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

3D Mapping for Urban Service Robots

Rafael Valencia, Ernesto H. Teniente, Eduard Trulls, and Juan Andrade-Cetto

Abstract— **We present an approach to the problem of 3D map building in urban settings for service robots, using three-dimensional laser range scans as the main data input. Our system is based on the probabilistic alignment of 3D point clouds employing a delayed-state information-form SLAM algorithm, for which we can add observations of relative robot displacements efficiently. These observations come from the alignment of dense range data point clouds computed with a variant of the iterative closest point algorithm. The datasets were acquired with our custom built 3D range scanner integrated into a mobile robot platform. Our mapping results are compared to a GIS-based CAD model of the experimental site. The results show that our approach to 3D mapping performs with sufficient accuracy to derive traversability maps that allow our service robots navigate and accomplish their assigned tasks on a urban pedestrian area.**

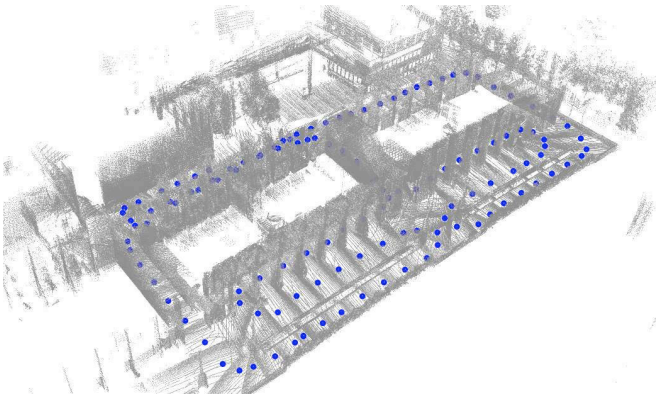


Fig. 1. Final map of the experimental site computed with our approach.

I. Introduction

3D mapping in urban environments has been recognized as a challenging task in the past. Urban settings have many specific characteristics, e. g., non-flat terrain, occasional poor GPS coverage, underpasses, points with aliasing, moderate vegetation, and sunlight exposure severely subject to shadows.

In order to avoid problems related to cameras, such as illumination issues, 3D mapping in outdoor environments has been usually addressed using 3D laser range finders. In this paper we describe a solution to the Simultaneous Localization and Mapping (SLAM) problem in urban environments, using three-dimensional laser range scans as the main data input.

The intended application of this solution is the building of the necessary maps for a heterogeneous fleet of service robots that navigate in urban settings during the execution of their tasks [1]. For this purpose, from 3D point cloud maps we also compute traversability maps, which are 2D grid layers with continuous-valued cells indicating the maximum traversability speed.

Our approach consists of the probabilistic alignment of 3D point clouds employing a delayed-state Extended Information Filter (EIF) SLAM algorithm. From consecutive 3D point clouds we compute relative pose constraints with the Iterative Closest Point (ICP) algorithm [2]. In our approach to point cloud fitting, we use a hierarchical correspondence search strategy, using a point-to-plane metric at the coarsest level and a point-to-point metric at finer levels. When searching for point matches, we use kd-trees to reduce the computational complexity. And, during the minimization step, we perform compensation between translation and rotation by using a weighted distance metric [3].

The pose constraints computed from consecutive 3D point clouds are then used as relative pose measurements in a 6 degrees-of-freedom (DOF) delayed-state information-form SLAM algorithm. State transition and measurement models are computed using motion compositions. In addition, motivated by [4], we employ a reordering of the information matrix and a QR decomposition to efficiently recover the covariance and state estimate.

Finally, once we have these 3D point clouds correctly registered we derive traversability maps by dividing the environment into cells and assigning velocities for each cell using the kinematic model of the mobile robot.

The reminder of this paper is organized as follows. A brief description of the related work is presented in section II. In section III we describe our strategy for computing pose constraints from 3D point clouds. Section IV is devoted to explain our SLAM algorithm. The experimental setup is described in section V and the corresponding results are shown in section VI. In Section VII we show how traversability maps are extracted from 3D scans. Finally, concluding remarks are depicted in Section VIII.

II. Related Work

Mapping with 3D laser range finders has been addressed in many ways. A non-probabilistic approach is proposed in [5], where the alignment of two scans is done mainly by improvements to the basic ICP algorithm proposed in [2].

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC. Llorens Artigas 4-6, Barcelona, 08028 Spain. {rvalencia,ehomar,etrulls,cetto}@iri.upc.edu.

This work has been supported by scholarships from the Mexican Council of Science and Technology to R. Valencia and E. Teniente, and from UPC to E. Trulls; by project CSIC-200850I107 from CSIC; by projects DPI-2007-614452, DPI-2008-06022, and MIPRCV Consolider-Ingenio 2010 from the Spanish Ministry of Science and Innovation; and by the EU URUS project IST-FP6-STREP-045062.

Nevertheless, probabilistic methods allow a straightforward way to distribute errors when closing a loop. One possibility for 3D probabilistic SLAM in outdoor environments is to employ a delayed-state framework with an Extended Kalman Filter (EKF) [6].

However, using an EIF within the delayed-state framework has better scalability properties compared to the EKF [7]. A delayed-state EIF generates exact sparse information matrices and, during open loop traverse, the information matrix becomes tri-block diagonal as consecutive robot poses are added to the state. At loop closure, the matrix structure is only modified sparsely, setting information links between non-consecutive robot poses. Thus, one advantage of the delayed-state information-form for SLAM is that predictions and updates take constant time, assuming an efficient or approximate way for state recovery is used to evaluate Jacobians.

The approximations performed by linearizations, together with covariance and state recovery and data association are issues of concern in the use of EIF filters for SLAM. In [8] we proposed an alternative to reduce the overconfidence effects of linearizations by closing only the most informative loops, decreasing the total number of loop closure links, maintaining the sparsity of the information matrix. The technique not only defers filter inconsistency but also has better scalability properties. As for state recovery in information form, efficient techniques for exact recovery of covariance and state estimates are proposed in [4] and [9]. Our latest work shows that during open loop traverse exact state recovery can be performed in constant time, and that we can perform efficient data association in $\mathcal{O}(log\ n)$ for the delayed-state EIF framework [10].

III. Computation of Relative Pose Constraints

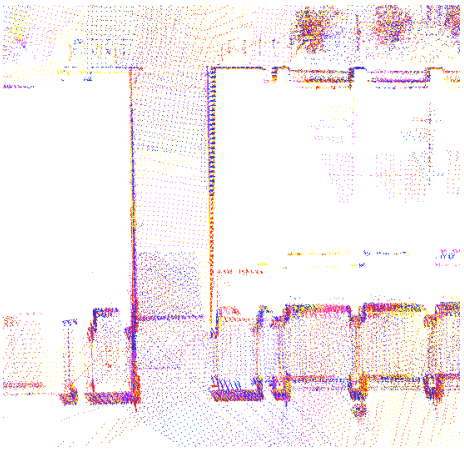
The purpose of the Iterative Closest Point (ICP) algorithm is to compute the relative motion between two partially overlapped 3D point clouds. The algorithm iteratively minimizes the Mean Square Error (MSE) over point matches proceeding as follows: for each point in one data set, the closest point in the second one is found or vice-versa (correspondence step), then the motion that minimizes the MSE between the correspondences is computed (registration step), finally, the point matches are updated (update step).

A. Point to Point Distance

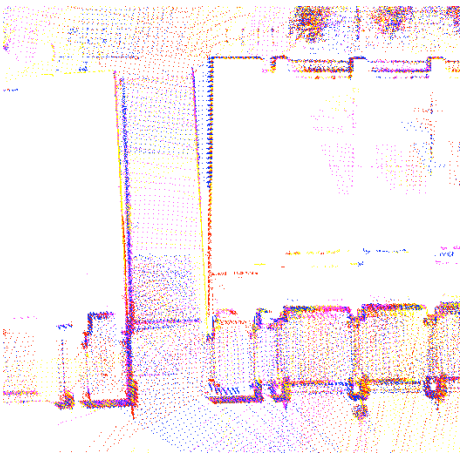
Traditionally, the minimization is performed over the sum of Euclidean distances. However, we resort to a metric that gives different weight to sensor rotation than translation [3]. This distance between points p_1 and p_2 is defined as

$$d(p_1, p_2) = \sqrt{\|\delta\|^2 - \frac{\|p_1 \times \delta\|^2}{k}}, \qquad (1)$$

with $k = \|p_1\|^2 + L^2$, $\delta = p_2 - p_1$, and L is a user specified weighting factor that trades-off between translation and rotation. Note that when $L \to \infty$ the new distance tends to the Euclidean distance. In our experiments $L = 30$, see Fig. 2.



(a) $L = 30$.



(b) $L \to \infty$ (Euclidean distance).

Fig. 2. Comparison of ICP metrics.

B. Sampling Strategy and Outlier Removal

The iterative nature of the ICP minimization step requires reduced sample sets to be used. To this end, point clouds are uniformly sampled, which helps also to reduce sensor noise. Unfortunately, performing ICP over sampled data is very sensitive to data content, e.g. noise level, occlusion areas, complexity of the range data, etc. When the number of outliers is large, many wrong correspondences are unavoidable, and would produce convergence to a local minimum leading to poor final overlap, or in the worst case, to divergence. We shall remember that the original ICP algorithm considers data sets without outliers, which is not our case. For this reason, after sampling, outlier removal is performed by keeping only those points with a mean distance from their k-nearest neighbors lower than an experimentally chosen threshold.

C. Correspondence Search

Several metrics can be used to compute feature correspondences in 3D range data, such as point-to-point, point-to-plane, and point-to-projection with triangular surfaces [11]. In our method we propose a hierarchical correspondence search, using a point-to-plane strategy at the coarsest level

and a point-to-point metric at finer levels.

The closest plane to a query point is computed by fitting a planar patch to the approximate nearest neighboring (ANN) points from the reference data [12]. The plane is least squares fitted [13], and the fitting error stored. If the plane fitting error is larger than a given threshold, we consider that the plane does not have sufficient support and the point-to-point metric is used instead.

Several approaches to dismiss possible outliers during the correspondence step in the ICP have also been proposed. We choose to remove wrong correspondences limiting the angle between adjacent normals and also rejecting those correspondences whose distances are larger than some multiple of the standard deviation of all distances within the matching set.

IV. 6 DOF POSE SLAM

We refer to Pose SLAM as the delayed-state information-form of SLAM in which one estimates the state vector $\mathbf{x}$, containing the history of poses from time 0 to k, given the history of proprioceptive observations Z and the set of motion commands U. Using the canonical parameterization,

$$p(\mathbf{x}|Z,U) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}^{-1}(\mathbf{x};\boldsymbol{\eta},\boldsymbol{\Lambda}), \tag{2}$$

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}, \text{ and } \boldsymbol{\eta} = \boldsymbol{\Sigma}^{-1}\,\boldsymbol{\mu}, \tag{3}$$

where $\boldsymbol{\mu}$ is the mean state vector and $\boldsymbol{\Sigma}$ its covariance matrix, and $\boldsymbol{\Lambda}$ and $\boldsymbol{\eta}$ are the information matrix and information vector, respectively.

In our implementation one robot pose (the k-th component of the state vector $\mathbf{x}$) is defined as

$$\mathbf{x}_k = \left[\mathbf{t}_k^\top, \boldsymbol{\Theta}_k^\top\right]^\top, \tag{4}$$

where $\mathbf{t}_k = [x_k, y_k, z_k]^\top$ indicates the position of the robot, and $\boldsymbol{\Theta}_k = [\phi_k, \theta_k, \psi_k]^\top$ is the vector of Euler angles to represent the orientation.

The noise-free motion model is defined using the compounding operation [14], and defines the state transition model, relating state components $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$,

$$\begin{aligned}\mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k)\\ &= \mathbf{x}_k \oplus \mathbf{u}_k,\end{aligned} \tag{5}$$

and $\mathbf{u}_k$ is the relative motion between consecutive poses as computed with the ICP algorithm, i.e. the relative travelled distance and the relative rotation change.

A first order Taylor series approximation of this model is given by

$$\mathbf{x}_{k+1} \approx f(\boldsymbol{\mu}_k, \mathbf{u}_k) + \mathbf{F}(\mathbf{x}_k - \boldsymbol{\mu}_k) + \mathbf{w}_k, \tag{6}$$

where

$$\mathbf{F} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\boldsymbol{\mu}_k, \mathbf{u}_k}, \tag{7}$$

and zero mean white noise $\mathbf{w}_k$ with covariance $\boldsymbol{\Sigma}_u$, used to accommodate for higher order terms and modeling errors.

We form our proprioceptive observation model also using the compounding operations. The noise-free measurement model is given by Equation 8, which tells us how much the robot has moved between any robot pose $\mathbf{x}_i$ and the current pose $\mathbf{x}_k$,

$$\begin{aligned}\mathbf{z}_{ik} &= h(\mathbf{x}_i, \mathbf{x}_k)\\ &= \ominus\mathbf{x}_i \oplus \mathbf{x}_k,\end{aligned} \tag{8}$$

The linearized measurement model is given by

$$\mathbf{z}_{ik} \approx h(\boldsymbol{\mu}_i, \boldsymbol{\mu}_k) + \mathbf{H}(\mathbf{x}_{i,k} - \boldsymbol{\mu}_{i,k}) + \mathbf{v}_k, \tag{9}$$

where $\mathbf{x}_{i,k} = [\mathbf{x}_i^\top, \mathbf{x}_k^\top]^\top$, $\mathbf{v}_k$ is zero mean white measurement noise with covariance $\boldsymbol{\Sigma}_z$, and

$$\mathbf{H} = \left[\left.\frac{\partial h}{\partial \mathbf{x}_i}\right|_{\boldsymbol{\mu}_i} \quad \left.\frac{\partial h}{\partial \mathbf{x}_k}\right|_{\boldsymbol{\mu}_k}\right]. \tag{10}$$

A. State Augmentation

In the delayed-state framework we do not marginalize out past robot poses as in other classical SLAM approaches such as the EKF and the EIF. Instead, we append the time-propagated robot pose $\mathbf{x}_{t+1}$ to the state vector, obtaining the prior probability distribution

$$\begin{aligned}&p\left(\mathbf{x}_{0:k}, \mathbf{x}_{k+1} | Z^k, U^{k+1}\right)\\ &\quad = p\left(\mathbf{x}_{0:k} | Z^k, U^k\right) p\left(\mathbf{x}_{k+1} | \mathbf{x}_k, \mathbf{u}_{k+1}\right),\end{aligned} \tag{11}$$

where $\mathbf{x}_{0:k}$ represents the robot trajectory before time $k+1$, and Z^k and U^k are the history of observations and odometry increments up to time k, respectively. This probability is factored into the product of the state posterior at time k and the transition probability multiplied by the prior probability —i.e. the posterior distribution computed at time k. For Gaussian distributions, the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\Lambda}$ of Eq. (11) in the form of (2) are given by

$$\boldsymbol{\eta}_{k,k+1} = \bar{\boldsymbol{\eta}}_{k,k+1} + \mathbf{F}_{\text{aug}}^\top \boldsymbol{\Sigma}_u^{-1}(f(\boldsymbol{\mu}_k, \mathbf{u}_k) - \mathbf{F}\boldsymbol{\mu}_k) \tag{12}$$

and

$$\Lambda_{k:k+1,k:k+1} = \bar{\Lambda}_{k:k+1,k:k+1} + \mathbf{F}_{\text{aug}}^\top \boldsymbol{\Sigma}_u^{-1} \mathbf{F}_{\text{aug}}, \tag{13}$$

in which $\mathbf{F}_{\text{aug}} = \begin{bmatrix} -\mathbf{F} & \mathbf{I} \end{bmatrix}$, and $\bar{\boldsymbol{\eta}}_{k+1}$ and $\bar{\boldsymbol{\Lambda}}_{k+1,k+1}$ represent the posterior information vector and information matrix at time k, with zero entries for time $k+1$, indicating infinite uncertainty for that robot pose.

The augmentation process introduces information only between the new robot pose $\mathbf{x}_{k+1}$ and the previous one $\mathbf{x}_k$. Moreover, the shared information between the new pose $\mathbf{x}_{k+1}$ and the rest of the robot trajectory $\mathbf{x}_{0:k-1}$ is always zero when we have not closed any loop. This matrix results in a naturally sparse information matrix with a tridiagonal block structure.

B. State Update

After augmenting the state we add observations with the EIF update equations,

$$\boldsymbol{\eta}_{i,k+1} = \bar{\boldsymbol{\eta}}_{i,k+1} + \mathbf{H}^\top \boldsymbol{\Sigma}_z^{-1}\left(\mathbf{z}_{k+1} - h(\boldsymbol{\mu}_i, \bar{\boldsymbol{\mu}}_{k+1}) + \mathbf{H}\bar{\boldsymbol{\mu}}_{i,k+1}\right) \tag{14}$$

$$\boldsymbol{\Lambda}_{i:k+1,i:k+1} = \bar{\boldsymbol{\Lambda}}_{i:k+1,i:k+1} + \mathbf{H}^\top \boldsymbol{\Sigma}_z^{-1}\mathbf{H}, \tag{15}$$

where $\mathbf{z}_{k+1}$ is the observation at time $k+1$, which is the relative pose measurements between the current pose $\mathbf{x}_{k+1}$ and any pose $\mathbf{x}_i$.

In the same way as with the prediction step, given the two-block size of the measurement Jacobian $\mathbf{H}$ in Eq. (10), only the four blocks relating poses i and $k+1$ in the information matrix will be updated.

C. Covariance and State Recovery

Motivated by [4], we employ a QR factorization of the information matrix to solve $\mathbf{\Lambda}\boldsymbol{\mu} = \boldsymbol{\eta}$ and $\mathbf{\Lambda}\mathbf{\Sigma} = \mathbf{I}$, for $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$.

In order to reduce the fill-in in the right triangular matrix from the QR factorization, we first reorder the information matrix using the column approximate minimum degree (COLAMD) ordering [15], then we apply QR factorization to the reordered information matrix and solve for each state variable via back substitution.

V. Experimental Setup

The goal application of the work presented in this paper is to build the required maps for a heterogeneous fleet of service robots in urban settings [1]. With these maps the robots should be able to perform path planning and navigate to accomplish their tasks, such as guidance, assistance, transportation of goods, and surveillance.

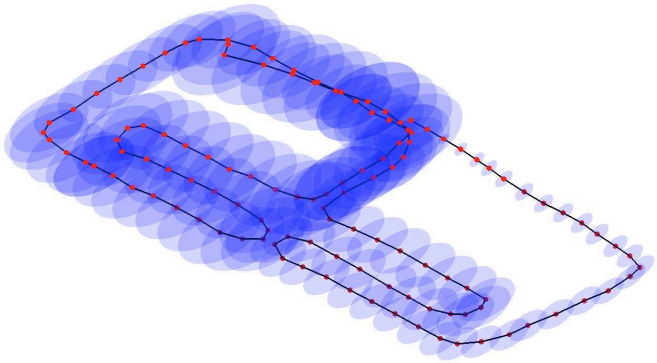
Our experimental site is the Barcelona Robot Lab, located at the Campus Nord of the Universitat Politècnica de Catalunya, part of the URUS project, and equipped with a camera network. This experimental area has over 15,000 square meters, several levels and underpasses, poor GPS coverage, moderate vegetation, several points with aliasing, large amounts of regularity from building structures, and sunlight exposure severely subject to shadows. An aerial view of the site is shown in Fig. 6.

In order to build the maps described in this paper, we built our proprietary 3D scanning system, using a Leuze RS4 scanner and controlling its pitch with a DC motor and a computer. The system was installed atop an Activmedia Pioneer 2AT robotic platform. The system yields 3D point clouds with ranges up to 30 meters, and sizes of about 76,000 points. The sensor noise level is ± 5 cm in depth estimation for each laser beam. Figure 3 portrays the complete device.

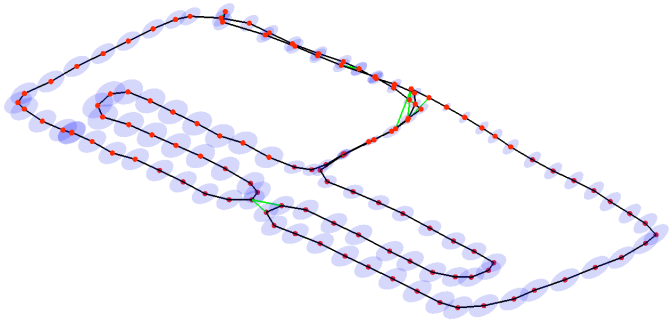
VI. 3D Mapping Results

The robot was teleoperated through the site along a path of over 600 m (see Fig. 4(a)). The figure contains results from state augmentation purely from concatenation of ICP computed motion constraints. The hyper-ellipsoids shown indicate marginal covariances of the robot position. Position uncertainty is larger along the direction perpendicular to the motion plane. This open loop traverse causes an increment of the accumulated estimation error. The mapping strategy discussed closes 19 loops, with the consequent improvement on localization uncertainty, as depicted in Fig. 4(b). The complete alignment of the 3D point clouds is shown in Fig. 1.

Fig. 3. 3D laser range finder mounted on our robotic platform.

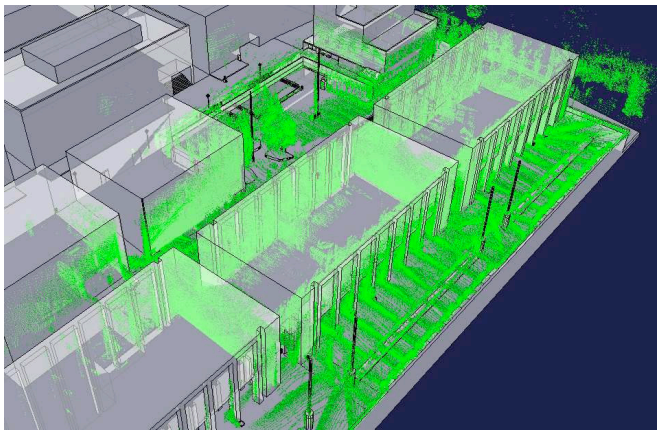


(a) State estimate before loop closure.

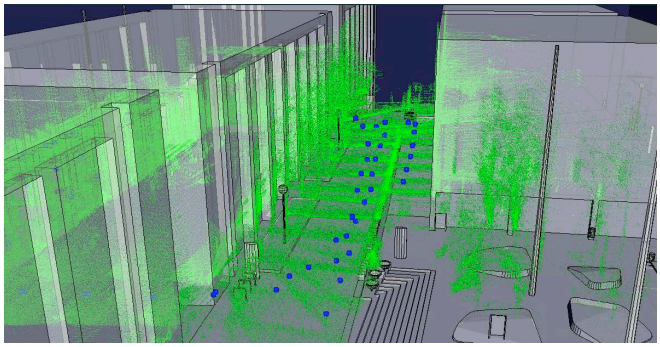


(b) Estimated trajectory after loop closure.

Fig. 4. 6D range-based SLAM results.



(a) Top view of the 3D map.



(b) A section of the map. The blue points indicate the robot position estimates.

Fig. 5. Projection of the 3D point cloud on a 3D georeferenced CAD model.

The accompanying video shows the complete map building process.

The results of our mapping technique are compared empirically to a manually built CAD model of the experimental site. The model is made using geo-referenced information. Figures 5(a) and 5(b) show two views of the final 3D point cloud map projected into the 3D model.

VII. Traversability Maps

After the 3D point clouds are aligned, we employ this final map to derive 2D layers that allow robots to perform localization. Additionally, from these 2D layers, we build traversability maps in the form of 2D grid maps where each cell indicates the maximum linear velocity in a given 2D robot position (x, y).

The 2D layers are extracted cutting the 3D map at the robot's frontal laser height. Then, using topological information the floor is removed. In the next step, the 2D layer is discretized and transformed into a binary image, which in turn is processed by morphological operations to increase the size of obstacles, to filter noise, and to fill gaps. Thus, this binary image is used as a grid map, with each cell representing the presence of an obstacle at that location in the environment.

Once a 2D layer is extracted and transformed into a grid map, we build its corresponding traversability map as follows. For every element in the configuration space we determine the maximum linear velocity that generates a collision-free path. Then, for a given robot position, we select the minimum velocity from all orientations. This process is detailed below.

We discretize the configuration space $\mathcal{C}$, in 10 cm for the robot position and 0.25 rad for robot orientation, and the action space $U = V \times \Omega$, in 0.1 m/s for linear velocity and 0.01 rad/s for angular velocity, where V and Ω are the sets of all possible linear and angular velocities, respectively.

Next, for each robot configuration $\mathbf{q}_i = (x, y, \theta)^\top$, we compute the set $A(\mathbf{q}_i)$ of all actions that generate a collision-free path for this configuration as follows. Given $\mathbf{q}_i \in \mathcal{C}$, we test every control action $u_j \in U$ using the kinematic model of our mobile robot, iterating k-times a fixed time step Δt, and for each action we generate a path

$$\tau : s \rightarrow X, \tag{16}$$

where $s \in [0, k\Delta t]$ and $X \in \mathcal{C}$. From this path, we add u_j to $A(\mathbf{q}_i)$ if, for every s, $\tau(s)$ is within the free space $\mathcal{C}_{\texttt{free}}$, wherein collision detection is performed using the previously computed grid map.

Finally, we compute a function that associates every robot position (x, y) to the maximum linear velocity $V_{\texttt{free}}$ that warrants a collision-free path

$$m : (x, y) \rightarrow V_{\texttt{free}}. \tag{17}$$

To compute $V_{\texttt{free}}$ we define the set $V(x, y, \theta)$, which contains all linear velocities from $A(\mathbf{q}_i)$, for configuration $\mathbf{q}_i = (x, y, \theta)^\top$, and the set

$$V(x, y) = \bigcup_{\theta \in \Theta} max\left(V(x, y, \theta)\right), \tag{18}$$

where Θ are all orientations from $\mathcal{C}_{\texttt{free}}$. In consequence, $V_{\texttt{free}} = min\left(V(x, y)\right)$ for a given robot position (x, y).

VIII. Conclusions

This paper presents an approach to the problem of 3D map building in urban settings for service robots, which consists of probabilistic alignment of 3D point clouds. A delayed-state EIF algorithm was employed to build the final map using only observations derived with our ICP algorithm. Relative pose constraints from consecutive robot poses were used to augment the state, and a sparse set of loop closures is used to refine the estimate reducing the accumulated drift.

Our version of the ICP algorithm employs both point-to-plane and point-to-point correspondence search at different levels of granularity. A distance metric that weighted differently rotations and translation was used. Values of L between 30 and 50 for this metric worked well for our data sets.

Regarding the estimation process, we can note that the delayed-state EIF allows an efficient and straightforward way to distribute error when closing a loop, since observation updates take constant time when an efficient or approximate way for state recovery is employed, such as the one we used

(a) 2D Layer superimposed on an aerial image

(b) Corresponding traversability map. Velocity varies from 0 m/s (blue) to 1 m/s (red).

Fig. 6. Traversability map from 2D layers of the aligned 3D point clouds.

here. Finally, with this approach we were able to close loops of aproximately 250 m of length, along a path of over 600 m.

Additionally we showed how, from the aligned 3D point cloud, one can compute maps useful for robots path planning and navigation; a much needed step usually neglected in most SLAM implementations. Traversability maps were derived by transforming the map of point clouds into a representation compatible with the robot kinematic chracteristics.

Given the empirical comparison with the geo-referenced CAD model and the orthographic views of the scene, our approach performed well enough to derive the traversability maps that allow service robots of our intended application [1] to navigate and accomplish their assigned tasks. We hypothesize that the overall estimation error of the presented method varies from 5 cm to 50 cm. These values however can not be verified with sufficient precision since no sufficiently accurate ground truth is available.

REFERENCES

[1] A. Sanfeliu and J. Andrade-Cetto, "Ubiquitous networking robotics in urban settings," in *Proc. IEEE/RSJ IROS Workshop Network Robot Syst.*, Beijing, Oct. 2006, pp. 14–18.

[2] P. Besl and N. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[3] L. Biota, L. Montesano, J. Minguez, and F. Lamiraux, "Toward a metric-based scan matching algorithm for displacement estimation in 3d workspaces," in *Proc. IEEE Int. Conf. Robot. Automat.*, Orlando, May 2006, pp. 4330–4332.

[4] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, 2008.

[5] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM-3D mapping outdoor environments," *J. Field Robot.*, vol. 24, no. 8-9, pp. 699–722, 2007.

[6] D. Cole and P. Newman, "3D SLAM in outdoor environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, Orlando, May 2006, pp. 1556–1563.

[7] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.

[8] V. Ila, J. Andrade-Cetto, R. Valencia, and A. Sanfeliu, "Vision-based loop closing for delayed state robot mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, Nov. 2007, pp. 3892–3897.

[9] S. Huang, Z. Wang, and G. Dissanayake, "Exact state and covariance sub-matrix recovery for submap based sparse EIF SLAM algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, Pasadena, Apr. 2008, pp. 1868–1873.

[10] V. Ila, J. Porta, and J. Andrade-Cetto, "Compact maps in pose SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Saint Louis, Oct. 2009.

[11] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3D Digital Imaging Modeling*, Quebec, May 2001, pp. 145–152.

[12] S. Arya and D. Mount, "Approximate nearest neighbor queries in fixed dimensions," in *Proc. ACM SIAM Sym. Discrete Algorithms*, Austin, Jan. 1993, pp. 271–280.

[13] J. Andrade-Cetto and M. Villamizar, "Object recognition," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed. New York: John Wiley & Sons, 2007, pp. 1–28.

[14] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, 1990, pp. 167–193.

[15] T. Davis, J. Gilbert, S. Larimore, and E. Ng, "A column approximate minimum degree ordering algorithm," *ACM T. Math. Soft.*, vol. 30, no. 3, pp. 353–376, 2004.