DMAT: Deformable Medial Axis Transform for Animated Mesh Approximation

Baorong Yang\textsuperscript{1}, Junfeng Yao\textsuperscript{1,†} and Xiaohu Guo\textsuperscript{1,‡}

\textsuperscript{1}Xiamen University, China
\textsuperscript{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 \textit{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

\hspace{1em} 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 \textit{et al.} [TBE16] innovatively proposed a volumetric structure called \textit{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, \textit{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 \textit{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-
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 structurally 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 an-tervalex is smaller than a user-specified threshold.

For a surface vertex \( v_i \) and a medial primitive \( P_j \), we define the projection of \( v_i \) onto \( P_j \) based on the minimizer of the following squared distances \( E_d(m) \):

\[
E_d(m) = \left\| (v_i - c) \cdot n_{ij} - r \right\|^2,
\]

where \( m = \{c, r\} \) is a sphere on the medial primitive, and \( n_{ij} \) is the outward normal of primitive \( P_j \) at \( v_i \), as shown in Figure 2. The details for computing the outward normal \( n_{ij} \) is given in the Supplementary Appendix. The sphere \( m \) minimizing \( E_d(m) \) is defined as the footprint of \( v_i \) on \( P_j \).

Without loss of generality, let us consider \( P_j \) being a medial cone \( e_{ij} \). In this case, \( m = \alpha m_i + (1 - \alpha) m_j \). By replacing \( e_{ij} \) and \( r_{ij} \)
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.

Then \( \alpha \) of the purple footprint to the yellow surface vertex.

\[ E_d(\alpha) = \left\| (v_i - c_i - \alpha(c_k - c_j)) \cdot m_{ij} - r_j - \alpha(r_k - r_j) \right\|^2. \]  \hfill (2)

Then \( \alpha \) can be solved by \( \frac{dE_d(\alpha)}{d\alpha} = 0 \). 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 \( \{S[t] = 0 \ldots m\} \) of \( m \) frames, our computed DMAT consists of a medial mesh sequence \( \{M'[t = 0 \ldots m]\} \) 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 \( C \) of medial mesh \( M \) can be used to approximate the given surface \( S \) [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 simplifed MAT \( M^0 \) for the referenced triangle mesh \( S^0 \), composed of \( n \) medial spheres \( \{m^0_i[k = 0 \ldots n]\} \). The computation of \( M^0 \) will be discussed in Sec. 4.1. Then we apply a two-stage ICP optimization method for computing the deformed medial spheres \( \{m_i[k = 0 \ldots n]\} \) of medial mesh \( M^t \) 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 \( C_j \) for primitive \( P_j \). The detail for grouping the correspondences will be discussed in Sec. 4.2.

For each corresponding vertex \( v_i \in C_j \) on the surface \( S \), we compute its projected footprint \( m_{ij} = \{c_i, r_j\} \) on medial primitive \( P_j \), and would like to maintain its “relative position” to \( P_j \) as much as possible through an energy optimization. Since the footprint is on medial primitive, it can be represented as follows:

\[ m_{ij} = \sum_{m \in V_i} \alpha_{ij} m_c, \]  \hfill (3)

where \( V_i \) is the set of medial vertices for medial primitive \( P_j \), and \( \{\alpha_{ij}\} \) are the barycentric coordinates of \( m_{ij} \). We take the barycentric coordinates \( \{\alpha_{ij}\} \) as the “relative position” of \( v_i \) w.r.t. \( P_j \). If we keep their “relative position” to be fixed, i.e., fixing \( \{\alpha_{ij}\} \), then the footprint \( m_{ij} \) of each vertex can be simply interpolated from the medial vertices of this primitive. For each vertex \( v_i \in C_j \), we define the vector from the center \( c_i \) of \( m_{ij} \) to \( v_i \) to be its footprint-ray:

\[ s_i = v_i - c_i. \]

We consider the first frame (frame 0) to be a reference frame. When the corresponding set of vertices \( C_j \) 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: \( Q_j = \|R_j s_i^0 - s_i\|^2 \), by assuming the primitive \( P_j \) is undergoing a rotation \( R_j \), and \( s_i^0 \) is the footprint-ray in the reference frame. In this way, the deformed projective spheres \( m_{ij} \) can be used to drive the deformation of medial primitives \( P_j \). Thus, for each medial primitive \( P_j \), we can define the ARAP footprint-ray energy \( Q_j \) as the following summation:

\[ Q_j = \sum_{v_i \in C_j} Q_{ij} \]
\[ = \sum_{v_i \in C_j} \left\| R_j s_i^0 - s_i \right\|^2 \]  \hfill (4)
\[ = \sum_{v_i \in C_j} \left\| R_j s_i^0 - v_i + \sum_{m \in V_i} \alpha_{ij} m_c \right\|^2. \]

We assume each medial primitive \( P_j \) has an “ideal” rigid-body motion with rotation \( R_j \) and translation \( t_j \). Then we can define the following ARAP medial primitive energy \( W_j \) as the following summation:

\[ W_j = \sum_{m \in V_j} \left\| R_j c_i^0 + t_j - c_i \right\|^2, \]  \hfill (5)

where \( c_i \) is the center position of medial vertex \( m_i \) at deformed frame \( t, c_i^0 \) 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:

\[ E \left( \{R_j, t_j\}, \{m_i\} \right) = \sum_j (Q_j + \omega W_j), \]  \hfill (6)

where the variables include the rigid-body motions \( \{R_j, t_j\} \) of all medial primitives \( \{P_j\} \), and the center positions \( \{c_i\} \) of all medial vertices \( \{m_i\} \). \( \omega \) is a weighting factor used to balance two energy
terms. In all of our experiments, we simply set \( w = 1 \). 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 \( \{ m_i \} \), we can solve for the rigid-body motion \( \{ R_i, t_i \} \) of all medial primitives; (2) by fixing \( \{ R_i, t_i \} \) of all medial primitives, we can solve for all medial vertices \( \{ m_j \} \).

### 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 \( v_i \) w.r.t. medial primitive \( P_j \) are fixed, i.e., in Eq. (3) the barycentric coordinates of \( m_j \) 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:

\[
Q_{ij} = \sum_{v_i \in C_j} \left( R_i u_{ij} \left( \sum_{m_k \in V_i} a_{ijkl} r_k \right) - v_i \right) \sum_{m_k \in V_i} a_{ijkl} r_k^2
\]

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).

- In the first stage, we fix the “relative position” of surface vertex \( v_i \) w.r.t. medial primitive \( P_j \), by using the same barycentric coordinates \( \{ a_{ijkl} \} \) computed from the projected sphere \( m_j^0 \) in the reference frame, and use \( u_{ij} = s_i^0/||s_i^0|| \) for \( Q_{ij} \) in Eq. (7). In this stage, we optimize for the rigid-body motions \( \{ R_i, t_i \} \) of all medial primitives, and the center positions of all medial vertices \( \{ m_i \} \), 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 \( v_i \) w.r.t. medial primitive \( P_j \), by allowing \( v_i \) to be regrouped to a “best-fit” medial primitive which is the closest medial primitive of \( v_i \), 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 \( u_{ij} \) for \( Q_{ij} \) in Eq. (7) as:

\[
u_{ij} = R_i^{-1} m_j^0 \]

where \( R_i \) is the current optimized rotation for medial primitive and \( m_j^0 \) 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 \( u_{ij} \) for these two stages, it is not difficult to show that the energy \( Q_{ij} \) is actually consistent. For solving the minimization problem, we adopt an iterative way for optimizing the rigid motions \( \{ R_i, t_i \} \) of medial primitives and the medial vertices \( \{ m_i \} \). 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 mesh has been discussed. We now describe in detail how to compute the reference medial mesh, partition the vertices on triangle mesh \( S \) 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 \( S^0 \) by using the method of Amenta and Bern [AB99], and use Q-MAT [LWS+15] to remove redundant spikes and simplify it to obtain the medial mesh \( M^0 \).

In Amenta and Bern’s approach [AB99], the initial medial axis is computed as the dual of Delaunay triangulation 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 \( m^0 \), denoted as \( C(m^0) \). Q-MAT follows the QEM framework [GMHP97] with edge-collapses. For each contraction, an edge \( e_i^0 \) is contracted to a new medial vertex \( m_i^0 \), and the correspondences are merged by \( C(m_i^0) = C(m_i^0) \cup C(m_i^0) \). Note
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.

4.2. Partition of Correspondences

We use the following signed distance \( d_j(v_i) \) from a surface point \( v_i \) to medial primitive \( P_j \), as a guidance to partition the surface into independent regions grouped by medial primitives:

\[
d_j(v_i) = (v_i - c_{ij}) \cdot n_{ij} - r_{ij},
\]

where \( m_{ij} = \{c_{ij}, r_{ij}\} \) is the footprint of \( v_i \), and \( n_{ij} \) is the outward normal of primitive \( P_j \) at \( v_i \). Similar to a Voronoi cell for Euclidean distance, the correspondences \( C_j \) stores the closest surface points for \( P_j \) 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 \( P_j \) and
\(P_k\) are neighbors and share either a cone or a sphere. For a nearby surface point \(v_i\), we would like to decide whether assigning \(v_i\) to \(P_j\) or to \(P_k\).

In the first case, suppose \(v_i\) has an inner-footprint \(m_{ij}\) on \(P_j\), and an inner-footprint \(m_{ik}\) on \(P_k\). We simply assign \(v_i\) to the medial primitive that has smaller signed distance of Eq. (9).

In the second case, suppose \(v_i\) has an inner-footprint \(m_{ij}\) on \(P_j\), and an outer-footprint \(m_{ik}\) on \(P_k\). In this case, we simply assign \(v_i\) to primitive \(P_j\).

**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.

The third case is when the footprints of \(v_i\) on both \(P_j\) and \(P_k\) are the same, i.e., \(m_{ij} = m_{ik}\). In this case, both \(m_{ij}\) and \(m_{ik}\) 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 \(v_i\) to the two boundary planes of the two connected cones, and assign \(v_i\) 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 \(v_i\) to the two boundary planes of the two connected slabs, and assign \(v_i\) 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 \(v_i\), we assign \(v_i\) to the slab, and vice versa.

- **Two medial slabs share a sphere.** We sum up the two signed distance between \(v_i\) and the boundary planes of two cones of each medial slab, and assign \(v_i\) 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:

\[
t_j = \frac{\sum_{m \in C_j} c_j}{|C_j|},
\]

where \(C_j\) is the set of medial vertices for primitive \(P_j\), and \(|C_j|\) is the number of medial vertices.

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

After getting the rotation and translation \(\{R_j, t_j\}\) for medial primitives, we can solve for all medial vertices \(\{m_l\}\) by minimizing \(E\) in Eq. (6) with fixed \(\{R_j, t_j\}\), 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 \(r_j\) and solve for the centers \(c_j\) of all medial vertices only. We perform \(N_1\) number of iterations of computing \(\{R_j, t_j\}\) and \(c_j\) 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 \(\mathcal{N}(m_l)\) of medial vertex \(m_l\):

\[
R_l = \max_{v_i \in C_j} \max_{P \in \mathcal{N}(m_l)} r_{ij},
\]

where \(r_{ij}\) is the radius of the projective sphere \(m_{ij}\) of surface vertex \(v_i\) on primitive \(P_j\). After solving the medial vertices \(\{m_l\}\) 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 \(\{R_j, t_j\}\) for medial primitives and \(\{c_j, r_j\}\) 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-projection, we perform \(N_3\) number of iterations of computing \(\{R_j, t_j\}\) and \(\{m_l\}\). 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 [LWS15], 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.

<table>
<thead>
<tr>
<th>Input Anim.</th>
<th>DMAT</th>
<th>ASM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#S</td>
<td>#E</td>
</tr>
<tr>
<td>Hand</td>
<td>34</td>
<td>18</td>
</tr>
<tr>
<td>Cat poses</td>
<td>85</td>
<td>17</td>
</tr>
<tr>
<td>Horse-gallop</td>
<td>46</td>
<td>30</td>
</tr>
<tr>
<td>Flamingo poses</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Samba</td>
<td>38</td>
<td>21</td>
</tr>
<tr>
<td>Jump</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Horse-collapse</td>
<td>46</td>
<td>28</td>
</tr>
</tbody>
</table>

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

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

Figure 10: Comparison with ASM on Horse-gallop anim. with 46 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: HD = max (H12, 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 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.
Figure 11: Comparison with ASM on Samba anim. with 38 spheres.

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.

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).

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

Figure 15: DMAT extraction result on Flamingo anim. with 46, 100, 200 and 400 spheres, represented as S46, S100, S200 and S400 respectively.
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 $\varepsilon_v$ of each vertex $v$ in the reference frame as: $\varepsilon_v = |d_j(v)|$. c) upsample the input mesh at the region where $\varepsilon_v$ is greater than a given threshold $\lambda$, d) iteratively apply a), b) and c) until all errors are below threshold $\lambda$. 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 show 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 $\{R_j,t_j\}$ and $\{m_i\}$. 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
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.

### 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 ren-
dering 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


[CL05] CHAZAL F., LIEUTIER A.: The $\kappa$-medial axis”. Graphical Models 67 (2005), 304–331. 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


© 2018 The Author(s)
Computer Graphics Forum © 2018 The Eurographics Association and John Wiley & Sons Ltd.