The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Loop Exploration for SLAM with Fusion of Advanced Sonar Features and Laser Polar Scan Matching

Fredy Tungadi and Lindsay Kleeman

Abstract—**SLAM is a well studied technique for robots to build a map of environments while at the same time keeping track of their pose (position and orientation). However SLAM does not provide control approaches for how the robot moves around the environment. This paper presents an integrated approach to create a fully autonomous exploring and mapping robot. An EKF-SLAM approach is used to fuse Advanced Sonar and Laser Scan-Matching. This also tackles the problem of map-drifts in some types of environment where lasers do not supply sufficient information in some directions such as along a corridor. In addition, the proposed exploration algorithm takes advantage of the characteristic of the Voronoi Graph to enable the robot to strategically explore the environments in a loop-closing fashion and safe manner. By revisiting areas to close loops as early as possible, the robot can build a more stable map incrementally while still reliably tracking its pose. Experimental results of the integrated approach are shown to demonstrate the algorithm provides real-time exploration of a mobile robot in an initially unknown real environment. Experimental comparisons of exploration strategies *with* and *without* early local loop closing demonstrate the benefits of the approach in the map quality.**

I. Introduction

In many cases, a mobile robot needs to know its pose and the location of obstacles/objects surrounding in order to perform useful operation autonomously in an initially unknown environment. The SLAM (Simultaneous Localisation and Mapping) algorithms [1-6] provide solutions to this problem. Recently, there has been significant progress on SLAM, though the solutions do not provide motion control of the robot and as a result, the robot has to be driven manually to produce useful results. An exploration procedure is required to create a complete autonomous exploring and mapping robot.

The most primitive exploration strategies are based on random-walks and wall-following. These techniques however, are generally not suitable in complex environments, because they cannot guarantee map completeness and are also inefficient. Path planning is needed to be incorporated in an exploration strategy to make it more efficient and effective. One example of this strategy is the Frontier-Based exploration [7], where occupancy grids are used to detect the boundary between explored and unexplored cells (called frontiers) and then a grid-based path planning technique such as distance transform [8] is used to guide the robot to each frontier. In addition, similar approaches such as Voronoi-Graph-Based exploration [9] uses Voronoi Graph to explore unknown nodes in the graph and feature-based exploration [10] uses exploration paths that are dictated by mapped geometric features to explore environments.

These approaches do not adequately take account of the fact the path executed by the robot influences the quality of the map which is built by the SLAM algorithm. More recent works have focused on minimising the robot's pose and the map uncertainties while exploring environments. Some researchers use learning algorithm such as reinforcement learning [11] and neural dynamics [12] to learn control policy that minimises uncertainties. This however depends on past experience and therefore may require extra time before the robot finds the optimal way of exploring environments. Another technique [13] uses the Extended Information Filter to compute a multi-step trajectory that minimises the uncertainty. This method assumes the map is a collection of point landmarks rather than a collection of raw laser scans or image data and therefore cannot be applied to scan matching SLAM [2-4] which is chosen in this paper for its robustness to data association errors.

This paper shares the same idea with active loop-closing exploration [14], in the sense that the robot needs to close loops while exploring environments in order to reduce uncertainties. The approach in [14] maintains both an occupancy grid map and a topological map while exploring an environment. These maps are then used to identify the possibility of loop closing. The loop is detected when the shortest path between current pose and previously visited locations is large in the topological map but small in the grid map. Whenever the loop is not detected, the robot uses the frontier-based technique for exploration. The main advantages of this method are that it is directly applicable to scan-matching SLAM and there is no extra time for learning required to acquire a consistent and accurate map. However the loop is only detected when the robot is already almost closing the loop, so there is a possibility that the robot omits a loop. Furthermore, it does not attempt to execute the loop-closings strategically, which means that it may choose poor visited areas for closing the loops. In contrast, the approach in this paper allows the robot to attempt to close smaller loops first before attempting to close bigger loops. As

Manuscript received March 1, 2009.
F. Tungadi is with the Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Victoria, Australia.
(E-mail: fredy.tungadi@eng.monash.edu.au).
L. Kleeman is with the Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Victoria, Australia.
(E-mail: lindsay.kleeman@eng.monash.edu.au).

shown later in this paper experimentally, this strategy allows a less risky and more stable exploration.

The paper also contributes by presenting an integrated fusion of Advanced Sonar and Laser Scan-Matching SLAM – Exploration algorithm / strategy that utilises Voronoi Graph to strategically explore environment in a loop-closing fashion and thus ensures an accurate and consistent map-building process. This paper also presents an algorithm to automatically extract loop-paths from the Voronoi Graph.

This paper is organised as follows. The robot configuration is briefly described in section II, the SLAM algorithm used for exploration is described in section III, and then in section IV, the exploration strategy including loop-path extraction is discussed. Finally, section V presents the experimental results of the proposed integrated SLAM-exploration technique in real-life environments.

II. ROBOT CONFIGURATION

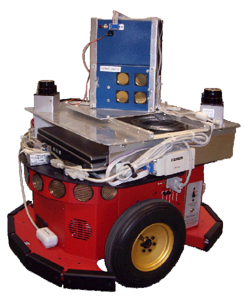


Figure 1 – The experimental mobile robot with front and rear laser range finders and Advanced Sonars.

The robot used for experimental work is shown in Figure 1. It has two Hokuyo URG Laser Range Finders and two Advanced Sonar systems which are mounted on an ActivMedia Pioneer 3 DX mobile robot. Each laser has a field of view of 240 degrees and an angular resolution of 0.36 degrees with a scanning refresh rate of up to 10 Hz. The laser range finders then provide the robot with a 360 degree field of view with reported ranges between 20 mm to 4000 mm. The advanced sonar systems [15] report accurate range and bearing as well as classification of targets. The maximum range is 5 meters with error standard deviation of 0.2 mm for range and 0.1 degrees for bearing. The classification types are planes, right angle concave corners and edges. The Advanced Sonars are positioned to face 45 degrees and -135 degrees with respect to the robot coordinate frame to allow detection as well as range and bearing measurement of wall moulding and door frames in corridors. All the measurements are time stamped to allow time synchronisation of sensors data.

III. FUSION LASER PSM AND ADVANCED SONAR SLAM

The SLAM is implemented using Extended Kalman Filter (EKF) which is employed based on the description by A. Davison [6] where all map features are included in SLAM state vector and updated on each observation step. The prediction model is derived as in [1], where the error model assumes error sources are additive white noise on the wheel separation as well as the left and right wheel distance measurements.

The landmarks for SLAM are defined as two types namely templates of the two lasers' raw-scans data (reference scans) and a collection of advanced sonars' point features (corners and edges). The lasers' landmarks are collected approximately every meter of robot travel and observed by a process of Polar Scan Matching (PSM) [3] for multiple lasers [2] at approximately every 250ms. PSM refers to the process of aligning an observed set of scan-points with a reference set of scan-points. PSM aligns the current scan with respect to the reference scans by minimising the sum of square range residual. After each successful PSM, the reference scans are updated to include the previously unseen features to allow the environment to be mapped more completely and hence to allow better path planning. In contrast, the advanced sonar corner and edge features are collected and updated as soon as they are detected and classified.

The purpose of using both lasers and advanced sonars is to increase the accuracy of SLAM by increasing the variety of measurements. The advanced sonars are mounted at a fixed facing angle, and therefore do not provide enough feature detection and re-observation for SLAM alone. It is feasible to scan the sonar across a wide pan angle, but this slows the sensor cycle considerably. Laser scan-matching may suffer from ill-conditioned scan matches in some environments for example, in a featureless corridor. The scan-matching may have a large positional error in the direction of corridor because the two scans taken from different positions look alike and so the corresponding scan-points pair do not necessarily represent the same physical point in the environment. Proper error modelling of the scan matching such as in [21] can cope with the problem but it makes the EKF rely on inaccurate odometry and therefore causes the estimation to accumulate error especially when the robot is travelling on a rough or slippery surface. In the implementation, the lasers provide most of the measurement for pose correction and the advanced Sonars complement the lasers by providing information that is useful when the lasers fail to perform accurate scan-matchings. The sonar can produce range and bearing angles to small corner cube type targets naturally occurring in doorjambs and other corridor wall features not resolvable by the laser. The fusion of the measurements is carried out with an EKF.

The augmented state vector containing the state of the robot (θ_v , x_v , y_v) and the state of landmark locations is defined as follows:

$$X = [\theta_v, x_v, y_v, p_1, p_2, ..., p_n]^T \tag{1}$$

where the lasers' landmarks are represented by the pose of the centroid of the reference scans in global coordinate frame, $p_i = (x_{Li}, y_{Li}, \phi_{Li})$ and the advanced sonars' landmarks are represented by a two dimensional point in global coordinate frame, $p_i = (x_{si}, y_{si}, 1)$. The observation model for the pose of the laser landmark with respect to the robot is calculated as follows:

$$H_L(k) = \begin{bmatrix} x_{hi}(k) \\ y_{hi}(k) \\ \phi_{hi}(k) \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} (x_{Li}(k) - x_v(k))\cos(\theta_v(k)) + (y_{Li}(k) - y_v(k))\sin(\theta_v(k)) \\ -(x_{Li}(k) - x_v(k))\sin(\theta_v(k)) + (y_{Li}(k) - y_v(k))\cos(\theta_v(k)) \\ \phi_{Li}(k) - \theta_v(k) \end{bmatrix}$$

while the observation model for the advanced sonar measurements is calculated as follows:

$$H_s(k) = \begin{bmatrix} range_i \\ bearing_i \\ 1 \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} \sqrt{(x_{si}(k) - x_v(k))^2 + (y_{si}(k) - y_v(k))^2} \\ a\tan 2\left(\dfrac{y_{si}(k) - y_v(k)}{x_{si}(k) - x_v(k)}\right) - \theta_v \\ 1 \end{bmatrix}$$

As in previous work [2], standard EKF-SLAM validation gate (nearest neighbour) and update equations are applied. It is important to note that each update will update the entire covariance matrix and the entire state vector.

Each stored feature is tagged with their corresponding types and only associated with new measurements with agreed feature types. Simple heuristic error modelling as in [2] is used for lasers scan matching measurements and advanced sonars' error modelling of [1] is applied. More detailed explanation on EKF-SLAM implementation can be found in [6]. The example experimental result of the SLAM implementation in a corridor environment is shown in Figure 2. In Figure 2, phantom advanced sonar features are observed outside the physical corridor due to multi-path echoes. These features are not used for navigation and associated with new measurement only if they are both echoed features.

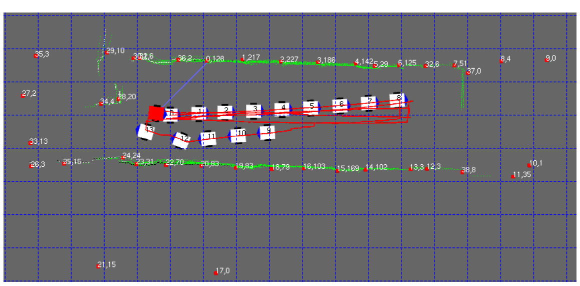


Figure 2 – Example of result of fusion advanced sonar and laser scan matching SLAM in a corridor environment. The red dots are the advanced sonar point features which represent wall moulding and door jambs in the corridor while the green dots are the lasers' map features which represent the wall in the corridor. The grid size is 1m.

A SLAM experiment with and without the advanced sonars in a long laser-featureless corridor was carried out to further demonstrate the benefit of the fusion system. The experimental result is shown in figure 3. The building plans are shown in black whilst the robot measured map is green for laser and red for sonar. The length of the corridor was measured from and to the wall perpendicular to the corridor which exists at the start and at the end of the corridor. The length of the corridor was ~22.3m as measured using a tape measure. Figure 3a shows the final result of the fusion of laser PSM and advanced sonar features SLAM in the laser-featureless corridor, and the length of the corridor was found to be ~22.3m. While, figure 3b shows the result of laser PSM SLAM in the laser-featureless corridor while the robot was still inside the corridor and figure 3c shows the final results when the robot finally fixed the error after coming back to the starting position. In the first experiment (with Advanced Sonar), thanks to the assistance of advanced sonars the robot was always able to localise well in the corridor and the total error is found to be less than 0.5%. While without the help of advanced sonar, the drift in the direction of the corridor is found to be as large as 1m (~5% error) when the robot was still inside the corridor and as small as 0.2m (~1% error) when the robot is finally back to initial position and fixed the error after a successful PSM, thanks to the perpendicular wall at the start of the corridor. These results show the significant improvement of incorporating advanced sonar features into the classical scan-matching SLAM.

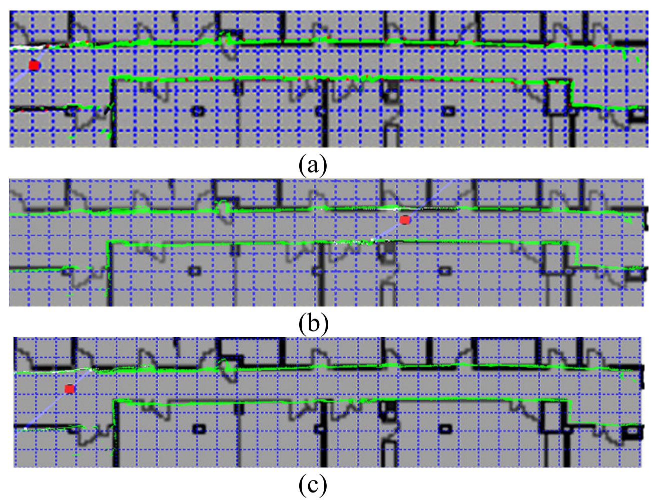


Figure 3 – (a) Final map result of the fusion of laser PSM and advanced sonar features SLAM in a laser-featureless corridor. (b) Map result of the laser PSM SLAM (without advanced sonar) while the robot was still inside the corridor and unable to recover from the drift. (c) Final map result of the laser PSM SLAM (without advanced sonar) when the robot was finally back at the start of the corridor and able to recover from drift after a successful scan matching thanks to the existence of the wall perpendicular to the corridor which exists at the start of the corridor. The grid size is 1m.

IV. Exploration Algorithm

A. *Voronoi Graph*

Voronoi Graph can be described as a collection of line segments which are equidistant from nearby points (called sites) in a d-dimensional Euclidean space. If these sites are treated as obstacles detected by ranging sensors, then the safest paths to travel around the space can be provided by the Voronoi line segments. There are a number of algorithms for computing Voronoi Graph among which S. Fortune's plane sweep algorithm [16] provides a simple O (n log n) solution to the problem and is thus used in this paper (the code for S. Fortune's Voronoi Graph generation method which is modified by S. O'Sullivan is available in [19]). Here, the obstacles in the environment are represented by a set of points which are supplied directly from a set of stored reference scans acquired from the aforementioned SLAM algorithm. These set of reference scans are used as the input for the Voronoi Graph generation and all the resulting Voronoi line segments with distance to the nearest site less than the radius of the robot is eliminated.

Next, for the Voronoi Graph to be useful in exploration, the connectivity between Voronoi vertices needs to be constructed. In this work, the Voronoi Graph is converted into an undirected-weighted graph structure [18] which is represented by an adjacency list data structure [18] where the end points of the line segments become the nodes and then the links between the nodes are established by the lines and weighted by the length of the lines. An example of a Voronoi Graph and its corresponding adjacency list is shown in Figure 4 and Figure 5 respectively. Once the adjacency list is built, then existing graph algorithms such as Breadth First Search (BFS) [18] and Depth First Search (DFS) [18] can be used to find path between any vertices in the Voronoi Graph.

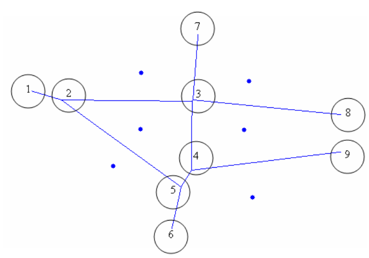


Figure 4 – Example of Voronoi Graph and its graph structure.

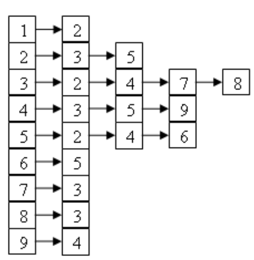


Figure 5 – Adjacency-list representation of the Voronoi Graph example.

B. *Exploration Algorithm*

In the proposed exploration strategy, a Voronoi Graph is used as possible safe paths to destinations. The characteristic of Voronoi Graph allows the robot to plan its collision-free path to anywhere in the environment. The main advantage of utilising Voronoi Graph over grid-based path planning is that the path is safer and more natural. The grids traversal of grid-based path planning is limited to eight directions and thus produces unnecessary turns in a lot of its planned path. Another important characteristic of Voronoi Graph is that it exhibits loop-paths that can help in guiding the robot to close loops during exploration. The loop-paths extraction algorithm is discussed in section IV.C.

It is well known that the loop closing is an important attribute of most SLAM algorithms as it increases the stability and consistency of both map features and robot's pose estimation. As the robot travels further from its starting position, the uncertainty in the state estimation increases due to the addition of new map features with accumulative positional error into the SLAM state vector as a result of imperfect robot's pose estimation and correction. Moreover, these new features are never associated with the old features and as a result the joint correlation between elements in the SLAM state vector is weakened over time. If the robot re-enters previously visited areas, this uncertainty is generally reduced and the joint correlation is strengthened. Note that, re-entering places which are too close to the current position has a small effect of improvement, while re-visiting places further from its current position has a larger effect on improvement. However, if the loop-path is too large, the map can be already too distorted for the loop-path to exist in the Voronoi Graph prior to closing the loop and furthermore there is a need to integrate additional loop-closing algorithms such as in [17] to successfully close the loop. The exploration strategy is designed based on the above deduction where loop-paths which are considered too small are ignored and the robot attempts to execute smaller loop-paths first before attempting to execute larger loop-paths. This will ensure a stable partial map creation before the robot travels further to explore the environment.

The priority of the proposed strategy is performing loop-paths while exploring environments, but the existence of loop-paths cannot be guaranteed in Voronoi Graph and/or in some environments such as corridors. Moreover, there is no assurance that the robot will fully map the environment by just executing the loop-paths. In the event where no loop-paths exist, Voronoi-Graph based exploration is used to visit all the unknown nodes. An occupancy grid-map is maintained throughout the whole exploration process which is then used to determine whether a node has been explored. To determine whether or not a node is explored, the node position in the occupancy grid-map is calculated and then the probability value of its corresponding cell is checked. If the probability value is not 0.5, then it is explored and vice versa. The selection of nodes to explore is done by firstly extracting paths to all unexplored nodes from the robot's current position using Dijkstra method [20] and then followed by sorting the paths by their error costs. The first on the list will be the path with minimum error cost and therefore executed. In order to avoid oscillation, if the destination point changes but the difference in error cost is not large, the robot keeps going to the current destination point. The error cost is calculated as the propagated odometry error [1] of following the path. The complete strategy on the proposed algorithm is shown as flow-chart in Figure 6.

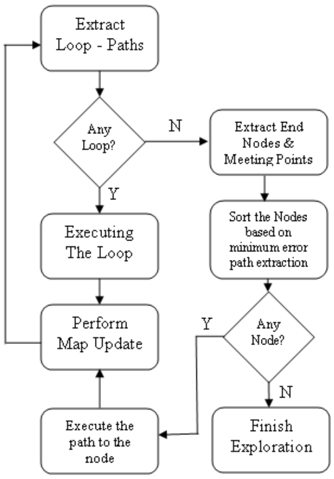


Figure 6 – Flow chart for exploration

C. Loop-Paths Extraction

Loop-paths or cycle-paths refer to paths that start and end at the same node in the graph. The loop-paths can be extracted by using a modified DFS (Depth First Search) which is implemented using a recursive function. By using the DFS, all the nodes are explored to see if there exist paths which end at their corresponding starting node. The implementation uses the following objects:

1. *adj*: a map data structure; to store the adjacency list of the graph.
2. *loop*: a map data structure; to store a list of loop-paths
3. *q*: a queue data structure; to store current loop-paths extraction.
4. *curr*: an integer data type; to store the current node number.
5. *start*: an integer data type; to store the starting node number.

In the following implementation shown in pseudo code form in Algorithm 1, the size of the queue, *q* is initially zero and as the DFS is visiting nodes, the nodes are pushed back into *q* if it does not already exist in *q*. When the DFS reach back to the starting node, the whole *q* is pushed back to the loop-path list, *loop*. Every time the DFS reach the end of path or finish the loop, then the queue is popped back to try another path. The algorithm recognises the end of path without a loop when the entire next nodes at the current node in the adjacency list, *adj* already exist in *q*. the full pseudo-code of the algorithm is shown in Algorithm 1.

The starting node is required to be one of the inputs of the loop-paths extraction algorithm, because it determines where the loop-paths should start and end. At the start of the exploration, the starting node is found by selecting the closest node to the starting position of the robot. The selected node has to have at least two links to other nodes, where one link is used for starting path and the other links are used for the ending paths. However, only those nodes with more than two links are considered in the search in order to reduce the search complexity. This starting node is used until all the resulting loop-paths have been executed. If the loop-paths are not found in the starting node, then the starting node is changed to the next closest node and the search stopped when all the nodes have been attempted. The loop-paths extraction is repeated every time the robot is 1 m away from its previous location, since as the robot explores the environment, the map expands and the Voronoi Graph also changes.

Algorithm 1 – The Loop-Paths Extraction Algorithm

```
int loop_extract (map adj, map  loop, queue q, int curr, int start)
{
   if (q.size == 0)
      q.pushback (curr);
   count = 0;
   for I = 0:adj[curr].size
      next_curr = adj[curr][i];
      if ( next_curr is not in q)
         q.pushback (next_curr);
         flag = 1;
      else
         count++;
         flag = 0;
      endif
      if ( next_curr == start )
         if (q.size > MIN_SIZE)
            loop.pushback (q);
         endif
      else if (flag == 1)
         if ( loop_extract(adj, loop, q, next_curr, start) )
            q.popback();
            count++;
         endif
      endif
      if (count == adj[curr].size)
         return 1;
      else
         return 0;
      endif
   end for
}
```

V. EXPERIMENTAL RESULTS

The exploration algorithm described in this paper has been implemented in real-time C++ using the abovementioned robot. The experiments were carried out in a robotics laboratory with size of 10m x 10m. The integration of the SLAM algorithm and the exploration strategy enabled the robot to explore and build the map of the environment without any human intervention. However, as the proposed method assumes a static environment, the experiment was done without any human and/or other robots moving in the environment.

Figure 7 shows the final Map and the resulting Voronoi Graph of the robotics laboratory while Figure 8 shows the footage of the exploration. The white robots in Figure 8 represent the collection of the pose SLAM estimation of the reference scans. They were numbered according to their creation time. The robot started where the white robot was numbered 0 and then moved to the selected unexplored node.

As it moved to the selected node, then a loop was detected, then the robot changed to follow the loop-path until it successfully closed the loop. Once the small loop had been closed, the robot then decided to follow the bigger loop as detected by loop-extraction algorithm. As depicted by Figure 7 and 8, only two loops existed in the environment, and so as soon as the robot closed the second loop, the robot then used graph-based exploration to complete the map. It is also worth mentioning that the robot can close the loop without any special loop-closing algorithm e.g. [5] because the size of the loop is relatively small. A similar but larger experimental result with three loops and a corridor is shown in figure 9.

Another experiment was carried out by manually driving the robot around the laboratory to show the potential benefit of the proposed algorithm over the other exploration algorithms. In this experiment, the robot was driven as far as possible (while performing SLAM) before attempting to close the loop. Some other algorithms such as Frontier-based allow the robot to travel as far as the map permit without closing the loop and therefore risky in terms of maintaining a consistent map in SLAM. As shown in Figure 10, the map error of the manual drive is significantly larger than the map error of the proposed method prior to closing the loop which is about 0.2m larger in both x and y direction. This result demonstrates that by closing smaller loop first before attempting to close a larger loop which is encouraged by the proposed algorithm, the robot can build a more stable map and perform less risky loop-closings.

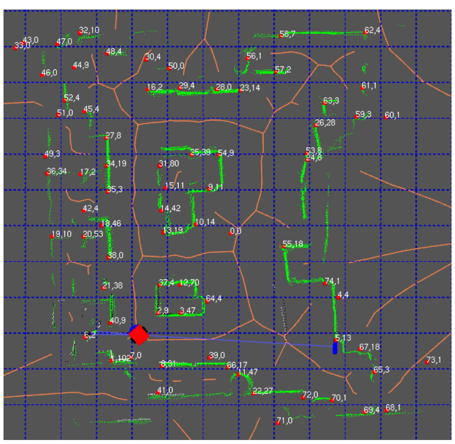
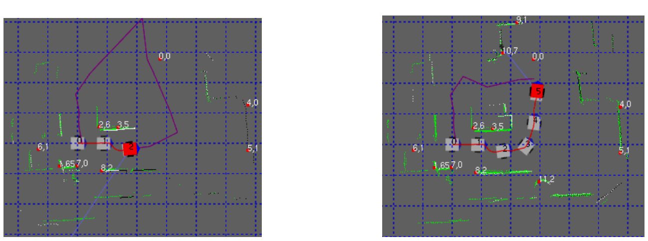
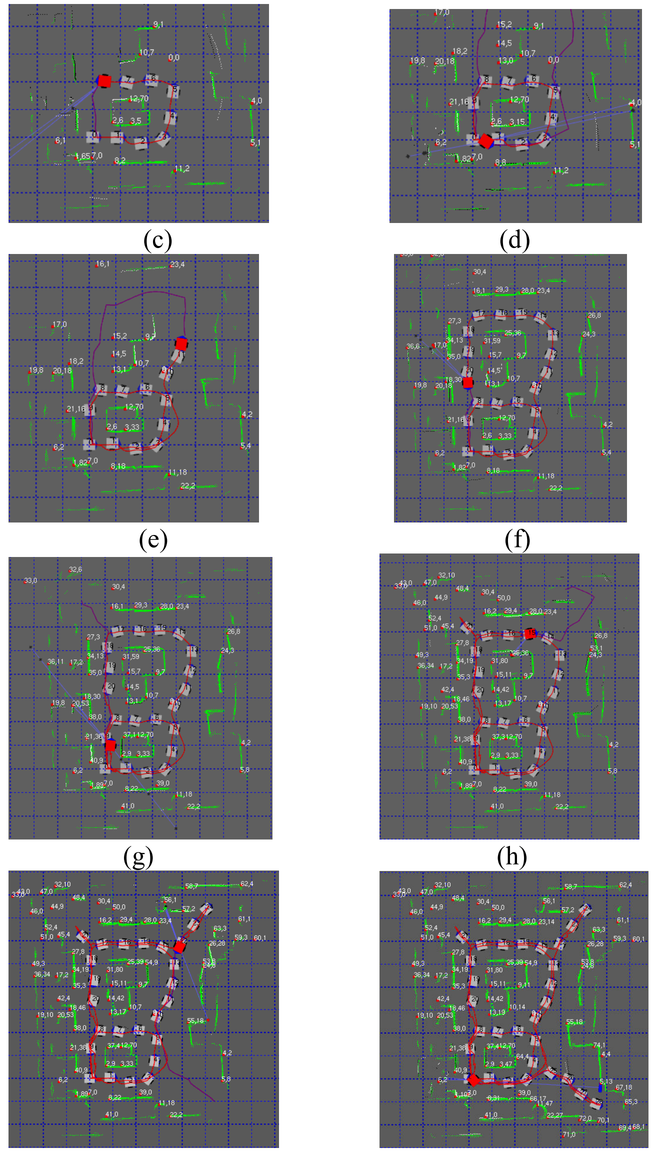


Figure 7 – The final Map and Voronoi graph of autonomous SLAM-exploration experiment with 2 loops. The orange lines are the resulting Voronoi Graph of the Robotics laboratory. The green dots are the laser map features while the red dots are the advanced sonar map features. The sonar map features is labelled with its captured index and its re-observation count. The gird size is 1m.



(a) (b)



(c) (d)

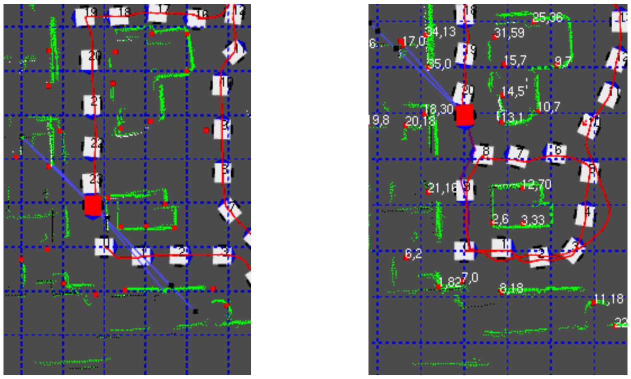
(e) (f)

(g) (h)

(i) (j)

Figure 8 – The footage of the exploration experiment in the robotics lab. The red coloured robot shows the current position of the robot, while the white coloured robot shows the pose of the stored reference scans. Violet lines show the extracted exploration path and the red lines shows the exploration path. The grid size is 1m.

Figure 9 - The final Map of autonomous SLAM-exploration experiment with 3 loops and a corridor. The gird size is 1m.



(a) (b)

Figure 10 – The SLAM Map result before loop-closure. The right picture shows the map result of integrating Voronoi loop exploration prior to the attempt to close the second loop (red to 8), while the left picture shows the map result of manual drive prior to the attempt to close the first loop (red to 0). As shown above, the map error of trying to close a big loop without closing a small loop first is significantly larger. The grid size is 1m.

VI. Conclusion and Future Work

The proposed integrated SLAM-exploration algorithm has been shown to successfully enable the robot to explore and build a stable and consistent map of environment without human intervention. This approach is especially suitable in an environment where loop-paths can be extracted and hence a more accurate map of the environment is obtainable by performing loop-closing frequently. The method has also illustrated the benefits of deploying different sensor modalities, such as sonar and laser where each has its limitations. The laser sensing *cannot* localise well in long corridors where drift is a problem due to the lack of accurate lengthwise information that the sonar *can* detect – namely doorjambs and protrusions into the corridor.

The proposed method has not been designed to scale well with the size of environment and hence the results presented are for relatively small scale environments. Operation in large scale environments requires a map decomposition or local map approach to avoid storage and computational complexities for both the EKF-SLAM, and the Voronoi Graph generation and the loop extraction. Local map strategies for SLAM exist and similar approaches can be applied to the loop extraction process so that real time performance is maintained in larger environments. This paper has shown that the early loop closing strategy is beneficial and in future can be applied to larger scales.

Acknowledgment

The financial support of the ARC Centre for Perceptive and Intelligent Machines in Complex Environments is gratefully acknowledged.

References

[1] L. Kleeman, "Advanced sonar and odometry error modelling for simultaneous localization and map building", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, pp 699-704, 2003.

[2] F. Tungadi and L. Kleeman, "Multiple laser polar scan matching with application to SLAM", Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia, 2007.

[3] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates", International Journal of Robotics Research, Vol. 26, pp 1125-1153, October 2007.

[4] J. Nieto, T. Bailey, and E. Nebot, "Recursive Scan-Matching SLAM". Journal Robotics and Autonomous Systems, 55(1): p. 39-49, 2007.

[5] M. Bosse, P. Newman, J. Leonard, S. Teller. "Simultaneous Localization and Map Building in Large-Scale Cyclic Environments using ATLAS Framework". International Journal of Robotics Research, 23(12):1113-1139, 2004.

[6] A. Davison, "Mobile Robot Navigation Using Active Vision". PhD thesis, University of Oxford, 1998.

[7] B. Yamauchi, A. Schultz, and W. Adams, "Mobile Robot Exploration and Map-Building with Continuous Localization", Proceedings of the IEEE International Conference on Robotics & Automation, Leuven, Belgium 1998.

[8] R. A. Jarvis, " Collision-Free Trajectory Planning Using Distance Transform", in Mechanical Engineering Transactions, Journal of the Institution of Engineers, Vol.ME10, No.3, 187-191, 1985.

[9] K. Nagatani and H. Choset, "Toward Robust Sensor Based Exploration by Constructing Reduced Generalized Voronoi Graph", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1687-1692, 1999.

[10] P. Newman, M. Bosse, and J. Leonard, "Autonomous Feature-Based Exploration", Proceedings of the IEEE International Conference on Robotics & Automation, Taipei, Taiwan, pp 1234-1240, 2003.

[11] T. Kollar and N. Roy, "Using Reinforcement Learning to Improve Exploration Trajectories for Error Minimization", Proceedings of the IEEE International Conference on Robotics & Automation, Orlando, 2006.

[12] J. Bueckert and S. X. Yang, "Neural Dynamics Based Exploration Algorithm for a Mobile Robot", Proceedings of the International Conference on Artificial Neural Network, Porto, Portugal, pp. 640-649, 2007.

[13] R. Sim and N. Roy, "Active Exploration Planning for SLAM using Extended Information Filters", 2004. Available: http://www.mit.edu/~nickroy/papers/uai04.pdf

[14] C. Stachniss, D. Hahnel, W. Burgard, "Exploration with Active Loop-Closing for FastSLAM", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, pp. 1505 – 1510, 2004.

[15] L. Kleeman, "On-The-Fly Classifying Sonar with Accurate Range and Bearing Estimation", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2002, pp. 178-183.

[16] S. Fortune, "Voronoi Diagrams and Delaunay Triangulations", Handbook of discrete and computational geometry, CRC Press, Inc., Boca Raton, FL, 1997.

[17] P. M. Newman and K. L. Ho, "SLAM-Loop Closing with Visually Salient Features", Proceedings of the IEEE International Conference on Robotics & Automation, pp. 635-642, 2005.

[18] A. Gibbons, "Algorithmic Graph Theory", Cambridge University Press, Cambridge, 1985.

[19] S. Fortune and S. O'Sullivan, "Voronoi Diagram Generator C++ code", 1994. Available: http://mapmanager.cvs.sourceforge.net/mapmanager/src/voronoi

[20] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", in Numerische Mathematik, pp. 269-271, 1959.

[21] O. Bengtsson and A.J Baerveldt, "Localization in changing environments – estimation of a covariance matrix for the IDC algorithm" Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1931 – 1937 vol. 4, 2004.