To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2013, 1–14

FULL PAPER

Simultaneous Kinematic Calibration, Localization, and Mapping (SKCLAM) for Industrial Robot Manipulators

Jinghui LI[a], Akitoshi ITO[a], Hiroyuki YAGUCHI[b] and Yusuke MAEDA[c*]

[a] *Department of Mechanical Engineering, Materials Science, and Ocean Engineering, Graduate School of Engineering Science, Yokohama National University, Yokohama, Japan*; [b] *Department of Systems Integration, Graduate School of Engineering, Yokohama National University, Yokohama, Japan*; [c] *Division of Systems Research, Faculty of Engineering, Yokohama National University, Yokohama, Japan*

(*v1.0 released January 2013*)

Recently, the demand for more accurate, productive, and economical robot manipulators is increasing in the robotics industry. However, a manipulator will produce kinematic errors during production. Thus low-cost kinematic calibration is demanded. Moreover, environmental mapping is also demanded to plan the motions of the manipulator. In this paper, we proposed a simultaneous kinematic calibration, localization, and mapping (SKCLAM) method, which can simultaneously calibrate the kinematic parameters of an industrial robot manipulator using a commercial RGB-D camera attached to its end effector to reconstruct its surroundings. In our method, the kinematic calibration is achieved with feature detection and epipolor geometry. Synthetic and real data experiments were conducted to verify the SKCLAM method. We succeeded in reducing the kinematic errors of the manipulator and reconstructing dense 3D maps of the workspace in the experiments.

Keywords: industrial robot manipulators; kinematic calibration; 3D reconstruction; RGB-D cameras; visual SLAM

1. Introduction

Robot technology has experienced a boom in recent years, particularly in the field of industrial robots. The industrial robot manipulator is an important component in industrial robotics and plays a key role in the industrial community. However, the kinematic parameters of the robot manipulators have initial errors and could be changed as a result of mechanical aging or manufacturing accidents during operation. Traditional kinematic calibration approaches require expensive equipment and complex steps that increases cost and decreases productivity.

Here, in order to calibrate the kinematic parameters economically, efficiently, and safely, a method in which simultaneous localization and mapping (SLAM) is applied to the kinematic calibration of robot manipulators is considered, which will be named as simultaneous kinematic calibration, localization, and mapping (SKCLAM) method. Unlike traditional calibration approaches, it can simultaneously calibrate the kinematic parameters of a robot manipulator and create a map of its surroundings using an RGB-D camera. In addition, this map can be used for manipulator localization and motion planning to avoid obstacles in the workspace. This would reduce manual maintenance and improve productivity.

In this paper, we propose a SKCLAM approach that can simultaneously calibrate the kinematic parameters of a 6-degrees of freedom (DOF) industrial robot manipulator and reconstruct its

*Yusuke MAEDA. Email: maeda@ynu.ac.jp

environment using an RGB-D camera mounted on the end effector. We also verify the SKCLAM approach in both the virtual and real environments.

The remainder of this paper is arranged as follows: Initially, several related works are introduced in Section 2, and the SKCLAM problem is introduced in Section 3, where we explain our method in detail. In Section 4, we perform experiments to verify our method in the virtual and real environments. We also evaluate the performance of our method in this section. Finally, we provide a conclusion and a few remarks of this work in Section 5.

2. Related works

The SKCLAM method is to combine the kinematic calibration of the robot manipulator and the RGB-D camera-based visual SLAM. Kinematic calibration is a process of identifying the kinematic parameters of the manipulator to obtain better absolute positioning accuracy. The RGB-D SLAM method is a way for the robot to reconstruct its surroundings and localize itself while it explores in an unknown environment using the image data from the RGB-D camera.

2.1 *Kinematic calibration*

With the development of industrial robotics, robot manipulators are being increasingly applied to various workspaces, where they are expected to perform various tasks accurately and efficiently. However, some errors always exist between the nominal and true kinematic parameters owing to aging and assembly/fabrication tolerances. This kinematic error could decrease the accuracy of the robot manipulator, which means that the quality of products cannot be guaranteed. Therefore, kinematic calibration of the robot manipulator is necessary to avoid decreased productivity. A large number of existing studies focus on this field. For example, Hollerbach [1] reviewed the traditional methods of kinematic calibration. Furthermore, most of the calibration methods that have been proposed require large amounts of measurement to calibrate the robot [2–4]. Nowadays, there are plenty of approaches that can achieve kinematic calibration. With the restriction of cost, vision-based kinematic calibration methods are an effective way to improve the absolute positioning accuracy of a robot using low-cost equipment. Zhuang and Roth [5] published a weighty book on camera-based robot calibration. Watanabe et al. [6] proposed an effective method that achieves kinematic calibration using a CCD camera. This method not only improves the accuracy of the industrial manipulator but also increases the efficiency of visual measurement. In [7], Du and Zhang proposed a vision-based self-calibration method for serial-link robot manipulators. It can calibrate the kinematic parameters of the manipulator by employing the information from an RGB camera, which is attached to the manipulator end effector. A stereo vision system for autonomous kinematic calibration was reported in [8]. This system achieves kinematic calibration using a fixed stereo camera and a planar marker attached to the end effector. In contrast to previous studies on kinematic calibration, SKCLAM focus on not only calibration itself but also simultaneous environmental mapping.

2.2 *RGB-D SLAM*

The SLAM technology has developed rapidly over the past decades [9]. This was achieved by employing expensive or economical sensors, such as sonar sensors [10], laser scanners [11], and IR sensors [12]. Recently, RGB-D cameras are achieving superior performances at low prices. Meanwhile, the computational capability of computers is becoming more powerful, along with well-developed probabilistic information processing technology. Several studies have introduced RGB-D cameras, such as Microsoft Kinect and Intel RealSense, to solve the SLAM problem. They can provide appearance and depth information for mapping the environment at low cost. The scientific RGB-D SLAM system was first proposed by Henry et al. [13]. They generated a

dense 3D map of indoor environments using an RGB-D camera. They chose the iterative closest point (ICP) algorithm to match features between the images and employed the random sample consensus (RANSAC) method to optimize the feature matches. In [14], Enders et al. presented a 3D mapping system which creates a highly accurate map of the environment using an RGB-D camera (Microsoft Kinect). This RGB-D SLAM system succeeded in mapping challenging environments robustly.

2.3 *Simultaneous calibration and mapping*

The simultaneous kinematic calibration and mapping (SLAM) approach has been used in several recent studies. In [15], Kümmerle et al. achieved parameter estimation for a mobile robot while performing SLAM with an on-board RGB-D camera and a wheel encoder. Zhi and Schwertfeger [16] proposed an approach for achieving the hand–eye calibration without using calibration targets. They employed triangulation and bundle adjustment to optimize both the hand–eye calibration and robot–world calibration. They not only calibrated the kinematic parameters but also reconstructed the environment by employing the SIFT features. However, only sparse point clouds could be obtained. In contrast, our approach calibrates kinematic parameters using RGB information and reconstructs the environment densely with the use of corresponding depth information. That is a benefit while planning the motion of the manipulator. The work by Klingensmith et al. [17] is the most similar to our study. They developed articulated robot motion for simultaneous localization and mapping (ARM-SLAM), in which a multi-jointed robot manipulator with a hand-mounted RGB-D camera (Microsoft Kinect) simultaneously localized itself and reconstructed the surrounding environment. However, they considered only joint angle errors because the employed robot manipulator was lightweight and wire-driven with low rigidity and the internal sensor error was assumed to be large. In our study, we consider other kinematic errors in the SKCLAM method for a 6-DOF industrial manipulator whose internal sensor error is comparatively smaller than wire-driven robot arms.

The original contributions of this paper can be summarized as follows:

- **The proposal of the concept of SKCLAM that can calibrate kinematic parameters of industrial manipulators and obtain 3D dense maps around them simultaneously.**
- **Concrete implementation of the SKCLAM with an RGB-D camera in a cost-effective way.**
- **Quantitative validation of the SKCLAM both in virtual and real environments.**

3. Method

The components of the SKCLAM system include a 6-DOF industrial robot manipulator and an RGB-D camera mounted on its end effector. The RGB-D camera is used to capture the RGB-D information for the task to calibrate the kinematic parameters of the robot manipulator while reconstructing the workspace around the manipulator.

Therefore, the following procedure for implementing SKCLAM method is proposed:

(1) Employ an RGB-D camera mounted on the manipulator end effector to capture RGB images and corresponding depth images. Meanwhile, record the variables of each manipulator's joint at that moment.
(2) Detect and describe the feature points from the captured RGB images, and match these feature points.
(3) Employ the joint variables from step 1 to calculate the homogeneous transformation matrix from the base coordinate system of the robot manipulator to that of the RGB-D camera. Meanwhile, the fundamental matrix of the RGB-D camera also needs to be clarified.

(4) Employ the feature matches from step 2 and the fundamental matrix of the camera from step 3 to draw epipolar lines. Two images of the same scene are related by epipolar geometry.
(5) Define an error function based on the epipolar geometry and minimize the error function for the kinematic calibration.
(6) Reconstruct a map of the environment by employing the calibrated kinematic parameters and the point cloud data from the depth images.

The SKCLAM procedure will be discussed in greater detail below.

3.1 *Kinematic model*

In order to accomplish the SKCLAM method, some preparatory work needs to be done. First, it is necessary to formulate the kinematic model for the SKCLAM problem. To address this issue, we mainly use the Denavit–Hartenberg (DH) parameters. The Hayati parameters are also used to avoid the singularity that arises when trying to express the relationship of adjacent joint axes [18].

In the SKCLAM problem, the kinematic parameters include four transformation parameters (a_i, α_i, d_i or β_i, θ_i, $i = 2, \ldots, 6$). However, in this paper, we only evaluate the calibration performance of the DH parameters α_i and θ_i due to the lower sensitivity of the other parameters to RGB images. Furthermore, we assume that the intrinsic camera parameters are already calibrated. In fact, we calibrate the offset of $\theta_i^{'}$ which is defined by

$$\theta_i = \theta_i^{'} + \Delta\theta_i, \tag{1}$$

where $\Delta\theta_i$ is the variation of the joint angle.

The SKCLAM method employs the captured images to calibrate the kinematic parameters and map the workspace; therefore, the relationship between the RGB-D camera and the robot manipulator also needs to be established. The details of step 3 are as follows:

(1) Calculate the homogeneous transformation matrix from the base coordinate system of the robot manipulator to that of the RGB-D camera. Let us define $\boldsymbol{T}_1$ as the homogeneous transformation matrix of the RGB-D camera in the previous manipulator posture and $\boldsymbol{T}_2$ as that in the current posture. Then, we obtain the homogeneous transformation matrix from the previous posture to the current posture as follows:

$$\boldsymbol{T}_{21} = \boldsymbol{T}_1^{-1}\boldsymbol{T}_2. \tag{2}$$

(2) Calculate the fundamental matrix $\boldsymbol{F}$ of the RGB-D camera by employing the intrinsic parameter $\boldsymbol{K}$ as

$$\boldsymbol{F} = \boldsymbol{K}^{-T}\boldsymbol{E}\boldsymbol{K}^{-1}, \tag{3}$$

where the essential matrix $\boldsymbol{E}$ is written by

$$\boldsymbol{E} = [\boldsymbol{t}]_{\times}\boldsymbol{R}, \tag{4}$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotation matrix and the translation vector of $\boldsymbol{T}_{21}$, respectively; $[\boldsymbol{t}]_{\times}$ is the following skew-symmetric matrix calculated from $\boldsymbol{t}(= (t_x, t_y.t_z))$:

$$[\boldsymbol{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \tag{5}$$

So far, we have formulated the kinematic model and established the relationship between the camera coordinate system and robot manipulator coordinate system.

3.2 *Feature detection, description, and matching*

The SIFT [19] is often used for feature description owing to its good robustness to scale, rotation, and viewpoint changes in RGB images. In the SKCLAM method, considering both good robustness and fast computation speed, we utilize the AKAZE algorithm [20] to detect, describe, and match features from consecutive captured RGB images using OpenCV.

For feature matching, we choose the brute force matching method, which is the most basic and obvious feature matching method. As for the SKCLAM problem, the quality of feature matches will have a significant effect on the kinematic calibration. Therefore, we propose a combined method to filter out bad feature matches to obtain sufficient performance of the SKCLAM. This method is the one that combines the cross-check, ratio test, and epipolar geometry-based elimination methods.

3.2.1 Cross-check

The cross-check is also one of the most basic methods for filtering out bad feature matches. In simple terms, if two feature points in a training image and a query image can match one another, they are viewed as a good match.

3.2.2 Ratio test

In the problem of feature matching, many features will not have any reliable match in the training images because they arise from background clutter or are not detected in the training images. When some feature descriptors are much more discriminative than others, using the cross-check method to obtain the closest feature probably does not perform well.

To obtain more reliable feature matches, Lowe [19] developed an alternative measure known as the ratio test. This measure compares the Euclidean distances of the closest and second-closest feature points to obtain the best feature match. First, we employ the k-nearest neighbors (kNN) algorithm, which returns k best matches. Here, $k = 2$. Then, suppose d_1 and d_2 are the distances to the closest and second-closest feature points, respectively. In order to obtain the best match, the d_1/d_2 ratio ($0 < d_1/d_2 < 1$) should be smaller than a threshold. In other words, if the distance of the closest feature point is significantly closer than that of the second-closest feature point, it will be regarded as a reliable match. Otherwise, it is an unreliable match.

The selection of the threshold influences the feature matching result. In [19], Lowe recommended 0.8 as the threshold of the ratio test. Theoretically, a smaller threshold will select fewer matches although the matches will be more accurate. However, each environment has different feature information. If the images can provide rich feature information, choosing a smaller threshold for the ratio test will return a sufficient number of reliable matches. On the other hand, if the images have poor feature information, setting a larger threshold for the ratio test will be a better choice to obtain sufficient matches. This is because a smaller threshold will return fewer matches.

Consequently, we chose 0.6 as the threshold of the ratio test for obtaining accurate feature matches in our experiment environment.

3.2.3 Epipolar geometry-based elimination

Some outliers are generated by the kinematic errors of the robot manipulator. This will have a negative effect on feature matching and will influence the performance of the kinematic calibration. We use the nominal value of kinematic calibration to calculate the epipolar lines in query images. When a feature point deviates from the epipolar lines in the query image by more than the threshold, this feature point will be regarded as an outlier. That is, the matches that correspond to this outlier will be regarded as unreliable. We call this method “epipolar

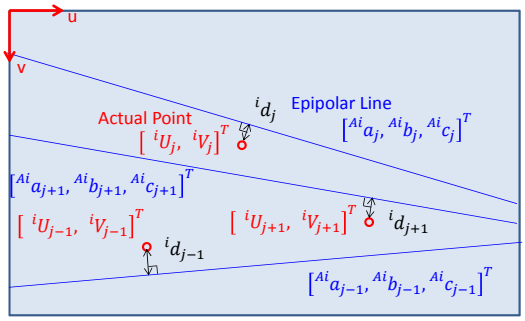


Figure 1. Error function (Epipolar geometry)

geometry-based elimination". d_{th}, a threshold, is used to eliminate outliers. It is related to the horizontal resolution and horizontal field of view (HFOV) of the camera properties, as follows:

$$d_{th} = \frac{\theta_{th}}{\theta_h} q_h, \tag{6}$$

where θ_h is the HFOV of the camera. q_h is the horizontal resolution of the shot image; θ_{th} is a threshold of the angle, which is adjusted according to the actual circumstances.

In our method, we choose $d_{th} = 200$px (equivalently, $\theta_{th} = 7.3°$) as the maximum distance for keeping an inlier within the epipolar geometry-based elimination.

As mentioned above, we desire to combine these three methods for filtering out bad feature matches. First, we employ the brute force matching to search all of the possible feature matches. Then, we return two pairs of better feature matches between the training images and query images using the kNN algorithm. Next, we choose reliable feature matches among them using the ratio test. After that, employ epipolar geometry-based elimination to discard the questionable feature matches, which are selected by the ratio test. Lastly, employ the cross-check method to check the rest of the feature matches between the query images and training images. If these two feature matches have the same feature descriptors, these are identical and should not be discarded.

Finally, the 30 best matches are selected by Euclidean distance to serve the kinematic calibration. Additionally, the RANSAC algorithm is used to eliminate bad matches.

3.3 *Error function*

In the SKCLAM problem, epipolar geometry is utilized to express the relationship between any RGB images captured in different postures. In order to minimize the error, we define an error function $F(\mathbf{\Phi})$ based on the distance between each of the epipolar lines and its corresponding point (Figure 1) as

$$F(\mathbf{\Phi}) = \sqrt{\frac{E_e(\mathbf{\Phi})}{2\sum_{i=1}^{N} M_i}}, \tag{7}$$

where $E_e(\mathbf{\Phi})$ is the sum of the distances written as

$$E_e(\mathbf{\Phi}) = \sum_{i=1}^{N-1} \left(\sum_{j=1}^{M_i} {}^{Ai}d_j(\mathbf{\Phi},\ {}^{Ai}\mathbf{\Psi}_j(\mathbf{\Phi},\ \boldsymbol{J}_{Ai},\ \boldsymbol{J}_{Bi})) \right) + \sum_{i=1}^{N-1} \left(\sum_{j=1}^{M_i} {}^{Bi}d_j(\mathbf{\Phi},\ {}^{Bi}\mathbf{\Psi}_j(\mathbf{\Phi},\ \boldsymbol{J}_{Ai},\ \boldsymbol{J}_{Bi})) \right). \tag{8}$$

In the equation of $E_e(\mathbf{\Phi})$, some notations are explained as follows:

- $\mathbf{\Phi}$: The DH or Hayati parameters of the robot manipulator to be calibrated.
- N: The number of posture pairs used for kinematic calibration.
- A_i, B_i: Previous and current postures of the robot manipulator in the i-th posture pair, respectively.
- $\boldsymbol{J}_{Ai}$, $\boldsymbol{J}_{Bi}$: The joint variables for A_i and B_i, respectively.
- M_i: The number of feature matches in the i-th posture pair.
- ${}^{Ai}\mathbf{\Psi}_j = \left({}^{Ai}a_j\ {}^{Ai}b_j\ {}^{Ai}c_j\right)$, ${}^{Bi}\mathbf{\Psi}_j = \left({}^{Bi}a_j\ {}^{Bi}b_j\ {}^{Bi}c_j\right)$: The parameters of the epipolar line corresponding to the feature point j ($j = 1, 2, \ldots, M_i$) for A_i and B_i, respectively.
- ${}^{Ai}d_j$, ${}^{Bi}d_j$: The distance between the feature point j ($j = 1, 2, \ldots, M_i$) and its corresponding epipolar line for A_i and B_i.

In order to minimize the error function $F(\mathbf{\Phi})$, we choose the Nelder–Mead method. Thus, the problem of kinematic parameter calibration can be converted to minimization of the error function $F(\mathbf{\Phi})$.

4. Experiments

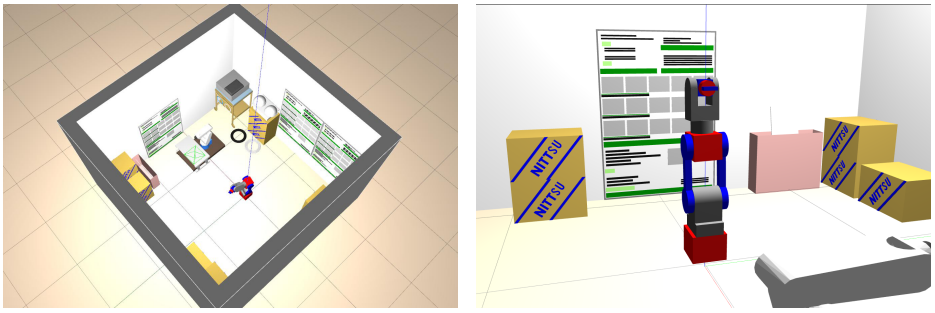
In this section, we conduct simulations to validate our method in a virtual environment because a synthetic data experiment is a necessary step before testing in practice. Moreover, we also conduct verification in the real environment. All of the experiment results were obtained on a PC with an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz in Ubuntu 16.04 LTS.

4.1 *Synthetic data experiments*

First, we built a virtual environment to simulate the workspace (Figure 2(a)) using Gazebo. According to [21], a virtual 6-DOF industrial robot manipulator (Figure 2(b)) was constructed. Then, we ran the SKCLAM program in the Robot Operating System (ROS). In synthetic data experiments, we can change the postures of a virtual manipulator to capture scenes through a virtual RGB-D camera. The scenes are then used to calibrate the kinematic parameters of the manipulator and evaluate the SKCLAM method.

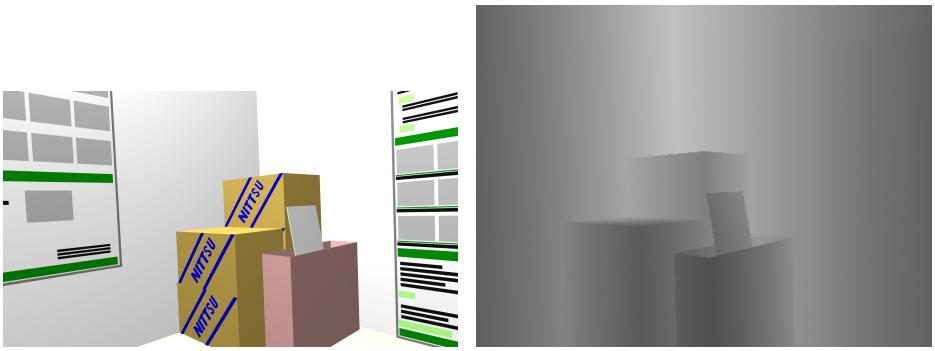
4.1.1 Kinematic calibration

We used a virtual RGB-D camera mounted on the end effector of the manipulator to capture scenes of the surroundings without distortion. First, we specified a pose of the end effector of the manipulator for capturing RGB-D information of the workspace. Then the inverse kinematics problem of the 6-DOF manipulator for the pose was solved to generate postures of the manipulator; that is, we can obtain eight different postures of the manipulator from one end effector pose by inverse kinematics. The first posture of the eight was used for calibration. Then the second posture among the eight was taken but we rotated the joint 1 of the manipulator by 5°, and the posture was used for calibration, too. Similarly, the other postures among the eight were taken with 5° joint 1 rotation. Thus we have eight postures of the manipulator for calibration



(a) Virtual environment (b) Virtual manipulator

Figure 2. Virtual environment



(a) RGB image (1920×1080) (b) Depth image (480×360)

Figure 3. An example of captured RGB-D images

that cover 0°∼ 35° view orientations. We set 5° for joint 1 rotation to obtain sufficient overlaps of camera images. Repeating the above process nine times, 72 postures were generated to cover 0°∼ 355° view orientations. We added the 73rd posture manually, which was the same as the first posture. These postures can support the virtual RGB-D camera to capture 73 RGB images (1920×1080) (Figure 3(a)) and the corresponding depth images (480×360) (Figure 3(b)) with view orientations from 0° to 360°. Next, we used adjacent postures of them as pairs (72 pairs) for calibration according to the procedure in Section 3.

For the purpose of verifying the SKCLAM method, we added 1° error into the kinematic parameters to be calibrated. Subsequently, we performed simulations and obtained the results presented in Table 1. Here, the results of the joint values before and after kinematic calibration by our method are listed. Furthermore, we recorded the CPU time of the SKCLAM program for evaluation.

Additionally, the robustness to image noise was also considered. We employed OpenCV to add colored Gaussian noise (20,000 colored pixels scattered randomly) and Gaussian blur (Gaussian kernel size is 9×9) into every captured RGB image. We then used these noise-added images to calibrate the kinematic parameters in the same virtual environment. The results are also listed in Table 1.

From Table 1, the difference between the true and calibrated values remained in $[-0.06°, +0.04°]$. We calculated the mean and standard deviation of the position errors of the end effector from the 73 postures before and after kinematic calibration. Note that the ground truth of the end effector position is available in the virtual environment. In addition, we calculated the variation of the error function for evaluating the calibration results. The further evaluations

Table 1. Kinematic parameters

DH parameters	Nominal value	True value	Calibrated value (without noise)	Calibrated value (with noise)
α_2 [deg]	90	91	91.00	91.00
$\theta_2^{'}$ [deg]	90	91	90.96	91.04
α_3 [deg]	180	181	180.98	181.00
$\theta_3^{'}$ [deg]	−90	−89	−89.06	−88.98
α_4 [deg]	−90	−89	−89.00	−89.02
$\theta_4^{'}$ [deg]	0	1	1.00	1.02
α_5 [deg]	90	91	91.00	91.03
$\theta_5^{'}$ [deg]	0	1	0.98	0.98
α_6 [deg]	−90	−89	−89.03	−89.06
$\theta_6^{'}$ [deg]	−90	−89	−88.86	−89.00
CPU Time [s]	/	/	164.7	185.1

Table 2. Absolute position error of end effector (virtual environment) [mm]

	Nominal value	Calibrated value
Without Noise	6.94	0.25±0.05
With Noise	6.94	0.27±0.05

[a]The number of calibrated represent mean ± standard deviation

Table 3. The error function variation (virtual environment) [px]

	Before Calibration	After Calibration
Without Noise	55.35	0.68
With Noise	53.12	0.80

are reported in Tables 2 and 3.

As shown in Tables 2 and 3, both the absolute position error of the end effector and the error function were significantly optimized, regardless of the captured RGB images with or without Gaussian noise and Gaussian blur. That is, the SKCLAM method is able to achieve kinematic calibration.

4.1.2 Mapping

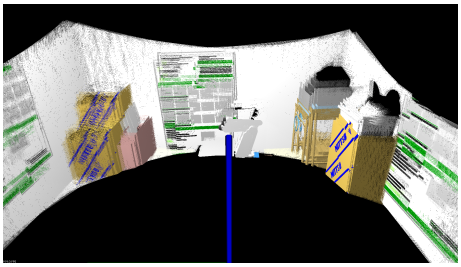
We employed the RGB images (1920×1080) and their corresponding depth images (480×360) to calculate the point cloud data. We reconstructed 3D models of the virtual environment using the point cloud data and calibrated kinematic parameters in the PCL.

According to Figure 4, the reconstruction accuracy is improved after kinematic calibration. However, the mapping quality in Figures 4(b), 4(c), and 4(d) are almost the same because in this ideal virtual simulation, the SKCLAM method performed well. Consequently, verification of the SKCLAM method in the real data experiment is necessary.

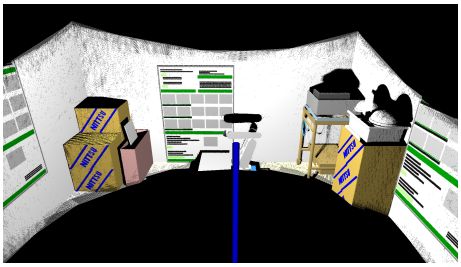
4.2 *Real data experiments*

We confirmed the SKCLAM method in the real environment (Figure 5(a)) using a 6-DOF industrial robot manipulator (Yaskawa Motoman-HP3J [21]) equipped with an RGB-D sensor (Intel RealSense R200 [22]) on the end effector (Figure 5(b)).

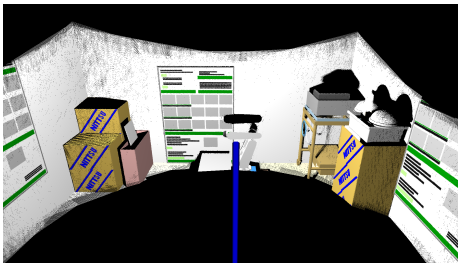
In the real data experiment, we did not know the true values of the kinematic parameters. However, the optimization results of the error function could evaluate the performance of the kinematic calibration to some extent. Moreover, the quality of the reconstructed 3D environment model also could directly reflect the results of the kinematic calibration. Additionally, an external laser displacement sensor was employed for evaluating the error of the end effector before and after calibration.



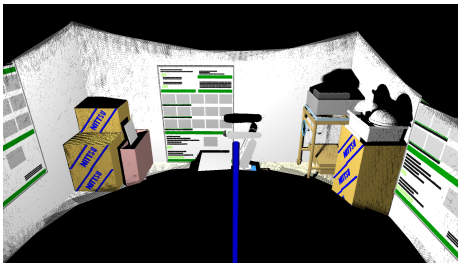
(a) Nominal value (without noise)



(b) True value (without noise)

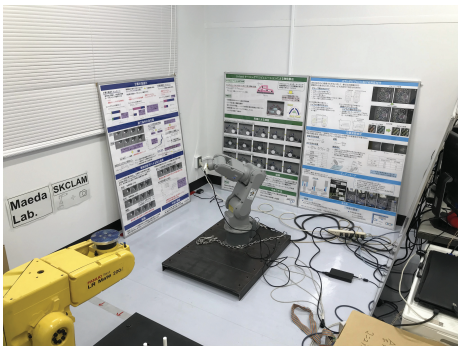


(c) After calibration (without noise)

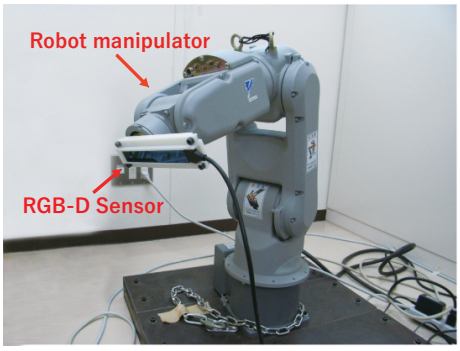


(d) After calibration (with noise)

Figure 4. Environment mapping



(a) Real environment



(b) SKCLAM target

Figure 5. Real environment

Table 4. The error function variation (real environment)

Before Calibration [px]	After Calibration [px]	CPU Time [s]
63.63	21.43	442.6

4.2.1 Kinematic calibration

In our real data experiments, we evaluated only the calibration performance of the DH parameters α_i and θ_i $(i = 2, \ldots, 6)$. Here, we calibrate the offset of $\theta_i^{'}$, which satisfies equation (1). We used the RGB-D sensor mounted on the end effector of the manipulator to capture RGB-D images for 73 different postures, as in Section 4.1. Finally, we obtained 73 RGB-D images (1920×1080). We used neighbor postures as pairs for calibration (72 pairs).

In terms of the error function variation, Table 4 reflects the optimization result that the value

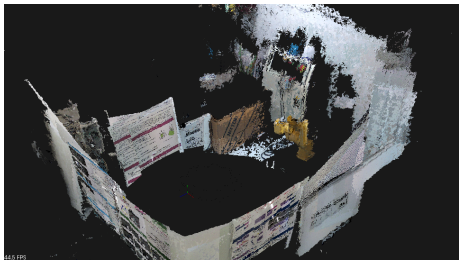


Figure 6. Reconstructed laboratory

of the error function decreased after kinematic calibration.

4.2.2 Mapping

We constructed the point cloud map by employing the calibrated kinematic parameters (Figure 6). To evaluate the quality of the map and the accuracy after kinematic calibration, some details will be compared between the non-calibrated and calibrated maps in Figure 7. The non-calibrated map is shown as Figures 7(a), 7(c), and 7(e), and the calibrated map is shown as Figures 7(b), 7(d), and 7(f). These reconstructed maps show some differences. For instance, in Figures 7(a) and 7(b), the restoration accuracy is improved around the control box in the left by the calibration.

4.2.3 End effector position error

An external laser displacement sensor (SUNX HL-C135C-BK10 [23], $10\mu m$ resolution) was employed for evaluating the position error of the end effector before and after calibration. Because it is difficult to directly measure the position errors of the end effector, we measured the distances between the laser sensor and the end effector in some poses in which the end effector pose should be identical, but joint variables are not identical. If the kinematic calibration is successful, the discrepancy of the distances should be smaller. The poses were obtained by solving inverse kinematics numerically with kinematic parameters before or after calibration. Firstly, five different poses of the end effector were selected, and eight inverse kinematics solutions for each of the pose were obtained with calibrated and non-calibrated parameters (total 40 postures for each). Next, we used the laser displacement sensor to measure the distance between the end effector and itself in X, Y, and Z directions in each of the postures. For clarifying the positional variations of the end effector before and after calibration, we calculated the standard deviation of the distances in each posture group for the specified end effector poses. The results are presented in Figure 8.

As shown in Figure 8, the distance variation of the end effector was decreased after calibration, which indicates the accuracy improvement by SKCLAM.

5. Conclusion

In this paper, we proposed an approach called SKCLAM. The SKCLAM method can simultaneously calibrate the kinematic parameters of an industrial robot manipulator and reconstruct the environment of its surroundings using an RGB-D camera attached to its end effector. This way a more economical and uncomplicated method of kinematic calibration and environment reconstruction was achieved.

We performed experiments in both virtual and real environments to confirm that our method calibrates the kinematic parameters of industrial robot manipulators and builds a dense environment map in 3D. In the synthetic data experiments, the proposed method could tolerate image noise. However, considering the error function variation and the quality of the calibrated maps in real data experiments, the calibration and reconstruction accuracy were not perfect. Because SKCLAM is an integrative problem, we could also acquire better performance by mod-

(a) Front of the manipulator (non-calibrated)

(b) Front of the manipulator (calibrated)

(c) Corner of laboratory (non-calibrated)

(d) Corner of laboratory (calibrated)

(e) Back of the manipulator (non-calibrated)

(f) Back of the manipulator (calibrated)

Figure 7. Reconstructed real environment in different viewpoint

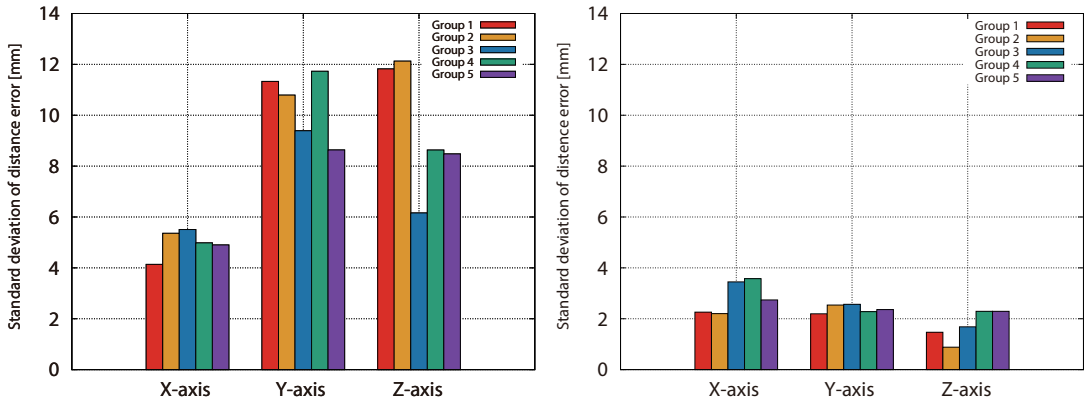
ifying other components in this problem, such as camera calibration. In future work, more real data experiments will be conducted to test and evaluate our approach in various environments, including more challenging ones.

Disclosure statement

No potential conflict of interest was reported by the authors. Jinghui Li and Akitoshi Ito contributed equally to this work.

Funding

This work was supported by JSPS KAKENHI Grant Number JP18K04045.



(a) Before calibration (b) After calibration

Figure 8. The standard deviation of distances between end effector and laser sensor

ORCID

Jinghui Li: https://orcid.org/0000-0001-9953-2354
Yusuke Maeda: https://orcid.org/0000-0002-9654-6117

Notes on contributors

Jinghui LI received the B.E. degree from Shenyang University of Chemical Technology in 2015 and his M.S. degree from Gunma University in 2018. He is currently a Ph.D. candidate in Graduate School of Engineering Science, Yokohama National University.

Akitoshi ITO received the B.E. degree from Yokohama National University in 2018. He is currently a Master's student in Graduate School of Engineering Science, Yokohama National University.

Hiroyuki YAGUCHI received the B.E. and M.E. degrees from Yokohama National University in 2016 and 2018, respectively.

Yusuke MAEDA received the B.E., M.E. and D. Eng. degrees from The University of Tokyo in 1995, 1997 and 2003, respectively. From 1997 to 1999, he worked in Dai Nippon Printing Co., Ltd. From 1999 to 2004, he was a Research Associate at Department of Precision Engineering, The University of Tokyo. He joined Division of Systems Research, Faculty of Engineering, Yokohama National University as a Lecturer in 2004. From 2005, he is an Associate Professor of the division. His research interests includes robotic manipulation, robot programming, and decentralized manufacturing systems. Dr. Maeda is a member of RSJ, JSME, JSPE, SICE and IEEE-RAS.

References

[1] Hollerbach JM, Wampler CW. The calibration index and taxonomy for robot kinematic calibration methods. The International Journal of Robotics Research. 1996;15(6):573-591.
[2] Park I, Lee B, Cho S, et al. Laser-based kinematic calibration of robot manipulator using differential kinematics. IEEE/ASME Transactions on Mechatronics. 2012;17(6):1059–1067.

[3] Hwangbo M, Kim J, Kanade T. IMU self-calibration using factorization. IEEE Transactions on Robotics. 2013;29(2):493–507.
[4] Du G, Shao H, Chen Y, et al. An online method for serial robot self-calibration with CMAC and UKF. Robotics and Computer-Integrated Manufacturing. 2016;42:39–48.
[5] Zhuang H, Roth ZS. Camera-aided robot calibration. Boca Raton (FL): CRC Press; 1996.
[6] Watanabe A, Sakakibara S, Ban K, et al. A kinematic calibration method for industrial robots using autonomous visual measurement. CIRP Annals. 2006;55(1):1–6.
[7] Du G, Zhang P. Online robot calibration based on vision measurement. Robotics and Computer-Integrated Manufacturing. 2013;29(6):484–492.
[8] Zhang X, Song Y, Yang Y, et al. Stereo vision based autonomous robot calibration. Robotics and Autonomous Systems. 2017;93:43–51.
[9] Yousif K, Bab-Hadiashar A, Hoseinnezhad R. An overview to visual odometry and visual SLAM: Applications to mobile robotics. Intelligent Industrial Systems. 2015;1(4):289–311.
[10] Kleeman L. Advanced sonar and odometry error modeling for simultaneous localisation and map building. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2003 Oct 27-31; Las Vegas (NV), USA. IEEE; 2003. p. 699–704.
[11] Brenneke C, Wulf O, Wagner B. Using 3D laser range data for SLAM in outdoor environments. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2003 Oct 27-31; Las Vegas (NV), USA. IEEE; 2003. p. 1556–1563.
[12] Abrate F, Bona B, Indri M. Experimental EKF-based SLAM for mini-rovers with IR sensors only. In: Proceedings of the 3rd European Conference on Mobile Robots (ECMR); 2007 Sept 19-21; Freiburg, Germany. .
[13] Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research. 2012;31(5):647–663.
[14] Endres F, Hess J, Sturm J, et al. 3D Mapping with an RGB-D Camera. IEEE Transactions on Robotics. 2014;30(1):177–187.
[15] Kümmerle R, Grisetti G, Burgard W. Simultaneous parameter calibration, localization, and mapping. Advanced Robotics. 2012;26(17):2021–2041.
[16] Zhi X, Schwertfeger S. Simultaneous hand-eye calibration and reconstruction. In: Proceedings 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2017 Sept 24-28; Vancouver (BC), Canada. IEEE; 2017. p. 1470–1477.
[17] Klingensmith M, Srinivasa SS, Kaess M. Articulated robot motion for simultaneous localization and mapping (ARM-SLAM). IEEE Robotics and Automation Letters. 2016;1(2):1156–1163.
[18] Hollerbach J, Khalil W, Gautier M. Model Identification. In: Siciliano B, Khatib O, editors. Springer Handbook of Robotics 2nd Edition. Springer International Publishing; 2016. p. 115–122.
[19] Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004;60(2):91–110.
[20] Alcantarilla PF, Nuevo J, Bartoli A. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In: Proceedings of the British Machine Vision Conference (BMVC); BMVA Press; 2013 Sept 9-13; Bristol (UK); 2013.
[21] Yaskawa Motoman-HP3J: Industrial robot MOTOMAN-HP3J [Internet]; Available from: `http://www.motoman.dk/uploads/tx_catalogrobot/HP3J_EN_02.pdf`.
[22] Intel(R) RealSense(TM) Camera R200: Specifications [Internet]; Available from: `https://ark.intel.com/products/92256/Intel-RealSense-Camera-R200`.
[23] SUNX HL-C135C-BK10: Specifications [Internet]; Available from: `https://www3.panasonic.biz/ac/e/search_num/index.jsp?c=detail&part_no=HL-C135C-BK10`.