Symmetrization

Niloy J. Mitra
TU Vienna

Leonidas J. Guibas
Stanford University

Mark Pauly
ETH Zurich

Figure 1: Different 3D objects symmetrized with our algorithm. The transparent shapes indicate the original models. The example in the center shows a fully automatic correspondence computation that can be formulated as a symmetrization of the two poses of a scanned human.

Abstract

We present a symmetrization algorithm for geometric objects. Our algorithm enhances approximate symmetries of a model while minimally altering its shape. Symmetrizing deformations are formulated as an optimization process that couples the spatial domain with a transformation configuration space, where symmetries can be expressed more naturally and compactly as parametrized point-pair mappings. We derive closed-form solution for the optimal symmetry transformations, given a set of corresponding sample pairs. The resulting optimal displacement vectors are used to drive a constrained deformation model that pulls the shape towards symmetry. We show how our algorithm successfully symmetrizes both the geometry and the discretization of complex 2D and 3D shapes and discuss various applications of such symmetrizing deformations.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling.

Keywords: symmetry, shape analysis, shape optimization, correspondence, matching

1 Introduction

Symmetry is a central concept in many natural and man-made objects and plays a crucial role in visual perception, design, engineering, and art. Symmetries are often approximate or partial, due to variations in growth processes, imperfections in manufacturing or acquisition procedures, or shortcomings in manual design applications. Several recent efforts in shape analysis have focused on detecting symmetries in two- and three-dimensional shapes [Podolak et al. 2006], [Martinet et al. 2006], [Mitra et al. 2006]. Numerous applications have successfully utilized this type of information, e.g., for model reduction [Mitra et al. 2006], scan completion [Thrun and Wegbreit 2005], segmentation [Simari et al. 2006], shape matching, and viewpoint selection [Podolak et al. 2006].

We propose a novel algorithm for *symmetrization* that strives to enhance symmetries in a given model, while minimally altering its shape. The challenge in symmetrization lies in the fact that simply copying parts of the model and applying the desired symmetry transform does not lead to satisfactory results in general. For example, the global symmetries of the shapes shown in Figure 1 cannot be realized with such a naïve approach. The key insight of our method is that coupling the spatial domain with a symmetry transformation space, where symmetries can be expressed more naturally, leads to a simple objective function that encodes the desired symmetry-enhancing deformations. The resulting optimization minimally deforms the shape while maximizing its symmetry score using existing shape deformation tools (e.g. [Igarashi et al. 2005] or [Botsch et al. 2006]).

Symmetrization has numerous applications. In shape design, imperfect input data often does not exhibit the symmetries intended by the designer or those present in some ideal model. Examples include shapes modeled using sketch-based interfaces or data stemming from 3D model acquisition. Manually correcting symmetry deficits in the data is time-consuming, cumbersome, and virtually impossible for large-scale models. This problem is even more chal-

lenging for articulated or deformable models, where the full shape symmetry is only apparent in certain poses. Similarly, reverse engineering applications can benefit from a symmetrized model to recover the semantic structure of a given shape. Machine recognition of parts can be improved by first applying symmetrization to repetitive elements in an acquired shape. This process can also lead to reduced noise and better compression performance. Certain classes of non-rigid shapes, e.g. articulated models, can be symmetrized to improve the performance of shape matching algorithms, which can be beneficial for pose-independent model retrieval. More generally, a symmetrized object exhibits a structure that is easier for us to perceive, to classify, and to understand.

Contributions. Our contributions are two-fold: Given an explicit pairing of points, we derive closed-form solutions for the optimal reflective and/or rigid transformation *and* the corresponding minimal displacements to exactly achieve this symmetry. Based on this theoretical framework we introduce a practical algorithm for symmetrization. Our method alternates between symmetry detection and shape deformation, using an optimization that couples the spatial domain with the symmetry transform space to successively enhance symmetries in a given model. This process is fully automatic and requires no user interaction beyond specifying a few parameters for symmetry detection and shape deformation. We also show how our symmetrization algorithm can be used for other applications such as correspondence computations and shape filtering.

Previous Work. Related research has mostly been done in the area of symmetry detection for geometric objects. Early papers focused on detecting exact symmetries in 2D and 3D planar point sets [Atallah 1985], [Wolter et al. 1985], which limits their applicability for more complex geometries. A method for approximate symmetry detection has been proposed by Zabrodsky et al. [1995], [Zabrodsky and Weinshall 1997]. They define a symmetry measure for a single given transformation as the distance of a shape to the closest symmetric shape. Martinet et al. [2006] find global symmetries of 3D objects by analyzing the extrema and spherical harmonic coefficients of generalized moments. There has also been increasing interest in more general symmetry transforms, see [Podolak et al. 2006] and references therein. Here the goal is to define a continuous measure for all possible transformations of a certain symmetry class. This enables various applications including shape matching, alignment, segmentation or viewpoint selection. Other techniques that analyze distributions in transformation space include the RANdom Sample Consensus (RANSAC) method [Fischler and Bolles 1981], as well as geometric hashing techniques [Wolfson and Rigoutsos 1997] that have recently been applied successfully for partial shape matching [Gal and Cohen-Or 2006]. The medial axis transform captures local reflective symmetries with respect to a point [Blum 1967], which can be accumulated to extract more global symmetries. The rather involved computations and the inherent instability of the medial axis have prevented a wide-spread use so far. However, recently proposed stable versions of the medial axis (see [Attali et al. 2004] for an overview) can potentially alleviate these problems. Symmetry has also been exploited to define shape descriptors [Kazhdan et al. 2002] that represent global reflective and rotational symmetries with respect to a shape's center of mass. This approach has been applied successfully for alignment, classification and shape matching [Kazhdan et al. 2004]. More recent work uses symmetry detection for segmentation [Simari et al. 2006], or scan completion [Thrun and Wegbreit 2005]. Various symmetrizations are known in classical geometry, e.g. symmetrization of convex sets [Grunbaum 1963], or Steiner symmetrization that maps a subset of Euclidean space to a set of spheres, while preserving volume and convexity [Hadwiger 1957]. Symmetrization methods are also used in function theory [Faber 1920] and tensor algebra [Schouten 1951]. However, to the best of our knowledge, no method for symmetrization of general geometric shapes has been proposed before.

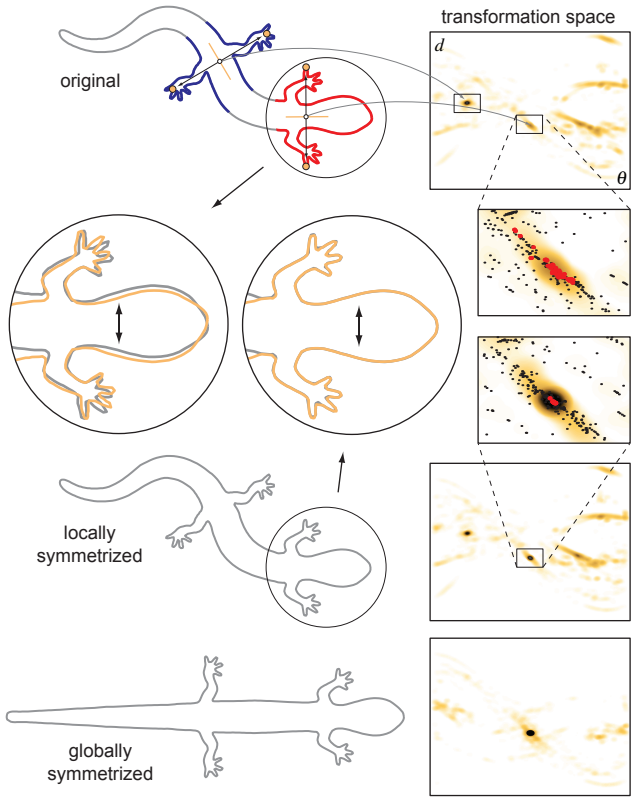

Figure 2: Symmetrization in 2D. The original shape is sampled, and compatible sample pairs are mapped into the space of reflective transformations shown on the right as density plots. Correspondences are found by checking the spatial consistency of points within clusters, yielding the marked regions on the model (and the red points shown in the zooms). After contracting clusters in transformation space, the shape is locally symmetric as shown by the zoomed overlaid reflected geometry. Merging the clusters and further incremental contraction finally yields the globally symmetrized model shown with corresponding transform plot at the bottom.

2 Overview

We briefly review the symmetry detection method introduced in [Mitra et al. 2006] that forms the basis of our symmetrization algorithm. Partial symmetry relations of a shape S can be defined as the invariance of subparts $S_1, S_2 \subseteq S$ under a certain transformation T such that $T(S_1) = S_2$. Transformations typically include reflections, translations, rotations, or scaling. Such a symmetry transformation induces a point-wise correspondence between S_1 and S_2, i.e., every point $\mathbf{p} \in S_1$ is paired with a symmetric point $T(\mathbf{p}) \in S_2$. To find an unknown symmetry transformation T, we estimate this correspondence by sampling the model and pairing compatible samples on S. Compatible here means that the local geometry at the two sample points can be matched with a transformation of the specific symmetry class in consideration.

Figure 2 illustrates our approach for reflective symmetries in 2D. The boundary curve of the gecko is sampled with 577 points, which results in 4780 matching point pairs. Each pair defines a unique re-

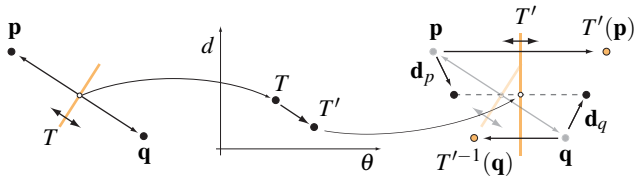

Figure 3: Optimal displacements $\mathbf{d}_p$ and $\mathbf{d}_q$ for a point pair $(\mathbf{p},\mathbf{q})$ when changing the transformation from T to T'.

flection line, parametrized by an angle θ and the distance d to the origin, that can be seen as evidence for this specific (local) symmetry relation. By accumulating such evidence in transformation space, more extensive symmetries can be found using spatial clustering methods[1]. Since a cluster does not necessarily correspond to a connected region in the spatial domain, the algorithm extracts the symmetric patches using incremental region growing. This process effectively computes the correspondence between the symmetric patches by discarding points in transformation space, and corresponding point pairs in the spatial domain, that do not belong to spatially connected parts (see [Mitra et al. 2006] for details). For the gecko model this yields two clusters that correspond to the red and blue regions shown in the figure.

Our goal is symmetrization, i.e., finding a deformation that enhances approximate symmetries present in a given shape. The deviation from perfect symmetry can be observed as variance of the clusters in transformation space, as shown in the zoom of the red cluster. Our algorithm deforms the shape in such a way that the extracted clusters are contracted, while altering the shape as little as possible. As shown in Figure 2, contracting clusters nicely symmetrizes the shape locally, but fails to recognize more complex global transformations, such as the reflective symmetry along the curved spine of the gecko. The algorithm thus proceeds by merging different clusters in transformation space, yielding a global deformation that unbends the spine. This symmetrization process is formulated as an optimization that is based on optimal displacement vectors for which we derive closed-form solutions in Section 3. More details of the algorithm will be discussed in Section 4.

3 Optimal Displacements

The goal of symmetrization is to find the minimal deformation that achieves perfect symmetry. For this purpose we define a coupling between transformation space and the spatial domain that specifies how pairs of points are displaced when their corresponding local symmetry mappings are modified.

Suppose we are given two points $\mathbf{p}$ and $\mathbf{q}$ defining a unique reflective transformation T that maps $\mathbf{p}$ to $\mathbf{q} = T(\mathbf{p})$ (see Figure 3). T is represented by the line through $(\mathbf{p}+\mathbf{q})/2$ with normal direction $\mathbf{p}-\mathbf{q}$ and can be expressed as a point $T = (\theta, d)$ in a 2D transformation space, where θ is the angle with some fixed reference line and d denotes the distance to the origin. If we now want to move the transformation T to $T' = (\theta', d')$, the points $\mathbf{p}$ and $\mathbf{q}$ need to be displaced by some vectors $\mathbf{d}_p$ and $\mathbf{d}_q$ such that $T'(\mathbf{p}+\mathbf{d}_p) = \mathbf{q}+\mathbf{d}_q$. To minimize the effect of this local change of transformation on the shape of the model, we are interested in the displacements with smallest magnitude, i.e. the ones that minimize $\|\mathbf{d}_p\|^2 + \|\mathbf{d}_q\|^2$. Simple geometric considerations show that these optimal displacements are given as

$$\mathbf{d}_p = \frac{T'^{-1}(\mathbf{q}) - \mathbf{p}}{2} \quad \text{and} \quad \mathbf{d}_q = \frac{T'(\mathbf{p}) - \mathbf{q}}{2}, \tag{1}$$

[1] A similar pairing strategy has been used in [Podolak et al. 2006] to efficiently compute a planar reflective symmetry transform.

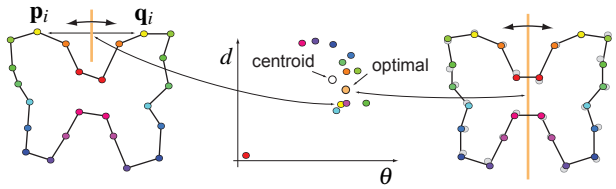

Figure 4: Optimal reflection plane and displacements for a set of corresponding point pairs. Correspondences are indicated by color, e.g., the transformation of the yellow sample pair is indicated by the yellow dot in transformation space. Gray dots on the right show the original positions.

effectively moving each point halfway to its perfectly symmetric counterpart, thus evenly splitting the displacements. This relation also holds for other mappings such as rigid transformations. Displacing a point T to T' in transformation space thus induces these minimal displacements of the corresponding sample points in the spatial domain.

Given the single pair displacements, we derive closed-form solutions for the optimal symmetrizing transformation of a given *set* of corresponding point pairs. For illustration we will again consider a 2D example with reflective symmetry shown in Figure 4. Assume we are given a set of point pairs $\{(\mathbf{p}_1,\mathbf{q}_1),\ldots,(\mathbf{p}_m,\mathbf{q}_m)\}$ on the boundary of a shape. The goal is to find the optimal reflective symmetry transform T and corresponding displacements that make the point set symmetric with respect to T. Optimal here means that the transformation T minimizes the symmetry cost

$$E = \sum_{i=1}^{m} (\|\mathbf{d}_{p_i}\|^2 + \|\mathbf{d}_{q_i}\|^2) = 2\sum_{i=1}^{m} \|\mathbf{d}_{p_i}\|^2.$$

We show in the appendix that this problem can be reduced to a 3×3 eigenvalue problem, where the eigenvector with smallest corresponding eigenvalue equals the normal of the symmetry plane. Given the optimal transformation T, the minimal displacements are then computed using Equation 1. Similarly, we derive a closed-form relation for rigid transformations (see appendix), which computes the parameters of the transformation using singular value decomposition. Note that this optimal transformation ($E_{opt} = 69.4$) differs from the centroid of the pair transformations ($E_{cen} = 142.6$), see Figure 4.

4 Optimization

The optimal displacements derived in the previous section form the basis of our symmetrization algorithm. Note that in general we cannot simply apply these displacements directly to symmetrize the shape. This fails for two reasons: The initial random sampling of the model does not respect symmetries, and the correspondences estimated during the symmetry detection stage are potentially inaccurate and incomplete. Sample points on the surface might not be paired and will thus not be displaced, leading to incorrect results. In this section we propose solutions to both problems. We first discuss a procedure to locally optimize the sample points on the surface, before we describe an optimization method that continuously deforms the shape to maximize its symmetries.

4.1 Optimizing Sample Positions

The example of Figure 2 illustrates that the extracted symmetry clusters exhibit variance in the distribution of a cluster's points in transformation space. Two different factors contribute to this variance: Firstly, geometric deviations from perfect symmetry lead to

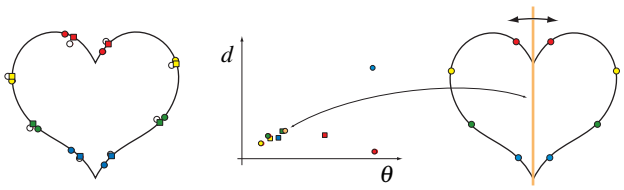

Figure 5: Optimizing sampling positions. Colored dots on the left show the original positions of matching pairs, white dots indicate the displaced positions and colored squares show the samples after re-projection. The corresponding transformations are shown in the center. The samples on the right show the result after five iterations.

variations in the transformations defined by each sample pair. Secondly, even if the geometry is perfectly symmetric, the samples generated on the model typically do not respect this symmetry exactly, leading to additional variance.

Deformations of the shape should only be triggered by geometric deviations from symmetry, not by non-symmetric sample placement. Therefore, we first perturb sample points on the surface of the model to optimize the sampling positions with respect to a cluster's symmetry transform. Every sample $\mathbf{p} \in S$ will be shifted in the direction of its optimal displacement $\mathbf{d}_p$. Since this will typically move samples off the surface, we project them back onto the surface and re-compute the optimal transformation and displacements vectors. We additionally restrict the sample movement to a sphere with radius equal to the local sample spacing to avoid excessive drifting. This procedure is iterated until the variance of the cluster is no longer reduced. Effectively, samples glide along the surface to positions where the symmetry score is maximized. This reduces the variance of a cluster without modifying the geometry and thus leads to smaller subsequent deformations. Figure 5 shows an example. For this perfectly symmetric shape all transformations are contracted to a single point, i.e., on the right all samples are symmetric with respect to the same reflection line, shown in yellow.

4.2 Symmetrizing Deformation

The deformation required for symmetrization can be formulated as an optimization where the displacement vectors are considered as directions of locally steepest descent pulling the shape towards the desired symmetry. Since each point pair is considered independently, we need a coupling between neighboring points on the surface of the model to obtain a coherent surface deformation. This regularization can be achieved using existing shape deformation methods that can incorporate the symmetrizing displacements as either positional constraints or forces acting on the shape. An important requirement for our application is detail-preserving deformation. In particular, shape features should be rotated correctly to avoid unnatural shearing effects that could destroy symmetries. In addition, we need a quantitative measure of the magnitude of the deformation, e.g., in form of a deformation energy, which allows the algorithm to favor symmetrizing deformations that involve the least change of shape.

We found two methods to be particularly suitable for our purposes. For 2D models we use the as-rigid-as-possible shape manipulation method of [Igarashi et al. 2005] shown in Figure 6. In this illustrative example we deform the shape by moving only two points in transformation space to their centroid. The corresponding surface samples are displaced according to Equation 1 and used as constraints for the 2D deformation method. The interior meshing required by this approach is computed similar to [Alliez et al. 2005]. For 3D shapes we use the non-linear PriMo deformation model of [Botsch et al. 2006] that is based on a simplified thin-shell model and has shown to provide intuitive detail preservation. However, other shape deformation methods can also be applied, e.g. [Pauly et al. 2005], as long as they satisfy the above requirements.

Contracting clusters. The symmetry detection algorithm of [Mitra et al. 2006] provides us with sample pairs $\{(\mathbf{p}_i, \mathbf{q}_i)\}$ on the initial model S corresponding to points $\{T_i\}$ of a cluster in transformation space. We first optimize sample positions on S as described in Section 4.1. Then we compute the optimal transformation (Section 3) and apply the resulting displacements $\mathbf{p}_i \rightarrow \mathbf{p}_i + \Delta t \mathbf{d}_{p_i}$ (analogously for $\mathbf{q}_i$), where Δt is the time step of the optimization (we chose $\Delta t = 0.1$ for all our examples). These new sample locations are used as positional constraints when evaluating the deformation model to obtain a deformed shape S'. Sample positions are then re-positioned on S', new optimal displacement vectors are computed, and the procedure is iterated. Since deforming the shape can also affect the point pair matching, we frequently re-apply the symmetry detection stage to update correspondences. For all our examples we found it sufficient to re-compute the pairing every 5 time steps, which avoids unnecessary computational overhead by exploiting the temporal coherence of the pair correspondences.

Merging clusters. Our symmetrization requires reliable correspondences that are extracted from sufficiently pronounced clusters in transformation space. When contracting such a cluster as described above, the regularization of the deformation model causes adjacent points that are not part of the cluster to move with the constrained samples to minimize the deformation energy. This shifts the transformations of these sample pairs towards the cluster. Figure 7 shows this behavior for the gecko model (see also Figure 2). If the separation of clusters is more distinct, however, the automatic contraction can fail to recognize certain global symmetries. An example can be seen in Figure 8 where the two distinct clusters correspond to the approximative reflective symmetries of the bunny's head and body. Contracting each cluster individually will not produce a deformation strong enough to rotate the head into the globally symmetric position. Therefore, we merge neighboring clusters after contraction by defining additional displacements that attract clusters to each other. Two clusters at T and T' are merged by moving all cluster points to their common centroid (see also Figure 6).

Our algorithm proceeds as follows: Clusters are first sorted by height and the most pronounced cluster is selected for symmetrization. We apply the symmetrizing deformations until the deformation energy indicates that the deviations from the original shape have exceeded a certain threshold. The process is then repeated with the next biggest cluster until all clusters of a suitable size have been symmetrized. Finally, we greedily merge clusters based on their distance in transformation space.

Control. Apart from these threshold parameters we provide two additional means to control the optimization: In the deformation model we treat the displaced positions as soft constraints, i.e., the

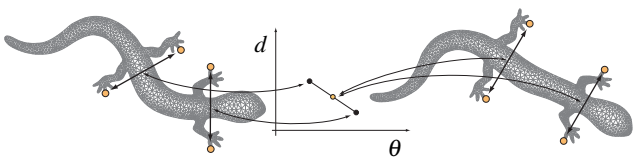

Figure 6: As-rigid-as-possible shape deformation with interior meshing. Moving two points in transformation space to their centroid induces the deformation shown on the right.

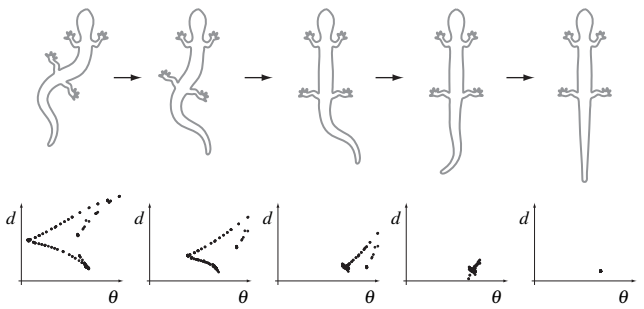

Figure 7: Symmetrizing deformation. The transform plots illustrate the paths taken by all pairs of the final contracted cluster. Not all of these points are initially part of a cluster, but get continuously attracted to the cluster due to the regularization of the deformation.

symmetrizing displacements are not met exactly in general. This allows the user to control the deformation by modifying the stiffness of the shape's material. Soft materials will allow for better symmetrization at the cost of potentially greater deviation from the rest shape, while stiffer materials more strongly resist the symmetrizing deformations to retain the original shape. Since both deformation models applied in our system allow spatially varying stiffness, this gives the user flexible control over the behavior of the shape. In addition to the stiffness parameter, a simple interface allows the user to control the symmetrization by interactively selecting clusters for contraction or merging. This can be useful when a certain semantic meaning is associated with a symmetry that is not recognized by the system automatically.

5 Results and Discussion

Figures 8 to 12 illustrate our symmetrization algorithm on a variety of different shapes. In all examples we consider the symmetry group of reflections, rotations, and translations, i.e. a 6D transformation space. While all computations are performed in the full 6D space, the visualizations show 2D projections computed using multi-dimensional scaling [Cox and Cox 1994]. The circular structures noticeable in some of the density plots are artifacts of this projection. All shapes have been symmetrized fully automatically

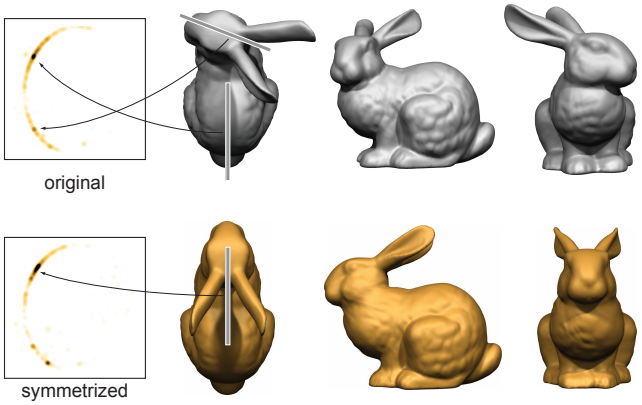

Figure 8: Symmetrizing the Stanford bunny. Two clusters corresponding to reflective symmetries of the head and body of the model are merged to obtain a globally symmetric shape. The transform plots on the left are projected from 6D to 2D for visualization.

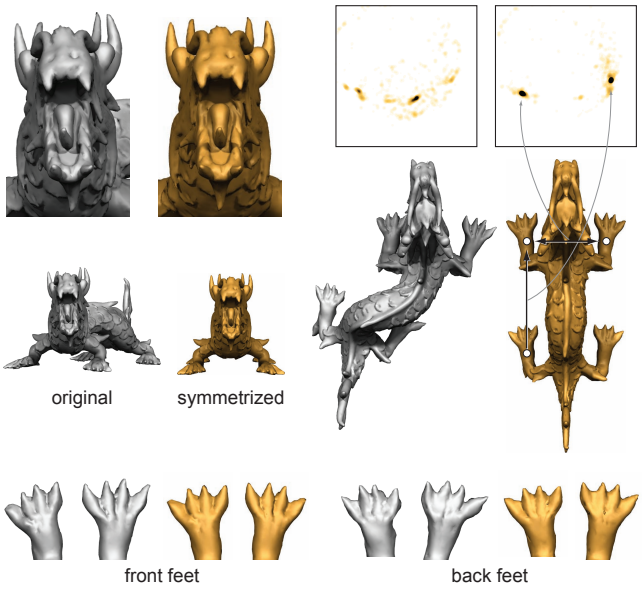

Figure 9: Symmetrization of a sculpted model. Large-scale deformations achieve a global reflective symmetry by straightening the spine, while many small-scale deformations symmetrize the model locally, e.g., the horns, tongue, or toes.

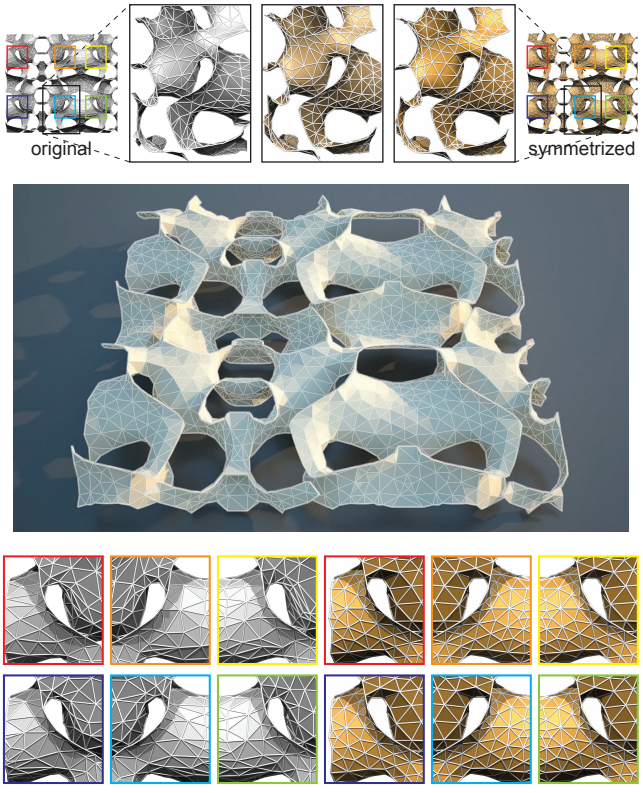

Figure 10: An architectural design study. The zooms in the top row show how the meshing of the extracted symmetric element evolves during the optimization. This element appears six times in different location and orientation, as illustrated in the bottom row.

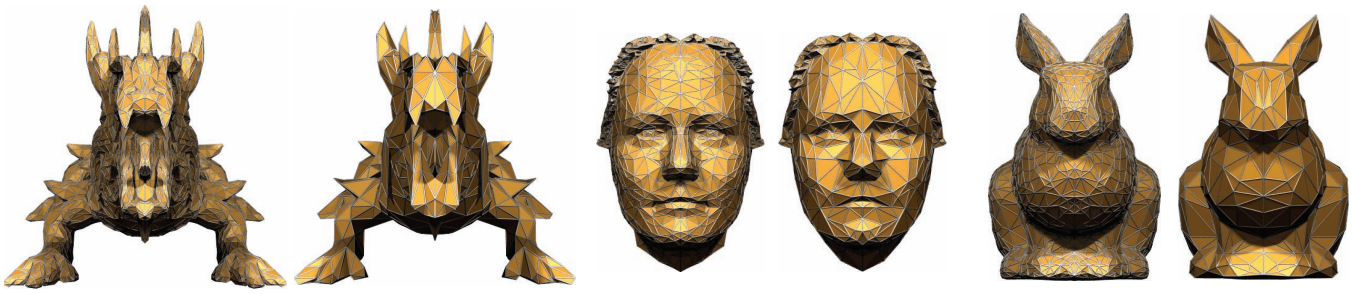

Figure 11: Perfectly symmetric meshes can be obtained using symmetrizing deformations, symmetric sampling, and symmetry-aware meshing. The number below each model indicate the corresponding vertex count.

using a fixed global stiffness in the PriMo deformation model.

Symmetrization of the Stanford bunny is shown in Figure 8. The head is rotated by merging the two dominant clusters. Prior contraction of these cluster leads to more subtle changes noticeable in the fur, the ears and the hind legs.

Figure 9 shows the symmetrization of a scanned hand-sculpted statue with many approximate symmetries. Our algorithm successfully symmetrizes the feet, head and tail of the dragon using cluster contraction and achieves a global reflective symmetry using cluster merging. The two main clusters in the final transform plot correspond to the global reflective symmetry across the spine and the partial translational symmetry that matches the front and back feet.

The design study of Figure 10 contains many repetitive, partially symmetric elements. This example illustrates how the symmetry-aware sampling of Section 4.1 can be used to also symmetrize the discretization of the model, which is important in certain areas such as CAD or architectural design. Similarly, Figure 11 shows remeshed versions of the bunny, dragon, and face models. After symmetrization of the geometry and placement of symmetric samples, the mesh connectivity of these samples is inferred from the connectivity of the original model. Different resolution levels can be obtained by adapting iterative simplification methods based on edge contraction, e.g., [Garland and Heckbert 1997], such that edges coupled by a symmetry relation are always contracted together. These perfectly symmetric meshes can be beneficial for compression and efficient rendering, as the inherent redundancy is explicitly represented in the mesh. Similarly, physical simulations can better preserve the symmetries of the underlying dynamics, since these symmetries are matched by the discretization.

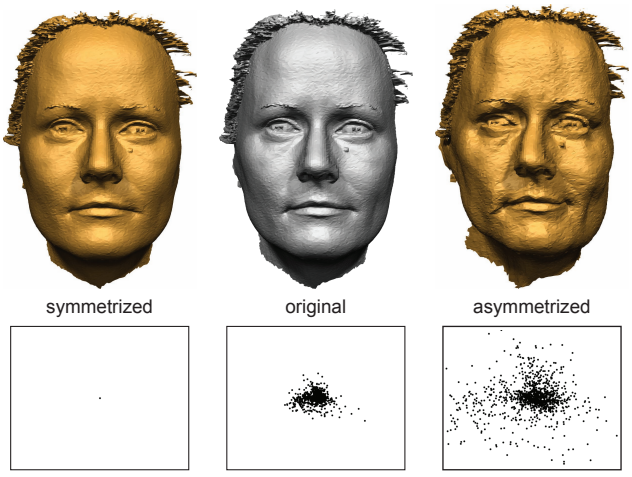

Figure 12: Symmetrization and its inverse operation. The original face scan has been symmetrized using cluster contraction. Moving points in the opposite direction yields interesting feature-enhancing effects. The transform plots show transformations of consistent corresponding pairs of the dominant cluster.

Figure 12 shows the inverse of a symmetrizing deformation, i.e., asymmetries are amplified by moving points along the negative direction of optimal displacement. This creates a certain enhancement effect that can bring out distinctive characteristics of a face. Notice also how this process amplifies high-frequency noise, which is reduced in the symmetrized version.

Figure 13 shows an example where we apply our algorithm to compute point-wise correspondences between two different poses of a scanned human. Sample pairs are restricted to always contain one sample from each model, i.e. each point in transformation space describes a local mapping from one shape to the other. Significant clusters thus define approximately rigid subparts of the shape that can then be contracted and merged to warp the two models towards each other. Regions that have already been matched are not considered in subsequent symmetry detection stages, which allows successively smaller clusters to be recognized by the algorithm. Note that these computations are fully automatic, in particular, no markers, manual pre-alignment or other user interaction are necessary.

Model	# Triangles	Detection	Contraction	Merging
Bunny	70,550	8.43	7.38	10.63
Dragon	64,378	36.03	22.57	34.66
Human	24,998	7.64	1.91	105.95
Face	34,889	4.48	3.58	-
Design	3,380	4.12*	2.56	-

*includes time for remeshing

The above table shows the accumulated total time in seconds for symmetry detection, cluster contraction and merging, measured on an Intel dual-core 2.4GHz with 2GB RAM.

The symmetrized shapes shown in the figures exhibit symmetries that cannot be found using existing symmetry detection algorithms. For example, prior methods, e.g., [Podolak et al. 2006], [Mitra et al. 2006], will not detect the global reflective symmetry of the model in Figure 9 due to the smooth bend in the spine of the dragon. Contracting and merging clusters effectively fuses different local symmetries into more global ones, thus extending the types of symmetries that can be detected in a given shape. However, similar to the above algorithms, our method currently cannot handle continuous symmetries, e.g., the rotational symmetries that define surfaces of revolution.

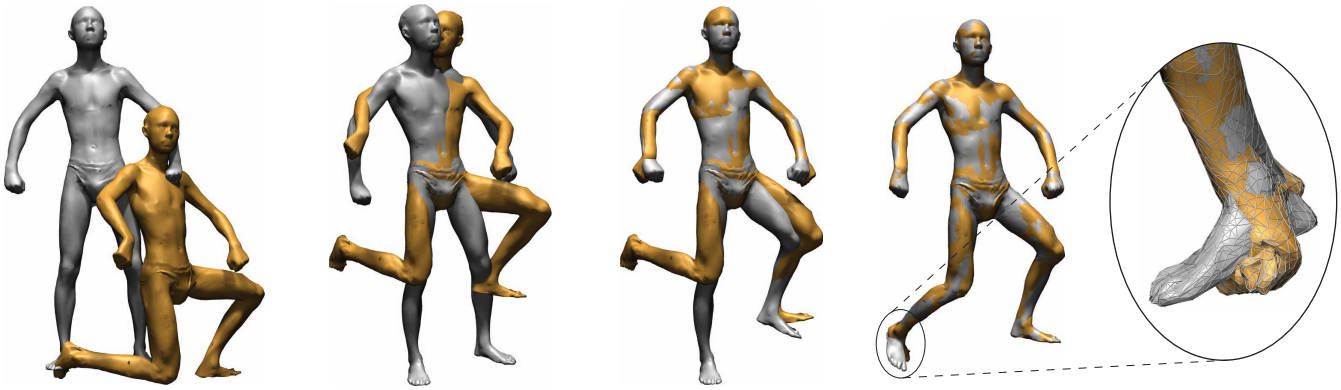

Figure 13: Different stages of a fully automatic correspondence computation. The two poses are deformed towards each other by successively contracting and merging the most prominent clusters. The right foot shown in the zoom cannot be matched due to insufficient pair matching.

Limitations. Figures 8 and 13 show two cases where our method fails to process the entire model correctly. The front feet of the bunny and the right foot of the male character did not get symmetrized properly. The reason in both cases is insufficient local matching. The curvature-based pairing does not find corresponding points in these regions due to significant differences in local geometry. Small-scale features are sometimes ignored for similar reasons, e.g., the eye brows in Figure 12. A possible solution are multi-scale descriptors for point pair matching as proposed for example in [Pauly et al. 2003] or [Funkhouser and Shilane 2006]. This can potentially improve the robustness of the point pairing step during symmetry detection. In its current form the correspondence computation illustrated in Figure 13 is only applicable for shapes that contain substantial rigid components. As such it can be used for registration of articulated bodies, but is not suitable for computing correspondence for general models, e.g. in order to perform complex morphing operations. A general limitation of our approach is that the deformation model does not respect the semantics of the shape. In particular, for certain shapes it might not be appropriate to merge cluster that are close in transformation space, because they do not belong to the same semantic symmetry. Such cases need to be disambiguated by the user. However, no such user intervention was required for any of the examples shown in the paper.

6 Conclusion

We have shown how geometric objects can be symmetrized using a coupled optimization that successively deforms a shape towards a symmetric configuration. Our algorithm is robust and efficient, requires minimal user intervention, and can handle both local and global symmetries. It will be interesting to further explore symmetrizing deformations that take into account a more complete physical model of the object being symmetrized, as currently our symmetrization is purely driven by geometric error. An interesting alternative would be to use a perceptual error measure. Maximum symmetrization with minimum perceptual change could lead to improved compression performance for symmetry-based compression algorithms like fractal compression. While in this paper we concentrate on enhancing symmetries in a given model, other operations in transformation space are conceivable, as illustrated in Figure 12. Such filters can possibly result in useful operations in the spatial domain that might be difficult to achieve otherwise. Imposing on one object the symmetries of another is an interesting possibility. We also want to explore the use of symmetrization for symmetry-aware segmentation, shape matching, and database retrieval.

A Optimal Transformations

Reflection. We derive an expression for the optimal reflection T that makes a set of point pairs $(\mathbf{p}_1,\mathbf{q}_1),\ldots,(\mathbf{p}_m,\mathbf{q}_m)$ symmetric with respect to T such that $\mathbf{q}_i = T(\mathbf{p}_i)\ \forall i$. Optimal here means that the displacements are minimal, i.e.,

$$E_T = \sum_i (\|\mathbf{d}_{p_i}\|^2 + \|\mathbf{d}_{q_i}\|^2) = 2\sum_i \|\mathbf{d}_{p_i}\|^2 = \sum_i ||(T(\mathbf{q}_i) - \mathbf{p}_i)||^2/2$$

is minimized. If we represent the reflection plane T by its normal $\mathbf{n}$ and distance d from origin, then for any point $\mathbf{p}$, $T(\mathbf{p}) = \mathbf{p} + 2(d - \mathbf{n}^T\mathbf{p})\mathbf{n}$. Thus we have

$$E_T = \sum_i \|(\mathbf{q}_i + 2(d - \mathbf{n}^T\mathbf{q}_i)\mathbf{n} - \mathbf{p}_i)\|^2/2.$$

We can now solve for the normal $\mathbf{n}$ and d that minimizes E_T, subject to the constraint $\mathbf{n}^T\mathbf{n} = 1$. In analogy to standard least squares fitting, it follows that E_T is minimized by the smallest eigenvector $\mathbf{n}^*$ of the matrix $(A - \mathbf{b}\mathbf{b}^T/2m)$, where $A = \sum_i(\mathbf{p}_i\mathbf{q}_i^T + \mathbf{q}_i\mathbf{p}_i^T)$ and $\mathbf{b} = \sum_i(\mathbf{p}_i + \mathbf{q}_i)$. The corresponding distance of the plane from the origin is given by $d^* = \mathbf{n}^{*T}\mathbf{b}/2m$.

Rigid Transform. Similarly, we derive an optimal rigid transformation for a given set of point pairs $(\mathbf{p}_1,\mathbf{q}_1),\ldots,(\mathbf{p}_m,\mathbf{q}_m)$. Let $C_p = [\mathbf{e}_p^1\ \mathbf{e}_p^2\ \mathbf{e}_p^3]$ be the local coordinate frame of $\mathbf{p}$ consisting of the surface normal, and the two principal curvature directions, respectively. We want to make the point pairs symmetric with respect to some rigid transform composed of a rotation R followed by a translation $\mathbf{t}$, i.e. $\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t}$ and $RC_{p_i} = C_{q_i}\ \forall i$. The cost of the corresponding minimal displacement of points, and alignment of the coordinate frames is proportional to

$$E_{(R,\mathbf{t})} = \sum_i \left(||R\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i||^2 + \lambda ||RC_{p_i} - C_{q_i}||_F^2 \right),$$

where λ is a positive constant weighting the two error components, and $||.||_F$ denotes matrix Frobenius norm [Hofer et al. 2004]. Our goal is to find the optimal rigid transform $(R^*,\mathbf{t}^*)$ which minimizes $E_{(R,\mathbf{t})}$. Setting $\partial E_{(R,\mathbf{t})}/\partial\mathbf{t} = 0$, we get $R\bar{\mathbf{p}} + \mathbf{t} = \bar{\mathbf{q}}$, where $\bar{\mathbf{p}} = \sum_i \mathbf{p}_i/m$ and $\bar{\mathbf{q}} = \sum_i \mathbf{q}_i/m$. Using $\tilde{\mathbf{p}}_i = \mathbf{p}_i - \bar{\mathbf{p}}$ and $\tilde{\mathbf{q}}_i = \mathbf{q}_i - \bar{\mathbf{q}}$, we have,

$$E_{(R,\mathbf{t})} = \sum_i \left(||R\tilde{\mathbf{p}}_i - \tilde{\mathbf{q}}_i||^2 + \lambda ||RC_{p_i} - C_{q_i}||_F^2 \right)$$

$$= \sum_i ||R\tilde{\mathbf{p}}_i - \tilde{\mathbf{q}}_i||^2 + \sum_i \sum_{j=1}^{3} ||\sqrt{\lambda}(R\mathbf{e}_{p_i}^j - \mathbf{e}_{q_i}^j)||^2$$

$$= \sum_{k=1}^{4m} ||R\mathbf{u}_k - \mathbf{v}_k||^2,$$

where we use suitable $\mathbf{u}_k$ and $\mathbf{v}_k$ to simplify notation. This reduces the problem to finding the rotation that best aligns a set of point pairs for which a closed form solution exists [Eggert et al. 1997]. Using this result, we can minimize $E_{(R,\mathbf{t})}$ as follows: Let $H = \sum_{k=1}^{4m} \mathbf{u}_k \mathbf{v}_k^T$ and $H = U\Lambda V^T$ its singular value decomposition. The optimal rotation is then given by $R^* = VU^T$ and the corresponding optimal translation vector is $\mathbf{t}^* = \bar{\mathbf{q}} - R^*\bar{\mathbf{p}}$.

Acknowledgements

This work is supported by grant S9206-N12 of the Austrian Science Fund, grant SNF 200021-112122/1 of the Swiss National Science Foundation, NSF grant CARGO 0310661, ARO grant DAAD19-03-1-0331, NIH grant GM-072970, and DARPA grant 32905. The authors are grateful to Takeo Igarashi and Mario Botsch for their help with shape deformation tools for curves and surfaces, respectively. Special thanks to Helmut Pottmann for his insightful feedback and support. The design example in Figure 10 was conceived by Benjamin Scheider.

References

ALLIEZ, P., COHEN-STEINER, D., YVINEC, M., AND DESBRUN, M. 2005. Variational tetrahedral meshing. *ACM Trans. Graph. 24*, 3, 617–625.

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

ATTALI, D., BOISSONNAT, J., AND EDELSBRUNNER, H. 2004. Stability and computation of the medial axis — a state-of-the-art report. *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*.

BLUM, H. 1967. A transformation for extracting descriptors of shape. In *Models for the Perception of Speech and Visual Forms*, MIT Press, 362–380.

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. Primo: Coupled prisms for intuitive surface modeling. In *Proc. Symposium on Geometry Processing*, 11–22.

COX, T., AND COX, M. 1994. *Multidimensional Scaling*. Chapman and Hall, London.

EGGERT, D. W., LORUSSO, A., AND FISHER, R. B. 1997. Estimating 3-d rigid body transformations: a comparison of four major algorithms. In *Mach. Vision Appl.*, vol. 9(5-6), 272–290.

FABER, G. 1920. Ueber potentialtheorie und konforme abbildung. *Sitzungsber. Bayer. Akad. Wiss. Math.-Naturwiss. Kl.*, 49–64.

FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Comm. of the ACM*, 381–395.

FUNKHOUSER, T., AND SHILANE, P. 2006. Partial matching of 3D shapes with priority-driven search. In *Symposium on Geometry Processing*, 131–142.

GAL, R., AND COHEN-OR, D. 2006. Salient geometric features for partial shape matching and similarity. *ACM TOG 25*, 1.

GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH '97*, 209–216.

GRUNBAUM, B. 1963. Measures of symmetry for convex sets. *Proc. Symposium Pure Math. 7*, 233–270.

HADWIGER, H. 1957. *Vorlesungen ueber Inhalt, Oberflaeche und Isoperimetrie*. Springer.

HOFER, M., POTTMANN, H., AND RAVANI, B. 2004. From curve design algorithms to the design of rigid body motions. *The Visual Computer*, 279–297.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph. 24*, 3, 1134–1141.

KAZHDAN, M. M., CHAZELLE, B., DOBKIN, D. P., FINKELSTEIN, A., AND FUNKHOUSER, T. A. 2002. A reflective symmetry descriptor. In *ECCV*, 642–656.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3d shape matching. In *Sympos. on Geometry Processing*, 116–125.

MARTINET, A., SOLER, C., HOLZSCHUCH, N., AND SILLION, F. 2006. Accurate detection of symmetries in 3d shapes. *ACM Trans. Graph. 25*, 2, 439–464.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph. 25*, 3, 560–568.

PAULY, M., KEISER, R., AND GROSS, M. 2003. Multi-scale feature extraction on point-sampled models. In *Proceedings of Eurographics*, 281–289.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. 2005. Example-based 3d scan completion. In *Symposium on Geometry Processing*, 23–32.

PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3d shapes. *ACM Trans. Graph. 25*, 3, 549–559.

SCHOUTEN, A. 1951. *Tensor analysis for physicists*. Cambridge Univ. Press.

SIMARI, P., KALOGERAKIS, E., AND SINGH, K. 2006. Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Proc. Symposium on Geometry Processing*.

THRUN, S., AND WEGBREIT, B. 2005. Shape from symmetry. In *Int. Conference on Computer Vision*.

WOLFSON, H. J., AND RIGOUTSOS, I. 1997. Geometric hashing: An overview. *IEEE Comput. Sci. Eng. 4*, 4, 10–21.

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

ZABRODSKY, H., AND WEINSHALL, D. 1997. Using bilateral symmetry to improve 3D reconstruction from image sequences. *Computer Vision and Image Understanding: CVIU 67*, 1, 48–57.

ZABRODSKY, H., PELEG, S., AND AVNIR, D. 1995. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*, 12, 1154–1166.