2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Merging of 3D Visual Maps Based on Part-Map Retrieval and Path Consistency

Masahiro Tomono

Abstract— **This paper presents a map-merging method which builds a 3D visual map by connecting part maps. The purpose is to increase flexibility and robustness in 3D mapping. Map merging is performed by combining image retrieval using bag-of-words, geometric verification, and pose adjustment. A key issue is robust data association. To cope with false matches, we perform group matching with pose propagation, which checks the geometric consistency of point correspondences over multiple frames. Also, we perform path consistency check, which examines the accumulated errors along a loop to eliminate inconsistent part map connections. Experiments show our method successfully built detailed 3D maps of indoor environments.**

I. Introduction

Map merging is an important issue in robotic mapping, and there have been studies mainly for multi-robot mapping [11], [7], [9], [2], [8], [27], [4]. This paper considers map merging as a tool for increasing flexibility and robustness in 3D mapping.

In most SLAM systems, sensor data are assumed to be sequential, and adjacent data in the sequence are associated with each other using wheel/visual odometry. If the data sequence is interrupted for some reasons, the mapping process will be terminated. There are many reasons for interruption: motion blur caused by quick motions or collisions, large occlusions caused by obstacles in front of the sensor. Furthermore, sensor data can be interrupted intentionally by the user or robot. Large or complex environments are hard to be covered efficiently by a single sequence, and sensor data must be collected separately. If the system accepts only a single data sequence without interruption, it will be inconvenient in practical use.

A solution to this problem would be a submap-based scheme, in which sensor data are obtained as a set of subsequences, and a whole map is built by automatically connecting submaps generated from the subsequences. This scheme will free the user/robot from obtaining a long, complete data sequence. Here, we call submaps as *part maps* to emphasize their role as parts of a whole map. This paper proposes a method of 3D mapping from stereo image sequences based on the map merging scheme. This system builds detailed 3D maps using the edge-point based SLAM (EdgeSLAM for short) [24], which provides rich shape information. The system uses a stereo camera only. When connecting part maps, pose adjustment [13] is used to adjust the shape of the whole map.

A key issue in map merging is data association between part maps. Similarly to the conventional vision-based approaches, our method employs image retrieval based on bag-of-words (BoW) and geometric verification based on camera pose estimation using 2D-3D matching. However, difficulty in data association emerges in indoor environments such as corridors and halls, which have non-textured objects and similar structures which repeatedly appear. Such places can generate false positives and false negatives.

To cope with these false matches, this paper proposes two techniques. One is group matching with pose propagation, in which data association by BoW and geometric verification is performed over multiple frames. The relative poses between adjacent frames obtained by EdgeSLAM can be additional geometric constraints to reduce accidental false matches. The other is path consistency check, which finds loops in a connected graph of part maps, and eliminates false connections based on loop constraint. This is effective for false positives due to similarly looking places which are hard to distinguish by appearance.

The contributions of the paper are twofold. First, it proposes a robust data association scheme based on group matching with pose propagation and path consistency in the context of map merging. Second, it provides a scheme of 3D map merging by integrating EdgeSLAM, image retrieval, data association, and pose adjustment.

II. Related Work

Many studies on map merging are motivated by multi-robot mapping [11], [7], [9], [2], [8], [27], [4]. Some studies assume that relative poses between robots can be obtained when the robots encounter one another [11], [8], [27]. This information greatly helps data association. On the other hand, there have been studies on merging submaps without relative poses between robots [7], [9], [2], [4]. Our approach belongs to this class although our purpose is single-robot mapping. In [7], 2D laser local maps are joined into a single global map based on data association using image sequence matching. In [2], 2D occupancy grid maps are connected by finding maximum overlaps between them using a stochastic search algorithm. In [9], 2D

M. Tomono is with Future Robotics Technology Center, Chiba Institute of Technology, Narashino, Chiba 275-0016, Japan. tomono@furo.org

topological maps are merged using subgraph matching and image registration techniques. In [4], appearance based maps (topological maps) are joined using BoW matching and geometric verification. Our method connects submaps using BoW and 3D pose estimation with group matching and path consistency check for outlier rejection. To our knowledge, there are no conventional map merging systems which build a detailed 3D map using image edge points.

Bundler [22] collects a large amount of unordered images from web sites and builds a large scale 3D model based on structure-from-motion and bundle adjustment. This system does not need the sequentiality of the images, and makes an image graph which represents neighborhood relationships between images. This makes the data collection procedure much flexible, but at least one perfect image sequence must be found in the images to reconstruct the whole scene. This needs a lot of images if images are collected randomly. This approach is good for modeling from web site images, but for visual SLAM, combining short image sequences would be a good tradeoff between flexibility and efficiency. Our approach is in the middle between the two extremes: conventional SLAM with perfectly ordered data and Bundler with unordered data.

A key issue in map merging is how to find connections between submaps. Visual SLAM can provide strong data association using image descriptors such as SIFT [17]. The BoW scheme has been used for data association in robot localization and place recognition [26], [6], [5], [12], [25]. The BoW uses no geometric constraints between feature points and can generate many false positives. To filter out false positives, many systems employ geometric verification such as 2D-2D feature matching based on epipolar geometry [26], [6], [4] or 2D-3D feature matching based on reprojection error minimization [12], [25]. Such geometric computation is used also to obtain relative poses between submaps to create a pose graph for pose adjustment.

Grouping of matched pairs are used to improve place recognition [19], [16]. In [19], geometrically consistent pairs are grouped using a graph partitioning method in order to eliminate false positives. In [16], grouping is performed based on BoW matching followed by geometric verification in order to reduce redundant matches, but it is also useful to select correct matches. In this paper, we employ group matching with pose propagation, where we first reduce false negatives using pose propagation and then reduce false positives using group matching.

The concept of our path consistency check is similar to the recent studies on outlier rejection in loop closing [14], [20], [23]. In [14], false loop closures are eliminated based on consensus among loop closures and with the robot trajectory. In [20] and [23], outlier rejection process is incorporated into the pose graph optimization. In these methods, a whole robot trajectory obtained by odometry plays an important role for efficient outlier rejection. On the other hand, in map merging, no whole robot trajectory is provided beforehand. Map merging includes a combinatorial problem of finding a complete trajectory by connecting submaps, which makes it hard to apply these methods to map merging.

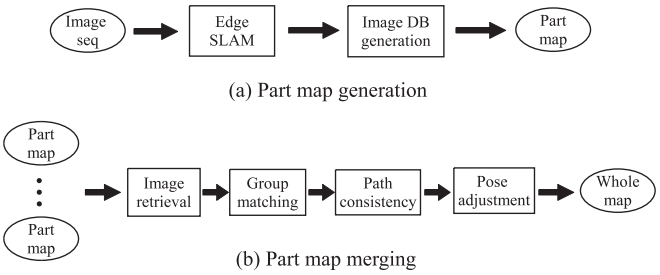


Fig. 1. Flow diagram of the proposed method.

III. Part Map Generation

A. Problem Statement

We assume that map merging is performed for the following cases; (1) Sensor data sequences are separately collected from divided regions, and a whole map is created from the set of data sequences; (2) A data sequence is interrupted by disturbances, and data acquisition is resumed around the failure point. In both cases, it is possible to narrow down the candidates of part map connections if prior information is available, e.g., the region where the previous data sequence was collected. This is useful especially for large-scale mapping. In this paper, however, we assume that the system has no prior information about part map connections. This is the most general case, and it can be a basis for improving efficiency using prior information in future.

Fig. 1 shows the flow diagram. The first stage is part map generation. Part maps are generated using EdgeSLAM, and key frames are stored in an image database. The second stage is part map merging. The system builds a whole map from multiple part maps by finding the connections between them and by adjusting the poses of the part maps. Sensor data can be divided arbitrarily at any points, but they must have overlapping regions to connect part maps. We assume the length of an overlapping region is typically 3 to 10 meters.

B. Local Map Building

A 3D map is built based on the method proposed by our previous work [24]. We briefly review the method for completeness.

A local map is built using EdgeSLAM [24], which uses image edge points detected by the Canny detector [3]. Note that edge points can be obtained from not only long segments but also fine textures. Edge points are reconstructed for each frame based on the parallel-stereo reconstruction.

The camera motion from time $t-1$ to t is estimated by matching the edge points in frame I_{t-1} and those in frame I_t. Our method employs 2D-3D matching, in which the 3D points reconstructed from I_{t-1} are matched with the 2D points detected in I_t. The registration is performed using a variant of ICP algorithm [1] on the image plane. Let r_t be the camera pose at t, P^i_{t-1} be the i-th 3D edge point reconstructed at $t-1$, and $\hat{p}^i_{t-1}$ be the projected point of P^i_{t-1} onto image I_t. Let p^i_t be the image edge point at t which corresponds to $\hat{p}^i_{t-1}$. A cost function F is defined as follows.

$$F(r_t) = \frac{1}{N}\sum_{i=1}^{N} d(p^i_t, \hat{p}^i_{t-1}) \quad (1)$$

Here, $d(p^i_t, \hat{p}^i_{t-1})$ is the perpendicular distance between $\hat{p}^i_{t-1}$ and the edge segment on which p^i_t lies.

Camera pose r_t and edge point correspondences are searched by minimizing $F(r_t)$ using the ICP. The initial value of r_t is set to r_{t-1}, and the initial correspondence p^i_t of $\hat{p}^i_{t-1}$ is set to the edge point which is the closest to $\hat{p}^i_{t-1}$ in terms of Euclidean distance. By repeating the minimization of $F(r_t)$ and edge point matching, the optimal r_t and edge point correspondences are obtained.

Based on the obtained camera pose, a 3D map is built by transforming the stereo 3D points from the camera coordinate system to the world coordinate system.

C. Part Map

A part map is created from a local map generated in the previous section. A part map m_n is defined as (F_n, G^I_n, G^E_n, C_n). F_n is a set of map elements, G^I_n is an internal graph, and G^E_n is an external graph. C_n is the reference frame of m_n. C_n is initially set at the origin of the world frame, and is transformed by map merging.

A map element (*mapel* for short) is a key frame with extra information, and is used as the smallest building block for mapping. A mapel $f_{k,n}$ of m_n is defined as $(I_{k,n}, r_{k,n}, E_{k,n}, P_{k,n})$. $I_{k,n}$ is a key frame (left image of a stereo frame), and $r_{k,n}$ is the camera pose of $I_{k,n}$ in C_n. $E_{k,n}$ is the set of the edge points detected in $I_{k,n}$, and $P_{k,n}$ is the set of the 3D points reconstructed from $E_{k,n}$. All of r_{k_n}, $E_{k,n}$ and $P_{k,n}$ are obtained by EdgeSLAM. A key frame is extracted from the image sequence when the camera movement from the previous key frame exceeds a threshold (±200 [mm] and ±10 [deg] in implementation). The key frame extraction reduces redundancies in the image sequence since the distance between two successive frames are small to make the ICP work stably in EdgeSLAM.

D. Pose Graphs for Part Maps

A part map has two kinds of pose graphs: internal graph and external graph. Fig. 2 shows an example.

An internal graph $G^I_n = (N^I_n, A^I_n)$ is a pose graph which represents the internal structure of a part map.

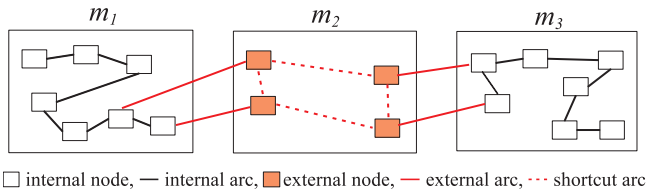


Fig. 2. Pose graphs. This figure illustrates internal graphs for part maps m_1 and m_3, and an external graph for m_2.

N^I_n is a set of internal nodes, each of which corresponds to a mapel. There is a one-to-one correspondence between N^I_n and F_n, and we identify nodes with mapels. A^I_n is a set of internal arcs. An internal arc has the relative pose and its covariance between two nodes. Internal arcs are created between neighborhood mapels based on the camera poses estimated by EdgeSLAM.

An external graph $G^E_n = (N^E_n, A^E_n)$ is used to connect part maps. N^E_n is a set of the nodes connected to other part maps. Note that $N^E_n \subset N^I_n$. A^E_n consists of two sets of arcs as $A^J_n \bigcup A^S_n$. A^J_n is a set of external arcs between N^E_n and nodes of other external graphs. A^S_n is a set of shortcut arcs between nodes in N^E_n. The nodes in N^E_n may have no internal arcs since $N^E_n \subset N^I_n$. To avoid such nodes being isolated, we create shortcut arcs. The relative pose of a shortcut arc is computed simply from the current poses of the nodes. The covariance is calculated by accumulating the covariances along the path from one node to the other node.

The method how to perform pose adjustment using these graphs is explained in Section IV-E.

IV. 3D Mapping by Merging Part Maps

As shown in Fig. 1 (b), a whole 3D map is generated by image retrieval, group matching, path consistency check and pose adjustment.

A. Image Retrieval

In order to build a whole map from part maps, connections between part maps must be found. To find the connections, we adopt the image retrieval scheme proposed by our previous work [25]. Since key points extracted by SIFT can be sparse in non-textured environments, we employ edge points as key points. For each edge point in $E_{k,n}$ of mapel $f_{k,n}$, a SIFT descriptor is computed through the scale-space analysis [15] to make the descriptor scale-invariant. Then, the descriptor is converted to a visual word through the vocabulary tree [18].

The vocabulary tree is an image database based on tree search and inverted files. Each node in the tree corresponds to a visual word and has an inverted file to store the identifiers of the images that contain the features corresponding to the visual word. When a query mapel $f_{k,n}$ is given, the mapels matched with $f_{k,n}$

are retrieved from the vocabulary tree and the inverted files based on the TF-IDF scoring scheme [21], [18].

An inverted file is created for each part map. This makes image retrieval available for every combination of part maps. If the system has prior information on connections of part maps, retrieval is performed for a small set of part maps pruned by the prior. Otherwise, retrieval is performed for all the part maps.

Image retrieval generates a TF-IDF score matrix, which is used for pose estimation. The row and column of the score matrix indicates the mapels in the current part map and those in the reference part map, respectively. Each element of the score matrix has a TF-IDF score of a mapel pair. An example of score matrix is shown in Fig. 7 in Section V-A.

B. Pose Estimation

After retrieving part maps, we calculate the relative pose between part maps. One purpose is to create pose graphs. To create a whole map through pose adjustment, we need the relative poses between mapels. The other purpose is to reduce false positives. Image retrieval using BoW can generate false positives, and pose estimation can be used as geometric verification mentioned in Section II.

We apply pose estimation to the mapel pairs each of which has a score larger than a threshold th_1 from the TF-IDF score matrix. The relative pose is obtained by estimating the camera pose of a mapel $f_{j,2}$ with respect to a mapel $f_{i,1}$ in another part map m_1 in a similar manner to our previous work [25]. First, we find point correspondences between the 3D points of $f_{i,1}$ and the edge points of $f_{j,2}$. This is done based on two kinds of point correspondences. One is the point correspondences between the 3D points and the edge points in $f_{i,1}$, which have been obtained by EdgeSLAM. The other is the point correspondences between edge points in $f_{i,1}$ and $f_{j,2}$ obtained through the image retrieval mentioned above. Then, we compute the camera pose of $f_{j,2}$ relative to $f_{i,1}$ by minimizing the reprojection errors of the 3D points of $f_{i,1}$ onto the edge points of $f_{j,2}$. This is done by the ICP algorithm based on Eq. (1).

We create a pose score matrix based on the following score $S_{i,j}$, which is used for group matching.

$$S_{i,j} = \frac{2N_{i,j}}{N_i + N_j} \tag{2}$$

N_i and N_j are the numbers of the edge points in f_i and f_j respectively. $N_{i,j}$ is the number of the edge points matched between f_i and f_j.

It is difficult to completely eliminate false positives due to noises, occlusions, sparse features, and similar appearances. Group matching and path consistency in the next sections cope with this problem. Furthermore, due to bad initial values, the ICP can fail to estimate correct poses, which generate false negatives. The randomized ICP in [25] can cope with this problem, but applying it to every candidate mapel pair is time consuming. Instead, group matching with pose propagation addresses this problem.

C. Group Matching with Pose Propagation

We introduce group matching with pose propagation to cope with false matches between part maps. Since an overlapping region of two part maps has multiple images, the accuracy of part map matching will be improved by exploiting geometric constraints over the images. For this purpose, we employ group matching based on a consensus scheme among estimated poses. However, false negatives in pose estimation will make the consensus less reliable. Therefore, before group matching, pose propagation is performed to reduce false negatives.

Another purpose of group matching is to reduce the computation time of the path consistency check in the next section. Since its computational complexity is exponential in the number of arcs between part maps, we reduce the arcs by grouping them into one arc (map arc defined later).

Fig. 3 shows an example of the geometric constraint over multiple images. By the method mentioned in the previous section, mapel $f_{j,2}$ in part map m_2 is matched with mapel $f_{i,1}$ in part map m_1, and $f_{j,2}$ has a pose $q_{j,1}$ in the reference frame of m_1. Likewise, $q_{n,1}$ is obtained from $f_{k,1}$ and $f_{n,2}$. If these matches are correct, the relative pose between $q_{j,1}$ and $q_{n,1}$ is equivalent with $rel_{j,n}$, which is the relative pose between $r_{j,2}$ and $r_{n,2}$ obtained by EdgeSLAM. Thus, we have the following equation.

$$q_{n,1} \quad = \quad q_{j,1} \oplus rel_{j,n} \tag{3}$$

False negatives are reduced by pose propagation using these constraints. We find the mapel pairs M_1 each of which has a score larger than a threshold th_2 from the pose score matrix. Using the pose $q_{j,1}$ estimated for $(f_{i,1}, f_{j,2})$ in M_1, the pose $\tilde{q}_{n,1}$ of a nearby mapel pair $(f_{k,1}, f_{n,2})$ is calculated according to Eq. (3), that is, $\tilde{q}_{n,1} = q_{j,1} \oplus rel_{j,n}$. Then, we calculate the matching score based on Eq. (1) for $\tilde{q}_{n,1}$. If the score is better than that for the original one $q_{n,1}$, $\tilde{q}_{n,1}$ overrides $q_{n,1}$ in the pose score matrix. Repeating this process for each mapel pair in M_1, the pose score matrix is updated.

In this process, for efficiency, Eq. (1) is calculated for $\tilde{q}_{n,1}$ without the ICP. When the distance between $f_{i,1}$ and $f_{k,1}$ is small, the accumulated errors by EdgeSLAM or visual odometry will be small and the simple prediction by Eq. (3) will provide a good matching score.

Next, false positives are reduced by group matching using the updated pose score matrix based on a consensus based scheme. We find the mapel pairs M_2 each of which has a score larger than a threshold th_2 from the updated pose score matrix. Then, for mapel pair $p_{i,j} = (f_{i,1}, f_{j,2}) \in M_2$, we find a *group* for $p_{i,j}$, which is defined as the mapel pairs having a relative pose

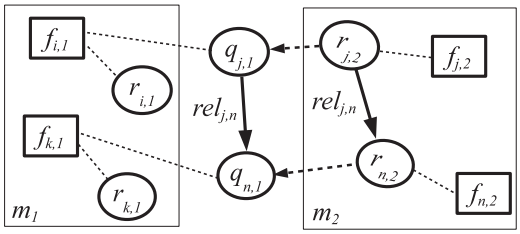


Fig. 3. Geometric constraints between robot poses. $q_{j,1}$ is the camera pose which makes the image of $f_{j,2}$ matched with 3D points of $f_{i,1}$. The relative pose between $q_{j,1}$ and $q_{n,1}$ is equivalent with that between $r_{j,2}$ and $r_{n,2}$.

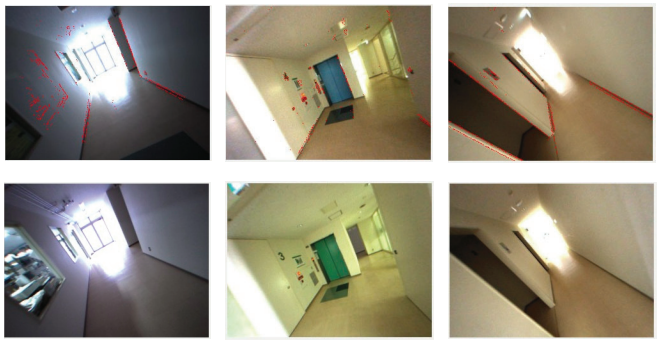


Fig. 4. Examples of similar structures. (a) Similar looking corridor except the window in the front. (b) The floor number is different. (c) Completely same appearance on different floors.

consistent with the relative pose of $p_{i,j}$. We search mapel pairs in the region of radius N from $p_{i,j}$ for efficiency. If $\sqrt{D_{n,1}} \leq th_3$ in Eq. (4) holds, $q_{n,1}$ is added to the group as an inlier. Otherwise, $q_{n,1}$ is an outlier for the group.

$$D_{n,1} = (q_{n,1} - \tilde{q}_{n,1})^T \Sigma_{j,n}^{-1} (q_{n,1} - \tilde{q}_{n,1}) \quad (4)$$

$\Sigma_{j,n}$ is a covariance of $rel_{j,n}$. We examine all the mapel pairs for each group, and select the groups having inliers more than a threshold th_4. An external arc is created for each mapel pair in the groups obtained.

In implementation, the threshold values are as follows; $th_1 = 0.15$, $th_2 = 0.4$, $th_3 = 3$, $th_4 = 10$, and $N = 5$.

D. Path Consistency Check

Man-made environments such as buildings have similar structures which repeatedly appear. Fig. 4 shows examples. The group matching is effective for accidental false positives, but it is not for systematic ones. It is possible to distinguish such places by checking differences on a long trajectory, but sharing a long trajectory by part maps is not suitable for our motivation, and we employ another approach.

We exploit path consistency to address the false positives which escape group matching. The path consistency here is the constraint that one returns to the same position after traversing along a loop. We introduce *map graph* to examine path consistency. A map graph is a set

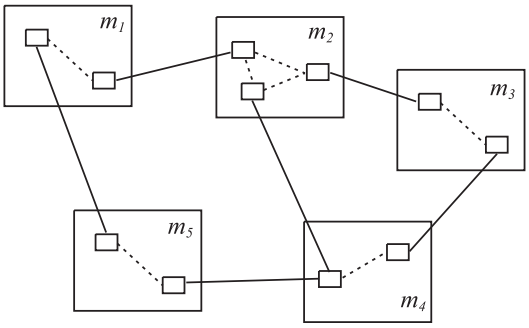


Fig. 5. A map graph is created by connecting external graphs using map arcs. This graph is used for path consistency check.

of external graphs combining with one another. We use *map arcs* to reduce redundancies in the external arcs. A map arc is a representative of the external arcs in a group obtained by group matching. The representative arc has the relative pose and its covariance of the mapel pair which has the best score in the group. Fig. 5 shows a map graph. If the connections between part maps is correct, the loops along map arcs and shortcut arcs in the map graph must be true loops in the real world.

We detect all the simple loops (or circuits in graph theory [10]) in the map arcs and examine whether or not each loop is a true one. Simple loops can be detected using a cycle basis [10]. For the nodes n_1 and n_2 of a map arc a_i in each loop, we calculate the relative pose $rel(n_1, n_2)$ and its covariance $\Sigma(n_1, n_2)$ between n_1 and n_2 along the longer path in the loop. These are calculated by accumulating the relative pose and covariance of each map arc and shortcut arc in the longer path. Then, based on Eq. (5), we calculate the distance between the relative pose along the longer path and that of the shorter path, which is a_i itself. Here, rel_i is the relative pose of a_i.

$$E = (rel(n_1, n_2) - rel_i)^T \Sigma(n_1, n_2)^{-1} (rel(n_1, n_2) - rel_i) \quad (5)$$

If $\sqrt{E} \leq th_3$, the loop is regarded as consistent. Otherwise, the loop has one or more false positive arcs. These false positives are hard to find directly. Thus, we select good arcs by combining the consistent loops in the following manner.

We generate a map graph by combining the consistent loops. Note that the combination of individually consistent loops might generate inconsistent loops. Therefore, we check path consistency again for the loops in the map graph. If an inconsistent loop is found, that combination is discarded. By repeating this process, we build a consistent map graph. If multiple candidates are obtained, we select the candidate which firstly has the largest set of part maps and secondly has the smallest average errors in terms of Eq. (5).

The computation cost of this method is exponential in the number of map arcs. A solution to this issue would be a hierarchical scheme, which is future work.

E. Pose Adjustment

In order to perform pose adjustment, external arcs A_n^J are created as follows. We explain it with Fig. 3 as an example. For each pose pair $(q_{j,1}, r_{i,1})$ in the matched group, an external arc is created between $f_{i,1}$ and $f_{j,2}$. The arc has the relative pose and its covariance between $f_{i,1}$ and $f_{j,2}$. Note that the relative pose is computed between $q_{j,1}$ and $r_{i,1}$ since $f_{j,2}$ must be at $q_{j,1}$ to connect m_1 and m_2. The covariance is simply calculated as the sum of those of $q_{j,1}$ and $r_{i,1}$.

Pose adjustment is performed in a two stage fashion. The first stage is done using the external graphs connected by the external arcs which belong to the map arcs selected by the path consistency check. At the second stage, pose adjustment is applied to a whole graph, which is generated by combining all the internal graphs using the external arcs. The poses of the nodes in the whole graph are updated by the first pose adjustment, and they provide good initial values for the second pose adjustment.

In implementation, we use the Sparse Pose Adjustment (SPA) software developed by Konolige et al. [13].

V. Experiments

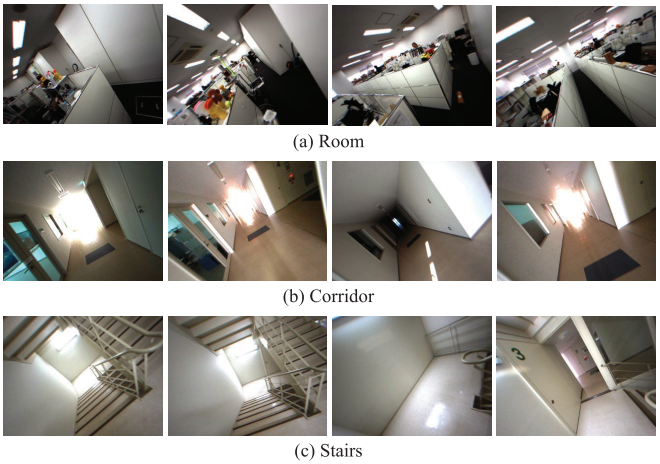
We conducted experiments using Point Grey Research's stereo camera Bumblebee2. Images were captured manually by a walking human. The image size was reduced to 320×240 pixels. The system is implemented in C++ and runs on a laptop PC with Core i7-2960XM 2.7GHz. Neither multicore computing nor GPGPU were used.

A. Part Map Retrieval

We conducted experiments on data association in indoor environments (a room, corridor, and stairs). In these environments, image sequences were captured twice, and two sets of part maps were built from the image sequences. Then, we examined the performance of data association between the two part maps. Fig. 6 shows images of the environments. While the room have many features and a variety of appearances, the corridor and stairs have similar structures and appearances.

We evaluated the performance of part map retrieval using BoW-only, BoW+single matching, and BoW+group matching. Single matching means geometric verification by pose estimation using a single image. BoW-only and BoW+single matching are widely used by conventional vision-based SLAM systems. BoW+group matching includes the pose propagation.

Fig. 7 shows the score matrices by image retrieval using BoW as presented in Section IV-A. Brightness indicates the magnitude of the matching score. The diagonal elements tend to be brighter since the camera moved along similar paths in this experiment. Note that in the case of map merging the overlapping region will be much smaller. Fig. 7 (a) is the result for the room, and the scores along the diagonal are clearly high. This indicates that the room is well distinguished by BoW due to salient features. Fig. 7 (b) and (c) are the results for the corridors (1/4 of the whole trajectory) and stairs respectively, and non-diagonal elements have high scores, which are false positives.



(a) Room

(b) Corridor

(c) Stairs

Fig. 6. Images of the environments.

Fig. 8 shows the score matrices for the stairs dataset. As shown in (b), the score matrix for BoW+single matching has false negatives due to the failure of the pose estimation by the ICP. As shown in (c), the pose propagation reduced the false negatives.

Fig. 9 shows the precision-recall curves. In all the matching methods, the precision and recall were evaluated using the matches with the top score. In the room experiment, there are no significant differences between the three methods. The single matching has relatively small recall value due to several false negatives, which was improved by the group matching. In the corridor experiment, BoW-only generates several false positives. BoW+group matching reduce false positives using geometric verification. In the stairs experiment, all the methods generates several false positives, since this environment has similarly looking places (e.g., the first and second images in Fig. 6(c)). The group matching can keep multiple matches for similar places, but this experiment records only the matches having the top score, which might be a false positive with slightly better score than a true positive.

B. 3D Mapping

We conducted experiments of 3D mapping from images captured in our university building. Fig. 10 shows some images of the environments.

1) Building 4th Floor: We built a map of a building floor from seven part maps. This dataset consists of three image sequences, one of which captured the outer loop of the floor. Five part maps were created by dividing randomly this image sequence. The remaining two image sequences captured paths across two long

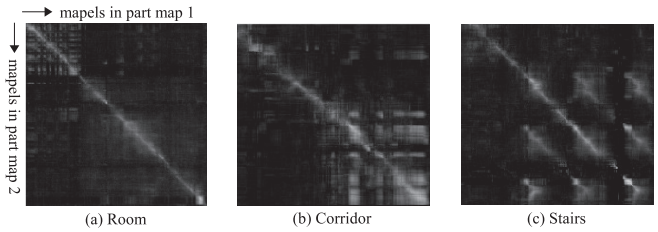


Fig. 7. Score matrices by image retrieval.

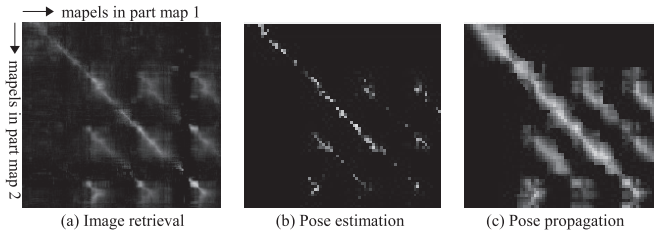


Fig. 8. Score matrices for the stairs dataset.

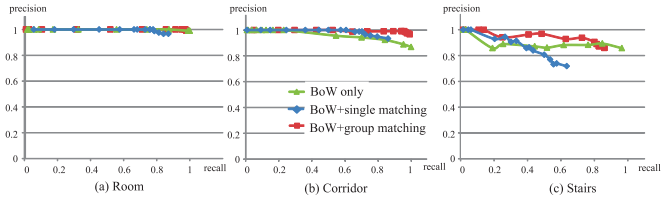


Fig. 9. Precision-recall curves for part map retrieval.

Fig. 10. Images of the environments.

corridors in the outer loop. Two part maps were created from these image sequences.

Fig. 11 (a) shows the seven part maps built from the image sequences, and (b) shows the partial map corresponding to the external graphs after pose adjustment. Note that an external graph has only the mapels connecting to other part maps.

Fig. 12 shows the whole map generated after the second pose adjustment. The total number of the key frames of all the part maps is 1461. The whole map has 1,552,991 landmarks (3D points). The computation time for this dataset is as follows. Part map generation took 23 [msec/frame] in average for EdgeSLAM, and 134 [msec/key frame] in average for image database registration with SIFT descriptor generation for key frames. Part map retrieval took totally 11.5 [sec], most of which was spent by group matching. Path consistency took 43 [msec] for 17 map arcs, but 574 [msec] for 31 map arcs. As mentioned above, the computational complexity of the path consistency check is exponential. Therefore, it is very important to eliminate spurious map arcs by group matching. The software has yet to be customized to improve the performance.

We evaluated the stability of map merging at several precisions with respect to true positives (TP) and false positives (FP) of part map connections (map arcs); precision=0.82 (#TP=14, #FP=3), precision=0.71 (#TP=15, #FP=6), and precision=0.52 (#TP=16, #FP=15). In all the cases, the proposed methods eliminated the false positives by path consistency and generated a 3D whole map with correct topology.

We currently have no ground truth for this environment and the evaluation of the accuracy of the whole map is future work.

2) Building 3rd and 4th Floors: We built a map of two floors of the same building. The map has a loop along two floors connected by stairs. This dataset consists of four image sequences captured on different days and they were randomly divided into eight part maps. Fig. 13 shows the part maps.

Fig. 14 shows the whole map. The total number of the key frames of all the part maps is 1498. The whole map has 1,576,462 landmarks.

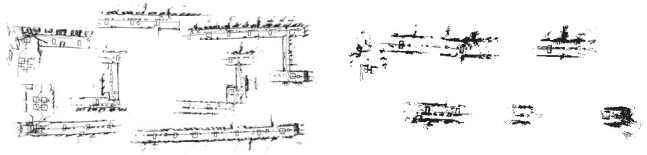
We evaluated the stability at several precisions with respect to part map connections; precision=0.74 (#TP=14, #FP=5), precision=0.67 (#TP=14, #FP=7), and precision=0.57 (#TP=16, #FP=12). In all the cases, eliminating the false positives, the proposed methods generated a 3D whole map with correct topology.

The 3rd and 4th floors have similar structures and appearances. In particular, the regions around the elevators and stairs have very similar appearances and led to false positives. Our method successfully built the whole map in such a condition.

VI. Conclusions

This paper has presented a map-merging method which builds a 3D visual map by connecting part maps. This method provides flexibility and robustness in the mapping procedure. A key issue is robust data association achieved by group matching with pose propagation, which checks the geometric consistency of point correspondences over multiple frames; and also path consistency check, which examines the accumulated errors along a loop to eliminate inconsistent part map connections. Experiments show our method successfully built detailed 3D maps of indoor environments.

Future work includes the framework of part map management for efficient map merging in large-scale environments.



(a) Seven part maps (b) Connected regions

Fig. 11. Connections of the part maps.

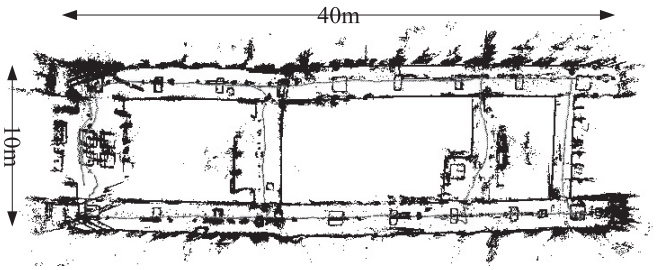


Fig. 12. Whole 3D map generated by map merging.

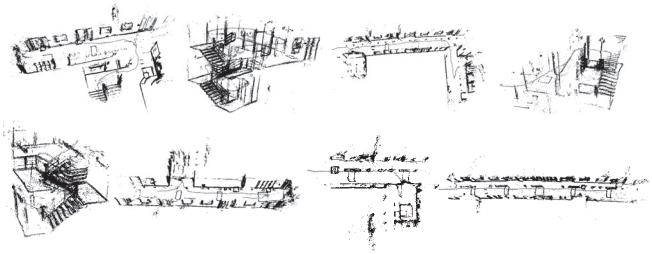
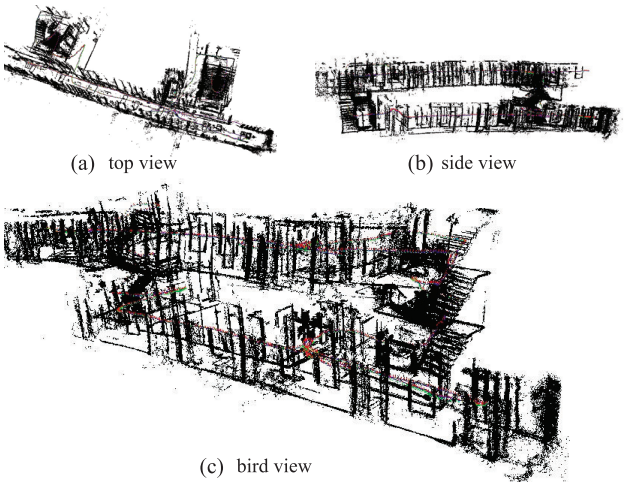


Fig. 13. Eight part maps generated by EdgeSLAM.



(a) top view (b) side view

(c) bird view

Fig. 14. Whole 3D map generated by map merging.

References

[1] P. J. Besl and N. D. Mckay: A Method of Registration of 3-D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256, 1992.

[2] A. Birk and S. Carpin: Merging occupancy grid maps from multiple robots, *Proc. of the IEEE: Special Issue on Multi-Robot Systems*, vol. 94, no. 7, pp. 1384–1387, 2006.

[3] J. Canny: A Computational Approach to Edge Detection, *IEEE Trans. PAMI*, Vol. 8, No. 6, pp. 679–698 (1986).

[4] G. Erinc and S. Carpin: "Anytime merging of appearance based maps," *Proc. of ICRA2012*, pp. 1656–1662, 2012.

[5] D. Filliat: "A visual bag of words method for interactive qualitative localization and mapping," *Proc. of ICRA2007*, 2007.

[6] F. Fraundorfer, C. Engels, and D. Nistér: "Topological mapping, localization and navigation using image collections," *Proc. of IROS2007*, 2007.

[7] K. L. Ho and P. Newman: "Multiple map intersection detection using visual appearance," *Proc. of International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.

[8] A. Howard: Multi-robot Simultaneous Localization and Mapping using Particle Filters, *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.

[9] W. H. Huang. and K. R. Beevers, Topological map merging, The International Journal of Robotics Research, vol. 24, no. 8, pp. 601–613, 2005.

[10] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, and K. Zweig: Cycle bases in graphs: Characterization, algorithms, complexity, and applications, *Computer Science Review*, Vol.3, No. 4, pp. 199–243, 2009.

[11] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Steward: "Map merging for distributed robot navigation," *Proc. of IROS2003*, pp. 212–217, 2003.

[12] K. Konolige, J. Bowman, J.D.Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua: "View-based Maps," *Proc. of RSS2009*, 2009.

[13] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and V. Regis: "Efficient Sparse Pose Adjustment for 2D Mapping," *Proc. of IROS2010*, 2010.

[14] Y. Latif, C. Cadena, J. Neira: Robust loop closing over time, *Proc. of RSS2012*, 2012.

[15] T. Lindberg: "Feature Detection with Automatic Scale Selection," *Int. J. of Computer Vision*, 30(2), pp. 79-116 (1998).

[16] D. Galvez-Lopez and J. D. Tardos: Bags of Binary Words for Fast Place Recognition in Image Sequences, *IEEE Trans. on Robotics*, 28(5):1188-1197, 2012.

[17] D. G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

[18] D. Nister, and H. Stewnius: Scalable Recognition with a Vocabulary Tree, *Proc. of CVPR2006*, 2006.

[19] E.Olson: Recognizing places using spectrally clustered local matches, *Robotics and Autonomous Systems*, 57(12), 1157-1172, 2009

[20] E. Olson, P. Agarwal: Inference on networks of mixtures for robust robot mapping, *Proc. of RSS2012*, 2012.

[21] J. Sivic and A. Zisserman: Video Google: A text retrieval approach to object matching in videos, *Proc. of ICCV2003*, 2003.

[22] N. Snavely, S. M. Seitz, and R. Szeliski: Modeling the World from Internet Photo Collections, *International Journal of Computer Vision*, 2008.

[23] N. Sunderhauf and P. Protzel: Towards a Robust Back-End for Pose Graph SLAM, *Proc. of ICRA2012*, 2012.

[24] M. Tomono: "Robust 3D SLAM with a Stereo Camera Based on an Edge-Point ICP Algorithm," *Proc. of ICRA2009*, pp. 4306–4311, 2009.

[25] M. Tomono: "3D Localization Based on Visual Odometry and Landmark Recognition Using Image Edge Points," *Proc. of IROS2010*, 2010.

[26] J. Wang, R. Cipolla, and H. Zha: "Vision-based Global Localization Using a Visual Vocabulary," *Proc. of ICRA2005*, 2005.

[27] X. S. Zhou and S. I. Roumeliotis: "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case," *Proc. of IROS2006*, pp. 1785–1792, 2006.