2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Improving Indoor Localization: A Low-Cost, Multi-Marker and Multi-Camera System for Robot Tracking

Iuri Barros[1*], Ranulfo Bezerra[1], Rawin Assabumrungrat[2], Shotaro Kojima[1], Yoshito Okada[1], Masashi Konyo[1], Kazunori Ohno[1], Satoshi Tadokoro[1]

***Abstract*— Localization is a fundamental requirement for a wide range of robotic applications, but existing systems often require complex, resource-intensive and costly setups. We propose a cost-effective localization system that integrates multiple fiducial markers and multiple cameras for enhancing both pose estimation accuracy, detection range and frequency while reducing costs and providing camera placement flexibility. Our system reduces position RMSE from 13.45 to 3.6 centimeters (73% improvement) and can achieve 100% detection coverage while leveraging 3 to 5 cameras instead of 10, no IMU or odometry compared to our previous single-marker multi-camera system, MoCArU. When tested at different camera heights, our system outperforms the previous one in all evaluated conditions. It also increases the frequency of estimates, as determined by a qualitative analysis. Additionally, we evaluate various pose fusion methods, demonstrating that a simple and quick mean-based approach effectively maintains tracking accuracy with our system. This flexible, low-cost system provides a reliable and practical solution for indoor localization, making it a valuable option for various indoor tracking and monitoring applications.**

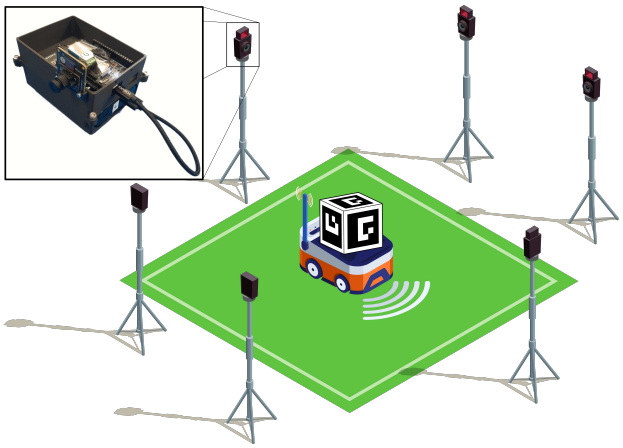

Fig. 1. A diagram illustrating the proposed robot localization system, which employs multiple ArUco markers using a cubic structure and Spresense cameras to track the robot's position.

I. INTRODUCTION

Spatial localization, the process of determining the xyz position coordinates of a target object relative to a reference frame, plays a crucial role across various fields. In robotics, for example, accurate localization is fundamental for tasks such as navigation [1], mapping [2], and object manipulation [3], enabling robots to operate effectively within their environments [4]. In asset tracking, indoor positioning systems ensure materials are correctly routed within buildings, enhancing operational efficiency. Furthermore, in Augmented Reality (AR) [5] and Virtual Reality (VR) [6], localization aligns virtual objects with the real world, creating immersive experiences where digital and physical elements seamlessly merge.

Despite significant advancements in localization technologies, many existing systems are designed primarily with high precision and accuracy in mind, often necessitating complex setups, high costs, and substantial computational resources. These systems are typically tailored for applications that demand fine-grained positional accuracy and high sampling rates, making them well-suited for certain high-stakes tasks but overly complex and costly for others [7] [8]. In scenarios where the primary need is monitoring rather than precision, such as tracking robots or cargo in indoor environments, these high-end systems can become unnecessarily burdensome. This creates a research gap for a more accessible, lightweight, and cost-effective localization solution that is simpler to deploy and better suited for tasks where high precision is not the primary requirement [9] [10]. An example of such a system is MoCaRu [11], our previous work, which is a low-cost IoT solution that localizes objects using an ArUco marker and the OpenCV library [12].

However, the challenge with systems like MoCaRu lies in their coverage and the requirement for cameras to be positioned at specific heights to ensure proper functionality. The system typically demands that cameras be placed at elevated positions to capture the top-facing ArUco markers on the robots or objects being tracked. This restriction not only limits the flexibility in camera placement but also reduces the system's accuracy and ability to cover larger areas effectively. Moreover, the previous system highly relied on odometry and IMU information of the robot, which in other applications may not be available. This height, accuracy and coverage limitations are significant obstacles that must be addressed to improve the system's versatility and overall effectiveness in resource localization and tracking.

This paper proposes a novel approach to address these challenges by integrating a cube with multiple ArUco markers on different faces [13] [14] with the previously proposed multi-camera system, enhancing it's range, accuracy and flexibility. This setup (shown in Fig. 1) also reduces the number of cameras required, as each camera can now detect multiple markers, improving the robustness and accuracy of pose estimation while reducing costs. Additionally, the paper

[1]Graduate School of Information Sciences, Tohoku University, Japan.
[2]School of Engineering, Tohoku University, Japan.
*Corresponding author barros.iuri@rm.is.tohoku.ac.jp

introduces a camera pose calibration algorithm based on Cumulative Moving Average (CMA), which enables reliable camera pose estimation even in the absence or occlusion of a world marker, further enhancing system reliability.

The contributions of this paper are as follows:

- An improved localization system that does not rely on robot odometry or IMU information to generate relatively accurate pose estimations.
- The introduction of a design that enhances camera placement flexibility and reduces the number of cameras required for effective operation.
- A simple and effective calibration method for camera pose estimation that mitigates noise from detection and eliminates non-detection issues due to marker occlusion, thereby improving final pose accuracy.
- A comparative analysis of different pose estimation techniques, providing insights into the effectiveness of various pose fusion methods.

II. METHODOLOGY

A. Materials and system overview

As in our previous work, the localization system combined an ArUco marker, tripods, power banks, and IoT devices. We now integrate an additional 3D printed structure for holding five ArUco markers in the shape of a cube. The result is a low-budget, standalone, portable and easy to use localization system. Additionally, the experimental setup also involves a TurtleBot [15] for evaluation of the system in a real application scenario. Fig. 2 depicts a TurtleBot with the proposed ArUco cube on the left side, and a photo of the experiment on the right side.

Image capturing and wireless distribution is performed by up to five IoT devices. Each device consists of a Spresense Extension Board [16] carrying one Spresense Main Board [17], with a Spresense HDR Camera board [18] and Spresense Wi-Fi iS110B Add-on attached to it, which can be seen in Fig. 1. These boards were responsible for broadcasting JPEG images in a VGA format, 640 pixels wide by 480 pixels tall, over HTTP at 24 FPS. On average, each wireless camera was delivering raw image messages at a rate of 5 to 10 Hz when observed from the host computer. At the host computer, ROS nodes are responsible to receive the MJPEG streams, perform ArUco recognition, relative pose estimation and pose fusion as shown in Fig. 3.

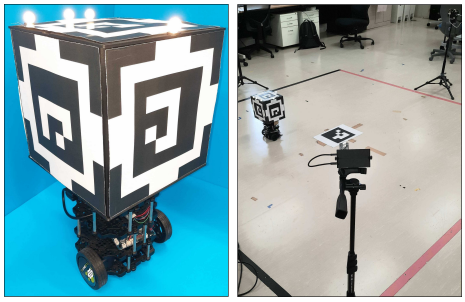

Fig. 2. The TurtleBot equipped with the ArUco cube (left) and a photo capturing the experimental setup (right).

B. ArUco cube

Relying on a single marker means that a camera can only produce one estimate of an object's pose. The ArUco board concept, in which multiple markers are used to estimate a single pose, was then extended to a 3D layout to improve pose estimation. For instance, [19] implemented a dodecahedron for improved detectability, [20] a 3D grid board, and others placed markers along the robot's body [21] [22]. In this study, we utilize an ArUco cube [14] to perform a single pose estimation, integrating multiple cameras to detect these multiple markers.

Unlike previous studies, our cube makes use of a single ArUco marker ID. In multi-robot applications, assigning five unique markers to each robot can significantly scale the ArUco dictionary, potentially leading to delays in detection or costing time for dictionary generation [23]. Knowing that the cube has a predefined geometry, the markers were strategically placed so that each can be identified by its orientation relative to a East-North-Up (ENU) coordinate system. For instance, the top marker generally has its z-axis pointing upward, while the right marker has its y-axis oriented upward. Based on this configuration, we can determine which side of the cube is being referenced, all while using the same marker ID. Every pose estimation from each ArUco marker is then assigned a suffix of its side, which is then used for fusion by translating and rotating these poses to a desired location. Please refer to [11] for further details on how the pose is estimated. Fig. 4 illustrates the transformation and fusion process. In this study, we transformed all four sides' information to the top as this was also the face holding the motion capture's reflective markers.

C. Camera Calibration

We implemented a camera pose calibration step to the low-cost localization system to overcome two issues. First, once the world marker becomes occluded, the localization of the camera, and thus of the robot, cannot be estimated. Second, every attempt to estimate the camera position relative to the world marker produces a slightly different estimate that is close, but not equal to the true position. By implementing a calibration step, we can use the calibrated camera pose to perform localization operations even in the absence of the world marker. Moreover, the noise on the robot's pose is eliminated and replaced by a small error derived from the calibration process.

The calibration begins by initializing the camera pose estimate with the first data point generated from detection of the world marker. As new data points are generated, the estimate is refined using the CMA method. This iterative process is performed for a predefined number of desired data points. In our application, convergence was often achieved after collecting approximately 200 points. Essentially, the calibration process refines the cameras' pose estimate over time by incorporating new data and using a statistical method

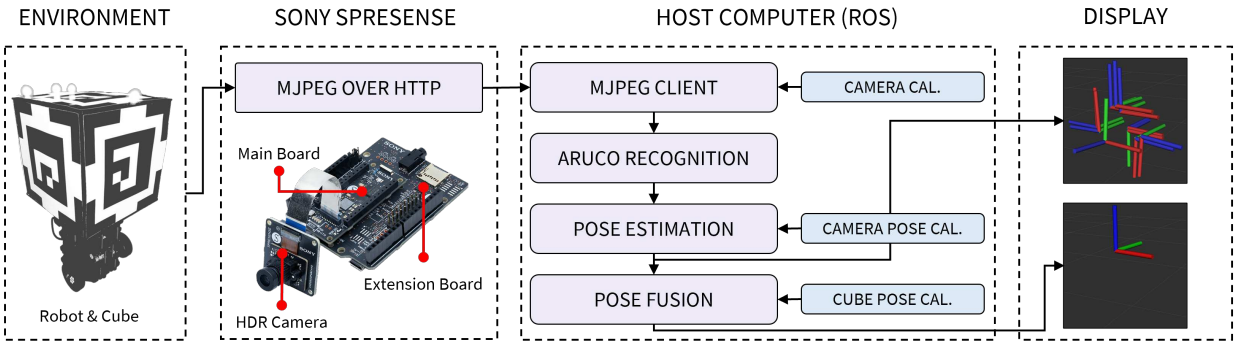

Fig. 3. System overview diagram. Sony Spresenses detect the robot equipped with the ArUco cube in the environment and stream MJPEG images over HTTP to a ROS system, which is responsible to perform the pose estimation.

to reduce noise. The calibrated pose information is then published as a ROS parameter, which can be accessed by nodes that require it for robot pose estimation. Once calibrated, there is no need for continuous re-estimation of the camera pose.

We employ the recursive version of the CMA to estimate translation (x, y and z) and rotation (quaternion's x, y, z, and w) of the camera relative to the world frame. The world frame is a 25x25 cm ArUco marker with ID 0, printed and placed on the floor. The recursive formula for the CMA is shown in (1).

$$CMA_t = CMA_{t-1} + \frac{d_t - CMA_{t-1}}{t+1} \tag{1}$$

where CMA_{t-1} is the CMA at the previous time step, and d_t is the new data point at the current time step t. This approach ensures convergence as the new values are slowly less effective in changing the estimate.

D. Pose fusion methods

To obtain a single, unified pose estimate from multiple sources, a fusion process is needed [24]. In this study, we compare three pose fusion methods: arithmetic mean, area-weighted mean, and area-based Gaussian combination. These methods are lightweight, making them suitable for real-time implementation and relatively straightforward to implement. Here, we discuss the implementation details of these methods, diving into their underlying principles, and how they were adapted to the specific application of this study.

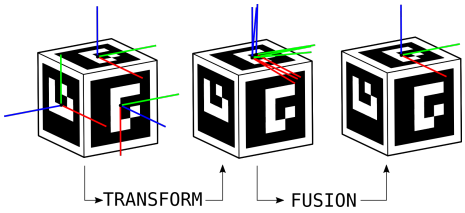

Fig. 4. The transform fusion process: detected poses are transformed to a common coordinate system and then fused for an accurate position and orientation estimate.

1) Arithmetic mean: The most straightforward way for fusing pose estimates is to calculate the arithmetic mean of the position coordinates. We regard this approach as a baseline because its simplicity makes it the most likely starting point for many. The resulting pose estimate, denoted as $\bar{p}$, is then obtained by averaging the individual poses p_i given by each i-th camera, as shown in (2).

$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{2}$$

2) Area-weighted mean: In [14], the authors increase pose estimation accuracy by leveraging weights into the arithmetic mean, where the weights are derived from the area of the detected marker in the image. The hypothesis is that cameras closer to a marker and with a direct line of sight can provide more accurate pose estimates, and that this information can be used to improve the final pose estimation [25]. In our novel approach for motion capture, we also implement this area-weighted mean, but instead of fusing the estimates of a single camera image, we combine multiple estimates generated by various cameras and markers.

The pseudo-area can be calculated from the pixel information of the detected marker corners and employed as a quality measurement of the estimated pose. Once the pose information and pixel area is available, we can combine these into a single estimate, weighted by the pixel area. To estimate the pixel area, we utilize the Shoelace Theorem [26] as shown in (3).

$$a_i = \frac{1}{2}\left|\sum_{j=0}^{4}(x_j^m \; y_{j+1}^m - y_j^m \; x_{j+1}^m)\right| \tag{3}$$

In the equation, a_i is the pixel area of a marker detected by the i-th camera, x^m and y^m are the Cartesian coordinates of each marker corner and j is the corner number, which in a squared marker goes up to four. After obtaining the pixel areas, the final position estimate of the robot can be calculated using a weighted-mean approach as shown in (4).

$$\bar{p} = \frac{1}{A}\sum_{i=1}^{n} a_i p_i, \quad A = \sum_{i=1}^{n} a_i \tag{4}$$

Here, A is the total pixel area from the set of n markers being used in the current estimation.

3) Area-based Gaussian combination: In our prior work [11], we proposed a Gaussian combination approach for fusing pose information. This approach relies on a variance measurement to make a Gaussian distribution, where the mean is the pose detected from the fiducial marker. The variance was derived from the difference between the ground truth and ArUco estimates in a given time frame. However, obtaining ground truth data can be challenging for many users who lack access to such systems. To address this, we propose a novel approach that uses the pixel area proposed by [14] as a proxy for variance in the Gaussian combination process [27].

Pixel areas can be directly used to perform the combination, or be used as the input to a function that maps areas to a proxy variance. During our tests, we tested different mapping functions like the square, cube, exponential and other linear approximations, while some gave similar results, a square function offered the best results, being simple and efficient. We employed the inverse of the square function in the combination process. Equation (5) depicts the function used to map pixel area into the proxy variance.

$$\sigma_i^2 = 1/a_i^2 \tag{5}$$

This variance should reflect the confidence of each camera's pose. By taking the inverse of the squared area, we have that smaller markers produce higher variances, and vice versa. The equation for the pose estimate using this area-based Gaussian combination is displayed in (6).

$$\bar{p} = \sigma^2 \sum_{i=1}^{n} \frac{p_i}{\sigma_i^2}, \quad \sigma^2 = \left(\sum_{i=1}^{n} \frac{1}{\sigma_i^2} \right)^{-1} \tag{6}$$

where σ^2 is the combined variance.

III. EVALUATION

We investigate the system's localization capabilities under various configurations: three varying camera heights (80, 100 and 120 centimeters); different number of cameras (3 and 5), and with and without the camera pose calibration data. At each height, the camera inclination was changed for better field of view (20, 30 and 40 degrees, respectively). For each height setting, four samples were collected to account for variability in the data.

A. Comparison of position estimations

Evaluation of the proposed system is initially performed using a qualitative comparison between the ground truth and the estimated positions. We provide scatter plots with projections of the position data on the xy-plane for different camera heights. These projections allow for a qualitative analysis of tracking accuracy and frequency of each system, highlighting any advantages of the novel system at different experimental conditions.

B. Tracking the position error

The primary goal of any motion capture system is to precisely track the pose of an object. To evaluate the proposed system effectiveness in tracking the pose of the robot, we employ the root mean square error (RMSE) metric on the pose information resulting from fusion as shown in (7). These are compared to the ground truth generated by a commercially available motion capture system, Vicon [28].

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2]} \tag{7}$$

Where N is the number of position messages, x, y, and z are the positions reported by Vicon at the closest time step relative to the $\hat{x}$, $\hat{y}$, and $\hat{z}$ positions reported by our system.

C. Detection ratio

In the former single-marker system, the ability to detect the robot was heavily tied to the camera positions. We hypothesized that placing additional markers to the robot at distinct angles (using the proposed 3D geometry) would improve detection performance. The detection ratio (DR) is thus employed as a metric to evaluate the system's effectiveness in detecting the robot.

The detection duration (Δt_{det}) represents the time during which the robot is successfully detected by the proposed system. The total duration (Δt_{total}) is the time span between the first and last received pose messages reported by the system, regardless of whether the robot was detected or not. The detection ratio is then the detection duration divided by the total duration. Therefore, a DR of 0 means the system never detects the robot during the time period in question, while a DR of 1 means the system always detects the robot.

To measure the period when the robot was successfully detected (Δt_{det}), we impose a message timeout of 0.5 seconds. When the robot is undetected, the time duration from the last received pose message to the next received message is deducted from the total duration (Δt_{total}). This method provides a quantitative way to evaluate how well the system can detect the robot over time, and can be represented mathematically as in (8).

$$DR = \Delta t_{\text{det}} / \Delta t_{\text{total}} \tag{8}$$

IV. RESULTS

A. Qualitative analysis of position tracking

The plots in Fig. 5 compare the robot's trajectory estimated using both the single-marker and the proposed multi-marker approaches at various camera heights, with the ground truth trajectory. Ground truth data is represented by a continuous black line, while the estimated positions from each approach are identified by colored markers: blue for the single approach, and orange for the multiple.

Investigating the marker density at varying camera heights, the multi-marker approach shows higher density at the 80

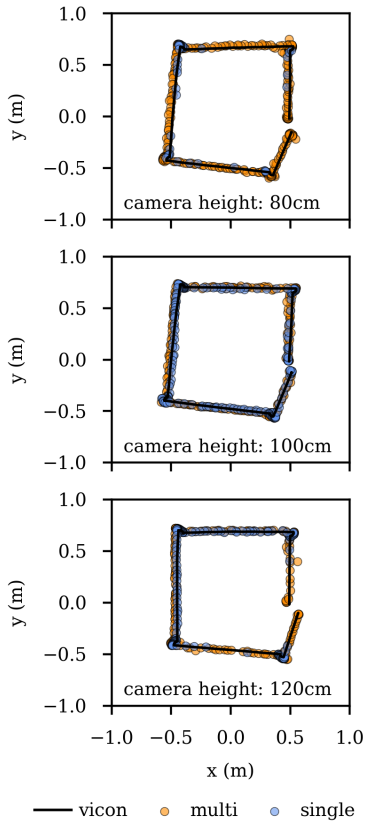

Fig. 5. Trajectory of the moving robot in the xy plane as detected by the single-marker (blue) and multi-marker (orange) approaches, and ground truth (black).

and 120 centimeters settings, whereas both methods exhibit similar densities at a height of 100 centimeters. In addition, both methods result in markers tightly clustered around the ground truth, with minor deviations observed along the robot's trajectory. However, when it comes to coverage, our method excels, effectively covering the entire robot's path at all camera heights, while the single method fails to do so in the 80 and 120 centimeters settings. Overall, both methods generate estimates that remain close to the ground truth, while our method offers greater marker density and coverage.

B. Investigation of fusion methods

The bar plot shown in Fig. 6 displays the tracking error between different fusion methods and the ground truth for various camera heights. Standard deviations are also shown with error bars. The y-axis represents RMSE values, and the x-axis denotes the assessed camera heights both in centimeters. Additionally, these results originated from a system configuration with three cameras, calibration, and multiple markers. Assessing the values, it can be observed that the methods led to similar results. At a height of 80 cm, errors of 3.96±0.20 cm, 4.01±0.27 cm and 3.97±0.24 cm were observed for the mean, Gaussian and weighted fusion methods. When the cameras were elevated to 100 cm, the errors were 3.67±0.10 cm, 3.71±0.12 cm and

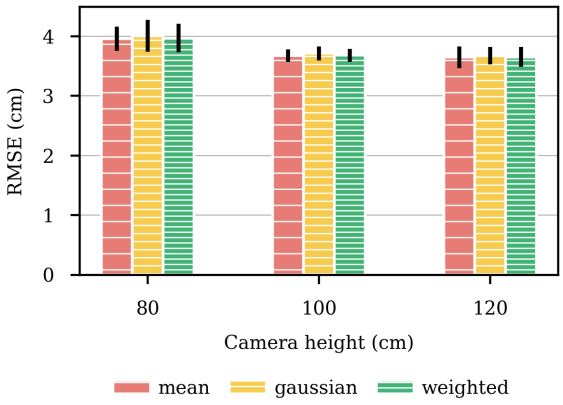

Fig. 6. RMSE of position estimates obtained from various pose fusion algorithms across different camera heights.

3.68±0.11 cm. Finally, at 120 cm, the methods yielded 3.65±0.18 cm, 3.68±0.15 cm and 3.65±0.16 cm. Overall, the mean approach offers the lowest RMSE and variance across heights, except for the 120 cm case, where it shows similar RMSE, but a higher variance than the Weighted approach.

C. Tracking the position error

Fig. 7 provides a bar plot comparing the RMSE values and standard deviations (error bars) of several system configurations at varying camera heights. All values were obtained applying the mean fusion method. The multi-marker, three cameras with calibration configuration resulted in errors of 4.0, 3.7 and 3.6 centimeters when the cameras were elevated to heights of 80, 100 and 120 centimeters, respectively. When we increased the number of cameras to five, the RMSE also increased, reaching mean values of 5.8, 5.3 and 6.1 centimeters. When keeping the same number of

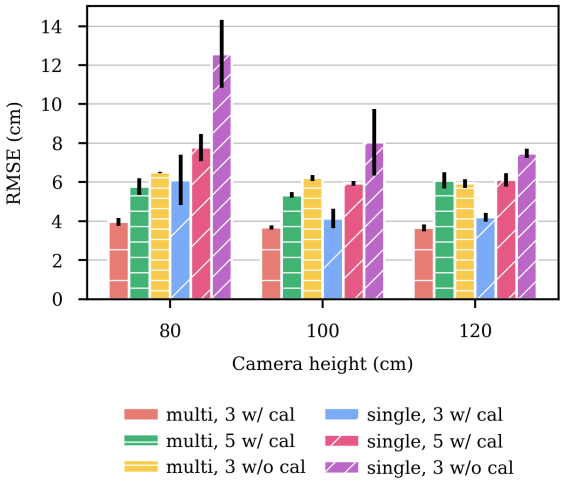

Fig. 7. RMSE of position estimates obtained from different system setups across different camera heights. The setups include multi and single-marker configurations, with either three or five cameras, and with or without calibration.

cameras (three), but removing the calibration, the RMSE also increased to 6.5, 6.2 and 5.9 centimeters. As for the single-marker approach, the errors for the setups were 6.1, 4.1 and 4.2 centimeters (three cameras, with calibration), 7.8, 5.9 and 6.1 (five cameras, with calibration), and 12.6, 8.0 and 7.5 (three cameras, no calibration).

D. Detection ratio

In Fig. 8, a comparison of detection ability of the system with the investigated setups are displayed using a bar plot with error bars. The y-axis provides the DR values, while the x-axis informs the various camera heights. In the multi-marker configuration with three cameras and without calibration, DRs of 61.6%, 55.4%, and 53.6% were observed when the cameras were placed at heights of 80, 100, and 120 centimeters, respectively. When calibration was applied, the ratios increased to 95.3%, 86.5%, and 78.5% for the same camera heights. Increasing the number of cameras to five further improved detection, reaching 99.1%, 100%, and 98.8%. Employing the single-marker approach, the errors for the same configurations were 1.5%, 11.5%, and 15.6% (three cameras without calibration), 16%, 50.6%, and 54.5% (three cameras with calibration), and 35.9%, 75.9%, and 79.8% (five cameras with calibration).

V. DISCUSSIONS

A. Qualitative analysis of position tracking

The plots demonstrate that both the single and multi-marker methods used for pose estimation yield results that are close to the ground truth, indicating that these approaches are viable for localization with a low-cost system. However, the greater number of markers of the latter method enhances the system's ability to estimate poses despite the cameras' position, allowing greater flexibility in camera placement. Consequently, when using the cube, the system can more effectively cover the object's path, regardless of the camera's

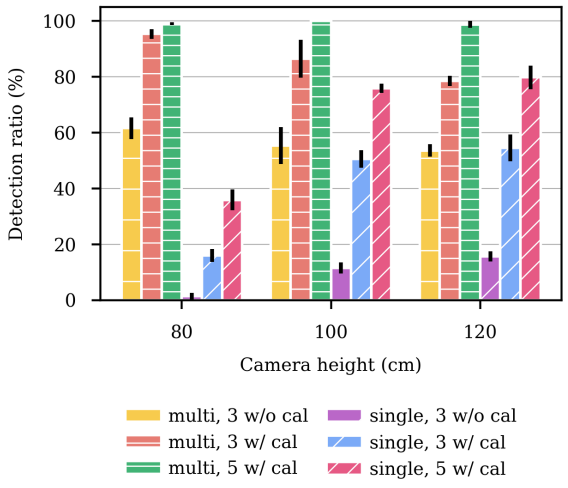

Fig. 8. Detection ratio from different system setups across different camera heights. The setups include multi and single-marker configurations, with either three or five cameras, and with or without calibration.

positioning. Furthermore, the results indicate that the multi-marker approach can offer more detailed information about the robot's position as indicated by the higher density of samples in the plots, once more despite camera position.

B. Investigation of fusion methods

Our analysis of RMSE values yielded from different fusion methods revealed that, in general, the mean approach achieved the lowest RMSE and variance. The simplicity of the algorithm may have provided a slight edge in publishing the pose estimate, leading to a shorter delay between the ground truth and the other methods, and consequently to lower errors. The slightly higher RMSEs from area-based methods (Gaussian and Weighted) suggest that incorporating area information as proposed by [14] does not enhance the system's ability to track the robot. In contrast, we demonstrated that area information can be a proxy standard deviation for Gaussian fusion, but it does not necessarily improves tracking accuracy. Hence, a simple arithmetic mean can still be employed for pose fusion and effectiveness preserved, and pixel area can be a proxy for Gaussian fusion. Nonetheless, future research could explore novel methods that incorporate area information for enhanced tracking performance.

C. Tracking the position error

From the plot in Fig. 7, we observed that removing calibration information significantly increases RMSE, while using more markers reduces it. These observations demonstrate that our approach can significantly improve tracking accuracy compared to the previous method. As noted in [11], the best result achieved was an RMSE of 13.45 centimeters using ten cameras. In contrast, our approach with a multi-marker and calibrated configuration achieves an RMSE of 3.6 centimeters, a 73% improvement, with only three cameras and no sensor information. Additionally, as observed in [11], increasing the number of cameras can lead to an unexpected increase in error. Upon investigation, we found that adding more cameras caused interference in the network, which in turn caused delays in image reception and pose estimation. Then, by eliminating two cameras, we obtained lower RMSE results, highlighting the importance of network optimization in such systems.

D. Detection ratio and processing time

As hinted by the qualitative analysis, the multi-marker approach is highly effective in detecting the robot under different conditions. This effectiveness can be attributed to the camera's ability to identify at least one of the five markers to generate a pose estimate. The single-marker approach, however, requires the camera to be accurately positioned relatively to the marker for its detection. Any robot that partially occludes the single marker results in loss of tracking. Rather than searching for an optimal camera position, users can use the cube and place the cameras wherever it is most convenient. Moreover, despite increasing RMSE, when the number of cameras was increased to five and calibration was used, DR was 100%. These findings underscore the

importance of a careful balance between system complexity, such as the number of cameras and the overall detection performance. Moreover, the processing time of the system varies, depending mainly on OpenCV parameters and the size of images, but overall it was negligible compared to image latency.

VI. CONCLUSIONS

This study aimed at integrating a multi-marker structure into a low-cost multi-camera localization system to improve pose estimation accuracy and reliability while further reducing costs, and the analysis of different pose fusion techniques. Our multi-marker, multi-camera system produces estimates that are 73% more accurate than MoCArU, reducing the RMSE from 13.45 to 3.6 centimeters while leveraging fewer cameras, no IMU or odometry. Additionally, when compared to the single-marker approach, our system can consistently provide DRs of almost 100% (not previously obtainable) while offering flexible camera placement. We also confirm that the fusion method does not significantly impact tracking accuracy and a simple arithmetic mean can be used. Future work will focus on improving the system's network potentially through integration of antennas and the use of libraries like Zenoh [29]. Overall, we have developed a cost-effective, multi-camera, multi-marker localization system that significantly outperforms previous iterations in terms of accuracy, flexibility, and detection reliability, offering an affordable solution for tracking robots.

ACKNOWLEDGMENT

This work was supported by the Sony Semiconductor Solutions Corporation.

REFERENCES

[1] J. Zheng, S. Bi, B. Cao, and D. Yang, "Visual localization of inspection robot using extended kalman filter and aruco markers," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 742–747, 2018.

[2] A. Sampathkrishna, "Aruco maker based localization and node graph approach to mapping," 2022.

[3] Y. Zhang, G. Tian, and X. Shao, "Safe and efficient robot manipulation: Task-oriented environment modeling and object pose estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[4] B. Xing, Q. Zhu, F. Pan, and X. Feng, "Marker-based multi-sensor fusion indoor localization system for micro air vehicles," *Sensors*, vol. 18, no. 6, 2018.

[5] D. Avola, L. Cinque, G. L. Foresti, C. Mercuri, and D. Pannone, "A practical framework for the development of augmented reality applications by using aruco markers," in *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, pp. 645–654, INSTICC, SciTePress, 2016.

[6] R. G. Lupu, P. Herghelegiu, N. Botezatu, A. Moldoveanu, O. Ferche, C. Ilie, and A.-M. Levinta, "Virtual reality system for stroke recovery for upper limbs using aruco markers," in *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 548–552, 2017.

[7] M. Menolotto, D.-S. Komaris, S. Tedesco, B. O'Flynn, and M. Walsh, "Motion capture technology in industrial applications: A systematic review," *Sensors*, vol. 20, no. 19, 2020.

[8] S. Al Habsi, M. Shehada, M. Abdoon, A. Mashood, and H. Noura, "Integration of a vicon camera system for indoor flight of a parrot ar drone," in *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, pp. 1–6, 2015.

[9] N. J. Sie, S. Srigrarom, and S. Huang, "Field test validations of vision-based multi-camera multi-drone tracking and 3d localizing with concurrent camera pose estimation," in *2021 6th International Conference on Control and Robotics Engineering (ICCRE)*, pp. 139–144, 2021.

[10] D. Chen, Z. Peng, and X. Ling, "A low-cost localization system based on artificial landmarks with two degree of freedom platform camera," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 625–630, 2014.

[11] R. Assabumrungrat, R. Bezerra, I. Barros, S. Kojima, Y. Okada, M. Konyo, K. Ohno, and S. Tadokoro, "Mocaru: Low-cost wireless portable robot localization system using iot," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3458–3465, 2023.

[12] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[13] L. V. Vargas, A. C. Leite, and R. R. Costa, "On the discrete implementation of the filtered inverse method for serial robots," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1030–1037, 2023. 22nd IFAC World Congress.

[14] S. S. Tørdal and G. Hovland, "Relative Vessel Motion Tracking using Sensor Fusion, Aruco Markers, and MRU Sensors," *Modeling, Identification and Control*, vol. 38, no. 2, pp. 79–93, 2017.

[15] "ROBOTIS e-Manual." `https://emanual.robotis.com`. Accessed: 27-08-2024.

[16] "Spresense Extension Board — Developer World." `https://developer.sony.com/spresense/products/spresense-ext-board/`. Accessed 26-08-2024.

[17] "Spresense Main Board." `https://developer.sony.com/spresense/products/spresense-main-board/`. Accessed: 2024-8-27.

[18] "Spresense HDR Camera Board — Developer World." `https://developer.sony.com/spresense/products/spresense-hdr-camera-board/`, 2024. [Accessed 26-08-2024].

[19] P. García-Ruiz, R. Muñoz-Salinas, R. Medina-Carnicer, and M. J. Marín-Jiménez, "Object localization with multiplanar fiducial markers: Accurate pose estimation," in *Pattern Recognition and Image Analysis* (A. Pertusa, A. J. Gallego, J. A. Sánchez, and I. Domingues, eds.), (Cham), pp. 454–465, Springer Nature Switzerland, 2023.

[20] P. Oščádal, D. Heczko, A. Vysocký, J. Mlotek, P. Novák, I. Virgala, M. Sukop, and Z. Bobovský, "Improved pose estimation of aruco tags using a novel 3d placement strategy," *Sensors*, vol. 20, no. 17, 2020.

[21] S. Roos-Hoefgeest, I. A. Garcia, and R. C. Gonzalez, "Mobile robot localization in industrial environments using a ring of cameras and aruco markers," in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1–6, 2021.

[22] S. S. Katta, J. Adnan, S. Chaudhary, S. D. Roy, C. Arora, S. K. Saha, and M. E, "Pose estimation of 5-dof manipulator using on-body markers," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pp. 897–902, 2021.

[23] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[24] S. Faria, J. Lima, and P. Costa, "Sensor fusion for mobile robot localization using extended kalman filter, uwb tof and aruco markers," in *Optimization, Learning Algorithms and Applications* (A. I. Pereira, F. P. Fernandes, J. P. Coelho, J. P. Teixeira, M. F. Pacheco, P. Alves, and R. P. Lopes, eds.), (Cham), pp. 235–250, Springer International Publishing, 2021.

[25] R. Wang, H. Guo, X. Wang, and L. Han, "The effect of aruco marker size, number, and distribution on the localization performance of fixed-point targets," in *2023 6th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pp. 118–123, 2023.

[26] B. Braden, "The surveyor's area formula," *The College Mathematics Journal*, vol. 17, no. 4, pp. 326–337, 1986.

[27] J. O. Smith, *Spectral Audio Signal Processing*. W3K Publishing, 2023. online book, 2011 edition.

[28] "Tracker: Delivering precise real-world data." `https://www.vicon.com/software/tracker/`. Accessed 27-08-2024.

[29] W.-Y. Liang, Y. Yuan, and H.-J. Lin, "A performance study on the throughput and latency of zenoh, mqtt, kafka, and dds," 2023.