J Intell Robot Syst (2012) 67:7–24
DOI 10.1007/s10846-011-9647-4

Vision-Based Kidnap Recovery with SLAM for Home Cleaning Robots

Seongsoo Lee · Sukhan Lee · Seungmin Baek

Received: 1 June 2011 / Accepted: 28 November 2011 / Published online: 15 December 2011

Abstract Emerged as salient in the recent home appliance consumer market is a new generation of home cleaning robot featuring the capability of Simultaneous Localization and Mapping (SLAM). SLAM allows a cleaning robot not only to self-optimize its work paths for efficiency but also to self-recover from kidnappings for user convenience. By kidnapping, we mean that a robot is displaced, in the middle of cleaning, without its SLAM aware of where it moves to. This paper presents a vision-based kidnap recovery with SLAM for home cleaning robots, the first of its kind, using a wheel drop switch and an upward-looking camera for low-cost applications. In particular, a camera with a wide-angle lens is adopted for a kidnapped robot to be able to recover its pose on a global map with only a single image. First, the kidnapping situation is effectively detected based on a wheel drop switch. Then, for an efficient kidnap recovery, a coarse-to-fine approach to matching the image features detected with those associated with a large number of robot poses or nodes, built as a map in graph representation, is adopted. The pose ambiguity, e.g., due to symmetry is taken care of, if any. The final robot pose is obtained with high accuracy from the fine level of the coarse-to-fine hierarchy by fusing poses estimated from a chosen set of matching nodes. The proposed method was implemented as an embedded system with an ARM11 processor on a real commercial home cleaning robot and tested extensively. Experimental results show that the proposed method works well even in the situation in which the cleaning robot is suddenly kidnapped during the map building process.

Keywords Home cleaning robot · Kidnap recovery · SLAM · Global localization

S. Lee · S. Lee (✉)
School of Information and Communication Engineering and Department of Interaction Science, Sungkyunkwan University, Suwon, South Korea
e-mail: lsh@ece.skku.ac.kr

S. Lee
e-mail: seongsu.lee@lge.com

S. Lee · S. Baek
Future IT Laboratory, LG Electronics Inc., Seoul, South Korea
e-mail: seungmin2.baek@lge.com

1 Introduction

Home cleaning robots are the most successful consumer robot introduced over the past few years. However, the conventional home cleaning robots frequently become stuck on something in a manner such that a wheel is off the ground during navigation. When this happens, users usually relocate the robot so that it can continue to perform the cleaning task that it was given before

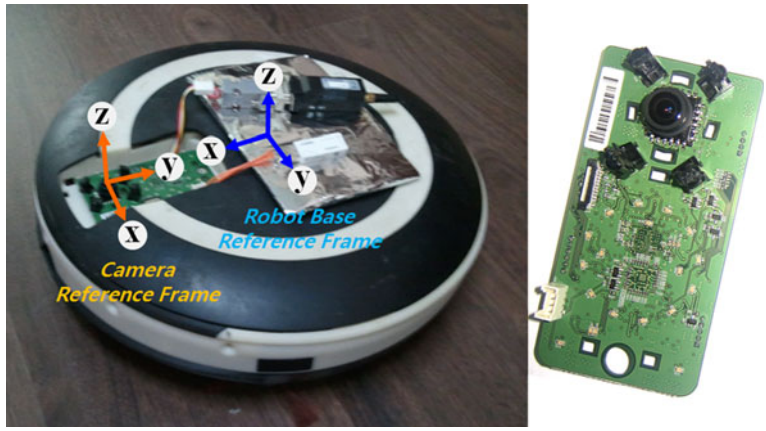

Fig. 1 Home cleaning robot and our embedded vision board (including a camera, an ARM11 processor, and 64 MB RAM)

it got stuck. Unfortunately, the conventional home cleaning robots cannot resume the previous cleaning task in this situation, because they cannot recover from the localization error without any odometric feedback about the actual motion. In this situation, users can only command the robot to start a new cleaning because the robot will erase the previous cleaning area in its memory. The more this situation is repeated, the more increased an unnecessary cleaning time becomes. In robotics, this situation is referred to as the "*kidnapped robot problem*" and is a variant of the global localization problem [6]. The kidnapped robot problem is more challenging as the robot is typically confused.

Simultaneous Localization and Mapping (SLAM) [1, 15] can be used to estimate a home cleaning robot pose within an incrementally built map and concurrently remember a cleaning area. However, most SLAM algorithms cannot correct the significant localization error such as a kidnapping situation. In order to solve this problem, we propose a practical global localization algorithm that efficiently solves kidnapping problems in home cleaning robots. The original contribution of this work is in addressing major problems that must be solved for developing a reliable vision-based global localization system, as follows:

- Detecting a kidnapping situation.
- Matching features extracted from the current image with map elements composed of numerous features.
- Removing the pose ambiguity caused by matching error.
- Fusing the uncertain multiple hypotheses about the current robot pose obtained from the matching results.

The organization of this paper is as follows. First, we describe related works in Section 2 and a brief review of our vision-based SLAM system in Section 3. Second, we propose a kidnapping detection method using a wheel drop switch in Section 4 and a novel localization algorithm based on the matching process at multiple levels of resolution that takes advantage of the use of an upward-looking camera with a wide-angle lens in Section 5. Third, we apply the proposed global localization algorithm to an embedded system with an ARM11 processor of a home cleaning robot (See Fig. 1), and demonstrate the validity of the proposed method through indoor experiments in Section 6. Finally, we conclude this paper in Section 7.

2 Related Works

Self-localization addresses the problem of determining a robot pose relative to a given map of the environment [9]. This map can be built by SLAM [1]. Such a setup is feasible and could be useful in many applications. Self-localization is essential for most applications, because safe navigation (including path planning) requires knowledge of the current pose of the mobile robot. This problem

has received considerable attention over the last decades [20–24]. Depending on the types of information available initially and at runtime, the localization problem can be characterized as local or global. Local techniques aim to update a robot pose by using its current sensor data as well as sensor data that have already been accumulated. One popular approach to robot localization is the use of the Kalman filter [25]. Kalman filter based approaches are computationally efficient, but are only suitable for pose tacking because they require bounding of the initial localization error [26, 27]. Localization on a local level is mandatory for algorithms that address the SLAM problem.

In contrast to local techniques, global techniques are capable of localizing a robot without prior knowledge about its pose. One approach for the global techniques based on visual data is to find certain easily identifiable objects in an environment when one or more of those objects are in the camera's field of view. When the robot sees any of these landmarks, the perceived locations of the landmarks can be compared to their known locations within the environment, and the robot pose can be deduced. The landmarks can be either artificial or natural.

A key problem in vision-based global localization concerns the matching of features extracted from an image with map elements without prior knowledge of the robot pose. Solving this matching problem becomes more challenging when the map contains a very large number of landmarks. Recent studies have substantially improved the matching process for localization by using invariant local descriptors [6, 7]. In [6], a global localization algorithm is extended by proposing a submapping strategy that relies on highly specific Scale Invariant Feature Transform (SIFT) features [5] to match a local submap to a pre-built SIFT database map. Three-dimensional submaps are built by aligning multiple frames while the global map is built by aligning and merging multiple 3D submaps. The SIFT features help to preselect potential matches in order to reduce the computational complexity of the matching process. In [7], the Speeded-Up Robust Features (SURF) method [2] is applied in feature matching between omni-directional images. The camera pose is computed from two images that have their location information determined by a triangulation method. The experimental results demonstrate that the SURF is better in terms of accuracy and computational costs when compared with the SIFT.

Blob-like features generated by the SIFT or SURF methods can change dramatically according to the lighting conditions. In addition, feature matching with SIFT or SURF descriptors may yield comparatively high false-positive rates with a significant change in the illumination. In order to overcome these problems, some researchers empirically demonstrated the robustness of their proposed localization method under various outdoor lighting conditions using both monocular camera images and 3D laser point cloud data [12]. The main advantage of this method over previous approaches is the ability to autonomously generate a 3D edge map.

However, the aforementioned approaches to global localization are not suitable for low-cost applications such as home cleaning robots because they use relatively expensive sensors (such as laser range finders, stereo cameras, or omni-directional cameras), or require a high-performance processor that cannot be easily applied in low-cost applications.

In particular, schemes based on cameras pointing straight up at the ceiling, [30–33], have several advantages compared to other schemes in indoor environments. First, there is no significant effect caused by dynamic obstacles such as moving people since the camera points at the ceiling. In addition, there is no scale change for features between neighboring images in most indoor environments. Therefore, under the aforementioned conditions, visual features can be extracted quickly from an image, while robust matching results are obtained. In [30], a camera pointed to the ceiling is used to match the current image to a large ceiling mosaic covering the whole operational space of the robot. The mosaic has to be constructed in advance, which involves a complex state estimation problem. In [32], a vision-based simultaneous localization and mapping technique is proposed, which uses corner features as landmarks in the scene and localizes and reconstructs 3D corner landmarks at the same time using an extended Kalman filter based framework. With the reconstructed 3D

corner landmarks, the robot pose is globally localized based on the Hough clustering technique but it is not suitable for embedded systems because the feature matching problem in a map including numerous landmarks is not addressed. In [33], a panorama image of the ceiling is generated by using a visual motion between adjacent images and is used to estimate the robot pose. However, it is not designed for an embedded system based low-cost robot applications.

For home cleaning robots as shown in Fig. 1, we have developed a vision-based global localization algorithm using a simple camera that satisfies the small size, light weight, and low energy requirements. The camera points in the zenith direction perpendicular to the ground plane due to the aforementioned advantages, and a wide-angle lens of 160° is also utilized to localize the robot globally with a single image in the matching process.

3 Brief Review of Vision-Based SLAM

In this work, the SLAM algorithm employs a pose graph optimization technique [10, 11, 16–18] that successfully solves large SLAM problems with many loops. There is no privileged pose, and recovering the landmark estimates requires a graph traversal. Each node on a graph represents a robot pose at which a sensor measurement was acquired. The edges in the graph represent the spatial constraints between the nodes. In this paper, the constraint $z_{i,j}$ is obtained using sensor measurements and describes a relative transformation (composed of the 3-dimensional vector $z_{i,j} = [\Delta x_{(z_{i,j})}, \Delta y_{(z_{i,j})}, \Delta\theta_{(z_{i,j})}]$) between two poses $x_i = [x_{(x_i)}, y_{(x_i)}, \theta_{(x_i)}]^T$ and $x_j = [x_{(x_j)}, y_{(x_j)}, \theta_{(x_j)}]^T$ to represent the relative pose of x_j with respect to the pose x_i. Let $(z_{i,j} - x_j \ominus x_i)$ be the error introduced by the constraint $z_{i,j}$, where $\ominus$ is the inverse of a compounding operation $\oplus$ and it is the same operation as defined in [14]. Assuming that the constraints are independent, a SLAM formula that minimizes the sum of the weighted square costs introduced by the error $(z_{i,j} - x_j \ominus x_i)$ is given by:

$$\sum_{i,j} \left(z_{i,j} - x_j \ominus x_i\right)^T W_{i,j} \left(z_{i,j} - x_j \ominus x_i\right), \qquad (1)$$

where $W_{i,j}^{-1}$ is a 3×3 covariance matrix of $z_{i,j}$.

One approach for efficiently solving Eq. 1 is to linearize a constraint $z_{i,j}$ with respect to the orientation $\theta_{(x_i)}$, as described in [11]. The linearized version is solved by a linear solver that handles a linear least-square problem such as $AX{=}B$.

There are two types of constraints to be considered for Eq. 1: *the incremental constraint and the loop closure constraint*. The incremental constraint describes a relative transformation between successive poses which have continuity with the time. The loop closure constraint models a relative transformation between non-consecutive poses.

Algorithm 1 Loop closure constraint calculation

- Input: 2d image features at the pose x_i, 3D feature points at the pose x_j

1. Choose five random pairs.
2. Compute the relative transformation $z_{i,j}$ using a non-linear optimization algorithm that minimizes Eq. 2 by adjusting $z_{i,j}$.
3. Check the number of inliers and their reprojection error.
4. Repeat the line 1 to 3 until n relative transformations for $z_{i,j}$ are obtained.
5. Determine the best hypothesis among the n relative transformations.

In this work, odometry is primarily utilized to obtain the incremental constraint $z_{i,j}$ between successive poses based on an odometry motion model algorithm described in [29]. The loop closure constraint $z_{i,j} = [\Delta x_{(z_{i,j})}, \Delta y_{(z_{i,j})}, \Delta\theta_{(z_{i,j})}]^T$ is obtained based on a non-linear optimization technique given local point features in the image coordinate system of the pose x_i and the corresponding 3D points in the robot base reference frame of the pose x_j. The cost function to be minimized by adjusting the optimized variables $\{\Delta x_{(z_{i,j})}, \Delta y_{(z_{i,j})}, \Delta\theta_{(z_{i,j})}\}$ is given by:

$$\sum_s \left(\underbrace{\psi_{(x_i)}[s] - C_{calib.} T_{Robot}^{Camera} \left(R p_{(x_j)}[s] + T\right)}_{=\varepsilon(s)} \right)^T \times \left(\psi_{(x_i)}[s] - \underbrace{C_{calib.} T_{Robot}^{Camera} \left(R p_{(x_j)}[s] + T\right)}_{=h(s)} \right), \qquad (2)$$

where

$$R = \begin{pmatrix} \cos\left(\Delta\theta_{(z_{i,j})}\right) & -\sin\left(\Delta\theta_{(z_{i,j})}\right) & 0 \\ \sin\left(\Delta\theta_{(z_{i,j})}\right) & \cos\left(\Delta\theta_{(z_{i,j})}\right) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

$$T = \begin{pmatrix} \Delta x_{(z_{i,j})} \\ \Delta y_{(z_{i,j})} \\ 0 \end{pmatrix}, \quad (4)$$

C_{calib} is a known camera calibration matrix, T_{Robot}^{Camera} is a known coordinate transformation from 3D robot base coordinates to 3D camera coordinates, $\psi_{(x_i)}[s]$ is the s^{th} measured feature location $[u, v, 1]^T$ in the image coordinate system of the pose x_i, and $p_{(x_j)}[s]$ is the s^{th} 3D feature point $[x, y, z]^T$ in the robot base reference frame of the pose x_j. Equation 2 is solved by iteratively computing a correction term $\delta_{z_{i,j}}$ using the Gauss-Newton form [28], as follows:

$$\delta_{z_{i,j}} = \left(J^T J\right)^{-1} J^T \varepsilon, \quad (5)$$

where J is the Jacobian matrix $\partial h/\partial z_{i,j}$, and ε is the matrix form of $\varepsilon(s)$. Here, $(J^T J)^{-1}$ obtained in the final iteration process means the 3×3 covariance matrix of the solution and is used as the covariance matrix $W_{i,j}^{-1}$ of $z_{i,j} = [\Delta x_{(z_{i,j})}, \Delta y_{(z_{i,j})}, \Delta\theta_{(z_{i,j})}]^T$. An algorithm for computing the loop closure constraint $z_{i,j}$ is briefly described in Algorithm 1 and a method for feature matching between neighboring images will be presented in Section 5.1. In our tests, good result in terms of the map accuracy were achieved when diagonal terms of the covariance matrix for the loop closure constraint were about 1 to 4 times greater than those for the incremental constraint. This condition was satisfied by heuristically adjusting the odometry motion model parameters described in [29].

In our SLAM system, the 3D locations of features for poses stored in the map are needed for obtaining the loop closure constraint using Algorithm 1. With the matching results between successive images, we initialize a 3D location of each feature using a triangulation method [19] which is popular in computer vision. Odometry is used to predict a camera motion between successive images while capturing images at 30 cm intervals.

In order to reduce the computational complexity of the SLAM process, we introduce two strategies as follows. First, the current pose stored in the map can have a single loop closure constraint. Second, the largest loop closure constraint among multiple loop closure candidates is selected for the current pose. These strategies are necessary to apply a pose graph optimization technique in a low-cost system because the computational complexity is proportional to the number of poses and the number of loop closure constraints.

The details of our SLAM system are not presented in this paper, because the objective of this paper is to focus on proposing a vision-based global localization algorithm that can efficiently recover from kidnapping of the robot.

Figure 2 shows the 3D locations of features reconstructed from successive images in a local

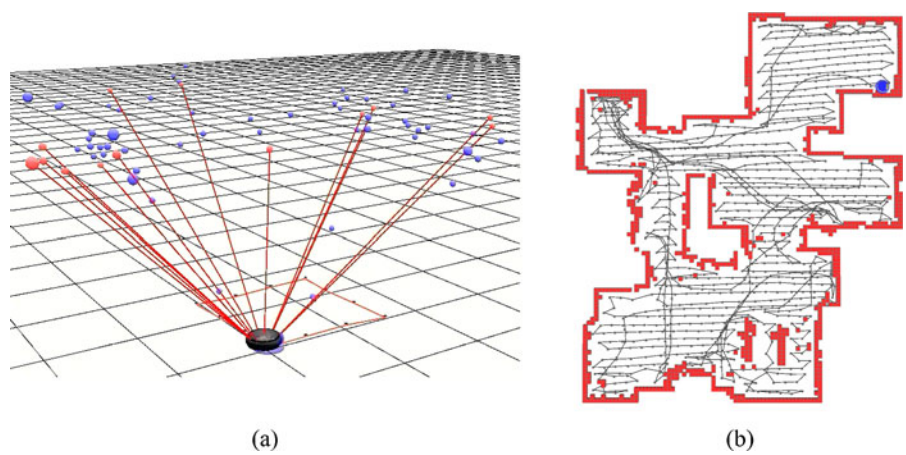

Fig. 2 Experimental results obtained by our SLAM system. **a** 3D visual map. **b** Obstacle grid map constructed following exploration of the experimental environment

area and the obstacle grid map built by our SLAM system in an indoor environment. A red obstacle grid map (See Fig. 2b) superimposed onto the robot's movement trajectory was built with Position Sensitive Device (PSD) sensors that can measure a short-range distance, because it was better than the 3D visual map in depicting an indoor environment. The robot trajectory estimated by our SLAM system is drawn with a black line.

4 Kidnapping Situation Detection

In the kidnapped robot problem, a robot is usually lifted and moved to a different location; therefore, it does not have any odometric feedback on this motion. This problem aims to test the system in situations where the robot's odometric information is totally wrong or when the robot has to recover from incorrect localization. If the robot is moved to a new location without perceiving the kidnapping, it will never recover from the localization error.

One approach to detecting kidnapping of the robot is to examine a tracking failure because it indicates that the robot may have been kidnapped [6]. However, it is unclear how long the tracking failure must happen in order to determine the kidnapping situation. When the robot is kidnapped during map building, this problem can be more serious, because a distortion of the map is not avoidable if the robot is not responsive to kidnapping. For this reason, a wheel drop switch that is able to immediately detect kidnapping was employed in this work.

Many home cleaning robots have been developed over the past few years. Most have the ability to detect when a wheel has dropped; this usually happens if the robot gets picked up and carried to an arbitrary location during its operation, or if it is stuck on something in a manner such that a wheel is off the ground. The wheel drop is detected using a small switch located in the vicinity of the wheel motor, as shown in Fig. 3. The switch sends an on/off indication according to the wheel location.

Another alternative is to attach range sensors such as sonar or PSD sensors to the bottom of a robot platform in order to measure the distance between the robot and the floor. Changes in the distance indicate that the robot may have been lifted.

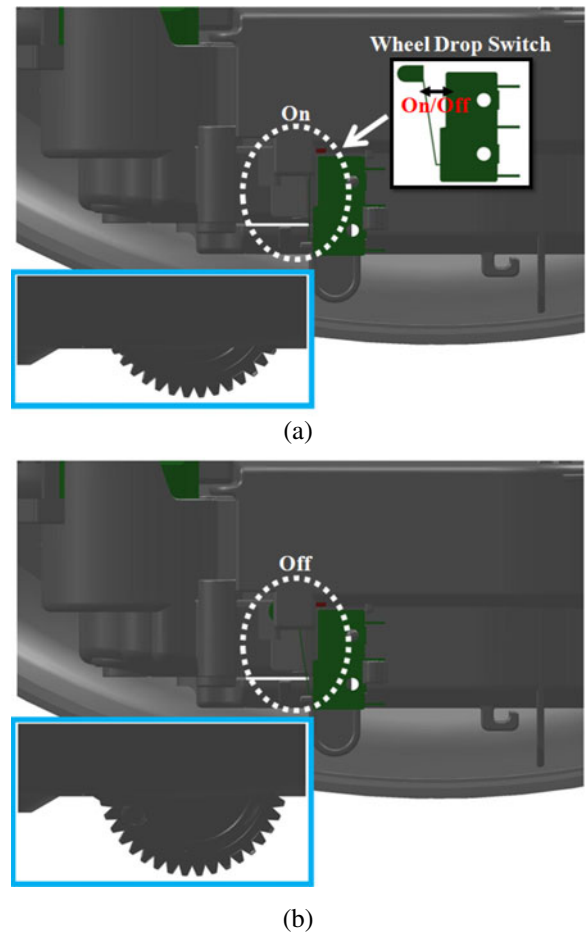

Fig. 3 Example of detecting kidnapping using a wheel drop switch. **a** Normal situation. **b** Wheel drop situation

5 Vision-Based Global Localization

This section presents a novel localization algorithm that efficiently recovers from kidnapping with a map built by our SLAM system, which was briefly described in Section 3. The proposed localization algorithm contains several main modules, and Fig. 4 shows the algorithm overview that illustrates a work-flow between the modules. The feature matching (FM) and recursive poses estimation (RPE) modules are commonly used in the coarse and fine level localization modules.

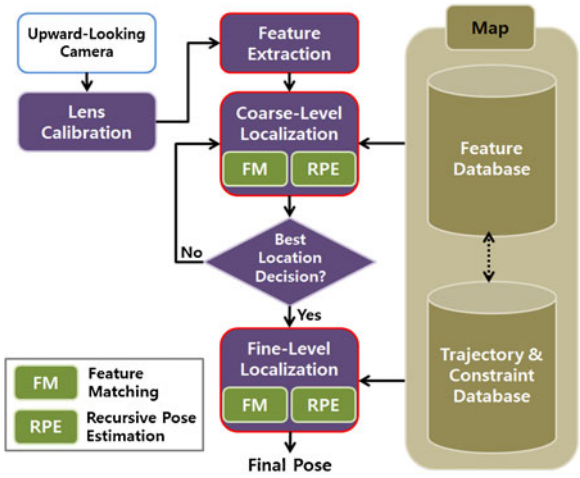

Fig. 4 Overview of the proposed localization algorithm

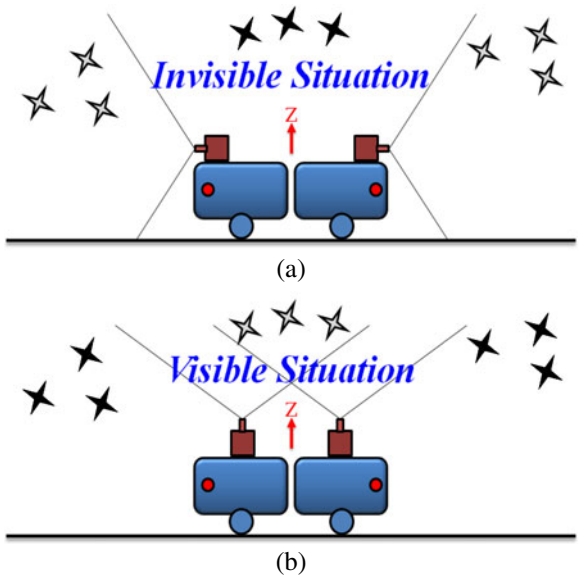

Fig. 5 Advantage of an upward-looking camera in feature matching when compared with a frontal view camera. **a** Frontal view camera. **b** Upward-looking camera

5.1 Feature Extraction & Matching Based on an Upward-Looking Camera

Feature matching is performed with an upward-looking camera that looks in the zenith direction perpendicular to the ground plane, because such a camera has considerable advantages in terms of cost, efficiency, and robustness compared to others in indoor environments. First, visual features can be quickly extracted because there is no need to consider the scale invariance property of features. In addition, feature matching between adjacent images captured from the upward-looking camera can be performed regardless of a rotation of the robot about the z-axis of the robot base reference frame.

Figure 5 shows an example that demonstrates an advantage of the upward-looking camera in feature matching when compared with a frontal view camera. The frontal view camera is strictly restricted by a rotation of the robot about the z-axis of the robot base reference frame in feature matching between two images despite the fact that the two images are captured at adjacent locations (See Fig. 5a). On the other hand, feature matching between two adjacent images captured from the upward-looking camera is possible regardless of a rotation of the robot about the z-axis of the robot base reference frame, as shown in Fig. 5b. The maximum distance required between two camera reference frames for obtaining the appropriate matching result depends on the camera's field of view and a ceiling height.

The SURF method [2] is a well-known approach to wide-base line matching. However, the location accuracy of a blob-like feature extracted by the SURF method is relatively poor compared to that of a corner-like feature. In addition, the feature location extracted by the SURF technique is sensitive to changes in scene illumination. In order to extract more stable features, the Harris corner detector [3] is employed to use corner-like features as points of interest. The mid-points of straight-lines are also considered to be points of interest because they allow for improvement in the feature matching performance. The straight-lines are extracted from an image based on the edge following approach [4].

Features are only extracted at the original image scale because there is no need to consider the scale invariance property of features extracted with a camera pointing in the zenith direction perpendicular to the ground plane. This assumption is true for most indoor environments. The SURF descriptor with 64 dimensions is used as a local descriptor for robust feature matching.

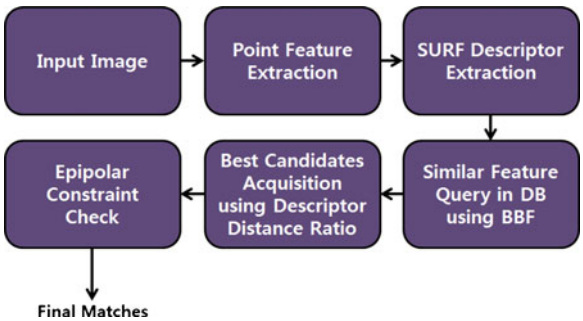

Fig. 6 Flowchart for feature matching between an input image and the database

In order to match features between different images, we employ the Best-Bin-First algorithm [5] to query for similar features in the database and the best candidate matches are founded by checking the ratio of distance from the closest neighbor to the distance of the second closest in the feature descriptor space. When the distance ratio of a certain match is less than 0.6, it is considered as a correct one.

To reduce false matches, we additionally apply the epipolar constraint of the corresponding points ψ_i and ψ_j between two images such as $\psi_i^T F \psi_j = 0$ described by the fundamental matrix F between two cameras. The fundamental matrix F is computed using the Random Sample Consensus (RANSAC) based algorithm described in [13] and the algorithm requires at least eight matched features. In this work, the Euclidean distance of local descriptors and the distance from ψ_i or ψ_j to the epipolar line were used as a quality of correspondence between the two points ψ_i and ψ_j.

A flow chart of the feature matching algorithm presented in this section is shown in Fig. 6. In order to evaluate the feature matching performance, we examined the correct-matching-ratio using 100 image pairs captured from an upward-looking camera. In this test, we considered matches with less than 5-pixel error as correct ones and 81% was the average correct-matching-ratio. This result was suitable for a robot pose estimation method based on a recursive Bayesian formula and the method will be presented in the next section.

Figure 7 graphically illustrates the matching results between upward looking scenes arbitrarily

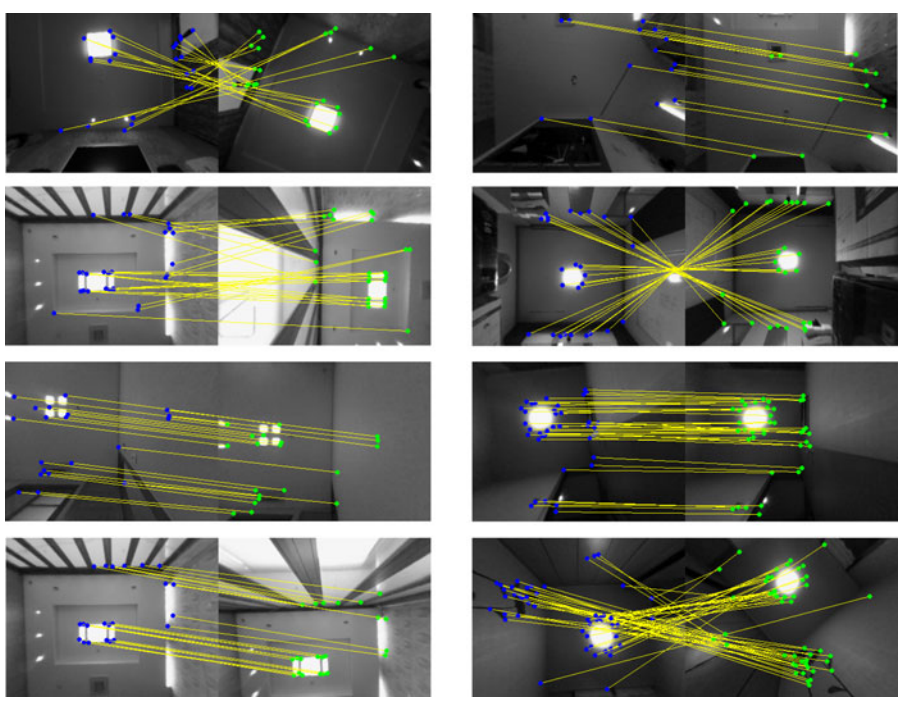

Fig. 7 Matching results in different views

selected among the 100 image pairs. Most matches were good ones and certain features were partially invariant to illumination changes.

5.2 Recursive Pose Estimation from a Single Image

This section describes how to estimate the robot pose on the coarse or fine level using a recursive Bayesian estimation approach given a single image. Here, we apply the recursion (which is limited to a single time-stamp) to obtain the optimal pose estimate. As our upward-looking camera is restricted to approximate only a 2D planar motion, a 3D point p_i associated with a node x_v of the graph can be re-projected to its expected image coordinates at the robot pose x_k to be estimated at time k using the pinhole camera model based on a 2D robot pose $\left[x_{(x_k)}, y_{(x_k)}, \theta_{(x_k)}\right]^T$. Under this condition, the observation model for estimating the robot pose x_k is given by:

$$p\left(\Psi_i | x_k, x_v, p_i\right) \Leftrightarrow \Psi_i = g\left(x_k, x_v, p_i\right) + v_i, \tag{6}$$

where Ψ_i is the measured image coordinates, and v_i is Gaussian observation noise. Figure 8 shows a schematic diagram for Eq. 6.

A recursive Bayesian formula was derived in this work in order to efficiently estimate the robot pose with Eq. 6. In a Bayesian sense, the posterior of the robot pose x_k given a node x_v of the graph, a set of 3D points $[p_1,\ldots,p_n]^T$, and their observations $[\Psi_1,\ldots,\Psi_n]^T$ is represented by:

$$p\left(x_k | x_v, p_1, ..., p_n, \Psi_1, ..., \Psi_n\right). \tag{7}$$

Equation 7 can be rewritten as the following equation because the node x_v of the graph and the set of 3D points $[p_1,\ldots,p_n]^T$ do not provide new information about the robot pose x_k without the observations $[\Psi_1,\ldots,\Psi_n]^T$ in a global localization algorithm that deals with the kidnapped robot problem.

$$\begin{aligned} &\propto p(\Psi_1, ..., \Psi_n | x_k, x_v, p_1, ..., p_n) p(x_k | x_v, p_1, ..., p_n) \\ &\propto p(\Psi_1, ..., \Psi_n | x_k, x_v, p_1, ..., p_n) p(x_k). \end{aligned} \tag{8}$$

By making two assumptions that the set of 3D points $[p_1,\ldots,p_n]^T$ are mutually independent and that Ψ_i is associated with x_k, x_v, and p_i, Eq. 8 can be factored as follows:

$$\propto p\left(x_k\right) \prod_{i=1}^{n} p\left(\Psi_i | x_k, x_v, p_i\right). \tag{9}$$

Note that the correlation between the locally referenced features that know their 3D locations at each node can be disregarded in a pose graph optimization technique to SLAM because the map consistency relies on the spatial constraints identifying a relative transformation between poses rather than the geometric information between the features. Here, Eq. 9 is one of the recursive Bayesian formulas. One major difference between Eq. 9 and the traditional formula for recursive Bayesian estimation is the dynamic system model. The dynamic system model does not need to be considered in Eq. 9, because the updated state denotes only a single robot pose.

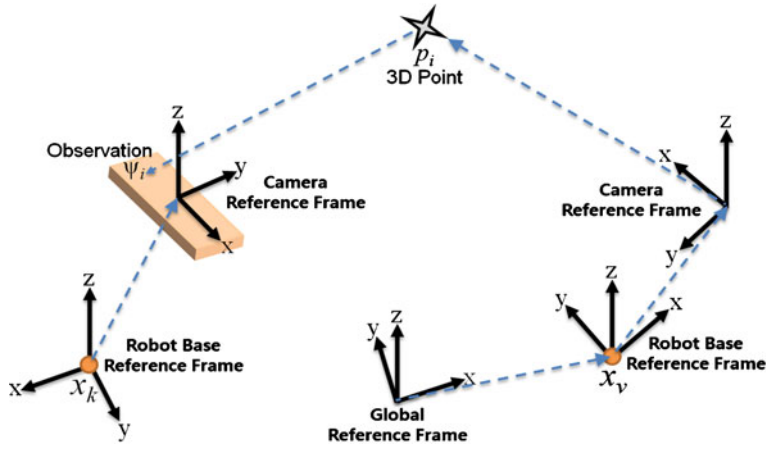

Fig. 8 Geometric relationship between the two poses x_k and x_v

In this work, $p(x_k)$ is approximated by Gaussian distribution with any mean $\hat{x}_k^-$ and a large amount of error covariance matrix C_k^-, and is updated with the same formula used for updating the state in an extended Kalman filter (EKF) [8]. The filtering result is also represented by Gaussian distribution. The Gaussian distribution is again used as the prior probability distribution in updating the estimate of the robot pose x_k with the next observation. The above process is repeated until the posterior of the robot pose x_k is obtained by n iterations. The algorithm for computing Eq. 9 is briefly described in Algorithm 2.

Algorithm 2 Calculation of Eq. 9

- Input: Node x_v, 3D points $[p_1, ..., p_n]^T$, observations $[\Psi_1, ..., \Psi_n]^T$

1. Initialize $p(x_k) \sim N(\hat{x}_k^-, C_k^-)$.
2. **for** $i = 1$ **to** n
   if $i = 1$ **then**
   $Prior = p(x_k)$
   else
   $Prior = p(x_k \mid x_v, p_1, ..., p_{i-1}, \Psi_1, ..., \Psi_{i-1})$
   end
   EKF_update ($Prior$, $p(\Psi_i \mid x_k, x_v, p_i)$)
   end

Algorithm 3 Recursive pose estimation algorithm (RPEA)

- Input: Node x_v, 3D points $[p_1, ..., p_n]$, observations $[\Psi_1, ..., \Psi_n]^T$.

1. Initialize $p(x_k) \sim N(\hat{x}_k^-, C_k^-)$.
2. **for** $j = 1$ **to** max_iter
   Estimate the posterior $N(\hat{x}_k^+, C_k^+)$ using the line 2 of Algorithm 2.
   if *convergence criteria* $=$ *true* **then**
   Return $N(\hat{x}_k^+, C_k^+)$.
   else
   Initialize $p(x_k) \sim N(\hat{x}_k^+, C_k^-)$.
   end
   end

In order to find a local minimum that is identical to that attained with the Levenberg-Marquardt algorithm (LMA, one of nonlinear optimization techniques), Eq. 9 is solved iteratively until some convergence criteria are matched. One difference between this iterative scheme and the LMA is that the initial state covariance of the EKF is replaced with a relatively large amount of error covariance compared to the observation noise at all iterations. The recursive pose estimation algorithm that iteratively solves Eq. 9 will henceforth be referred to as the RPEA, and is summarized in Algorithm 3.

Algorithm 4 RPEA based on the ransac approach

- Input: Node x_v, 3D points $[p_1, ..., p_n]^T$, observations $[\Psi_1, ..., \Psi_n]^T$.

1. Choose 5 random pairs among $[p_1, ..., p_n]^T$ and $[\Psi_1, ..., \Psi_n]^T$.
2. Estimate the robot pose using the RPEA (See Algorithm 3).
3. Check the number of inliers and their re-projection error.
4. Repeat the line 1 to 3 until n poses are obtained.
5. Decide the best hypothesis.

Using all of the observations, the simple estimation of the robot pose is sensitive to outliers. To cope with this problem, the RANSAC approach is employed to obtain the best pose against the outliers. In our implementation, the best result was achieved using 5 random pairs of correspondences between the image features and their 3D locations in a reference frame. In order to find the best hypothesis, each hypothesis is checked for inliers by re-projecting the 3D points to their expected image coordinates. The best hypothesis is determined by the number of inliers and their re-projection errors. The algorithm is briefly described in Algorithm 4.

However, the RPEA implemented by the EKF may be unstable because it is a suboptimal solution like the LMA. It depends strongly on the initial guess in the case of multiple minima. Therefore, the RPEA has a problem with divergence when the initial condition is far from the global minimum. Shown in Fig. 9 is a simulation result that includes a few divergence cases among 3,600 runs with different initial guess values. In the simulation, 5 features were used to estimate the robot pose. In order to solve the divergence

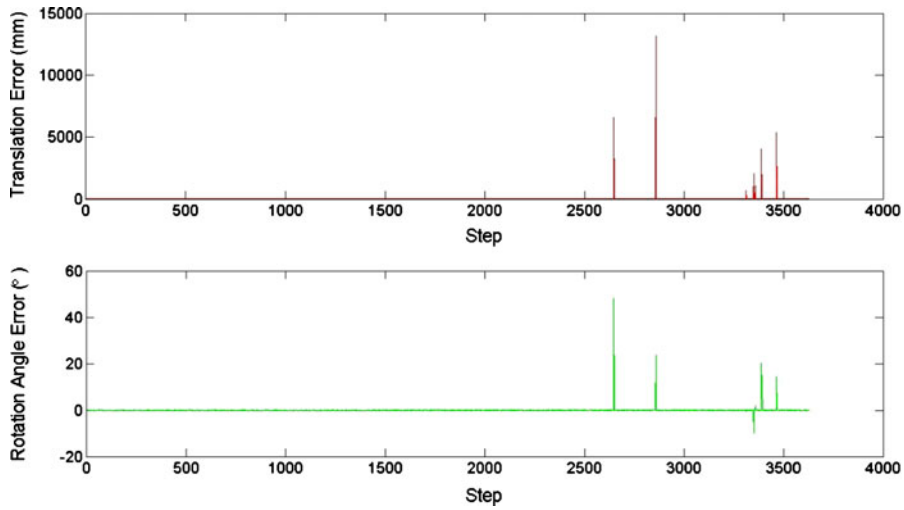

Fig. 9 Divergence cases of the RPEA under certain initial conditions. The translation and rotation angle errors were obtained by using 3,600 runs with randomly generated initial guess values

problem, multiple initial guess values were used in this work. Fortunately, the RPEA mainly converged to the ground truth regardless of the initial conditions in various simulation tests, as shown in Fig. 9. Under this condition, the robot pose that is identically estimated from the majority of initial guess values is more likely to converge to the global minimum. Here, we used 4 initial guess values to find the global minimum.

5.3 Multi-Level Localization

In this work, a key problem in localizing a robot without prior knowledge regarding its pose concerns feature matching between the current image and a graph composed of a very large number of nodes. In order to solve this problem, the matching process is performed at multiple levels of resolution. The basic idea is to discretize the graph at different levels of resolution. The matching process on a coarse level effectively limits the search area for matching between features extracted from the current image and a set of auxiliary features associated with nodes of the graph. The matching process on a finer level then allows for more accurate pose estimation.

A hierarchy of coarser levels is easily constructed by discretizing the graph onto grids with increasing grid spacing, (i.e., onto fewer sampling points). Figure 10 shows an example with a three-level hierarchy for efficiently obtaining the nodes that yield a large number of matches with an input image in the graph composed of numerous nodes. The grid size of the 1st coarse level depends on the camera's field of view and a ceiling height. A representative node is selected in the vicinity of each grid center and is used to match features extracted from the input image with the auxiliary features associated with the node, as shown in Fig. 10a. With the feature matching results between the input image and the representative nodes, multiple poses are generated by the RPEA based on the RANSAC approach. In order to decide the search area of the feature matching on the fine level, an X-Y histogram with 0.5 m $\times$ 0.5 m bins is built from the multiple poses, as shown in Fig. 10b. The highest peak in the histogram is detected, and the weighted average location is then calculated in the vicinity of the peak.

There are two thresholds to consider when evaluating the validity of the weighted average location: the minimum number of poses and the ratio of the poses used to calculate the weighted average location to the total number of poses. The two thresholds are decided empirically; we used 3 poses and a ratio of 0.7, respectively. If the two thresholds are satisfied, the weighted average location is considered to be the best location on the coarse level. Otherwise, the above process is repeated on the next coarse level until the best loca-

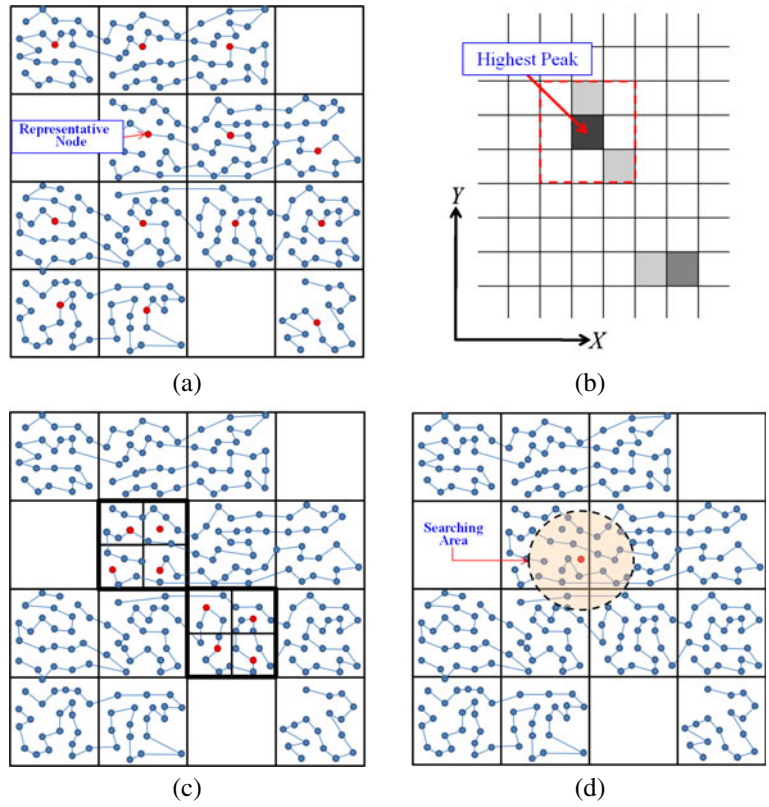

Fig. 10 Example of a three-level hierarchy to reduce the computational complexity of the matching process. **a** 1st coarse level. **b** X-Y histogram. **c** 2nd coarse level. **d** Fine level

tion is determined. The next coarse level is formed by simply reducing the sizes of grids that include the poses generated on the previous coarse level (See Fig. 10c). Note that the grid size of the 1st coarse level must be reduced if no valid poses exist on the 1st coarse level. Of course, this scheme may continuously fail. In this case, the robot moves to another place and tries again to recover from the kidnapping.

If the best location is decided on the coarse level, the fine level matching process is performed with nodes that are located in the vicinity of the best location (See Fig. 10d). The multiple poses on the fine level are ultimately used to estimate the final robot pose in the kidnapping situation. Because the multiple poses that are obtained by the RPEA are represented by Gaussian distribution with mean and covariance matrix, the final robot pose at time k can be simply estimated using the standard Kalman filter formulation, as follows:

$$\hat{x}_k^* = \left(\sum_{i=1}^{n} \left(C(i)_k^+\right)^{-1}\right)^{-1} \sum_{i=1}^{n} \left(\left(C(i)_k^+\right)^{-1} \hat{x}(i)_k^+\right),$$

$$C_k^* = \left(\sum_{i=1}^{n} \left(C(i)_k^+\right)^{-1}\right)^{-1}, \tag{10}$$

where $\hat{x}(i)_k^+$ and $C(i)_k^+$ are the mean and covariance matrix of the i^{th} pose on the fine level, respectively. In order to reduce the likelihood of incorrect data association, the validity of the final robot pose is evaluated using the standard deviation of multiple poses used in the calculation of Eq. 10 because the standard deviation is able to measure the spread of multiple poses about

the final robot pose. This standard deviation is given by:

$$\sigma\left(\hat{x}_k^*\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{x}(i)_k^+ - \hat{x}_k^*\right)^2}. \quad (11)$$

In this work, three thresholds for the standard deviation $\sigma(\hat{x}_k^*)$ were set to 15 cm, 15 cm, and 3°. If these criteria are not matched, the matching process on the fine level is iteratively performed while capturing a new image in the vicinity of the current location until the criteria are matched.

Another aspect to be considered in global localization is the handling of ambiguous situations. The ambiguity of initial poses occurs due to symmetry of a certain environment or multiple areas with similar textures. In vision-based localization, a camera with a wide-angle lens reduces the likelihood of ambiguous situations. However, there is a trade-off between the camera's field of view and localization accuracy. In the proposed localization algorithm, the pose ambiguity is naturally identified in the process that decides the best location from multiple poses generated on the coarse level. If an ambiguous situation is detected while recovering from a kidnapping situation, the robot simply moves to another place until the ambiguity of the robot pose disappears.

6 Experimental Results

We implemented the proposed kidnap recovery method using C programming language on an embedded vision board with an ARM11 processor of a real home cleaning robot (See Fig. 1). The ARM11 processor was running at 533 MHz. A single camera on the embedded vision board was pointing in the zenith direction perpendicular to the ground. To demonstrate the validity of the proposed method, we extensively tested the proposed method in home environments. In all of the tests, the robot moved at a speed of 0.3 m/s. All sensor data and experimental data were collected during all of the tests through a wireless LAN. Logged time-stamps were again used to recreate and analyze the experimental results on a laptop computer. A 320 × 240 resolution image captured by the camera with a wide-angle lens of 160° was used after the lens calibration. The ceiling height from the camera in the experiments was about 2.3 m.

6.1 Home Environment Test

The first experiment was conducted in a home environment. The experimental environment was 7 × 9 m in size. The sensor data were divided into three categories. First, numerous images of the robot being moved by a zigzag motion-based navigation system were saved in order to incre-

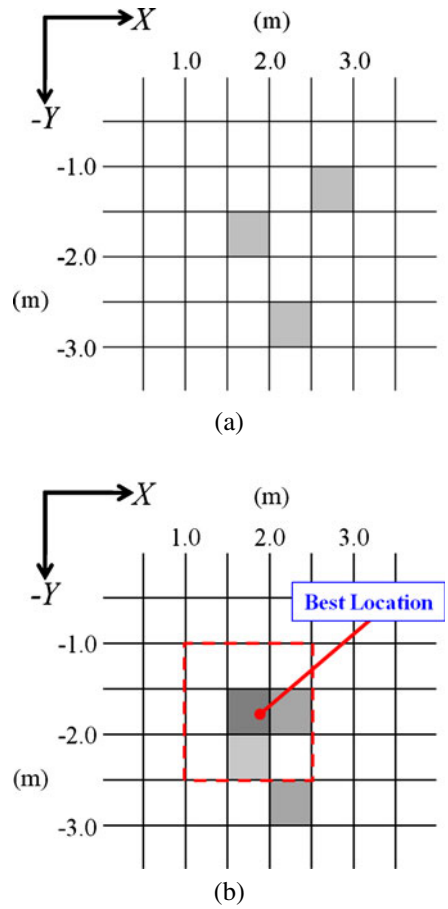

Fig. 11 X-Y histograms of the two coarse levels. **a** 1st coarse level. **b** 2nd coarse level

mentally build a map. This type of navigation system was effective for exploring the entire experimental environment. Second, additional images of the robot were collected when it was kidnapped. Third, new images of the robot continuing to explore after recovering from the kidnapping situation were saved in order to build a map of the entire experimental environment. The map was constructed using our SLAM system described in Section 3.

The global localization capability of the proposed method is illustrated in Figs. 11, 12 and 13. The black path shows the robot trajectory estimated by our SLAM system. An obstacle grid map with red superimposed onto the robot trajectory was constructed with PSD sensors. In the experiment, the robot was suddenly carried to an arbitrary unknown location while the robot was building a map. At that time, the robot detected its kidnapping by a signal sent from the wheel drop switch, and then switched to kidnap recovery mode. A new image was autonomously captured when the robot was placed on the ground. With the new image, the best location on the coarse level was determined on the 2nd coarse level with a grid size of 1 m × 1 m. Figure 11 shows X-Y histograms for each coarse level and multiple poses generated on each coarse level were voted to fixed bins with a 0.5 m × 0.5 m size regardless of grid sizes in each coarse level. Twelve poses were then generated in the vicinity of the best location (See Fig. 12a), and were used to estimate the final robot pose using the standard Kalman

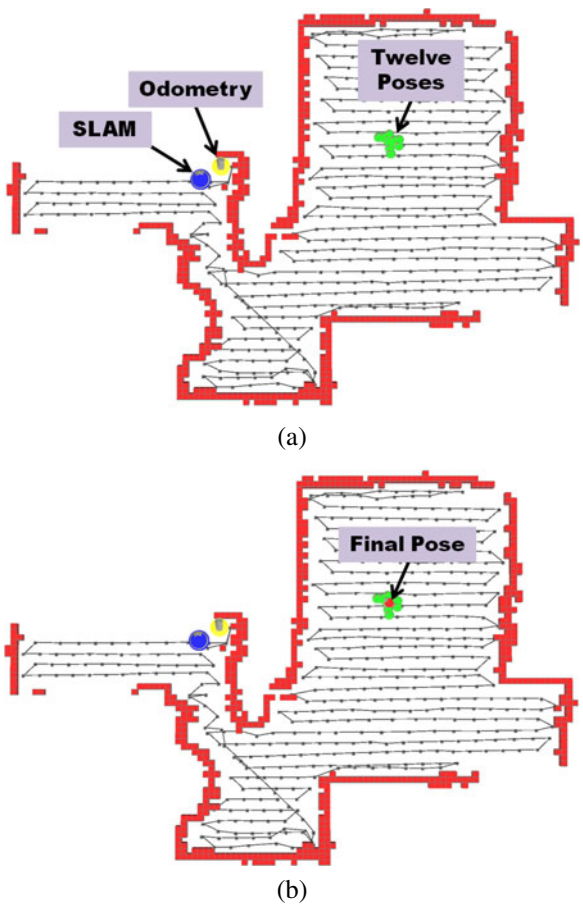

Fig. 12 Localization result of the first experiment. **a** Poses on the fine level. **b**Achieved localization

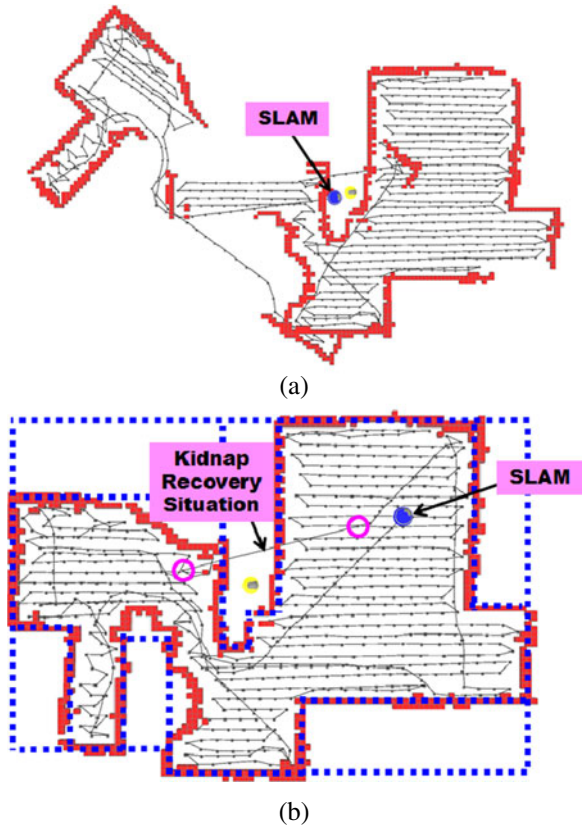

Fig. 13 Maps constructed at the end of the exploration during the first experiment before and after applying the proposed method in a kidnapping situation. The ground truth (including big obstacles such as sofa, furniture, doors, and walls) is shown by blue dotted lines. **a** Without kidnap recovery. **b** With kidnap recovery

Table 1 Computation time of the proposed method in the experiment

Functional Module	Time (ms)
Feature Extraction	253
Feature Matching (1st coarse level)	954
X-Y Histogram (1st coarse level)	19.3
Feature Matching (2nd coarse level)	636
X-Y Histogram (2nd coarse level)	27.5
Feature Matching (fine level)	1272
Final Pose Estimation	68.8

filter (Refer to Section 5.3 for details). The final robot pose is shown in Fig. 12b.

In the experiment, the grid size of the 1st coarse level was set to 2 m × 2 m and the total execution time for global localization was 3.23 s given 287 nodes with approximately 36,000 features. The computational costs of the proposed method in this experiment are presented in Table 1. The highest processing time was consumed by the feature matching module in the fine level, but the processing time is almost constant regardless of the map size because we use a fixed searching range in the fine level. For this reason, the speed of the proposed method depends on the map size and the grid size of the 1st coarse level to determine the best location. The smaller the grid size of the 1st coarse level is, the more increased the total execution time becomes.

For verification of the localization accuracy, a map depicting the entire experimental environment was built in addition to the previous map constructed before the kidnapping. The final maps built before and after applying the proposed method in the kidnapping situation are illustrated in Fig. 13. The map built without the aid of the kidnap recovery algorithm had large distortions, and was difficult to use for navigation. However, the map error accumulated when applying the proposed method was much smaller, and the obstacle location error between the proposed method and the ground truth data was less than 17 cm (See Fig. 13b).

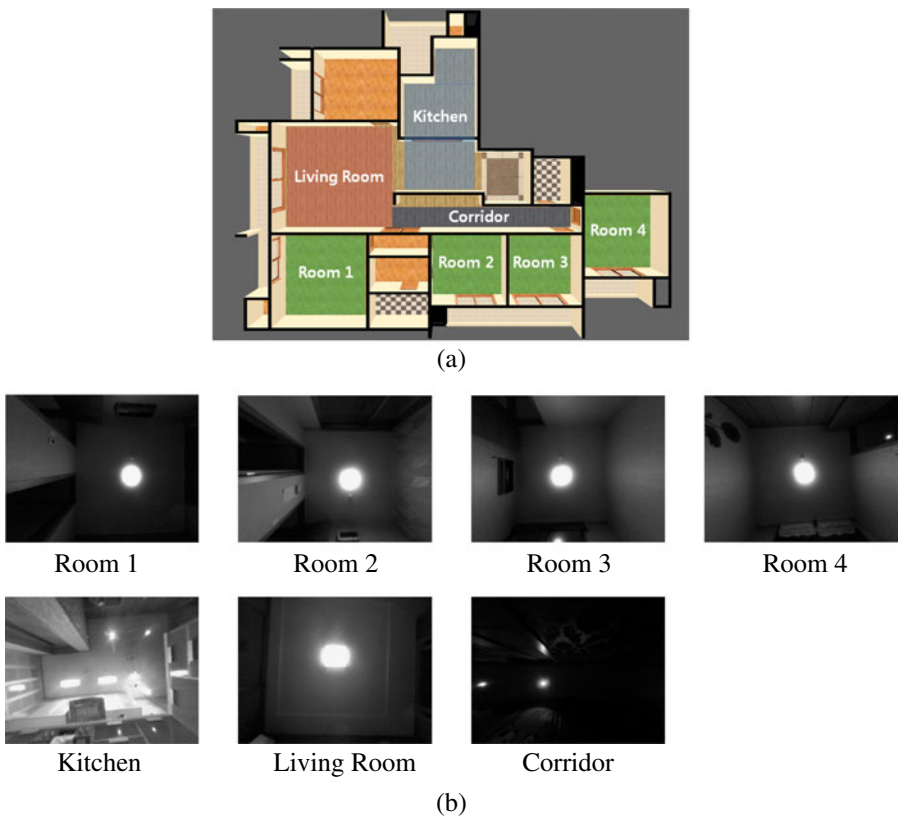

Fig. 14 Second experimental settings. **a** Seven areas in a home environment. **b** Sample images in the seven areas

6.2 Statistical Analysis

The second experiment was conducted to statistically evaluate the performance of the proposed method in a home environment. Low-cost service robots such as home cleaning robots are frequently kidnapped in real-world situations because the robots are often stuck on something in a manner such that a wheel is off the ground during navigation, and are then carried to another location relatively close to the stuck place by a user. In this case, the user will want the robot to continue to perform the task that it was given before it got stuck. In order to solve this problem, the localization system must ensure that it can quickly recover from such a kidnapping with a relatively high probability, regardless of the kidnapped place in a home environment.

The experimental environment was 17 × 20 m in size, and was divided into multiple small areas, as shown in Fig. 14. The robot was suddenly carried to another location in each area while it was building a map. For statistical analysis, 10 tests were performed in each area and the maximum trials allowed to recover from the kidnapping in each test were limited to 5 times.

Figure 15 shows the success rate when applying the proposed method in various kidnapping situations. The highest success rate is shown in a kitchen because there were highly textured patterns (See Fig. 14b). On the other hand, the lowest success rate is shown in two rooms and a corridor, with 80% of success. Similar or poorly textured patterns of the ceiling in some indoor environments were likely to make it harder to localize the robot using an upward-looking camera. In this experiment, the localization failure was caused by two reasons. First, sufficient matches required for place recognition in the feature matching process did not exist in some places. Second, the robot pose estimation was impossible due to a high reprojection error caused by depth errors or false matches. A naive approach to solving these problems may be to allow the robot to continuously wander until it recovers from its kidnapping. If we applied this approach to this experiment, the success rate of the proposed method would reach almost 100% in all areas of this experimental environment. However, this approach is a comparatively time-consuming procedure.

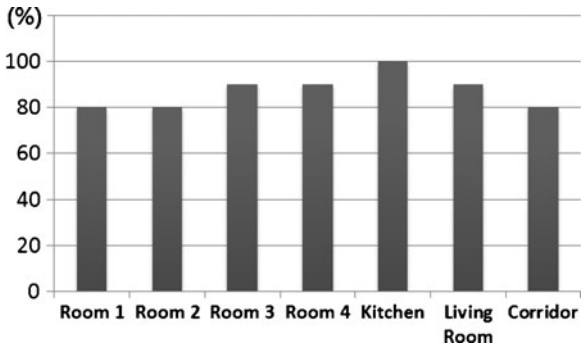

Fig. 15 Success rate when applying the proposed method in the kidnapping scenario of the second experiment

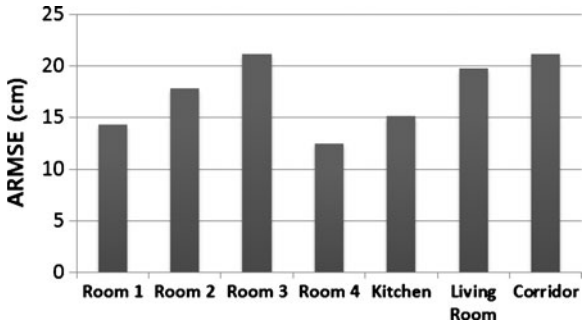

Fig. 16 Estimation error of the proposed method

Figure 16 shows the average root mean square error (ARMSE) between the ground truth location of the robot and its estimated location by the proposed method in the experiment. For the proposed method, the robot location error is clearly bounded and does not diverge in the experiment.

7 Conclusions

In this paper, we have presented a global localization algorithm using a wheel drop switch and an upward-looking camera that efficiently solves the problem of robot kidnapping in unstructured indoor environments. The proposed method was applied to a real home cleaning robot that yields poor odometric estimates and low resolution images taken by an upward-looking camera. The camera, which has a wide-angle lens, allowed us to localize the robot globally with a single image. In

order to recover from a kidnapping situation in a reasonable amount of time, the matching process was performed at multiple levels of resolution. The matching process on a coarse level effectively limited the search area for feature matching between the current image and the map elements. The finer matching process was used to estimate a more accurate pose. The experimental results show that the proposed method can achieve a high level of localization accuracy on an embedded system for a home cleaning robot in a situation in which a robot is suddenly kidnapped.

In our research, there still remain some problems to be addressed. The first arises when global localization continuously fails due to poorly textured patterns of the ceiling in some indoor environments. To solve this problem, we have to improve the feature matching performance using additional range sensors such as PSD sensors or develop an active path-planning method to relocate the kidnapped robot to highly textured places in the map through trial and error. The second problem is global localization in large-scale indoor environments. The proposed localization method iteratively performs feature matching between two images through the matching process at multiple levels of resolution. To reduce computing time, a more efficient and robust feature matching method must be developed in future research. The last problem is depth errors of visual point features caused by a ceiling height. Theoretically, the proposed method works well in indoor environments including different ceiling heights if feature depths are accurately estimated regardless of the ceiling height. However, the higher the ceiling height is, the more increased the feature depth error becomes. For home cleaning robots, we have to solve this problem without resorting to expensive hardware.

Acknowledgements This work was supported in part by the Future IT Laboratory of LG Electronics Inc., in part by the KORUS-Tech (Korea-US Tech Program KT-2010-SW-AP-FS0-0004) program sponsored by the Ministry of Knowledge Economy, in part by the NRF (National Research Foundation) WCU program (R31-2010-000-10062-0) sponsored by the Ministry of Education, Science and Technology, and in part by the Gyeonggi International Collaboration Research Program (Code I09200) sponsored by the Gyeonggi Province of Korea.

References

1. Durrant-Whyte, H.F., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). Int. J. Comput. Vis. Image Understand. **110**(3), 346–359 (2008)
3. Harris, C., Stephens, M.: A combined corner and edge detector, In: Proceeding of the 4th Alvey Vision Conference, pp. 147–151 (1988)
4. Lee, S., Kim, E., Park, Y.: 3D object recognition using multiple features for robotic manipulation. In: Proceeding of the International Conference on Robotics and Automation, pp. 3768–3774. Orlando (2006)
5. Lowe, D.G.: Distinctive image features from scale-invariant key-points. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
6. Se, S., Lowe, D.G., Little, J.J.: Vision-based global localization and mapping for mobile robots. IEEE Trans. Robot. **21**(3), 364–375 (2005)
7. Murillo, A.C., Guerrero, J.J., Sagues, C.: SURF features for efficient robot localization with omnidirectional images. In: Proceedings of the International Conference on Robotics and Automation, pp. 3901–3907. Roma (2007)
8. Sorenson, H.W., Stubberud, A.R.: Non-linear filtering by approximation of the a posteriori density. Int. J. Control. **8**(1), 33–51 (1968)
9. Thrun, S.: Bayesian landmark learning for mobile robot localization. Mach. Learn. **33**(1), 41–76 (1998)
10. Grisetti, G., Stachniss, C., Grzonka, S., Burgard, W.: A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In: Proceedings of Robotics: Science and Systems, Atalanta (2007)
11. Konolige, K.: SLAM via variable reduction from constraint maps. In: Proceedings of the International Conference on Robotics and Automation, Barcelona, pp. 667–672 (2005)
12. Borges, P., Zlot, R., Bosse, M., Nuske, S., Tews, A.: Vision-based localization using an edge map extracted from 3D laser range data. In: Proceedings of the International Conference on Robotics and Automation, pp. 4902–4909. Anchorage (2010)
13. Zhang, Z.: Determining the epipolar geometry and its uncertainty: a review. Int. J. Comput. Vis. **27**(2), 161–195 (1998)
14. Lu, F., Milios, E.: Globally consistent range scan alignment for environment mapping. Auton. Robot. **4**(4), 333–349 (1997)
15. Bailey, T., Durrant-Whyte, H.F.: Simultaneous localization and mapping: part II. IEEE Robot. Autom. Mag. **13**(2), 108–117 (2006)
16. Martinelli, A., Nguyen, V., Tomatis, N., Siegwart, R.: A relative map approach to SLAM based on shift and rotation invariants. Robot. Auton. Syst. **55**(1), 50–61 (2007)

17. Steder, B., Grisetti, G., Stachniss, C., Burgard, W.: Visual slam for flying vehicles. IEEE Trans. Robot. **24**(5), 1088–1093 (2008)
18. Gutmann, J.S., Konolige, K.: Incremental mapping of large cyclic environments. In: Proceedings of the International Symposium of Computational Intelligence in Robotics and Automation, pp. 318–325. Monterey (1999)
19. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge (2000)
20. Moreno, L., Garrido, S., Munoz, M.L.: Evolutionary filter for robust mobile robot localization. Robot. Auton. Syst. **54**(7), 590–600 (2006)
21. Wolf, J., Burgard, W., Burkhardt, H.: Robust vision-based localization by combining an image retrieval system with monte carlo localization. IEEE Trans. Robot. **21**(2), 208–216 (2005)
22. Nuske, S., Roberts, J., Wyeth, G.: Robust outdoor visual localization using a three-dimensional-edge map. J. Field Robot. **26**(9), 728–756 (2009)
23. Gasparri, A., Panzieri, S., Pascucci, F., Ulivi, G.: Monte carlo filter in mobile robotics localization: a clustered evolutionary point of view. J. Intell. Robotic. Syst. **47**(2), 155–174 (2006)
24. Penne, R., Mertens, L., Veraart, J.: Mobile camera localization using apollonius circles and virtual landmarks. J. Intell. Robotic. Syst. **58**(3–4), 287–308 (2010)
25. Kalman, R.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)
26. Baltzakis, H., Trahanias, P.: Hybrid mobile robot localization using switching state-space models. In: Proceedings of the International Conference on Robotics and Automation, pp. 366–373. Washington D.C. (2002)
27. Crowley, J.L.: Mathematical foundations of navigation and perception for an autonomous mobile robot. In: Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics, pp. 9–51. Berlin (1995)
28. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (1999)
29. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
30. Thrun, S., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., H¨ahnel, D., Rosenberg, C., Roy, N., Schulte, J., Schulz, D.: MINERVA: a second generation mobile tour-guide robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, (1999)
31. Folkesson, J., Jensfelt, P., Christensen, H.: Vision SLAM in the measurement subspace. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 30–35. Barcelona (2005)
32. Jeong, W.Y., Lee, K.M.: CV-SLAM: A new ceiling vision-based SLAM technique, In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3195–3200. Edmonton (2005)
33. Matsumoto, T., Takahashi, T., Iwahashi, M., Kimura, T., Salbiah, S., Mokhtar, N.: Visual compensation in localization of a robot on a ceiling map. Sci. Res. Essay **6**(1), 131–135 (2011)