Supplementary Material Planar Shape Based Registration for Multi-modal Geometry

Muxingzi Li
muxingzi.li@inria.fr
Florent Lafarge
florent.lafarge@inria.fr
Inria, Université Côte d'Azur
Sophia Antipolis, France

Scale estimation for two-step baselines

Existing two-step pipelines for model-to-scene alignment has shown that bounding boxes provide a good estimate for the relative scale between objects with sufficient overlap [3], which can be used for a later 6DoF alignment step [5, 7]. We adopt the bounding box based method for estimating a relative scale, which consists of an outlier filtering step and a bounding box estimation step. Outlier filtering is done with the KNN algorithm as the following: First compute for each point the mean distance to its K nearest neighbors. Average KNN distances for all points give a distribution with a sample mean μ and sample standard deviation σ. All points with a mean KNN distance greater than $\mu + 2\sigma$ are classified as outliers, which approximately corresponds to an outlier ratio of 2.3% under the normal distribution assumption. Optimal bounding boxes are estimated using the algorithm of Chang et al.[1]. Let B_d and B_m denote the bounding boxes of the source and the target, respectively. The scale is set as the ratio between the lengths of the bounding box diagonals, $s_{\text{init}} = \text{Diag}(B_m)/\text{Diag}(B_d)$.

Candidate rotations after alignment

As discussed in section 3.1, the output of the rotational initialization is a list of 24 rotation matrices $\{R_i\}_{i=1}^{24}$. This is done by aligning the 3-frames with all permutations of the three axes (3! in total) and combinations of the orientation of each axis (2^3 in total). Simple application of the right-hand rule (or left-hand rule) can eliminate half of the candidates, giving a final number of 24. Fig. 1 shows the complete list of rotational initialization.

EM algorithm for estimating 3-frames

Notations. Let:

- $\mathbf{n}_i \in \mathbb{R}^3$ be the surface normal of the i-th planar shape.
- $\mathbf{N} = \{\mathbf{n}_1, ..., \mathbf{n}_m\}$ be the set of surface normals.
- $a_i \in \mathbb{R}$ be the surface area of the i-th planar shape.

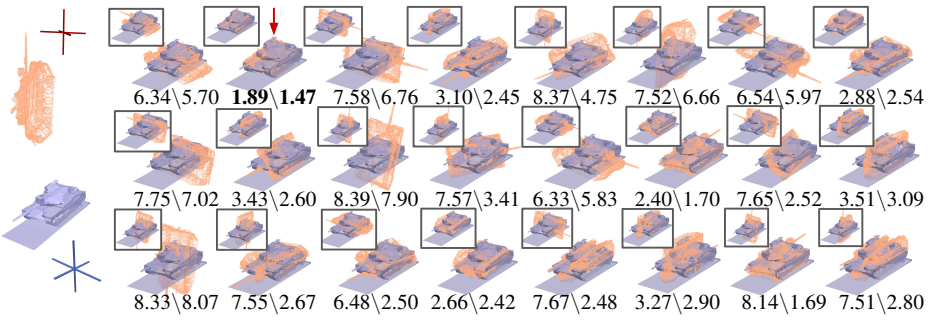

Figure 1: Demonstration of the 24 initial rotations and the corresponding output after refinement. The input point cloud and the mesh are shown on the left, together with their 3-frames (red and blue, resp.). For each initial rotation, the alignment after applying the rotation matrix is shown on the top-left frame, and the refinement output is at the bottom right. The loss before and after refinement is shown below. The best path is indicated by a red arrow.

- $w_i = f(a_i) \in \mathbb{R}$ be a non-negative weight assigned to the i-th planar shape. In this thesis, we simply define the function f as $f : x \in \mathbb{R} \mapsto x \in \mathbb{R}$ such that the weight is proportional to the shape area.

- $\mathbf{W} = \{w_1, ..., w_m\}$ be the set of weights.

- $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ be the parameters defining the k-th Gaussian component, with mean $\boldsymbol{\mu}_k \in \mathbb{R}^3$ and covariance $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3\times3}$.

- $p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{N}_g(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ be the probability density of a multivariate Gaussian distribution at $\mathbf{x}$, under a distance metric defined by the function $g : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$.

- $\{\pi_1, ..., \pi_K\}$ be mixture coefficients, where $\pi_k \leq 0$ is the coefficient for the k-th Gaussian component, satisfying $\sum_{k=1}^{K} \pi_k = 1$.

The algorithm clusters the surface-normals of a set of planar shapes into 3 clusters, under the Gaussian mixture model assumption for weighted data. The distance metric is the absolute cosine similarity metric for measuring the distance between two normal vectors on the spherical surface. The pseudo-code is given in Algorithm 1.

Derivation for continuous optimization

Transformations in 3D space can be represented as Lie groups [8, 11]. In particular, the similarity transform matrix belongs to the Sim(3) Lie group. The associated Lie algebra sim(3) of the Sim(3) group is a vector space of dimension 7. The basis element of the Lie algebra are called generators. The generators of sim(3) correspond to differential translations (G_1, G_2, G_3), derivatives of rotation around the each of the standard axes (G_4, G_5, G_6), and

Algorithm 1 Weighted-data clustering with the EM algorithm

Input: $\mathbf{N} = \{\mathbf{n}_1, ..., \mathbf{n}_m\}$, $\mathbf{W} = \{w_1, ..., w_m\}$
Output: $\{\pi_1, ..., \pi_3\}$, $\{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3\}$

Initialize π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ for all k
while not converging **do**

$$p(z_i = k|\mathbf{n}_i; \boldsymbol{\theta}, w_i) := \frac{\pi_k \mathcal{N}_g(\mathbf{n}_i; \boldsymbol{\theta}_k, \frac{1}{w_i}\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}_g(\mathbf{n}_i; \boldsymbol{\theta}_j, \frac{1}{w_i}\boldsymbol{\Sigma}_j)}$$

$$\pi_k := \frac{1}{m}\sum_{i=1}^{m} p(z_i = k|\mathbf{n}_i; \boldsymbol{\theta}, w_i)$$

$$\boldsymbol{\mu}_k := \frac{\sum_{i=1}^{m} w_i p(z_i=k|\mathbf{n}_i;\boldsymbol{\theta},w_i)\mathbf{n}_i}{\sum_{i=1}^{m} w_i p(z_i=k|\mathbf{n}_i;\boldsymbol{\theta},w_i)}$$

Normalize $\boldsymbol{\mu}_k$

$$\boldsymbol{\Sigma}_k := \frac{\sum_{i=1}^{m} w_i p(z_i=k|\mathbf{n}_i;\boldsymbol{\theta},w_i)(\mathbf{n}_i-\boldsymbol{\mu}_k)(\mathbf{n}_i-\boldsymbol{\mu}_k)^T}{\sum_{i=1}^{m} p(z_i=k|\mathbf{n}_i;\boldsymbol{\theta},w_i)}$$

end while

the derivative of scale change (G_7):

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$G_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, G_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{1}$$

$$G_7 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

An element of sim(3) is represented as a linear combination of generators,

$$u_1 G_1 + u_2 G_2 + u_3 G_3 + \omega_1 G_4 + \omega_2 G_5 + \omega_3 G_6 + \lambda G_7 \in \text{sim}(3) \quad \text{where } (\boldsymbol{u};\ \boldsymbol{\omega};\ \lambda) \in \mathbb{R}^7. \tag{2}$$

We denote $\boldsymbol{\xi} = (\boldsymbol{u};\ \boldsymbol{\omega};\ \lambda) \in \text{sim}(3)$ for convenience. The Lie algebra is linked to the Lie group by an exponential map which is a bijective function converting any element in the tangent space to exactly one transformation in the group. Specifically, the exponential map from sim(3) to Sim(3) is the matrix exponential given by

$$\begin{aligned} \exp(\boldsymbol{\xi}) &= \exp\begin{bmatrix} \boldsymbol{\omega}^\wedge & \boldsymbol{u} \\ \mathbf{0}^T & -\lambda \end{bmatrix} \\ &= \mathbf{I} + \begin{bmatrix} \boldsymbol{\omega}^\wedge & \boldsymbol{u} \\ \mathbf{0}^T & -\lambda \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} (\boldsymbol{\omega}^\wedge)^2 & \boldsymbol{\omega}^\wedge\boldsymbol{u} - \lambda\boldsymbol{u} \\ \mathbf{0}^T & \lambda^2 \end{bmatrix} + ... \\ &= \begin{bmatrix} \exp(\boldsymbol{\omega}^\wedge) & \boldsymbol{v} \\ \mathbf{0}^T & \exp(-\lambda) \end{bmatrix} \in \text{Sim}(3) \end{aligned} \tag{3}$$

where $\boldsymbol{\omega}^\wedge$ is the skew-symmetric matrix associated to vector $\boldsymbol{\omega}$, $\exp(\boldsymbol{\omega}^\wedge)$ is the rotation matrix, $\boldsymbol{\nu}$ denotes the translation component and $\exp(-\lambda)$ is the scale change. For $S \in \mathrm{Sim}(3)$ and a point $\mathbf{p} \in \mathbb{R}^3$, differentiation of $S\mathbf{p}$ by $\boldsymbol{\xi}$ is performed by implicitly left multiplying the transformation by the generators representing infinitesimal perturbations.

Qualitative analysis

Fig. 2 visualizes the registration results of each method on two examples (a free-form object and a regular object). The input point clouds are significantly perturbed from the ground-truth alignment. For local methods, we show both results with and without an FPFH-based RANSAC initialization. Classical feature-based global methods like FGR do not offer enough robustness to variations in the shape characteristics. On the free-form object (top), the family of probabilistic approaches (CPD, BCPD++, and FilterReg) benefits more from the RANSAC initialization than the ICP approaches. Point-based learning method DCP fails to register point clouds with large numbers of points, due to its high memory consumption. Volumetric learning-based approach DGR is able to produce more accurate correspondences, but requires accurate estimation of the relative scale as a pre-processing step. Our method recovers well the alignment for both types of objects, achieving satisfying accuracy.

Fig. 3 visualizes our results on all 19 pairs of objects in the dataset.

References

[1] Chia-Tche Chang, Bastien Gorissen, and Samuel Melchior. Fast oriented bounding box optimization on the rotation group so(3,r). *ACM Transactions on Graphics*, 30(5), 2011.

[2] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020.

[3] Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, Riccardo Gherardi, Andrea Fusiello, and Roberto Scopigno. Fully automatic registration of image sets on approximate geometry. *IJCV*, 102, 2013.

[4] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *CVPR*, 2019.

[5] Saurabh Gupta, Pablo AndrÃl's ArbelÃąez, Ross B. Girshick, and Jitendra Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *CVPR*, 2015.

[6] Osamu Hirose. Acceleration of non-rigid point set registration with downsampling and gaussian process regression. *TPAMI*, 2021.

[7] Hamid Izadinia and Steven M. Seitz. Scene recomposition by learning-based icp. In *CVPR*, 2020.

[8] Alexander Kirillov. *An Introduction to Lie Groups and Lie Algebras*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2008.

[9] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *TPAMI*, 32(12), 2010.

[10] Szymon Rusinkiewicz. A symmetric objective function for ICP. *ACM Transactions on Graphics*, 38(4), 2019.

[11] J. M. Selig. Lie groups and lie algebras in robotics. In Jim Byrnes, editor, *Computational Noncommutative Algebra and Applications*, pages 101–125. Springer Netherlands, 2004.

[12] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019.

[13] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *ECCV*, 2016.

[14] Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. Point set registration with integrated scale estimation. In *International Conference on Pattern Recognition and Image Processing*, 2005.

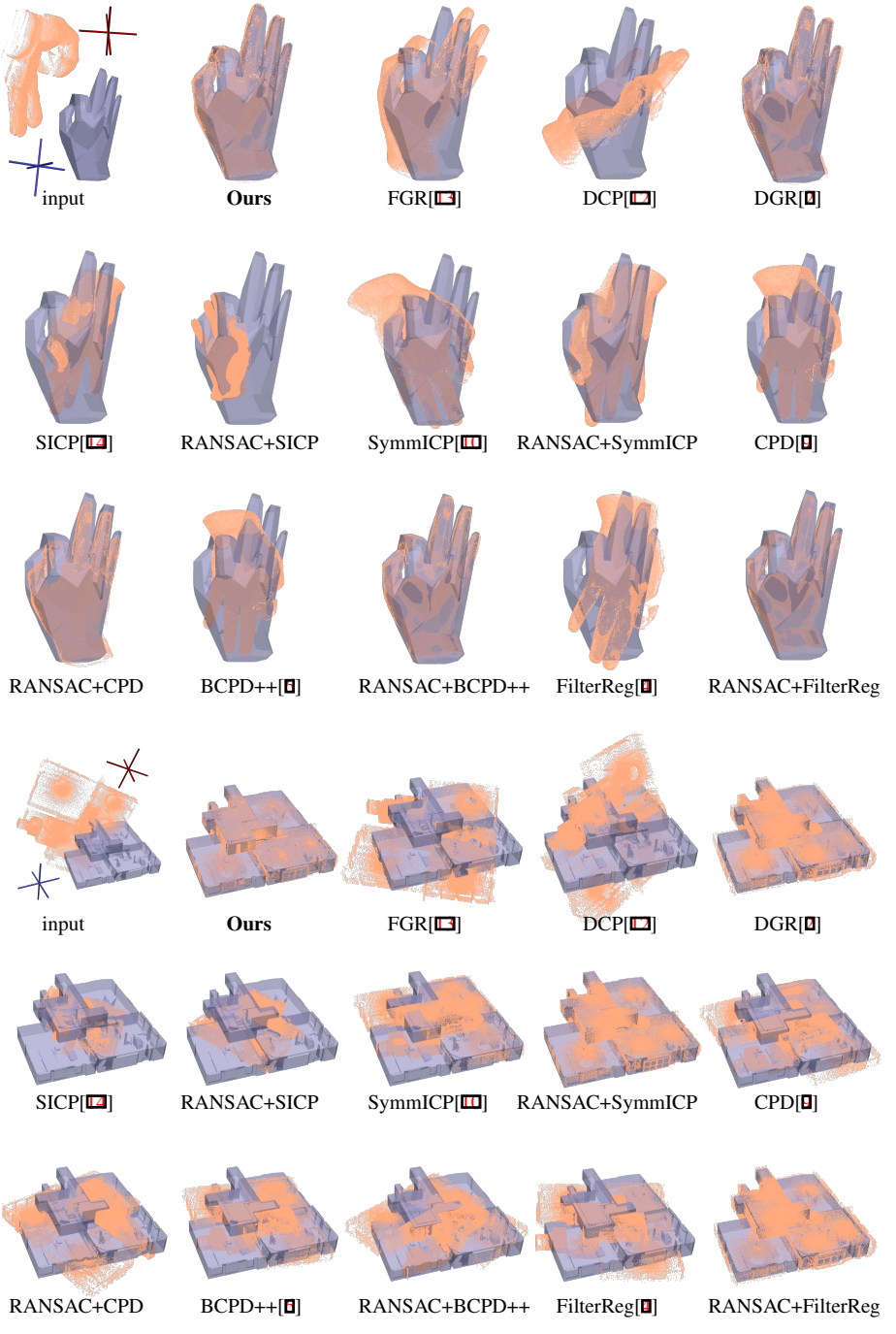

Figure 2: Visualization of results registered by each method on a free-form object (top) and a regular object (bottom). The 3-frames of the input point cloud and the mesh are shown (red and blue, respectively). Our method achieves satisfying alignment for both types of objects under large transformation.

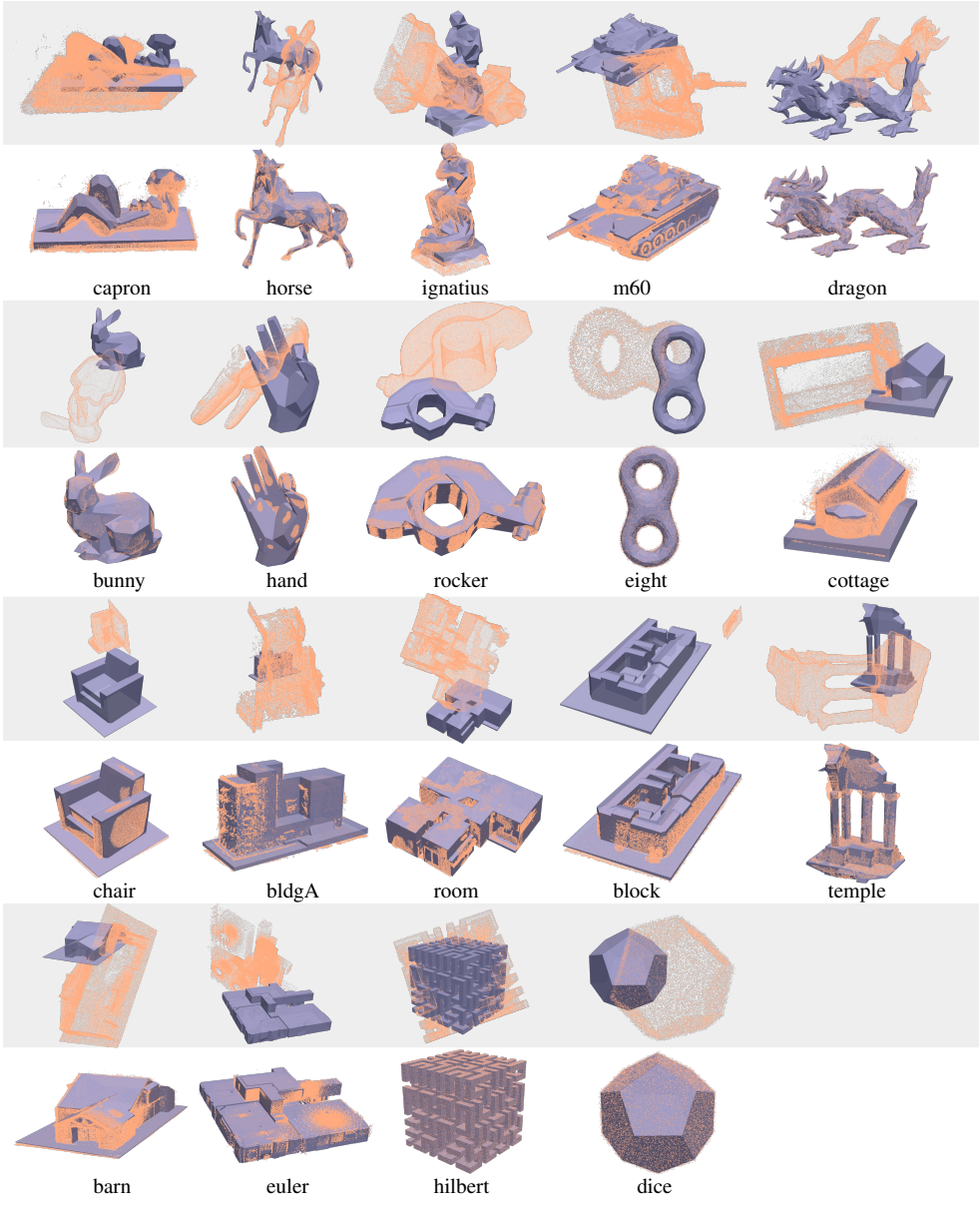

Figure 3: Visualization of our results on 19 examples from the dataset, with random perturbations in rotation between 90° and180 °, in translation between 0 and 100% of the diameter, and in scaling between 0.25 and 2. For each example, the input is shown on the top (gray region), with the aligned result on the bottom.