A Planar-Reflective Symmetry Transform for 3D Shapes

Joshua Podolak   Philip Shilane   Aleksey Golovinskiy   Szymon Rusinkiewicz   Thomas Funkhouser

Princeton University
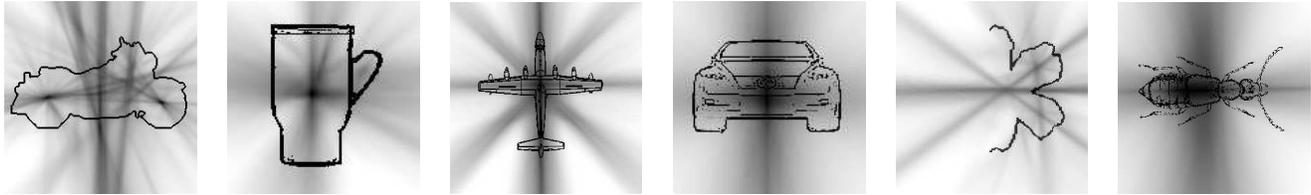
Figure 1: The Planar Reflective Symmetry Transform captures the degree of symmetry of arbitrary shapes with respect to reflection through all planes in space. Although symmetry measures are computed for planes (lines in 2D), for this visualization, points are colored by the symmetry measure of the plane with the largest symmetry passing through them, with darker lines representing greater symmetries.

Abstract

Symmetry is an important cue for many applications, including object alignment, recognition, and segmentation. In this paper, we describe a planar reflective symmetry transform (PRST) that captures a continuous measure of the reflectional symmetry of a shape with respect to all possible planes. This transform combines and extends previous work that has focused on global symmetries with respect to the center of mass in 3D meshes and local symmetries with respect to points in 2D images. We provide an efficient Monte Carlo sampling algorithm for computing the transform for surfaces and show that it is stable under common transformations. We also provide an iterative refinement algorithm to find local maxima of the transform precisely. We use the transform to define two new geometric properties, center of symmetry and principal symmetry axes, and show that they are useful for aligning objects in a canonical coordinate system. Finally, we demonstrate that the symmetry transform is useful for several applications in computer graphics, including shape matching, segmentation of meshes into parts, and automatic viewpoint selection.

Keywords: symmetry, shape analysis, registration, matching, segmentation, viewpoint selection

1 Introduction

Symmetry is an important feature of almost all shapes: nearly every man-made object contains at least one plane of perfect symmetry, and most natural objects exhibit near-perfect symmetries (look around your office and count the symmetric objects).

While symmetry is known to provide visual cues for human vision [Ferguson 2000], it has seen only limited use as a feature for applications in computer graphics and geometric processing. In recent years, symmetry information has been used to detect local features in 2D images [Reisfeld et al. 1995], guide reconstruction of 2D curves and range scans with missing data [Zabrodsky

et al. 1993; Thrun and Wegbreit 2005], rotate shapes into a canonical coordinate frame [Kazhdan et al. 2003a], match the shapes of meshes [Kazhdan et al. 2004], and recognize instantiation of parts in assembled scenes [Martinet et al. 2005]. However, most of this work has focused on local symmetry with respect to points and global symmetry with respect to planes through the center of mass.

In this paper, we describe a *planar reflective symmetry transform* (PRST), a transform from the space of points to the space of planes that provides a continuous measure of the reflective symmetry of an object with respect to all planes through its bounding volume. This transform represents both perfect and imperfect symmetries; it captures global symmetries of an object; and, it highlights local symmetries of salient parts.

As an example, consider the visualizations of the PRSTs for several 2D outlines (black lines) shown in Figure 1. Note that the point of maximal symmetry (darkest point) usually corresponds to the conceptual center of the object, and the main ridges in the PRST frequently correspond to the main axes, even for partial objects (e.g., the flower). Also notice that symmetries in parts of objects (e.g., the wheels of the motorcycle) appear as local maxima in the transform, while small extra asymmetric features (e.g., the handle of the mug) do not affect the global maxima. These properties are advantageous for several applications, including selection of a canonical coordinate system, matching of 3D shapes, segmentation into parts, and optimal viewpoint selection (see results in Section 5).

The contributions of our work are six-fold. First, we define the planar reflective symmetry transform (Section 3). Second, we present a new algorithm based on Monte Carlo integration for computing a discrete version of the PRST (Section 4.3). Third, we provide an iterative refinement algorithm for finding local maxima in the PRST with arbitrary precision (Section 4.4). Fourth, we define the new shape properties *center of symmetry* and *principal symmetry axes*, and demonstrate their utility for alignment of 3D meshes (Section 5.1). Fifth, we investigate using the PRST as a representation of shape in matching and retrieval experiments (Section 5.2). Finally, we explore the use of local symmetries in segmentation and viewpoint selection (Sections 5.3 and 5.4 respectively).

2 Previous Work

Perfect Symmetries: Traditional approaches to symmetry detection work with discrete symmetries—perfect symmetries under rotation, reflection, or translation. For instance, efficient al-

gorithms have been described for finding whole-object symmetries using substring matching [Atallah 1985; Wolter et al. 1985; Zhang and Huebner 2002], using an octree representation [Minovic et al. 1993], the extended Gaussian image [Sun and Sherrah 1997] and the singular value decomposition of the points of the model [Shah and Sorensen 2005]. Further methods are available for describing local symmetries with a respect to a point — e.g., the medial axis [Blum 1967]. However, since these methods consider only perfect symmetries, they are unstable with added noise or missing data and fail to recognize the potentially important cues of imperfect symmetries.

As a recent example, Thrun and Wegbreit [2005] detect perfect symmetries in scanned models by explicitly searching ever-growing sets of points while maintaining a list of possible rotational and reflectional perfect symmetries. In this way, scans of models composed of symmetrical parts may be completed by extending the measured symmetries to the entire model.

Also recently, Martinet et al. [2005] have introduced a method based on generalized moments to detect perfect symmetry in 3D shapes accurately. Their approach combines the property that even order moments contain the same symmetries as the model with an efficient method for calculating moment coefficients using a spherical harmonic decomposition. While their method is efficient and can detect perfect symmetries of a segmented model, they do not work with imperfect symmetries, nor do they investigate the range of applications addressed in this paper.

Imperfect Symmetries: In the last decade, methods have been provided for measuring imperfect symmetries. For example, Zabrodsky et al. defined the *symmetry distance* of a shape with respect to a transformation as the distance from the given shape to the closest shape that is perfectly symmetric with respect to that transformation [Zabrodsky et al. 1995; Zabrodsky et al. 1993]. They provide an algorithm to find the symmetry distance for a set of connected points for any given reflective or rotational transformation, and they use it for completing the outline of partially-occluded 2D contours, for locating faces in an image, and for determining the orientation of a 3D shape. However, their method considers symmetry with respect to only one point or plane at a time, and thus it is not as general nor as descriptive as the methods proposed in this paper.

Symmetry descriptors: Kazhdan et al. used the same continuous measure of imperfect symmetries to define a *shape descriptor* that represents the symmetries of an object with respect to all planes and rotations through its center of mass [Kazhdan et al. 2003a; Kazhdan et al. 2004]. They describe an efficient algorithm for computing the descriptor from a 3D voxel representation, and show that planar symmetries can be used for alignment and classification of 3D meshes. We build upon this work to consider symmetries with respect to all planes through an object's bounding volume, an extension that enables more robust alignment and matching, as well as new applications based on local symmetries.

Symmetry Transforms: More general symmetry transforms defined for all points in space have been used for a decade in computer vision, mainly for robust detection of radially symmetric features in natural images prior to segmentation. For example, Reisfeld et al. defined a Generalized Symmetry Transform for local point symmetries [Reisfeld et al. 1995], and related approaches have been described in [Bigun 1997; Di Gesù et al. 1997; Loy and Zelinsky 2002; Choi and Chien 2004]. These methods have been shown to be effective for finding features in noisy images (e.g., eyes on a face [Reisfeld and Yeshurun 1992]), discriminating textures [Bonneh et al. 1994; Chetverikov 1995], and segmenting images based upon local symmetries [Kelly and Levine 1995]. However, they are designed mainly for detecting local point symmetries in unseg-
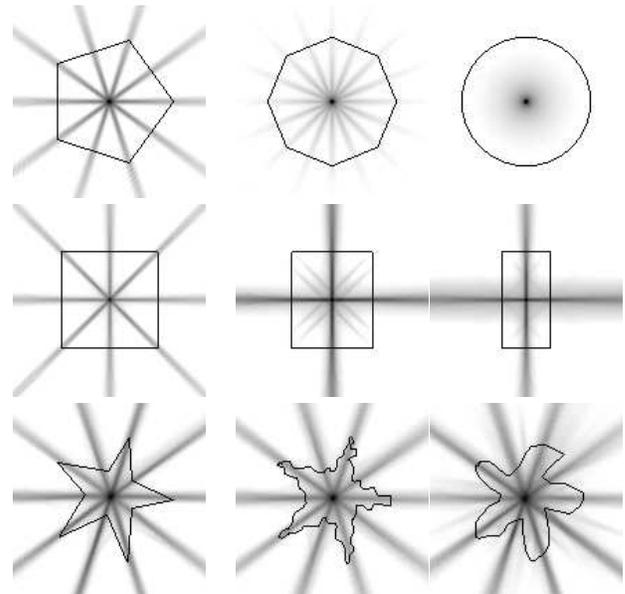

Figure 2: Visualization of the PRST for several simple 2D shapes.

mented 2D images, as opposed to global planar symmetries in 3D meshes, and thus they do not address the mesh processing problems and computer graphics applications targeted in this paper.

3 Planar Reflective Symmetry Transform

The *Planar Reflective Symmetry Transform* (PRST) is a mapping from a scalar-valued function f defined over a d-dimensional space of points to a scalar-valued function $PRST(f, \gamma)$ defined over the d-dimensional space of planes, such that the scalar value associated with every plane reflection γ is a measure of f's symmetry with respect to that plane.

Following previous work on continuous symmetry analysis [Zabrodsky et al. 1995; Kazhdan et al. 2003a; Kazhdan et al. 2004], we define the symmetry distance, $SD(f, \gamma)$ of f with respect to a plane reflection γ as the L_2 distance between f and the nearest function that is invariant to that reflection:

$$SD(f, \gamma) = \min_{g \mid \gamma(g) = g} \|f - g\|.$$

Since the symmetry distance lies between 0 and $\|f\|$ and provides a measure of the "anti-symmetry" of the shape with respect to γ, we complement it and divide by the magnitude of f, to produce a normalized symmetry measure for our PRST, such that

$$PRST^2(f, \gamma) = 1 - \frac{SD^2(f, \gamma)}{\|f\|^2}$$

This definition was chosen so that $PRST(f, \gamma)$ is 1 if f is perfectly symmetric with respect to γ, 0 if f is perfectly anti-symmetric with respect to γ, and an intermediate value for partial symmetries. It is also important for computing the PRST efficiently, since the square of the PRST for any plane reflection reduces to a dot product under this formulation, as will be seen in the following section (for functions on a grid, a dot product is the sum of the product of each pair of corresponding elements).

To give an intuitive sense for the information provided by the PRST, visualizations for several simple 2D shapes are shown in Figures 1 and 2. In these images, symmetry is measured for every plane (line

in 2D), and the darkness of every point represents the maximum of PRST values over all planes passing through the point (darker values represent larger symmetries). While these images do not show the PRST directly, since the PRST is defined on the space of planes, not points, they are easier to understand than visualizations shown in plane coordinates, and they give a good sense for where planes of high symmetry can be found. Thus, we display visualizations of this type throughout the paper unless otherwise noted.

These images demonstrate two important properties of the PRST. First, we see that the dominant points and planes of symmetry match our human intuition of the "center" and "major axes" of the object. For example, trivially, the planes with maximal symmetry through a circle intersect at the center, and the major and minor axes of a rectangle appear as global and local maxima in the PRST, respectively. In general, the local maxima of the PRST coincide with the axes of the large (nearly) symmetric parts of an object (Figure 1).

Second, since the PRST for every plane takes into account the shape of the entire object, it is not sensitive to noise and varies continuously with deformations. These provable properties can be seen empirically in Figure 2 — i.e., the PRST is stable when a regular polygon deforms into a circle, when a square deforms into a rectangle, and when noise is added to the boundary of a starfish. This stability is in stark contrast to the medial axis transform, which finds local point symmetries but is sensitive to small boundary perturbations, shooting off a new branch for every small bump on the boundary (Figure 3).



PRST   MAT
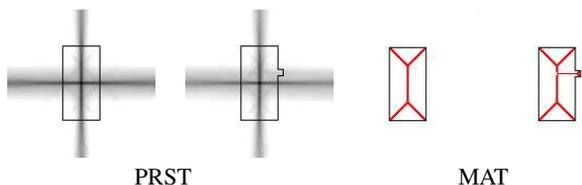
Figure 3: The PRST and the medial axis transform (MAT) for a rectangle and a rectangle with a bump. The bump leaves the PRST largely unaffected (left), but dramatically changes the MAT (right).

While these visualizations are shown for contours in 2D, reflective symmetry transforms can be defined in any dimension, for any symmetry group, for any symmetry measure, and for any shape representation, including point sets, surface meshes, and volumetric functions. For the sake of simplicity and generality, we focus our initial discussion in the following sections on planar symmetries for 3D volumetric functions, however other shape representations can trivially be converted to this representation.

4   Computation

Computing the planar symmetry transform is challenging, as an infinite number of planes can pass through an object, and a measure of symmetry for an entire object must be determined for all of them. In the following subsections, we provide background information that can be used to gain insight into computational solutions, and then we describe three algorithmic strategies. The first two provide methods for computing a discrete version of the PRST for volumetric functions and surface meshes, respectively, while the third provides a continuous method for finding local maxima of the PRST precisely.

4.1   Background

Given a function f in a d-dimensional space, we aim to develop an algorithm that will compute

$$PRST^2(f,\gamma) = 1 - \frac{SD^2(f,\gamma)}{\|f\|^2}$$

for every plane reflection γ, where $SD(f,\gamma)$ represents the L_2 distance between f and the closest function that is symmetric with respect to γ.

In previous work, [Kazhdan et al. 2003a] has observed that the nearest symmetric function to f is simply the average of f and $\gamma(f)$:

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

Combining the two equations, we get

$$PRST^2(f,\gamma) = 1 - \frac{SD^2(f,\gamma)}{\|f\|^2} = 1 - \frac{\|f-\gamma(f)\|^2}{4\|f\|^2} =$$

$$1 - \frac{\|f\|^2 - 2f\cdot\gamma(f) + \|\gamma(f)\|^2}{4\|f\|^2}.$$

If f is normalized, then $\|\gamma(f)\| = \|f\| = 1$ (since norms are preserved by reflection), and we obtain

$$PRST^2(f,\gamma) = 1 - \frac{1 - 2f\cdot\gamma(f) + 1}{4} = \frac{1 + f\cdot\gamma(f)}{2}. \quad (1)$$

Therefore, the calculation of the symmetry measure for a single plane reduces to the calculation of a dot product between f and its reflection:

$$D(f,\gamma) = f\cdot\gamma(f). \quad (2)$$

Intuitively, this means that the $PRST^2(f,\gamma)$ for a single plane reflection γ is related to how well f correlates with $\gamma(f)$, and it can be computed with an integration of $f\cdot\gamma(f)$ over the bounding volume of f.

In order to apply the above definition to surfaces, it is necessary to convert them to volumetric functions. While we could simply rasterize the surfaces into a (binary) occupancy grid, this would result in sensitivity to noise and small features. Instead, we use the Gaussian Euclidean Distance Transform (GEDT), as was previously proposed by [Kazhdan et al. 2004]. For a model M and width σ, the GEDT at an arbitrary point in space x is defined as:

$$f(x,M,\sigma) = e^{-dist^2(x,M)/\sigma^2},$$

where $dist(x,M)$ represents the distance of the point x to the nearest point on M. This allows surfaces to be slightly misaligned (by the Gaussian width σ) under reflection, allowing us to capture imperfect symmetries of the surface. We choose σ for the Gaussian according to the maximal frequency of the grid in order to avoid aliasing and to provide a gradually decreasing "penalty" for imperfect symmetries. The net result is a 3D function that is exactly one at the surface of the object and gradually drops off to zero at both interior and exterior points. Similar methods can be used for other shape representations.

4.2   Discrete Computation for Volumetric Functions

With this background, we can propose several algorithms for computing a discrete version of the PRST for a function f represented on a regular $n \times n \times n$ grid. These algorithms are efficient for characterizing all the planar symmetries of a densely sampled volume

(a) Sampling orientations   (b) Sampling translations
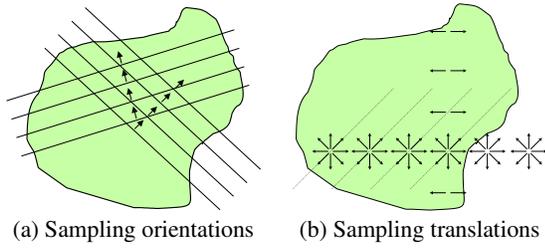
Figure 4: Computing the PRST for many planes at once by: (a) sampling orientations and convolving over plane translations, or (b) sampling positions and convolving over plane rotations.

(e.g., medical imaging data). However, as we will show in the following section, more efficient algorithms are possible for sparse functions representing surfaces and point clouds.

As a first step, we could naively apply Equation 1 to evaluate $PRST^2(f, \gamma)$ for every possible plane reflection γ separately. Since there are $O(n^3)$ possible planes through a $n \times n \times n$ grid, and the evaluation of a dot product over the grid for each plane requires $O(n^3)$, the total complexity of this brute force algorithm is $O(n^6)$, which is prohibitively expensive for most applications. However, since the PRST values for planes with the same orientation require dot products of functions successively shifted at regular intervals with respect to one another (Figure 4a), we can compute them all at the same time with a single convolution. Since the convolution for a single direction takes $O(n^3 \log n)$, and there are $O(n^2)$ possible directions through the grid, the total running time of this algorithm is $O(n^5 \log n)$.

Equivalently, we can consider convolutions over rotations at a discrete set of points (Figure 4b). In this case, we use the frequency domain algorithm described in [Kazhdan et al. 2003a] to compute the PRST for all planes through $O(n)$ points (Figure 4b). Since each invocation of Kazhdan et al.'s algorithm takes $(n^4 \log n)$, the total running time is again $O(n^5 \log n)$. A multiresolution approximation is possible in $O(n^4 \log n)$.

We have investigated all three of these approaches. In our implementation, we discretize the space of planes to match the resolution of the grid (finer sampling of the planes yields no additional information about the band-limited PRST). When working in 2D, we use a uniform parameterization of the set of lines by their angles $\theta \in [0, \pi]$ and distance from the origin $r \in [-r_{max}, r_{max}]$. Note the slightly unusual choice of angles on a semicircle and both positive and negative radii, which avoids a singularity at the origin. Similarly, we parameterize planes in 3D by the spherical coordinates of their normals $\theta \in [0, \pi/2], \phi \in [0, 2\pi]$ and distance from the origin $r \in [-r_{max}, r_{max}]$. (Note that the "buckets" of planes are not of uniform size, shrinking towards the poles as $\sin \theta$.)

We find that these convolution algorithms take 40 seconds on average for grids with $64 \times 64 \times 64$ voxels on a 3GHz processor.

4.3 Discrete Computation for Surfaces

While the algorithms discussed so far are equally efficient for all functions, rasterized surfaces and point sets naturally lead to sparsity over the volume. In this section, we describe a Monte Carlo algorithm for computing the PRST that takes advantage of this sparsity to increase efficiency.

Our discussion of the algorithm begins with the brute-force approach presented in Section 4.2:
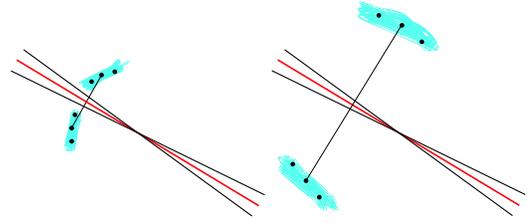

Figure 5: The efficient Monte Carlo algorithm selects a pair of points and votes for the plane between them. The vote must be weighted, accounting for the fact that as a point is father away from the plane of reflection, the chance of finding a reflection point is increased (the size of the blue area is larger as the points are farther apart).

> **for each** plane γ:
>   **for each** point x:
> $x' \leftarrow \gamma(x)$
> $PRST^2(f, \gamma) \mathrel{+}= f(x) \cdot f(x')$

We observe that for sparse functions this is inefficient, since it performs useless computation whenever either $f(x)$ or $f(x')$ is near zero. Instead, we interchange the order of computations and perform *importance sampling* in a Monte Carlo framework:

> **for** sampled points x:
>   **for** sampled points x':
> $\gamma \leftarrow$ reflection plane(x, x')
> $PRST^2(f, \gamma) \mathrel{+}= w(x, x', \gamma) \cdot f(x) \cdot f(x')$

Intuitively, this algorithm repeatedly picks a pair of points and "votes" for the plane between them. The sampling of x and x' is performed according to the energy in the function f, allowing us to focus effort on computations that will contribute to the final answer. For a typical 3D surface, non-negligible values appear in only $O(n^2)$ voxels, and thus this algorithm requires only $O(n^4)$ operations to compute the entire PRST.

Weighting: In order for the above algorithm to compute the PRST correctly, it is necessary to weight the contribution of each "vote" appropriately. This is the role of the function $w(x, x', \gamma)$, which consists of two terms. The first term accounts for the importance sampling that we perform, and is simply the reciprocal of the probability of having selected x and x':

$$w_{samp}(x, x', \gamma) = \frac{1}{f(x) \cdot f(x')}.$$

The second term represents a *change-of-variables*, accounting for the two different ways we have of sampling the space of planes: as a pair of points (x, x') and with our discretized bins over (r, θ, ϕ). While part of this change-of-variables term is intuitive (accounting for the $\sin \theta$ decrease in bin size), another part accounts for the fact that bins of planes will receive more votes if x and x' are far apart than if they are nearby. For example, both parts of Figure 5 consider a single bin of planes. However, for a fixed x, it is clear that more points x' will vote for that bin if the points are further apart, so we should weight the contribution of such pairs lower than votes by nearby points.

To derive the change-of-variables weight, we simply compute the determinant of the Jacobian of the transformation between the parameterization of the planes of reflection and the reflected points themselves. If we let

$$\hat{n} = \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix}$$

be the normal of the plane of reflection, then we can write

$$
\begin{aligned}
d &= \|x - x'\| \\
&= 2(r - \hat{n} \cdot x) \\
x' &= x + d\hat{n} \\
&= x + 2r\hat{n} - 2(\hat{n} \cdot x)\hat{n} \\
J &= \left(\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \left(\frac{\partial x'}{\partial r}\right) & \left(\frac{\partial x'}{\partial \theta}\right) & \left(\frac{\partial x'}{\partial \phi}\right) \\ \downarrow & \downarrow & \downarrow \end{array} \right)
\end{aligned}
$$
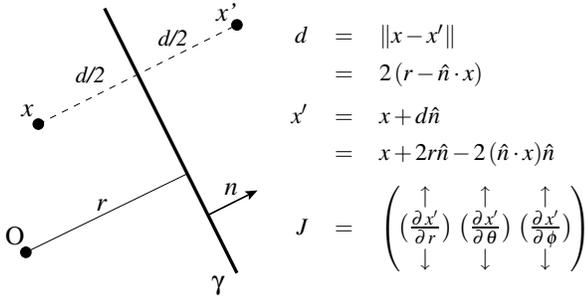
and solve for the determinant:

$$
w_{change\text{-}of\text{-}variables} = |J| = 2d^2 \sin\theta.
$$

Therefore, we have

$$
\begin{aligned}
w(x, x', \gamma) &= w_{samp} \cdot w_{change\text{-}of\text{-}variables}, \\
&= \frac{1}{f(x)\,f(x')\,2d^2\sin\theta}.
\end{aligned}
$$

So, overall, our Monte Carlo estimator is:

$$
D(f, \gamma) = \frac{1}{N_{samp}} \sum_{i=1}^{N_{samp}} \frac{1}{2d^2 \sin\theta}
$$

Computation time: By exploiting sparsity in the volume, the Monte Carlo algorithm is able to compute the PRST of 3D surfaces efficiently. As with all randomized algorithms, noise in the final approximation decreases with additional samples, but as shown in Figure 6, the algorithm converges quite quickly. For example, for the 64^3 grid resolution used throughout this paper, computing the PRST to 1% noise takes an average of 8 seconds on a 3 GHz processor, corresponding to two million sampled point pairs. These results are typical - there is little variation in computation time, except for very large models (for which the rasterization time can begin to dominate).
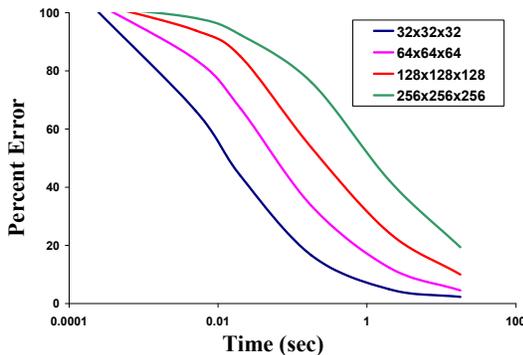

Figure 6: Comparison of error in the Monte Carlo approximation to the PRST, as a function of time. For typical grid sizes, such as the 64^3 used throughout this paper, computing the PRST takes only a few seconds.

4.4 Continuous Refinement for Local Maxima

While the PRST taken as a whole characterizes all of an object's symmetries, its *local maxima* form an important and intuitive subset. They may correspond to the principal symmetries of the whole object, weaker or partial local symmetries, or perfect symmetries of parts. Due to their intuitive nature, they are important for several applications, including all those described in Section 5. We now discuss an approach that builds upon the algorithm presented in the previous section to find local symmetry maxima precisely: we extract candidates for local maxima from the discrete PRST, then refine their locations using an iterative local optimization algorithm. This algorithm is able to find local maxima of the PRST with arbitrary precision.

Given the full 3D symmetry transform, tabulated at a moderate resolution, we first look for cells with a higher symmetry value than all their immediate neighbors. This yields a large number of candidates, so we apply a number of thresholds to extract only the strongest symmetries. First, we apply a threshold on the strength of the symmetry at that cell. While we could use a single fixed threshold, we have observed that portions of the model away from the center naturally have lower symmetry values (since there is less of the model that could potentially map onto itself under those reflections), so it is more natural to use a *lower* threshold near the edges of the model than near the center. In particular, we 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 from the center of mass. On top of the symmetry threshold, we also discard shallow maxima, which are potentially subject to noise: we impose a threshold on the discrete Laplacian (sum of second partial derivatives) of the PRST. The thresholds are set automatically to $^{1}/_{10}$ of the values at the strongest local symmetry.

Once we have a list of candidate local maxima, we refine them to find the planes of symmetry with high precision. This approach, of finding maxima of a function by first tabulating it then locally refining candidate maxima, is commonly used for numerical maximization in general, and also resembles the local optimization performed by Martinet et al. [2005]. Our refinement method is inspired by the Iterative Closest Points algorithm [Besl and McKay 1992], commonly used to perform pairwise alignment of meshes, but solves for a plane of reflection rather than a rigid-body transformation.

Our "Iterative Symmetric Points" or ISP algorithm begins by randomly sampling points from the mesh (we typically use around 10,000 points per iteration), then reflecting them across the candidate plane. We match each reflected point to the closest point on the mesh, then solve for the three parameters of the reflection plane that minimizes the sum of distances (weighted to account for the Gaussian Euclidean Distance Transform) between corresponding points (note that the minimal sum of weighted distances provides a maximum for Equation 2 when f is the GEDT of the surface). The process is iterated until it converges to a local maximum of the PRST. Figure 7, left, shows an iteration of ISP, with source points in red, the candidate plane in gray, and reflected points in green. The support of the final maximum is shown at right.

If the iteration causes the reflection plane to leave its cell (in the discrete PRST), the candidate is determined to be an unstable local maximum and discarded. Of course, this should not happen if the function f is sufficiently bandlimited by the GEDT. However, we have found this check is necessary since different point sampling strategies are used by the discrete and iterative algorithms.

In our experiments, this two-stage process of first tabulating the PRST then refining candidate local maxima has proven both robust and efficient. The local refinement converges in a few seconds for each plane, and we typically find 10–20 strong local maxima of
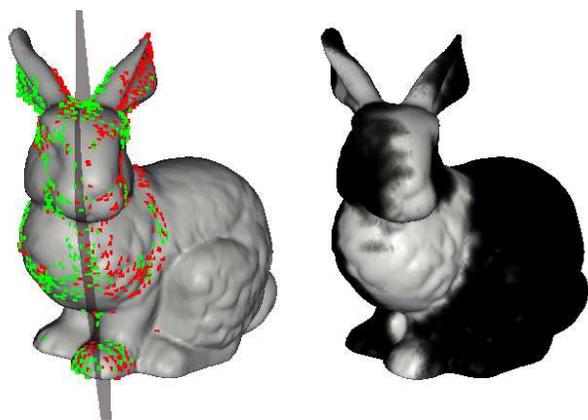
Figure 7: At left, we show an iteration of *ISP*. We select random points (red), reflect them through the candidate plane of symmetry (gray), and find closest points on the surface (green). We then update the plane of reflection to optimize the sum of Gaussian distances between corresponding point pairs (samples with low weight have been culled in this visualization). At right, we show the support of the final local symmetry maximum, as indicated by the gray level.

symmetry for models of moderate complexity. Figure 8 shows the four strongest local maxima for a bull model, together with the surface support of each plane reflection (white regions of the surface reflect onto each other across the chosen plane). Note that we find planes capturing the global symmetries of the bull (1), as well as separate local maxima capturing symmetries of the neck (2), body (3), and head (4).

5 Applications

The PRST is a general-purpose transform with potential applications in computer vision, medical imaging, and a variety of other fields. In this section, we investigate four applications in computer graphics.

5.1 Alignment

Alignment of objects into a canonical coordinate frame is an important preprocessing step for a variety of tasks, including visualization, studying the variation of models across different classes, composition of scenes, and indexing of 3D model databases. To perform this alignment, a global point of reference must be selected for the origin and a set of axes must be chosen to determine the orientation.

Typically, alignments of this type are computed with principal component analysis (PCA): the center of mass is chosen as the origin, and the principal axes are used to determine the orientation [Duda et al. 2001]. However, it is well known that PCA does not always produce compatible alignments for objects in the same class [Kazhdan et al. 2003b], and it certainly does not produce alignments similar to what a human would select. Consider, for example, the mugs shown in Figures 9 and 10. Most humans would suggest that the central axis of these mugs runs straight up and down through the middle of the cup, and the center is somewhere along this axis. However, the center of mass and principal axes (shown in green in Figure 9) are biased towards the handle to different degrees depending on the size and shape of the handle, producing alignments

(1) (2)

(3) (4)
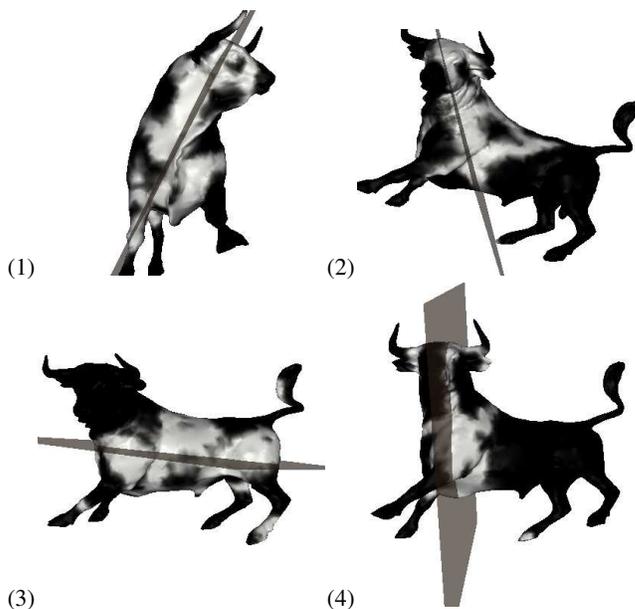
Figure 8: In this visualization, the triangles of the bull are colored to show how symmetric they are with respect to the plane of symmetry displayed, with black meaning no support of the plane reflection. Planes representing the four strongest local maxima of the PRST are shown here. Note how points supporting reflections across planes (2), (3), and (4) tend to cluster into regions corresponding to the neck, body, and head, respectively.

inconsistent with other mugs and undesirable for most applications (e.g., placement in a scene). Similarly, since the center of mass is shifted off the central axis of the cup, the symmetry descriptor of Kazhdan et al. [2003a] will not detect its perfect symmetries and will produce poor alignments.
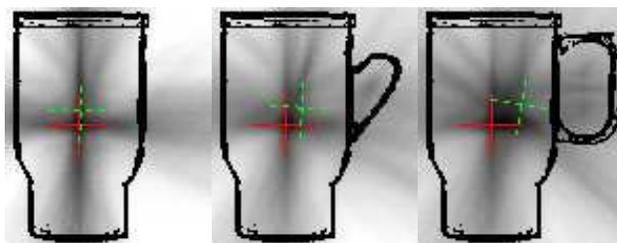
Figure 9: A line drawing of a mug with and without handles. The the center of mass and PCA axes are drawn in dotted green — note that they move depending on the presence of handles. A visualization of the PRST is overlaid on the drawings, and the center of symmetry and principal symmetry axes are shown in solid red — they remain stable under perturbation of the shape.

In this section, we investigate the use of the PRST to produce better alignments. Specifically, we introduce two new concepts, the *principal symmetry axes* (PSA) and the *center of symmetry* (COS), as robust global alignment features of a model. Intuitively, the principal symmetry axes are the normals of the orthogonal set of planes with maximal symmetry, and the center of symmetry is the intersection of those three planes. Specifically, given a PRST we select the first principal symmetry axis by finding the plane with maximal symmetry. We then select the second axis by searching for the plane with maximal symmetry among those perpendicular to the first, and

Center of mass and PCA

Center of symmetry and principal symmetry axes

Center of mass and PCA

Center of symmetry and principal symmetry axes
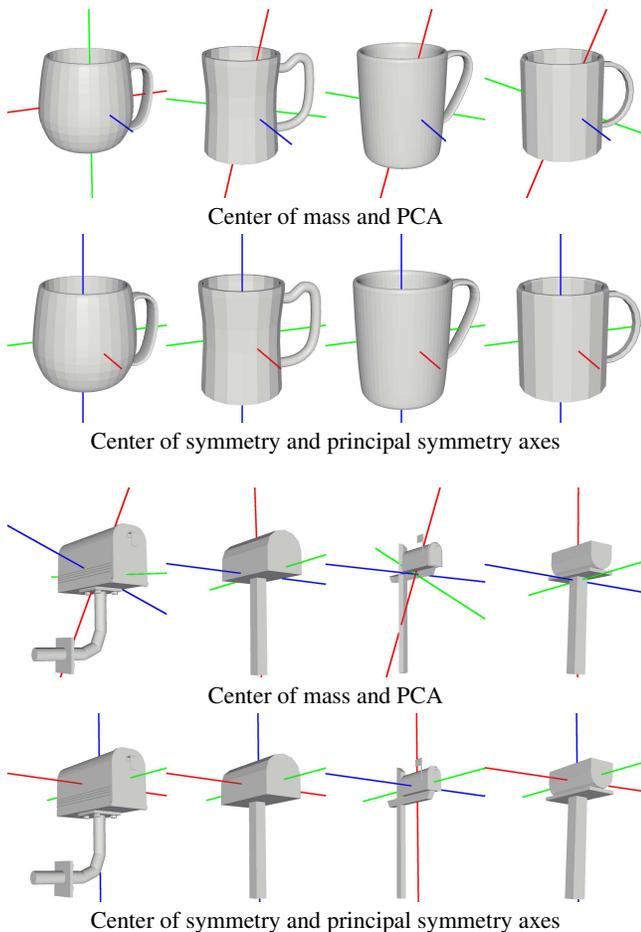
Figure 10: Alignment for translation and rotation based on centers of symmetry and principal symmetry axes, as compared to center of mass and PCA. In each case, the red, green, and blue lines represent the first, second, and third principal axes, and their intersection is the computed center.

finally we choose the third axis in the same way, searching only the planes perpendicular to both the first and second selections.

We find that this simple method produces coordinate frames that are both robust and semantically meaningful for most objects. For example, for the mugs shown in Figures 9 and 10, the center of symmetry and principal symmetry axes appear right in the middle of the cylindrical cup. Similarly, for the mailboxes shown in Figure 10, the center of symmetry and principal symmetry axes consistently reside in the middle of the box — unlike the center of mass and principal axes, they are not affected by the shapes of the stands. In general, the principal symmetry axes and center of symmetry are determined by an object's large parts with significant symmetries, and thus they closely match our intuition of an object's canonical coordinate frame.

In order to test whether the PRST is robust, even for partial surfaces, we experimented with alignments of synthetically generated range scans. This experiment is motivated by an object recognition application in which (partial-object) scans are acquired and registered to (whole-object) meshes stored in a database, with the hope of automatically recognizing which object was scanned [Shan et al. 2004]. For this application, it is useful to align the partial scan to the complete object automatically.

(Center of Symmetry)   (Center of Mass)
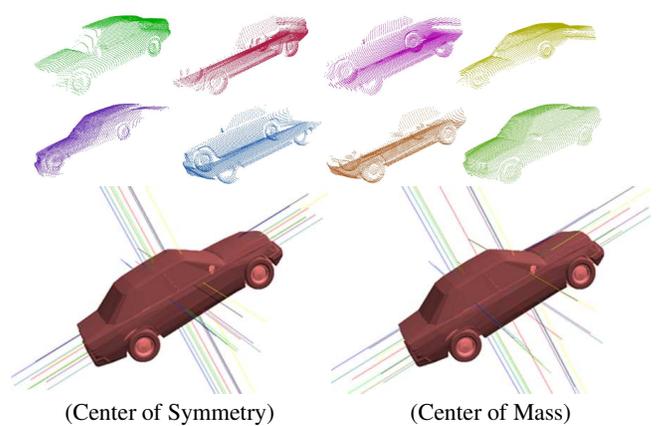
Figure 11: To evaluate the stability offered by symmetry-based alignment in a scan recognition application, we computed eight virtual scans of 907 models (top two rows of images), and for each computed coordinate frames using our symmetry-based approach and PCA. Note how the centers of symmetry computed from the partial scans (shown as cross-hairs) cluster near the center of the whole car better than do the centers of mass.

(a)   (b)
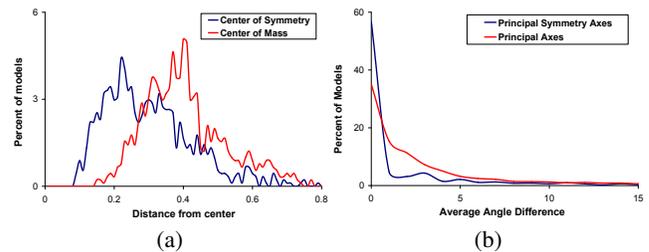
Figure 12: Histograms of translational (a) and rotational (b) misalignment in our virtual-scan experiment. In blue we show alignment based on center of symmetry and PSA, with center of mass and PCA in green. Larger values near the left of each graph indicate better matching performance.

For the experiment, we used a ray tracer to generate eight synthetic range scans of approximately 10,000 points (Figure 11, top) for the 907 meshes provided as part of the Princeton Shape Benchmark test set [Shilane et al. 2004]. For each mesh, the virtual scanner was in turn placed at each of the eight corners of a cube surrounding the model, always pointing toward the center of the mesh bounding box. The view distance was twice the length of the bounding box diagonal, and the field of view was 0.4 radians (Figure 11). For each scan, we voxelized the point cloud, computed the PRST, and extracted the principal symmetry axes and center of symmetry (Figure 11, bottom left). Then, we evaluate how well these coordinate frames match the frames computed for the complete meshes, and compare to the accuracy of the frames computed using the principal axes and center of mass (Figure 11, bottom right).

Figure 12 shows the results of this experiment. In the first plot, we see histograms of the translational misalignment between the partial range scans and the whole objects when aligned with the center of symmetry (blue curve) and center of mass (green curve). In general, the partial scans were sufficient to recover the major symmetries of the object correctly, leading to lower average errors for center-of-symmetry alignment as compared to center-of-mass. Of the 907 models tested, the center of symmetry for a scan was closer to that of the entire model 90% of the time. On average, they were closer by a factor of 1.5, with better results occurring when a particular viewpoint caused significant portions of the model to be missing.
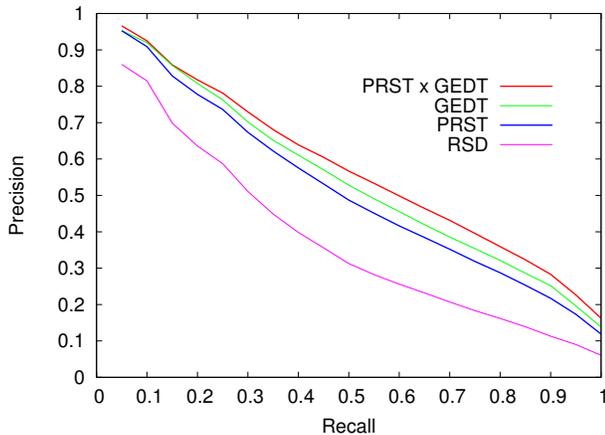
Figure 13: Plot of precision versus recall achieved with four shape matching methods: PRSD (magenta), PRST (thick blue), GEDT (green), and the combination of PRST and GEDT (thick red). Note that the PRST captures information complementary to other shape matching methods, hence can be used to augment their retrieval performance.

Figure 12b shows histograms of the rotational misalignment of the partial scans with respect to the full object when aligned with principal symmetry axes (PSA) and with the principal axes (PCA). Note the large peak near zero in the blue curve, indicating that PSA recovered the rotation for most range scans to within a few degrees of that computed for the entire object. In contrast, PCA provided a larger spread of misalignment angles. Moreover, even though less than half of the surface was available in any scan, we found that the coordinate frame chosen with PSA was within 5 degrees of a human chosen set of axes for the whole object in 70% of the scans, as opposed to only 50% for PCA.

5.2 Matching

For many applications, it is important to be able to classify 3D models or retrieve them from a database based on their geometric properties. The goal is to find a representation of shape (a *shape descriptor*) that can be computed robustly, matched efficiently, and used to discriminate different classes of objects effectively. In this section, we investigate using the PRST as such a shape descriptor.

Our efforts are motivated by the observation that symmetry properties are often consistent within a class of objects. For example, although chairs may vary in their size, whether or not they have arms, etc., their reflective planar symmetries are almost always the same (perfect left-right symmetry, a weaker global symmetry between the back and seat, local symmetries through the seat and back, etc.). This is true for many other object classes as well, including airplanes, tables, people, etc. Perhaps it is possible to classify 3D meshes automatically by comparing their computed symmetries to those of meshes in a supervised training set.

In previous work, Kazhdan et al. investigated this approach using their reflective symmetry descriptor. They employed the maximum difference between the symmetry measures of any two corresponding planes through the center of mass as a dissimilarity measure for a pair of 3D meshes. We extend that previous work by considering symmetries with respect to all planes through an object's bounding volume. In our matching method, we measure the dissimilarity between a pair of aligned meshes as the L_2 distance between their discrete PRSTs (Section 4.3), weighting the differences between
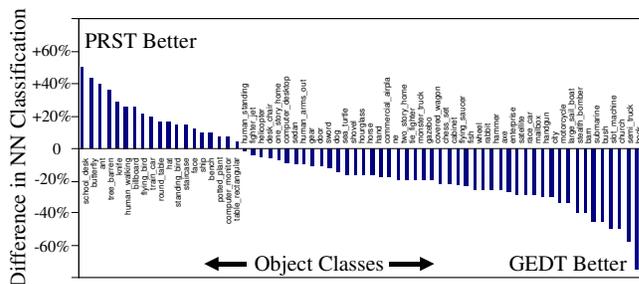
Figure 14: The PRST provides better retrieval performance for some classes (top), while the GEDT is better for others (bottom). Combined, they produce better retrieval performance than either alone.

corresponding bins of the PRST by $\sqrt{\sin \theta}$ (where θ is the polar angle of the plane represented by the bin) to account for different bin sizes. This measure produces a large distance when there are planes for which one object is (nearly) symmetric, while the other is not.

In order to evaluate the PRST as a shape descriptor for shape-based retrieval and classification applications, we ran a set of "leave-one-out" experiments with the Princeton Shape Benchmark test set [Shilane et al. 2004], a database of 907 polygonal models partitioned into 92 classes commonly used for shape matching evaluations. In order to focus our study on shape representation rather than alignment, we manually registered all models into a common coordinate frame before matching every model against all the others. The L_2 distances between PRSTs were used to produce a ranked retrieval list for each "query" model, and then statistics were computed to evaluate how often models within the same class appear at the front of the computed retrieval lists.

Figure 13 shows average precision-recall plots comparing our retrieval performance with that of Kazhdan et al.'s planar reflective symmetry descriptor (PRSD) and the Gaussian Euclidean Distance Transform (GEDT), which is currently used in at least one shape based search engine [Funkhouser et al. 2003]. The horizontal axis of this plot represents increasing recall values (fraction of the query's class retrieved), while the vertical axis represents retrieval precision (fraction of the retrieved models that are in the same class as the query). Higher curves represent better performance.

We find that the precision achieved when matching with the PRST (thick blue curve) is higher than with the PRSD (magenta curve) for every recall value. This confirms the expectation that the extra information provided by the PRST (symmetries for off-center planes) adds precision for shape matching. Of course, it is more expensive to store ($32,768$ floats versus $1,024$ floats) and to compare (0.1 ms versus 0.004 ms), but the extra cost seems worth the improved performance for most applications.

We also find that both the PRSD and PRST provide less matching precision than the GEDT (green curve) on average. We believe that this is because many object classes within the Princeton Shape Benchmark have the same symmetries (e.g., almost all manmade objects have perfect left-right symmetry). Even though the symmetries may be consistent within a class, they do not always help discriminate between classes. However, we observe that the PRST provides better matching results for some types of objects (i.e., ones with distinctive symmetries), while the GEDT provides better results for others (Figure 14). So, we find that combining the two shape descriptors (by simply multiplying the L_2 distances computed separately for the two descriptors) provides better retrieval performance than either alone (the thick red curve in Figure 13).

The nearest neighbor classification rate and discounted cumulative gain scores for the combined method were 69.2% and 68.6%, respectively, which represent good retrieval performance for this data set [Shilane et al. 2004]. This leads us to conclude that the PRST, while perhaps not the best shape representation for retrieval of this type of data on its own, can provide useful information for shape-based matching and can be used to discriminate classes of objects that are difficult to distinguish with other methods.

5.3 Segmentation

Although the PRST naturally represents the symmetries of an entire object, it also implicitly captures the symmetries of its parts. We propose to use this information for segmentation: we decompose a mesh such that the faces with each segment have the same distinct symmetries. This criterion for automatic decomposition into parts is quite different from previous methods (e.g., [Chazelle et al. 1995; Li et al. 2001; Mangan and Whitaker 1999] in that it incorporates local shape information for many different parts of the mesh simultaneously.

Our segmentation algorithm follows recent work that has used k-means clustering as a primitive operation. However, instead of clustering based on a simplification [Garland et al. 2001] or based on geodesic and angular distance between points [Katz and Tal 2003], we cluster based on support for local maxima in the PRST. To do this, we find the significant local maxima of the PRST (Section 4.4) and compute, for each face and for every symmetry plane, the degree to which the face contributes to the symmetry with respect to that plane — i.e., how well does the face map onto the surface after reflection across the plane (a visualization of this measure is shown in Figure 8). If there are m local maxima in the PRST, then every point has m values representing its support for symmetry with respect to each of the m planes. We treat these m values as a feature vector and cluster faces according to their proximity in the m-dimensional feature space. Intuitively, this method clusters faces that support the same, distinct set of planar symmetries.

Our segmentation algorithm proceeds hierarchically, in a manner similar to the method of Katz and Tal [2003]. For each split, we perform k-means clustering (with $k = 2$) to establish a rough segmentation, and then take the two largest connected components and find the exact boundary between them by computing a weighted min-cut along the edges of the mesh. The discrete PRST is recomputed and its local maxima are refined after every split. Segmentation is terminated at a user-supplied depth, or when the only planes of local maxima reflect either more than 90% or less than 10% of the surface onto itself.

The result of this process is a segmentation tree, with the property that lower levels in the tree capture increasingly local symmetries, hence allowing strong symmetries of even small parts to influence the segmentation. Figure 15 shows some examples of the segmentation produced by our method. Note that for the Teapot the strongest planes of symmetry pass through the body of the pot. So, the handle, spout, and top are removed precisely because they are *not* symmetric with respect to those planes — i.e., the body of the pot is removed from the smaller parts, rather than vice-versa. For the Octopus and Skeletal Hand models, local symmetries of parts are important for obtaining the segmentation shown. Finally, a weakness of our scheme can be seen in the segmentation of the legs of the Dinopet and Bull. Because we use a min-cut to smooth our initial guess, the final segmentation will seek a shorter cut, and thus avoid the upper sections of the thigh. Integrating symmetry information into the min-cut algorithm is a topic for future work.
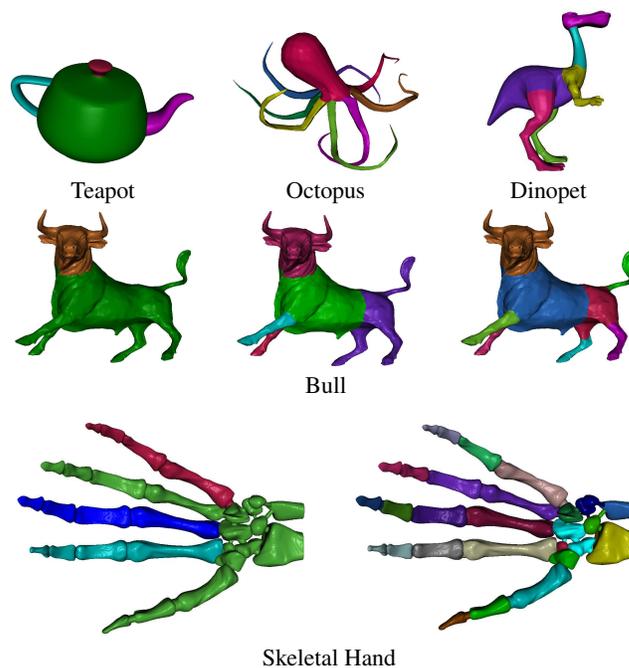

Figure 15: These images show segmentation of a range of models. For the bull we show segmentation into 2, 4, and 8 segments. The skeletal hand is shown segmented into 4 and 18 parts.

5.4 Viewpoint Selection

3D models may look considerably different when viewed from different directions, thus computing good viewpoints for 3D models has always been important for applications such as rapidly viewing a large number of models, generation of icons, selection of viewpoints for Image Based Rendering, and robot motion. The optimal "canonical view" of a model may differ depending on the purpose, and there have been a number of methods suggested to find such viewpoints for various applications. Kamada et al. [1988] seek to minimize the number of degenerate faces in the image. Both Vázquez et al. [2001] and Lee et al. [2005] try to find quality viewpoints by maximizing the (interesting) information content for a view. Abbasi et al. [2000] and Lee et al. [2004] find optimal viewpoints for recognition based on image contours, pruned by various imaging constraints. Finally, Blanz et al. [1999] performed a user study to determine factors that influence the canonical views used to display 3D models. They report that users prefer off-axis views to front or side axis views.

We introduce a method to choose good viewpoints automatically by minimizing the symmetry seen. The intuition behind our approach is that symmetry in an object presents redundant information to the user and is therefore to be avoided. Our method begins with the primary symmetry of the object and then uses the local maxima extracted from the PRST to minimize the amount of symmetry in the direction of the viewer. More specifically, for each plane appearing as a local maximum in the PRST, the preferred viewing direction is along the normal to the plane. We compute the viewpoint score for a view direction v as $S(v) = \sum_{u \in W} |v \cdot u| \cdot M(u)$ where $u \in W$ is a plane of local symmetry and $M(u)$ is the symmetry score for that plane.

With this relatively simple scoring function it is enough to do an exhaustive search to find the optimal viewpoint, although a gradient decent method such as introduced by [Lee et al. 2005] may be used to accelerate the computation.

| Score	Best	Worst
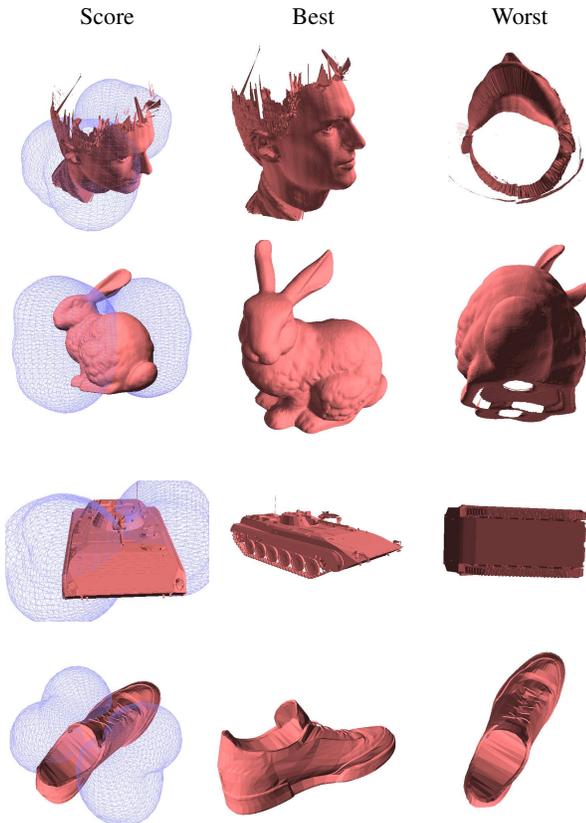
Figure 16: At left, we show the viewpoint score for each model as a spherical function. The visualization is obtained by scaling unit vectors on the sphere in proportion to the quality of the viewpoint from that direction. The images at center show the best viewpoint selected by our algorithm. The images at right show the worst viewpoint selected.

We show the results of our approach in the middle column of Figure 16 — each object's local symmetries repel the viewpoint, such that the final selected view is off of the major axis of the objects. Note that even though we only show the best viewpoint, it is possible to extend this implementation to produce multiple "interesting" viewpoints for the user's selection (left column in Figure 16) and/or viewpoints to be avoided (right column).

6 Discussion

The investigation of the PRST presented in this paper is a first step. Our implementation has several limitations, and there are many avenues for future research.

First, we have investigated only the transform that maps a 3D object to its planar reflective symmetries. While this type of symmetry is perhaps the most prevalent in real-world objects, and thus makes a good starting point, it is possible to consider other types of symmetry in future work. For example, one might define a rotational symmetry transform, in which a measure of symmetry is computed for every possible rotation around every possible axis (5D), or a point reflectional symmetry around every point (3D). We can further investigate the mapping of the reflective symmetry transform back from the space of planes to the space of points, by storing with each point a function of the symmetry distances for the planes that

pass through it. This was the motivation for our visualization in Figures 1 and 2. We hypothesize that the average, maximum, or some similar function of symmetry distances for all planes through each point may produce a 3D function worthy of future investigation.

Second, our transform measures symmetries of an entire object. However, for objects containing multiple symmetric parts a useful investigation would be to understand how symmetries can be detected at multiple scales, corresponding to different sized regions of local support [Manmatha and Sawhney 1997]. While we have shown an automatic segmentation algorithm that extract symmetries of large parts of the model hierarchically, interesting local symmetries may "be drowned" and not found during this process. A multi-resolution scheme, where the PRST is computed on local sections of the model would be interesting to investigate.

Finally, since the PRST is a $3D \rightarrow 3D$ mapping, we raise the question of whether the transform is invertible. While at first glance the transform is composed of operations that cause irrecoverable loss of information, precluding inversion, we believe that there are strong constraints provided by, for example, the requirement that the model be bounded (i.e. that the function is zero at the boundary) that make inversion possible. Using these constraints we can already show that the transform is invertible in the 1D and 2D cases, and we believe we can extend this to 3D. The inversion is currently sensitive to noise however, so further research is necessary to determine if additional constraints or stronger variational relations might make our method practical. We hypothesize that the ability to invert the transform will lead to applications in a variety of domains, with the ability to not only analyze but also synthesize symmetries.

7 Summary

In summary, we have defined the *planar reflective symmetry transform*, which measures the symmetry of an object with respect to all planes through its bounding volume. We have described an efficient Monte Carlo algorithm for computing the transform for surface meshes, shown that it is stable under small perturbations, and investigated its utility for several geometric processing applications. In particular, we propose that the *center of symmetry* and *principal symmetry axes* are useful for aligning 3D objects in a common coordinate frame. We also show that the reflective symmetry transform can be used for registering 3D range scans into a common coordinate system, matching 3D polygonal models of the same class, segmenting 3D models into parts, and finding good viewpoints for visualization of meshes. In future work, we hope to investigate other variants of the symmetry transform and their applications for geometric processing.

Acknowledgments

References

ABBASI, S., AND MOKHTARIAN, F. 2000. Automatic view selection in multi-view object recognition. In *Proc. ICPR*, vol. 1, 1013.

ATALLAH, M. 1985. On symmetry detection. *IEEE Trans. on Computers 34*, 663–666.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-D shapes. *IEEE Trans. PAMI 14*, 2, 239–256.

BIGUN, J. 1997. Pattern recognition in images by symmetries and coordinate transformations. *Computer Vision and Image Understanding 68*, 3, 290–307.

BLANZ, V., TARR, M., BUELTHOFF, H., AND VETTER, T. 1999. What object attributes determine canonical views. *Perception 28*.

BLUM, H. 1967. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, MIT Press, W. Whaten-Dunn, Ed., 362–380.

BONNEH, Y., REISFELD, D., AND YESHURUN, Y. 1994. Quantification of local symmetry: application to texture discrimination. *Spatial Vision 8*, 4, 515–530.

CHAZELLE, B., DOBKIN, D. P., SHOURABOURA, N., AND TAL, A. 1995. Strategies for polyhedral surface decomposition: an experimental study. In *SCG '95: Proceedings of the eleventh annual symposium on Computational geometry*, ACM Press, New York, NY, USA, 297–305.

CHETVERIKOV, D. 1995. Pattern orientation and texture symmetry. *Computer Analysis of Images and Patterns 970*, 222–229.

CHOI, I., AND CHIEN, S. 2004. A generalized symmetry transform with selective attention capability for specific corner angles. *IEEE Signal Processing Letters 11*, 2 (Feb.), 255–257.

DI GESÙ, V., VALENTI, C., AND STRINATI, L. 1997. Local operators to detect regions of interest. *Pattern Recognition Letters 18*, 11–13 (Nov.), 1088–1081.

DUDA, R., HART, P., AND STORK, D. 2001. *Pattern Classification, Second Edition*. John Wiley & Sons, New York.

FERGUSON, R. W. 2000. Modeling orientation effects in symmetry detection: The role of visual structure. In *Proc. Conf. Cognitive Science Society*.

FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., AND JACOBS, D. 2003. A search engine for 3D models. *ACM Trans. Graph. 22*, 1, 83–105.

GARLAND, M., WILLMOTT, A., AND HECKBERT, P. S. 2001. Hierarchical face clustering on polygonal surfaces. In *SI3D '01: Proceedings of the 2001 Symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 49–58.

KAMADA, T., AND KAWAI, S. 1988. A simple method for computing general position in displaying three-dimensional objects. *Comput. Vision Graph. Image Process. 41*, 1, 43–56.

KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *Proceedings of ACM SIGGRAPH 22*, 3, 954–961.

KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. A reflective symmetry descriptor for 3D models. *Algorithmica 38*, 1 (Oct.).

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3D shape matching. In *Proc. Symposium on Geometry Processing*.

KELLY, M. F., AND LEVINE, M. D. 1995. Annular symmetry operators: A method for locating and describing objects. In *Proc.*

ICCV, 1016–1021.

LEE, J., MOGHADDAM, B., PFISTER, H., AND MACHIRAJU, R. 2004. Finding optimal views for 3d face shape modeling. In *FGR*, IEEE Computer Society, 31–36.

LEE, C. H., VARSHNEY, A., AND JACOBS, D. W. 2005. Mesh saliency. *Proceedings of ACM SIGGRAPH 24*, 3, 659–666.

LI, X., TOON, T., TAN, T., AND HUANG, Z. 2001. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 Symposium on Interactive 3D graphics*, 35–42.

LOY, G., AND ZELINSKY, A. 2002. A fast radial symmetry transform for detecting points of interest. In *Proc. ECCV*, 358–368.

MANGAN, A. P., AND WHITAKER, R. T. 1999. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics 5*, 4, 308–321.

MANMATHA, R., AND SAWHNEY, H. 1997. Finding symmetry in intensity images. Tech. Rep. UM-CS-1997-007, University of Massachusetts, Jan.

MARTINET, A., SOLER, C., HOLZSCHUCH, N., AND SILLION, F. 2005. Accurately detecting symmetries of 3D shapes. Tech. Rep. RR-5692, INRIA, September.

MINOVIC, P., ISHIKAWA, S., AND KATO, K. 1993. Symmetry identification of a 3D object represented by octree. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 5 (May), 507–514.

REISFELD, D., AND YESHURUN, Y. 1992. Robust detection of facial features by generalized symmetry. In *Proc. ICPR*, 117.

REISFELD, D., WOLFSON, H., AND YESHURUN, Y. 1995. Context-free attentional operators: The generalized symmetry transform. *IJCV 14*, 2, 119–130.

SHAH, M. I., AND SORENSEN, D. C. 2005. A symmetry preserving singular value decomposition. *SIAM Journal of Matrix Analysis and it's Application* (October).

SHAN, Y., MATEI, B., SAWHNEY, H. S., KUMAR, R., HUBER, D., AND HEBERT, M. 2004. Linear model hashing and batch ransac for rapid and accurate object recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*.

SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The Princeton Shape Benchmark. In *Proc. Shape Modeling International*.

SUN, C., AND SHERRAH, J. 1997. 3D symmetry detection using the extended Gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence 2*, 2 (February), 164–168.

THRUN, S., AND WEGBREIT, B. 2005. Shape from symmetry. In *Proceedings of the International Conference on Computer Vision (ICCV)*, IEEE, Bejing, China.

VÁZQUEZ, P.-P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2001. Viewpoint selection using viewpoint entropy. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*, Aka GmbH, 273–280.

WOLTER, J. D., WOO, T. C., AND VOLZ, R. A. 1985. Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer 1*, 37–48.

ZABRODSKY, H., PELEG, S., AND AVNIR, D. 1993. Completion of occluded shapes using symmetry. In *Proc. CVPR*, 678–679.

ZABRODSKY, H., PELEG, S., AND AVNIR, D. 1995. Symmetry as a continuous feature. *Trans. PAMI 17*, 12, 1154–1166.

ZHANG, J., AND HUEBNER, K. 2002. Using symmetry as a feature in panoramic images for mobile robot applications. In *Proc. Robotik*, vol. 1679 of *VDI-Berichte*, 263–268.