequency lidar odometry for autonomous driving,” *Applied Sciences*, vol. 9, no. 7, p. 1506, Apr. 2019, doi: 10.3390/app9071506.

[11] F. Spiess, J. Friesslich, T. Kaupp, S. Kounev, and N. Strobel, “Survey and experimental comparison of RGB-D indoor robot navigation methods supported by ROS and their expansion via fusion with wheel odometry and IMU data,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 12, pp. 1532–1540, 2020, doi: 10.18178/ijmerr.9.12.1532-1540.

[12] A. Aljaafreh *et al.*, “A real-time olive fruit detection for harvesting robot based on YOLO algorithms,” *Acta Technologica Agriculturae*, vol. 26, no. 3, pp. 121–132, Sep. 2023, doi: 10.2478/ata-2023-0017.

[13] J. Wang *et al.*, “Apple rapid recognition and processing method based on an improved version of YOLOv5,” *Ecological Informatics*, vol. 77, p. 102196, Nov. 2023, doi: 10.1016/j.ecoinf.2023.102196.

[14] Y. Sun, D. Zhang, X. Guo, and H. Yang, “Lightweight algorithm for apple detection based on an improved YOLOv5 model,” *Plants*, vol. 12, no. 17, p. 3032, Aug. 2023, doi: 10.3390/plants12173032.

[15] D. Yang, C. Su, H. Wu, X. Xu, and X. Zhao, “Shelter identification for shelter-transporting AGV based on improved target detection model YOLOv5,” *IEEE Access*, vol. 10, pp. 119132–119139, 2022, doi: 10.1109/ACCESS.2022.3220665.

[16] D. Yang, C. Su, H. Wu, X. Xu, and X. Zhao, “Research of target detection and distance measurement technology based on YOLOv5 and depth camera,” in *2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE)*, IEEE, May 2022, pp. 346–349, doi: 10.1109/CISCE55963.2022.9851025.

[17] P. Maolanon, K. Sukvichai, N. Chayopitak, and A. Takahashi, “Indoor room identify and mapping with virtual based SLAM using furnitures and household objects relationship based on CNNs,” in *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, IEEE, Mar. 2019, pp. 1–6, doi: 10.1109/ICTEmSys.2019.8695966.

[18] H. W. Cheong and H. Lee, "Concept design of AGV (Automated Guided Vehicle) based on image detection and positioning," *Procedia Computer Science*, vol. 139, pp. 104–107, 2018, doi: 10.1016/j.procs.2018.10.224.

[19] Q. Yang, Y. Lian, Y. Liu, W. Xie, and Y. Yang, "Multi-AGV tracking system based on global vision and AprilTag in smart warehouse," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 104, no. 3, p. 42, Mar. 2022, doi: 10.1007/s10846-021-01561-5.

[20] W. E. Barioni, I. P. Latini, A. Lazzaretti, M. Teixeira, F. Neves, and L. V. R. De Arruda, "AGV detection in industrial environments through computer vision," in *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, IEEE, Oct. 2022, pp. 1–6, doi: 10.1109/LARS/SBR/WRE56824.2022.9995994.

[21] S. Nakamura, S. Muto, and D. Takahashi, "Short-range Lidar SLAM utilizing localization data of monocular localization," *ROBOMECH Journal*, vol. 8, no. 1, p. 23, Dec. 2021, doi: 10.1186/s40648-021-00211-7.

[22] J. Peng, Y. Hou, H. Xu, and T. Li, "Dynamic visual SLAM and MEC technologies for B5G: a comprehensive review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, p. 98, Oct. 2022, doi: 10.1186/s13638-022-02181-9.

[23] Yahboom, "Yahboom RosmasterX1." Accessed: Oct. 16, 2023.[Online].Available: https://category.yahboom.net/collections/ros-robotics/products/rosmaster-x1

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[25] E. Daoud, D. Vu, H. Nguyen, and M. Gaedke, "Improving fake product detection using AI-based technology," in *Proceedings of the 18th International Conference on e-Society (ES 2020)*, IADIS Press, 2020, pp. 119–125, doi: 10.33965/es2020_202005L015.

[26] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, pp. 1–17, Feb. 2021, doi: 10.3390/f12020217.

[27] A. Punyathansombat, S. Pawako, K. Chamniprasart, and J. Srisertpol, "Visual inspection of the Hard Disk Drive components in fully automated assembly line using deep learning techniques," in *The SUT International Virtual Conference on Science and Technology 2021 (IVCST 2021)*, Nakhon Ratchasima, Thailand, 2021

[28] L. Tan, Y. Wang, H. Yu, and J. Zhu, "Automatic camera calibration using active displays of a virtual pattern," *Sensors*, vol. 17, no. 4, p. 685, Mar. 2017, doi: 10.3390/s17040685.

[29] S. Pawako, N. Khaewnak, P. Sripho, and J. Srisertpol, "Analyzing and Improving the Scale of Pixels in Images for Locating Moving Objects from Surveillance Cameras Based on Their Surroundings," in *The National RGJ and RRI Conference 2023*, Bangkok, Thailand, 2023

BIOGRAPHIES OF AUTHORS

Siripong Pawako is a Ph.D. Candidate in Mechatronics Engineering at Suranaree University of Technology. He holds a B.Eng. in Mechanical Engineering and an M.Eng. in Mechatronics Engineering, both from Suranaree University of Technology. During his master's studies, he specialized in analyzing and resolving vibration issues in industrial machinery, collaborating with Suranaree Medical Equipment Ltd. to develop machinery for medical laboratories in hospitals. His current research focuses on developing machine software integrated with artificial intelligence systems and creating low-cost sensor technology for the industrial sector. He can be contacted via email at: siripongpawako@gmail.com.

Nopparut Khaewnak received the B.Eng. and M.Eng. degree in Mechanical Engineering from Suranaree University of Technology, in 2007 and 2011, respectively. He worked at Rajamangala University of Technology Tawan-Ok Bang Phra Campus, Chonburi Province in 2014, and subsequently joined the Department of Mechatronics Engineering in 2015. He then took a leave of absence to pursue a Doctor of Philosophy program in Mechanical and Process System Engineering, starting in 2022 and continuing to the present. He can be contacted via email at: nopparut_kh@rmutto.ac.th.

Dr. Jirephon Srisertpol is an Associate Professor at the School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand. He got his Ph.D. degree in system analysis and control and processing information from St. Petersburg State University of Aerospace Instrumentation in Russia. He is the Head of the System and Control Engineering Laboratory. His research interests are in the area of mathematical modelling, adaptive systems and vibration analysis. He can be contacted at email: jiraphon@sut.ac.th.