Unified Representation of Heterogeneous Sets of Geometric Primitives

Federico Nardi　　Bartolomeo Della Corte　　Giorgio Grisetti

I. Introduction

Acquiring models of the environment from data gathered by a moving sensor is an essential building block of many robotics applications. These "models" should support several tasks such as navigation and manipulation. Common representations of 3D scenes are: voxel maps, octrees [3], implicit surfaces [1], sample based [5], [6] or surface distributions [4]. All representations mentioned above do not exploit any structure in the environment. Exploiting structure to describe a scene has clear advantages on the storage: describing a room as a set of walls and a floor requires far less memory than storing the corresponding point cloud.

In this work we propose a unified representation for hybrid scenes consisting of points, lines, planes and surfels and we refer to these primitives within a scene as "matchables". By using this unified representation we can define correspondences among items in the scene that belong to different classes: besides enforcing that two planes or two points in the scenes are the same, we can also express constraints such as "a point lies on a line", or "a line lies on a plane", as shown in Fig. 1.

A registration approach built upon the presented representation is expected to capture in a uniform manner several ICP variants. This would allow the system to benefit from environmental structure when present, while smoothly degrading to regular ICP [2] when the structure is missing.

II. Unified Scene Representation

A matchable $\mathbf{m}$ is a data-structure that contains the information to represent the geometric primitives previously mentioned, and it is parametrized as follows:

$$\left\langle \mathbf{v_m} = \begin{pmatrix} \mathbf{p_m} \\ \mathbf{d_m} \end{pmatrix}, \mathbf{\Omega_m} \right\rangle. \quad (1)$$

More in detail, $\mathbf{p_m} \in \Re^3$ is the centroid of the primitive, $\mathbf{d_m} \in S(2)$ its direction vector and $\mathbf{\Omega_m} \in \Re^{3\times3}$ a 3D information matrix obtained from $\mathbf{d_m}$ according to Tab. I, note that $\mathbf{R_m}$ is a rotation matrix that applied to the direction vector $\mathbf{d}$, returns the $\mathbf{x}$ axis, i.e., $\begin{pmatrix}1 & 0 & 0\end{pmatrix}^T = \mathbf{R_m d}_m$. Albeit not minimal, this formulation allows to represent points, lines, planes and surfels by using the same set of parameters.

Applying a transformation $\mathbf{X} \in SE(3)$ to a matchable $\mathbf{m}$ results in a new matchable $\mathbf{m}' = \mathbf{X} \cdot \mathbf{m}$ with the following parameters:

Federico, Bartolomeo and Giorgio are with Sapienza University of Rome, Department of Computer, Control, and Management Engineering Antonio Ruberti, Rome, Italy.

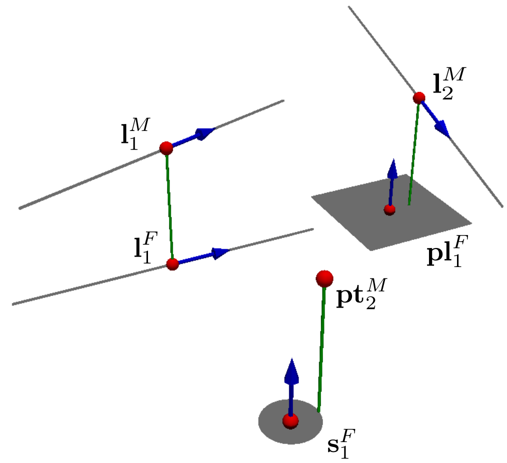


Fig. 1: Two scenes, moving (**M**) and fixed (**F**)), with different types of matchables: points (**pt**), lines (**l**), planes (**pl**) and surfels (**s**).

$$\mathbf{v}' = \begin{pmatrix} \mathbf{R p_m} + \mathbf{t} \\ \mathbf{R d_m} \end{pmatrix} \qquad \mathbf{\Omega}'_\mathbf{m} = \mathbf{R \Omega_m R}^T \quad (2)$$

A straightforward way to compute a 6D vector difference between matchables consists in stacking the difference between the corresponding points $\mathbf{e}_\mathrm{p} = \mathbf{p_m} - \mathbf{p}'_\mathbf{m}$ and direction vectors $\mathbf{e}_\mathrm{d} = \mathbf{d_m} - \mathbf{d}'_\mathbf{m}$. Extending this difference vector with an additional component $e_o = \mathbf{d}_\mathbf{m}^T \mathbf{d}'_\mathbf{m}$ allows us to capture the orthogonality between two direction vectors. Accordingly, we define the difference between two matchables as the following 7D vector:

$$\mathbf{e}(\mathbf{v_m}, \mathbf{v}'_\mathbf{m}) = \begin{pmatrix} \mathbf{e}_\mathrm{p} \\ \mathbf{e}_\mathrm{d} \\ e_\mathrm{o} \end{pmatrix} = \begin{pmatrix} \mathbf{p_m} - \mathbf{p}'_\mathbf{m} \\ \mathbf{d_m} - \mathbf{d}'_\mathbf{m} \\ \mathbf{d}_\mathbf{m}^T \mathbf{d}'_\mathbf{m} \end{pmatrix} \quad (3)$$

The difference vector is not necessarily zero even for identical primitives since the representation is not minimal. To carry on the minimization we need to define a metric that is immune to this issue. To this extent, we construct a *distance* metric that is zero whenever a constraint is satisfied.

A distance $e(\mathbf{m}, \mathbf{m}')$ between matchables is a non-negative scalar computed from the difference vector. If the distance is null, the constraint between the two matchables is satisfied. To compute the distance, we employ adapted Ω-norm to difference vector: as follows $e(\mathbf{m}, \mathbf{m}') = \|\mathbf{e}(\mathbf{m}, \mathbf{m}')\|^2_{\mathbf{\Omega}(\mathbf{m},\mathbf{m}')}$. The information matrix $\mathbf{\Omega}(\mathbf{m}, \mathbf{m}') \in \Re^{7\times7}$ activates the appropriate components of the difference vector during the minimization, based on

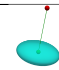
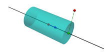
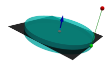
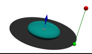
type	$\mathbf{p_m}$	$\mathbf{d_m}$	$\mathbf{\Omega_m}$	Shape of $\mathbf{\Omega_m}$
point	$\mathbf{p}$	$\mathbf{0}$	$\mathbf{I}_{3\times 3}$	
line	$\mathbf{p_l}$	$\mathbf{d_l}$	$\mathbf{R_m}\mathrm{diag}(0,1,1)\mathbf{R}_\mathbf{m}^T$	
plane	$\mathbf{p}_\pi$	$\mathbf{d}_\pi$	$\mathbf{R_m}\mathrm{diag}(1,0,0)\mathbf{R}_\mathbf{m}^T$	
surfel	$\mathbf{p_s}$	$\mathbf{d_s}$	$\mathbf{R_m}\mathrm{diag}(1,\epsilon,\epsilon)\mathbf{R}_\mathbf{m}^T$	

TABLE I: The shape of $\mathbf{\Omega_m}$ discriminates the type of primitive represented by the matchable. The confidence ellipsoid obtained from $\mathbf{\Omega_m}$ is a sphere if the matchable is a point, or it has a disk-like shape with $\mathbf{d_m}$ as direction of minimal variation in case of a sufel. If the primitive is a line or a plane the confidence ellipsoid degenerates respectively to a cylinder oriented as $\mathbf{d_m}$ or to two parallel planes having normal $\mathbf{d_m}$.

		$\mathbf{m}'$			
		point	**line**	**plane**	**surfel**
$\mathbf{m}$	**point**	$\mathbf{\Omega}_\mathrm{p}=\mathbf{I}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$
	line	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{I}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=1$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=1$
	plane	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=1$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{I}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{I}$ $\Omega_\mathrm{o}=0$
	surfel	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{0}$ $\Omega_\mathrm{o}=1$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{I}$ $\Omega_\mathrm{o}=0$	$\mathbf{\Omega}_\mathrm{p}=\mathbf{\Omega}'_\mathbf{m}$ $\mathbf{\Omega}_\mathrm{d}=\mathbf{I}$ $\Omega_\mathrm{o}=0$

TABLE II: Information Matrix $\mathbf{\Omega}(\mathbf{m},\mathbf{m}')$, green: $\mathbf{\Omega}_\mathrm{p}$ does not depend on $\mathbf{m}$, pink: $\mathbf{\Omega}_\mathrm{p}$ depends on $\mathbf{m}$.

the type of constraint, according to Tab. II. We enforce the following block diagonal structure for $\mathbf{\Omega}(\mathbf{m},\mathbf{m}')$:

$$\mathbf{\Omega}(\mathbf{m},\mathbf{m}') = \begin{pmatrix} \mathbf{\Omega}_\mathrm{p} & 0 & 0 \\ 0 & \mathbf{\Omega}_\mathrm{d} & 0 \\ 0 & 0 & \Omega_\mathrm{o} \end{pmatrix} \quad (4)$$

With this formulation, the generic distance between two matchables is computed as

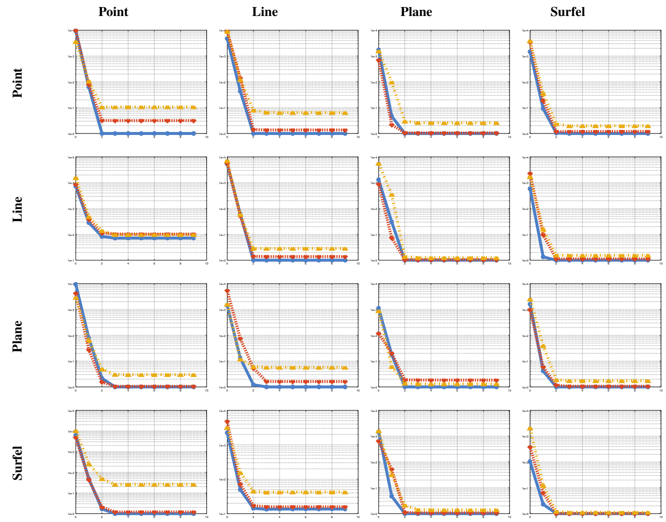
$$\begin{aligned} e(\mathbf{m},\mathbf{m}') &= \|\mathbf{e}(\mathbf{v_m},\mathbf{v}'_\mathbf{m})\|^2_{\mathbf{\Omega}(\mathbf{m},\mathbf{m}')} \quad (5) \\ &= \|\mathbf{e}_\mathrm{p}\|^2_{\mathbf{\Omega}_\mathrm{p}} + \|\mathbf{e}_\mathrm{d}\|^2_{\mathbf{\Omega}_\mathrm{d}} + \|e_\mathrm{o}\|^2_{\Omega_\mathrm{o}} \end{aligned}$$

Having defined how to transform the matchables and a way to compute how "far" is a constraint from being satisfied, the described representation is suitable to be used within a registration context by seeking for the optimal transformation $\mathbf{X}^*$ that better alignes two scenes:

$$\begin{aligned} \mathbf{X}^* &= \underset{\mathbf{X}}{\operatorname{argmin}} \sum e(\mathbf{X}\mathbf{m}_k,\mathbf{m}'_k) \\ &= \underset{\mathbf{X}}{\operatorname{argmin}} \sum \|\mathbf{e}(\mathbf{X}\mathbf{m}_k,\mathbf{m}'_k)^T\|^2_{\mathbf{\Omega}(\mathbf{X}\mathbf{m}_k,\mathbf{m}'_k)} \quad (6) \end{aligned}$$

where the matchables of the "moving" scene $\mathbf{m}_{1:K}$ are linked to the matchables of the "fixed" scene $\mathbf{m}'_{1:K}$.

TABLE III: Synthetic Experiments - Error Evolution. In *solid blue* is shown the evolution of the error without noise, while in *dashed red* is reported the low-noise case and in *point-dashed yellow* the high-noise case.



III. Results

To validate the proposed representation in a registration context, we designed an experiment with synthetic data, isolating the effect of homogeneous and heterogeneous constraints. For each constraint type, we build an optimization problem with 10 couples of matchable in three cases: *no-noise*, *low-noise* $[\sigma_t^2 \sim 0.05, \sigma_r^2 \sim 0.025]$ and *high-noise* $[\sigma_t^2 \sim 0.1, \sigma_r^2 \sim 0.05]$, where the noise is applied to the measurements. In Tab. III we report the error evolution of the Gauss-Newton iterative solver for 10 iterations, under different levels of noise. In all cases the solver was able to find the correct minimum.

References

[1] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, , P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[2] PJ Besl and Neil D McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[3] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[4] M. Magnusson, T. Duckett, and A. J. Lilienthal. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal on Field Robotics*, 2007.

[5] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1–4, May 2011.

[6] J. Serafin and G. Grisetti. Nicp: Dense normal based point cloud registration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 742–749, Sept 2015.