The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Mobile Robot Self-Localization Based on Tracked Scale and Rotation Invariant Feature Points by Using an Omnidirectional Camera

Tsuyoshi Tasaki, Seiji Tokura, Takafumi Sonoura, Fumio Ozaki and Nobuto Matsuhira

Abstract—**Self-localization is important for mobile robots in order to move accurately, and many works use an omnidirectional camera for self-localization. However, it is difficult to realize fast and accurate self-localization by using only one omnidirectional camera without any calibration. For its realization, we use "tracked scale and rotation invariant feature points" that are regarded as landmarks. These landmarks can be tracked and do not change for a "long" time. In a landmark selection phase, robots detect the feature points by using both a fast tracking method and a slow "Speed Up Robust Features (SURF)" method. After detection, robots select landmarks from among detected feature points by using Support Vector Machine (SVM) trained by feature vectors based on observation positions. In a self-localization phase, robots detect landmarks while switching detection methods dynamically based on a tracking error criterion that is calculated easily even in the uncalibrated omnidirectional image. We performed experiments in an approximately 10 [m] x 10 [m] mock supermarket by using a navigation robot ApriTau™ that had an omnidirectional camera on its top. The results showed that ApriTau™ could localize 2.9 times faster and 4.2 times more accurately by using the developed method than by using only the SURF method. The results also showed that ApriTau™ could arrive at a goal within a 3 [cm] error from various initial positions at the mock supermarket.**

I. Introduction

At airports or big supermarkets, robots are expected to navigate people or convey baggage. For navigation, it is important for robots to localize their own position. Many works deal with self-localization problems. Recently, most of them use a laser range finder (LRF). They realize self-localization, matching a map to data measured using the LRF [1]. Self-localization methods based on the LRF data can be performed fast and accurately. However, when robots using the methods do not know their initial position at all, errors occur at airports or big supermarkets. The errors occur because there are many similarly shaped objects such as poles and shelves.

On the other hand, many works use an omnidirectional camera. The omnidirectional camera can get not only shape information but also texture information in a wide area around a robot. Therefore, even if there are many similarly shaped objects and a robot does not know its initial position at all, the self-localization succeeds by using the omnidirectional camera.

Some works based on the omnidirectional camera calibrate its mirror parameters [2]. After changing omnidirectional images to non-distorted images, the works apply traditional methods based on general cameras that do not have a wide field of view. These works have an advantage in that they can use many traditional methods. However, it is troublesome to calibrate mirror parameters. Some works [3] use scale and rotation invariant (Local Invariant: LI) feature points like SIFT [4] or SURF [5] without calibrating mirror parameters. Even if the robot moves and omnidirectional images change, using LI feature points enables the robot to localize its position. However, it performs more slowly and some feature points cannot always be used in the omnidirectional image. Some works dealing with SLAM [6][7] use tracking methods [8] or select feature points that have small errors while making a map [9] in order to perform fast. However, few works related to SLAM focus on feature points that can be tracked for a long time. Therefore, in the case of the previous works, there are many feature points that are not detected from various positions in an area where the robot moves.

We deal with two problems. The first is the fast self-localization of a robot using an uncalibrated omnidirectional camera. The second is selecting landmarks that are detected easily from various positions. Our aim is to have a robot arrive at goals accurately from various positions at a supermarket that has many similarly shaped objects. Additionally, we confirm that the developed method can enhance the traditional method based on the LRF.

In order to solve the problems, we focus on the existence of many LI feature points that can be tracked and do not change their feature vectors for a long time. We regard the tracked LI feature points as candidates of new landmarks. In a landmark selection phase, we select landmarks from among the tracked LI feature points by using Support Vector Machine (SVM). In a self-localization phase, our robot detects landmarks, switching detection methods dynamically based on a tracking error criterion. The criterion is calculated easily by directions of detected tracked LI feature points even in the uncalibrated omnidirectional image. Because the tracked LI feature points

Tsuyoshi Tasaki is with the Corporate Research & Development Center, TOSHIBA CORPORATION, Japan , mail: tsuyoshi.tasaki@toshiba.co.jp
Seiji Tokura is with the Corporate Research & Development Center, TOSHIBA CORPORATION, Japan , mail: seiji.tokura@toshiba.co.jp
Takafumi Sonoura is with the Corporate Research & Development Center, TOSHIBA CORPORATION, Japan , mail: takafumi.sonoura@toshiba.co.jp
Fumio Ozaki is with the Corporate Research & Development Center, TOSHIBA CORPORATION, Japan , mail: fumio.ozaki@toshiba.co.jp
Nobuto Matsuhira is with the Corporate Research & Development Center, TOSHIBA CORPORATION, Japan , mail: nobuto.matsuhira@toshiba.co.jp

can be tracked for a long time and do not change their feature vectors very much, our robot can detect them from various positions. Moreover, our robot can continue to detect them fast and localize its position even if it does not know their initial position at all.

Section II defines tracked LI feature points and describes how the points are selected. Section III describes how our robot localizes its position by using tracked LI feature points. In Section IV, we confirm the ability of our self-localization method at the mock supermarket. Section V concludes this paper.

II. Landmark Selection from among Tracked Scale and Rotation Invariant Feature Points

A. Detection of Tracked LI Feature Points

The SURF method is used for detecting LI feature points. The SURF method performs faster than the SIFT method does. However, not all points detected by the SURF method can be tracked. For example, the SURF method can detect feature points at the center of wall and floor. The tracking method is not good at tracking such points. Moreover, some LI feature points change their feature vectors in the uncalibrated distorted omnidirectional image.

In order to avoid using LI feature points that are not tracked easily and change their feature vectors much, our robot moves along whole paths in the landmark selection phase. Our robot can move under ideal environments before supermarkets open. That is, we can assume our robot has enough time, knows the initial position, localizes its position using LRF and there are no people. While moving, our robot detects tracked LI feature points while capturing continuous images. The top K LI feature points satisfying (1) and (2) for a long time are regarded as the tracked LI feature points.

$$\left|\mathbf{F}_G^{(t)} - \mathbf{F}_g^{(0)}\right| < T_{\mathbf{F}} \tag{1}$$

$$\left|\mathbf{x}_G^{(t)} - \mathbf{x}_{g'}^{(t)}\right| < T_{\mathbf{x}} \tag{2}$$

Here, g denotes one LI feature point. The robot started to move at time 0, G denotes one LI feature point detected at time t, and g' denotes a point obtained by tracking g until t. $\mathbf{F}_i^{(t)}$ denotes a feature vector calculated by a SURF 64-dimensional feature vector of a point i at t. The members of the feature vector $\mathbf{F}_i^{(t)}$ are normalized. A 2-dimensional vector $\mathbf{x}_i^{(t)}$ denotes the position of a point i at t on image coordinates. An origin of image coordinates is defined as a center of the image. $T_{\mathbf{F}}$ and $T_{\mathbf{x}}$ are constant thresholds. Equation (1) enables the robot to detect an LI feature point that does not change the SURF feature vector very much while moving. Equation (2) enables the robot to detect the LI feature point close to a point obtained by tracking. When the tracking succeeds, the position of G is close to the position of g' in image coordinates. Therefore, it is easy to track g that satisfies both (1) and (2) for a long time and to detect it from various positions. The feature vectors of g do not change very much even in the distorted images.

B. Landmark Selection Based on Observation Positions

Tracked LI feature points detected by (1) and (2) are candidates of the landmark. Our robot regards tracked LI feature points whose positions on world coordinates are measured correctly as landmarks. Here, measuring correctly means that a distance between a measured position and correct position is less than a threshold T_{d}.

In order to measure a position (X_m, Y_m) of one tracked LI feature point m on world coordinates, m is measured from various positions. The robot can memorize an observation position (X_{A1}, Y_{A1}) and a posture θ_{A1} on world coordinates because the robot can localize its position under ideal environments in the landmark selection phase. From one position by using only one omnidirectional camera, the robot cannot measure (X_m, Y_m), but can measure a direction θ_{I1} from the robot to m. When m cannot be tracked at t and similar tracked LI feature point m' is detected after t, (1) is used in order to judge whether those two points are the same or not.

In order to measure (X_m, Y_m), a straight line L_{A1} passing through (X_m, Y_m) and (X_{A1}, Y_{A1}) is calculated by (X_{A1}, Y_{A1}), θ_{A1} and θ_{I1}. When m is observed from various observation positions (X_{A2}, Y_{A2}), (X_{A3}, Y_{A3}), …, (X_{An}, Y_{An}), the lines L_{A2}, L_{A3}, …, L_{An} are also calculated. The point m is at the intersection of lines. However, not all lines intersect because observations often include errors. When the robot observes m from (X_{A1}, Y_{A1}), (X_{A2}, Y_{A2}), …, (X_{An}, Y_{An}), the intersection is calculated by the least square error method as shown by (3). Here, $\mathbf{M}^+$ denotes a para-inverse matrix of matrix $\mathbf{M}$.

$$\begin{pmatrix} X_m \\ Y_m \end{pmatrix} = \begin{pmatrix} \tan(\theta_{A1}+\theta_{I1}) & -1 \\ \tan(\theta_{A2}+\theta_{I2}) & -1 \\ \vdots & \vdots \\ \tan(\theta_{An}+\theta_{In}) & -1 \end{pmatrix}^{+} \begin{pmatrix} \tan(\theta_{A1}+\theta_{I1})\cdot X_{A1} - Y_{A1} \\ \tan(\theta_{A2}+\theta_{I2})\cdot X_{A2} - Y_{A2} \\ \vdots \\ \tan(\theta_{An}+\theta_{In})\cdot X_{An} - Y_{An} \end{pmatrix} \tag{3}$$

An example of a measured landmark is shown in Fig. 1. Fig. 1 shows 4 omnidirectional images taken from 4 observation positions. The landmark is the center point of a dotted line circle in each image. The same place is detected as the

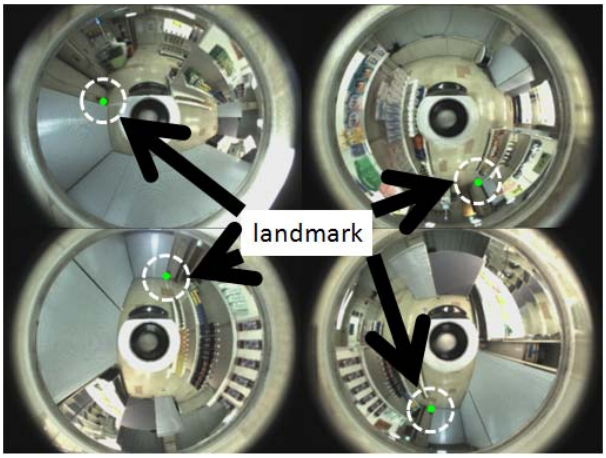

Fig. 1. A landmark from various observation positions captured by an omnidirectional camera.

landmark from various observation positions.

(X_m, Y_m) is calculated accurately by the least square method when the same landmark is detected accurately from various observation positions as seen in Fig. 1. However, the calculation sometimes fails. For example, when there are many similar things and the robot regards different objects as the same, the robot cannot calculate the position of the intersection. Additionally, when observations do not fail but the observation positions are close to each other, the intersection cannot be calculated.

Considering a relationship between observation positions and the accuracy of the landmark measurement, tracked LI feature points that may fail to be measured are deleted. Tracked LI feature points that can be measured correctly are regarded as landmarks. Here, we use SVM in order to classify the points as landmarks or not.

6-dimensional vector is used as a feature vector for SVM. The feature vector consists of 6 feature quantities as follows:

1. the number of observations
2. the least square error of the measurement [rad^2]
3. the average distance from the robot to the points in the image [pixel]
4. the distribution of the observation positions' x-coordinate [m^2]
5. the distribution of the observation positions' y-coordinate [m^2]
6. the distribution of the directions from the robot to the points [rad^2]

These 6 feature quantities relate to the observation positions and postures. At the end of the landmark selection phase, the feature vectors of all tracked LI feature points are calculated. The vectors are calculated by the result of the measurements and the information of the observation positions. SVM that has already been trained by training data classifies the points as landmarks or not. In order to make the training data, we first make the robot move to another place and obtain N landmark candidates. Next, we manually label the candidates as landmarks or not by comparing measured positions with correct positions of landmarks. This manual step can be omitted once SVM has been trained.

III. Self-localization Based on Tracked Scale and Rotation Invariant Feature Points

A. A Tracking Error Criterion

While the robot moves, it localizes its position while taking an omnidirectional image continuously. The landmark is detected from the continuous images. The position and posture of the robot can be calculated by detecting more than 3 landmarks. In order to realize the fast self-localization, once the robot detects landmarks by using the SURF method that performs slowly, the landmarks are tracked fast continuously. It is easy to track the landmarks for a long time because the landmarks are tracked LI feature points. However, tracking landmarks will certainly fail. For example, when people move between the robot and the landmarks, targets of the tracking method change. As long as the robot does not know that the targets change, self-localization fails and the robot continues to track different targets.

We propose a tracking error criterion calculated by the directions to the tracked landmarks in the uncalibrated omnidirectional image. Our robot stops to track the landmarks that have a high tracking error criterion. Stopping tracking minimizes the self-localization error.

The tracking error criterion E is defined as (4), (5) and (6). Here, (x_i, y_i) and $(x_{i'}, y_{i'})$ denote the position on world coordinates and image coordinates of landmark i, respectively. (X, Y, Θ) denotes the robot's position and posture on world coordinates estimated by using more than 3 landmarks.

$$E = \left| \alpha_{i'} - \beta_i \right| \tag{4}$$

$$\alpha_{i'} = \mathrm{Tan}^{-1} \frac{y_{i'}}{x_{i'}} \tag{5}$$

$$\beta_i = \mathrm{Tan}^{-1} \frac{y_i - Y}{x_i - X} - \Theta \tag{6}$$

$\alpha_{i'}$ denotes the direction to the landmark i that can be measured by an omnidirectional image. β_i denotes the direction to the landmark i that can be calculated on world coordinates. Fig. 2 shows a relationship between $\alpha_{i'}$ and β_i. As shown in Fig. 2, when both self-localization and detecting landmarks succeed, $\alpha_{i'}$ is equal to β_i. When $\alpha_{i'}$ is equal to β_i, the tracking error criterion E is equal to 0 which is the smallest value.

On the other hand, when either self-localization or detecting landmarks fails, E increases. For example, when the tracked target changes from the landmark to the person who walks between the robot and the tracked landmark, $\alpha_{i'}$ differs from β_i gradually.

B. Self-localization System Switching between the SURF Method and the Tracking Method

Our self-localization system is shown in Fig. 3. "Sub Flow" in Fig. 3 denotes a process path that is passed in the case that the tracking error criterion E is high. In "Sub Flow",

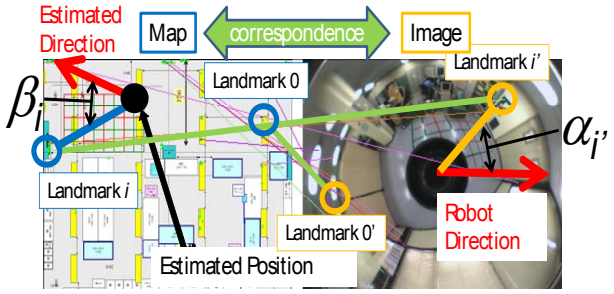

Fig. 2 Relationship between $\alpha_{i'}$ and β_i

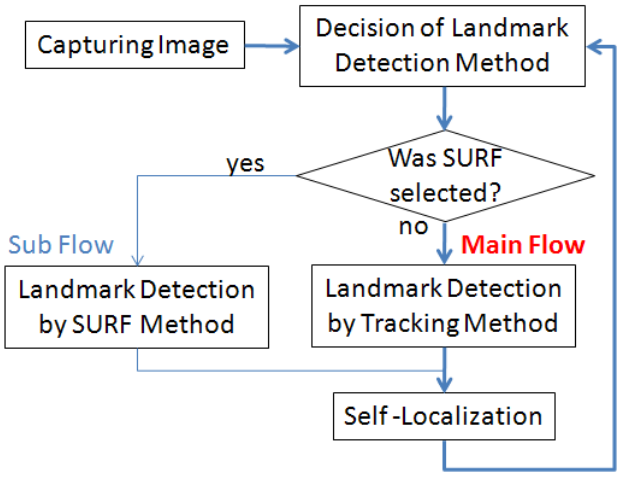

Fig. 3 The developed self-localization system

landmarks are detected slowly by the SURF method. "Main Flow" denotes a process path that is usually passed. In "Main Flow", landmarks are detected fast by the tracking method. The number of passing "Main Flow" is more than that of "Sub Flow". Once our system detects landmarks through "Sub Flow" and our robot localizes its position, our robot localizes its position by using the tracking method through "Main Flow". Our robot also removes the landmark that has the high E while moving. When the number of landmarks is less than a threshold, our robot detects landmarks again by using the SURF method. Details of the self-localization steps are as follows:

1. The robot captures an omnidirectional image.
2. Self-localization is performed through "Sub Flow".
3. The tracking error criteria E of all landmarks are calculated.
4. The landmark whose E is higher than a threshold T_p is removed.
5. If more than T_n landmarks exist in step 4, self-localization is performed through "Main Flow" after the next omnidirectional image is captured. Otherwise, self-localization is performed through "Sub Flow".
6. The robot repeats the process from step 3 to step 5.

IV. Evaluation of a Self-Localization Ability

A. Navigation Robot: *ApriTau*™

Our self-localization system was implemented on our robot called ApriTau™[10] as shown in Fig. 4. ApriTau™ is used for evaluations of self-localization and movement ability. It is 120 [cm] tall and 65 [cm] wide. An omnidirectional camera is mounted on the top of its head and does not move with the head motion, because the camera is fixed on the inside frames. ApriTau™ takes images whose size is 320x240 [pixels] continuously at 15 [fps]. The size is small because we focus on the computational time rather than the accuracy of the self-localization. It can move 1.0 [m/s] using its vehicle. In

Fig. 4 Navigation robot ApriTau™

Fig. 5 An experimental room for a supermarket

the landmark selection phase, ApriTau™ can localize its position by using odometry data and LRF data.

We performed experiments at a mock supermarket as shown in Fig. 5. The size of the mock supermarket is 10 [m] by 10 [m]. There are some shelves, a refrigerator and a self-checkout machine at the mock supermarket.

In these experiments, the thresholds T_F, T_x and the parameter K as shown in section 2.1 are 40, 1.5 and 9, respectively. The threshold T_d as shown in section 2.2 is 60 [cm]. The SVM is trained by 100 tracked LI feature points. The thresholds T_p and T_n are 5 [deg] and 5 points respectively. These thresholds and parameters are decided experimentally, considering the resolution of the image and the size of the mock supermarket.

B. Accuracy and Computational Time Evaluation of Self-Localization System

1) *Setting:*

In order to confirm the effectiveness of the tracked LI feature points, we compare two methods. One is our developed method that switches between the SURF method and the tracking method. The other is a simple method that uses only the SURF method. The experimental steps are as

follows:

1. ApriTau™ moves along whole paths at the experimental room and selects landmarks.
2. ApriTau™ moves along a fixed route for evaluation. In this case, we use a 4 [m] line by a wall for simplicity. ApriTau™ takes continuous omnidirectional images while moving.
3. Both our method and the simple method estimate the positions of ApriTau™ at each time and measure the computational time of processing one omnidirectional image.
4. We use the Euclidean distance between the estimated positions and the route at each time as the error. For evaluation, we use an average (avg.) and a standard deviation (SD) of the error calculated by each method. We also use an avg. and a SD of the computational time. We compare our method with the simple method based on these 4 values.

2) Results and Discussions:

Fig. 6 shows the estimated position of both our method and the simple method. A line by the y-coordinate 2 [m] shows the route of ApriTau™. Squares and triangles denote the estimated positions of our method and the simple method, respectively.

Table 1 shows the avg. and SD of the error and the avg. and SD of the computational time calculated by both methods. As shown in Table 1, the error of our method is 4.2 times less than that of the simple method. The computational time of our method is also 2.9 times faster than that of the simple method.

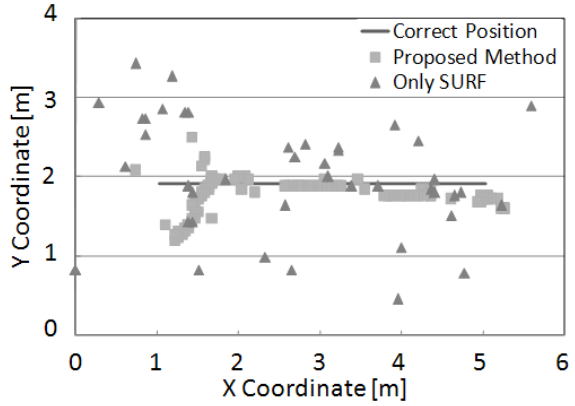

Fig. 6 Localization results of the developed method and the simple method

Table 1 Comparing our method with the simple method for localization errors and computational time

Method	Error Avg.	Error SD	Time Avg.	Time SD
Developed	0.38 [m]	0.21 [m]	16.4 [ms]	4.65 [ms]
Simple	1.61 [m]	1.36 [m]	46.8 [ms]	1.81 [ms]

The computational time of our method is much shorter than that of the simple method, which is confirmed by t-test. However, the computational time SD of our method is longer than that of the simple method. We think the SD is long because the computational time of "Sub Flow" is much longer than that of "Main Flow". In a real supermarket, there are many people. Therefore, we think our system often switches between the SURF method and the tracking method, and our method performs slowly. In future work, we have to confirm around how many people our method can perform fast.

Our method is more accurate, because the robot uses tracked LI feature points that can be detected at various positions and have salient feature vectors. However, as shown in Fig. 6, our method cannot work accurately from 1.0 [m] to 1.5[m] on x-coordinate. This error occurs because a white wall without landmarks exists on an x-coordinate 0 [m] line. If the robot uses only a general camera that has a narrow field of view, it cannot localize its position at all in front of the white wall. On the other hand, ApriTau™ can localize fast with about 60 [cm] errors even in this case, which is an advantage. For example, if the self-localization method using LRFs uses the output of our method as an initial position, the robot can localize its position very accurately.

C. Mobility Evaluation Based on the Self-localization

1) Setting:

In order to confirm that the output of our method enhances the self-localization method using LRFs, we make ApriTau™ move to the fixed goal from various initial positions. ApriTau™ is activated without knowing its position. First, it localizes its position by using our method. Next, it localizes its position more accurately by using the LRF method based on the result of our system. In this experiment, we use ICP [11] as the LRF method. When ApriTau™ used only the LRF method, it could not localize itself accurately because there are many similar shelves at the mock supermarket. For evaluation of the robot movement ability, we use the distance from the fixed goal position to the position where ApriTau™ can arrive. ApriTau™ moves to the goal position (5.55 [m], 3.11 [m]) on world coordinates from various initial positions. The correct positions of the robot can be measured by outside measurement systems. There are no people around the robot at the mock supermarket. ApriTau™ has already selected landmarks.

2) Results and Discussions:

The output route and the distance between the goal and the position where ApriTau™ arrived are shown in Fig. 7 and Table 2, respectively.

As shown in Fig. 7, the outputs of our method in Trial 1 and 2 were modified and finally ApriTau™ arrived at the circle that indicated the goal. The average of distance errors is 3 [cm]. We think the error is small enough for mobile robots to move at a supermarket.

This result shows that using the LRF method with our

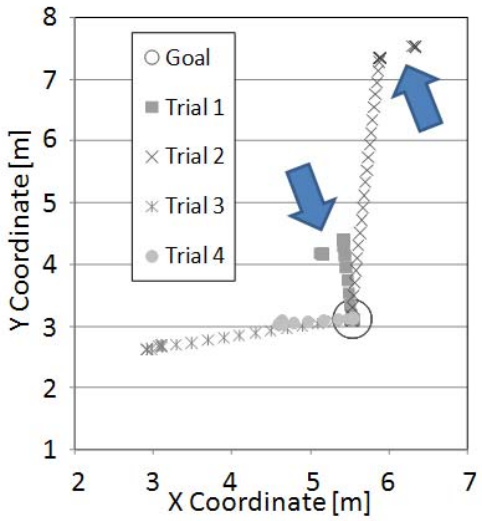

Fig. 7 Routes from various initial positions to the goal

Table 2 Distances between the goal and arrival positions

Trial Number	1	2	3	4
X-coordinate [m]	5.54	5.51	5.60	5.56
Y-coordinate [m]	3.11	3.14	3.15	3.11
Distance [m]	0.01	0.04	0.06	0.01

method is effective, when the robot moves at a place such as a supermarket that has many rectilinear shaped objects with different textures. In this case, the ICP works well especially, even if the robot only knows its approximate position.

ApriTau^{TM} selects signs that show what kinds of goods the shelves have as landmarks. The signs are effective because robot can detect signs from various positions and they are easy to identify. Other landmarks such as a pole or a watch can be also detected from various positions. This experiment shows that the landmarks based on the tracked LI feature points can be taken from various positions and enhance the robot movement ability.

V. Conclusion

We deal with two problems. The first is the fast self-localization of a robot using an uncalibrated omnidirectional camera. The second is selecting landmarks that are detected easily from various positions. In order to solve these problems, we developed the method whereby the robot selected landmarks based on the tracked LI feature points. The tracked LI feature points are defined as the feature points that can be detected by both the slow SURF method and the fast tracking method. The robot selects landmarks from among the tracked LI feature points that can be measured accurately. SVM judges whether the position of points can be measured accurately or not based on the observation positions. Moreover, in the self-localization phase, we developed the fast self-localization method that switched between the SURF method and the tracking method based on a new criterion. The criterion is called the tracking error criterion. The criterion is calculated by the direction of landmarks that can be calculated even in the uncalibrated omnidirectional image.

In order to confirm the ability of the developed self-localization method, we implemented the method into the navigation robot ApriTau^{TM} and performed two experiments. The first experiment showed that ApriTau^{TM} could localize 2.9 times faster and 4.2 times more accurately by using our method than by using only the SURF method. The second experiment showed that ApriTau^{TM} could arrive at a goal within a 3 [cm] error from various initial positions at the mock supermarket. ApriTau^{TM} could not localize its position by using only the LRF self-localization method at the mock supermarket. In future work, we intend to confirm the effectiveness of our method in a crowded place such as a real supermarket.

Acknowledgment

This research was supported by New Energy and Industrial Technology Development Organization (NEDO, Japan) Project for Strategic Development of Advanced Robotics Elemental Technologies, Conveyance Robot System in the Area of Service Robots, Robotic Transportation System for Commercial Facilities.

References

[1] M. Tomono, “Efficient Global Scan Matching Using Saliency-based Scan Point Resampling,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1433–1438, 2005.

[2] R. Bunschoten and B. Krose, “Robust Scene Reconstruction From an Omnidirectional Vision System,” IEEE Trans. Robotics and automation, vol. 19, no. 2 pp. 351–357, 2003.

[3] A. C. Murillo, J. J. Guerrero and C. Sagues, “SURF features for efficient robot localization with omnidirectional images”, IEEE International Conference on Robotics and Automation, pp. 3901 - 3907, 2007.

[4] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, 60(2), pp. 91 - 110, 2004.

[5] H. Bay, T. Tuytelaars, and L. V. Gool: “Surf: Speed up robust features”, in the ninth European Confernce on Computer Vision 2006.

[6] A. Davison, L. Reid, N. Molton, O. Stasse, ”MonoSLAM: Real-time single camera SLAM,” IEEE Trans. Pattern Analysis and Machine Intelligence 29, pp.1052-1067, 2007.

[7] S. Ahn, W. K. Chung and S. R. Oh, “Construction of Hybrid Visual Map for Indoor SLAM”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1695 – 1701, 2007.

[8] B. D. Lucas and T. Kanade, “An Interative Image Registration Technique with an Applicatino to Stereo Vision”, Proceeding of 7th International Joint Conference on Artificial Intelligence, pp. 674 - 679, 1981.

[9] S. Hochdorfer and C. Schlegel, “Landmark rating and selection according to localization coverage: Addressing the challenge of lifelong operation of SLAM in service robots,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 382–387, 2009

[10] S. Tokura, T. Tasaki, T. Sonoura, M. Sano, N. Matsuhira, K. Komoriya, “Robotic Transportation System for Shopping Support Services”, 18th IEEE International Symposium on Robot and Human Interactive Communication, pp. 320 - 320, 2009.

[11] P. J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256, 1992.