Simultaneous Localization and Mapping with Infinite Planes

Michael Kaess

***Abstract*— Simultaneous localization and mapping with infinite planes is attractive because of the reduced complexity with respect to both sparse point-based and dense volumetric methods. We show how to include infinite planes into a least-squares formulation for mapping, using a homogeneous plane parametrization with a corresponding minimal representation for the optimization. Because it is a minimal representation, it is suitable for use with Gauss-Newton, Powell's Dog Leg and incremental solvers such as iSAM. We also introduce a relative plane formulation that improves convergence. We evaluate our proposed approach on simulated data to show its advantages over alternative solutions. We also introduce a simple mapping system and present experimental results, showing real-time mapping of select indoor environments with a hand-held RGB-D sensor.**

I. Introduction

The recent popularity of real-time 3D simultaneous localization and mapping (SLAM) is explained by its wide range of possible applications including mobile robot navigation, surveying and inspection, augmented and virtual reality, search and rescue, and reconnaissance. Laser-range finders are commonly used for 3D mapping, but the high cost of a 3D laser or actuated 2D laser limit their wide-spread adoption. With the recent availability of cheap camera-based RGB-D sensors, 3D mapping capabilities are now becoming more widely accessible. But many mapping systems place high demands on processing and hence also power requirements, negating the advantages in both mobility and price of mobile RGB-D sensors.

When assuming the presence of structure in the surroundings, cheaper solutions are possible by making use of higher level features, such as planes, rather than points or dense volumetric representations. While there has been prior work on using planar features for mapping, discussed in more detail below, a least-squares estimation with planar features faces some challenges that have so far not been addressed in the SLAM literature.

A specific challenge for nonlinear least-squares estimation is presented by the overparametrized representation of an infinite plane. While a Gauss-Newton solver is preferable to other iterative least-squares solvers because of its fast convergence, the overparametrization leads to a rank-deficient information matrix, causing matrix inversion to fail. Commonly used in practice, the Levenberg-Marquardt algorithm adds a regularization term, but is not suitable for incremental inference [13, 14]. Alternatively, a Lagrangian multiplier can eliminate the extra degree of freedom, but at the cost of additional computation and a 67% larger state space compared to the minimal representation. And finally, a commonly used minimal representation using spherical coordinates for the plane normal suffers from singularities not unlike the well-known gimbal lock for Euler angles.

In this paper we introduce a minimal parametrization that avoids all of these problems and has so far not been described in the context of SLAM with planar features. We will relate the overparametrization of an infinite plane to that of a rotation represented by a quaternion and provide a solution that is also related to our earlier work on a minimal parametrization for homogeneous point representations [12].

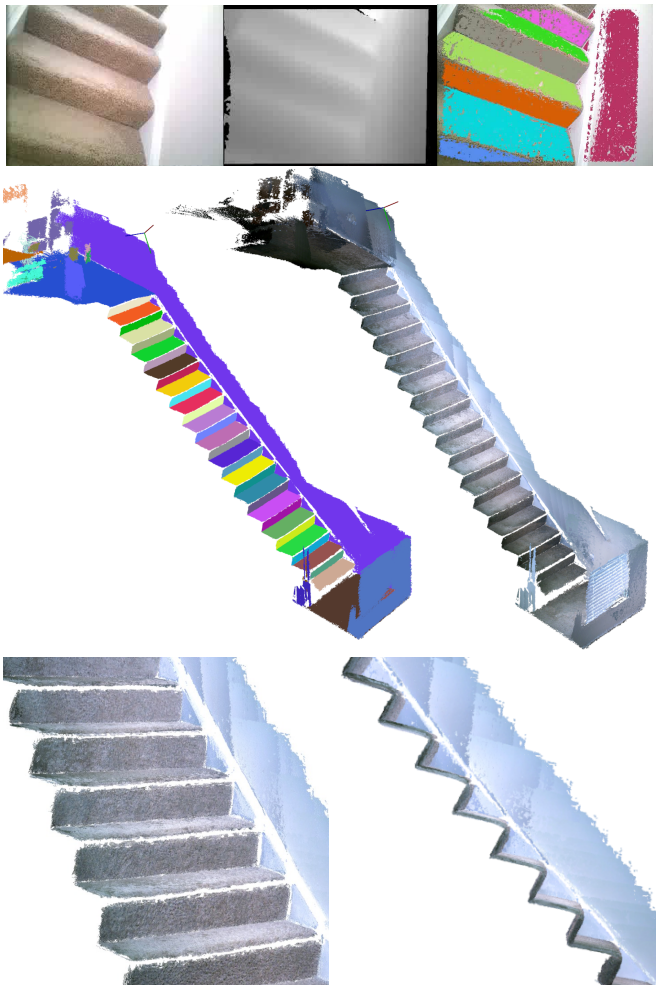


Fig. 1: The stairs sequence. (top) Single frame, from left to right: raw image, depth image and planar segmentation. (center) 3D model, showing unique planes on the left and the colored point cloud on the right. (bottom) Enlarged view and side view of the 3D model.

Michael Kaess is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. kaess@cmu.edu

This work was partially supported by the Office of Naval Research under awards N00014-12-1-0093 and N00014-14-1-0373. Partial support was also provided by the American Bureau of Shipping.

Our contributions in this paper are three-fold:

1) We introduce a minimal parametrization of infinite planes suitable for least-squares estimation with Gauss-Newton methods and related incremental solvers.
2) We present a relative plane formulation that improves convergence.
3) We show experimental results for mapping with planes using a hand-held RGB-D sensor (see Fig. 1).

In the following section we discuss related work. In section III we describe our approach to mapping with infinite planes, with a focus on the minimal representation for least-squares optimization. In section IV we provide an evaluation using simulated data, followed in section V by experimental results from a real-time system using a hand-held RGB-D sensor. We summarize and offer concluding remarks in section VI.

II. Related Work

We perform simultaneous localization and mapping (SLAM) by explicitly including infinite planes into the estimation formulation. One of the earliest works that makes planes explicit in the estimation is presented by Weingarten and Siegwart [19]. They use 3D laser data and odometry derived from a 2D laser in combination with an EKF formulation. The EKF formulation is also used by Servant et al. [16], starting with partial knowledge and tracking a monocular camera, and by Gee et al. [2] and Martinez-Carranza and Calway [10] for monocular mapping. The main problem of using an EKF formulation is the computational cost caused by maintaining the dense covariance matrix, which limits application to a small number of planes. Instead, we apply a graph-based smoothing formulation that yields a sparse estimation problem that can efficiently be solved.

Lee et al. [9] use a graph formulation in combination with a spherical parametrization of the planes. The spherical parametrization is commonly used [19, 16] and is attractive because it is a minimal representation. However, the spherical parametrization suffers from singularities that are similar in nature to the well-known gimbal lock for Euler angles.

Another common approach uses four parameters to represent a plane by its normal and distance to the origin. Trevor et al. [18] uses this overparametrized representation for a smoothing solution that combines data from an RGB-D sensor and a 2D laser range finder. The overparametrized formulation is also used by Taguchi et al. [17] for real-time mapping with a hand-held RGB-D sensor, combining both mapping of points and planes. However, for an over-parametrized system the information matrix is singular and any attempt to invert it within Gauss-Newton will fail. Hence Levenberg-Marquardt is typically used instead, which adds a regularization term and thereby hides the overparametrization, negatively affecting convergence speed. In contrast, in this paper we describe a minimal parametrization for the optimization that is not affected by singularities.

In recent work, Salas-Moreno et al. [15] introduced a dense planar SLAM system that generates a surfel map and identifies and grows planar regions over time, with applications to augmented reality. Similar work by Whelan et al. [21] incrementally performs planar simplification of a triangle mesh that is built using a dense volumetric approach. In contrast, we avoid dense mapping and explicitly represent and optimize a graph of infinite planes.

III. Mapping with Infinite Planes

We start by introducing the state and plane representation as well as basic operations related to those states. For over-parametrized quantities we then discuss minimal representations. Finally, we discuss the formulation of plane SLAM as a least-squares optimization problem with a suitable plane observation model, and show how to solve this estimation problem.

A. State and Plane Representation

We represent the location and attitude of the sensor, or short its pose, with respect to some coordinate frame w by the tuple

$$x = (\mathbf{t}, \mathbf{q}) \in \mathbb{R}^3 \times S^3, \tag{1}$$

where $\mathbf{t}$ is the translation and $\mathbf{q}$ is the spatial orientation of the sensor represented by a unit quaternion. A unit quaternion

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \mathbf{v}\sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{pmatrix} = \begin{pmatrix} \mathbf{q}_v \\ q_w \end{pmatrix} \in S^3, \;\; \|\mathbf{q}\| = 1 \tag{2}$$

represents a rotation around the vector $\mathbf{v}$ through angle θ. Whenever needed, we convert pose x into the Euclidean transformation matrix

$$\mathrm{T}_{wx} = \begin{pmatrix} \mathrm{R}(\mathbf{q}) & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4}, \tag{3}$$

where for quaternion $\mathbf{q}$ the expression $\mathrm{R}(\mathbf{q}) \in SO(3)$ is the rotation matrix

$$\mathrm{R}(\mathbf{q}) = \begin{bmatrix} -q_1^2+q_2^2+q_3^2-q_4^2 & 2(q_3q_4-q_1q_2) & 2(q_2q_4+q_1q_3) \\ 2(q_1q_2+q_3q_4) & -q_1^2+q_2^2-q_3^2+q_4^2 & 2(q_1q_4-q_2q_3) \\ 2(q_1q_3-q_2q_4) & 2(q_2q_3+q_1q_4) & q_1^2+q_2^2-q_3^2-q_4^2 \end{bmatrix} \tag{4}$$

A point in projective space is represented by homogeneous coordinates $\mathbf{p} = (p_1, p_2, p_3, p_4)^\top \in \mathbb{P}^3$, where the corresponding Euclidean point for $p_4 \neq 0$ is $(p_1/p_4, p_2/p_4, p_3/p_4)^\top \in \mathbb{R}^3$. A point $\mathbf{p}_x$ in the local sensor frame is moved into the world frame by left multiplication with the Euclidean transformation matrix

$$\mathbf{p}_w = \mathrm{T}_{wx}\mathbf{p}_x. \tag{5}$$

Conversely, a world point is moved into the local frame $\mathbf{p}_x = \mathrm{T}_{wx}^{-1}\mathbf{p}_w$ using the inverse transformation.

We represent a plane as a homogeneous vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^\top \in \mathbb{P}^3$ in projective space. (Hartley and Zisserman [5] provide an excellent introduction to projective geometry and the homogeneous representation of planes used in this work.) A point $\mathbf{p} = (p_1, p_2, p_3, p_4)^\top \in \mathbb{P}^3$ lies on the plane iff

$$\pi_1p_1 + \pi_2p_2 + \pi_3p_3 + \pi_4p_4 = 0, \tag{6}$$

or short $\boldsymbol{\pi}^\top \mathbf{p} = 0$. Mapping into $\mathbb{R}^3$ for $x_4 \neq 0$ yields the more familiar plane equation $\pi_1 x + \pi_2 y + \pi_3 z + \pi_4 = 0$, or

$$\mathbf{n}^\top \mathbf{x} = d, \tag{7}$$

where $\mathbf{x} = (x, y, z)^\top = (x_1/x_4, x_2/x_4, x_3/x_4)^\top$. After normalization, the first three components

$$\mathbf{n} = \frac{(\pi_1, \pi_2, \pi_3)^\top}{\sqrt{\pi_1^2 + \pi_2^2 + \pi_3^2}} \tag{8}$$

form the normal vector of the plane, and the remaining component

$$d = \frac{-\pi_4}{\sqrt{\pi_1^2 + \pi_2^2 + \pi_3^2}} \tag{9}$$

its distance from the origin. The homogeneous plane representation is transformed from local to world frame according to the *inverse transpose* of the corresponding point transform

$$\boldsymbol{\pi}_w = \mathrm{T}_{wx}^{-\top} \boldsymbol{\pi}_x. \tag{10}$$

B. Minimal Parametrization

The homogeneous plane representation is overparametrized and therefore requires some special care during optimization—we will apply a similar solution as is commonly used for quaternions, and therefore start with a review of the quaternion case.

The unit sphere $S^3 = \{\mathbf{q} \in \mathbb{R}^4 : \|\mathbf{q}\| = 1\}$ can be identified with the set of unit quaternions, which form a 3-dimensional Lie group under quaternion multiplication (Hamilton product)

$$\mathbf{q}\mathbf{q}' = \begin{bmatrix} q_1 q_4' + q_2 q_3' - q_3 q_2' + q_4 q_1' \\ -q_1 q_3' + q_2 q_4' + q_3 q_1' + q_4 q_2' \\ q_1 q_2' - q_2 q_1' + q_3 q_4' + q_4 q_3' \\ -q_1 q_1' - q_2 q_2' - q_3 q_3' + q_4 q_4' \end{bmatrix}. \tag{11}$$

They are related to the set of 3D rotation matrices $SO(3)$ as follows: There is a two-to-one covering map from S^3 onto $SO(3)$: antipodal points in S^3 are identified because $-\mathbf{q}$ represents the same rotation as $\mathbf{q}$. The matrix Lie algebra of $SO(3)$ is $\mathfrak{so}(3)$, the set of skew-symmetric matrices

$$[\boldsymbol{\omega}]_\times = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{12}$$

for an incremental rotation $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^\top$, see Hall [4]. This Lie algebra forms a tangent space $\mathbb{R}^3$ of $SO(3)$ at the identity. Because the elements have three parameters, they provide a minimal local parametrization of rotations. The *exponential map*

$$\exp([\boldsymbol{\omega}]_\times) = \mathrm{I} + \sin\|\boldsymbol{\omega}\| [\hat{\boldsymbol{\omega}}]_\times + (1 - \cos\|\boldsymbol{\omega}\|) \left(\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}^\top - \mathrm{I}\right) \tag{13}$$

is also known as Rodrigues' formula, where $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} / \|\boldsymbol{\omega}\|$ and $\|\boldsymbol{\omega}\|$ are the axis and angle of rotation, respectively. The exponential map allows updating an existing rotation R by an increment $\boldsymbol{\omega}$

$$\mathrm{R}' = \exp([\boldsymbol{\omega}]_\times)\mathrm{R}. \tag{14}$$

Similarly, the elements of the Lie algebra $\mathfrak{su}(2)$ of S^3 can be identified with the tangent space $\mathbb{R}^3$ of S^3 at the identity. We use the mapping of $\mathbb{R}^3$ to S^3 from Grassia [3]

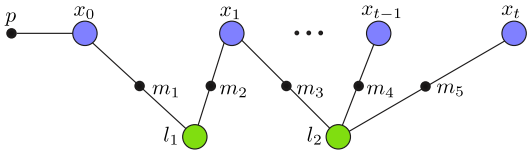
$$\exp(\boldsymbol{\omega}) = \begin{pmatrix} sinc\left(\frac{1}{2}\|\boldsymbol{\omega}\|\right)\boldsymbol{\omega} \\ \cos\left(\frac{1}{2}\|\boldsymbol{\omega}\|\right) \end{pmatrix} \tag{15}$$

that coincides with the axis/angle representation above. Here, $sinc$ is the cardinal sine function. Note that $\exp((0,0,0)^\top) = (0,0,0,1)^\top$ is the identity of S^3 as expected. As for every minimal representation of rotations there are singularities, here at multiples of 2π, but, unlike for Euler angles, they can be avoided by forcing $\boldsymbol{\omega}$ to fall into the range $(-\pi, \pi]$, while still allowing for all possible rotations. An existing quaternion $\mathbf{q}$ is updated by an increment $\boldsymbol{\omega}$ using quaternion multiplication

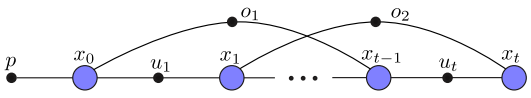
$$\mathbf{q}' = \exp(\boldsymbol{\omega})\,\mathbf{q}. \tag{16}$$

Now we are ready to take on the overparametrization in the plane representation. Note that both homogeneous and non-homogeneous (normal and distance) representations are overparametrized, because there are only three degrees of freedom in a plane: its orientation, which can be specified by two angles α and β, and its orthogonal distance d from the origin. But why not just use these three parameters (α, β, d) as a minimal representation? They are indeed sufficient to specify any plane, but there are singularities, not unlike those for Euler angles, that cause problems when encountered during optimization. And why is the overparametrization problematic? The information matrix becomes rank-deficient and cannot be inverted as needed for Gauss-Newton type optimization. One solution is regularization, as provided by the Levenberg-Marquardt algorithm. However, slow convergence can be expected because there are directions in which the cost function is flat so that an infinite number of solutions exist, forcing the algorithm toward linear convergence speed of the gradient descent method instead of the quadratic convergence of Gauss-Newton. Furthermore, overparametrization does not allow the use of trust-region methods such as Powell's Dog Leg or incremental inference with methods such as incremental smoothing and mapping (iSAM) [7, 13].

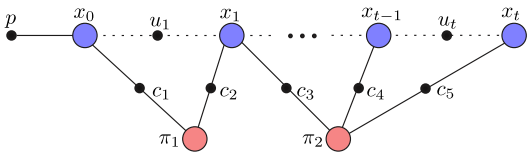
We can find a minimal representation by restricting the ambiguity in the homogeneous representation. The homogeneous plane representation is not unique: all $\lambda\boldsymbol{\pi}$ represent the same plane for $\lambda \in \mathbb{R}\backslash 0$, which defines an equivalence class. We identify unique representatives by normalizing the vector $\boldsymbol{\pi}$ to lie on the unit sphere of $\mathbb{R}^4$ as $\boldsymbol{\pi}' = \boldsymbol{\pi} / \|\boldsymbol{\pi}\| \in S^3$. The vector $(0,0,0,1)^\top$ represents the north pole of the unit sphere (think of a sphere in $\mathbb{R}^3$) and corresponds to the plane at infinity [5]. The vector $(1,0,0,0)^\top$ is a point on the equator that represents a plane through the origin. In fact, $(a,b,c,0)^\top$ contains all the planes through the origin, where $(a,b,c)^\top$ describes the normal vector. Leaving the equator, for increasing d we obtain planes that are further from the origin with distance given by 9. Equivalently, negative d lead to planes with the normal pointing away from the origin—remembering that S^3 is a double cover of $SO(3)$, an equivalent plane is found by negating both the normal and the



(a) Structure-from-motion (SFM)



(b) Pose graph SLAM



(c) Planar SLAM

Fig. 2: Factor graph representation for three SLAM variants. Variable nodes include poses x, point features l and plane features π. Factor nodes relate to point measurements m, plane measurements c, odometry measurements u, loop closures o, and the prior p on the first pose. While plane constraints between pairwise frames could be converted into a pose graph formulation, for planar SLAM, we choose to explicitly model the infinite planes, similar to the point features in structure from motion. Additional sensor information such as odometry can be added similar to the common pose graph formulation of SLAM. Note that at least three general planes have to be observed per pose for (c) to be fully constrained, and similarly the example in (a) is underconstrained as shown.

distance. Therefore we can further reduce the representation by restricting $\boldsymbol{\pi}'$ to lie on the upper hemisphere, i.e. $d \geq 0$. There is still some ambiguity left, as for the unit circle/sphere at $d = 0$ opposite sites are also identified, but we can safely ignore that here.

How do we restrict the optimization to remain on the unit sphere? One option is to add a Lagrangian multiplier to enforce unit norm, adding an additional parameter to the optimization as well as a nontrivial cost term. Instead of the minimum of three parameters, we now need five parameters for each plane, considerably increasing the size of the state space, while also adding to the computational cost with the additional constraint. A better solution is to use a minimal representation to update a plane *during* optimization. Recalling the unit quaternion case discussed earlier for rotations, we realize that both can be identified with S^3. We will simply treat the normalized homogeneous plane parametrization as a quaternion, and use the exponential map to update planes during optimization.

We will later also need the inverse of the exponential map, the so-called *log map*, that maps an element of S^3 referenced to the identity $(0, 0, 0, 1)^\top$ into the tangent space $\mathbb{R}^3$

$$\omega = \log(\mathbf{q}) = \frac{2\cos^{-1}(q_w)}{\|\mathbf{q}_v\|}\mathbf{q}_v, \tag{17}$$

which is useful for measuring the distance $\log(\mathbf{q}(\boldsymbol{\pi})^{-1}\mathbf{q}(\boldsymbol{\pi}'))$ between two planes $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ in the tangent space.

C. SLAM Formulation

We formulate the planar mapping problem as a least-squares optimization, estimating the sensor poses $x_0, \ldots, x_t$ and planes $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_m$ given the plane measurements.

We use a *factor graph* to represent the estimation problem as a graphical model [8]. Factor graphs of three SLAM variants are shown in Fig. 2. Planar SLAM is similar to structure from motion, but instead of point locations we estimate the parameters of infinite planes. And similar to the pose graph SLAM formulation, we may include any available odometry constraints, e.g. from a robot platform or from inertial sensors. It is also possible to convert the planar constraints between frames into constraints within a pose graph formulation. However, explicit modeling is preferable for planar mapping and is also feasible in real-time because the number of planes visible per image is typically small.

Formally, a factor graph is a bipartite graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ with two node types: *factor nodes* $f_i \in \mathcal{F}$ that relate to our measurements and *variable nodes* $\theta_j \in \Theta$ that represent the poses and planes to be estimated. An edge $e_{ij} \in \mathcal{E}$ connects a factor node with a variable node. A factor graph G defines the factorization of a function

$$f(\Theta) = \prod_i f_i(\Theta_i), \tag{18}$$

where dependence relationships are encoded by the edges e_{ij}: each factor f_i is a function of the variables in Θ_i. Our goal is to find the variable assignment Θ^* that maximizes the factorization (18)

$$\Theta^* = \arg\max_\Theta \prod_i f_i(\Theta_i), \tag{19}$$

where $\Theta_i = \{\theta_j\}$ is the set of variables θ_j involved in the factor f_i. For Gaussian measurement models

$$f_i(\Theta_i) \propto \exp\left(-\frac{1}{2}\left\|h_i(\Theta_i) - z_i\right\|^2_{\Sigma_i}\right), \tag{20}$$

the factored objective function to maximize (19) is equivalent to the nonlinear least-squares problem

$$\arg\min_\Theta \left(-\log f(\Theta)\right) = \arg\min_\Theta \frac{1}{2}\sum_i \left\|h_i(\Theta_i) - z_i\right\|^2_{\Sigma_i}, \tag{21}$$

where $h_i(\Theta_i)$ is a measurement prediction function and z_i a measurement, and $\|\mathbf{y}\|^2_\Sigma := \mathbf{y}^T\Sigma^{-1}\mathbf{y}$ is the squared Mahalanobis distance with covariance matrix Σ.

D. Plane Measurement Model

A measurement of a plane is given by its normal and distance from the local sensor frame, and an associated measurement uncertainty. We model the uncertainty of the measurement of the plane $\boldsymbol{\pi}$ from pose x

$$\boldsymbol{\pi}_x = \mathrm{T}_{wx}^{-\top}\boldsymbol{\pi} + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}(0, \Sigma) \tag{22}$$

by zero-mean Gaussian noise $\boldsymbol{v}$ with covariance Σ. Then, the probability of a plane estimate $\hat{\boldsymbol{\pi}}$ and a pose estimate

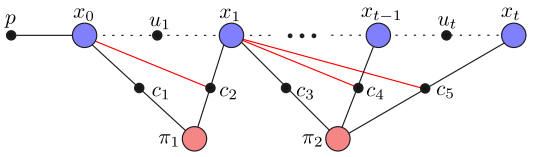


Fig. 3: Relative formulation for plane measurements, based on Fig. 2c. Planes are expressed in the local coordinate frame of a base pose, for example the first pose from which they have been observed. Subsequent plane observations result in ternary factors that also connect to the respective base pose (new connections shown in red).

$\hat{x}$ given an actual measurement $\tilde{\boldsymbol{\pi}}_x$ is given by the normal distribution

$$p(\hat{x}, \hat{\boldsymbol{\pi}} | \tilde{\boldsymbol{\pi}}_x) = \frac{1}{\sqrt{(2\pi)^3 |\Sigma|}} \exp\left(-\frac{1}{2} \left\| h(\mathrm{T}_{w\hat{x}}, \hat{\boldsymbol{\pi}}) \ominus \tilde{\boldsymbol{\pi}}_x \right\|_\Sigma^2\right). \tag{23}$$

The operator $\ominus$ calculates the difference in the tangent space by applying the log map. The corresponding cost function for the least-squares formulation is simply the quadratic term from the exponent

$$\begin{aligned} c_{x\pi}(\hat{x}, \hat{\boldsymbol{\pi}}) &= \left\| h(\mathrm{T}_{w\hat{x}}, \hat{\boldsymbol{\pi}}) \ominus \tilde{\boldsymbol{\pi}}_x \right\|_\Sigma^2 \\ &= \left\| \log\left(q(\mathrm{T}_{w\hat{x}}^{-\top} \hat{\boldsymbol{\pi}})^{-1} q(\tilde{\boldsymbol{\pi}}_x)\right) \right\|_\Sigma^2. \end{aligned} \tag{24}$$

Taking the difference in tangent space, we treat planes as quaternions, calculate the difference by quaternion multiplication with one side inverted, and finally apply the log map.

We propose a relative formulation for plane measurements. Planes are expressed relative to the first pose that observed them, changing the cost function (24) to first express the observing pose in the frame of the base pose

$$c_{x\pi x_b}(\hat{x}, \hat{\boldsymbol{\pi}}, \hat{x}_b) = \left\| h(\mathrm{T}_{w\hat{x}_b}^{-1} \mathrm{T}_{w\hat{x}}, \hat{\boldsymbol{\pi}}) \ominus \tilde{\boldsymbol{\pi}}_x \right\|_\Sigma^2 \tag{25}$$

In the factor graph, this results in ternary factors that additionally connect to the base pose, see Fig. 3. Similar to our relative formulation for homogeneous point features in McDonald et al. [12], this allows for faster convergence. Intuitively, when mapping a large building, loop closures might cause a complete room to move, but the relative configuration of planes and poses inside the room will likely not be affected. Keeping planes anchored relative to poses automatically moves the planes along with any global changes that affect the poses.

Extraction of plane measurements from point data is described by Weingarten and Siegwart [19]. Förstner [1] provides a theoretical account for how the probabilistic uncertainty of homogeneous entities (including planes) is correctly dealt with in the context of least squares estimation.

E. Solution

Solving the nonlinear least-squares problem typically involves repeated linearization. For nonlinear measurement functions h_i in (20), nonlinear optimization methods such as Gauss-Newton iterations or the Levenberg-Marquardt algorithm solve a succession of linear approximations to (21) in order to approach the minimum. At each iteration of the nonlinear solver, we linearize around the current estimate Θ to get a new, *linear* least-squares problem in $\boldsymbol{\Delta}$

$$\arg\min_{\boldsymbol{\Delta}} \left(-\log f(\boldsymbol{\Delta})\right) = \arg\min_{\boldsymbol{\Delta}} \left\| A\boldsymbol{\Delta} - \mathbf{b} \right\|^2, \tag{26}$$

where $A \in \mathbb{R}^{m \times n}$ is the measurement Jacobian consisting of m measurement rows, and $\boldsymbol{\Delta}$ is an n-dimensional vector. Note that the covariances Σ_i have been absorbed into the corresponding block rows of A, making use of

$$\|\boldsymbol{\Delta}\|_\Sigma^2 = \boldsymbol{\Delta}^T \Sigma^{-1} \boldsymbol{\Delta} = \boldsymbol{\Delta}^T \Sigma^{-\frac{T}{2}} \Sigma^{-\frac{1}{2}} \boldsymbol{\Delta} = \left\| \Sigma^{-\frac{1}{2}} \boldsymbol{\Delta} \right\|^2. \tag{27}$$

Once $\boldsymbol{\Delta}$ is found, the new estimate is given by $\Theta \oplus \boldsymbol{\Delta}$, which is then used as linearization point in the next iteration of the nonlinear optimization. The operator $\oplus$ is often simple addition, but for overparametrized representations such as quaternions for 3D orientations or our homogeneous plane representations, an exponential map is used instead.

The minimum of the linear system $A\boldsymbol{\Delta} - \mathbf{b}$ is obtained by Cholesky or QR matrix factorization. By setting the derivative in $\boldsymbol{\Delta}$ to zero we obtain the normal equations $A^T A \boldsymbol{\Delta} = A^T \mathbf{b}$. Cholesky factorization yields $A^T A = R^T R$, and a forward and backsubstitution on $R^T \mathbf{y} = A^T \mathbf{b}$ and $R\boldsymbol{\Delta} = \mathbf{y}$ first recovers $\mathbf{y}$, then the actual solution, the update $\boldsymbol{\Delta}$. Alternatively we can skip the normal equations and apply QR factorization directly to A, yielding $R\boldsymbol{\Delta} = \mathbf{d}$, which is solved by backsubstitution. Note that Q is not explicitly formed; instead $\mathbf{b}$ is modified during factorization to obtain $\mathbf{d}$, see [7] for details.

When considering a sequence of measurements, as is typically the case for SLAM, it is possible to obtain a more efficient solution by updating the previous solution with the new measurements. iSAM [7] appends the new linearized measurement rows to the previous square root information matrix R and applies Givens rotations to again form an upper triangular matrix. A fully nonlinear incremental algorithm is also possible by exploiting the connection between sparse linear algebra and probabilistic graphical models, see iSAM2 [8] for details.

TABLE I: Batch optimization, comparing relative and absolute formulation as well as overparametrization and our minimal parametrization for a simulated sequence (76 poses, 31 planes, 450 plane measurements).

	Gauss-Newton	Levenberg-Marquardt	Powell's Dog-Leg
Overpar., absolute	not possible	76 it (475ms)	not possible
Overpar., relative	not possible	15 it (208ms)	not possible
Minimal, absolute	diverged	58 it (340ms)	17 it (126ms)
Minimal, relative	**5 it** (105ms)	**5 it** (106ms)	**7 it** (129ms)

IV. Evaluation

First we show the advantages of our minimal parametrization as well as the relative plane formulation. We use a simulated dataset of 76 poses distributed in $1m$ steps along a straight line with 31 random planes yielding 450 plane measurements based on a maximum sensor range of $5m$. We simulate the measurements from the ground truth and add Gaussian noise with standard deviations of $0.1m$ in

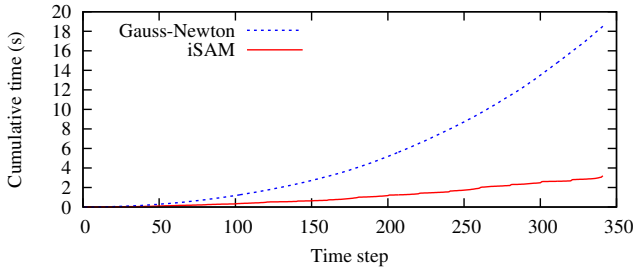


Fig. 4: Cumulative computation time for Gauss-Newton compared to incremental solution by iSAM, using our minimal parametrization and relative plane formulation for a simulated sequence (343 poses, 28 planes, 2736 plane measurements).

translation, $0.01rad$ in orientation and 0.005 on the plane measurements. In Table I we compare batch optimization using Gauss-Newton (GN), Levenberg-Marquardt (LM) and Powell's Dog-Leg (PDL). Neither GN nor PDL can handle overparametrization, because the information matrix is rank deficient and cannot be inverted. Intuitively, the cost function locally is not bowl shaped with a single minimum, but resembles a valley of infinitely many minima. While LM does find the correct solution, its regularization masks the underlying problem at the cost of convergence speed. In contrast, the minimal representation also allows use of GN and PDL. Generally, PDL has a slightly higher cost, but is more stable, as can be seen by comparing with the second to last row, where GN diverges. In summary, our minimal representation combined with the relative formulation outperforms the alternatives.

Second, we demonstrate the performance advantage of incremental optimization enabled by our minimal representation. In SLAM, measurements arrive sequentially, and at every time step a solution is calculated. Exploiting this sequential nature, an incremental solver such as iSAM [7] reuses previous calculations to significantly lower computational cost compared to performing a batch solution at every step. Using the same sensor range and artificial noise as in the previous experiment, we have simulated a longer random robot trajectory through a Manhattan world with random planes. Fig. 4 compares per step cumulative timing for batch Gauss-Newton and the incremental solver iSAM. The results indicate that incremental optimization is possible with our minimal representation, which is not true in the overparametrized case. The plot further confirms that incremental optimization significantly reduces computational cost, which is consistent with our prior results [7] for landmark-based and pose graph SLAM.

V. Experimental Results

We present experimental results with a hand-held RGB-D sensor. We use the ASUS Xtion Pro Live sensor at 640×480 resolution. Experiments are run on a laptop computer with i7-3920XM 2.9GHz CPU. No GPU is used. The implementation is multi-threaded, with separate threads for plane detection, graph optimization, and visualization. The system runs at 15 frames per second.

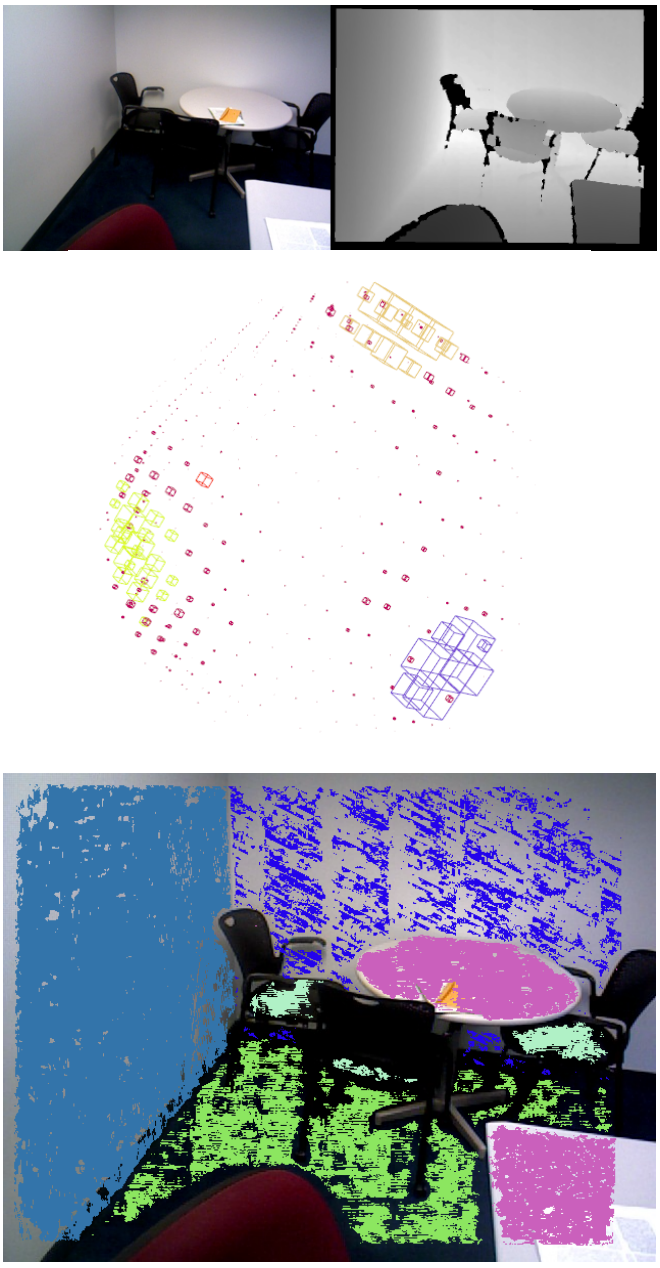


Fig. 5: Plane segmentation. (top) Raw image and depth image. (center) Voxelization of normal vectors on the unit sphere with clusters indicated by different colors. (bottom) Final plane segmentation after additional clustering for each normal direction along distance from origin.

Plane segmentation follows Holz et al. [6] with some modifications. The top row in Fig. 5 shows the raw sensor data (color and depth) for a single frame. We only use depth information for the segmentation. We start by finding surface normals for each available depth measurement by smoothing over neighboring measurements. We use the PCL 1.7 implementation of IntegralImageNormalEstimation with depthChangeFactor=0.01, smoothingSize=20 and depth dependent smoothing enabled. We then cluster the points in normal space (surface of the unit sphere S^2 in $\mathbb{R}^3$) by computing a three dimensional voxel grid (cube of side length 20) and merging neighboring voxels that exceed a

TABLE II: Statistics and per frame timing for the stairs sequence in Fig. 1.

Length of sequence	$58s$
Number of planes	81
Number of poses	868
Number of plane observations	$5,934$
Time for normal computation	$34ms$
Time for plane segmentation	$33ms$
Time for data association	$< 1ms$
Time for graph optimization	$24ms$

threshold (1500) for the number of binned normals. For each cluster, we then perform clustering in depth by discretizing in one dimension ($0.05m$ steps) and merging of neighboring cells that exceed a second threshold (1000). Finally, we fit planes to each cluster of points using the eigenvector corresponding to the smallest eigenvalue as normal direction and filter out points that lie more than $0.02m$ from the plane.

We use a simple approach to data association. The optimized previous pose is used to initialize the new pose. Planes detected in the current frame are matched against all planes in the map, selecting the best match for each plane within a given threshold of both angle ($8°$) and distance ($0.1m$). While not sufficient for general mapping applications, this approach is suitable for simple environments such as those presented here.

Results from walking up a flight of stairs with the RGB-D sensor are shown in Fig. 1. For a single frame, the top row shows the raw color and depth image as well as the detected planes, where each plane is assigned a random color. The center row shows the final 3D model of the stairs, both as randomly colored planes and as point cloud with color taken from the input frames. In the bottom row, the closeups of the 3D model from front and side show the quality of the reconstruction. Statistics and timing for this sequence are provided in Table II. Parallelization allows running at 15 frames per second with about 1.5 frames of lag.

Results from an office environment are shown in Fig. 6. Even though the floor has been observed in two disconnected components left and right of the table, data association has combined those into a single infinite plane. Similarly, the two disconnected components of the ceiling have been merged.

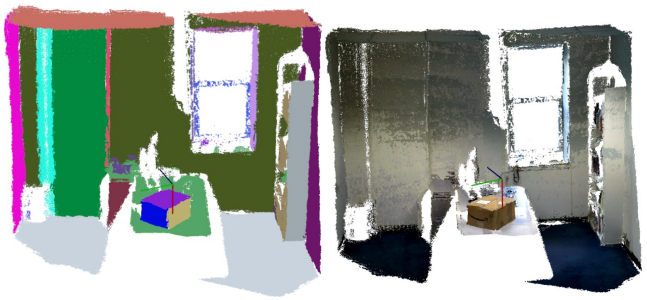


Fig. 6: Office sequence. (left) The 3D model with a random color assigned to each plane. (right) The 3D model with color from the input images.

VI. Conclusion

We have presented a novel minimal representation for planar features that, unlike previous formulations, is suitable for least-squares optimization with Gauss-Newton, Powell's Dog-Leg and iSAM. Using simulated data we have demonstrated that the minimal representation allows for a faster and more stable optimization. We have also introduced a relative formulation for planes and shown that it improves convergence properties. We have presented real-time experimental results for a simple planar 3D mapping system using a handheld RGB-D camera.

Our planar mapping system is suitable to demonstrate the planar mapping concept but can be improved in many aspects. Plane detection could be sped up by prediction, or, to also work at lower frame rates, integration of inertial measurements. Data association will need to take into account the boundaries of the physical planar surfaces to successfully work over larger environments. And a keyframe approach could further reduce computational cost. While the plane optimization provides an efficient formulation of the estimation problem, its expressiveness in terms of dense 3D models is limited when compared to more general volumetric solutions [20]. A promising approach may be to locally use TSDF methods or surfels, anchored relative to the planes, so that they automatically adapt to any large-scale adjustments of the plane graph.

While presented in isolation here, the planar features can be combined with other geometric entities to handle frames with insufficient planar constraints. Examples include point features using a similar homogeneous parametrization introduced in our prior work [11, 12], or line features using Plücker coordinates. Beyond homogeneous 4-vectors, finding a minimal representation for objects of other dimensions can be generalized even in absence of a suitable Lie algebra, as described by Förstner [1]. And even for the homogeneous 4-vector, there are alternative minimal representations, for example based on Householder matrices as described for general homogeneous entities in the appendix of [5].

References

[1] W. Förstner. Minimal representations for uncertainty and estimation in projective spaces. In *Asian Conf. on Computer Vision (ACCV)*, pages 619–632, 2010.

[2] A.P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering higher level structure in visual SLAM. *IEEE Trans. Robotics*, 24(5): 980–990, 2008.

[3] F.S. Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3:29–48, March 1998. ISSN 1086-7651.

[4] B.C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2000.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

[6] D. Holz, S. Holzer, R.B. Rusu, and S. Behnke. Real-time plane segmentation using RGB-D cameras. In *RoboCup 2011: Robot Soccer World Cup XV*, volume 7416 of *Lecture Notes in Computer Science*, pages 306–317. Springer, 2012.

[7] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, December 2008.

[8] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31:217–236, February 2012.

[9] T. Lee, S. Lim, S. Lee, S. An, and S. Oh. Indoor mapping using planes extracted from noisy RGB-D sensors. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1727–1733, 2012.

[10] J. Martinez-Carranza and A. Calway. Unifying planar and point mapping in monocular SLAM. In *British Machine Vision Conf. (BMVC)*, Aberystwyth, Wales, August 2010.

[11] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J.J. Leonard. 6-DOF multi-session visual SLAM using anchor nodes. In *European Conference on Mobile Robotics (ECMR)*, Orbero, Sweden, September 2011.

[12] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J.J. Leonard. Real-time 6-DOF multi-session visual SLAM over large scale environments. *J. of Robotics and Autonomous Systems*, 61(10):1144–1158, October 2013.

[13] D.M. Rosen, M. Kaess, and J.J. Leonard. An incremental trust-region method for robust online sparse least-squares estimation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1262–1269, St. Paul, MN, May 2012.

[14] D.M. Rosen, M. Kaess, and J.J. Leonard. RISE: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Trans. Robotics*, 2014. doi: 10.1109/TRO.2014.2321852. To appear.

[15] R.F. Salas-Moreno, B. Glocker, P.H.J. Kelly, and A.J. Davison. Dense planar SLAM. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, Munich, Germany, September 2014.

[16] F. Servant, E. Marchand, P. Houlier, and I. Marchal. Visual plane-based simultaneous localization and model refinement for augmented reality. In *Intl. Conf. on Pattern Recognition (ICPR)*, December 2008.

[17] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. Point-plane SLAM for hand-held 3D sensors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013.

[18] A.J.B. Trevor, J.G. Rogers, and H.I. Christensen. Planar surface SLAM with 3D and 2D sensors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3041–3048, May 2012.

[19] J. Weingarten and R. Siegwart. 3D SLAM using planar segments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3062–3067, October 2006.

[20] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *Intl. J. of Robotics Research*, 2014. To appear.

[21] T. Whelan, L. Ma, E. Bondarev, P.H.N. de With, and J. McDonald. Incremental and batch planar simplification of dense point cloud maps. *J. of Robotics and Autonomous Systems*, 2014. To appear.