

3D-To-2D Mapping For User Interactive Segmentation Of Human Leg Muscles From MRI Data

Nilanjan Ray¹, Satarupa Mukherjee¹, Krishna Kanth Nakka¹, Scott T. Acton^{2,3}, Silvia S. Blemker³

¹Computing Science, University of Alberta, Canada

²Dept. of Electrical & Computer Engineering, University of Virginia, USA

³Dept. of Biomedical Engineering, University of Virginia, USA

Abstract – The automated computation of muscle volume from MRI of human legs is an open problem in the biomedical imaging community. Such automation has the potential to provide an objective measure of effectiveness of pre- and post-surgery treatments. In this paper, we take a step toward automation by proposing a framework for user interactive segmentation of MRI of human leg muscles. Our framework is built upon the strategy of bootstrapping: after the first few tedious segmentation results are achieved, and once a small database of segmented muscles is built, user interaction is reduced. Further, as the database of segmented muscles grows, the user interaction becomes more efficient. At the heart of this proposed framework is a simple, computationally attractive 3D representation of muscles. By a generalized cylinder model, we represent a 3D human leg muscle by two smooth 2D images, which enables application of 2D image processing and analysis methods in this complex multi-segment 3D problem. The smoothness of a leg muscle is modeled by the smoothness of the 2D images. Interdependence and relative positions of leg muscles are represented by a linear combination (basically, convolutions) of such 2D images. We demonstrate that fitting and editing of these models during user interactive segmentation of MRI data are computationally efficient, because our linear interaction model can be represented in the Fourier domain.

Index Terms— MRI, muscle segmentation, user interactive segmentation, 3D modeling.

I. INTRODUCTION

Skeletal muscle is the most abundant tissue in the human body and is essential for movement. Muscle volume is an important determinant of muscle functional capacity. In recent years, magnetic resonance imaging has become a ubiquitous tool to measure muscle volumes in vivo (e.g., Handsfield *et al.* [1], Hozlbaur *et al.* [2]). Muscles are outlined individually across several two-dimensional images, and three-dimensional volumes are reconstructed from the 2D outlines. Currently, segmentation of muscles from magnetic resonance imaging (MRI) requires a high level of user input; for example, segmentation of one lower limb requires approximately 20 hours of user segmentation time. This intense level of user input limits the ultimate clinical or broad applicability of these measurements. New algorithms

for segmentation time would have a major impact on the biomechanics and physical medicine communities.

In the recent past, automated segmentation of portions of individual human leg muscles has been reported as in [3] and [4]. However, segmentation of the entire leg muscle system still remains an open problem.

II. FRAMEWORK FOR INTERACTIVE SEGMENTATION

Within the current practice within MRI muscle analysis, the final goal of this study is the drastic reduction of user interaction time. Toward this goal, our proposed framework will rely on the user-generated sets of volume-segmented MRI (Handsfield *et al.* [1]). A database consisting of MRI data and corresponding muscle volume segmentation is depicted in Fig. 1, which shows the overall scheme of the proposed user interactive segmentation framework. When new MRI data are presented to the system, our method matches the new MRI with the MRIs in the database. If a satisfactory match (as evaluated by the user) is not found, the user will carry on the current practice of interactive segmentation.

On the other hand, if a satisfactory match is found, the volume segmentation corresponding to the matched MRI is brought in to the user. ‘Smart User Interaction’ as indicated in Fig. 1 then begins. So, instead of building the volume segmentation from scratch, the user practically makes corrections to already existing volume segmentation. This is the principal source of savings in interaction time. Note that as the database size grows, the entire framework will be able to present the user with accurate initial volume segmentation to start the interaction process. So, our proposed framework works on the principle of a bootstrapping method, which becomes more accurate and efficient as it matures with time.

Some of the essential elements of the smart user interaction module include: (a) movement of the muscle boundary in 2D on a slice, (b) movement of one muscle entirely in 2D or 3D, (c) movement of all muscles together in 2D or 3D, (d) delete muscles, one at a time, (e) deletion of the upper or lower part of a muscle, and (f) addition of a muscle *de novo*.

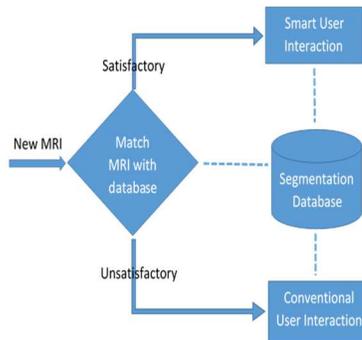


Figure 1. Schematic of user interactive segmentation.

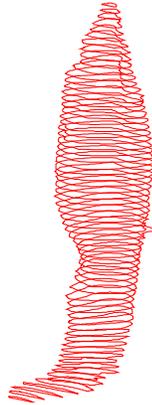


Figure 2. A human leg muscle represented by stacked closed contours, i.e., a generalized cylinder model.

III. MUSCLE SEGMENTATION TOOLBOX

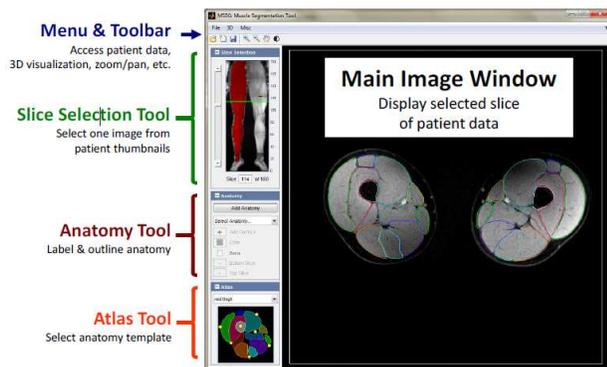


Figure 3. Snapshot of Muscle Segmentation Tool.

A snapshot of the muscle segmentation toolbox is shown in Fig. 3. Some of the important technical activities which can be carried out with the help of this toolbox are as follows –

- (i) Navigating 3D patient data
- (ii) Labeling and outlining anatomical structures
- (iii) Visualizing anatomical surfaces in 3D
- (iv) Exporting anatomical measurements
- (v) Exporting high resolution images and videos

The main parts of the toolbox are shown and labeled in Fig. 3.

The **slice selection tool** contains a maximum intensity projection of the 3D volume in the posterior/anterior direction. The main display panel contains a single cross section of the 3D volume in the left/right and posterior/anterior plane (convention displays each axial cross-section from the feet looking towards the head i.e., from the inferior to the superior direction). These images are oriented with the right hand side of the patient on the left hand of the screen. Thus this slice selection tool allows the user to explore the 3D volume slice by slice by clicking a location within the image display and adjusting the slider. The green line highlights the current slice displayed within the main image panel and the red regions contain some anatomical structures.

The **anatomy tool** allows users to manually delineate anatomical structures from the 3D volume. Each 3D anatomical structure is comprised of a set of 2D contours, each contour outlining an anatomical structure on a single cross section of the 3D volume.

The **atlas tool** maintains a library of plausible contours for a specific slice of 3D dataset.

IV. MATCHING MRI WITH DATABASE

In this section we will briefly discuss how a new MRI image is automatically matched with the database.

Once a new MRI image (stack of 2D slices) comes in, we compute histograms of oriented gradients (HOG) [7] feature vector for each slice for this query image. We also pre-compute HOG feature vectors for all the slices within each study in the database. The database at the moment has 65 fully segmented studies of legs from athletes, non-athletes, cerebral palsy patients and post-surgery patients with knee injuries. To compute HOG score between a query and a database image, for each slice in the query image, the best slice (defined by minimum L_1 distance) within the database image is found. Then, these L_1 distances are averaged over all the slices in the query image. This way each study in the database receives a HOG score. For a query image, the best match is then found by the minimum HOG score.

To validate this scheme, we consider an experiment, where each of the database studies take its turn as a query image and the corresponding database has 64 entries with query image deleted from the database. With this setup, along with the HOG scores, we are also able to compare user created segmentations for the studies. To compare two segmentations we utilize Jaccard scores (ratio of number of pixels belonging to the intersection over union of the two segmentations). Note that a higher Jaccard score implies a better match. In the left panel of Fig. 4, for a randomly chosen query image, we plot HOG scores versus Jaccard scores for all the database studies. Clearly, a negative trend in the scatter plot is observed, as expected. In the right panel of Fig. 4, we plot the minimum HOG scores and corresponding Jaccard

scores for all the query images. This right panel shows the spread of the Jaccard scores that we obtained by matching studies with the HOG feature. In the future, we plan to explore other features for MRI image retrieval.

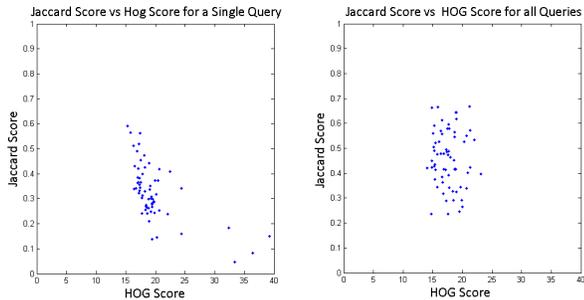


Figure 4. Plots of HOG Scores and Jaccard Scores.

V. 3D-TO-2D MAPPING

In order to attain the goal of user interactive segmentation efficiently, we propose to use generalized cylinders to model 3D human leg muscles. These cylinders are a series of closed contour is stacked in the z -direction as shown in Fig. 2. In this 3D-to-2D mapping, a muscle can be represented by two 2D images: $x(z, t)$ and $y(z, t)$, so that a point in 3D on the muscle boundary has the coordinates: $(x(z, t), y(z, t), z)$, where $t \in [0, 1]$ is the normalized distance that parameterizes the contour. This is shown in Fig. 5. The top left and right panels of Fig. 5 respectively show the images x and y by heat maps. Note that these images are not smooth, because when the contours were drawn by a user on the MRI slices, the starting points for contours were not necessarily close to each other for nearby slices. Moreover, on some slices, contours were drawn in the clockwise manner, while on other slices counter clockwise convention was followed.

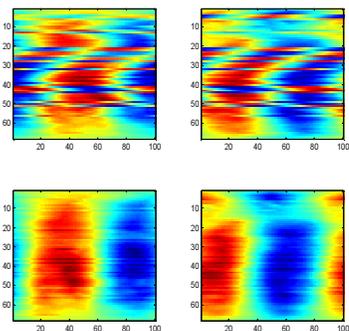


Figure 5. The top left and right panels show $x(z, t)$ and $y(z, t)$ respectively. Bottom left and right panels show reparameterized versions of $x(z, t)$ and $y(z, t)$, respectively.

We can choose the starting point of a contour on each slice and fix an order, either clockwise or counter clockwise by enforcing a smoothness constraint. We refer to this smoothing as reparameterization operation. This can be achieved with the following minimization:

$$\min_{s_z} \left[\sum_{z,t} |x(z, t + s_z) - x(z - 1, t + s_z)| + |y(z, t + s_z) - y(z - 1, t + s_z)| \right].$$

The variable s_z represents re-parameterizations of t , i.e., s_z is the amount of circular shift of the z^{th} row of x and y matrices. The minimization can be performed by dynamic programming, but, in practice, it actually reduces to a fast sequential row-by-row greedy optimization. The bottom panels of Fig. 5 show the reparameterized x and y images that appear smooth in the heat map. Notice that reparameterization has no effect on the shape of the 3D muscle.

VI. INTRA-MUSCLE MODEL

When a user moves a contour point on a muscle slice, the entire 3D stack of contours should move in a meaningful way. If the user moves a point on a slice, the neighboring points on the contour and neighboring contours should move and fit to the data, e.g., to the edges representing muscle boundaries. Note that moving a contour point is same as pushing down and/or pulling up a point on the 2D image surfaces. We model this interaction among pixels values with a locally linear model:

$$x(z, t) = \sum_{i=z-1, i \neq z}^{z+1} \sum_{j=t-1, j \neq t}^{t+1} a_{i,j} x(i, j),$$

$$y(z, t) = \sum_{i=z-1, i \neq z}^{z+1} \sum_{j=t-1, j \neq t}^{t+1} a_{i,j} y(i, j),$$

such that the affine constraint is satisfied:

$$\sum_{i=z-1, i \neq z}^{z+1} \sum_{j=t-1, j \neq t}^{t+1} a_{i,j} x(i, j) = 1.$$

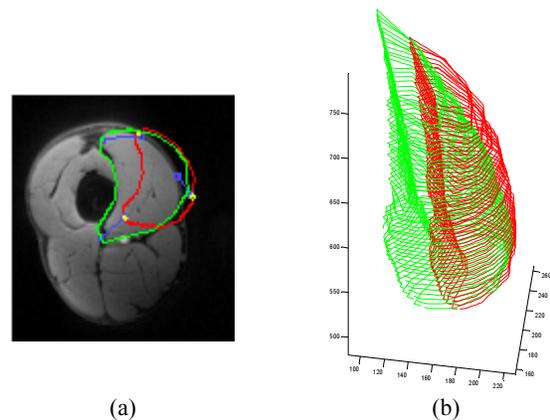


Figure 6. (a) example user interaction – moving the red contour to the green contour shown by moving three points, (b) 3D muscle shape before (red) and after (green) user interaction.

A locally linear model has been used in the past in many significant applications (e.g., non-linear dimensionality reduction [5]). It is easy to verify that this model is invariant to affine transformations. The linear coefficients $a_{i,j}$ can be solved efficiently by a minimum norm least squares method, because the linear system is underdetermined here.

Once we find out the coefficients $a_{i,j}$, we assume that they remain unchanged during user interaction. For example, when a user moves three points on the red contour the green contour is generated. This is shown in Fig. 6(a). Fig. 6(b) shows the positions of 3D muscle before (in red) and after (in green) the user interaction. Notice that the linear intra-muscle model has been able to preserve its smooth shape after the interaction.

Computationally, such interaction is enabled in real time, because, here, the software solves a sparse linear system with a few boundary conditions, much like the manner in which one solves the Laplace equation on a rectangular 2D grid subject to Dirichlet boundary conditions [6].

VII. MODELING INTERACTION AMONG MUSCLES

In practice, modeling the dependency within each muscle is not enough, because the leg muscles co-exist with each other in an orderly way in 3D. To model the interdependencies, let us denote the leg muscles by $(x_1, y_1), \dots, (x_n, y_n)$. Once again these are 2D image pairs. To capture the interrelations among these images, we consider the following linear convolution model:

$$\begin{aligned} x_i &= w_{i1} * x_{i1} + w_{i2} * x_{i2} + c_i^x, & i &= 1, 2, \dots, n, \\ y_i &= w_{i1} * y_{i1} + w_{i2} * y_{i2} + c_i^y, & i &= 1, 2, \dots, n, \end{aligned}$$

where,

$$\begin{aligned} x_{i1} &= \sum_{j=1, j \neq i}^{n/2} x_j, & y_{i1} &= \sum_{j=1, j \neq i}^{n/2} y_j, \\ x_{i2} &= \sum_{j=1+n/2, j \neq i}^n x_j, & y_{i2} &= \sum_{j=1+n/2, j \neq i}^n y_j. \end{aligned}$$

The convolution kernels (w) are linear coefficients for the model. Thus, each muscle is expressed as a linear combination of the rest of the muscles. The constant terms, c_i^x and c_i^y can be chosen in such a way that the linear model becomes affine transformation invariant up to a constant. Computing the convolution kernels is straightforward in the frequency domain. Two of the salient advantages of the above model discussed below.

A. Affine transformation invariance

It can be shown that for the following choices of c_i^x and c_i^y the model becomes invariant to affine transformation:

$$\begin{aligned} c_i^x &= \text{mean}(x_i) - 0.5\text{mean}(x_{i1}) - 0.5\text{mean}(x_{i2}), \\ c_i^y &= \text{mean}(y_i) - 0.5\text{mean}(y_{i1}) - 0.5\text{mean}(y_{i2}). \end{aligned}$$

In fact, for such choices, the convolution kernels yield a constant image of grayscale 0.5 when operated on a constant image of grayscale 1.

B. Computational efficiency during user interaction

As mentioned before, one of the characteristics of the convolution model is lack of computational burden during

user interaction. Suppose, dx_i^0 and dy_i^0 are deformations of the i th muscles produced by any ‘‘external force’’ such as the user, then we need to make sure that the actual deformations dx_i and dy_i of the muscles follow the linear model. So, we need to solve the following optimization problem:

$$\min \sum_{i=1}^n \|dx_i^0 - dx_i\|^2 + \|dy_i^0 - dy_i\|^2,$$

such that

$$\begin{aligned} dx_i &= w_{i1} * dx_{i1} + w_{i2} * dx_{i2}, & i &= 1, 2, \dots, n, \\ dy_i &= w_{i1} * dy_{i1} + w_{i2} * dy_{i2}, & i &= 1, 2, \dots, n. \end{aligned}$$

Even though the minimization is convex, computationally it is daunting because the linear system is quite large and dense. Instead, we solve the optimization in the Fourier domain:

$$\min \sum_{i=1}^n \|Fdx_i^0 - Fdx_i\|^2 + \|Fdy_i^0 - Fdy_i\|^2,$$

such that

$$\begin{aligned} Fdx_i &= Fw_{i1}Fdx_{i1} + Fw_{i2}Fdx_{i2}, & i &= 1, 2, \dots, n, \\ Fdy_i &= Fw_{i1}Fdy_{i1} + Fw_{i2}Fdy_{i2}, & i &= 1, 2, \dots, n. \end{aligned}$$

Notice that the objective function remains the same because of the orthonormality property of DFT. Note also that now the optimization problems are much smaller, decoupled, and still convex. In fact, the aforementioned minimization problem can be solved point-wise in the DFT domain independently of others. Each of these independent minimization problems involves only n variables, where n is the total number leg muscles.

VIII. CONCLUSIONS

An important first step has been achieved in solving the open problem of multi-muscle segmentation from 3D MRI data. The method contributes two major strides: (1) a 3D-to-2D mapping that allows a single 3D muscle to modeled by two 2D images representing shape in x and y coordinates; and (2) the employment of previous manual segmentations to guide interactive semi-automated segmentation of new MRI data. This approach has benefits of low computational cost and of invariance to standard transformations.

ACKNOWLEDGEMENTS

The authors would like to thank Katherine Read, Geoffrey Handsfield and Craig Meyer for imaging and preparing MRI data and for manual segmentation. The authors acknowledge the funding support from UVA-Coulter Translational Partnership.

REFERENCES

- [1] GG Handsfield, CH Meyer, J Hart, MF Able, and SS Blemker. Size and Scaling of 35 Lower Limb Muscles Quantified Using MRI, *Journal of Biomechanics*, in revision, 2013.
- [2] Holzbaur, K. R., Murray, W. M., Gold, G. E., Delp, S. L., 2007b. Upper limb muscle volumes in adult subjects. *Journal of Biomechanics* 40, 742-749.
- [3] S. Essafi, G. Lings, J. -F. Deux, A. Rahmouni, G. Bassez, and N. Paragios, "Wavelet-Driven Knowledge-Based MRI Calf Muscle Segmentation," *IEEE ISBI 2009*, page 225-228.
- [4] S. Andrews, G. Hamarneh, A. Yazdanpanah, B. HajGhanbari, W.D. Reid, "Probabilistic multi-shape segmentation of knee extensor and flexor muscles." *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*. Springer Berlin Heidelberg, 2011. 651-658.
- [5] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science* 290 (5500): 2319-2323, 22 December 2000
- [6] W.F. Ames, *Numerical Methods for Partial Differential Equations*, Third Edition, Academic Press, 1992.
- [7] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", In *CVPR*. 886–893, 2005.