On Reproducibility & Performance: an Addendum to *Symmetry and Orbit Detection via Lie-Algebra Voting*

Zeyun Shi[1] Pierre Alliez[2] Mathieu Desbrun[2,3] Hujun Bao[1] Jin Huang[1*]

[1]Zhejiang University [2]Inria [3]Caltech

Symposium of Geometry Processing, 2016

1 Introduction

In this short note, we demonstrate that despite the fact that there are a few parameters involved in our algorithm to detect symmetries and orbits, these parameters do not need to be tuned—or can be adjusted easily.

2 Implementation details

Implementation details are provided in the main paper, but we mention a few more details here for completeness and reproducibility.

1. **Sampling.** We uniformly sample a set of points P on the model with a default size of $|P| = 1000$. Around each sample $p_i \in P$, we extract its neighboring patch $\mathcal{C}(p_i)$ based on a radius r, set by default to 3 times the minimum distance between samples.

2. **Analysis.** A local frame is constructed for each point $p_i \in P$ based on local ($\mathcal{C}(p_i)$) estimation of normal, principal curvatures ($\lambda_{\min}, \lambda_{\max}$), and principal curvature directions.

3. **Sample pruning.** A sample for which $\lambda_{\min} = \lambda_{\max}$ is invariant under rotations around its normal, thus a point pair containing such a sample can not define a unique transformation. We reject this kind of samples by using a threshold θ on the ratios of curvatures by requiring $|\frac{\lambda_{\min}}{\lambda_{\max}}| < \theta$, where θ is set to 0.9 by default (instead of 0.75 in [Mitra et al., 2006]). The remaining set of samples forms the set $\bar{P}$.

4. **Pairing.** From a random subset $P' \subset \bar{P}$ with $|P'| = |P|/5$, a transformation T_{ij} is then computed for each pair $p_i \in P'$, $p_j \in \bar{P}$ that aligns their local frames. Since T_{ij} has two possible values (because the principal curvature direction is a "two-rotational" symmetry vector), we chose the one with the smaller alignment error between $\mathcal{C}(p_i)$ and $\mathcal{C}(p_j)$ (in terms of the average distance between the closest point pairs). Default size of P' is 200.

5. **Pair pruning.** Not all sample pairs give reliable transformation for symmetry (and orbit) detection. By definition, a transformation T_{ij} generated from pair p_i, p_j is likely to be a candidate symmetry only if regions surrounding p_i and regions surrounding p_j match. Thus, we use a threshold ϵ on alignment error between $\mathcal{C}(p_i)$ and $\mathcal{C}(p_j)$ to prune sample pairs (and their corresponding transformations), the unit of ϵ is also the minimum distance between samples. (In [Mitra et al., 2006], pair pruning is done by picking pairing samples close enough in signature space, where the signature of a sample is its principal curvature ratio.)

6. **Clustering.** After removing unsuitable pairs, the remaining set of transformations is T. For symmetry detection, mean shift clustering is performed using Gaussian kernel with kernel size δ_m, while for orbit detection, RANSAC is used with model fitting threshold δ_r. Both δ_m and δ_r are set by an initial calibration process: we randomly pick a small number (1000 by default) of pairs of transformations from T and evaluate their variational distance to estimate the average distance $\bar{\delta}$. Finally, δ_m and δ_r is set to $0.05\bar{\delta}$ by default.

In Table 1, we list the parameters used in our tests and figures. Notice that most of the parameters take default values, and only the alignment error threshold ϵ is sometimes tuned. In practice, we first select $\epsilon = 0.1$, then gradually increase it if no prominent cluster appears after clustering (for example, the *Thai status* model, which contains many fine details, has few sample pairs that could be accurately aligned); or inversely, if the most prominent clusters correspond to inaccurate or meaningless symmetries (or orbits), we gradually decrease ϵ (for example, the *tower* model, since it contains many small components). We also used a larger size of P for the *Lamp* model as it contains a long and thin supporting holder which interfere with the sampling density of the bulbs. We used a larger size of P' for *indoor scene*, as it contains many potential symmetries and orbits.

*Corresponding author: hj@cad.zju.edu.cn

Model	Vertices	$\|P\|$	r	θ	$\|\bar{P}\|$	$\|P'\|$	ϵ	$\|T\|$	δ_m and δ_r
Church with side railing	178394	default	default	default	813	default	0.1	920	default
Lego brick	10981	default	default	default	932	default	0.2	2444	default
Lamp	16000	2×default	default	default	1789	default	0.2	1576	default
Sydney Opera House (partial)	15670	default	default	default	960	default	0.1	223	default
Man	20000	default	default	default	990	default	0.2	1056	default
Elk	9552	default	default	default	712	default	0.2	1390	default
Bunny	5000	default	default	default	970	default	0.2	380	default
Bumpy torus	16815	default	default	default	889	default	0.2	767	default
Thai status	30000	default	default	default	955	default	0.3	1621	default
Workpiece	100000	default	default	default	779	default	0.2	1183	default
Fire hydrant	25000	default	default	default	913	default	0.1	1130	default
Octopus	30000	default	default	default	985	default	0.1	339	default
Carter	10000	default	default	default	901	default	0.2	610	default
Nautilus	29997	default	default	default	900	default	0.1	382	default
Filigree	50000	default	default	default	954	default	0.1	205	default
Tower	107736	default	default	default	767	default	0.05	3386	default
Indoor scene	240861	default	default	default	901	2×default	0.1	14096	default
Teapot	11144	default	default	default	971	default	0.2	368	default

Table 1: Statistic analysis of the parameters used in our method.

Model	Vertices	Clustering (Lie algebra)	Clustering (R^7)
Lamp	16000	0.354s	0.347s
Sydney Opera House (partial)	15670	0.190s	0.190s
Man	20000	0.228s	0.227s
Bunny	5000	0.188s	0.208s
Elk	9552	0.238s	0.363s
Bumpy torus	16815	0.214s	0.218s
Thai status	30000	0.403s	0.437s

Table 2: Timings in seconds on a 3.5 GHz Xeon E5 with 16GBytes main memory.

3 Performance

Our adjoint invariant distance is more complex than the Euclidean distance, hence it could appear marginally more time consuming. However, this is not correct: in fact, the clustering step (logarithm mapping with adjoint invariant distance) will often be faster and/or more efficient at finding clusters than if one uses the R^7 mapping with Euclidean distance [Mitra et al., 2006]. As shown in Table 2, our method is thus at least comparable with R^7 method in terms of speed: as discussed in Sec. 5.1 and Sec. 5.2 in our paper, this is related to the better spatial distribution of transformations with our logarithm mapping. Moreover, recall that our method does find symmetries *without having to test various positions*, so performance judged from a user perspective is far superior to existing symmetry detection methods.

References

[Mitra et al., 2006] Mitra, N. J., Guibas, L., and Pauly, M. (2006). Partial and approximate symmetry detection for 3D geometry. *ACM Trans. Graph.*, 25(3):560–568.