Auton Robot (2008) 24: 315–335
DOI 10.1007/s10514-007-9083-2

A practical approach for EKF-SLAM in an indoor environment: fusing ultrasonic sensors and stereo camera

SungHwan Ahn · Jinwoo Choi · Nakju Lett Doh · Wan Kyun Chung

Received: 28 May 2007 / Accepted: 19 December 2007 / Published online: 5 January 2008

Abstract Improving the practical capability of SLAM requires effective sensor fusion to cope with the large uncertainties from the sensors and environment. Fusing ultrasonic and vision sensors possesses advantages of both economical efficiency and complementary cooperation. In particular, it can resolve the false data association and divergence problem of an ultrasonic sensor-only algorithm and overcome both the low frequency of SLAM update caused by the computational burden and the weakness to illumination changes of a vision sensor-only algorithm. In this paper, we propose a VR-SLAM (Vision and Range sensor-SLAM) algorithm to combine ultrasonic sensors and stereo camera very effectively. It consists of two schemes: (1) extracting robust point and line features from sonar data and (2) recognizing planar visual objects using a multi-scale Harris corner detector and its SIFT descriptor from a pre-constructed object database. We show that fusing these schemes through EKF-SLAM frameworks can achieve correct data association via the object recognition and high frequency update via the sonar features. The performance of the proposed algorithm was verified by experiments in various real indoor environments.

S. Ahn (✉) · J. Choi · W.K. Chung
Robotics Lab., Department of Mechanical Engineering, Pohang University of Science & Technology (POSTECH), Pohang, Korea
e-mail: ash@postech.ac.kr

J. Choi
e-mail: shalomi@postech.ac.kr

W.K. Chung
e-mail: wkchung@postech.ac.kr

N.L. Doh
School of Electrical Engineering, Korea University, Seoul, Korea
e-mail: nakju@korea.ac.kr

Keywords Ultrasonic sensor · Sonar feature detection · Stereo camera · Visual object recognition · SLAM · Mobile robot

1 Introduction

Simultaneous localization and map building (SLAM) based on onboard sensor data is a common way to autonomously navigate a mobile robot. Until now, much research has been done on theoretical SLAM algorithms (Guivant and Nebot 2001; Tardós et al. 2002; Montemerlo et al. 2003; Bosse et al. 2004; Estrada et al. 2005), achieving excellent results.

The importance of a practical SLAM solution under the large uncertainties from the sensors and environment is also remarkably emphasized along with these studies of the SLAM algorithms. Specifically, real applications such as home cleaning robots and service robots in an indoor environment attract public and industrial attention. For that purpose, concurrent utilization of range and vision sensors has been popularly adopted for multiple sensor fusion. It has mainly used a laser range finder as the range sensor because of its superior accuracy. As a result, the multi-sensor fusion method generally outperforms both laser-only and vision-only methods. Ortin et al. (2003) successfully re-localized the robot on the given map by using two dimensional laser scan and its corresponding image. It provides accurate measurement with the geometric information of laser scan and reliable data association with the photometric information of gray scale image. The SLAM method of Folkesson et al. (2005) handled different kinds of landmarks such as the extracted edges from ceiling images and the line features of laser scan in an efficient and systematic way. Newman and

Ho (2005, 2006) showed how a judicious vision-based retrieval system for a loop closure can improve the performance of a laser-only method in both indoor and outdoor environments.

Despite the excellent performance of the above combination, ultrasonic sensors are more suitable for practical SLAM solutions in an indoor environment instead of laser range finders because they are cheap and have relatively good cost performance. However, they suffer from large angular uncertainty (aperture angle larger than 22.5°) and range uncertainty due to specular reflection. Many researchers have extracted robust sonar features with post-processing of sonar data to overcome the uncertainties. Leonard and Durrant-Whyte (1991) obtained region of constant depths (RCDs) corresponding to planes, corners, edges and cylinders using a rotating ultrasonic sensor. Wijk and Christensen (2000) developed a point feature detection method, a triangulation-based fusion (TBF) algorithm which detected point features with arc intersections between current and previous sonar data. Tardós et al. (2002) used a Hough transform for point and line feature detection. In spite of these efforts, the performance of sonar-only SLAM algorithms has been restricted because of their inherent uncertainty. In particular, when a robot executes data association using ultrasonic sensors, it is prone to make false data associations. Then it causes the estimated states such as robot pose and landmark position to diverge inevitably.

Vision sensors are appropriate to compensate for the weakness of ultrasonic sensors as the combination of laser range finder and vision sensor because they have abundant and salient information which can achieve reliable data association. Many studies on feature-based approaches of vision-only SLAM have been done in order to utilize reliable data association ability of vision sensors. Davison (2003) efficiently estimated the location of a single camera and related visual features by means of feature tracking. Se et al. (2002) used scale invariant feature transform (SIFT) features generated from a trinocular vision in an indoor environment and maintained the robot pose and the 3-D map of the features separately. The vSLAM (Karlsson et al. 2005) was realized by using a single camera similarly to Se et al. (2002). CV-SLAM (Jeong and Lee 2005) proposed SLAM and kidnapping solutions using a ceiling vision sensor which used Harris corners and their orientation information from ceiling and side walls. Elinas et al. (2006) and Barfoot (2005) directly estimated the motion of the robot and position of SIFT features from a stereo camera with the help of Rao-Blackwellized particle filter. These vision-only algorithms have great data association performance due to abundant visual information. However, the computational burden can be increased by processing large image data and handling a large number of landmarks. Therefore, vision-only SLAM has a lower update frequency than the case of using range sensors, and it is also sensitive to illumination changes.

Notwithstanding these limitations, ultrasonic sensors and vision sensors have a mutually complementary relation which could increase the SLAM performance. The false data association and large uncertainty of ultrasonic sensors can be supplemented by vision sensors. Moreover, the low update frequency and the weakness to illumination changes of vision sensors can be alleviated by using ultrasonic sensors. In this paper, we propose a VR-SLAM (Vision and Range sensor-SLAM) method to practically combine ultrasonic sensors and a stereo camera. It extracts point and line features from sonar data and recognizes visual objects based on salient visual features.

The proposed VR-SLAM is organized as follows; (1) extracting robust sonar features (Choi et al. 2005), (2) recognizing visual objects (Ahn et al. 2006) and (3) fusing both features via EKF (Extended Kalman Filter)-SLAM frameworks.

First, we propose a robust feature detection scheme of both point and line features for ultrasonic sensors. The feature detection is based on the TBF algorithm (Wijk and Christensen 2000) because it gives a good framework for detecting robust point features in an indoor environment. Unlike TBF algorithm, RCDs (Leonard and Durrant-Whyte 1991) are difficult to apply to arbitrary configurations of ultrasonic sensors and cannot be classified into certain types of geometrical features such as point and line features. Besides, Hough transform (Tardós et al. 2002) might be suffered from computational burden due to its inherent voting scheme and noise sensitivity due to the use of threshold values. Unfortunately, the TBF algorithm also has some limitations which need to be improved upon. It occasions false point features along line segments because of the sonar uncertainty, and the map representation with only point features is not sufficient to perform SLAM estimation successfully in an indoor environment. Therefore, we add two processes to the original TBF framework for improving feature detection: (1) filtering unstable intersections and (2) extracting line features. They increase robustness by removing false point features on walls and also gain more opportunity to detect effective sonar features for SLAM update than the original TBF algorithm.

Second, we define a visual object as a set of visual features which represents a physical object and regard it as a landmark in the SLAM map. The appearance-based approach using already available object models in an indoor environment has more advantages as a practical SLAM solution than the aforementioned feature-based approaches of vision-only SLAM. It can reduce the number of landmarks, making SLAM more computationally feasible. And it can improve the performance of data association process because object recognition is more reliable than individual matching between a large number of similar visual features. Also looking for certain objects in the database allows the robot to filter out moving objects and helps it to work in a dynamic environment where people may be moving around the robot. The procedure is as follows. (1) We extract salient visual features composed of multi-scale Harris corners and SIFT descriptors. (2) Then we use a RANSAC clustering method based on a dual error measure of homography to recognize a visual object robustly by retrieving object information from a previously constructed database. (3) Additionally, we obtain accurate range and bearing data of the recognized object for SLAM by filtering outliers which have wrong distance information obtained from the stereo camera.

Finally, we apply both features to two different EKF-SLAM frameworks: standard EKF-SLAM (Dissanayake et al. 2001) and hierarchical SLAM (Estrada et al. 2005) in order to fuse the point and line features of ultrasonic sensors and the visual objects of the stereo camera simultaneously. (1) The standard EKF-SLAM handles the locations of the sonar features and the visual objects as SLAM states simultaneously, maintaining consistency of localization and map building. When the visual objects are recognized, the robot relies on the objects more than the sonar features because of their excellent capability of discrimination for correct data association. Otherwise, the sonar features are used for frequent SLAM updates instead of the visual objects. (2) On the other hand, the hierarchical SLAM creates local maps based on the sonar features. Each local map has its own representative visual objects. Then global consistency between the local maps can be maintained via loop closure which is detected by visual object recognition. Odometry is playing an important role in both systems to estimate planar motion of the robot, though the hierarchical SLAM is less affected by odometry than the standard EKF-SLAM.

The proposed method has an advantage over the previous works which only use either sonar features (Choi et al. 2005) or visual objects (Ahn et al. 2006), especially, in a large environment. Sonar-only SLAM works well in a small environment or local parts of a large environment. In a large environment, however, sonar-only SLAM suffers from loop closure problems. When the vehicle uncertainty grows unboundedly in the large environment, it can be difficult to match the current observation with the correct landmark of the SLAM map due to the limitations of data association ability of sonar features. In case of using only visual objects, the object recognition executes correct data association even in the large environment because of their abundant visual information. However, the resulting map cannot represent the details of the environment because the visual objects are sparsely located to cover the large area. Moreover, the estimated locations of the recognized objects can be erroneous due to the low frequency of SLAM update to compensate odometry information between the sparsely-located objects. On the other hand, the proposed method generates a globally consistent map with the details of the environment because it achieves complementary cooperation of sonar features and visual objects. The cooperation of those features can be found more clearly in the hierarchical SLAM because it uses the proposed sensor suite more systematically than the standard EKF-SLAM. While the standard EKF-SLAM uses both features with the same reliability level, the hierarchical SLAM uses sonar features to estimate the local maps and visual objects to maintain the consistency of the global map, respectively.

This paper is organized as follows. Section 2 describes schemes for robust point and line feature detection using ultrasonic sensors. Section 3 explains an object recognition method based on compatible visual features, and Sect. 4 gives EKF-SLAM frameworks to simultaneously manage the sonar features and the visual objects. Then Sect. 5 shows the experimental results of the proposed VR-SLAM algorithm in three different kinds of indoor environments, and a conclusion follows.

2 Sonar feature detection for VR-SLAM

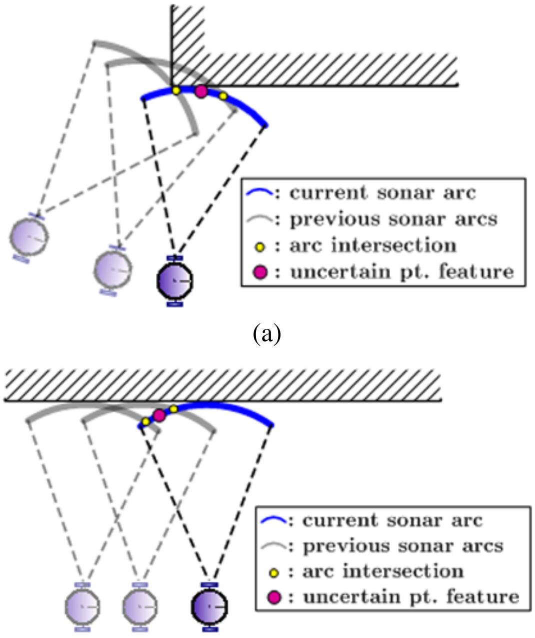
2.1 Robust sonar features in an indoor environment

2.1.1 Requirements of sonar feature extraction algorithm

When a robot extracts landmarks from ultrasonic sensors for SLAM, point and line features are generally used. Point features offer information about corners, edges and pole-type elements of the environment. Line features describe the planes of objects (walls, appliances, furniture and so on) in two dimensional space.

Extraction algorithms of the sonar features should satisfy the following conditions in order to use these features for SLAM estimation.

- It should always provide correct information about the features in a real unstructured environment. The locations of the features have to be determined accurately and robustly even with the large range and angular uncertainties of ultrasonic sensors.
- It should be able to filter out false sonar features, the erroneous point features obtained along line segments due to the sonar uncertainties from walls.
- It should be able to describe the real environment properly with the detected features and guarantee a sufficient number of observations. Using only point features is insufficient for SLAM estimation in an indoor environment because it can not represent the entire environment and causes shortage of landmarks. Therefore, the feature detection method should give as many point and line features as possible.



(a)

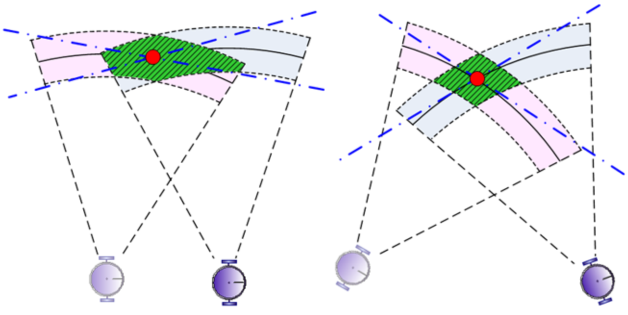
(b)

Fig. 1 Remaining problems of the TBF algorithm: (**a**) inaccurate position of a point feature due to uncertain sensor data and (**b**) generation of a false point feature along a wall

- It should be applicable to any sensor configuration, such as a rotating ultrasonic sensor module, a ring arrangement and other various arrangements.
- It should be computationally efficient to guarantee real-time implementation. Thus, a voting scheme and a delayed decision by using accumulated sensor data are not suitable. The sonar features need to be determined from the current sensor data and that of the previous few steps, if possible.

The representative methods to extract sonar features, RCD (Leonard and Durrant-Whyte 1991), Hough transformation (Tardós et al. 2002) and TBF (Wijk and Christensen 2000) can be evaluated by the above conditions. The TBF algorithm, which gives a framework for point feature detection in an indoor environment, is the most noticeable one among them because it satisfies most of the prerequisite conditions although there is room for improvement. Several weak points of TBF to be resolved for SLAM implementation are as follows.

- The location of obtained point features can be easily affected by range and angular errors of ultrasonic sensors. The TBF algorithm can not extract point features on the same location (corner) repeatedly in the condition of Fig. 1(a), so it limits the performance of SLAM estimation.
- There is no way to remove false point features which are extracted from line segments by the above sensor errors (Fig. 1(b)). These features increase the number of unnecessary landmarks for SLAM and, as a result, its computational burden.
- Extracting line features was not mentioned in the TBF framework.



(a) (b)

Fig. 2 Stability of intersections: (**a**) unstable intersection (*large intersection area*) and (**b**) stable intersection (*small intersection area*)

Therefore, we propose a modified TBF (mTBF) that can improve on the weak points of the original TBF framework while preserving its advantageous properties.

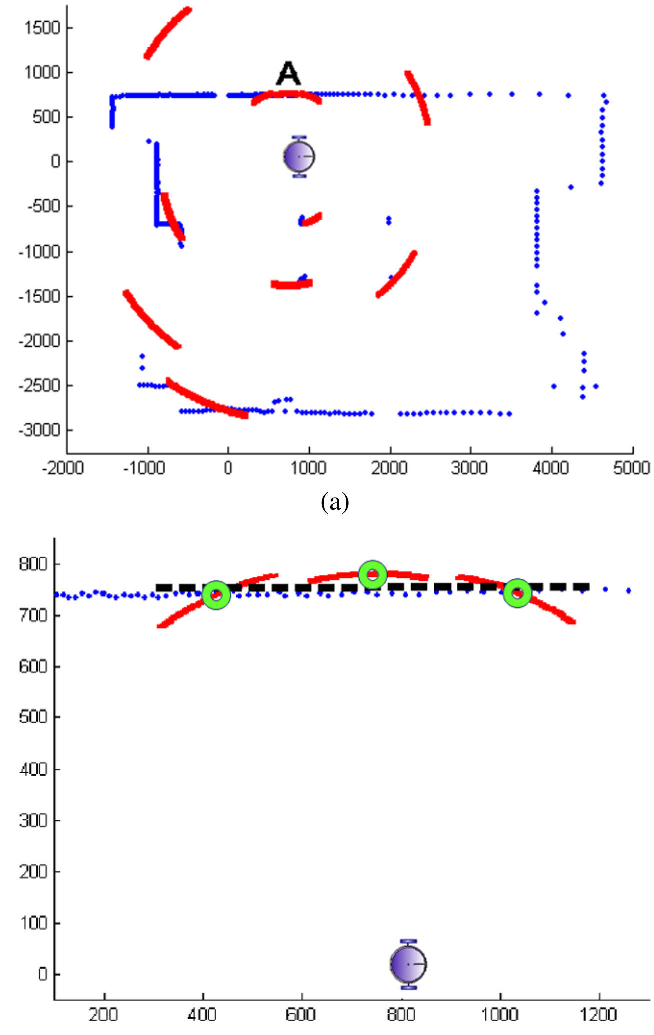
2.1.2 Modified TBF algorithm

The original TBF algorithm is a point feature detection method. Its main process is as follows.

a. All sonar information is represented as arcs with their own aperture angle.
b. Arc intersections are determined for a specific number of previous sonar arcs and the current sonar arcs.
c. If the number of the obtained intersections is more than a predetermined threshold value, the center of gravity of the intersections is selected as a point feature.

First, we propose the concept of stable intersections (Choset et al. 2003) to choose confident arc intersections from uncertain sonar data to make the original TBF algorithm suitable for our application. Here, the stable intersection means an intersection whose in-between angle of the crossing arcs is larger than a given threshold.

Stable intersections can compensate for two shortcomings of the original TBF algorithm. (1) It can improve the robustness of feature detection with the range and angular errors of ultrasonic sensors. When any kinds of intersections are allowed to extract point features, the arc intersection might be located at a large unstable intersection area (hatched area in Fig. 2(a)). The area is widely spread from a real point feature because it is significantly affected by the sensor errors. However, the stable intersection allows a small intersection area (hatched area in Fig. 2(b)) to be formed close to the real point feature. It is not significantly disturbed by the uncertainty of the sensors, so the point features thus



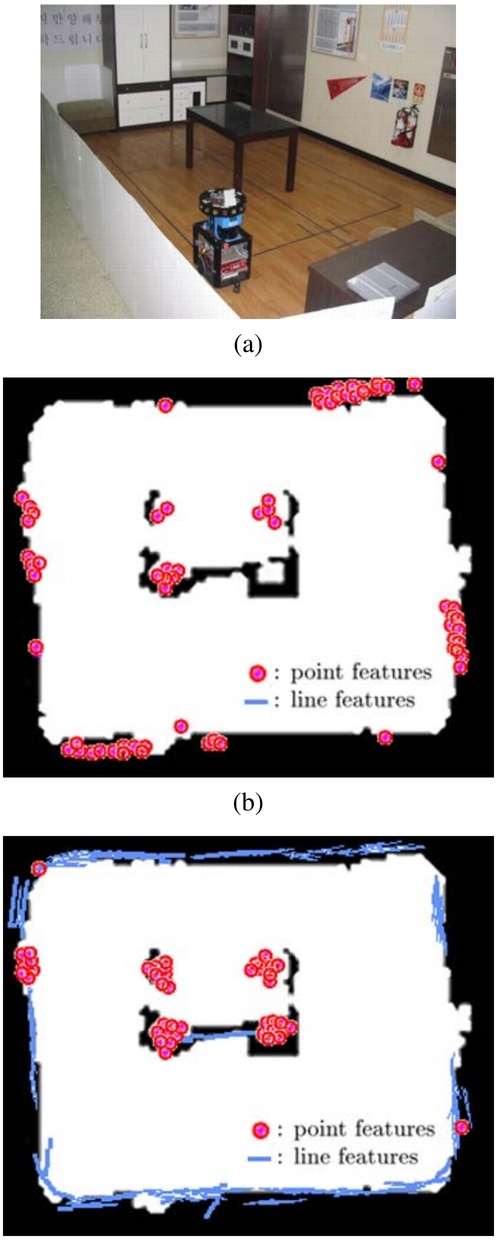
(a)

(b)

Fig. 3 (**a**) Presentation of ultrasonic sensor data (*arcs*) and laser sensor data (*dots*) with respect to a robot position in the environment shown in Fig. 4(a) and (**b**) extraction of a line feature (*dashed line*) using three adjacent sonar data (magnifying part A of (**a**))

can be extracted robustly. (2) This also filters false point features on walls. Because of the geometric configurations of sonar arcs along walls, the arc intersections obtained from walls should be closer to the unstable intersections than the stable ones (Fig. 1(b)). Consequently, the false point features can be effectively removed.

Second, we propose a line feature detection scheme. The proposed method can be combined simply and easily with the original framework by considering basic characteristics of ultrasonic sensors. We use the following properties of sonar data reflected by a line segment in order to classify a line feature: (1) Three adjacent ultrasonic sensors have similar range readings. (2) The middle one has minimum range among them (Fig. 3(b)).



(a)

(b)

(c)

Fig. 4 (**a**) Home-like environment to verify the performance of the modified TBF (mTBF), and two resulting maps: (**b**) remaining false point features along walls by the original TBF algorithm and (**c**) exact point features and well-aligned line features by the modified TBF algorithm

Therefore, line features can then be extracted by

a. representing the three adjacent sonar arcs as three points (Fig. 3(b)) using the centerline sensor model (Choset et al. 2003)
b. executing least square line fitting with the obtained points and
c. determining the length of the extracted line feature by the aperture angle of the ultrasonic sensor (dashed line)

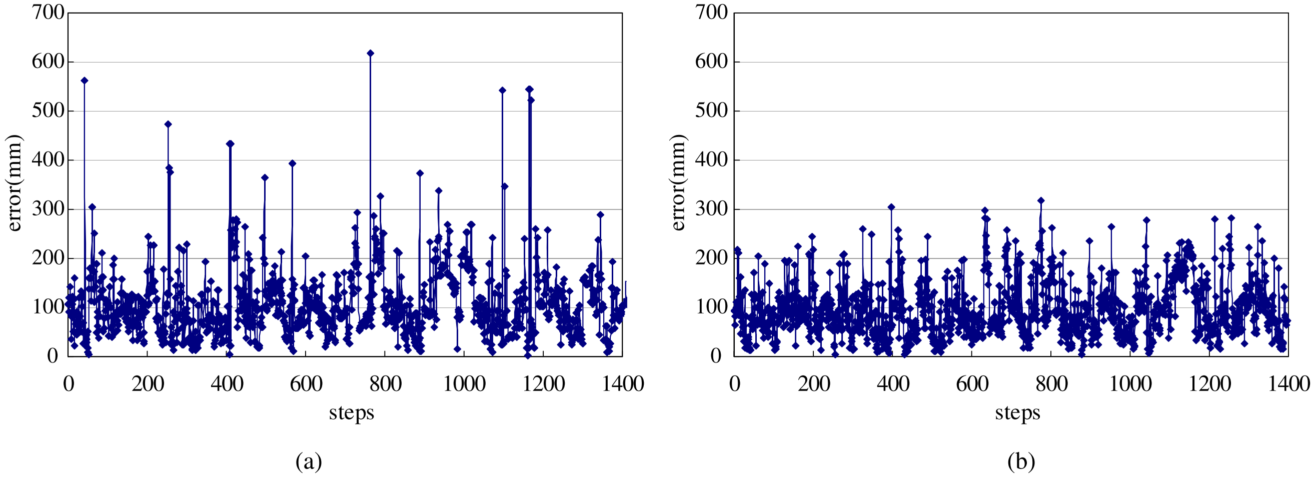


Fig. 5 Comparison of position errors of estimated point features extracted from the four legs of table: (**a**) the original TBF and (**b**) the modified TBF

The line features improve the overall performance of the TBF algorithm in two ways. (1) It can filter out false point features on walls in advance, as the stable intersection does. The false point features on walls can be removed by ignoring sensor data forming a line feature before arc intersections are determined. Here, the ignored sensor data have a characteristic that three adjacent ultrasonic sensors have similar ranges in front of a line segment. (2) Adding the line feature detection method to the TBF framework can naturally improve SLAM results because it gives more features than the original TBF algorithm. Moreover, the resulting map using the line features can represent a real environment better than a map using only point features.

2.2 Experimental verification of sonar feature detection

An experiment was executed in a home-like environment (Fig. 4(a)) to verify the performance of the modified TBF (mTBF). After the robot moved along the same path twice by wall-following using ultrasonic sensors, the resulting sonar features of the original TBF and the mTBF were compared by drawing them on a common sonar grid map. The original TBF gave many false point features spread along the walls (Fig. 4(b)). On the other hand, the mTBF extracted point features from the legs of the table and the corners of furniture exactly (Fig. 4(c)) and the number of their observations for successful feature detection was almost doubled. Moreover, the mTBF found line features that were well-aligned to real line elements instead of detecting false point features. These line features can supplement landmarks for SLAM. This shows that the mTBF method can improve the practical capability of ultrasonic sensors in an indoor environment.

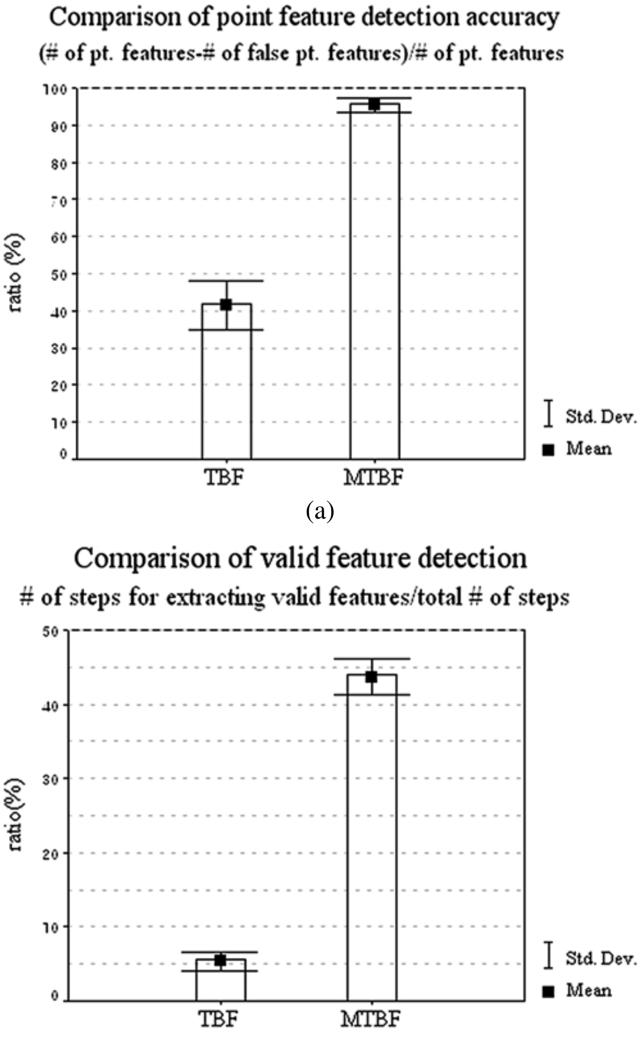
The same experiment in the environment (Fig. 4(a)) was repeated ten times in order to accurately compare performance, and the resulting sonar features were analyzed into three different ways.

First, we compared position errors of the extracted point features. The errors were calculated as the difference between real and estimated feature locations of the four table legs. The original TBF algorithm produced more outliers (Fig. 5(a)) than mTBF (Fig. 5(b)). These outliers, which might come from range and angular errors of the ultrasonic sensors, had quite large inaccuracies (300∼700 mm). They can lower the performance of SLAM estimation by increasing the chance of false data association.

Second, we investigated the incidence rate of correct point features (Fig. 6(a)) to confirm the ability to remove false point features along walls due to the unstable intersections of sonar data. The original TBF could not discriminate false point features, and only 41.5% of the extracted point features were correct ones. On the other hand, the ratio of correct point features via mTBF was 95.5%.

Finally, Fig. 6(b) shows the ratio of the number of valid features to total steps of sampling sonar data in order to check the observation frequency of both sonar features for SLAM update. The ratio was 5.4% and 43.7% for the original TBF and the mTBF, respectively. The mTBF increased the number of observations of the sonar features almost eight times over that of the original TBF. This should allow more frequent SLAM update with the abundant sonar features.

The above three comparisons verify the effectiveness of the proposed sonar feature detection scheme. The improved capability of the mTBF can help to improve SLAM performance with more accurate and frequent SLAM updates.



(a)

(b)

Fig. 6 Comparison of (**a**) the rate of correct point feature detection to show how the modified TBF can remove false point features along walls effectively and (**b**) the ratio of detecting valid features over all the sampled data of both methods

3 Visual object recognition for VR-SLAM

In addition to the above sonar features, we propose a method to handle visual features effectively in this section: extracting suitable visual features for object recognition in an indoor environment, grouping the features into a visual object and treating the object as a visual landmark in VR-SLAM.

3.1 Salient visual features for object recognition in an indoor environment

3.1.1 Motivation of visual features

Salient visual features extracted from images are used in many vision applications such as image matching, stitching, object recognition and so forth. When the salient visual features are applied to object recognition, object models composed of the visual features should be stored in a database beforehand (Lowe 2004). After the object database is built, an object can be discriminated from the others by clustering the visual features corresponding to the same object in the database.

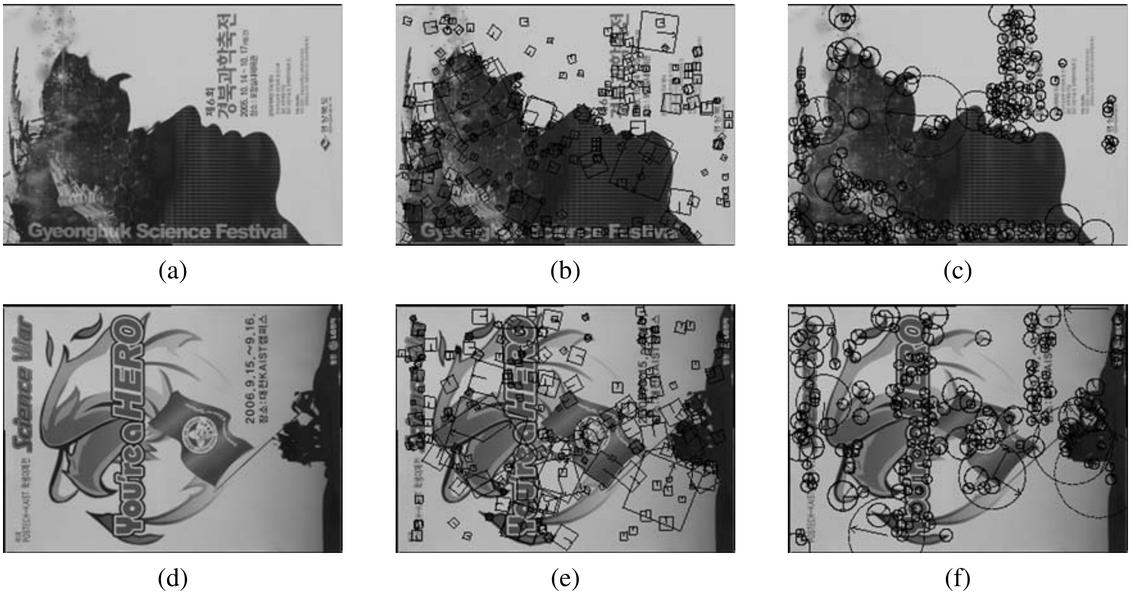
Robust object recognition requires that the visual features should be invariant to many variables such as translation, rotation, scale, viewpoint, illumination changes and occlusion. One of the most popular salient features satisfying the aforementioned conditions is SIFT (Scale Invariant Feature Transform) (Lowe 2004). First, it detects interest points (SIFT detectors) which are local extremes of Difference-of-Gaussian (DoG) images through location and scale space. Then, SIFT descriptor vectors are constructed to describe local gradient information around the detectors and their dominant orientations are determined. The SIFT feature is invariant to image translation, rotation, scaling and partially invariant to illumination changes, affine and projective transformations. These properties make it widely utilized in many object recognition applications.

Despite these excellent properties, however, when the SIFT detectors are generated from objects, their blob-like feature representation has many chances to extract less-informative features which have similar local information around the detectors. As shown in Fig. 7(b) and (e), we extracted 250 SIFT detectors from the original images (Fig. 7(a) and (d)), respectively. Due to the use of the DoG images, they were also detected on the single-tone background of images where it has less visually distinctive information. Because these less-informative features have similar local information, they increase the chance to make outliers of feature matching and lower the success rate of object recognition, which, in turn, affects SLAM performance eventually. Moreover, large numbers of the less-informative features increase the computational load of the recognition.

Therefore, when the robot recognizes objects in an indoor environment for SLAM, a method which effectively reduces the less-informative features is required for correct data association.

3.1.2 Multi-scale Harris corners as detectors

Multi-scale Harris corner (MSHC) is an adequate visual feature of object recognition for SLAM due to the following reasons. It is as invariant as the SIFT feature against rotation, scale, affine and illumination changes (Lin et al. 2005). Moreover, it is invariant to viewpoint changes of a planar motion to a considerable extent (Fig. 8), which could happen in the proposed indoor SLAM method using visual objects. The performance of the MSHC and SIFT detectors was also evaluated with a repeatability criterion using INRIA Graffiti data set (Mikolajczyk and Schmid 2004). It matched

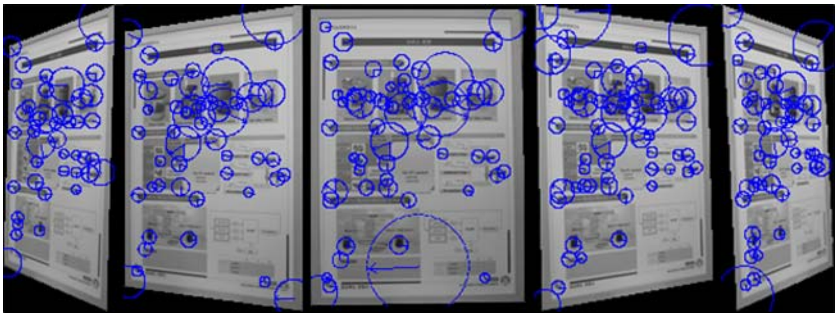


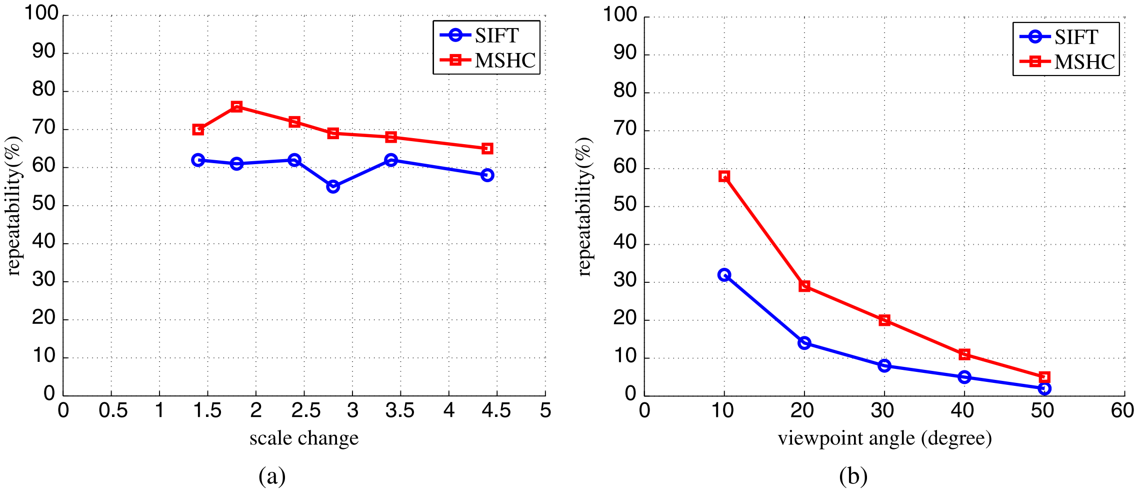
(a) (b) (c)

(d) (e) (f)

Fig. 7 (**a**), (**d**) Original images and their 250 visual features of (**b**), (**e**) SIFT detectors (*squares*) and (**c**), (**f**) Multi-scale Harris corners (*circles*)

Fig. 8 High repeatability of multi-scale Harris corners with respect to various viewpoint changes (−60°, −30°, 0°, 30°, 60°)

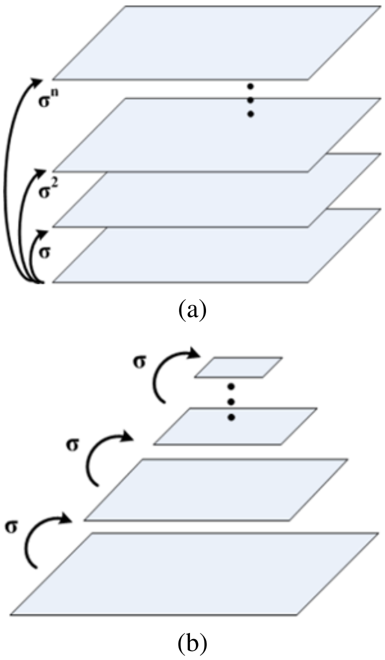




(a) (b)

Fig. 9 Repeatability of the MSHC and SIFT detectors with respect to (**a**) 6 scale changes and (**b**) 5 viewpoint changes

the corresponding features between images by changing 6 scales and 5 viewpoints (Fig. 9). And the 250 MSHC detectors were extracted from the same images (Fig. 7(a) and (d)) to compare with the SIFT detectors. They were mainly located on a visually informative points as shown in Fig. 7(c) and (f) because of their structure-like (corner-like) feature representation and effectively reduced the number of less-informative features than the SIFT detectors.



(a)

(b)

Fig. 10 Comparison of: (**a**) a successive smoothing process and (**b**) a sub-sampling process for a scale representation

The extraction process is as follows. First, a Harris corner is determined from the scale-adapted second moment matrix at each pixel **x** in an image

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}, \quad (1)$$

where σ_I is the integration scale, σ_D is the differentiation scale, L_x and L_y are the gradients of the smoothed image with a Gaussian kernel of size σ_D in column and row direction, respectively. And $g(\sigma_I)$ is a smoothing with a Gaussian window of size σ_I. If a Harris measure,

$$\lambda = \det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) - \alpha \mathrm{tr}^2(\mu(\mathbf{x}, \sigma_I, \sigma_D)) \quad (2)$$

at a point is bigger than a given threshold value, the point can be classified as a Harris corner. Subsequently, the Harris corners are extracted from each scale space by changing the integration and the differentiation scales. Finally, multi-scale Harris corners, which have local maxima of the Harris measures in scale space, are chosen among the extracted Harris corners.

However, the original multi-scale Harris corner has a significant weak point in terms of computational efficiency because it requires a successive smoothing process of the given image with Gaussian kernels for a scale representation (Fig. 10(a)). So we replaced the scale space representation with a pyramid representation which uses a sub-sampling process for representing different scale changes (Fig. 10(b)). The pyramid representation can reduce the computational time by half and guarantee 2 Hz extraction for a real-time implementation of the multi-scale Harris corner using a 2.0 Ghz Pentium-IV.

3.1.3 SIFT descriptors

After the multi-scale Harris corners are extracted, descriptor vectors to imply local characteristic around each detector point should be determined. Zernike moment can be used as the descriptor of the multi-scale Harris corner, as in Lin et al. (2005), however it has a large computational burden and low discriminating capability. We use the SIFT descriptor (Lowe 2004) for a real-time implementation instead; it is the state-of-the-art descriptor invariant to various image changes (Mikolajczyk and Schmid 2005).

First, gradient vectors are computed by neighbor pixels of the detector within the area covered by its scale. The dominant direction of the gradient vector set determines the orientation of the detector. Then, a descriptor vector which has 128 elements reflecting local information around the detector is calculated from the histogram of direction and magnitude of the gradient vector set.

3.2 Construction of object database

Extraction of physical objects by segmenting out the background region from images with a previously constructed database is widely used for object recognition in a cluttered environment. We also construct an object database in advance as an off-line process to implement SLAM with visual object recognition. First, we take a picture of an object in order to construct an object database. Then the object is segmented manually by selecting its vertices (left: four vertex points in Fig. 11). The resulting object database consists of the proposed visual features (right: circles) extracted from the segmented image along with a record of their row and column positions in the image coordinate, orientations, scales and descriptor vectors as shown in Fig. 11. Also it has row and column position of the center point (left: cross) of the object to define a point landmark in SLAM.

Our database has the segmented images with three different scales and five different viewpoints (fifteen images for one object, Fig. 12). It helps data association of visual objects to cope with image changes that exceed inherent invariant abilities of the proposed visual features (MSHC detectors and SIFT descriptor). Consequently, the database can adapt to the large scale and viewpoint changes which frequently happen to SLAM in an indoor environment. It can maximize the advantages of using pre-constructed model database in a practical aspect.

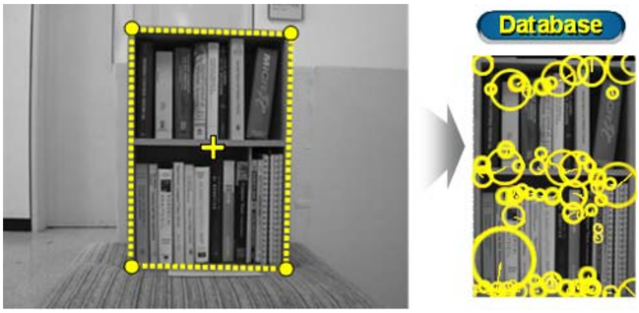


Fig. 11 Construction of an object database which includes vertices (*left*: four points), local invariant features (*right*: circles) and center position (*left*: cross) of an object

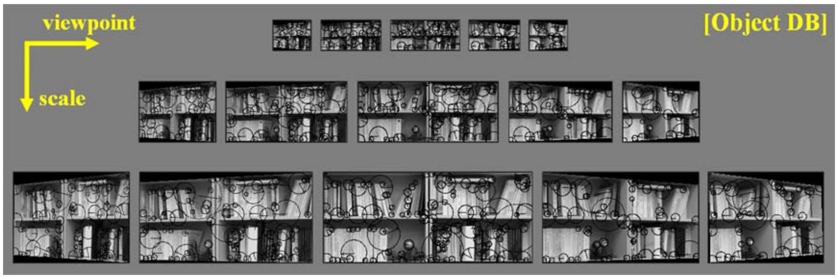


Fig. 12 Construction of an object database which has three different scale images with five viewpoint changes

The robot only knows what kinds of the objects exist in the given environment during construction of the database, but information on their absolute spatial position is not recorded in advance. After data association is achieved by the object recognition method, the position of the visual objects and the robot will be estimated simultaneously by solving the SLAM problem.

3.3 Object recognition based on RANSAC clustering

First of all, object recognition should match the visual features of a current scene to those of the object database by means of comparing the descriptor vectors. A k-d tree structure (Lowe 2004) is adopted to construct the database for speeding up data retrieval time from the database. Because of this, the matching process of the object in a current scene with a large database can be done efficiently by using approximate nearest-neighbor search based on the k-d tree.

Although the SIFT descriptor is highly specific and invariant to image variations, outliers still remain when only Euclidean distance is used for matching between descriptor vectors of similar values. It excludes any information of geometric relation between the matched pairs of the visual features, so that the remaining outliers can degrade matching performance.

Therefore, we propose a method to geometrically constrain the matched features of the current scene into the same object in order to resolve the above outlier problem. We basically assume that the database is composed of planar objects. In this case, the geometric constraint is based on the fact that there exists a non-degenerate homography transformation between the object database and the object in the current scene. This means that the probability to find a group of the features satisfying the geometric constraint can be increased by proper estimation of the homography transformation between the matched features,

$$\mathbf{X}_{s,i} = \mathbf{M}_{o\to s}\mathbf{X}_{o,i}, \tag{3}$$

where $\mathbf{X}_{s,i}$ is the position of the ith matched point in the scene image, $\mathbf{X}_{o,i}$ is the position of the ith matched point in the object model image and $\mathbf{M}_{o\to s}$ is a homography transformation from the object database to the scene image.

RANSAC clustering (Fischler and Bolles 1981) improves the homography transformation estimates in a systematic way. It recovers eight parameters of the homography transformation between the matched pairs. An error measure for the clustering procedure uses a dual error measure,

$$d = \frac{1}{2n}\sum_{i=1}^{n}\Big[\|\mathbf{M}_{o\to s}\mathbf{X}_{o,i} - \mathbf{X}_{s,i}\| + \|\mathbf{M}_{s\to o}\mathbf{X}_{s,i} - \mathbf{X}_{o,i}\|\Big], \tag{4}$$

where n is the number of clustered features and $\mathbf{M}_{s\to o}$ is a homography transformation from the scene to the object image. Consequently, the homography parameters are obtained and the matched features can be grouped into the same recognized object.

Figure 13 presents the performance of the proposed object recognition algorithm in a cluttered environment. It also shows the capability of detecting multiple objects simultaneously and adapting to partial occlusion of the object. Furthermore, the proposed scheme of object recognition works well with considerable changes of scale and illumination, as shown in Fig. 14.

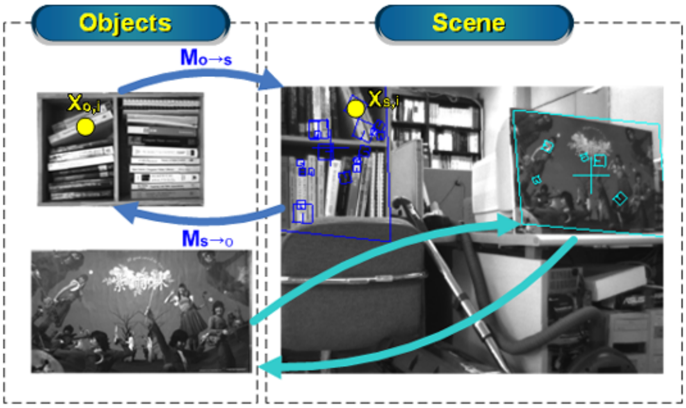


Fig. 13 Result of object recognition by descriptor matching and RANSAC clustering (*Left*: an object database, *Right*: a current scene with recognized objects)

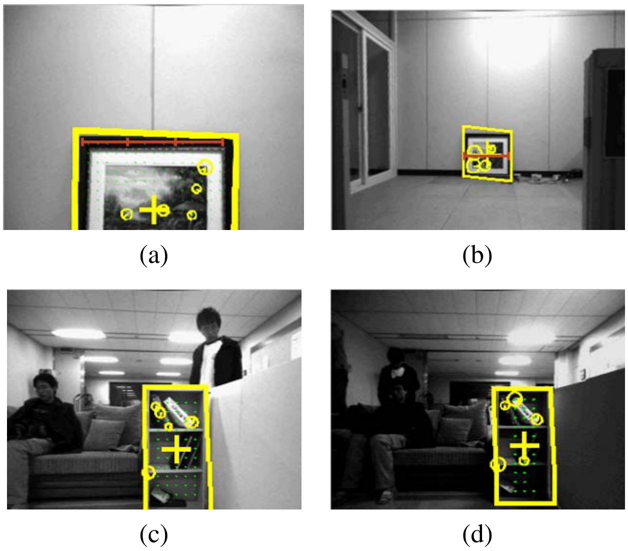

(a)
(b)
(c)
(d)

Fig. 14 Performance of object recognition in (**a**), (**b**) three-times scale change and (**c**), (**d**) illumination change

3.4 Getting accurate metric distance to the recognized object

When the robot uses visual objects in SLAM, the excellent discriminant ability of object recognition should be decisive in correct data association. Additionally, relative range and bearing information of the recognized object with respect to the robot needs to be accurate for the purpose of dealing with the objects as point landmarks directly in the SLAM state.

Thus, a stereo camera is used to get metric distance to the recognized object immediately. The center point of the object is assigned as a representative point for an observation and then regarded as a point landmark.

When the stereo camera measures the distance of the center point directly, it is easy to obtain an accurate position, (x_m, y_m) in the camera coordinate x and y parallel to the image plane. On the other hand, depth information, z_m, has an uncertainty that comes from unreliable stereo matching. This is also a problem even when we use a comparatively accurate stereo camera such as Bumblebee. The outliers caused by the false stereo matching make it difficult to get accurate depth information. Therefore, we propose a method to overcome this problem by using distance information of other points within the recognized object as well as the center point to increase the accuracy of the distance calculation.

First, the aforementioned planar assumption of the visual object is used again to get more accurate depth information of the center point. We use the fact that pixels which have accurate distance information within the recognized object can be located on the same plane, but inaccurate outliers would not. Therefore, sample points which have their distance information are randomly chosen within the recognized object and then the RANSAC clustering is applied with the points again to estimate a common planar equation such as $ax + by + cz + d = 0$, where $x,\ y,\ z$ are the position of points within the recognized object in the camera coordinate and $a,\ b,\ c,\ d$ are the parameters of a 3D planar equation. The 3D planar equation of the object can be obtained as a result of RANSAC clustering and the center point, (x_m, y_m, z_m) in the camera coordinate should belong to the obtained 3D plane (5).

$$ax_m + by_m + cz_m + d = 0. \tag{5}$$

Second, the center point in the image coordinate of current scene, (r_m, c_m) can be estimated through the obtained homography transformation, $\mathbf{M}_{o\to s}$ in (3). The relation between the center point in the image coordinate and the corresponding point in the camera coordinate is determined by the camera model as follows:

$$r_m = v_o + f\frac{y_m}{z_m} \quad\Rightarrow\quad y_m = \frac{r_m - v_o}{f}z_m, \tag{6}$$

$$c_m = u_o + f\frac{x_m}{z_m} \quad\Rightarrow\quad x_m = \frac{c_m - u_o}{f}z_m, \tag{7}$$

where (u_o, v_o) is a camera center point in the image coordinate and f is the focal length of the camera.

Finally, (5), (6) and (7) can determine the depth information of the center point of the recognized object,

$$z_m = -d/\left(a\frac{c_m - u_o}{f} + b\frac{r_m - v_o}{f} + c\right). \tag{8}$$

Therefore, the depth information obtained from the RANSAC plane equation can naturally be more accurate than the directly acquired one from the disparity of stereo matching. The accurate metric information and thus correct data association are very helpful and effective for convergent and consistent EKF-SLAM estimation, which will be described in the next section.

4 EKF-SLAM frameworks for VR-SLAM

We implement localization and map building based on two kinds of EKF-frameworks, which are the standard EKF-SLAM (Dissanayake et al. 2001) and the hierarchical SLAM (Estrada et al. 2005). In both case, we assume the planar motion of a robot in an indoor environment. As described earlier, the complementary combination of sonar features and visual objects should enhance their performance.

4.1 Standard EKF-SLAM: global map approach

We use point and line features of ultrasonic sensors and visual objects of vision sensor at the same time in a standard EKF-SLAM framework. The visual objects used as point landmarks have smaller measurement covariances than the sonar features to reflect the outstanding reliability of the visual objects.

The standard EKF-SLAM composed of prediction and update stages can estimate the SLAM state (9) and its covariance matrix (10) in a recursive way:

$$\mathbf{x}(k) = [\mathbf{x}_v^T \quad \mathbf{p}_1^T \quad \cdots \quad \mathbf{p}_N^T]^T, \tag{9}$$

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}, \tag{10}$$

where $\mathbf{x}_v$ is the robot pose, $\mathbf{p}_i$ is the ith landmark position, $\mathbf{P}_{vv}$ is the robot covariance submatrix, $\mathbf{P}_{vm}$ is the covariance between robot and landmarks, and $\mathbf{P}_{mm}$ is the landmark covariance.

The prediction stage is performed by a kinematic model of the robot, $\mathbf{f}(\cdot)$, and the wheel velocity between $k-1$ and k computed from odometry data, $\mathbf{u}(k-1)$, in (11). The covariance is propagated by the Jacobian of the prediction model, $\mathbf{F}(k-1)$, and the covariance of control input, $\widetilde{\mathbf{Q}}(k-1)$, in the prediction step (12):

$$\hat{\mathbf{x}}_v^-(k) = \mathbf{f}(\hat{\mathbf{x}}_v^+(k-1), \mathbf{u}(k-1), k), \tag{11}$$

$$\mathbf{P}^-(k) = \mathbf{F}(k-1)\mathbf{P}^+(k-1)\mathbf{F}^T(k-1) + \widetilde{\mathbf{Q}}(k-1). \tag{12}$$

Subsequently, the update stage is accomplished by the innovation (14), the difference between the predicted observation (13) and the real observation, $\mathbf{z}(k)$, and the innovation covariance (15):

$$\hat{\mathbf{z}}(k) = \mathbf{h}(\hat{\mathbf{x}}_v^-(k), \hat{\mathbf{p}}^-(k)), \tag{13}$$

$$\mathbf{e}(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k), \tag{14}$$

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}^-(k)\mathbf{H}^T(k) + \widetilde{\mathbf{R}}(k), \tag{15}$$

where $\mathbf{h}(\cdot)$ is an observation model, $\mathbf{H}(k)$ is the Jacobian of the observation model and $\widetilde{\mathbf{R}}(k)$ is the measurement covariance. The observation model of a point landmark is a distance to the location of the point landmark with respect to the predicted robot position. A line landmark uses the foot of perpendicular line from the predicted robot to the line.

The update step for the state and the covariance is accomplished by the innovation as follows:

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{W}(k)\mathbf{e}(k), \tag{16}$$

$$\hat{\mathbf{P}}^+(k) = \hat{\mathbf{P}}^-(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}(k), \tag{17}$$

where

$$\mathbf{W}(k) = \mathbf{P}^-(k)\mathbf{H}^T(k)\mathbf{S}^{-1}(k) \tag{18}$$

is the Kalman gain.

4.2 Hierarchical SLAM: local map approach

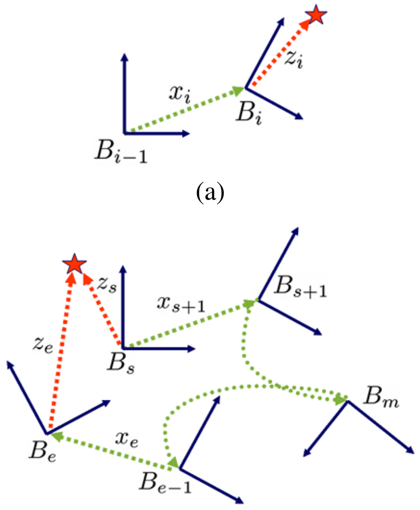
Hierarchical SLAM is a local map approach to overcome linearization error in a large environment and highly uncertain odometry. It can reduce the computational cost of the standard EKF-SLAM while maintaining the consistency of the estimation.

The proposed method using both sonar features and visual objects should be very effective in a hierarchical SLAM framework, since both features have different characteristics as landmarks. A local map is created by a set of the sonar features because they have high update frequency for SLAM. On the other hand, we rely more on information from the visual objects than the sonar features for maintaining global consistency between the generated local maps due to the excellent discriminant ability of the visual objects.

4.2.1 Construction of local map

Creating a local map is similar to the standard EKF-SLAM approach using the proposed point and line features of ultrasonic sensors and the visual objects. Each local map has one or more visual objects for its identification to impose a successful loop closure.

During the SLAM estimation, mutually independent local maps are sequentially built as follows. Once a visual object which is different from one in the current local map is



(a)

(b)

Fig. 15 (**a**) Constructing a new local map based on B_i which has a measurement, z_i to the recognized visual object and (**b**) closing a loop linking the reference base frame B_s to B_e

detected, the current local map is closed and a new local map is created. The resulting new ith local map has base reference frame, B_i, and measurement of the visual object, z_i, as shown in Fig. 15(a). When the robot detects more than one visual object at the same scene, it registers all of them as the representatives of the local map with respect to the base reference frame, B_i. Then, the relative transformation, x_i, between the base references of the previous local map, B_{i-1}, and the current one, B_i, is kept separately, which should be updated to maintain the global consistency of the local maps.

The proposed method of dividing local maps depends on distinct visual objects, unlike usual approaches relying on the number of features, the size of robot uncertainty or spatial separation.

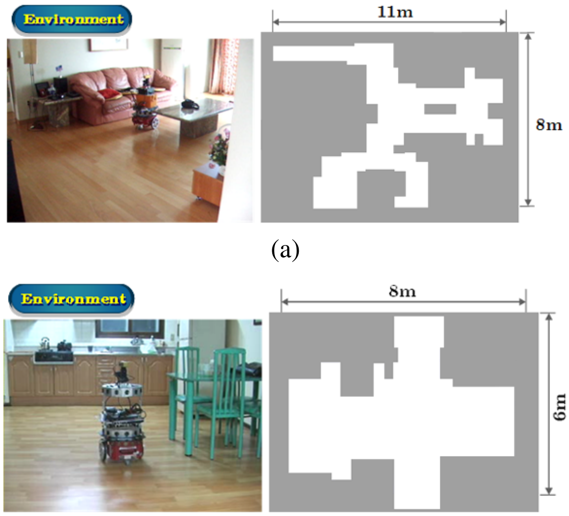
4.2.2 Maintenance of global consistency

After sequential local maps are constructed, the unconstrained relative transformation between the local maps are represented as $\mathbf{x}_u$, where

$$\mathbf{x}_u = [\,\mathbf{x}_1 \quad \dots \quad \mathbf{x}_{s+1} \quad \dots \quad \mathbf{x}_m \quad \dots \quad \mathbf{x}_e \quad \dots \quad \mathbf{x}_n\,]^T \quad (19)$$

and its corresponding covariance matrix is $\mathbf{P}_u$. When a loop linking a base reference frame B_s to B_e is detected by visual object recognition (Fig. 15(b)), a constraint can be generated between local maps and measurements of the visual objects as follows:

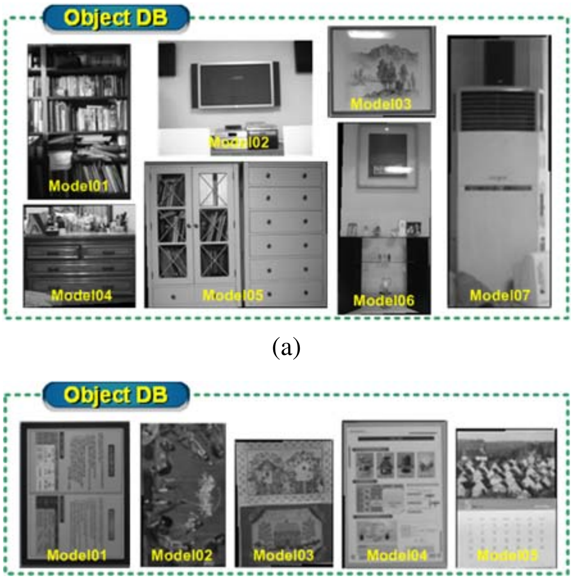
$$\mathbf{h}(\hat{\mathbf{x}}) \equiv \hat{\mathbf{x}}_{s+1} \oplus \cdots \oplus \hat{\mathbf{x}}_e \oplus \mathbf{z}_e = \mathbf{z}_s. \quad (20)$$



(a)

(b)

Fig. 16 Various home environments and their floor plans: (**a**) 1 and (**b**) 2



(a)

(b)

Fig. 17 Object model databases: (**a**) 1 and (**b**) 2

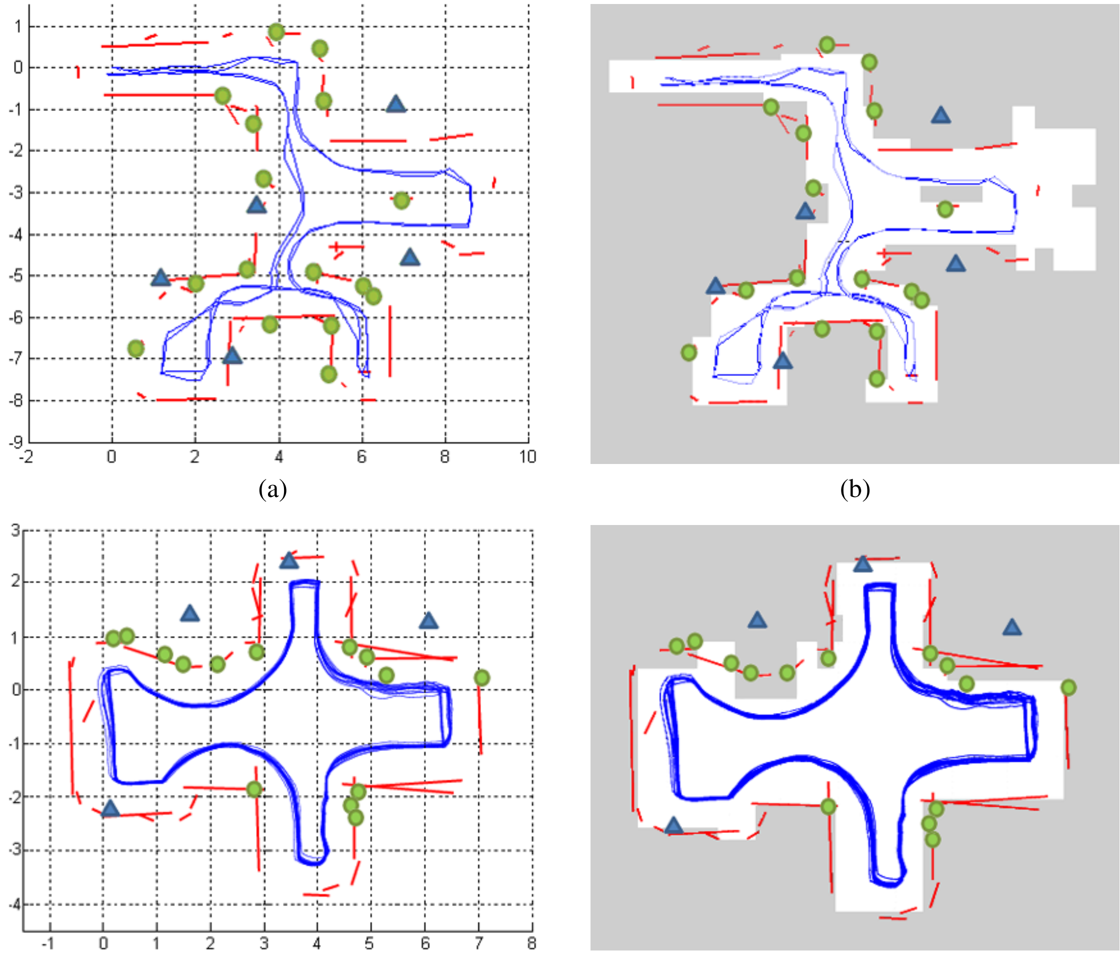
We solve the constrained optimization problem for maintaining global map consistency, as in hierarchical SLAM:

$$\min_{\hat{\mathbf{x}}} \frac{1}{2}(\hat{\mathbf{x}} - \mathbf{x}_u)^T \mathbf{P}_u^{-1}(\hat{\mathbf{x}} - \mathbf{x}_u) \quad (21)$$

based on the loop constraint (20) via iterated extended Kalman filter (IEKF).

The IEKF approach can produce the following iterative equation:

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + \mathbf{P}_u\mathbf{H}_i^T\left(\mathbf{H}_i\mathbf{P}_u\mathbf{H}_i^T + \mathbf{R}\right)^{-1}$$



(a) (b)

(c) (d)

Fig. 18 Experimental results using the standard EKF-SLAM: (**a**) and (**c**) are the result of SLAM estimation in the environment 1 and 2, respectively. (**b**) and (**d**) are the overlay of the SLAM results on the corresponding floor plans (Fig. 16). (*triangle*: visual object, *circle*: sonar point feature, *line segment*: sonar line feature, *center loop*: estimated robot path)

$$\times \left[\mathbf{H}_i \left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_u \right) + \left(\mathbf{z}_s - \hat{\mathbf{h}}_i \right) \right], \tag{22}$$

where

$$\mathbf{H}_i = \left[\left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_1}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_2}\right|_{\hat{\mathbf{x}}_i} \cdots \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_{n-1}}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_n}\right|_{\hat{\mathbf{x}}_i} \right] \tag{23}$$

is the Jacobian of **h** whose terms are represented by

$$\begin{aligned} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_m}\right|_{\hat{\mathbf{x}}_i} &= \left.\frac{\partial \mathbf{h}}{\partial (\mathbf{x}_{s+1} \oplus \cdots \oplus \mathbf{x}_m)}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial (\mathbf{x}_{s+1} \oplus \cdots \oplus \mathbf{x}_m)}{\mathbf{x}_m}\right|_{\hat{\mathbf{x}}_i} \\ &= \mathbf{J}_{1\oplus}((\mathbf{x}_{s+1} \oplus \cdots \oplus \mathbf{x}_m), \ominus(\mathbf{x}_{s+1} \oplus \cdots \oplus \mathbf{x}_m) \oplus \mathbf{h}) \\ &\quad \times \mathbf{J}_{2\oplus}((\mathbf{x}_{s+1} \oplus \cdots \oplus \mathbf{x}_m) \ominus \mathbf{x}_m, \mathbf{x}_m) \end{aligned} \tag{24}$$

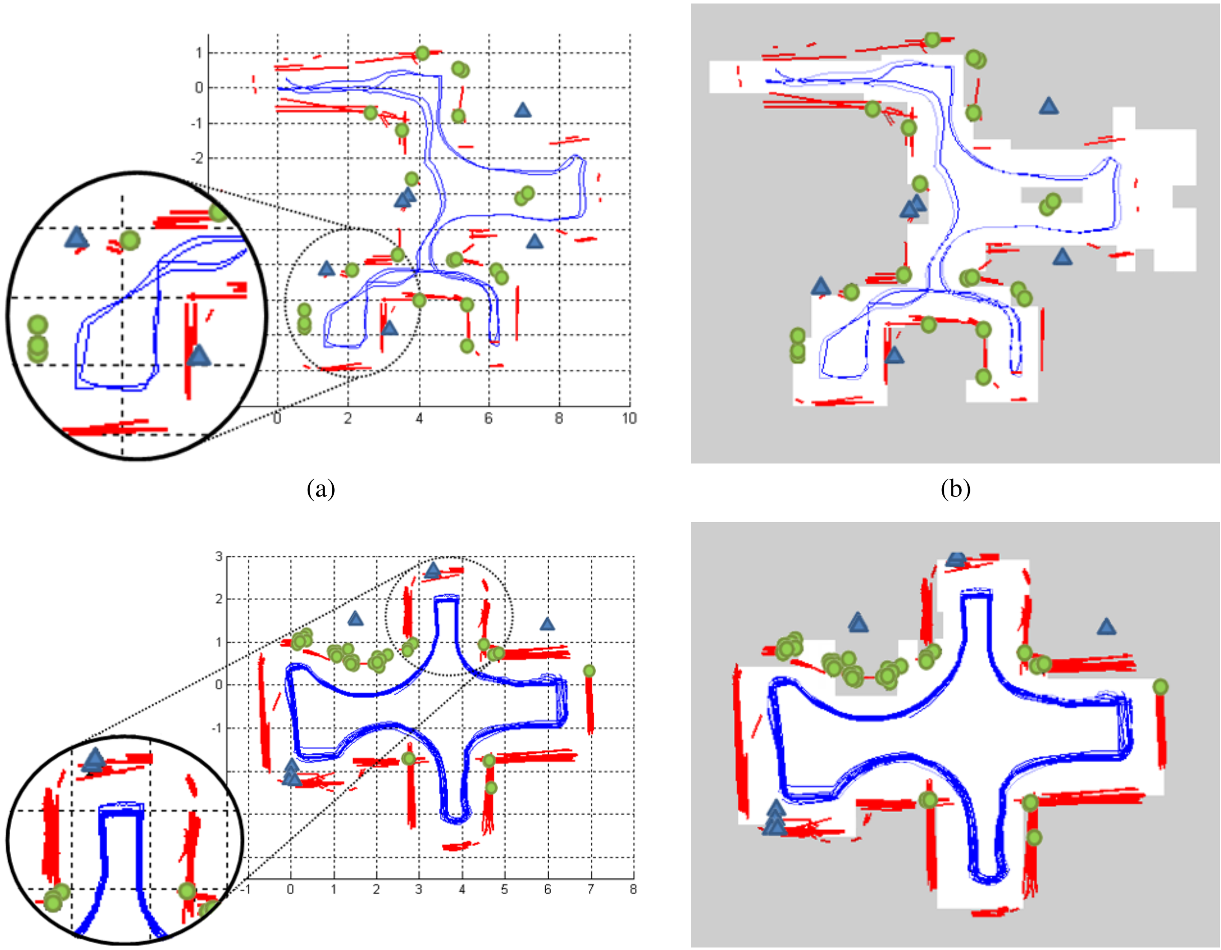
and the measurement covariance of the visual object is **R**. We iterate the IEKF equation (22) until $\hat{\mathbf{x}}_i$ is converged and, finally, we can achieve a globally consistent SLAM map, $\hat{\mathbf{x}}$, with the help of the visual objects.

5 Experimental results

Experiments were carried out using a Pioneer3-DX in three different kinds of indoor environments to verify the proposed VR-SLAM algorithm. The robot is equipped with 12 piezo-electric ultrasonic sensors from the Murata company and a Bumblebee stereo camera from Point Grey Research. It moved along the wall several times by a sensor-based navigation using ultrasonic sensors, with an average speed of about 0.15 m/s. Data acquisition of the sonar features and the visual objects was performed at almost 4 Hz and 0.5 Hz, respectively.

5.1 Home environments

Applicability of the VR-SLAM algorithm to real home environments for practical purposes and proper feature extraction of ultrasonic and vision sensors were evaluated in two environments (Fig. 16). The first experiment was executed



(a) (b)

(c) (d)

Fig. 19 Experimental results using the hierarchical SLAM: (**a**) and (**c**) are the result of SLAM estimation in the environment 1 and 2, respectively. The magnified parts of the maps show the global consistency between the resulting local maps. (**b**) and (**d**) are the overlay of the SLAM results on the corresponding floor plans (Fig. 16). (*triangle*: visual object, *circle*: sonar point feature, *line segment*: sonar line feature, *center loop*: estimated robot path)

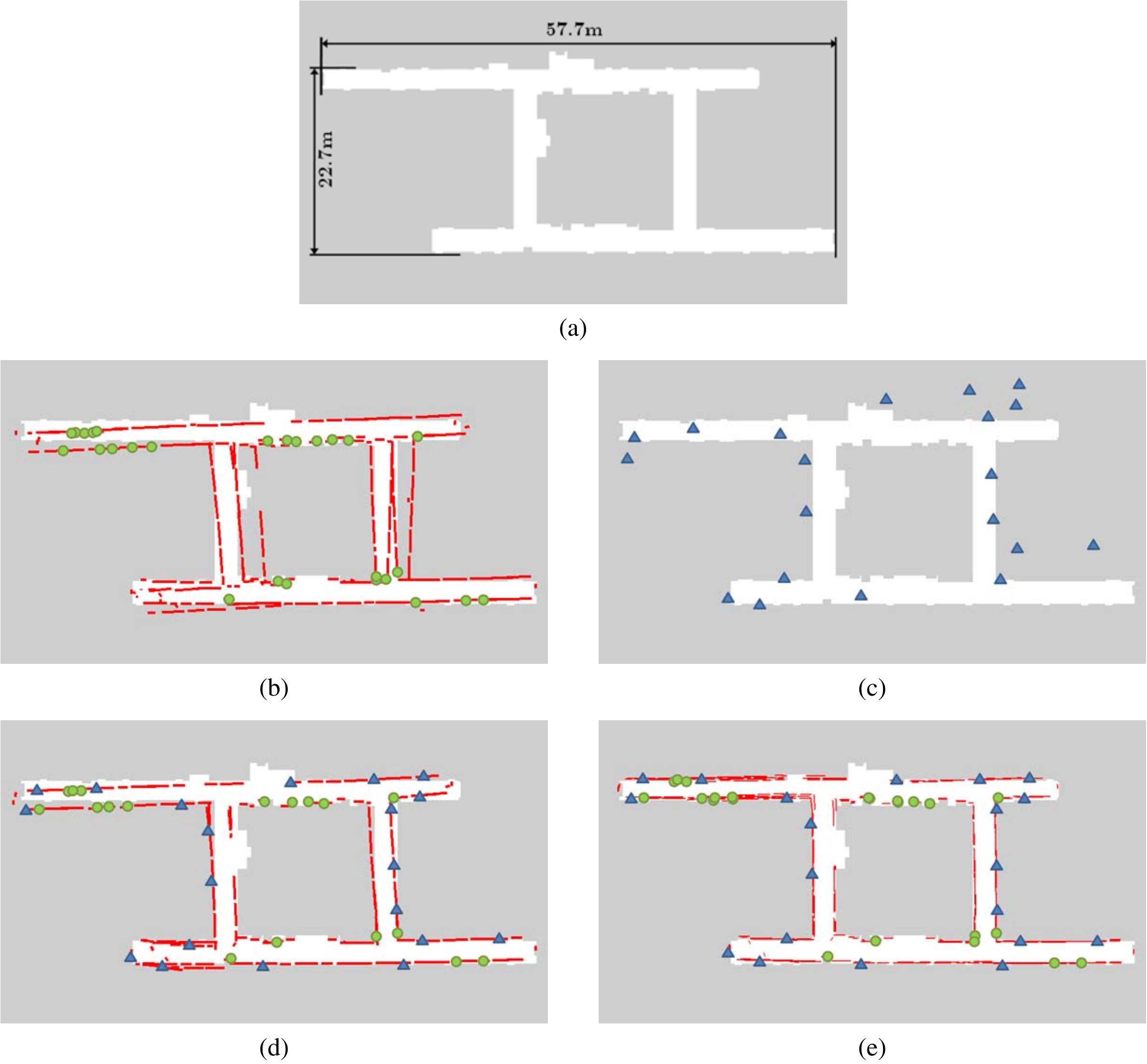
in a real apartment where a family is living (Fig. 16(a)). The robot navigated two bedrooms, a living room and a dining room in a wall following manner, and the covered area was 11 m × 8 m. And the second experiment was conducted in a different home environment (Fig. 16(b)), consisting of three bedrooms, a living room and a dining room. The robot covered just the living room and the dining room because the other rooms were separated by doorsills. The covered area was 8 m × 6 m, however, the robot performed the wall-following process continuously until the battery was exhausted after about one and half hours to verify the consistency of sustained SLAM estimation.

The visual objects used in the experiments are shown in Fig. 17. Appliances such as televisions and air conditioners, and pieces of furniture such as book shelves, drawers and picture frames which generally exist in home environments were selected. The object database was constructed using images captured by a commercial digital camera in advance. Seven (Fig. 17(a)) and five (Fig. 17(b)) planar objects were used for the object databases, reflecting the size of the different environments, which were 11×8 and 8×6 m^2, respectively. These visual objects were selected to sparsely cover the entire environment.

5.1.1 Applying to standard EKF-SLAM as global map approach

The robot moved by sensor-based navigation (wall following) and simultaneously ran the standard EKF-SLAM framework in the global reference frame. The experimental results of our VR-SLAM algorithm are presented in Fig. 18(a) and (c), where the center loop is the estimated robot path.

The resulting SLAM maps have circles and solid line segments that denote the estimated point and line features, respectively. The estimated visual objects are shown as trian-



(a)

(b)

(c)

(d)

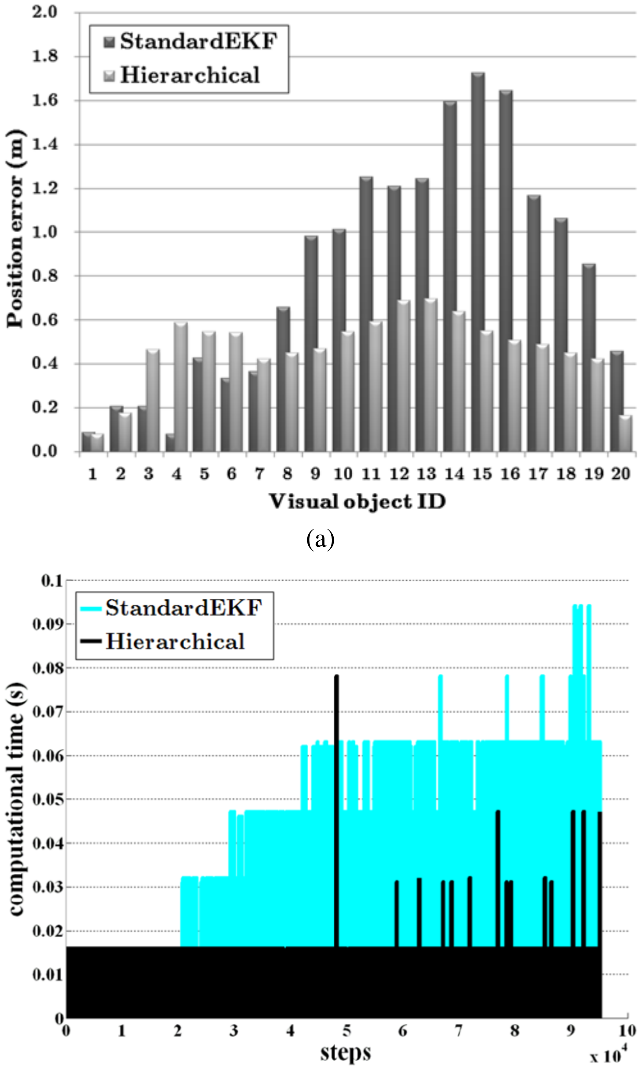
(e)

Fig. 20 Experimental results in a large building environment: (**a**) floor plan of the environment, (**b**) sonar-only SLAM, (**c**) vision-only SLAM, (**d**) standard EKF-SLAM and (**e**) hierarchical SLAM (*triangle*: visual object, *circle*: sonar point feature, *line segment*: sonar line feature, *center loop*: estimated robot path)

gles. The detected landmarks of the SLAM maps overlaid on the floor plans (Fig. 16) of the given environments. They matched well with the floor plans (Fig. 18(b) and (d)), confirming that the standard EKF-SLAM framework operated excellently in these environments in real-time. Moreover, we need to keep in mind that the results were obtained using only ultrasonic sensors and a stereo camera, which are more imprecise than laser sensors. The high discriminating ability of visual objects maintains correct data association, and the more frequent update of sonar features compensates for the low update frequency of visual objects caused by sparse observation and illumination effect.

5.1.2 Applying to hierarchical SLAM as local map approach

The standard EKF-SLAM algorithm was used to build each local map for the first step of implementing the hierarchical SLAM (Estrada et al. 2005). In every local map, sonar data were processed to obtain point and line features and a local map was created whenever one or more visual objects, which were different from the ones in the previous local map, were detected. As a result, totals of 10 and 54 local maps were generated with the databases of 7 and 5 object models in each experimental environment during 2 and 20 cycles, respectively. Ideally, the number of obtained local maps should be equal to the number of object models in the database times the number of running cycles. However, the



(a)

(b)

Fig. 21 Comparison between the standard EKF and the hierarchical SLAM results: (**a**) position error of estimated visual objects and (**b**) processing time per update step

local maps are generated less than the ideal value because the objects are occasionally missed.

Then the visual object recognition gives a constraint for consistency to the global map via a loop closing procedure. For every loop closure, we can increase the consistency of the global map by solving constraint non-linear optimization problems as described before. The resulting maps were obtained by 3 and 25 loop closures during 2 and 20 cycles, respectively. In all cases, at least one loop closing event per cycle was accomplished. This means that enough loop closures were executed to maintain global consistency for the given environment.

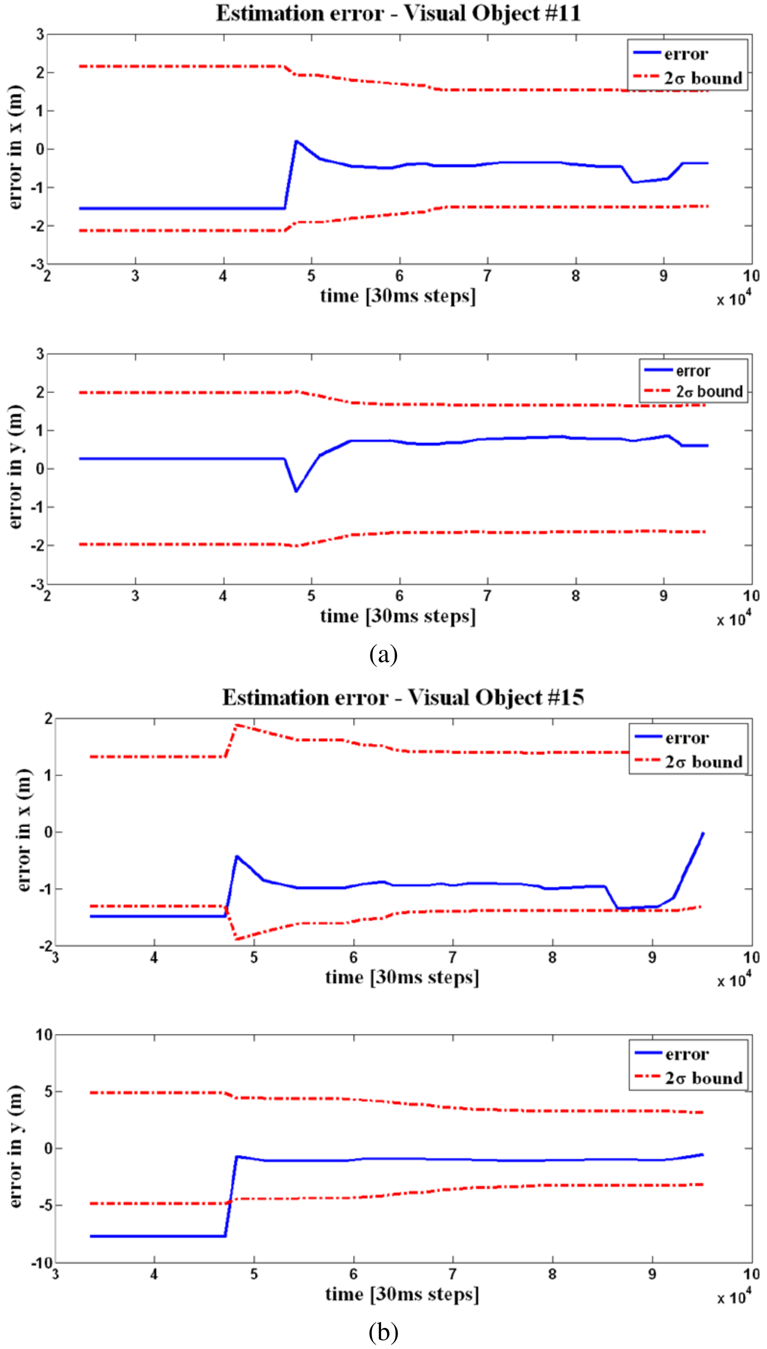
The results of the local map approach are shown in Fig. 19. It displays the overlapped local maps to illustrate the global consistency of our method. The overlapped local maps look similar to the single global maps of the standard EKF-SLAM (Fig. 18) because the loop closure can maintain global consistency of the local maps successfully. We can also check the performance by investigating the magnified regions of resulting maps (Fig. 19(a) and (c)).

The results present similar performance to the standard EKF-SLAM. However, the local map approach has high potential to remedy the linearization errors in a large indoor environment and the computational problem of handling a large number of landmarks. In particular, our local map approach can lead VR-SLAM to rely more on visual object information than the sonar features for global consistency, and it makes both features to be fused more hierarchically.

5.2 Building environment

We expand the experiment in a larger environment like a building environment (Fig. 20(a)) where several people move around the robot. The robot navigated along the right wall with a loop of about 210 m at an average speed of 0.15 m/s twice. To cover the whole environment, we registered 20 visual objects at the model database in advance. In this section, we provide quantitative analysis of the proposed method as follows.

First, we compare four estimated maps of sonar-only, vision-only, and two applications of the proposed method such as standard EKF-SLAM and hierarchical SLAM with both features by overlaying the resulting SLAM maps on the floor plan (Fig. 20(a)). As expected, sonar-only SLAM fails a loop closure between the first and second loop of the robot path in the large environment. Besides, it makes duplicated line segments in vertical hallways due to false association between right and left walls. The failures of association result in the final robot pose error as (3.23 m, −0.90 m, −7.03°). Vision-only SLAM enables to do correct association between visual objects with no effect of vehicle uncertainty. However, the estimated locations of visual objects are mismatched with the floor plan because it estimates the robot pose by using only odometry data when the sparsely-located visual objects are not detectable. Thus, the estimated errors become larger as the objects are located distantly from the origin (left bottom of the map) and the maximum error of object location is up to about 5 m. However, the final robot pose error is obtained as (0.06 m, 0.02 m, 3.32°) with the help of visual objects near the origin. The standard EKF-SLAM and hierarchical SLAM using both features show better performance with the final pose errors as (0.04 m, −0.08 m, 0.55°) and (0.03 m, 0.01 m, −1.61°), respectively. The frequent update with sonar features and correct association with visual object recognition make it possible to prevent the standard EKF-SLAM from divergence of estimation. Unfortunately, however, the standard EKF-SLAM fails to perform loop closing of sonar features near the origin as sonar-only SLAM does, even though the estimation



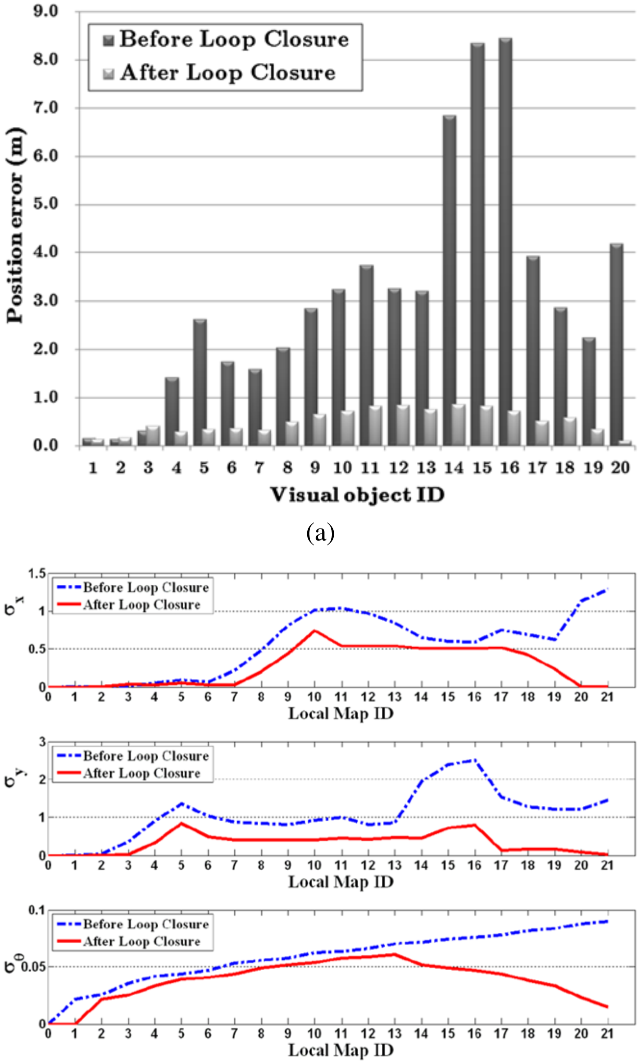
(a)

(b)

Fig. 22 Position errors between the actual and estimated location for (**a**) visual object #11 and (**b**) #15. After the loop closure at 48,000 time step, the error are maintained with the 95% confidence boundary

errors are bounded by both features.On the other hand, the hierarchical SLAM achieves globally consistent estimation because of the successful loop closing process using visual objects. Moreover, the estimated map matches better with floor plan than any other cases.

Second, we compare the performance between the proposed methods applying to the standard EKF-SLAM and the hierarchical SLAM framework in terms of accuracy, computational overhead and estimation consistency. The accuracy of the resulting maps can be compared with the estimated position error of visual objects (Fig. 21(a)). The local map approach generated a globally consistent map with about 2-fold enhancement on the average by using reliable matching of visual objects. Figure 21(b) depicts the computational time for each SLAM update of the proposed method on a 2.0 Ghz Core2 CPU. The computational burden of the standard EKF-SLAM, which scales quadratically in the number of landmarks, was increased over time with handling 175 landmarks. Overall, the hierarchical SLAM was faster than the standard EKF-SLAM except the moment of the first loop



(a)

(b)

Fig. 23 Before and after the first loop closure: (**a**) position error of estimated visual objects and (**b**) σ uncertainty bound of the base reference frame of each local map with respect to the global reference frame

closure. Although it takes a bit longer time when the robot closed the loop, it shows the local map approach can efficiently reduce the computational burden by constructing small local maps sequentially. Figure 22 shows consistency of the hierarchical SLAM. The standard EKF-SLAM failed to maintain its consistency, but the estimated errors of the hierarchical SLAM were bounded within the 2σ limits after the first loop closure is finished.

Finally, we show the role of the loop closure by evaluating mapping accuracy and covariance changes before and after the first loop closure. The first loop closure was detected between the 1st and the 21st local maps with their representative visual object. After the loop closing event, the position error of the estimated visual objects was dramatically decreased (Fig. 23(a)). About 6-fold enhancement on the average of the experiment denotes that we can obtain more accurate SLAM map representation by closing the loop successfully with the reliable visual object recognition. And the accumulated position uncertainty of the base reference frame for the local maps is presented with respect to the global reference frame (Fig. 23(b)). It also reduced the position uncertainty by using the loop constraint of the corresponding local maps.

6 Conclusion

This paper presented VR-SLAM (Vision and Range Sensor-SLAM). The VR-SLAM could work with vision and various types of range sensors such as laser range finders, ultrasonic sensors and infrared sensors. In this paper, especially, a practical solution for EKF-SLAM in an indoor environment is proposed. It mainly focuses on fusing the features obtained from both ultrasonic sensors and a stereo camera in real-time.

The VR-SLAM algorithm has the following salient traits. (1) It has a robust sonar feature detection scheme of point and line features. It can generate both accurate point features which are hardly affected by the range and angular errors of ultrasonic sensors and a sufficient number of line features using the modified TBF algorithm. (2) It has a visual object recognition scheme which has excellent discriminating ability for data association. The scheme makes SLAM more computationally feasible and improves the data association process by grouping a set of visual features as a physical object using RANSAC clustering. (3) It is a multi-sensor SLAM fusing the advantages of ultrasonic and vision sensors to achieve complementary cooperation. It can efficiently overcome both the false data association problem of ultrasonic sensors and the low frequency SLAM update and the weakness to illumination changes of vision sensors. We applied these methods in the frameworks of standard EKF-SLAM and hierarchical SLAM to show the superiority of the proposed methods. The proposed algorithm can be used effectively in the hierarchical SLAM, since each local map can have a distinct visual object, which enhances loop closing performance. The applicability to the hierarchical SLAM helps to expand the method to larger environments.

Experimental results from three different indoor environments verified the performance and robustness of the proposed method.

Acknowledgement This work was supported in part by the IT R&D program of MIC/IITA [2005-S-033-02, Embedded Component Technology and Standardization for URC], by the Korea Science and Engineering Foundation (KOSEF) grant of MOST [No. R0A-2003-000-10308-0], and by the grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare [A020603], Republic of Korea.

References

Ahn, S., Choi, M., Choi, J., & Chung, W. K. (2006). Data association using visual object recognition for EKF-SLAM in home environment. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 2588–2594).

Barfoot, T. D. (2005). Online visual motion estimation using FastSLAM with SIFT features. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 579–585).

Bosse, M., Newman, P., Leonard, J., & Teller, S. (2004). SLAM in large-scale cyclic environments using the ATLAS framework. *International Journal on Robotics and Research*, *23*(12), 1113–1139.

Choi, J., Ahn, S., & Chung, W. K. (2005). Robust sonar feature detection for the SLAM of mobile robot. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 3415–3420).

Choset, H., Nagatani, K., & Lazar, N. A. (2003). The arc-transversal median algorithm: A geometric approach to increasing ultrasonic sensor azimuth accuracy. *IEEE Transactions on Robotics and Automation*, *19*(3), 513–523.

Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Proc. of international conference on computer vision* (pp. 1403–1410).

Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, *17*(3), 229–241.

Elinas, P., Sim, R., & Little, J. J. (2006). σSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution. In *Proc. of IEEE international conference on robotics and automation* (pp. 1564–1570).

Estrada, C., Neira, J., & Tardós, J.D. (2005). Hierarchical SLAM: Real-time accurate mapping of large environment. *IEEE Transactions on Robotics*, *21*(4), 588–596.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Folkesson, J., Jensfelt, P., & Christensen, H. I. (2005). Graphical SLAM using vision and the measurement subspace. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 325–330).

Guivant, J. E., & Nebot, E. M. (2001). Optimization of the simultaneous localization and map-building algorithms for real-time implementation. *IEEE Transactions on Robotics and Automation*, *17*(3), 242–257.

Jeong, W., & Lee, K. M. (2005). CV-SLAM: A new ceiling vision-based SLAM technique. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 3195–3200).

Karlsson, N., Bernardo, E. D., Ostrowski, J., Goncalves, L., Pirjanian, P., & Munich, M. E. (2005). The vSLAM algorithm for robust localization and mapping. In *Proc. of IEEE international conference on robotics and automation* (pp. 24–29).

Leonard, J. J., & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, *7*(3), 376–382.

Lin, Z., Kim, S., & Kweon, I. S. (2005). Recognition-based indoor topological navigation using robust invariant features. In *Proc. of IEEE/RSJ international conference on intelligent robots and systems* (pp. 2309–2314).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Mikolajczyk, K., & Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, *60*(1), 63–86.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615–1630.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. of the sixteenth international joint conference on artificial intelligence* (pp. 1151–1156).

Newman, P., & Ho, K. (2005). SLAM-Loop closing with visually salient features. In *Proc. of IEEE international conference on robotics and automation* (pp. 635–642).

Newman, P., Cole, D., & Ho, K. (2006). Outdoor SLAM using visual appearance and laser ranging. In *Proc. of IEEE international conference on robotics and automation* (pp. 1180–1187).

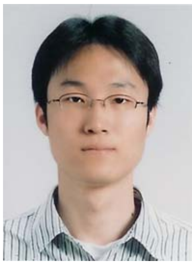
Ortin, D., Neira, J., & Montiel, J. M. M. (2003). Relocation using laser and vision. In *Proc. of IEEE international conference on robotics and automation* (pp. 1505–1510).

Se, S., Lowe, D. G., & Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, *21*(8), 735–758.

Tardós, J. D., Neira, J., Newman, P. M., & Leonard, J. J. (2002). Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, *21*(4), 311–330.

Wijk, O., & Christensen, H. I. (2000). Triangulation-based fusion of sonar data with application in robot tracking. *IEEE Transactions on Robotics and Automation*, *16*(6), 740–752.

SungHwan Ahn received his BS and MS degree in Mechanical Engineering from Pohang University of Science and Technology (POSTECH), Korea, in 2002 and 2003, respectively. He is currently a PhD candidate in Mechanical Engineering from POSTECH, Korea. His current research interests are mainly concentrated on SLAM with vision and ultrasonic sensors, mapping with visual features, and intelligent robot navigation.



Jinwoo Choi received his BS and MS degree in Mechanical Engineering from Pohang University of Science and Technology (POSTECH), Korea, in 2003 and 2005, respectively. He is currently a PhD candidate in Mechanical Engineering from POSTECH, Korea. His current research interests are mainly concentrated on SLAM, robust sonar features, and data association.

Nakju Lett Doh received his BS, his MS, and his Ph.D. degree in Mechanical Engineering from the Pohang University of Science and Technology (POSTECH), Korea, in 1998, 2000, and 2005, respectively. In 2006, he joined to the school of electrical engineering, Korea University, as an assistant professor. In 2003, he won the best student paper award in IEEE International Conference on Robotics and Automation. He also got the gold and the bronze prize in Humantech Thesis Competition hosted by Samsung Electronics in 2005 and 2000, respectively. His research area is the mobile robotics including localization, mapping, and motion planning of robots.

Wan Kyun Chung National University in 1981, his MS degree in Mechanical Engineering from KAIST in 1983, and his Ph.D. in Production Engineering from KAIST in 1987. He is Professor in the school of Mechanical Engineering, POSTECH (he joined the faculty in 1987). In 1988, he was a visiting professor at the Robotics Institute of Carnegie-Mellon University. In 1995 he was a visiting scholar at the university of California, Berkeley. His research interests include the localization and navigation for mobile robots, underwater robots and development of robust controller for precision motion control. He is a director of National Research Laboratory for Intelligent Mobile Robot Navigation. He is serving as an Associate Editor for IEEE Tr. on Robotics, international editorial board for Advanced Robotics.