

Dynamic Calibration between a Mobile Robot and SLAM Device for Navigation

Ryoichi Ishikawa¹, Takeshi Oishi¹, and Katsushi Ikeuchi²

Abstract—In this paper, we propose a dynamic calibration between a mobile robot and a device using simultaneous localization and mapping (SLAM) technology, which we termed as the SLAM device, for a robot navigation system. The navigation framework assumes loose mounting of SLAM device for easy use and requires an online adjustment to remove localization errors. The online adjustment method dynamically corrects not only the calibration errors between the SLAM device and the part of the robot to which the device is attached but also the robot encoder errors by calibrating the whole body of the robot. The online adjustment assumes that the information of the external environment and shape information of the robot are consistent. In addition to the online adjustment, we also present an offline calibration between a robot and device. The offline calibration is motion-based and we clarify the most efficient method based on the number of degrees-of-freedom of the robot movement. Our method can be easily used for various types of robots with sufficiently precise localization for navigation. In the experiments, we confirm the parameters obtained via two types of offline calibration based on the degree of freedom of robot movement. We also validate the effectiveness of the online adjustment method by plotting localized position errors during a robots intense movement. Finally, we demonstrate the navigation using a SLAM device.

I. INTRODUCTION

Navigation is one of the most important issues for a robot to perform many tasks in various environments such as homes, factories, and outdoor fields. The robot navigation system estimates the 6-DoF pose of the robot in a 3D space and leads the robot to a destination. The robot uses different types of sensors including a camera, LiDAR, GPS, and IMU based on system and environmental requirements. Specifically, there are several studies on visual navigation because optical sensors and algorithms for camera tracking are widely available.

Various approaches for localizing a robot in a given environmental map were proposed over the past few decades. Global algorithms conduct the scan-to-map matching via the iterative closest point (ICP) algorithm [1], 2D Monte Carlo localization (MCL) [2], 3D MCL [3], view-sequence matching [4], and particle filters [5]. Local feature-based methods use certain types of landmarks such as two-dimensional bar code [6] and natural feature points [7].

Some parts of the experiments were conducted with the assistance of the Strategic prototyping group, Microsoft. We particularly appreciate the help extended by Yutaka Suzue. We also appreciate Yoshihiro Sato, Computer Vision Laboratory, The University of Tokyo. This work is partly supported by JSPS Research Fellow Grant No. 16J09277

¹Ryoichi Ishikawa and Takeshi Oishi are with Institute of Industrial Science, The University of Tokyo, Japan {ishikawa, oishi}@cvl.iis.u-tokyo.ac.jp

²Katsushi Ikeuchi is with Microsoft, Washington, U.S.A katsuike@microsoft.com

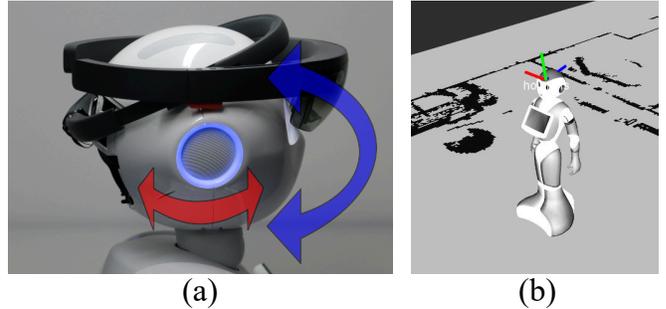


Fig. 1. (a) Attaching the SLAM device on a robot and calibrating the relative pose via moving the robot head, (b) After calibration, the SLAM device is located in front of the robot head, and the robot is localized in the map coordinate system.

Recently, simultaneous localization and mapping (SLAM)-based approaches are popular for robot navigation. Specifically, SLAM provides a 3D map while simultaneously localizing the robot via a monocular camera [8], [9], an RGB-D camera [10], and a LiDAR [11]. The entire SLAM system can be used as odometry to navigate a robot with several types of devices [12], [13], [14], [15].

We refer to this type of a 3D localization device as SLAM device. The device exhibits SLAM functionality and provides a 6-DoF pose for the device in real-time. We also refer to multi-functional devices, such as smartphone and head-mounted display (HMD) as the SLAM device. The devices are designed for augmented reality (AR) and mixed reality (MR). Thus, AR and MR require self-localization for rendering virtual objects in the real world and exhibit SLAM functionality.

In this paper, we propose an online calibration between a device and robot for a framework of the robot navigation using the SLAM device. The purpose of this method is easy navigation implementation for robots which is no visual sensor which is not suitable for SLAM. Since we assume that a SLAM device is loosely mounted on a robot because of easy and quick use, an online calibration between the device and robot is important for removing localization error. We also present offline calibration for computing relative pose between a device and a robot as shown in Fig. 1.

The online calibration considers consistency between the robot, environment, and the SLAM device. The position estimated by the SLAM device and inner state of the robot includes error due to the robot-device calibration error and the joints lags of the articulated robot. We introduce a calibration method between the SLAM device and the complete body to

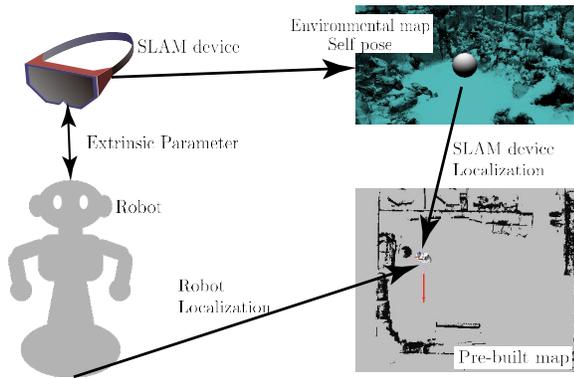


Fig. 2. Overview of the proposed localization system for navigation

refine the positioning of the robot whereas dynamic motion-based sensor calibrations calibrate two fixed sensors [19], [20]. In the process, calibration parameters are dynamically adjusted during robot moving to reduce the positioning error given the assumption that the robot is in contact with a floor.

The offline calibration method is based on hand-eye calibration [16], [17], [18]. The hand-eye calibration estimates the relative pose of two different systems from several egomotions of the systems. The motion-based approach is also used for calibration between camera-IMU [19], [20], camera-camera or LiDAR-camera [21], [22], [23]. However, the studies do not consider the degree of freedom of the systems. Therefore, we clarify necessary and efficient movements for the calibration with respect to the degree of freedom of the robot.

The proposed calibration method between the SLAM device and robot can be easily applied to various types of robots although the robot does not possess sufficient degrees of freedom. We demonstrate that accurate navigation is performed via the proposed online adjustment.

Figure 2 shows an overview of our localization system. The SLAM device offers an environmental map and a self-pose in the map. The SLAM device is localized in the pre-built map via aligning the environmental map in the SLAM device. The system also localizes a robot footprint in the map by using the extrinsic calibration parameter and robot pose information. In the following sections, we describe offline calibration between the SLAM device and robot and online adjustment of the localized robot footprint.

II. ONLINE WHOLE-BODY CALIBRATION

Online calibration is required between the SLAM device and the whole-body of the robot while offline calibration is used to calculate the extrinsic parameter between the SLAM device and the part of robot where the device is attached. The error in the robot-device calibration is unavoidable given the play of the attachment, encoder error at the calibration, and movement of the SLAM device as the robot is moving. Additionally, motion lags exist when controlling an articulated robot such as humanoid robot. Thus, the pose of the robot given by SLAM device is inaccurate and should be calibrated

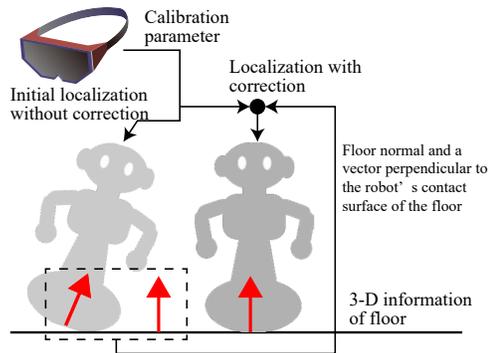


Fig. 3. Overview of the on-line adjustment method: Loop closure is partially created by aligning the floor normal and a vector perpendicular to the robot's contact surface of the floor.

online for accurate positioning of the robot. The overview of the process is shown in Fig. 3.

We calibrate the SLAM device and whole-body of the robot by closing a loop between the SLAM device, environmental map, and robot. The robot does not estimate the calibration error and lags only from the inputs by the SLAM device while moving. Conversely, the SLAM device knows the pose of the device in the 3D environmental map, and the robot knows its foot contacts to the floor which constitutes part of the environment. It is possible to correct the error by performing a consensus between them, and this requires a loop closing scheme.

The rotation error mainly contributes to the estimated position of the robot's foot if the SLAM device is attached to a relatively high position. The estimated position error is in proportion to r_x and r_y rotational errors aside from a rotation around vertical direction axis r_z (See Fig. 5) and SLAM device height. Even the r_x and r_y rotational errors are small, the localized position error becomes large whereas the small translation error not largely affects the position. The rotation parameter r_z also mainly contribute to the locomotion of the robot. The robot moves in wrong directions if a rotation error exists.

Therefore, we adjust only the rotation parameters. We assume that the robot stands perpendicular to the floor. The robot is inclined to the floor if there are errors in rotation parameters r_x and r_y . The remaining r_z parameter is corrected by using horizontal movement described in Sec. III-B.3. We align the vertical direction of the robot n to the surface normal of the floor n_f . The estimated vertical direction of the robot is calculated by the pose given by the SLAM device and inner state of the robot. Specifically, n_f is given by the map in the SLAM device. We define R_a that aligns n to n_f as follows:

$$n_f = R_a n. \quad (1)$$

Specifically, R_a is not uniquely determined because the rank of R_a corresponds to two. We compute R_a using the angle between n and n_f : θ . The Rodrigues rotation formula gives

		t_x	t_y	t_z	r_x	r_y	r_z
	Horizontal rotation	○	○		○	○	
	Vertical rotation	○		○	○		○
	Forward movement					○	○

Fig. 4. Relations between movements and obtained parameters

R_a as follows:

$$R_a = \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{a} \mathbf{a}^\top + \sin \theta [\mathbf{a}]_\times \quad (2)$$

where,

$$\mathbf{a} = -\frac{\mathbf{n}_f \times \mathbf{n}}{|\mathbf{n}_f \times \mathbf{n}|} \quad (3)$$

$$\theta = \cos^{-1}(\mathbf{n}_f \cdot \mathbf{n}). \quad (4)$$

Thus, R_a is the smallest rotation that satisfies Eq. 1. We can apply R_a to determine the refined position of the robots foot t_o from the SLAM device as follows:

$$t_o = -R_a \hat{R}^{-1} \hat{t} + R_a \hat{R}^{-1} t_b \quad (5)$$

where, \hat{R} and \hat{t} are the rotational and translational calibration parameter between the SLAM device and the robot, respectively. t_b corresponds to the vector from the origin of the robot coordinate for the calibration to the footprint of the robot. Specifically, t_b is obtained by the inner state of the robot (See Fig. 7).

This R_a removes 2-DoF rotational error aside from a rotation around vertical direction axis. These 2-DoF error components increase in proportion to r_x and r_y rotational error and SLAM device height h . The proposed adjustment is effective especially when h is large like humanoid type robots.

III. MOTION-BASED OFFLINE CALIBRATION

We introduce a motion-based offline calibration method between the SLAM device and parts of robot where the device is attached by considering the degree of freedom of the robot. First, we clarify the relation between motions for calibration and calibration parameters. Next, we describe the mathematical formulations for a detailed understanding. Finally, we present implementations of the motion-based calibration based on the degrees of freedom of the robot.

A. Relation between motion and calibration parameters

The motions that can be used for the calibration are limited by the degree of freedom of the robot. We assume that the SLAM device is attached to a body part of the robot. When we attach the device to a 6-DoF arm, the calibration process exactly corresponds to hand-eye calibration. Conversely, the device is attached to a body part, and possible motions are generally limited. For example, in the case that we attach the device to the head of the humanoid robot, the robot rotates

the device vertically and horizontally by swinging the neck in addition to translating horizontally. Conversely, the rover-type robot can rotate only in the horizontal direction and translate horizontally.

The primitive motions include horizontal rotation, vertical rotation, and forward translation. We do not consider any motion that contains multiple primitive motions as aforementioned for the purpose of simplicity and efficiency. A combination of the motions should correspond to the combination to solve all 6-DoF of the extrinsic parameters: 3-DoF translation parameter $\{t_x, t_y, t_z\}$, and 3-DoF rotation parameter $\{r_x, r_y, r_z\}$ around the x -axis, y -axis, and z -axis, respectively. As shown in Fig. 4, we obtain two translation parameters and two rotation parameters from a rotational motion. The parameters are orthogonal to the rotation axis (Figure 5 shows an example of calibration by horizontal rotation). Thus, it is not possible to obtain the relative translation and rotation information along the rotation axis. Hence, we can estimate 6-DoF calibration parameters if the robot rotates the SLAM device in two directions. If the robot cannot rotate in two orthogonal directions, we use the translation and additional information. The details are described in the following subsections.

B. Motion-based calibration

1) *Fundamentals*: We estimate the relative pose $\mathbf{X} = \{\hat{R}, \hat{t}\}$ between the SLAM device and robot from several motions. Let $\mathbf{A} = \{R_r, t_r\}$ correspond to the relative pose of the robot before and after a movement. Let $\mathbf{B} = \{R_s, t_s\}$ correspond to the relative pose of SLAM device. Specifically, \mathbf{A} is obtained by encoders or other sensors embedded in the robot. Additionally, \mathbf{B} is obtained by the SLAM function. As shown in Fig. 6, the pose of the SLAM device after the movement is expressed in two forms $\mathbf{A}\mathbf{X}$ and $\mathbf{X}\mathbf{B}$. Given that the two transformations are identical, the following two equations are derived

$$R_r \hat{R} = \hat{R} R_s \quad (6)$$

$$R_r \hat{t} + t_r = \hat{R} t_s + \hat{t}. \quad (7)$$

The relative rotation \hat{R} is solved via aligning rotation axes of R_s and R_r . Let k_s and k_r correspond to the rotation axes of R_s and R_r , respectively. The following expression is obtained from Eq. 6:

$$k_r = \hat{R} k_s. \quad (8)$$

Thus, \hat{R} is solved by several rotations. When \hat{R} is given, \hat{t} is solved by Eq. 7. Please refer to [16], [17], and [18] for more details.

2) *Horizontal/vertical rotation*: Here we consider the calibration by horizontal rotation. We first solve the rotation parameters. The horizontal rotation indicates that k_r in Eq. 8 corresponds to the z -axis in Fig. 5. The rotation axis of the SLAM device k_s is given by the device. Rotation is one dimensional, and thus 2-DoF rotation parameters of \hat{R} are obtained by solving Eq. 8. The parameters correspond to rotations around the axes that are orthogonal to the rotation axis z ; r_x and r_y are obtained.

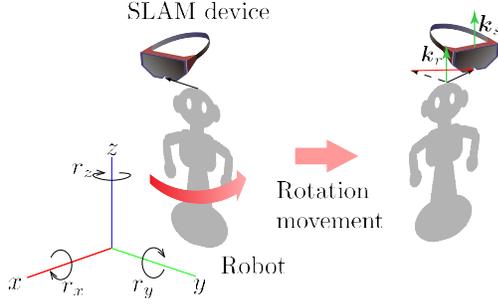


Fig. 5. Calibration by horizontal rotation movement. Alignment of the axes of rotation (Green arrows) gives rotational parameters r_x and r_y orthogonal to the axes of rotation. Translation difference (Red arrow) also offers rotational parameter t_x and t_y orthogonal to the axes of rotation.

Next, we solve the translation parameters, t_x and t_y . Equation 7 is transformed as follows;

$$(\mathbf{I} - \mathbf{R}_r)\hat{\mathbf{t}} = \mathbf{t}_r - \hat{\mathbf{R}}\mathbf{t}_s. \quad (9)$$

The rank of $(\mathbf{I} - \mathbf{R}_r)$ in the left term of Eq. 9 corresponds to two because the motion corresponds to a rotation. We decompose $\hat{\mathbf{t}}$ into \mathbf{k}_r and two unit vectors, which are orthogonal to \mathbf{k}_r and mutually orthogonal. The unit vectors are arbitrarily determined although they can correspond to the x and y directions.

$$\hat{\mathbf{t}} = a\mathbf{k}_r + b\mathbf{t}_x + c\mathbf{t}_y, \quad (10)$$

where, a , b , and c correspond to the variables. Additionally, \mathbf{k}_r does not change when it is rotated by \mathbf{R}_r since \mathbf{k}_r corresponds to the rotation axis of \mathbf{R}_r as follows:

$$\mathbf{k}_r = \mathbf{R}_r\mathbf{k}_r. \quad (11)$$

Therefore, Eq. 9 corresponds to the simultaneous equations for b , c :

$$(\mathbf{I} - \mathbf{R}_r)(b\mathbf{t}_x + c\mathbf{t}_y) = \mathbf{t}_r - \hat{\mathbf{R}}\mathbf{t}_s. \quad (12)$$

The translation parameters t_x , t_y are given by solving Eq. 12. The vertical rotation is identical to the case of horizontal rotation with a different rotation axis. The vertical rotation is performed around the y -axis, and we obtain the rotation parameters r_x , r_z and translation parameters t_x , t_z .

3) *Forward translation*: The rotation parameters are obtained via a forward translation of the robot. Here we assume that the robot moves straight. We can approximate $\mathbf{R}_r \approx \mathbf{I}$. Equation 7 is transformed as follows:

$$\mathbf{t}_r = \hat{\mathbf{R}}\mathbf{t}_s, \quad (13)$$

The form of the equation is identical to Eq. 8. Therefore, we obtain rotation parameters r_y , r_z when the robot moves in the x -direction.

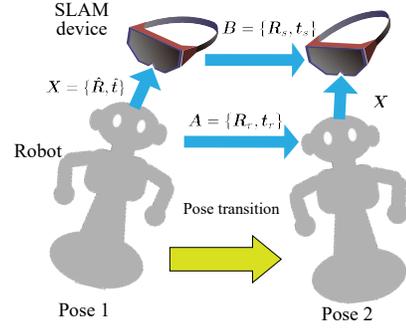


Fig. 6. Motion-based sensor calibration

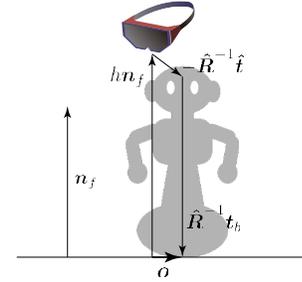


Fig. 7. Robot calibration using the height of a SLAM-device

C. Implementation

1) *Calibration by bi-directional rotations*: As shown in Fig. 4, we can obtain t_x , t_y , r_x , and r_y by a horizontal rotation. We can obtain the remaining t_z and r_z by rotating the device in the vertical direction. If the horizontal rotation and vertical rotation are performed at least once, then 6-DoF calibration parameters are obtained.

2) *Calibration by horizontal rotation and translation*: In the case of a robot without the capability for vertical rotation, we use the relations between SLAM device, robot, and the floor. We obtain the calibration parameters other than the translation in t_z as shown in Fig. 4 via a horizontal rotation and a translation. Specifically, we determine the remaining parameter r_z .

We use a perpendicular from the SLAM device to the floor. As shown in Fig. 7, we assume that \mathbf{n}_f is the surface normal of the floor and h is the height of the SLAM device from the floor. We assume that \mathbf{n}_f and h are given by the environmental map generated by the SLAM device. The difference vector \mathbf{o} between the foot of the perpendicular from the SLAM device and foot of the robot is described as follows:

$$\mathbf{o} = h\mathbf{n}_f - \hat{\mathbf{R}}^{-1}\hat{\mathbf{t}} + \hat{\mathbf{R}}^{-1}\mathbf{t}_b. \quad (14)$$

Thus, \mathbf{o} must be parallel to the floor and perpendicular to \mathbf{n}_f . Therefore, the following equation holds,

$$\mathbf{n}_f \cdot \mathbf{o} = \mathbf{n}_f \cdot (h\mathbf{n}_f - \hat{\mathbf{R}}^{-1}\hat{\mathbf{t}} + \hat{\mathbf{R}}^{-1}\mathbf{t}_b) = 0. \quad (15)$$

The remaining translation parameter in the vertical direction is given by solving Eq. 15.

TABLE I
OBTAINED EXTRINSIC PARAMETER

		x (m)	y (m)	z (m)	angle(rad)	rx	ry	rz
bi-directional rotation	mean	0.083285	0.031271	0.129084	1.66594	0.511814	-0.49433	-0.70262
	standard deviation	0.000531	0.000906	0.00169	0.000391	0.000877	0.000786	0.000232
horizontal movement	mean	0.101651	0.031098	0.11713	1.642449	0.52515	-0.47133	-0.70838
	standard deviation	0.004845	0.002489	0.002671	0.010782	0.015054	0.00668	0.007034

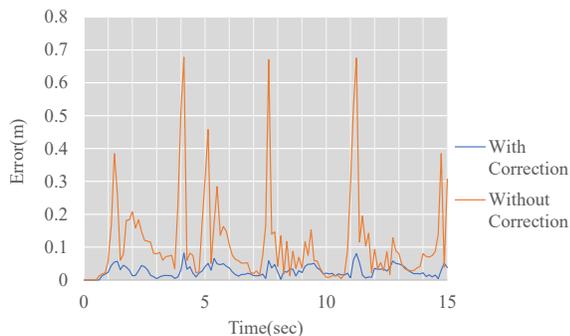


Fig. 8. Localized position error during motion swinging Pepper head wearing HoloLens drastically with the adjustment (Blue line) and without it (Orange line).

IV. EXPERIMENTAL RESULTS

We use SoftBank Pepper¹ as a robot and Microsoft HoloLens² as a SLAM device. The Robot Operating System (ROS)³ assumes the role of a host system. HoloLens is attached to the head of Pepper as shown in Fig. 1 (a). First, we evaluate the accuracy of the calibration parameters obtained by the two methods explained in Sec. III. Next, we evaluate the online correction method by plotting the position error when the joint of the robot moves frequently. Finally, we demonstrate a robot navigation system in an indoor scene using SLAM device.

A. Offline Calibration

We first plot the robot-device calibration parameters via the proposed offline calibration methods with two-directional rotations and horizontal rotation/translation. The two-directional rotation method is performed via swinging the neck twice for each in the horizontal and vertical directions. The horizontal method attempts forward translation and horizontal rotation twice each at five different positions for different conditions for the floor recognition. Table I shows the average value and standard deviation of the estimated parameters when each calibration method is performed five times. The standard deviation of translation parameters by the two-way rotation method is less than 2 mm and that of the rotation angle is less than 4.0×10^{-4} rad. The performance of two-directional rotation method exceeds that

of the method using horizontal rotation/translation. However, the accuracies of both methods are sufficient for navigation.

B. Online Adjustment

We demonstrate the effectiveness of the online device-body calibration method. We move the robots joints to create lags in actuator control without locomotion of the robot. Subsequently, we record the position of the robots foot in $x - y$ coordinate in the case with and without calibration. We can observe the positioning errors due to the time lag although the foot position should not change during the movements of the joints.

Figure 8 shows the error of the online device-body calibration. The error is calculated as the distance from the initial position. We use the two-directional rotation method for the offline calibration. We observe a maximum error exceeding 60 cm in the case without correction. Conversely, the error of the proposed method is less than 10 cm. The result indicates that the proposed sensor-body calibration method absorbs errors for accurate localization of the robot.

C. Navigation

We use a host system for managing the map and navigation. The procedures of navigation are as follows:

- 1) Attach HoloLens to the head of Pepper
- 2) Establish communication with the host system, Pepper, and HoloLens
- 3) Calibrate Pepper and HoloLens
- 4) Run localization and navigation program

We used two-directional rotation method for offline calibration (Fig. 1 (a)). The localization program locates the HoloLens in the global coordinate system via aligning the map provided by the HoloLens to a pre-built 2D map with an initial position that is given manually. The navigation program provides a destination on the 2D map and creates a route plan via implementing the Dijkstra method in ROSs NavFn⁴ package with a cost map [24]. We use the dynamic window approach [25] for local route planning.

Figure 9 shows a sequence of snapshots where the Pepper walked through a long corridor and entered into a room. A movie of the navigation scene is available on <http://www.cvl.iis.u-tokyo.ac.jp/~ishikawa/video/ARobotNav.mp4>

¹<https://www.softbank.jp/en/robot/>

²<https://www.microsoft.com/hololens>

³<http://www.ros.org/>

⁴<http://wiki.ros.org/navfn>

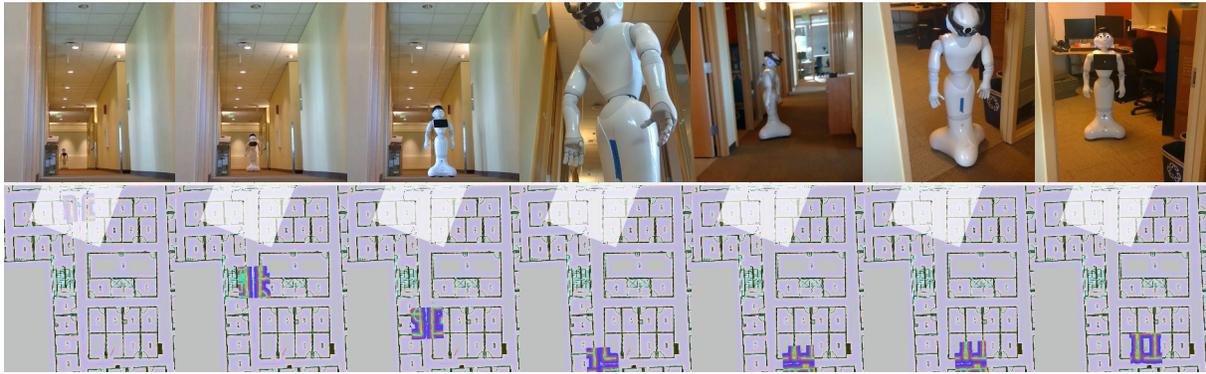


Fig. 9. Navigation with our system using external SLAM device. Lower sequence is floor map in GUI. First, Pepper walks along the corridor and then turns at the corner. Finally, Pepper enters a room and reaches the destination.

V. CONCLUSION

In this paper, we proposed a navigation system with external SLAM device with offline calibration and online localization adjustment. Although a humanoid type robot attached with a SLAM device is used for the demonstration, it is possible to easily apply the proposed method to various robots with a self-position estimation function. Although we used pre-built 2D floor maps created in advance, The system can be extended to navigate without a pre-built map and three-dimensional navigation via a 3D map provided by a SLAM device.

REFERENCES

- [1] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," *Journal of Intelligent and Robotic systems*, vol. 18, no. 3, pp. 249–275, 1997.
- [2] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *AAAI/IAAI*, vol. 1999, no. 343-349, pp. 2–2, 1999.
- [3] A. Hornung, S. Oßwald, D. Maier, and M. Bennewitz, "Monte carlo localization for humanoid robot navigation in complex indoor environments," *International Journal of Humanoid Robotics*, vol. 11, no. 02, p. 1441002, 2014.
- [4] J. Ido, Y. Shimizu, Y. Matsumoto, and T. Ogasawara, "Indoor navigation for a humanoid robot using a view sequence," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 315–325, 2009.
- [5] W. Winterhalter, F. Fleckenstein, B. Steder, L. Spinello, and W. Burgard, "Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 3138–3143.
- [6] L. George and A. Mazel, "Humanoid robot indoor navigation based on 2d bar codes: Application to the nao robot," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 329–335.
- [7] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele, "Real-time image-based 6-dof localization in large-scale environments," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1043–1050.
- [8] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.
- [9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [10] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.
- [11] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 393–398.
- [12] Y. Misono, Y. Goto, Y. Tarutoko, K. Kobayashi, and K. Watanabe, "Development of laser rangefinder-based SLAM algorithm for mobile robot navigation," in *SICE, 2007 Annual Conference*. IEEE, 2007, pp. 392–396.
- [13] G. Klančar, L. Teslić, and I. Škrjanc, "Mobile-robot pose estimation and environment mapping using an extended Kalman filter," *International Journal of Systems Science*, vol. 45, no. 12, pp. 2603–2618, 2014.
- [14] A. Oliver, S. Kang, B. C. Wünsche, and B. MacDonald, "Using the Kinect as a navigation sensor for mobile robotics," in *Proceedings of the 27th conference on image and vision computing New Zealand*. ACM, 2012, pp. 509–514.
- [15] S. Wang, Y. Li, Y. Sun, X. Li, N. Sun, X. Zhang, and N. Yu, "A localization and navigation method with ORB-SLAM for indoor service mobile robots," in *Real-time Computing and Robotics (RCAR), IEEE International Conference on*. IEEE, 2016, pp. 443–447.
- [16] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX=XB$," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.
- [17] F. C. Park and B. J. Martin, "Robot sensor calibration: solving $AX=XB$ on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.
- [18] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation $AX=XB$," *Journal of Field Robotics*, vol. 22, no. 9, pp. 497–506, 2005.
- [19] J. Kelly and G. S. Sukhatme, "Fast relative pose calibration for visual and inertial sensors," in *Experimental Robotics*. Springer, 2009, pp. 515–524.
- [20] J. D. Hol, T. B. Schön, and F. Gustafsson, "Modeling and calibration of inertial and vision sensors," *The international journal of robotics research*, vol. 29, no. 2-3, pp. 231–244, 2010.
- [21] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1793–1800.
- [22] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, 2016.
- [23] R. Ishikawa, T. Oishi, and K. Katsushi, "LiDAR and Camera Calibration using Motions Estimated by Sensor Fusion Odometry," in *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*, Oct 2018, pp. 7342–7349.
- [24] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 709–715.
- [25] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.