Context-aware Volume Modeling of Skeletal Muscles

Zhicheng Yan†  Wei Chen‡  Aidong Lu§  David S. Ebert¶

1State Key Lab of CAD&CG, Zhejiang University, China
2University of North Carolina at Charlotte, USA
3Purdue University, USA

Abstract
This paper presents an interactive volume modeling method that constructs skeletal muscles from an existing volumetric dataset. Our approach provides users with an intuitive modeling interface and produces compelling results that conform to the characteristic anatomy in the input volume. The algorithmic core of our method is an intuitive anatomy classification approach, suited to accommodate spatial constraints on the muscle volume. The presented work is useful in illustrative visualization, volumetric information fusion and volume illustration that involve muscle modeling, where the spatial context should be faithfully preserved.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations—

1. Introduction
Modeling and illustrating muscle volumes are of great interests in many applications, such as medical illustration, treatment planning for musculoskeletal disorder, and anatomical dynamics simulation [STS90, SPCM97, BTS*05]. In this paper, we address a challenging problem of interactive volume modeling of skeletal muscles, which is to construct outlier shapes and internal fibrous structures of skeletal muscles in the volumetric representation. The challenge is to achieve the quality currently attainable from three-dimensional modeling systems with simple yet effective interactions.

Existing geometric modeling approaches [SPCM97, KHS01] exploit the anatomical structures provided by an underlying skeleton to construct reliable muscle geometry in the three-dimensional space. They will require a conversion between geometric models and volumetric representations if real-world anatomy is used. In the meantime, many solutions [BTS*05, TSB*05] build musculoskeletal systems by taking biomechanics into account. They could be slow and labor intensive, especially when there is complex anatomy around the muscle volume.

Volume modeling approaches that work directly with measured volumetric datasets can be very sophisticated in terms of efficiency and fidelity. Many intuitive interfaces [SS04, CSC06, BGKG06, CCI*07] have been introduced to reduce user interactions for general volume data to a great extent, yet without sacrificing the rendering quality. However, this trend has not happened in muscle modeling, mainly because modeling muscle volume requires a correct anatomical context. A better user interface is needed to accurately specify the relationships between underlying muscle and surrounding anatomy.

Specifically, there are two important properties required for a muscle volume modeling system. On one hand, the modeled muscle should conform to its anatomy context, e.g., not intersecting with bones and preserving the anatomical relationships. On the other hand, the user interactions should be operated on a simple form (two-dimensional interface), which provides great convenience to the non-professional modeling users such as medical illustrators.

In this paper, we present a novel volume modeling system for designing and editing skeletal muscles based on an input volume. Our focus is not the accuracy and performance of the shape modeling that have been extensively studied. In-
stead, we are seeking an optimal two-dimensional painting interface that allows users to easily classify an input volume, and thereby discover the anatomical context useful for modeling. This offers effective simplification of modeling operations, and it is generally suitable for novices. With our system, we demonstrate how users can classify volume dataset and create illustrative visualization of muscle volume in an intuitive fashion.

The rest of this paper is organized as follows. We review related work in Section 2. Section 3 introduces the pipeline. A novel semi-supervised anatomy classification method is explained in Section 4. Implementation details are provided in Section 5. We present results for a variety of examples in Section 6, and conclude the paper in Section 7.

2. Related Work

Muscle modeling Earlier muscle modeling methods [KHS01] have concentrated on using geometric shapes to edit muscle models and generate animations. For example, the anatomy-based muscle modeling technique [SPCM97] generates and edits muscle models by exploiting the relationship between exterior forms and corresponding underlying structures. They represent a muscle with a set of parameters including location, orientation, general shape and volume, which can be flexibly modified to account for the deformation and movement of an underlying articulated skeleton. To achieve a high degree of realism, we think that the generation of muscle models should incorporate precise knowledge of anatomy.

In terms of physical reality, data-driven muscle modeling methods are inherently superior to geometric modeling methods. For instance, three-dimensional finite-element models of muscle [BTS+05] can improve the representations of muscles with complex geometry. Meanwhile, real-world volume datasets can also be employed to construct high-fidelity muscle models [TSB+05] by taking biomechanics into account.

Note that these schemes perform well in many biomedical applications, although they typically require intensive user interactions and certain anatomy knowledge. This paper compliments existing methods with an intuitive anatomy classification interface. All studied muscle simulation and manipulation techniques can be used in our system, although not explicitly shown in our results.

Context-aware volume modeling Volume modeling is a fundamental problem in volume visualization [CCT+07]. Extensive research has been engaged in the segmentation, manipulation, and deformation of volumetric datasets [MTB03, CCT+07]. For example, Singh and Silver [SS04] proposed a method that allowed users to choose geometrical components and highlight them to depict the context embedded in a volume dataset effectively. The feature-aligned volume manipulation approach [CSC06] provides an interactive volume deformation interface which effectively enables illustrative visualization. The VolumeShop system [BG05] makes full use of volume modeling techniques to create a fully dynamic three-dimensional volume illustration user interface. In [BKW08], an interactive volume editing and painting system with an efficient three-dimensional brushing toolkit is described. Our approach provides an add-on to these existing methods by enhancing the visual expressiveness with a modeled muscle volume.

Interactive volume classification A transfer function [PLB+01] maps data values onto optical properties. Designing a transfer function is logically identical to the process of identifying features of interest, namely, volume classification. However, standard histogram-based transfer function design can hardly incorporate human intervention into this pattern-recognition process. There has been a vast body of medical segmentation methods [FRB07, SWY99, ZBS01]. By regarding a volume data as a three-dimensional image, existing image classification and segmentation approaches [BJ01, NN04, RLC04, FRB07] could be extended to volume classification. Challenges in this include how to quickly obtain the user specification and how to supervise the classification procedure.

Recent work makes great progress to fill the gap between high-level perception and low-level features by means of advanced supervised learning techniques. The pioneering work by Tseng et al. [TLM05] allows the user to specify semantic regions on some volume slices and forward this information to a classifier for high-dimensional classification. With an attempt to simplify the user interactions, volume catcher [ONI05] assumes that the classification boundaries form a silhouette in the rendered image. The user only needs to specify the silhouette in two-dimensional images to generate a set of seeding points for subsequent three-dimensional volume classification. The volume cutout approach [YZNC05] elegantly combines the advantages of the stroke-based user interface and graphcut based segmentation algorithms. A recently developed volume coloring approach [BKW08] works in the three-dimensional domain, facilitating semi-automatic classification and segmentation of opaque iso-surfaces. Our approach is different from these methods in that it is augmented by the visualization results, and only requires simple point-based specification in two-dimensional images.

3. Overview

As depicted in Figure 1, our muscle modeling system consists of two stages: volume classification and muscle construction. The input of our approach includes a volume dataset and its associated transfer function. The user determines semantically different regions by marking points on the rendered image. These indications are extended into the three-dimensional space along the viewing direction, resulting in a collection of sampling points. Subsequently, we em-
ploy a semi-supervised learning algorithm to classify the set of the sampling points and the marking points in the image plane, and thus label the voxels in which the sampling points lie. Then, we use a statistical region merging approach to perform volume classification. The volume classification can be incrementally enhanced by specifying more marking points under different viewports or transfer functions.

The muscle modeling is performed interactively with a set of modeling widgets including the position locator, the slicing metaphor, and the shape generator. Every operation is performed in two-dimensional space. The classification results give the user a clear anatomical context for muscle design, such as the permitted size, location and orientation of muscle volume as well as the spatial relationship with other objects. The modeled muscle geometry is converted into a list of fibers, and reformulated into a volume representation.

4. Intuitive Anatomy Classification

With the assistance of volume visualization, the user could recognize the anatomical structures and design the intended muscle volume. Therefore, it is highly desirable that both tasks are performed following the intuitions of the user. In our approach the user is required to only specify a list of points on the rendered image to indicate schematically meaningful regions. The visual information guides the user to directly manipulate and justify the classification. Meanwhile, the classification results in turn provide effective hints on optimizing transfer functions because the colors are derived from the employed transfer function.

4.1. Region labeling in two-dimensional space

When a set of labeled points are drawn in the rendered image, they are cast into the volume space along the viewing direction (Figure 2). To capture all potential regions, densely distributed points are first generated along each ray, together with their colors and opacities under the underlying transfer function. We group the sampling points sequentially along each ray based on their similarity in the RGB color space, and choose several representative samples from each grouped set. The voxels corresponding to those representative samples are considered as unlabeled points. Note that, the colors of the labeled points are different from those of the sampling points.

Figure 2: In the image plane, five points are labeled as three classes. Other points are unlabeled.

4.2. Semi-supervised classification in the color space

We now have two sets of points, namely, the labeled points in the rendered image, and the unlabeled sampling points in the volume space. Our classification scheme is inspired by the following assumption: one observed color is the composition of the colors of its associated sampling points, and statistically tends to be similar to the ones that dominate the composition. As a consequence, we employ a semi-supervised learning algorithm to classify the unlabeled sampling points in RGB color space.

For the sake of brevity we describe our method with the similar notations from [ZGL03]. By regarding each point as a node, we construct a connected graph $G = (V, E)$ where $V$ consists of $n$ points, of which $L = \{l_1, ..., l_l\}$ are the labeled points with labels $y_1, ..., y_l$, and $U = \{l_1 + 1, ..., l + u\}$ are the unlabeled points. In Figure 3 (a), five labeled points...
are displayed in the first column, while the remaining ones are unlabeled. Each labeled point is composed of the ones in the same row.

Figure 3: Illustration of our semi-supervised classification algorithm. Various glyphs are used to encode different classes.

To assign a label to each node in $U$, we compute an $n \times n$ symmetric weight matrix $W$ for the edges of $G$:

$$\omega_{ij} = \exp(- \sum_{d=1}^{3} (x^d_i - x^d_j)^2)$$  \hspace{1cm} (1)

where $x^d_i$ is the $d$-th color channel of the $i$th point $x_i$. The objective is to compute a real-valued function $f : V \rightarrow \mathbb{R}$ on $G$ to satisfy the property that $f(i) = f(i) \equiv y_i$ on the labeled data $i = 1, \ldots, l$, and to minimize a quadratic energy function:

$$E(f) = \frac{1}{2} \sum_{ij} \omega_{ij} (f(i) - f(j))^2$$  \hspace{1cm} (2)

The minimum energy function $f^* = \arg \min_{f \in \mathcal{F}} E(f)$ is harmonic in the sense that it is identical to $f_i$ on the labeled point sets $L$ and satisfies $\Delta f = 0$ on the unlabeled point set $U$. $\Delta$ is the combinatorial Laplacian operator and can be represented in a matrix form $\Delta = D - W$, where $D = \text{diag}(d_i)$ is the diagonal matrix with entries $d_i = \sum_j w_{ij}$, and $W = [w_{ij}]$ is the weight matrix in Equation 2. We rewrite $\Delta f = 0$ as $f = Pf$, where $P = D^{-1}W$.

By dividing the matrix $W$, $D$ and $P$ into four blocks along the $l$th row and column, we have:

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}$$  \hspace{1cm} (3)

With the denotation $f = \begin{bmatrix} f_l \\ f_u \end{bmatrix}$ where $f_u$ represents the values on the unlabeled data points, the harmonic solution $\Delta f = 0$ is given by:

$$f_u = (D_{uu} - W_{uu})^{-1}W_{ul}f_l = (I - P_{uu})^{-1}P_{ul}f_l$$  \hspace{1cm} (4)

where $I$ is an $n \times n$ identity matrix.

Suppose that there are $m$ regions $\{r_1, r_2, r_3, \ldots, r_m\}$ specified by the user. $f_u$ gives a probability $f_u(j)$ to each unknown point. We determine its label (Figure 3 (b)) by applying a threshold $\theta$ to $f_u$:

$$x_i \in r_j \quad \text{if} \quad |f_u(j) - \theta| < \theta \quad j \in 1, 2, 3, \ldots, m$$  \hspace{1cm} (5)

A typical choice for $\theta$ is 0.2. We ignore the points that do not belong to any region. This guarantees that only the points approximating the observed colors are labeled, yielding a conservative classification result.

4.3. Statistical region merging

In the next step, all voxels are classified with a statistical region merging algorithm [NN04] by using the identified sampling points as the seeding points. Rather than using solely the similarity of the scalar values as the merging criteria, we use the scalar and gradient magnitude of voxels to guide the region merging. The classified volume is then rendered by assigning different colors to individual regions. The user can justify the result and improve it by changing viewpoints or transfer functions, and label additional points in the image plane to trigger the modifications to the classification. This process can be repeated until satisfying results are achieved. For a $256 \times 128 \times 256$ volume dataset and 800 unlabeled points, our un-optimized implementation consumes about 1.5 seconds to perform semi-supervised learning, and 5 seconds for statistical region merging. Therefore, the entire anatomy classification could be accomplished in several minutes, even with multiple iterations.

5. Modeling of the Skeletal Muscles

Before the muscle modeling, the user can check the classified anatomies to get a clear context. Similar to the classification process, the modeling operations are fully carried out in the two-dimensional space by leveraging the context built from the anatomy classification. In the following sections, we describe how a user can easily generate the shape, internal structures and the volume of a muscle model with a group of convenient widgets.

5.1. Widgets for modeling the muscle geometry

A set of muscle geometry modeling widgets are designed, including:

- **The slicing metaphor** A slicing metaphor is used to adjust the slicing plane to eliminate the occlusion, and as a designing space in which the user specifies the muscle shape. It could be translated, uniformly scaled or rotated around its local axes (Figure 4 (a)). In our system, one slicing plane is sufficient to generate a skeletal muscle.

- **The shape generator and modifier** The shape of a muscle is described as a surface formed from a medial axis and several contour curves produced from the user-specified strokes. Guided by the display in the volume rendering window, the user manipulates a slicing plane and draws two strokes in the slicing plane to indicate the muscle contour. Our system converts the strokes into B-spline curves and highlights their controlling points whose positions can be interactively altered. Then the B-spline curves are...
rotated along a computed medial axis to produce multiple contour lines, forming the geometry of a muscle (Figure 4 (b) and (c)).

- **The position editor** In addition to conventional rotating, scaling and translating operators, the modeled muscle geometry can be arbitrarily located, added, deleted and cloned. Figure 4 (d) shows the muscles with simple cloning and modification operations.

![Image of muscles showing different positions](image_url)

**Figure 4:** Widgets used for modeling the muscle geometry. (a) The slicing metaphor; (b) The shape generator in a slicing plane; (c) The shape generator in a volume rendering window; (d) The position editor.

All these widgets enable the user to explicitly determine the muscle volume without a massive amount of efforts. A huge benefit from the anatomy classification is its available context. Specifically, we employ the following schemes to optimize the modeling operations:

- **Relationship** The user can study the relationships among anatomies by visualizing the classified results, e.g., the one shown in Figure 1. A rough planning on how to place the muscle volume could be formed.
- **Location and size** Typically, classified regions include skin, bone, vessel, muscle, air and other unclassified ones. Based on the locations of the skin and bone regions, the user is allowed to interactively compute the axes of the slicing planes, and estimate the bounding box of the underlying muscle volume.
- **Penetration** Biologically correct muscle volumes should be embedded in the classified muscle region. By assigning a mask to each voxel, our system automatically prevents the penetrations of the designed muscle into other regions when specifying the controlling vertices of the B-spline curves. Given a vertex sampled from the muscle contour specified by the user, all voxels formed by rotating the vertex along the medial axis are checked. If all of them belong to the muscle region, the position of this vertex is permitted. Otherwise, the user has to choose another conservative position. Figure 5 illustrates an example of preventing penetration.

![Image of fibers with and without penetration](image_url)

**Figure 5:** Designing fibers (a) without and (b) with preventing the penetration. In both cases, the red lines indicate prohibited drawings.

Our experimental results demonstrate that these schemes facilitate generating satisfying muscle volumes. More sophisticated and complicated contexts can be used for professional modeling users, as introduced in [BTS’05, TSB’05].

### 5.2. Modeling the muscle outlier shape

To model the shape of a skeletal muscle, we compute several B-spline curves from the controlling points, and evenly sample a collection of points on the curves. To find the medial axis, we pair the sample points on two curves and compute their midpoints. Then they are sequentially connected to form the medial axis (Figure 6 (a)). To generate more contours, we rotate the existing contours along the medial axis, with which the scaling of the contour deviation from the medial axis can be easily decided. In addition, we build a local coordinate system of the muscle by orienting the axis to be along the z-axis of the volume space (Figure 6 (b)).

![Image of generating internal fibers](image_url)

**Figure 6:** Generating the outlier shape and internal fibers of a skeletal muscle.
5.3. Modeling the internal muscle structures

Generating the internal muscle structures is performed in two steps. By assuming that the muscle cross-section is perpendicular to the z-axis of the local coordinate system, we sequentially sample a list of points on the contour curves and group them into a sequence of cross-sections. Subsequently the boundary of each cross-section is generated (Figure 6 (c)). For the plane of each cross-section, we interpolate a set of points inside the boundary formed by the contour curves. Thereafter, a number of fibers are generated by connecting all points along the medial axis (Figure 6 (d)). The set of the generated fibers forms a fiber bundle, and represents a skeletal muscle. Our system also allows the user to interactively or automatically choose the rear parts of a fiber as a tendon component of a muscle, e.g., the white parts shown in the final results.

5.4. Modeling of the muscle volume

We convert each fiber into a volumetric representation using the three-dimensional gaussian-weighted line voxelization algorithm [WKZL04], and replace the corresponding parts in the input volume. A mask is assigned to each voxel to indicate its classification. Specifically, the tendon part of each fiber is denoted with a different mask in the volume. To allow for line-based illumination [Ban94], the direction of each fiber is recorded, and replaces the per-voxel gradient during volume illustration.

6. Results and Discussions

We applied the proposed approach to four volume datasets. All experimental results were collected at a PC equipped with an Intel Core 2 Duo E6600 2.4 GHz CPU, 3G host memory and Nvidia Geforce GTX 280 graphics card. Table 1 summarizes the experimental configuration, the user time (UT) for the classification, the solving time (ST) for the classification, and the user time (MT) for modeling muscles in seconds for the results reported in this paper. The third and fourth columns list the number of the sampling points (#P) used for the color-space classification, and the number of the modeled fiber bundles (#F).

<table>
<thead>
<tr>
<th>Data</th>
<th>size</th>
<th>#P</th>
<th>#F</th>
<th>UT (s)</th>
<th>ST (s)</th>
<th>MT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feet</td>
<td>256×128×256</td>
<td>5000</td>
<td>10</td>
<td>450</td>
<td>6.5</td>
<td>3030</td>
</tr>
<tr>
<td>Hand</td>
<td>256×256×128</td>
<td>3000</td>
<td>17</td>
<td>360</td>
<td>4.4</td>
<td>5410</td>
</tr>
<tr>
<td>Knee</td>
<td>256×128×256</td>
<td>2500</td>
<td>4</td>
<td>260</td>
<td>6.5</td>
<td>1240</td>
</tr>
<tr>
<td>Abdomen</td>
<td>256×128×256</td>
<td>2400</td>
<td>-</td>
<td>650</td>
<td>7.5</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: The time statistics in seconds for four datasets.

An unoptimized volume renderer was used to render the results at the image resolution of 800 × 800. The average rendering performance is 5 fps. With the volume mask that indicates the volume classification, varied shading or color can be applied to different objects, e.g., the tendon part of a fiber is rendered in white while the other parts are in red. Expressive muscle illustration requires a high resolution of the modeled volumetric fibers. In our approach, a fiber is modeled with a volumetric line whose width is seven voxels. Due to the limited volume resolution of the input volume data, the fiber number in a fiber bundle might be low. We upsample the volume datasets with higher resolutions, and increase the number of generated fibers in each muscle. In all examples listed in Table 1, the resolutions of the modeled muscle volumes are 5123.

The anatomy classification in our approach is enabled by a semi-supervised learning in the color space, while the observed colors are determined by the employed transfer function. In the meantime, a transfer function can be used to generate a volume classification. For each example, we have tested a variety of transfer functions, and found that the efficiency of our approach is stable. Figure 7 demonstrates that our approach can achieve better classifications than the standard transfer function design. Even if there are certain vague regions in the rendered image, our system still performs well. This robustness is facilitated by the sufficiently large sampling points in the volume space.

![Figure 7: Top row: the classification results with a conventional transfer function. The other pictures show six consecutive classification results with our approach. Each result is associated with a specific user indication in the image plane.](image)
illustration is shown. It is apparent that the modeled muscles greatly enhance the depiction and the expressiveness of the volume illustrations of both datasets, moving closer to the hand-drawn illustrations than conventional volume rendered images. The result of the Feet dataset is presented in Figure 10, where we compare a hand-drawn illustration and two volume illustrations with and without the modeled muscle. We conclude that our approach does improve the shape depiction by fusing additional anatomical information of some volumetric muscle.

Figure 11 compares the classification results with a standard transfer function and our approach. Our approach effectively separates all objects including the regions of a colon and two kidneys, while the conventional transfer functions can hardly fulfill this task.

7. Conclusions

We have described an easy-to-use system for modeling the muscle volume with an intelligent two-dimensional interface. One distinctive feature of our method is that the modeling is context-aware, and the classification of anatomical structures is performed in the color space. With our system, the muscle volume can be easily built by means of a group of convenient sketch-based volume modeling widgets, providing great convenience to the non-professional modeling users such as the medical illustrators. In addition to be a simple yet efficient muscle modeling tool, the presented work is useful in muscle rendering such as medical illustration and anatomy simulation.

In our current implementation, the medial axis of target muscle is estimated by the user and then coincided with the slicing plane. This task would require more efforts when the geometry of the target muscle is non-trivial. We plan to extract related information such as the bounding box and the orientation of fibers from the input volume data to facilitate locating the medial axis. In addition, since the surface geometry of the target muscle is formed by rotating the con-
tours, the types of muscle shapes are limited. An interesting future work would be to utilize more flexible muscle modeling toolkits like shape deformation, finite element models and kinetic animation modeling. We also plan to explore an interactive approach to generate real-world muscle volumes from measured diffusion tensor images (DTI) datasets, and integrate them into other volume data for expressive information fusion.

8. Acknowledgements

This work has been funded by Natural Science Foundations of China (No. 60873123), the US DOE DE-FG02-06ER25733 and NSF 0633150, the US NSF under Grant 0328984 and by the US Department of Homeland Security Regional Visualization and Analytics Center (RVAC) Center of Excellence.

References


Figure 11: Classification results for the Abdomen dataset with a well-tuned transfer function (left) and with our approach (right). The region (denoted by a red circle) of a colon and two kidneys can not be separated with the transfer function while ours can.