EKF-based 3D SLAM for Structured Environment Reconstruction

Jan Weingarten and Roland Siegwart
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Autonomous Systems Lab
CH-1015 Lausanne, Switzerland
Email: jan.weingarten@epfl.ch

***Abstract*— This paper presents the extension and experimental validation of the widely used EKF[1]-based SLAM[2] algorithm to 3D space. It uses planar features extracted probabilistically from dense three-dimensional point clouds generated by a rotating 2D laser scanner. These features are represented in compliance with the Symmetries and Perturbation model (SPmodel) in a stochastic map. As the robot moves, this map is updated incrementally while its pose is tracked by using an Extended Kalman Filter. After showing how three-dimensional data can be generated, the probabilistic feature extraction method is described, capable of robustly extracting (infinite) planes from structured environments. The SLAM algorithm is then used to track a robot moving through an indoor environment and its capabilities in terms of 3D reconstruction are analyzed.**

***Index Terms*— Extended Kalman Filter, 3D SLAM, Probabilistic Plane Extraction, SPmodel**

I. Introduction

A key issue in mobile robot navigation is localization and mapping. Only a robot capable of localizing itself in its environment can perform navigation tasks reliably. To achieve this, the robot needs an internal map representing its surrounding area. In metric navigation, commonly used map representations are based on geometrical features or occupancy grids. As the robot moves, it takes measurements of its environment and tries to match these appropriately to its internal map in order to keep track of its position. Hence, this requires a map which in general is not available a priori. A robot capable of creating a map of its environment and tracking its position at the same time is performing SLAM.

A. Motivation

Robot localization and mapping in structured or indoor environments is mainly done in 2D space until now. Most current metric approaches rely on a laser scanner sensor (see e.g. [1], [2], [3]) and use a probabilistic representation. All of these approaches can fail if the environment is non-planar, if the sensors are masked by a group of surrounding people at an exposition for example or simply if the robot is not restricted to the ground plane. This work attempts to address these problems by using a 3D sensor on a mobile robot and extending the popular EKF-based SLAM approach to 3D space.

Besides making robot navigation possible in 3D space, other interesting applications arise with the availability of precise and dense 3D data. Then the robot could be used to create precise and rich maps of its environment. An example of such an application is mapping of abandoned mines (see [4],[5]), while other applications may be found in fields like architecture, design, virtual reality and urban-search-and-rescue.

As 3D sensors suitable for mobile robots are still rare on the market, many groups build custom setups using the popular 2D laser scanner on a rotating support (see [4], [6], [7]). The data of these sensors can be composed of several hundred 2D scans leading to dense point clouds. Efficient data processing and a compact representation is therefore required to handle it efficiently.

B. Related Work

Horn et al. [8] corrected the planar (2D) pose of a mobile robot by using vertical planes extracted from 3D data. Sequeira et al. [9] used a single point laser mounted on a pan-tilt unit to create 3D models of indoor scenes. Their work mainly focussed on next-view planning and 3D reconstruction. Nuechter et al. [4] presented a scan alignment approach for 3D scans gathered by a rotating laser scanner sensor. They use the iterative closest point (ICP) algorithm to minimize a global error measure to generate consistent models. The approach presented in this paper relies on extracted features rather than raw data points leading to more compact and meaningful representations. Kohlhepp et al. [6] also use extracted planes to track the robot pose along with a EKF-based approach but represented the map in separate submaps. In this approach a single consistent stochastic map is used to represent the robot pose and all features.

II. Prerequisites

This section presents the sensor used throughout this work, a rotating Sick laser scanner and the used framework for processing uncertain geometric data, the Symmetries and Perturbations model (SPmodel).

A. The 3D sensor

The 3D sensor used (see Figure 1) is a custom setup based on a 2D laser scanner mounted on a support rotated by a

[1] Extended Kalman Filter

[2] Simultaneous Localization and Mapping

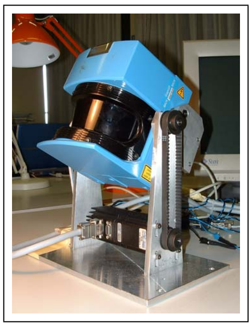


Fig. 1. The 3D sensor used is a Sick laser scanner mounted on a support which is rotated by a stepping motor via a v-ribbed belt. The angular resolution of the step motor is set to 0.45° with a range of 270°. The sensor produces 216961 data points (601 2D scans) per 3D scan.

stepping motor via a belt transmission. The angular resolution of the stepping motor is set to $0.45°$, the chosen number of steps is 601 with an angular range covering 270 degrees from $\phi_m^{min} = -45°$ to $\phi_m^{max} = 225°$ with respect to the horizontal plane. As the 2D laser scanner produces 361 per scan with an angular resolution of $0.5°$, the number of points of a complete 3D scan reaches $361 \times 601 = 216961$. The scanner currently takes around a minute for a complete scan which is not the fastest possible, but slow enough to avoid synchronization problems between the laser scanner and the stepping motor.

The main error sources are assumed to be found in the laser measurements along the direction of the laser and the stepping angle of the motor rotating the setup which is not completely free of play. The laser range error is assumed to be independent of the measured distance ρ_l and modelled by a Gaussian distribution with mean $\mu_l = \rho_l$ and variance $\sigma_{\rho_l}^2 = 0.02^2$m, the used angular positioning error of the stepping motor is also assumed to be Gaussian with mean $\mu_m = \phi_m$ and variance $\sigma_{\phi_m}^2 = 0.1^{2°}$. A simple calibration method using the laser scanner to find the front edge of the sensor setup was used to initialize the step counter correctly at each power-on.

B. Probabilistic Representation using the SPmodel

The Symmetries and Perturbations model (SPmodel) provides means to represent and process uncertain geometrical data [3]. A reference F is associated to a feature and the location of F with respect to the base reference W is defined by a transform $\mathbf{t}_{WF}$ or a location vector $\mathbf{x}_{WF}$ which is equivalent to the homogeneous matrix representation[3] H_{WF}

$$\begin{aligned}
\mathbf{t}_{WF} &= Trans(x, y, z)Rot(Z, \phi)Rot(Y, \theta)Rot(X, \psi)\\
\mathbf{x}_{WF} &= (x, y, z, \phi, \theta, \psi)^T\\
H_{WF} &= \begin{pmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi & x \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi s_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & y \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi & z \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

The composition of two location vectors is denoted by $\oplus$, the inverse of a location vector by $\ominus$.

In order to be able to cope with different geometric entities, the concept of symmetry representing the degrees of freedom of a geometrical object is introduced. An infinite plane for example has 3 degrees of freedom as it is defined by only 3 of the total 6 parameters (a distance and two angles). The set of all preserving translations and rotations with respect to the feature's local reference frame F, in this case the translations T_x, T_y in the plane, and the rotation R_ϕ around the plane normal, define the entity's symmetries. These are represented by a *row selection matrix* B_F, the so-called *binding matrix* of F, which for an infinite follows as

$$B_F = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

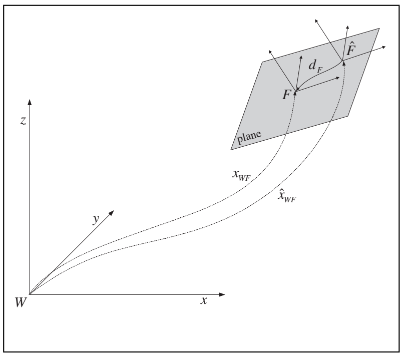


Fig. 2. A plane defined in compliance with the SP-model with a local coordinate frame $\hat{F}$ attached to the plane with its (local) x- and y-axes lying in the plane, a differential location vector d_F representing the estimation error and the true location of the plane depicted by F.

As depicted in Figure 2, the estimated location vector of an *uncertain* feature F is denoted by $\hat{\mathbf{x}}_{WF}$ and the associated error is expressed by a *differential location vector* $\mathbf{d}_F$ attached to the local reference frame, yielding the true location of the feature

$$\mathbf{x}_{WF} = \hat{\mathbf{x}}_{WF} \oplus \mathbf{d}_F$$

The components of $\mathbf{d}_F$ affected by preserving transforms are set to zero. The *perturbation vector* is formed by the non-zero elements of the differential vector $\mathbf{d}_F$:

$$\mathbf{d}_F = B_F^T \mathbf{p}_F; \qquad \mathbf{p}_F = B_F \mathbf{d}_F$$

Further on, the SPmodel defines an uncertain location L_{WF} of a feature F with respect to the world origin W as a quadruple

$$L_{WF} = (\hat{\mathbf{x}}_{WF}, \hat{\mathbf{p}}_F, C_F, B_F)$$

where $\hat{\mathbf{x}}_{WF} = (x, y, z, \phi, \theta, \psi)^T$ (the angles ψ,θ,ϕ are *RPY* angles) is the above-mentioned best estimate from the world origin W to the local reference of the feature F, $\hat{\mathbf{p}}_F$ is the local perturbation vector with associated error covariance matrix $C_F = E\left[(\mathbf{p}_F - \hat{\mathbf{p}}_F)(\mathbf{p}_F - \hat{\mathbf{p}}_F)^T\right]$ and B_F is the binding matrix encoding symmetries. Please refer to the literature for more details.

III. The Probabilistic Feature Extraction Process

This section describes the feature extraction process from the generation of 3D data up to the extraction of the infinite planes emphasizing the segmentation and probabilistic extraction of infinite planes. All modelled errors are assumed to be Gaussian which is required for the Kalman Filter.

[3] note that s_θ stands for $\sin(\theta)$, c_ψ for $\cos(\psi)$ etc.

A. Generating probabilistic 3D data

The uncertain location L_{RD_i} of a reconstructed 3D point in cartesian space is found by the composition of several uncertain SP-locations using standard error propagation

$$L_{RD_i} = L_{RS_{rot}} \oplus L_{S_{rot}S} \oplus L_{SD_i},$$

with $L_{RS_{rot}}$ being the transform from the robot base to the center of the rotation axis of laser scanner support. $L_{S_{rot}S}$ is the transform from the center of the rotation axis to the (rotated) rotating deflection mirror of the laser scanner and L_{SD_i} is the transform from this mirror to the actual measured value (see Figure 3).

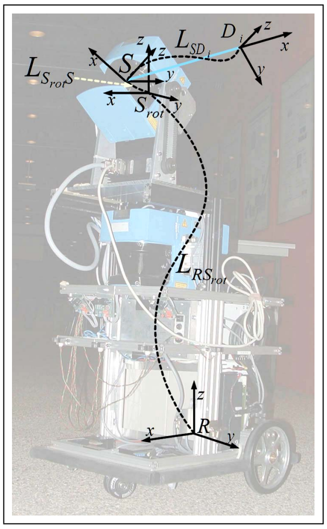


Fig. 3. The generation of uncertain cartesian data in the robot reference frame R starts with the measurement of D_i. The associated SP-location L_{SD_i} contains the modelled sensor errors affecting the measurement range ρ_l and the angular error e_{ϕ_m}. This error is then propagated through $L_{S_{rot}S}$ and $L_{RS_{rot}}$ leading to L_{RD_i}, which contains the measured point in cartesian space with associated covariance matrix.

B. Segmentation

The used segmentation method is based on the method presented in [10]. It starts by decomposing the space into regular cells which in this work are chosen to be cubes with a side length of 0.25m. After every raw data point has been associated to its corresponding cell, a plane is fitted to the points contained in every cell by using a Ransac algorithm [11] for segmentation with subsequent least-square fitting. The result of applying this plane extraction to every cell is depicted in Figure 4. The Ransac algorithm was chosen due to its simplicity and its robustness to outliers. After the extraction of a plane in every cell, a recursive region growing algorithm fuses similar planes from neighboring cells together (see Figure 4).

In general, the quality of this segmentation step could be improved by considering the uncertainty information associated to the raw data. However, in this specific case, as the sensor produces precise data and only large planes are extracted composed of thousands of data points, this method is considered appropriate.

C. Probabilistic Fitting

An infinite plane P can be described by two angles and an orthogonal distance to the origin of the coordinate frame. See [12] for a list of models describing infinite planes. Here, a normal vector $\mathbf{n} = (n_x, n_y, n_z)^T$ and a distance d_i is chosen as this notation is more convenient for the fitting process. The least-square linear regression starting point is the sum

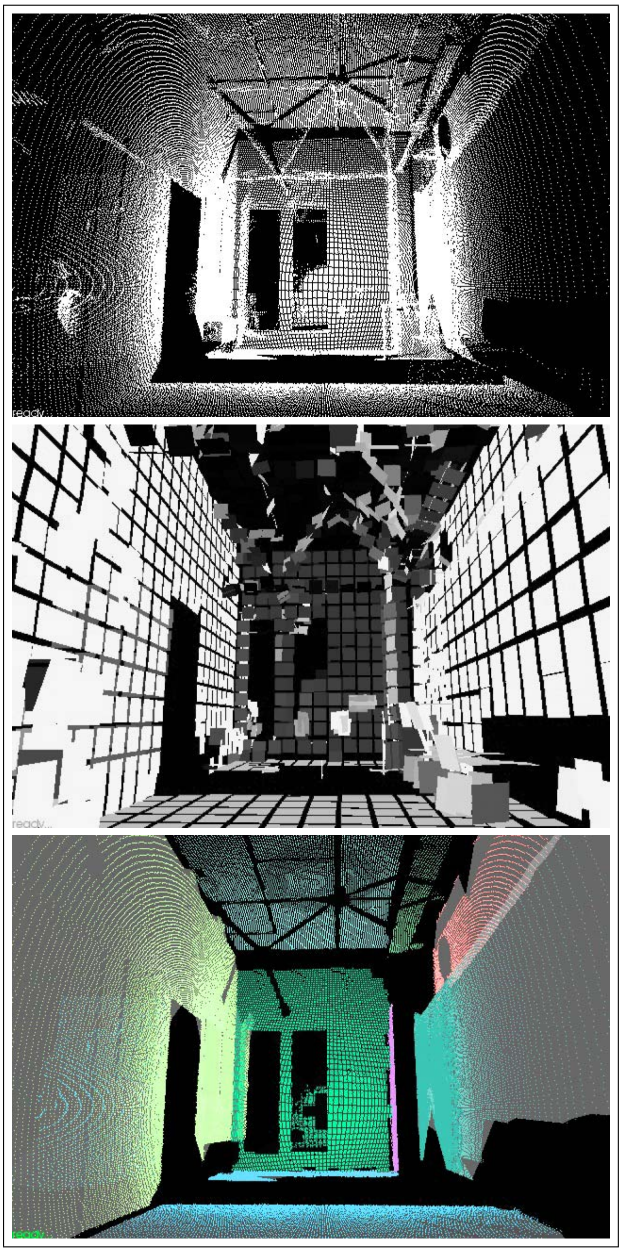


Fig. 4. The top image shows the raw 3D scan composed of 601 2D laser scans with 361 data points each. The image in the middle shows the result of extracting a plane for every cell. The bottom image shows the result of the region growing algorithm after merging similar planes of neighboring cells together.

$$S(n_x, n_y, n_z, d) = \sum_{i=1}^{N} w_i (n_x x_i + n_y y_i + n_z z_i - d)^2$$

which has to be minimized. The (x_i, y_i, z_i) with $i = 1 \ldots N$ are the points generated by the 3D sensor which are to be represented by a plane. $w_i = 1/trace(C_i)^2$ is a possible choice of a weighting factor allowing to incorporate uncertainty information described by the covariance matrix C_i of each data point. The plane normal $\mathbf{n} = (n_x, n_y, n_z)^T$ is found by calculating the eigenvector corresponding to the smallest eigenvalue of

$$\mathbf{A} = \begin{pmatrix} \sum_{i=1}^{N} w_i x_{t_i}^2 & \sum_{i=1}^{N} w_i x_{t_i} y_{t_i} & \sum_{i=1}^{N} w_i x_{t_i} z_{t_i} \\ \sum_{i=1}^{N} w_i x_{t_i} y_{t_i} & \sum_{i=1}^{N} w_i y_{t_i}^2 & \sum_{i=1}^{N} w_i y_{t_i} z_{t_i} \\ \sum_{i=1}^{N} w_i x_{t_i} z_{t_i} & \sum_{i=1}^{N} w_i y_{t_i} z_{t_i} & \sum_{i=1}^{N} w_i z_{t_i}^2 \end{pmatrix}$$

where $x_{t_i} = x_i - \bar{x}$, $y_{t_i} = y_i - \bar{y}$ and $z_{t_i} = z_i - \bar{z}$ are the raw data points translated to the center of gravity $(\bar{x}, \bar{y}, \bar{z})^T = \sum_{i=1}^{N} w_i (x_i, y_i, z_i)^T$. As the best-fitting plane passes through the center of gravity $\mathbf{g}$, the orthogonal distance to the origin is found by $d = \mathbf{g} \cdot \mathbf{n}$. The uncertainty of the extracted plane is calculated as described in [12].

IV. SLAM IN 3D

This section presents some aspects of the implemented SLAM algorithm which otherwise follows strictly the algorithm presented by Castellanos et al. [3]. It uses the Extended Kalman Filter to update the *SPmap*, which is the common stochastic map framework defined for the SPmodel. The next section briefly presents this SPmap, whereas the other section discusses the data association problem.

A. The SPmap

Similar to a location in 3D, the SPmap is defined by a quadruple $SPmap = (\hat{\mathbf{x}}^W, \hat{\mathbf{p}}^W, C^W, B^W)$ where $\hat{\mathbf{x}}^W = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{x}}_{WF_1}, \ldots, \hat{\mathbf{x}}_{WF_N})^T$ is a vector containing the location vector from the world frame W to the robot frame R as well as N location vectors to the feature references $F_1 \ldots F_N$. The vector $\hat{\mathbf{p}}^W$ contains all stacked perturbation vectors from the robot and the feature locations, C^W contains all covariance matrices and additionally correlation information, and $B_W = diag(B_R, B_{F_1}, \ldots, B_{F_N})$ is a block-diagonal matrix containing all binding matrices.

Note that the size of the covariance matrix C^W grows quadratically with an increasing number of features, which is a typical drawback of the stochastic map approach not further considered here. Refer to [3] for explicit update equations.

B. Data Association

The most crucial step of a feature-based SLAM approach is the data association, where pairings of predicted features and newly observed features have to be found in order to update the robot and feature positions represented in the SPmap respectively. Generally, the performance of the data association step depends on the quality of the odometry data, the update frequency of the exteroceptive sensor, and the model used to identify features. In this work, pairings $p_{ij} = (\hat{z}_i, z_j)$ of corresponding features are found by minimizing the Mahalanobis distance

$$m = (z_j - \hat{z}_i) S_{ij}^{-1} (z_j - \hat{z}_i)^T$$

where the $\hat{z}_i$'s are predicted features, the z_j's newly observed ones and S_{ij} is the innovation covariance of the pairing. If this found minimum satisfies the validation test

$$min(m) \leq \chi^2_{\alpha,n}$$

the pairing is accepted. $\chi^2_{\alpha,n}$ is taken from a χ^2-distribution with $n = 3$ degrees of freedom (in the case of infinite planar features) and the significance level α. A non-matched observation is added to the SPmap as new feature.

V. EXPERIMENTAL RESULTS

This section presents experimental results and is twofold. The first part deals with the analysis of the implemented Kalman Filter approach using a simple simulation, which allows to compare the estimated values to ground truth. The second one shows results using real data from the robot taken at our lab.

A. Simulation

A simple idealized environment composed of three orthogonal planes was used representing the minimal environment necessary to update all six degrees of freedom of the robot state vector $\mathbf{x}_{WR} = (x, y, z, \phi, \theta, \psi)^T$. The robot performs a sequence of 70 movements with a simulated 3D odometry composed of unsystematic and systematic errors. Figure 5 shows a visualization of the above-mentioned environment as well as the ground truth path, the odometry data and the corrected path of the robot. It can be seen that if the exteroceptive data is used (bottom image), the robot follows closely the ground truth trajectory (depicted in red). Figure 6 shows the quantitative error analysis of the same experiment. It can be seen, that the localization error of the robot pose stays bounded in all six components.

Many similar simulations were carried out with different trajectories and modified parameters all leading to satisfactory results. Consequently, this shows that 3D SLAM using the EKF and the SPmodel representation works. However, as the Jacobian matrices of the transforms in 3D space contain singularities [13], it is expected, that under certain conditions, the algorithm could show unexpected behavior, which will be investigated in the future. Further on, it should be mentioned that the simulation is not realistic in the sense that the data association is trivial. In real environments, wrong data associations can occur which may lead to filter divergence.

B. Real Experiments

Figure 7 shows a robot moving through a corridor and into two different rooms. On its way it takes 40 3D scans composed of 601 2D scans with 361 data points each. The small image on the right shows the reconstruction of the scene using odometry information only. The large image shows the same scene this time using the information of the exteroceptive sensor. It can be seen that by using the extracted planes, the estimated map and robot trajectory are much more consistent.

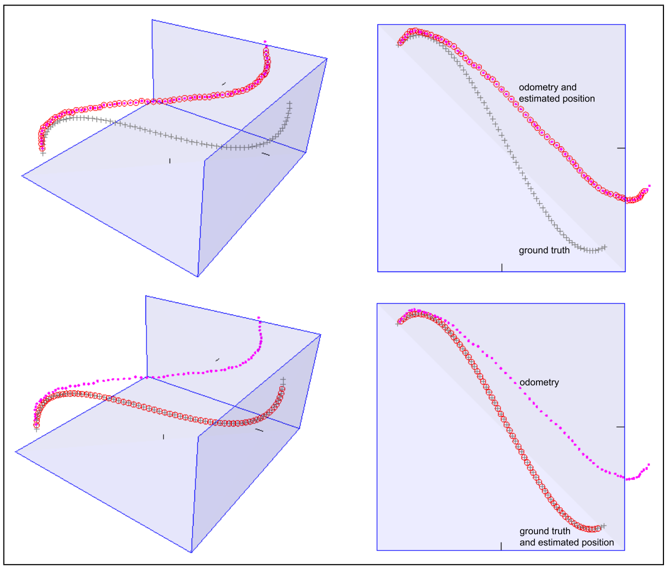


Fig. 5. This figure shows a visualization of the performed 3D SLAM simulation. The idealized environment is composed of 3 orthogonal planes (blue) with plane normals pointing inwards. The ground truth robot path is depicted by gray crosses and the odometry values by magenta dots. The top 2 images show a (trivial) SLAM experiment using odometry data only (right view is the scene from the top). As expected, the robot follows strictly the odometry path and its position error grows monotonically. The bottom images show the same experiment including a simulated exteroceptive sensor generating 3D data. It can be observed that when the robot considers the extracted planes, it follows well the ground truth trajectory.

The motion model used is based on 2D differential drive kinematics using the wheel encoder information. As this sensor cannot measure slope changes in undulated terrain, a sufficiently high error is associated to the missing components allowing to estimate the full 6-DOF robot pose even without a 6-DOF odometry.

Note that the resulting SPmap represents an environment with a size of about 150 square meters and contains 52 infinite planes which can be stored in several kBytes of memory only.

VI. Conclusion and Future Work

This paper presents how an EKF-based SLAM algorithm can be used in 3D space. It shows that by relying on infinite planar features extracted form three-dimensional point clouds, the robot pose error accumulated by the odometry can be corrected and a consistent map of the environment can be built. Further on, the algorithm leads to a compact map representation utilizing several kByte of memory only, which can be useful for efficient robot localization for example. It is also shown that such a SLAM algorithm used in conjunction with a 3D sensor generating dense data, rich maps of the robot's environment can be created.

Future work will consist of refining this method by thoroughly investigating the issues rising from orientations in 3D. Then, more discriminatory features like plane segments will be used to increase the performance of data association and produce more meaningful maps. The ultimate goal is to develop a system capable of generating polygonal models of unknown environments autonomously.

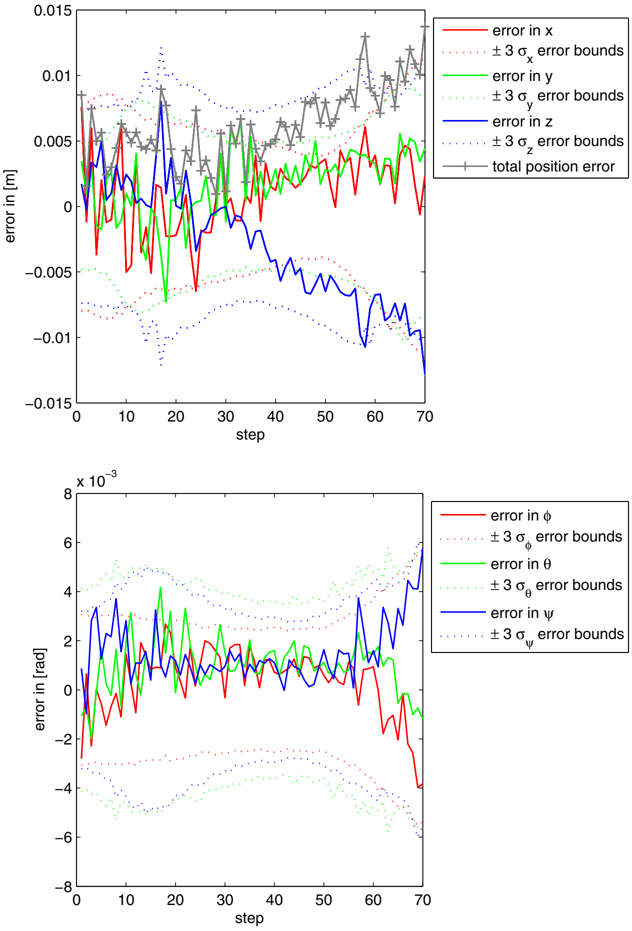


Fig. 6. The results of the 3D SLAM simulation showing the position $(x, y, z)^T$ and angular $(\phi, \theta, \psi)^T$ error analysis of the corrected robot trajectory. It can be observed that all 6 state vector components stay bounded.

References

[1] D. Fox, W. Burgard, S. Thrun, "Active Markov Localization for Mobile Robots", *Robotics and Autonomous Systems*, 1998, v. 25, n. 3-4, pp. 195-207.

[2] K.O. Arras , J.A. Castellanos, M. Schilt and R. Siegwart, "Feature-based multi-hypothesis localization and tracking using geometric constraints", *Elsevier, Robotics and Autonomous Systems*, 2003, v. 1056, pp. 1-13.

[3] José A. Castellanos, Juan D. Tardós, "Mobile Robot Localization And Map Building", *Kluwer Academic Publishers*, 1999.

[4] A. Nüchter, H. Surmann, "6D SLAM with an Application in Autonomous Mine Mapping", *Proceedings of ICRA*, 2004, New Orleans.

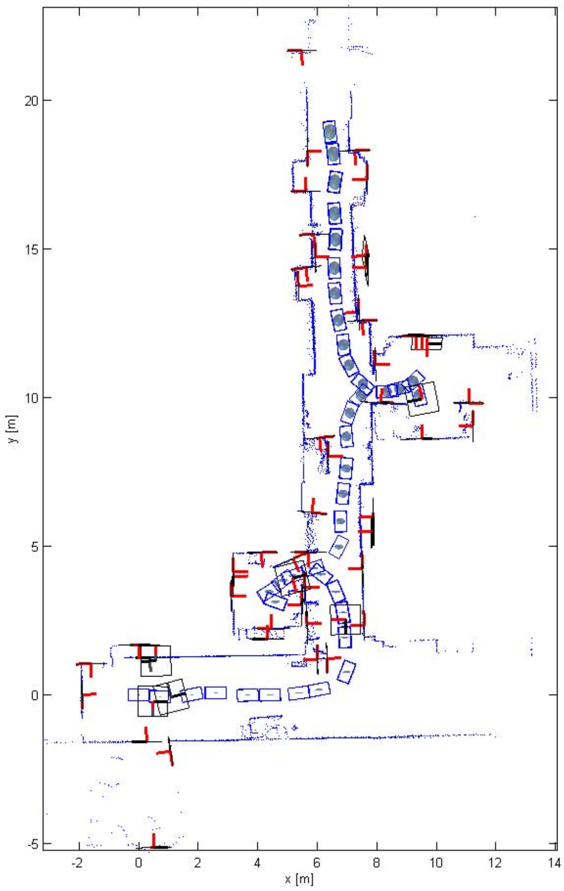
[5] D. Ferguson, A. Morris, D. Hähnel, C. Baker, Z. Omohundro, C. Reverte, S. Thayer, W. Whittaker, W. Burgard, S. Thrun, "An Autononous Robotic System for Mapping Abondoned Mines", *Advances in Neural Information Processing Systems*, 2003.

[6] P. Kohlhepp, P. Pozzo, M. Walter, R. Dillmann, "Sequential 3D-SLAM for mobile action planning", *Proceedings of IROS*, 2004, Sendai.

[7] D. Hähnel, D. Schulz, W. Burgard, "Map Building with Mobile Robots in Populated Environments", *Proceedings of IROS*, 2002.

[8] J. Horn, G. Schmidt, "Continuous localization of a mobile robot based on 3D-laser-range-data, predicetd sensor images, and dead-reckoning", *Robotics and Autonomous Systems*, 1995, v. 14, pp. 99-118

[9] V. Sequeira, J. G. M. Gonçalves, M.I.Ribeiro,"3D Reconstruction of Indoor Environments"*Proceedings of Internation Conference on Image Processing*, 1996, v. 2, pp. 405-408.



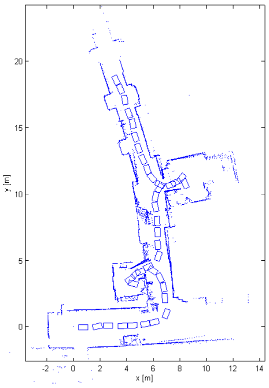


Fig. 7. This figure shows the results of one of the performed SLAM experiments. The robot moves along a corridor entering 2 rooms on its way starting from the bottom left. The travelled distance is about 40 meters. On its way it stops 40 times to scan its environment in 3D. Planes are extracted from these scans and appropriately matched to the stochastic map following the presented EKF-SLAM algorithm. Note that 3σ uncertainty ellipses represent the uncertainty of the estimated robot pose and coordinate frames depict the mapped features, most of which are perpendicular to the xy-plane as they represent walls, doors etc. The shown 2D data represents part of the 3D scans to clarify visualization. The small view on the right shows the result using odometry data only.

[10] J. Weingarten, G. Gruener, R. Siegwart, “A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction”, *Proceedings of ICAR*, 2003, Coimbra.

[11] Martin A. Fischler and Robert C. Bolles, ”Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, *Communications of the ACM*, June 1981, v. 24, n. 6, pp. 381-395.

[12] J. Weingarten, G. Gruener, R. Siegwart, “Probabilistic Plane Fitting in 3D and an Application to Robotic Mapping”, *Proceedings of ICRA*, 2004, New Orleans.

[13] Randall Smith, Matthew Self, Peter Cheeseman, ”Estimating Uncertain Spatial Relationships in Robotics”, *in I. J. Cox and G. T. Wilfong, editors, Autonomous Robot Vehicles, Springer-Verlag*, 1990, pp. 167-193.

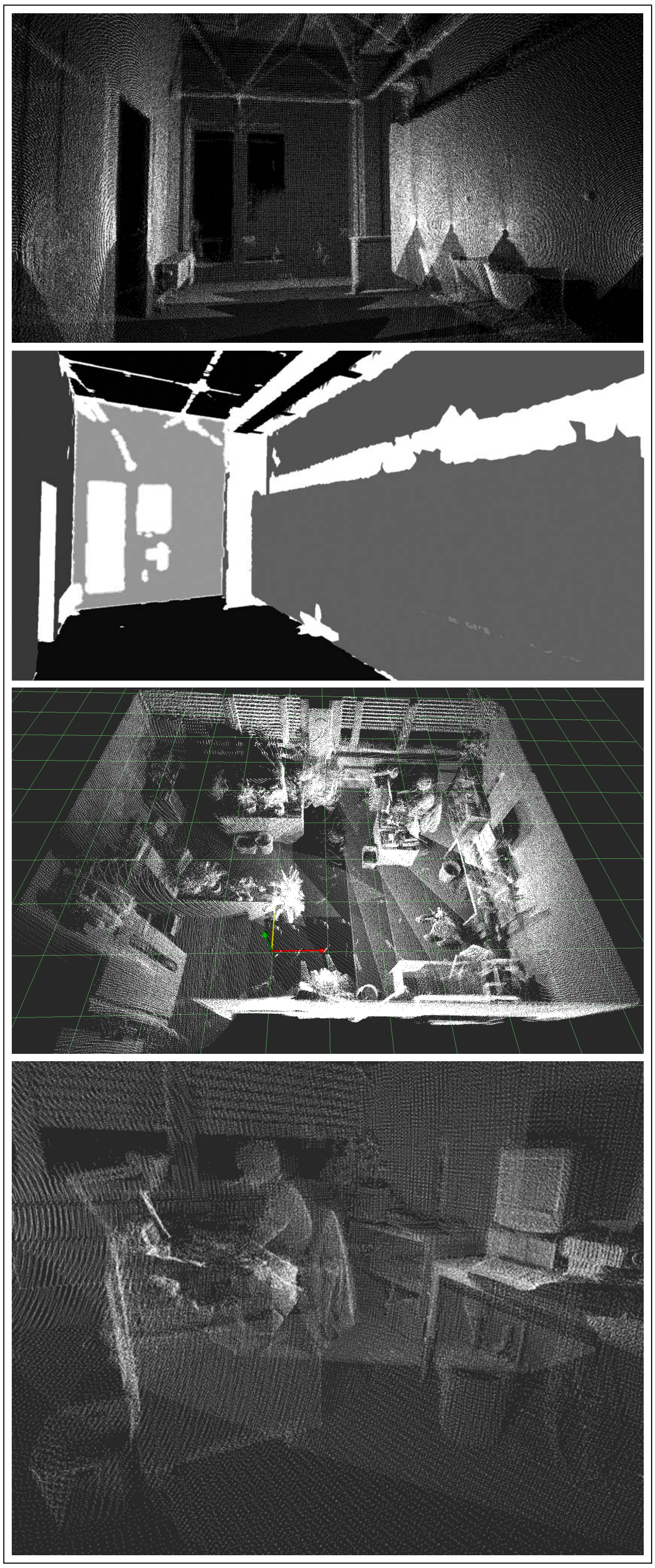


Fig. 8. This figure shows the 3D reconstruction capabilities of the algorithm. The top image shows a part of a corridor of our lab reconstructed as a point cloud. The image below shows the learned planes included in the SPmap representing the same scene. These planes are visualized as alpha-shapes using the supporting point information. The two bottom images show a reconstructed office based on 7 aligned scans. Note that the bottom image represents a part of the above images showing clearly distinguishable details like for example a computer, a working person, a trash can and a lamp.