Line Segment Based Scan Matching for Concurrent Mapping and Localization of a Mobile Robot

Tahir Yaqub, Michal J Tordon and Jayantha Katupitiya
School of Mechanical and Manufacturing Engineering
University of New South Wales
Sydney, Australia
Email: t.yaqub,m.tordon,j.katupitiya@unsw.edu.au

***Abstract*— This paper describes a new approach to align the laser scans in order to build a consistent map of the environment of a mobile robot. There are no physical landmarks and the environment is completely unknown. Our method finds the nearest matching lines in the scans and resolves the data correspondence problem. The global consistency and final map is achieved by the formulation of geometrical constraints and solving them using conjugate gradient method. The method can handle the multiple loop closing problem. This is easy to implement and fully integrated with a complete navigational architecture. The proposed algorithm is implemented using C++ programming libraries and tested in a real-time robotic simulation environment.**

***Keywords*—Mobile robot, CML, SLAM, Scan matching, Localization.**

I. Introduction and Related Work

Representation of the environment of a mobile robot is of fundamental importance for a successful navigation. This representation is usually called the map of the environment. If there is no prior information, then the robot has to build the map, during the navigation, using its sensors. Over the past decade there has been a lot of work on map building for mobile robots. Some approaches use the landmarks (installed or natural) as a reference for map building while some use the relative positions of landmarks as invariant features for map building. The major problem with all map building approaches is the data association problem also known as correspondence problem. This problem arises due to the fact that landmarks can not be identified uniquely. There always exist an uncertainty associated with it. Scan matching approaches are very popular for the purpose of map building and localization because of the efficiency. By merging laser scans at different locations, a more complete environment presentation can be obtained. Most of the scan matching methods are based on Iterative Closest Point (ICP) algorithm [1], [2] and its variants, the idea borrowed from vision community. In the field of robotics Lu and Milios [3] described a method of data association in their highly influential paper. They were the first to formulate the SLAM problem as a set of links between robot poses and to formulate a global optimization algorithm. A similar approach has been used by [4] for the localization of a mail delivery robot. Another scan matching approach is presented by Biber [5] which is independent of the correspondence problem. Gutman [6] used a combine scan matcher in AMOS project. Another method [7] use the exhaustive search strategy for corresponding points which can be used as an initial alignment. Some other hybrid approaches for environment modelling are Pradalier [8], Tomatis [9], Ortin, [10] and Dufourd [11]. Hough transform based approach of scan matching [12] are computationally very expensive. Some approaches [13] use Expectation Maximization (EM) Algorithm [14] to solve the data association problems in a hill climbing manner but these can not be implemented in an on-line fashion because EM algorithm requires multiple passes. The problem with iterative methods used in most of the scan matching approaches is that we can not use these methods recursively in an efficient manner. This is the reason due to which scan matching approaches are sometimes considered incapable of handling large map building problems. We present a method which reduces the computational burden and suitable for real-time applications. Instead of finding a point to point correspondence like ICP, we extract straight lines from the laser scans and find the rotational and translational shifts for a proper match. This shift is then used to update the pose reported by the dead-reckoning. This reduces the computations by nearly a factor of ten. The beauty of our method is that it aligns the scan locally in a single iteration. For map building however, the global localization is achieved by formulating the geometrical constraints and solving those using conjugate gradient method.

II. Line Segment Based Scan Matching

A. Feature Extraction

Although there could be many important features present in the environment of a mobile robot but the most important features are the straight lines. The obvious reason is that all the man made environments contain walls and other polygonal objects. We use the real-time line extraction algorithm proposed in [15]. This algorithm does the preprocessing for statistical removal of outliers, segmentation and the straight line extraction. Some attributes of the straight lines which can be extracted are length, slope and terminal points. The feature extraction algorithm can also extract the number of lines and corners from each scan. However in the particular method described here, we do not use the slope and corner information. We are using LRF from SICK electro-optics with the resolution of 0.5^o and it covers a field of view of 180^o.

Thus we get 361 range measurements z_i in the form of (r_i,θ). The detailed characteristics can be seen in [16]. We noticed that in practice, the number of lines extracted from the environment are usually not very large. These are of the order of magnitude less than the 361 points obtained in each of laser scan if we were to use the raw data. We also noticed that in a real laboratory like environment, maximum 30 to 40 lines were extracted and in an office like environment, the lines were usually between 20 to 30. Although this also depends on the threshold used in the line extraction algorithm but here we are talking about the situations in which we were able to detect the maximum variation in the environment.

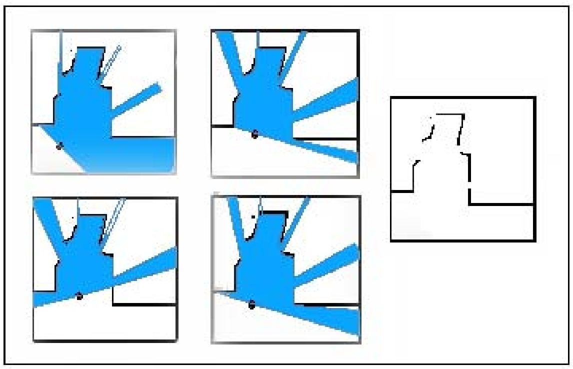


Fig. 1. The simulated Robot with a laser scanner at four different positions in an irregular indoor environment shown right most. The array of remaining four figures is as this Top Left: Initial Pose Top Right: Final Pose if there would not be errors Bottom Left: Final Pose with both Rotational and Translational errors Bottom Right: Final Pose with only Translational error

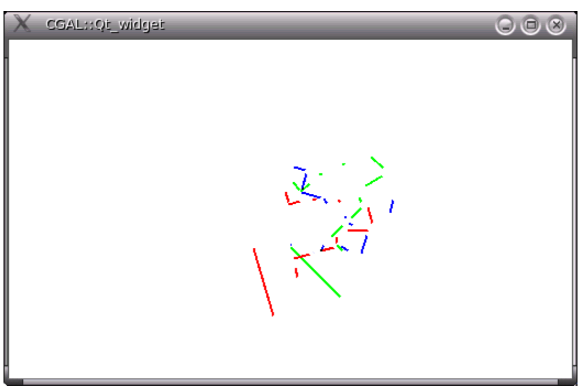


Fig. 2. The reference scan is shown in green which is used for global optimization. The scans shown in red and blue are matched and used for local pose corrections.

B. Scan Acquisition

A robot is shown in Fig.1. The details can be seen there. At initial position, robot scans the environment and extracts the straight lines as shown green in Fig.2. This is the first reference scan and would be used for global optimization. For local scan matching, we move this scan in the frame of the second scan. The moved scan is shown in red and the new scan is shown in blue. This process is carried out at every motion command. The scans shown in Fig.2 are corresponding to the final position shown in bottom left position of Fig.1. The second scan has both rotational and translational errors with respect to the reported pose from the odometry. The rotational error is around 40^o which is considerably large. This can be considered as one of the the difficult scenario as far as the scan matching is concerned.

C. Scan Matching Algorithm

The purpose of scan matching is to improve the pose estimation obtained by dead reckoning (odometry) and hence accurately localize the robot locally. All the scan matching approaches assumes that an initial guess is available for matching the scans. In most robotic applications this approximate pose is obtained by dead-reckoning using the wheel encoders. But since the error in this estimate accumulates without bounds, therefore we have to make the frequent corrections. These corrections are made using scan matching. The problem is to align the scans so they can be merged. We use a two step procedure to align the scans. First we find the rotational shift and the overlap region of two scans simultaneously and establish a data correspondence and finally we determine the translational shift using the established overlap region. The overlap region precisely refers to those line segments which corresponds to same physical objects in the environment. To find the nearest line in the first scan we use three points from each line segment of the other scan. The points we use are both the end points and the mid-point of all the line segments in the scan. For these three points, we find the closest line of the other scan. If two out of three points of a line segment finds the same line of the other scan as the closest line, then both the lines are considered as part of the common region visible from both the poses and selected for further processing. Since the two scans differ in rigid motion, therefore without the loss of generality, we can say that the angles between the lines in an overlapping portion of the scan would be the same within a small variance. Hence we find the rotational shift of the scans. Three of most commonly used algorithm ICP, IDC [17] and one proposed by Cox [18] find the distance of every point to the closest line segments. If there are p points and l line segments, the complexity of this operation is $O(pl)$. This varies with the number of points. The advantage of our method is that we use only three points from each line segment and hence the complexity is $O(3l)$ and is fixed. The substantial efficiency we get due to the feature extraction is at the cost of some loss of information during the feature extraction process. However in practice, this does not cause serious problems due to the differences in data presentation of camera and a laser range finder. Let first scan has $l1$ segments $\{s_1, s_2, ..., s_{l1}\}^{[1]}$ and second scan has $l2$ segments $\{s_1, s_2, ..., s_{l2}\}^{[2]}$, and $l1 < l2$. Note that the superscript [1] denotes the scan number (first/second) of two scans. Let us now consider that there exists lines as entities of infinite length passing through the start and end points of each segment of both scans. It should be noted that whenever we talk about the intersection of segments, actually this refers to the intersection of corresponding infinite lines. Since all the segments of both scans are in the same reference frame, hence the corresponding lines will either be parallel or intersect at some point. Let us say that $\{P_{1i}, P_{2i}, P_{3i}\}^{[1]}$ are the three points on the i_{th} segment of the first scan, and all three points (or at least 2 out of 3) find the same line of the other scan as the closest line (corresponding to this segment on the other scan). If every one of three points finds a different line segment as closest then all three points would not be considered for further analysis. Further suppose that θ_i is the angle between this segment and the closest line (segment) on the other scan. Let k are the number of line segments which have been chosen based on 2 out of 3 scoring scheme and are the potential candidates for overlapping region. Now at this stage we have to find the outlier amongst the k segments chosen for further processing. This is done by taking into consideration the fact that the two scans only differ due to a rigid motion and hence in order to be part of overlapping region, $corresponding\ line\ segments$ on both scans must have approximately same length (within a very small threshold). Secondly amongst $\theta_i.....\theta_k$, only those angles are realistic which fall within a small threshold (ideally the same) because these are the angles between all those segments which are suppose to refer to the same physical objects in the environment. We discard the angles and the associated segments if the angle value is far apart the mean of the rest. Therefore we first find the largest occurring angle (there must be one if we have an overlapping region) within a threshold and then take the mean to get the θ_r as the rotation component of the scan. Now all overlapping segments are rotated by this much of angle. This is equivalent of applying this simple transformation on two end points of all the segments. Let (x, y) is any point(start and end point in our method) on a segment at a local origin (x_p, y_p). This is to be rotated through the currently obtained θ_r in order to align with the other scan at the same local origin. Then the new position of (x, y) is denoted by $(x^{'}, y^{'})$ and can be found as:

$$\begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta_r & -sin\theta_r & x_p(1-cos\theta_r)+y_p sin\theta_r \\ sin\theta_r & cos\theta_r & y_p(1-cos\theta_r)-x_p sin\theta_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

we find the new positions only for the start and end points of all overlapping segments of the scan to be aligned. Since we are interested in the overlapping region as defined above, therefore within this region, the rotated end points correspond to the same physical locations. If the coordinates of other scan are taken as $(x^{''}, y^{''})$ and noting that both the scans now differ only in translation, we can simply take the differences in $(x^{'}, y^{'})$ and $(x^{''}, y^{''})$ and solve using least square to find the final translation of the scan. If $\Delta x = x^{''} - x^{'}$ and $\Delta y = y^{''} - y^{'}$, we notice that the required translation is simply the mean of these values for all segments. Finally we find the final position (x_t, y_t) of scan points by the following transformation.

$$\begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix}$$

The overlapping region of both scans has been aligned now and at this stage we can double check by using a point to point match between the end points of aligned segments if desired. Otherwise the pose correction of $(\Delta x, \Delta y, \theta_r)$ should be made to the pose reported by the odometry.

We now describe the algorithm in a formal way.

Algorithm 1: Line Segment Based Scan Matching

Data: 2 sets of line segments with lengths extracted from 2 different scans and an initial guess of their relative position
Result: Improved Estimate of the Pose Difference between two scans
Initialization : Move the first scan to the frame of reference of second scan using the guess from odometry;
while *NOT the last line reached-:of the scan having less lines* **do**
    Find the nearest line from both end points and middle point of every line.;
    if *two of the three points finds the same line as nearest* **then**
        Take this line for further processing
    end
    else
        Ignore the nearest line

    Take one line from both scans starting from the left OR right depending on the direction of rotation from odometry.;
    calculate the angle of intersection (on the plausible side) of lines and add this angle to a DEQUE;
    Move to the next pair of lines from both scans;
end
Locate, in the final DEQUE of angles, the most frequent angle within a tolerance (say it the candidate angle);
for *all lines corresponding to this frequently occurring angle* **do**
    select the sets of lines occurring consecutively within each of the both scans;
    for *all lines from both sets* **do**
        if *lengths of corresponding lines are same within a tolerance* **then**
            Add these lines to overlap region;
        else
            Discard the corresponding lines from both scans

    end
    Also add the single line next OR after the lines comprising the overlap region (depending on the plausible direction) from both scans to the overlap region if both the lines have the intersection equals to the candidate angle (however length may vary)
end
Rotate one of the scan by this angle and solve for the difference in (x,y) coordinates of the end points of lines segments (from both scans) which have been included in the overlap region.;

III. CONCURRENT MAPPING AND LOCALIZATION

There are many techniques which use a known map for the localization and navigation of mobile robot (not discussed here). Some are based on scan matching such as Gutmann and Nebel [19]. However it is a highly desirable capability for a robot to build its own map during the navigation process. It is called simultaneous localization and map building (SLAM) [20], [21] or concurrent mapping and localization CML [22]. These techniques are based on the Extended Kalman Filter. Although the techniques based on Kalman filter are considered efficient but have number of limitations such as linearization and quadratic update complexity. Another alternative is based on the formulation of Lu and Milios [3] as described earlier.

Our method is also loosely based on this approach. However we use *conjugate gradient* to solve the constraints. The conjugate gradient is the most prominent iterative method to solve the sparse system of linear equations. We do not describe the method here, and refer to an excellent tutorial by Shewchuck [23] but we do explain our strategy in a nutshell as follows:

- Store both the corrected pose and the corresponding lines (local map) in a map implementation. (map in the sense of a data structure of C++ STL library). Since lines can be considered as the entities joining two end points, therefore we store set of these two points against the corresponding pose.
- The rotational component in pose causes non-linearity therefore approximate the pose difference using Taylor series approximation.
- Form the pose constraints using first reference scan and using the fact that the relative pose changes are invariant with respect to the reference scan (first scan). We can take first scan as reference without the loss of generality.
- Solve the linearized constraints using conjugate gradient method and get the optimized set of poses.
- Register the line segments associated to each pose *at the pose* obtained after optimization.

We solve for the correspondence of physical objects at each step but do not register the line segments as global objects and hold them until the constraint optimization has been carried out. Loop closing is not problematic because we can not find the multiple correspondence at any stage. Since we do not take along a huge set of point landmarks, instead we use a relatively much smaller set of end points of line segments, therefore memory requirements are also not much of real concern.

IV. EXPERIMENTAL VALIDATION

We use a real-time robotic software Player/Stage [24] for the experimentation. The advantage is that we can get the truth device for comparing the results. The feature extraction algorithm was implemented using C++ Standard Template Library (STL) and the scan matching algorithm uses the state of the art Computational Geometry Algorithm Library (CGAL). Matrix operations were performed using GNU Scientific Library. Like Ref. [18], we also feel enthusiastic about the capabilities of C++ language for robotics applications [25]. This language and the associated libraries available provide excellent capabilities to implement robotics algorithms. For testing we use two environments in addition to the one used during the algorithm development process, which is shown earlier in section-II. Firstly we use a single loop obstacle free environment shown in Fig. 4 (a), and build the map using only the odometric information. This map is badly misaligned and shown in Fig. 3. Then we use our algorithm to build the map which is shown in Fig. 3 (b). The robot took a complete anti-clockwise trip starting from lower left corner. During this run, 38 scans were taken and analyzed. The map built is shown in Fig. 4 (b). Secondly we use a more complex environment for map building shown in Fig. 5. The dimensions of the environment

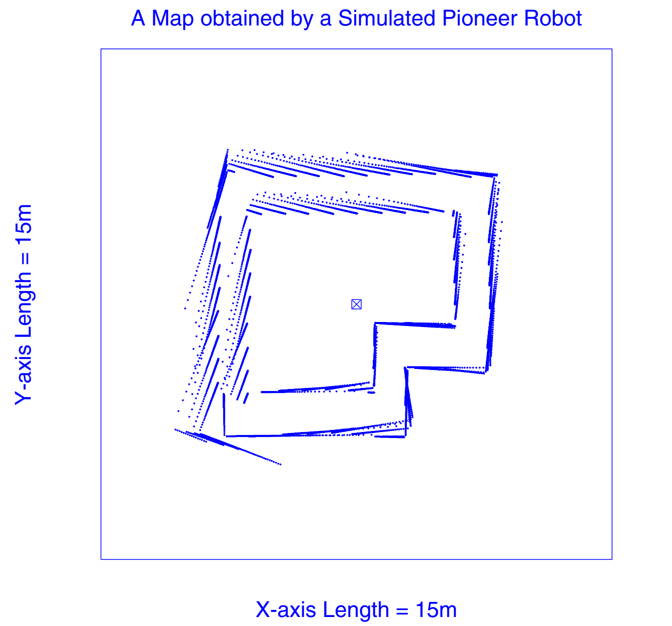
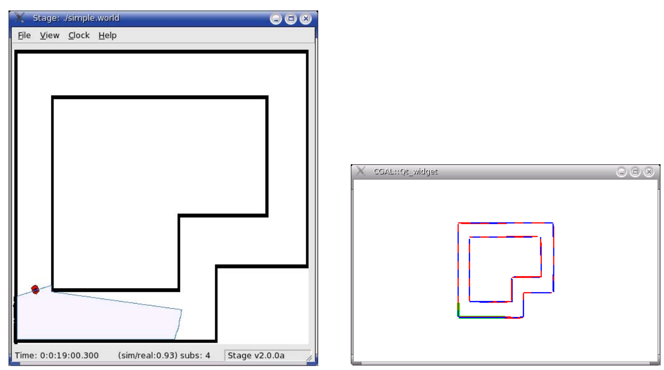


Fig. 3. The map of an environment obtained by using the odometric information only. The robot started and stopped at lower left corner after an anti-clockwise run of 38 scans. Here all the scan points are shown instead of the line segments for clarity



(a) (b)

Fig. 4. (a)The Robot at the starting position for map building with first reference laser scan , (b) Map obtained by scan matching. The reference scan is shown in green, then alternate scans are shown in blue and red colors. CGAL Library is used for implementation of the algorithm. Actual dimensions are shown in Fig. 3.

and the final map are shown in Fig.7. The complete run and mapping took nearly 20 minutes. The strategy we followed was to move the robot in a left wall following manner. The trajectory of the robot is not shown here but it can be guessed by observing the starting point shown in the figures and the left wall following behavior. Local obstacle avoidance was carried out using the Vector Field Histogram [26]. This method is good but has the problem of *local minima*, where the robot traps and keep moving in circles. These traps in local minima were used to check the robustness of the algorithm and it happened quite a few times during the run but even then we were able to capture all the fine details in the environment and built a map consistent with the environment as shown in Fig.6 and Fig.7.

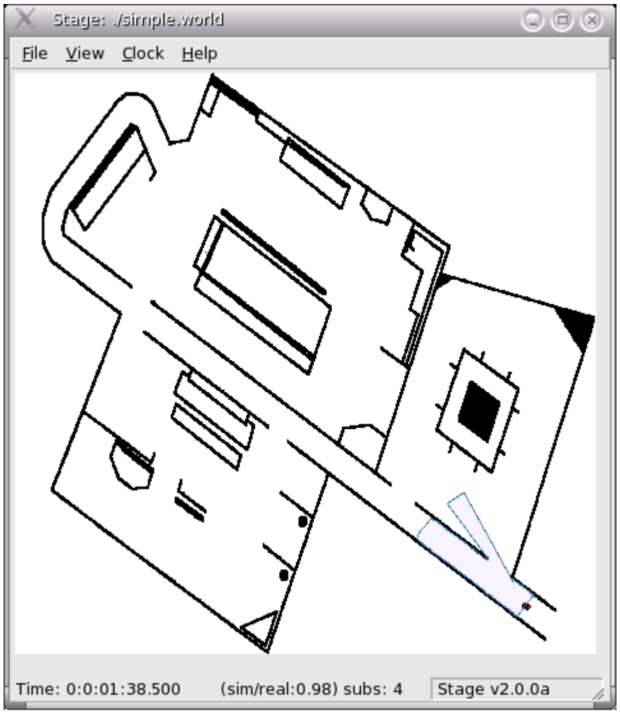


Fig. 5. A simulated pioneer robot is starting mapping of a more complex environment having multiple loop closing points. The size of environment is shown in Fig. 7.

V. Conclusion and Future Work

The main contribution of this paper is a line segment based scan matching algorithm for real-time concurrent map building and localization. The experimental result show that the algorithm is fast and robust and can build the high quality maps. This algorithm can easily be integrated into the hardware because of its modular implementation structure. The memory requirements are not much because we do not store the history of raw scan and only a part of it is stored. The method is capable of handling the multiple loop closing situations. The simulation results are encouraging and currently we are implementing this on a wheelchair.

Acknowledgment

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

[1] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings of IEEE International Conference on Robotics and Automation, 1991.*, pages 2724–2729 vol.3, 1991.

[2] P. J. Besl and H. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992. 0162-8828.

[3] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping, 1997.

[4] K. O. Arras and S. J. Vestli. Hybrid, high-precision localisation for the mail distributing mobile robot system mops. In *Proceedings of IEEE International conference on Robotics and Automation ICRA.*

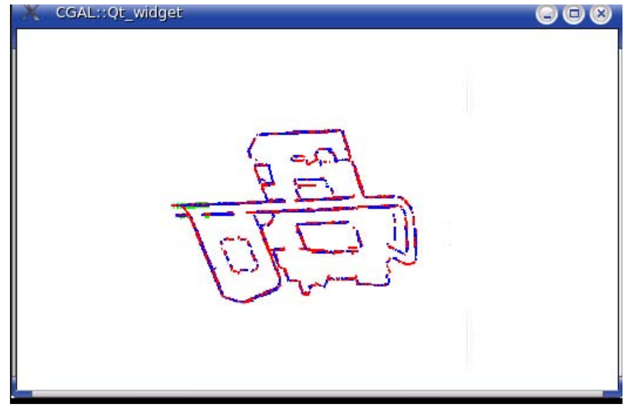


Fig. 6. On-Line Registration of scans obtained by scan matching. The first reference scan is shown in green, then alternate scans are shown in blue and red colors.

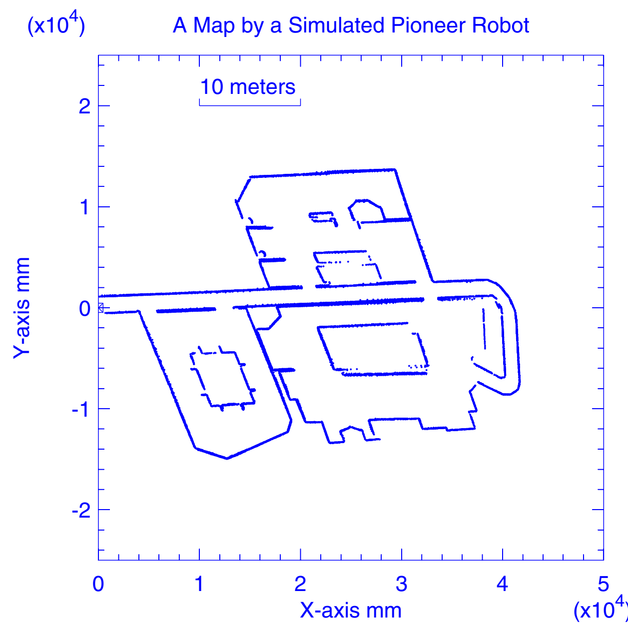


Fig. 7. The Final Map of the environment shown in Fig. 5.

[5] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743–2748, 2003.

[6] J. S. Gutmann and C. Schlegel. Amos: comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the first Euromicro Workshop on Advanced Mobile Robot*, pages 61–67, 1996.

[7] Chen Chu-Song, Hung Yi-Ping, and Cheng Jen-Bo. A fast automatic method for registration of partially-overlapping range images. pages 242–248, 1998.

[8] C. Pradalier and S. Sekhavat. Concurrent matching, localization and map building using invariant features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 514–520 vol.1, 2002.

[9] N. Tomatis, I. Nourbakhsh, K. Arras, and R. Siegwart. A hybrid approach for robust and precise mobile robot navigation with compact environment modeling. In *Proceedings of IEEE International conference on Robotics and Automation ICRA 2001.*

[10] D. Ortin, J. Neira, and J. M. M. Montiel. Relocation using laser and vision. In *Proceedings of IEEE International conference on Robotics and Automation ICRA*, volume 2, pages 1505–1510, 2004.

[11] D. Dufourd, R. Chatila, and D. Luzeaux. Combinatorial maps for simultaneous localization and map building (slam). In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1047–1052 vol.2, 2004.

[12] M. Tomono. A scan matching method using euclidean invariant signature for global localization and map building. In *Proceedings of IEEE International conference on Robotics and Automation ICRA 2004.*

[13] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Em, mcmc, and chain flipping for structure from motion with unknown correspondence.

[14] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society Ser. B 39:1–38*, 1977.

[15] Yaqub T. and Katupitiya J. Modelling the environment of a mobile robot using feature-based principal component analysis. In *IEEE interantional conference on Robotics, Automation and Mechatronics"*, Bangkok, June 2006.

[16] Ye Cang and J. Borenstein. Characterization of a 2d laser scanner for mobile robot obstacle negotiation. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 3, pages 2512–2518, Washington DC, May 2002.

[17] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *CVPR94*, pages 935–938, 1994.

[18] I. J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *Robotics and Automation, IEEE Transactions on*, 7(2):193–204, 1991. 1042-296X.

[19] J. Gutmann, T. Weigel, and B. Nebel. A fast, accurate, and robust method for self-localization in polygonial environments using laserrange finders. *Advanced Robotics*, 14(8):651–668, 2001.

[20] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainly. *Int. Journal of Robotics Research*, 5:56–68, 1986.

[21] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In *Proceedings. ICRA '00. IEEE International Conference on Robotics and Automation, 2000*, volume 2, pages 1009–1014 vol.2, 2000.

[22] J. Leonard and H. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D. Koditschek J. Hollerbach, editor, *International Symposium on Robotics Research*, 1999.

[23] J.R. Shewchuck. An introduction to conjugate gradient method without the agonizing pain. 1994.

[24] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, pages 317–323, Coimbra, Portugal, 2003.

[25] Jayantha Katupitiya and Kim Bentley. *Interfacing with C++ : programming real-world applications*. Springer Berlin Heidelberg, New York, NY, 2006.

[26] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on*, 7(3):278–288, 1991. 1042-296X.