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

Mapping with Synthetic 2D LIDAR in 3D Urban Environment

Z. J. Chong[1], B. Qin[1], T. Bandyopadhyay[2], M. H. Ang Jr.[1], E. Frazzoli[3], D. Rus[3]

***Abstract*— In this paper, we report a fully automated detailed mapping of a challenging urban environment using single LIDAR. To improve scan matching, extended correlative scan matcher is proposed. Also, a Monte Carlo loop closure detection is implemented to perform place recognition efficiently. Automatic recovery of the pose graph map in the presence of false place recognition is realized through a heuristic based loop closure rejection. This mapping framework is evaluated through experiments on the real world dataset obtained from NUS campus environment.**

I. Introduction

Long term navigation in an urban environment requires prior knowledge of the environment. Our previous work [3] showed that a map with abstract representation of the environment is required for a successful autonomous navigation. In particular, through the use of a prior map, precise navigation in an urban environment can be achieved by the effective use of a fixed LIDAR augmented with synthetic LIDAR measurements. We believe that the prior knowledge can augment sensor packages to create an autonomous system with a minimalist infrastructure.

Murphy et al. [17] introduced Rao-Blackwellized Particle Filters to perform mapping. In their seminal work, the robots trajectory and the associated map is represented by each particle. It was later extended by Montemerlo et al. [16] to perform landmark based mapping.

The other approach is graph-based mapping, proposed by Lu and Milios [15], where it proved to be highly effective in solving large scale mapping problems by exploiting the inherent sparsity of a map. In [8], a simple loop closure detection is performed by pair wise matching for any observations that are within a vicinity from the current location. On the other hand, Granstorm et al. [7] used machine learning to perform loop closure detection. In the work, AdaBoost is used as a strong classifier to find a positive match for a loop closure. Newman et al. in [18] described an automated loop closure detection that performs probabilistic measurement using a sequence of images. When an image was found to have a high confidence of loop closure, the corresponding laser scans are retrieved to obtain the relative transformations.

While many graph type SLAM back-end programs exist, for example TreeMap[6], TORO[9], iSAM [11], iSAM2 [10] and g^{2}o [12], recent researches have shown promising results that showcase the capability of SLAM back-ends to reject false loop closure, some are the extension to the current SLAM back-ends. The leading examples include Switchable Constraints [24], Max-Mixture Model [19], and Realizing, Reversing, Recovering (RRR) [13]. However, recent analysis [25] comparing those methods have shown that none of these algorithms work perfectly.

The contribution of this paper is threefold. First, to improve the correctness of a scan matching, an extended correlative scan matcher is proposed. Second, we show that a Monte Carlo based loop closure detection is an efficient technique to perform place recognition. Lastly, we implemented a heuristic based loop closure rejection to a graph based optimization method that allows robust mapping of the environment.

This paper is organized as follows, in Section II, we give an overview to the synthetic 2D LIDAR. Section III provides details on the implementation of our approach to mapping. Finally, experimental results for the implemented framework is included in Section IV.

II. Synthetic 2D LIDAR Overview

The synthetic LIDAR [3] is constructed in real time using interest points extracted from a 3D rolling window assuming that many surfaces in the urban environment are rectilinear in the vertical direction. To form a synthetic LIDAR, interest points that are extracted from the rectilinear surface are projected on a horizontal 2D plane.

The synthetic LIDAR uses 3D rolling window that allows accurate reconstruction of the real environment. More specifically,

$$P_n = \bigcup_{k=n-w} \{p_k, \ldots, p_n\} \quad n > w \tag{1}$$

where P_n is the points accumulated, p_n the points collected on the n-th scan and w is the window size. This is illustrated in Fig. 1, where β is used to ensure a minimum distance is maintained between 2 scan line in a rolling window.

To select the interest points efficiently, 2 rolling windows are maintained. The first rolling window contains only the interest points, while the other much smaller window contains the raw, unprocessed points. This is reflected by the following:

$$P^{\phi}_{n+1} = P^{\phi}_{n} \bigcup \Phi(P_{n+1} \setminus P_n) \tag{2}$$

where Φ can be any points classification function, P^{ϕ} consists of the processed points and P contains the raw points.

[1]Z. J. Chong, B. Qin, M. H. Ang Jr. are with the National University of Singapore, Kent Ridge, Singapore {chongzj, baoxing.qin, mpeangh} at nus.edu.sg

[2]T. Bandyopadhyay is with the Singapore-MIT Alliance for Research and Technology, Singapore tirtha at smart.mit.edu

[3]E. Frazzoli and D. Rus are with the Massachusetts Institute of Technology, Cambridge, MA., USA frazzoli at mit.edu, rus at csail.mit.edu

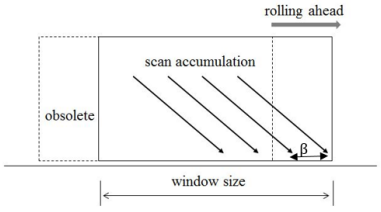


Fig. 1. A 3D rolling window

One such point classification function is the surface normal estimation. The normal of each point is calculated using a first order 3D plane fitting[1], where it is approximated by performing least-square plane fitting to a point's local neighborhood P^K [23].

The plane is represented by a point x, a normal vector $\vec{n}$ and distance d_i from a point $p_i \in P^K$, where d_i is defined as

$$d_i = (p_i - x) \cdot \vec{n} \tag{3}$$

By taking $x = \overline{p} = \frac{1}{k}\sum_{i=1}^{k} p_i$ as the centroid of p_k, the values of $\vec{n}$ can be computed in a least-square sense such that $d_i = 0$. The solution for $\vec{n}$ is given by computing the eigenvalue and eigenvector of the following covariance matrix $C \in \mathbb{R}^{3x3}$ of P^K [22]:

$$C = \frac{1}{k}\sum_{i=1}^{k}(p_i - \overline{p}) \cdot (p_i - \overline{p})^T, C \cdot \vec{v_j} = \lambda_j \cdot \vec{v_j}, j \in \{0, 1, 2\} \tag{4}$$

Where k is the number of points in the local neighborhood, $\overline{p}$ as the centroid of the neighbors, λ_j is the j^{th} eigenvalue with $\vec{v_j}$ as the j^{th} eigenvector. The principal components of P^K corresponds to the eigenvectors $\vec{v_j}$. Hence, the approximation of $\vec{n}$ can be found from the smallest eigenvalue λ_0. Once the normal vector $\vec{n}$ is found, the vertical points can then be obtained by taking the threshold of $\vec{n}$ along the vertical axis, e.g. 0.5. This can vary depending on how noisy the sensor data is.

Once a complete P_n^{ϕ} is obtained, the construction of synthetic LIDAR is completed by performing projection into a virtual horizontal plane (z=0).

III. Robust Mapping with Single LIDAR

The overall mapping process is depicted in Fig. 2, where it can be separated into two general categories, the front-end and back-end. The front-end usually has different modules depending on the sensor types, the back-end usually consist of least-square optimization that relies on the information matrix given by the front-end data.

The mapping process starts with the acquisition of raw sensor data. In our work the sensors are 2D LIDAR, encoder and IMU. The raw sensor data are processed through a real time 2D synthetic LIDAR, as discussed in Section II. The scan matching works together with loop closure detection to generate a constraint graph. Finally, the graph is used by the back-end to generate a globally consistent map while at the same time performs loop constraint rejection. The following will provide detailed description to each of the modules.

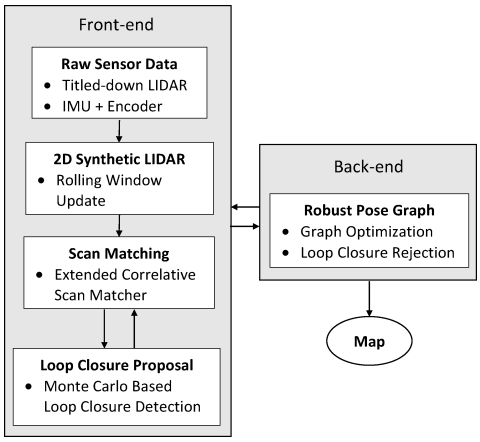


Fig. 2. The mapping framework

A. Extended Correlative Scan Matcher

The correlative scan matcher (CSM) [20] is a family of cross-correlation scan matching algorithms. It employs the probabilistic framework to search for a rigid transformation that maximizes the probability of having the observed data. Correlative scan matching has an advantage that it is robust to large initialization error while taking advantage of fast computation using lookup-table rasterization.

1) Multiple Lookup-table Rasterization: Instead of using only point information in the lookup-table m, other features associated with the point can be included to form multiple lookup-table m_n, where n corresponds to the number of features used. In a single lookup-table rasterization, each point m_i in a map is computed its conditional probability that a nearby point p_i can be observed. In a multiple lookup-table rasterization, other feature of the point p_{n_i} can be encoded. This can either be the point's curvature, normal, depth/height, or color value. To encode an additional lookup-table for a different feature, the nearest point p_is feature value is used. Loosely speaking, the features from the nearest point are simply cast to the additional lookup-table, taking the value at m_{n_i}.

The observation model is then becomes:

$$p(z|x_i, m, m_n) = \prod_j p(z_j|x_i, m, m_n) \tag{5}$$

The additional term m_n is included to take into account the extra features that is encoded in the additional look-up table. To obtain the value of $P(z_j|x_i, m, m_n)$, a weighted sum of each feature can be used:

$$p(z_j|x_i, m, m_n) = \sum_n w_n \; p(z_j|x_i, m) f(z_j, x_i, m_n) \tag{6}$$

where w_n are weighting factors with $\sum w_n = 1.0$, $f(z_j, x_i, m_n)$ is an evaluation function that has a value of

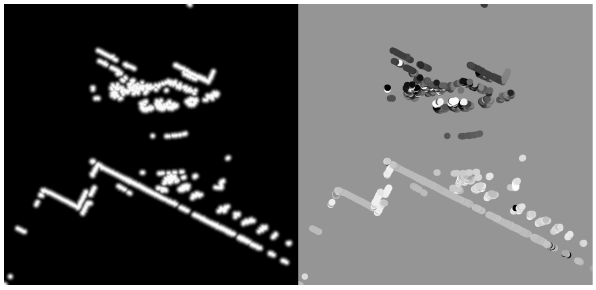


Fig. 3. Multiple look-up table rasterization showing different encoding of point features. Left: Euclidean distance Right: Surface normal

$[0, 1]$. Should none but only the Euclidean point information is used, this implies that $f(x_i, m_n) = 1.0$ and the observation model is reduced to

$$p(z|x_i, m) = \prod_j p(z_j|x_i, m) \tag{7}$$

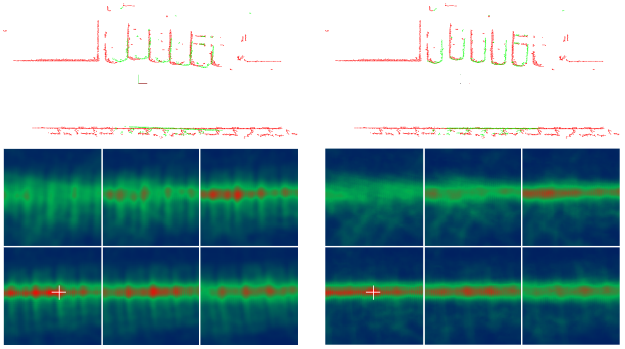
In this work, surface normal of a point is used as the additional feature. To build the look-up table, encoding of the elements in the table is done by obtaining the point's surface normal, derived from the $\vec{n}$ along the horizontal plane. In this case, the information is readily available as the result from the construction of the synthetic LIDAR.

Fig. 3 shows such rasterized look-up table with the additional surface normal as the new prior. The seemingly striking contrast is expected since the rotation of the surface normal on the horizontal plane operates on a range of $[-\pi, \pi)$. To speed out the initialization of the raster table, only cells that contain non-zero value from the Euclidean distance table is encoded with the additional information. Then, the evaluation function can be designed based on the included angle between 2 surface normals, i.e. $\angle(\vec{m}_{xy}, \vec{o}_{xy})/\pi$, where $\vec{m}_{xy}$ is the vector of the surface normal from the matching points, and $\vec{o}_{xy}$ can be inferred directly from the additional look-up table. The normalization factor, π is used to scale the value into range [0,1].

A comparative result showing the advantage of using multiple raster table is provided in Fig. 4, together with the 3D correlation cost function. As evidence from the matching, the use of multiple raster table allows precise matching that are correspond to the true match. The value of the cost function also reveals the matching problem contains many local minima. By using multiple raster table, the true global maxima can be found successfully.

2) Scan Matching Verification: While searching for the maximum likelihood of a scan with a prior could lead to a correct matching, directly inferring the score from the scan matching can sometimes result in a wrong match. This is especially true when covering a large urban environment, as two places can share some similarities geometrically although there are located far away.

The verification is essentially the reversed process of a standard scan matching. Since the transformation of the supposed match is known, the result is used to transform the prior points and test against the matching points. An



(a) CSM with points information only (b) Extended CSM with points and surface normal information

Fig. 4. Comparative examples with extended correlative scan matcher. The cost function is visualized with each tile represents a slice of the cost volume for a fixed orientation. In each example, the maximum numerical value is marked with a crosshair

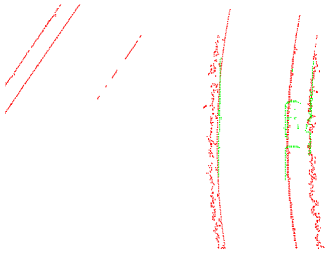


Fig. 5. Wrong matching resolved through scan matching verification. The scan matching score was 64% while verification through reversed process shows the score of only 17%

example of the use of the verification process is shown in Fig. 5, where the wrong match shows a lower score through the verification process.

B. Monte Carlo Based Loop Closure Detection

The well-known Monte Carlo Localization [4], [5] has been applied successfully that allows accurate positioning [21], [3] of a robot within a known environment. The use of the particle filters are the key that enable localization to be done efficiently. Similar to the work of [14], where particle filters are used to perform place recognition between 2 visual pair images, particle filters are used to perform loop closure detection that designed to take the advantage of the view point invariant of a LIDAR scan.

In a particle filter based loop closure detection, each particle corresponds to a node in the map and the particle set:

$$\begin{aligned} X_t &= x_t^{[1]}, x_t^{[2]}, \cdots, x_t^{[M]} \\ x_t^{[j]} &\in 1, 2, \cdots, N \end{aligned} \tag{8}$$

where $x_t^{[j]}$ is j^{th} particle at time t, M is the total number of particles and N gives the current total number of nodes. The particles are the representation of the probability distribution of the current active node. As more nodes are being added to the map, loop closure is detected when there is high density

of particles on a different node. The details of the loop detection process is described below:

1) Motion Model: A motion model is applied to propagate the particles when a new node is being added, which is given by:

$$x_t^{[j]} = p(x_t|u_t, x_{t-1}^{[j]}) \tag{9}$$

where u_t is the robot's control input. The motion model is part of the important function in the iterative nature of the particle sampling framework. Due to the 1D structure of the graph topology, a simple forward motion is used with large probability of particles moving from a node to another at each step. For the LIDAR measurements, determining forward motion of a node can be ambiguous since the measurements are rotational invariant. A forward motion may not necessary imply an increment to the node number. To overcome this ambiguity, heading information from the scan matching result is used when propagation of the particles is performed. The motion models is given as follows:

$$\begin{aligned} p(x_t = n|x_{t-1}, u_t) &= p_0 \\ p(x_t = (n+1)|x_{t-1}, u_t) &= p_1 \\ p(x_t = (n+2)|x_{t-1}, u_t) &= p_2 \\ p(x_t = (n+3)|x_{t-1}, u_t) &= p_3 \end{aligned} \tag{10}$$

Here p_1 is normally assigned with a high probability while a small value is given to the rest of the p_n.

Input: X_{t-1}, z_t, N
Output: X_t
1 $X_t = \bar{X}_t = \emptyset$;
2 **for** $i = 1 \rightarrow M$ **do**
3 $\quad x_t^{[i]} = p(x_t|u_t, x_{t-1}^{[i]})$;
4 $\quad w_t^{[i]} = p(z_t|x_t^{[i]})$;
5 $\quad \bar{X}_t = \bar{X}_t + \langle x_t^{[i]}, w_t^{[i]} \rangle$;
6 **end**
7 **for** $i = 1 \rightarrow M$ **do**
8 $\quad$ draw $j \in \{1, \cdots, N\}$ with probability $\propto w_t^{[j]}$;
9 $\quad$ add $x_t^{[j]}$ to X_t;
10 **end**
11 **for** $i = 1 \rightarrow \alpha M$ **do**
12 $\quad j = R[1, N]$;
13 $\quad k = U[1, M]$;
14 $\quad x_t^{[k]} = j$;
15 **end**
16 return X_t

Algorithm 1: Monte Carlo Close Loop Detection

2) Monte Carlo Loop Closure Detection Algorithm: The overall algorithm is summarized in Alg. 1. The input to the algorithm are the previous particles set X_{t-1}, the measurements z_t and the current total number of nodes N.

At line 11-15, random samples is added to maintain diversity of the particles. To add a random sample, different random number generators are used. To generate j, a particle index is drawn according to the weighted Euclidean distance from the current node based on the latest optimized position available from the back-end. This is to increase the chances of discovering close loop even when a low number of particles is used. On the other hand k is generated with a discrete uniform distribution.

At each the new particle set, the proposal of possible loop closure can be obtained. In this work, it is estimated by a node that has the maximum number of particles.

C. Automatic Loop Closure Rejection

Even though a robust scan matching can be achieved, there are areas where observations are very similar in an urban environment. The pose graph is known to have issues with a falsely recognized loop closure. When the false loop is included in the optimization, it is often ended up having an inconsistent map.

To address the problem, heuristic based loop closure rejection is implemented. Although there are efforts in having a robust pose graph optimizer, our experiment shows that a simple threshold based loop closure rejection have a more desirable graph convergence properties that leads to a high quality map. By keeping track on the development of the residual error value available from the optimizer, a decision on a good or bad close loop can be made based on the growth rate of the error values. This can be achieved by iterative optimization, where a constraint can be retained/removed based on a threshold value. When the error rate exceeds certain threshold, the loop constraint is removed from the graph and optimization is performed, thereby regaining an accurate map.

IV. Experiments

A. Experiment Setup

All experiments are based on the raw sensor data collected around the NUS campus area. This is an urban environment containing many different types of building structures and natural foliage. There are hilly roads with significant difference in elevation and real traffic with many pedestrians traveling through the environment. Our robot is an autonomous Yamaha G22E golf cart equipped with various types of sensors [2]. The vehicle has a simple sensor package with full autonomous capability. The data collected for the mapping purpose consist of sensor data from a single tilted down LIDAR and odometry derived from encoders and a low cost IMU.

B. Mapping of NUS campus area

A dataset is created by driving the vehicle around the campus area covering an area of 720 m x 900 m while traveling more than 6 km. For the first experiment, we used the recorded dataset to perform scan matching. In order to quickly identify areas that may contain the correct match, a multi-level resolution correlation is used. The correlation is started with an exhaustive search on the low resolution search space, covering large volumes of 3D searches (x, y

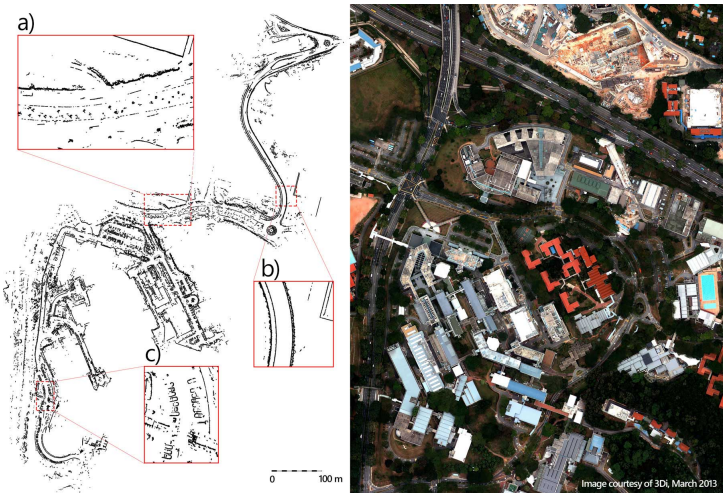


Fig. 6. Mapping of the NUS campus area

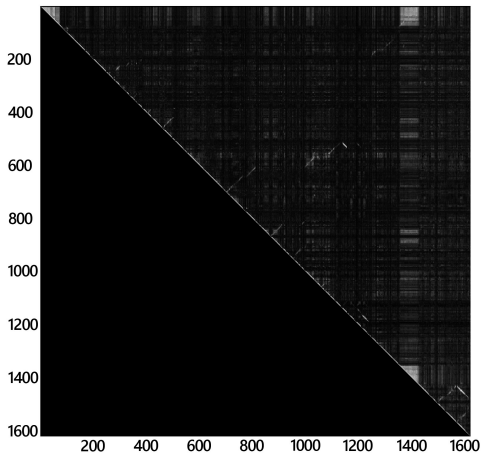


Fig. 7. The similarity matrix of the mapping environment.

TABLE I

PERFORMANCE COMPARISON BETWEEN CSM AND ECSM

Scan matcher	CSM	ECSM
Accuracy (%)	6.7	2.3
Average Scores (%)	73.9	64.6
Scores Std. Dev	12.4	12.5
Average run time (ms)	225	328

TABLE II

MONTE CARLO LOOP DETECTION AND LOOP REJECTION RESULTS

M	Loops Detection			Loops Rejection			Net
	Exp.1	Exp.2	Ave.	Exp.1	Exp.2	Ave.	
30	456*	482	469	142	146	144	325
40	492	507*	500	143	160	152	348
50	499	525	512	154	160	157	355
60	520	520	520	185	155	170	350
80	527	520	524	156	153	155	369
100	518	542	530	140	145	143	387

and θ). Then, the search is continued by evaluating through slices that might have the global maximum.

To compare the performance between the Correlative Scan Matcher (CSM) and the extended CSM, matching pairs that contain only the true matches are used. The scan matching is performed against a subset of the dataset containing 342 pairs of true scan data, then the results are verified manually. In all the experiment, the window of the scan matching is set to cover search space of up to ± 15 m in the longitudinal direction, ± 22 m in the lateral direction and $\pm 32^{\circ}$ rotation. A fairly large translational window is used due to the nature of an outdoor urban area scan matching. The lateral window of 22 m is about the length of a four lanes traffic with a lane seperator (fig. 6a). To maintain a reasonable scan matching speed, the resolution of translation and rotation is set to 0.1 m and 1° respectively. This is to match the occupancy grid size for localization and the LIDAR's angular resolution.

Table I shows the CSM has a false matching of 6.7%. This is improved to 2.3% with the extended CSM while the average run time is increased by 0.1 s. Fig. 7 shows the upper triangular similarity matrix for the complete dataset from the scan matcher. It reflects the nature of the complex environment that is characterized by several bright spots.

For Monte Carlo loop closure detection experiment, different number of particles are used to perform mapping on the same dataset. For the back-end, iSAM [11] is used to perform graph optimization. To accept a loop closure, the proposed loop constraint is added temporary to the graph and perform optimization. It is accepted when the difference of optimized resultant residual error is smaller than a threshold, in which the value is selected through experiment on the same dataset with different number of particles.

Table II shows the experimental result. A consistent map is produced in all the attempts except the experimental set that is marked with *. It shows that only 30 particles are required to produce a consistent map. However, at least 50 particles are required to perform a successful mapping. Generally, the number of loop closure detected increases with the number of particles. The same is applied to the number of net loop closures after taking into account the rejected loop constraints.

Fig. 8 shows a snapshot of the mapping process. The green lines are the visualized factor graph, red circles represent the position of the particles at a node and the radius of the circle scales linearly with the density of particles at the current node. The snapshot shows the scan matcher is able to perform accurate matching of large offset at the

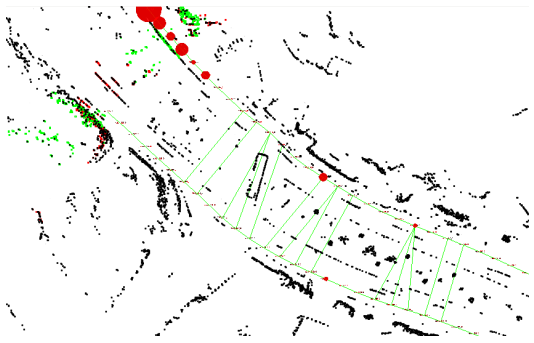


Fig. 8. Visualization showing the mapping process.

opposite direction. Fig. 6b highlights the most challenging area. It is an overhead bridge of about $250\ m$ in length, connecting the NUS main campus with NUS UTown. It poses a stiff challenge to the mapping framework due to the highly uniform structure of a bridge. In this area, the use of Monte Carlo loop detection and loop constraint rejection successfully creates an accurate map of the environment. A video showing the process of the mapping can be found at `http://db.tt/cYrDdtOV`

V. Conclusions and Future Work

In conclusion, a robust mapping platform that allows detailed mapping of the environment is established. To enable accurate scan matching, multiple look-up table is introduced as an extension to the correlative scan matcher . For efficient discovery of loop closure, Monte Carlo based loop closure detection is used. We also showed a heuristic approach to perform reliable loop constraint rejection. For the future work, we would like to investigate the use of robustified pose graph SLAM, and establish a flexible framework that allows multi-vehicle mapping operation. We would also like to perform more in-depth analysis to the underlying topological structure of the map to allow efficient map maintenance in a large environment.

Acknowledgment

This research was supported by the National Research Foundation (NRF) Singapore through the Singapore MIT Alliance for Research and Technology's (FM IRG) research programme, in addition to the partnership with the Defence Science Organisation (DSO). We are grateful for their support.

References

[1] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 11, pp. 1114 –1116, nov 1994.

[2] Z. J. Chong, B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, B. Rebasmen, P. Dai, S. Kim, M. H. Ang Jr., D. Hsu, D. Rus, and E. Frazzoli, "Autonomy for Mobility on Demand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[3] Z. Chong, B. Qin, T. Bandyopadhyay, M. Ang, E. Frazzoli, and D. Rus, "Synthetic 2d LIDAR for precise vehicle localization in 3d urban environment," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.

[4] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.

[5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proceedings of the National Conference on Artificial Intelligence*. JOHN WILEY & SONS LTD, 1999, pp. 343–349.

[6] U. Frese and L. Schroder, "Closing a million-landmarks loop," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 5032–5039.

[7] K. Granstrom, J. Callmer, F. Ramos, and J. Nieto, "Learning to detect loop closure from range data," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 15–22.

[8] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.

[9] G. Grisetti, C. Stachniss, and W. Burgard, "Non-linear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, p. 2009.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3281–3288.

[11] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1365–1378, 2008.

[12] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[13] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time," *Proceedings of Robotics: Science and Systems (RSS), Sydney, Australia*, 2012.

[14] Y. Liu and H. Zhang, "Visual loop closure detection with a compact image descriptor."

[15] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous robots*, vol. 4, no. 4, pp. 333–349, 1997.

[16] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the National conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002, pp. 593–598.

[17] K. Murphy *et al.*, "Bayesian map learning in dynamic environments," *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, pp. 1015–1021, 1999.

[18] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1180–1187.

[19] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," *Proceedings of Robotics: Science and Systems (RSS), Sydney, Australia*, 2012.

[20] E. B. Olson, "Real-time correlative scan matching," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 4387–4393.

[21] B. Qin, Z. J. Chong, T. Bandyopadhyay, M. H. Ang Jr., E. Frazzoli, and D. Rus, "Curb-Intersection Feature Based Monte Carlo Localization on Urban Roads," in *IEEE International Conference on Robotics and Automation*, 2012.

[22] R. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[23] C. M. Shakarji, "Least-squares fitting algorithms of the nist algorithm testing system," in *Journal of Research of the National Institute of Standards and Technology*, 1998, pp. 633–641.

[24] N. Sunderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1254–1261.

[25] N. Sünderhauf and P. Protzel, "Switchable constraints vs. max-mixture models vs. rrr–a comparison of three approaches to robust pose graph slam," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.