Symmetry and Orbit Detection via Lie-Algebra Voting

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

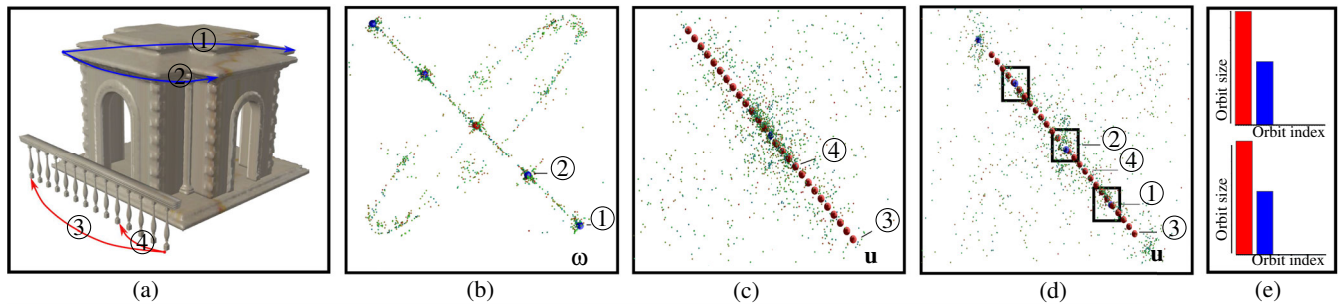[1]Zhejiang University   [2]Inria   [3]Caltech

Figure 1: *Adjoint invariant distance to orbit. Our Lie algebra voting approach to symmetry and orbit detection maps $SE(3)$ transformations into points in a logarithmic space composed of a rotation part $\omega \in \mathbb{R}^3$ and a translation part $\mathbf{u} \in \mathbb{R}^3$. The rotational orbit of the church and the translational orbit of the side railing (a) are mapped into collinear blue and red spheres respectively (a few transformations within these two orbits are marked with circled numbers to enhance comprehension). When the scene is centered, the two lines are orthogonal to each other and easy to distinguish (b). However, after a rigid translation of the scene, the rotational orbit now has $\mathbf{u}$-values near the translation orbit points, making it impossible to automatically distinguish these two orbits using a Euclidean distance (d), while our adjoint invariant distance for orbit shows no discernible difference in results as evidenced by a binning of detected orbit sizes for both situations (e).*

Abstract

In this paper, we formulate an automatic approach to the detection of partial, local, and global symmetries and orbits in arbitrary 3D datasets. We improve upon existing voting-based symmetry detection techniques by leveraging the Lie group structure of geometric transformations. In particular, we introduce a logarithmic mapping that ensures that orbits are mapped to linear subspaces, hence unifying and extending many existing mappings in a single Lie-algebra voting formulation. Compared to previous work, our resulting method offers significantly improved robustness as it guarantees that our symmetry detection of an input model is frame, scale, and reflection invariant. As a consequence, we demonstrate that our approach efficiently and reliably discovers symmetries and orbits of geometric datasets without requiring heavy parameter tuning.

1 Introduction

Symmetry plays a central role in geometry. Whether they be in natural or man-made shapes or scenes, symmetric features represent visually defining properties that are easily recognizable, and can thus be exploited for semantic classification. Detecting these (global or local) symmetries is thus a core task in many applications such as modeling, reconstruction, and segmentation as it helps increasing robustness of further processing.

Previous work. The importance of symmetry in geometry processing has resulted in a variety of approaches to detect near-

symmetries in geometric datasets—a comprehensive review is provided in [MPWC13]. Symmetry detection methods can be categorized as intrinsic or extrinsic. While intrinsic symmetry detection uses geodesic distances and considers isometric deformation as in [LTSW09, XZT*09, JXCZ13], extrinsic symmetry detection typically looks for rigid transformations instead. We restrict our exposition to methods detecting partial, approximate and extrinsic symmetry in 3D meshes.

A number of methods have been proposed to identify the presence of global symmetries, based for instance on spectral analysis [BAK10, LCDF10], feature graph matching [BBW*08, BBW*09, THW*14], descriptors [KFR04], or symmetry transforms [PSG*06]. While a few of these methods can be used

---

to detect local symmetries as well, finding partial symmetry within an input geometry is most efficiently achieved through voting [MGP06], by proceeding as follows. After an initial probing of the geometry that finds pairs of small, similar regions, a series of candidate transformations between each pair of neighborhoods are mapped as points in a high-dimensional space—typically $\mathbb{R}^7$ to encode the Euler angles, translations, and scale factor of each transformation. Clustering in this transformation domain is then performed to find high-density regions, which are evidence of potential symmetries. Depending on the type of symmetry that one wishes to look for, various mappings to the transformation domain can be designed: any parameterization of the relevant symmetry group can be used, and clustering is performed in this parameter domain identified with $\mathbb{R}^n$, so that a notion of distance between transformations can be trivially evaluated using the Euclidean metric. This voting framework is intuitive and flexible, and has been shown to properly confirm a number of important symmetries.

Finding orbits, i.e., regular structures generated by combining k transformations, is more challenging, and as such, it has been significantly less studied. Translational orbits, both discrete and continuous, were discussed in the context of pattern-aware shape deformation [BWKS11]. Pauly et al. [PMW*08] demonstrated that the voting framework can handle orbit detection too: with a proper choice of the map to the transformation domain, orbits are identifiable as lattice structures. Therefore, a more thorough analysis of the geometric configuration of samples in the transformation domain can detect more than just partial symmetries. However, specific maps are required to recognize specific orbits, stressing the importance of the choice of transformation domain.

Symmetry detection for urban scenes and building facades have also been sought after. Methods for this purpose often make assumptions on the orientation of the scene and of the facades. For example, in [KBWS] an urban scene is considered to have consistent upward orientation, leading to a use of orientation histogram descriptors to detect symmetry candidates. In [ZXJ*13], atomic elements of facade images are extracted via box abstraction, before being hierarchically decomposed by splitting and layering of facade structures. Instead, our approach detects local (even approximate) symmetries with no prior knowledge on the original position and orientation of the data to be robust to arbitrary inputs.

Current limitations. Despite the success of voting methods to find symmetries and orbits, we point out that this family of approaches suffers from a number of important shortcomings that limit the practical detection of shape symmetries. For instance, they rely on a *customized* mapping function from the space of transformations to a linear space based on which symmetries (or composition thereof) are looked for. Repeated mappings and clustering may thus be needed to detect a complete set of relevant symmetries. This lack of a unified transformation mapping also results in a need for various clustering methods to detect the presence of features such as concentration points or lattices; note that most of these clustering methods require parameter tuning of their own to perform efficiently and reliably. Finally, the notion of distance with which clustering is achieved is often coordinate system dependent, i.e., the symmetry and orbit detection can be significantly impacted by the translation, rotation, or scaling of the input geometry. This problem

(detailed in Sec. 4.1) is particularly undesirable since what we call a symmetry in Euclidean space is, by very definition, frame and scale invariant. In practice, these limitations imply that one can only find symmetries reliably when one already *knows* they are present in the input geometry since manual tuning is necessary (see, e.g., Fig. 6). With all these issues, robust, "*sight-unseen*" symmetry discovery of geometric datasets is still a challenging task.

Contributions. We propose a simple approach to (partial, local and global) symmetry and orbit detection for 3D datasets, which unifies all existing voting-based techniques into a single, versatile Lie-algebra voting framework. Using the group structure of transformations that are combinations of rigid motions and scaling, we advocate the use of a logarithmic map to naturally encode symmetries and orbits as linear subspaces of the transformation domain. We also identify the importance of using an adjoint-invariant metric on the resulting (Lie algebra) space to enable robust, coordinate frame independent clustering of linear subspaces from a series of sampled transformations obtained through probing of the input geometry. Our group theoretic approach helps discover, systematically and automatically, relevant symmetry and orbit in arbitrary geometric datasets by removing many of the limitations of current methods.

2 Background

We begin our exposition by reviewing concepts and mathematical definitions which will be at the core of our contribution.

2.1 Transformations as Lie groups

Symmetry in shapes in 3D Euclidean space has been extensively studied since the antiquity. Most relevant to computer graphics are the notion of isometries, scaling, and reflections, as what we call *shape* is always defined up to an arbitrary translation, rotation, scale or even flip. Consequently, the transformations from the Euclidean space to itself that do not affect the shape of an object are indispensable tools to detect symmetries and orbits. The groups of all 3D rotations (SO(3)), rigid body motions (rotations and translations, denoted $\text{SE}(3) \equiv \text{SO}(3) \ltimes \mathbb{R}^3$), or even the more general case of similarities (rotation, translation and uniform (non-zero) scaling, denoted $\text{SIM}(3) \equiv \text{SE}(3) \ltimes \mathbb{R}^+$) have thus found countless applications in graphics (see, for instance, [Ale02]). This latter group of similarities will be most relevant to our exposition, as the orientation-reversing case of reflections will be treated algorithmically instead of geometrically (see Sec. 5.1). We note that similarity transformations in SIM(3) form not only a group (elements can be composed and always have an associated inverse), but a Lie group [MLS94], i.e., it has a differential structure.

Lie algebra. The tangent space at the identity of a Lie group canonically defines a Lie algebra, which we will denote as $\mathfrak{sim}(3)$. Intuitively, the Lie algebra of the group of similarities represents infinitesimal similarity transformations, and thus offers a *linear space* where one can easily sum or subtract infinitesimal elements. A Lie group and its Lie algebra are strongly related, and one can perform calculations in one by first mapping to the other: the exponential (exp) and logarithm (log) maps can convert back and forth between Lie group and Lie algebra. Note that some restrictions apply in general; however, our work will only consider rotations limited to 2π, removing a number of ambiguities present in the general case.

Representation of elements. A transformation $T \in \mathrm{SIM}(3)$ is typically represented as a 4×4 matrix of the form:

$$T = \begin{pmatrix} R & \mathbf{t} \\ 0 & w^{-1} \end{pmatrix} \qquad (1)$$

where $R \in \mathrm{SO}(3)$ is a 3×3 rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ a translation, and $w \in \mathbb{R}$ is a scale. Applying this transformation to a geometric object $\mathcal{S}$ simply rotates $\mathcal{S}$ by R, translates it by $\mathbf{t}$, then scales it by w. The matrix logarithm of this group element T maps it to an element of the Lie algebra which is also typically written out as a 4×4 matrix, this time expressed as [MLS94, Ead13]

$$\log(T) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 & u_1 \\ \omega_3 & 0 & -\omega_1 & u_2 \\ -\omega_2 & \omega_1 & 0 & u_3 \\ 0 & 0 & 0 & -s. \end{pmatrix} \qquad (2)$$

The vector $\omega = (\omega_1, \omega_2, \omega_3)$ can be understood as an infinitesimal rotation (where the direction of ω encodes the axis of rotation, while its norm is an angular speed between 0 and π), while $\mathbf{u} = (u_1, u_2, u_3)$ is akin to an infinitesimal translation and s to an infinitesimal change of scale factor. These 7 independent values in $\log(T)$ can be grouped into a single vector $\widehat{\log}(T) \equiv (\omega^t, \mathbf{u}^t, s)^t$. It should be noticed that in general, $\mathbf{u} \neq \mathbf{t}$ since rotations and translations are intertwined in the calculation of the log matrix. The mapping $T \to \widehat{\log}(T)$ is often referred to as the *logarithmic map*. Note also that this logarithm of a transformation T has a clear geometric meaning, particularly when $w = 1$: in that case, $s = 0$, so ω and $\mathbf{u}$ represent the constant angular and linear velocity of a screw motion [vKC99] that brings the identity element of $\mathrm{SIM}(3)$ to T. With the scale added, one can think of the logarithm of a transformation T as a dilating screw motion, since the scale is also changing (i.e., growing or shrinking) along the screw motion.

Adjoint action. While a Lie algebra is the tangent space of its Lie group at the identity element, it is often important to work in the tangent space of an arbitrary transformation. The *adjoint* serves exactly this purpose, since it linearly transforms tangent vectors from one tangent space of the Lie group to another [MLS94]. This adjoint map is easy to define from the notion of conjugation by an element $F \in \mathrm{SE}(3)$: a similarity T is conjugated by F through the composition FTF^{-1}. The adjoint map is the differential of conjugation at identity; that is, for an arbitrary element F of $\mathrm{SE}(3)$, the adjoint map Ad_F with respect to F is the linear operator acting on Lie algebra elements as:

$$\mathrm{Ad}_F = \begin{pmatrix} R & 0 & 0 \\ \mathbf{t}_\times R & R & -\mathbf{t} \\ 0 & 0 & 1 \end{pmatrix}, \qquad (3)$$

where $\mathbf{t}_\times$ is a skew-symmetric matrix satisfying $\mathbf{t}_\times \mathbf{d} = \mathbf{t} \times \mathbf{d}$ for an arbitrary vector $\mathbf{d} \in \mathbb{R}^3$. Conjugation and adjoint will be especially useful when we talk about change of coordinate frames later. These group actions and geometric entities are thus not purely abstract: they, in fact, offer a coherent framework to manipulate 3D transformations in a principled manner.

2.2 Transformation voting strategy

Now that we have a general notion of shape congruence through the action of similarity transformations, we can define what we

call a *partial* symmetry in a geometric object $\mathcal{S}$: if a local part $\mathcal{P}$ of the object and a transformation $T \in \mathrm{SIM}(3)$ exist such that $T(\mathcal{P}) \cap \mathcal{S}$ is itself a large patch of the object, we infer a visual symmetry for the object corresponding to the transformation T. An exploration of closely-matching pairs of small patches on an input geometry was proposed in [MGP06] to find candidate transformations T_i. Mapping this multitude of transformations into a well-chosen *transformation domain* creates a point cloud, in which high-density regions indicate the presence of a large number of similar transformations—and thus a potential symmetry.

The same strategy can also discover k-orbits, i.e., subgroups of $\mathrm{SIM}(3)$ generated by a set of k generative transformations $\{G_1, \cdots, G_k\}$ which commute with each other [PMW*08]. An element T of a k-orbit is thus of the form

$$T = \prod_i G_i^{r_i}$$

for arbitrary integers or reals r_i (in continuous orbits), and the presence of such orbits signals the existence of repetitive patterns in the shape. Unlike a symmetry which is captured by a single transformation (e.g., a point in the transformation domain), a k-orbit is usually mapped to a k-dimensional linear structure (or lattice if $r_i \in \mathbb{N}$)—and can thus be detected by applying a robust shape detection approach such as RANSAC (repeated stochastic sampling consensus [FB81]) to the point cloud in the transformation domain.

In the remainder of this paper, we show that *adopting a Lie algebra viewpoint* when using the voting framework provides significantly more robust foundations for the discovery of symmetries and orbits.

3 Lie Algebra Domain

While previous works formulated various mappings from elements of $\mathrm{SIM}(3)$ to the transformation domain, we propose the use of a logarithmic map instead. This simple, easily computable map to the Lie algebra of similarity transformations satisfies key properties that will ease the process of symmetry and orbit detection through voting. It is also generic, as it can be used for most previously-studied symmetries and orbits.

3.1 Logarithmic mapping

We introduced the logarithm map $\widehat{\log}$ in Sec. 2.1 which converts any element T of $\mathrm{SIM}(3)$ to a vector $\widehat{\log} T \in \mathbb{R}^7$ encoding the matrix logarithm of T. This operation offers a very convenient map to a general transformation domain. Just like the initial approach of Mitra et al. [MGP06], it is seven dimensional; it is, however, a local diffeomorphism as well: transformations that are quite similar will be mapped to nearby points in $\mathbb{R}^7$. Such a basic requirement was not satisfied by the Euler angle based mapping of [MGP06]. Using this log-based map thus leverages the differential structure of Lie groups, since we directly use the Lie algebra representation of a transformation as its coordinates in the transformation domain.

3.2 Orbits as linear structures

Because the Lie algebra $\mathfrak{sim}(3)$ is a linear space, we reap additional benefits as well. If F and G are two transformations of $\mathrm{SIM}(3)$ that *commute*, then the composition of F and G satisfies:

$$FG = \exp(\log(F) + \log(G)).$$

Thus, if a k-orbit T is generated by transformations $\{G_1, ..., G_k\}$ which commute with each other, then its logarithmic mapping to the transformation domain has the following property:

$$\widehat{\log}(T) = \widehat{\log}\left(\prod_i G_i^{r_i}\right) = \sum_i r_i \widehat{\log}(G_i). \tag{4}$$

Consequently, a k-orbit *always* generates a k-dimensional linear subspace (line for $k=1$, plane for $k=2$, etc...) in the mapped domain. Our logarithmic mapping thus *generalizes* the orbit detection proposed in [PMW*08] as we no longer have to construct a particular mapping to detect each type of k-orbit: our log-based transformation reveals the linear structure of all types of orbits in a unified manner. Note finally that our approach, much like previous work, cannot handle the case of non-commutative orbit; fortunately, most relevant orbits are commutative [PMW*08].

3.3 Basic notion of distance

Finally, the $\widehat{\log}$ map has a last natural property that is particularly convenient to extract orbits. For each mapping to a transformation domain, a reliable distance measurement $\text{dist}(F,G)$ between two transformations F and G must be formulated. One could consider the composition of F and G^{-1} as a notion of difference $\delta = FG^{-1}$ between the two, then use a matrix norm of δ to obtain a scalar measurement. However, this leads to inconsistencies since $\|\delta\| \neq \|\delta^{-1}\|$ in general, which means that the distance between F and G is not equal to the distance between G and F.

Instead, the matrix logarithm does satisfy $\log(\delta) = -\log(\delta^{-1})$; we can thus define a consistent distance via the Euclidean norm (or any weighted variant) of the vector $\widehat{\log}(\delta)$, such that:

$$\text{dist}^2(F,G) = \widehat{\log}(\delta)^t \begin{pmatrix} \alpha\,\text{Id} & 0 & 0 \\ 0 & \beta\,\text{Id} & 0 \\ 0 & 0 & \gamma \end{pmatrix} \widehat{\log}(\delta) \equiv \|\widehat{\log}(\delta)\|_E^2.$$

The parameters α, β and γ have to be fixed by the user as they reflect the arbitrariness of the importance ratio between a unit translation vector, a unit rotation and a unit scaling. Note that for $\alpha = \beta = \gamma = 1$, the norm $\|\cdot\|_E$ becomes the usual Euclidean norm of a vector in $\mathbb{R}^7$. The distance between F and G can be measured *directly* in the transformation domain as the (possibly weighted) Euclidean distance between the two points corresponding to F and G:

$$\text{dist}(F,G) = \|\widehat{\log}(F) - \widehat{\log}(G)\|_E.$$

While this simple notion of distance in our transformation space is already substantially more reliable than previously proposed distances, we show next that another metric can be devised to further increase robustness of the symmetry discovery process.

4 Optimal Distances in Lie Algebra

Due to the shape congruence we discussed in Sec. 2.1, what constitutes a symmetry or orbit of an input object should not depend on its location in Euclidean space or on its scale. This invariance, which is key to robustness in shape analysis, has been surprisingly overlooked in previous work. Next, we discuss the specific mathematical requirements it imposes on the metric in our transformation domain, and deduce a practical notion of distance that vastly improves robustness over previous work.

4.1 Adjoint invariance

Consider an arbitrary rigid transformation $F \in \text{SE}(3)$. If one applies this transformation to an input geometry, and feeds this rotated and translated object to a symmetry detection algorithm, one should expect the output to be exactly the same as if F had not been applied: after all, applying the rigid F does not change the equivalence class in which the object is since its shape is entirely preserved. However, the transformations T_i generated by the voting strategy described in Sec. 2.2 will be each transformed into their conjugation by F, i.e., they become of the form FT_iF^{-1}. Alas, these conjugated transformations do not have the same pairwise distances as their non-conjugated counterparts in *any* of the transformation domains proposed in the past—and even the canonical metric $\|\cdot\|_E$ defined in Sec. 3.3 is *only* invariant to rotations of the input. This is not just a mathematical wrinkle: this lack of invariance to similarity is a huge impediment to the discovery of obvious symmetries as demonstrated in Fig. 2, where a simple rotation and translation of a 3D model is enough to disrupt the proper extraction of trivial symmetries. Indeed, following the approach of [MGP06] (Sec. 2) and discarding the scale for simplicity, two sample points p_i and p_j on this model would find a rotation R_{ij} to align their respective local frames and a translation $t_{ij} = p_j - R_{ij}p_i$. If the brick model is now translated globally by a vector T, then the new translation would become $t'_{ij} = t_{ij} + (I - R_{ij})T$, which can be very different from t_{ij} in general (even on flat surfaces if noise is present), preventing the detection of even simple structures.
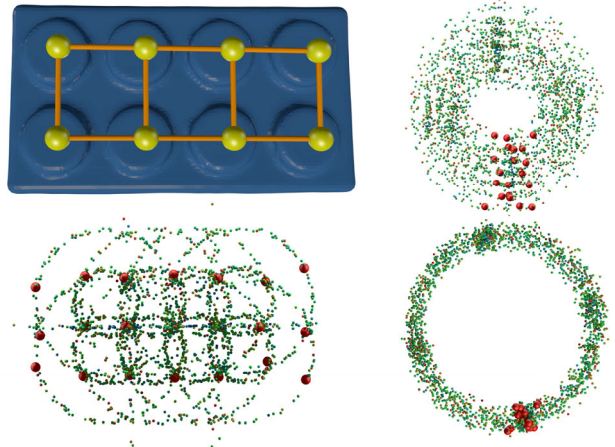

Figure 2: *Lack of adjoint invariance. A lego brick has a 2-generator translational orbit highlighted with the orange grid and yellow spheres (top left). While a "good" placement of this brick in space exhibits clear structures in $\mathbb{R}^7$ (bottom left), simply rotating and translating the brick breaks this structure (right, two different views), rendering automatic orbit detection completely impossible.*

In order to enforce invariance to the placement and scale of the input, we instead need to find a special metric in the transformation domain that is *adjoint invariant*, i.e., that satisfies for any pair of transformations A and B:

$$\|\widehat{\log}(A) - \widehat{\log}(B)\| = \|\text{Ad}_F\,\widehat{\log}(A) - \text{Ad}_F\,\widehat{\log}(B)\|.$$

Intuitively, this is a Lie group extension of the idea of frame invariance for pairwise distances between points; but in our more

complex setup, this is far from an easy requirement to satisfy. Not surprisingly, this type of requirement on the metric has been studied in robotics at the end of the '90s for the case of $SE(3)$, see for instance [MLS94, vKC99]. We can easily extend their findings to $\text{SIM}(3)$: once we write out the requirements that an adjoint-invariant metric must satisfy, we discover that only a 3-parameter family of quadratic forms M of the form:

$$M = \begin{pmatrix} \alpha\,\text{Id} & \frac{1}{2}\beta\,\text{Id} & 0 \\ \frac{1}{2}\beta\,\text{Id} & 0 & 0 \\ 0 & 0 & \gamma \end{pmatrix} \qquad (5)$$

is able to enforce this invariance. For these quadratic forms, one can trivially verify the adjoint-invariance property:

$$\forall \delta \equiv (\omega^t, \mathbf{u}^t, s)^t \in \mathfrak{sim}(3),$$
$$\|\text{Ad}_F\,\delta\|_M = \|\delta\|_M = \alpha\langle\omega,\omega\rangle + \beta\langle\omega,\mathbf{u}\rangle + \gamma s^2. \qquad (6)$$

Unfortunately, the resulting notion of distances (whose geodesics are screw motions [vKC99]) corresponds to only *pseudo-Riemannian* metrics: even after restricting the three parameters to be positive, this distance in the transformation domain can be negative, or even blind to translations when $\omega = 0$. In other words, there is *no* perfect metric for our purpose. Despite this disappointing realization, one can still produce a useful metric that will be adjoint-invariant over most of the transformation domain, which will be enough to render clustering significantly more robust.

4.2 Variational Distance

Although there is no perfect adjoint invariant metric on $\text{SIM}(3)$, we propose a variationally-defined metric which is adjoint-invariant *almost everywhere*. We define distance $\|\cdot\|_A$ based on the smallest norm over all adjoints of rigid body transformations, i.e.,

$$\forall \delta \in \mathfrak{sim}(3), \quad \|\delta\|_A = \min_F \|\text{Ad}_F\,\delta\|_E^2, \quad F \in \text{SE}(3). \qquad (7)$$

That is, this distance should represent the smallest (weighted) Euclidean metric of δ over all possible coordinate frames defined by a coordinate system change F, which corresponds to the "shortest" screw motion from Id to δ. Expanding this definition via Eq. (3) leads to a simpler expression of the form:

$$\|\text{Ad}_F\,\delta\|_E^2 = \alpha\|\omega\|_2^2 + \beta\|\mathbf{t}_\times R\omega + R\mathbf{u} - s\mathbf{t}\|_2^2 + \gamma s^2,$$

The s component in δ here measures the difference of scaling between two sampled transformations, and is usually small. Thus, we approximate the above equation as

$$\|\text{Ad}_F\,\delta\|_E^2 \approx \alpha\|\omega\|_2^2 + \beta\|\mathbf{t}_\times R\omega + R\mathbf{u}\|_2^2 + \gamma s^2$$
$$= \alpha\|\omega\|^2 + \beta\left(\|\mathbf{u}\|^2 + \Psi_{R,\mathbf{t}}(\omega,\mathbf{u})\right) + \gamma s^2,$$

where: $\Psi_{R,\mathbf{t}}(\omega,\mathbf{u}) = \omega^T R^T \mathbf{t}_\times^T \mathbf{t}_\times R\omega + 2\omega^T R^T \mathbf{t}_\times^T R\mathbf{u}$. The fact that $R^T R = \text{Id}$ and $R^T \mathbf{t}_\times R = (R^T \mathbf{t})_\times$ also implies

$$\Psi_{R,\mathbf{t}}(\omega,\mathbf{u}) = \Psi_{\text{Id},R^T\mathbf{t}}(\omega,\mathbf{u}).$$

Thus, we can always assume $R = \text{Id}$ without changing the value of the minimum over all F. Substituting $R = \text{Id}$ into the optimization problem defining the norm, one gets:

$$\|\delta\|_A = \min_{\mathbf{t}} \alpha\|\omega\|^2 + \beta\|\mathbf{t}_\times\omega + \mathbf{u}\|^2 + \gamma s^2. \qquad (8)$$

This quadratic problem has a simple closed form solution. When $\omega \neq 0$, $\mathbf{u}$ can be decomposed in the component $\mathbf{u}_\parallel = \langle\omega/\|\omega\|, \mathbf{u}\rangle\omega$ parallel to ω and the remaining part $\mathbf{u}_\perp = \mathbf{u} - \mathbf{u}_\parallel$ orthogonal to ω, thus picking $\mathbf{t} \times \omega = -\mathbf{u}_\perp$ leads to the solution. Thus, we get:

$$\|\delta\|_A = \begin{cases} \beta\|\mathbf{u}\|^2 + \gamma s^2 & \omega = 0 \\ \alpha\|\omega\|_2^2 + \beta\langle\omega/\|\omega\|, \mathbf{u}\rangle^2 + \gamma s^2 & \text{otherwise} \end{cases}, \qquad (9)$$

Note that this formulation changes the term $\langle\omega,\mathbf{u}\rangle$ in Eq. (6), eliminating the issue of negative distances and returning a reasonable value when $\omega = 0$. However, we traded positive-definiteness for continuity: this distance is now *discontinuous* at $\omega = 0$. We regularize this distance by defining a vector $\bar{\omega}$ increasingly aligned with $\mathbf{u}$ as the norm of ω approaches zero, and equal to the normalized vector $\omega/\|\omega\|$ otherwise, through:

$$\bar{\omega} = \omega/(\|\omega\| + \varepsilon)$$

for ε small (we use $\varepsilon = 10^{-6}$ in all our experiments). We can now define the final expression of this regularized distance as:

$$\boxed{\|\delta\|_A \equiv \alpha\|\omega\|_2^2 + \beta((1 - \|\bar{\omega}\|)\|\mathbf{u}\| + \|\bar{\omega}\|\langle\bar{\omega}, \mathbf{u}\rangle)^2 + \gamma s^2.} \qquad (10)$$

This new distance definition combines the benefits of continuity and adjoint invariance (and consequently, of robustness) away from $\omega = 0$. Equipped with this distance on $\mathfrak{sim}(3)$, we can proceed as advocated in [MGP06] with a mean shift algorithm [Che95] to cluster samples in the transformation domain, where the sample mean $m(\delta)$ of a sample set $\mathcal{S}$ at δ is computed as

$$m(\delta) = \sum_{\delta_s \in S} K(\|\delta - \delta_s\|_A)\,\delta_s \,\Big/\, \sum_{\delta_s \in S} K(\|\delta - \delta_s\|_A), \qquad (11)$$

where K used in our method is the unit Gaussian kernel. The resulting cluster centers will serve as reliable detectors of the significant symmetries present in the input.

4.3 Adjoint-invariant distance to orbit

The distance defined in Eq. (10) provides a measurement between mapped transformations, so that a voting strategy can be used to find the most significant symmetries. In order to detect orbits, however, we also need to provide a distance between a point in the transformation domain and an orbit. As orbits have a linear structure in our construction (see Sec. 3.2), a k-generator orbit can be represented by its k generators as a base matrix G of size $7 \times k$. Any element in the orbit can be written as Gr with $r \in \mathbb{R}^k$; thus an intuitive definition of point-to-orbit distance could be the minimal Euclidean distance from a point p to this linear structure, i.e.,

$$\min_r \|p - Gr\|_E. \qquad (12)$$

The k dimensional vector r can be viewed as the coefficients of the projection point of p in the orbit structure. From this quadratic optimization, we can directly derive an explicit formula for r, and compute the distance. However, this definition suffers from not being frame or scale invariant.

Extending our design of a distance between points, we can also define an adjoint-invariant point-to-orbit distance using:

$$\text{dist}_G(p) = \min_{F,r} \|\text{Ad}_F\,p - \text{Ad}_F\,Gr\|_E. \qquad (13)$$

Following the same reasoning as in Section 4.2, we find that the rotation R of F is also irrelevant to this distance definition: the above optimization problem can be solved by fixing R in F as Id, and solving in alternation for $\mathbf{t}$ and r via simple quadratic optimization. Because the Hessian of $\mathbf{t}$ can be singular, we use Singular Value Decomposition to compute the Hessian's pseudo-inverse. Starting from $r = 0$ as the initial value, the optimization usually ends in less than 10 iterations, where convergence is tested through the gradient norm. After getting $\mathbf{t}$ and r, we use Eq. (10) to compute the distance to avoid the discontinuity issue.

We can now use RANSAC [FB81] to find the k-generator orbits present in the sampled transformations. Grid fitting can also be applied to each orbit as in [PMW*08] if needed.

5 Results

Using the Lie algebra representation of transformations and our new adjoint-invariant distances, we can now reliably employ the voting framework to discover symmetries and orbits. In this section, we discuss how to efficiently implement our approach, and provide various results and comparisons with previous work.

5.1 Implementation Details

Our new transformation mapping and distances can be readily plugged into any existing voting framework. Our implementation mostly follows the pipeline advocated in [MGP06] (see their Fig. 2), which we briefly review to highlight our changes:

1. **Sampling.** A set P of sample points is uniformly selected from the model S; after pruning [MGP06], we get a reduced sample set $\bar{P} = \{p_i\}$.
2. **Pairing.** A local frame is constructed for each point p_i based on estimations of normal and principal curvature directions. 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. Unlike Mitra et al. [MGP06], who reject pairs if their distance in the signature space is too large, we compute the alignment error of neighboring patches surrounding p_i, p_j in terms of the average distance between the closest point pairs, and pairs with large alignment error are discarded. This procedure improves robustness to noise (see Fig. 13).
3. **Clustering.** The remaining set of transformations $\{T_{ij}\}$ is then clustered in logarithmic space using our adjoint-invariant distance metric. For symmetry detection, mean shift [Che95] clustering is performed, while RANSAC [FB81] is used to find orbits.
4. **Extracting.** As in [MGP06], we use each resulting cluster center as a representative transformation, and extract its associated symmetry region using a region growing algorithm.

We also point out a few important details that we used to further increase the accuracy and efficiency of symmetry and orbit detection.

Handling inverses. One simple, yet important issue that we identified is the ambiguity that may rise between a transformation and its inverse. As the pairs of samples that the initial sampling stage of the voting strategy generates are not ordered, one can arbitrarily find either a matching transformation T or its inverse depending on the order in which the samples are compared. While a large enough set of samples will create high densities around both transformations,

this approach weakens the density of each transformation unnecessarily. Using our Lie algebraic representation of transformations, this problem can be addressed easily, since T maps to $\widehat{\log}(T)$ while T^{-1} is mapped to $-\widehat{\log}(T)$. Thus, the distance between two transformation A and B is further altered to read:

$$\min(\|\widehat{\log}(A) - \widehat{\log}(B)\|_A, \|\widehat{\log}(A) + \widehat{\log}(B)\|_A), \quad (14)$$

to account for the inevitable presence of these doublets. This modified distance also avoids the typical issue due to the 2π periodicity of rotations: the distance between two rotations along the same axis by $-\pi + \varepsilon$ and $\pi - \varepsilon$ respectively will be small as expected.

Culling via metric estimates. Since symmetry and orbit detection relies on finding local accumulation of samples, any pair of transformations that are sufficiently dissimilar should be efficiently culled instead of being submitted to an exact distance computation. We obtain a speedup of a factor 3 by first checking the following lower bound on the distance defined in Eq. (10): $\alpha\|\omega\|^2 + \gamma s^2 \leq \|\delta\|_A$. If this lower bound is beyond a given large threshold, we can safely skip further computations for this pair during clustering.

Dealing with reflections. Recall that we did not consider reflections in our method so far, as their orientation-reversing effect creates complications in the mathematical definitions since reflection matrices have no real logarithm [Cul66]. We remedy this issue by applying a reflection to the whole model and adding the result to the input; more specifically:

1. Points are sampled from an input scene S and its YZ-plane reflection $\tilde{S} = T_{YZ}S$. (Note than any planar reflection of the original scene can be used instead: the purpose is simply to add *one* reflection, since all other reflections will be reachable via composition with rigid body transformations—which our method captures precisely thanks to our use of a logarithmic mapping.)
2. In the pairing step, we construct the set of transformations $\{T_{ij}\}$ for which at least one point of $p_i, p_j \in S \cup \tilde{S}$ is in the original scene, i.e., we only deal with pairs satisfying $\{p_i, p_j\} \cap S \neq \emptyset$ to avoid unnecessary computations.
3. In the clustering step, if more than half of the transformations in a cluster involves points coming from $\tilde{S}$, this cluster with transformation T is declared a reflection symmetry $T_{YZ}T$.
4. In the extracting step, for reflection symmetry $T_{YZ}T$, one of the seeds for patch growing p_i, p_j is in $\tilde{S}$, and after mapping it to S, the method proposed in [MGP06] can be directly used to extract the symmetry patches of S.

While this procedure is hidden from the user, this algorithmic trick is enough to deal with plane reflections with minimal changes to the code—see Fig. 3. However, it does double the computational cost of symmetry and orbit discovery. Note that a similar procedure can be applied to find point reflections.

Parameters. We used $\alpha = \beta = 1$ and $\gamma = 10$ in all our experiments. The weight γ was set larger than α and β because we have found that the evaluation of the scale component of a transformation is typically not as reliable as the rotation and translation parts. The main reason is that the estimation of principal curvature uses a fixed neighborhood radius, which is suboptimal for point pairs with scaling symmetry [MGP06, PMW*08]. The regularization parameter $\varepsilon = 1e\text{-}6$ is used in all our results, but any value in the range
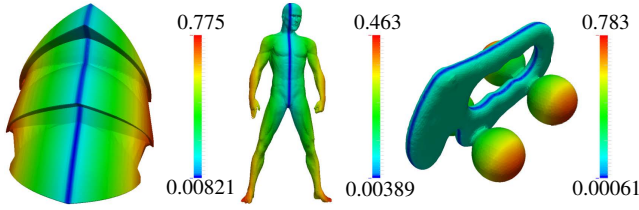
Figure 3: *Reflections. Each point of these models is colored by its distance from the global reflection plane detected by our approach.*

[1e-6, 1e-3] produces visually indistinguishable results. Higher values of ε makes our metric closer to the Euclidean metric, it thus loses some of its valuable properties.

Orbit fitting. As previously identified in [PMW*08], the voting distribution of orbit parameters has a special characteristic: for the translation orbit T^k of the railing in Fig. 1 for instance, a plot of the voting histogram as a function of k (inset) reveals a linear drop (T and T^{-1} are grouped together to increase reliability as we mentioned above). This pattern indicates that the symmetry close to the end of the orbit is harder to detect because of a smaller number of votes. Exploiting this characteristic helps us infer finer symmetries even for low sample counts.
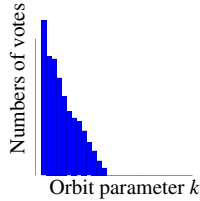
Mean shift clustering. An important parameter rarely discussed in symmetry detection is the kernel size σ used in mean shift clustering. We automatically set it through an initial calibration process: we randomly pick a small number of pairs of transformations and evaluate their variational distance $\|\cdot\|_A$ to estimate the average distance $\bar{h}$ in the transformation set sampled from the object. We experimentally found that choosing a kernel size σ in the interval $[0.01\bar{h}, 0.1\bar{h}]$ consistently leads to robust symmetry detection.

5.2 Comparisons of transformation spaces

In order to further demonstrate the advantages of our logarithmic mapping space over the usual $\mathbb{R}^7$ space [MGP06], we compare the distribution of transformation samples on a series of model to provide a fair, parameter-free estimation of how a clustering algorithm is likely to find symmetries.

For a cluster C_i of transformation samples (with center c_i), we first define its *radial spatial distribution* (inspired by [WS09]) as the function $\tau_i(r) = |f(c_i, r) \cap C_i| / |f(c_i, r) \cup C_i|$, where $f(c_i, r)$ represents the set of samples that are within a sphere centered at c_i with radius r, and $|\cdot|$ represents the volume of a set. This notion of radial spatial distribution reveals the quality of a cluster, as shown in Fig. 4: when a cluster is clearly separated from other samples, its radial distribution will reach a plateau at its maximum value 1, from a minimum radius r_{min} to a maximum radius r_{max}—see Fig. 4(a). Note that the ratio r_{max}/r_{min} thus measures the compactness of the cluster. Instead, if a cluster overlaps other nearby clusters, then the maximum value that τ reaches indicates a separation distance of this cluster from other samples, and can thus be seen as a good measure of compactness—see Fig. 4(b).

In order to compare how the choice of transformation space affects symmetry detection, we need to generate a set of transformation
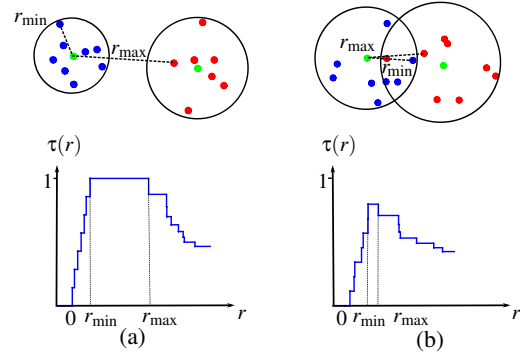

Figure 4: *Radial spatial distributions. When two clusters (blue and red, with their respective centers in green) are well separated, the radial spatial distribution τ of the blue cluster has a clear plateau with $\tau(r) = 1$ (a). When two clusters overlap, the maximum of τ is lower than 1, but represents a good measure of compactness (b).*

samples for which we know what the results are supposed to be. We thus generate input scenes as follows:

1. For a given model M, we randomly generate three transformations T_1, T_2, T_3, and apply them to M to get a new scene $\bar{M}$ composed of $T_1 M$, $T_2 M$ and $T_3 M$.

2. We perform the sampling and pairing step of the traditional voting approach for $\bar{M}$ and get the transformation samples.

3. These transformation samples are then mapped to either our logarithmic, or the original $\mathbb{R}^7$ transformation space.

By construction, the ground truth symmetries in this scene are known to be $T_2 T_1^{-1}$, $T_3 T_1^{-1}$ and $T_3 T_2^{-1}$ (and their inverses, ignored for simplicity). Instead of performing the usual clustering which would be unfairly biased by the choice of ($\mathbb{R}^7$ or variational) distance, we assign for *each* pair of transformations one of the three ground truth symmetries based on the one minimizing the local ICP error [BM92]—or we simply discard this transformation if the error is larger than 1% of the bounding box diagonal size. Note that this assignment is costly to compute, but has the advantage of being purely geometrical, and thus really indicate how intrinsically good a mapping is *independent* of the clustering method used on it.

Finally, we measure the quality of the results using voting based on radial spatial distribution: a cluster (corresponding to one of the three symmetries) casts a vote for our logarithmic space or the $\mathbb{R}^7$

space based on which embedding space leads to a larger max value of its radial spatial distribution—or for the embedding space with the larger ratio of r_{max}/r_{min} if both spaces leads to the same maximum. This voting is done for 30 scenes that were randomly generated from a model. In our tests, the logarithmic space *systematically* outvotes the $\mathbb{R}^7$ space by at least 50%, as demonstrated in the inset. Therefore, even if one ignores the advantages of our adjoint invariant clustering, the use of a logarithmic space is valuable in itself.

Adjoint invariance. Using radial spatial distributions, we can also demonstrate that our adjoint-invariant distance leads to improved
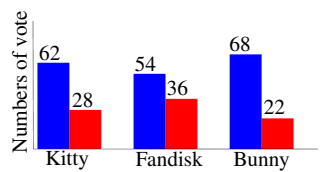
robustness in the choice of kernel size used in mean shift clustering: as Fig. 5 illustrates, distributions change significantly when the scene is rotated and translated for the $\mathbb{R}^7$ space mapping, while they remain unchanged in our method. This invariance to rigid transformations is crucial to robust symmetry and orbit detection.

2D visualization of transformation distributions. The role of adjoint invariance is illustrated in a simple example in Fig. 6, where a *lamp* model contains three identical flower-shape heads. As the visualization of a distribution of transformation samples in dimension 7 is difficult, we used *multidimensional scaling* (MDS) and *multivariate kernel density estimation* (MKDE) [VR02] to compute a 2D projection that best preserves the distances between samples in $\mathbb{R}^7$ (a-d). Such a 2D visualization has the advantage of making the structure of this high-dimensional distribution visually obvious. While the distribution of transformations in our logarithm space is virtually unchanged for two different orientations and positions of the lamp (a-b), the original symmetry detection algorithm generates extremely different transformation distributions in $\mathbb{R}^7$ (c-d). This behavior makes it impossible to pick a threshold for the clustering technique to decide which prominent clusters are potential symmetries. As a consequence, a lack of adjoint invariance leads to the inability to detect symmetries automatically and reliably. We point out that simply "normalizing" the position of the object or the scene by a global transformation does not help either as illustrated in Fig. 11, where multiple symmetries and orbits cannot be recovered with this procedure. In contrast, our metric $\|\cdot\|_A$ performs as expected with no parameter tuning. Moreover, notice that these three symmetries (and their inverses) have cluster sizes that are clearly much larger than any other, so even without knowing the number or type of symmetries present in this example, the sudden change in cluster size (Fig. 6, bottom graph) is a clear indication that only three main symmetries are present. Our approach thus offers a *sight-unseen* detection of symmetries.

5.3   Ground truth comparison for a complex model

The *necklace* in Fig. 7 is composed of 32 components with identical links $\{C_k\}, k = 1, \cdots, 32$. The symmetry transformations between these components being known in advance, we can measure the accuracy of a symmetry detection algorithm by comparing the resulting clusters $\mathcal{S} = \{S_i\}$ to the ground truth transformations $\mathcal{G} = \{G_i\}$, where each transformation G_i maps a link C_{i-} to another link C_{i+} of the necklace. Ideally, the largest clusters in $\mathcal{S}$ should match the ground-truth symmetries $\mathcal{G}$. We test this hypothesis for both our Lie-algebra voting and the $\mathbb{R}^7$ voting of Mitra et al. [MGP06]. We
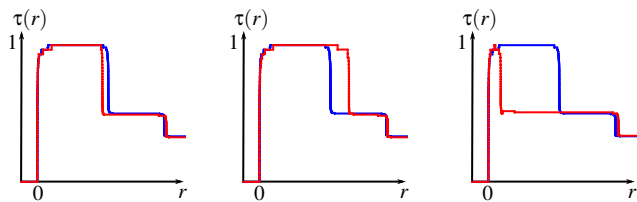

Figure 5: *Invariance of radial spatial distribution. While a typical cluster's radial spatial distribution in our logarithmic space does not change if the scene is rotated and translated in space (blue; 3 different poses), the $\mathbb{R}^7$ mapping space behaves unreliably (red).*
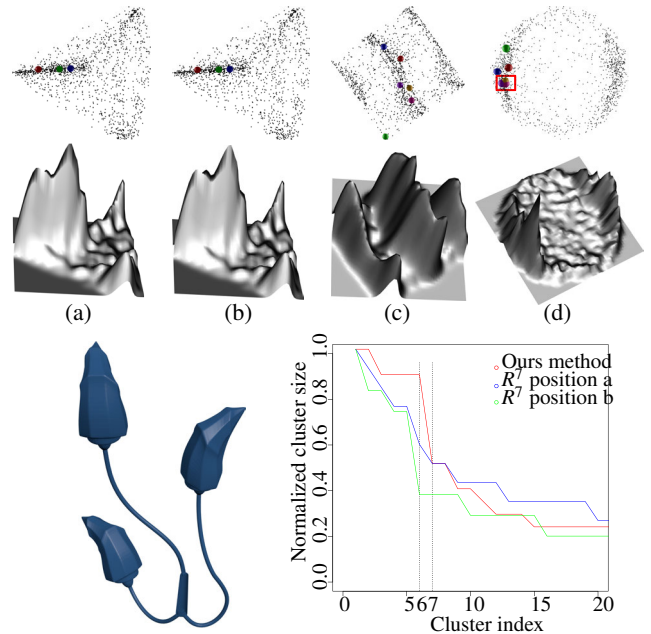

Figure 6: *Adjoint invariance. The* lamp *model (bottom left) has 3 significant symmetries due to identical heads. For two different positions of the lamp in space, we plot both multidimensional scaling (MDS) and multivariate kernel density estimation (MKDE) [VR02] 2D visualizations of the resulting transformation distributions for our method ((a) and (b)) vs. the original $\mathbb{R}^7$ method ((c) and (d)). Cluster centers corresponding to the correct transformations between the three heads are colored. While our Lie algebra approach is unchanged for different positions of the lamp, the original method differs widely based on position and orientation, with symmetries being sometimes co-located (see rectangle in (d)). We also show the distribution of cluster sizes (number of the transformations in each cluster, normalized by the largest one; bottom right), when a sudden change is visible with our Lie-algebra voting after the first six clusters—thus clearly identifying the three symmetries (and their inverses) as the only relevant ones.*

first find the "closest" transformation S_j for each transformation $G_i \in \mathcal{G}$ via $\min_j D(FG_iF^{-1}, S_j)$, where $F \in \mathrm{SE}(3)$ is a random rigid transformation applied to the whole necklace to place the necklace into different positions, and D stands for the distance measurement (Euclidean distance for $\mathbb{R}^7$ parameterization, adjoint invariant distance for our logarithm parameterization). We then define an error $E(G_i)$ for each G_i as the average pointwise matching error from C_{i-} to C_{i+} using S_j (normalized by the bounding box diagonal of the model), since it measures how well we match each known symmetry. Finally, the average of all $E(G_i)$ indicates how accurate the set of discovered symmetries $\mathcal{S}$ is. Fig. 7 shows the results (in log scale) of this test with mean shift clustering for 3 different positions (translation and rotation) of the necklace model for a large number of kernel sizes—a parameter that clustering heavily depends on. Our method produces identical results for all positions, while the $\mathbb{R}^7$ approach exhibits very different behaviors. Moreover, our approach has better accuracy for most kernel sizes, or near equivalent accuracy for the best of the three positions. Finally, the $\mathbb{R}^7$

approach fails to detect some of the symmetries G_i in two out of the three positions. Note that the unit of kernel size on the graph in Fig. 7 is set to the minimal distance among the transformations in $\mathcal{G}$, i.e. $\min_{G_i,G_j \in \mathcal{G}} D(G_i, G_j)$.
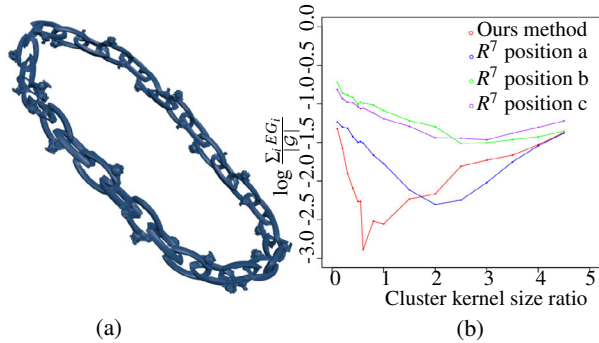

Figure 7: *Ground truth comparison. (a) The necklace model. (b) The mean shift based symmetry detection error $\sum_i E(G_i)$ as a function of cluster kernel size ratio is generally much smaller for our Lie-algebra voting approach compared to the $\mathbb{R}^7$ method, and does not depend on the position of the model. Notice that for positions b and c, the $\mathbb{R}^7$ method fails to detect some of ground truth symmetries no matter how the cluster kernel size is tuned.*

5.4 Orbit detection

Linear structures. Fig. 8 demonstrates that our transformation domain exhibits clear linear structures when orbits are present, while previous $\mathbb{R}^7$ representations do not share this property. The *bumpy torus* has a 6-fold rotational symmetry and many partial symmetries in the form of discrete rotations, while the *Thai status* model has a global 3-fold rotational orbit and the *workpiece* model has a clear translation orbit. The $\mathbb{R}^7$ representation have no clear structures for these models, but our approach exhibits obvious (and relevant) structures.

Orbit detection. Our approach is able to discover both continuous and discrete orbits [PMW*08, BWKS11]. Fig. 9 depicts the *fire hydrant* model containing a continuous rotational symmetry (highlighted by the dominant red line), the *octopus* model containing a 8-fold rotational orbit (clearly visible as 8 separate dots in the transformation domain), and the *carter* model with a 6-fold rotational orbit and a partial continuous rotational orbit (exhibiting a continuous line in the ω subspace, with 6 clear dots). Such clear features are thus easily detected and characterized by common clustering and linear subspace detection algorithms, without having to experimentally tune a variety of parameters. Robustness of the symmetry and orbit discovery process is thus much improved. The example in Fig. 1 contains a rotational orbit (represented by the blue dots) along the z axis and a translation orbit (represented by the red dots) along the x axis. When the scene is centered, orbits are easy to detect, even with a Euclidean distance, as the translation component of the rotational symmetry appears at the origin of the transformation domain. If the scene is translated along the y axis, however, the Euclidean distance can no longer distinguish the two orbits correctly. Our adjoint invariant orbit distance has no such issue, and all of our results were robustly handled without having to worry
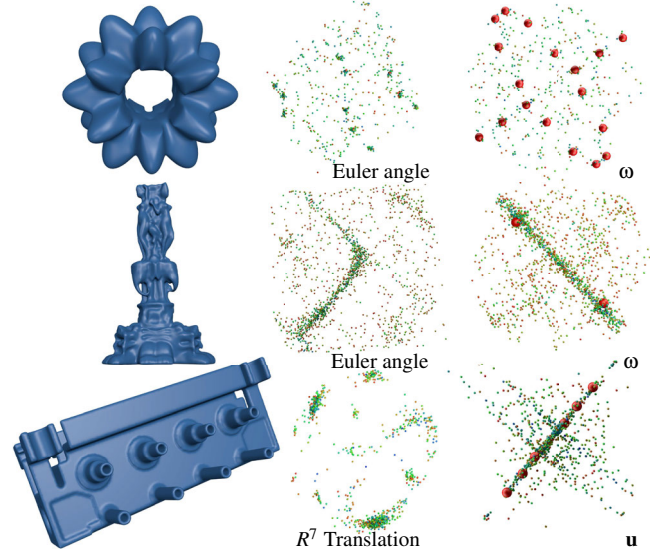

Figure 8: *Linear structure of orbits. Our transformation domain (right, visualized through its ω or **u** coordinates for simplicity) always exhibits clear linear structures when orbits are present, while an Euler-angle based domain is unreliable. Transformation samples are colored by their alignment error (blue indicating small values). Cluster centers are shown as red dots.*

about whether the scene has to be relocated to detect symmetries and orbits.

5.5 Unified symmetry and orbit discovery

Detecting symmetries and orbits in a complex scene is a much harder task than detecting symmetries and orbits in a single model. Not only do the computational costs associated to both patch sampling and clustering increase, but the resulting complexity of the density field in the transformation domain can be quite high. However, because our approach satisfies the linear orbit property as well as an invariance to similarities, visualizing the transformation domain once basic clustering and RANSAC has been performed to hunt for symmetries and/or orbits is significantly easier. The scene in Fig. 11 includes various symmetries and orbits; without having to resort to many different mappings, all the SIM(3) symmetry and orbits are detected in a unified way. More importantly, if one uses a coordinate system dependent method instead, significant structures in the scene are missed because of their arbitrary relative locations.

6 Conclusions

We presented in this paper a new Lie algebra variant of the voting framework to discover, automatically and reliably, symmetries and orbits in geometric scenes. By leveraging the Lie group nature of symmetries and constructing a consistent metric that is fundamentally invariant to rigid body and scale change of the input models, we significantly improved the robustness and accuracy of the results. In particular, we introduced a logarithm mapping of transformations that unifies most previous symmetries and orbits, allowing for a significantly easier analysis of complex scenes since no manual tuning is required to discover symmetry and orbit.
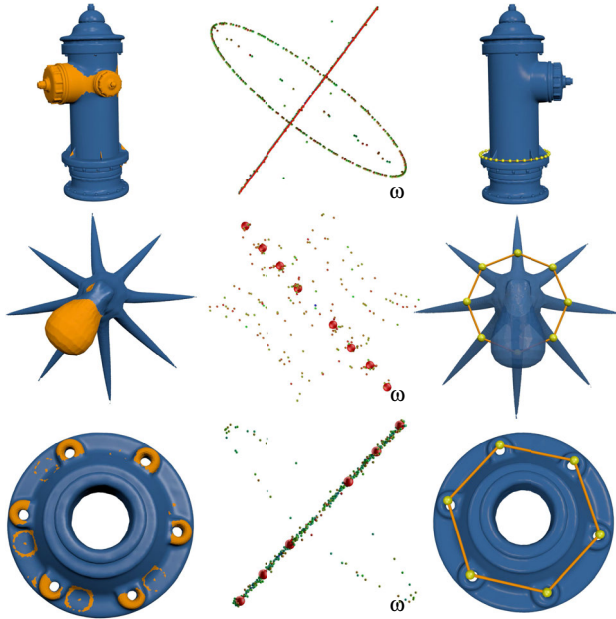
Figure 9: *Rotational orbits. Our choice of Lie algebra coordinates renders rotational orbits particularly obvious in the transformation domain (middle), be they continuous (fire hydrant), discrete (octopus), or both (carter). We can then color the regions of the object exhibiting orbits (left) and/or visualize particular orbits (right).*
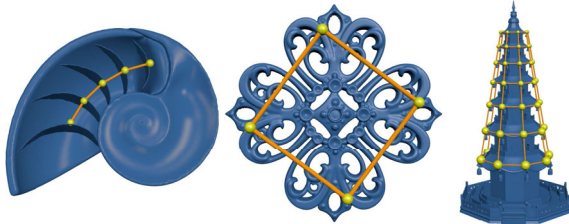

Figure 10: *Gallery. Our symmetry and orbit detection approach can find all the symmetries in these classical models with a single (logarithmic) mapping, and without the need to carefully move the model to a specific location.*

This unified Lie algebraic framework may also benefit common geometry processing tasks such as editing or reconstruction as it offers a principled way to identify, even without user interaction, partial symmetry in geometric data. In particular, we plan to investigate how symmetry and orbit detection can be used as a reliable prior in the context of reconstruction and abstraction of manmade geometry (where symmetries and orbits abound). Additionally, while we focused on finding simple geometric patterns (linear subspaces) in the voting space, detecting complex patterns is an interesting future work as it may allow detection of hierarchical and/or semantic structures in scenes.

Limitations. While we improved the reliability and robustness of the clustering step in the symmetry and orbit detection pipeline through a Lie-algebra voting and an adjoint-invariant distance, a few limitations of the original voting framework remain. First, our voting strategy is based on uniform sampling [MGP06], generating fewer points for symmetries involving small regions. Consequently, small symmetries have a smaller probability to be detected. For ex-
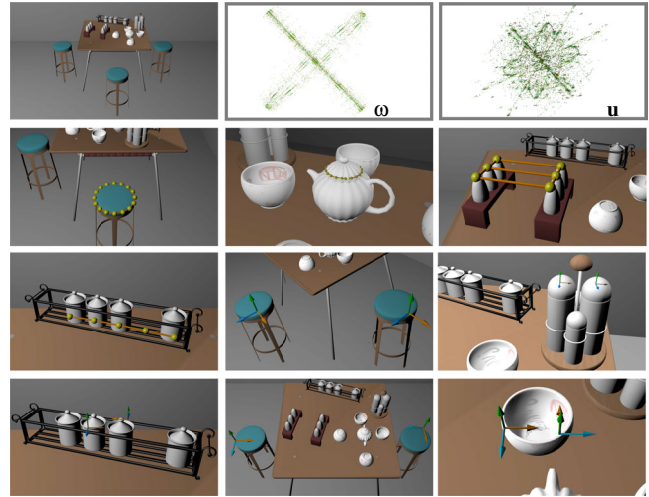

Figure 11: *Indoor scene. The dining room scene contains many symmetries and orbits; linear structures of orbits are very clear as illustrated by our visualization of ω and u components, despite the complexity of the scene. Bottom rows show the most significant orbits and some representative symmetries found in this scene using coordinate frames and lattices to indicate symmetries and orbits.*

ample, the $2\pi/18$ symmetry of the corrugated teapot in Fig. 12 may be missed if too few transformation samples are used.

We showed in Fig. 6 that our method can automatically determine the number of significant symmetries without manually tuning the parameters by just tracking sudden changes in (normalized) cluster sizes. However, such clear, sudden changes may not be present in complex examples. For instance, our approach identifies correctly all the 992 significant symmetries of the necklace model from 45845 sampled transformations, but there is no clear demarcation that indicates that other clusters are simply small artifacts as the graph in the inset indicates. However, we significantly outperform the $\mathbb{R}^7$ approach since it cannot even find all the symmetries for certain positions of the model.
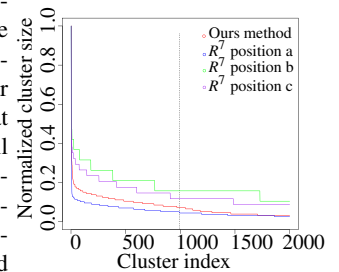
Additionally, as our method depends on the accuracy of transformation estimates from sample pairs, very noisy data (Fig. 13) or symmetries with significantly different scales may incur estimation errors which will affect clustering quality. Similarly, while our experimental kernel size estimation for mean shift clustering (described in Section 5.1) works well in most of the cases, an adaptive strategy may be required to detect complex symmetries and orbits with many different scales.
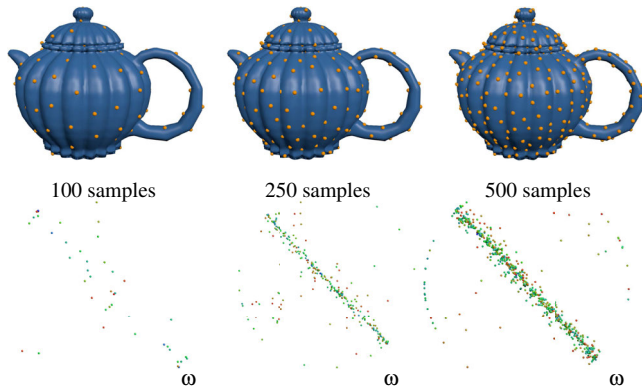
Figure 12: *Sampling rates and orbits. From left to right, we show increasing sample density of a corrugated teapot model and the corresponding logarithm space (visualizing the ω component). The linear structure of the rotational orbit of the model only becomes clear when the number of samples is high enough, at the cost of an increased computation time as well.*
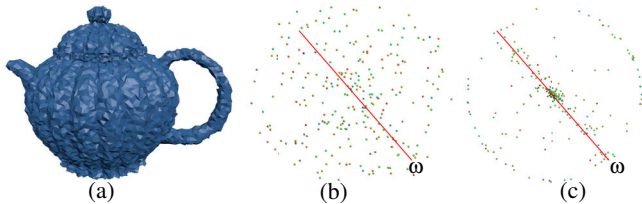

Figure 13: *Effect of noise. We add random Gaussian noise to the corrugated teapot model and use 500 samples for symmetry detection (a). Since transformations computed from local frame alignment become unreliable, the underlying linear structure of the rotational orbit is less crisp (b), but increasing neighboring patch size for the sampling points reduces this issue (c).*

References

[Ale02] ALEXA M.: Linear combination of transformations. *ACM Trans. Graph. 21*, 3 (2002), 380–387. 2

[BAK10] BERMANIS A., AVERBUCH A., KELLER Y.: 3D symmetry detection and analysis using the pseudo-polar Fourier transform. *Int. J. Comput. Vision 90*, 2 (Nov. 2010), 166–182. 1

[BBW*08] BERNER A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: A graph-based approach to symmetry detection. In *Symp. Point-Based Graph.* (2008), pp. 1–8. 1

[BBW*09] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Comp. Graph. Forum 28*, 2 (2009), 697–706. 1

[BM92] BESL P., MCKAY N. D.: A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell., 14*, 2 (Feb 1992), 239–256. 7

[BWKS11] BOKELOH M., WAND M., KOLTUN V., SEIDEL H.-P.: Pattern-aware shape deformation using sliding dockers. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 123:1–123:10. 2, 9

[Che95] CHENG Y.: Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 8 (Aug. 1995), 790–799. 5, 6

[Cul66] CULVER W. J.: On the existence and uniqueness of the real logarithm of a matrix. *Proc. Amer. Math. Soc 17* (1966), 1146–1151. 6

[Ead13] EADE E.: Lie groups for 2D and 3D transformations. http://www.ethaneade.org/lie.pdf, 2013. 3

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395. 3, 6

[JXCZ13] JIANG W., XU K., CHENG Z.-Q., ZHANG H.: Skeleton-based intrinsic symmetry detection on point clouds. *Graph. Models 75*, 4 (2013), 177–188. 1

[KBWS] KERBER J., BOKELOH M., WAND M., SEIDEL H.-P.: Scalable symmetry detection for urban scenes. *Comp. Graph. Forum 32*, 1. 2

[KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Symmetry descriptors and 3D shape matching. In *Symp. Geom. Process.* (2004), pp. 115–123. 1

[LCDF10] LIPMAN Y., CHEN X., DAUBECHIES I., FUNKHOUSER T.: Symmetry factored embedding and distance. *ACM Trans. Graph. 29*, 4 (2010), 103:1–103:12. 1

[LTSW09] LASOWSKI R., TEVS A., SEIDEL H.-P., WAND M.: A probabilistic framework for partial intrinsic symmetries in geometric data. In *Int. Conf. on Comput. Vision* (Sept 2009), pp. 963–970. 1

[MGP06] MITRA N. J., GUIBAS L., PAULY M.: Partial and approximate symmetry detection for 3D geometry. *ACM Trans. Graph. 25*, 3 (2006), 560–568. 2, 3, 4, 5, 6, 7, 8, 10

[MLS94] MURRAY R. M., LI Z., SASTRY S. S.: *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994. 2, 3, 5

[MPWC13] MITRA N. J., PAULY M., WAND M., CEYLAN D.: Symmetry in 3D geometry: Extraction and applications. *Comp. Graph. Forum 32*, 6 (2013), 1–23. 1

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *ACM Trans. Graph. 27*, 3 (2008), 43:1–11. 2, 3, 4, 6, 7, 9

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Trans. Graph. 25*, 3 (2006), 549–559. 1

[THW*14] TEVS A., HUANG Q., WAND M., SEIDEL H.-P., GUIBAS L.: Relating shapes via geometric symmetries and regularities. *ACM Trans. Graph. 33*, 4 (2014), 119:1–119:12. 1

[vKC99] ŽEFRAN M., KUMAR V., CROKE C.: Metrics and connections for rigid-body kinematics. *International Journal of Robotics Research 18*, 2 (1999), 242:1–16. 3, 5

[VR02] VENABLES W. N., RIPLEY B. D.: *Modern Applied Statistics with S*, Fourth ed. Springer, New York, 2002. 8

[WS09] WEINBERGER K. Q., SAUL L. K.: Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research 10* (2009), 207–244. 7

[XZT*09] XU K., ZHANG H., TAGLIASACCHI A., LIU L., LI G., MENG M., XIONG Y.: Partial intrinsic reflectional symmetry of 3d shapes. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 138:1–138:10. 1

[ZXJ*13] ZHANG H., XU K., JIANG W., LIN J., COHEN-OR D., CHEN B.: Layered analysis of irregular facades via symmetry maximization. *ACM Trans. Graph. 32*, 4 (2013), 104:1–104:10. 2