Pairwise Harmonics for Shape Analysis

Youyi Zheng Chiew-Lan Tai Eugene Zhang Pengfei Xu

Abstract—This paper introduces a simple yet effective shape analysis mechanism for geometry processing. Unlike traditional shape analysis techniques which compute descriptors per surface point up to certain neighborhoods, we introduce a shape analysis framework in which the descriptors are based on pairs of surface points. Such a *pairwise* analysis approach leads to a new class of shape descriptors that are more global, discriminative and can effectively capture the variations in the underlying geometry. Specifically, we introduce new shape descriptors based on the isocurves of harmonic functions whose global maximum and minimum occur at the point pair. We show that these shape descriptors can infer shape structures and consistently lead to simpler and more efficient algorithms than the state-of-the-art methods for three applications: intrinsic reflectional symmetry axis computation, matching shape extremities, and simultaneous surface segmentation and skeletonization.

Index Terms—Shape Analysis, Pairwise Harmonics, Intrinsic Symmetry, Shape Correspondence, Segmentation and Skeletonization.

1 Introduction

Shape analysis is essential in a wide range of geometry processing applications such as content-based shape matching and retrieval, part recognition, and shape editing [1], [2]. An effective, numerically robust, and computationally tractable shape descriptor is key in shape analysis. Most existing shape descriptors, such as the curvature map [3], mesh saliency [4], and spin images [5], are defined in terms of shape features extracted in local surface regions. Consequently, they are often not sufficient to abstract more global structures in the shape, such as global symmetry.

In recent years, more global shape descriptors have been introduced. The Shape Diameter Function (SDF) [6], for instance, is a scalar quantity based on the local thickness of the shape around each measurement point and is robust against noise and pose variations. A number of approaches [7], [8], [9] exploit the spectrum of the eigenspace of heat diffusion operators to define signatures around each surface point which capture isometrics. The intrinsic maps [10], [11], [12] define surface embedding in low-dimensional conformal spaces where the search for correspondence mappings becomes more tractable. These descriptors defined for each surface point are either expensive to compute or are designed for specific applications (e.g., shape correspondence, symmetry computation), therefore are not easy to generalize directly.

We introduce *pairwise harmonics*, in the context of manifold surfaces, for generic shape processing at global scales. At the core of our approach is the notion of *pairwise analysis* (Section 2). Unlike previous approaches which mostly focus on processing the whole surface in a point-wise manner, our approach analyzes the shape in a *pairwise* manner and computes shape descriptors for pairs of surface points. Specifically, the shape descriptors are defined in terms of properties of isocurves uniformly sampled from pairwise harmonic fields.

Defining descriptors for a point pair offers several benefits over traditional descriptors defined in terms of one point. First, pairwise descriptors can provide not only insights about the geometry between the point pair, but also their relationship. This makes the analysis framework more global and discriminative in revealing shape structures, such as symmetry. Second, pairwise analysis constrains the measurements over point pairs (in our case along a local geodesic path), and thus can reveal shape structures at different scales for different point pairs. It naturally encapsules multi-scale analysis into a single framework with no additional cost. Further, exploring shape geometry in both global and local scales can tolerate partially missing information when matching different shape parts. Third, pairwise descriptors often overlap (see the isocurves on right legs of the Armadillo in Figure 1), making a sparse set of pairwise descriptors sufficient in representing the underlying geometry. This property greatly decreases the computation complexity for applications that require examining global shape properties (e.g., part recognition, skeletionization, shape matching, etc).

Inferring inter-shape correspondence and understanding the symmetry structure are known to be challenging problems due to the nature of the computation difficulties in recovering such high-level shape information from the geometry alone. Hence, the state-of-the-art approaches for computing the symmetry and correspondence often employ rather complicated and computationally expensive techniques based on statistics [13], [14], heat diffusion [7], [8], [9] and conformal maps [10],

- *Y. Zheng is with the Geometric Modeling and Scientific Visualization Center, King Abdullah University of Science and Technology. E-mail: {youyi.zheng}@kaust.edu.sa*
- *C.-L. Tai and P. Xu are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. E-mail: {taicl, xupengfei}@cse.ust.hk*
- *E. Zhang is with the School of Electrical Engineering and Computer Science, Oregon State University.E-mail: {zhange}@eecs.oregonstate.edu*

[11], [12], etc. These sophisticated mechanisms are often costly in computation. Recently, Xu et al. [14] discovers intrinsic reflectional symmetry axis by narrowing down the search space through a series of tests. However, their approach can still take up to hours to compute the symmetry axis. We show that the pairwise harmonics can lead to simpler and more efficient solutions for these problems (Section 3.1, 3.2).

We demonstrate the effectiveness of the *pairwise harmonics* with three applications: intrinsic reflectional symmetry axis computation, shape correspondence, and simultaneous skeletonization and segmentation. We compare the results with the state-of-the-art techniques and show that the pairwise harmonics enable simpler and more efficient algorithms. Additionally, focusing on computing the pairwise global properties of the shape can lead to algorithms that are more robust towards local surface noise, pose variations and shape topology.

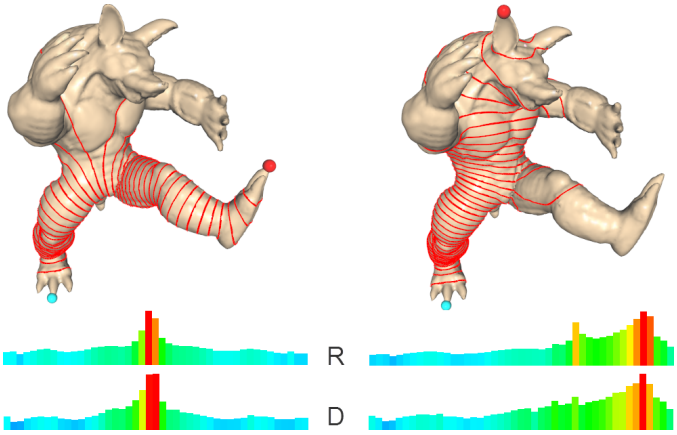


Fig. 1. Two examples of pairwise harmonics and the histogram visualization of the extracted shape descriptors: the perimeter descriptor R (top histograms) and the distance descriptor D (bottom histograms). Notice the near symmetry of both distributions if the two points are symmetry (left).

2 Pairwise Harmonics

Previous Work on Shape Descriptors. *Shape descriptors* (or *shape signatures*) have been extensively researched in the context of shape retrieval and shape matching [1]. Shape descriptors provide informative abstractions of the surface contents and offer effective means for similarity matching. Different descriptors reveal different shape information, e.g., the curvature map described the curvature distribution and the average geodesic distance reveals the protrusions in the surface [15], and surface visibility helps interior and exterior classification [16].

Shape descriptors are often defined in terms of feature vectors or histograms (e.g., the spin image [5]). Graph-based descriptors [17] or statistical-based descriptors [18] have also been employed. There exists a wealth of shape descriptors in the domain of shape retrieval and matching; we refer readers to [1], [19] for reviews on the state-of-the-art shape descriptors, and to [7], [8], [9], [20] for the more recently proposed descriptors based on the spectrum of Laplace-Beltrami operators.

Recently, Tevs et al. [21] presented a geodesic-fan-like [22] shape descriptor for surface matching. The descriptor is defined as the distribution of the lengths of equally distanced geodesic rings propagating from a single point. They show that the descriptor has strong discriminating power in dissimilarity measurements among points, and employ a landmark-based algorithm to find surface correspondences. Our descriptor resembles their form in the sense that it is also defined on properties of propagating isocurves. However, to our best knowledge, there have been no previous descriptors proposed based on harmonic fields of point pairs.

Shape descriptors from pairwise harmonics. *Harmonic* function is a well-established mathematical concept and has been widely applied in geometry processing [23], [24], [25] due to its desirable hamonicity properties, such as smoothness and local extrema coinciding with the singularities in the harmonic field. We exploit the harmonic funtions coupling with a pairwise analysis mechanism for geometry analysis. The shape descriptors are defined on isocurves of harmonic functions for point pairs. To our knowledge, this is the first time harmonic field is examined in such a manner, which can resolve the complexity of existing shape analysis algorithms.

We now formally describe the pairwise harmonics. On a mesh surface, a harmonic function f defines a scalar field that satisfies the Laplace equation $\Delta f = 0$, subject to certain Dirichlet boundary constraints. Given a pair of points (p, q) on the mesh surface, the pairwise harmonic for (p, q), which we denote as $f_{p,q}$, is defined as the harmonic field with Dirichlet boundary constraints $f(p) = 0$ and $f(q) = 1$. The discretization leads to a sparse linear system $Lf = b$, with L denoting the Laplacian matrix with the cotangent weights [26] (we weight each vertex i with its one-ring voronoi area). We use the cotangent-weight because it is simple, approximates well a pure harmonicity [27], and has a set of desirable properties including inherent smoothness, iso-metric invariance and insensitive to noise and tessellation (see Figure 2).

Notice that such functions have a range between 0 and 1. We consider the levelsets of this function on the surface. Close to p and q, the levelsets can be approximated by disks on the surface. Away from p and q, the geometry and topology of the levelsets can become more complex (e.g., multiple connected components, critical points). This motivates us to quantify similarity measurements using the properties of the function levelsets, in particular the length of the isocurves and their average distance to the end points p and q, as our shape descriptors.

We uniformly sample K isocurves between the interval $[0, 1]$ (see Figure 1). This set of isocurves is the key ingredient of our shape analysis algorithms. From the isocurves we extract two types of descriptors and demonstrate their use in different applications.

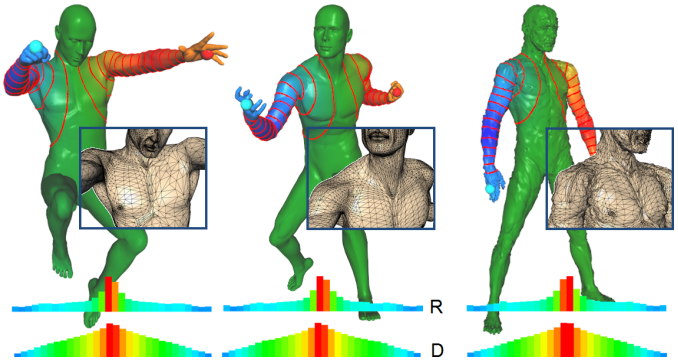


Fig. 2. A pairwise harmonic field defined by two points (one on each hand) and the uniformly sampled isocurves. Observe that the perimeter descriptor R and the distance descriptor D are insensitive to pose variation, tessellation, and surface noise.

The *perimeter descriptor* is defined as the distribution of the length of the isocurves between the pair (p, q):

$$R_{pq} = \{r^1_{pq}, r^2_{pq}, ..., r^K_{pq}\} \quad (1)$$

where r^i_{pq} is the length of the ith isocurve. Figure 1 shows two examples of the perimeter distribution. This distribution reveals the shape variation between the point pair, hence facilitates shape matching. The perimeter descriptor also captures symmetry of shape. Indeed, if the local geometry near the point pairs are similar, the perimeter distribution of the isocurves will demonstrate similarity (Figure 1 left).

We define another descriptor, called *distance descriptor*, which captures shape information orthogonal to the perimeter descriptor:

$$D_{pq} = \{d^1_{pq}, d^2_{pq}, ..., d^K_{pq}\} \quad (2)$$

where d^i_{pq} is the average geodesic distance from points on the isocurves to either p (for first half of isocurves) or q (for second half of isocurves); i.e., $d^i_{pq} = geo(i, s)$ where $s = p$ if $i < K/2$ and $s = q$ otherwise. This distance can be viewed as horizontal geodesic information along the medial axis connecting centers of isocurves. Figure 1 also shows two examples of the distance distribution.

We will show that the two descriptors can reveal symmetries of the shape between a point pair. Intuitively, if two points merit an intrinsic symmetry, then both descriptors exhibit nearly symmetric profiles. Specifically, the distribution histograms of both descriptors are symmetric if the two points (p, q) are geometrically symmetric, i.e., $r^i_{p,q} = r^{K-i}_{q,p}$ and $d^i_{p,q} = d^{K-i}_{q,p}$ for any $0 \leq i < K$. In Section 3.1 we combine the two descriptors to define a *path intrinsic symmetry* measure which is capable of capturing both symmetry and partial symmetry of shapes.

Thanks to the inherent smoothness of the cotangent-weighted harmonic fields, the computed pairwise descriptors are insensitive to surface noise and isometric deformation (Figure 2). These desirable properties lead to robust algorithms we shall present in the next sections.

3 Applications

We now show the practical benefits of our pairwise harmonics and and their corresponding shape descriptors with three applications.

Updating a pairwise harmonic is super-linear [28], however, computing the harmonic for every surface point pair would still be computational prohibitive. Therefore, surface sampling to identify a sufficient small number of good pairwise harmonics for specific applications is essential in our solutions. Our shape descriptors are defined on the diffusion path constrained by two points. Consequently, the isocurves extracted from different fields constrained by different point pairs can overlap with each other, (e.g., in Figure 1, the isocurves from one leg to another leg overlap with those from one leg to an ear), leading to the descriptors having redundancy geometry infromation. This characteristic allows sampling of only sparsely distributed points for applications like shape matching and skeletonization where isocurves tautly encircling the surface regions suffice. For symmetry axis computation, more samples are required to identify and validate possible candidate symmetry pairs.

3.1 Intrinsic Reflectional Symmetry Axis

Previous Work on Symmetry. Symmetry is one of the most fundamental phenomena in nature and man-made objects. It is also a crucial element for human recognition in many domains, such as arts, biology, mechanical design, and architecture [29], [30]. In computer graphics, symmetry has also been well examined and has been used in shape processing applications like remeshing [31], beautification [32] and shape editing [2].

Previous approaches on symmetry computation can be divided into two classes. One class focuses on computing the explicit symmetry, especially the Euclidean reflectional and translational symmetries [33], [34] or parts symmetry under rigid transformations [13]. The other class focuses on computing intrinsic shape symmetries (isometric homeomorphism between the shape surface and itself, non-rigid and possibly only partially) [35], [11], [8], [12].

Recently, algorithms have been proposed to find structures that characterize intrinsic symmetries [36], [37]. We also focus on computing such symmetry structures without explicitly establishing the point-to-point symmetries. We show that the *pairwise harmonics* can be naturally exploited to compute intrinsic reflectional symmetry axis as of Xu el al. [36]. In their method, a series of geometric filters were designed to first filter out redundant pairs and then a symmetry-support voting mechanism is employed to summarize the symmetry on mesh points and vote for the final symmetry axes. Promising results were obtained, however it usually takes substantial time (up to hours) to compute intrinsic symmetries. As reported in their paper, the complexity of their algorithm is $O(n^4)$ where n is the sample size (typically $2k - 10k$).

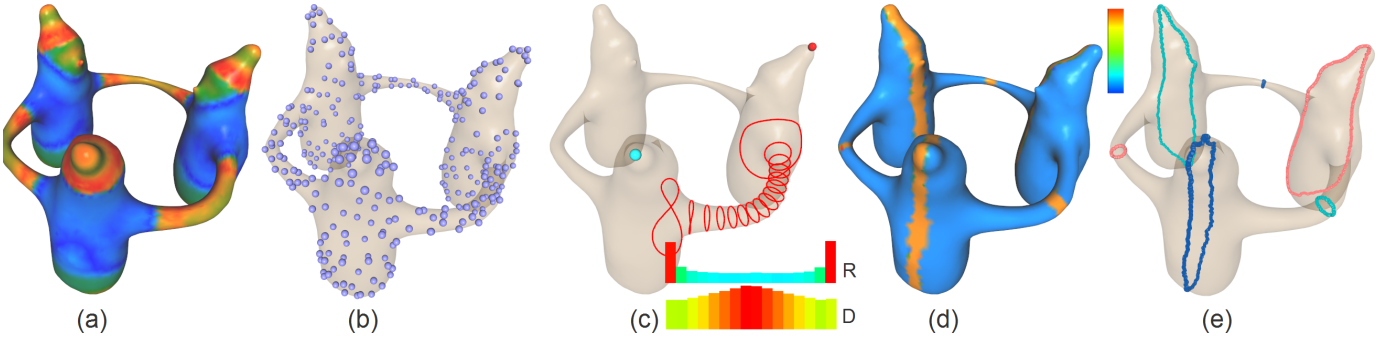


Fig. 3. The main flow of our intrinsic symmetry axis computation algorithm. Based on the vertex geodesic entropy [21] (a), we sample the surface (b) and define a pairwise harmonic for each valid point pair (c). We measure the symmetry of each point pair and let it vote for the symmetry axes. Votes are accumulated at surface faces (d); observe that those lying in the symmetry regions receive large amounts of votes (d). Final symmetry axes (three here) are extracted based on the votes (e).

Our approach. Since the descriptors extracted from *pairwise harmonics* contain intrinsic symmetry information (Figure 1 left), our main challenge here is to identify good candidate point pairs efficiently to reduce the search space.

Exhaustive search over the entire model for possible good symmetry point pairs proves to be prohibitive given its $O(n^4 \log n)$ complexity. Hence, surface sampling is essential in reducing the search space at the outset. A naïve sampling strategy may lead to potentially good candidates being missed out or too many error pairs introduced in the final symmetry computation. Our goal is to reduce the amount of computation during this search while still ensuring that prominent symmetry pairs can be identified. We leverage the fact that if a point pair is found to merit good intrinsic symmetry, the region along the field propagation path is likely to contain many good symmetry pairs. Hence, if a pair with good symmetry is found, the symmetry of the region along the path is well represented which significantly reduces the samples needed to validate the established symmetry as required in [14], allowing us to devise a simpler and more efficient method.

To benefit our goal of sampling potentially good symmetry pairs, we adopt a sampling strategy that utilizes the geodesic entropy proposed by Tevs et al. [21]. They define a likelihood of point i being equal to point j as follows:

$$\eta_{ij} = e^{-\frac{\|g_i - g_j\|^2}{2\sigma^2}} \tag{3}$$

where $\|.\|$ denotes the L^2 distance, σ is the standard derivation and g_i is the geodesic descriptor of point i defined as the lengths of the geodesic rings propagating from point i at equal distances. We do not normalize g_i and g_j as did in [21] because we wish to generate as many distinct points as possible. For example, points on a hand have profiles of geodesic descriptors that are rather different from points on a foot, even though as limbs they have similar appearances. This observation leads to a probability function for each point. The geodesic entropy of a point i is defined as the entropy of the probability function in equation 3 [21]. Figure 3 (a) shows such an entropy map for the Snowman model. The geodesic descriptors possess the power of clearly distinguishing regions of different semantics (see the distinct regions (warm-colored) in Figure 3 (a)), leading to potential symmetry regions having similar colors (note the similar colors in symmetry parts).

We then employ a sampling strategy that strives to cover all prominent regions of the mesh while encouraging the points to span the entire mesh surface. The sampling method iteratively selects the point with the largest geodesic entropy value, while disallowing any point within a specified radius of already selected points. Figure 3 (b) shows the sampled points for the Snowmen model, which are distributed throughout the model and many of them are potential symmetry pairs.

We now describe the pipeline of our algorithm for intrinsic symmetry identification (Figure 3). First, we compute a set of points on the surface as described above, which we denote by Ω. Second, we iterate through all point pairs (p, q) from Ω and reject pairs that we consider uncorrelated using two geometric tests (the details of which will be described later). Denote the set of remaining point pairs by Σ. Next, for each point pair p, q in Λ, we compute a corresponding pairwise harmonic function (order of p and q is not important) and extract a uniformly sampled set of isocurves as described in Section 2. We measure the path intrinsic symmetry (*PIS*) of the point pair, denoted as ς_{pq} (defined below), and check if it is a good candidate of symmetric point pair. Finally, we employ a voting strategy similar to [36] to find the final symmetries. Instead of using surface points [36], we vote to the surface faces since the symmetry axes may not exactly pass through surface points.

The two geometric tests that we use to reject point pairs are similarity test and geodesic distance test. For the similarity test, we use the geodesic descriptors of the two points defined above, i.e., g_i and g_j. We reject a pair of points if their distance $\psi(g_i, g_j)$ is larger than

a threshold ϱ (0.01 in our experiment). $\psi(.)$ is defined as $\psi(g_i, g_j) = \frac{1}{n}\sum_{k=1}^{n} |\frac{g_i^k - g_j^k}{g_i^k + g_j^k}|$ ($n = 16$ is the number of geodesic rings [21]). To omit close point pairs, the geodesic distance test rejects a point pair if their geodesic distance is shorter than $1/5$ of the maximum geodesic distance among any sampled pairs.

We define the path intrinsic symmetry (*PIS*) between a pair of points (p, q), denoted ς_{pq}, as follows:

$$\varsigma_{pq} = \Pi_{j=1}^{K/2} L(r_{pq}^j, r_{pq}^{K+1-j}) L(d_{pq}^j, d_{pq}^{K+1-j}) \quad (4)$$

where L is defined as

$$L(x, y) = 1 + \frac{|x - y|}{x + y}. \quad (5)$$

Such a definition introduces consistent penalty for intrinsic symmetry measurements. The value of ς_{pq} increases dramatically as the number of non-symmetry isocurves rises. For example, if one corresponding pair of the isocurves has a large value of $L(x, y)$, it will lead to an increase in the magnitude of the value of ς_{pq}. For a perfect symmetry point pair (p, q) we have $\varsigma_{pq} = 1$. We reject a point pair as a symmetry candidate if its ς_{pq} value is larger than a user-defined threshold ϵ (2.0 in experiments).

Before voting, we convolve all the computed *PIS* values by a Gaussian function to boost those pairs with strong symmetries. Specifically, we compute a new ς'_{pq} as:

$$\varsigma'_{pq} = e^{-\varsigma_{pq}^2/(2\tau^2)}, \quad (6)$$

where $\tau = \epsilon/2$. For each candidate point pair, we compute its symmetry axis as in [36] and let the point pair vote to the axis curve by sending a score to the faces on the symmetry axis. Each mesh face can receive multiple votes if it is passed by multiple axes. Finally, the symmetry axes are iteratively selected using a greedy algorithm considering the average votes of all faces they pass through. We ignore a new axis curve if it intersects any previous selected ones.

Results and Evaluation. Figure 3 shows a result of our algorithm. Note that the algorithm is able to find intrinsic symmetries using a relatively small set of sample points (we use 300 in this example). This demonstrates the strength of our pairwise descriptors in terms of both effectiveness and computational efficiency. Our algorithm runs at a much faster rate than the method of Xu et al. [14]. The whole process typically takes under 5 minutes for a sample rate of up to 500 vertices for moderate mesh size (30k), and the complexity basically grows quadratically with the sample size. We use the fast Choleskey factorization updating schemes as in [38] to speed up the computation of the harmonic fields.

Figure 4 shows some results from more challenging models. Our algorithm consistently locates symmetry axes in desired regions. For the menmento model, our method finds only one symmetry axis, even though there are three. This is because the three symmetry axes intersect each other and our greedy method simply ignores the other two. The interleaved strategy employed in [36] can solve this problem, but at the cost of a significant time complexity. As our algorithm works well for most models, for simplicity, we did not implement that strategy.



Fig. 4. More examples on partial intrinsic symmetry computation. Our method consistently locates the symmetry axes at desired regions. For the rightmost model, our method finds only one symmetry axis due to intersection with the other symmetry axes (see detailed discussion in Section 3.1). The middle model is coupled with 30% Gaussian noise.

It is interesting to note that only a relatively small set of samples (< 1000) usually suffices. This is because our sampling strategy is effective in locating potential symmetry points (Figure 11). The pairwise harmonics also render the method simpler and more efficient. The symmetry information encoded in the isocurves along the propagation path lends itself to an effective way of measuring shape symmetry within a surface region. Such information serves naturally as symmetry evidence to a symmetry axis, without requiring any localized support voting scheme to validate a found symmetry as done in [36], hence significantly reduces the complexity.

3.2 Matching Shape Extremities

In this section we describe our algorithm that uses the *pairwise harmonics* to establish a correspondence between two sets of shape extremal points on two given shapes. The shapes may have rather different surface details as well as poses and may even be only partially matchable. After establishing a correspondence between the shape extremities, a denser correspondence can be found using the landmark algorithm of Tevs et al. [21].

Previous Work on Shape Correspondence. Finding a correspondence in a pair of models is a well-researched area. In general, shape correspondence is a more difficult problem than shape retrieval, since shape retrieval mainly seeks to measure the similarity between two shapes (e.g., by comparing shape descriptors in the same domains) while shape correspondence addresses the problem of establishing a one-to-one concrete mapping between two sets of shape points. Recently, various methods achieving compelling results have been proposed

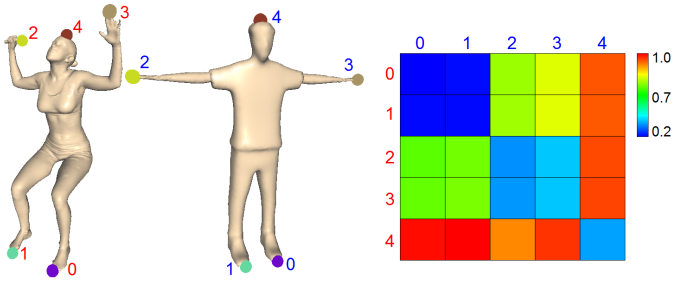


Fig. 5. Visualization of the feature-to-feature distance matrix for the two human shapes. Our pairwise descriptors have a strong distinguishing power to classify these features. For example, points on arms in one model are most similar to points on arms in the other model.

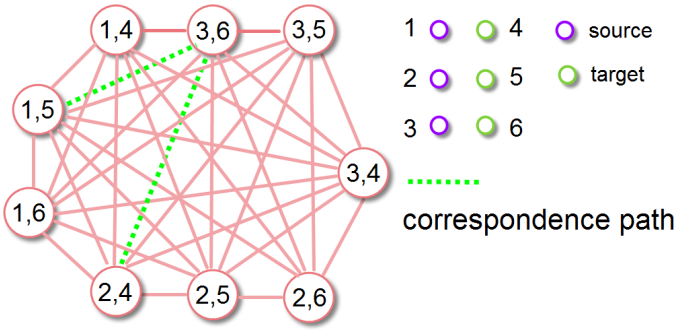


Fig. 6. The graph model for our matching framework. Each pair of points corresponds to a graph node. The green path shows a possible correspondence.

for finding correspondence between isometric shapes, shapes with different surface details and large deformations, or even only partially matchable shapes [10], [12], [39], [40], [41], [42], [43]. These methods commonly leverage low-dimensional spaces (e.g., conformal mappings [12]) or effective representations (e.g., shape skeleton [42]) to reduce exponential searching complexities, and subsequently employ efficient searching algorithms to narrow down the search space while retaining good mapping candidates. Correspondences are established either between a dense subset or complete set of surface points [10], [12], a sparse set of surface extreme points [41], [40], nodes on skeletons [42] or shape parts [43].

Despite the complex matching frameworks, the underlying key ingredients of these matching systems are the descriptors and combinatorial measurements (e.g., transformation cost [10], [41]) to compare the similarities among the elements to be matched. Most of the above-mentioned methods employ rather complicated techniques and are time consuming even when finding a correspondence for dozens of points [10], [41], [12]. The recent approach of Au et al. [42] gains a much faster speed for matching the ending and junction nodes of two skeletons. A search tree is iteratively spanned whose nodes eventually vote for the best correspondence. As a drawback, computing shape skeletons can be an overhead. Further, it often fails for shapes with no well-defined skeletons, such as tables or chairs in Figure 8.

Our Approach. Our method finds correspondence between two sets of shape extreme points [41] (a typical size < 20). Let us denote the two sets of points as $P = \{p_1, p_2, ..., p_m\}$ and $Q = \{q_1, q_2, ..., q_n\}$. Without loss of generality, we assume $m \leq n$.

For each extreme point p_i (q_i), we first compute a pairwise harmonic field from p_i (q_i) to each other extreme point p_j (q_j) and compute the set of descriptors from all these fields, denoted by $H_i = \{..., h_{ij}, ...\}_{j \neq i}$, where h_{ij} is defined as follows:

$$h_{ij} = (R_{ij}, g_{ij}). \tag{7}$$

Here R_{ij} is the perimeter distribution vector in Equation 1 and g_{ij} denotes the normalized geodesic distance from point i to j (normalized to the range $[0, 1]$ by the maximum geodesic distance among all pairs). We then define the point-to-point difference between two point pair $p \in P$ and $q \in Q$, denoted by $D(p, q)$, as the minimum distance between two sets of corresponding descriptors H_p and H_q. This computes the minimum sum of pairwise distances of any injective mappings between the two sets. We solve it using a fast implementation of the Hungarian algorithm [44], where the difference between any two h_{ij}s is defined by the distance metric E:

$$E(h_{ij}, h_{kl}) = \frac{1}{K} \sum_{m=1}^{K} \frac{|r_{ij}^m - r_{kl}^m|}{r_{ij}^m + r_{kl}^m} + \frac{|g^{M_1}(i,j) - g^{M_2}(k,l)|}{g^{M_1}(i,j) + g^{M_2}(k,l)}. \tag{8}$$

Figure 5 (right) visualizes the values of D(.) of of two feature sets as a distance matrix, which clearly illustrates the discriminate power of the pairwise descriptors. In particular, one can see that the arms, legs, and heads are clearly classified (shown in similar blue colors).

Now we consider the correspondence problem. For any mapping between two subsets of P and Q, $f : (i_1, i_2, ..., i_\kappa) \subseteq P \mapsto (j_1, j_2, ..., j_\kappa) \subseteq Q$, we define the mapping error (in terms of the set of descriptors defined at the relevant extreme points) as follows:

$$\chi_f = \sum_{k=1}^{\kappa} \sum_{l=1, l \neq k}^{\kappa} E(h_{i_k i_l}, h_{j_k j_l}). \tag{9}$$

Given the definition of χ, our task is to search through a space of possible mappings between different subsets of P and Q to find the best one. A naïve search in the entire space clearly leads to exponential complexity [42], hence we first filter out obvious invalid mappings before searching (using the computed point-to-point difference $D(.)$, see details below). Next, we formulate the searching as a priority-driven graph search problem. We let each point pair $p_i \in P$ and $q_i \in Q$ be a graph node and connect each pair of graph nodes by an edge to obtain a complete graph G (see an example in Figure 6). A path in the graph then represents a possible correspondence mapping (e.g., the green path in Figure 6). The goal of our algorithm is to find a best path in the graph.

Algorithm 1 Our ant probing algorithm.

Require: The complete graph G formed from two point sets P and Q. Assume $|P| = m \leq n = |Q|$.
Ensure: A best graph path denoting a correspondence;
Place an ant on each graph node;
for each ant **do**
  while The ant still have possible nodes to go **do**
    Find the set of next top k quality nodes it can move to;
    if the set is not empty **then**
      move to one next node with the best quality;
      Compute the mapping error χ of the $path$ just probed and record the path if its length equals m and has the least χ value among all paths probed so far;
      if $\chi \geq leastcost$ probed so far **then**
        move back to the previous node;
      end if
    else
      move back to the previous node;
    end if
  end while
end for
Find the least-cost graph path with length m among all ants, denoted as $bestPath$;
return $bestPath$;

An exhaustive search for a best path is computationally prohibitive. Our goal of exploring all potential good paths is similar to the famous travelling salesman problem (TSP). We design an algorithm which operates like the well-known Ant Colony System (ACS) [45] and runs in polynomial time.

More specifically, we start by placing one ant at each graph node and let them probe simultaneously. An ant on each graph node finds several possible nodes to probe in the next step. Only those next possible nodes whose inclusion form paths that represent valid correspondences are considered. To limit the search space, at each node, we consider only the top k best quality nodes. The quality is measured as the path compatibility of the new node to all previously found nodes (i.e., by comparing $E(.)$'s, see details below). The ants-probing process is summarized in Algorithm 1.

Each ant probes in a DFS manner to first find a valid path, then it backtracks along the path to look for other possible paths that have not been probed. This is achieved by recording only the best k possible next nodes at each graph node of the path. Each ant records the best path (of length m) during probing, by measuring the mapping error χ (equation 9) of the current path. Probing into a specific node is terminated if the mapping error introduced by the node is already larger than the best mapping error found so far since χ is monotonically increasing. The probing stops when an ant returns to its starting node and all its k best next nodes have been probed. Finally, the path with the best quality recorded among all ants is selected as the final correspondence.

We apply several tests in our algorithm to reduce the search complexity. First, before graph construction, we filter out unrelated extremal point pairs by a similarity test. We omit (p_i, q_j) if the difference of their $D(,)$'s is larger than a user-defined threshold ρ (0.9 in our experiments). Second, when finding the next possible node for an ant on a node, we perform the following pruning tests on each of its adjacent node (p_k, q_l) in G:

1) If either k or l has already appeared in a previous node, we omit it. This ensures a 1:1 mapping and thus a valid correspondence.
2) For each previous node (p_i, q_j) in the ant's probed path, we check the path compatibility of $p_k \sim p_i$ with $q_l \sim q_j$ and disregard (p_k, q_l) if any $E(h_{ij}, h_{kl})$ is greater than a threshold (0.7 in our experiments);
3) If a pair passes the above two tests, we check the transformation error as defined in [42]. That is, we perform the $T4$ spatial configuration pruning test which filters out configurations with high deformation cost (such as symmetry switch) by computing the transformation error when deforming one set of points to another, based on a given point-to-point mapping (please refer to the paper for details).

Note that in traditional ACS each ant lays on every node of the path a certain amount of pheromone during probing, indicating the likelihood of the node being on a shortest path. The pheromone will have a certain probability to evaporate and the path with the most amount of pheromone will increasingly be more likely visited in later iterations and finally form an ideal shortest path (may not be the global solution). In our framework, we do not perform this task since the term $E(.)$ is path-dependent for different ants and hence cannot be aligned by a summarization procedure. In practice, we found our pruning strategies could significantly reduce the search space. Coupled with our effective cost function, our algorithm is able to find the solution that minimizes the mapping error very quickly.

Results and Evaluation. Figure 8 shows some results of our correspondence algorithm. Observe that our algorithm is capable of handling shapes with very different geometry details as well as only partially matchable shapes. The examples include shapes with rather different poses (teddy and ants models), shapes with very different surface details or topologies (the two dinosaur models, dog and wolf, two hands, and table and chair), and shapes that are only partially matchable (two armadillo models). Our correspondence algorithm is tolerant to different shape poses and surface variation since the harmonic fields are robust to isometric deformations and the descriptors are computed at both global and local scales, hence are not sensitive to differences in local shape details. The descriptors extracted for each extreme point encode the shape variations along different propagation paths between the point and all

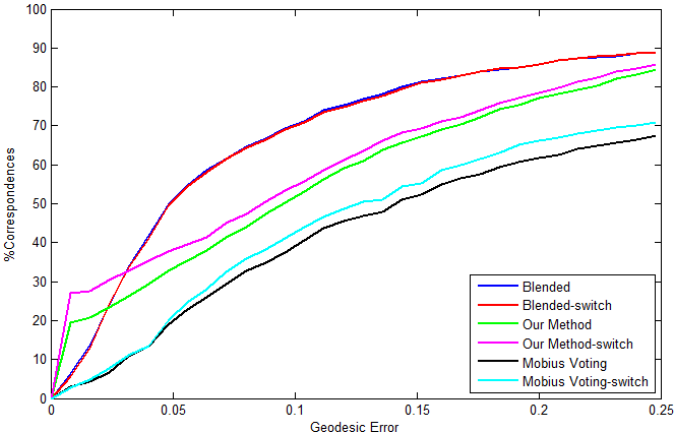


Fig. 7. Benchmark evaluation of our method on the animal data set of Kim et al. [12]. Our algorithm performs slightly worse than their blending method while being simpler and more efficient in finding the sparse correspondences.

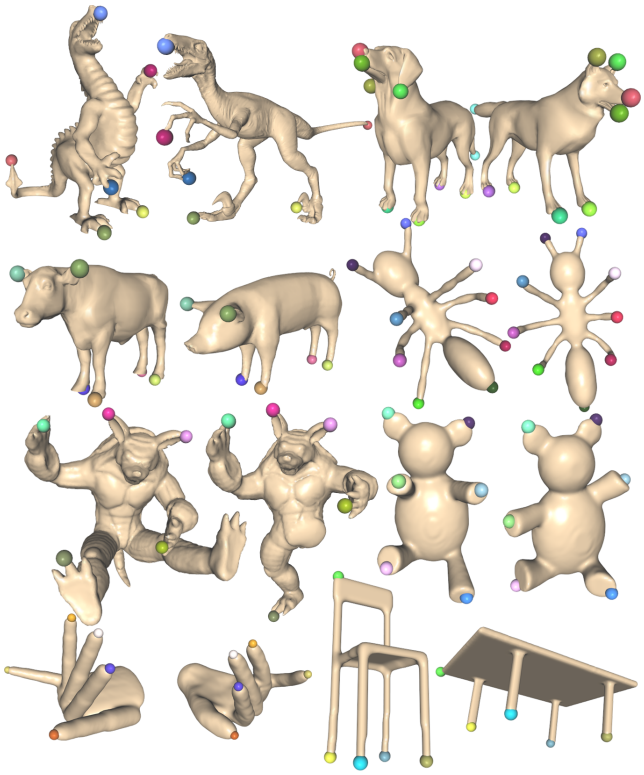


Fig. 8. Some correspondence results obtained using our framework. Our algorithm is able to produce good results for shapes with similar semantics but may differ in pose, geometry and be only partially matchable (e.g., Armadillo model).

other extreme points. Such shape information naturally distinguish points lying on different parts of the shape (e.g., a point on a hand from a point on the head in Figure 5).

We qualitatively analyze our correspondence algorithm using the most recent protocols of [12]. Specifically,

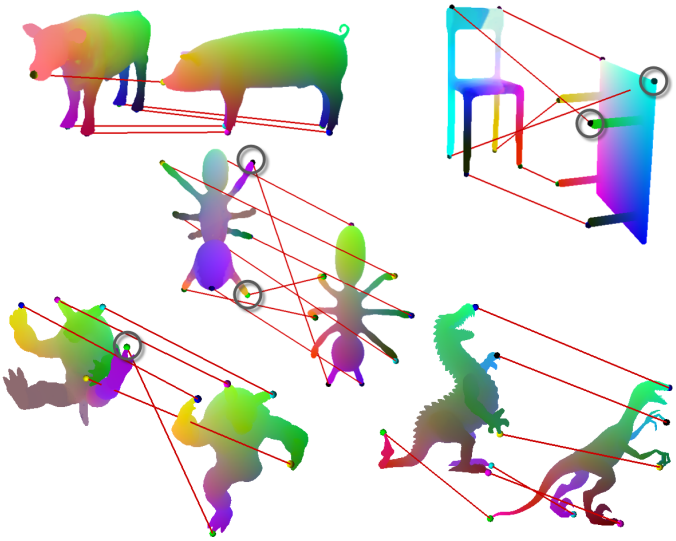


Fig. 9. Correspondence results from the blending method [12] for some models in Fig. 8.

we collect the animal data (51 meshes) as did in [12] and compute the sparse correspondence for each model with a randomly selected model (from the 51 animals, self-exclusive). We compare the results of our method with those of the blending method [12] and the Möbius voting method [10]. Figure 7 shows the evaluation results. The geodesic error is computed as the average geodesic error to the ground truth correspondences (measured in percentages). Our method performs slightly worse than the blending method and better than the method of Möbius voting in finding sparse correspondence. Note that here the curve is slightly different from the ones shown in [12] since we are using random comparison. In experiments, we found the blending method performs very well for those animal models while our method sometimes generates error in matching small branches like ears or symmetries due to pose variations. We also apply their method on some complex models used in our experiments, with topological changes or missing information as shown in Figure 8. It is interesting to observe that our method performs better than the blending method for models with large difference in both topology and geometry (chair-table), or twisted branches (the ants model), or even partially missing information (the Armadilos), which demonstrates the effective of the pairwise descriptors in measuring global shape characteristics. As for the Möbius voting method, we found that it does not work well for models with two large geometry differences, like a giraffe and a cat. The evaluation of our method is slightly better if accepting symmetry flips; this is because the deformation computation we employed [42] is not applied in a commonly embedded space (e.g., with MDS), hence if the shape undergoes large deformation (e.g., a man hugging himself), symmetry flips would be hard to recognize.

Like many other correspondence algorithms, our

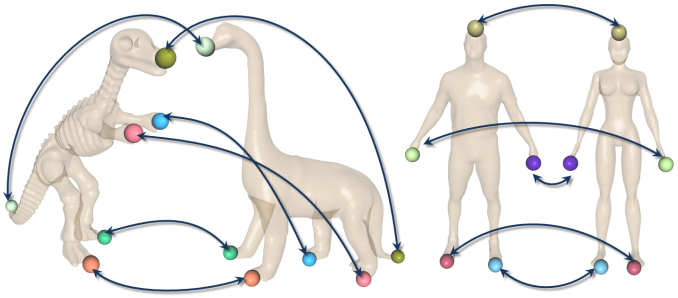


Fig. 10. Any correspondence algorithms that purely rely on geometry of input shapes (including ours) may fail on such ambiguities.

method may fail when the desired correspondence depends more on the shape semantics rather than the shape geometry, such as the two dinosaurs shown in Figure 10. Another limitation is that our algorithm can fail to identify a symmetry switch as shown in the right of Figure 10 where a front-faced model is matched with a back-faced one. This is true since such a symmetry switch can not be sensed by any purely intrinsic approaches. Nevertheless, Our algorithm runs one to two orders of magnitude faster than the state-of-the-art algorithm of [10], [41], typically taking under 30 seconds to find the correspondences. For most examples, we find that only a small number of k suffices to find the best mapping. We used a sufficiently large number ($k = 30$) for all models tested. A detailed evaluation of the parameter k is presented in Section 4.

3.3 Simultaneous Skeletonization and Segmentation

Previous Work. Shape skeletonization can aid in finding a rigid segmentation of the underlying shape [46] while segmentation can be exploited to extract a skeleton [47], [48]. A large body of recent research has been devoted to mesh segmentation [49], [50], [51] and skeletonization [46], [52]. We refer readers to a survey on segmentation techniques by Shamir [53] and a short review by Tagliasacchi et al. [52] on recent skeletonization methods.

Although shape skeletonization algorithms can often be easily extended to finding shape segmentation and vice versa, the method of Lien et al. [48] has been, to our knowledge, the only work that computes them simultaneously. It extracts a skeleton from the segmentation boundaries and iteratively refines the skeleton by further decomposition of the segmentation. We design a simple simultaneous decomposition and skeletonisation algorithm that uses the pairwise harmonics. We show that, compared with their method, ours extracts skeletons that more resemble those obtained by rigid shape decomposition, with the junction nodes located at the branching surface and each skeletal branch corresponds to one rigid shape component.

Our Method. Our method is motivated by the dual roles of the isocurves. Each isocurve of a pairwise harmonic field encircles the local shape, capturing the cross sectional profile. Connecting the centroids of the isocurves produces a skeletal segment (Figure 11 (a)). On the other hand, isocurves are also good candidates for segmentation boundaries.

The outline of our method is shown in Figure 11. We first sample the surface to obtain a sparse set of points that generate pairwise harmonic fields spanning the whole model. Connecting the centroids of isocurves of each field produces a skeletal segment. To obtain reliable skeletal segments from all pairwise harmonics, we first assign *node rigidity* at each skeletal node indicating whether that node lies within a rigid or branching part, and divide the skeletal segments at nodes with low rigidity. We then employ a greedy algorithm to select a set of disconnected prominent skeletal segments, forming a partial skeleton. These chosen skeletal segments induce an initial segmentation. Next, we leverage the shape segmentation to guide the completion of the skeleton. Specifically, we compute center nodes at junction segmentation regions and connect up the disconnected skeletal segments based on segmentation adjacency. Finally, the segmentation result is refined according to the completed skeletal structure.

We employ the max-min sampling strategy [10] to get a set of sparsely distributed points, denoted by Δ. These points are mostly located at prominent shape parts (Figure 11 (a)) so that the resulting pairwise harmonics populate the entire mesh surface. For each point pair in Δ, we compute a pairwise harmonic and connect the centers of all isocurves sequentially to form a skeletal segment (Figure 11 (a)). In this way, each point pair produces exactly one skeletal segment.

Denote the set of all skeletal segments by Φ and the set of all center points (skeletal nodes) by C. Note that there may be more than one skeletal segment corresponding to a specific shape component (e.g., arm in (Figure 11) extracted from different fields. We define the node rigidity ξ of each skeletal node $j \in C$ as follows:

$$\xi_j = \frac{max\{\lambda_1, \lambda_2, \lambda_3\}}{\lambda_1 + \lambda_2 + \lambda_3}, \tag{10}$$

where λ_k is the eigenvalue of the k-th principle component computed on a neighborhood of node j. For this purpose, we use all points ($\in C$) within a sphere centered at node j whose radius equals to the radius of the associated isocurve (defined as perimeter divided by 2π). A node is defined to be non-rigid if its ξ value is smaller than 0.9. Figure 11 (b) shows all the skeletal nodes, with the colors blue, green to red indicating increasing rigidity. One can see that non-rigid nodes are mostly located in the branching parts of the shape.

Next, for each skeletal segment in Φ, we divide it into sub-segments at each non-rigid node, resulting in a larger set Φ' of segments. For example, the single skeletal segment in Figure 11 (a) would be divided into three sub-segments if there are two non-rigid nodes. We assign a score ρ to each resulting segment s indicating

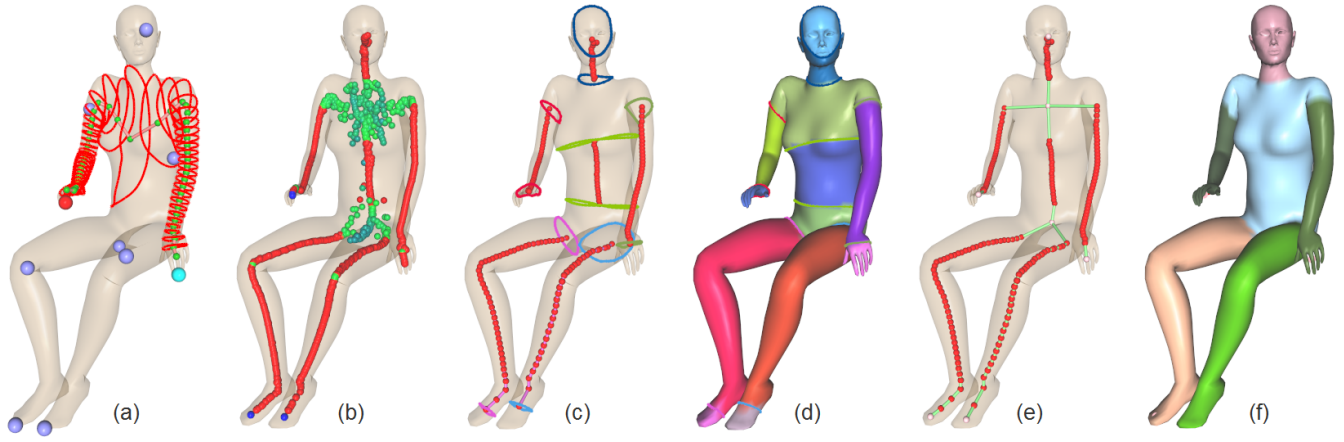


Fig. 11. Simultaneous shape segmentation and skeletonisation. (a) We sample the shape to get a sparse set of points that generates pairwise harmonic fields that span the model. Isocurves extracted from each pairwise harmonic produces a potential skeletal segment. (b) Each node is assigned a rigidity score indicating its likelihood of lying on a rigid part (blue, green to red indicates increasing rigidity); (c) segments with the highest rigidity are selected as candidate skeletal segments; (d) an initial segmentation is obtained by cutting along the isocurves at two ends of each selected skeletal segments; (e) the skeleton is completed according to the segmentation adjacency; (f) the segmentation result is in turned refined according to the completed skeleton.

its likelihood of being a constituent of the final skeleton. That is,

$$\rho_s = \xi_s * l_s \tag{11}$$

where ξ_s denotes the segment rigidity, averaged over all the nodes on that segment, and l_s is the segment length. To construct the initial partial skeleton, we iterate through all the segments in Φ', each time selecting the segment with the highest ρ. We omit any new segment that overlaps with any already selected one, thereby eliminating nearby skeletal segments (e.g., only one segment that run along an arm is selected). Figure 11 (c) shows the selected skeletal segments. Note that each segment corresponds to a rigid shape component, while most non-rigid nodes are in the junction regoins.

The selected skeletal segments induces an initial shape segmentation. Specifically, we partition the mesh using the isocurves associated with the two end nodes of each skeletal segment (Figure 11 (c)). This produces a segmentation where each skeletal segment corresponds to a major shape component, and the remainder components correspond to either junctions or extrema of a skeleton (Figure 11 (d)). Next, to complete the skeleton, we simply compute the centers of components corresponding to junction or extrema nodes and connect these centers to adjacent skeletal segments according to the segmentation adjacency (Figure 11 (e)).

Finally, we revert to segmentation refinement. First, we merge a component that corresponds to a skeletal end point (e.g., hand in Figure 11 (d)) to its sole adjacent component. We then exploit the constructed skeleton for further segmentation refinement at junction parts. In particular, we consider each skeletal segment that is incident to two junction nodes (e.g., the body part shown in blue in Figure 11) and try merging its associated component with the two adjacent components (e.g., the blue body part is merged with its adjacent components corresponding to two junctions nodes). This process is repeated by selecting the skeletal segments with the largest component volume until no further merging is possible. If any component is merged with a junction node, the junction node is tagged and removed from further consideration. Figure 11 (f) shows the final segmentation. In addition, Figure 12 shows more skeletonization and segmentation results. Observe that the use of the rigidity term and a skeleton in guiding the segmentation produces a partition that mostly agrees with the shape's rigidity without over-segmentation. Hence, our method generally conforms more to the shape's articulatory structure whereas most state-of-the-art segmentation methods tend not to recover such information.

Results and Evaluation. Comparing with the mesh contraction algorithm [46], we find that our skeletonization results are comparable for many models. In some cases when a component does not merit a good principle axis, our algorithm can fail to identify a branch (e.g., the body part of the teddy shown in Figure 13). In such cases, the segmentation results are also affected (see the segmentation of teddy where the body and head form one component). Nevertheless, our algorithm is simpler and more efficient than the method of Au et al. [46] which involves iteratively solving a large linear system. Note that, like their method, our algorithm operates directly on mesh surfaces and preserves shape topology.

For segmentation, we quantitatively evaluate our results using the protocol of Chen et al. [54]. Since our segmentation algorithm builds upon skeleton extraction, we exclude the head, vase, fish and CAD categories whose models do not have good definition of skeletons. Our method is capable of handling fairly complex

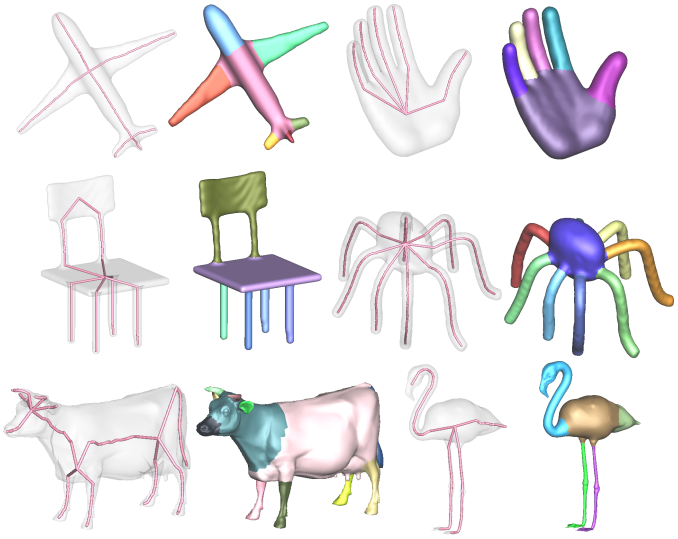


Fig. 12. Segmentation and skeletonisation results of some representative models.

models such as the cow model (Figure 12) and the menmento model (Figure 13) and can deal with models with different topologies (the chair model). Observe that our algorithm leads to segmentation results that divide a shape into near-rigid components, thanks to the support of the underlying shape skeleton. From the evaluation results (Figure 14 and 15), we find that our simple method produces results comparable with state-of-the-art algorithms but is much faster. It typically takes under 30 seconds to segment a model. Since our methods are built on harmonic fields, which are not sensitive to surface noise and isometric deformation, our results are robust to surface noise, tessellations and poses [38].

4 Parameters

In this section, we briefly examine the parameters used in the three applications.

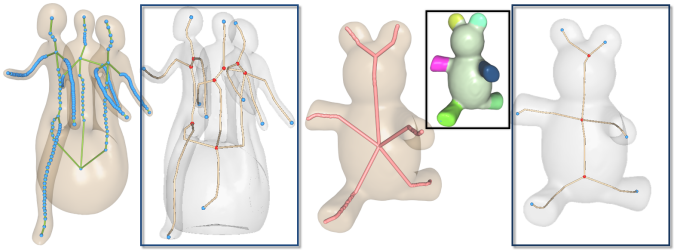


Fig. 13. Our skeletonization algorithm generates comparable results to the method of Au et al. [46] for most models. However, when no obvious principle axis exists in a component, our method may fail to locate a skeletal segment (e.g., the bottom piece of the menmento model), leading to a degraded result. The generated skeleton influences the segmentation results (e.g., the body of the teddy bear is not segmented from its head). Images in blue frames are results from the method of Au et al.

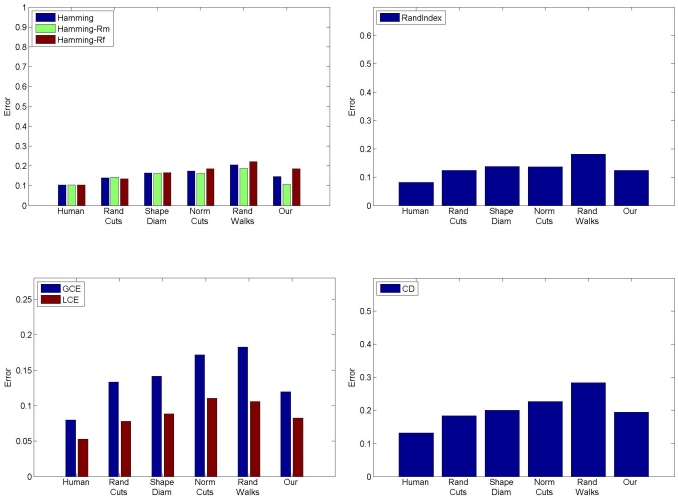


Fig. 14. Quantitative analysis of our segmentation results using protocols of Chen et al. [54]. Our method performs slightly better than the state-of-the-art methods while being much simpler and efficient.

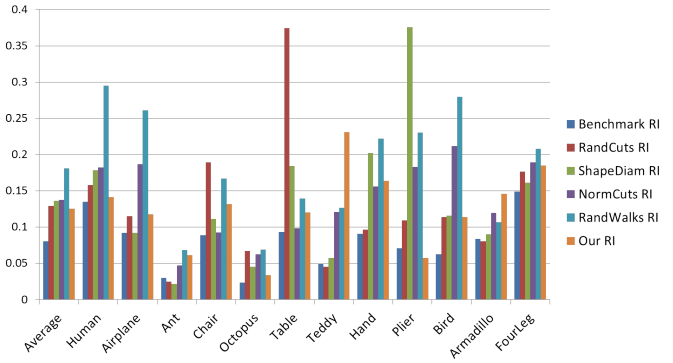


Fig. 15. By class rand index error from protocols of Chen et al. [54]. Our method generally performs better than the state-of-the-art methods. For the teddy category, our method performs worse due to the small branching parts of the teddy models which can lead to degraded skeletons (see Figure 13) since we are using a fixed rigidity threshold.

Intrinsic Symmetry Detection. the similarity test used in our intrinsic symmetry computation method is to filter out uncorrelated point pairs before the harmonics computation. This is mainly for efficiency and does not affect the final results if we increase the value, i.e., allow more pairs to pass through, since they will finally be filtered out by the user-defined symmetry parameter ϵ. The parameter ϵ controls the number of voting candidate point pairs. Although increasing its value may lead to errors in the final symmetry selection due to more undesired pairs participating in the voting, we found that the results tend to be stable against this parameter since true symmetry axes will still be more likely to receive more votes (Figure 16). The range of $[2, 4]$ produces satisfactory results.

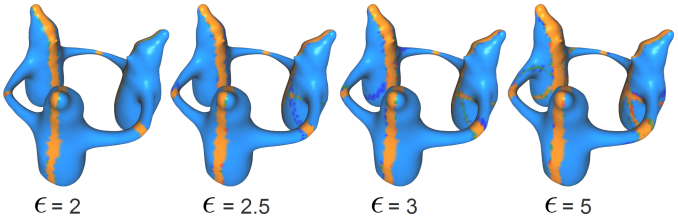


Fig. 16. Increasing the value of ϵ does not affect the results significantly since true symmetry axes are more likely to receive more votes.

Shape Extremities Matching. For our correspondence algorithm, we used five parameters during graph construction and the pruning tests. The user-defined threshold ρ and the geodesic test parameter are to filter out invalid nodes at the stage of graph construction. A small value of ρ can improve the efficiency by reducing the graph size but may lead to potential candidate pairs being rejected if two shapes have large variations such as the two dinosaurs shown in Figure 8. A large value allows more pairs to pass the test (hence allowing matching of shapes with very different geometry) but also decreases the efficiency and may introduce errors in the final mapping. We find that values between $0.8-0.95$ tend to give satisfactory results. As for the value of the spatial configuration parameter (T4) used in our pruning test during probing, we simply set it to the same range as the one used in [46].

The last parameter used in our correspondence algorithm is the searching parameter k. When $k = 1$, each ant only probes the current best possible node hence may easily lead to local minima. When $k = |G|$ the entire space is explored. Our experiments show that the algorithm converges rather quickly to the global optimum, in other words, a small k suffices to lead the algorithm to the best solution. Table 1 shows the value of k used to find the best solution for the models in Figure 8. It can be noticed that surfaces with large variations and many branches may require a larger value of k to find the best solution. We found that a k value smaller than 10 could already lead to a good correspondence solution for most models; for complex ones such as the ants model, a small k value produces results up to some symmetry switches. Hence, in our experiments, we set k to a fixed number of 30 for all our examples.

Segmentation and Skeletonization. The only parameter used in our segmentation and skeletonization method is the threshold to define a stable node. For consistency, we use 0.9 for all our examples in the paper and for the evaluation. This value of the parameter has rather minor impact on the final results. Specifically, smaller values of this parameter can cause more nodes to be classified as being stable, which can be helpful in some cases to identify less prominent branches, such as the body of the teddy bear shown in Figure 13. On the other hand, larger values lead to less noisy results. We found that a value between 0.85-0.95 produces satisfactory results.

two raptors	dog-wolf	pig-cow	two ants
9	6	19	20
armadillo	teddy	hands	table-chair
2	5	5	5

TABLE 1
Number of k's in finding the best solutions.

All Applications. An important parameter used in all three applications is the number of isocurves extracted from each harmonic field. Naturally, it would be desirable to sample a number of isocurves according to the geodesic distance of the two points. However, for simplicity, we used a constant number for each application. For symmetry computation and shape matching applications, we found that a small set of isocurves suffices to render both algorithms robust, and we use $K = 16$. For segmentation and skeletonization, to faithfully locate desired skeletal nodes in all surface regions, we sample 50 isocurves per harmonic field.

5 Conclusion and Future Work

In conclusion, this paper introduces a simple and effective shape analysis tool based on point pairs on the surface. Unlike previous shape descriptors which are derived mostly based on analysis around individual points, in this work, we define a set of shape descriptors from pairwise harmonics which is rather simple yet capable of capturing important shape geometry and global shape structures. We show the benefits of the pairwise analysis framework with three applications.

Though the concept of harmonic fields has been widely adopted by the geometry community, we exploit it in a rather different context, i.e., in a pairwise manner. We believe such a coupled way of analyzing a shape will open up new possibilities for shape analysis in the future. Indeed, it is interesting to note that in the independent parallel work of Xu et al. [55] and Kaick et al [56], point pairs have been effectively examined and employed for shape analysis. In the work of [55], intrinsic symmetries at various scales are detected using point-pair symmetry support voting and clustering and in the work of [56] fuzzy geodesics between point pairs are exploited for partial shape correspondence and shape retrieval [56]. In addition to this, we believe such a pairwise analysis method can be naturally applied to more applications such as shape co-analysis and part recognition.

The pairwise analysis is capable of sensing the shape semantics, which is exploited for the first time in this work. This capability will encourage us to explore more on shape analysis tools that can capture shape semantics in the future. For example, it would be interesting to explore the extension of pairwise analysis to a multi-way analysis, which has the potential of revealing even richer information in the surface. A natural application is shape matching. Finally, we wish to explore means to automate the choice of points in the point pairs in various applications.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments, Kevin Xu for providing the models for symmetry axes extraction, Xin Zhao and Han Liu for their help in the evaluation with the correspondence benchmark. This work was supported in part by grants from the Hong Kong Research Grant Council (Project No. GRF619611, GRF619012). Eugene Zhang was partially supported by NSF Award IIS-0917308.

REFERENCES

[1] A. D. Bimbo and P. Pala, "Content-based retrieval of 3d models," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, pp. 20–43, 2006.

[2] R. Gal, O. Sorkine, N. J. Mitra, and D. Cohen-Or, "iwires: An analyze-and-edit approach to shape manipulation," *ACM Transactions on Graphics (Siggraph)*, vol. 28, no. 3, pp. #33, 1–10, 2009.

[3] J.-P. Vandeborre, V. Couillet, and M. Daoudi, "A practical approach for 3d model indexing by combining local and global invariants," in *3DPVT'02*, 2002, pp. 644–647.

[4] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH '05, 2005, pp. 659–666.

[5] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, pp. 433–449, 1999.

[6] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonization using the shape diameter function," *The Visual Computer*, vol. 24, no. 4, pp. 249–259, 2008.

[7] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Eurographics Symposium on Geometry Processing (SGP)*, 2009, pp. 1383–1392.

[8] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, "One point isometric matching with the heat kernel," in *Eurographics Symposium on Geometry Processing (SGP)*, vol. 29, no. 5, 2010, pp. 1555–1564.

[9] T. K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang, "Persistent heat signature for pose-oblivious matching of incomplete models," *Comput. Graph. Forum*, pp. 1545–1554, 2010.

[10] Y. Lipman and T. Funkhouser, "Möbius voting for surface correspondence," in *ACM SIGGRAPH 2009 papers*, ser. SIGGRAPH '09, 2009, pp. 72:1–72:12.

[11] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser, "Mobius transformations for global intrinsic symmetry analysis," *Computer Graphics Forum (Proc. of Symposium on Geometry Processing)*, 2010.

[12] V. G. Kim, Y. Lipman, and T. Funkhouser, "Blended intrinsic maps," *Transactions on Graphics (Proc. of SIGGRAPH 2011)*, 2011.

[13] N. J. Mitra, L. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 25, no. 3, pp. 560–568, 2006.

[14] K. Xu, H. Zhang, D. Cohen-Or, and Y. Xiong, "Dynamic harmonic fields for surface processing," *Computers and Graphics (SMI '09)*, vol. 33, pp. 391–398, 2009.

[15] E. Zhang, K. Mischaikow, and G. Turk, "Feature-based surface parameterization and texture mapping," *ACM Trans. Graph.*, vol. 24, no. 1, pp. 1–27, 2005.

[16] E. Zhang and G. Turk, "Visibility-guided simplification," in *Proceedings of the conference on Visualization '02*, ser. VIS '02, 2002, pp. 267–274.

[17] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proceedings of the Shape Modeling International 2003*, 2003, pp. 130–.

[18] C. B. Akgul, B. Sankur, Y. Yemez, and F. Schmitt, "3d model retrieval using probability density-based shape descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1117–1133, 2009.

[19] "A survey of content based 3d shape retrieval methods," in *Proceedings of the Shape Modeling International 2004*, 2004, pp. 145–156.

[20] M. Ben-Chen and C. Gotsman, "Characterizing shape using conformal factors," in *3DOR*, 2008, pp. 1–8.

[21] A. Tevs, A. Berner, M. Wand, I. Ihrke, and H.-P. Seidel, "Intrinsic shape matching by planned landmark sampling," in *Eurographics*, 2011, pp. 543–552.

[22] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature maps for local shape comparison," in *In Shape Modeling International*, 2005, pp. 244–256.

[23] V. Kraevoy, A. Sheffer, and C. Gotsman, "Matchmaker: constructing constrained texture maps," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03, 2003, pp. 326–333.

[24] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel, "Harmonic guidance for surface deformation," *Computer Graphics Forum (Proc. of EUROGRAPHICS 2005)*, vol. 24, no. 3, pp. 601–609, 2005.

[25] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart, "Spectral surface quadrangulation," *ACM Trans. Graph.*, vol. 25, pp. 1057–1066, July 2006.

[26] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," *Visualization and Mathematics III*, pp. 35–57, 2002.

[27] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete laplace operators: no free lunch," in *Proceedings of the fifth Eurographics symposium on Geometry processing*, 2007, pp. 33–37.

[28] T. Davis, *User Guide for CHOLMOD: a sparse Cholesky factorization and modification package*, 2008.

[29] H. WEYL, *Symmetry*. Princeton University Press, 1983.

[30] M. Leyton, *A generative theory of shape*. Berlin, Heidelberg: Springer-Verlag, 2001.

[31] J. Podolak, A. Golovinskiy, and S. Rusinkiewicz, "Symmetry-enhanced remeshing of surfaces," in *Proceedings of the fifth Eurographics symposium on Geometry processing*, 2007, pp. 235–242.

[32] N. J. Mitra, L. Guibas, and M. Pauly, "Symmetrization," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 26, no. 3, pp. #63, 1–8, 2007.

[33] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A reflective symmetry descriptor for 3d models," *Algorithmica*, vol. 38, pp. 201–225, 2003.

[34] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A planar-reflective symmetry transform for 3d shapes," *ACM Trans. Graph.*, vol. 25, pp. 549–559, 2006.

[35] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Symmetries of non-rigid shapes," *IEEE 11th International Conference on Computer Vision (2007)*, pp. 1–7, 2007.

[36] K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng, and Y. Xiong, "Partial intrinsic reflectional symmetry of 3D shapes," *ACM Trans. Graph.*, vol. 28, pp. 138:1–138:10, 2009.

[37] Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, "Symmetry factored embedding and distance," *ACM Transactions on Graphics (SIGGRAPH 2010)*, Jul. 2010.

[38] Y. Zheng and C.-L. Tai, "Mech decomposition with cross-boundary brushes," in *Computer Graphics Forum (In Proc. of Eurographics 2010)*, vol. 29, no. 2, 2010, pp. 527–535.

[39] F. Mémoli and G. Sapiro, "A theoretical and computational framework for isometry invariant recognition of point cloud data," *Found. Comput. Math.*, vol. 5, no. 3, pp. 313–347, 2005.

[40] B. M. M. BRONSTEIN, A. M. and R. KIMMEL, *Numerical geometry of non-rigid shapes*. Springer-Verlag New York Inc., 2008.

[41] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi, "Deformation-drive shape correspondence," *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing 2008)*, vol. 27, no. 5, pp. 1431–1439, 2008.

[42] O. K.-C. Au, C.-L. Tai, D. Cohen-Or, Y. Zheng, and H. Fu, "Electors voting for fast automatic shape correspondence," in *Computer Graphics Forum (In Proc. of Eurographics 2010)*, vol. 29, no. 2, 2010, pp. 645–654.

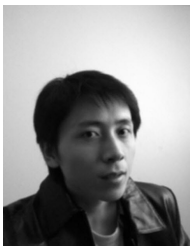
[43] O. van Kaick, A. Tagliasacchi, O. Sidi, H. Zhang, D. Cohen-Or, L. Wolf, and G. Hamarneh, "Prior knowledge for shape correspondence," *Computer Graphics Forum (Special Issue of Eurographics)*, vol. 30, no. 2, pp. 553–562, 2011.

[44] H. W. Kuhn, "The hungarian method for the assignment problem," in *Naval Research Logistics (NRL)*, vol. 52, no. 2, 2005, pp. 7–21.

[45] M. Dorigo and L. Gambardella, "Ant colony system : A cooperative learning approach to the traveling salesman problem," vol. 1, no. 1, pp. 53–66, 1997.

[46] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[47] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts," *ACM Trans. Graph.*, pp. 954–961, 2003.
[48] J.-M. Lien, J. Keyser, and N. M. Amato, "Simultaneous shape decomposition and skeletonization," in *SPM '06*, 2006, pp. 219–228.
[49] L. G. KATZ, S. and A. TAL, "Mesh segmentation using feature point and core extraction." in *The Visual Computer*, vol. 21, 2005, pp. 649 – 658.
[50] A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3D mesh analysis," *ACM Trans. Graph.*, pp. 1–12, 2008.
[51] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D Mesh Segmentation and Labeling," *ACM Transactions on Graphics*, vol. 29, no. 3, 2010.
[52] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Transactions on Graphics, (Proceedings SIGGRAPH 2009)*, vol. 28, no. 3, pp. 71:1–71:9, 2009.
[53] A. Shamir, "A survey on mesh segmentation techniques," *Computer Graphics Forum*, vol. 27, no. 6, pp. 1539–1556, 2008.
[54] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graph.*, pp. 1–12, 2009.
[55] K. Xu, H. Zhang, W. Jiang, R. Dyer, Z. Cheng, L. Liu, and B. Chen, "Multi-scale partial intrinsic symmetry detection," *ACM Transactions on Graphics (Special Issue of SIGGRAPH Asia)*, vol. 31, no. 6, p. to appear, 2012.
[56] O. van Kaick, H. Zhang, and G. Hamarneh, "Bilateral maps for partial matching," *Computer Graphics Forum*, p. to appear, 2013.



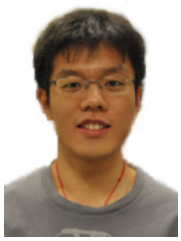
Youyi Zheng is a Postdoc at the Geometric Modeling and Scientific Visualization Center, King Abdullah University of Science and Technology. He received his B.Sc. and M.Sc. in Mathematics both from the Zhejiang University and his PhD degree from the Hong Kong University of Science and Technology. His research interests include interactive techniques, digital geometric processing, computer graphics and computer vision.



Chiew-Lan Tai is currently an associate professor of computer science at the Hong Kong University of Science and Technology. She received the BSc degree in mathematics from the University of Malaya, MSc in computer and information sciences from the National University of Singapore, and DSc degree in information science from the University of Tokyo. Her research interests include geometry processing, computer graphics, and interactive techniques.



Eugene Zhang received the PhD degree in computer science in 2004 from Georgia Institute of Technology. He is currently an associate professor at Oregon State University, where he is a member of the School of Electrical Engineering and Computer Science. During part of 2011 and 2012 he was a guest professor at Free University of Berlin and the Max-Planck-Institute in Informatics. His research interests include computer graphics, scientific visualization, geometric modeling, and computational topology. He received an National Science Foundation (NSF) CAREER award in 2006. He is a senior member of the IEEE and the ACM.



Pengfei XU is a Ph.D. candidate at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. He received his B.Sc. in Mathematics from the Zhejiang University. His research interests include digital geometric processing, interactive techniques and computer graphics.