

# Symmetry in 3D Geometry: Extraction and Applications

Niloy J. Mitra

UCL

Mark Pauly

EPFL

Michael Wand

MPII

Duygu Ceylan

EPFL

---

## Abstract

*The concept of symmetry has received significant attention in computer graphics and computer vision research in recent years. Numerous methods have been proposed to find, extract, encode, and exploit geometric symmetries and high-level structural information for a wide variety of geometry processing tasks. This report surveys and classifies recent developments in symmetry detection. We focus on elucidating the key similarities and differences between existing methods to gain a better understanding of a fundamental problem in digital geometry processing and shape understanding in general. We discuss a variety of applications in computer graphics and geometry processing that benefit from symmetry information for more effective processing. An analysis of the strengths and limitations of existing algorithms highlights the plenitude of opportunities for future research both in terms of theory and applications.*

**Keywords:** symmetry detection, shape analysis, transformation groups

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, systems

---

## 1. Introduction

*The mathematical sciences particularly exhibit order, symmetry, and limitation; and these are the greatest forms of the beautiful.* - Aristotle

Symmetry is a universal concept in nature, science, and art (see Figure 1). In the physical world, geometric symmetries and structural regularity occur at all scales, from crystal lattices and carbon nano-structures to the human body, architectural artifacts, and the formation of galaxies. Many biochemical processes are governed by symmetry and as a result we experience a wealth of biological structures that exhibit strong regularity patterns. This abundance of symmetry in the natural world has inspired mankind from its origins to incorporate symmetry in the design of tools, buildings, artwork, or even music. Besides aesthetic considerations, physical optimality principles and manufacturing efficiency often lead to symmetric designs in engineering and architecture.

Geometric data, acquired via scanning or modeled from scratch, is traditionally represented as a collection of low-level primitives, e.g., point clouds, polygon meshes, NURBS patches, etc., without explicit encoding of any underlying high-level structure. Finding symmetries in such geometric data is thus an important problem in geometry process-

ing that has received significant attention in recent years. Numerous applications immediately benefit from extracted symmetry information, e.g., shape matching, segmentation, retrieval, geometry completion, beautification, meshing, or procedural modeling.

This survey reviews the state-of-the-art in symmetry detection methods for geometric data sets. We introduce basic mathematical terms and present various high-level criteria to organize existing work into a set of categories, emphasizing their similarities and differences (see Table 1). We hope that this comparative survey will help readers navigate through the constantly expanding literature on symmetry detection and inspire researchers to contribute to this emerging field in the future.

The rest of the survey is structured as follows: In Section 2 we start with a discussion of the classical mathematical theory of symmetry groups that characterizes the structure of globally and exactly symmetric objects. In Section 3 we look at more general cases including partial and approximate symmetry, which are particularly relevant in practical applications in computer graphics and vision. The main part of this paper surveys different algorithms for symmetry detection (Sections 4–7) with a special focus on partial symmetry

detection methods (Section 6). We then examine what types of geometrical structures these different algorithms discover and how symmetry information is encoded (Section 8). Finally, we discuss various applications of symmetry detection (Section 9) and conclude with thoughts on future challenges in the field (Sections 10 and 11).

## 2. Classical Theory

Symmetry is a general concept in mathematics [Wey52]; broadly speaking, a symmetry or invariance preserves a certain property (e.g., geometric similarity) of an object under some operation applied to the object. This notion of invariance is formalized in an elegant branch of mathematics called *group theory* [Rot94]. In the context of geometry, we will consider geometric transformations as the symmetry operations, such as reflections, translations, rotations, or combinations thereof.

We say that a geometric object  $M$  is *symmetric* with respect to a transformation  $T$ , if  $M = T(M)$ , i.e.,  $M$  is invariant under the action of transformation  $T$ . The set  $\mathcal{S}$  of all symmetry transformations of a shape has a very specific structure, namely that of a group. A *symmetry group* is a set of transformations that satisfies the following group axioms with composition as the group operation:

- **Closure:** If  $M$  is symmetric with respect to two transformations  $T_1$  and  $T_2$ , then it will also be symmetric with

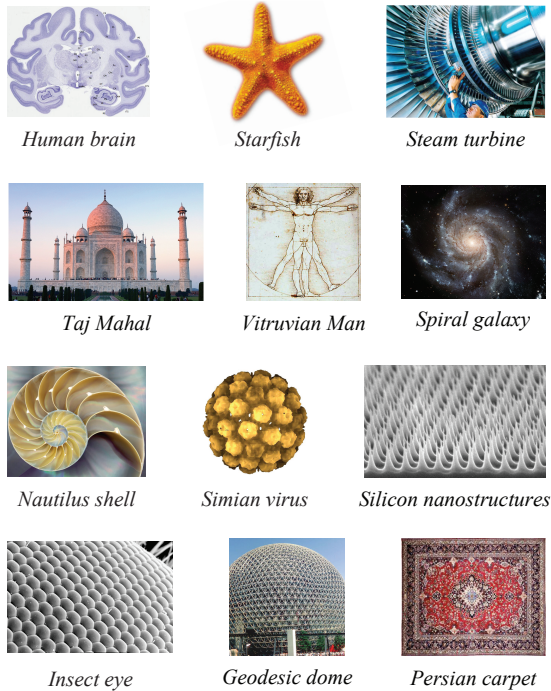
respect to the composition  $T_1 T_2$ . Thus if  $T_1, T_2 \in \mathcal{S}$ , it follows that  $(T_1 T_2) \in \mathcal{S}$  also.

- **Identity element:** The identity transform  $I \in \mathcal{S}$  is always a symmetry transformation, since it trivially leaves any object unchanged, i.e.,  $I(M) = M$ .
- **Inverse element:** For each symmetry transformation  $T \in \mathcal{S}$  there exists an inverse element  $T^{-1} \in \mathcal{S}$ , such that  $T^{-1}T = TT^{-1} = I$ .
- **Associativity:** The compositions of multiple transformations is independent of the priority of composition, i.e.,  $(T_1 T_2)T_3 = T_1(T_2 T_3) \forall T_1, T_2, T_3 \in \mathcal{S}$ .

Note that while the priority of composition is irrelevant, the order of transformations can be important. For example, composing two rotations in 3D about different axes in general leads to a different transformation when switching the order of application. Groups for which the relation  $T_1 T_2 = T_2 T_1$  holds  $\forall T_1, T_2 \in \mathcal{S}$  are called *commutative* or *Abelian* groups, e.g., any two rotations about the same (rotation) axis.

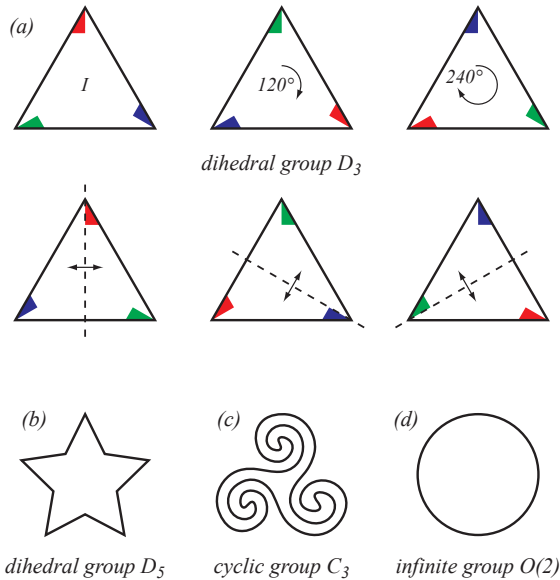
The notion of symmetry as invariance under transformations is a powerful concept that has been long advocated prominently by Felix Klein in his Erlanger Programm [Kle93]. Klein proposed to characterize different classes of geometry, such as projective geometry or Euclidean geometry, based on the underlying symmetry groups. For example, distances and angles are invariants in Euclidean geometry. These properties are preserved under transformations of the Euclidean group, the group of all isometries with respect to the Euclidean metric. This notion of classifying geometries based on symmetry groups can be transferred to geometric objects as well.

Let us consider the example of a 2D equilateral triangle shown in Figure 2(a). We observe that a rotation of  $120^\circ$  around the triangle center maps the triangle onto itself. It follows that all integer multiples of this rotation are also symmetries of the triangle. However, only three of these rotations are unique, since a rotation by  $360^\circ$  is equal to the identity transformation. We also see that the triangle has three reflectional symmetries across the lines from each vertex through the triangle center. Together, these transformations form the *dihedral group*  $D_3$  consisting of six elements, three rotations (of which one is the identity transformation) and three reflections. In general, the dihedral group  $D_n$  represents the symmetries of a regular  $n$ -gon. These symmetries can be represented as finite combinations of two *generating transformations*. For example, repeated application of a  $72^\circ$  rotation and a reflection can generate all elements of a dihedral group  $D_5$  (see Figure 2(b)). Shapes that have rotational symmetries but no reflectional symmetries, such as the triskelion shown in Figure 2(c) can be characterized by a *cyclic group*  $C_n$  that is generated by a rotation of  $360^\circ/n$ . The cyclic and dihedral groups are *finite* point groups. In two dimensions, they are subgroups of the orthogonal group  $O(2)$ , the group of all Euclidean isometries that leave the origin fixed. This *infinite* group is the symmetry group of the circle, which is sym-



**Figure 1:** Examples of symmetry in nature, engineering, architecture, and art.





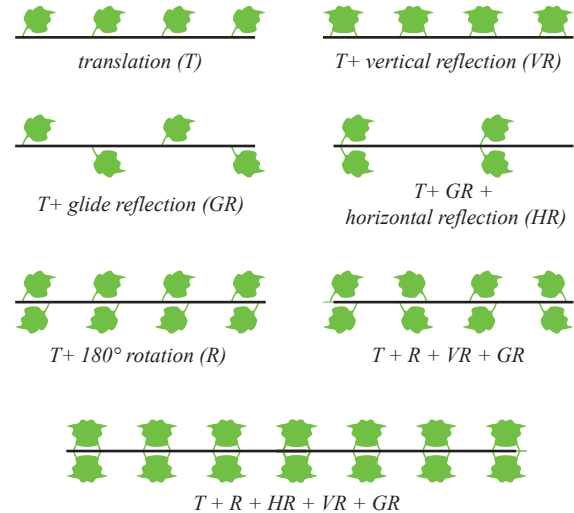
**Figure 2:** The dihedral group  $D_3$  represents the symmetries of the equilateral triangle (the colored flags are added to illustrate the transformation), while  $D_5$  is the symmetry group of the five-sided star. The triskelion has three rotational symmetry but no reflectional symmetries and is represented by the cyclic group  $C_3$ . All of these finite point groups are subsets of the isometry group  $O(2)$  that represents the symmetries of the circle.

metric with respect to rotations of arbitrary angle around its center and reflections across arbitrary lines through the center (Figure 2(d)).

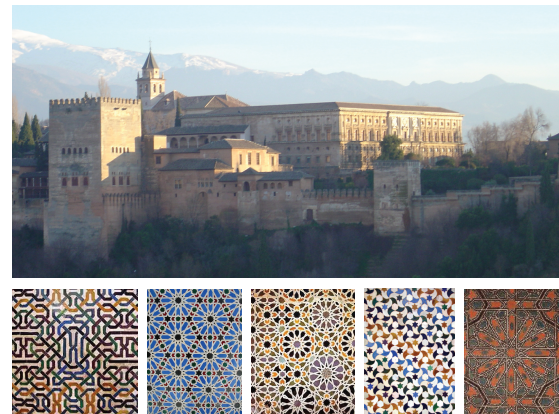
Symmetry groups have been used extensively in the study of decorative art and structural ornaments. The symmetries of a two-dimensional surface that is repetitive in one direction and extends to infinity along that direction can be classified by one of exactly seven *Frieze groups* (see Figure 3). If repetition occurs in two different directions, seventeen distinct groups are possible, denoted as *wallpaper groups*. These groups combine reflections, rotations, and translations so that all these transformations and all combinations of them leave the entire grid unchanged. This leads to a wealth of repetitive patterns that can, for example, be observed in the beautiful tiling patterns of the Alhambra in Granada, Spain (see Figure 4).

In summary, the classical theory of symmetry groups describes the structure of transformations that map objects to themselves *exactly*. Such exact, global symmetry leads to a group structure because after applying a transformation, we end up with the same situation as before, creating a closed algebraic structure.

In computer graphics applications, however, we often



**Figure 3:** Frieze groups are composed of translation, rotation by 180 degrees, glide reflection, reflection about a horizontal line, or reflection about a vertical line.



**Figure 4:** The symmetries of the tiling patterns of the Alhambra can be described by wallpaper groups of which 17 distinct types exist.

face a more general problem, where symmetry is approximate and partial. For example, for a simple building facade with windows related by translational symmetry, the closure property is violated since the facades have finite extent. Further, we have to handle different classes of transformations. Finally, we need efficient algorithms to compute such symmetries. In the following subsection, we discuss these issues.

### 3. Classification

This survey investigates algorithmic paradigms for extracting symmetries and higher-order relations in 3D geometric

Reference	Method	Input				Output						
		mesh	points	volume	image	global	partial	disc.	cont.	extrin.	intrin.	Type of Features
Alt et al. 1988 [AMWW88]	object-space graph isomorph.	✓				✓		✓		✓		pairwise rigid
Atallah et al. 1985 [Ata85]	string pattern matching	✓				✓		✓		✓		pairwise reflections
Bermanis et al. 2010 [BAK10]	spectral analysis			✓		✓		✓		✓		pairwise rotations, reflections angular difference functions
Berner et al. 2008 [BBW*08]	feature-graph matching	✓	✓	✓		✓	✓			✓		segmentation rigid slippage features
Berner et al. 2009 [BBW*09a]	feature-graph matching	✓	✓			✓	✓			✓	✓	segmentation (relaxed) isometries Gaussian curvature, curvature lines
Bokeloh et al. 2009 [BBW*09b]	feature graph matching	✓	✓			✓	✓			✓		segmentation rigid lines
Ben-Chen et al. 2010 [BCBSG10]	flow of killing vector fields	✓				✓		✓		✓		continuous continuous isometries Killing vector fields
Bokeloh et al. 2011 [BWKS11]	RANSAC-based transformation verification	✓				✓	✓	✓	✓	✓		symmetry groups, continuous translations sharp creases
Berner et al. 2011 [BWM*11]	feature-graph matching	✓	✓			✓	✓			✓		segmentation subspace symmetries lines of large principle curvature
Chertok et al. 2010 [CK10]	spectral analysis		✓	✓		✓	✓			✓		pairwise rotations, reflections local image features
Gal et al. 2006 [GCO06]	geometric hashing	✓				✓	✓			✓		segmentation similarity transform. curvature based salient features
Gelfand et al. 2004 [GG04]	slippage analysis	✓	✓			✓		✓		✓		continuous rigid local slippage signatures
Kazhdan et al. 2003 [KCD*03]	descriptor computation with Fourier methods			✓		✓			✓	✓		pairwise reflections
Kazhdan et al. 2004 [KFR04]	descriptor computation with Fourier methods			✓		✓			✓	✓		pairwise rotations, reflections
Kim et al. 2010 [KLCF11]	search in möbius transformations	✓				✓		✓		✓		pairwise isometries average geodesic distance
Lipman et al. 2010 [LCDF10]	spectral analysis in correspondence space	✓	✓			✓		✓		✓	✓	symmetry aware embedding rigid
Lasowski et al. 2009 [LTSW09]	belief propagation	✓	✓			✓		✓		✓		segmentation isometries
Mitra et al. 2010 [MBB10]	multi-dimensional scaling	✓				✓	✓			✓		symmetry groups isometries discrete Laplacians
Mitra et al. 2006 [MGP06]	transformation voting	✓	✓			✓	✓			✓		segmentation, hierarchy similarity transform. curvature
Martinet et al. 2006 [MSHS06]	generalized moment functions	✓	✓			✓		✓		✓		pairwise, hierarchy rotations, reflections moments
Ovsjanikov et al. 2008 [OSG08]	search in signature embedding	✓				✓		✓		✓		pairwise isometries global point signatures
Pauly et al. 2008 [PMW*08]	transformation voting	✓	✓			✓	✓			✓		symmetry groups similarity transform. curvature
Podolak et al. 2006 [PSG*06]	symmetry transform computation			✓	✓	✓		✓		✓		pairwise reflections
Raviv et al. 2007 [RBBK07]	generalized multi-dimensional scaling	✓				✓			✓	✓		pairwise isometries geodesic distances
Raviv et al. 2009 [RBBK09]	generalized multi-dimensional scaling	✓	✓			✓			✓	✓		pairwise isometries geodesic distances
Raviv et al. 2010 [RBS*10]	matching of distance histograms	✓				✓			✓	✓		pairwise isometries diffusion distances
Simari et al. 2006 [SKS06]	reweighted least squares auto alignment	✓				✓	✓			✓		hierarchy reflections PCA axes
Sun et al. 1997 [SS97]	search in orientation histograms		✓	✓	✓	✓		✓		✓		pairwise reflections, rotations extended Gaussian image
Thrun et al. 2005 [TW05]	symmetry space scoring			✓		✓			✓	✓		segmentation reflections, rotations
Xu et al. 2009 [XZT*09]	voting for symmetry transforms	✓				✓			✓	✓		segmentation reflections SDF
Zabrodsky et al. 1995 [ZPA95]	symmetry distance computation	✓		✓		✓			✓	✓		pairwise isometries

**Table 1:** The table classifies the related work on symmetry detection according to the method used, type of input (meshes, point sets, volume data, and images), and type of output (global vs. partial, discrete vs. continuous, extrinsic vs. intrinsic, structure of the symmetries, class of transformations, and features used.)

data. We first formalize the problem of symmetry detection and present several classification categories to highlight the key similarities and differences of existing symmetry detection algorithms. In contrast to the classical theory discussed above, we now consider practically important aspects such as approximate symmetry and partiality.

**Correspondences as Building Blocks.** The elementary building block for *symmetry detection* is the identification of matching geometry: Given a shape, the goal is to identify and extract pairs of *regions* such that each pair of regions, under an appropriate distance measure, is *similar* when the respective regions are aligned using an *allowable transformation*. Specifically, given a geometric model  $M$ , the goal is to identify subsets  $M_1, M_2 \subseteq M$  such that  $M_1 \approx T(M_2)$ , where  $T$  denotes an allowable transformation and  $\approx$  denotes equality under the chosen distance measure. In case of partial symmetry detection, the surface patches are often required to be non-overlapping, i.e.,  $M_1 \cap M_2 = \emptyset$ . In this survey, we focus on shapes represented as surfaces, e.g., point cloud data, triangle meshes, or NURBS surfaces, rather than 2D images or volumetric data.

Research efforts target variations of the symmetry detection problem primarily based on the choice of (i) how the shape is segmented, (ii) how distance between two surface patches is measured, and (iii) what classes of transformations are allowed to bring surface patches into alignment (see Table 1 for a classification of recent related work). The symmetry detection problem is challenging because we have to simultaneously segment the shape and establish correspondence across the resultant segments, while solving for the respective aligning transforms. Note that even the decoupled versions of the problem are non-trivial with various solution strategies: we refer the readers to respective surveys on mesh segmentation [Sha08] and surface correspondence [vKZHC01]. Next we discuss some common variants of the problem.

**Global vs. Partial Symmetries.** For *global symmetry* detection we seek transformations that map the *whole* object to itself, i.e.,  $M_1, M_2 = M$ . Consequently, we do not have to solve the segmentation problem, which greatly simplifies the symmetry detection process.

For global symmetries of a finite object the centroid of an object is a fixed-point, i.e., the point is invariant under the shape's symmetry transformations. For example, symmetry rotations have the object centroid as rotation centers, while planes of reflection must pass through the object centroid. Methods for global symmetry detection exploit this property to significantly reduce the search space.

While a number of common shapes exhibit global symmetries (see Figure 1), self-similarities often occur only on parts of a shape. In order to capture these regularity patterns and enable a fine-grain analysis of geometric objects, we consider *partial symmetries*. There are two aspects to

partial symmetries. Symmetry can be restricted to a subset  $M' \subseteq M$  as shown in Figure 5(a). If we consider  $M'$  as a separate shape, then we can apply the notion of symmetry groups as defined above. Symmetry detection thus amounts to segmenting the shape into subsets that exhibit global symmetries represented by a transformation group.

In many instances, however, we do not have a complete symmetry as defined by a symmetry group. For example, translational structures in bounded shapes are very common, such as the repetitive patterns of the steps of the stairs shown in Figure 5(b). For such a structure, we can find a transformation that maps, e.g., the three lower-most steps to the three upper-most ones, but there is no self-similarity of the entire set of steps, except for the identity transform. Specifically, we say a shape  $M$  has a *partial symmetry* with respect to a transformation  $T$ , if there exist two subsets  $M_1, M_2 \subseteq M$  such that  $T(M_1) = M_2$ . This definition coincides with the definition of a global symmetry if  $M_1, M_2 = M$ , thus global symmetry is a special case of partial symmetry.

Partial symmetry allows handling a broader class of symmetries, but typically does not preserve the group structure. However, we can classify partial symmetries by the smallest group that contains the partial symmetry transformations. Conceptually, we can compute the closure of the symmetry set with respect to composition, which is analogous to repeating the pattern to infinity (or until a full rotation is achieved for a cyclic rotation group) as illustrated in Figure 5.

**Exact vs. Approximate Symmetries.** In another axis of variation, we look at the notion of equivalence  $\approx$  under transformations. Physical objects as shown in Figure 1 are typically not symmetric in a precise sense; according to the definition given in Section 2, none of these objects is symmetric. Deviations in biological growth, manufacturing imprecisions, or stochastic fluctuations in physical processes commonly invalidate a precise symmetry transformation. These shapes, however, exhibit strong regularity patterns that are close to ideal symmetries.

In order to enable more practical symmetry detection algorithms, we need a mathematical definition of approximate symmetry that is suitable for computation. For this purpose we define a distance function  $d(M, T(M))$  that measures the distance between the two shapes  $M$  and  $T(M)$ . We say that a shape  $M$  is  $\epsilon$ -symmetric with respect to a transformation  $T$ , if  $d(M, T(M)) < \epsilon$ . For  $\epsilon = 0$  we obtain the definition of exact symmetry, i.e.,  $T(M) = M$ .

Different variants of distance functions have been proposed. Zabrodsky et al. [ZPA95] introduced a measure of approximate symmetry as the “minimum effort required to transform a given object into a symmetric one”. This effort of transformation is quantified through a distance function that computes the mean of squared distances of each point on a shape to the location on the closest symmetric shape. If



(a) complete symmetry group on parts of a shape



(b) partial translational symmetry



(c) partial rotational symmetry

**Figure 5:** Partial symmetries commonly occur in geometric data sets.

a shape is defined through a scalar-valued function  $f$ , e.g., a gray-scale image, then the closest symmetric function is simply the average of  $f$  and  $T(f)$  [PSG\*06]. Thus the symmetry distance can be defined as

$$d(f, T) = \left\| f - \frac{f + T(f)}{2} \right\| = \left\| \frac{f - T(f)}{2} \right\|.$$

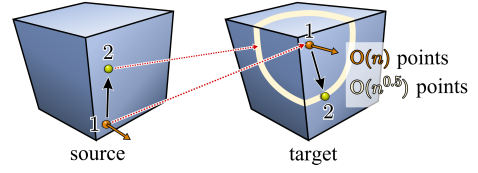
A distance measure that is often used in shape matching and registration is defined as,

$$d(M, T(M)) = \int_{\mathbf{x} \in M} \|T\mathbf{x} - \Gamma_M(T\mathbf{x})\|^2 d\mathbf{x}, \quad (1)$$

where  $\Gamma_M(\mathbf{y})$  is the closest point from  $\mathbf{y}$  on the surface  $M$ . Note that this is not a symmetric definition. Another popular definition is the Hausdorff distance

$$d(M, T(M)) = \max \left\{ \sup_{\mathbf{x} \in M} \inf_{\mathbf{y} \in T(M)} \|\mathbf{x} - \mathbf{y}\|, \sup_{\mathbf{y} \in T(M)} \inf_{\mathbf{x} \in M} \|\mathbf{x} - \mathbf{y}\| \right\}.$$

For a given transformation  $T$  and subset  $M' \subseteq M$  we can compute the symmetry distance using suitable discretizations of the continuous measures defined above. For exam-



**Figure 6:** A rigid correspondence is determined by a surface point and a direction. In practice, we expect that an object encoded by  $n$  points has no more than  $O(n^{1.5})$  candidate transformations that a brute-force search algorithm would need to test.

ple, the integral distance measure of Equation 1 can be approximated as

$$d(M, T(M)) \approx \frac{A(M)}{|P|} \sum_{\mathbf{p}_i \in P} \|T\mathbf{p}_i - \Gamma_P(T\mathbf{p}_i)\|^2,$$

where,  $A(M)$  is the surface area of  $M$  and  $P$  is a uniform sampling of  $M$  with  $|P|$  samples.

Another difficulty with approximate symmetries is that the group structure is not preserved — it is well possible that  $d(M, T_1(M)) < \epsilon$  and  $d(M, T_2(M)) < \epsilon$ , while  $d(M, (T_1 T_2)(M)) > \epsilon$ . Hence  $\epsilon$ -symmetries are not closed under composition and thus do not form a group. Alternately,  $A \approx B$  and  $B \approx C$  do not necessarily imply that  $A \approx C$ .

For a partial and approximate symmetry to be meaningful, the size of the symmetric subset needs to be sufficiently large, and the approximation threshold  $\epsilon$  sufficiently small. We say a tuple  $(M', T)$  with  $M' \subseteq M$  is a  $(\epsilon, \delta)$ -symmetry, if  $d(M', T(M')) < \epsilon$  and  $\delta = |M'|/|M|$ , where  $|\cdot|$  denotes an area measure. In general, we want to find symmetry transformations that minimize  $\epsilon$  and maximize  $\delta$ . A higher approximation threshold  $\epsilon$  typically allows larger symmetric patches and vice versa. The appropriate tradeoff depends on the specific application.

**Intrinsic vs. Extrinsic Symmetry.** Another mode of distinction between symmetry detection paradigms is how distance is measured. In most cases, we consider Euclidean distance between points. In few cases, however, an intrinsic distance measure is more appropriate: for example, human bodies undergoing isometric deformations. We return to this topic later in Section 7.

#### 4. Complexity

In this section, we consider a simple “brute-force” baseline algorithm in order to understand the complexity of the symmetry detection problem and to motivate the need for more sophisticated algorithms discussed in the subsequent sections.

The simplest option is to uniformly sample the space of



all allowed transformations, and then for each such transformation  $T$  check whether  $d(M, T(M)) < \epsilon$ . Since such a distance function typically has many local minima, a local gradient descent optimization from a sparse sampling of the transformation space is unreliable. On the other hand, since the space of transformations is high-dimensional (e.g., 6-dimensional for rigid transforms), naively sampling the space can be highly inefficient. Instead, any practical algorithm first analyzes the input to identify good regions of the transformation space to investigate.

To analyze the complexity of the symmetry detection problem, let us assume we have a uniformly sampled point set  $P$  consisting of  $n$  points that represent a shape  $M$ . Suppose we want to detect if  $M$  is globally symmetric under rigid transformations. Since a rigid transformation is uniquely determined by three pairs of (non-degenerate) corresponding points, a brute force approach can be as follows: We randomly select a triplet of points from  $P$  and try out all possible  $\binom{n}{3}$  correspondences from points in  $P$ . For each correspondence assignment, we verify if  $M$  is globally symmetric under the action of the corresponding transformation  $T$ , i.e., we evaluate if  $d(M, T(M)) < \epsilon$ . The verification test requires  $O(n)$  distance computations, with each distance computation taking  $O(1)$  (for example using an  $\epsilon$ -grid as approximate search data structure, and excluding degenerate cases). Thus, our brute force approach detects global symmetry in  $O(n^4)$ . This estimate is an upper bound and is rather conservative: If reliable surface normal information is available, two corresponding point pairs are sufficient to fix a rigid mapping that matches the local surface orientation. This reduces the cost to  $O(n^2)$  trials with  $O(n)$  cost each for verification, i.e., leads to a total complexity of  $O(n^3)$ .

In practice, further optimizations are possible for non-degenerate geometry: Rigid transformations preserve Euclidean distances, which excludes all matches of point pairs where the pairwise distance changes. Thus we only need to fix one point and a tangential direction. Fixing a pair of source points and one target point, the second target point should be chosen from a sphere intersecting the geometry, a set that typically resembles a distorted circle with typically  $O(n^{0.5})$  points. In other words, we would expect not more than  $O(n^{0.5})$  relevant orientations, reducing the complexity of finding transformation candidates to  $O(n^{1.5})$  trials (Figure 6 illustrates the idea). The  $O(n)$  cost for verification can be addressed by random sampling: If the current proposal for a transformation  $T$  is wrong, it is likely that a random point  $\mathbf{x} \in P$  will not be mapped close to  $P$ . We can expect that the likelihood of not detecting a mistake will drop exponentially with the number of trials, rejecting wrong matches in expected  $O(1)$  time (we still need  $O(n)$  time for correct and nearly-correct solutions). Thus, in a practical scenario, with non-degenerate geometry, this would lead to an algorithm with expected costs of  $O(n^{1.5})$  for finding a global rigid symmetry.

In the case of partial symmetry the complexity is higher. First, we need to try all points of the object as source points for finding a matching transformation, increasing the (expected) number of transformations to  $O(n^{2.5})$ . Furthermore, we cannot any longer rely on the randomized verification, as this will miss partial symmetries more easily. The overall complexity estimate therefore amounts to  $O(n^{3.5})$  point-to-point comparisons.

Classes of allowable transformations with additional degrees of freedom further increases the complexity: Similarity mappings (rigid motion + scaling) requires triplets rather than pairs to be matched. Affine mappings (a general linear transformation) requires a match of four point pairs, increasing the costs accordingly (global symmetries require testing of  $O(n^4)$  candidate transformations, partial symmetries require  $O(n^5)$  candidates).

The goal of the algorithms presented in the following is to find pairwise correspondences more efficiently. The key will be to find the relevant transformations more quickly rather than by applying a brute-force search.

## 5. Global Symmetry Detection

The main focus of this survey is on partial symmetry detection methods (see Section 6). For completeness, we briefly mention a few global symmetry detection algorithms that have been proposed for point clouds, triangle meshes, or volumetric grids. Global Euclidean symmetries for finite objects can only occur as reflections or rotations. Models exhibiting such symmetries share an important property: the planes of reflection and/or the axes of rotation pass through their center of mass. This property greatly reduces the search space for symmetry extraction.

Theoretical characterization of symmetry detection algorithms has been a topic of interest in computational geometry. Atallah et al. [Ata85] propose an  $O(n \log n)$  optimal algorithm for enumerating all reflective symmetries of a planar figure consisting of segments, circles, and points. Alt et al. [AMWW88] present efficient algorithms to handle rigid transformations and also general congruences for point sets.

The computation of global symmetries can be further simplified if reliable global shape descriptors can be computed. Popular methods detect  $n$ -fold rotational symmetry based on the correlation of the extended Gaussian image [SS97], Fourier analysis to handle voxel data [KCD\*03], spherical harmonic coefficients [KFR04], or moment coefficients [TMS09].

Martinet et al. [MSHS06] propose an efficient method for global symmetry detection using generalized even moments. Given a surface mesh  $M$ , the generalized moment of order  $2p$  in a direction  $\mathbf{v}$  is given by,

$$\mathcal{M}^{2p}(\mathbf{v}) = \int_{\mathbf{x} \in M} \|\mathbf{x} \times \mathbf{v}\|^{2p} d\mathbf{x}.$$

For a shape that is symmetric about an axis  $\mathbf{v}$ , they show that  $\nabla \mathcal{M}^{2p}(\mathbf{v}) = 0$ , i.e., the parameters of the global symmetry transformations can be extracted as the directions that lead to zero gradient of the moment functions. Since this condition on the gradient can produce false positives, they verify the candidate solutions in the last step of the algorithm.

Raviv et al. [RBBK07] employ generalized multi-dimensional scaling to detect global intrinsic symmetries. They find an embedding that preserves the geodesic distances on the object as corresponding Euclidean distances, thereby reducing the isometric symmetry detection problem into an instance of Euclidean symmetry detection. An alternative strategy has been proposed by Ovsjanikov et al. [OSG08], who use eigenfunctions of Laplace Beltrami operators to transform intrinsic symmetries of a shape into Euclidean symmetries in a signature space. As feature points they use a restricted version of a Global Point Signature (GPS) embedding of manifold introduced by Rustamov [Rus07b].

## 6. Partial Symmetry Detection

As discussed in Section 3, global symmetry detection is a special case of partial symmetry detection. We now discuss five main approaches for partial symmetry detection: The geometric hashing algorithm of Gal and Cohen-Or [GCO06], the transformation space voting schemes of Mitra et al. [MGP06] and Pauly et al. [PMW\*08], the planar reflective symmetry transform of Podolak et al. [PSG\*06], the graph-based approach of Bokeloh et al. [BBW\*09b], and symmetry-factored encoding by Lipman et al. [LCDF10]. At a high level, these methods share many similarities, even though the individual algorithmic components are different. We identify three main stages that appear more or less prominently in all these methods:

1. **feature selection** restricts the computations to a small set of relevant geometric features,
2. **aggregation** accumulates local symmetry information in a suitable data structure, and
3. **extraction** retrieves meaningful partial symmetries from the aggregate symmetry representation.

Feature selection typically uses local shape descriptors to significantly reduce the search space by considering geometric properties of the shape that are invariant under the considered symmetry transformations. For example, rigid transforms preserve principal curvatures [Car76]. As a consequence, if two points have significantly different curvatures, then no rigid transformation can sufficiently align the local surfaces around the points, and hence the points cannot be symmetric counterparts. In general, feature descriptors should satisfy the following requirements: The features should be (i) distinctive and characteristic of local geometry, (ii) invariant under the considered symmetry transformations, (iii) efficiently computable, and (iv) robust under missing data, noise, and outliers.

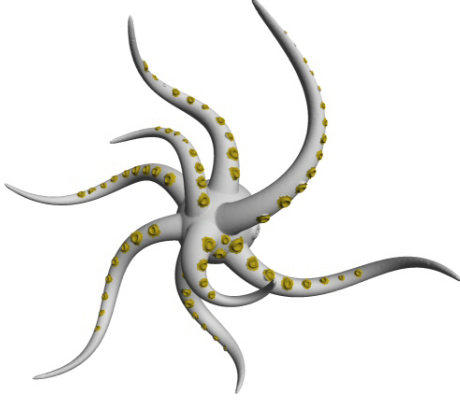
For symmetry under rigid transformations, a natural choice for a feature descriptor are differential properties of the surface that are commonly averaged over the local feature area. Curvatures can be reliably estimated from mesh representations using various approaches including curvature tensors [Tau95], osculating jets [CP03], finite-differences [Rus04], polynomial approximations [MGPG04], integral invariants [PWY\*07], or heat kernels [SOG09]. Under intrinsically isometric transformations, Gaussian curvatures are preserved at the surface point and hence are commonly used as local features. Heat kernels, which are also functions of Gaussian curvature for small neighborhoods, are also employed as they can be robustly computed.

However, even with feature-based pruning, an exhaustive validation of all remaining symmetry candidate transformations is computationally too expensive. A successful strategy is to further reduce the search space by aggregating local information of symmetries in a suitable representation of the symmetry *transformation space*. Given the aggregated local symmetry information, different strategies can be employed to extract the most likely candidates for meaningful partial symmetries. The extracted symmetry candidates are finally verified and refined in the spatial domain commonly leading to a segmentation of the model.

### 6.1. Geometric Hashing

A fundamental technique often employed for indexing is geometric hashing. Gal and Cohen-Or [GCO06] propose a solution for self-similarity or symmetry detection based on this approach. Originally introduced by Schwartz and Sharir [SS87], geometric hashing was later popularized by Wolfson [WR97] in the context of object retrieval. The algorithm proceeds in two main stages: In a preprocessing phase, a large hash table is constructed consisting of a collection of features. In the query phase, the hash table is then used to efficiently prune out unnecessary retrievals for testing matches.

**Feature selection.** Gal and Cohen-Or use a local feature descriptor based on surface curvature. First, they use least squares to locally fit quadratic patches to vertex neighborhoods of a given mesh. The curvature tensor and curvature derivatives at vertices are then computed using the coefficients of the fitted quadratic patches and the mesh vertices are sorted according to the magnitude of the Gaussian curvature, i.e., the product of principal curvature values. Subsequently, they build local patches around the vertices by greedily adding neighboring faces to the current patch as long as the patch can be quadratically approximated, within a user-defined threshold. Finally, they mark patches as *salient* based on a score computed using an empirical combination of curvature values associated with a patch, the number of curvature extrema in the patch, the curvature



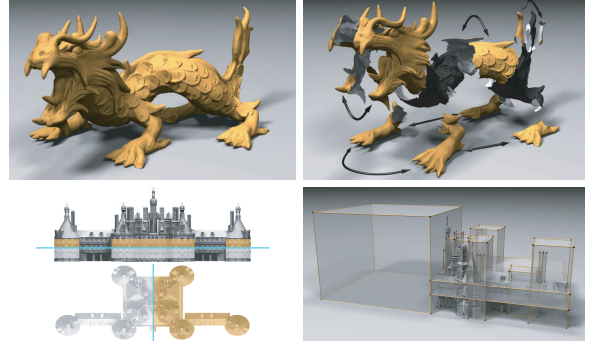
**Figure 7:** The suction cup on the tentacle of the octopus is identified as a salient feature and its similar occurrences are detected [GCO06]. (Images courtesy of Gal et al.)

variance in the patch cluster, and the patch area. This rather sophisticated method of feature selection is necessary for sufficient pruning in order to reduce the memory overhead for storing the index tables.

**Aggregation.** They aggregate transformations based on the following observation: several transformation families have only a few degrees of freedom and are uniquely determined by a small number of correct corresponding point pairs. For example, rigid transformations are uniquely specified by only three corresponding point pairs [GG04]. So for rigid transformations, they pick all sets of triplets of points as *bases*, and use each base to bring the point set to a canonical position. For each choice of base, the remaining points in their canonical positions define a hash function, where each point is indexed based on their spatial location. When a base is so chosen that the corresponding point sets indeed match, a large number of point pairs will be in agreement. Based on this observation, they then handle query objects by probing against this hash table. Essentially, this can also be seen as an example of voting, but the cost of creating the data structure is very high since the entire space of possible matching space is voxelised for hashing.

**Extraction.** In the geometric hashing step, they first bring each query patch to a canonical position by indexing a small number of rotation-invariant features. The remaining points in the aligned pointset are tested against the geometric hashing grid to check if there is any consensus for a matching base index. The method determines the best aligning transformation from a large set of transformations using a voting scheme, quantizing the transformation space using a six-dimensional table.

Suppose using the current base, the aligned point set positions are given by  $\{\mathbf{p}_i\}$ . They first retrieve the cells  $\{c_i\}$  where the aligned points lie, and for each cell  $c_i$  find the base



**Figure 8:** Symmetry detection results under similarity transformations (rotation, translation, reflection, and scaling) on 3D geometry using a transformation domain voting scheme [MGP06].

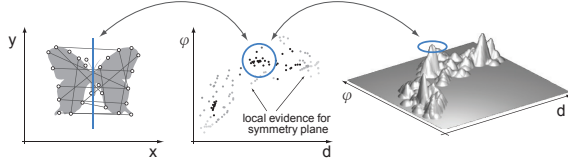
indices  $B_i = \{b_j^i\}$  that vote for the current cell. A consensus is reached when there is a significant number of cells voting for any particular base(s). A consensus, if found, yields a potential aligning transformation using the transformation between the starting base and the matched base index. This process is repeated over all possible choices of bases to list the candidate transformations. In practice, instead of iterating over all possible bases it is sufficient to try a random selection of bases [WR97, GCO06, AMCO08]. Figure 7 shows an example of searching for similar occurrences of a query patch in an input model using this approach.

## 6.2. Transformation Space Voting

Mitra et al. [MGP06] propose a method for computing pairwise partial and approximate symmetries based on accumulating local symmetry votes in a symmetry transformation space. The method was extended by Pauly et al. [PMW\*08] to handle repetitive patterns based on one- and two-parameter transformation groups.

**Feature selection.** An important aspect of the voting scheme is to aggregate local geometric information to capture even small-scale symmetries. They employ simple feature descriptors based on differential geometry invariants: Principal curvatures  $\kappa_1$  and  $\kappa_2$  for rigid transformations, and the ratio of principal curvatures  $\kappa_1/\kappa_2$  for similarity transformations.

**Aggregation.** They map point-pairs with matching signatures of the original object to points in a transformation domain. When considering mirror symmetries, the algorithm exploits the fact that any two points  $\mathbf{p}$  and  $\mathbf{q}$  define a unique plane of reflection. This plane is the orthogonal bisector of the two points, i.e., passes through the point  $(\mathbf{p} + \mathbf{q})/2$  and has normal vector  $(\mathbf{p} - \mathbf{q})/\|\mathbf{p} - \mathbf{q}\|$ . Notice that the orientation of the normal vector has a flip ambiguity. The reflective



**Figure 9:** To detect symmetries in geometric models, the boundary of the shape is uniformly sampled (left) [MGP06]. Every pair of samples with compatible local surface geometry provides local evidence for a symmetry transformation (center). In this example we consider reflections that are parameterized by an angle  $\phi$  and the distance  $d$  to the origin. Accumulating such evidence using a clustering approach yields the dominant symmetries of the model (right).

plane can be represented as a point in a 3D space consisting of two angles that define the normal vector and the distance of the plane from the origin. Thus a pair of spatial points gets mapped to a point in the 3D space of reflective planes, called *transformation space* (see Figure 9). This idea does not immediately generalize to handle rigid transformations as a point pair alone is not sufficient to uniquely determine an aligning transformation. Instead, in order to extract a unique aligning transformation, they encode the local neighborhood at each point, using the intrinsic local frame composed of the surface normal and the principal curvature directions. Since umbilical points, i.e., points with locally spherical neighborhoods, do not have uniquely defined principal curvature directions, they are left out of the point-pairing. In the case of rigid transformations each point-pair along with the respective intrinsic coordinate frame produces a rigid transformation parameterized by a translation vector and three Euler angles, i.e., a point in a 6D transformation space.

**Extraction.** Each point in transformation space can be interpreted as a witness or vote for a symmetry transformation. To extract meaningful symmetries at larger scales such local evidence is accumulated, i.e., groups of pairs with a similar transformation are found that correspond to symmetric subsets of the model surface. Suppose the transformation space is equipped with a distance metric, i.e., for any two transformation points  $T_i$  and  $T_j$ , their distance  $d(T_i, T_j)$  is defined (typically using Euclidean distance). Under such a distance measure, *dense* regions in the transformation space can be identified using a clustering approach. For this purpose, Mitra et al. [MGP06] use a mean-shift clustering method, which does not require a priori knowledge of the number of potential clusters or equivalently number of potential symmetry transformations. The kernel width of the mean-shift clustering method is a user parameter that controls the approximation level of the detected symmetries. For a fixed approximation level, the strength or height of clusters relates to the number of votes the region receives in the transformation space. Each cluster corresponds to a potential pairwise par-



**Figure 10:** Regular structures discovered using transformation domain grid-detection [PMW\*08] on an amphitheater, a nautilus shell, a laser-scan of a building, and a procedurally generated helical segment.

tial symmetry of the shape and the extracted cluster centers act as symmetry transformation candidates that are subsequently validated and refined. Figure 8 shows example symmetries detected by this method under similarity transformations.

Transformation domain voting can be extended to detect one- and two-parameter patterns as demonstrated by Pauly et al. [PMW\*08]. The algorithm is based on the observation that regularity in spatial domain reveals itself as a lattice structure in the transformation domain. Such lattices can be effectively extracted by analyzing appropriate 2D planar slices passing through the origin of the 6D transformation space. A robust grid fitting algorithm is then applied on the 2D slices to estimate the generating transformations of the symmetry lattice. Figure 10 shows typical results obtained using this approach.

### 6.3. Planar Reflective Symmetry Transform

Motivated by the notion of continuous symmetry introduced by Zabrodsky et al. [ZPA95], Podolak et al. [PSG\*06] investigate the notion of a symmetry transform under reflective transformations. They propose the *planar reflective symmetry transform* (PRST) to encode a continuous notion of symmetry of an object about any reflective line in 2D, or about any reflective plane in 3D. Given a reflective plane  $\gamma$ , the PRST for a function  $f$  is defined as:

$$\text{PRST}^2(f, \gamma) := 1 - d(f, \gamma(f))^2 / \|f\|^2.$$

The symmetry distance  $d(f, \gamma(f))$ , as originally introduced by Zabrodsky, is defined as the distance of the function  $f$  to the closest function  $g$  that is symmetric with respect to the transformation defined by  $\gamma$ , i.e.,  $d(f, \gamma(f)) := \min_g \|f - g\|^2$ , such that  $\gamma(g) = g$ . Computationally, the closest function is simply the average of  $f$  and its reflection  $\gamma(f)$ , and hence the symmetry distance simply [KCD\*03]:

$$d(f, \gamma(f)) = \|f - (f + \gamma(f))/2\| = \|f - \gamma(f)\|/2.$$



This greatly simplifies the computation of the PRST reducing it to evaluating dot products:

$$\text{PRST}^2(f, \gamma) = \frac{1 + f \cdot \gamma(f)}{2},$$

where the function  $f$  is assumed to be normalized, i.e.,  $\|f\| = 1$ . Since the above definition applies to a function  $f$ , the algorithm transforms a given surface geometry into a suitable functional representation. A typical solution is to embed the surface in a volume grid and rasterize the surface using a Gaussian blur kernel to smooth out effects of noise and small features. An alternative is to use the Gaussian Euclidean Distance Transform (GEDT) as proposed by Kazhdan and colleagues [KFR04]. Podolak et al. [PSG\*06] observe that functions arising from rasterized surfaces are typically sparse over the volume grids, and propose a Monte Carlo sampling algorithm to efficiently compute the PRST. Figure 11 shows the PRST for several simple curves. The definition has subsequently been extended by Rustamov [Rus07a] to incorporate local neighborhood of points or *contexts* using a scale factor for controlling the neighborhood range.

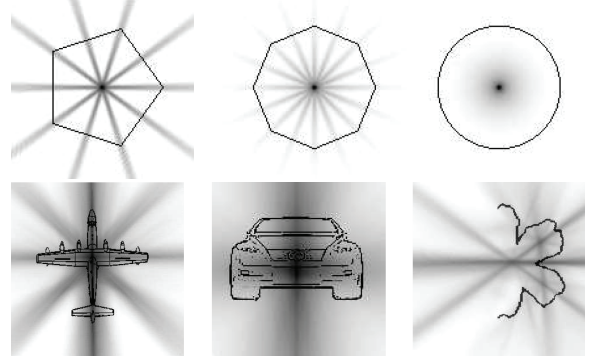
**Feature selection.** PRST being a symmetry transform assigns scores to all possible reflective planes, and hence does not involve any feature selection to limit the possible choices of potential candidate transformations.

**Aggregation.** In practice, the PRST is evaluated over a discrete space using a volume grid discretization. Typically the local maxima of the symmetry function are of interest, as they reveal potential principal reflective symmetries, both partial and approximate. The PRST values are computed over a medium resolution volume grid, and then candidate maxima are identified via a thresholding step. Instead of a global threshold value, Podolak et al. suggest using a lower threshold near the edges of the boundary than near its center. They use a threshold proportional to  $(1 - r/R)$ , where  $R$  is the radius of the object, and  $r$  is the distance of the candidate plane of reflection from the center of mass of the object. Additionally, shallow maxima, typically caused by noise, are suppressed using a discrete Laplacian smoothing of the computed PRST. Potential symmetry candidates are locally refined using the Iterated Closest Point (ICP) algorithm [BM92, CM92, MGPG04].

#### 6.4. Graph-Based Symmetry Detection

Instead of operating at the level of sample points, it is sometimes more practical to work at the level of feature curves, in particular for data sets where these feature curves can be extracted robustly. This leads to substantial reduction in computational overhead, since the number of feature curves is typically significantly smaller than the number of sample points.

Bokeloh et al. [BBW\*09b] explore this idea by mapping



**Figure 11:** Planar Reflective Symmetry Transforms (PRST) for various curves. For visualization, each pixel is colored by the score of the plane with maximum symmetry score among all the planes passing through the pixel. (Images courtesy of Podolak and et al. [PSG\*06]).

the symmetry extraction problem to a mapping problem of a network of curves. They extract line features that form a very sparse approximation of the original geometry, arguing that for many objects the crease lines carry the most important information. Feature lines are then used to define local coordinates or bases. Matching subsets of such bases then amounts to arrangements of line features with compatible local neighborhoods. Such matching subsets are treated as potential symmetry candidates and are subsequently tested and extracted in a validation stage. Although working with a sparse set of linear features makes the approach substantially more efficient than directly comparing local geometry, the approach assumes input models with a dominating set of such characteristic linear features. The graph of linear features is built using detected feature lines as nodes and using an edge set obtained by connecting k-nearest line segment neighbors.

**Feature selection.** As a general criterion for a line feature, the local geometry on the line should be invariant under continuous motions along that line. The algorithm considers rigid motions as potential symmetry transformations and detects lines by slippage analysis [GG04]. Effectively, the method finds regions that map to themselves under infinitesimal rotations and translations, yielding straight and circular line directions. In order to place the actual features, points of maximum principal curvature orthogonal to the line direction are chosen. The actual detection is formulated as a moving least-squares problem that sparsely samples the geometry and then iteratively moves points within line-regions towards curvature extrema. The sparse sampling yields rather moderate computation times. Other feature selection criteria have been proposed in the context of similar symmetry detection methods [BBW\*09a, GSMCO09, MZL\*09].



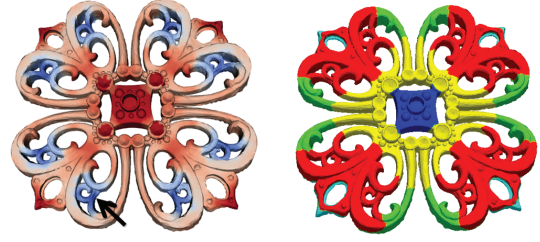
**Figure 12:** Symmetry detection result using line features and sub-graph matching [BBW\*09b]. Detected symmetric or repeating parts are highlighted.

**Aggregation.** The algorithm matches pairs of line features that intersect in a locally similar way (similar curvatures, similar angles). All such base-pairs yield potential transformations that could be checked: The combination of non-parallel lines gives a fixed coordinate frame for the initial transformation. The algorithm then proceeds by checking the line-pattern in the neighborhood iteratively, thereby refining the transformations by iterative alignment. For efficiency reasons, not all pairs of bases are checked but instead random sampling is applied. The underlying assumption is that the most salient symmetries are more likely to receive a random sample. By testing local line configurations in growing radii, false positives can be rejected rapidly.

**Extraction.** The previous step yields a list of potential transformations that are true positives with high probability as the main crease lines match in a local neighborhood. Potential transformations are verified using a region growing algorithm that compares actual points in terms of their distance to the matching surface and also fixes the final extent of the symmetric regions. They simultaneously grow regions that stop if symmetric areas collide, thereby partitioning the model into symmetric pieces. Such a partitioning, however, is non-canonical since it depends on arbitrary choices (such as where the region growing is started). Later, Bokeloh et al. [BWS10] introduce a similar algorithm that lifts this restriction but outputs overlapping symmetric areas.

### 6.5. Symmetry Factored Embedding

Instead of working in the transformation space, one can also work directly in the space of correspondences. Specifically, if a model is sampled with  $n$  points, then one can represent symmetry using a *symmetry correspondence matrix*  $C$  of  $n \times n$  non-negative entries, where each entry encodes symmetry relations between point pairs. Lipman et al. [LCDF10] make an important observation that such a matrix is sparse and all points symmetric to each other form an orbit, which



**Figure 13:** Symmetry factored distance and the symmetry orbits detected. (Left) Symmetry factored distance from the marked point (with arrow) to all other points on the model, with blue denoting small distance and red denoting large distance. Note that symmetric points are at similar distances. (Right) Segmentation in the symmetry-factored embedding space produces symmetry-aware segmentation. (Images courtesy of Lipman et al. [LCDF10].)

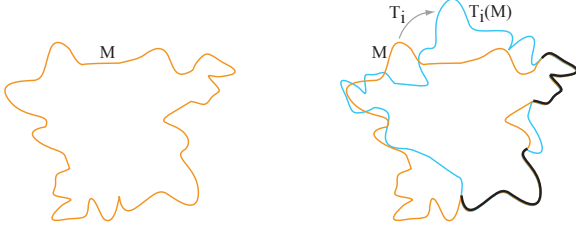
appears as a clique in the graph induced by the symmetry correspondence matrix  $C$ . They extract connectedness of the graph using spectral methods. Specifically, given a model, first a correspondence matrix is selected using a randomized algorithm similar in spirit to voting-based methods [MGP06]. The symmetry factored embedding is then defined using the spectral analysis of the correspondence matrix  $C$ . If  $C\psi_k = \lambda_k\psi_k$  for  $k = 1, \dots, n$  with  $\psi_k$  and  $\lambda_k$  are respectively the eigenvectors and eigenvalues of  $C$ , the embedding is defined as:

$$\Phi^t(x_i) := (\lambda_1^t \psi_1(x_i), \dots, \lambda_n^t \psi_n(x_i))$$

where,  $t$  is in the range of 20 – 80 (see also [NLCK05]). Subsequently, symmetry factored distance between any two points  $x_i$  and  $x_j$  on the mesh can be simply computed as the Euclidean distance in this embedded space, i.e.,  $d^t(x_i, x_j)^2 = \|\Phi^t(x_i) - \Phi^t(x_j)\|^2 = \sum_k \lambda_k^{2t} \|\psi_k(x_i) - \psi_k(x_j)\|^2$  (see Figure 13).

### 6.6. Symmetry Verification

**Local Refinement.** Most symmetry detection algorithms first detect symmetry transformations approximately, and later refine the transformations for improved accuracy. Suppose a mesh  $M$  is symmetric (partially) under transformation  $T$ . This implies that  $M$  and  $T(M)$  are closely aligned, though only parts of the object may be matched in case of partial symmetry. To further improve the matching, one can treat  $M$  and  $T(M)$  as two independent shapes and improve their alignment — this is the classic local registration problem. A popular solution to this problem is the Iterated Closest Point (ICP) algorithm [BM92, CM92, MGP04]. Suppose the local (rigid) registration brings  $M$  in alignment with  $T(M)$  under transformation  $\tau$ , i.e.,  $\tau(M) \approx T(M) \Rightarrow M \approx \tau^{-1}T(M)$ , assuming  $\tau$  is invertible. In case of symmetry de-



**Figure 14:** Under the action of transformation  $T_i$ , the input shape  $M$  gets mapped to  $T_i(M)$ . The overlap regions, shown in black, between  $M$  and  $T_i(M)$  are the parts that are symmetric under the action of transformation  $T_i$ . Overlap is measured with respect to a tolerance  $\epsilon$  as specified by the approximation level.

tection under rigid transformations, the refined rigid transformation is given by  $\tau^{-1}T$ . In case of reflective symmetry, we can factor in  $\tau$  appropriately to solve for the refined transformation plane. In case of line features the approach can be extended to Iterated Closest Line algorithm [BBW\*09b].

In the case of regularity detection, when instead of looking for isolated symmetries one searches for a pattern among the detected transformations, the refinement process is coupled, i.e., such local refinements amount to a simultaneous registration problem in the object domain. This is a generalization of ICP based refinement, but additionally 1-parameter or 2-parameter detected regularities are maintained [PMW\*08].

**Patch Extraction.** After extracting potential symmetry transformations and then refining the coarse estimates, the last step of most symmetry detection algorithms involves extracting patches of the mesh that are symmetric under the detected symmetry transformations. Recall that for given a mesh  $M$ , the goal is to identify regions  $\{\mathcal{R}_i\}$  of the mesh that are symmetric to other regions of the mesh under transformations  $\{T_i\}$ . The challenge is to simultaneously determine  $\{\mathcal{R}_i\}$  and  $\{T_i\}$ . Once  $\{T_i\}$  is known, however, the problem becomes much simpler. Thus given a transformation  $T_i$ , the problem amounts to finding the overlapping region  $\mathcal{R}_i$  between  $M$  and  $T_i(M)$  (see Figure 14).

The notion of overlap is defined up to a margin of approximation  $\epsilon$ . Let  $V$  be the vertex set of mesh  $M$ . The set of vertices  $\{v_i\} \in V$  can then be extracted such that for all the chosen vertices  $d(v_i, M) < \epsilon$  under an appropriate choice of distance. Finally, the selected points  $\{v_i\}$  can be grouped together into connected components using the original connectivity of the input mesh  $M$ . Most algorithms retain only the largest connected component as the symmetric patch  $\mathcal{R}_i$ , and analyze the remaining mesh for further symmetries.

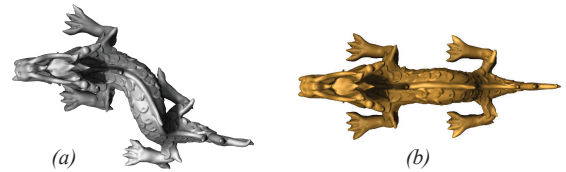
## 7. Intrinsic Symmetries

Euclidean symmetries depend on the embedding of an object in the ambient space. For example, while the geometry of the dragon in Figure 15(a) does not exhibit any global symmetries, the pose of Figure 15(b) exhibits a global mirror symmetry. Thus, the symmetries of the object change with changes of the embedding. In the spirit of the Erlangen Program initiated by Felix Klein, it is possible to avoid the dependence on the embedding by looking at a broader class of transformations. *Intrinsic* symmetry can be defined by considering self-isometries of a surface with respect to an intrinsic metric of the surface. For example, we can specify as an intrinsic metric the geodesic distance function  $d_M$  that measures the distance between any two points  $x, y \in M$  as the length of the shortest path between  $x$  and  $y$  on  $M$ . We say that a shape  $M$  is *intrinsically symmetric* with respect to a homeomorphism  $T : M \rightarrow M$ , if the intrinsic metric is preserved by the mapping, i.e., if  $d_M(x, y) = d_M(T(x), T(y))$  for all points  $x, y \in M$ .

Extrinsic, i.e., Euclidean, symmetries as defined previously are a subset of intrinsic symmetries, since all Euclidean transformations are isometries of the embedded surface. Note that whether extrinsic or intrinsic symmetries are more appropriate to characterize an object depends on the type of object and the specific application.

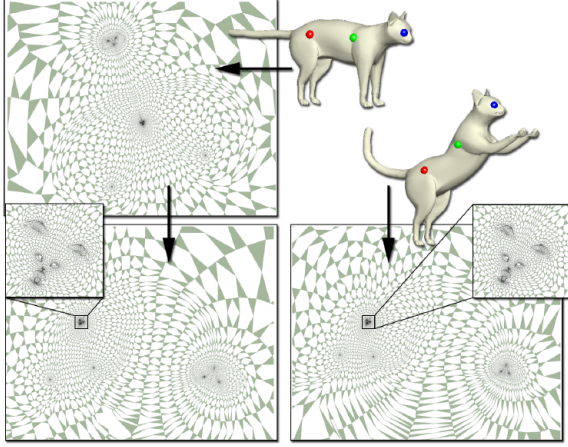
For isometric transformations, a natural choice for local descriptor is the Gaussian curvature. At any point on a surface Gaussian curvature can be evaluated using only length measurement on the surface, i.e., using the first fundamental form of the local surface, and hence remains invariant under isometric deformations [Car76]. A Gaussian curvature based feature descriptor has been used for performing non-rigid registration under isometric deformations by Huang et al. [HAWG08]. Xu et al. [XZT\*09] use shape diameter functions [SSCO08] computed on equally spaced geodesic rings around surface points as an alternative to conventional curvature estimates.

In the context of point-to-point correspondence between surface pairs with approximate and partial isometry, Lipman et al. [LF09] propose an automatic and efficient algo-



**Figure 15:** Changes of pose cause a change in extrinsic symmetries. While the pose in (a) does not exhibit a global mirror symmetry, a (nearly) isometric deformation exposes the global reflection (images from [MGP07]).





**Figure 16:** Isometric point-to-point correspondence between two meshes  $M_1$  and  $M_2$ . The meshes are first conformally flattened to the plane. Any pair of triplet of correspondences, then uniquely specifies a Möbius transform between the flattened meshes. Each such Möbius transform is ranked based on the votes from other mesh vertices, and the highest rated transform is retained as the isometric deformation linking  $M_1$  and  $M_2$ . Intuitively, a correct pair of correspondence triplets (like shown in red, green, blue) brings the flattened meshes to alignment (bottom row). (Images courtesy of Lipman and Funkhouser [LF09].)

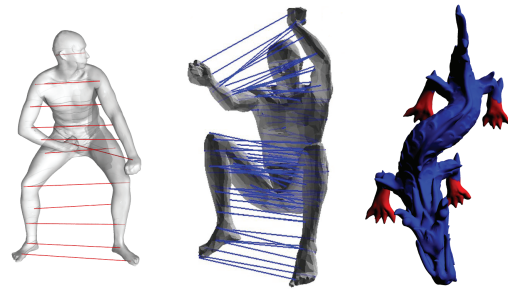
rithm (see Figure 16). They observe that isometry is a subset of Möbius group, which has only 6 degrees of freedom for genus zero surfaces and only 3 degrees of freedom for topological discs. Based on this observation, they efficiently extract candidate Möbius transforms, which are then verified to extract the best isometric deformation, thus producing point-to-point correspondence. Specifically, given two surfaces  $M_1$  and  $M_2$ , they first uniformly sample the surfaces to produce point sets  $P_1$  and  $P_2$ , respectively using same sample spacing. In a key step, they observe that a pair of triplets of points from the two point sets is sufficient to uniquely determine a Möbius transform linking  $M_1 \rightarrow M_2$ . Each such Möbius transform can then be used to *vote* for correspondence between shapes  $M_1$  and  $M_2$ . The algorithm successfully finds correspondence across model pairs even under isometric deformations resulting in large Euclidean deformations. Kim et al. [KLCF11] extend this algorithm to detect global intrinsic symmetry on 3D models (see Figure 17).

Earlier, Raviv et al. [RBBK07] employ generalized multi-dimensional scaling to compute a new embedding that best preserves the original geodesic distances on the object in the form of corresponding Euclidean distances in the new space. This mapping reduces the isometric symmetry detection problem to an instance of Euclidean symmetry detection. In another attempt, Ovsjanikov et al. [OSG08] use

eigenfunctions of Laplace Beltrami operators to transform intrinsic symmetries of a shape to Euclidean symmetries in the signature space. As feature points they use a restricted version of Global Point Signature (GPS) embedding of a manifold, originally introduced by Rustamov [Rus07b]. Lipman et al. [LCDF10] use the pointwise absolute values of the Laplacian eigenvectors as a measure of the global intrinsic similarity in their spectral clustering framework to extract global intrinsic symmetries. In a recent effort, Xu et al. [xu2] introduce a two-level clustering framework to detect partial intrinsic symmetries at multiple scales. They first perform a clustering to identify the different symmetry scales and then extract symmetries at each scale by a second-level spectral clustering. In another effort, Xu et al. [XZT\*09] extend PRST based reflectional symmetry detection [PSG\*06] and introduce an algorithm to extract partial intrinsic reflectional symmetries in 3D geometry. Given a closed 2-manifold mesh, they use a voting scheme to compute a scalar field, whose local extrema are accentuated to reveal local reflectional symmetry support along curves. An iterative refinement procedure combined with region growing is used to finalize the results and identify the isometrically repeating parts (see Figure 18). Alternately, Mitra et al. [MBB10] extract intrinsic regularity, where the repetitions are on an intrinsic grid, without any apparent Euclidean pattern to describe the shape, but rising out of (near) isometric deformation of the underlying surface. They employ multidimensional scaling to reduce the problem of intrinsic structure detection to a simpler problem of 2D grid detection, which is performed using an autocorrelation analysis.

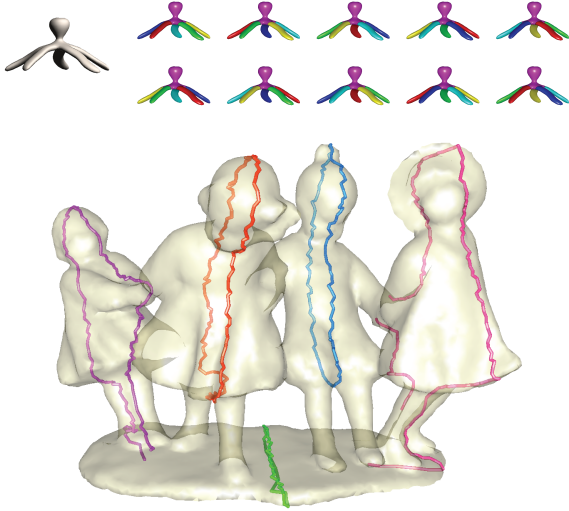
## 8. Encoding Extracted Symmetries

In addition to detecting symmetries, as discussed in the previous sections, it is also important to consider how the output is encoded. Independent of the specific symmetry detection algorithm, a number of different representations of the extracted symmetries are possible with the preferred representation depending on the target application.



**Figure 17:** Intrinsic symmetries detected on 3D geometry. (Images courtesy of Ovsjanikov et al. [OSG08], Kim et al. [KLCF11] and Lasowski et al. [LTSW09], respectively.)





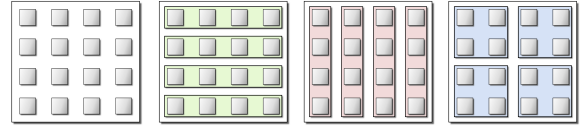
**Figure 18:** Intrinsic symmetries detected on 3D geometry. (Image courtesy of Raviv et al. [RBBK07] and Xu and et al. [XZT\*09], respectively.)

**Pairwise symmetries.** The simplest form of output is a list of pairwise symmetries, i.e., a list of the pairwise correspondences: For each such entry, the detection algorithm reports a transformation  $\mathbf{T}$  and a subset of points (or surface patch)  $M_1 \subset M$  such that  $\text{dist}(\mathbf{T}(M_1), M)$  is small and  $\mathbf{T}$  has desirable properties such as rigidity, isometry, similarity, etc. In the case of approximate symmetries, every point  $\mathbf{x} \in M_1$  might be additionally tagged with a confidence measure, e.g., based on the distance to the surface.

Pairwise symmetries couple parts of surfaces in two ways: First, they define a pairwise similarity measure for points on the surface. This can be used to non-locally transport information, such as denoising or simultaneous editing. Second, the information that two regions  $M_1, M_2 = \mathbf{T}(M_1)$  are coupled by a symmetry transformation  $\mathbf{T}$  can be exploited to characterize shapes for recognition or for structure-preserving editing.

Pairwise relations capture the symmetry structure of an object only indirectly. While the set of pairwise relations in principle encode all symmetries within an object, further processing is necessary to extract relevant aspects.

**Segmentation.** One option for representing the symmetry information is to segment the input shape  $M$  into symmetric *building blocks*. A simple implementation is through region growing (see [MGP06, BBW\*09b], Figure 12 shows a typical result): For each pairwise symmetry  $(M_1, \mathbf{T})$ , seed points  $\mathbf{x}, \mathbf{T}(\mathbf{x})$  are created and a simultaneous region growing is started, stopping when regions collide. The technique of Lipman et al. [LCDF10] processes the correspondence information by clustering consistent orbits, computing a symmetry-



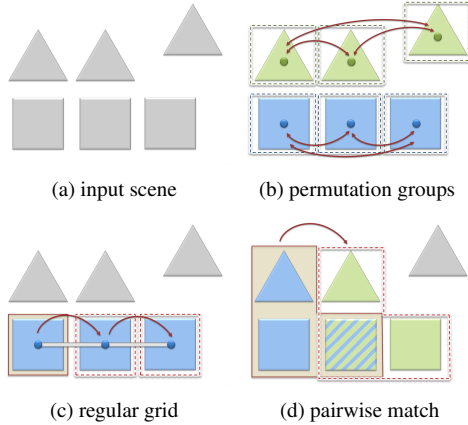
**Figure 19:** Establishing a hierarchical structure is ambiguous; there are usually several equivalent solutions that all explain the structure compactly.

aware distance function as output. Points that form consistent cliques of equivalent points (under the mappings considered) are embedded close to each other. In the (optional) segmentation step, such points will obtain the same label.

Segmentation into symmetric parts has a number of limitations: Hierarchy and regularity are not explicitly encoded, and the decomposition does not necessarily lead to building blocks that can be reassembled to create new composite shapes. Therefore a number of alternatives have been proposed, as described next.

**Hierarchy.** The segmentation algorithm can be augmented to provide a hierarchical encoding: Mitra et al. [MGP06] perform a hierarchical reduction where salient symmetric parts (indicated by strong responses in transformation space) are reduced to a single part, before recursively continuing the symmetry search. In another effort, Simari et al. [SKS06] present a folding tree data structure to encode non-redundant regions of the original mesh as well as the reflection planes by recursively applying a symmetry detection algorithm. The encoded structure can then be *unfolded* to recover back the original shape. Wang et al. [WXL\*11] refine the idea by using perceptual grouping for estimating the hierarchy. Such hierarchical encodings facilitate compression and structure-aware editing. In many cases, however, symmetries are not fully captured by a hierarchical model. For example, for a grid of  $4 \times 4$  windows, many, equally well-justified hierarchical decompositions exist (for example, see Figure 19). A purely hierarchical encoding therefore has to make choices during the construction of the hierarchy, making the representation non-canonical. Finding an optimal hierarchy with respect to compression (minimal coding costs) is NP-hard, e.g., a two-level hierarchy reduces to the NP-hard set-cover problem.

**Building blocks.** The issue of finding a segmentation that leads to building blocks that can be connected to form new shapes has been considered by Bokeloh et al. [BWS10]: Instead of cutting pieces at Voronoi cells of a region growing algorithm, their approach cuts at the boundaries of partial symmetries. The set of such boundaries leads to *docking sites* that form adapters for connecting alternative pieces. The proposed algorithm, however, still uses a non-canonical hierarchical encoding of the docking for efficiency reasons and only works for exact symmetries. Kalojanov et



**Figure 20:** Partial symmetry decomposes an input scene (a) into exchangeable pieces (b) [KBW\*12]. This yields regularity (a group structure) with respect to exchanging of parts. Regularity of transformations (c) is stricter and does not capture all redundancy. Looking at a single pairwise match (d), we see that it comprises pieces from multiple permutation groups. Due to the mixing of different types of regularity, the set of pairwise matches in general does not form a group and the matched regions might overlap partially.

al. [KBW\*12] study this idea more systematically: Similar to Lipman et al. [LCDF10], they consider cliques of equivalent points, induced by pairwise input matchings. They group points with the same set of outgoing transformations into an elementary building block, and consider all blocks within the same clique of transformations as one class of tiles. This yields “microtiles”, a canonical, non-hierarchical decomposition of pairwise matches into building blocks from which all partial symmetry information can be efficiently retrieved.

**Symmetry groups.** The structure of global, exact symmetries is understood by the formalism of symmetry groups, as discussed in Section 2. The idea is easily extended to finite excerpts of such groups, for example, to detect the regularity in a finite grid of windows of a building. The detection can be restricted to certain types of symmetry groups: For example, commutative symmetry groups (such as translational patterns) are concisely described by a set of generator transformations  $\mathbf{T}_1, \dots, \mathbf{T}_k$  that replicate a base instance  $M_1 \subset M$  by a set of transformations  $\{\mathbf{T}_1^{i_1} \circ \dots \circ \mathbf{T}_k^{i_k}, i_1, \dots, i_k \in \mathbb{Z}\}$ . This structure is isomorphic to an integer lattice (commutativity is important since it allows us to freely reorder the transformations). Such models are useful for shape analysis and structure-aware shape editing (e.g., unlike complex symmetry groups, grids can be easily augmented with additional elements) [PMW\*08, MP08, BWKS11].

For partial symmetry of shapes, usually no group structure exist in the set of transformations involved, as symmet-

ric pieces might be placed irregularly, and even regular arrangements are most of the time only excerpts such that the group is not closed. Kalojanov et al. [KBW\*12] show that in their decomposition into “microtiles” gives rise to canonical, maximal *permutation* groups of pieces. These building blocks can be exchanged with a maximal number of different instances (see Figure 20a,b). In their view, regularity of partial symmetry is captured best in the *actions* of the transformation on appropriately chosen subsets, not (necessarily) in the *transformations* themselves (see Figure 20 c). If we look at a specific pairwise match (under a single, fixed transformation), it will usually contain a union of multiple pieces from different such regularity patterns (see Figure 20d). Thus, the group structure is lost and this causes the confusions in how to interpret the resulting structure.

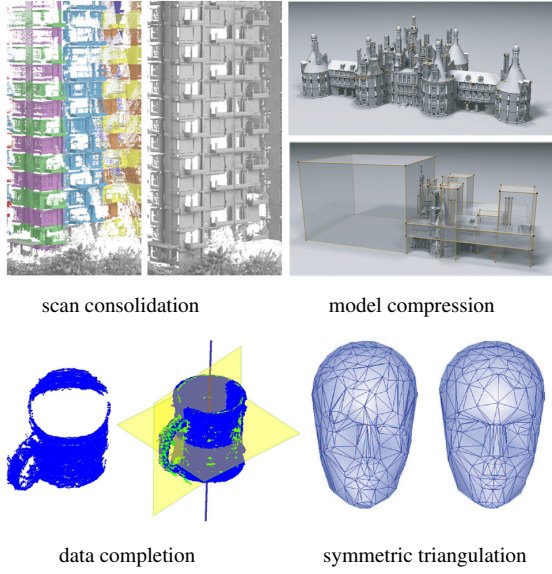
**Continuous symmetry.** For shapes that contain areas such as flat planar surfaces or straight edges, infinitely many partial symmetries exist, with a continuum of transformations. Continuous symmetries with respect to rigid motion have been studied by Gelfand et al. [GG04] who find *slippable motions* using zero eigenvalues of the Hessian matrix of a rigid auto-alignment problem (local patches of geometry are matched to themselves). The output is a segmentation into different types of slippable regions (with zero to three rotational and/or translational degrees of freedom). An analogue for intrinsic rather than extrinsic isometries (Killing vector fields) has been studied by Ben-Chen et al. [BCBSG10].

In summary, while different models are possible to structure the space of pairwise symmetries, the choice often depends on the target application. Importantly, while global and exact symmetries are well understood by models from group theory, we still lack a similar, canonical model for approximate and partial symmetry.

## 9. Applications

Extracted symmetries, global or partial, exact or approximate, essentially provide relations across different parts of a shape, as discussed in the previous section. Such high-level understanding of the input 3D geometry enables a range of applications, which are otherwise difficult to perform on unorganized low-level primitives. We now discuss the main application areas where the extracted non-local symmetry relations are used.

**Model acquisition and representation.** Symmetry relations are essentially equivalence among different parts of an object. In other words, symmetric parts represent redundant geometric regions, and thus are repeated measurements of the same underlying geometry. Such redundant observations allow denoising the input in presence of unbiased noise. More specifically, say surface patch  $S_0$  is symmetric with surface patches  $S_1, S_2, \dots$  under transforms  $T_1, T_2, \dots$ , respectively, i.e.,  $S_i \approx T_i(S_0)$  for  $i = 1, \dots$ . Then for any

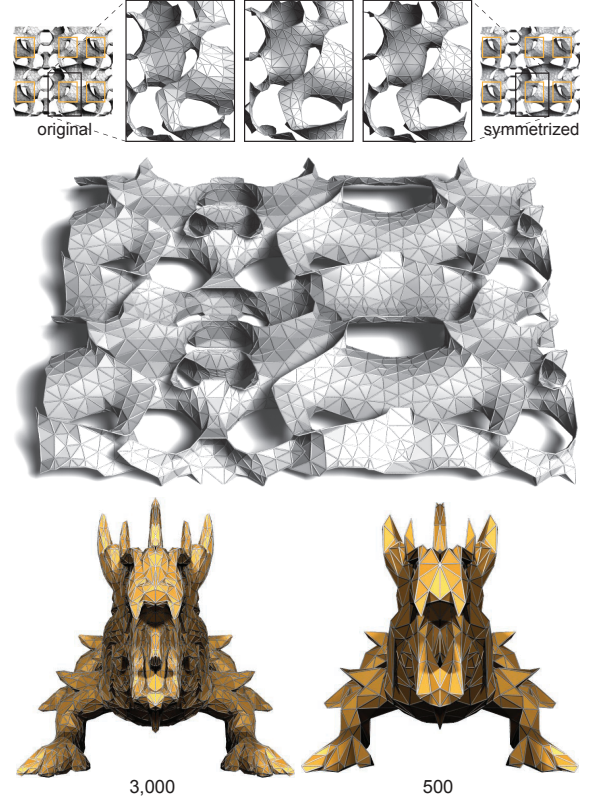


**Figure 21:** Applications of symmetry in model acquisition and representation (Images courtesy of Zhang et al. [ZSW\*10], Mitra et al. [MGP06], Thrun et al. [TW05], and Podolak et al. [PGR07] respectively.)

point  $\mathbf{p} \in S_0$  the corresponding point on the  $i$ -th patch is given by the point  $\mathbf{q}_i \in S_i$  such that  $d(\mathbf{q}_i, T_i(\mathbf{p}))$  is minimized. Note that under perfect symmetry  $\mathbf{q}_i = T_i(\mathbf{p})$ . Such correspondence information is used for non-local denoising [BCM08, LLZM10] and scan consolidation [ZSW\*10] of man-made objects (e.g., building facades, 3D ornamentation, etc.) where symmetry relations are common. Further, in regions of missing data, instead of looking for points  $\mathbf{q}_i$ , surface information is propagated using symmetry information  $T_i(\mathbf{p})$  for data completion [PMG\*05, TW05, XZT\*09] (see Figure 21).

In addition to model improvement, symmetry relations also suggest a novel acquisition framework. The main observation is that for redundant object regions essentially one has to recover the transformations  $T_i$  while using the base patch  $S_0$  for model construction. This effectively allows a smart acquisition process where base patch  $S_0$  is to be acquired at a high resolution, while for the corresponding symmetric relations only the symmetry transforms are needed, and hence significantly coarser scanning suffice. The challenge, however, is reliable symmetry detection, which can be challenging in severely undersampled data.

Detected symmetries, if organized by explicitly storing the symmetry transforms, factor out model redundancies and thus produce a *compressed representation*, e.g., if geometry is organized as a tree-structure [MGP06, SKS06] or in a hierarchy [WXL\*11] (see Figure 21). However, as discussed in Section 8, such representations are typically non-canonical.

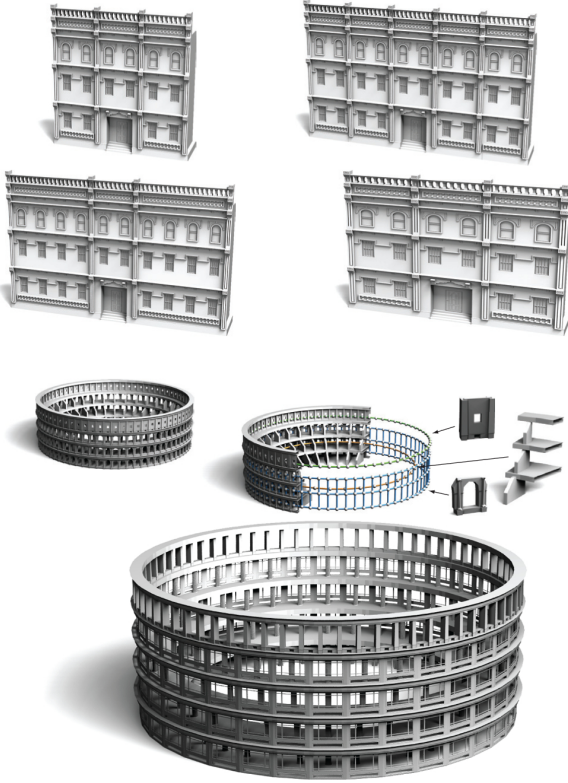


**Figure 22:** (Top panel) An input architectural design is analyzed to reveal six symmetric patches, which are then simultaneously optimized to create exactly symmetric surfaces and compatible meshes. (Bottom panel) Approximate symmetries of inputs are made exact, and the resultant meshes are simplified while respecting the symmetry relations. The number below each model indicate the corresponding vertex count [MGP07].

Intuitively such factored representations can provide a measure of information content of shapes, rather than simply measuring the model complexity in terms of vertex count.

Geometric models, however, are rarely perfectly symmetric. Deviations from perfect symmetry arise due to high-frequency noise, or due to low-frequency global deformations. Hence, researchers have proposed to *symmetrize* models, i.e., factor out deformations or deviations to increase the symmetry or redundancy of 3D models, by allowing certain model deformations [MGP07, PGR07]. Symmetrized models can also be used to produce equivalent or symmetric triangulation, which is particularly useful for symmetric geometry processing [PGR07], or to reuse of identical surface patches especially for efficient and economical manufacturing [EKS\*10, FLHCO10], etc. In the future, we expect to see more efforts on how to characterize the space of such

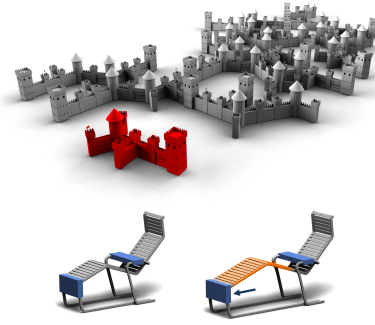




**Figure 23:** Detecting regularity in shapes specially building facades and similar man-made structures, allow us to infer plausible procedural rules [PM01, MWH\*06, MZWVG07] to recreate the models. This allows smart geometric modeling to perform global geometric edits by inserting and deleting geometric patches, or replacing the base geometric patches, while conferring to the inferred procedural rules [PMW\*08, MP08].

model variations, especially ones with low degrees of freedom [BWM\*11, OLGMI1, KLM\*12].

**Model manipulation.** We discussed how properties relating to symmetries in the object-domain manifest as clusters in the transform domain, with the fuzziness of clusters indicating the extent of approximation in the detected symmetries. Based on this observation, Mitra et al. [MGP07] present a non-local geometry editing framework wherein fuzzy clusters are tightened and the corresponding point-pair positions suitably adjusted to enhance object symmetries, both local and global. Similarly one can perform symmetric remeshing and simplification where the progressive simpler meshes still retain and respect the original object symmetries [MGP07, PGR07] (see Figure 22). Intuitively, since symmetry manifests itself better in the transform domain, a



**Figure 24:** Inverse procedural modeling from symmetry: By attaching building blocks derived from boundaries between non-symmetric area (docking sides), shape variations can be created automatically (top; original in red). Sliding dockers (bottom, orange) generalize the idea to partial patterns.

coupled spatial and transform domain editing paradigm is natural for symmetry-aware manipulations.

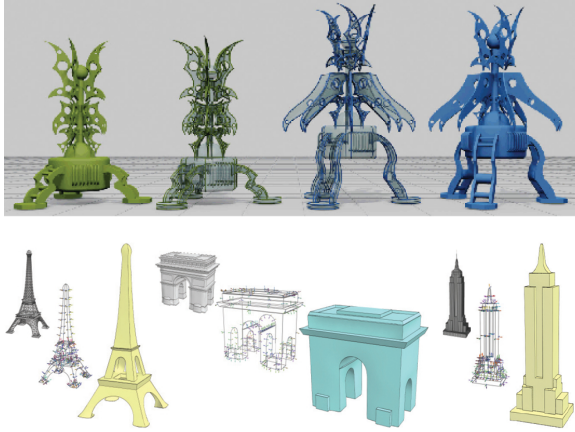
Recently, Panozzo et al. [PLPZ12] have introduced a framework to synthesize direction, line, or cross fields that respect the symmetries of the surfaces. They explore generalized reflections of surfaces to construct fields that adhere both to the coarse global symmetries and local non-symmetric features of the surfaces. These symmetric fields are used in various applications including remeshing and non-photorealistic rendering.

Symmetry of an object often relates to its characteristic form, specially for man-made objects. Gal et al. [GSMCO09] observe that such objects can be characterized using 1D feature curves or wires. Such features are then analyzed to reveal their mutual relations and symmetry properties, which are preserved using an optimization in course of subsequent manipulations. Explicitly distilling such symmetry relations as a minimalist curve network, which can act as intuitive interaction handles for manipulation of shape abstractions [MZL\*09] (see Figure 25).

**Model synthesis.** Symmetry is ubiquitous in natural objects, man-made shapes, and architectural forms. Extracted symmetries provide valuable understanding about the regularity of the structures, which often relate to form, function, and aesthetics. The extracted symmetric patches can be treated as alphabets and combined with the detected symmetry transforms in order to construct an inverse-shape grammar [SG71] for the input shapes. The constructed grammar can then be used to procedurally create variations of the input geometry of similar characteristics [PM01, MWH\*06, MZWVG07]. Such a workflow has been demonstrated in the context of modeling architectural buildings and facades [PMW\*08, MP08] (see Figure 23).

A formal framework for inferring shape grammars from

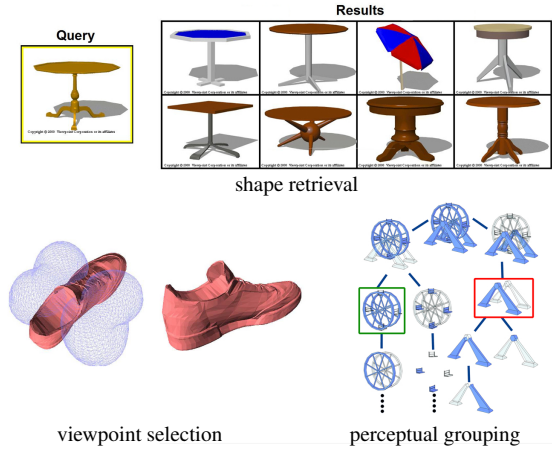




**Figure 25:** (Top panel) Preservation of analyzed symmetry relations in iWires a shape manipulation framework [GSMCO09], (bottom panel) distilling and highlighting of symmetry relations for creating abstractions of man-made shapes [MZL\*09]. (Images courtesy of Gal et al. and Mehra et al. respectively.)

symmetry information has been developed by Bokeloh et al. [BWS10]. The main challenge is how to extract building blocks that fit together, i.e., symmetric parts that can be exchanged without changing the object – this, however, does not create any shape variations. Therefore, in the context of synthesis, interesting regions comprise of the boundaries between symmetric and non-symmetric areas. Such boundaries are referred to as *docking sites* that allow different non-symmetric parts to be exchanged. This exchange of parts encapsulated in symmetry defines a rewriting system for creating shape modifications, which can be transformed into a constructive grammar towards creating a rich set of shape variations. Additionally, such a technique detects repetitive regular patterns that form a regular grid (see Figure 3) and also allows the user to change the repetition count. This has been demonstrated by Bokeloh et al. [BWKS11], where *sliding dockers* are extracted to describe partial regularity within more general shapes (see Figure 24). In general, we can consider the regularity structure of a scene as an invariant for structure preserving shape editing [BWSK12].

**Shape classification.** Kazhdan and colleagues [KFR04] use symmetry as a distinguishing feature for classification and retrieval of shapes. They relate global rotational and reflective symmetries of objects to properties of their spherical harmonic expansions (see also work of Martinet et al. [MSHS06]) and the coefficients as *symmetry descriptors*. The symmetry augmented representations were then used to compare shapes. The method presented a compact, rotation invariant shape descriptor that has been shown to be an effective descriptor on the Princeton shape database [SMKF04] (see Figure 26). The approach has been extended later to



**Figure 26:** (Top panel) Symmetry descriptors are used to query a database for shape retrieval [KFR04]. (Bottom panel) Good viewpoints are automatically selected to minimize the symmetry in the scene [PSG\*06] and a symmetry-induced hierarchical organization of the parts of a model is presented [WXL\*11]. (Images courtesy of Kazhdan et al., Podolak et al., and Wang et al. respectively.)

make use of symmetry transforms of objects [PSG\*06]. More recently, Giachetti et al. [GL12] have presented the area projection transform to detect regions in a shape which are likely to be the center of radial symmetry. This transform is similarly used to build a shape descriptor for shape retrieval applications.

Symmetry can also be utilized to find mappings across different surfaces as it captures recognizable and semantic features of many objects. Liu et al. [LKF12] use the global reflective symmetry curves as global features to find a coarse alignment between two shapes and use this information to compute surface mappings. These mappings are important for shape analysis applications.

Symmetry is also believed to be a dominating factor in shape perception and viewpoint selection [RWY95]. Podolak et al. [PSG\*06] use the detected reflective symmetries to automatically select preferred viewpoints for objects and to pose them better. Wang et al. [WXL\*11] discuss how to group objects hierarchically by a perceptually motivated scheme that combines symmetry and proximity information. The result is an intuitive though non-canonical decomposition of shapes that is used for structure-preserving editing (see Figure 26).

## 10. Future Directions

Symmetry of geometric shapes is an important tool in geometry processing. Arguably, it takes us one step closer to the ultimate goal of a computationally *understanding* three-dimensional objects. The state-of-the-art provides a wide

range of algorithms for detecting extrinsic and intrinsic symmetries in various types of data, ranging from clean meshes to point clouds from 3D scanners. Nevertheless, there are many problems yet to be resolved. Maybe, the bigger part of the challenge is even still ahead of us. Below, we give a brief overview of open problems where no good solution is known. We structure these into three areas: (i) *improving detection*, (ii) extracting *more general* notions of symmetry, and (iii) *structural models* of redundancy.

**Improving symmetry detection.** Although significant advances have been made towards symmetry detection, there are still challenging situations that cannot be handled. One aspect is data quality. For very poor data, with a large amount of noise and outliers, or substantial portions of data missing, existing approaches run into problems, in particular in the feature extraction and aggregation stages. Intrinsic approaches are particularly troubled – currently, none of the techniques handle objects with large scale topological noise, such as acquisition holes and contacts in a raw 3D scan. On the one hand, directions to address these issues could include traditional measures such as multi-resolution techniques, inpainting and feature extrapolation approaches, various data filtering and preprocessing steps, and combining information from multiple data sources when available [JTC11, LZS\*11, CML\*12, HYM13]. Further, uncertainty in the input at some point unavoidably shows up as uncertainty in the output, which could be modeled explicitly. Statistical representation of potential symmetry is a direction that could possibly contribute to addressing such challenges.

Another aspect of the detection itself is computational performance: Although known techniques are efficient enough to handle objects such as larger buildings within a few minutes, there is no technique that can process a scan of an entire city, or other instances of massive data sets. Neither the absolute computational costs nor the asymptotic scaling behavior for growing amounts of data is adequate for such scenarios. Similarly, there are no online algorithms that could for example process data that is coming in in real-time from a dynamic 3D scanner such as Microsoft Kinect.

**More general notions of symmetry.** Existing symmetry detection methods have largely explored actions of translations, reflections, rotations, scaling, as well as intrinsic isometries of a fixed manifold. However, many real-world objects, in particular organic shapes, show redundancy that cannot be characterized by these notions of symmetry, not even in an approximate sense. For example, the leaves of a plant might all roughly look alike, although no pair of them is actually related by a similarity transform. Further, there exist semantic categories such as “windows” or “cars” that share common traits that are impossible to capture by just fixing a priori set of transformation parameters. There are two key challenges: First, what is a good model of general similarity? The main conceptual pitfall is that very general

mappings between shapes are prone to overfitting; the fewer restrictions we place upon the matching of parts, the easier it becomes to relate things that, to a human, are clearly unrelated. Finding a good model for *significant* structural redundancy is probably one of the major and very difficult challenges in the field. The second problem is efficiency; given such general matching models, how can the solution be computed in a reasonable amount of time? As an example, there has been a recent attempt to generalize matchings to linear subspaces of variations [BWM\*11] but efficiency remains a challenge.

**Structural models.** Finally, there is the challenge of understanding the structure of the redundancy in the symmetry information. The known approaches use tools from group theory to structure the symmetries, but there are several restrictions: While mutually commutative transformations yield easily understood grid structures [PMW\*08], complex non-commutative structures are more difficult to handle. Once we only observe excerpts, there is a risk of overfitting, i.e., misinterpretation of the data in terms of a structure that is unnecessarily complex. Aside from group structures of the symmetries themselves, there is also the question what else the structural redundancy tells us about the shape. For example, what is the information content of a shape with partial symmetry or regularity, and what is the relative information if we know a set of shapes and observe a new one that is partially similar? In the end, this might lead to an *information theory of shapes* that unlike traditional shape compression takes self-similarity and symmetry into account.

## 11. Conclusion

Symmetry is ubiquitous in nature and in man-made objects. It captures self-similarity of objects under the action of transformations, and can be naturally thought of in terms of transformation groups. Automatic detection of symmetry depends on the type of symmetries that are of interest, i.e., whether exact or approximate, global or partial, and Euclidean or intrinsic. The choice of symmetry type for shape analysis often depends on the family of objects and target applications. We presented a framework to organize state-of-the-art symmetry detection algorithms based on their choices in the key phases of the algorithms. Finally, we discussed a wide range of applications organized into groups depending on the phase of a typical geometry processing framework they affect. Symmetry-aided processing can be applied in the shape acquisition phase, in the geometry manipulation phase, and also can be used for categorizing and organizing the captured 3D geometry of shapes.

In the future we expect research efforts to be devoted towards development of symmetry detection algorithms to handle very large volumes of data, in the form of a parallel approach or as an online algorithm. The chief difficulty in developing such algorithms come from the global nature of

symmetry that tightly couples the overall geometry making it difficult to work solely with local reasoning, without any prior knowledge or preprocessing of the data.

Symmetry amounts to redundancy in the context of information theory. This raises questions about what geometric information is fundamental, and how we can efficiently represent and store geometric shapes. Similar to many image compression procedures, it is conceivable to store a compact base representation as a symmetrized version of the shape, and then subsequently store deviations from the symmetrized shape. Intuitively this can be thought to be a level of symmetry representation, similar to frequency band representations for images. But, we still lack a canonical algorithm to quantify information content of 3D geometry.

In contrast to point sets or triangle meshes, a symmetry-aware representation provides high-level cues about the objects: symmetry properties describe the relation of object parts to themselves, thereby abstracting from the concrete shape and emphasizing on structural relations. Thus, symmetry is an important building block in gaining a high level understanding of geometry. A grand goal, however, is to infer meaningful shape semantics via computational shape analysis. Semantics can be in terms of motion, in terms of function of object parts, or degrees of freedom. A combination of geometric insights [XWY\*09, GSMCO09, MYY\*10] with data-driven approaches [FCODS08, KHS10] to capture complex patterns in the geometric structure and their relation to semantic attributes [ZCOM13], may possibly bring us closer solving this grand goal. Symmetry – the redundancy within a geometric object and the structure of such redundancy – will continue to play an important role in model acquisition, understanding, manipulation, manufacturing, and also towards efficient, economic, aesthetic, and functional object designs.

**Acknowledgement.** This research has been supported by the Marie Curie Career Integration Grant 303541, an UCL Impact grant, an Adobe research grant, and the ERC Starting Grant 257453 COSYM. The authors thank Hao (Richard) Zhang and the anonymous reviewers for their comments on various versions of this manuscript.

## References

- [ACOM12] AIGER D., COHEN-OR D., MITRA N. J.: Repetition maximization based texture rectification. *Computer Graphics Forum (EUROGRAPHICS)* (2012). 21
- [AMCO08] AIGER D., MITRA N. J., COHEN-OR D.: 4-points congruent sets for robust surface registration. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 27, 3 (2008), 85:1–85:10. 9
- [AMWW88] ALT H., MEHLHORN K., WAGENER H., WELZL E.: Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.* 3, 3 (1988), 237–256. 4, 7
- [Ata85] ATALLAH M. J.: On symmetry detection. *IEEE Trans. Comput.* 34, 7 (1985), 663–666. 4, 7
- [BAK10] BERMANIS A., AVERBUCH A., KELLER Y.: 3-d symmetry detection and analysis using the pseudo-polar fourier transform. *Int. Journal of Computer Vision* 90, 2 (2010), 166–182. 4
- [BBW\*08] BERNER A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: A graph-based approach to symmetry detection. In *Symposium on Volume and Point-Based Graphics* (2008), pp. 1–8. 4
- [BBW\*09a] BERNER A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: *Generalized intrinsic symmetry detection*. Research Report MPI-I-2009-4-005, Max-Planck-Institut für Informatik, 2009. 4, 11
- [BBW\*09b] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Computer Graphics Forum (Proc. EUROGRAPHICS)* 28, 2 (2009), 697–706. 4, 8, 11, 12, 15
- [BCBSG10] BEN-CHEN M., BUTSCHER A., SOLOMON J., GUIBAS L.: On discrete killing vector fields and patterns on surfaces. In *Proc. of Symp. of Geometry Processing* (2010). 4, 16
- [BCM08] BUADES A., COLL B., MOREL J.-M.: Nonlocal image and movie denoising. *Int. Journal of Computer Vision* 76, 2 (2008), 123–139. 17
- [BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2 (1992), 239–256. 11, 12
- [BWKS11] BOKELOH M., WAND M., KOLTUN V., SEIDEL H.-P.: Pattern-aware shape deformation using sliding dockers. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA)* 30, 6 (2011), 123:1–123:10. 4, 16, 18
- [BWM\*11] BERNER A., WAND M., MITRA N. J., MEWES D., SEIDEL H.-P.: Shape analysis with subspace symmetries. *Computer Graphics Forum (Proc. EUROGRAPHICS)* 30, 2 (2011), 277–286. 4, 17, 20
- [BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 4 (2010), 104:1–104:10. 12, 15, 18
- [BWSK12] BOKELOH M., WAND M., SEIDEL H.-P., KOLTUN V.: An algebraic model for parameterized shape editing. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 31, 4 (2012), 78:1–78:10. 18
- [Car76] CARMO M. D.: *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976. 8, 13
- [CK10] CHERTOK M., KELLER Y.: Spectral symmetry analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 7 (2010), 1227–1238. 4
- [CM92] CHEN Y., MEDIONI G.: Object modelling by registration of multiple range images. *Image Vision Comput.* 10, 3 (1992), 145–155. 11, 12
- [CML\*12] CEYLAN D., MITRA N. J., LI H., WEISE T., PAULY M.: Factored facade acquisition using symmetric line arrangements. *Computer Graphics Forum (Proc. EUROGRAPHICS)* 31, 1 (2012), 671–680. 19
- [CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. of Symp. of Geometry Processing* (2003), pp. 177–187. 8
- [EKS\*10] EIGENSATZ M., KILIAN M., SCHIFTNER A., MITRA N. J., POTTMANN H., PAULY M.: Paneling architectural freeform surfaces. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 3 (2010), 45:1–45:10. 17
- [FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 27, 3 (2008), 42:1–42:7. 21

- [FLHCO10] FU C.-W., LAI C.-F., HE Y., COHEN-OR D.: K-set tilable surfaces. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 3 (2010), 44:1–44:6. [17](#)
- [GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. on Graphics* 25, 1 (2006), 130–150. [4, 8, 9](#)
- [GG04] GELFAND N., GUIBAS L. J.: Shape segmentation using local slippage analysis. In *Proc. of Symp. of Geometry Processing* (2004), pp. 214–223. [4, 9, 11, 16](#)
- [GL12] GIACHETTI A., LOVATO C.: Radial symmetry detection and shape characterization with the multiscale area projection transform. In *Proc. of Symp. of Geometry Processing* (2012), pp. 1669–1678. [19](#)
- [GSMCO09] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: iwires: an analyze-and-edit approach to shape manipulation. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 28, 3 (2009), 1–10. [11, 18, 19, 21](#)
- [HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. In *Proc. of Symp. of Geometry Processing* (2008), pp. 1449–1457. [13](#)
- [HYM13] HALAWANI S., YANG Y., MITRA H. L. N. J.: Interactive facades: Analysis and synthesis of semi-regular facades. *Computer Graphics Forum (Proc. EUROGRAPHICS)* 32, 1 (2013). [19](#)
- [JTC11] JIANG N., TAN P., CHEONG L. F.: Multi-view repetitive structure detection. In *Proc. of the IEEE Inter. Conf. on Computer Vision* (2011), pp. 535–542. [19](#)
- [KBW\*12] KOLOJANOV J., BOKELOH M., WAND M., GUIBAS L., SLUSALLEK P., SEIDEL H.-P.: Microtiles: Extracting building blocks from correspondences. In *Proc. of Symp. of Geometry Processing* (2012). [15, 16](#)
- [KCD\*03] KAZHDAN M., CHAZELLE B., DOBKIN D., FUNKHOUSER T., RUSINKIEWICZ S.: A reflective symmetry descriptor for 3d models. *Algorithmica* 38, 1 (2003), 201–225. [4, 7, 10](#)
- [KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Symmetry descriptors and 3D shape matching. In *Proc. of Symp. of Geometry Processing* (2004), pp. 115–123. [4, 7, 11, 18, 19](#)
- [KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 3 (2010), 102:1–102:12. [21](#)
- [KLCF11] KIM V., LIPMAN Y., CHEN X., FUNKHOUSER T.: Möbius transformations for global intrinsic symmetry analysis. In *Proc. of Symp. of Geometry Processing* (2011), pp. 1689–1700. [4, 14](#)
- [Kle93] KLEIN F.: Vergleichende betrachtungen u"ber neuere geometrische forschungen. *Mathematische Annalen* 43 (1893). [2](#)
- [KLM\*12] KIM V. G., LI W., MITRA N. J., DiVERDI S., FUNKHOUSER T.: Exploring collections of 3d models using fuzzy correspondences. *ACM Transactions on Graphics* 31, 4 (2012), 54:1–54:11. [17](#)
- [LCDF10] LIPMAN Y., CHEN X., DAUBECHIES I., FUNKHOUSER T.: Symmetry factored embedding and distance. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 4 (2010), 103:1–103:12. [4, 8, 12, 14, 15](#)
- [LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 28, 3 (2009), 72:1–72:12. [13, 14](#)
- [LKF12] LIU T., KIM V. G., FUNKHOUSER T.: Finding surface correspondences using symmetry axis curves. In *Proc. of Symp. of Geometry Processing* (2012), pp. 1607–1616. [19](#)
- [LLZM10] LI G., LIU L., ZHENG H., MITRA N. J.: Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA)* 29, 6 (2010), 152:1–152:10. [17](#)
- [LTSW09] LASOWSKI R., TEVS A., SEIDEL H.-P., WAND M.: A probabilistic framework for partial intrinsic symmetries in geometric data. In *Proc. of the IEEE Inter. Conf. on Computer Vision* (2009), pp. 963–970. [4, 14](#)
- [LZS\*11] LI Y., ZHENG Q., SHARF A., COHEN-OR D., CHEN B., MITRA N. J.: 2d-3d fusion for layer decomposition of urban facades. In *Proc. of the IEEE Inter. Conf. on Computer Vision* (2011), pp. 882 – 889. [19](#)
- [MBB10] MITRA N. J., BRONSTEIN A., BRONSTEIN M.: Intrinsic regularity detection in 3d geometry. In *Proc. of the European Conf. on Computer Vision* (2010), pp. 398–410. [4, 14](#)
- [MGP06] MITRA N. J., GUIBAS L., PAULY M.: Partial and approximate symmetry detection for 3d geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 25, 3 (2006), 560–568. [4, 8, 9, 10, 12, 15, 16, 17](#)
- [MGP07] MITRA N. J., GUIBAS L. J., PAULY M.: Symmetrization. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 26, 3 (2007), 63:1–63:8. [13, 17](#)
- [MGP04] MITRA N. J., GELFAND N., POTTMANN H., GUIBAS L.: Registration of point cloud data from a geometric optimization perspective. In *Proc. of Symp. of Geometry Processing* (2004), pp. 23–31. [8, 11, 12](#)
- [MP08] MITRA N. J., PAULY M.: Symmetry for architectural design. In *Advances in Architectural Geometry* (2008), pp. 13–16. [16, 18](#)
- [MSHS06] MARTINET A., SOLER C., HOLZSCHUCH N., SIL-LION F.: Accurate detection of symmetries in 3d shapes. *ACM Trans. on Graphics* 25, 2 (2006), 439 – 464. [4, 7, 19](#)
- [MWH\*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 25, 3 (2006), 614–623. [18](#)
- [MY\*10] MITRA N. J., YANG Y.-L., YAN D.-M., LI W., AGRAWALA M.: Illustrating how mechanical assemblies work. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 4 (2010), 58:1–58:12. [21](#)
- [MZL\*09] MEHRA R., ZHOU Q., LONG J., SHEFFER A., GOOCH A., MITRA N. J.: Abstraction of man-made shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 28, 5 (2009), 137:1–137:10. [11, 18, 19](#)
- [MZWVG07] MÜLLER P., ZENG G., WONKA P., VAN GOOL L.: Image-based procedural modeling of facades. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 26, 3 (2007). [18](#)
- [NLCK05] NADLER B., LAFON S., COIFMAN R. R., KEVREKIDIS I. G.: Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems* (2005), pp. 955–962. [12](#)
- [OLGM11] OVSJANIKOV M., LI W., GUIBAS L., MITRA N. J.: Exploration of continuous variability in collections of 3d shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 30, 4 (2011), 33:1–33:10. [17](#)
- [OSG08] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. In *Proc. of Symp. of Geometry Processing* (2008), pp. 1341–1348. [4, 8, 14](#)
- [PGR07] PODOLAK J., GOLOVINSKIY A., RUSINKIEWICZ S.: Symmetry-enhanced remeshing of surfaces. In *Proc. of Symp. of Geometry Processing* (2007), pp. 235–242. [16, 17](#)



- [PLPZ12] PANOZZO D., LIPMAN Y., PUPPO E., ZORIN D.: Fields on symmetric surfaces. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 31, 4 (July 2012), 111:1–111:12. [17](#)
- [PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proceedings of SIGGRAPH* (2001), pp. 301–308. [18](#)
- [PMG\*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L.: Example-based 3d scan completion. In *Proc. of Symp. of Geometry Processing* (2005), pp. 23–32. [17](#)
- [PMW\*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 27, 3 (2008), 43:1–43:11. [4, 8, 9, 10, 13, 16, 18, 20](#)
- [PSG\*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 25, 3 (2006), 549–559. [4, 6, 8, 10, 11, 14, 19](#)
- [PWY\*07] POTTMANN H., WALLNER J., YANG Y.-L., LAI Y.-K., HU S.-M.: Principal curvatures from the integral invariant viewpoint. *Comput. Aided Geom. Des.* 24, 8–9 (2007), 428–442. [8](#)
- [RBBK07] RAVIV D., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Symmetries of non-rigid shapes. In *Proc. of the IEEE Inter. Conf. on Computer Vision* (2007), pp. 1–7. [4, 8, 14, 15](#)
- [RBBK09] RAVIV D., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Full and partial symmetries of non-rigid shapes. *Int. Journal of Computer Vision* 89, 1 (2009), 18–39. [4](#)
- [RBS\*10] RAVIV D., BRONSTEIN M. M., SAPIRO G., BRONSTEIN A. M., KIMMEL R.: Diffusion symmetries of non-rigid shapes. In *Proc. 3DPVT* (2010). [4](#)
- [Rot94] ROTMAN J.: *An Introduction to the Theory of Groups*. 4th edition, Springer, 1994. [2](#)
- [Rus04] RUSINKIEWICZ S.: Estimating curvatures and their derivatives on triangle meshes". In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium* (2004), pp. 486–493. [8](#)
- [Rus07a] RUSTAMOV R. M.: Augmented symmetry transforms. In *Proc. of Int. Conf. on Shape Modeling and Applications* (2007), pp. 13–20. [11](#)
- [Rus07b] RUSTAMOV R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proc. of Symp. of Geometry Processing* (2007), pp. 225–233. [8, 14](#)
- [RWY95] REISFELD D., WOLFSON H., YESHURUN Y.: Context free attentional operators: the generalized symmetry transform. *Int. Journal of Computer Vision* 14 (1995), 119–130. [19](#)
- [SG71] STINY G., GIPS J.: Shape grammars and the generative specification of painting and sculpture. In *Proceedings of IFIP Congress* (1971). [18](#)
- [Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum* 27, 6 (2008), 1539–1556. [5](#)
- [SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Proc. of Symp. of Geometry Processing* (2006), pp. 111–119. [4, 15, 17](#)
- [SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Proc. of Shape Modeling International* (2004), pp. 167–178. [19](#)
- [SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. of Symp. of Geometry Processing* (2009), pp. 1383–1392. [8](#)
- [SS87] SCHWARTZ J. T., SHARIR M.: Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *International Journal of Robotics Res* 6, 2 (1987), 29–44. [8](#)
- [SS97] SUN C., SHERRAH J.: 3d symmetry detection using the extended gaussian image. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 2 (1997), 164–168. [4, 7](#)
- [SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *Visual Computer* 24, 4 (2008), 249–259. [13](#)
- [Tau95] TAUBIN G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. of the IEEE Inter. Conf. on Computer Vision* (1995), pp. 902–. [8](#)
- [TMS09] TZIMIROPOULOS G., MITIANOUDIS N., STATHAKI T.: A unifying approach to moment-based shape orientation and symmetry classification. *Trans. Img. Proc.* 18, 1 (2009), 125–139. [7](#)
- [TW05] THRUN S., WEGBREIT B.: Shape from symmetry. In *Proc. of the IEEE Inter. Conf. on Computer Vision* (2005), pp. 1824–1831. [4, 16, 17](#)
- [vKZHC011] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707. [5](#)
- [Wey52] WEYL H.: *Symmetry*. Princeton University Press, 1952. [2](#)
- [WR97] WOLFSON H. J., RIGOUTSOS I.: Geometric hashing: An overview. *IEEE Comput. Sci. Eng.* 4, 4 (1997), 10–21. [8, 9](#)
- [WXL\*11] WANG Y., XU K., LI J., ZHANG H., SHAMIR A., LIU L., CHENG Z., XIONG Y.: Symmetry hierarchy of man-made objects. *Computer Graphics Forum (Proc. EUROGRAPHICS)* 30, 2 (2011), 287–296. [15, 17, 19](#)
- [xu2] [14](#)
- [XWY\*09] XU W., WANG J., YIN K., ZHOU K., VAN DE PANNE M., CHEN F., GUO B.: Joint-aware manipulation of deformable models. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 28, 3 (2009), 35:1–35:9. [21](#)
- [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. on Graphics (Proc. SIGGRAPH)* 28, 5 (2009), 138:1–138:10. [4, 13, 14, 15, 17](#)
- [ZCOM13] ZHENG Y., COHEN-OR D., MITRA N. J.: sfarr: Functional substructures for part compatibility. *Computer Graphics Forum (Proc. EUROGRAPHICS)* 32, 1 (2013). [21](#)
- [ZPA95] ZABRODSKY H., PELEG S., AVNIR D.: Symmetry as a continuous feature. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 12 (1995), 1154–1166. [4, 5, 10](#)
- [ZSW\*10] ZHENG Q., SHARF A., WAN G., LI Y., MITRA N. J., COHEN-OR D., CHEN B.: Non-local scan consolidation for 3d urban scenes. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 29, 3 (2010), 94:1–94:9. [16, 17](#)