IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Robot Arm Self-Calibration using RGB-D camera

Jiyong Lee[1], KangGeon Kim[1*], and Bum-Jae You[1*]

***Abstract*—Kinematic and hand-eye calibration of robotic arms is a critical research area in robotics, essential to ensuring the accuracy of manipulation tasks. The widely adopted methods for robotic arm calibration typically rely on specialized markers or external sensors to achieve precise measurements. However, these approaches are often expensive and require additional effort, such as the installation and maintenance of auxiliary equipment. Furthermore, many downstream tasks require separate hand-eye calibration steps because of differences between the sensors used for calibration and those used for task execution. Comprehensive calibration of both the robot arm and sensors plays a vital role in optimizing system performance. However, the robot's posture could be constrained due to either the sensor's limited range or textureless scenes when a camera is used. To address these limitations, this study proposes a cost-effective self-calibration method that simultaneously calibrates the robot arm and determines the spatial relationship between the robot and an RGB-D camera, allowing for data collection at multiple locations. The proposed approach leverages recent advancements in machine learning to identify correspondences between images captured at different robot postures, facilitating automatic data selection. Furthermore, the removal of location constraints increases flexibility, enabling the collection of sufficient data as the robot's location changes. The method is evaluated using a Franka Emika Panda robotic arm, and the experimental results demonstrate its effectiveness in achieving accurate calibration without the need for external devices or markers.**

***Index Terms*—Calibration and Identification; Computer Vision for Automation**

I. Introduction

RECENTLY, manipulation tasks have garnered significant interest within the robotics community [1] and in 2019, the OECD reported that many jobs could disappear due to automation [2]. Robots serve as powerful tools for automating repetitive and labor-intensive tasks. In particular, multiple-joint robot manipulators have been increasingly utilized in various industrial applications, including factories and construction sites. However, numerous factors contribute to mechanical errors, such as issues during assembly, transportation, or collisions, which result in deviations in the robot's motion. Furthermore, calibration is not a once-in-a-lifetime task; robots may require recalibration due to mechanical wear or extended periods of use.

Traditional calibration methods are mainly based on external devices [3], [4] or specialized patterns [5], [6]. Although these methods effectively correct for errors in the kinematic chain, they require additional time and space for equipment installation. For already deployed systems, on-site recalibration is often impractical or impossible.

Some studies have attempted to calibrate robot manipulators using an RGB-D camera [7], [8] as a substitute for external devices such as laser trackers. Since RGB-D cameras capture 3D points through depth information, they can replace the expensive external equipment with a low-cost alternative. However, this may degrade the calibration results as the depth accuracy of RGB-D cameras varies depending on the measurement method and is generally less precise than that of laser trackers.

For accurate manipulation of target objects in contact-rich tasks, a robot must respond accurately to environmental information provided by sensors such as cameras mounted on the robot. Traditional kinematic calibration methods using external sensors, such as laser trackers, do not account for the relationship between the robot and the camera. Similarly, eye-in-hand calibration often overlooks potential errors in the robot arm itself. Furthermore, it would be ideal to utilize the same sensor for both calibration and task execution since the operational range of a robot manipulator varies depending on the sensor type. For instance, sensors like short-range finders have a limited sensing range, which can restrict the manipulator's range of motion. In such cases, relocating the robot would enable various postures, even within a constrained range of motion.

This work presents a novel self-calibration method for jointly calibrating a robot arm and its hand-eye system. Image feature data for calibration is collected at multiple locations by relocating the robot base, as it may be difficult to obtain sufficient data in texture-less environments due to a lack of keypoints [10], even when using the image-based correspondence algorithm, SuperGlue [9], for feature point extraction. Additionally, the limited sensing range can be compensated for by moving the robot closer to nearby objects, thereby bringing them within the sensor's effective range.

Although sufficient data may be collected, the data inherently include sensor-induced noise. To enhance robustness, a local-to-global pose estimation scheme is employed for efficient outlier rejection across the collected data. This is a parallel pose estimation method adaptively performed in a coarse-to-fine manner. This is originally proposed as a replacement for RANdom SAmple Consensus (RANSAC) in the machine learning community [10]. This concept is suitable for efficient outlier rejection, which is performed during the

Manuscript received: March 21, 2025; Revised June 27, 2025; Accepted July 31, 2025

This paper was recommended for publication by Editor L. Pallottino upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Research Foundation of Korea through the Ministry of Science and ICT (MSIT), Korean Government under Grant 2022M3C1A3098746. This work was also supported by the Korea Institute of Science and Technology Institutional Programs (Project No. 2E33602)

[1]Jiyong Lee, KangGeon Kim, and Bum-Jae You are with Artificial Intelligence and Robotics Institute, Korea Institute of Science and Technology, Seoul, South Korea jiyonglee@kist.re.kr, danny@kist.re.kr, ybj@kist.re.kr

*Corresponding Authors

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

local estimation to make the global estimation more accurate.

Lastly, it is shown in our study that the Structural Similarity Index Measure (SSIM) is an effective criterion for outlier rejection. The proposed method improves calibration accuracy by 7% compared to separate calibration using a laser tracker, as measured by Root Mean Square Error (RMSE).

The contributions of this work can be summarized as follows:

- A novel self-calibration methodology is proposed for both robot arm and hand-eye calibration in an eye-in-hand setting. It automatically collects data from multiple locations using an image-based correspondence matching algorithm [9], and performs optimization by merging the data into a unified coordinate system.
- To address the potential data scarcity of the image-based algorithm, data are collected at multiple locations by relocating the robot base, and we demonstrate how calibration can be performed using the data.
- To enable efficient outlier rejection, a local-to-global pose estimation method is employed. Outliers are first rejected during the local pose estimation step using a distance metric, after which the global pose is estimated.
- We demonstrate that SSIM is effective in environments with severe depth noise.

II. Related Work

A. Correspondence Matching

This has been a long-standing research topic in computer vision, with various applications such as Structure From Motion (SFM) [11], [12], visual localization [13], and optical flow [14]. Traditionally, feature descriptors have been extracted using hand-crafted methods such as SIFT [15] and SURF [16], and correspondences are found using additional matching techniques such as Markov Chain Monte Carlo (MCMC) [17] or graph matching [18], leveraging the distinctiveness of the descriptors. Recently, advancements in machine learning have led to the development of more reliable algorithms. SuperGlue [9] addresses this problem using a two-step approach. Similar to traditional methods, feature descriptors are first extracted and then matched. The key difference is that both the feature descriptors and the matching process are based on learning algorithms. Another line of research has attempted to integrate the two steps into a single process. LoFTR [19] adopts a coarse-to-fine scheme to perform local feature extraction and matching within a unified network model. COTR [20] addresses both sparse and dense matching problems and demonstrates its performance through various matching demonstrations, including single image pairs, facial landmarks, and two-view reconstruction.

B. Kinematic & Hand-Eye Calibration

Kinematic calibration and hand-eye calibration are often coupled. However, many studies assume that the robot is already calibrated and focus on estimating the hand-eye transformation. These approaches typically use specialized target objects, such as dots [5] or chessboard patterns [6]. Other works on full-system calibration treat the hand-eye transformation as an additional robot segment, making the overall calibration procedure similar to standard hand-eye calibration.

Recently, some studies have attempted calibration without using patterns, instead relying on features such as straight edges [21] or CAD models [22]. In particular, studies like [7], [8] have addressed the problem of calibrating a robot manipulator while it simultaneously performs another downstream task. In [7], SIFT [15] feature points are extracted, and their correspondences are determined using epipolar constraints. The effectiveness of the proposed method is demonstrated on both synthetic and real-world datasets. In [8], only range data is used, excluding color information. Relative poses are estimated using the Iterative Closest Point (ICP) algorithm, and calibration is performed using a small object. While these methods achieve accurate calibration, they require the sensor to be specifically aimed at the target object, and the robot must remain fixed in a specific location throughout the process. Moreover, when using a small object, the range of possible robot postures is limited.

III. Methodology

A robot arm is designed with an ideal configuration based on its design specifications. However, factors such as assembly processes or transportation may introduce mechanical distortions that deviate from the original design. This work is intended to perform calibration without the use of external devices. Moreover, to address data scarcity, it enables the relocation of the robot base.

To achieve this, two key challenges are addressed: (1) obtaining accurate measurements from the environment, and (2) merging data from multiple locations into a unified coordinate system to utilize this information for robot arm and hand-eye calibration. To ensure accurate measurements, three criteria are applied. During outlier rejection, discrepancies are evaluated in both 2D and 3D distances to enhance robustness. In the image pair selection step, final image pairs are determined using an additional metric (SSIM). For data merging and utilization, two types of extended links are introduced: one to connect different locations, and another for hand-eye calibration. By introducing these two types of extended links, all data can be transformed into a unified coordinate system, enabling the calibration process.

As illustrated in Fig. 1, the overall procedure consists of four steps: (1) data acquisition at multiple locations, (2) 3D point cloud generation, (3) pairwise pose estimation, and (4) optimization. The pairwise pose estimation step comprises four sub-modules: local pose estimation, outlier rejection, global pose estimation, and image pair selection.

A. Problem Statement

The goal of calibration is to determine the optimal parameters of the kinematic chain using images captured from multiple locations. Let $\mathcal{D} = \{I, J, L\}$ represent the collected data, where I denotes images acquired from various robot arm postures, J the corresponding joint angles, and L the indices of the robot's locations. The parameters to be optimized include the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

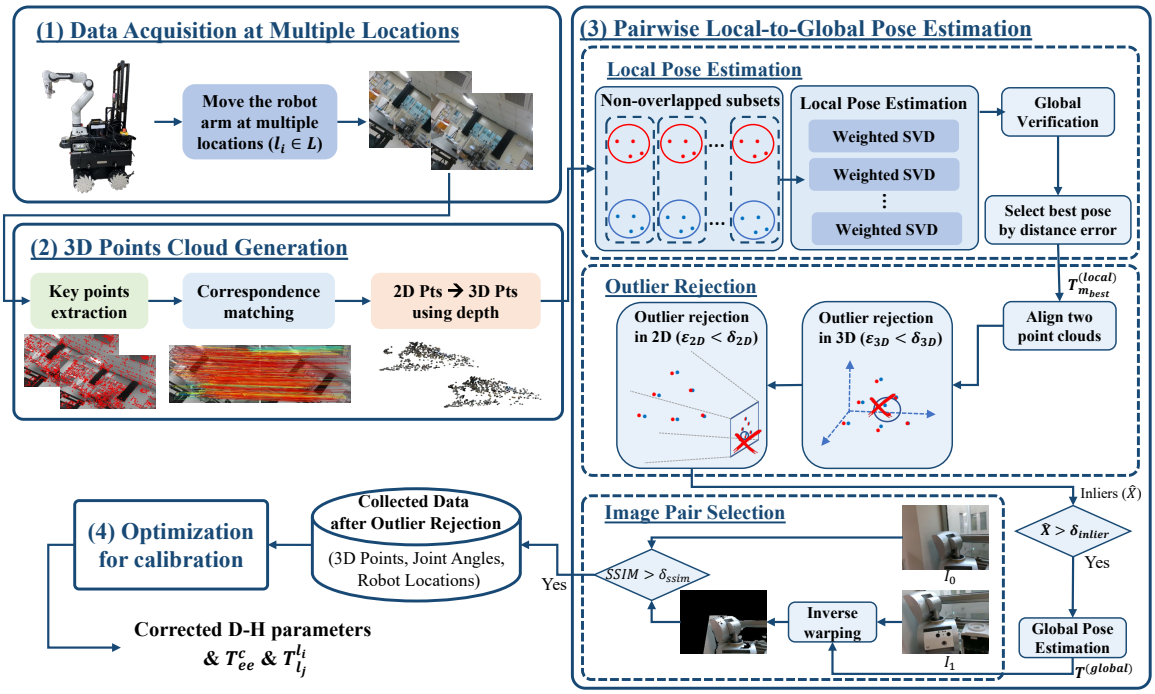


Fig. 1. **Overall Diagram** The calibration procedures are composed of four steps. (1) Data are collected at multiple locations, then (2) 2D keypoints are extracted using SuperGlue [9] and converted to 3D points using depth images. (3) The outlier rejection and pose estimation are performed in the pairwise pose estimation step. Finally, (4) the calibration is conducted with the collected data.

corrected Denavit–Hartenberg (DH) parameters, the eye-to-hand transformation T_{ee}^{c}, and the inter-location transformations $T_{l_j}^{l_i}$ $(l_i, l_j \in L)$.

During the data acquisition step, the wheel-based mobile robot moves within an area where feature matching is feasible. The joint angles are selected randomly, and the captured images may or may not overlap. Therefore, selecting appropriate image pairs is crucial for extracting useful information to be used in the calibration process.

Additionally, the robot is allowed to move during data acquisition to address data deficiency caused by textureless environments or limited sensor range. More data can be gathered by moving closer to textured objects. Even though the camera remains focused on a single object, the entire workspace can be covered by relocating the mobile base. In such cases, the transformation between data collected at different locations must be computed.

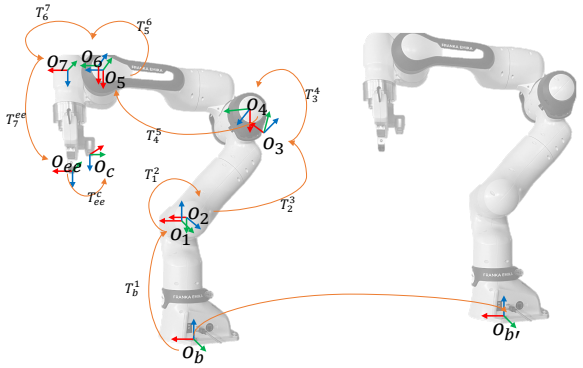


Fig. 2. **Kinematic Chain** The entire kinematic chain includes the links of the robot arm, along with two additional links: T_{ee}^{C} and $T_{b}^{b'}$. T_{ee}^{C} represents the transformation between the camera and the end-effector for hand-eye calibration, while $T_{b}^{b'}$ denotes the transformation between O_b and $O_{b'}$ for robot base relocation.

B. Kinematic Modeling

The robot arm used in this work is the Franka Panda, with its configuration represented using DH parameters. The kinematic chain is modeled with $9+(|L|-1)$ links: 7 for the arm itself, 1 for the gripper, 1 for hand-eye calibration, and $(|L|-1)$ for the transformation between locations. The kinematic chain is illustrated in Fig. 2. The complete transformation of this system includes the robot arm, defined from O_b (the base frame) to O_{ee} (the end-effector frame), along with an additional robot base $O_{b'}$. The tip of the gripper is designated as the end-effector (ee), making the target transformation for hand-eye calibration T_{ee}^{c}.

The transformation T_7^{ee} does not include any calibration parameters, as it is directly computed from the robot's specifications. Additionally, two components are excluded from calibration due to redundancy: the first link and the last parameter of T_6^7. The origin of the base frame (O_b) is undefined with respect to the world coordinate system, preventing calibration of the first link's parameters. Similarly, the last parameter of T_6^7 is offset by T_{ee}^{c}. Therefore, the first link (4 parameters) and the last parameter (1 parameter) of T_6^7 are fixed and excluded from the calibration process.

To eliminate constraints on the robot's posture, data collection is performed as the robot moves to different locations. To enable this, additional parameters must be introduced to

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

estimate the transformation between the consecutive locations, resulting in extra $6 \times (|L| - 1)$ parameters.

As a result, the total number of parameters to be calibrated is calculated as $4 \times (\Omega - 1) - 1$ (for the arm links) + 6 (for T_{ee}^{c}) + $6 \times (|L| - 1)$ (for the robot's locations). Here, Ω is the number of the manipulator links, with each link contributing four parameters. The value 6 represents the degrees of freedom in a transformation: three for rotation and three for translation, and $|L|$ is the number of locations.

C. Data Acquisition at Multiple Locations

As the first step, images and joint angles are collected for each posture. Many existing calibration approaches assume that the robot remains fixed at a specific location, while others require the robot arm to follow a predefined pattern or trajectory [8]. Additionally, the most common and straightforward calibration method using a camera involves a chessboard pattern. However, if the vertical orientation of the images is reversed, the resulting data may become inconsistent, potentially leading to inaccuracies.

This work imposes no such constraints. The robot stops at one location, collects images using randomly generated joint angles, and then moves to the next location. This process is repeated until a sufficient amount of image feature data is collected. Each raw data point consists of an RGB-D image, joint angles, and a location index.

D. 3D Point Cloud Generation

The calibration process begins with the extraction of 2D keypoints from the raw images, followed by the conversion to a 3D point cloud.

Given two input images, 2D keypoints, called SuperPoints [23], are first extracted from each image, and correspondences are then established between them. Given known 2D-2D correspondences between two images, the pose is typically estimated using the 3D-2D Perspective-n-Point (PnP) RANSAC algorithm by minimizing the reprojection error [24]. While this method is robust to noise, it often yields less accurate pose estimates. Therefore, in this work, the pose is estimated using two 3D point clouds, which are directly constructed from depth images. Although this approach is more sensitive to measurement noise, its impact is mitigated through an outlier rejection process applied during pose estimation.

E. Pairwise Local-to-Global Pose Estimation

Poses have to be estimated while efficiently rejecting outliers from point clouds to ensure precise measurements, as the point clouds contain erroneous data due to sensor noise. In GeoTransformer [10], a local-to-global pose estimation scheme was originally proposed to integrate a pose estimation block into a machine learning algorithm in parallel, serving as an alternative to the inherently sequential RANSAC method. Their experiments demonstrate that this scheme is robust even when the data contains a high proportion of outliers. Inspired by this approach, we propose a pairwise local-to-global pose estimation method that enables both efficient outlier rejection and accurate pose estimation. This coarse-to-fine strategy begins by estimating local poses based on nearby data, then refines the results by rejecting outliers, thereby improving the global pose estimation. The pseudocode for the pairwise local-to-global pose estimation is provided in Algorithm 1.

Algorithm 1 Pairwise Local-to-Global Pose Estimation

Inputs: two point clouds $X^{(0)}, X^{(1)}$ having K points with known correspondence matching.

POSE ESTIMATION$(X^{(0)}, X^{(1)})$
  shuffle points indexes
  divide points into M groups having the n points
    $\chi_m \leftarrow \{x_j^i \in X^{(i)} | i \in \{0,1\},$
      $j \in \{0, ..., K-1\}\}, m \in \{0, ..., \lfloor K/n \rfloor - 1\}$
  At each group in parallel (**Local pose**)
    $T_m^{(local)} \leftarrow$ local pose from χ_m
    $\mathcal{E}_m \leftarrow distance(X^{(0)}, T_m^{(local)} \cdot X^{(1)})$
  $m_{best} \leftarrow \text{argmin}_m \, \mathcal{E}_m$
  $\hat{X}^{(0)}, \hat{X}^{(1)} \leftarrow$ **OUTLIER_REJECTION**$(X^{(0)}, X^{(1)}, T_{m_{best}}^{(local)})$
  $T^{(global)} \leftarrow$ global pose from $\hat{X}^{(0)}, \hat{X}^{(1)}$ (**Global pose**)
  return $T^{(global)}$

OUTLIER_REJECTION$(X^{(0)}, X^{(1)}, T)$
  $\mathcal{E}_{3D} \leftarrow distance(X^{(0)}, T \cdot X^{(1)})$
  $\mathcal{E}_{2D} \leftarrow distance(proj(X^{(0)}), proj(T \cdot X^{(1)}))$
  $\mathbb{IN} \leftarrow \{i | \mathcal{E}_{3D,i} < \delta_{3D}, \mathcal{E}_{2D,i} < \delta_{2D}, i \in \{0, ..., K-1\}\}$
  $\hat{X}^{(0)} \leftarrow \{x_j^{(0)} | j \in \mathbb{IN}\}$
  $\hat{X}^{(1)} \leftarrow \{x_j^{(1)} | j \in \mathbb{IN}\}$
  return $\hat{X}^{(0)}, \hat{X}^{(1)}$

1) Local-to-Global Pose Estimation: This process is illustrated in the pairwise pose estimation section of Fig. 1. Local pose estimation, the coarse step, is performed using a subset of keypoints, while global pose estimation, the fine step, is conducted using only the inlier points.

In the local pose estimation, given K points obtained from RGB-D images, $m(= \lfloor K/n \rfloor)$ relative poses are calculated in parallel using n points each (with a total of $n \times m$ points). These poses are then globally verified using all the points. This parallel computation replaces the sequential repetition in RANSAC. Unlike RANSAC, which sequentially performs the 'estimation-verification' routine until a condition is satisfied, this method uses non-overlapping point subsets to estimate poses in parallel. Furthermore, instead of counting the number of inlier points within a threshold, the reprojection error between corresponding points is calculated because it is unclear which one to choose in cases where multiple local poses have the same number of inlier points. To resolve this, after aligning the two sets of points, the local pose with the minimum distance error between corresponding points is selected as the best local estimate. In the global pose estimation, the final pose ($T^{(global)}$) is calculated using only inlier points ($\mathbb{IN}$).

This method helps achieve an accurate pose without requiring repeated trials. In both local and global pose estimation, the pose is computed using weighted Singular Value Decomposition.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

2) Outlier Rejection: Outliers are rejected based on two criteria: (1) 2D pixel error ($\mathcal{E}_{2D}$) and (2) 3D distance error ($\mathcal{E}_{3D}$). Once the best local pose ($T^{(local)}_{m_{best}}$) is obtained, one set of points is transformed to align with the other. The displacement between corresponding points varies depending on the dimension in which it is measured. Some points that exhibit a small reprojection error in 2D may show a large distance error in 3D. Therefore, errors are measured in both dimensions for effective outlier rejection.

3) Image Pairs Selection: To accept a pair of images as stable data, image pairs are selected based on three criteria: (1) the number of keypoints, (2) the number of inliers, and (3) the Structural Similarity Index Measure (SSIM) [25].

First, keypoints are extracted from each image as a measure of useful information. Images with fewer than $\delta_{keypoint}$ keypoints are excluded, as they are considered to lack sufficient information. Second, during pose estimation, inliers are determined based on the best local pose. Image pairs with fewer than δ_{inlier} inliers are discarded, as they are unlikely to produce a stable pose.

Lastly, because the depth values captured by an RGB-D camera tend to become less accurate with increasing distance, SSIM is used to evaluate the confidence of the estimated pose (see Eqn (1)).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{1}$$

SSIM is a metric that measures the similarity between two images. μ_x, μ_y and σ_x, σ_y represent the mean and standard deviation of the two images, and σ_{xy} denotes the covariance between them. c_1, c_2 are constants used to ensure numerical stability. To apply SSIM, an inverse warping is performed on one image using the pose matrix, and the SSIM score is calculated between the warped image and the other image. The warped image may contain blank pixels due to differences in viewpoint; these regions are excluded from the calculation. Only pixels with non-zero values in both images are considered. If the SSIM score falls below δ_{SSIM}, the image pair is excluded, as it probably contains significant depth noise.

F. Optimization for Calibration

As the last step, optimization is performed using the collected data to estimate the offsets of the DH parameters, the matrix T^c_{ee}, and the transformation matrices between locations. The cost function used in the optimization consists of two components: the local cost function ($\mathcal{C}_l$) and the global cost function ($\mathcal{C}_g$). The local cost function accounts for data collected at the same location, while the global cost function incorporates data from different locations.

All points captured at the same location can be transformed into the robot base coordinates using Eqn (2).

$$\begin{aligned} T(\theta)^b_c &= (T^b_1 \cdot T^1_2 \cdot T^2_3 \cdot T^3_4 \cdot T^4_5 \cdot T^5_6 \cdot T^6_7 \cdot T^7_{ee} \cdot (T^c_{ee})^{-1})(\theta) \\ \theta &= \{\theta_{DH}, \theta_{c\to ee}\}, \end{aligned} \tag{2}$$

where θ_{DH} represents the DH parameters and $\theta_{c\to ee}$ denotes the 6-DoF transformation parameters (3 for rotation and 3 for translation) between the camera and the end-effector.

To find the optimal parameters, the cost function is defined as in Eqn (3). The function calculates the reprojection errors between all data pairs. It is more stable than the approach used in [4], which minimizes displacement error after transforming all measurements into a common coordinate frame.

$$\begin{aligned} &\mathcal{C}_l(\delta\theta \mid \theta, X^c) \\ &\quad = \sum_{i}^{N-1} \sum_{j=i}^{N} \|x^c_i - (T(\theta_i + \delta\theta)^b_c)^{-1} T(\theta_j + \delta\theta)^b_c x^c_j\|^2, \\ &\qquad\qquad x^c_i, x^c_j \in X^c \end{aligned} \tag{3}$$

To extend the robot's range of motion, data is captured at multiple locations. Consequently, an additional cost function, referred to as the global cost function, is introduced to estimate the transformations between different locations. Similar to the local cost function, the global cost function computes the reprojection error between data captured at different locations, as expressed in Eqn. (4).

$$\begin{aligned} &\mathcal{C}_g(\delta\theta, \theta_0, ..., \theta_{\Gamma-1} \mid \theta, X^c_{l_p}, X^c_{l_q}) \\ &\quad = \sum_{i\in l_p} \sum_{j\in l_q} \|x^c_i - (T(\theta_i + \delta\theta)^b_c)^{-1} T^{l_p}_{l_q} T(\theta_j + \delta\theta)^b_c x^c_j\|^2, \\ &\qquad T^{l_p}_{l_q} = T^{l_p}_{l_{p+1}}(\theta_p) \cdot ... \cdot T^{l_{q-1}}_{l_q}(\theta_{q-1}), \\ &\qquad\qquad (p < q), x^c_i \in X^c_{l_p}, x^c_j \in X^c_{l_q} \end{aligned} \tag{4}$$

where $T^{l_p}_{l_q}$ is the transformation matrix from the location l_p to the location l_q. The total cost function is defined as:

$$\mathcal{C} = \mathcal{C}_l + \mathcal{C}_g \tag{5}$$

The cost function is minimized using the Levenberg-Marquardt algorithm. The optimal solution ($\delta\theta$) in Eqn. 6 is used to determine the DH offsets ($\delta\theta_{DH}$), the transformation matrix from the camera to the end-effector (T^c_{ee}), and the parameters $\theta_0, ..., \theta_{\Gamma-1}$, which represent the 6-DoF transformations between different locations.

$$\delta\theta, \theta_0, ..., \theta_{\Gamma-1} = \underset{\delta\theta, \theta_0, ..., \theta_{\Gamma-1}}{\operatorname{argmin}} \mathcal{C}(\delta\theta, \theta_0, ..., \theta_{\Gamma-1} \mid \theta, X^c) \tag{6}$$

IV. EXPERIMENTS

The proposed method is validated through several experiments. First, its effectiveness is demonstrated in both large and small areas. The results resemble Simultaneous Localization And Mapping (SLAM) in a large area and Structure From Motion (SFM) in a small area. Next, in terms of calibration, the results are compared with those obtained using other sensors or patterns (e.g., laser trackers, chessboard patterns). Traditionally, laser trackers have been widely used for robot kinematic calibration due to their highly accurate measurements. Cameras with chessboard patterns have been commonly employed, as they provide a convenient way to establish clear correspondences. These methods are considered de facto standards, so the proposed method is evaluated against them.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

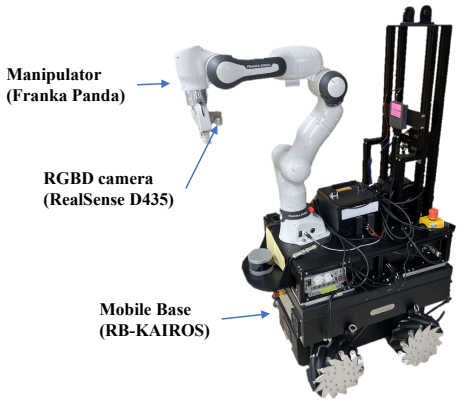


Fig. 3. **Experimental Setup** The manipulator (Franka Panda) is mounted on a mobile base (RB-KAIROS), with an RGB-D camera (RealSense D435) installed beneath the gripper.

A. Experimental Setup

This calibration method is tested on a Franka Emika Panda manipulator with a RealSense D435 camera mounted on the robot arm. The manipulator is fixed to a mobile base (RB-KAIROS), as shown in Fig. 3. Calibration is performed in both small and large areas. The small area consists of a round table (height: 0.72 m, radius: 0.35 m) with a book placed on top, while the large area is a corridor with a width of 2.27 m. According to the D435 datasheet, the depth noise level is less than 2% at a distance of 2 m. Due to varying depth noise levels at different distances, different evaluation criteria are applied for each case.

For calibration in a small area, the robot collects data at five locations, acquiring 10 samples at each location. The depth range is limited to 100 mm to 700 mm. The thresholds δ_{2D}, δ_{3D}, and δ_{SSIM} are set to 1 pixel, 1 mm, and 0.9, respectively. Within this range, depth data is less noisy, enabling accurate calibration verification.

In the large area, the robot moves forward through the corridor and collects 10 samples at each of five locations, as in the small-area case. Due to higher noise in the depth values at longer distances, the focus shifts from accuracy to evaluating the method's applicability. The usable depth range is set between 500 mm and 3000 mm, and δ_{2D}, δ_{3D}, and δ_{SSIM} are set to 1 pixel, 5 mm, and 0.7, respectively. $\delta_{keypoint}$ and δ_{inlier} are set to 30 and 10 in both scenarios. Optimization is performed using the Levenberg-Marquardt algorithm with finite differences.

B. Experimental Result

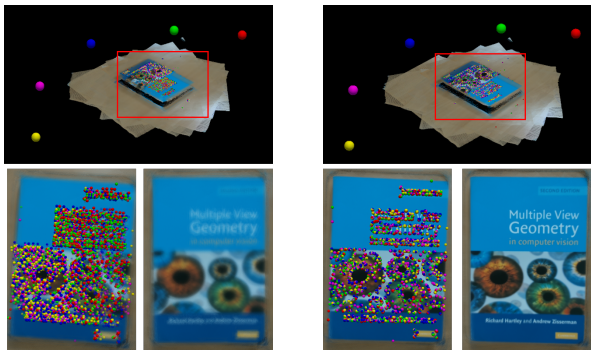
To quantitatively verify the results, 60 additional samples are captured within 1 meter of the camera at six different locations (10 samples per location). The corner points of the chessboard are detected using the *findChessboardCorners* function from the OpenCV library, and corresponding 3D points are recovered using the depth images. These points are reprojected across all pairs of images, and 2D pixel errors and 3D distance errors (in millimeters) are calculated. The results are presented in TABLE I. Although the errors in the large area are greater than those in the small area due to increased depth noise, they are still sufficient for performing large-scale downstream tasks.

TABLE I
QUANTITATIVE RESULT OF CALIBRATION. ($mean \pm std$)

method	error type	small area	large area
Proposed w/o SSIM	2D error (px)	1.65 ± 0.96	2.98 ± 1.82
	3D error (mm)	1.50 ± 0.70	3.15 ± 1.61
Proposed	2D error (px)	1.52 ± 0.90	2.91 ± 1.76
	3D error (mm)	1.43 ± 0.69	3.11 ± 1.61

As qualitative results, the point clouds extracted from the collected data are visualized. The calibration results for the small and large areas are shown in Fig. 4 and Fig. 5, respectively. In Fig. 4, the first row displays the aligned point clouds, where colored spheres represent the robot locations where data were collected. Fig. 4a shows the result using only hand-eye calibration, while Fig. 4b presents the result after full calibration, including both arm calibration and hand-eye calibration. In the enlarged views shown in the second row, small colored dots indicate the keypoints used for calibration, and the large colored dots represent the robot locations, using the same color as the corresponding collected data.

In Fig. 5, the first and second columns show the aligned point clouds from the side and top views, respectively. The third column provides enlarged views of the areas highlighted in red rectangles. Similar to the small area visualization, colored spheres indicate the data collection locations, and small colored dots in the third column represent the keypoints used for calibration. In both figures, the aligned point clouds obtained through full calibration appear clearer, and the corresponding points (dots) are more accurately aligned to a single position compared to the results from hand-eye calibration alone, even though the data were collected at different locations.

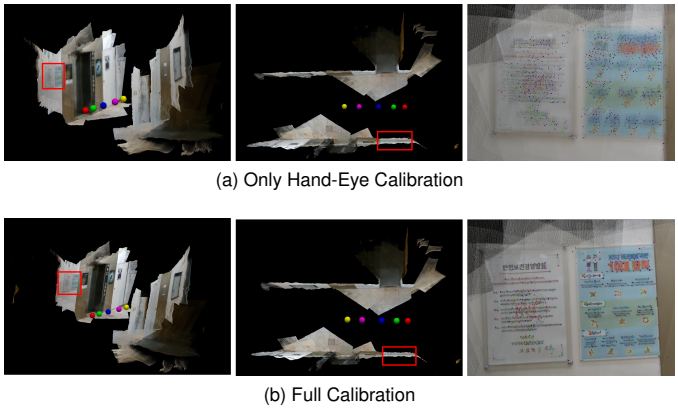


(a) Only Hand-Eye Calibration (b) Full Calibration

Fig. 4. **Calibration in a Small Area** This reflects a scenario in which the sensor's range limits the robot's movement. Relocating the robot around the table (large dots) allows for broader workspace coverage and enables calibration, as illustrated by the small dots in (b) compared to those in (a).

To evaluate whether the calibration result is accurate enough for object manipulation, an approach test is conducted using a chessboard. The calibration result from the small-area setup is applied to the robot arm. After the chessboard is detected using

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Only Hand-Eye Calibration

(b) Full Calibration

Fig. 5. **Calibration in a Large Area** This represents a scenario in which calibration is performed in a large environment, where feature points may or may not be detected in the surrounding images. By relocating the robot (large dots), feature points (small dots) can be extracted in a new environment, enabling successful calibration.

the camera mounted on the robot, the arm moves to a point 5 cm away from the target location (Fig. 6). Although the error cannot be precisely measured because the camera is positioned very close to the chessboard after the approach, the result provides a strong qualitative indication of the calibration's accuracy.

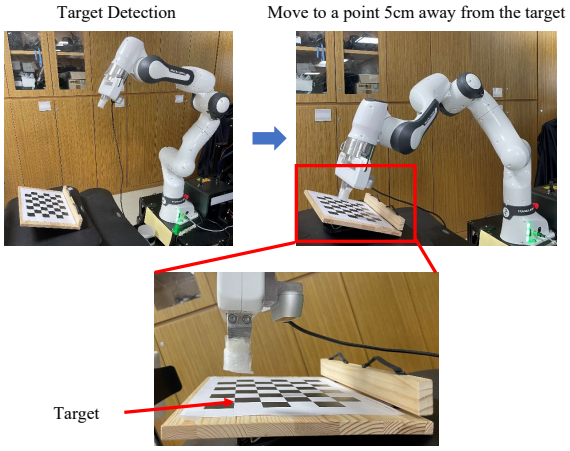


Fig. 6. **Approaching Test** A real robot (Franka Panda) approaches a target (chessboard) based on the results of chessboard recognition. The application of calibrated parameters enables accurate movement to the designated points (5 cm away from the target) in a qualitative manner.

A further experiment is conducted to evaluate the robustness of the proposed method. Data are collected in a cluttered environment, which presents a challenging scenario due to various sources of error, including thin desk edges, transparent glass surfaces, and other visual inferences (Fig. 7). Despite these challenges, the proposed method performs effectively (Fig. 7b). Compared to Fig. 7a, the point cloud of the chair within the red rectangle is better aligned. Although the results are not accurate enough for precise manipulation tasks due to depth noise, the method remains suitable for applications requiring lower precision.

TABLE II shows the time taken by each component. To obtain the optimal solution more effectively, hand-eye calibration is performed first. Using the resulting matrix T_{ee}^{c} for initialization, the full calibration is then carried out. The time of feature matching, outlier rejection, and pose estimation is for a pair of images.

(a) Only Hand-Eye Calibration (b) Full Calibration

Fig. 7. **Calibration in a Cluttered Area** This is intended for robustness testing. In a cluttered environment, various factors, such as the thin edge of the desk and transparent glass, can cause visual distractions. Despite these challenges, the calibration process can still be applied to data collected in such environments.

TABLE II
COMPUTATIONAL COST

		Runtime (s)
Feature Matching	2D Feat. Extract.	0.021
	Correspondence	0.049
Outlier Rejection	2D/3D error	0.002
	SSIM	0.037
Pose Estimation	Local	0.002
	Global	0.001
Optimization	only Hand-Eye	190.181
	Full	291.277
Total		682.284

C. Performance Comparison

To evaluate the effectiveness of the proposed method, its results are compared with two alternative approaches: (1) using a laser tracker and (2) using a chessboard pattern.

For the laser tracker approach, a reflector marker attached to the robot arm is measured using the Hexagon AT960 Leica Laser Tracker, and robot arm calibration is performed using RoboDyn software. Hand-eye calibration is then conducted separately using a chessboard pattern. In the chessboard pattern approach, the calibration process follows the same procedure as the proposed method, except that pose estimation is carried out by detecting the corners of the chessboard.

In TABLE III, the best result is achieved using the chessboard pattern. This is due to the high accuracy of pattern detection and clear correspondence matching. However, relying on a chessboard pattern placed at a fixed location limits the robot's operational range and does not account for the full workspace. The laser tracker approach may be optimal for

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE III
RESULT COMPARISON OF DIFFERENT METHODS.

method	error type	$mean \pm std$	RMSE
Laser	2D error (px)	1.85 ± 1.07	2.13
Tracker	3D error (mm)	1.55 ± 0.75	1.72
Chessboard	2D error (px)	0.96 ± 0.62	1.14
Pattern	3D error (mm)	1.10 ± 0.60	1.25
Proposed	2D error (px)	1.52 ± 0.90	1.76
	3D error (mm)	1.43 ± 0.69	1.59

calibrating only the robot arm, but it does not incorporate hand-eye calibration during the process. Furthermore, while the laser tracker itself may introduce calibration errors, there is no mechanism to compensate for them after calibration. In contrast, both the proposed method and the chessboard-based method consider errors in both arm and hand-eye calibration, which is why they outperform the laser tracker approach. Since downstream tasks rely on data from terminal sensors such as cameras, minimizing errors in this sensor data is essential.

V. LIMITATION AND CONCLUSION

This work proposes a self-calibration method that utilizes an RGB-D camera. The approach collects data using an image-based feature extraction method. However, image-based methods have a limitation: they struggle to extract feature points in textureless environments. Additionally, sensors often have a limited range, which can prevent feature extraction. To overcome these challenges, the robot is allowed to relocate, and data are collected from multiple positions. This enables the robot to approach textured areas or bring surrounding objects within the effective range of the sensors.

Moreover, to enhance robustness, a pairwise local-to-global pose estimation method is employed, incorporating outlier rejection based on 2D and 3D distance criteria. Additionally, the Structural Similarity Index Measure (SSIM) is used to exclude erroneous image pairs. We demonstrate that the proposed method is effective in both small and large environments. It also performs comparably to calibration techniques that rely on laser trackers or chessboard patterns.

The proposed method is theoretically applicable to a wide range of manipulators, but it has certain limitations. Although this work attempts to address data scarcity by relocating the robot base, it may still struggle in completely textureless environments, such as a featureless white room. Alternative approaches, such as dense matching (e.g., optical flow), could enhance the system's robustness. Furthermore, since an RGB-D camera serves as the primary sensor, calibration accuracy heavily depends on the precision of the depth measurements. While the proposed outlier rejection method partially alleviates this issue, it may still result in insufficient data under certain conditions.

Despite these limitations, the proposed method achieves precise calibration using data from multiple locations without requiring external devices or specialized patterns. The only equipment needed is the camera used for downstream tasks, which can positively impact the success rate of those tasks.

REFERENCES

[1] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022001312

[2] OECD, *OECD Employment Outlook 2019*, 2019. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/9ee00155-en

[3] T. Khawli, M. Anwar, and S. Islam, "A calibration method for laser guided robotic manipulation for industrial automation," 10 2018.

[4] Y. Liu, Y. Li, Z. Zhuang, and T. Song, "Improvement of robot accuracy with an optical tracking system," *Sensors*, vol. 20, no. 21, 2020.

[5] Z. Zhang, L. Zhang, and G.-Z. Yang, "A computationally efficient method for hand–eye calibration," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, 07 2017.

[6] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach." in *ISER*, ser. Springer Tracts in Advanced Robotics, vol. 79. Springer, 2010, pp. 211–225.

[7] H. Y. Jinghui Li, Akitoshi Ito and Y. Maeda, "Simultaneous kinematic calibration, localization, and mapping (skclam) for industrial robot manipulators†," *Advanced Robotics*, vol. 33, no. 23, pp. 1225–1234, 2019.

[8] A. Peters, "Robot self-calibration using actuated 3d sensors," *Journal of Field Robotics*, vol. 41, pp. 327 – 346, 2022.

[9] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[10] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 143–11 152.

[11] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, "Image matching across wide baselines: From paper to practice," *Int. J. Comput. Vision*, vol. 129, no. 2, p. 517–547, Feb. 2021.

[13] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[14] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, p. 91–110, Nov. 2004.

[16] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008, similarity Matching in Computer Vision and Multimedia.

[17] F. Dellaert, S. Seitz, S. Thrun, and C. Thorpe, "Feature correspondence: A markov chain monte carlo approach," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2000.

[18] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 596–609.

[19] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," *CVPR*, 2021.

[20] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi, "COTR: Correspondence Transformer for Matching Across Images," 2021.

[21] J. Xu, J. L. Hoo, S. Dritsas, and J. G. Fernandez, "Hand-eye calibration for 2d laser profile scanners using straight edges of common objects," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102221, 2022.

[22] N. Heide, T. Emter, and J. Petereit, "Calibration of multiple 3d lidar sensors to a common vehicle frame," in *ISR 2018; 50th International Symposium on Robotics*, 2018, pp. 1–8.

[23] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–33 712.

[24] "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry*," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 2, pp. 103–107, 2015.

[25] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.