

Probabilistic Plane Fitting in 3D and an Application to Robotic Mapping

Jan W. Weingarten

Autonomous Systems Lab

Swiss Federal Institute Of Technology (EPFL)

Lausanne, Switzerland

Email: jan.weingarten@epfl.ch

Gabriel Gruener

Swiss Center for Electronics

and Microtechnology

Alpnach Dorf, Switzerland

Roland Siegwart

Autonomous Systems Lab

Swiss Federal Institute Of Technology (EPFL)

Lausanne, Switzerland

Abstract— This paper presents a method for probabilistic plane fitting and an application to robotic 3D mapping. The plane is fitted in an orthogonal least-square sense and the output complies with the conventions of the Symmetries and Perturbation model (SPmodel). In the second part of the paper, the presented plane fitting method is used within a 3D mapping application. It is shown that by using probabilistic information, high precision 3D maps can be generated.

I. INTRODUCTION

One main goal within the research area of mobile robotics is to reach the highest possible grade of robustness and precision. In localization and mapping applications, this depends directly on the quality of the used perceptive sensors, the odometry, the computer hardware and the software implementing the underlying models and algorithms. With the appearance of accurate and affordable 2D laser scanners on the market some years ago, a prerequisite for big progress was set resulting in robots that can safely navigate and interact in highly dynamic and populated environments like museums or exhibitions (see Arras et. al [9] or Thrun et al. [5]).

A. Related Work

Recent work from Hähnel et al. [3], Liu et al. [4], Moravec et al. [14] or Surmann et al. [15] shows that extending localization and mapping methods to 3D is very promising. On the one hand, automatically generated 3D models are useful as a visualization for architects, fire-fighters or virtual reality applications. On the other hand, the dense three-dimensional information could help the robot itself improve its navigation capabilities. Up to now, the used sensor systems for 3D are either based on 2D laser scanners (see Thrun et al. [7], [3], [4], [15]) or stereo vision cameras (see Iocchi et al. [16] or [14]) still limiting the associated applications to stop-and-go motion navigation or pure 3D mapping. Cameras producing three-dimensional data in real-time have recently become available (see Lange et al. [17]) ensuring further progress.

B. Motivation

Current navigation approaches using 2D information can fail due to limitations of the perceptive sensors. When using a standard horizontal laser scanner for example, people standing around the robot can make the robot go blind and objects like

table tops or staircases can lead to severe crashes. Using 3D information is therefore a logical next step in order to develop more robust and powerful robots with various new applications for example within the areas of virtual reality and flying robots (e.g. Thrun et. al [19]).

In comparison to two-dimensional data, the denser 3D data requires more memory and computer power. Feature-based approaches provide means of overcoming this problem by representing hundreds of raw data points as a single feature.

The crucial part of feature-based approaches is the feature extraction itself, more precisely the segmentation and fitting procedure. Due to sensor limitations, it can be very hard to find and extract features reliably especially in dynamic and complex environments. Nevertheless, a good feature extraction rewards with very robust and precise results.

State-of-the-art feature extraction algorithms use probabilistic information and have proven to be superior to classical approaches. The goal of this work is to create the basic framework for probabilistic plane fitting and to show its usefulness in a typical application for robotic 3D mapping.

The paper is composed of two main parts. The first part deals with probabilistic plane-fitting using the SPmodel framework and the second presents a 3D mapping application using the presented plane fitting method. The next section covers the prerequisites.

II. PREREQUISITES

A. SPmodel

The Symmetries and Perturbations Model (SPmodel) is a framework for representing and processing erroneous geometrical data (Castellanos et al. [2] and Smith et al. [1]). Within this framework, the location \mathbf{L}_{WF} of a geometrical object F is defined by four parameters: The transformation $\mathbf{x}_{WF} = (x, y, z, \phi, \theta, \psi)^T$ ¹ from the world coordinate frame W into the local object coordinate frame F , the binding matrix \mathbf{B}_F accounting for symmetries, the locally defined perturbation vector² \mathbf{p}_F representing the error with its associated covariance matrix \mathbf{C}_F containing the uncertainty information.

¹The angles ϕ, θ, ψ represent the RPY-rotations (roll, pitch, yaw) around the z-axis, the y-axis and the x-axis, respectively

²The perturbation vector \mathbf{p}_F is formed by multiplying the binding matrix \mathbf{B}_F with the full rank differential vector $\mathbf{d}_F = (d_x, d_y, d_z, d_\phi, d_\theta, d_\psi)^T$

It is important to mention, that the SPmodel is generally defined for geometrical objects with six parameters (6 degrees of freedom), which typically is a location in 3D, using the concept of a row selection matrix or binding matrix \mathbf{B} to adapt the dimension to the actual case. A point in 3D space for example has no orientation and hence can be described by three translations. An infinite plane can also be described by three parameters, two angles and the perpendicular distance to the origin (see Castellanos et al. [2] for details).

The SP-framework provides several operations to transform geometrical objects defined by so-called locations: the most important are the *composition* \oplus of two locations and the *inversion* \ominus of a location. In visual terms, the composition of two locations is their concatenation considering the propagation of associated errors. Within the scope of this work, the SPmodel is used wherever uncertain information appears and has to be transformed.

B. Hardware

A mobile robot equipped with two opposing horizontal SICK Laser Range Scanners (LMS 200) sensors (for a field of view of 360 degrees) is used. It includes an Extended Kalman Filter-based 2D localization system using linear features and assuming Gaussian error distributions developed by Arras et al. [8]. In this case, it requires an a priori map of the environment. This feature-based localization method reaches subcentimeter precision. An additional, vertically mounted laser scanner is used to generate the three-dimensional data (see Figure 1) like it is done by Liu et al. [4] or Hähnel et al. [3] for example.

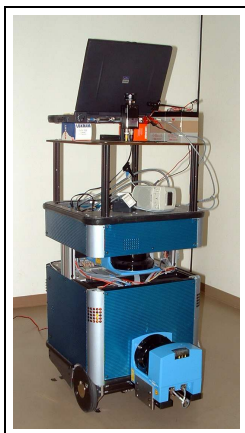


Fig. 1. Pygmalion robot with two horizontal SICK LMS 200 scanners with a total field of view of 360 degrees for two-dimensional localization and a vertical scanner of the same type for 3D data generation.

C. Data Registration Process

Associating the data of the vertical SICK sensor with the chronologically correct robot position³ estimate yields a registered point cloud $P = \{\mathbf{p}_i = (x_i, y_i, z_i)^T | i = 1..N\}$ in 3D. More precisely, the transformation⁴ \mathbf{x}_{WR} from the world coordinate frame W into the robot coordinate frame R defined by $\mathbf{x}_{WR} = (x_{robot}, y_{robot}, 0, \phi_{robot}, 0, 0)^T$ is compounded

³The vertical sick scans are synchronized with the robot localization system through linear interpolation

⁴In the context of the SPmodel, transformation is used as a synonym of location

with the transformation $\mathbf{x}_{RS_{p_i}}$ from the robot coordinate frame R into the frame of a single sensor value S_{p_i} defined by $\mathbf{x}_{RS_{p_i}} = (0, y_{sensor_i}, z_{sensor_i}, 0, 0, 0)^T$, yielding the transformation from the world frame W into the sensor value frame S_{p_i} defined by $\mathbf{x}_{WS} = \mathbf{x}_{WR} \oplus \mathbf{x}_{RS_{p_i}}$.

The error sources of the generated N points P are two-fold and have to be combined appropriately. Firstly, the robot position error describing how precisely the robot pose $(x, y, \theta)^T$ on the floorplane is known has to be considered:

$$\epsilon_{robot} \sim N(0, \Sigma_{robot}), \Sigma_{robot} = \begin{pmatrix} \sigma_{x^2} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_{y^2} & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_{\theta^2} \end{pmatrix}$$

Secondly the sensor error ϵ_{sensor} , which in turn is two-fold, composed of the angular error and the distance error, has to be taken into account. As the distance error is much greater than the angular error, the latter is omitted (see e.g. Wang et al. [18]). Assuming a normal distribution can be justified by the central limit theorem of statistics, yielding $\epsilon_{sensor} \sim N(0, \sigma_{sensor})^5$.

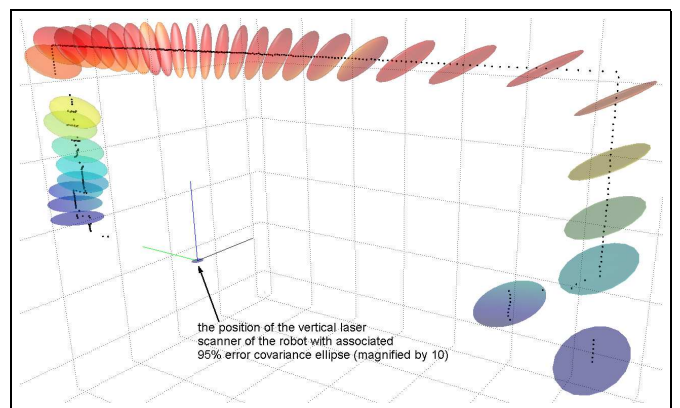


Fig. 2. A single scan composed of 361 data points with associated uncertainty information. The uncertainty resulting from the composition of the robot estimate error and the sensor error is visualized for every tenth data point and depicted by 95% error ellipsoids magnified by 10.

The overall error of a single three-dimensional point \mathbf{p}_i is found automatically by the SPmodel framework which relies on the standard laws of error propagation calculus [1]. Figure 2 shows a single laser scan with associated uncertainty ellipsoids.

III. PROBABILISTIC PLANE FITTING IN 3D

This section describes a way of fitting infinite planes to uncertain three-dimensional data. It is adapted to the representation conventions of the SPmodel in the way that it firstly finds a transform to a local coordinate frame lying within the plane and then analyzes the error locally.

The range data is given as a set of three-dimensional points $\{\mathbf{p}_i = (x_i, y_i, z_i)^T | i = 1..N\}$ with associated covariance matrices $\{C_i | C_i \in \mathbb{R}^{3 \times 3}\}$. To find the best-fitting (infinite) plane in a weighted least-square sense, a representation has

⁵ $\sigma = (0.015)^2$ [m] throughout this work

#	Equation	Plane Parameters
1	$\mathbf{n}\mathbf{x} - d = 0$	$\mathbf{n} = (n_x, n_y, n_z)^T, d$
2	$x \cos \theta \cos \varphi + y \cos \theta \sin \varphi + z \sin \theta - \rho = 0$	θ, φ, ρ
3	$Z = aX + bY + d$	a, b, d
4	$ax + by + cz + 1 = 0$	a, b, c

TABLE I. Representations of a plane in 3D

to be chosen and weights w_i have to be defined. The weights $w_i = 1/\text{trace}(C_i)^2$ have proven to be an adequate choice. A more thorough investigation of weighting factors will be faced in the future.

A. Representations

Different plane models can be found in the literature (see Table II). Not all of them are well-suited for least-square fitting problems and error analysis. The Hesse notation (Model 1 of table I) is the most general model that can be converted into all the other models:

$$\mathbf{n}\mathbf{p}_i - d \equiv n_x x_i + n_y y_i + n_z z_i - d = 0 \quad (1)$$

All data points $\mathbf{p}_i = (x_i, y_i, z_i)^T$ that lie on the plane defined by the normal $\mathbf{n} = (n_x, n_y, n_z)^T$ and the perpendicular distance to the origin d satisfy the above equation. In reality however, only very few data points lie exactly on the plane, hence the value ϵ is introduced on the right of (1) standing for the fitting error, which corresponds to a perpendicular distance to the plane:

$$n_x x_i + n_y y_i + n_z z_i - d = \epsilon_i \quad (2)$$

Taking the sum over all squared distances yields the regression problem $R(n_x, n_y, n_z, d) = \sum_{i=0}^N \epsilon_i^2 = \sum_{i=0}^N (n_x x_i + n_y y_i + n_z z_i - d)^2$ solved in the next section. Dividing (2) by $(-d)$ and substituting accordingly yields model 4:

$$ax_i + by_i + cz_i + 1 = \frac{\epsilon_i}{-d}$$

The associated regression problem does not minimize the sum of the squared distances but $\sum_{i=0}^N (\epsilon_i/(-d))^2$.

Model 3 is found dividing (2) by n_z and substituting again.

$$Z - aX - bY - d = \frac{\epsilon_i}{-n_z}$$

Again it can be seen, that the sum to be minimized $\sum_{i=0}^N (\epsilon_i/(-n_z))^2$ corresponds to the orthogonal least-square distance only when $n_z = \pm 1$.

Model 2 leads to a nonlinear regression problem which is difficult to tackle analytically.

From the comparison above, it can be seized that the Hesse notation (Model 1) of a plane is the most flexible as it has the least number of constraints. In this work, it is used in combination with Model 3 (see next section).

B. Plane Fitting

Plane fitting is done in three steps:

1. Plane Fitting With Principal Component Analysis (PCA)

2. Transformation of the input data points into the global xy -plane
3. Uncertainty analysis using standard regression methods

These steps are described in more detail in the following.

1) *Plane Fitting With PCA*: Choosing the Hesse-model yields the function

$$R(n_x, n_y, n_z, d) = \sum_{i=0}^N w_i (n_x x_i^r + n_y y_i^r + n_z z_i^r - d)^2 \quad (3)$$

which has to be minimized. "r" stands for raw data. Deriving (3) with respect to d and setting it equal to 0 yields

$$\begin{aligned} \partial_d R(n_x, n_y, n_z, d) &= 0 \\ \equiv \partial_d \sum_{i=0}^N w_i (n_x x_i^r + n_y y_i^r + n_z z_i^r - d)^2 &= 0 \\ \Rightarrow 2 \sum_{i=0}^N w_i (n_x x_i^r + n_y y_i^r + n_z z_i^r - d)(-1) &= 0 \\ \Leftrightarrow \sum_{i=0}^N w_i (n_x x_i^r + n_y y_i^r + n_z z_i^r) &= \sum_{i=0}^N w_i d \\ \Leftrightarrow \frac{1}{\sum_{i=0}^N w_i} \sum_{i=0}^N w_i (n_x x_i^r + n_y y_i^r + n_z z_i^r) &= d \\ \Leftrightarrow \frac{1}{\sum_{i=0}^N w_i} \begin{pmatrix} \sum_{i=0}^N w_i x_i \\ \sum_{i=0}^N w_i y_i \\ \sum_{i=0}^N w_i z_i \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} &= d \\ \Leftrightarrow \mathbf{cog} \cdot \mathbf{n} &= d \end{aligned}$$

This means that the best-fitting plane passes through the center of gravity represented by \mathbf{cog} . Translating the data points into the origin yields $S(n_x, n_y, n_z) = \sum_{i=0}^N w_i (n_x x_i + n_y y_i + n_z z_i)^2$. The plane normal $\mathbf{n} = (n_x, n_y, n_z)^T$ can be found by calculating the eigenvector corresponding to the smallest eigenvalue of

$$\mathbf{A} = \begin{pmatrix} \sum_{i=0}^N w_i x_i^2 & \sum_{i=0}^N w_i x_i y_i & \sum_{i=0}^N w_i x_i z_i \\ \sum_{i=0}^N w_i x_i y_i & \sum_{i=0}^N w_i y_i^2 & \sum_{i=0}^N w_i y_i z_i \\ \sum_{i=0}^N w_i x_i z_i & \sum_{i=0}^N w_i y_i z_i & \sum_{i=0}^N w_i z_i^2 \end{pmatrix}$$

2) *Transformation of the data*: With the aid of the perpendicular distance $d = \|\mathbf{cog} \cdot \mathbf{n}\|$ from the plane to the origin a transformation $x_{SP\text{Plane}XY}$ can be defined in order to transform the raw data S_P into the global xy -plane⁶.

3) *Uncertainty Analysis*: After having transformed the data into the global xy -plane this point cloud $P^t = \{p_i^t = (x_i^t, y_i^t, z_i^t)^T | i = 1..N\}$ and the covariance matrices $\{C_i^t | i = 1..N\}$ will again be fitted to a plane model, but this time using the Model 3 of Table II. This allows to find an analytical expression for the first and second moments of the plane, the latter corresponding to the covariance matrices. As the best

⁶Transforming data into a plane means rotating and translating the point cloud in a way that its principal axes (found through PCA) correspond to the world coordinate system

fitting plane now has to lie within the global xy -plane and therefore $(n_x, n_y, n_z) = (0, 0, 1)^T$, the regression equation $T(n_x, n_y, n_z, d) = \sum_{i=0}^N w_i (n_x x_i^t + n_y y_i^t + n_z z_i^t - d)^2$ can be divided by n_z and with the substitutions $n_x/n_z = b_1$, $n_y/n_z = b_2$ and $(-d)/n_z = b_0$ simplified into:

$$T(b_0, b_1, b_2) = \sum_{i=0}^N w_i (z_i^t + b_0 + b_1 x_i^t + b_2 y_i^t)^2$$

After taking partial derivatives with respect to b_0 , b_1 and b_2 , the parameters for the best-fitting plane are:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = A^{-1} \begin{pmatrix} -\sum_{i=0}^N w_i z_i^t \\ -\sum_{i=0}^N w_i x_i^t z_i^t \\ -\sum_{i=0}^N w_i y_i^t z_i^t \end{pmatrix}$$

$$A = \begin{pmatrix} \sum_{i=0}^N w_i 1 & \sum_{i=0}^N w_i x_i^t & \sum_{i=0}^N w_i y_i^t \\ \sum_{i=0}^N w_i x_i^t & \sum_{i=0}^N w_i (x_i^t)^2 & \sum_{i=0}^N w_i x_i^t y_i^t \\ \sum_{i=0}^N w_i y_i^t & \sum_{i=0}^N w_i x_i^t y_i^t & \sum_{i=0}^N w_i (y_i^t)^2 \end{pmatrix}$$

The covariance matrix of the parameter vector $\mathbf{b} = (b_0, b_1, b_2)^T$ can be found with the standard law of error propagation:

$$C_{plane} = F \begin{pmatrix} C_1^t & 0 & \dots & 0 \\ 0 & C_2^t & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_N^t \end{pmatrix} F^T$$

with $F = \nabla(b_0, b_1, b_2)^T$. A second Taylor approximation leads to the final result for the plane parameters $(z, \theta, \psi)^T = (\arctan b_0, \arctan b_1, \arctan b_2)^T$ and corresponding covariance matrix $C_{SPplane}$, which can directly be input into the SPmodel.

$$C_{SPplane} = \begin{pmatrix} \sigma_{zz^2} & \sigma_{z\theta} & \sigma_{z\psi} \\ \sigma_{z\theta} & \sigma_{\theta^2} & \sigma_{\theta\psi} \\ \sigma_{z\psi} & \sigma_{\theta\psi} & \sigma_{\psi^2} \end{pmatrix} = F C_{plane} F^T$$

$$F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{1+b_1^2} & 0 \\ 0 & 0 & \frac{1}{1+b_2^2} \end{pmatrix}$$

C. Remark

Singularities of the Jacobian matrices exist in the rotational part (ϕ, θ, ψ) of a transform L [1]. They have not been taken into account here because 3D points are only represented by a translation transform (x, y, z) .

IV. ROBOTIC MAPPING APPLICATION

A. Introduction

A typical application for plane-fitting can be found within the area of robotic 3D mapping. There, the goal is to convert a generally vast set of raw data into a compact representation, overcoming problems due to noise and low rendering capability (see Figure 3). Furthermore, extracted features like planes have a richer semantical value than points and can be used for higher level machine learning tasks like robot

navigation. Using specific features obviously narrows the area of application of the mapping system. But as this work focuses on mapping indoor or structured environments which mainly consist of planar structures like walls, ceilings, closets, doors, etc., it is assumed that representing the environment by means of planar features is an appropriate choice. Thrun et al. [6] call this the "structured environment assumption".

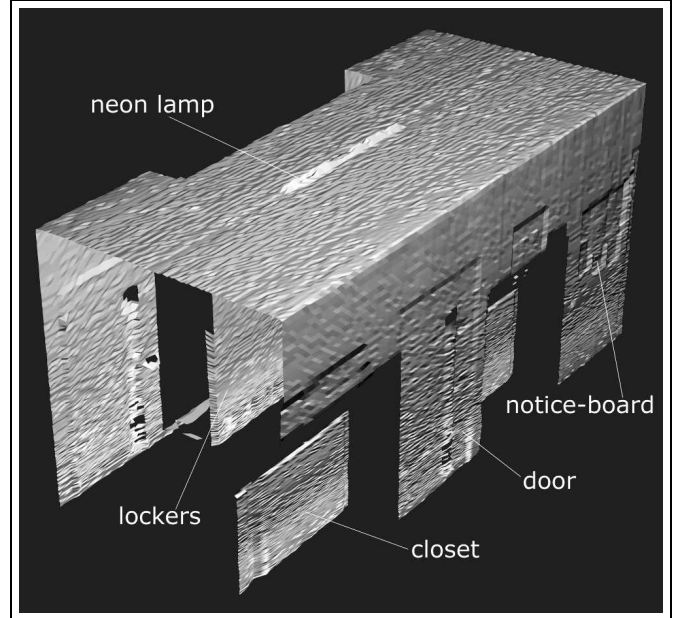


Fig. 3. This image shows 100 consecutively taken SICK scans (3610 points) representing a part of a corridor at the Autonomous Systems Lab. Note that doors, lockers, closets, neon lamps and even notice-boards can be recognized.

B. Applications of 3D representations

Up to now feature-based three-dimensional mapping was used purely for visualization, which can be interesting for different applications, like in architecture, quality control, virtual reality, etc. In the future, 3D mapping will become increasingly important for mobile robot navigation, namely for localization and SLAM (Simultaneous Localization and Mapping).

This section presents some steps into the above mentioned direction, as the presented 3D mapping algorithm works on local data and outputs probabilistic features which are necessary for feature-based localization and mapping approaches. It shows that by incorporating probabilistic data through standard error propagation techniques the mapping process can be improved and a 3D model with associated uncertainty information can be obtained.

C. Algorithm

The algorithm presented in this section can be divided into three stages. The first stage filters the data to get rid of outliers and detect holes, the second stage performs the initial data segmentation using a specialized agglomerative hierarchical clustering approach, and the final stage consists of a simple probabilistic merging operation fusing planar regions with

similar model parameters. Furthermore, an outer loop to this algorithm enables incremental map building (see Figure 4).

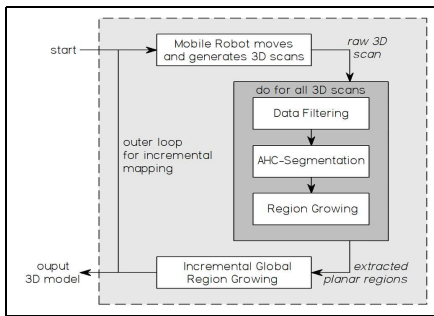


Fig. 4. A scheme describing the general structure of an incremental 3D mapping algorithm.

As a single scan of a laser scanner is a two-dimensional manifold, two consecutive non-located scans are regarded as a 3D scan in the following. Note that the presented algorithm does not require a special laser range scanner but works with every kind of 3D sensor.

1) *Data Filtering*: Outliers are filtered and a representation based on planar regions (a list of vertices lying approximately in a plane) is generated which is necessary for step 2.

If p_{ij} and p_{i+1j} are three-dimensional data points of the scan i , and p_{i+1j} and p_{i+1j+1} are data points of the scan $i + 1$, $Q := \{p_{ij}, p_{ij+1}, p_{i+1j+1}, p_{i+1j}\}$ forms an initial planar region with the edges $E := \{e_k = (\alpha, \beta) | e_1 = (p_{ij}, p_{i+1j}), e_2 = (p_{i+1j}, p_{i+1j+1}), e_3 = (p_{i+1j+1}, p_{ij+1}), e_4 = (p_{ij+1}, p_{ij})\}$ which in this case is a quad patch. To filter outliers and detect holes that may occur due to hallways or surfaces parallel to the laser beam, the quad Q is discarded under certain conditions. This can be achieved by defining constraints all edges e_k ($k = 1 \dots 4$) have to match by means of a χ^2 hypothesis test in order to accept quad Q for further processing. The Mahalanobis distance from an edge point $e_{k\alpha}$ with associated covariance matrix $\Sigma_{e_{k\alpha}}$ to the multivariate weighted mean (also known as the information filter) μ_{e_k} of the edge e_k is a possible choice and yields the following expressions (see Arras [10]):

$$\begin{aligned} \delta_{e_k} &= (e_{k\alpha} - \mu_{e_k})^T \Sigma_{e_k}^{-1} (e_{k\alpha} - \mu_{e_k}) \\ \Sigma_{e_k} &= (\Sigma_{e_{k\alpha}}^{-1} + \Sigma_{e_{k\beta}}^{-1})^{-1} \\ \mu_{e_k} &= \Sigma_{e_k} (\Sigma_{e_{k\alpha}}^{-1} e_{k\alpha} + \Sigma_{e_{k\beta}}^{-1} e_{k\beta}) \end{aligned}$$

The δ_{e_k} 's can be interpreted as χ^2 -variables with 3 degrees of freedom each. If any of the δ_{e_k} 's ($k = 1 \dots 4$) is larger than a selected maximum threshold value $\delta_{max} = \chi_{r,a}^2$ the quad Q is discarded with significance level a [11]. The resulting array Q of quads, for example, would look like $q_1, q_2, \dots, q_j, q_{j+2}, \dots, q_k, q_{k+2}, \dots, q_M$ (q_{j+1} and q_{k+1} have been discarded as they did not fulfill the above constraints).

2) *AHC-Segmentation*: Agglomerative Hierarchical Clustering (AHC) also called nearest neighbor filter is a standard clustering method used within the field of computer vision for a long time [13]. A specialized version taking into account the structure of the underlying data is used. All initial planar regions Q that passed the data filter constitute the

starting set of clusters $SC = \{sc_s | s = 1 \dots T\}$ defined as follows: Series of clusters sc_s , containing a contiguous array of initial regions are processed one after another. The first series to be processed is in this case $sc_1 = \{c_{1u} | c_{1u} \in \{(q_1, q_2), (q_2, q_3), (q_3, q_4), \dots, (q_{j-1}, q_j)\}\}$, the second one $sc_2 = \{c_{2u} | c_{2u} \in \{(q_{j+1}, q_{j+2}), \dots, (q_{k-1}, q_k)\}\}$, and the last $sc_3 = \{c_{3u} | c_{3u} \in \{(q_{k+1}, q_{k+2}), \dots, (q_{M-1}, q_M)\}\}$. It can be seen that the gaps detected in the filtering step delimit the series of clusters.

Every region q_j within a cluster c_{su} being part of a series of clusters sc_s has a maximum of two neighbors. This corresponds to the inherent topology of the input scan data in which every scan point has two direct neighbors⁷. In a standard AHC method on the other hand, all clusters are interconnected.

After having defined the initial clusters a best-fitting plane is found for every cluster. The average distance between the data points included in the regions of the cluster and the plane found is evaluated. The regions within the cluster with the smallest associated average distance are fused⁸. After updating the series of clusters, the next smallest distance is searched and the process repeats until a predefined minimum number of allowed regions is achieved. In this work this predefined minimum number of clusters is proportional to the size of the input series of clusters sc_s .

3) *Region Growing*: The clustering process ends when the minimal number of allowed clusters is reached. Since this number is set to be larger than the optimal value, an ensuing cleaning operation merges neighboring regions that meet certain criteria defined again by means of a χ^2 hypothesis test.

4) *Outer Loop*: When the planar regions approximating the data of two consecutive scans are found, another region growing mechanism converts the loose regions into global regions describing ceilings, walls, doors, etc.

D. Experimental Results

The office environment of the Autonomous Systems Lab was used for testing purposes. It is a highly structured environment with many planar objects like walls, closets, lockers and doors but it also contains many cylindrical steel beams close to the ceiling.

Several measurement missions were performed with the mobile robot generating 3D models (see for example Figure 6) of various sizes in different parts of the environment (see Table II). During the last mission which took place in the most complex part of the testing environment, 1083 scans were taken and still over 95% of the input points were represented by planes. All other missions took place in a less cluttered area which explains their higher percentage of represented points.

Concerning the quality of the extracted planes, a part of the environment was measured by hand and modelled in 3D to provide ground truth information. Figure 5 shows a visual comparison of the ground truth (left) and the reconstructed

⁷Neighboring scan points of a single measurement are scanned consecutively

⁸This can be implemented efficiently by using binary trees (heap sort)

# scans	# points	# planes	ratio [%]	time [s]	mem. [KB]
2	722	3	100	0.28	56.7
3	1083	3	99.72	0.64	58.8
10	3610	4	99.78	3.70	61.9
100	36100	40	99.08	56.52	222.4
344	124128	233	98.7	235.45	1158.8
1083	403598	1742	95.76	334.74	4512.2

TABLE II. Results of the algorithm produced by a standard Pentium IV with 1.7 GHz.

What is measured ?	ground truth	reconstructed model
height of the ceiling [m]	2.700	2.696
width of the corridor [m]	2.268	2.266
orientation ceiling (normal)	$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} -0.00098 \\ 0.00188 \\ -1.00000 \end{pmatrix}$
orientation lockers (normal)	$\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0.00447 \\ -1.00000 \\ 0.00106 \end{pmatrix}$

TABLE III. Quality of reconstructed map

3D model generated by the robot (right). Table III shows a comparison of real measures with the reconstructed model.



Fig. 5. Three-dimensional hand-made model of a part of a corridor at ASL representing ground truth (left). The same scene generated by the mobile robot measurement and 3D mapping algorithm (right). A superimposition of the two (center). Note that on the left side of the image a cupboard can be identified; on the right side, a group of lockers.

V. CONCLUSION

The first part of this paper presented a generic method for plane-fitting in an orthogonal least-square sense. The method fits well into the SPmodel framework, as it generates the parameters of the fitted plane as well as the propagated uncertainty information from the raw data.

The second section outlines an application for this plane-fitting method. It uses consecutive laser range scanner scans as 3D scans, filters the data probabilistically to overcome outliers and holes, performs a plane segmentation with a specialized AHC approach, and fuses matching planes together to find a compact 3D model.

It differs from related 3D mapping methods because it uses probabilistic information extensively. It has been shown that this not only improves the precision of the reconstructed model compared to the physical reality (see last section) but also attaches uncertainty information to the extracted features, therefore lending itself for feature-based robot navigation applications, which will be approached in future work.

REFERENCES

- [1] 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.
- [2] José A. Castellanos, Juan D. Tardós, "Mobile Robot Localization And Map Building", *Kluwer Academic Publishers*, 1999.
- [3] Dirk Hähnel, Wolfram Burgard, Sebastian Thrun, "Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot", *The fourth European workshop on advanced mobile robots (EUROBOT'01)*, 2001, pp. 91-98.
- [4] Yufeng Liu, Rosemary Emery, Deepayan Chakrabarti, Wolfram Burgard, Sebastian Thrun, "Using EM to Learn 3D Models of Indoor Environments with Mobile Robots", *Proceedings of the IEEE International Conference on Machine Learning (ICML)*, 2001, June 28 - July 1, 2001, pp. 329-336.
- [5] Sebastian Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hhnel, C. Rosenberg, N. Roy, J.Schulte, D. Schulz, "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva", *Journal of Robotics Research*, 2000.
- [6] Sebastian Thrun, "Robotic Mapping: A Survey", *Exploring Artificial Intelligence in the New Millennium*, Morgan Kaufmann, February 2002.
- [7] Sebastian Thrun, Wolfram Burgard, Dieter Fox, "A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping", *ICRA*, San Francisco, April 2000, Volume 1, pp. 321-328
- [8] K.O. Arras, N. Tomatis, B. Jensen, R. Siegwart, "Multisensor On-the-Fly Localization: Precision and Reliability for Applications", *Elsevier, Robotics and Autonomous Systems*, 2001, 34 (2-3), 131-143.
- [9] 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, Vol. 1056, pp. 1-13.
- [10] K.O. Arras, "Feature-Based Robot Navigation in Known and Unknown Environments", *PhD Thesis No. 2765*, EPFL, 2003.
- [11] K. Kanatani, "Statistical Optimization for geometric computation: theory and practice", *Elsevier Science*, 1996
- [12] Richard P. Paul, "Robot Manipulators: Mathematics Programming and Control", *MIT Press*, 1981.
- [13] Olivier Faugeras, "Three-Dimensional Computer Vision: Geometric Viewpoint", *MIT Press*, 1993.
- [14] Hans P. Moravec, "Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids", *CMU technical report*, CMU-RI-TR-96-34, 1996.
- [15] Hartmut Surmann, Andreas Nchter, and Joachim Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments", *Journal Robotics and Autonomous Systems*, 2003.
- [16] Luca Iocchi, Kurt Konolige, Max Bajracharya, "Visually Realistic Mapping of a Planar Environment with Stereo", *Proc. of Seventh International Symposium on Experimental Robotics (ISER)*, Hawaii, 2000, pp. 521-532.
- [17] Robert Lange, Peter Seitz, "Solid-State Time-of-Flight Range Camera", *IEEE Journal of quantum electronics*, 2001, Vol. 37, No. 3, pp. 390-397.
- [18] Caihua Wang, Hideki Tanahashi, Hidekazu Hirayu, Yoshonori Niwa, Kazuhiko Yamamoto, "A Probabilistic Approach to Plane Extraction and Polyhedral Approximation of Range Data", *IEICE Trans. Inf. & Syst.*, 2002, Vol. E85-D, No. 2, pp. 402-410.
- [19] Sebastian Thrun, Mark Diel, Dirk Hähnel, "Scan Alignment and 3-D Surface Modeling with a Helicopter Platform", *The 4th International Conference on Field and Service Robotics*, July 14-16, 2003.

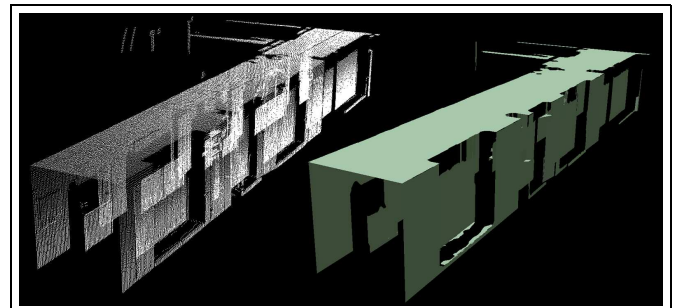


Fig. 6. The raw data (left) and the reconstructed model of a hallway representing 98 % of the 124128 input data points.