Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

DMAT: Deformable Medial Axis Transform for Animated Mesh Approximation

Baorong Yang[1], Junfeng Yao[†1] and Xiaohu Guo[‡2]

[1]Xiamen University, China
[2] University of Texas at Dallas, USA

Abstract

Extracting a faithful and compact representation of an animated surface mesh is an important problem for computer graphics. However, the surface-based methods have limited approximation power for volume preservation when the animated sequences are extremely simplified. In this paper, we introduce Deformable Medial Axis Transform *(DMAT), which is deformable medial mesh composed of a set of animated spheres. Starting from extracting an accurate and compact representation of a static MAT as the template and partitioning the vertices on the input surface as the correspondences for each medial primitive, we present a correspondence-based approximation method equipped with an* As-Rigid-As-Possible *(ARAP) deformation energy defined on medial primitives. As a result, our algorithm produces DMAT with consistent connectivity across the whole sequence, accurately approximating the input animated surfaces.*

CCS Concepts

•***Computing methodologies*** → ***Volumetric models;***

1. Introduction

High-resolution representations for deforming 3D surfaces such as meshes can be redundant and expensive for storage, streaming, and processing. Coarse control structures, such as animated skeletons [KOF04, HRS10, LD12, LD14], lattice-based Freeform Deformation (FFD) [SP86, HJCW06], deformation cages [JZP*08], subdivision surfaces [GB18] and pose signal [CK12], have been used as alternatives of mesh representations because of their simplicity and editability.

However, the surface-based animation representations have limited approximation power for volume preservation when the animated sequences are extremely coarse. Ideally, the underlying deformation structures as well as the geometric details should be captured from the shape representation for the animated surfaces, and remain coarse enough for efficient streaming and intuitive editing of the sequence. For doing this, Thiery *et al.* [TBE16] innovatively proposed a volumetric structure called *Animated Sphere-Mesh* (ASM), for faithful approximation of animated surfaces. It can be used to rig a single mesh of the original sequence and reproduce faithfully the full animation sequence. Similar to the sphere-mesh, *Medial Axis Transform* (MAT) of 3D surfaces, first proposed by Blum [Blu67], also consists of a piecewise linear interpolation of spheres. Because medial axis is sensitive to the noise of surfaces which leads to numerous unstable spikes, significant efforts have been made on removing spikes and producing simple and accurate MAT [LWS*15, YSC*16]. However, existing MAT methods are only targeting on representing static shapes, because the MAT for every single mesh of the dynamic sequence could be quite different from each other.

In this paper, we propose the Deformable Medial Axis Transformation (DMAT) for representing dynamic surfaces, which is a deformable medial mesh composed of a set of time-varying medial spheres. Starting from extracting an accurate and compact MAT as a template for the reference frame, our method partitions the vertices of the input surface as the correspondences for each medial primitive, and conducts a correspondence-based approximation by minimizing an As-Rigid-As-Possible (ARAP) deformation energy defined on medial primitives. Our computed DMAT has consistent connectivity across the whole animation sequence, and accurately approximates the input surfaces.

2. Related works

2.1. Animated Mesh Approximation

Similar to ASM [TBE16], we use a sphere-based volumetric representation, i.e., dynamic medial mesh, for surface sequence ap-

[†] Corresponding author, Email: yao0010@xmu.edu.cn
[‡] Corresponding author, Email: xguo@utdallas.edu

proximation in this paper. ASM firstly extracts a dense sphere-mesh [TGB13] for surface sequence, then rely on a quadric error metric (QEM) [GMHP97] to define the cost of edge-collapse. Tkach *et al.* [TPT16] proposes a use of sphere-meshes as a geometric representation for real-time generative hand tracking.

The essential idea of example-based rigging methods [KOF04, ATTS10, HRS10, LD12, LD14] is to perform motion segmentation on the vertices of the input animation with similar rigid transformations and fit the subdomains of the motion, then estimate joint locations and bone sizes using linear or non-linear least squares, and finally optimize the bone transformations and skinning weights [JS11, KCvO08, Kv05]. The input animation is then approximated by linear blending skinning (LBS) [MTLT88]. Most of these methods are working on the explicit form of animated mesh surface. Inspired by those methods, we intend to segment the vertices of the input animation by medial primitives (medial slabs and medial cones), in which the spheres are considered to have similar rigid transformation, and capture the deformation of medial axis by the segmented vertices (we call the grouped vertices *correspondences* of medial primitives, see Sec. 4.2 for more details).

To drive the deformation of medial axis while making less distortion of local medial axis, we will integrate a geometric approach similar to Lan *et al.*'s technique [LYHG17]. They adopt an As-Rigid-As-Possible (ARAP) scheme to initially deform the medial axis so that its local transform is as-close-as-possible to a rigid transform, then the deformed medial axis is computed in an iterative way. The ARAP deformation has been widely used in shape deformation [CGLX17, IMH05, SSP07] by locally deforming shape primitives.

2.2. Medial Axis Computation

Computing the Voronoi diagram of a set of sampled points on object boundary [AB99] is the most commonly-used method for MAT extraction from a 3D shape, and the medial axis is simply the vertices of the Voronoid diagram. However, the medial axis is sensitive to the boundary noise of 3D shape, which generates many undesirable spikes, making it unsuitable for further practical applications.

To obtain a structually simple and compact medial axis, a series of criteria are developed to identify and prune the spikes from the initial medial axis. *Angle-based filtering method* [AM96, ACK01, FLM03, DZ02, SFM05] adopts a global threshold, the angle formed by a point of medial axis with its two closest points on the shape boundary. A medial point is directly removed if the associated angle is smaller than a user-specified threshold. *λ-medial axis method* [CL05, SP08, CCT09] uses the circumradius of closest points of a medial point as a pruning criterion and removes the medial points of which the associated circumradius is smaller than a given threshold λ. *Scale Axis Transformation* (SAT) [MGP10] employs a local pruning factor s as follows: firstly, enlarges all medial spheres by the factor $s > 1$, then removes the medial spheres that are contained in any other medial sphere, and the final medial axis is obtained by scaling back the surviving medial spheres by the factor $1/s$. Although the methods of using pruning criterion defined on the medial points are highly effective, the simplification quality depends on the specified threshold or factor. Sun *et al.* [SCYW15] used an error metric defined by the one-sided Hausdorff distance from the original shape to approximate volumes during simplification. *Progressive MAT* (PMAT) [FTB13] performs MAT simplification by collapsing edges of medial axis. The pruning criterion is defined as the ratio of the edge length to the difference of the medial radii at the two endpoints of the edge and used as the cost of edge-collapse. Li *et al.* [LWS*15] presented an efficient and effective method, called *Q-MAT* by using a quadratic error metric [GMHP97] to measure the edge-collapse error and the stability ratio to distinguish the spikes of medial axis. Yan *et al.* [YSC*16] proposed a global measure criterion based on the *Erosion Thickness* (ET) which performs very well in differentiating small boundary noises from shape features. Note that all these methods are for computing MAT for static surfaces, instead of dynamic sequences.

3. Deformable Medial Axis Transform

Following the representation in Q-MAT [LWS*15], we use a non-manifold mesh $\mathbf{M}$ consisting of edges and triangles, called *medial mesh*, to approximate the medial axis of a 3D surface $\mathbf{S}$ by volumetric enveloping.

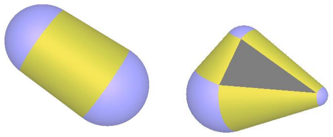

(a) (b)

Figure 1: *(a) A medial cone; (b) a medial slab.*

3.1. Medial Mesh and Footprint

In the medial mesh $\mathbf{M}$, a medial vertex is denoted as a *medial sphere* $\mathbf{m} = \{\mathbf{c}, r\}$, which contains its center $\mathbf{c}$ and radius r. An edge $\mathbf{e}_{ij}$ incident to two medial vertices $\mathbf{m}_i$ and $\mathbf{m}_j$, is defined by linear interpolation $\mathbf{m} = \alpha\mathbf{m}_i + (1-\alpha)\mathbf{m}_j$, $\alpha \in [0,1]$. Its enveloping primitive is called a *medial cone* bounded by two spherical caps and a truncated cone, as shown in Figure 1(a). Similarly, a triangle $\mathbf{f}_{ijk}$ incident to medial vertices $\mathbf{m}_i$, $\mathbf{m}_j$ and $\mathbf{m}_k$, is defined by linear interpolation $\mathbf{m} = \beta_i\mathbf{m}_i + \beta_j\mathbf{m}_j + \beta_k\mathbf{m}_k$, $\beta_i \in [0,1]$, $\beta_j \in [0, 1-\beta_i]$, and $\beta_k = 1 - \beta_i - \beta_j$. Its enveloping primitive is called a *medial slab* bounded by three spherical caps, three conical patches, and two triangles, as shown in Figure 1(b).

For a surface vertex $\mathbf{v}_i$ and a medial primitive $\mathbf{P}_j$, we define the *projection* of $\mathbf{v}_i$ onto $\mathbf{P}_j$ based on the minimizer of the following squared distances $E_d(\mathbf{m})$:

$$E_d(\mathbf{m}) = \left\|(\mathbf{v}_i - \mathbf{c}) \cdot \mathbf{n}_{ij} - r\right\|^2, \tag{1}$$

where $\mathbf{m} = \{\mathbf{c}, r\}$ is a sphere on the medial primitive, and $\mathbf{n}_{ij}$ is the outward normal of primitive $\mathbf{P}_j$ at $\mathbf{v}_i$, as shown in Figure 2. The details for computing the outward normal $\mathbf{n}_{ij}$ is given in the Supplementary Appendix. The sphere $\mathbf{m}$ minimizing $E_d(\mathbf{m})$ is defined as the *footprint* of $\mathbf{v}_i$ on $\mathbf{P}_j$.

Without loss of generality, let us consider $\mathbf{P}_j$ being a medial cone $\mathbf{e}_{kl}$. In this case, $\mathbf{m} = \alpha\mathbf{m}_k + (1-\alpha)\mathbf{m}_l$. By replacing $\mathbf{c}_{kl}$ and r_{kl}

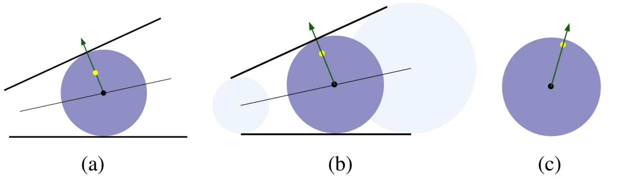


Figure 2: *Footprint and the corresponding outward normal on (a) medial slab, (b) medial cone, (c) medial sphere. The green outward normal represents the unit vector pointing from the center (the black point) of the purple footprint to the yellow surface vertex.*

into Eq. (1), it becomes a quadratic minimization problem with $\alpha \in [0,1]$ being the only variable to be solved, and Eq. (1) becomes:

$$E_d(\alpha) = \left\|(\mathbf{v}_i - \mathbf{c}_l - \alpha(\mathbf{c}_k - \mathbf{c}_l)) \cdot \mathbf{n}_{ij} - r_l - \alpha(r_k - r_l)\right\|^2. \quad (2)$$

Then α can be solved by $\mathrm{d}E_d(\alpha)/\mathrm{d}\alpha = 0$. If $0 \leq \alpha \leq 1$, we call the footprint to be an *inner-footprint*. If $\alpha < 0$ or $\alpha > 1$, we clamp it to $\alpha = 0$ or $\alpha = 1$, respectively, and call it an *outer-footprint*, even though we clamp the footprint to the boundary of the primitive. On a medial slab, the computation of footprint and the notions of *inner-* and *outer-footprint* are defined similarly.

3.2. Correspondence-based Approximation

Inspired by the idea of *Animated Sphere-Meshes* (ASM) [TBE16], we approximate an animated mesh sequence by a deformable medial mesh, where each vertex is associated with a time-varying sphere. For an animated surface sequence $\{\mathbf{S}^t | t = 0\ldots m\}$ of m frames, our computed DMAT consists of a medial mesh sequence $\{\mathbf{M}^t | t = 0\ldots m\}$ which has consistent connectivity.

In Thiery *et al.*'s *Sphere-Meshes* [TGB13], they perform edge-collapses for simplifying the initial sphere-mesh, where a set of vertices on triangle mesh can be used for approximating one sphere in the simplified sphere-mesh. Since the enveloping surface $\mathbf{C}$ of medial mesh $\mathbf{M}$ can be used to approximate the given surface $\mathbf{S}$ [Ede99], we cast our shape approximation problem into partitioning the given surface based on medial primitives, and approximating each partitioned region of the animated surface with DMAT by deforming the corresponding medial spheres. These deformed spheres, together with their connectivity, result in an animated medial mesh for our DMAT approximation.

We initialize the animated medial mesh with a simplified MAT $\mathbf{M}^0$ for the referenced triangle mesh $\mathbf{S}^0$, composed of n medial spheres $\left\{\mathbf{m}_k^0 | k = 0\ldots n\right\}$. The computation of $\mathbf{M}^0$ will be discussed in Sec. 4.1. Then we apply a two-stage ICP optimization method for computing the deformed medial spheres $\left\{\mathbf{m}_k^t | k = 0\ldots n\right\}$ of medial mesh $\mathbf{M}^t$ at frame $t \in [1,m]$, which will be discussed in Sec. 4. In this section, we focus on how to use the partitioned regions for DMAT approximation. In order to simplify notation, we remove the superscript t for all symbols in later discussions wherever the context is about a particular frame t in the animated sequence.

The vertices on the partitioned regions of input surfaces are called the *correspondences* of their corresponding medial primitives. We denote them as $\mathcal{C}_j$ for primitive $\mathbf{P}_j$. The detail for grouping the correspondences will be discussed in Sec. 4.2.

For each corresponding vertex $\mathbf{v}_i \in \mathcal{C}_j$ on the surface $\mathbf{S}$, we compute its projected footprint $\mathbf{m}_{ij} = \{\mathbf{c}_{ij}, r_{ij}\}$ on medial primitive $\mathbf{P}_j$, and would like to maintain its "relative position" to $\mathbf{P}_j$ as much as possible through an energy optimization. Since the footprint is on medial primitive, it can be represented as follows:

$$\mathbf{m}_{ij} = \sum_{\mathbf{m}_l \in \mathcal{V}_j} \alpha_{ijl} \mathbf{m}_l, \quad (3)$$

where $\mathcal{V}_j$ is the set of medial vertices for medial primitive $\mathbf{P}_j$, and $\{\alpha_{ijl}\}$ are the barycentric coordinates of $\mathbf{m}_{ij}$. We take the barycentric coordinates $\{\alpha_{ijl}\}$ as the "relative position" of $\mathbf{v}_i$ *w.r.t.* $\mathbf{P}_j$. If we keep their "relative position" to be fixed, i.e., fixing $\{\alpha_{ijl}\}$, then the footprint $\mathbf{m}_{ij}$ of each vertex can be simply interpolated from the medial vertices of this primitive. For each vertex $\mathbf{v}_i \in \mathcal{C}_j$, we define the vector from the center $\mathbf{c}_{ij}$ of $\mathbf{m}_{ij}$ to $\mathbf{v}_i$ to be its *footprint-ray*: $\mathbf{s}_i = \mathbf{v}_i - \mathbf{c}_{ij}$.

We consider the first frame (frame 0) to be a reference frame. When the corresponding set of vertices $\mathcal{C}_j$ on the surface are deformed in a later frame (frame t), we define an As-Rigid-As-Possible (ARAP) energy for each footprint-ray as the following squared L^2 distance: $Q_{ij} = ||\mathbf{R}_j \mathbf{s}_i^0 - \mathbf{s}_i||^2$, by assuming the primitive $\mathbf{P}_j$ is undergoing a rotation $\mathbf{R}_j$, and $\mathbf{s}_i^0$ is the footprint-ray in the reference frame. In this way, the deformed projective spheres $\mathbf{m}_{ij}$ can be used to drive the deformation of medial primitives $\mathbf{P}_j$. Thus, for each medial primitive $\mathbf{P}_j$, we can define the *ARAP footprint-ray energy* Q_j as the following summation:

$$\begin{aligned} Q_j &= \sum_{\mathbf{v}_i \in \mathcal{C}_j} Q_{ij} \\ &= \sum_{\mathbf{v}_i \in \mathcal{C}_j} \left\|\mathbf{R}_j \mathbf{s}_i^0 - \mathbf{s}_i\right\|^2 \\ &= \sum_{\mathbf{v}_i \in \mathcal{C}_j} \left\|\mathbf{R}_j \mathbf{s}_i^0 - \mathbf{v}_i + \sum_{\mathbf{m}_l \in \mathcal{V}_j} \alpha_{ijl} \mathbf{c}_l\right\|^2. \end{aligned} \quad (4)$$

We assume each medial primitive $\mathbf{P}_j$ has an "ideal" rigid-body motion with rotation $\mathbf{R}_j$ and translation $\mathbf{t}_j$. Then we can define the following *ARAP medial primitive energy* W_j as the following summation:

$$W_j = \sum_{\mathbf{m}_l \in \mathcal{V}_j} \left\|\mathbf{R}_j \mathbf{c}_l^0 + \mathbf{t}_j - \mathbf{c}_l\right\|^2, \quad (5)$$

where $\mathbf{c}_l$ is the center position of medial vertex $\mathbf{m}_l$ at deformed frame t, $\mathbf{c}_l^0$ is its position at reference frame.

By combining the above two energies, we can define the *ARAP total energy* E for the whole medial mesh:

$$E\left(\{\mathbf{R}_j, \mathbf{t}_j\}, \{\mathbf{m}_l\}\right) = \sum_j \left(Q_j + \omega W_j\right), \quad (6)$$

where the variables include the rigid-body motions $\left\{\mathbf{R}_j, \mathbf{t}_j\right\}$ of all medial primitives $\left\{\mathbf{P}_j\right\}$, and the center positions $\{\mathbf{c}_l\}$ of all medial vertices $\{\mathbf{m}_l\}$. ω is a weighting factor used to balance two energy terms. In all of our experiments, we simply set $\omega = 1$. By minimizing this ARAP total energy E, we will be able to solve for the deformed medial mesh.

We use an alternating optimization strategy for this minimization problem by iterating the following two steps until convergence: (1) by fixing all medial vertices $\{\mathbf{m}_l\}$, we can solve for the rigid-body motion $\{\mathbf{R}_j, \mathbf{t}_j\}$ of all medial primitives; (2) by fixing $\{\mathbf{R}_j, \mathbf{t}_j\}$ of all medial primitives, we can solve for all medial vertices $\{\mathbf{m}_l\}$.

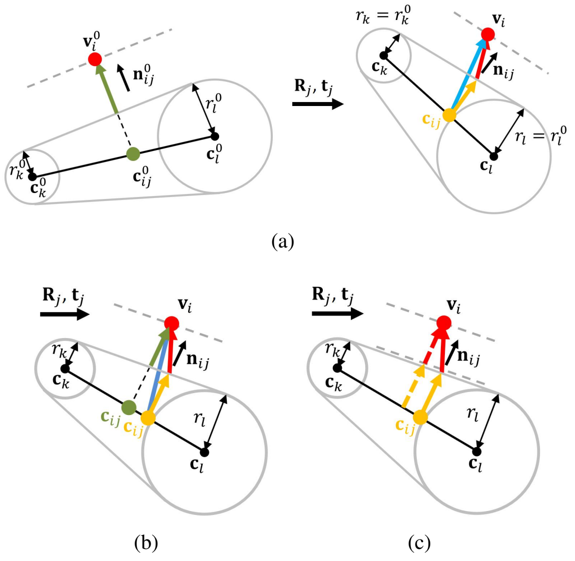
3.3. Enhanced ARAP Energy

The ARAP energy introduced in Eqs. (4)–(6) has two limitations: (1) the energy E does not depend on the radii of the medial vertices – in other words, the radii of medial vertices are fixed throughout the deformation process; (2) the "relative position" of surface vertex $\mathbf{v}_i$ *w.r.t.* medial primitive $\mathbf{P}_j$ are fixed, i.e., in Eq. (3) the barycentric coordinates of $\mathbf{m}_{ij}$ is fixed, which may not be a reasonable assumption as the surface undergoes non-rigid deformations.

In order to solve the first limitation, we modify the footprint-ray energy Q_{ij} as: $Q_{ij} = ||\mathbf{R}_j \mathbf{u}_{ij} r_{ij} - \mathbf{s}_i||^2$, where $\mathbf{u}_{ij}$ is a reference unit vector initially defined in the $\mathbf{s}_i^0$ direction, i.e., $\mathbf{u}_{ij} = \mathbf{s}_i^0 / ||\mathbf{s}_i^0||$, and r_{ij} is the radius of footprint $\mathbf{m}_{ij}$ for projecting vertex $\mathbf{v}_i$ onto medial primitive $\mathbf{P}_j$. In this way, the ARAP footprint-ray energy Q_j can be modified as:

$$Q_j = \sum_{\mathbf{v}_i \in \mathcal{C}_j} \left\| \mathbf{R}_j \mathbf{u}_{ij} \left(\sum_{\mathbf{m}_l \in \mathcal{V}_j} \alpha_{ijl} r_l \right) - \mathbf{v}_i + \sum_{\mathbf{m}_l \in \mathcal{V}_j} \alpha_{ijl} \mathbf{c}_l \right\|^2. \qquad (7)$$

By minimizing the ARAP total energy E in Eq. (6), each medial vertex will try to adjust its radius in order to provide best fitting for the footprint-ray in Eq. (7).



(a)

(b)

(c)

Figure 3: *Illustration of footprint-ray matching.*

In order to solve the above second limitation, we propose a two-stage optimization strategy:

- In the first stage, we fix the "relative position" of surface vertex $\mathbf{v}_i$ *w.r.t.* medial primitive $\mathbf{P}_j$, by using the same barycentric coordinates $\{\alpha_{ijl}\}$ computed from the projected sphere $\mathbf{m}_{ij}^0$ in the reference frame, and use $\mathbf{u}_{ij} = \mathbf{s}_i^0 / ||\mathbf{s}_i^0||$ for Q_j in Eq. (7). In this stage, we optimize for the rigid-body motions $\{\mathbf{R}_j, \mathbf{t}_j\}$ of all medial primitives, and the center positions of all medial vertices $\{\mathbf{m}_l\}$, as shown in Figure 3(a), we are computing the difference between the yellow reference ray and the current footprint-ray (in blue), which is the red ray.
- In the second stage, we relax the "relative position" of surface vertex $\mathbf{v}_i$ *w.r.t.* medial primitive $\mathbf{P}_j$, by allowing $\mathbf{v}_i$ to be re-grouped to a "best-fit" medial primitive which is the closest medial primitive of $\mathbf{v}_i$, and re-evaluating its barycentric coordinates through re-projection, as shown in Figure 3(b), the difference changes from the red ray to the green ray. We modify $\mathbf{u}_{ij}$ for Q_j in Eq. (7) as: $\mathbf{u}_{ij} = \mathbf{R}_j^{-1} \mathbf{n}_{ij}$, where $\mathbf{R}_j$ is the current optimized rotation for medial primitive and $\mathbf{n}_{ij}$ is the outward normal of the current primitive, as shown in Figure 3(c), the difference switches from the red ray to the dashed red ray. We modify the alternating optimization strategy by adding one additional step of re-grouping and re-projection for each vertex.

Even though we use different $\mathbf{u}_{ij}$ for these two stages, it is not difficult to show that the energy Q_j is actually consistent. For solving the minimization problem, we adopt an iterative way for optimizing the rigid motions $\{\mathbf{R}_j, \mathbf{t}_j\}$ of medial primitives and the medial vertices $\{\mathbf{m}_l\}$. The details will be discussed in Sec. 4.3. Our alternating optimization guarantees that the consistent ARAP total energy E is monotonically minimized, and the approximation accuracy is further improved, as shown in the examples of Figures 16–17.

4. Algorithm Details

In the previous section, correspondence-based approximation of medial primitives for extracting the animated medial mesh has been discussed. We now describe in detail how to compute the reference medial mesh, partition the vertices on triangle mesh $\mathbf{S}$ and use the partitioned correspondence of medial primitives as well as the corresponding projective spheres to deform the medial mesh. The flow of the approximation algorithm is shown as Figure 4.

4.1. Computation of Medial Mesh in Reference Frame

We compute an initial medial axis from reference frame $\mathbf{S}^0$ by using the method of Amenta and Bern [AB99], and use Q-MAT [LWS*15] to remove redundent spikes and simplify it to obtain the medial mesh $\mathbf{M}^0$.

In Amenta and Bern's approach [AB99], the initial medial axis is computed as the dual of Delaunay trianglulation of the sample points on the surface without any filtering. Each medial sphere in the initial medial axis is computed from a tetrahedron, the set of vertices of the tetrahedron are the correspondences of medial sphere $\mathbf{m}_i^0 = \left\{\mathbf{c}_i^0, r_i^0\right\}$, denoted as $\mathcal{C}(\mathbf{m}_i^0)$. Q-MAT follows the QEM framework [GMHP97] with edge-collapses. For each contraction, an edge $\mathbf{e}_{jk}^0$ is contracted to a new medial vertex $\mathbf{m}_g^0$, and the correspondences are merged by $\mathcal{C}(\mathbf{m}_g^0) = \mathcal{C}(\mathbf{m}_j^0) \cup \mathcal{C}(\mathbf{m}_k^0)$. Note

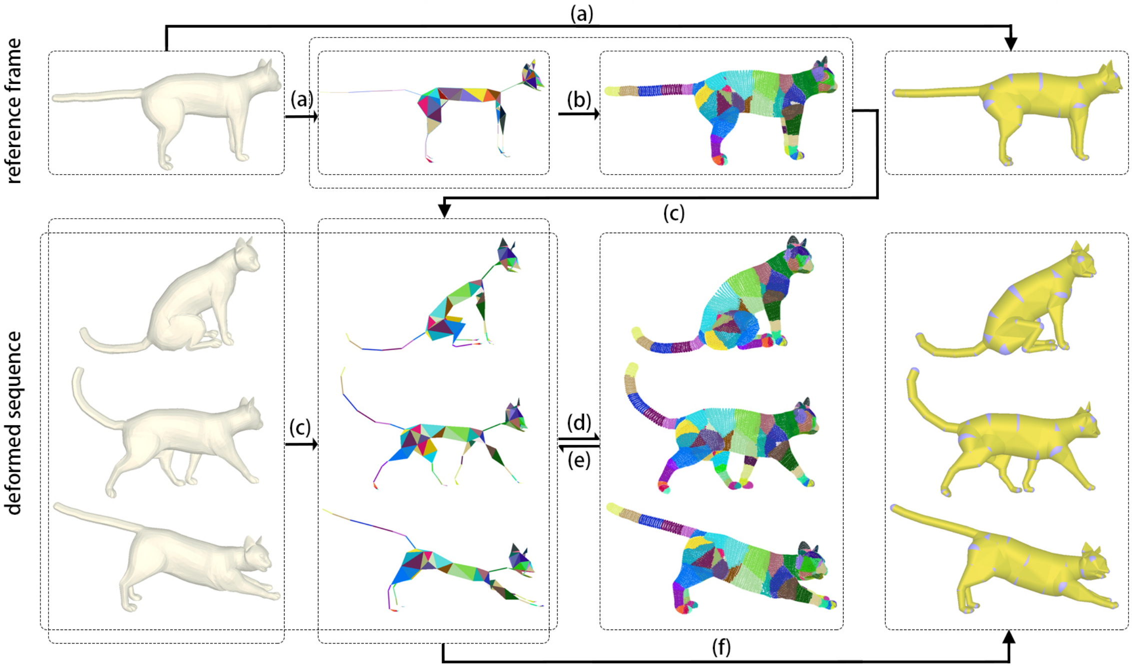


Figure 4: *Overview of our algorithm. From left to right, the four columns represent triangle meshes of input surfaces, medial primitives of medial meshes, partitioned correspondences for medial primitives and envelop of computed DMAT. (a) Compute the medial mesh of reference frame by Q-MAT [LWS*15]. (b) Partition the correspondences of medial primitives in the reference frame. (c) First-stage optimization for DMAT. (d) Second-stage optimization by re-grouping the correspondences for medial primitives in DMAT, and (e) optimize DMAT with the re-grouped correspondences. (f) Output the DMAT results.*

that the sampling density of input surface is crucial to the approximation accuracy of resulting medial axis. The detail for selecting a proper sampling density will be discussed in Sec. 5.

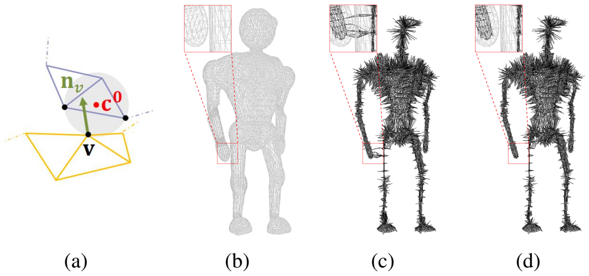


Figure 5: *Illustration of eliminating wrongly-connected edges. (a) Illustration of boundary-connected medial sphere in grey which is generated by the black points. (b) Input triangle mesh. Medial axis before (c) and after (d) deleting the wrongly-connected edges.*

Care must be taken when we use the Delaunay triangulation based method to extract the initial medial axis, as the boundary spheres could be connected by mistake when the two tetrahedra generating the spheres are close to each other while indeed they shouldn't connect with each other. For a medial sphere $\mathbf{m}^0 = \{\mathbf{c}^0, r^0\}$, if there exists a correspondence $\mathbf{v} \in \mathcal{C}(\mathbf{m}^0)$ satisfying the following condition, we call $\mathbf{m}^0$ a *boundary-connected* medial sphere:

$$\left(\mathbf{v} - \mathbf{c}^0\right) \cdot \mathbf{n}_v < 0, \tag{8}$$

where $\mathbf{n}_v$ in Figure 5(a) is the normal of vertex $\mathbf{v}$. As shown in Figure 5(c), the medial mesh of Humanoid is wrongly-connected between the right hand and right leg. It is obvious that it will cause some topological mistakes if we don't deal with it, i.e., a circle loop of the hand and the body. We can delete those edges which contain at least one boundary-connected medial sphere, as shown in Figure 5(d).

4.2. Partition of Correspondences

We use the following signed distance $d_j(\mathbf{v}_i)$ from a surface point $\mathbf{v}_i$ to medial primitive $\mathbf{P}_j$, as a guidance to partition the surface into independent regions grouped by medial primitives:

$$d_j(\mathbf{v}_i) = \left(\mathbf{v}_i - \mathbf{c}_{ij}\right) \cdot \mathbf{n}_{ij} - r_{ij}, \tag{9}$$

where $\mathbf{m}_{ij} = \{\mathbf{c}_{ij}, r_{ij}\}$ is the footprint of $\mathbf{v}_i$, and $\mathbf{n}_{ij}$ is the outward normal of primitive $\mathbf{P}_j$ at $\mathbf{v}_i$. Similar to a Voronoi cell for Euclidean distance, the correspondences $\mathcal{C}_j$ stores the closest surface points for $\mathbf{P}_j$ based on the above signed distance.

However, different from a Voronoi diagram in Euclidean space, there are several cases that we need to handle for the partition of correspondences on surface. Suppose two medial primitives $\mathbf{P}_j$ and

$\mathbf{P}_k$ are neighbors and share either a cone or a sphere. For a nearby surface point $\mathbf{v}_i$, we would like to decide whether assigning $\mathbf{v}_i$ to $\mathbf{P}_j$ or to $\mathbf{P}_k$.

In the first case, suppose $\mathbf{v}_i$ has an *inner-footprint* $\mathbf{m}_{ij}$ on $\mathbf{P}_j$, and an *inner-footprint* $\mathbf{m}_{ik}$ on $\mathbf{P}_k$. We simply assign $\mathbf{v}_i$ to the medial primitive that has smaller signed distance of Eq. (9).

In the second case, suppose $\mathbf{v}_i$ has an *inner-footprint* $\mathbf{m}_{ij}$ on $\mathbf{P}_j$, and an *outer-footprint* $\mathbf{m}_{ik}$ on $\mathbf{P}_k$. In this case, we simply assign $\mathbf{v}_i$ to primitive $\mathbf{P}_j$.

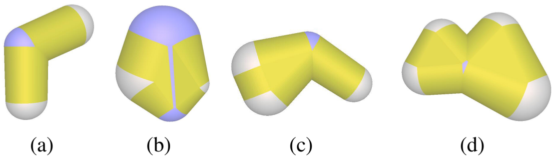


Figure 6: *Same footprint on two medial primitives. (a) two cones share a sphere, (b) two slabs share a cone, (c) A cone and a slab share a sphere, (d) two slabs share a sphere.*

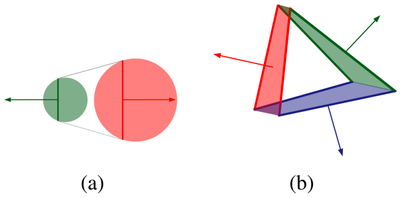


Figure 7: *Boundary planes of (a) medial cone and (b) medial slab.*

The third case is when the footprints of $\mathbf{v}_i$ on both $\mathbf{P}_j$ and $\mathbf{P}_k$ are the same, i.e., $\mathbf{m}_{ij} = \mathbf{m}_{ik}$. In this case, both $\mathbf{m}_{ij}$ and $\mathbf{m}_{ik}$ are *outer-footprints*. We introduce *boundary plane* which is the bisecting plane of two connected medial primitives for this case. There are four sub-cases that need to be considered, as shown in Figure 6:

- **Two medial cones share a sphere**. For each cone, we can find its two boundary planes as shown in Figure 7(a). We compare the signed Euclidean distance from $\mathbf{v}_i$ to the two boundary planes of the two connected cones, and assign $\mathbf{v}_i$ to the cone with smaller distance.
- **Two medial slabs share a cone**. For each slab, we can find its three boundary planes as shown in Figure 7(b). We compare the signed Euclidean distance from $\mathbf{v}_i$ to the two boundary planes of the two connected slabs, and assign $\mathbf{v}_i$ to the slab with smaller distance.
- **A medial cone and a medial slab share a sphere**. We treat this case as three cones sharing a sphere (two of them are the edges of the medial slab). We use the above strategy to compare two groups of cones. When one of the cones in the slab is closer to $\mathbf{v}_i$, we assign $\mathbf{v}_i$ to the slab, and vice versa.
- **Two medial slabs share a sphere**. We sum up the two signed distance between $\mathbf{v}_i$ and the boundary planes of two cones of each medial slab, and assign $\mathbf{v}_i$ to the slab with smaller summed distance.

4.3. Medial Mesh Deformation

As mentioned in Sec. 3.2 and 3.3, we apply a two-stage optimization method for computing the deformed medial mesh.

For minimizing our ARAP total energy E of Eq. (6), the optimal translation can be simply obtained as the barycenter of medial primitive:

$$\mathbf{t}_j = \frac{\sum_{\mathbf{m}_l \in \mathcal{V}_j} \mathbf{c}_l}{|\mathcal{V}_j|}, \tag{10}$$

where $\mathcal{V}_j$ is the set of medial vertices for primitive $\mathbf{P}_j$, and $|\mathcal{V}_j|$ is its number of medial vertices.

Then we fix the translation $\{\mathbf{t}_j\}$ and medial vertices $\{\mathbf{m}_l\}$ to compute the optimal rotation $\{\mathbf{R}_j\}$. This least-square rigid motion problem [Sor17] can be solved by using Singular Value Decomposition (SVD). The details on solving the optimal rigid motion $\{\mathbf{R}_j, \mathbf{t}_j\}$ are discussed in the Supplementary Appendix.

After getting the rotation and translation $\{\mathbf{R}_j, \mathbf{t}_j\}$ for medial primitives, we can solve for all medial vertices $\{\mathbf{m}_l\}$ by minimizing E in Eq. (6) with fixed $\{\mathbf{R}_j, \mathbf{t}_j\}$, which results in a quadratic optimization problem and can be easily solved by a linear system. The detail on optimizing the medial spheres is discussed in the Supplementary Appendix. In the first stage of optimization, we fix all radii $\{r_l\}$ and solve for the centers $\{\mathbf{c}_l\}$ of all medial vertices only. We perform N_1 number of iterations of computing $\{\mathbf{R}_j, \mathbf{t}_j\}$ and $\{\mathbf{c}_l\}$ in the first stage.

In the second stage, both the centers and radii are optimized. Note that the radii solved in the second stage could be negative, so we restrict the radii to be nonnegative. Besides, we bound the spheres' radii to avoid overly-large spheres. This upper bound R_l is set by the maximum radius of all projective spheres of the adjacent medial primitives $\mathcal{N}(\mathbf{m}_l)$ of medial vertex $\mathbf{m}_l$:

$$R_l = \max_{\mathbf{v}_i \in \mathcal{C}_j, \mathbf{P}_j \in \mathcal{N}(\mathbf{m}_l)} r_{ij}, \tag{11}$$

where r_{ij} is the radius of the projective sphere $\mathbf{m}_{ij}$ of surface vertex $\mathbf{v}_i$ on primitive $\mathbf{P}_j$. After solving the medial vertices $\{\mathbf{m}_l\}$ for the second stage, we check the solved radii: if it is out of the bound, then we clamp it to be either 0 or R_l respectively.

In the second stage of optimization, we also perform re-grouping and re-projection for all surface vertices after solving for both $\{\mathbf{R}_j, \mathbf{t}_j\}$ for medial primitives and $\{\mathbf{c}_l, r_l\}$ for medial vertices, before entering the next round of iteration. We perform N_2 number of re-grouping and re-projection in the second stage. After each re-reprojection, we perform N_3 number of iterations of computing $\{\mathbf{R}_j, \mathbf{t}_j\}$ and $\{\mathbf{m}_l\}$. In all of our experiments, we set $N_1 = 10$, $N_2 = 8$ and $N_3 = 5$.

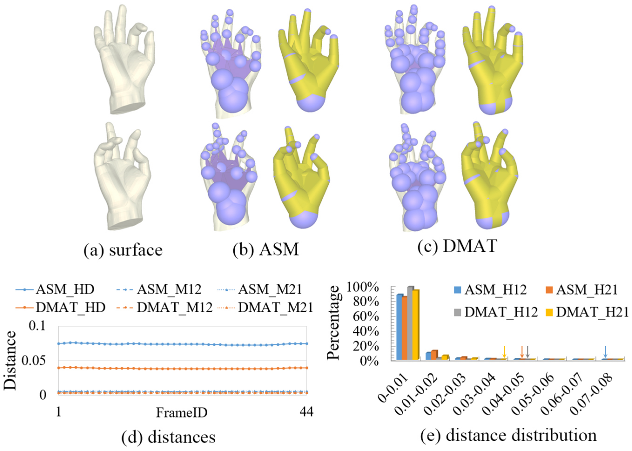
5. Results

We have implemented our DMAT algorithm in C++ and conducted the experiments on an Intel(R) Core(TM) i7-6700 CPU running at 3.10GHz with 8GB of main memory. By using the CGAL package "Delaunay Triangulation 3", reference medial mesh is extracted and simplified with Q-MAT [LWS*15], and correspondences of the

Table 1: *Approximation accuracy for our DMAT compared with ASM [TBE16]. #S / #E / #T: number of medial vertices (spheres), edges without incidental triangles, and triangles in the output medial mesh. The comparison is based on the same number of medial vertices (spheres).* **HD**, **M12**, *and* **M21** *are evaluated across all frames of the animation sequences. Note: largest errors of each animation are in bold font.*

	DMAT						ASM					
Input Anim.	**#S**	**#E**	**#T**	**HD**	**M12**	**M21**	**#S**	**#E**	**#T**	**HD**	**M12**	**M21**
Hand	34	18	17	3.873%	0.304%	0.404%	34	8	43	**7.393%**	**0.534%**	**0.565%**
Cat poses	85	17	77	2.760%	0.231%	0.272%	85	4	145	**5.430%**	**0.368%**	**0.666%**
Horse-gallop	46	30	13	2.372%	0.287%	0.286%	46	14	51	**2.853%**	**0.343%**	**0.416%**
Flamingo poses	20	21	0	2.637%	0.386%	0.491%	20	8	10	**3.267%**	**0.543%**	**0.723%**
Samba	38	21	16	2.731%	0.354%	0.338%	38	11	40	**5.582%**	**0.405%**	**0.531%**
Jump	10	11	0	8.407%	**2.044%**	**1.855%**	10	6	2	**9.127%**	1.774%	1.476%
Horse-collapse	46	28	14	4.286%	0.503%	0.699%	46	14	51	**6.527%**	**0.689%**	**1.167%**

simplified medial spheres are used for computing the correspondences of its neighboring medial primitives, then the two-stage optimization is applied to compute the deformable medial mesh for approximating the input surface sequences.

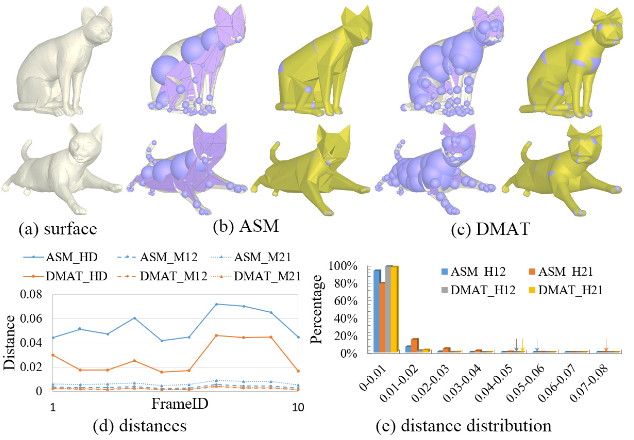


(a) surface (b) ASM (c) DMAT
(d) distances (e) distance distribution

Figure 8: *Comparison with ASM on Hand anim. with 34 spheres.*

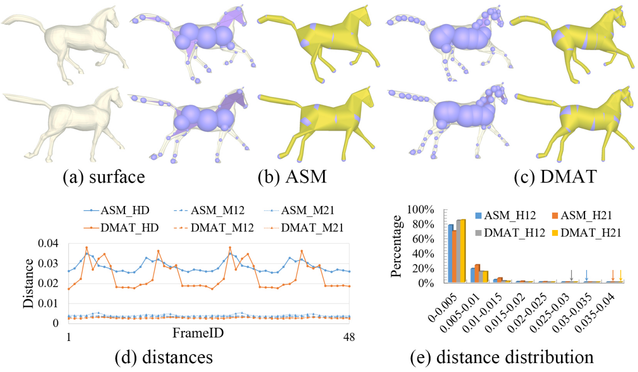
We use the two-sided Hausdorff distance error, denoted **HD**, to evaluate the approximation accuracy of the extracted DMAT in this section. For a medial mesh, we use **H12** and **M12** to denote the maximum and mean of the minimum Euclidean distances from the vertices on the input mesh to the surface reconstructed from medial mesh, and use **H21** and **M21** to denote the maximum and mean of the minimum Euclidean distances from a set of densely-sampled points on the reconstructed surface to the input surface mesh. And the Hausdorff distance **HD** is defined by: $\mathbf{HD} = \max(\mathbf{H12}, \mathbf{H21})$. Note that **H12**, **M12**, **H21** and **M21** are normalized with respect to the diagonal length of the corresponding surface.

We report the distances to input animated surfaces in Table 1, and compare our method with ASM [TBE16]. For the ASM method, we use the results provided on the author's website. Table 1 shows that our dynamic medial mesh tends to be structurally cleaner than ASM results for all the sequences, which means there are less primitives in the structure when using the same number of spheres.



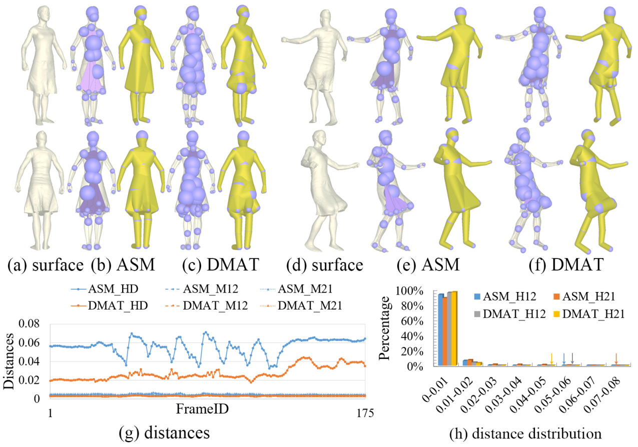
(a) surface (b) ASM (c) DMAT
(d) distances (e) distance distribution

Figure 9: *Comparison with ASM on Cat anim. with 85 spheres.*



(a) surface (b) ASM (c) DMAT
(d) distances (e) distance distribution

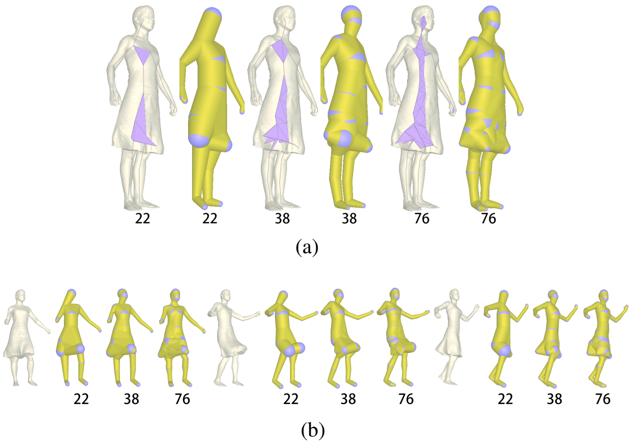
Figure 10: *Comparison with ASM on Horse-gallop anim. with 46 spheres.*



(a) surface (b) ASM (c) DMAT (d) surface (e) ASM (f) DMAT

(g) distances (h) distance distribution

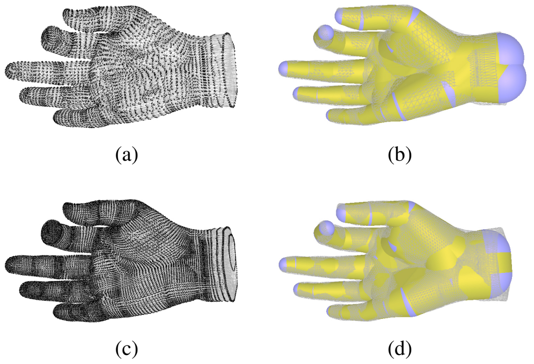
Figure 11: *Comparison with ASM on Samba anim. with 38 spheres.*



(a)

(b)

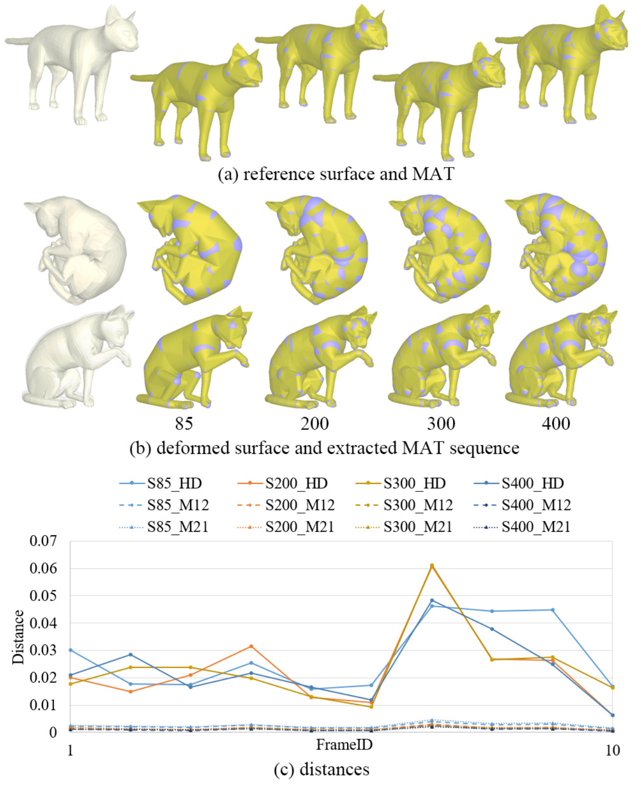
Figure 12: *Samba anim. approximated with dynamic medial mesh with 22, 38 and 76 spheres. (a) reference medial mesh. (b) deformed mesh and the extracted medial meshes.*



(a) (b)

(c) (d)

Figure 13: *Upsampling triangle mesh of the Hand. Initial mesh (a) with 7929 vertices and its related medial mesh (b), and the upsampled mesh (c) with 31710 vertices and its related medial mesh (d).*

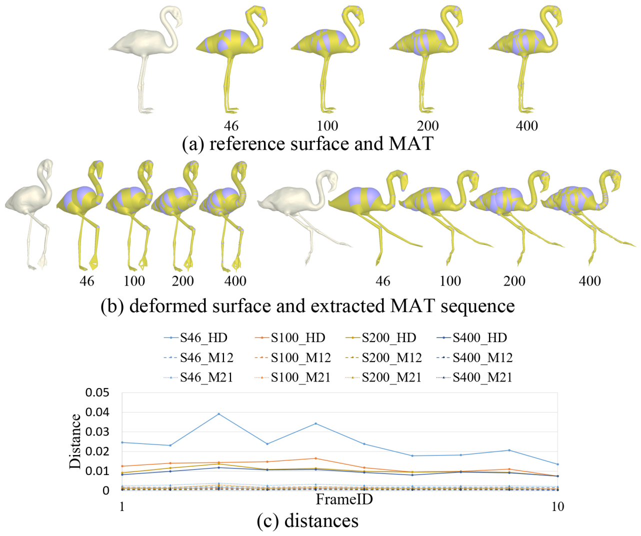


(a) reference surface and MAT

(b) deformed surface and extracted MAT sequence

(c) distances

Figure 14: *DMAT extraction result on Cat anim. with 85, 200, 300 and 400 spheres, represented as S85, S200, S300 and S400 respectively.*



(a) reference surface and MAT

(b) deformed surface and extracted MAT sequence

(c) distances

Figure 15: *DMAT extraction result on Flamingo anim. with 46, 100, 200 and 400 spheres, represented as S46, S100, S200 and S400 respectively.*

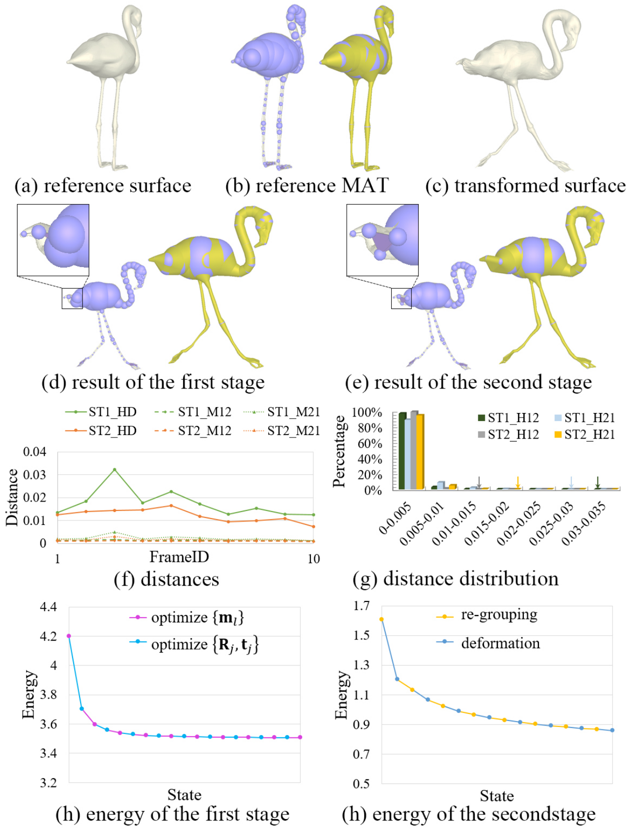


Figure 16: *Comparison on the results of the first (ST1) and the second (ST2) stage on Flamingo anim. with 100 spheres.*

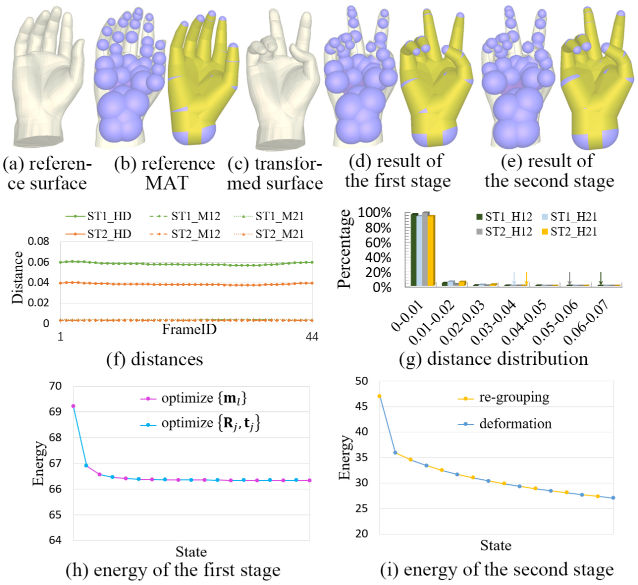


Figure 17: *Comparison on the results of the first (ST1) and the second (ST2) stage on Hand anim. with 34 spheres.*

The results illustrate that our DMAT tends to better approximate "volume-preserving" animations, such as the Hand, Cat, Horse-gallop and Flamingo sequences, as shown in Figures 8–10, especially the middle and ring finger of the hand, belly of the cat and right rear leg of the horse, or animated surfaces with large tubular parts, such as the sweep of the Samba sequence as shown in Figure 11. Our DMAT reconstruction errors for the Samba, Hand and Cat sequences are only around halves of ASM's errors, when approximating with 38, 34 and 85 spheres respectively. Our DMAT tends to have smaller Hausdorff errors as well as smaller mean distances if more primitives are required in order to capture the geometric details of the shape. Figure 12 illustrates that our method can work well on the Samba sequence which is a "volume-preserving" sequence with large tubular parts when setting an appropriate number of spheres for Q-MAT-based simplification of the reference medial mesh. When the number is 38 or 76, the dynamic medial mesh captures not only the structure of arms and legs, but also the fine details of the dress. However, the structure of the reference medial mesh is bad when it is extremely coarse with 22 medial spheres, where the extracted medial meshes are not able to capture the fine details of the dress or the head any more. The Jump sequence is of the same case that the extracted DMAT has large errors due to insufficient number of spheres.

As shown in Figure 13, the sampling density of the input surface mesh is crucial to the DMAT approximation result, and we choose it as follows: a) compute the simplified MAT by Q-MAT, b) compute the error ε_v of each vertex v in the reference frame as: $\varepsilon_v = \left|d_j(v)\right|$. c) upsample the input mesh at the region where ε_v is greater than a given threshold λ, d) iteratively apply a), b) and c) until all errors are below threshold λ. When we use the same number of spheres (34 for this case) for the dynamic medial mesh extraction, it is obvious that upsampling the input surface meshes decreases the errors. We also experiment on using another frame other than the first frame as the reference frame and compare the reconstruction errors with the result of using the first frame as a reference. The results shows the reconstruction errors are not directly affected by the reference surface, but related to the reference medial mesh. Due to the page limit, we put the result and discussion in the Supplementary Appendix.

Figures 14–15 show that our DMAT method can be used on a much denser reference medial mesh, to better capture small features of the surface with more medial spheres and primitives. While increasing the number of spheres in the structure, the errors will decrease on volume-preserving domains of the input surface.

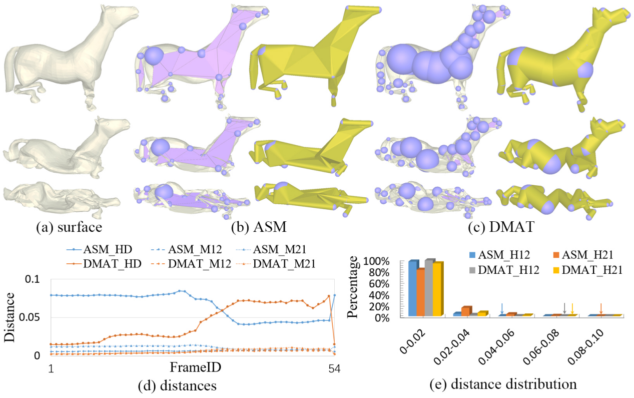
To validate the effectiveness of the second optimization stage, we compare the results of the first and the second stage, as shown in Figures 16–17. We also plot the curves of energy *w.r.t.* iterations in our two stage optimizations, to illustrate the convergence of our alternating energy minimization strategy. For the second stage, "re-grouping" includes re-grouping the correspondences and re-projecting them onto their corresponding medial primitives, while "deformation" represents iteratively optimizing $\{\mathbf{R}_j, \mathbf{t}_j\}$ and $\{\mathbf{m}_l\}$. It shows that the second stage of our optimization method further decreases the energy monotonically.

Table 2 lists the timings of our DMAT computation on various models. It is clear that our method is fast in the first stage, while

Table 2: *Timings for our DMAT approximation algorithm. #V / #F: number of vertices and frames in the input mesh sequence. #S: number of spheres in the output medial mesh. The initialization time (Init.) includes the time for initialization of rotation and translation of medial primitives while excluding the time of computing the reference medial mesh. GCT: total time of grouping the correspondences of medial primitives. RTT: total time of optimizing the rotation and translation of medial primitives. MST: total time of optimizing the medial spheres in both stages.*

Input Anim. (#V /#F)	(#S)	**Init.** (s)	**Stage 1** (s)	**Stage 2** (s)	**Total** (s)	**GCT** (s)	**RTT** (s)	**MST** (s)
Samba(9971/175)	38	0.976	13.766	152.720	167.462	76.627	11.921	57.430
Jump(10002/150)	10	1.048	10.269	83.638	94.955	32.212	10.392	40.591
Flamingo poses(26398/10)	46	0.225	2.418	22.526	25.169	9.395	2.466	9.852
Horse-collapse(8431/48)	46	0.314	3.479	36.641	40.434	16.594	3.115	15.822
Horse-gallop(8431/48)	46	0.324	3.993	38.850	43.167	15.568	3.625	17.874
Horse-gallop(8431/48)	100	0.371	5.018	62.936	68.325	26.269	4.047	28.604
Horse-gallop(8431/48)	200	0.449	6.874	116.681	124.004	44.385	4.265	66.309
Horse-gallop(8431/48)	400	0.640	15.531	295.720	311.891	74.688	6.389	217.486
Hand(7929/44)	34	0.259	3.412	37.030	40.701	18.831	2.844	14.042
Hand(31710/44)	34	1.085	12.634	143.967	157.686	77.069	12.346	49.977
Cat poses(28822/10)	85	0.231	3.568	39.354	43.153	24.585	2.698	11.161
Cat poses(115282/10)	85	1.000	12.454	149.643	163.097	89.814	10.136	43.726

the second stage takes much more time, and most the time (more than 80%) is spent on the re-grouping of correspondences and optimizing the medial spheres, which causes that our method is much slower than ASM [TBE16]. For example, the time for computing the animated sphere-mesh on the Horse-gallop for ASM is 5.394 seconds while it takes 43.167 seconds for computing dynamic medial mesh with the same number of spheres. In our method, the time for re-grouping the correspondence is related to the number of spheres, the sampling density of input animation, as well as the number of iterations. The time for optimizing the medial spheres is relevant to the number of medial spheres as well as the number of iterations. Note that increasing the number of iterations for the second stage will decrease the reconstruction errors of the dynamic medial mesh and increase the time of optimization. A balance should be found between the approximation accuracy of dynamic medial mesh and the efficiency of approximation by selecting a proper number of iterations.


(a) surface (b) ASM (c) DMAT

(d) distances (e) distance distribution

Figure 18: *Comparision with ASM on Horse-collapse anim. with 46 spheres.*

6. Discussion and Future Work

Compared with ASM [TBE16], our DMAT can extract a more accurate and structurally-cleaner medial mesh on “volume-preserving” animated surfaces. However, our current method would fail on mesh collapsing sequences. Figure 18 shows that when working on the Horse-collapse case, our DMAT method fails to compute a plausible dynamic medial mesh and a consistent medial axis won’t be able to approximate animation with severe volume change. In the future, we would like to consider optimizing the connectivity of medial mesh, instead of keeping the connectivity fixed as in our current work, to better reconstruct the collapsing surfaces. Also, we would like to explore the solutions for animation approximation with other volumetric meshes. Besides, our approach relies on the correspondences of medial axis. Once the topology of the input surface has changed, we won’t be able to approximate the animation with current ARAP energy. Therefore, we would like to handle this issue in the future.

Our DMAT method for approximating the input sequence is slow due to its two-stage optimization framework. Since the computation of nearest medial primitive for each surface point is independent to the computation for other points, GPU-based parallel computing methods could be designed on the grouping of correspondences to speed up our second stage of optimization. In addition, we can incorporate a GPU-based iterative solver for sparse linear system to speed up the optimization of medial spheres.

Finally, since our medial mesh is concise and accurate, it has potential to serve as a volumetric structure for other applications such as collision detection, motion modeling, and motion analysis, etc. We would like to investigate those potentials in our future work.

Acknowledgements

The authors would like to thank the anonymous reviewers and Yiqi Cai for their valuable suggestions, Lei Lan for helping with the rendering and Mengwei Lin for making the video. This work was partially supported by the Fundamental Research Funds for the Central Universities in China (No.20720170056) and the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No. BUAA-VR-16KF-22).

References

[AB99] AMENTA N., BERN M.: Surface reconstruction by Voronoi filtering. *Discrete & Computational Geometry 22*, 4 (1999), 481–504. 2, 4

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust, unions of balls, and the medial axis transform. *Computational Geometry Theory & Applications 19*, 2 (2001), 127–153. 2

[AM96] ATTALI D., MONTANVERT A.: Modeling noise for a better simplification of skeletons. In *International Conference on Image Processing, 1996. Proceedings* (1996), vol. 3, pp. 13–16. 2

[ATTS10] AGUIAR E. D., THEOBALT C., THRUN S., SEIDEL H. P.: Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum 27*, 2 (2010), 389–397. 2

[Blu67] BLUM H.: A transformation for extracting new descriptors of shape. *Models for the Preception of Speech & Visual Form 19* (1967), 362–380. 1

[CCT09] CHAUSSARD J., COUPRIE M., TALBOT H.: A discrete λ-medial axis. In *International Conference on Discrete Geometry for Computer Imagery* (2009), pp. 421–433. 2

[CGLX17] CHEN S.-Y., GAO L., LAI Y.-K., XIA S.: Rigidity controllable as-rigid-as-possible shape deformation. *Graphical Models 91* (2017), 13–21. 2

[CK12] CASHMAN T. J., KAI H.: A continuous, editable representation for deforming mesh sequences with separate signals for time, pose and shape. *Computer Graphics Forum 31*, 2 (2012), 735–744. 1

[CL05] CHAZAL F., LIEUTIER A.: The "λ-medial axis". *Graphical Models 67* (2005), 304–331. 2

[DZ02] DEY T. K., ZHAO W.: Approximate medial axis as a Voronoi subcomplex. In *ACM Symposium on Solid Modeling and Applications* (2002), pp. 356–366. 2

[Ede99] EDELSBRUNNER H.: Deformable smooth surface design. *Discrete & Computational Geometry 21*, 1 (1999), 87–115. 3

[FLM03] FOSKEY M., LIN M. C., MANOCHA D.: Efficient computation of a simplified medial axis. In *Eighth ACM Symposium on Solid Modeling and Applications 2003, Seattle, Washington, Usa, June* (2003), pp. 96–107. 2

[FTB13] FARAJ N., THIERY J. M., BOUBEKEUR T.: Progressive medial axis filtration. In *SIGGRAPH Asia 2013 Technical Briefs* (2013), p. 3. 2

[GB18] GRESHAKE S. H., BRONSART R.: Application of subdivision surfaces in ship hull form modeling. *Computer-Aided Design 100* (2018), 79 – 92. 1

[GMHP97] GARLAND, MICHAEL, HECKBERT, PAUL S.: Surface simplification using quadric error metrics. *Proc. SIGGRAPH 1997* (1997), 209–216. 2, 4

[HJCW06] HONG M., JUNG S., CHOI M. H., WELCH S. W. J.: Fast volume preservation for a mass-spring system. *IEEE Computer Graphics & Applications 26*, 5 (2006), 83. 1

[HRS10] HASLER N., ROSENHAHN B., SEIDEL H. P.: Learning skeletons for shape and pose. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2010), pp. 23–30. 1, 2

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics 24*, 3 (July 2005), 1134–1141. 2

[JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. *ACM Transactions on Graphics 30*, 6 (2011), 165:1–165:8. 2

[JZP*08] JU T., ZHOU Q. Y., PANNE M. V. D., COHEN-OR D., NEUMANN U.: Reusable skinning templates using cage-based deformations. *ACM Transactions on Graphics 27*, 5 (2008), 1–10. 1

[KCvO08] KAVAN L., COLLINS S., ŽÁRA J., O'SULLIVAN C.: Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics 27*, 4 (2008), 105. 2

[KOF04] KIRK A. G., O'BRIEN J. F., FORSYTH D. A.: Skeletal parameter estimation from optical motion capture data. In *IEEE Conference on Computer Vision & Pattern Recognition* (2004), p. 29. 1, 2

[Kv05] KAVAN L., ŽÁRA J.: Spherical blend skinning: A real-time deformation of articulated models. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games* (2005), pp. 9–16. 2

[LD12] LE B. H., DENG Z.: Smooth skinning decomposition with rigid bones. *ACM Transactions on Graphics 31*, 6 (2012), 1–10. 1, 2

[LD14] LE B. H., DENG Z.: Robust and accurate skeletal rigging from mesh sequences. *ACM Transactions on Graphics 33*, 4 (2014), 1–10. 1, 2

[LWS*15] LI P., WANG B., SUN F., GUO X., ZHANG C., WANG W.: Q-MAT: Computing medial axis transform by quadratic error minimization. *ACM Transactions on Graphics 35*, 1 (2015), 8. 1, 2, 4, 5, 6

[LYHG17] LAN L., YAO J., HUANG P., GUO X.: Medial-axis-driven shape deformation with volume preservation. *Visual Computer 33*, 4 (2017), 1–12. 2

[MGP10] MIKLOS B., GIESEN J., PAULY M.: Discrete scale axis representations for 3D geometry. *ACM Transactions on Graphics 29*, 4 (July 2010), 101:1–101:10. 2

[MTLT88] MAGNENAT-THALMANN N., LAPERRIÈRE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. *Proc Graphic Interface '88* (1988), 26–33. 2

[SCYW15] SUN F., CHOI Y., YU Y., WANG W.: Medial meshes: a compact and accurate medial shape representation. *IEEE Transactions on Visualization & Computer Graphics 22*, 3 (2015), 1278–1290. 2

[SFM05] SUD A., FOSKEY M., MANOCHA D.: Homotopy-preserving medial axis simplification. In *ACM Symposium on Solid and Physical Modeling* (2005), pp. 39–50. 2

[Sor17] SORKINE O.: Least-squares rigid motion using SVD. *Technical notes* (2017). 6

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. In *Conference on Computer Graphics & Interactive Techniques* (1986), pp. 151–160. 1

[SP08] SIDDIQI K., PIZER S.: Medial representations: Mathematics, algorithms and applications. *Computational Imaging & Vision 37* (2008). 2

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. *ACM Transactions on Graphics 26*, 3 (July 2007). 2

[TBE16] THIERY J. M., BOUBEKEUR T., EISEMANN E.: Animated mesh approximation with sphere-meshes. *ACM Transactions on Graphics 35*, 3 (2016), 30. 1, 3, 7, 10

[TGB13] THIERY J.-M., GUY E., BOUBEKEUR T.: Sphere-meshes: shape approximation using spherical quadric error metrics. *ACM Transactions on Graphics 32*, 6 (2013), 178:1–178:12. 2, 3

[TPT16] TKACH A., PAULY M., TAGLIASACCHI A.: Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics 35*, 6 (2016), 1–11. 2

[YSC*16] YAN Y., SYKES K., CHAMBERS E., LETSCHER D., JU T.: Erosion thickness on medial axes of 3D shapes. *ACM Transactions on Graphics 35*, 4 (2016), 38. 1, 2