Graphical Models 105 (2019) 101041

Contents lists available at ScienceDirect

Graphical Models

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

3D skeleton transfer for meshes and clouds

Çağlar Seylan, Yusuf Sahillioğlu*

METU, Dept. of Computer Engineering, Ankara, Turkey

ARTICLE INFO

Keywords:
skeleton transfer
skeleton extraction
3D shape abstraction

ABSTRACT

We present a curve-skeleton extraction method in the form of curve-skeleton transfer from the source shape to the target. The shapes we deal with need to be in correspondence. They are discretized as meshes embedded in 3D and are not necessarily watertight. They can even be in arbitrary topology, and furthermore be disconnected point clouds. Our method associates the source skeleton with the source shape and computes the optimal rigid transformations towards the corresponding query shape points. This transformation for each source skeleton point is followed by a post-processing operation. Fixed skeleton connectivity maintained throughout the transfer guarantees that the resulting skeleton is as complex as the source skeleton, e.g., no redundant branches and complicated joint hierarchies. The performance and versatility of our method as well as its advantages over a direct skeleton extraction method are demonstrated.

1. Introduction

The set of points having more than one closest point to the boundary of a shape is called the medial-axis or skeleton. Since its first introduction in [1], many of its variants have been developed and used in shape analysis, retrieval, deformation, and many other applications.

One problematic issue with the initial version of the medial-axis is being too sensitive to small variations on the boundary. Even with a boundary with small noise, the produced medial-axis is too complex to be processed or to be practical especially for 3D shapes because medial-axis of a 3D shape not only consists of curves, but also contains 2D surfaces. Such structures are unnecessarily complex for some applications. To promote usage of medial-axis, 1D curve representations of medial-axis, called curve-skeletons, have been introduced. In spite of their differences, the terms curve-skeleton and skeleton are used interchangeably in the literature. Throughout the paper, we refer to curve-skeleton by the phrase "skeleton".

Numerous methods have been proposed for computing medial-axes of shapes. In cases where there are several shapes in correspondence such that they represent different poses of the same entity at different times, e.g., animation sequences, some recent techniques like [2] allow transfer of medial-axis computed for one shape to other shapes. Analogously, many methods have been proposed for extracting curve skeletons of shapes, however, no study has been done specifically for transferring an already extracted curve-skeleton between shapes in correspondence. There exist common scenarios where this correspondence information is available, e.g., animation sequences, simulated models, or registered models. Computing the skeleton for each shape directly in such scenarios with the existing methods has three disadvantages. First, the extraction process for all the shapes may take a long time as each one is treated independently in isolation. Second, it is highly unlikely that the resulting skeletons will be registered to each other vertex-by-vertex, however, this missing one-to-one correspondence property may also be desirable. Third, many of the skeleton extraction methods require that the mesh, arguably the most common shape representation, is watertight, i.e., possess no holes, voids, or non-manifold structures, and complete, and represented by fine tessellations. Majority of these direct methods are also likely to suffer from producing artifacts such as redundant branches.

In this study, we propose a simple yet accurate method that enables curve-skeleton transfer between shapes in correspondence. To do this, we need an already extracted skeleton, which we refer to as the source skeleton throughout the paper. Firstly, for each source skeleton vertex, we associate a set of vertices of the mesh from which it is extracted. This mesh is called the source mesh throughout the paper. The correspondence process to achieve this association is explained in detail in Section 3.1. Subsequently, for each skeleton vertex, the optimum rigid transformation that maps its associated vertices to the corresponding point set of the target shape is computed separately. Note that, the target shape in our versatile method admit many representations but, without loss of generality, we assume mesh representation hereafter. Note also that this target mesh is also referred to as the input mesh since we want to extract the skeleton for this query mesh. Optimum transformation computation results in a rotation matrix and a translation vector for each skeleton vertex. The resulting transformations are applied to their skeleton vertices one-by-one, transferring the skeleton vertices to

* Corresponding author.
E-mail address: ys@ceng.metu.edu.tr (Y. Sahillioğlu).

https://doi.org/10.1016/j.gmod.2019.101041
Received 22 March 2019; Received in revised form 13 June 2019; Accepted 29 July 2019
Available online 30 July 2019
1524-0703/© 2019 Elsevier Inc. All rights reserved.

the input mesh. This transfer process is explained in Section 3.2. Finally, 1D curve skeleton is smoothed during the post-processing stage whose details are given in Section 3.3.

The study has two main contributions.

- A curve-skeleton transfer method which works quite fast, does not require skeleton computation for each target mesh from scratch, preserves connectivity and complexity of the source skeleton and works even if the target meshes have degeneracies up to some extent. Meshes are not necessarily watertight and in fixed topology, and furthermore can be disconnected triangle soups or point clouds.
- Two novel error metrics that allow us to measure the accuracy of centeredness property of a curve-skeleton inside of tubular structures quantitatively and to compare the results with other methods in a more systematic way. To our knowledge, no quantitative error metric was proposed in previous studies to assess centeredness property of curve-skeletons.

2. Related work

Unlike medial-axis, curve-skeleton is not formally defined. Although some attempts were made to mathematically define what a curve-skeleton is, e.g., Dey and Sun [3], these definitions have not been accepted widely by the community. Due to the lack of formal definition, there has not been an exact algorithm to compute the curve-skeleton of a shape. However, several properties that should be satisfied by a curve-skeleton is listed at least: Topology preserving, invariance under isometric transformations, reconstructability, thinness, centeredness, reliability, component-wise differentiation, connectedness, robustness, smoothness, and hierarchicalness. We will not explain what those properties mean for the brevity of the discussion, those who are interested in brief definitions and analysis of those properties are encouraged to read the survey [4]. Identification of the properties does definitely not imply that a curve-skeleton should satisfy all of them. In fact, it is impossible for a curve-skeleton to satisfy all of the properties because some of them are conflicting. For example, centeredness and robustness are conflicting properties which further implies a curve-skeleton does not have to be a subset of the medial-axis invalidating the definition in [3]. Instead of fulfilling all of the properties, non-conflicting and different subsets of them are satisfied for different applications. This has led to development of numerous techniques and algorithms to compute the curve-skeleton of a shape.

First computing medial-axis and then pruning it to obtain the curve-skeleton is one of the most widely used strategies for computing curve-skeleton of a shape, Li et al. [5] and Yan et al. [6] are recent examples to algorithms adopting this strategy. Low robustness to surface noise and computational expensiveness are two major drawbacks of such approaches. Some techniques to produce robust skeletons have also been developed. For example, a generalized potential field generated by charges placed on the surface of the object is used by Chuang et al. [7] in order to define a repulsive force that pushes the surface towards centering skeleton. Even though curve-skeletons produced by such such techniques are almost excellent in terms of robustness and smoothness, it is challenging to trace the field discontinuities due to numerical issues and their computational cost is quite high. Exploiting curve-skeleton extraction in 2D which is a relatively well-studied area is another strategy for 3D curve-skeleton extraction. For example in [8], the shape is first orthographically projected onto 2D scenes, its curve-skeleton is computed in 2D domain, and it is projected back to 3D. Although robust curve-skeletons can be computed with a low computational cost with this method, it may be out of the shape and a large number of projections may be required. Contraction-based methods have been becoming more popular in curve-skeleton extraction area. Tagliasacchi et al. [9] is a good representative of this family, which was also used in this study to extract the source skeleton. Major drawbacks of such methods are they usually require many parameters to tune, quite different skeletons can be generated for different discretization of the same surface, and they usually assume that the surface is watertight, possessing no holes. Other than these, many different methods were developed for curve-skeleton computation from probabilistic methods [10] to the ones using shape diameter function [11]. Curve-skeleton extraction methods are certainly not limited the mentioned ones, those who are interested are encouraged to read the surveys [4] and [12]. Both medial-axis and curve-skeleton computation has drawn noticeable attention up to now, and this trend will most probably continue in the future due to the diverse range of applications taking advantages of them.

One of the applications in which skeletons are quite useful is the compact encoding of animation sequences. A naïve approach that stores each animation frame model explicitly with the full mesh details would be too redundant and expensive in terms of streaming, storage, and processing. Extracting and rigging the skeleton of a single model is a much more efficient alternative for the faithful reproduction of the sequence poses [13–15]. These methods have been improved recently, at the expense of increased computational load, by automatic pruning of redundant bones [16] and introduction of deformable medial spheres [2]. The input to this family of methods is a set of example poses with temporal consistency, i.e., different poses with the same fixed connectivity. We should note that while most of the sequences come with this information naturally, e.g., meshes produced by animations, deformations, simulations, and registrations, one can also employ fully automatic dense correspondence algorithms to obtain this knowledge [17–21].

Another usage of skeletons is to represent static models efficiently for various purposes such as animation or deformation. Given a static character mesh and a generic template skeleton, for instance, Baran and Popović [22] adjusts the skeleton so it can fit into the mesh properly and then attaches it to the surface with the appropriate blend skinning weights that derive the animation afterwards. Electors voting approach [23] performs similarly for this so-called animation re-targeting task except it extracts the skeleton automatically instead of fitting a template. Regardless of how skeleton is placed in this family of animation re-targeting methods, one needs to bind the skeleton to the surface vertices with the appropriate blend skinning weights to enable the animation of the skeletonized object. This is either a manual burdensome process or an automated process up to some accuracy issues [24,25]. Note that, if the source mesh is weighted properly then the skeletonized object through our transfer is already equipped with this important weight information since the new skeleton comes with its binding to the new surface through perfect correspondences.

There are several other useful static shape representations through skeletons. Shape retrieval application, for instance, greatly benefits from skeletons while comparing the similarity between a query model and many database models without getting confused and slowed down by the geometric factors [4,26,27]. Sometimes, however, these geometric factors may be useful, e.g., skeletons of a horse and a camel are similar but they are geometrically different and should not be matched [28]. Consequently, one needs to keep this trade-off in mind while employing skeletons for this task.

Several applications like the ones mentioned above usually use a series of poses of the same model. A computationally lightweight method for computing curve-skeletons for all of the poses of a model such that the resulting curve-skeletons will be topologically equivalent and have the same number of skeletal elements is lacking. At this point, one may attempt to use medial-axis deformation based methods like [29] and [2]. Nevertheless, such methods are proposed for generating surfaces for animating mesh sequences and using them for curve-skeleton transfer comes with a few important disadvantages. For example, in the approach in [2], which was based on [30], the two stage optimization process is computationally costly. Even for a small number of medial primitives ($\sim$ 50), the computation takes dozens of seconds yet it takes less than a second with the method proposed here for a large number of skeleton vertices ($\sim$ 500) (See Section 4.9). Moreover, the proposed method allows curve-skeleton transfer when the input subject changes, e.g., transfer from a male to a female, or from a slim subject to a fat one (See Section 4.3). However, it is unclear how deformation based methods behave when used for such a purpose. As mentioned, medial-axis deformation methods and this study bring solutions to different problems. If the objective is to embed surface information into a medial structure and reconstruct the animated surface, then deformation methods like [2] should be favoured. Nevertheless, if the objective is to compute curve-skeletons for a series of poses of the same model, or even a different model up to some extend (See Section 4.3), in a very fast manner, it is advantageous to use our transfer approach.

3. Method

The method we propose consists of three stages. A correspondence between the source mesh and its skeleton is established in the first stage which we call as surface correspondence. Once the surface correspondence is established, the optimum rigid transformations transferring the skeleton of the source mesh to the possible skeletons of the other input meshes are estimated. Skeleton transfer stage is followed by a post-processing stage in which the estimated skeletons -which are 3D medial curves- are smoothed by using Laplacian smoothing.

3.1. Surface correspondence

In order to transfer to the input mesh, we need a source mesh and its skeleton, which can be created manually or computed by any skeleton extraction method. We should make sure the skeleton extraction method we are using can extract the skeleton as correct as possible. We used the Mean Curvature Flow (MCF) method proposed in [9] to extract the source skeleton from the source mesh. The source skeletons used in this study are shown in Fig. 1. Please note that the part of the extracted source skeleton in the right foot of the human mesh does not precisely satisfy the medial-axis property. Consequently, the transferred skeletons also do not satisfy the medial-axis property precisely in that region of the meshes (Section 4).

A correspondence between the source mesh and the source skeleton should be established so that the source skeleton can be transferred to the input meshes using this correspondence. We continue to provide details for this correspondence computation.

It is assumed that the source skeleton consists of piece-wise linear 1D curves wherein joints are represented as vertices (which we call as skeleton vertices) and if any two vertices are connected, they form a line segment. For the ith vertex of the source skeleton $\mathbf{s}_i = [s_{i_x}\ s_{i_y}\ s_{i_z}]^T$, we first determine the number of mesh vertices N_i with which we are going to establish correspondence. Let $(\mathbf{s}_i, \mathbf{v}_{i,1})$ be a pair of vertices where $\mathbf{v}_{i,1} = [v_{i,1_x}\ v_{i,1_y}\ v_{i,1_z}]^T$ is the closest vertex of the source mesh to $\mathbf{s}_i$ in terms of Euclidean distance. Let us denote the distance between $\mathbf{s}_i$ and $\mathbf{v}_{i,1}$ as r_i. We define $r_{\min}$ as the smallest of such distances for all the skeleton vertices, i.e., $r_{\min} = \min\{r_1, r_2, \ldots, r_N\}$ where N is the total number of vertices in the source skeleton.

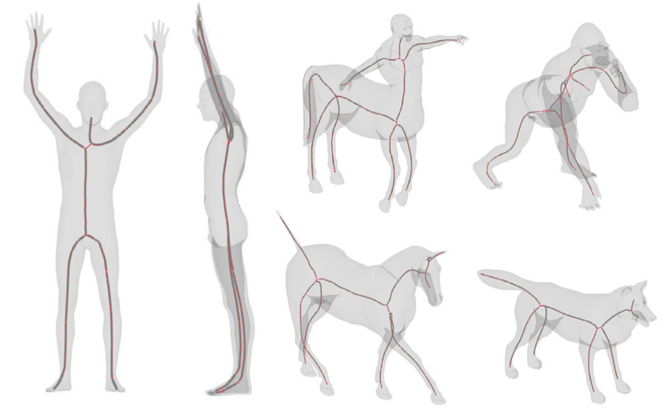


Fig. 1. Source skeletons to be transferred (human in two views).

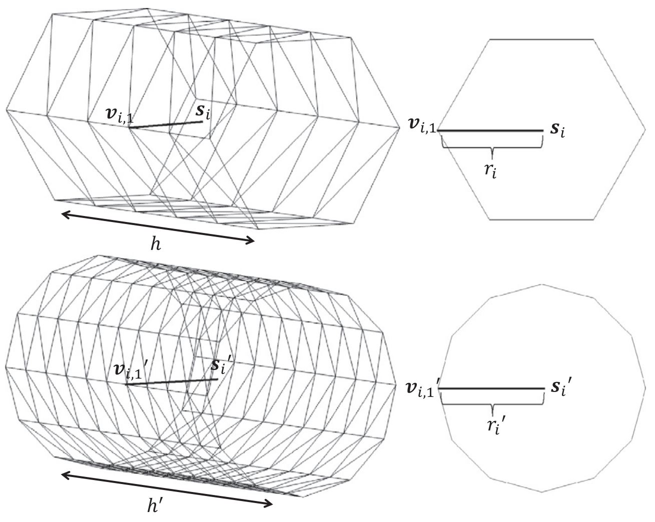


Fig. 2. Two cylinders represented as triangular meshes. The cylinder at bottom is thicker than the one at top, that is, $r_i' > r_i$, and also $h' \geq h$. Subfigures at right show the top views of the cross sections of the cylinders to better depict the setup. As r_i increases, the area of the cylinder increases non-linearly.

Let us consider a hypothetical case in which $\mathbf{s}_i$ is inside of a cylindrical shape with height h. $\mathbf{v}_{i,1}$, r_i, and the setup is depicted at the top row of Fig. 2. The top view of the cross section of the cylinder is shown at right to better see the setup. $\mathbf{s}_i$ is corresponded with a number of vertices chosen from the surface of the cylinder. At the bottom row of Fig. 2, a thicker cylinder is given such that $r_i' > r_i$. It is clear that the surface area of the cylinder increases with increasing r_i if $h' \geq h$. As the cylinder gets thicker, h' tends to be larger than h implying that the surface area non-linearly depends on r_i. Consequently, N_i should be increased non-linearly with increasing r_i, otherwise, lacking number of corresponding vertices leads to suboptimal skeleton transfer results. Besides, it is not a proper way to directly use Euclidean distance between $\mathbf{s}_i$ and $\mathbf{v}_{i,1}$ while deciding N_i because the scales of meshes may differ. Using the ratio $\rho_i := r_i / r_{\min}$ instead of r_i by-passes this problem. This leads us to choose N_i as

$$N_i = \begin{cases} 8 + \lfloor 8\rho_i \rfloor, & \text{if } \rho_i \leq 3 \\ 8 + \lfloor 12\rho_i \rfloor, & \text{if } 3 < \rho_i \leq 6 \\ 8 + \lfloor 16\rho_i \rfloor, & \text{if } 6 < \rho_i. \end{cases} \quad (1)$$

When we do not use any constant term and ρ_i is close to 1, N_i becomes small for a robust transformation estimation. Thus, we added a constant term to the equation. To reflect the non-linearity between ρ_i and N_i mentioned at the previous paragraph, we divided the equation into three cases and we increase the coefficient of ρ_i with increasing ρ_i. The value of the initial term which is 8 and the coefficients of ρ_i which are 8, 12, and 16 are determined through various experiments covering many different cases. Please note that there can exist different procedures for determining N_i, the one presented in this study produces fairly accurate results which will be discussed thoroughly in Section 4.

Determination of N_i is followed by choosing N_i vertices on the mesh. It is better to choose the corresponding vertices on the rigid body part of the source mesh to which $\mathbf{s}_i$ is related so that the estimated rotation matrix and translation vector will be consistent. Choosing corresponding vertices on an unrelated body part would lead to suboptimal rotation matrix and translation vector estimation. Nevertheless, choosing simply closest N_i mesh vertices to $\mathbf{s}_i$ may result in corresponding vertices clustered in a small area which would provide redundant information or again lead to suboptimal rotation matrix and translation

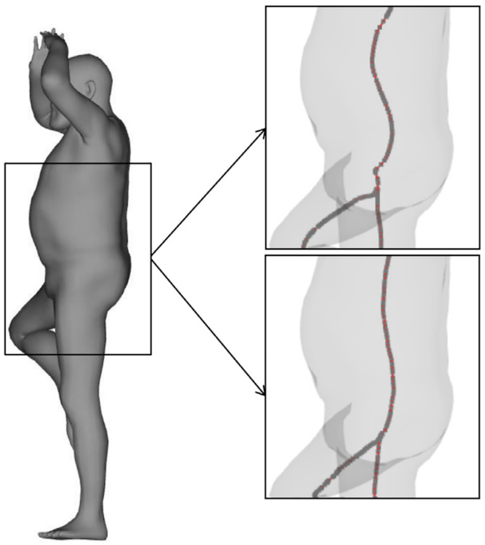


Fig. 3. An inter-subject skeleton transfer case (from the source in Fig. 1 to this different human) showing the resulting skeleton when just the closest mesh vertices to skeleton vertices were chosen as corresponding vertices (top-right) and when ρ_i parameter was used (bottom right).

vector estimation. The corresponding vertices of $\mathbf{s}_i$ should lie on the body part of the mesh to which $\mathbf{s}_i$ is related as much as possible while the distance between them should be as large as possible. To achieve this, we choose N_i vertices on the mesh which are closest to $\mathbf{s}_i$ and they do not belong to each others' 1-ring neighbourhoods. More formally, let $\mathcal{N}_1(\mathbf{v})$ be the set of vertices in 1-ring neighbourhood of vertex $\mathbf{v}$ and $C_i = \{\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \dots, \mathbf{v}_{i,N_i}\}$ be the set of correspondent points of $\mathbf{s}_i$, we choose the N_i closest vertices on the source mesh to $\mathbf{s}_i$ such that $\mathcal{N}_1(\mathbf{v}_{i,l}) \cap \mathcal{N}_1(\mathbf{v}_{i,m}) \notin C_i$ for $l \neq m$. The selection routine can easily be implemented first by sorting the mesh vertices according to their distances to $\mathbf{s}_i$ and then constructing the set C_i.

Before finishing this subsection, we would like to justify the need of making the number of corresponding vertices for a skeleton vertex $\mathbf{s}_i$ dependent on ρ_i and making them not to belong to each others' 1-ring neighbourhoods. As mentioned, if we correspond $\mathbf{s}_i$ just with a number of mesh vertices closest to it, then the corresponding vertices have the risk of clustering in a small area on a part of the mesh. This may further lead to distorted transferred skeletons especially between inter-subjects. Fig. 3 depicts an example of this phenomenon. In this example, the clustered vertices for the skeleton vertices going through the torso of the human were all selected from the back of the subject. When the skeleton was transferred to a mesh of another subject, the skeleton segment in the torso became distorted, given in the top-right subfigure. This was due to the fact that, as all corresponding points for the distorted part of the skeleton were from the back, the estimated transformations for those skeleton vertices cannot capture the surface on the stomach of the mesh. Resulting skeleton when we used ρ_i alleviates this problem as shown at the bottom-right subfigure.

3.2. Skeleton transfer

While transferring the source skeleton to a target with correspondence, its number of vertices and connectivity of vertices are preserved so that we obtain a skeleton for the target mesh registered to the source skeleton vertex-by-vertex. To achieve this, we first compute the optimum rotation matrix $\hat{\mathbf{R}}_i$ and translation vector $\hat{\mathbf{t}}_i$ that map the corresponding mesh vertices C_i of the source skeleton vertex $\mathbf{s}_i$ to the corresponding vertices D_i of the input mesh, and then apply the computed transformation to $\mathbf{s}_i$ to find the corresponding skeleton vertex of the input mesh $\hat{\mathbf{s}}_i$. Detailed derivation of the optimum rotation matrix and translation vector computation method we used can be found in [31]. Not to extend the discussion by repeating the steps in the indicated study, we just state the optimization problem and give its solution.

Let $D_i := \{\mathbf{w}_{i,1}, \mathbf{w}_{i,2}, \dots, \mathbf{w}_{i,N_i}\}$ be the set of vertices of the input mesh corresponding to its skeleton vertex $\hat{\mathbf{s}}_i$. As the input mesh is registered to the source mesh, the vertices in D_i are also registered to the vertices in the set C_i with the same id numbers. To obtain $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{t}}_i$, we define the following error metric:

$$E_i(\mathbf{R}_i, \mathbf{t}_i) := \sum_{j=1}^{N_i} ||(\mathbf{R}_i\mathbf{v}_{i,j} + \mathbf{t}_i) - \mathbf{w}_{i,j}||^2. \tag{2}$$

Our aim is to solve the optimization problem

$$(\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i) = \underset{\mathbf{R}_i,\mathbf{t}_i}{\operatorname{argmin}}\, E_i(\mathbf{R}_i, \mathbf{t}_i) \tag{3}$$

subject to $\mathbf{R}_i\mathbf{R}_i^T = \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}$.

Please note that optimum rotation matrix and translation vector for each skeleton vertex of the input mesh, i.e., $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{t}}_i$ are computed for each $\mathbf{s}_i$, $1 \le i \le N$, by using the method described in [31]. Subsequently, we find the location of the source skeleton vertex $\mathbf{s}_i$ when it is transferred to the input mesh by using the linear transformation

$$\hat{\mathbf{s}}_i = \hat{\mathbf{R}}_i\mathbf{s}_i + \hat{\mathbf{t}}_i. \tag{4}$$

Computation of skeleton vertices $\hat{\mathbf{s}}_i$ for the input mesh is followed by copying the edges in the source skeleton directly to the skeleton for the input mesh. In other words, if there is an edge between $\mathbf{s}_i$ and $\mathbf{s}_j$ in the source skeleton, we connect $\hat{\mathbf{s}}_i$ and $\hat{\mathbf{s}}_j$ by adding an edge between them.

Note that, we build our approach on the observation that even if the shape exhibits large non-rigid deformations, majority of the skeleton moves locally rigidly, e.g., from ankle to elbow. The optimal rotations and translations handle the rigid movement of the local parts as well as the non-rigid movement of the minor joint regions which are sufficiently small, e.g., elbow. This is validated through our comprehensive evaluations which involve skeleton transfer under very large non-rigid transformations (Section 4).

3.3. Postprocessing

When we follow the procedure mentioned up to now, we obtain a skeleton for the input mesh registered to the source skeleton, and we call it as the transferred skeleton. 1D curve representing the transferred skeleton may have ripples like in the figure at right due to the nonzero error values resulting in Eq. (2). In order to get rid of those ripples as much as possible, we applied Laplacian smoothing method to the 1D curve representing the transferred skeleton.

We introduce the notion of skeleton segment to explain the smoothing process more rigorously. Let $\mathbf{s}_i$ and $\mathbf{s}_j$ be skeleton vertices having only one neighbour or more than two neighbours. If all the edges between $\mathbf{s}_i$ and $\mathbf{s}_j$ on the path from $\mathbf{s}_i$ to $\mathbf{s}_j$ have exactly two neighbours, we call this path as a skeleton segment having $\mathbf{s}_i$ and $\mathbf{s}_j$ as terminals.

Let $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{L-1}, \mathbf{s}_L\}$ be a skeleton segment and let $\{\mathbf{s}'_1, \mathbf{s}'_2, \dots, \mathbf{s}'_{L-1}, \mathbf{s}'_L\}$ be the smoothed version of it. Positions of the terminal vertices are preserved, i.e., $\mathbf{s}'_1 = \mathbf{s}_1$ and $\mathbf{s}'_L = \mathbf{s}_L$. Let $\mathbf{s}_j$ be a non-terminal vertex in the segment. Let $\mathbf{s}_{j-1}$ be the preceding vertex

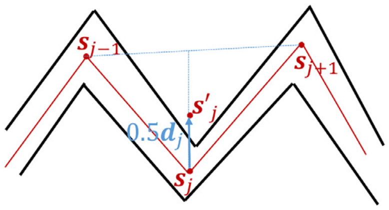


Fig. 4. An extreme case in 2D setup in which the shape boundary is zigzaggy.

and $\mathbf{s}_{j+1}$ be the succeeding vertex of it. Weight w_{j-1} of the edge between $\mathbf{s}_{j-1}$ and $\mathbf{s}_j$ and weight w_j of the edge between $\mathbf{s}_j$ and $\mathbf{s}_{j+1}$ are defined as

$$w_{j-1} := \frac{1}{||\mathbf{s}_j - \mathbf{s}_{j-1}||} \tag{5}$$

and

$$w_j := \frac{1}{||\mathbf{s}_{j+1} - \mathbf{s}_j||}, \tag{6}$$

respectively. After computation of the weights, the displacement vector $\mathbf{d}_j$ for $\mathbf{s}_j$ is computed as

$$\mathbf{d}_j := \frac{w_{j-1}\mathbf{s}_{j-1} + w_j\mathbf{s}_{j+1}}{w_{j-1} + w_j} - \mathbf{s}_j. \tag{7}$$

Finally, $\mathbf{s}'_j$ is computed by the relation

$$\mathbf{s}'_j = \mathbf{s}_j + 0.5\mathbf{d}_j. \tag{8}$$

Applying Laplacian smoothing procedure just one time may not be enough to reach a satisfying smooth skeleton segment. Thus, the smoothing procedure is applied repeatedly to a skeleton segment until the displacement is sufficiently small. To this end, iterations continue until $0.5||\mathbf{d}_j|| < 0.05\lambda_{avg}$ condition is satisfied for any $\mathbf{s}_j$ where λ_{avg} is the average edge length of the source skeleton segment.

The smoothing operation we used is computationally lightweight, as supported by the timing values given in Section 4.9. It smoothes the curve-skeleton mostly in one pass, acting just like a low-pass filter, if not, in a few passes. Note that, it does not check any condition about surface boundary which may arise the question whether it can push the transferred skeleton out of the shape boundary. For this to happen for a skeleton vertex $\mathbf{s}_j$, $0.5||\mathbf{d}_j|| > r_j$ should be satisfied where r_j is the distance to the closest mesh vertex to $\mathbf{s}_j$. Although it is mathematically possible, this can be considered as an extreme case. An example setup to this case is given in Fig. 4 in 2D to easily explain the case. In the figure, black lines represent shape boundary, red lines represent its curve-skeleton, $\mathbf{s}_{j-1}$, $\mathbf{s}_j$, $\mathbf{s}_{j+1}$ represent three consecutive skeleton vertices, $0.5\mathbf{d}_j$ represents half of the displacement vector, and $\mathbf{s}'_j$ represents new position of $\mathbf{s}_j$. It is hard to encounter such very thin and zigzaggy mesh structures in animation characters. Legs of the horse and centaur models are closest examples to such structures in the datasets we used and the smoothing operation did not push the transferred skeleton out of those structures. Despite all of these, one can still modify the smoothing operation adding a boundary check condition or use a completely different one to ensure the skeleton is inside of the shape boundary if the dataset includes such very thin and zigzaggy structures as in Fig. 4. However, this may make the smoothing operation computationally expensive relative to surface correspondence and transfer stages.

4. Experiments

In this section, we will first present the datasets used in the experiments. Subsequently, we will briefly explain the experiments we conducted to test the performance of the method in various cases and discuss the corresponding results. Finally, we will present the time consumed by the method for each model.

4.1. Datasets

We used the FAUST dataset [32] and the TOSCA dataset [33] in the experiments, for human and non-human skeleton extraction demonstrations, respectively. FAUST comes with triangulated meshes of 10 human subjects, each in 10 different poses. The meshes were obtained by registering a template mesh to all these poses, hence one-to-one correspondence is available over the dataset. We used a subset of this dataset that consists of 4 subjects. One pose of the first subject is used as the source mesh from which the source skeleton was extracted. 9 other meshes of this first subject are used as inputs to our intra-subject skeleton transfer experiments (Section 4.2), and the remaining 30 meshes are the inputs to the inter-subject part (Section 4.3). We also create and use the downsampled point clouds (Section 4.4) and punctured versions (Section 4.5) of these meshes for other experiments that demonstrate versatility.

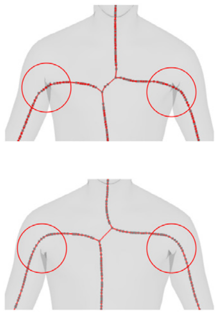
TOSCA, on the other hand, is a synthetic dataset obtained by deforming reference mesh models towards new poses, which also brings one-to-one correspondence information. We used Centaur, Horse, Gorilla, and Wolf subjects from TOSCA. One mesh of each subject class provides the respective source skeleton. In addition to the original meshes, we again create and use the downsampled point clouds and punctured versions of these meshes to perform a thorough evaluation (Section 4.6).

4.2. Intra-subject skeleton transfer experiments (FAUST)

In this set of experiments, intra-subject skeleton transfer ability of the method was tested, that is, transferring the source skeleton to the other poses of the same subject. Our method transferred the source skeleton, which is inside of the human body shown in Fig. 1, to the input meshes successfully. The transferred skeletons for the 9 poses of the first subject can be seen in Fig. 5 which shows that results maintain the medial axis property. Note that the deviation of medial axis in the right foot of the source skeleton (Fig. 1) is reflected in our transfer results. The accurate and smooth results verify the robustness of our surface correspondence, rigid transformation, and postprocessing schemes in Section 3.

4.3. Inter-subject skeleton transfer experiments (FAUST)

In this set of experiments, inter-subject skeleton transfer ability of the method was tested. In these experiments the source skeleton was transferred to the 10 meshes representing all 10 poses of the second, third and fourth subjects of the FAUST dataset. Some of the resulting skeletons transferred from the source mesh to the meshes of the second, third and fourth subjects of this dataset are shown in Fig. 6. Our method successfully transferred the source skeleton to 30 different poses extremely quickly in-one shot. The results show that the established rotation matrices and translation vectors can represent pose changes not only between intra-subjects but also between inter-subjects providing the method with the abilities such as skeleton transfer between genders and subjects with different weights and heights.



Fast transfer of the source skeleton to several input meshes in one-shot is not the only advantage of the proposed method. In some cases, the transferred skeleton is also more accurate than the skeleton for the input mesh computed from scratch via a direct method. An example to

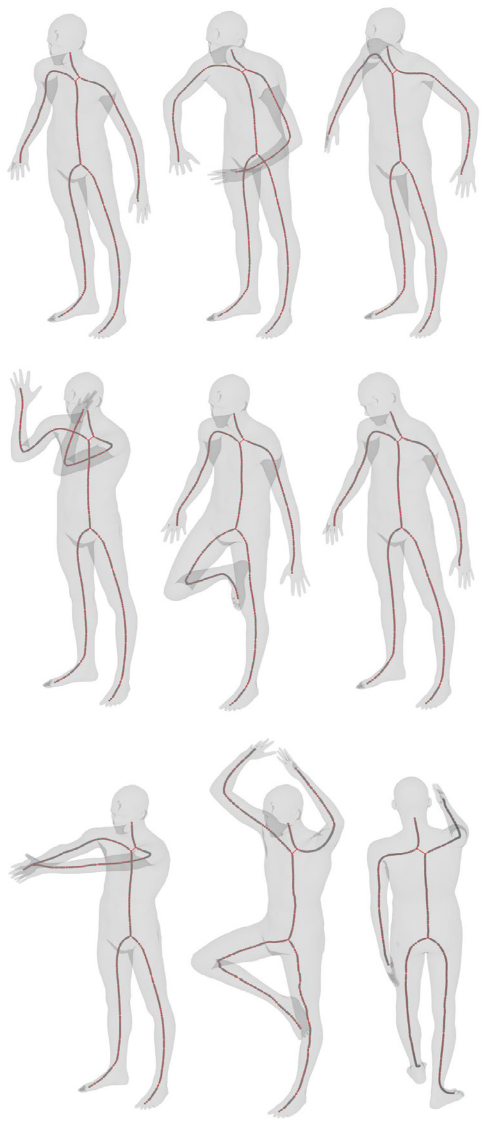


Fig. 5. Input meshes for the first subject of FAUST dataset with the transferred skeletons inside of them, resulting from the intra-subject experiments.

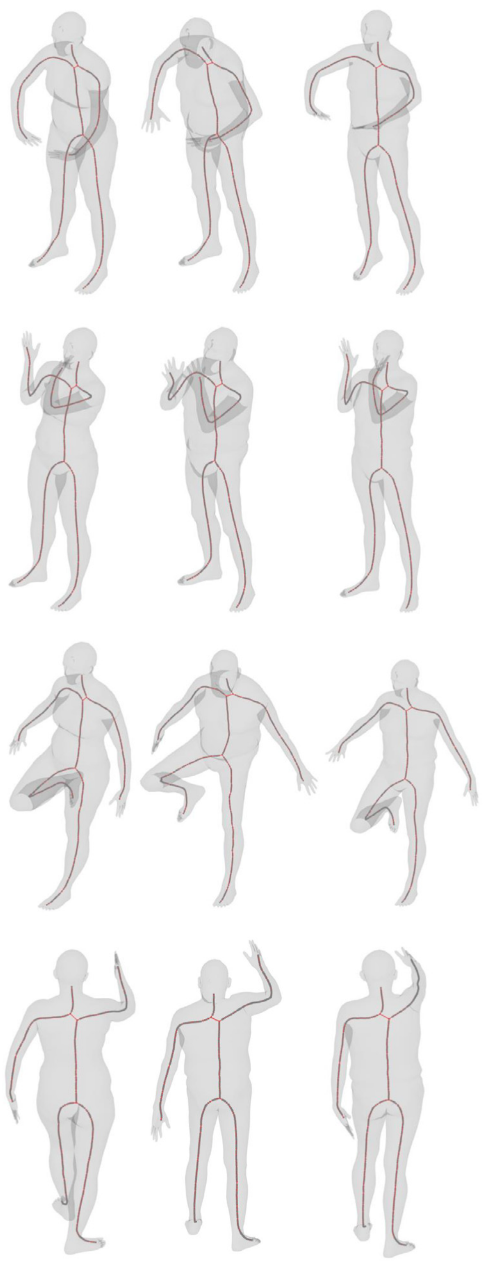


Fig. 6. Some of the inter-subject skeleton transfer results. Each column shows different poses of the same subject and each row shows the same poses of different subjects. Please note that we do not provide all of the results here for space issues. All of the results for this experiment are provided in the supplementary file.

this case can be observed in the figure at right. The skeleton at the top was computed by using MCF [9] and the one at the bottom was transferred from the source skeleton by using our method. As it can be seen, our result is better than that of MCF in terms of medial-axis property around the armpit (region inside of the red circles) of the subject. We also note that MCF, and many other existing algorithms, require watertight input meshes, meaning that they will fail on various other input types that we support, e.g., point clouds (Section 4.4) and punctured models with boundaries (Section 4.5).

4.4. Experiments for skeleton transfer to downsampled point clouds (FAUST)

In this set of experiments, the method was tested on input meshes having missing vertices and no connectivities. For this purpose, firstly, a number of vertices of the source mesh were selected randomly uniformly and the id numbers of these vertices were recorded. After that, vertices with the recorded id numbers were removed from all of the meshes, including the source mesh and all the input meshes of all of the subjects. We did not do any remeshing operations subsequently, instead, for each removed vertex, we further removed all the edges emanating from that vertex, and all the triangles associated with that vertex. To observe the performance degradation clearly, the number of removed vertices were increased 1000-by-1000 from 1000 to 6000, creating a downsampled point cloud data in each step. Subsequently, surface correspondences were established between the source skeleton and every downsampled versions of the source mesh, separately. Please note that, it is better to assume the inputs as downsampled point clouds at this point for this experiments because while transferring the source skeleton to them, we do not use and need any face or edge information. After the surface correspondence stage, the source skeleton was transferred to the point clouds of inputs by using the approach presented in Section 3.2 and the resulting skeletons were smoothed by using the approach presented in Section 3.3.

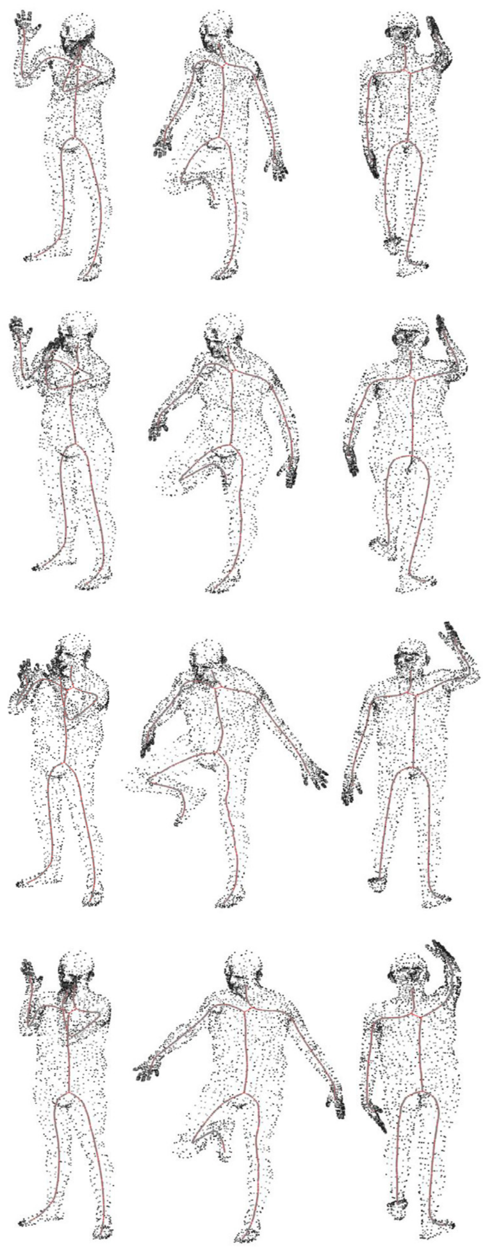


Fig. 7. Some of the skeleton transfer results when 4000 out of 6890 vertices were removed from the meshes; the subjects are represented as disconnected point clouds with 2890 points. Each column shows same poses of different subjects and each row represents different poses of the same subject. Please note that we do not provide all of the results here for space issues. All of the results for this experiment are provided in the supplementary file.

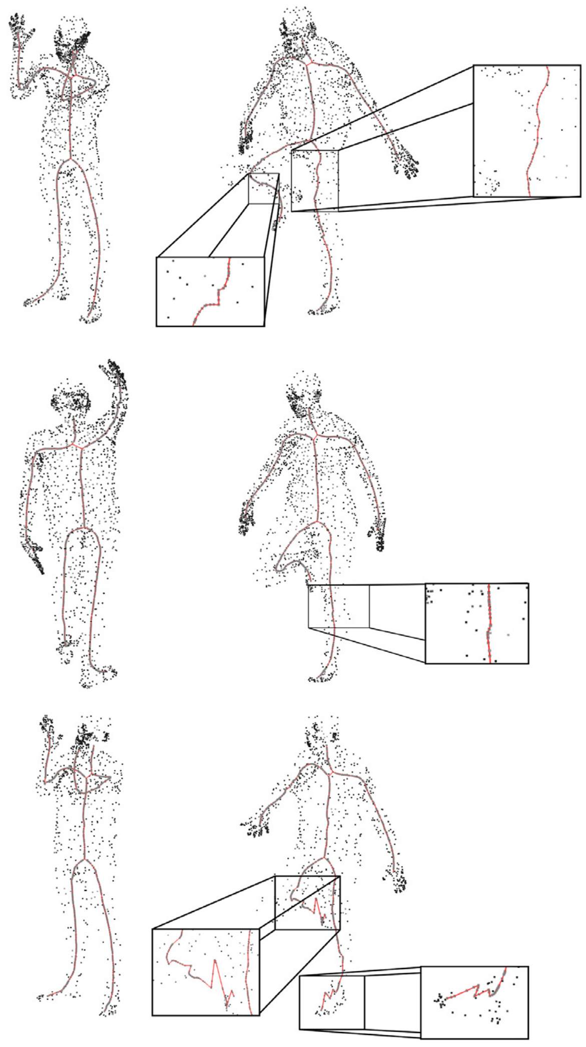


Fig. 8. Subfigures of the first two rows show four of the skeleton transfer results when 5000 of 6890 vertices were removed from the meshes. Subfigures of the last row show two of the skeleton transfer results when 6000 of 6890 vertices were removed from the meshes.

The performance of the method did not decrease noticeably even when approximately 60% of the vertices of the meshes were uniform randomly removed. Some of the results of this set of the experiments are shown in Fig. 7. This high level of robustness is due to the fact that, as long as the correspondence method mentioned in Section 3.1 can find alternative vertices for the removed vertices in close neighbourhoods of skeleton vertices, the established local transformations are able to represent both intra-subject and inter-subject pose changes.

The transferred skeletons started to deteriorate as the number of removed vertices were increased. The first two rows of Fig. 8 show four of the results when 5000 of 6890 (72.5% approx.) vertices of the meshes were uniform randomly removed. As one can verify from the figure, although the transferred skeletons are still valid for some input meshes (e.g., first two subfigures of the first column), they start to become inaccurate for some other meshes (e.g., first two subfigures of the second column). When 6000 of 6890 (87% approx.) vertices were removed uniform randomly, the method could not compute valid skeletons for the input meshes. Two of the results of this case are shown at the last row of Fig. 8, where one can see that although the transferred skeleton is not bad in some cases (e.g., left subfigure of the last row), most of them are deteriorated (e.g., right subfigure of the last row). The severe decrease in the accuracy is due to the fact that, when the surface correspondence method cannot find enough number of points in close neighbourhood of a skeleton vertex, it starts to correspond it to far-away vertices, resulting in erroneous rotation matrix and translation vector.

4.5. *Experiments for skeleton transfer to punctured meshes (FAUST)*

In this set of experiments, a set of connected vertices in a specific area of the meshes was removed instead of random removal, resulting in punctured meshes with boundaries. The id numbers of the removed vertices were the same for all the meshes including the source mesh and all the input meshes of all of the subjects. We generated three types of punctured meshes each one having a hole in a specific area: on the upper left arm, on the upper right leg and on the torso. As in the case for

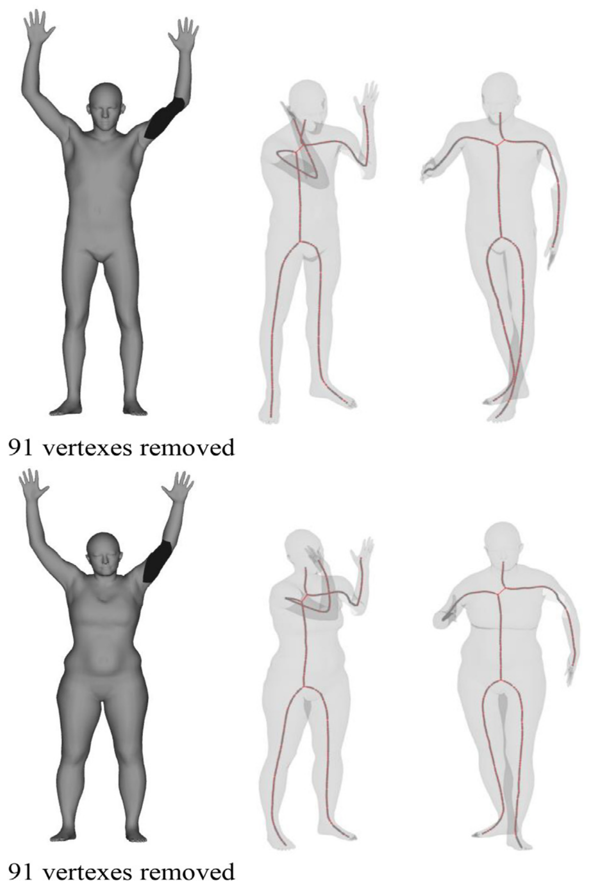


Fig. 9. Four of the skeleton transfer results when 91 vertices connected together on the upper-left arm of the subjects were removed, forming a hole. Each row represents a different subject. Shaded versions of the subjects are in the first column. The other two columns show the resulting skeletons for two different poses.

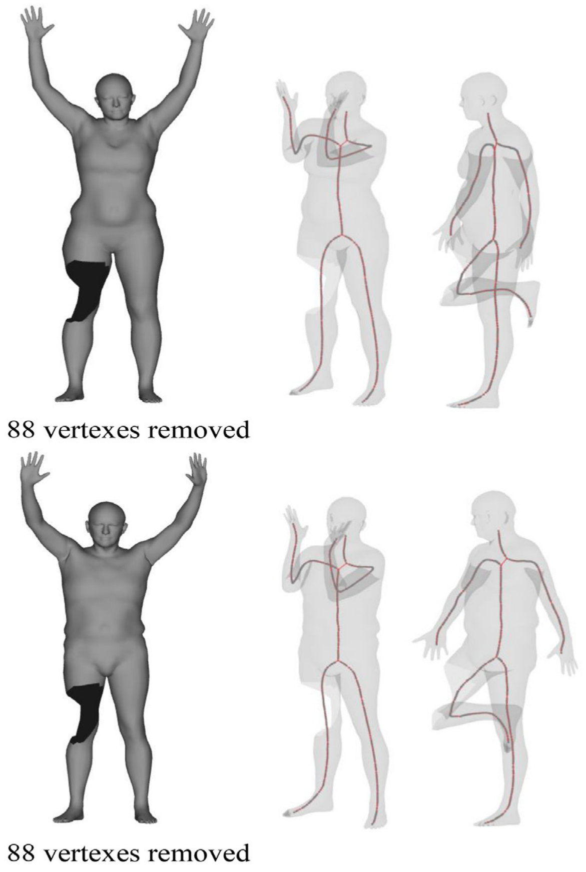


Fig. 10. Four of the skeleton transfer results when 88 vertices connected together on the upper-right leg of the subjects were removed, forming a hole. Each row represents a different subject. Shaded versions of the subjects are shown in the first column. The other two columns show the resulting skeletons for two different poses.

downsampling, after creation of new input meshes, surface correspondences were established between the source skeleton and every punctured version of the source mesh, separately. Surface correspondence step was followed by the skeleton transfer and smoothing steps in sequence.

The method transferred the source skeleton accurately in various cases of the punctured meshes. The results related with the inputs from which 91 connected vertices were removed resulting a hole in the upper left arm are shown in Fig. 9. In this case, when the vertices were removed, the skeleton vertices close to the hole were corresponded to the mesh vertices in the same rigid part with them, that is, in the upper left arm. As a consequence, the computed rotation matrices and translation vectors could capture the pose changes accurately. Of course, as the number of vertices gets larger, the results would become deteriorated.

The accuracy decrease in cases for punctured meshes depends both on the size and place of the hole. For example, when 88 vertices (almost same with the previous case) were removed from right-legs of the meshes, the accuracy dropped relatively severely for the most cases. Four examples to this case are given in Fig. 10. While the results at the middle column are accurate due to little pose change at the legs, the ones at the last column have inaccurate skeletons.

Four of the results can be seen in Fig. 11 when we removed 176 vertices from the torsos of the meshes. As it can be verified from the figure, when the position of torso of the subject changes significantly, the quality of the transferred skeleton has a high potential to decrease. The subfigure at the bottom-right is an example to this case where the skeleton segment in the torso has an abrupt change. The results associated with the punctured test cases implies that as the method corresponds a skeleton vertex $\mathbf{s}_i$ to the vertices in different rigid parts of the mesh than the part in which $\mathbf{s}_i$ lies, the estimated transformation for $\mathbf{s}_i$ becomes erroneous.

4.6. Experiments for skeleton transfer between non-human subjects (TOSCA)

To assess how well the method can be generalized, we used some meshes from the TOSCA dataset which contains meshes of non-human subjects. More specifically, for each set of Centaur, Horse, Gorilla and Wolf meshes from TOSCA dataset, we chose a source mesh separately. Following that, we extracted the source skeletons for each kind of non-human source mesh by using the MCF method as for the FAUST dataset. Subsequently, for each type, we transferred the source skeleton to other meshes for the same type. Fig. 12 shows the results of this set of experiments. Each row corresponds to a different non-human type, the first column shows the shaded versions of the source meshes whose skeletons can be seen in Fig. 1. The other columns show the transferred skeleton to the poses different than the source pose for each type. The mesh models in the dataset do not only correspond to non-human models but also have different number of vertices and sampling densities compared to the models in the FAUST dataset. We did not modify any parameter of the proposed method specifically for this dataset implying that the method is highly and easily generalizable.

We also tested the performance of the method on downsampled and punctured models from the TOSCA dataset to check whether the

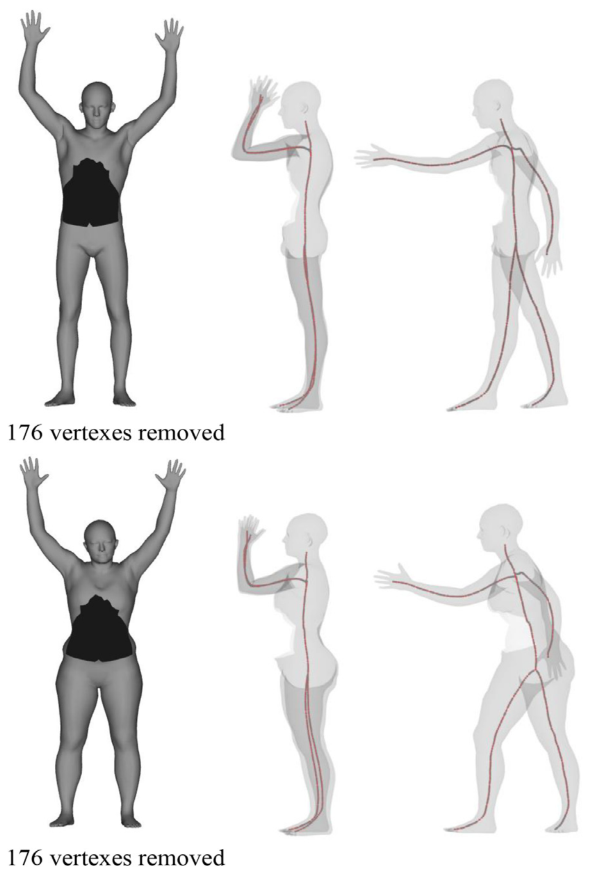


Fig. 11. Four of the skeleton transfer results when 176 vertices connected together on the torso the subjects were removed, forming a hole. Each row represents a different subject. Shaded versions of the subjects are shown in the first column. The other two columns show the resulting skeletons for two different poses.

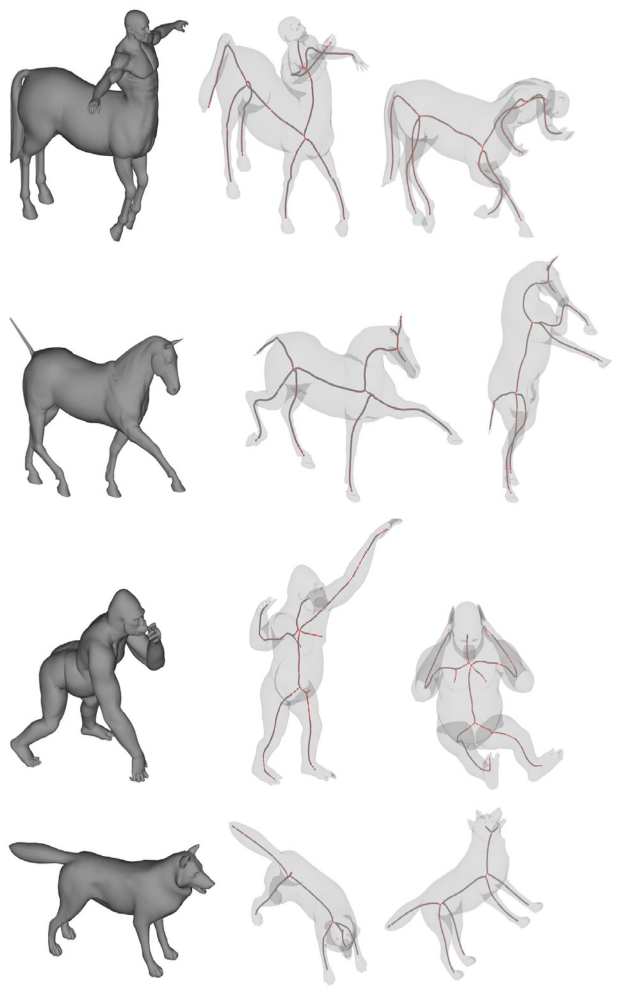


Fig. 12. Results obtained from the TOSCA dataset. Each row corresponds to a different subject. The first column shows shaded versions of the source meshes, the other columns show the transferred skeleton to a different pose than the source pose for each type. All of the results for these experiments are provided in the supplementary file.

successful results from the FAUST dataset still hold. To prepare thee downsampled models, we removed $\sim$60% of the vertices uniform randomly. The rest of the procedure was the same as the case for the downsampled models from the FAUST dataset described in Section 4.4. Results for two of the downsampled models of two different subjects from TOSCA dataset are shown in Fig. 13 as point clouds, each row showing a different subject. We removed 9500 out of 15,768 vertices randomly and uniformly from the centaur model and 11,500 out of 19,248 from the horse model. The source skeletons are shown in the first column while the transferred ones to different poses are shown in the second column. The results imply that conclusions for the models from the FAUST dataset can be generalized to this dataset seamlessly.

To prepare the punctured models, a set of connected vertices from a specific part of the mesh were removed. The rest of the procedure was the same as the case for the punctured models from the FAUST dataset described in Section 4.5. Results for two of the punctured models from the TOSCA dataset are shown in Fig. 14, each row showing a different subject. We removed 579 connected vertices out of 25,438 from the back of the Gorilla model and 464 connected vertices out of 4344 from the neck of the Wolf model. Shaded versions of the models and the source skeletons are shown in the first and second columns, respectively. The transferred skeletons to different poses are shown in the last columns. As it can be seen by the red circles in the figure, the transferred skeletons started to become deteriorated, similar to the case for FAUST models, when we removed a set of vertices from the models non-uniformly, like creating large holes on the surface.

4.7. *Experiments for skeleton transfer between non-registered shapes*

In the experiments of Sections 4.2 and 4.3, the source mesh and target meshes were registered vertex-by-vertex. The ones used in the experiments of Sections 4.4 and 4.5, a set of vertices were removed from the meshes but the remaining vertices were still registered vertex-by-vertex. In this new set of experiments, on the other hand, we test how well the method performs when true correspondence information between source and target meshes is unavailable.

To this end, we first need to choose source FAUST meshes for each of the second, third, and fourth subjects in order to perform intra-subject skeleton transfer experiments over these subjects. We chose the meshes in the same pose with the source mesh for the first subject, and then extracted their skeletons by using MCF [9]. After skeleton extraction, we registered source meshes and target meshes to each other for each subject by using the automatic shape correspondence method given in [17]. The rest of the skeleton transfer method is the same as in the other experiments.

The shape correspondence method [17] was run with 9 maximum levels and 8 initial samples. The details of these parameters can be found in [17]. In this setting, the algorithm computes approximately 4600 corresponding vertices on average between the source meshes and target

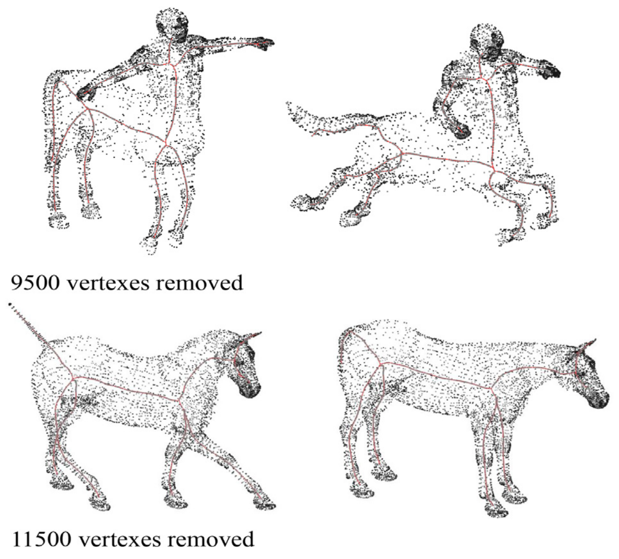


Fig. 13. Skeleton transfer results for two downsampled point cloud models from the TOSCA dataset. Each row represents a different subject. The first column shows the source skeletons in the point clouds along with the number of removed vertices. The second column shows the transferred skeletons to other poses again inside the point clouds. All of the results for these experiments are provided in the supplementary file.

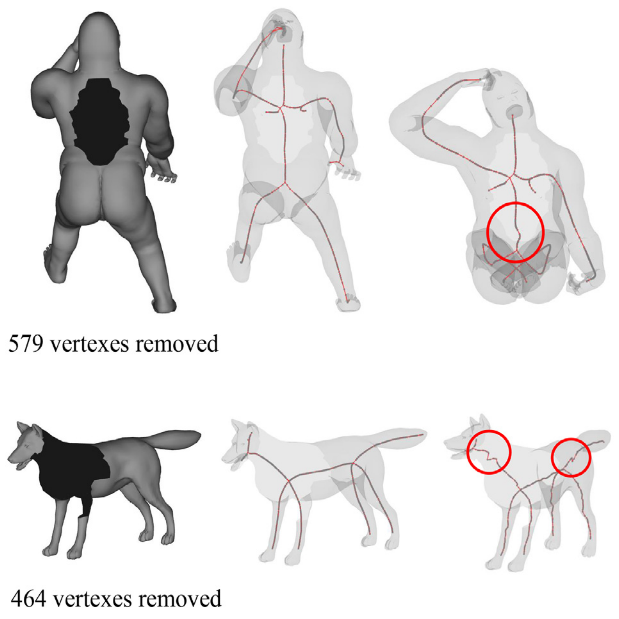


Fig. 14. Skeleton transfer results for two punctured models from TOSCA dataset. Each row represents a different subject. Shaded versions of the models along with the number of removed vertices are shown in the first column. The middle and last columns show the source and transferred skeletons, respectively.

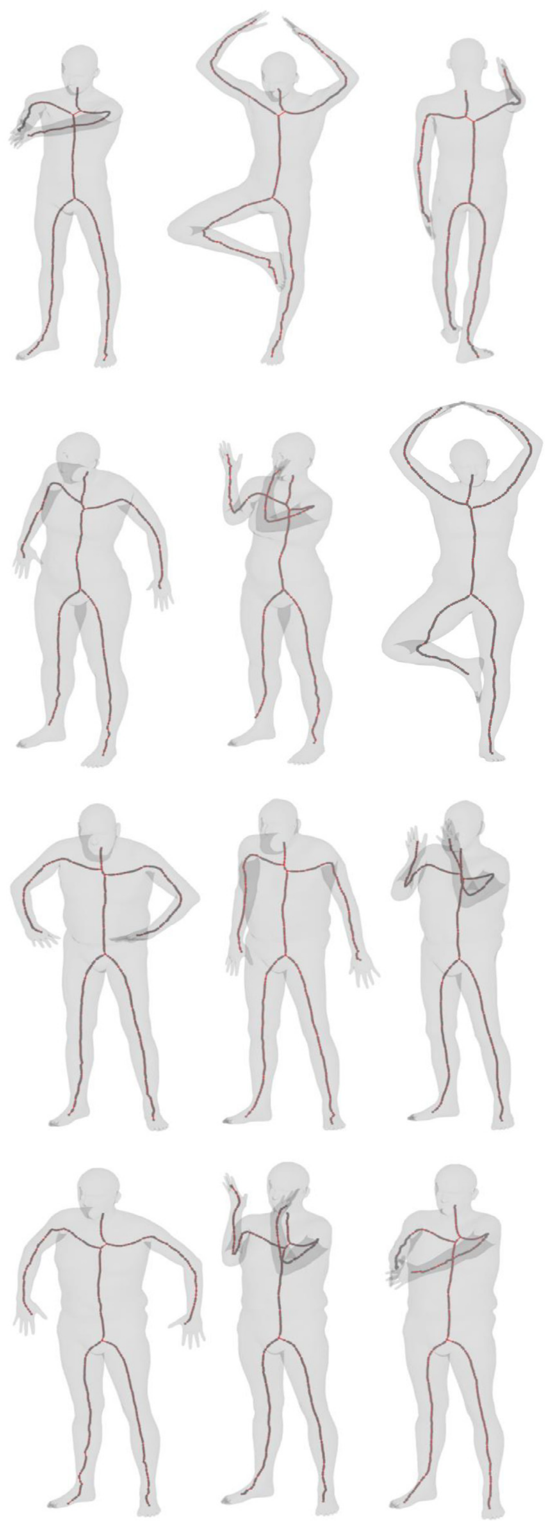


Fig. 15. Some of the skeleton transfer results between non-registered shapes. The meshes in each row belongs to a different subject.

meshes that have 6890 vertices each. Some of the results from FAUST dataset are given in Fig. 15. Each row represents a different subject.

We conducted the same experiment on TOSCA dataset for Centaur, Gorilla and Wolf models. For all of the models, [17] was run with 10 maximum levels and 8 initial samples for Centaur model. The respective parameters for Gorilla models were 9 and 8, and for Wolf model they were 8 and 8. Besides, it computed approximately 5000 corresponding vertices out of 15,768 for Centaur model, 3400 corresponding vertices out of 25,438 for Gorilla model and 2800 corresponding vertices out of 4344 for Wolf model on average. We provide one result for each model in Fig. 16. The number of computed corresponding vertices for Centaur and Gorilla models are far below ∼40% of number of original vertices, which is insufficient for estimating optimum rigid transformations. Their results are consequently not as robust as the Wolf case, which aligns with the conclusions drawn in Section 4.4.

Accuracy of the skeleton transfer method certainly depends on the accuracy of the shape correspondence method if vertex-by-vertex

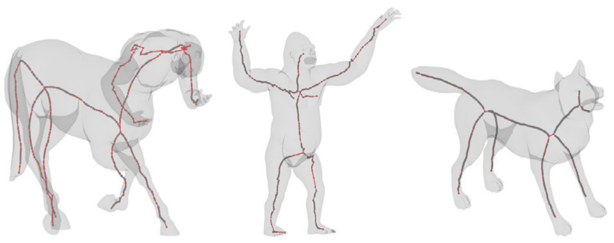


Fig. 16. One skeleton transfer result for each of centaur, gorilla and wolf models from TOSCA dataset.

correspondence is unknown beforehand. Therefore, it would be appropriate to share the accuracy of the shape correspondence method here. Average ground-truth correspondence cost for a computed map ϕ defined in [17] is given as

$$D_{\text{grd}}(\phi) = \frac{1}{|\phi|} \sum_{(s_i,t_j)\in\phi} g(s_i, t_j) \tag{9}$$

where (s_i, t_i) is the ground-truth match and (s_i, t_j) is the computed one. g gives the normalized geodesic distance between any two vertices such that the maximum distance is 1. Average D_{grd} values for FAUST and TOSCA are $\sim$0.042 and $\sim$0.033, respectively, which verify that our skeleton transfer method works fine in the presence of deviations from the ground-truth zero-cost maps.

Accuracy and robustness of skeleton transfer between non-registered shapes depend on two key points: The number of computed corresponding vertices and accuracy of the computed corresponding vertices. The accuracy and robustness of the transferred skeleton increase as these two quantities increase. To conclude, the proposed method allows skeleton transfer between non-registered meshes if the shape correspondence method run beforehand is robust and produces sufficient number of corresponding points.

4.8. Quantitative analysis

In addition to visually inspecting the transferred skeletons through all the preceding figures, we propose two novel error metrics to quantify the quality of centeredness propoperty of the results. Let $\mathbf{p}_{i,1}$ be the closest point to the skeleton point $\mathbf{s}_i$ lying on the mesh in terms of Euclidean distance. Note that $\mathbf{p}_{i,1}$ does not have to be a vertex of the mesh, it can lie on a triangle or edge of it. Let $\mathbf{d}_{i,1} := \mathbf{p}_{i,1} - \mathbf{s}_i$. We compute intersection points of the mesh and the ray $\mathbf{s}_i - t\mathbf{d}_{i,1}$, and denote the closest one to $\mathbf{s}_i$ as $\mathbf{p}_{i,2}$. Let $\mathbf{d}_{i,2} := \mathbf{p}_{i,2} - \mathbf{s}_i$. The setup can be seen at left in Fig. 17.

With this setup, we define ε_i as

$$\varepsilon_i := ||\mathbf{d}_{i,1} + \mathbf{d}_{i,2}|| \tag{10}$$

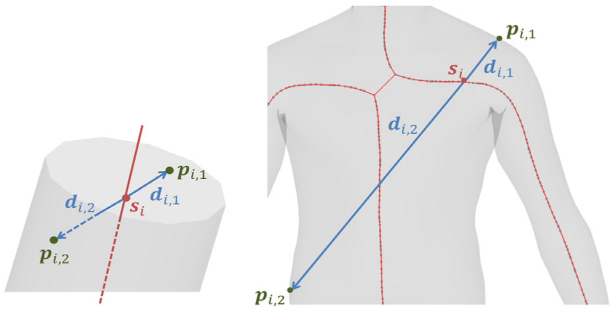


Fig. 17. The subfigure at left shows a didactic example introducing the variables pictorially used in the error metrics. The subfigure at right shows a real case in our models in which using the error metrics as they are would be inappropriate.

which gives us a value measuring how much $\mathbf{s}_i$ deviates from the medial-axis of the mesh. Close to zero values imply that $\mathbf{s}_i$ is close to the medial-axis of the mesh. We denote $\bar{\varepsilon}$ as the average value of ε_i among all skeleton vertices not outliers, which is going to be explained later in this section. It indicates how much a skeleton vertex of the mesh deviates from the medial-axis on average.

For the second error metric, we define κ_i as

$$\kappa_i := \frac{||\mathbf{d}_{i,2}||}{||\mathbf{d}_{i,1}||} - 1. \tag{11}$$

Unlike ε_i, κ_i provides us a unitless quantity measuring how much $\mathbf{s}_i$ deviates from the medial-axis of the mesh. Values close to zero imply that $\mathbf{s}_i$ is close to the medial-axis of the mesh. We denote $\bar{\kappa}$ as the average value of κ_i among all skeleton vertices not outliers. Like $\bar{\varepsilon}$, it also indicates how much a skeleton vertex of the mesh deviates from the medial-axis on average yet unlike $\bar{\varepsilon}$ it is unitless.

Error metrics in Eqs. (10) and 11 implicitly assume the curve-skeleton goes through the center of a tubular structure. The models in this study mostly consists of such structures: Legs, arms, torsos, necks, and tails. However, this assumption is not true for some parts of the models like shoulders. An example to this case is shown at right in Fig. 17. In this case, it is clear that $||\mathbf{d}_{i,1}||$ is not equal to $||\mathbf{d}_{i,2}||$ yet we can say that position of $\mathbf{s}_i$ is appropriate, at least visually. Such skeleton points should somehow be discarded from the accuracy computation. We call such vertices as outliers and spot them using the ratio $||\mathbf{d}_{i,2}||/||\mathbf{d}_{i,1}||$. If the ratio for $\mathbf{s}_i$ is above a threshold value determined beforehand, which we denote as η, we exclude $\mathbf{s}_i$ from the computations of the error metrics. Please note that similar outlier removal actions are common in the literature. For example, long rays are eliminated while computing the well-known and commonly-used shape diameter function [11]. According to our evaluations and observations, setting $\eta = 4$ is enough to spot most of the outliers for FAUST models. η values for the centaur, horse, gorilla, and wolf models are set to be 6, 6, 7, and 7, respectively. With these thresholds, $\sim$5% of vertices of FAUST models were spotted as outliers. Outlier percentages for the centaur, horse, gorilla, and wolf models are $\sim$9%, $\sim$11%, $\sim$11%, and $\sim$12%, respectively.

Quantitative results for various experiments using the error metrics introduced are provided in Table 1. In order to have a baseline to compare the results with, we also computed skeletons of the models using MCF [9], evaluated them, and report the results in the table. $\mu_{\bar{\varepsilon}}$ and $\sigma_{\bar{\varepsilon}}$ for an experiment represent average of the error values $\bar{\varepsilon}$ and standard deviation of $\bar{\varepsilon}$, respectively, among the models used in the experiment. Likewise, $\mu_{\bar{\kappa}}$ and $\sigma_{\bar{\kappa}}$ for an experiment represent average of the error values $\bar{\kappa}$ and standard deviation of $\bar{\kappa}$, respectively, among the models used in the experiment. As $\mu_{\bar{\varepsilon}}$ represents length, it would be better to have a quantity to compare it. For that purpose, we report l_{avg} which represents average edge length of the models used in an experiment. For the experiments with the downsampled models, results with the meshes whose $\sim$60% of the vertices were removed are reported.

It is important that the source skeleton ensures medial-axis property as much as possible so we selected the skeleton with the lowest error values as the source skeleton among a set of models. Error values related with them are provided separately in the last 5 lines of Table 1. For FAUST dataset, errors values for both original and downsampled models for intra-subject skeleton transfer are just a little below the error values for MCF. This is most probably due to the phenomenon explained in Section 4.3. Error values for inter-subject skeleton transfer are slightly above error values for MCF. For TOSCA dataset, error values for both the original and downsampled wolf and horse models are a little below the error values for MCF. For both the original and downsampled centaur and gorilla models, they are slightly above the ones for MCF. Furthermore, errors for downsampled models tend to be slightly greater than, if not they are almost equal with, the values for the original models, as expected. To sum up, the method provides us a tool allowing fast skeleton transfer between shapes in correspondence, possibly with boundaries or

Table 1
Resulting error values for various experiments. $\mu_{\bar{\varepsilon}}$ and $\sigma_{\bar{\varepsilon}}$ for an experiment represent average of the error values $\bar{\varepsilon}$ and standard deviation of $\bar{\varepsilon}$, respectively, among the models used in the experiment. $\mu_{\bar{\kappa}}$ and $\sigma_{\bar{\kappa}}$ for an experiment represent average of the error values $\bar{\kappa}$ and standard deviation of $\bar{\kappa}$, respectively, among the models used in the experiment. l_{avg} for an experiment represents average edge lengths of the models used in the experiment.

Experiment	$\mu_{\bar{\varepsilon}}$	$\sigma_{\bar{\varepsilon}}$	l_{avg}	$\mu_{\bar{\kappa}}$	$\sigma_{\bar{\kappa}}$
Intra-Subject Skeleton Transfer (FAUST)	0.0238	0.0016	0.0163	0.5386	0.0583
Inter-Subject Skeleton Transfer (FAUST)	0.0286	0.0028	0.0168	0.5487	0.0515
Intra-Subject Skeleton Transfer on Downsampled Models (FAUST)	0.0239	0.0027	0.0163	0.5487	0.0715
Inter-Subject Skeleton Transfer on Downsampled Models (FAUST)	0.0289	0.0032	0.0168	0.5830	0.0726
Mean Curvature Flow on Human Models (FAUST)	0.0248	0.0042	0.0167	0.5482	0.0728
Skeleton Transfer on Centaur Model (TOSCA)	6.6116	0.5257	1.6696	0.9887	0.0679
Skeleton Transfer on Downsampled Centaur Model (TOSCA)	6.7015	0.4830	1.6696	1.0030	0.0514
Mean Curvature Flow on Centaur Model (TOSCA)	6.1178	0.7676	1.6564	0.8953	0.0871
Skeleton Transfer on Horse Model (TOSCA)	8.6509	0.8597	2.0634	0.7819	0.1015
Skeleton Transfer on Downsampled Horse Model (TOSCA)	8.6794	0.9241	2.0634	0.7910	0.1140
Mean Curvature Flow on Horse Model (TOSCA)	8.7558	0.6345	2.0590	0.7664	0.0497
Skeleton Transfer on Gorilla Model (TOSCA)	6.7356	0.6250	1.0729	1.3140	0.2074
Skeleton Transfer on Downsampled Gorilla Model (TOSCA)	6.7600	0.5930	1.0729	1.3118	0.1190
Mean Curvature Flow on Gorilla Model (TOSCA)	5.8962	0.2415	1.0680	1.2381	0.2664
Skeleton Transfer on Wolf Model (TOSCA)	4.5667	0.1731	2.2529	0.8899	0.0118
Skeleton Transfer on Downsampled Wolf Model (TOSCA)	4.5556	0.1402	2.2529	0.8875	0.0075
Mean Curvature Flow on Wolf Model (TOSCA)	4.7830	0.1582	2.2538	0.9596	0.0462
Source Skeleton of Human Models (FAUST)	0.0226	N/A	0.0165	0.4972	N/A
Source Skeleton of Centaur Model (TOSCA)	5.5089	N/A	1.6298	0.7675	N/A
Source Skeleton of Horse Model (TOSCA)	8.2753	N/A	2.0218	0.7375	N/A
Source Skeleton of Gorilla Model (TOSCA)	5.3832	N/A	1.0531	0.9107	N/A
Source Skeleton of Wolf Model (TOSCA)	4.3227	N/A	2.2557	0.8108	N/A

missing vertices at the expense of a slight, or for some cases no, decrease in accuracy.

We believe that our error metrics are valuable in the sense they enable further quantitative assessments for visually similar skeletons. For example, thanks to our metrics we notice that, although visually similar, inter-subject transfer affects centeredness more than downsampling does. Also, these error metrics serve the purpose of comparing methods in terms of centeredness of the resulting curve-skeletons.

4.9. Timings

The experiments were conducted on a computer with Intel® Core^{TM} i7-2630QM CPU running at 2 GHz. Time consumed by the surface correspondence stage and by the skeleton transfer plus smoothing stages along with mesh and skeleton vertex counts for each subject are reported in Table 2. Time consumed by smoothing stage along is almost negligible compared to the other stages so we reported time elapsed in the skeleton transfer and smoothing stages together. Human model in Table 2 refer to the models in the FAUST dataset in which all the models have same number of vertices. Centaur, Horse, Gorilla, and Wolf models are from TOSCA dataset. We also report the timings related with the downsampled versions of the same models obtained by removing ~60% of their vertices in Table 3. One can notice the ~60% decrease in time consumed by the surface correspondence stage. This phenomenon is expected because of the linear dependence of surface correspondence to the mesh vertex count. However, time consumed in skeleton transfer and smoothing stage did not decrease at the same ratio as these stages depend on skeleton vertex count rather than mesh vertex counts. In punctured models, the number of removed vertices is small so timings of processing the punctured inputs are almost the same as the original inputs (Table 2). Please note that the time consumed in the surface correspondence and skeleton transfer stages can further be decreased by making the implementation parallel.

Table 2
Time elapsed in the different stages of the method along with number of input mesh vertices and skeleton vertices. Original meshes are used. All timings are reported in seconds.

Model	Mesh vertex count	Skeleton vertex count	Surface corr.	Transfer + Smoothing (Avg.)
Human	6890	522	4.121	0.588
Centaur	15,768	664	12.038	1.216
Horse	19,248	654	14.840	1.344
Gorilla	25,438	493	14.752	1.360
Wolf	4344	508	2.684	0.867

Table 3
Time elapsed in the different stages of the method along with number of input mesh vertices and skeleton vertices. Downsampled (~60% removed) point clouds are used. All timings are reported in seconds.

Model	Mesh vertex count	Skeleton vertex count	Surface corr.	Transfer + Smoothing (Avg.)
Human	2890	522	1.820	0.512
Centaur	6268	664	4.732	0.976
Horse	7748	654	5.787	1.065
Gorilla	10,238	493	5.888	0.852
Wolf	1744	508	1.160	0.768

4.10. Limitations

Eq. (1) implicitly assumes that models has enough vertices on the part of the mesh to which $\mathbf{s}_i$ is related for a reliable surface correspondence. Drawback of this implicit assumption can best be explained by an example. Let us discuss about punctured leg sequence given in the top row of Fig. 18. As the hole gets larger, the number of mesh vertices at the vicinity of skeleton vertices in the upper right leg decreases. Consequently, the surface correspondence stage of the method constructs correspondence sets for the skeleton vertices in the upper right part of the man with points from upper left leg, lower right leg and lower torso, which is a faulty correspondence. As those parts undergo different transformations than the upper right leg, transferred skeleton becomes inaccurate. However, when the faulty correspondence sets undergo similar transformations with the removed part and/or the pose change is little

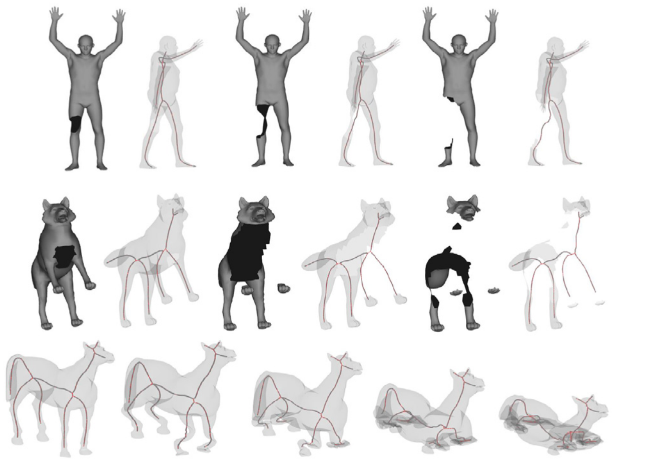


Fig. 18. The top row shows the punctured leg sequence and the middle row shows the punctured wolf sequence. For the mesh pairs at both rows, the shaded figure at left shows the source mesh and target meshes with transferred skeletons are shown at right. Bottom row shows mesh transfer in horse collapse sequence, the leftmost figure in this row shows the source mesh and skeleton..

between shapes, the skeleton does not deteriorates drastically. Punctured wolf sequence given at the middle row of Fig. 18 shows an example to this case. Full results of all of the experiments with punctured mesh sequences are provided in the supplementary file.

In addition to that, when the surface correspondence set undergoes an extreme deformation, Eq. (2) is incapable to capture the required transformation mapping C_i to D_i. This failure can be observed in the experiments with horse and camel collapse sequences [34]. For example, in the horse collapse sequence shown at the bottom row of Fig. 18, the front legs of the horse undergo such a drastic deformation that Eq. (2) cannot capture the true mapping for correspondence sets on the front legs of the shape. Results for the camel collapse sequence are provided in the supplementary file.

Another limitation of the method emerges when the deformation between the source shape and a target shape is not distance preserving, i.e., non-isometric. The method can tolerate such deformations and perform transfer successfully in those cases up to some extent; inter-subject skeleton transfer (Section 4.3) can be regarded as mild cases of such deformations. Nonetheless, if the degree of non-isometry is high, the resulting skeletons will not be accurate because of the faulty correspondences. For example, if we transfer skeleton of a cat to a giraffe, the result would be inaccurate due to high non-isometry (Distance between cat's chin and leg is not preserved over the giraffe's chin and leg due to stretched neck.). In this case, one can employ an automatic non-isometric correspondence method, or can even manually provide these correspondences to overcome this shortcoming.

5. Conclusion and future work

We have introduced skeleton transfer between two shapes in correspondence. The query shape can be discretized as a mesh (not necessarily watertight, not necessarily with fixed topology) or point cloud. To perform the transfer, firstly, the method is supplied a source shape and its skeleton that can be obtained manually or with any of the existing methods. For each point of the source skeleton, we associate a set of source shape points. After that, for each set of associated points, optimum rotation matrix and translation vector mapping the point set to the corresponding query point set is computed. By using these transformations, the source skeleton is transferred to the query shape point by point. Skeleton transfer process is completed by a smoothing postprocessing stage.

We conducted experiments to test both intra-subject and inter-subject human skeleton transfer performance of the method as well as non-human cases. Various types of shape discretizations are used in order to show versatility. The results of the experiments demonstrated visually appealing skeletons, even for the cases where the number of vertices in the meshes are decreased uniformly by ~60%. Nevertheless, the transferred skeletons started to loose their accuracy when we removed more of the vertices than this percentage. When the removal is done non-uniformly, i.e., in the form of a hole around a specific region, our method is still successful but not as tolerant as the ~60% case. These two removal cases, consequently, can be cast as the limitations of our method if they exceed a certain amount. The method was also tested on cases when no correspondence information available. In such cases, we first established a correspondence between the source shape and the query shapes by an existing shape correspondence method. Results of these experiments show that the method is still capable of transferring the skeleton accurately with accuracy depending on the performance of the shape correspondence method used.

We currently handle skeleton transfer between non-rigid shapes such as different poses of an articulated model. Another interesting research direction is to extend the scope of the work so that it can handle shapes with arbitrarily scaled parts, e.g., transferring horse skeleton to a giraffe which has a longer neck. Moreover, as the method keeps topology of the source skeleton during transfer operation, it cannot handle the cases when genus of the source mesh changes across input meshes. We leave enhancement of the method to manage such cases as a future work. In addition to skeleton transfer problem, investigating the inverse problem, that is, transferring the outer shape to other target skeletons, can be addressed as a future work.

Declaration of Competing Interest

None.

Acknowledgement

This work was supported by TUBITAK under the project EEEAG-115E471.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.gmod.2019.101041.

References

[1] H. Blum, A transformation for extracting new descriptors of shape, in: Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, 1967, pp. 362–380.
[2] B. Yang, J. Yao, X. Guo, DMAT: deformable medial axis transform for animated mesh approximation, Comput. Graph. Forum 37 (7) (2018) 301–311.
[3] T.K. Dey, J. Sun, Defining and computing curve-skeletons with medial geodesic function, in: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, 2006, pp. 143–152.
[4] N.D. Cornea, D. Silver, P. Min, Curve-skeleton properties, applications, and algorithms, IEEE Trans. Visual. Comput.Graph. 13 (2007) 530–548.
[5] P. Li, B. Wang, F. Sun, X. Guo, C. Zhang, W. Wang, Q-MAT: computing medial axis transform by quadratic error minimization, ACM Trans. Graph. (TOG) 35 (1) (2015) 8.
[6] Y. Yan, K. Sykes, E. Chambers, D. Letscher, T. Ju, Erosion thickness on medial axes of 3D shapes, ACM Trans. Graph. (TOG) 35 (4) (2016) 38.
[7] J.-H. Chuang, C.-H. Tsai, M.-C. Ko, Skeletonisation of three-dimensional object using generalized potential field, IEEE Trans. Pattern Anal. Mach.Intell. 22 (11) (2000) 1241–1251.
[8] M.K. Livesu, R. Scateni, Extracting curve-skeletons from digital shapes using occluding contours, Vis. Comput. 29 (2013) 907–916.
[9] A. Tagliasacchi, I. Alhashim, M. Olson, H. Zhang, Mean curvature skeletons, Comput. Graph. Forum 31 (5) (2012) 1735–1744.
[10] J.K. Feldman, S. Manish, Bayesian estimation of the shape skeleton, Proc. Natl. Acad. Sci. 103 (47) (2006) 18014–18019.
[11] L. Shapira, A. Shamir, D. Cohen-Or, Consistent mesh partitioning and skeletonisation using the shape diameter function, Vis. Comput. 24 (4) (2008) 249.
[12] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, A. Telea, 3D skeletons: a state-of-the-art report, Comput. Graph. Forum 35 (2) (2016) 573–597.
[13] S. Schaefer, C. Yuksel, Example-based skeleton extraction, in: Symposium on Geometry Processing, Citeseer, 2007, pp. 153–162.

[14] E. De Aguiar, C. Theobalt, S. Thrun, H.-P. Seidel, Automatic conversion of mesh animations into skeleton-based animations, Comput. Graph. Forum 27 (2) (2008) 389–397.

[15] N. Hasler, T. Thormählen, B. Rosenhahn, H.-P. Seidel, Learning skeletons for shape and pose, in: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ACM, 2010, pp. 23–30.

[16] B.H. Le, Z. Deng, Robust and accurate skeletal rigging from mesh sequences, ACM Trans. Graph. (TOG) 33 (4) (2014) 84.

[17] Y. Sahillioğlu, Y. Yemez, Coarse-to-fine combinatorial matching for dense isometric shape correspondence, Comput. Graph. Forum 30 (5) (2011) 1461–1470.

[18] V.G. Kim, Y. Lipman, T. Funkhouser, Blended intrinsic maps, ACM Trans. Graph. (TOG) 30 (4) (2011) 79.

[19] Y. Sahillioğlu, Y. Yemez, Coarse-to-fine isometric shape correspondence by tracking symmetric flips, Comput. Graph. Forum 32 (1) (2013) 177–189.

[20] D. Nogneng, S. Melzi, E. Rodolà, U. Castellani, M. Bronstein, M. Ovsjanikov, Improved functional mappings via product preservation, Comput. Graph. Forum 37 (2) (2018) 179–190.

[21] Y. Sahillioğlu, A genetic isometric shape correspondence algorithm with adaptive sampling, ACM Trans. Graph. (TOG) 37 (5) (2018) 175.

[22] I. Baran, J. Popović, Automatic rigging and animation of 3d characters, ACM Trans. Graph. (TOG) 26 (3) (2007) 72.

[23] O. Kin-Chung Au, C.-L. Tai, D. Cohen-Or, Y. Zheng, H. Fu, Electors voting for fast automatic shape correspondence, Comput. Graph. Forum 29 (2) (2010) 645–654.

[24] L. Kavan, S. Collins, J. Žára, C. O'Sullivan, Geometric skinning with approximate dual quaternion blending, ACM Trans. Graph. (TOG) 27 (4) (2008) 105.

[25] A. Jacobson, I. Baran, L. Kavan, J. Popović, O. Sorkine, Fast automatic skinning transformations, ACM Trans. Graph. (TOG) 31 (4) (2012) 77.

[26] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, Skeleton based shape matching and retrieval, in: Shape Modeling International, 2003, IEEE, 2003, pp. 130–139.

[27] Y. Sahillioğlu, M. Sezgin, Sketch-based articulated 3d shape retrieval, IEEE Comput. Graph. Appl. 37 (6) (2017) 88–101.

[28] Y. Sahillioğlu, L. Kavan, Detail-preserving mesh unfolding for nonrigid shape retrieval, ACM Trans. Graph. (TOG) 35 (3) (2016) 27.

[29] L. Lan, J. Mao, P. Huang, X. Guo, Medial-axis-driven shape deformation with volume preservation, Vis. Comput. 33 (2017) 789–800.

[30] J.M. Thiery, E. Guy, T. Boubekeur, E. Eisemann, Animated mesh approximation with sphere-meshes, ACM Trans. Graph. 35 (3) (2016) 1–13.

[31] O. Sorkine-Hornung, M. Rabinovich, Least-squares rigid motion using SVD, 2014. Department of Computer Science, ETH Zurich.

[32] F. Bogo, J. Romero, M. Loper, M.J. Black, FAUST: dataset and evaluation for 3D mesh registration, in: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3794–3801.

[33] A.M. Bronstein, M.M. Bronstein, R. Kimmel, Numerical Geometry of Non-Rigid Shapes, Springer Science & Business Media, 2008.

[34] R.W. Sumner, J. Popovic, Deformation transfer for triangle meshes, ACM Trans. Graph. 23 (3) (2004) 399–405.