Graphical Models 75 (2013) 177–188

Contents lists available at SciVerse ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Skeleton-based intrinsic symmetry detection on point clouds ☆

Wei Jiang [a], Kai Xu [a,*], Zhi-Quan Cheng [a], Hao Zhang [b]

[a] HPCL, School of Computer, National University of Defense Technology, China
[b] School of Computing Science, Simon Fraser University, Canada

ARTICLE INFO

Article history:
Received 15 October 2012
Received in revised form 28 February 2013
Accepted 4 March 2013
Available online 19 March 2013

Keywords:
Symmetry detection
Intrinsic symmetry
Skeleton
Point clouds
Self-isometry

ABSTRACT

We present a skeleton-based algorithm for intrinsic symmetry detection on imperfect 3D point cloud data. The data imperfections such as noise and incompleteness make it difficult to reliably compute geodesic distances, which play essential roles in existing intrinsic symmetry detection algorithms. In this paper, we leverage recent advances in curve skeleton extraction from point clouds for symmetry detection. Our method exploits the properties of curve skeletons, such as homotopy to the input shape, approximate isometry-invariance, and skeleton-to-surface mapping, for the detection task. Starting from a curve skeleton extracted from an input point cloud, we first compute *symmetry electors*, each of which is composed of a set of skeleton node pairs pruned with a cascade of symmetry filters. The electors are used to vote for *symmetric node pairs* indicating the symmetry map on the skeleton. A symmetry correspondence matrix (SCM) is constructed for the input point cloud through transferring the symmetry map from skeleton to point cloud. The final symmetry regions on the point cloud are detected via spectral analysis over the SCM. Experiments on raw point clouds, captured by a 3D scanner or the Microsoft Kinect, demonstrate the robustness of our algorithm. We also apply our method to repair incomplete scans based on the detected intrinsic symmetries.

1. Introduction

Symmetry is a universal phenomenon in nature, science, and art. Recently, symmetry analysis and symmetry-aware shape processing have been intensively studied in computer graphics, where the vast majority of existing works have been on *extrinsic* symmetry detection over complete shapes represented by polygonal meshes [1]. In this paper, we are interested in the problem of *intrinsic* symmetry detection over point cloud data. In particular, the input point cloud is assumed to be imperfect with noise and missing data, which are typical results of acquisition via 3D capture/scanning devices.

Intrinsic symmetry is defined as a region over a shape that possesses a self-map that preserves geodesic distances. Naturally, existing approaches to intrinsic symmetry detection have relied on geodesic distances in one way or another and they have all been applied to closed meshes [2–5]. One exception is the work by Ovsjanikov et al. [6] which detects symmetries in signature space defined by the eigenfunctions of the Laplace–Beltrami operator. However, this method also works on meshes and cannot deal with substantial topological defects. However, accurate geodesic distance computation over a point cloud is challenging in general [7], making it difficult to adapt existing symmetry detection schemes on meshes to work on imperfect point clouds.

The key idea in this paper is to take advantage of the recent success on robust curve skeleton extraction from imperfect point cloud data [8–12] and transform the symmetry detection problem from an input point cloud to its extracted curve skeleton. Given an imperfect point cloud, we expect curve skeleton extraction to be an easier problem than that of intrinsic symmetry detection since the

☆ This paper has been recommended for acceptance by Michael Wand.

* Corresponding author.
E-mail address: kaixu@nudt.edu.cn (K. Xu).

1524-0703/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.gmod.2013.03.001

former relies primarily on local geometry analysis and the latter is a global problem that is particularly susceptible to missing data. That being said, imperfect point clouds typically result in imperfect skeletons, which pose challenges to the ensuing symmetry analysis. For 3D scans of human body obtained by a RGBD camera such as Microsoft Kinect, a pre-computed skeleton is fitted to the scan in real time, which can be directly used in our symmetry detection pipeline. The detected symmetry information can in turn be used to complete the low quality scans of Kinect.

In the paper, we propose an intrinsic symmetry detection algorithm over imperfect point clouds by exploiting several properties resulting from robust curve skeleton extraction [9,10,12]: (i) homotopy to the input shape; (ii) approximate isometry-invariance; and (iii) skeleton-to-surface mapping. In our work, by isometry, we mainly refer to articulated deformation (or skeletal deformation) of the input shape which is approximately isometric.

Given a point cloud, our algorithm returns a set of self-symmetric regions, as well as a set of pairs of symmetric regions, if any. Each detected symmetry represents a partial intrinsic symmetry and together they constitute a global intrinsic symmetry of the input shape, see Fig. 1(4). To accomplish this, we compute *symmetry electors*, each of which is composed of a set of node pairs filtered by a cascade of pruning tests. The electors are then used to vote for *symmetric node pairs* indicating the symmetry map on the skeleton. The symmetry map is transferred to the point cloud, leading to a symmetry correspondence matrix over which we perform spectral analysis to extract the final symmetry regions for the point cloud.

2. Related work

We concentrate on the most relevant works in intrinsic symmetry detection. For comprehensive review, we refer the reader to the survey [1].

Existing approaches to intrinsic symmetry detection have so far been working with closed meshes. To obtain the geodesic distance preserving self-mapping, Raviv et al. [2,5] directly minimize distance distortion in the space of Generalized Multi-Dimensional Scaling (GMDS) embedding. Ovsjanikov et al. [6] detect global intrinsic symmetry through exploiting the fact that the intrinsic symmetries of a shape are transformed into the Euclidean symmetries in the signature space defined by the eigenfunctions of the Laplace–Beltrami operator. Lipman and Funkhouser [13] develop the Möbius voting to find near-isometric correspondence for 3D shapes, which is extended by Kim et al. [14] in detecting global intrinsic symmetry. Through defining continuous symmetry as infinitesimal rigid transformations, represented as tangent vector fields, Ben-Chen et al. [15] detect cylindrical and translational symmetry on a smooth manifold surface.

Berner et al. [16] detect partial symmetries in 3D objects through matching graphs of feature lines. Lasowski et al. [17] propose a probabilistic framework for partial intrinsic symmetry detection based on Markov random field model. Mitra et al. [18] detect intrinsic regularity in 3D shapes using multidimensional scaling. Xu et al. [3] detect partial intrinsic symmetries for 3D shapes through directly voting for intrinsic reflectional symmetry axis transform on the input surface.

A few works has been devoted to symmetry detection on imperfect *point cloud* data. Some researchers [4,19] have attempted to detect *extrinsic* symmetry on point clouds. Their success relies on the robust extraction of symmetry-invariant features. Lipman et al. [20] propose to detect symmetry in the space of correspondences. They construct a random walk matrix for symmetry, called symmetry correspondence matrix (SCM), where each entry measures the probability of two points being symmetric. Spectral analysis is performed on the SCM to extract the symmetry regions as orbits reflecting the connectedness in the random walk graph. Due to the robustness of spectral analysis, the method can be used to detect extrinsic symmetry on point clouds. Xu et al. [21] extend this approach to detect multiscale partial intrinsic symmetries. However, none of the above works can detect *intrinsic* symmetry on point clouds. Our paper makes it possible through com-

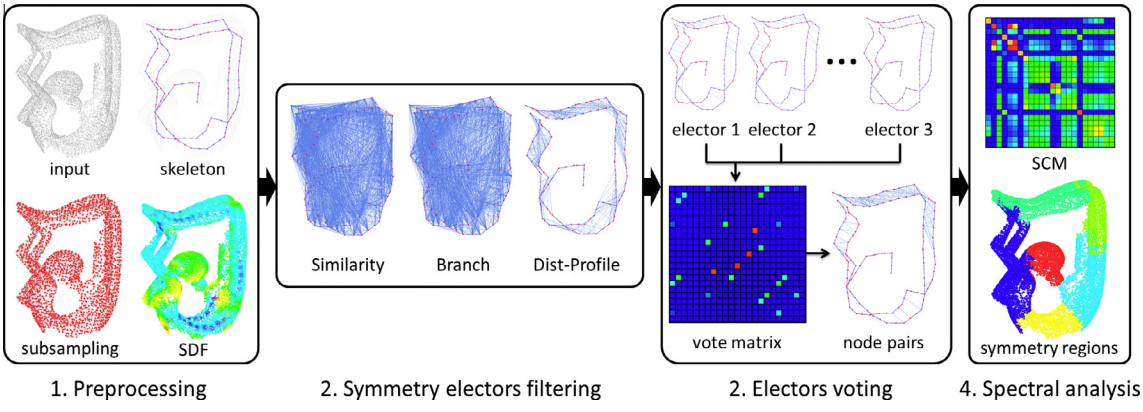

Fig. 1. The pipeline of our algorithm. (1) In the preprocessing step, the input point cloud is uniformly sampled. A curve skeleton is extracted and the symmetry invariant local shape descriptors are pre-computed. (2) A set of symmetry electors are selected based on a cascade of pruning tests. (3) The electors are used to vote for symmetric node pairs. (4) Based on the symmetric node pairs, a symmetry correspondence matrix is constructed over which spectral analysis is performed to extract symmetries.

bining the framework with a skeleton-based approach. Recently, Berger and Silva [22] propose a method for computing non-rigid matching for point clouds through deriving the so-called medial Laplacian and a diffusion process based on the medial axis.

3. Algorithm

Given a point cloud, we first extract its curve skeleton using existing approaches [9,10,12]. Some of the existing skeleton extraction approaches require point clouds with normals. However, our symmetry detection does not require normal information. Throughout this paper, we assume that the extracted skeleton is topologically identical, or homotopic, to the input shape. This assumption is practically reasonable for moderately incomplete 3D point clouds. For Kinect scans of human body, we can directly use the realtime fitted skeleton provided by the Kinect SDK.

The objective of our algorithm is to find from the input point cloud a set of non-overlapping regions (not necessarily covering the whole point cloud) each of which possesses a intrinsic symmetry. All these regions constitute the intrinsic symmetry of the point cloud. See Fig. 1 for an example.

3.1. Preprocessing

The curve skeletons extracted by the existing methods are typically of low resolution (contain a few skeleton nodes). To achieve a better accuracy of symmetry detection, we first refine the skeletons by inserting new nodes on the bones until the prescribed sampling rate is satisfied: the distance between every two adjacent nodes is about 0.01 times the maximal path length on the skeleton. The input point cloud is subsampled using the farthest point sampling approach based on Euclidean distance. In the preprocessing stage, we also perform some pre-computation, including geodesic distances on skeleton, symmetry-invariant local shape descriptors, etc., which will be detailed in the related subsections.

3.2. Overview

Let us denote the sampled point cloud by a set $\mathcal{P}$ of M *sample points*, and its refined curve skeleton by a set $\mathcal{S}$ of N *skeleton nodes*.

Algorithm 1 gives the pseudocode of our algorithm. Fig. 1 shows the pipeline of our algorithm. Our algorithm operates in four steps.

First, it constructs a set of *symmetry electors* from the skeleton nodes through a cascade of filtering tests and a voting based aggregation process (**SymmetryElectors**); see SubSection 3.3. This step is elaborated in Algorithm 2. Second, the electors cast votes on individual candidate node pairs to establish the symmetry correspondences on skeleton nodes (**VoteNodePairs**); see SubSection 3.4. Third, the symmetry correspondences on the skeleton are transferred onto the point cloud, leading to a *symmetry correspondence matrix* (SCM) (**BuildSymCorrMat**); see SubSection 3.5. In the last step, we perform spectral analysis over the SCM to extract symmetry clusters on the point cloud (**SpectralAnalysis**); see SubSection 3.6.

Algorithm 1. Symmetry detection via electors voting.

REQUIRE Input a point cloud $\mathcal{P}$ and its curve skeleton $\mathcal{S}$.
ENSURE Output symmetry regions $\Omega = \{C_i | i = 1, \ldots, n\}$.
$\mathcal{E} \leftarrow$ **SymmetryElectors**$(\mathcal{S})$; /* construct symmetry electors */
$\mathcal{P}_{node} \leftarrow$ **VoteNodePairs**$(\mathcal{E})$; /* vote for node pairs */
$\mathcal{M} \leftarrow$ **BuildSymCorrMat**$(\mathcal{P}_{node})$; /* build SCM from node pairs */
$\Omega \leftarrow$ **SpectralAnalysis**$(\mathcal{M})$; /* spectral analysis on SCM */

Algorithm 2. Symmetry electors selection.

$\mathcal{E} =$ **SymmetryElectors**$(\mathcal{S} = \{n_i | i = 1, \ldots, N\})$
$\mathcal{E} = \varnothing$.
for $n_i, n_j \in S$ **do**
  /* filters below executed *sequentially* */
if Similarity-Filter(n_i, n_j) passed **and**
  Branch-Filter(n_i, n_j) passed **and**
  Dist-Profile-Filter(n_i, n_j) passed
  $\Sigma_{ij} \leftarrow$ **Symmetry-Support**$(\{n_i, n_j\})$
  if $|\Sigma_{ij}| > p_d \cdot \binom{N}{2}$ **then**
    /* filter of symmetry support */
    $E = \{n_j, n_j\} \cup \Sigma_{ij}$;
    $\mathcal{E} \cup = \{E\}$;
    end if
  end if
end if
return $\mathcal{E}$.

3.3. Selection of symmetry electors

Since computing intrinsic symmetry map directly on point cloud is prohibitive, we opt to do so on its skeleton. Specifically, we seek for a set of symmetry correspondences over the skeleton nodes, which we call *symmetric node pairs*. However, detecting symmetric node pairs is difficult since skeleton is a reduced representation of the input shape which may lose the important geometric information useful to symmetry detection.

To this end, we adopt the electors voting scheme [23]. A symmetry elector contains a set of pairs of skeleton nodes, which represents a potential intrinsic symmetry of the input point cloud. Since an individual elector may not contain full information about the intrinsic symmetry of the point cloud, we let the electors cast votes to all node pairs and extract the symmetry correspondence from those node

pairs receiving sufficient supports from the electors (Section 3.4).

Specifically, we compute symmetry electors through filtering the bad correspondences according to a serial of intrinsic symmetry criteria tailored for skeleton nodes. Starting from a pair of nodes, we test it against four carefully designed symmetry filters: Similarity, Branch, Distance-Profile and Symmetry-Support. The four filters are applied in the ascending order of computational cost, for the efficiency consideration; see also Algorithm 2.

3.3.1. Similarity filter

The first filter compares the isometry-invariant local feature computed around the two nodes. Specifically, we compute the approximate Shape Diameter Function (SDF) [24] for the skeleton nodes. Most existing approaches to skeleton extraction naturally provide skeleton-surface mapping [12]. Thus the approximate SDF of a given skeleton node can be simply computed as the averaged Euclidean distances from the node to all its associated surface points according to the skeleton-surface mapping.

A pair of nodes passes the filter if the difference between their approximate SDF values is less than a threshold. We use $0.05SDF_{max}$ where SDF_{max} is the maximum SDF value of all skeleton nodes. To speed up the online comparison, we can pre-compute and store the approximate SDF values for all skeleton nodes.

3.3.2. Branch filter

Roughly speaking, a skeleton branch can be seen as the rotational symmetry axis of local geometry. This observation has been exploited by Tagliasacchi et al. [9] for curve skeleton extraction from point clouds. Knowing the skeleton of a point cloud, it is trivial to recover the local rotational symmetries based on the skeleton-surface mapping.

In this work, we wish to extract more prominent and nontrivial symmetries, e.g., the left–right reflectional symmetry of a human body. Therefore, we can filter the node pairs whose two nodes are from the same skeleton branch. To this end, we detect branches through extracting all junction nodes (with at least three incident bones) and tracing the branches one by one between every two adjacent junction nodes. The branch detection can be done off-line prior to the online filtering.

3.3.3. Distance-Profile filter

For a shape with global intrinsic symmetry, a pair of symmetric points should have similar configuration of intrinsic distances from the point to all other points, up to a symmetric permeation. If the distances are sorted into a vector, we obtain a more global feature, the distance profile, to filter erroneous point pairs.

This observation can be extended to skeleton nodes since skeleton is isometry invariant. Given two nodes, we can compute a similar vector to test if they are symmetric according to the global intrinsic symmetry of the input shape. Specifically, for each of the two nodes, we compute all distances from that node to all terminal nodes (with only one incident bone) and sort them in the ascending order, forming a N_t-dim vector (N_t is the number of terminal nodes). We then compare the two nodes by computing the L2 difference between their distance profile vectors. The node pair passes this filter if the difference is less than a prescribed threshold, i.e. 0.1 times the maximum profile difference between all pairs of nodes.

3.3.4. Symmetry-Support filter

The fourth filter examines whether the current node pair indicates a real intrinsic symmetry through testing whether the symmetry support for that node pairs is sufficiently strong. The intrinsic symmetry support between two pairs of nodes is defined similar to the symmetry criteria of Xu et al. [21]. Given two pairs of nodes, $\{n_p, n_q\}$ and $\{n_s, n_t\}$, we test whether the two pairs are supporting the same intrinsic symmetry (or say another way, they are supporting each other) with the following necessary condition of intrinsic involute symmetry

$$d_{\mathcal{S}}(n_p, n_s) = d_{\mathcal{S}}(n_q, n_t) \text{AND} d_{\mathcal{S}}(n_p, n_t) = d_{\mathcal{S}}(n_q, n_s), \tag{1}$$

where $d_{\mathcal{S}}$ is geodesic distance on skeleton. Fig. 2 illustrates the criteria on the skeleton nodes. If the number of supporting pairs a node pair received reaches a fraction p_d of $\binom{N}{2}$, the node pair passes the filter. We use $p_d = 0.01\%$ for all our experiments.

Once passing all the four filters, the current node pair, as well as its supporting node pairs, form a new symmetry elector. For each elector, we can measure its degree of symmetry by computing its intrinsic deviation defined based on the symmetry criteria (1):

$$\mathcal{D}_{E_i} = \frac{\sum_{\{n_p,n_q\},\{n_s,n_t\}\in E_i} d(\{n_p, n_q\}, \{n_s, n_t\})}{\binom{N_{E_i}}{2}}, \tag{2}$$

where N_{E_i} is the number of node pairs in elector E_i. The distance between two node pairs is defined as

$$d(\{n_p, n_q\}, \{n_s, n_t\}) = \max\{d_{\mathcal{S}}(n_p, n_s) - d_{\mathcal{S}}(n_q, n_t), d_{\mathcal{S}}(n_p, n_t) - d_{\mathcal{S}}(n_q, n_s)\}.$$

The node pairs from all the electors are referred to as *elector node pairs*, denoted as $\Pi_e = \cup_{E_i \in \mathcal{E}} E_i$. In Fig. 3, we show the effects of the different filters on the node pairs. With more filters applied, the node pairs left better reveal the intrinsic symmetry of the input shape.

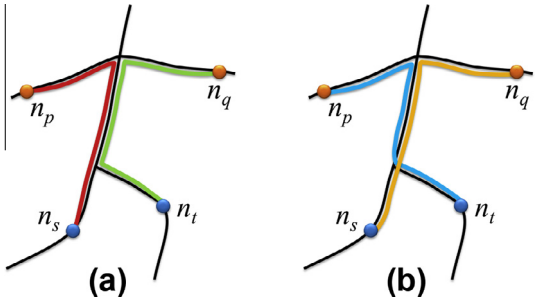

Fig. 2. Illustration of symmetry criteria on skeleton nodes.

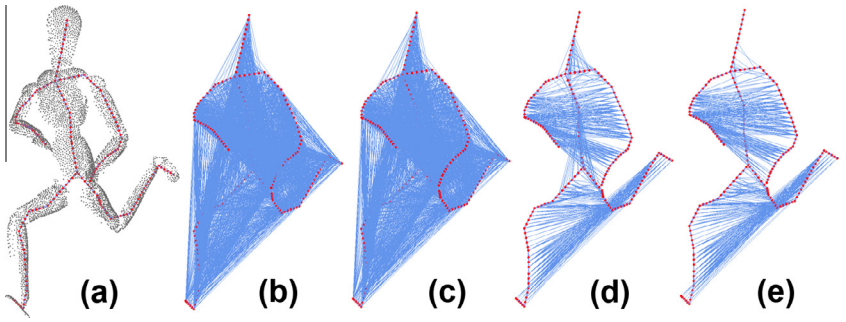

Fig. 3. The four symmetry filters at work. (a) The input point cloud and the skeleton. The figure shows the symmetric node pairs (indicated as blue lines) after the four filters are applied progressively: Similarity filter (b), Branch filter (c), Distance-Profile filter (d) and Symmetry-Support filter (e). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4. Electors voting for symmetric node pairs

We have selected a set of symmetry electors indicating potential symmetries. To accentuate the node pairs supporting the prominent intrinsic symmetry, we let the electors vote for the individual node pairs. The process is rather simplistic. The votes are casted into a $N \times N$ vote matrix as demonstrated in Fig. 4, where each entry corresponds to a node pair. Finally, those node pairs (corresponding to the entries in the vote matrix) whose value is greater than a threshold (0.1 times the maximum value in the vote matrix) is selected as *symmetric node pairs*, which will be use to find symmetry correspondence on the point cloud in the next subsection.

3.5. Symmetry correspondence matrix for point cloud

Since our ultimate goal is to find symmetry map on the point cloud, we need to transfer the symmetry map found on the skeleton, i.e., the symmetric node pairs to the sample points on the input point cloud, leading to a symmetry correspondence matrix (SCM) for the input point cloud.

Symmetry correspondence matrix is a symmetry affinity matrix where each entry measures the dissimilarity between the corresponding two sample points in the sense of symmetry, or say another way, the degree to which the two sample points are asymmetric [20]. We compute SCM for the sample points through interpolating the symmetry dissimilarity of the skeleton nodes which is computed as follows. Given a pair of skeleton nodes, we first collect all the electors that contains the node pair. Since we have computed an intrinsic deviation for each elector, the symmetry dissimilarity between a pair of nodes is simply the minimal intrinsic deviation among all the electors containing the node pair:

$$\mathcal{D}_{\{n_p,n_q\}} = \min_{\{n_p,n_q\}\in E_i} \{\mathcal{D}_{E_i}\}.$$

We then convert the symmetry dissimilarity of the node pairs to that of the sample point pairs through linear interpolation. To this end, we first compute for each skeleton node n_p a neighborhood η_p containing a set of k sample points using k-Nearest Neighbor (kNN) search. Given a pair of sample points $\{x_i, x_j\}$, we collect its *relevant node pairs* into Γ_{ij}.

$$\Gamma_{ij} = \{\{n_p, n_q\} | \{n_p, n_q\} \in \Pi_e \wedge (x_i \in \eta_p \wedge x_j \in \eta_q) \vee (x_j \in \eta_p \wedge x_i \in \eta_q)\},$$

where $\wedge$ and $\vee$ denote relation AND and OR respectively. Intuitively, an node pair is relevant to a point pair if it is an elector pair and each of its two neighbors contains ex-

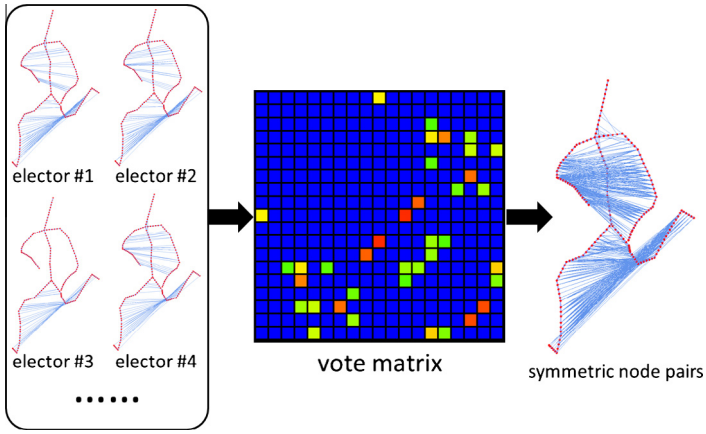

Fig. 4. Electors voting for symmetric node pairs (indicated as blue lines). Each entry of the vote matrix corresponds to a node pair. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

actly one of the two points in an exclusive manner. The symmetry dissimilarity of a point pair can be computed as the linear blending of the deviations of its relevant node pairs:

$$\mathcal{D}_{\{x_i,x_j\}} = \sum_{\{n_p,n_q\}\in\Gamma_{ij}} w_{pq}\mathcal{D}_{\{n_p,n_q\}}. \quad (3)$$

The linear blending weight is computed as:

$$w_{pq} = \left(1 - \tilde{d}(\{x_i,x_j\},\{n_p,n_q\})\Big/\tilde{d}_{ij}^{\max}\right)^2,$$

where $\tilde{d}$ is the distance between a point pair and its relevant node pair

$$\tilde{d}(\{x_i,x_j\},\{n_p,n_q\}) = \begin{cases} \max\{\|x_i-n_p\|_2,\|x_j-n_q\|_2\} \\ \quad \text{if } x_i\in\eta_p \quad \text{and} \quad x_j\in\eta_q, \\ \max\{\|x_i-n_q\|_2,\|x_j-n_p\|_2\} \\ \quad \text{if } x_i\in\eta_q \quad \text{and} \quad x_j\in\eta_p, \end{cases}$$

and $\tilde{d}_{ij}^{\max}$ is the maximum distance between point pair $\{x_i,x_j\}$ and all its relevant node pairs. Note that there is the case where a point pair may not have a relevant node pair. In this case, we simply set the symmetry dissimilarity of that point pair as infinite.

The symmetry dissimilarity between any pair of points is then converted to a symmetry correspondence matrix (SCM) $C\in R^{M\times M}$ using a Gaussian kernel $C_{ij} = e^{-(\mathcal{D}_{\{x_i,x_j\}}/(\sigma\cdot\mathcal{D}_{\max}))^2}$,where we use $\sigma = 0.01$, and $\mathcal{D}_{\max}$ is the maximum value of all $\mathcal{D}_{\{x_i,x_j\}}$. With the SCM, we perform spectral analysis to extract the symmetry regions [20,21], which we briefly review for completeness.

3.6. Symmetry extraction through spectral analysis

We perform eigendecomposition over the SCM and compute the symmetry factored embedding (SFE) for a sample point as

$$SFE^t(x_i) = (\lambda_1^t\psi_1(x_i),\ldots b,\lambda_n^t\psi_n(x_i)),$$

where ψ_k is the kth eigenfunction and λ_k the associated eigenvalue. We take the diffusion time $t = 20$ for all examples in this paper. The symmetry factored distance can be defined as the Euclidean distance within the space of SFE:

$$SFD^t(x_i,x_j)^2 = \sum_{k=1}^{n}\lambda_k^{2t}|\psi_k(x_i)-\psi_k(x_j)|.$$

With the SFE and SFD in hand, we can perform the standard K-means clustering in the space of SFE. To automatically determine the number of clusters, we employ the self-tuning spectral clustering [25], where the eigendecomposition of the symmetry scale matrix is computed with ARPACK [26]. After clustering, the set of sample points is divided into k clusters where each represents a symmetry orbit. Fig. 5 demonstrates the result of spectral clustering on the input point cloud based on SCM.

4. Results and discussion

In this section, we demonstrate and discuss results obtained by our skeleton-based intrinsic symmetry detection algorithm on a variety of imperfect point clouds. We also compare our detection method with other ones including those both for meshes and for point clouds. As a major application of our method, we demonstrate how the detected symmetries can be utilized to fill the missing regions of the incomplete point clouds.

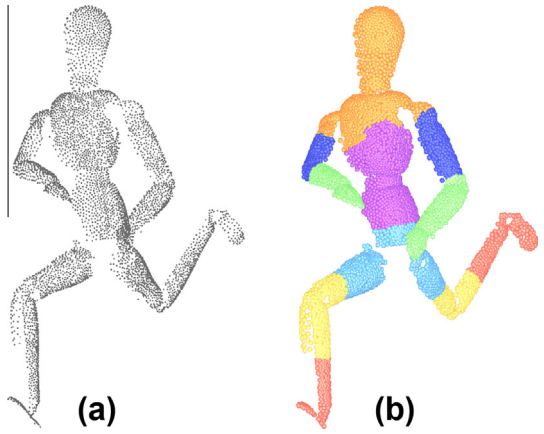

Fig. 5. The result of spectral clustering (b) on the input point cloud (a).

4.1. Intrinsic symmetry detection

The ability of our method to identify intrinsic symmetry from the input point clouds is evident. Figs. 1 shows the detection result on a raw scan. In Fig. 6, we show the results on a gallery of eight point clouds. For each model, we show the input point cloud, the extracted skeleton, as well as the symmetric node pairs on the skeleton. The detected symmetries on the point cloud are shown as a set of self-symmetric regions, as well as a set of pairs of symmetric regions, if any. Each of the symmetric regions is shaded with a unique color. It should be noted that for each model, the symmetric regions together constitute the global intrinsic symmetry of the model. Say another way, they share the same global intrinsic symmetry of the input point cloud. Our method cannot detect partial symmetries such as the self-symmetries of the human body and the trident of the Neptune model shown in Fig. 15a. On the Giraffe model in the gallery, there are a few node pairs connecting the head and feet of the giraffe. Our method failed to filter these erroneous pairs since the head and feed part have similar local thickness (SDF) and the related branches have almost the same length. However, these sparse wrong pairs did not affect the clustering result as shown in the figure.

4.2. Pose invariance

By using the consistent skeletons [27] extracted for point clouds of the articulated shape in different poses, our method returns consistent symmetry detection results for different poses. Fig. 7 demonstrates the effectiveness of our method on consistent symmetry detection on a set of scans of an articulated shapes captured in different poses, due to its intrinsic nature. This feature is potentially useful for cross-completion between different frames of an ani-

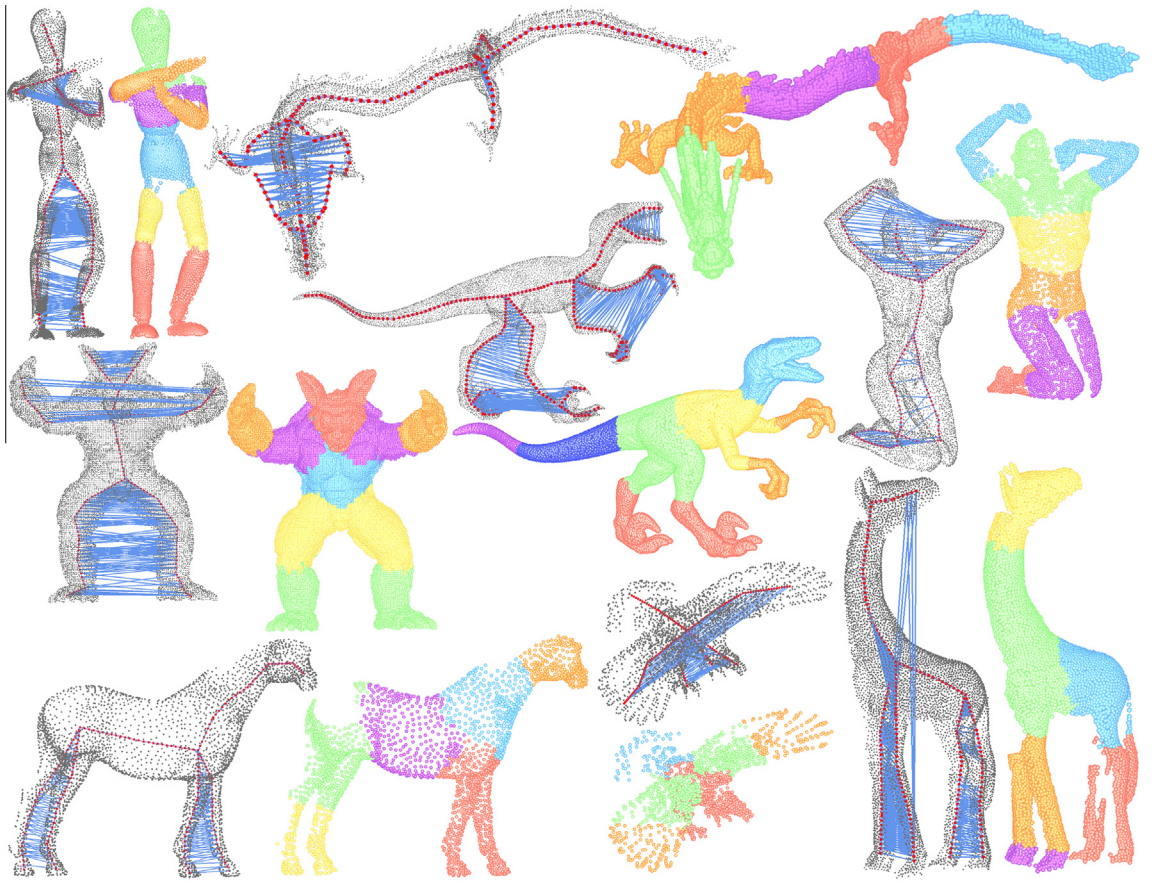

Fig. 6. A gallery of 3D symmetry detection results on eight point clouds. From top-left to bottom-right: Wooden doll, Dragon, Armadillo, Raptor, Woman, Horse, Eagle, and Giraffe. For each model, we show the input poind cloud, the extracted skeleton, the detected symmetric node pairs indicated as blue lines, as well as the detected symmetries as colored regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

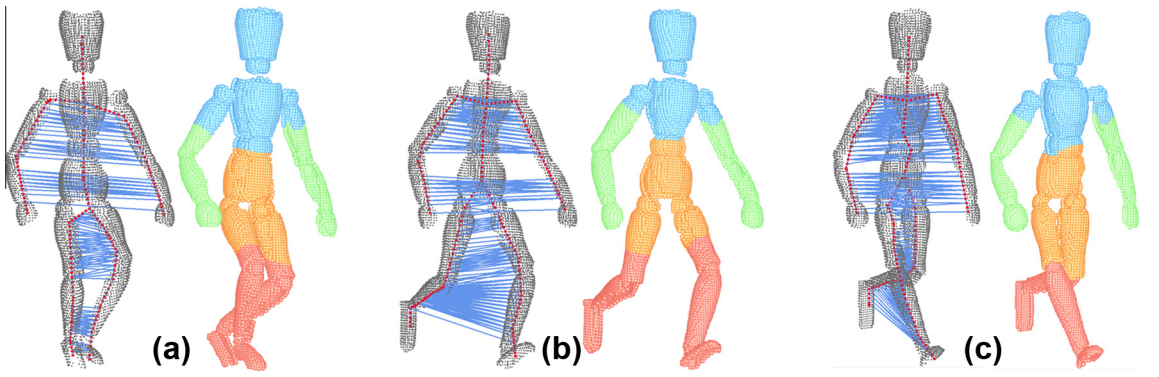

Fig. 7. Pose-invariant symmetry detection for an articulated Wooden doll. Our method returns stable results for the three different poses.

mated mesh sequence, which can be achieved by a trivial extension of the hole filling for single meshes proposed in SubSection 4.6.

4.3. Handling of low-quality and imperfect data

Most existing approaches to intrinsic symmetry detection rely on geodesic distance computed on the input surface. The imperfection of real world data, especially raw scans, often makes the estimation of geodesic distance prohibitive. In this case, extracting a skeleton first would help much in symmetry detection.

For the single-frame low-quality data captured by a Kinect, our algorithm produces reasonable results. In Fig. 8, we show our method can detect symmetries from a raw scan obtained by a Kinect camera. Although in very low

quality, the Kinect scan of a human body comes with a pre-computed skeleton which can be utilized by our method for symmetry detection. To the best of our knowledge, our method is the first one that detects intrinsic symmetry from such low quality point clouds.

Fig. 9 demonstrates the symmetry detection results on a Camel model with the number of scans is progressively reduced. The skeletons are extracted using the method of Tagliasacchi et al. [9]. Our method can detect the symmetry regions reasonably even for a partial scan with a large portion of missing data. When the data miss significantly and the global symmetry is broken, our method breaks down as expected (Fig. 9d).

In Fig. 10a–c, we demonstrate the robustness of our method against uniformly distributed Gaussian noise. Our method runs mainly on skeleton, the major step that relies on the point cloud is skeleton extraction. Base on the skeletons extracted by the noise-insensitive methods such as [12], our symmetry detection results are quite stable. The other two components of our algorithm which depend on point cloud are the computation of SDF for point cloud and the transfer of symmetry map from skeleton to point cloud. The latter does not affect the detection of symmetric node pairs and hence does not affect the symmetry detection significantly even if the noise is non-uniformly distributed. However, it may not be reliable to use SDF for node pair filtering in the case of non-uniform noise. For example, in Fig. 10d, the legs of the horse have different levels of noise. For this example, we increase the threshold of the Similarity filter (using $0.1SDF_{max}$) and our method still produces reasonable result.

4.4. Timing and statistics

Our experiments were performed on an Intel (R) Core (TM) Quad CPU 2.4 GHz machine with 2 GB RAM. For all 3D models, the number of sample points is about 2 K with small discrepancies due to criteria employed by the MeshLab sampling routine. Table 1 reports various statistics including timing. The most time-consuming parts are the construction of SCM and the symmetry extraction through spectral analysis.

4.5. Comparison

We compare our method to those both for meshes and point clouds. Our method naturally extends to surface meshes. In Fig. 11, we qualitatively compare the symmetry detection results on the Momento model with the other two methods for global intrinsic symmetry detection on surface meshes, i.e. [3,20]. Quantitative evaluation of the quality of symmetry detection results is out of the scope of this paper which we leave for future work. It can be seen that our method produces similar result to [20] where the global symmetry between the three human bodies are detected. Again, our method does not detect the partial symmetries of each human as is done by [3].

To the best of our knowledge, the only existing work that detects intrinsic symmetry on point clouds is proposed by Berger and Silva [22]. We make a comparison with their method on both a full and a partial scan of the Camel model (Fig. 12). In [22], although the medial diffusion is defined based on media structure, the method does not extract a skeleton explicitly. Therefore, we do not show the skeleton but only the symmetric maps (point pairs) found by each method. On the full scan (a), the two methods produce similar symmetry maps which mainly appear on the leg parts. On the partial scan (b), our method produces more accurate symmetry maps on the forelegs. The symmetry of the hindlegs is broken due to the large portion of missing data. Our method does not produce symmetry maps for the torso since the symmetric pairs are computed only on skeleton.

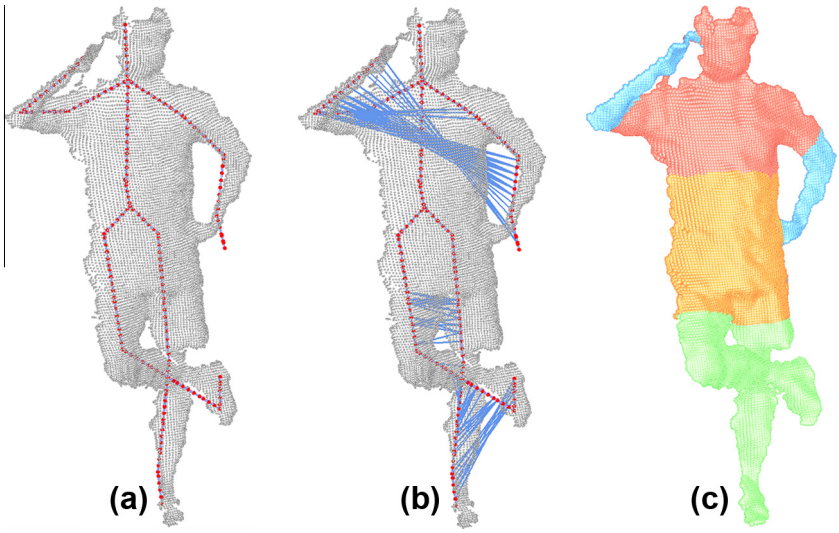

Fig. 8. Intrinsic symmetry detection results on a raw Kinect scan. (a) Input scan and skeleton. (b) Detected symmetric node pairs. (c) Extracted symmetry regions.

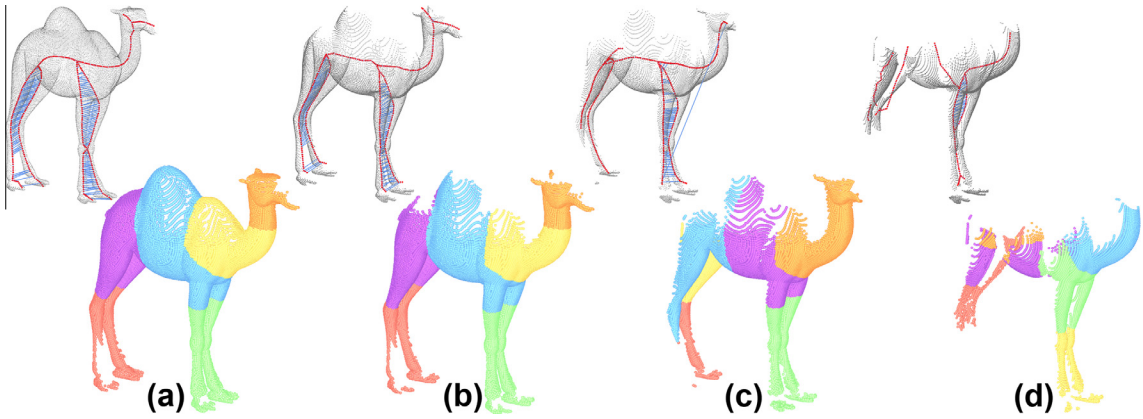

Fig. 9. Symmetry detection results on a Camel model with the virtual scans are progressively added. From (a) to (d), the number of scans are 5, 3, 2 and 1, respectively.

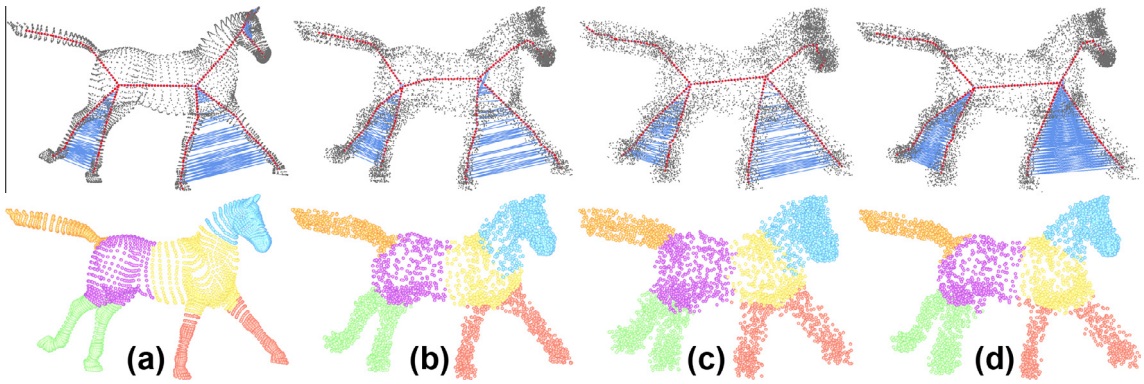

Fig. 10. Robustness against random noise. (a) Symmetry detection results on the original point cloud. (b and c) Results on the point cloud with different levels (0.2 and 0.5) of synthetic Gaussian noise added. The noise level is the ratio of the average vertex displacement over the average edge length in the original meshed model. (d) By relaxing the Similarity filter, our method still works when the global symmetry is broken by the severe non-uniform noise.

Table 1
Various statistics from our experiments. #Samp. denotes the number of sampling points; Running times are reported in seconds for skeleton refinement in the preprocessing step ("Preproc."), symmetric pairs electors voting ("Vote"), SCM matrix building ("SCM"), and symmetry extraction through spectral analysis ("Spect.").

Models	#Samp.	Preproc.	Vote	SCM	Spect.
Armadillo	2243	0.5	2.4	45.1	64.2
Camel	2182	0.4	2.4	44.7	43.7
Dragon	2491	0.2	1.5	59.3	68.0
Eagle	2775	0.3	3.1	77.3	79.0
Giraffe	2042	0.5	5.0	36.8	37.3
Horse	2149	0.3	2.7	42.4	44.8
Wooden doll	2826	0.4	2.3	79.3	107.3
Momento	2517	0.8	8.1	57.1	59.8
Raptor	2484	1.3	18.5	56.8	61.4
Woman	1458	0.4	3.2	16.3	15.7

In order to reveal the effect of skeletons on the symmetry detection results, we evaluate our method on the skeletons extracted by different methods; see Fig. 13. On the two scans of the Camel model, we extract skeletons using three methods including those from [9,10,12], and for each skeleton we run our method to detect the symmetry node pairs as well as the symmetry regions. Our method works well with all the three types of skeletons while producing the most accurate results for the high quality curve skeletons produced by [9].

4.6. Application: filling the missing regions

Symmetry induces redundancy. Such redundancy allows for filling the input point clouds in the presence of missing parts. In Fig. 14, we demonstrate the application of our symmetry detection method in the completion of imperfect point clouds. In a preliminary implementation, we employ a straightforward method where we utilize the detected symmetric node pairs, as well as the skeleton-to-surface mapping, to perform surface completion. Specifically, we transform the surface points associated with one of a pair of symmetric nodes along the skeleton to align them with those associated with its counterpart.

Given an input point cloud with missing data, we first let the user to identify the holes interactively using surface painting. Then our system automatically detects the relevant symmetric node pairs around the hole regions. Based on the node pairs, hole filling is performed through transforming the surface points associated with one node to the

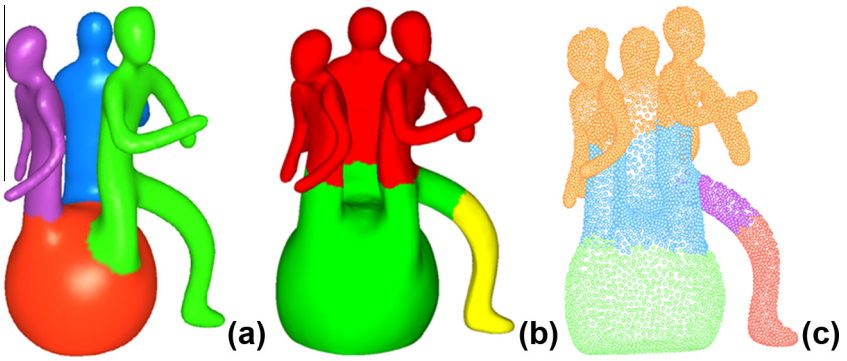

Fig. 11. Comparison of intrinsic symmetry detection on the Momento model (surface mesh) with different methods: (a) Xu et al. [3], (b) Lipman et al. [20] and (c) Ours.

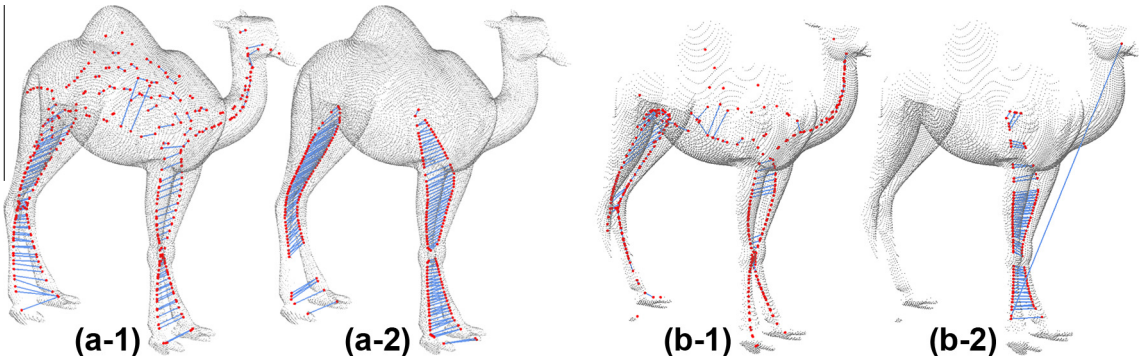

Fig. 12. Comparison of symmetry detection on point clouds between (1) our method and (2) the method of Berger and Silva [22]. For each point cloud, the detected symmetric point pairs (not on the surface) are shown.

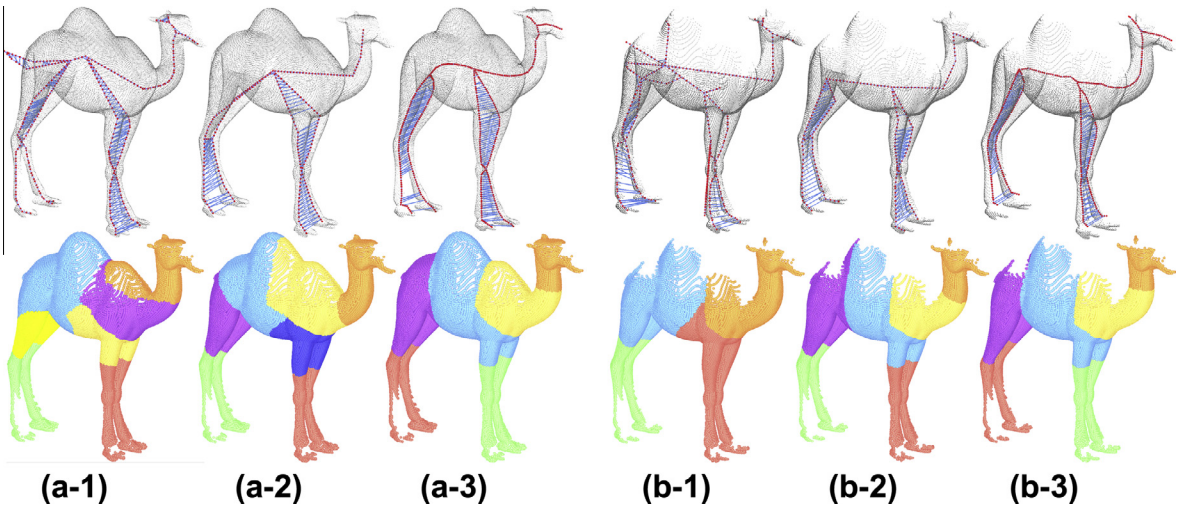

Fig. 13. Comparison of intrinsic symmetry detection results on two point clouds (a and b) using different skeleton extraction methods by: (1) Cao et al. [10], (2) Jiang et al. [12], and (3) Tagliasacchi et al. [9].

other. Suppose we want to complete a hole based on a symmetric node pair $\{n_s, n_t\}$. Without loss of generality, let us assume that we use the surface points associated with n_s, denoted as set $\mathcal{P}_s$, to complete the surface around n_t, denoted as $\mathcal{P}_t$. We first perform a mirror transformation for $\mathcal{P}_s$ along any plane whose normal is perpendicular to the tangent of the curve skeleton at node n_s, resulting in $\mathcal{P}'_s$. $\mathcal{P}'_s$ is then transformed along the geodesic path on the

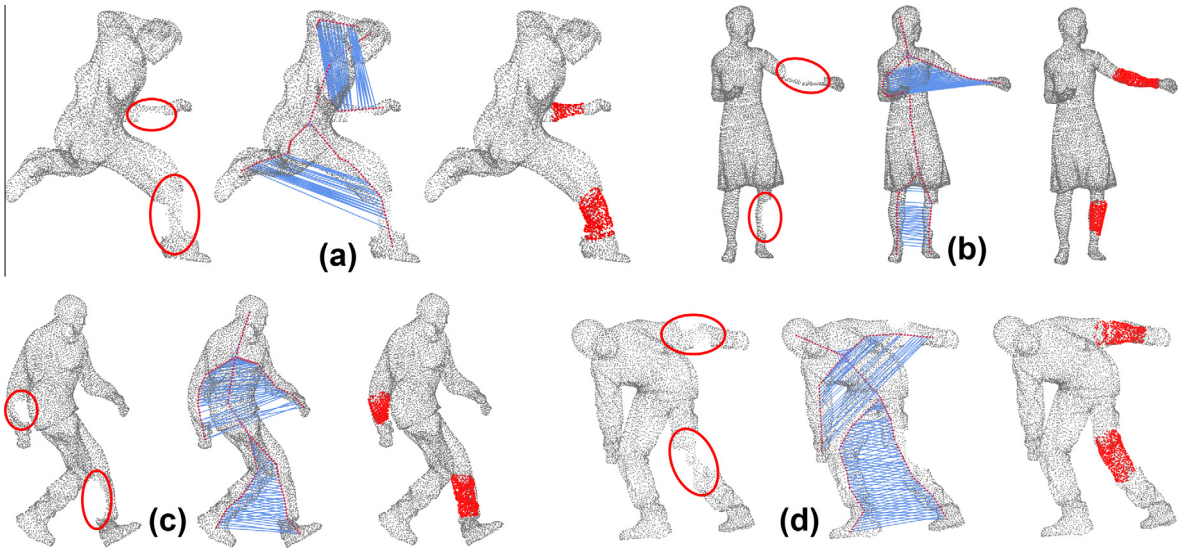

Fig. 14. For hole filling examples. For each example, the input scans containing missing parts (marked in red circles) is shown in the left, the detected symmetric node pairs on the skeletons in the middle and the pair-wise filling result in the right (newly filled point sets are colored in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

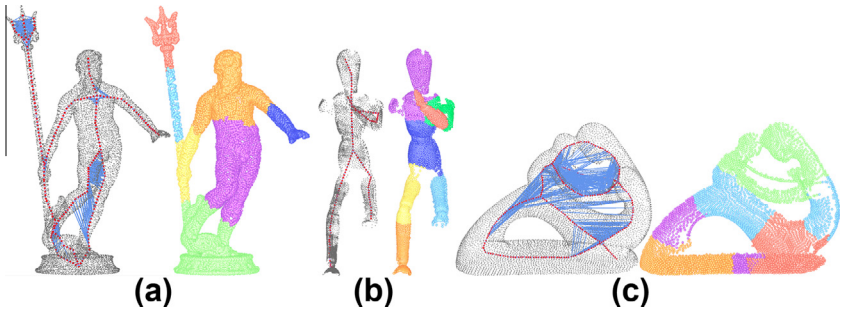

Fig. 15. Three failure cases. (a) Our method fails to detect the two self-symmetric regions of the Neptune model: the human body and the trident. (b) Our method fails to detect the partial symmetries of the incomplete Wooden doll model. (c) Our method fails since the model contains too few terminal nodes. For each example, the symmetric node pairs are shown in the left and the clustering result in the right.

skeleton connecting n_s and n_t in a rotation minimizing fashion [28], leading to a rough alignment between the two $\mathcal{P}'_s$ and $\mathcal{P}_t$. If a surface point is associated with multiple skeleton nodes, its final position is computed as the weighted average of the target positions transformed by each of the skeleton nodes, using a Gaussian kernel. Finally, we refine the alignment using the Iterative Closest Point (ICP) method [29] with the help of the partial overlap between the two point sets. As shown in Fig. 14, our method can robustly fill the small holes in the point clouds of human bodies taking various poses.

4.7. Limitations

Perhaps the most fundamental limitation of our approach is its reliance on geodesic distances on skeleton, which is topologically sensitive. In the case of a composite shape with several self-symmetric objects, as shown in Fig. 15a, the partial symmetries on each object cannot be detected correctly. This is a fundamental limitation of any approach that relies on intrinsic distances for shape analysis.

In Fig. 15b, we show another type of partial symmetry where some parts of the Wooden doll model are missing. Although the global symmetry is broken, partial symmetries (such as the upper parts of the two legs) remain. Our method failed to detect such symmetries due to the Distance-Profile filter we employed in the node pair detection: The missing parts cause the skeleton has different lengths of branches. Furthermore, our method cannot detect symmetry for shapes whose skeleton contains too few terminal nodes (only one for the example in Fig. 15c) so that the Distance-Profile filter cannot be applied.

Another limitation of our method is the dependence on the topologically correct skeletons. Therefore our method must work with an existing method for point cloud skeleton extraction. However, our method can seamlessly work with a Kinect system where the RGBD image of a human

body is fitted with a pre-computed skeleton in real time, showing the potential in augmenting the low quality scans of Kinect.

5. Conclusions

We have presented a novel and practical algorithm for intrinsic symmetry detection on imperfect point clouds based on accurate curve skeletons. The various experiments demonstrate the robustness of our method in terms of handling raw point clouds with severe noise, undersampling, and missing data. The detected symmetry can be applied to repair the scanned data by filling the missing regions.

Our current algorithm and implementation still leave much room for improvement. First and foremost, we would like to investigate how to detect symmetric node pairs for the independent partial intrinsic symmetries appearing on a shape, e.g. the Neptune model in Fig. 15. Another interesting venue is finding more efficient filtering mechanisms for the symmetry electors tailored for curve skeletons. We would also like to improve the performance of our algorithm, especially for the construction of SCM. Parallelization is obviously possible since the entries in the SCM can all be computed in parallel. Finally, it is desirable to more intelligently combine the symmetry criteria (Eq. (1)) with the conventional measures used in mesh segmentation problem, and develop an optimization-based framework for symmetry detection.

Acknowledgments

We would like to thank Andrea Tagliasacchi and Oscar Kin-Chung Au for sharing their code/executable on curve skeleton extraction. We are also grateful to Matthew Berger for producing the results in Fig. 12 using their method. Thanks also go to Junjie Cao and Qian Zheng for data sharing. This work was supported by NSFC (Nos. 60970094, 61103084, 61202333 and 61202334) and CPSF (No. 2012M520392).

References

[1] N.J. Mitra, M. Pauly, M. Wand, D. Ceylan, Symmetry in 3D geometry: extraction and applications, in: EUROGRAPHICS State-of-the-Art, Report, 2012, pp. 1–23.
[2] D. Raviv, A.M. Bronstein, M.M. Bronstein, R. Kimmel, Symmetries of non-rigid shapes, in: International Conference on Computer Vision, 2007, pp. 1–7.
[3] K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng, Y. Xiong, Partial intrinsic reflectional symmetry of 3D shapes, ACM Trans. Graph. 28 (5) (2009) 138:1–138:10.
[4] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, A. Schilling, Symmetry detection using line features, Comput. Graph. Forum 28 (2) (2009) 697–706.
[5] D. Raviv, A.M. Bronstein, M.M. Bronstein, R. Kimmel, Full and partial symmetries of non-rigid shapes, Int. J. Comput. Vision 89 (2010) 18–39. ISSN 0920–5691.
[6] M. Ovsjanikov, J. Sun, L. Guibas, Global intrinsic symmetries of shapes, Comput. Graph. Forum 27 (5) (2008) 1341–1348.
[7] F. Mémoli, G. Sapiro, Distance Functions and Geodesics on Point Clouds, IMA Preprint. <www.ima.umn.edu>, 2003.
[8] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, On-the-fly curve-skeleton computation for 3D shapes, Comput. Graph. Forum 26 (3) (2007) 323–328.
[9] A. Tagliasacchi, H. Zhang, D. Cohen-Or, Curve skeleton extraction from incomplete point cloud, ACM Trans. Graph. 28 (2009) 71:1–71:9.
[10] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, Z. Su, Point cloud skeletons via Laplacian-based contraction, in: Proc. Shape Modeling International, 2010, pp. 187–197.
[11] A. Bucksch, R. Lindenbergh, M. Menenti, SkelTre: robust skeleton extraction from imperfect point clouds, Visual Comput. 26 (10) (2010) 1283–1300. ISSN 0178-2789.
[12] W. Jiang, K. Xu, Z.-Q. Cheng, R.R. Martin, G. Dang, Curve skeleton extraction by coupled graph contraction and surface clustering, in: Computational Visual Media Conference, 2012.
[13] Y. Lipman, T. Funkhouser, Möbius voting for surface correspondence, ACM Trans. Graph. 28 (3) (2009) 72:1–72:12.
[14] V.G. Kim, Y. Lipman, X. Chen, T. Funkhouser, Möbius transformations for global intrinsic symmetry analysis, Comput. Graph. Forum (2010) 1689–1700.
[15] M. Ben-Chen, A. Butscher, J. Solomon, L. Guibas, On discrete killing vector fields and patterns on surfaces, in: Eurographics Symposium on Geometry Processing, 2010, pp. 1701–1711.
[16] A. Berner, M. Bokeloh, M. Wand, A. Schilling, H.-P. Seidel, Generalized Intrinsic Symmetry Detection, MPI Informatik Tech, Report MPI-I-2009-4-005, 2009.
[17] R. Lasowski, A. Tevs, H.-P. Seidel, M. Wand, A probabilistic framework for partial intrinsic symmetries in geometric data, in: Proc. Int. Conf. on Comp. Vis., 2009, pp. 963–970.
[18] N.J. Mitra, A. Bronstein, M. Bronstein, Intrinsic regularity detection in 3D geometry, in: Proc. Euro. Conf. on Comp. Vis., 2010, pp. 398–410.
[19] A. Berner, M. Wand, N.J. Mitra, D. Mewes, H.-P. Seidel, Shape analysis with subspace symmetries, Comput. Graph. Forum 30 (2) (2011) 277–286.
[20] Y. Lipman, X. Chen, I. Daubechies, T. Funkhouser, Symmetry factored embedding and distance, ACM Trans. Graph. 29 (4) (2010) 103:1–103:12.
[21] K. Xu, H. Zhang, W. Jiang, R. Dyer, Z. Cheng, L. Liu, B. Chen, Multi-scale partial intrinsic symmetry detection, ACM Trans. Graph. 31 (6) (2012) 1–10.
[22] M. Berger, C.T. Silva, Nonrigid matching of undersampled shapes via medial diffusion, Comput. Graph. Forum 31 (2012) 1587–1596.
[23] O.K.-C. Au, C.-L. Tai, D. Cohen-Or, Y. Zheng, H. Fu, Electors voting for fast automatic shape correspondence, Comput. Graph. Forum 29 (2010) 645–654.
[24] L. Shapira, A. Shamir, D. Cohen-Or, Consistent mesh partitioning and skeletonisation using the shape diameter function, Visual Comput. 24 (4) (2008) 249–259.
[25] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Proc. Advances in Neural Information Processing Systems (NIPS), vol. 17, 2004, pp. 1601–1608.
[26] R. Lehoucq, D. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, vol. 6, SIAM, 1998.
[27] Q. Zheng, A. Sharf, A. Tagliasacchi, B. Chen, H. Zhang, A. Sheffer, D. Cohen-Or, Consensus skeleton for non-rigid space-time registration, Comput. Graph. Forum 29 (2) (2010) 635–644.
[28] W. Wang, B. Jüttler, D. Zheng, Y. Liu, Computation of rotation minimizing frames, ACM Trans. Graph. 27 (2008) 2:1–2:18.
[29] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, Int. J. Comput. Vis. 13 (1994) 119–152.