EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Skeleton-Intrinsic Symmetrization of Shapes

Qian Zheng[1] Zhuming Hao[1] Hui Huang[1†] Kai Xu[1] Hao Zhang[2] Daniel Cohen-Or[3] Baoquan Chen[4]

[1]Shenzhen VisuCA Key Lab / SIAT [2]Simon Fraser University [3]Tel-Aviv University [4]Shandong University

Abstract

Enhancing the self-symmetry of a shape is of fundamental aesthetic virtue. In this paper, we are interested in recovering the aesthetics of intrinsic *reflection symmetries, where an asymmetric shape is symmetrized while keeping its general pose and perceived dynamics. The key challenge to intrinsic symmetrization is that the input shape has only approximate reflection symmetries, possibly far from perfect. The main premise of our work is that curve skeletons provide a concise and effective shape abstraction for analyzing approximate intrinsic symmetries as well as symmetrization. By measuring intrinsic distances over a curve skeleton for symmetry analysis, symmetrizing the skeleton, and then propagating the symmetrization from skeleton to shape, our approach to shape symmetrization is* skeleton-intrinsic. *Specifically, given an input shape and an extracted curve skeleton, we introduce the notion of a* backbone *as the path in the skeleton graph about which a self-matching of the input shape is optimal. We define an objective function for the reflective self-matching and develop an algorithm based on genetic programming to solve the global search problem for the backbone. The extracted backbone then guides the symmetrization of the skeleton, which in turn, guides the symmetrization of the whole shape. We show numerous intrinsic symmetrization results of hand drawn sketches and artist-modeled or reconstructed 3D shapes, as well as several applications of skeleton-intrinsic symmetrization of shapes.*

Beauty is bound up with symmetry - Hermann Weyl [Wey83].

1. Introduction

Symmetry is often considered synonymous to beauty. However, while symmetric shapes or patterns are attractive, an excess of symmetry tends to be perceived as predictable and uninteresting due to a certain sterile rigidity about it [McM05]. A primary example is demonstrated by shapes exhibiting *extrinsic* symmetries. These objects are predominantly man-made artifacts and by design, their symmetries are meant to be perfect, rigid, and unbreakable. In contrast, many organic objects do not show up in an extrinsically symmetric form; they are non-rigid and can articulate freely. When such objects are captured or depicted artistically, their images often exhibit a varying degree of (extrinsic) asymmetry, e.g., due to an asymmetric pose. The symmetries of these objects are *intrinsic* to the objects themselves. Such symmetries are still beautiful and the presence of extrinsic asymmetry even adds a certain dynamics or liveliness to them. In this paper, we use the term "intrinsic" to attribute any measure or property associated with a shape representation which remains largely invariant under varying poses.

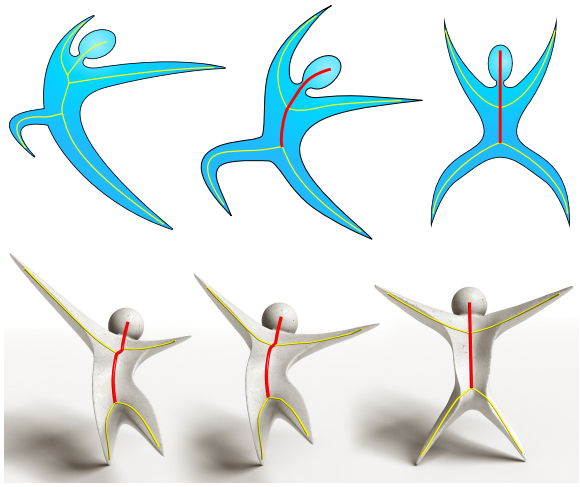


Figure 1: *Asymmetric shapes (left) are* intrinsically *symmetrized (middle) about their* backbones *(red), which facilitates further extrinsic symmetrization (right).*

† Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Symmetrization is a process that enhances the symmetry of a shape. The most notable such work, by Mitra et al. [MGP07], reveals and enhances the extrinsic reflection symmetries present in an intrinsically symmetric shape. For example, a wiggly figure would be "straightened up" with its corresponding limbs in the same poses. A complementary challenging objective is to enhance only the intrinsic reflection symmetry, while leaving the extrinsic asymmetries intact. In this paper, we introduce such an algorithm which we refer to as *intrinsic symmetrization*. Figure 1 contrasts intrinsic and extrinsic symmetrizations. In fact, starting with an asymmetric figure, intrinsic symmetrization is arguably a first step before extrinsic symmetrization becomes applicable.

The key challenge to intrinsic symmetrization is that the input shape is not assumed to be intrinsically symmetric. Rather than "extrinsicizing" existing intrinsic symmetries in a shape [MGP07], one must first search for and identify the *target* of intrinsic symmetrization. Current methods for intrinsic symmetry detection [OSG08, XZT*09, KLCF10] have all been designed to *reveal* existing symmetries, typically relying on strong cues of recurrent features and local similarities. In contrast, our algorithm must work with shapes which only exhibit weak evidences of symmetries.

The main premise of our approach is that curve skeletons provide a suitable shape abstraction for analyzing approximate intrinsic symmetries as well as symmetrization. First, any prominent intrinsic symmetry over a shape must correspond to some symmetry between its skeletal branches. Second, the need for intrinsic symmetrization mostly arises for imperfectly constructed shapes in practice, e.g., free-hand sketches or roughly modeled or captured 3D objects, which are often tempered with boundary noise and irregularities. Symmetry analysis on such boundary representations is expected to be less robust than on a high-level structural abstraction such as a curve skeleton. Finally, with only weak local symmetry cues, an effective analysis needs to resort to *global* search. Working with curve skeletons, a *reduced* shape representation, reduces the search cost.

Given an input shape with an extracted curve skeleton, we introduce the notion of a *backbone* as a sub-path in the skeleton graph about which a properly defined self-matching score of the input shape is maximized. We define an objective function for the reflective self-matching, which accounts for approximate intrinsic reflection symmetries, as well as potentially significant deviation from perfect symmetries. Since backbone extraction necessitates a global search, we develop an algorithm based on genetic programming which operates on the curve skeleton. The extracted backbone guides the intrinsic symmetrization of the curve skeleton, which in turn, guides the symmetrization of the entire shape, in a similar manner to linear blend skinning (LBS) but with non-uniform scaling.

We call our approach *skeleton-intrinsic* since our symmetry analysis is primarily based on intrinsic distances measured over a curve skeleton extracted from the input shape. Symmetrization results are propagated from skeletons to shapes and not obtained by working with intrinsic distances over the shape surface. As such, we make no claims on symmetrizing any surface metric. We symmetrize a shape via skeleton-intrinsic analysis and skeletal deformation.

We demonstrate the robustness of our backbone extraction scheme on numerous examples with varying degrees of asymmetries, as well as geometric and topological noise. In contrast, existing symmetry detection schemes are unable to recover approximate symmetries as effectively. We show that our technique can be applied to both 2D sketches and 3D models, enhancing them into intrinsic symmetric shapes to benefit a number of applications.

2. Related work

Symmetry analysis and processing has been extensively studied in computer vision and graphics [LHOKG10, MPWC12]. Here we focus on works on intrinsic symmetry detection. Most methods for symmetry detection explicitly search for the maximal distance-preserving global or partial self-maps [XZT*09, RBBK10, LKF12, XZJ*12]. Other methods reveal global intrinsic symmetries in a shape by mapping the shape into an embedding space so as to reduce the degrees of freedom of intrinsic transforms [OSG08, KLCF10]. All of these methods rely on geodesic computations over well-formed surfaces and can only tolerate slight deviations from perfect symmetries. In contrast, our method is designed to deal with rough shape descriptions and to detect and then enhance approximate intrinsic symmetries.

Curve skeleton and matching. Our skeleton-intrinsic symmetry analysis leverages recent developments on robust curve skeleton extraction from rough shape descriptions [RvWT08], even incomplete point clouds [TZCO09, HWCO*13]. Curve skeletons indeed provide an effective and compact form for abstracting and analyzing shapes [CMS07], e.g., for shape matching. Bai and Latecki [BL08] propose a skeletal graph matching algorithm by comparing geodesic paths between skeleton endpoints. Au et al. [ATCO*10] rely on elector voting to build correspondences between two curve skeletons and the scheme also rests on an assumption of low distance distortion.

Symmetry axes. While curve skeletons can be seen as the axes of local volumetric symmetries [TZCO09], the topological structure of the skeleton itself does not reflect global intrinsic *shape* symmetries. For example, the single branch left after an aggressive skeletal graph contraction would generally not be the backbone we seek. Earlier work by Xu et al. [XZT*09] develop a voting scheme to reveal prominent axes of intrinsic reflection symmetries *on* the surface of a 3D shape. Recently, Jiang et al. [JXCZ13] apply the voting scheme of Xu et al. [XZT*09] on curve skeletons for

symmetry detection on a 3D shape, where only slight deviation from perfect symmetries can be tolerated. In our work, a backbone roughly corresponds to a intrinsic reflection symmetry axis on the curve skeleton, but it must be able to tolerate significant devision from perfect symmetries.

Symmetrization. Mitra et al. [MGP07] "extrinsicize" intrinsic symmetries present in a shape by a deformation guided by the symmetric point pairs detected in the transformation space. Transformation-space analysis has proven to be effective for extrinsic symmetry detection and symmetrization, since extrinsic symmetries are captured by significant clusters or peaks in the transform space. However, the continuous nature of intrinsic symmetries implies that no such clear targets for symmetrization can be reliably identified. The work of Mitra et al. deals only with shapes that are close to being intrinsically symmetric. In contrast, we work on intrinsically asymmetric shapes, extract the approximate intrinsic symmetries therein, and enhance such symmetries while keeping the extrinsic asymmetries intact.

Enhancing free-hand drawings. Orbay and Kara [OK11] rectify a user sketch by stroke clustering and curving fitting. Lu et al. [LYFD12] collect a set of high-quality strokes from a trained artist and then transfer the learned styles to the drawings or writings of novices. Zitnick [Zit13] beautifies hand-writings using token means, which is based on the observation that the geometric average between handwritings of the same word is prettier than most of the individual instances. What is common about all of these methods is that they work at the stroke level. In contrast, our method aims to enhance the intrinsic symmetries of a curve skeleton and ultimately, of the corresponding sketched shape. Intrinsic symmetries provide views of a shape at a more global perspective beyond the appearance of individual strokes.

3. Overview

A shape representation S possesses a global intrinsic symmetry if there is a homeomorphism $T : S \to S$, which is an isometry [XZT*09]. The shape representation can be the boundary or a skeleton of an object. The key is for T to preserve intrinsic distances defined for the shape representation. The choice of the distance measure depends on the choice of the shape representation. For surfaces embedded in 3D, geodesic distances are most often employed. For 2D shapes, one often resorts to inner distance [LJ07], which measures the length of the shortest *interior* path between two points on the shape boundary. In our work, we focus on intrinsic reflection symmetries and perform a skeleton-intrinsic analysis, where intrinsic distances are measured over curve skeletons.

The input to our algorithm is a 2D or 3D shape defined by its boundaries. The input geometry can be rough, even a point cloud, as long as we are able to extract a proper curve skeleton. We employ the curve skeleton extraction scheme of Au et al. [ATC*08] for 3D shapes, Reniers et al. [RvWT08] for 2D shapes, and the ℓ_1-medial scheme of Huang et al. [HWCO*13] for incomplete shapes or point clouds. All the results reported in the paper have been obtained by working on *automatically* extracted skeletons.

By symmetrization, our goal is to enhance the approximate, yet global intrinsic reflection symmetries in the input shape. Directly identifying such symmetries for general shapes is quite challenging. Our symmetry analysis is skeleton-driven and skeleton-intrinsic. It focuses on the detection of a backbone, a path along the extracted curve skeleton of the shape about which a reflective self-mapping between all skeleton branches is maximized. Importantly, our symmetry analysis is not exclusively based on comparing intrinsic distances measured over curve skeletons, we also take into account matching between corresponding shape geometries.

Backbone extraction. Given a potential candidate for the backbone, i.e., a path along the curve skeleton, we evaluate it by a *self-matching score* of the input shape about the candidate path. The score serves as the objective function for backbone searching. Specifically, we search for an optimal self-matching between samples on the skeleton graph, considering not only the local geometric similarity between the samples, but also pairwise relationships, as defined by skeleton-intrinsic distances among them; see Figure 2. Both skeletal connectivity and shape geometry are taken into account and the candidate's self-matching score is derived from its corresponding optimal self-matching. To execute the global search for the optimal backbone, we develop an algorithm based on genetic programming; see Section 4.

Intrinsic symmetrization. Having obtained the backbone and corresponding optimal skeleton self-matching, we offer two options for intrinsic symmetrization of the input shape: computing an *average* between symmetric parts, or *copying* parts from one side (of the extracted approximate symmetry) to the other, where the latter can be carried out branch by branch. As copying is quite straightforward, we mainly describe the averaging scheme in Section 5, which alters the input shape in two steps. First, the skeleton is intrinsically symmetrized about the backbone. Then, the shape, or the *skin*, is symmetrized following the symmetric skeleton. The procedure is illustrated in Figure 6 using a 2D example. Various options based on size or shape quality, e.g., roughness transfer or smoothing, can be easily implemented, allowing intrinsic symmetrization to become a versatile modeling and shape enhancement mechanism.

4. Backbone detection

Given a shape S with its skeleton G as input, we define its backbone as the optimal path P^* in G about which a self-matching score of S is maximized. That is,

$$P^* = \underset{P \in \Pi}{\operatorname{argmax}} \mathcal{S}(M|P), \tag{1}$$

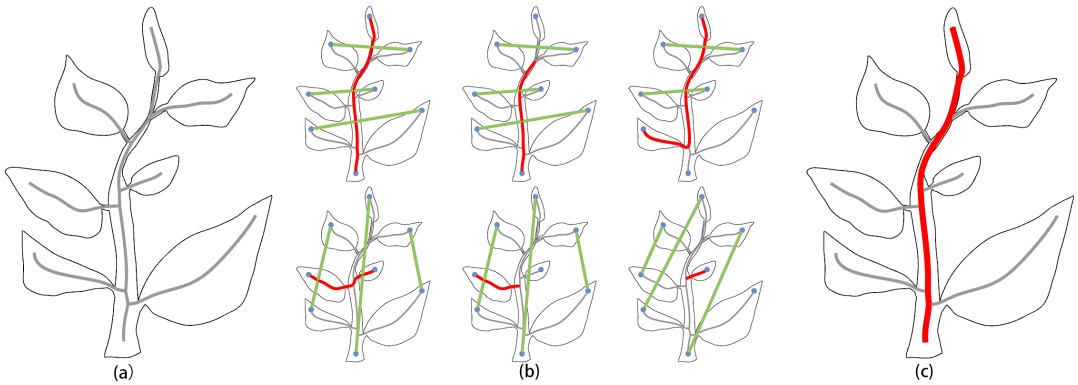


Figure 2: *Backbone detection: given a shape with an automatically extracted skeleton (a), red curves in (b) represent some potential backbone candidates, where green lines denote the established self-matching between end nodes (blue dots) about the candidate; the red path in (c) is our detected backbone with the optimal self-matching.*

where Π is the set of all paths over the skeleton between any two nodes whose valences are not equal two, i.e., end or junction nodes. $M|P$ represents the non-identity self-matching with path P as fixed points, i.e., reflective about P. Since we consider only reflective maps, we disallow any point other than those on P to be matched to itself.

In general, computing a non-identity self-matching, even a coarse one, for a shape with given fixed points is quite challenging. If S deviates moderately from being perfectly symmetric, the correspondence cannot be established by directly imposing distance preservation constraints. We search for a self-matching on the skeleton G while taking shape information from S into account. To this end, we first sample G into a point set, and then establish a self-map over the point set with the sample points in P fixed; see Figure 2.

The key component in the optimization objective function (1) is the matching score $\mathcal{S}(M|P)$, which measures the quality of mapping $M|P$. In the following, we first elaborate the computation of $M|P$ and then define the matching score $\mathcal{S}(M|P)$. Finally, we describe how to obtain the optimal backbone by a genetic algorithm.

4.1. Skeleton mapping

Given a backbone candidate P, finding the skeleton self-mapping is equivalent to establishing non-identity correspondence for two coincident skeletons. This determines a global intrinsic reflective symmetry about P. Since the self-map should be pose-invariant and tolerant to non-isometric distortion, methods like branch matching [BTST12] or electors voting [ATCO*10] are unsuitable.

In our implementation, we use end nodes, i.e., the skeleton nodes having only one adjacent node, as samples E, and search for an optimal self-matching M over the end nodes. Focusing on end nodes instead of the full skeleton has two advantages. Firstly, the topology of a curve skeleton can be unreliable while its end nodes, typically representing the tips of brunches, are generally stable. Secondly, the surface around an end node is geometrically salient and hence easier to characterize. Our method is inspired by [BL08], which utilizes end nodes for matching skeletons of 2D shapes. We extend this algorithm to deal with 3D shapes.

Graph matching problem. We formulate the end node mapping as a bipartite graph matching problem. To this end, we construct a kNN graph $G = \langle G_V, G_E \rangle$ from the end nodes through connecting each node to its k nearest neighbors based on geodesic distance, and then find a non-identity map M between G and its cloned counterpart $G' = \langle V_{G'}, E_{G'} \rangle$.

To measure the quality of a matching, we consider not only the similarity between end nodes (node affinity), but also that of pairwise relationships (edge affinity). We search for the optimal M such that the sum of both node and edge affinity, denoted by $J(M)$, is maximized:

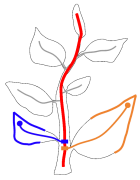
$$J(M) = \sum_{(i,i') \in M} K_p(i,i') + \sum_{\substack{\{i,i'\},\{j,j'\} \in M, \\ (i,j) \in E_G, (i',j') \in E_{G'}}} K_e((i,j),(i',j')). \tag{2}$$

Here $K_p(i,i')$ defines node affinity where $\{i,i'\} \in M$ is a pair of matching end nodes with $i \in V_G$ and $i' \in V_{G'}$; $K_e((i,j),(i',j'))$ defines edge affinity between edges $(i,j) \in E_G$ and $(i',j') \in E_{G'}$.

Node affinity. Given two end nodes i and i', the node affinity $K_p(i,i')$ is measured in terms of similarity between the nodes. To avoid a trivial self-mapping and encourage reflective maps about the candidate backbone, we use the candidate backbone to constrain the matching.

End node similarity is defined with three terms (see the embedded figure below). The first two terms are defined

with respect to the candidate backbone P. Specifically, we consider the branch connecting an end node i to P, whose length is denoted by l_{iP}. If a pair of end nodes match each other, their corresponding branches should match accordingly. Especially, the joining positions of their branches against P should coincide. Hence, we define the first term $d_b(i,i')$ as the geodesic distance between their joining positions along P. Meanwhile, we require their lengths to be close, so the second term is defined as a difference between branch lengths: $d_l(i,i') = |l_{iP} - l_{i'P}|/(l_{iP} + l_{i'P})$. The third term considers surface geometry in the vicinity of the two end nodes. We first identify for each end node an influenced surface region by examining the intrinsic distances between surface vertices and end nodes. For an end node, a surface vertex is influenced if its intrinsic distance to the end node is smaller than that to any other skeleton node. For each end node, we compute the average shape diameter function (SDF) values [SSCO08] over its influenced vertices. The geometric dissimilarity between two nodes is then defined as the difference between their average SDF values, denoted by $d_s(i,i')$.



Combining the three terms, end node similarity is defined as:

$$c(i,i') = \begin{cases} (c_b + c_l + c_s)/3, & \text{if } d_b < 2\sigma_b, d_l < 2\sigma_l \text{ and } d_s < 2\sigma_s \\ 0, & \text{otherwise.} \end{cases}$$

Here $c_b = g(d_b, \sigma_b)$, $c_l = g(d_l, \sigma_l)$, $c_s = g(d_s, \sigma_s)$, and $g(r,h) = e^{-2r^2/h^2}$ is a Gaussian kernel with support radius h. The parameters σ_b, σ_l and σ_s are used to tune the sensitivity of pair-wise differences. We set $\sigma_b = 0.04$, $\sigma_l = 0.3$ and $\sigma_s = \gamma$ by default (assuming the input shape, and hence skeleton, has been normalized into a unit cube). The parameter γ denotes the average difference between average SDF. If two end nodes are too different in terms of the measures, we set their similarity to zero to prevent a matching.

The node affinity $K_p(i,i')$ is defined with the cases:

A.1. If $i \neq i'$, i and i' can be matched (set $K_p(i,i') = c(i,i')$) except for the following two situations (set $K_p(i,i') = 0$):
A.1.a. i and i' are on the same sub-tree rooted at a node that directly connects to P;
A.1.b. In 2D case, i and i' lie on the same side of P. (End nodes have sidedness information with respect to P since P cuts the 2D shape and skeleton into two parts.)
A.2. If $i = i'$, there are two cases: if i is a node on P, it should match with itself so we set $K_p(i,i') = 1$. Otherwise, we avoid self-mapping by imposing a low self-affinity: $K_p(i,i') = 0.2$.

Edge affinity. The node affinity does not consider the spatial arrangement between skeleton nodes. To account for this, we consider the pairwise relationship between two end nodes i and j, and denote their shortest path on skeleton as $p(i,j)$. The edge affinity $K_e((i,j),(i',j'))$ measures the compatibility of two paths, $p(i,j)$ and $p(i',j')$, based on the statistic property derived from the distance-based symmetry criterion in [XZJ*12] which is pose-invariant:

$$D_d(p(i,j), p(i',j')) = \max\left(\frac{|l_{ij} - l_{i'j'}|}{l_{ij} + l_{i'j'}}, \frac{|l_{ij'} - l_{i'j}|}{l_{ij'} + l_{i'j}}\right),$$

where l_{ij} is the length of path $p(i,j)$.

The edge affinity is then defined as:

$$K_e((i,j),(i',j')) = \begin{cases} c_d/N_E, & \text{if } \substack{K_p(i,i')\neq 0,\ K_p(j,j')\neq 0,\\ D_d < 2\sigma_d} \\ 0, & \text{otherwise,} \end{cases}$$

where N_E is the number of edges, and $c_d = e^{-2D_d^2/\sigma_d^2}$ with $\sigma_d = 0.2$ penalizes the difference between path lengths.

Matching score. Given a candidate backbone P, we apply the factorized graph matching approach [ZDIT12] to find the optimal map $M|P$ that maximizes Equation (2). Note that the edge affinity generally favors one-to-one correspondences since an involute map would induce better mapping compatibility. In rare situations when $M|P$ has one-to-many correspondences, we simply remove the extra correspondences based on node affinity. From the resulting map, the matching score used in Equation (1) is:

$$\mathcal{S}(M|P) = J(M) - c_p, \tag{3}$$

where c_p penalizes the amount of unused skeleton segments. All segments on P are initialized as used while others are marked as unused. If end nodes i and j are matched, we mark the skeleton segments on $p(i,j)$ as used. We define $c_p = \sum_{i \in U}(l_i^2/(2\sigma_p^2))$, where U denotes the set of unused segments and l_i the length of segment i. We use $\sigma_p = 0.5\sigma_b$ as the default value.

4.2. Optimization by genetic algorithm

Since the optimization problem (1) is highly non-convex, we utilize a randomized search approach through enumerating candidate paths and choosing the one that maximizes the matching score in (3). The search space of candidate paths is in general very large, since any path between two skeleton nodes forms a candidate backbone. To solve this problem efficiently, we employ a genetic algorithm to evolve and explore a population of candidate solutions (paths), and use the matching score to evaluate the fitness of each individual.

To start, we randomly sample $s = 20$ candidate paths to form the initial population, and randomly select two nodes to generate a candidate path. When the first node is selected, a higher probability is assigned to other nodes, which are further away from the first selected one in order to favor the longer paths. Path crossover is performed between two intersecting paths by switching their end nodes, leading to two new paths. Figure 3 illustrates the crossover operation. Path mutation changes the end nodes of a path. Specifically, we randomly select a node which is not on the path and use it to

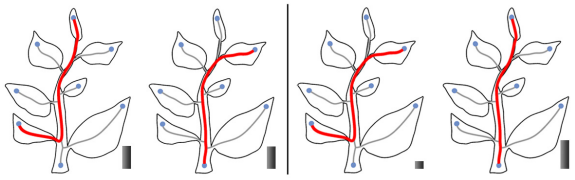


Figure 3: *The crossover operation of our genetic algorithm. Given two candidate backbones (left) of a shape, we generate two descendant backbones through switching their end nodes. The bars at the right corners represent the fitness score of the corresponding backbones.*

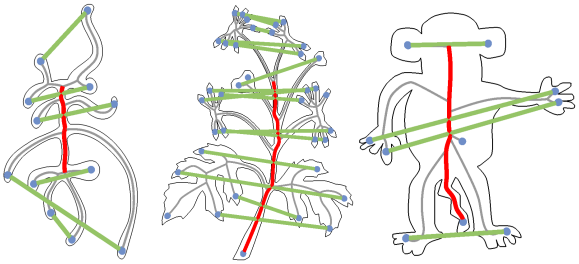


Figure 4: *Illustration of backbones we detect on three different 2D shapes.*

replace an end of the path. Here we prefer small changes, so a higher probability is given to nodes that are closer to end node to be replaced.

We execute a steady-state genetic algorithm to evolve the path population for $g = 20$ generations. In each generation, the top 50% fittest paths are migrated from the current generation to the next. The remaining population is formed by the newly generated paths using mutation and crossover. See Figure 4 for the backbones and associated end node correspondences we obtain on three different 2D shapes.

5. Backbone-guided symmetrization

Having obtained the backbone and end node correspondences, we perform skeleton-intrinsic symmetrization over the input shape via two steps based on the skeleton-shape mapping. We first symmetrize the input skeleton and deform the shape accordingly using a modified Linear Blending Skinning (LBS) method [JS11]; see Figure 6(c). Then we further adjust the shape to enhance its symmetry; see Figure 6(f). To obtain the skeleton-shape mapping, we first down-sample the curve skeleton into a skeleton composed of bone segments using Ramer-Douglas-Peucker (RDP) algorithm [Ram72]. A surface vertex receives an LBS influence weight from each bone, and is then associated with the bone which has the largest weight against it.

In our implementation, we opt to bring the two sides of the symmetric curve into their average shape. Other symmetrization options, such as warping one side to match the other, can be realized similarly based on this framework.

5.1. Skeleton symmetrization

In order to symmetrize the skeleton, we need to establish full correspondence for all skeleton nodes. This is achieved by matching the junction nodes along the branches.

Skeleton correspondence. The cost of matching two junction nodes is defined according to the backbone and end node correspondences. If a pair of end nodes matches, their corresponding branches attaching to the backbone should also match, and the junction nodes along these branches should be matched in a consecutive manner. We define the matching cost of junction nodes based on two factors, i.e., the difference between their connectivity degree and the difference between their position ratios which is the ratio of the distance from the node to the end node over the branch length.

The optimal matching between junction nodes should minimize the overall mapping cost of junction nodes on the skeleton, under the order constraint mentioned above. We adopt a greedy method to find the matching. Specifically, we greedily choose a pair of shortest unprocessed branches, with their end nodes matched, and find matching between the junction nodes along them using dynamic programming. The purple lines in Figure 6(b) illustrate the correspondence found for the junction nodes.

Symmetrization. Skeleton symmetrization strives to make the topology and geometry of the two sides of the backbone as similar as possible. If we view the two sides as two subgraphs of the input skeleton graph, this can be achieved by a greedy graph editing process. We define three graph edit operations, among which two are designed for topological symmetrization and one for geometric symmetrization. The first operation is to remove unused skeleton segments and their associated vertices on the shape; see Figure 5(b). The second operation is to move the joining nodes on the backbone of two corresponding branches to the middle position, if they do not coincide with each other; see Figure 5(c).

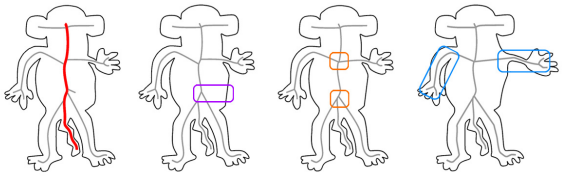


Figure 5: *Given the backbone (a), three operations are defined for topological symmetrization (b, c) and one for geometric symmetrization (d).*

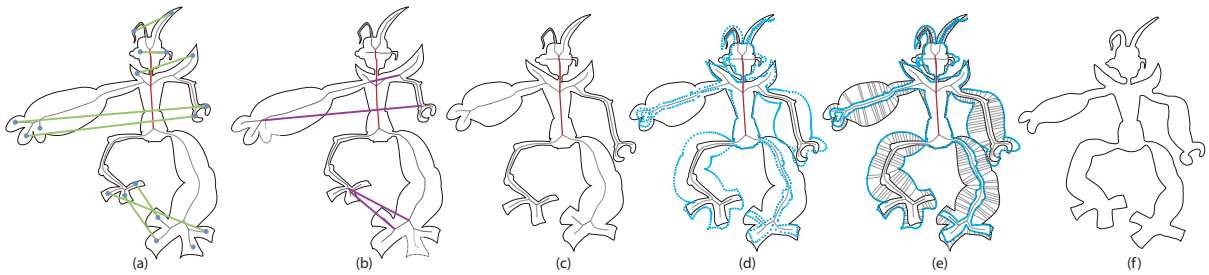


Figure 6: *Backbone-guided symmetrization: having obtained the backbone and the correspondence between end nodes (a), we first estimate the correspondences between junction nodes (the purple lines in b), and then symmetrize the skeleton about the backbone while deforming its skin accordingly (c). Note that unmatched skeleton segments and their associated surface (the dot lines in b) are simply discarded. The skin is then duplicated and reflected (blue dots in d). The two skins are correlated by matching each point on one skin to the nearest point on the other (e). The final intrinsically symmetric shape is simply the average between these two skins (f).*

To symmetrize the geometry of the skeleton, we uniformly stretch each pair of matched segments given by two pair of matched nodes. The shape is stretched accordingly using LBS; see Figure 5(d) for the finial symmetrized skeleton.

5.2. Shape symmetrization

The output shape after the skeleton symmetrization (denoted by M_S), e.g., in Figure 6(c), is deformed to its reflected version M_R, which serves as positional constraint to drive the symmetrization deformation. For this, we develop a non-rigid shape registration method based on the detail-preserving Laplacian shape deformation [SCOL*04].

Specifically, we first compute the reflection matrix of each bone, aligning it against its counterpart on the other side. By applying LBS transformations to the stretched shape M_S, we obtain M_R. Figure 6(d) demonstrates the reflected shape with blue dashed curves. In order to produce a smoother average shape of M_S and M_R, we perform Laplacian deformation over the shape M_S, under the positional constraints induced from M_R. That is to solve a least-squares problem:

$$\underset{V_A}{\operatorname{argmin}}(\|\Delta V_A - w_L L\|^2 + \sum_{i=0}^{m} W_{c,i}\|v_A^i - p^i\|^2) \qquad (4)$$

where Δ is the $n \times n$ curvature-flow Laplacian operator with conventional cotangent weights [MDSB03], L is the matrix encoding Laplacian coordinates and p^i is the constrained target position. The first term ensures the detail preservation and the second term imposes the positional constraints. The weight $w_L \in [0,1]$ is used to control the smoothness of the resulting shape, and $W_{c,i}$ is for tuning the importance of positional constraints. In our experiments, we use $w_L = 0.6$ by default to balance between the smoothness and geometric alignment. The reflected shape often exhibits skinning artifacts near the junction nodes. The vertices which are influenced by a single bone are more rigid and reliable as compared to those influenced by multiple bones with different transformations. Therefore, we compute $W_{c,i} \in [0,1]$ based on the rigidity of the targets [VBMP08].

Vertex correspondence. To compute constrained target position p^i for symmetrization, we build a vertex correspondence between the reflected shape M_R and the stretched shape M_S. Then p^i is the average position of every two corresponding vertices. Since the two shapes have the same pose (see the blue and black shapes in Figure 6(d)), we simply use a closest point matching. For each vertex v_S^i on M_S, we find its closest point v_R^i on M_R, to construct an initial correspondence. Then we iteratively improve the correspondence for v_S^i through looking for a better matched vertex within the K-nearest neighborhood of v_R^i, which minimizes the normal difference and Euclidean distance against v_S^i. We prune those correspondences which do not preserve the geodesic distance. This is achieved by a standard spectral matching [Szw05]. Figure 6(e) demonstrates the vertex correspondence obtained by our method.

Scaled Laplacian coordinates. Since Laplacian coordinates are not scale-invariant, we estimate the scaling of each bone based on the correspondences, and update the stretched shape M_S accordingly using LBS. The Laplacian coordinates L and the Laplacian operator Δ are computed on the updated shape. Specifically for each bone b_j, its scaling factor s_j is along the direction perpendicular to the bone. Thus, the scaling transformation of each bone is:

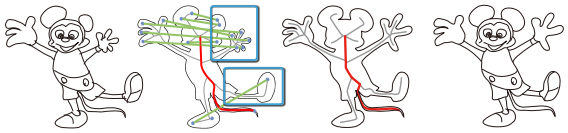
$$T_j = X_j{}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & s_j & 0 \\ 0 & 0 & s_j \end{pmatrix} X_j,$$

where X_i is the matrix which transforms b_j to X-axis.

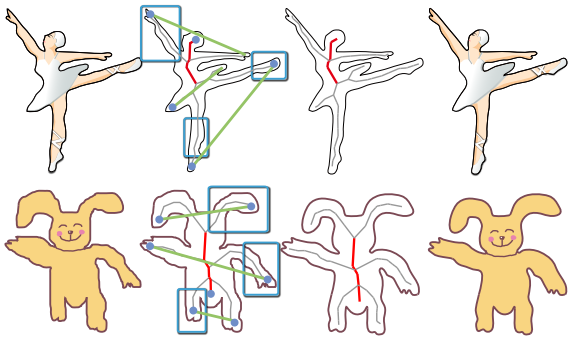
6. Results and applications

In this section, we show results of backbone extraction and skeleton-intrinsic symmetrization, evaluate their perfor-

(a) Symmetrization by copying enhances aesthetics of the structure.



(b) Symmetrization by copying enhances Mickey's left hand.



(c) Symmetrization by averaging smoothes sketch roughness.

Figure 7: *Skeleton-intrinsic symmetrization of 2D figures and sketches for beautification and enhancement. In each row, input and output are at the two ends. Middle two figures show skeletons and symmetrized skeletons. Extracted symmetric endpoint pairs are linked by green lines.*

mance, and present possible applications including sketch enhancement, extrinsic symmetrization, 3D model consolidation, and the extraction of symmetrized characters from images. Robustness of our algorithm is evaluated via stress tests and comparisons with existing methods.

Sketch enhancement. Figure 7 shows some results of skeleton-intrinsic symmetrization of 2D figures. In the top row, we show how the aesthetics of a garland figure be enhanced by symmetrization along the detected backbone. The remaining three examples show how skeleton-intrinsic symmetrization can be applied to enhance a rough 2D sketch. A rough sketch of Mickey's left hand causes a topological inconsistency between the two sides (near hands) of the skeleton. Symmetrization by copying replaces the rough portion by an enhanced version from the other side. For the dancer and the rabbit, symmetrization by averaging effectively smoothes the roughness in the original sketch.

Parameters and statistics. All skeletons of 3D shapes are extracted with default parameters provided in published works. For skeleton extraction from 2D shapes, we interactively tune the parameters to remove minor branches. After choosing the parameters for a 2D shape, we keep them the same during stress tests (see Table 2). For all the examples shown except for two, we use the following default parameter setting: $\{\sigma_b, \sigma_l, \sigma_s, \sigma_d, \sigma_p, s, g, W_L\} = \{0.04, 0.3, \gamma, 0.2, 0.02, 20, 20, 0.6\}$ and the parameter γ is defined in Section 4.1.

Shape	#V	#E	#P	t_0	t_1
Figure 1(top)	262	5	28	<1s	2s
Figure 2	583	8	91	<1s	3s
Figure 6	620	18	496	5s	3s
Figure 7(a)	975	32	1540	113s	12s
Figure 10	3854	5	28	<1s	31s
Figure 11(top)	7905	8	91	<1s	41s
Figure 12(top)	10002	7	66	<1s	63s
Figure 12(bottom)	5673	11	171	<1s	31s

Table 1: *Statistics and timing for skeleton-intrinsic symmetrization: #V denotes the number of input shape vertices; #E denotes the number of skeleton end nodes; #P gives the total number of backbone candidates; t_0 and t_1 are the computation times (in seconds) for the backbone detection and backbone-guided symmetrization, respectively.*

The first five parameters are used in graph matching. The only tuned parameter is σ_d, which indicates tolerance on the difference between skeleton lengths. For the result in Figure 4(left), a different value $\sigma_d = 0.3$ was chosen. The right embedded figure shows the inferior result of backbone detection when using the default set of parameters. Another exception is the result in Figure 12(bottom), where we set $\sigma_d = 0.1$. The default parameters s and g, for the genetic algorithm, were chosen according to the most complex example in Figure 4(middle).

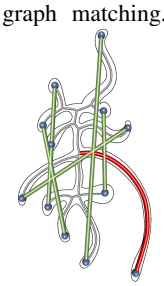


Table 1 provides running times and other statistics for intrinsic symmetrization of several 2D figures and 3D shapes. Timing is measured on an Intel Core I7-2600 machine with 8GB memory, NVIDIA GTX 460 GPU. Processing times are dictated by the number of end points, as we consider pairwise relationship to solve the fitness of each population. If the number of end nodes is too large, e.g., greater than 50, we suggest to quickly filter out small branches, detect the backbone on a simplified skeleton, and then estimate full end point correspondences on the detected backbone.

Comparison to symmetry detection. We compare our symmetry extraction scheme with two state-of-the-art methods for intrinsic global reflectional symmetry detection, first on asymmetric models and then on our intrinsically symmetrized output as a sanity check for our algorithm. The method of Ovsjanikov et al. [OSG08] operates on surface representations; we adapt it to 2D shapes using the interior mesh and apply the ensuing spectral analysis. The method

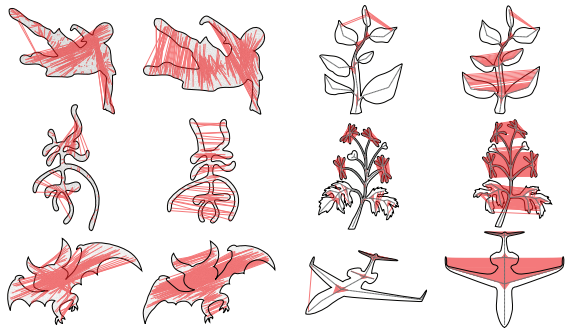


Figure 8: *Comparison with Ovsjanikov et al. [OSG08] (a) and Jiang et al. [JXCZ13] (b) on symmetry extraction. Their methods were unsuccessful on asymmetric inputs, shown in the left columns of (a) and (b). However, they both succeeded on the symmetrized outputs from our algorithm; see right columns of (a) and (b). Only a subset of the (interior) shape correspondences is shown in (a) to avoid over-cluttering. The full skeletal correspondences are shown in (b).*

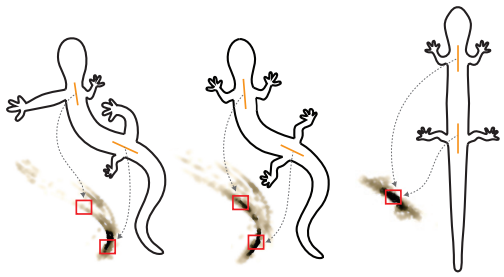


Figure 9: *An (intrinsically) asymmetric shape cannot be extrinsically symmetrized directly since the pattern of peaks in the transformation space do not adequately reveal the approximate symmetries, as shown on the left. Embedded plots of transformation spaces, following [MGP07], are shown on the side. The shape can be first intrinsically symmetrized by our method (middle) and then straightened to possess extrinsic symmetry (right).*

of Jiang et al. [JXCZ13] takes curve skeletons as input and is the most closely related to our method.

Figure 8 evidently shows that these two methods are unable to infer approximate symmetries in an intrinsically asymmetric shape. The spectral embeddings employed by Ovsjanikov et al. are likely to be more distorted away from approximate symmetry in the lower-dimensional embedding space. The method of Jiang et al. appears to settle for local symmetries that it can find, but not global ones, since the associated evidence for the latter is too weak. On the other hand, our method is able to extract the appropriate

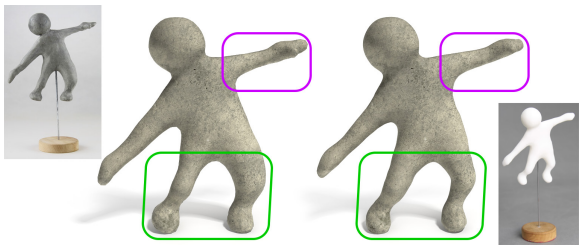


Figure 10: *A clay model (left inset) was created by a novice. We scanned it in, reconstructed it (left), and symmetrized it using our algorithm (right), and then 3D printed the final project (right inset). Compare the arms and legs highlighted in the boxes to see restoration of symmetries.*

backbones for all six shapes and symmetrize them; see right columns of (a) and (b). Then we apply the methods of Ovsjanikov et al. and Jiang et al. to these symmetrized shapes. With the restored intrinsic symmetries, both methods succeeded, providing validation and motivation for our work.

Extrinsic symmetrization. Perhaps an obvious application of skeleton-intrinsic symmetrization is that it provides the proper input for extrinsic symmetrization [MGP07]. Direct extrinsic symmetrization on an asymmetric shape, such as the ones shown in the left column of Figure 9, would likely not succeed since the symmetry-revealing transforms can be rather spread out in the transformation space. Our algorithm is able to detect the right backbone and the ensuing intrinsic symmetrization (middle) allows the shape to be extrinsically symmetrized; see the results in the right column. Both symmetrizations can be performed on the curve skeleton, with skeleton deformation followed by shape symmetrization. Figure 1 shows another similar example.

3D modeling consolidation. When novice users model a 3D shape, either physically or digitally, it is an intricate task to ensure that the modeling is performed to certain precision so as to ensure the (expected) symmetry of the final model. Figure 10 shows a clay model physically created by a novice, with effort. We scanned it in and it can be seen that despite the effort, there are subtle imperfections in the form of asymmetries of the limbs. Applying our symmetrization to the model fixes these problems so that an enhanced digital model can be used. The model can also be 3D printed, if possible in clay, to "enhance a physical creation".

Figure 11 shows how intrinsic symmetrization can be a valuable addition to the 3D modeling toolbox. A human model is to be endowed with a scorpion's claws and a dragon's feet via composition. As the composition was performed quickly by an artist, the arm is slightly misplaced and its length does not match the other side. Automatic intrinsic symmetrization can be employed to consolidate the result and restore the symmetry. The modeling process can be further simplified, thus requiring less effort from the artist, where only

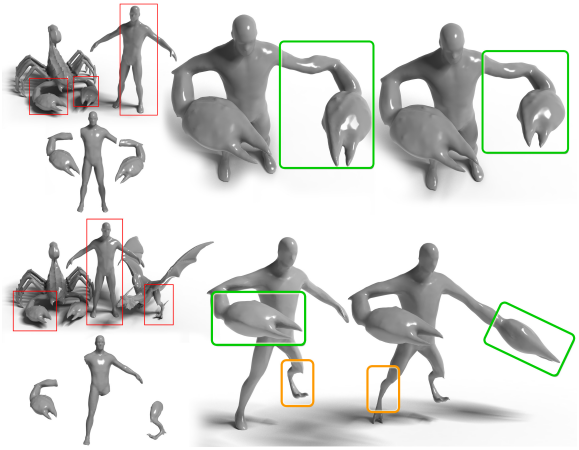


Figure 11: *A human model to be composed with a scorpion's claws or a dragon's feet. Top row: A rough composition which violates intrinsic symmetries is corrected by symmetrization (right). Bottom row: Compositions only done to one side (right hand and left foot) with a claw and a dragon's foot. Our algorithm detects the backbone and symmetrizes the 3D model by copying (right).*

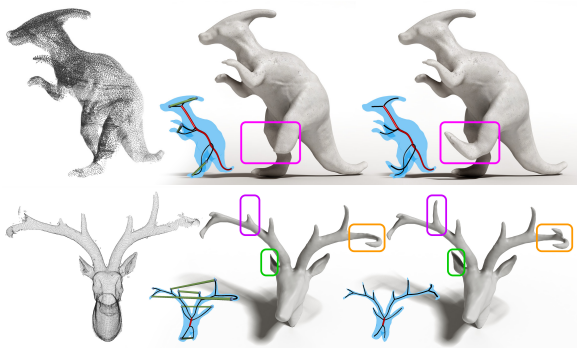


Figure 12: *Symmetrizing 3D reconstructions amid missing data. Reconstructions (middle) of the dinosaur and the deer head models from incomplete scans (left) are asymmetric. Symmetrization restores the missing parts (right).*

one claw and foot needs to be composed. Our symmetrization scheme is still able to detect the correct backbone and through averaging or copying, synthesize the other side.

Finally, symmetrization can be applied to a reconstructed model from incomplete point scans. The purpose again is to restore the symmetry of a model reconstruction which became asymmetric due to missing data; see Figure 12 for two examples. The dinosaur model also demonstrates our algorithm's ability to deal with curve skeletons embedded in 3D that are far from planar.

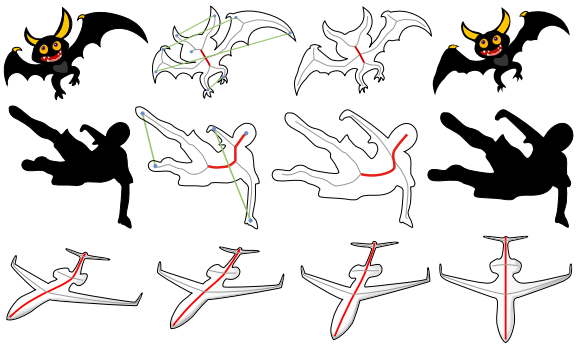


Figure 13: *Symmetrization of 2D shapes under strong perspective. In the first two rows, the left two images show input, one with skeleton and detected backbone. The right two images show the symmetrized results. Last row shows a series of gradually symmetrized (both intrinsically and then extrinsically) airplanes, providing the illusion of a "rotation".*

Shape symmetrization in image. Figures of characters or creatures in images under strong perspective offer another source of intrinsically asymmetric shapes. By applying our intrinsic symmetrization scheme, the strong perspective can be weakened or removed; see Figure 13 for a few such results. This offers two possible utilities. First, when a rough 3D shape needs to be extracted from a 2D contour, e.g., for model-driven image manipulation [ZFL*10], a perspective-corrected 2D input will likely make the task easier. Moreover, perhaps a side effect of our work, a series of gradually symmetrized 2D shapes, through both intrinsic and extrinsic symmetrizations, resemble a sequence resulting from rotation, as illustrated in the last row of Figure 13 for an airplane.

Stress tests. To stress test our backbone detection scheme, we impose perturbations to an input shape or its skeleton through stretching a skeleton branch, displacing a skeleton joint (i.e., moving a branch), and adding Gaussian noise to shape boundaries which may introduce topological noise to skeleton. We vary the amount of perturbations to a fairly large degree (see below), to make the shapes deviate significantly from its original (approximate) intrinsic symmetries.

Specifically, for stretching perturbation on a skeleton branch, we set the stretching range to be $[1.2^{-5}, 1.2^{5}]$ times the original branch length. For joint displacement, we take the range of movement of a skeleton joint to be $\pm 10\%$ of the total length of the skeleton, about its original position. The Gaussian noise added to the shape boundary is up to 1% of the diagonal length of the shape's bounding box. Within the range of each perturbation, we uniformly sample 60 shapes/skeletons, over which we run our test. A test succeeds only when the detected backbone coincides with the one detected for the original shape and skeleton. Figure 14 shows both successful and failure cases for each type of perturbation. Under significantly high level of perturbations, our

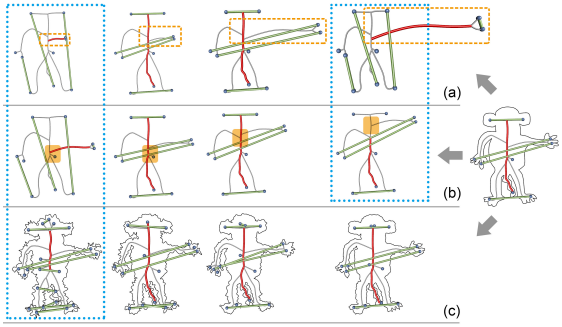


Figure 14: *A few visual results from our stress test of backbone (in red) detection, with failure cases shown in blue boxes. (a) Stretching of a skeleton branch (an arm of the figure) where the stretch factors are* 1.2^{-5}, 1.2^{-3}, 1.2^{3}, 1.2^{5}, *from left to right. (b) Movement of a skeleton branch (the same arm) where the displacements are* $0.06, 0.03, -0.07, -0.14$. *(c) Standard deviations of Gaussian noise on the boundary are* $0.01, 0.008, 0.006, 0.004$. *The figure on the rightmost shows the original input, its skeleton, and the backbone detected. We set the perturbation ranges for Table 2 based on failure cases on this example.*

Shape	Stretch	Displace	Noise
Figure 1(top)	78.3%	90%	95%
Figure 1(bottom)	70%	80%	81.3%
Figure 2, 3	86.7%	83.3%	73.3%
Figure 4(left)	80%	78.3%	76.7%
Figure 4(middle)	83.3%	88.3%	55%
Figure 4(right), 5, 14	93.3%	81.6%	90%
Figure 6	83.3%	91.6%	68.3%
Figure 7(a)	63.3%	73.3%	45%
Figure 7(b)	91.7%	88.3%	83.3%
Figure 7(c)	90.0%	91.6%	100%
Figure 9	80%	93.3%	95%
Figure 10	88.3%	85%	100%
Figure 11	93.3%	95%	100%
Figure 12(top)	86.7%	93.3%	100%
Figure 12(bottom)	73.3%	86.7%	100%
Figure 13(top)	91.6%	86.6%	88.3%
Figure 13(middle)	78.3%	75%	100%
Figure 13(bottom)	91.6%	93.3%	100%

Table 2: *Success rates from our stress tests on backbone detection for* all *examples shown in the paper, under varying levels of perturbation. The three kinds of perturbations are: stretching a skeleton branch, displacing a skeleton joint, and adding Gaussian noise to shape boundaries.*

algorithm does break down, however, its robustness is well demonstrated from the stress test. Table 2 reports the success rates for all the 2D and 3D examples shown in this paper. The average success rate for all examples is 85.3%.

7. Conclusion, limitation, and future work

We develop an algorithm for skeleton-driven and skeleton-intrinsic symmetrization of shapes. The key technical contribution is a scheme for extracting weak reflection symmetries from an intrinsically asymmetric shape. Our backbone extraction algorithm is shown to be robust and more capable of extracting approximate symmetries than existing methods, which have been designed to detect apparent symmetries. We demonstrate applications of skeleton-intrinsic symmetrization to sketch enhancement, 3D model consolidation, and symmetrized shape extraction from images.

In many of our examples and experiments, we deliberately present input shapes with moderate to significant intrinsic asymmetry. This is primarily for showing the robustness of backbone extraction and symmetrization scheme. Having said that, in Figures 7 and 10, for example, we also demonstrate subtle enhancements enabled by our method. While subtle, executing such transformations by hand is still tedious. An automatic scheme such as ours is desirable.

Limitations. Our symmetrization algorithm is designed to operate on boundary shape presentations in 2D or 3D. However, the input shape does not need to be watertight, as we demonstrate in Figure 12, so long as a reasonable curve skeleton can be extracted using the methods we employ. Like all skeleton-driven techniques, our skeleton-intrinsic symmetrization algorithm is only applicable to shapes that have appropriate curve skeleton abstractions, e.g., articulated figures. It does not handle objects such as a baseball hat or tea cup. Furthermore, our backbone optimization scheme does not deal with backbones that contain loops or topological mismatches due to loops in the skeleton graphs.

Our analysis is also restricted to reflectional symmetries, as it explicitly considers a reflective self-mapping on the curve skeleton. The symmetrization scheme is driven by the backbone only, thus the multi-scale nature of shape symmetries [XZJ*12] is not accounted for. Last but not the least, as a purely geometry based approach, our algorithm is unable to extract or enhance the expected symmetry if the associated evidence is too weak.

Future work. We would like to extend our backbone extraction to handle "intrinsic rotational symmetries" over a skeletal structure, instead of only having a reflection measure. An extension to multi-scale intrinsic symmetrization, as well as to co-symmetrization of a set of related shapes, would both be interesting. Rather than completely dependent upon extracted curve skeletons, a hybrid approach that considers both boundary representations and perhaps intermediate skeletal representations would allow us to sidestep certain limitations to existing skeleton extraction schemes. Finally, a natural question which will likely lead to interesting future work is how to asymmetrize a shape or pattern to beautify it, where the asymmetrization leads to neither extrinsic nor intrinsic symmetry.

Acknowledgments

The authors would like to thank all the reviewers for their valuable comments and feedback. This work is supported in part by grants from NSFC (61202224, 61331018, 61272327, 61202333), National 973 Program (2015CB352501), National 863 Program (2013AA01A604), Shenzhen Innovation Program (CXB201104220029A, KQCX20120807104901791, ZD201111080115A, JSG-G20130624154940238), CPSF China (2012M520392), NSERC (611370) and the Israel Science Foundation.

References

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 27*, 3 (2008). 3

[ATCO*10] AU O. K.-C., TAI C.-L., COHEN-OR D., ZHENG Y., FU H.: Electors voting for fast automatic shape correspondence. *Computer Graphics Forum 29* (2010), 645–654. 2, 4

[BL08] BAI X., LATECKI L. J.: Path similarity skeleton graph matching. *IEEE Trans. Pattern Analysis & Machine Intelligence 30*, 7 (2008), 1282–92. 2, 4

[BTST12] BHARAJ G., THORMÄHLEN T., SEIDEL H.-P., THEOBALT C.: Automatically rigging multi-component characters. *Computer Graphics Forum (Proc. of Eurographics) 31* (2012), 755–764. 4

[CMS07] CORNEA N. D., MIN P., SILVER D.: Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Visualization & Computer Graphics 13*, 3 (2007), 530–548. 2

[HWCO*13] HUANG H., WU S., COHEN-OR D., GONG M., ZHANG H., LI G., CHEN B.: ℓ_1-medial skeleton of point cloud. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 32*, 4 (2013), 65:1–65:8. 2, 3

[JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 30*, 6 (2011), 165:1–165:8. 6

[JXCZ13] JIANG W., XU K., CHENG Z.-Q., ZHANG H.: Skeleton-based intrinsic symmetry detection on point clouds. *Graphical Models 75*, 4 (2013), 177–188. 2, 9

[KLCF10] KIM V. G., LIPMAN Y., CHEN X., FUNKHOUSER T.: Möbius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum (Proc. Eurographics Symp. on Geometry Processing) 29*, 5 (2010), 1689–1700. 2

[LHOKG10] LIU Y., HEL-OR H., KAPLAN C. S., GOOL L. J. V.: *Computational Symmetry in Computer Vision and Computer Graphics*, vol. 5 of *Foundations and Trends in Computer Graphics and Vision*. 2010, pp. 1–195. 2

[LJ07] LING H., JACOBS D.: Shape classification using the inner-distance. *IEEE Trans. Pattern Analysis & Machine Intelligence 29*, 2 (2007), 286–299. 3

[LKF12] LIU T., KIM V. G., FUNKHOUSER T.: Finding surface correspondences using symmetry axis curves. *Computer Graphics Forum 31*, 5 (2012), 1607–1616. 2

[LYFD12] LU J., YU F., FINKELSTEIN A., DIVERDI S.: Helpinghand: example-based stroke stylization. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 31*, 4 (2012), 46:1–10. 3

[McM05] MCMANUS I. C.: Symmetry and asymmetry in aesthetics and the arts. *European Review 13* (2005), 157–180. 1

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential-geometry operators for triangulated 2-manifolds. 7

[MGP07] MITRA N. J., GUIBAS L. J., PAULY M.: Symmetrization. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 26*, 3 (2007), 63:1–63:8. 2, 3, 9

[MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D.: Symmetry in 3D geometry: Extraction and applications. In *Proc. of Eurographics STAR Report* (2012). 2

[OK11] ORBAY G., KARA L. B.: Beautification of design sketches using trainable stroke clustering and curve fitting. *IEEE Trans. Visualization & Computer Graphics 17*, 5 (2011), 694–708. 3

[OSG08] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. *Computer Graphics Forum (Proc. Eurographics Symp. on Geometry Processing) 27*, 5 (2008), 1341–1348. 2, 8, 9

[Ram72] RAMER U.: An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing 1*, 3 (1972), 244–256. 6

[RBBK10] RAVIV D., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Full and partial symmetries of non-rigid shapes. *Int. J. Computer Vision. 89* (2010), 18–39. 2

[RvWT08] RENIERS D., VAN WIJK J., TELEA A.: Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Trans. Visualization & Computer Graphics 14*, 2 (2008), 355–368. 2, 3

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proc. Eurographics Symp. on Geometry Processing* (2004), pp. 175–184. 7

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer 24*, 4 (2008), 249–259. 5

[Szw05] SZWOCH W.: A spectral technique for correspondence problems using pairwise constraints. In *Proc. Int. Conf. on Computer Vision* (2005), pp. 1482–1489. 7

[TZCO09] TAGLIASACCHI A., ZHANG H., COHEN-OR D.: Curve skeleton extraction from incomplete point cloud. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 28*, 3 (2009), 71:1–71:9. 2

[VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 28*, 3 (2008). 7

[Wey83] WEYL H.: *Symmetry*. Princeton University Press, 1983. 1

[XZJ*12] XU K., ZHANG H., JIANG W., DYER R., CHENG Z., LIU L., CHEN B.: Multi-scale partial intrinsic symmetry detection. *ACM Trans. on Graphics 31*, 6 (2012), 1–10. 2, 5, 11

[XZT*09] XU K., ZHANG H., TAGLIASACCHI A., LIU L., LI G., MENG M., XIONG Y.: Partial intrinsic reflectional symmetry of 3D shapes. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 28*, 5 (2009), 138:1–138:10. 2, 3

[ZDIT12] ZHOU F., DE LA TORRE F.: Factorized graph matching. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition* (2012), pp. 127–134. 5

[ZFL*10] ZHOU S., FU H., LIU L., COHEN-OR D., HAN X.: Parametric reshaping of human bodies in images. *ACM Trans. on Graphics 29*, 4 (2010), 126:1–126:10. 10

[Zit13] ZITNICK C. L.: Handwriting beautification using token means. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 32*, 4 (2013), 53:1–8. 3