

Systematic Handling of Heterogeneous Geometric Primitives in Graph-SLAM Optimization

Irvin Aloise, Bartolomeo Della Corte, Federico Nardi and Giorgio Grisetti

Department of Computer, Control and Management Engineering

Sapienza University of Rome

Email: {ialoise,dellacorte,fnardi,grisetti}@diag.uniroma1.it

Abstract—In this paper¹, we propose a pose-landmark graph optimization back-end that supports maps consisting of points, lines or planes. Our back-end allows representing both homogeneous (*point-point*, *line-line*, *plane-plane*) and heterogeneous measurements (*point-on-line*, *point-on-plane*, *line-on-plane*). Rather than treating all cases independently, we use a unified formulation that leads to both a compact derivation and a concise implementation. The additional geometric information, deriving from the use of higher-dimension primitives and constraints, yields to increased robustness and widens the convergence basin of our method. We evaluate the proposed formulation both on synthetic and raw data. Finally, we made available an open-source implementation to replicate the experiments.

I. INTRODUCTION

The Simultaneous Localization and Mapping (SLAM) problem is well-known in Robotics. The goal of a SLAM system is to concurrently estimate the position of the robot and a map of its operating environment from robot measurements. SLAM is a key component of more complex systems such as autonomous driving vehicles, home service robots or augmented reality applications. Due to its relevance, SLAM has been for more than two decades one of the most active research fields in robotics. Over the years, we have seen an impressive evolution in the hardware (sensors and computation). To use this more powerful hardware or to adapt to new sensors, a SLAM system has to be scalable. This consideration promoted a paradigm shift from filtering based SLAM approaches to more modular factor-graph based systems, the latter becoming nowadays the *de-facto* SLAM standard [1].

A factor-based SLAM system consists of two components: a *front-end* and a *back-end*. The former takes as input raw data, generating a factor-graph representing map and robot trajectory. The variables in a factor-graph are either robot poses or salient landmarks in the map. The factors encode measurements correlating one or more variables. In the special case of measurements involving only pairs of variables, a factor graph degenerates to a regular graph where the factors are encoded in the edges. If the variables represent only the poses of the robot the graph further specializes to a pose-graph [2]. The role of the back-end is to find a solution for the factor graph generated by the front-end. A solution of a factor graph is a configuration of the variables that better explains

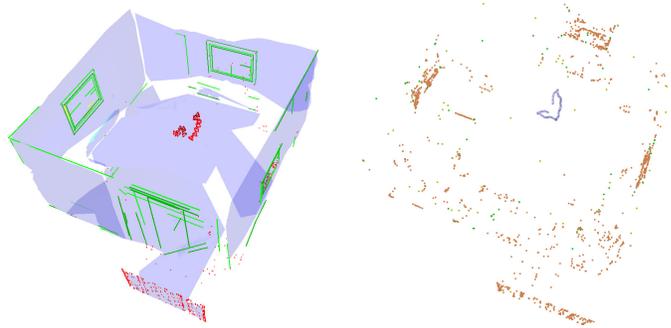


Fig. 1: Reconstruction and relative optimized graph obtained from the sequence *lr/tr1* of the ICL-NUIM dataset. The proposed approach takes advantage of the high-dimension primitives extracted from the environment, such as planes and lines, along with points, to perform accurate and robust optimization.

the measurements encoded in the factors. Possibly, a back-end should also provide an estimate of the solution’s uncertainty. Nearly all modern state-of-the-art SLAM systems are graph-based. The main distinction is between pose-landmark systems [3] [4] [5] [6] [7] where the variables consist of robot poses and landmarks and pose-pose systems [8], [9] where each node stores a local map or a dense measurement acquired at that robot position.

Most of the pose-landmark systems rely on point landmarks since they are easy to detect and there is solid literature on how to compute descriptors for visual features from point landmarks. Despite their attractiveness, using only points as landmarks neglects potentially useful structural information that can be captured by higher dimension primitives, such as planes or lines. Expressing a set of points extracted from a planar surface as a plane reduces the amount of information stored [10], while yielding to comparable accuracy in cloud registration problems [11]. The same considerations apply when representing a set of collinear points as a line [12].

In our previous work [10], we proposed a unified representation for local maps consisting of points, planes, and lines. A generic element of the map is called *matchable*, regardless of its type. We furthermore proposed both an iterative and a direct solver that can solve the registration problem between two sets of matchables, given a set of geometric constraints between them. Notably, the constraints between two matchables can be of any type (*point-point*, *point-line*, *point-plane*, *line-line*, *line-plane* and *plane-plane*). This system allows determining the single most likely robot position that registers one set of

¹This paper has been selected to appear also in IEEE Robotics and Automation Letters, under the same title.

primitives onto the other. The two scenes are regarded as rigid bodies.

In this work, we address the problem of global optimization, by proposing a **SLAM** back-end that takes a pose-landmark graph constructed from odometry and matchable measurements. The outcome of our back-end is the most likely configuration of robot poses and matchables with respect to the measurements. With this unified representation, we can model *heterogeneous* constraints, that is, a landmark can be observed as a primitive of a different type. As an example, a line-plane constraint can capture the observation of a wall landmark made by a planar laser scanner.

If we exclude the heterogeneous constraints that are characteristic of our system, our back-end is functionally equivalent to recent approaches that operate on high-dimension primitives [13] [14] [15]. Most of these systems rely on a front-end tailored for RGB-D data. In this case, the main difference between our approach and the ones mentioned above is the way primitives and constraints are represented inside the system. Our approach uses a single type of landmark, and a single type for the measurement, whereas other approaches rely on specific representations for each primitive and error function.

The uniform representation allows for a very compact implementation of a general system. As expected, using the additional information from high-dimension landmarks and heterogeneous constraints has a positive effect on the convergence properties of the back-end. We support our claims with synthetic and raw-data experiments. Fig. 1 shows the reconstruction and the relative optimized graph obtained from real data. Finally, we provide an open source C++ plugin² for *g²o* that implements our back-end approach.

II. RELATED WORK

SLAM is a well-known problem and many approaches have been proposed through the years [16] [17]. Early probabilistic solutions were based on filtering, mainly relying on Gaussian Filters (EKF and UKF and Information Filters) [18] or Particle Filters [19] [20]. Gaussian Filters have been reported to suffer from accuracy issues arising from the inherent linearizations [21]. Particle filters, in contrast, do not scale to large environments since the number of particles required might grow exponentially with the trajectory length. Currently, the community has converged towards Maximum-A-Posteriori (MAP) based approaches that leverage on the sparsity of the problem to achieve efficient computation. The graph-based formulation of the **SLAM** problem has been proposed by Lu and Milios [22]. They build a pose-graph from laser scans and odometry data, performing the batch graph optimization via least-squares. Subsequently, to overcome the computational limitations of the batch solution, Gutmann and Konolige [23] proposed a Local Registration and Global Correlation method (LRGC). The graph was incrementally built, and the optimization was performed when cycles were detected.

Least-squares based approaches, however, were suffering from computational issues due to the lack of efficient solvers for sparse linear algebra. To tackle these problems, the community proposed optimization systems based on *Relaxation* [24] [25] [26] and Stochastic Gradient Descent (SGD) [2] [27]. The first formalism is based on the concept of *relaxing* some optimization constraints. This leads to lower computational complexity, weighted by a slower convergence rate [28]. On the other hand, **SGD** approaches provide a decomposition of the standard gradient descent problem in many smaller problems. Thus, the optimization is performed considering the constraints individually.

All these works were still not considering the intrinsic sparsity of the problem. Dellaert *et al.* developed a system called $\sqrt{\text{SAM}}$ [29] that was based on least-squares and exploited the sparsity to efficiently compute a solution. $\sqrt{\text{SAM}}$ was further expanded adding incremental optimization and new more efficient data-structures, respectively in *iSAM* [30] and *iSAM2* [31]. Concurrently, Kümmerle *et al.* developed *g²o* [32], a tool to easily optimize generic graphs.

Whereas these systems have been designed as general factor graph optimization systems, the default factors supported assumed to operate on landmarks represented as points. These are generally employed since they are easy to detect and describe. Still, they do not provide information about the world structure. Moreover, in certain conditions the number of points detected could be insufficient - *e.g.* a robot navigating in a textureless environment. To add compactness and descriptiveness to the world reconstruction, many approaches embed other geometric primitives. Intuitively, line segments were firstly investigated [33] [34] since they were relatively easy to compute using different sensors [35] [36]. In particular, the work of Castellanos *et al.* [33] proposes a systematic paradigm to describe line-segments and potentially other geometric features, similarly to our approach: SPmap. For each landmark type it is required to provide a suitable *perturbation vector* and a *binding matrix* that will characterize its shape. Still, SPmap is restricted to 2D environments and designed to operate in conjunction with a Gaussian Filter. Accordingly, a system built this way shares the shortcomings of the estimator used. The work presented in this paper builds on SPmap and generalizes its ideas to 3D environments and non-linear estimators. Subsequently, Pedraza *et al.* investigated the use of generic B-splines [37] to correctly describe more complex shapes.

On the same line of thought, other researchers introduced the plane primitive in **SLAM** pipelines. De la Puente *et al.* [38] successfully embedded rectangles in a hierarchical graph-based pipeline, however, their approach is limited to 2D scenarios. Salas *et al.* [39] proposed a dense planar approach, detecting bounded planes and surfels from RGB-D images. On the contrary, Kaess [11] used infinite planes to perform **SLAM**, proposing a quaternion-based plane parametrization that improves the convergence of the optimization. The same parametrization has also been used in monocular context [40].

In our previous work [10], we exploited these works,

²Source available at: <https://srrg.gitlab.io/sashago.html>

providing a unified representation of heterogeneous primitives by means of *degenerate quadratics*. The provided unification allows the registration error to be computed regardless of the specific type of primitive. This includes the possibility to generate cross-constraints between primitives of different types.

In this work, we exploit *matchables* into our back-end. Thanks to the provided unification, the system implementation complexity is significantly bounded and it is possible to add cross-constraints between different primitives without specific adaptations of the back-end.

III. POSE-LANDMARK GRAPH OPTIMIZATION

A pose-landmark graph back-end aims at finding the configuration of robot poses $\{\mathbf{X}_{1:N^R}^R\}$ and landmarks $\{\mathbf{X}_{1:N^L}^L\}$ that is maximally consistent with the measurements. Let $\mathcal{X} = \{\mathbf{X}_{1:N^R}^R\} \cup \{\mathbf{X}_{1:N^L}^L\}$ be the set of all variables. Each robot pose and landmark is represented by a node in the graph. Nodes are connected by edges $(\mathbf{z}_k, \mathbf{\Omega}_k, \mathbf{h}_k(\cdot))$ that encode measurement mean \mathbf{z} and relative information matrix $\mathbf{\Omega}$. We indicate with $\mathbf{h}_k(\cdot)$ the observation function of the specific edge type.

To retrieve the indices of the first and second node of an edge from the measurement index k , we introduce the selector functions $i(k)$ and $j(k)$. Let $\hat{\mathbf{z}}_k = \mathbf{h}_k(\mathbf{X}_{i(k)}, \mathbf{X}_{j(k)})$ be the prediction function for the k^{th} measurement. In case of Euclidean variables, the residual error between prediction and measurement can be computed as the difference between the two vectors:

$$\mathbf{e}_k(\mathbf{X}_{i(k)}, \mathbf{X}_{j(k)}) = \mathbf{h}_k(\mathbf{X}_{i(k)}, \mathbf{X}_{j(k)}) - \mathbf{z}_k. \quad (1)$$

The optimal configuration \mathcal{X}^* of the nodes with respect to the measurements is the one that minimizes the cumulative Ω -norm of the residual error

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} \sum_k \mathbf{e}_k^T(\mathcal{X}) \mathbf{\Omega}_k \mathbf{e}_k(\mathcal{X}). \quad (2)$$

This optimization problem is usually solved using iterative approach, such as Iterative Least-Squares (ILS) or **SGD**. In general, **ILS** methods repeatedly resolve a linear approximation of Eq. (2). Considering a non-Euclidean state space, this is obtained expanding the error terms in Eq. (2) around the origin of a chart computed at current initial guess $\check{\mathcal{X}}$:

$$\begin{aligned} \mathbf{e}_k(\check{\mathcal{X}} \boxplus \Delta \mathbf{x}) &\approx \mathbf{e}_k(\check{\mathcal{X}}) + \left. \frac{\partial \mathbf{e}_k(\check{\mathcal{X}} \boxplus \Delta \mathbf{x})}{\partial \Delta \mathbf{x}} \right|_{\Delta \mathbf{x}=\mathbf{0}} \Delta \mathbf{x} \\ &= \mathbf{e}_k + \mathbf{J}_k \Delta \mathbf{x}. \end{aligned} \quad (3)$$

In Eq. (3), the \boxplus operator applies a perturbation $\Delta \mathbf{x}$ to the generic manifold space \mathcal{X} . Obviously, in the straightforward case of Euclidean states, the \boxplus operator degenerates to the regular vector addition. $\Delta \mathbf{x}$ should be a minimal parametrization for the increments of all state variables. Expanding Eq. (3) in Eq. (2) leads to the following quadratic form:

$$\Delta \mathbf{x}^T \left[\underbrace{\sum_k \mathbf{J}_k^T \mathbf{\Omega}_k \mathbf{J}_k}_{\mathbf{H}} \right] \Delta \mathbf{x} + 2 \left[\underbrace{\sum_k \mathbf{e}_k^T \mathbf{\Omega}_k \mathbf{J}_k}_{\mathbf{b}^T} \right] \Delta \mathbf{x} + \text{const} \quad (4)$$

Eq. (4) can be straightforwardly minimized in $\Delta \mathbf{x}$ by solving the following linear system:

$$\mathbf{H} \Delta \mathbf{x}^* = -\mathbf{b}. \quad (5)$$

The increment is applied to the current solution as follows:

$$\check{\mathcal{X}} \leftarrow \check{\mathcal{X}} \boxplus \Delta \mathbf{x}^*. \quad (6)$$

In our scenario, each measurement \mathbf{z}_k depends only on two nodes. For this reason, the Jacobian computed in Eq. (3) will be non zero only in the blocks relative to the state variables $\mathbf{X}_{i(k)}$ and $\mathbf{X}_{j(k)}$ connected by the edge k :

$$\mathbf{J}_k = [\mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{J}_{i(k)} \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \quad \mathbf{J}_{j(k)} \quad \mathbf{0} \quad \cdots]$$

By construction, $\mathbf{H} = \sum_k \mathbf{J}_k^T \mathbf{\Omega}_k \mathbf{J}_k$. Therefore, the sparsity will result also in the linear system described in Eq. (5). Exploiting this sparsity is a key aspect of modern optimization systems - e.g. *g²o* [32] or **iSAM** [30]. **ILS** repeats the above procedure until convergence.

Also the measurements may lie on a manifold space. In the remaining, we indicate with \mathbf{Z}_k a parametrization for the k^{th} manifold measurement. To compute the residual error in Eq. (1) we need to define a difference operator $\mathbf{Z}_b \boxminus \mathbf{Z}_a = \Delta \mathbf{z}_{ab}$ between manifold objects, that returns an Euclidean difference between two objects. Intuitively, if $\mathbf{Z}_a = \mathbf{Z}_b$, then $\Delta \mathbf{z}_{ab} = \mathbf{0}$. In Alg. 1 we reported the evolution of the vanilla Gauss-Newton (GN) algorithm. Some scenarios (including ours) lead to a rank deficient matrix \mathbf{H} and, hence, require a damping factor or a different optimization algorithm - e.g. Levenberg-Marquardt (LM).

To specialize this problem to a specific case we need to:

- define the domain and the perturbation for each state variable, along with a \boxplus operator that applies the perturbation to the variable;
- define an error function that computes a vector difference between prediction and measurement, for each type of edge.

We refer the reader to [41] for further details on this topic.

A. Example: 3D Pose and 3D Point Landmark Case

One of the most common scenario involves 3D robot poses \mathbf{X}^R and 3D point landmarks \mathbf{x}^L . In this sense, \mathbf{X}^R lies on the $SE(3)$ space and a possible non-minimal parametrization is a 3D isometry. The latter is composed by a rotation matrix \mathbf{R} and a translation vector \mathbf{t} . A minimal parametrization used to describe the increment $\Delta \mathbf{x}^R$ can be a 6-dimensional vector $\Delta \mathbf{x} = [\Delta \mathbf{t}^T \quad \Delta \alpha^T]^T = [x \ y \ z \ \theta \ \phi \ \psi]^T$. A landmark \mathbf{x}^L is a 3D vector that lies on the Euclidean space \mathbb{R}^3 . As a result, the \boxplus degenerates to a regular sum and the increments $\Delta \mathbf{x}^L$ live in the same space as the landmark. Tab. I summarizes how to apply the increment for such objects.

Analogously, measurements are of type pose-pose and pose-landmark and, thus, we should define different error functions. In the former case, to compute the distance between $SE(3)$ objects it is possible to use the matrix difference defined in [28].

Algorithm 1 GN minimization algorithm for manifold measurements and state spaces

Require: Initial guess $\check{\mathcal{X}}$; Measurements \mathcal{Z} ; Threshold ϵ .
Ensure: Optimal solution \mathcal{X}^*

- 1: $F_{new} \leftarrow F(\check{\mathcal{X}})$ ▷ compute the current error
- 2: **while** $\check{F} - F_{new} > \epsilon$ **do**
- 3: $\check{F} \leftarrow F_{new}$
- 4: $\mathbf{b} \leftarrow \mathbf{0}$
- 5: $\mathbf{H} \leftarrow \mathbf{0}$
- 6: **for** $\mathbf{Z}_k \in \mathcal{Z}$ **do**
- 7: $i \leftarrow i(k)$ ▷ extract indices
- 8: $j \leftarrow j(k)$
- 9: $\mathbf{e}_k \leftarrow \mathbf{e}_k(\mathbf{X}_i, \mathbf{X}_j)$ ▷ compute error vector
- 10: $\mathbf{J}_{i(k)} \leftarrow \left. \frac{\partial \tilde{\mathbf{e}}_k(\mathbf{X}_i, \mathbf{X}_j)}{\partial \Delta \mathbf{x}_i} \right|_{\Delta \mathbf{x}_i = \mathbf{0}}$ ▷ Jacobian i
- 11: $\mathbf{J}_{j(k)} \leftarrow \left. \frac{\partial \tilde{\mathbf{e}}_k(\mathbf{X}_i, \mathbf{X}_j)}{\partial \Delta \mathbf{x}_j} \right|_{\Delta \mathbf{x}_j = \mathbf{0}}$ ▷ Jacobian j
- 12: $\mathbf{J}_k \leftarrow [\mathbf{0} \cdots \mathbf{0} \ \mathbf{J}_{i(k)} \ \mathbf{0} \cdots \mathbf{0} \ \mathbf{J}_{j(k)} \ \mathbf{0} \cdots \mathbf{0}]$
- 13: $\mathbf{H} \leftarrow \mathbf{H} + \mathbf{J}_k^T \Omega_k \mathbf{J}_k$ ▷ contribution of \mathbf{Z}_k in \mathbf{H}
- 14: $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{J}_k^T \Omega_k \mathbf{e}_k$ ▷ contribution of \mathbf{Z}_k in \mathbf{b}
- 15: $\Delta \mathbf{x} \leftarrow \text{solve}(\mathbf{H} \Delta \mathbf{x} = -\mathbf{b})$ ▷ solve w.r.t. $\Delta \mathbf{x}$
- 16: $\check{\mathcal{X}} \leftarrow \check{\mathcal{X}} \boxplus \Delta \mathbf{x}$ ▷ update the state
- 17: $F_{new} \leftarrow F(\check{\mathcal{X}})$ ▷ compute the new error
- 18: **return** $\check{\mathcal{X}}$

	increment	$\mathbf{X} \boxplus \Delta \mathbf{x}$
\mathbf{X}^R	$\Delta \mathbf{x}^R = [\Delta \mathbf{t}^T \ \Delta \alpha^T]^T$	$[\mathbf{R}_{\Delta \alpha} \mathbf{R} \mid \mathbf{R}_{\Delta \alpha} \mathbf{t} + \Delta \mathbf{t}]$
\mathbf{x}^L	$\Delta \mathbf{x}^L \in \mathbb{R}^3$	$\mathbf{x}^L + \Delta \mathbf{x}^L$

TABLE I: Minimal parametrization of increments and \boxplus operator for $SE(3)$ and \mathbb{R}^3 objects.

measurement type	error function
pose-pose	$\mathbf{e}(\mathbf{X}_i^R, \mathbf{X}_j^R) = \text{flatten}(\mathbf{X}_i^{R-1} \mathbf{X}_j^R - \mathbf{Z}_{ij}^R)$
pose-landmark	$\mathbf{e}(\mathbf{X}_i^R, \mathbf{x}_j^L) = \mathbf{X}_i^{R-1} \mathbf{x}_j^L - \mathbf{z}_{ij}^L$

TABLE II: Standard error functions for $SE(3)$ and \mathbb{R}^3 measurements. The operator $\text{flatten}(\cdot)$ produces a 12D vector composed by the versors and the translation component of the isometry.

For the pose-landmark measurements, the prediction is obtained expressing the landmark in the robot frame. Consequently, the pose-landmark error is the Euclidean difference between prediction and observation. Tab. II summarizes these definitions.

We will apply the methodology outlined with this example to define a pose-landmark back-end where the landmarks can be either points, lines or planes and the measurements can express arbitrary constraints between landmarks.

IV. OPTIMIZATION USING MATCHABLES

In this section we recall the representation of heterogeneous primitives presented in [10], that allows a unification between different geometric primitives - *i.e.* points, lines and planes. In this formalism, the objects are referred to as *matchables*. In Sec. IV-A we will report a brief overview of this representation, while in Sec. IV-B and Sec. IV-C we show how to embed it in standard non-linear optimization algorithm like

GN or LM, according to the scheme defined in Sec. III-A.

A. Matchable Basics

A matchable expresses a point, a line or a plane through a degenerate quadric. As a result, a point \mathbf{x} belonging any of these primitives satisfies the following equation:

$$(\mathbf{x} - \mathbf{p})^T \mathbf{A} (\mathbf{x} - \mathbf{p}) = 0. \quad (7)$$

Since \mathbf{A} is symmetric, the eigenvalue decomposition allows to isolate the orientation \mathbf{R} of the primitive and its shape $\mathbf{\Lambda}$ as

$$\mathbf{A} = \mathbf{R} \mathbf{\Lambda} \mathbf{R}^T \quad \text{with} \quad \mathbf{R} = [\mathbf{r}_x \ \mathbf{r}_y \ \mathbf{r}_z]. \quad (8)$$

We recall that the column vectors \mathbf{r}_* in Eq. (8) are orthonormal and $\mathbf{\Lambda}$ is the diagonal matrix storing the eigenvalues of \mathbf{A} . In the remainder of this work we will refer to \mathbf{r}_x as the *direction* \mathbf{d}_M of the matchable. Intuitively, \mathbf{d}_M represents also the symmetry axis of the matchable.

We represent a matchable M as the following tuple:

$$M : \langle \mathbf{p}_M, \ \mathbf{R}_M, \ \mathbf{\Lambda}_M \rangle. \quad (9)$$

Here $\mathbf{p}_M \in \mathbb{R}^3$ is the centroid of the matchable, and describes its position in the space. $\mathbf{R}_M \in SO(3)$ is a rotation matrix and captures its orientation. Finally, $\mathbf{\Lambda}_M$ is a 3×3 diagonal matrix that characterizes the matchable shape. More specifically, a point is described as a sphere of null radius ($\mathbf{\Lambda} = \text{diag}(1, 1, 1)$), a line is a cylinder of null radius ($\mathbf{\Lambda} = \text{diag}(0, 1, 1)$) and a plane consists of two parallel planes with null distance ($\mathbf{\Lambda} = \text{diag}(1, 0, 0)$). Choosing a value of 1 for the non-null eigenvalues of $\mathbf{\Lambda}$ results in Eq. (7) being the squared Euclidean distance between a point \mathbf{x} and the closest point in the quadric.

B. Applying a Perturbation to a Matchable

We notice that the representation presented in Eq. (9) is clearly non-minimal and non-Euclidean, due to the presence of \mathbf{R} . To use matchables as landmarks in the schema proposed in Sec. III, we need to define a locally Euclidean perturbation $\Delta \mathbf{m}$ along a \boxplus operator that applies the perturbation to a matchable. Still, $\Delta \mathbf{m}$ should have enough degrees of freedom to move the matchable centroid \mathbf{p}_M and to change its orientation. Not all possible changes in orientation affect a primitive: a point has no orientation and rotating a plane along its normal or rotating a line along its direction has no effect on the primitive.

Albeit non-minimal, a reasonably small representation of a perturbation $\Delta \mathbf{m}$ for a matchable is the following:

$$\Delta \mathbf{m} = [\Delta \mathbf{p}^T \ \Delta \alpha_y^T \ \Delta \alpha_z^T]^T. \quad (10)$$

In this paradigm $\Delta \mathbf{m}$ is the concatenation of a perturbation of the centroid $\Delta \mathbf{p}$ and two rotations $\Delta \alpha_y$ and $\Delta \alpha_z$ along axes non parallel to the main direction of the matchable \mathbf{r}_x . Applying a perturbation to a matchable is done as follows:

$$M \boxplus \Delta \mathbf{m} \triangleq \langle \mathbf{p}_M + \Delta \mathbf{p}, \ \mathbf{R}_M \Delta \mathbf{R}, \ \mathbf{\Lambda}_M \rangle \quad (11)$$

where $\Delta \mathbf{R} = \mathbf{R}_y(\Delta \alpha_y) \cdot \mathbf{R}_z(\Delta \alpha_z)$ is the rotation matrix obtained from the orientation part of the $\Delta \mathbf{m}$.

C. Error Function

According to the schema in Sec. III, we need to define a way to compute the prediction of a matchable. Moreover, we should characterize an error function that computes the Euclidean vector describing the difference between prediction and observation.

The prediction is obtained simply by expressing the observed matchable $\mathbf{M}_j = \langle \mathbf{p}_j, \mathbf{R}_j, \mathbf{A}_j \rangle$ in the reference frame of the observing robot pose $\mathbf{X}_i = [\mathbf{R}_i \mid \mathbf{t}_i]$. Without loss of generality, we assume that our system stores the trajectory of the robot as “world-in-robot” transforms. Hence, the prediction $\hat{\mathbf{M}}_{ij}$ is computed as follows:

$$\hat{\mathbf{M}}_{ij} = \mathbf{h}(\mathbf{X}_i, \mathbf{M}_j) = \langle \mathbf{R}_i \mathbf{p}_j + \mathbf{t}_i, \quad \mathbf{R}_i \mathbf{R}_j, \quad \mathbf{A}_j \rangle \quad (12)$$

We need now to compute a difference vector between prediction and observation, that are potentially different primitives. Depending on the compared primitives, not all elements of a matchable are relevant: in the case of two points, the orientation is meaningless, or when comparing a plane and a line the direction vectors \mathbf{r}_x should be orthogonal.

The solution lies in computing an extended difference vector that captures all possible differences between two matchables and then to rely on the information matrix $\mathbf{\Omega}$ to suppress meaningless components of the difference. To this extent, considering two matchables \mathbf{M}_a and \mathbf{M}_b where \mathbf{M}_b represents a primitive of dimension equal or higher the one of \mathbf{M}_a , the 7D difference vector between the two is computed as follows:

$$\mathbf{e}(\mathbf{M}_a, \mathbf{M}_b) \triangleq \begin{pmatrix} \mathbf{e}_p \\ \mathbf{e}_d \\ e_o \end{pmatrix} = \begin{pmatrix} \mathbf{R}_b^T (\mathbf{p}_a - \mathbf{p}_b) \\ \mathbf{d}_a - \mathbf{d}_b \\ \mathbf{d}_a^T \mathbf{d}_b \end{pmatrix} \quad (13)$$

In Eq. (13)

- $\mathbf{e}_p \in \mathbb{R}^3$ describes the euclidean distance between the two matchables origins.
- if both \mathbf{M}_a and \mathbf{M}_b have a direction, $\mathbf{e}_d \in \mathbb{R}^3$ describes the misalignment between the two. The reader might notice that, since the distance is computed *locally*, singularities are avoided.
- $e_o \in \mathbb{R}$ describes whether $\mathbf{d}_a \perp \mathbf{d}_b$. This component is enforced in cases of line lying on a plane and viceversa.

The information matrix $\mathbf{\Omega}$ will shape the error, enabling and disabling components of Eq. (13) depending on the types of the considered matchables. Hence, considering measurement $\mathbf{Z}_k = \mathbf{Z}_{ij}$ connecting node i and j , characterized by Additive White Gaussian Noise (AWGN) $\mathcal{N}(\mathbf{0}, \bar{\mathbf{\Omega}}_k^{-1})$ the information matrix $\mathbf{\Omega}_k$ will be constructed as follows:

$$\mathbf{\Omega}_k = \mathbf{A} \bar{\mathbf{\Omega}}_k \mathbf{A}^T \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_p & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_d & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & A_o \end{pmatrix} \quad (14)$$

In Eq. (14), \mathbf{A} represents the “activation matrix” used to discard non-relevant components of the residual error in Eq. (13). Tab. III reports the composition of \mathbf{A} , depending on the possible matchable combinations. As a result, the final $\mathbf{\Omega}_k$ will properly scale the residual error by the AWGN statistics. We refer to the reader to [10] for further details on this topic.

		\mathbf{Z}_{ij}		
		point	line	plane
$\hat{\mathbf{Z}}_{ij}$	point	$\mathbf{A}_p = \mathbf{I}$ $\mathbf{A}_d = \mathbf{0}$ $A_o = 0$	$\mathbf{A}_p = \mathbf{A}_{\mathbf{Z}_{ij}}$ $\mathbf{A}_d = \mathbf{0}$ $A_o = 0$	$\mathbf{A}_p = \mathbf{A}_{\mathbf{Z}_{ij}}$ $\mathbf{A}_d = \mathbf{0}$ $\Omega_o = 0$
	line	—	$\mathbf{A}_p = \mathbf{A}_{\mathbf{Z}_{ij}}$ $\mathbf{A}_d = \mathbf{I}$ $A_o = 0$	$\mathbf{A}_p = \mathbf{A}_{\mathbf{Z}_{ij}}$ $\mathbf{A}_d = \mathbf{0}$ $A_o = 1$
	plane	—	—	$\mathbf{A}_p = \mathbf{A}_{\mathbf{Z}_{ij}}$ $\mathbf{A}_d = \mathbf{I}$ $A_o = 0$

TABLE III: Characterization of activation matrix \mathbf{A} for each possible combinations of matchables. We indicate with \mathbf{Z}_{ij} and $\hat{\mathbf{Z}}_{ij}$ respectively the predicted and the actual measurement. In this work we cover only cases when a primitive can be observed through a measurement of lower or equal dimension.

We now have all elements to specialize Alg. 1 to operate on a map whose landmarks are matchables and whose observations are arbitrary constraints between heterogeneous matchables. Denoting with \mathbf{Z}_{ij} the observation, the contribution of this edge to the cost function Eq. (2) is computed as

$$e_{ij}(\mathbf{X}_i, \mathbf{M}_j) = \|\mathbf{e}(\mathbf{X}_i \mathbf{M}_j, \mathbf{Z}_{ij})\|_{\mathbf{\Omega}_{ij}}. \quad (15)$$

Finally, the complete derivation of Jacobians \mathbf{J}_i and \mathbf{J}_j is described in the supplementary materials³. However, since our parametrization is redundant for low informative objects like points, the information matrix \mathbf{H} may be rank deficient in some scenarios. This means that, in some cases, it is not possible to use a first order optimization algorithm like Gauss-Newton. We overcome this issue using the Levenberg-Marquardt algorithm, that guarantees full rank of matrix \mathbf{H} .

In Eq. (13) we defined the error vector considering \mathbf{M}_b as a primitive of dimension higher or equal to the one of \mathbf{M}_a . These cases correspond to the upper triangular part of Tab. III and are the ones typically observed in SLAM. In the unlikely case that a landmark of lower dimension is observed through a higher dimension primitive⁴, Eq. (13) is still valid, but the two primitives have to be flipped. Doing so, however, results in the \mathbf{R} matrix of the predicted landmark to be dependant on the robot position, and the straightforward derivation of the Jacobians presented in the supplementary material does not hold. In this work we do not cover these cases.

V. EXPERIMENTAL EVALUATION

In this section, we provide experiments aimed to support the claims of larger convergence basin and improved robustness compared to point-only approaches. In particular, we report in Sec. V-A the results of the convergence analysis conducted on synthetically generated data. Sec. V-B reports a brief overview of the front-end used to generate graphs from standard RGB-D datasets. Moreover, we analyze the robustness of our approach evaluating the error on the optimized trajectories.

³The supplementary materials are attached to the RA-L version of the manuscript.

⁴An example of this situation might be a robot equipped with a large planar bumper allowed to slide only along the plane’s normal hitting the edge of an object.

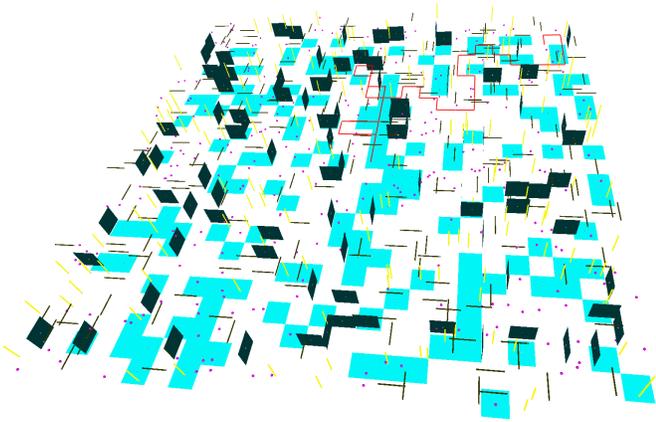


Fig. 2: Example of a synthetic randomly generated world containing points (purple), planes (cyan) and lines (yellow).

A. Synthetic Data

The aim of this set of experiments is to evaluate the effect of our back-end that can operate on higher order landmarks under different sensor noise and initial guess conditions. To generate the synthetic datasets we used an extension of `g2o_simulator` included in the `g2o` package to generate a synthetic world consisting of points, planes and lines. Fig. 2 illustrates a simulated scenario example.

The robot moves on a Manhattan-like trajectory along this maze, and senses the matchables. The measurement of a matchable can be of two types: homogeneous or not. As an example, the former occurs when a plane is sensed as a plane by the robot, while the latter occurs when a plane is sensed either as a line or as a point. In our experiments we isolate the effects of both homogeneous and non-homogeneous measurements.

On this synthetic world, we moved the robot on different trajectories of increasing length (`g0`, `g1` and `g2`). For each trajectory, we simulated the sensing of the matchables, and we distinguish the following four cases:

- `all`: a matchable in the map is sensed through all sensing modalities (homogeneous and not)
- `hom`: a matchable in the map generates a measure of the same type (point-point, line-line, plane-plane).
- `non-hom`: a matchable in the map generates a non homogeneous measurement.
- `point`: only the point matchables are considered.

Tab. IV reports the number of nodes and edges for each graph and sensing modality.

The `non-hom` datasets take into account only lower dimensional constraints generated from a matchable landmark - e.g. a point is never sensed as a plane. For this reason the number of variables of `non-hom` dataset is smaller, since point landmarks are not considered.

To analyze the effects under varying noise conditions, we applied to each scenario different noise figures - both on the odometry measurements and on the matchable measurements. The pose noise is sampled from $\mathcal{N}_t^p(0, \Sigma_t^p)$ and $\mathcal{N}_r^p(0, \Sigma_r^p)$ respectively for the translational and rotational pose components. The landmark noise, analogously, is sampled from

dataset	pose nodes	total nodes	total edges
<code>g0-all</code>	100	246	1797
<code>g0-hom</code>		246	1245
<code>g0-non-hom</code>		161	653
<code>g0-point</code>		151	554
<code>g1-all</code>	1000	2338	32426
<code>g1-hom</code>		2338	21756
<code>g1-non-hom</code>		1644	11671
<code>g1-point</code>		1439	8238
<code>g2-all</code>	10000	18163	198945
<code>g2-hom</code>		18163	143868
<code>g2-non-hom</code>		13191	65078
<code>g2-point</code>		13632	72613

TABLE IV: Synthetic dataset specifications. Graphs are collected according to the number of robot poses

	low-noise	mid-noise	high-noise
Σ_t	[0.01 0.01 0.001]	[0.1 0.1 0.01]	[1.0 1.0 0.01]
Σ_R	[0.001 0.001 0.005]	[0.01 0.01 0.05]	[0.01 0.01 0.1]
Σ_p	[0.005 0.005 0.005]	[0.05 0.05 0.05]	[0.5 0.5 0.5]
Σ_d	[0.001 0.001]	[0.01 0.01]	[0.1 0.1]

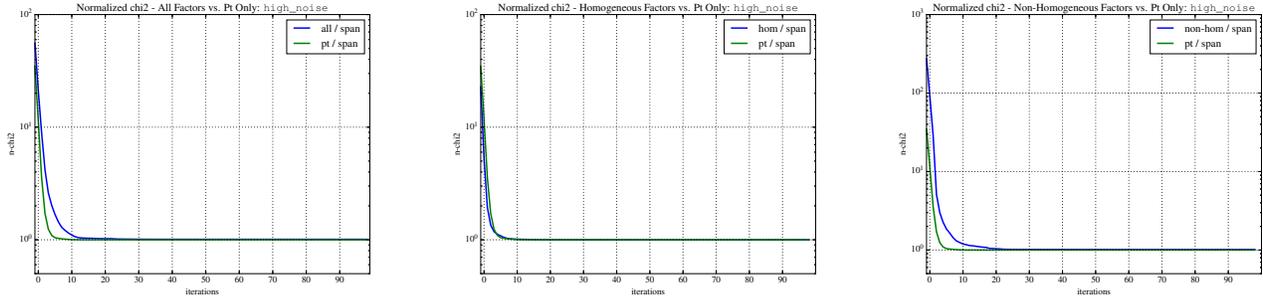
TABLE V: In this table we provide the value of noise added to the synthetic datasets. The values of Σ_t and Σ_p are expressed in meters while Σ_R and Σ_d in radians.

$\mathcal{N}_t^m(0, \Sigma_t^m)$ and $\mathcal{N}_r^m(0, \Sigma_r^m)$. In Tab. V we summarize the noise figures used in the experiments. To compare the chi2 of point-only cases with our approach, we normalized them to their chi2 at the respective global optima. For this reason all the plots might converge to a value of $\text{chi2} = 1$. Note that the global optimum is not the ground truth due to the noise in the measurements. To compute this optimum we execute `LM` until convergence, starting from the ground truth configuration.

In Fig. 3 we report the chi2 comparison of point-only and matchable based landmark obtained on the 100 poses datasets, using high-noise settings. Intuitively, when the graph dimension is relatively small there is no difference in terms of convergence speed. However, increasing the graph dimension leads to different results, showing the limits of point-based landmarks. The initial guess used in all the system is the spanning tree. In Fig. 4 we reported the chi2 comparison in the graphs with 1000 poses in high noise configuration. In all cases, our approach succeeds in reaching the optimum, while standard point based optimization remains stuck in suboptimal configurations. In bigger graph - i.e. 10000 poses - and under extreme noise conditions both approaches fail the optimization, due to nature of the problem. However, as shown in Fig. 5, point based optimization fails with every noise configuration, while our approach produces better results - that can be used as initial guess for further optimization.

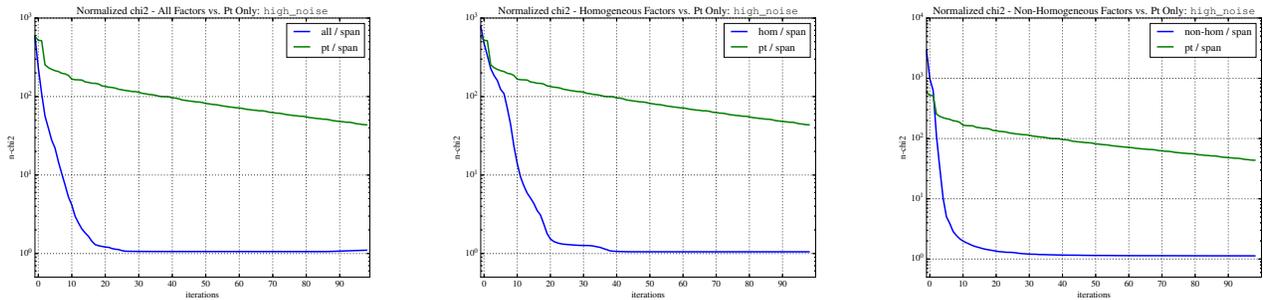
B. Raw Data

Albeit this work focuses on the back-end, we conducted a set of experiments to validate the usability of such a back-end on graphs generated from real data. To this end, we developed a simplistic front-end for RGB-D data. More in detail, we extended [10] by adding to the front-end the option to “export” the graph. Additionally, we implemented a straightforward loop detection on top of the recent Visual Place Recognition (VPR)



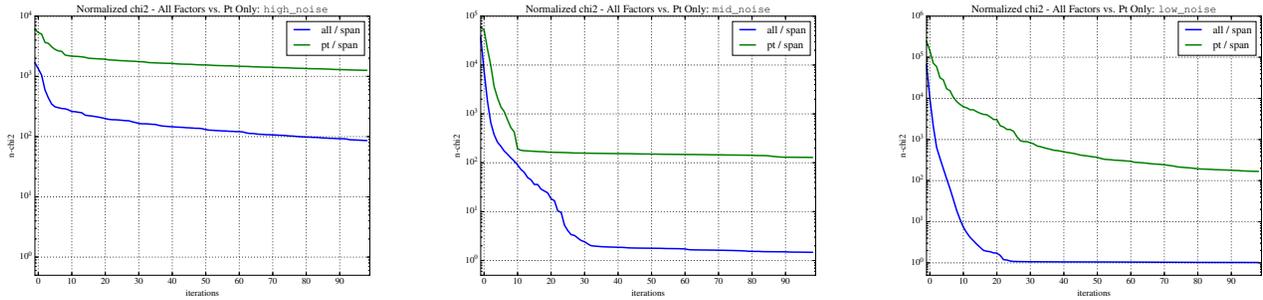
(a) All possible matchable edges compared to point only ones. (b) Homogeneous only matchable edges compared to point only ones. (c) Non-homogeneous only matchable edges compared to point only ones.

Fig. 3: Comparison between point and matchable based landmark graphs, using high-noise configurations. The dataset used are g_0 -all and g_0 -point in Fig. 3a, g_0 -hom and g_0 -point in Fig. 3b, g_0 -non-hom and g_0 -point in Fig. 3c. Minimal mismatch in the converge speed are present when including non homogeneous matchable edges, but they are due to the nature of the geometric constraint that we are considering - e.g. a point can slide on a line. Those effects are due to small dimension of the graph in this case. The chi2 of both approaches are normalized to their respective global optima chi2.



(a) All possible matchable edges compared to point only ones. (b) Homogeneous only matchable edges compared to point only ones. (c) Non-homogeneous only matchable edges compared to point only ones.

Fig. 4: Comparison between point and matchable based landmark graphs, using high-noise configurations. The dataset used are g_1 -all and g_1 -point in Fig. 4a, g_1 -hom and g_1 -point in Fig. 4b, g_1 -non-hom and g_1 -point in Fig. 4c. In this case it is clear how using matchables increases the convergence basin, reaching the optimum in all cases. The chi2 of both approaches are normalized to their respective global optima chi2.



(a) High noise configuration. (b) Mid noise configuration. (c) Low noise configuration.

Fig. 5: Comparison between point and matchable based landmark graphs. All plots refer to the datasets g_2 -all and g_2 -point. The illustrations reports different noise figures, respectively from left to right: high, mid and low noise configuration. The reader might notice how point only graph always fails in finding the optimal solution. The chi2 of both approaches are normalized to their respective global optima chi2.

	lr0	lr1	lr2	lr3	tr0	tr1	tr2	tr3
SA-SHA	0.081070	0.012452	0.018664	0.094783	0.027381	0.019387	0.019523	0.028203
SA-SHAGO	0.031051	0.008600	0.014552	0.068380	0.017060	0.076373	0.013735	0.010791
ProSLAM	0.004481	0.007028	0.011581	0.028300	0.168866	0.065404	0.032352	0.045774
ORB-SLAM 2	0.007354	0.173665	0.019236	0.010037	0.028518	0.041650	0.012535	0.136025

TABLE VI: Absolute trajectory RMSE [m] comparison on the ICL-NUIM dataset.

library HBST [42]. With this simple setup, we build a graph through the front-end. Once the processing of the raw data is terminated, we evaluate offline the trajectory error before and after the global optimization. The former is referred to as

SA-SHA, as in [10]. The optimized version proposed in this paper is labeled as *SA-SHAGO*.

In the remainder of this section, we report the results obtained by processing raw sensor data. The aim of this

paper is to present a new back-end that can process higher dimension landmarks, and not proposing a full SLAM system. Nonetheless, we report the result of two state-of-the-art systems, namely ProSLAM [5] and ORB-SLAM2 [3], to provide the reader with a baseline.

We conducted the experiments on the ICL-NUIM dataset [43], a photorealistic indoor RGB-D dataset. In Tab. VI we provide the results obtained evaluating the absolute trajectory RMSE - in meters - on the generated trajectory before and after the optimization. The gain in terms of accuracy is evident in almost all the datasets, but the sequence *tr1*. In this case, the non optimized trajectory presents already a reasonable accuracy, and the worsening in the absolute error is due to the insertion of wrong closures in the graph.

VI. CONCLUSIONS

In this work, we extended the pose-landmark optimization problem using different geometric primitives as landmarks - e.g. points, lines and planes. All these primitives are formalized with a unified representation based on the concept of degenerate quadrics, called *matchable*. Thanks to this unified representation, we can exploit the greater convergence properties of more descriptive primitives while preserving a concise implementation. Furthermore, this allows to model different observation of the same matchable landmark - e.g. observe a plane as a line. We support our claims of improved converged properties compared to standard point-based graph optimization. To this end we conducted quantitative experiments where we systematically analyze the effects of each class of constraint under varying conditions. We show the applicability of our approach in realistic scenarios by evaluating a full SLAM pipeline for RGB-D data, built on top of our back-end. Despite the front-end of the pipeline is rather simplistic, the greater descriptiveness of more complex geometric primitives yields to results comparable with other state-of-the-art SLAM systems.

REFERENCES

- [1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. **Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age.** *IEEE Trans. on Robotics (TRO)*, 32(6):1309–1332, 2016. 1
- [2] E. Olson, J. Leonard, and S. Teller. **Fast iterative alignment of pose graphs with poor initial estimates.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269. IEEE, 2006. 1, 2
- [3] R. Mur-Artal and J. D Tardós. **Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras.** *IEEE Trans. on Robotics (TRO)*, 33(5):1255–1262, 2017. 1, 8
- [4] R. Wang, M. Schwörer, and D. Cremers. **Stereo dso: Large-scale direct sparse visual odometry with stereo cameras.** In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, volume 42, 2017. 1
- [5] D. Schlegel, M. Colosi, and G. Grisetti. **ProSLAM: Graph SLAM from a Programmer’s Perspective.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 1–9. IEEE, 2018. 1, 8
- [6] C. Kerl, J. Sturm, and D. Cremers. **Dense Visual SLAM for RGB-D Cameras.** In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013. 1
- [7] J. Engel, T. Schöps, and D. Cremers. **LSD-SLAM: Large-Scale Direct Monocular SLAM.** In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, September 2014. 1
- [8] H. Wolfgang, K. Damon, R. Holger, and A. Daniel. **Real-Time Loop Closure in 2D LIDAR SLAM.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 1271–1278, 2016. 1
- [9] M. T. Lzaro, R. Capobianco, and G. Grisetti. **Efficient Long-term Mapping in Dynamic Environments.** In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 153–160, Oct 2018. 1
- [10] F. Nardi, B. Della Corte, and G. Grisetti. **Unified Representation and Registration of Heterogeneous Sets of Geometric Primitives.** *IEEE Robotics and Automation Letters (RA-L)*, 2019. 1, 2, 4, 5, 6, 7
- [11] M. Kaess. **Simultaneous localization and mapping with infinite planes.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, volume 1, page 2, 2015. 1, 2
- [12] J. Zhang and S. Singh. **Visual-lidar odometry and mapping: Low-drift, robust, and fast.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2174–2181. IEEE, 2015. 1
- [13] Y. Lu and D. Song. **Robust rgb-d odometry using point and line features.** In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 3934–3942, 2015. 2
- [14] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. **PL-SLAM: Real-time monocular visual SLAM with points and lines.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 4503–4508. IEEE, 2017. 2
- [15] L. Ma, C. Kerl, J. Stückler, and D. Cremers. **Cpa-slam: Consistent plane-model alignment for direct rgb-d slam.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 1285–1291. IEEE, 2016. 2
- [16] H. Durrant-Whyte and T. Bailey. **Simultaneous localization and mapping: part I.** *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 2
- [17] T. Bailey and H. Durrant-Whyte. **Simultaneous localization and mapping: part II.** *IEEE robotics & automation magazine*, 13(3):108–117, 2006. 2
- [18] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. **A solution to the simultaneous localization and map building (SLAM) problem.** *IEEE Trans. on Robotics (TRO)*, 17(3):229–241, 2001. 2
- [19] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. **FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges.** In *Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156, 2003. 2
- [20] G. Grisetti, C. Stachniss, and W. Burgard. **Improved techniques for grid mapping with rao-blackwellized particle filters.** *IEEE Trans. on Robotics (TRO)*, 23(1):34, 2007. 2
- [21] S. Huang, Y. Lai, U. Frese, and G. Dissanayake. **How far is SLAM from a linear least squares problem?** In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3011–3016, 2010. 2
- [22] F. Lu and E. Milios. **Globally consistent range scan alignment for environment mapping.** *Autonomous robots*, 4(4):333–349, 1997. 2
- [23] J-S Gutmann and K. Konolige. **Incremental mapping of large cyclic environments.** In *Computational Intelligence in Robotics and Automation, 1999. CIRA’99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325. IEEE, 1999. 2
- [24] A. Howard, M. J Mataric, and G. Sukhatme. **Relaxation on a mesh: a formalism for generalized localization.** In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, volume 2, pages 1055–1060. IEEE, 2001. 2
- [25] T. Duckett, S. Marsland, and J. Shapiro. **Fast, on-line learning of globally consistent maps.** *Autonomous Robots*, 12(3):287–300, 2002. 2
- [26] U. Frese, P. Larsson, and T. Duckett. **A multilevel relaxation algorithm for simultaneous localization and mapping.** *IEEE Trans. on Robotics (TRO)*, 21(2):196–207, 2005. 2
- [27] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. **A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent.** In *Proc. of Robotics: Science and Systems (RSS)*, volume 3, page 9, 2007. 2
- [28] I. Aloise and G. Grisetti. **Matrix Difference in Pose-Graph Optimization.** *arXiv preprint arXiv:1809.00952*, 2018. 2, 3
- [29] F. Dellaert and M. Kaess. **Square Root SAM: Simultaneous localization and mapping via square root information smoothing.** *Intl. Journal of Robotics Research (IJRR)*, 25(12):1181–1203, 2006. 2
- [30] M. Kaess, A. Ranganathan, and F. Dellaert. **iSAM: Fast incremental smoothing and mapping with efficient data association.** In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 1670–1677, 2007. 2, 3
- [31] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J Leonard, and F. Dellaert. **iSAM2: Incremental smoothing and mapping using the Bayes tree.** *Intl. Journal of Robotics Research (IJRR)*, 31(2):216–235, 2012. 2

- [32] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. [g2o: A general framework for graph optimization](#). In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3607–3613, 2011. 2, 3
- [33] J. A Castellanos, JMM Montiel, J. Neira, and J. D Tardós. [The SPmap: A probabilistic framework for simultaneous localization and map building](#). *IEEE Transactions on robotics and Automation*, 15(5):948–952, 1999. 2
- [34] D. Rodriguez-Losada, F. Matia, and R. Galan. [Building geometric feature based maps for indoor service robots](#). *Journal on Robotics and Autonomous Systems (RAS)*, 54(7):546–558, 2006. 2
- [35] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart. [A comparison of line extraction algorithms using 2D range data for indoor mobile robotics](#). *Autonomous Robots*, 23(2):97–111, 2007. 2
- [36] N. Aggarwal and W C. Karl. [Line detection in images through regularized Hough transform](#). *IEEE transactions on image processing*, 15(3):582–591, 2006. 2
- [37] L. Pedraza, D. Rodriguez-Losada, F. Matia, G. Dissanayake, and J. V. Miró. [Extending the limits of feature-based SLAM with B-splines](#). *IEEE Trans. on Robotics (TRO)*, 25(2):353–366, 2009. 2
- [38] P. de la Puente and D. Rodriguez-Losada. [Feature based graph SLAM with high level representation using rectangles](#). *Journal on Robotics and Autonomous Systems (RAS)*, 63:80–88, 2015. 2
- [39] R. F Salas-Moreno, B. Glocken, P. HJ Kelly, and A. J Davison. [Dense planar SLAM](#). In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 157–164. IEEE, 2014. 2
- [40] S. Yang, Y. Song, M. Kaess, and S. Scherer. [Pop-up slam: Semantic monocular plane slam for low-texture environments](#). In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1222–1229. IEEE, 2016. 2
- [41] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. [A tutorial on graph-based SLAM](#). *IEEE Trans. on Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010. 3
- [42] D. Schlegel and G. Grisetti. [HBST: A Hamming Distance Embedding Binary Search Tree for Feature-Based Visual Place Recognition](#). *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3741–3748, 2018. 7
- [43] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. [A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM](#). In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, Hong Kong, China, May 2014. 8