Maximum Likelihood Point Cloud Acquisition from a Mobile Platform

Todor Stoyanov and Achim Lilienthal
AASS Research Center, Örebro University

***Abstract*— This paper describes an approach to estimate maximum likelihood range data scans from a moving sensor platform. Data from a vertically mounted rotating laser scanner and odometry position estimates are fused to produce locally consistent point clouds. An expectation maximization algorithm is used to reduce the accumulated error after a full rotation of the range finder. A configuration consisting of a SICK laser scanner mounted on a rotational actuator is described and used to evaluate the proposed approach.**

I. Introduction

In the past robotic systems have been designed to operate mostly in controlled and semi-structured environments. In order to be able to perform higher-level tasks such as mapping, path planning or navigation, a popular approach is to equip the vehicle with a 2D laser range sensor. However, the domain of application of autonomous robots is constantly expanding to more complex, real-world environments. In order to cope with the new challenges, an increasing number of systems rely on three dimensional range data.

Various methods for obtaining consistent 3D range information have been used in the past. One approach is to use two perpendicular laser range finders, one in the horizontal and one in the vertical plane [1][2]. This technique relies on using the horizontal scanner to perform localization and the vertical to obtain range information. While the approach performs well in planar and indoor environments, it is difficult to extend to uneven terrain and has the major drawback that 3D data cannot be obtained while the vehicle is stationary. Another approach is to use a laser scanner in conjunction with an actuator to either “tilt” the scanner around a horizontal axis [3][4][5] or “spin” it around the vertical axis [6]. These methods have been used to produce dense point cloud data which can be utilized by higher level algorithms to create consistent environment models [7]. One problem with this technique is that the sensor platform has to remain stationary for the duration of the scanning, resulting in a *stop-scan-go* motion pattern. An increasingly popular approach to 3D range data acquisition is to use special sensor devices, like time of flight cameras[8]. While this technology has a lot of potential, it is still in an early development stage and does not yet provide the accuracy and illumination invariance of typical laser devices.

The work presented in this paper attempts to provide consistent point cloud data from a spinning SICK LMS200 laser range finder, while eliminating the constraint of halting the robot during data acquisition. Some work in this direction has already been done by Harrison et al. [9]. Their approach is to use a rapidly tilting laser scanner and odometry position data to produce a dense point cloud. Based on the assumption that nearly vertical objects in the scan are in fact vertical, they propose a Bayesian filtering technique to correct the inherent errors in the estimated position of the vehicle. In contrast to their approach, the algorithm presented here applies an Expectation Maximization technique to estimate a maximum likelihood point cloud.

This paper proceeds with Section II by describing the physical sensor setup and the calibration routines. Next, the core of the algorithms used is presented in Section III along with a definition of the operational domain and a discussion of possible limitations of the approach. Section IV continues by defining the evaluation strategy and presents the results obtained through the proposed approach. Finally, the conclusions and future research directions of this work are summarized in Section V.

II. Sensor Setup and Calibration

A. Physical Sensor Deployment and Experimental Vehicle

The data used for evaluating the proposed approach have been collected using the robot “Alfred” (Figure 1). Alfred is a custom built robot, developed in collaboration with Halmstad University and Danaher Motion Inc. It is based on a Permobil automatic wheelchair vehicle, enhanced with an on-board motion controller and a standard Linux machine running a player[10] server.

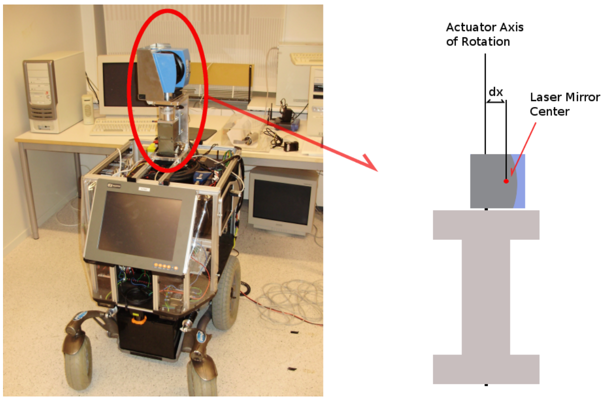


Fig. 1. Left: Alfred robot and sensor setup. Right: Diagram of the laser-actuator configuration

A SICK LMS200 laser scanner was mounted on the end effector of an Amtec PowerCube actuator and the whole setup was placed on top of the robot (Figure 1). This setup allows for continuous rotation of the laser scanner along the principal axis of the actuator, in this setup chosen to coincide

Dimension	Measured	Laser Estimation Error	σ
D_x	$3.400m$	$0.002m$	$0.003m$
D_y	$1.620m$	$0.012m$	$0.002m$

TABLE I

AVERAGE ERROR AND STANDARD DEVIATION OF THE ROOM DIMENSIONS OVER TEN RUNS

with the vertical axis of the robot centered reference system. Thus in principle acquisition of points in a full 360 degree field of view for a long time is possible. However, due to occlusions from the robot chassis, the lower 35 degrees of the laser FOV are ignored. The actuator used can spin at a constant speed between 0.1 and 1.5 rad/s and provides angular accuracy of 0.02 degrees. The laser scanner was set up to operate at 75Hz over an RS422 serial interface. In this mode the LMS provides 180 ranges with an angular resolution of 1° and a maximum range of 8 meters with an error of up to $15mm$.

B. Sensor Calibration

The presented approach utilizes three types of sensor data - robot pose, laser ranges and actuator orientation. The internal parameters of the SICK laser scanner are calibrated with high precision by the manufacturer and are usually not subject to further calibration [11]. The observed error on the PowerCube actuator was within the margins specified in the product data sheet, thus no further internal calibration was deemed necessary for this sensor as well. The odometry position estimate relies on two wheel encoders whose readings are fused through a differential drive motion model. State estimates obtained in this manner are known to be highly dependent on external parameters like wheel slippage and uneven terrain. However, over short distances and small changes in rotation, odometry can provide relatively precise values. Thus an effort was made to obtain suitable values for the encoder offsets, robot axle width and wheel radius. These parameters were tuned by hand and adjusted over several test runs in which the robot was commanded to either move straight or perform a turn in place.

The last calibration routine performed aimed to estimate the precise position of the laser scanner with respect to the axis of rotation of the actuator (illustrated in Figure 1(right)). In particular, the mirror offset d_x along the beam direction was estimated, while the offset perpendicular to the beam direction was assumed to be zero. The method used is similar to the one reported in [11], but adapted for a spinning laser scanner. An empty room was measured by hand, then the robot was placed inside and instructed to spin the actuator. The offset d_x was initially assumed to be zero and a 3D point cloud was generated by applying equation 2 from the next section. An efficient implementation of a RANSAC [12] plane fitting algorithm was then used to extract walls from the 3D laser data. The plane to plane distances D_{xLaser} and D_{yLaser} were compared to the corresponding real world room dimensions D_x and D_y. The mirror offset d_x was then calculated as

$$d_x = \frac{D_{xLaser} - D_x + D_{yLaser} - D_y}{4} \tag{1}$$

In this manner d_x was estimated to be $-2.5cm$. The corrected dimensions of the room along with errors over ten different scans are shown in Table I.

III. FILTERING APPROACH AND ALGORITHM DESCRIPTION

A. General Approach

The algorithms presented in this section attempt to estimate a consistent set of 3D range points from a moving robot. Assume a sensor platform R is moving through an unknown environment. A sequence of odometry state estimates $\bar{z} = z_0..z_t$ is available, where $z_i = \{x_i, y_i, \theta_i\}$ is the estimated 2D position and orientation of the vehicle at time t_i. Also available are the actuator states $\bar{a} = \alpha_0..\alpha_t$ and the laser scans $\bar{l} = l_0..l_t$ where α_i is the orientation of the actuator end effector at time t_i and $l_i = \{r_0..r_n, \beta_0..\beta_n\}$ is the set of ranges and corresponding bearings. Assuming perfect sensor readings, the 3D point measured by range reading k at time t_i is obtained through (2).

$$P_{i,k} = \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} + r_k \begin{bmatrix} \cos\beta_k \sin\alpha_i + \theta_i \\ \cos\beta_k \cos\alpha_i + \theta_i \\ \sin\beta_k \end{bmatrix} \tag{2}$$

The actuator and range measurements are assumed to be random variables corrupted by Gaussian noise with variances σ_α and σ_r, while the state measurements $\bar{z}$ are assumed to be corrupted by a multivariate Gaussian noise with covariance matrix C_{odom}

$$C_{odom} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} \end{bmatrix} \tag{3}$$

The random variables $\bar{z}$, $\bar{a}$ and $\bar{l}$ are assumed to be mutually independent and thus so are their respective noise distributions. In the rest of this paper, the state vector $\bar{z}$ will also be referred to as the robot *path*. In the case under consideration, the robot path is usually short (on the order of a couple of meters) and only refers to state measurements used to obtain a single consistent point cloud.

As noted in the previous section, the most significant source of error is the position estimate $\bar{z}$. This problem can be addressed by familiar filtering techniques from the SLAM domain. The proposed approach uses Multilevel Relaxation (MLR)[13] to estimate the maximum likelihood values for the state vector $\bar{z}$. MLR is a graph based optimization algorithm with vertices (or frames) encoding robot state values and edges (or relations) representing measurement uncertainties. The approach proposed here creates n frames from the corresponding odometry estimates. Consecutive frames are linked by relations derived from the odometry estimates and covariance matrix C_{odom}, resulting in a total of $n-1$ edges in the graph. To obtain a well defined estimation problem, at least one more relation must be added. This is derived by registering the best-estimate 3D points from the end of the robot path to the model points observed at the beginning of the path. The general idea of the approach

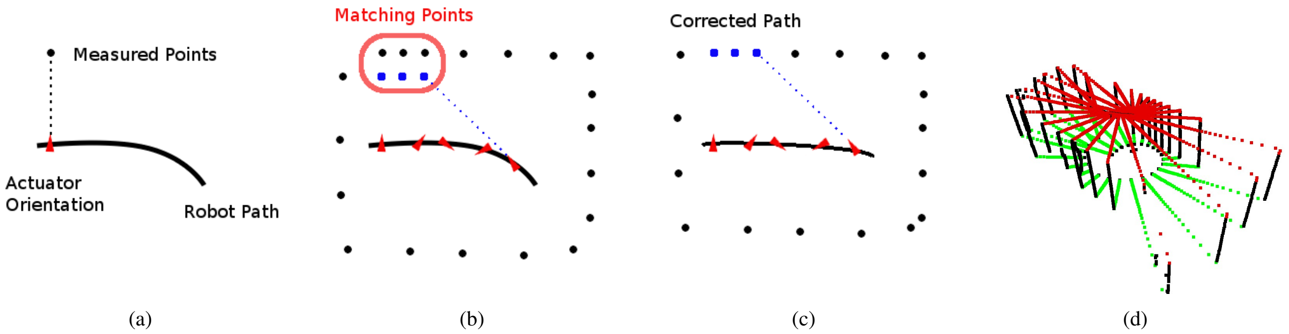


Fig. 2. Figures 2(a) through 2(c) - a projection into the xy plane of the robot as it moves through the environment and measures distances. After points are re-observed in 2(b) a maximum likelihood path is estimated and the error is minimized 2(c). In 2(d) sampled points from a typical scan, segmented into ceiling, floor and object classes.

is illustrated in Figure 2 and formalized in Algorithm 1. The major components of this algorithm are discussed in the subsequent sections.

Algorithm 1

```
1:  Start actuator rotation
2:  while $\alpha + \theta < 2\pi + \epsilon$ do
3:      Calculate estimated position and covariance
4:      Insert frame and relation in MLR graph
5:      Insert points in spatial index
6:  end while
7:  while # of matching points $< \mu$ do
8:      Propagate uncertainty to each point in the scan
9:      Search spatial index for nearest neighbor within uncertainty area
10:     Mark points as matching
11: end while
12: Register matching points, estimate position and uncertainty
13: Perform Multilevel Relaxation
```

B. Estimating the Odometry Covariance

In order to estimate the covariance matrix C_{odom} the technique presented in [14] was adopted. Thus, the uncertainty at time t_j from a reference point in time t_i ($i < j$) can be calculated as:

$$C_{i,j} = J_1 C_{i,j-1} J_1^T + J_2 C_{j-1,j} J_2^T \tag{4}$$

where J_1 is

$$J_1 = \begin{bmatrix} 1 & 0 & -\sin\theta_{j-1}dx - cos\theta_{j-1}dy \\ 1 & 0 & \cos\theta_{j-1}dx - sin\theta_{j-1}dy \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and J_2 is

$$J_2 = \begin{bmatrix} \cos\theta_{j-1} & -sin\theta_{j-1} & 0 \\ \sin\theta_{j-1} & cos\theta_{j-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The covariance $C_{j-1,j}$ is a measure of the uncertainty between the two consecutive frames $j-1$ and j. A good estimate is to define $C_{j-1,j}$ as

$$C_{j-1,j} = \begin{bmatrix} \sigma_{xd}\bar{d}_j + \sigma_{xt}\bar{\theta}_j & 0 & 0 \\ 0 & \sigma_{yd}\bar{d}_j + \sigma_{yt}\bar{\theta}_j & 0 \\ 0 & 0 & \sigma_{td}\bar{d}_j + \sigma_{tt}\bar{\theta}_j \end{bmatrix} \tag{7}$$

with $\bar{d}_j$ as the euclidean distance between z_{j-1} and z_j and $\bar{\theta}_j$ as the respective angular increment. Thus, in line 4 of Algorithm 1 the adjusted incremental covariance $C^*_{j-1,j} = J_2 C_{j-1,j} J_2^T$ is used to specify the MLR relation, while in the next section $C_{0,i}$ is used to obtain point covariance.

C. Covariance Propagation to Range Points

The uncertainty in the robot state z_i, the actuator a_i and laser l_i can be propagated to obtain the covariance matrix of a point P, obtained at time t_i. Using equation (2) and error propagation we obtain:

$$C_p = JCJ^T = J \begin{bmatrix} [C_{0,i}] & 0 & 0 \\ 0 & \sigma_{\alpha\alpha} & 0 \\ 0 & 0 & \sigma_{rr} \end{bmatrix} J^T \tag{8}$$

where C is the 5x5 state covariance matrix and J is the 3x5 Jacobian of (2) evaluated at time t_i

$$J = \begin{bmatrix} \frac{\delta P_x}{\delta x} & \frac{\delta P_x}{\delta y} & .. \\ \frac{\delta P_y}{\delta x} & \frac{\delta P_y}{\delta y} & .. \\ \frac{\delta P_z}{\delta x} & \frac{\delta P_z}{\delta y} & .. \end{bmatrix}_{x=x_i, y=y_i..} =$$

$$\begin{bmatrix} 1 & 0 & r(c_\alpha c_\theta - s_\alpha s_\theta) & c_\beta s_{\alpha+\theta} & r(c_\alpha c_\theta - s_\alpha s_\theta) \\ 0 & 1 & -r(c_\alpha s_\theta - s_\alpha c_\theta) & c_\beta c_{\alpha+\theta} & -r(c_\alpha s_\theta - s_\alpha c_\theta) \\ 0 & 0 & 0 & s_\beta & 0 \end{bmatrix} \tag{9}$$

where r is the range used to estimate the point coordinates, c_α c_β c_θ are the cosines of the actuator orientation, laser bearing and robot orientation, while s_α s_β and s_θ are the respective sines.

The so obtained covariance matrix is then used to test whether a point P might have already been observed (line 9 in Algorithm 1). Two points p_i and p_j are considered candidate matching points if the Mahalanobis distance $\xi = (p_i - p_j)C_{p_i}^{-1}(p_i - p_j)^T$ is bellow a threshold value ($\xi = 0.15m$ used in this paper).

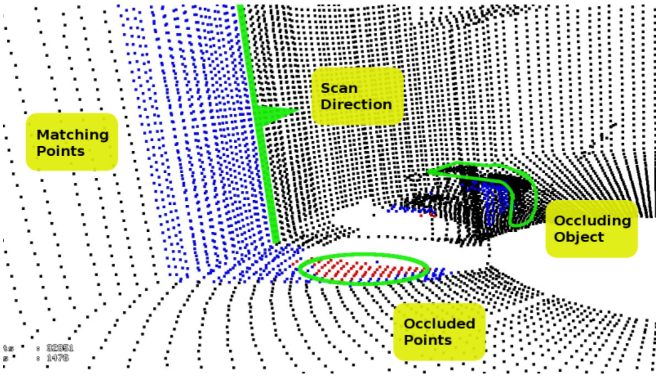


Fig. 3. Point matching for an example scan. The points used for ICP registration are marked in blue. Points collected at the end of the robot path, but occluded at the beginning are ignored (marked in red).

D. Registration and Error Estimation

The candidate matching points from the previous section are then used as input to an Iterative Closest Point (ICP)[15] algorithm. The ICP algorithm is a widely used registration method, which however is known to have some limitations. Most notably, ICP is likely to fail in cases of small overlap between candidate scans and featureless environments. The first issue is particularly problematic when point clouds with large viewpoint changes have to be registered. The method of candidate point set selection presented in the previous subsection addresses precisely this case. By considering only points within a covariance ellipsoid, the expected inconsistent and non-overlapping areas of the candidate point clouds are minimized (as illustrated in Figure 3). Thus, ICP is more likely to converge to an optimal solution, even with a large viewpoint change. For the implementation evaluated in Section IV the parameter μ from Algorithm 1 was set to 1000, while ϵ was set to $0.1rad$, resulting in compact candidate point sets.

The transformation and match covariance obtained are used to update the MLR graph structure. The path poses used for matching are transformed using the best-fit transformation obtained from ICP. The so obtained new positions are then transformed to the reference system of the first path point and these estimates are used to insert new relations to the initial path point. In order to obtain the covariance matrix needed to insert the final relations, a method similar to the one proposed in [16] was used. Assuming a true model Y, an estimated model $\hat{Y}$ and data observation M, we have

$$Y = MX + w, \hat{Y} = M\hat{X} \tag{10}$$

where X and $\hat{X}$ are the model parameter vectors and w is white Gaussian noise. Assuming a linear model function, the error minimized by ICP is

$$E(\hat{X}) = (Y - M\hat{X})^T (Y - M\hat{X}) \tag{11}$$

As shown in [16] the ICP match covariance can then be calculated using:

$$C_{match} = \sigma^2 \left(\frac{1}{2} H \right)^{-1} \tag{12}$$

where H is the hessian of the linearized error function $E(\hat{X})$ and σ^2 is an unbiased estimate of the error variance. The Multilevel Relaxation algorithm is then used to "distribute" the accumulated error over the poses $\bar{z}$ and obtain a maximum likelihood estimate for the robot path. This step is described in detail in [13], but essentially consists of generalizing the graph into a series of increasingly abstract levels, until it is possible to explicitly solve the estimation problem. The solution obtained is then propagated down from the top level of abstraction to obtain the maximum likelihood solution to the original problem. Finally, a new point set is generated using the corrected positions $\bar{z}_c$.

E. Theoretical Limitations

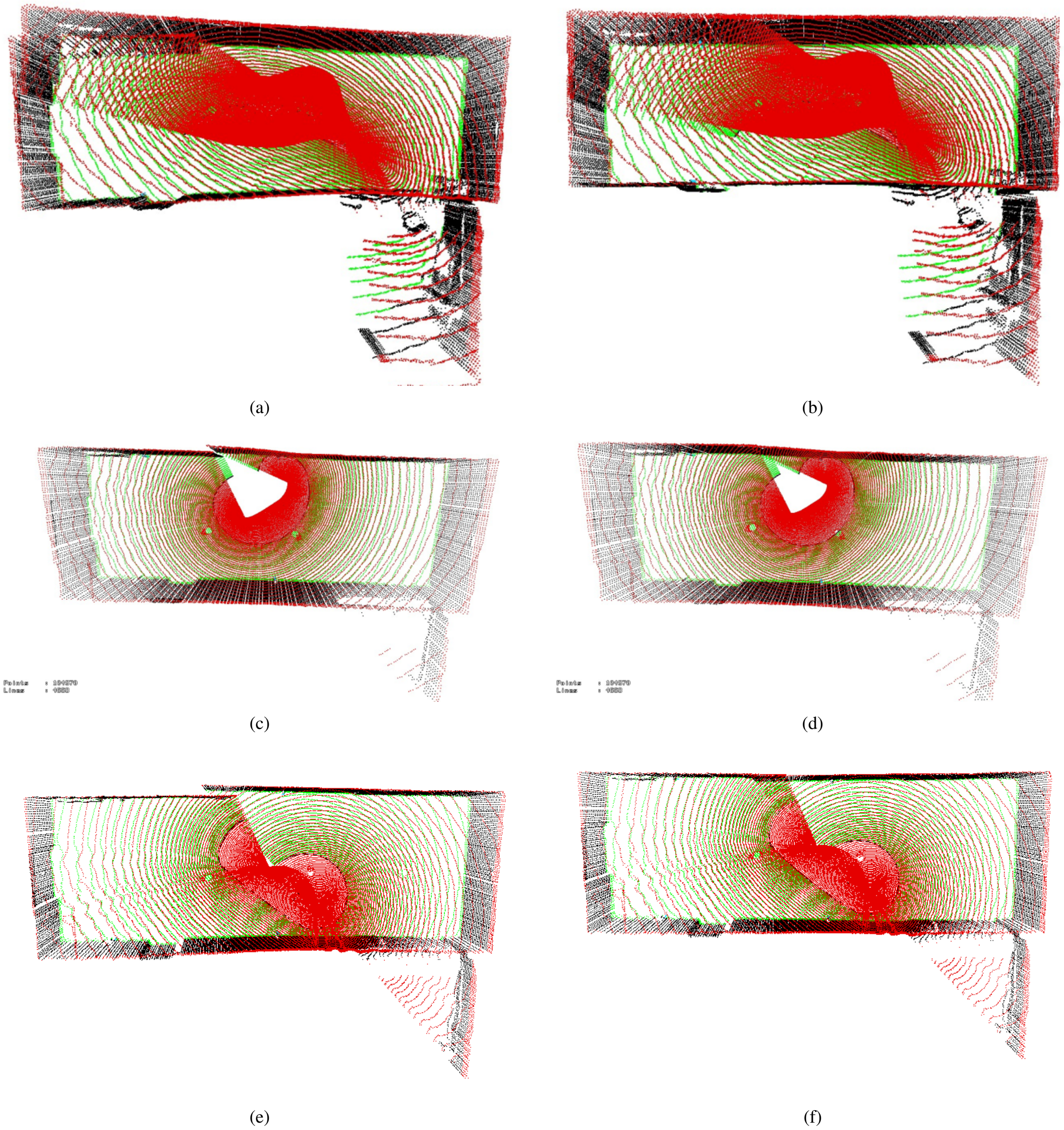
The presented approach has some theoretical limitations that deserve mention. First and foremost, the part of the environment scanned at the beginning of the path has to be at least partially observable from the final viewpoint. This constraint can be relaxed if matching is allowed for partial paths - i.e. if the final relation is between the last frame and an arbitrary previous frame, instead of the possibly occluded starting frame. In such cases, the error will be minimized only for the scan section in between the two matched poses. In practice environment observability can only be a problem if the robot makes sharp turns in cluttered environments.

The second limitation is that the algorithm is currently operating with a 2D pose vector, thus rough terrain can be problematic. This concern will be addressed in the future by obtaining the other three state variables from an Inertial Motion Unit (IMU) and solving a full 6DOF problem. An update to the MLR relaxation algorithm will also be necessary, as currently it is restricted to graph representations of a 2D environment.

Finally, the maximum speed and turn rate of the robot depend on the angular velocity of the actuator. Harrison et al [9] make an argument that the Nyquist sampling theorem can be applied to calculate the maximum speed of the vehicle in their tilting laser setup. The rotating setup used here imposes the same condition on the maximum turn rate of the robot - namely the sampling frequency (actuator turn rate) has to be at least double the signal frequency, thus $\dot{\theta} < \frac{\dot{\alpha}}{2}$. A theoretical constraint on the maximum speed is difficult to compute, as it depends on the observability of the environment as well. A strict upper bound however is that the distance traveled by the robot should be less then the maximum laser range, thus roughly $v < \frac{L_{range}\dot{\alpha}}{2\pi}$.

IV. Performance Evaluation

In order to evaluate the utility of the proposed filtering approach, point clouds were collected in a controlled indoor environment. The robot was put in an empty room and driven to random goal points. An additional laser scanner mounted in the front of the robot was used for collision avoidance.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4. Sample point clouds (100k points) acquired by simple odometry interpolation in 4(a) and 4(c) and after filtering in figure 4(b) and 4(d). The vehicle speed was limited to $0.2m/s$

Three sets of data were collected under maximum vehicle translational velocity of 0.1, 0.2 and $0.3m/s$ respectively. The maximum rotational speed of the robot was set to $0.75rad/s$ and kept constant in all three experiments. The actuator was rotated at a constant speed of $0.5rad/s$ to obtain dense point clouds. As stated in section III-E, the theoretical constraint for the platform's turn rate under this actuator velocity would be $0.25rad/s$, or three times lower then the one used. However, this maximal velocity is only reached in rare occasions, when the vehicle sharply approaches an obstacle and thus did not cause undersampling problems.

The robot was programmed to first acquire a scan of the environment while remaining stationary and then move to the next goal point. Thus, the measurements collected while stationary are used to generate ground truth point clouds, while the later data are used to evaluate the quality of the proposed filtering approach.

The log files collected while the robot was moving were used to generate two sets of point clouds - one with and one without applying the described filtering technique. Initially, the point clouds in the control data set were generated after each $2\pi + \epsilon$ absolute rotation of the laser scanner. An

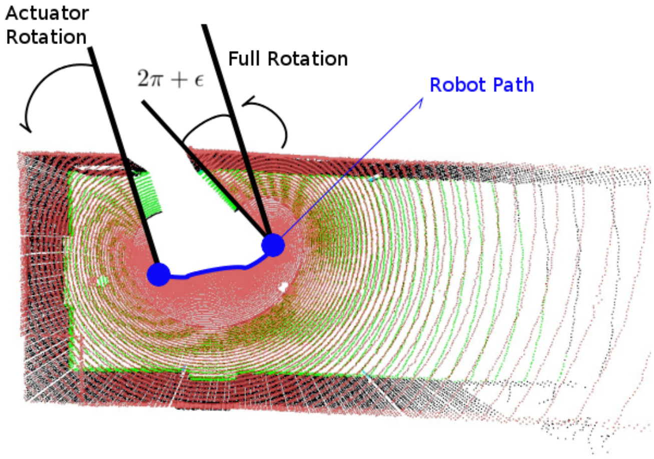


Fig. 5. A point cloud generated after a $2\pi + \epsilon$ combined robot and actuator rotation. Due to the robot motion, a part of the environment remains unobserved.

example measurement produced in this manner is shown in Figure 5. Evidently this method produced incomplete measurements, with parts of the environment missing due to viewpoint change. Therefore, the control measurements were generated by not performing the correction step in the proposed approach. A point cloud obtained in this manner is shown in Figure 4(a). The inconsistencies introduced by the error in the position estimate are more apparent in this case and can easily be identified by looking at the top-left corner of the picture. The corresponding filtered output points are shown in Figure 4(b). As expected, the error in the point positions is detected and distributed along the vehicle path to produce a locally consistent measurement.

It is worth noting that the quality of the filtered point cloud depends on the fidelity of the registration algorithm. Naturally, in order to correct the errors in position, a consistent transformation has to be obtained through ICP. In cases of an environment poor on features (as the one used here), sometimes not all errors can be corrected. For example when correcting the point cloud in Figure 4(c), it is not possible to accurately measure the error in the direction parallel to the wall, as the candidate matching points all belong to the wall plane and do not span the space along the wall. The covariance matrix returned by ICP reflects this issue by having a small variance perpendicular to the wall and a much larger component along it. The expectation maximization algorithm then detects the uncertainty of the transformation and puts more trust in the odometry estimate in this particular direction. The point cloud obtained through the filtering approach is shown in Figure 4(d).

V. Conclusion

This paper presented an Expectation Maximization algorithm to estimate maximum likelihood point clouds from a moving vehicle. Data sets from an indoor environment were collected and used to evaluate the quality of the point clouds generated by the proposed filtering algorithm. The results obtained attest to the feasibility of the presented approach for semi-structured indoor environments. Further work will focus on addressing some of the theoretical limitations of the algorithm, especially the constraint to a planar motion space.

References

[1] C. Früh and A. Zakhor, “Fast 3D Model Generation in Urban Environments,” *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, pp. 165–170, 2001.

[2] A. Howard, D. Wolf, and G. Sukhatme, “Towards 3D Mapping in Large Urban Environments,” *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1, pp. 419–424 vol.1, Sept.-2 Oct. 2004.

[3] A. Nüchter, K. Lingemann, and J. Hertzberg, “Mapping of Rescue Environments with Kurt3D,” in *IEEE International Workshop on Safety, Security and Rescue Robotics, SSRR*, 2005, pp. 158–163.

[4] H. Andreasson and A. J. Lilienthal, “Vision Aided 3D Laser Based Registration,” in *Proceedings of the European Conference on Mobile Robots (ECMR)*, 2007, pp. 192–197.

[5] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, “6D SLAM with an Application in Autonomous Mine Mapping,” in *IEEE International Conference on Robotics and Automation*, vol. 2, 2004, pp. 1998–2003.

[6] P. Lamon, C. Stachniss, R. Triebel, P. Pfaff, C. Plagemann, G. Grisetti, S. Kolski, W. Burgard, and R. Siegwart, “Mapping with an Autonomous Car,” in *IEEE/RSJ IROS Workshop: Safe Navigation in Open and Dynamic Environments*, 2006.

[7] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan Registration for Autonomous Mining Vehicles using 3D-NDT,” *Journal of Field Robotics*, vol. 24, no. 10, 2007.

[8] K. Pathak, A. Birk, and J. Poppinga, “Sub-Pixel Depth Accuracy with a Time of Flight Sensor Using Multimodal Gaussian Analysis,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots Systems, IROS 2008.*, Sept. 2008, pp. 3519–3524.

[9] A. Harrison and P. Newman, “High Quality 3D Laser Ranging Under General Vehicle Motion,” *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 7–12, May 2008.

[10] B. Gerkey, R. Vaughan, and A. Howard, “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems,” in *International Conference on Advanced Robotics (ICAR)*, 2003, pp. 317–323.

[11] J. Weingarten, “Feature-based 3D SLAM,” Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2006. [Online]. Available: http://library.epfl.ch/theses/?nr=3601

[12] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for Point-Cloud Shape Detection,” in *Computer Graphics Forum*, vol. 26, no. 2. Blackwell Publishing, 2007, pp. 214–226.

[13] U. Frese, P. Larsson, and T. Duckett, “A Multilevel Relaxation Algorithm for Simultaneous Localization and Mapping,” *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 196–207, April 2005.

[14] U. Frese, “A Discussion of Simultaneous Localization and Mapping,” *Autonomous Robots*, vol. 20, no. 1, pp. 25–42, 2006.

[15] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[16] O. Bengtsson and A. Baerveldt, “Localization in changing environments - estimation of a covariance matrix for the IDC algorithm,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4, 2001, pp. 1931–1937 vol.4.