

Robot Mapping for Rescue Robots

Nagesh Adluru
Temple University
Philadelphia, PA, USA
nagesh@temple.edu

Longin Jan Latecki
Temple University
Philadelphia, PA, USA
latecki@temple.edu

Rolf Lakaemper
Temple University
Philadelphia, PA, USA
lakaemper@temple.edu

Raj Madhavan
NIST
Gaithersburg, MD, USA
raj.madhavan@nist.gov

Abstract—Rescue robots operate in disaster areas where odometric information is known to be highly unreliable. This paper presents a new approach to odometry independent 2D robot mapping. Scans obtained from different robot positions are assembled to a single global map using alignment based on scan intrinsic information only, namely shape features. The approach detects geometric structures in different scans which are similar by shape and aligns the scans accordingly. The detection of shape similarity is the key to generate global maps even of highly cluttered environments and thus the proposed approach is particularly suitable for applications in the field of rescue robots.

I. INTRODUCTION

Rescue robots are mobile, autonomous robots, which are deployed in disaster areas mainly for search and rescue of victims. Even though actually rescue robots are remotely controlled by human operators, their navigation in such areas is usually extremely difficult. The only information that is available to a human operator is a stream of images transmitted from one or more cameras mounted on the robot. Due to limited view of the cameras, it is an extremely hard task for the operator to keep track of the location of the robot and navigate it. A global map of the robot environment with a robot path marked can be very helpful to the operator. The overview knowledge in the form of a global map is particularly important to localize victims in catastrophe scenarios (e.g., in collapsed buildings) and to ensure that the whole target region has been searched [1]. Moreover, when a victim is found, the map gives a responder team a much better situation awareness than currently used sketches which are manually drawn.

However, the highly cluttered environments (e. g., due to piles of rubble) and scans from sliding and tilted robots make mapping for rescue robots an especially challenging task. In particular, the odometry information is very unreliable. External sensor data like GPS to assist in this task is not available.

Currently, laser range scanners are the most popular sensing devices to generate maps of robot environments, but the existing robot mapping approaches are based on the assumption that consecutive laser scans are highly overlapping. However, when mapping is done in cluttered environments, frequent tilts of the robot and occlusions cause consecutive scans to be very different from each other.

Robot mapping, i.e. constructing the internal map autonomously and self-localization, i.e. localizing the robot within its internal map are of high importance to the field of mobile robotics [2]. Coping with unknown or changing environments requires to carry out both tasks simultaneously, this problem has been termed the 'SLAM problem: Simultaneous Localization and Mapping [3]. It has received considerable attention [2]–[4]. Successful stochastic approaches mostly based on particle filters have been developed to tackle representation and handling of uncertain data, which is one key problem in SLAM [5]. As current stochastic models are powerful, even linking them to a simple geometric representation like reflection points measured by a range sensor already yields impressive results. Though advances in stochastic means have improved the overall performance, they leave the basic spatial (geometric) representation untouched. Since the internal geometric representation is a foundation for these sophisticated stochastic techniques, shortcomings on the level of geometric representation affect the overall performance.

The highly cluttered environment, possible robot tilt, and unreliable odometry information cause the existing SLAM approaches to fail, since they assume large overlaps between scans and also good estimates of robot poses. Also current SLAM approaches fail in conditions where the environment changes rapidly, e.g. in a crowded shopping mall, due to occlusions.

This paper presents an approach that makes it possible to successfully complete the task of robot mapping of highly cluttered environments without any odometry information. We use shape information extracted from range data of robot scans to find and align similar structures among different scans. Shape information abstracts from actual positions of scan points in local scan maps. It allows to identify similar structures with assumption of minimal overlaps between consecutive scans only. The only requirement is the presence of similar structures so that it is possible to identify common parts of the robot environment.

The rest of the paper is organized as follows: Sec. II describes shape information and similarities used, Sec. III describes how we estimate odometry after extracting similar structures. Sec. IV describes global map construction using the estimated odometry. Finally Sec. V presents our experimental results. Our future work directions are presented in

the last section. The values on axes in all the figures are in centi-meters.

II. MATCHING STRUCTURES USING SHAPE SIMILARITY

The input data is a set of points obtained by a laser range scanner. This data contains information of the location of laser beam reflection points, hence it consists of a set of sample points of the environment. We abstract from the position of these scan points by extraction of the shape information utilizing *tangent directions* and *angular histograms*. Preserving the scan order and geometric properties of the scan allows us to match similarities between scans.

A. Tangent Directions

The points on each scan are approximated by polylines using Expectation Maximization and perceptual grouping as discussed in [6]. For each point we find the closest polyline, its tangent direction (angle ranging from 0° to 360°) is stored as the direction property of the point. Since we maintain the order of scan points, we obtain a sequence of tangent directions for each scan. Approximation of point sets with polylines using EM is a robust technique, and hence the sequence of tangent directions represents robust shape information of a scan. Fig. 1(a) shows polyline approximation of points. Two points in different scans are considered similar if their associated tangent directions are within 10° shift.

B. Angular Histograms

For each point in a scan we construct an angular histogram with 36 bins by counting the number of scan points for each bin within a radius of 50 cm. The circular frame and a sample angular histogram are shown in Fig. 1(b). This representation is similar to shape histograms used in [7] and to shape context used in [8]. In our approach a different similarity measure is used. χ^2 statistic used in [8] is not applicable in our case, because most bins are empty. Instead, we use the number of differing bins as the distance between two histograms. Two points in different scans are similar if they differ in at most 2 bins. We consider two bins b_1, b_2 as different if $|b_1 - b_2| > 0.25$ where b_i represents normalized value of the i^{th} bin.

C. Longest Common Subsequence

Once we obtain sequences of tangent directions and angular histograms between two scans, we use a well established algorithmic technique to obtain longest common subsequences (LCS). The LCS algorithm [9] considers two points to be similar based on similarity between both tangent directions and angular histograms as explained in Sections II-A-II-B. This technique finds corresponding points between two scans, but we face two problems here: first there may be no similar shape structures between two scans as illustrated in Fig. 2, and second there may be false positives in shape matching. The first problem can be solved by aligning the query scan to the partial global map

composed of a few recent scans instead of just the previous scan. To reduce computational demands we choose to align the query scan to a map composed of three previous scans. As shown in Fig. 3, this approach makes it possible to find similar structures.

To understand the second problem we can view the matchings generated by LCS as the hypotheses of correspondences. We have to eliminate false hypotheses. The false positives are eliminated using two deterministic filters, explained in Section II-D. To eliminate the problem of rotation of scans we perform the shape matching for several different rotations of the query scan. We chose the rotation that gives us the largest true overlap as the best rotation as defined in Section III.

D. Eliminating false positive correspondences

We eliminate false positives using two filters, namely collinear filter and segment filter.

- **Collinear filter:** Collinear structures, e.g. as gained from subsequent scans of walls of long hallways, do not contain a highly distinct amount of information. We therefore eliminate collinear matches from the match candidates extracted by LCS. If the direction of a correspondence vector (the translation vector between two corresponding points) is within 10° shift to the tangent direction of either of the corresponding points, then the correspondence is considered collinear and removed.
- **Segment filter:** We first divide the corresponding points detected by LCS into parts, called segments, and eliminate *bad* segments. Division of LCS into segments is based on distance. If there is a distance of 30 cm or more between one correspondence and the previous correspondence in either sequence we create a new segment. For each segment we compute a quality measure. Let the segment for which we compute the quality measure be called main segment. We first translate the query scan to the target scan based on the average translation vector calculated using the correspondences in the main segment. After putting the query scan onto the target scan, we compute the average distance between the corresponding points in the main segment. Now we compute average distances for all other segments and eliminate those that are above twice the average distance for the main segment by 20 cm. Thus we eliminate only very bad segments. The quality measure of the main segment is the number of the left over corresponding points. We then pick the segment with highest quality measure which leaves us with LCS correspondences that are good after translation. The effect of eliminating false positive matches obtained by LCS can be seen in Fig. 4.

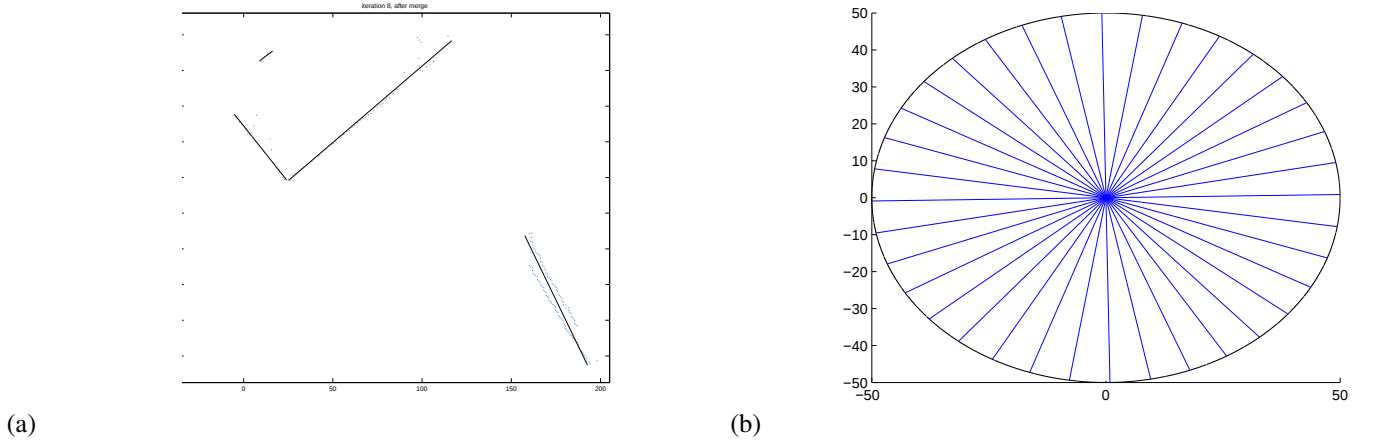


Fig. 1. (a) shows a piecewise linear approximation of points in part of a sample scan obtained using EM. Each point is assigned the tangent direction of the closest line segment. (b) The circular frame used for calculating angular histograms of points.

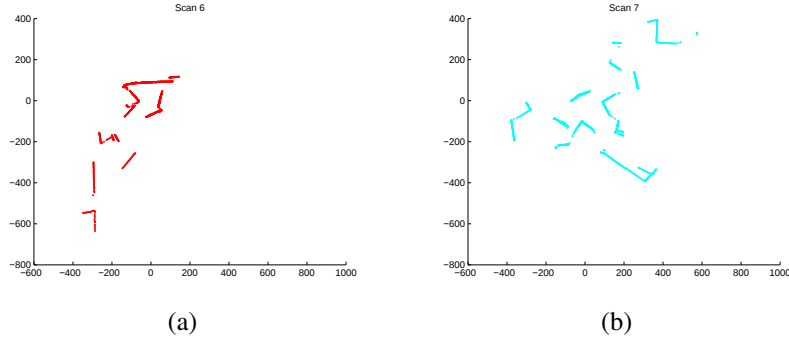


Fig. 2. This figure shows two consecutive scans 6 and 7 in our test data set that do not contain any similar structures.

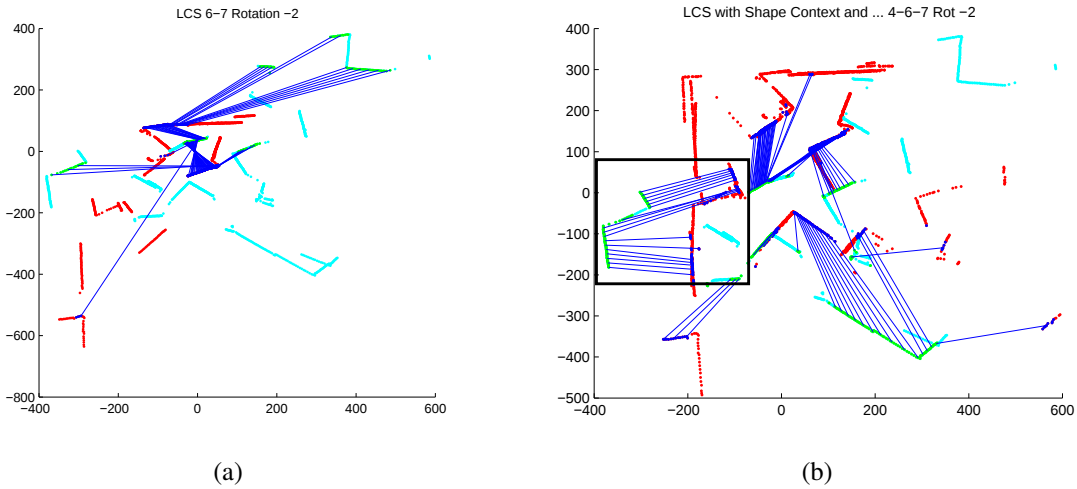


Fig. 3. The corresponding points computed using shape similarity in (a) between scans 6 (red) and 7 (cyan) are all false positives. (b) The rectangular region shows the true positive correspondences between a partial global map composed of scans 4 – 5 – 6 (red) and scan 7 (cyan). The blue lines link the corresponding points.

III. ROBOT POSE ESTIMATION

We compute the optimal translation and rotation for robot pose estimation. For each rotation, we pick the best segment wise translation, and score the rotation using *scan overlap*

measure described below. Then we select the best rotation and the corresponding translation.

- **Segment wise translation:** After eliminating false positive correspondences, the best translation and ro-

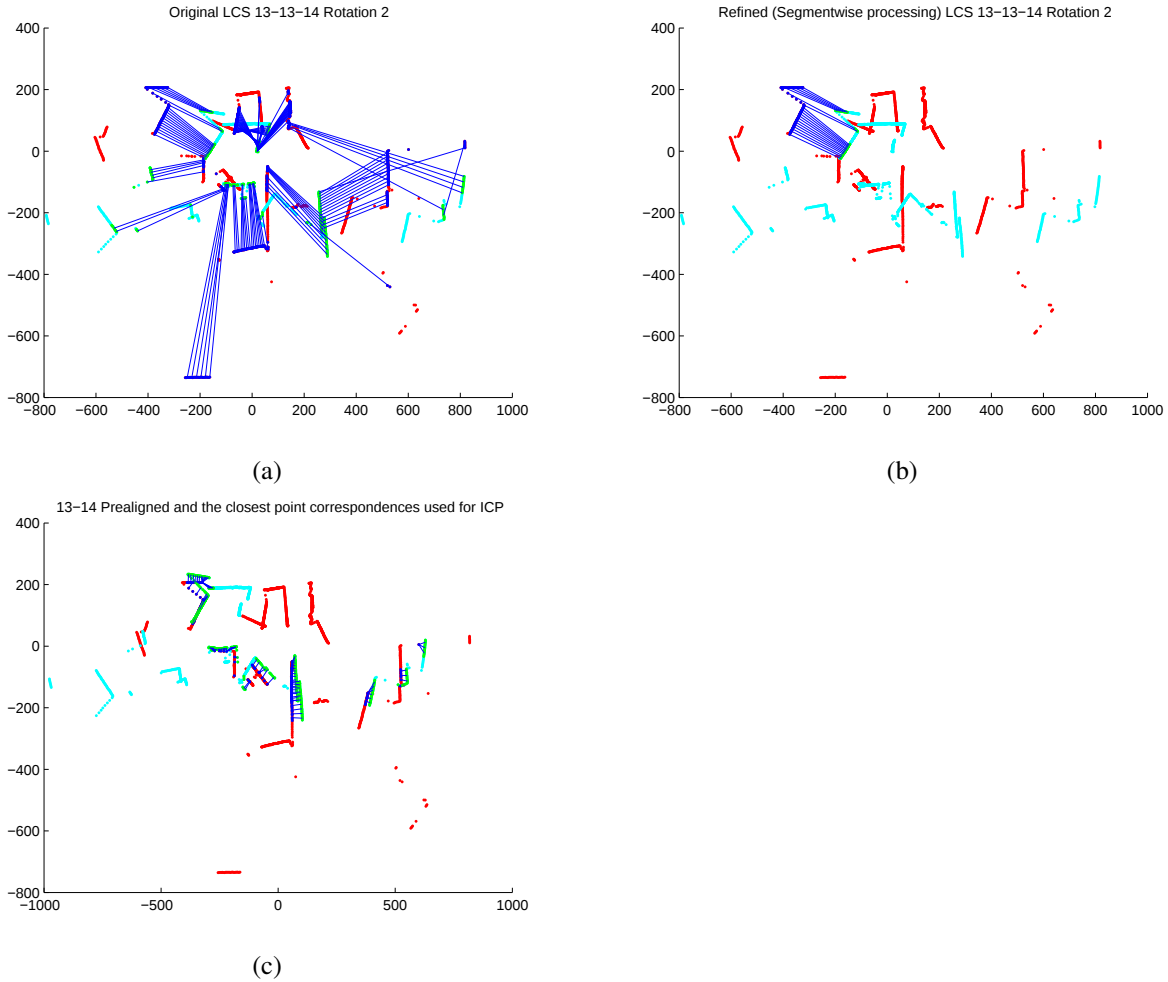


Fig. 4. This figure shows the effect of the segment filter. (a) The corresponding points between scans 13 and 14 extracted using LCS. (b) Actual correspondence after passing through our filters. We can see that only 3 good segments are left. (c) 13^{th} and 14^{th} scans aligned using segment wise transformation described in Section III.

tation of query scan points onto the model scan points are chosen, using the segments computed in segment filtering. For each of these segments the translated query scan is superimposed onto the target scan to compute the average distance between the correspondences in the segment used for translation. We eliminate all other point correspondences that are farther than the average distance. We pick the translation that gives us the largest number of point correspondences. The main difference between segment filter and this step is that in the segment filter false positives are eliminated segment wise, whereas in this case the elimination is done based on point pairs.

- **Scan overlap measure:** As mentioned in Section II-C, we perform the process for several rotations. For these experiments we rotated query scans from -2° to 2° with 1° interval. After we translate the query scan to target scan based on the best translation, we compute the number of points in query scan that are *close*, within a distance threshold to the points in the target

scan. The distance threshold is computed dynamically as follows: We compute the average distance between correspondences in the segment that was picked by the segment wise translation process. The distance threshold is the average distance plus 40 *cm*. We consider only those point correspondences for which the tangent directions of the corresponding points are within 10° shift. The length of this correspondence is the scan overlap measure.

IV. GLOBAL MAP CONSTRUCTION

Currently the best method for mapping robot scans is using particle filters [5]. Particle filtering is a powerful tool for mapping robot scans but it has two major constraints. One is that there needs to be large overlaps between immediate scans and also that the environment is not very cluttered or very dynamic. The other is that the pose estimates are good. We compute very good pose estimates but since the overlaps between immediate scans are small and the environment is cluttered particle filters fail in our

case.

Therefore, we use a deterministic technique to construct the global map and correct the errors in robot poses. We use *Iterative Closest Point (ICP)* [10] and [11]. We align maps by the translation and rotation obtained using our shape similarity approach. The correspondences used for ICP are selected using distance and tangent thresholds. Points from two scans should be below a distance threshold and also within a tangent threshold to correspond. The distance threshold is dynamically calculated using the average distance between the scan points aligned using pose estimates computed. We run the ICP algorithm iteratively as long as the quality of alignment increases with an upper bound of 20 iterations. The quality is measured using scan overlap measure and the average distance among the corresponding points in the scan overlap (overlap distance). If in an iteration the scan overlap measure decreases and the overlap distance increases, we roll back the iteration and break the ICP loop.

V. EXPERIMENTAL RESULTS

We use benchmark data from National Institute of Standards and Technology. NIST collected robot scans in a simulated urban search and rescue arena at 16 different positions. Four scans are taken at each position at four different orientations that differ by 90° . Each complete scan is composed of 1604 readings, 401 in each direction with the angle span of 100° and with 0.25° scan interval. We performed our experiments on the data without any odometry information. Fig. 5(c) shows the global map generated by our approach. For comparison, we show in 5(d) the map based on manually measured odometry. The global map generated using currently the best SLAM approach (based on particle filtering) is shown in 5(b). It was obtained using the CARMEN software described in Section V-B with the input shown in 5(a).

Fig. 6(a) shows the estimated and the manually measured robot poses. Fig. 6(b) shows the error plot of the estimated robot poses. As can be seen, although the error in pose estimate increases in 10-14, our approach is able to recover and significantly reduce the error in 15.

A. Decoupled ICP

Since each scan is composed of four subscans taken after 90° rotations, we perform decoupled ICP after the ICP on the complete scan. The measurements between different regions have overlaps and manually introduced errors due to the data acquisition process. The ICP applied to the complete scan cannot correct these local errors without sacrificing quality of global alignment. Hence we run the ICP algorithm on the four regions separately. We iterate on each region with quality check as explained in the Section IV.

B. CARMEN

CARMEN is a robot navigation software package developed by CMU. It has various modules useful for autonomous robots. The scan matching module of CARMEN is called VASCO and it is used to generate global maps from the robot scans. We input the data from NIST with the estimated odometry computed by the proposed approach to VASCO. Our input to VASCO is shown in Fig. 5(a). As can be seen in 5(b), VASCO was not able to improve the input map (shown in 5(a)).

VI. CONCLUSION AND FUTURE WORK

We have demonstrated that shape features of scans can be very powerful in scan matching in cluttered environments like rescue situations or dynamic environments. Our shape similarity technique is very useful in extracting overlaps among scans even when consecutive maps do not have large overlaps.

Once we get the overlaps using shape we can generate *a priori* distribution of the particles and use the particle filtering approach. We are working on using particle filters with the prior generated by our approach to compute the actual robot pose.

ACKNOWLEDGMENT

We would like to acknowledge the support from the National Institute of Standards and Technology, award Nr. NIST 70NANB5H11119, and discussions with Elena Messina and Adam Jacoff. We thank Dirk Haehnel for his prompt help in using CARMEN's scan matching module.

REFERENCES

- [1] A. Jacoff, E. Messina, and J. Evans, "Performance evaluation of autonomous mobile robots," *Industrial Robot: An Int. Journal*, vol. 29(3), 2002.
- [2] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.
- [3] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. of Robotics and Automation*, 2001.
- [4] D. Hähnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environments," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS'02)*, 2002.
- [5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press Cambridge, 2005.
- [6] L. J. Latecki and R. Lakaemper, "Polygonal approximation of laser range data based on perceptual grouping and em," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006.
- [7] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl, "3D shape histograms for similarity search and classification in spatial databases," in *Advances in Spatial Databases, 6th Int. Symposium, SSD'99*, R. Güting, D. Papadias, and F. Lochovsky, Eds., vol. 1651. Hong Kong, China: Springer, 1999, pp. 207–228.
- [8] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 705–722, 2002.
- [9] E. W. Myers, "An $o(ND)$ difference algorithm and its variations," *Algorithmica*, vol. 1, no. 2, pp. 251–266, 1986.
- [10] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

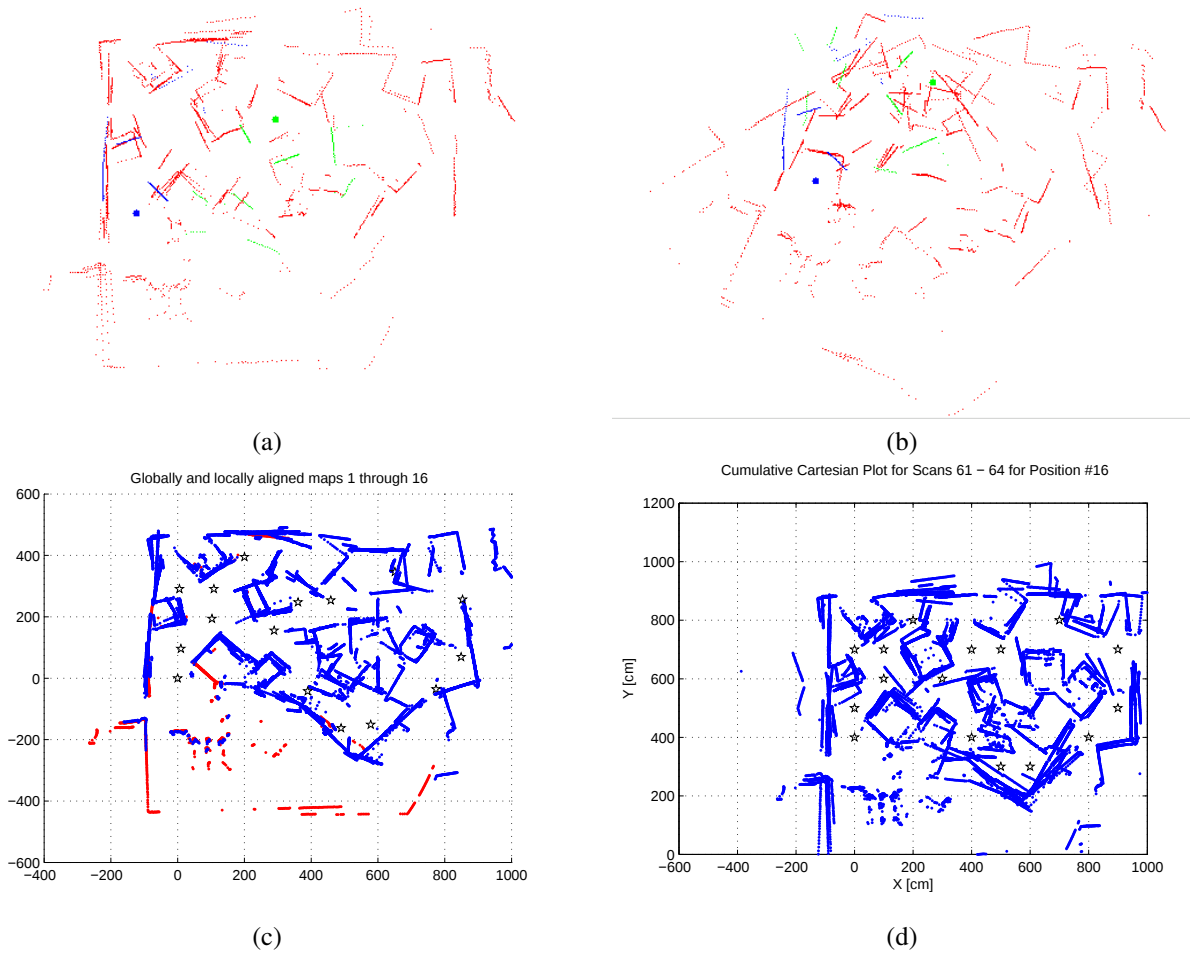


Fig. 5. (a) The input global map of NIST log data with the odometry estimated by the proposed approach. (b) The global map generated by VASCO module of CARMEN. (c) The global map generated by our approach. (d) The global map created by NIST based on manually measured odometry.

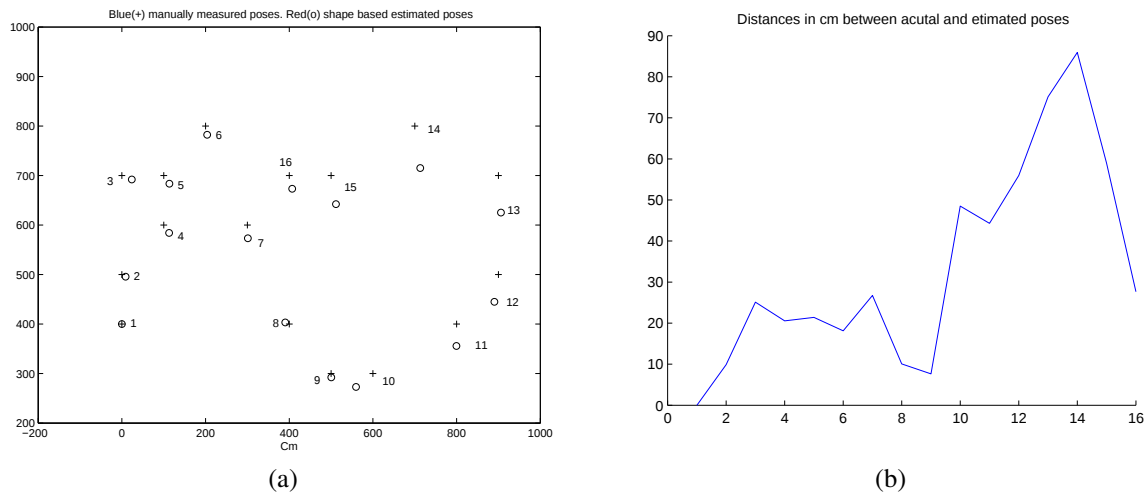


Fig. 6. The distance differences between the estimated (plus) and manually measured (circle) robot poses in (a) and error plot in (b).