

Robust Muscle Cell Quantification Using Structured Edge Detection and Hierarchical Segmentation

Fujun Liu¹, Fuyong Xing¹, Zizhao Zhang², Mason Mcgough³, and Lin Yang^{1,3}

¹Department of Electrical & Computer Engineering, ²Department of Computer & Information Science & Engineering, ³J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, USA

Abstract. Morphological characteristics of muscle cells, such as cross-sectional areas (CSAs), are critical factors to determine the muscle health. Automatic muscle cell segmentation is often the first prerequisite for quantitative analysis. In this paper, we have proposed a novel muscle cell segmentation algorithm that contains two steps: 1) A structured edge detection algorithm that can capture the inherent edge patterns of muscle images, and 2) a hierarchical segmentation algorithm. A set of nested partitions are first constructed, and a best subset selection algorithm is then designed to choose a maximum weighted subset of non-overlapping partitions as the final segmentation result. We have experimentally demonstrated that the proposed structured edge detection based hierarchical segmentation algorithm outperforms other state of the arts for muscle image segmentation.

1 Introduction

The ability to accurately and efficiently quantify the morphological characteristics of muscle cells, such as cross-sectional areas (CSAs), is essential to determine the health conditions of muscles. Automatic muscle cell segmentation plays a significant role in calculating the morphological parameters. There are two major challenges for muscle cell segmentation: 1) Because muscle cells always touch with one another, it is critical to design an efficient and effective muscle cell edge detection algorithm, and 2) because the strength of the muscle cell boundary varies significantly, segmentation algorithms using a fixed parameter often fail to produce satisfactory results for all the regions in one digitized muscle specimens.

In this paper, we propose a structured edge detection based hierarchical segmentation algorithm for muscle image segmentation. The structured edge detection, which can better capture inherent muscle image edge structures, is achieved by extending a random decision forest framework. It is noted in [1,2] that, by storing structure information at the leaf nodes of the random decision tree other than class probabilities, random decision forest can be conveniently used for structured learning. Different from the traditional edge detection method [3],

edge masks instead of edge probability values will be stored at the leaf nodes in our proposed structured edge detection algorithm. In order to accurately segment each muscle cell exhibiting both strong and weak boundaries, a hierarchical segmentation method is proposed, which takes a set of partitions produced by applying a segmentation algorithm with varying parameters as inputs, and selects a best subset of non-overlapping partition regions as the final result.

An overview of our proposed muscle image segmentation algorithm is shown in Fig. 1. Given an image patch, 1) an edge map is generated by the proposed structured edge detection algorithm; 2) an Ultrametric Contour Map (UCM) [3] is constructed and a set of segmentation candidates is generated by adjusting the thresholds of UCM; and 3) an efficient dynamic programming based subset selection algorithm is then used to choose the best regions for muscle image segmentation based on a constructed tree graph.

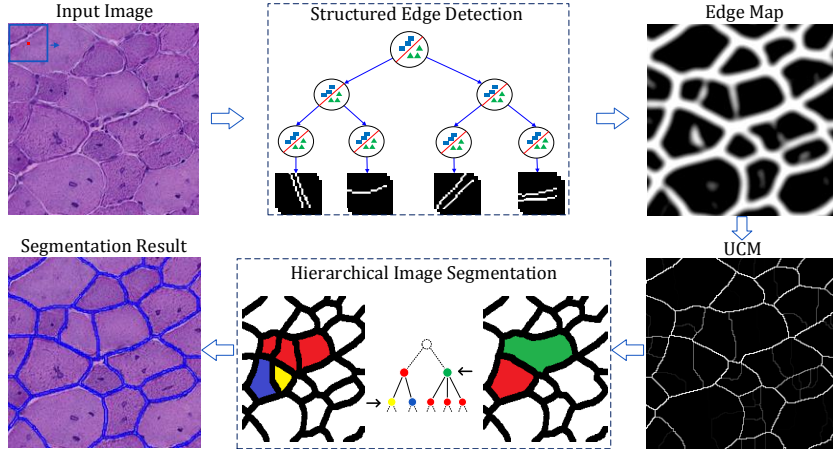


Fig. 1. An overview of the proposed muscle image segmentation algorithm.

2 Structured Edge Detection

Random Decision Forest. We start with a brief review of the random decision forest. A decision forest $\mathcal{F} = \{T_t\}$ is an ensemble of decision trees T_t , which are trained independently on randomly selected samples $S = \{s_i = (x_i \in \mathcal{X}, y_i \in \mathcal{Y})\}$, where \mathcal{X} and \mathcal{Y} denote the input features and output labels, respectively. A decision tree $T_t(x)$ produces the prediction results by recursively branching a feature sample $x \in \mathcal{X}$ left or right down the tree until a leaf node is reached. For a decision forest, the predictions $T_t(x)$ from individual trees are combined together using an ensemble model. Majority voting and averaging are common ensemble choices for classification and regression problems, respectively.

During the training of a decision tree, at each node n , a split function $h(x, \theta_n)$ is chosen to split the samples S_n into left S_n^L or right S_n^R . For multiclass classification, the split function $h(x, \theta_n)$ is optimized by maximizing the information

gain

$$\mathcal{I}(S_n) = H(S_n) - \left(\frac{|S_n^L|}{|S_n|} H(S_n^L) + \frac{|S_n^R|}{|S_n|} H(S_n^R) \right), \quad (1)$$

where $H(\cdot)$ is the class entropy function. The split function $h(x, \theta_n)$ can be an arbitrary classifier. A common choice is a stump function that is found to be computationally efficient and effective in practice [2]. The training procedure continues to split the samples until either a maximum depth is reached, or too few samples are left, or information gain falls below a certain threshold.

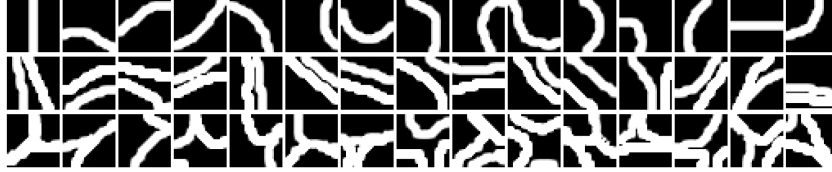


Fig. 2. Some sample edge masks learned and stored at the leaf nodes of the random decision trees.

Structured Edge Detection. Since a decision tree classifier generates the actual prediction at the leaf nodes, more information can be stored at the leaf nodes than class likelihoods. For example, in [1], structured class label information is stored at leaf nodes for semantic image segmentation. Inspired by [2,4], we propose to store edge structure information at the leaf nodes for structured muscle image edge detection. Different from traditional edge detection algorithms [3], which take an image patch x as an input and compute the probability of the edge existence at the center pixel p , the output of our proposed structured edge detection algorithm is an edge mask around the central pixel p instead of the likelihood value. As shown in Fig. 1, after learning the decision tree, the median of the edge masks sent to the leaf node will be stored as the leaf node output.

The information gain criteria in Equation (1) is effective in practice for decision tree training. In order to follow this criteria, the edge masks must be explicitly assigned proper class labels at each internal node of the tree during the training stage. One straightforward idea is to group the edge masks at a node into several clusters by an unsupervised clustering algorithm such as k-means or mean-shift, and then treat each cluster *id* as the class label for the sample belonging to that cluster. However, the edge masks \mathcal{Y} do not reside in the Euclidean space and directly grouping them may not generate desired results. In addition, clustering in high dimension space ($y \in \mathbb{R}^{256 \times 1}$ for an 16×16 edge mask) is computationally expensive. In this paper, we proposed to reduce the high dimension edge masks $\mathcal{Y} \in \mathbb{R}^n$ to a lower dimensional subspace $\mathcal{Z} \in \mathbb{R}^m$ ($m \ll n$) using an autoencoder [5] before clustering the edge masks. In this paper, we use the matrix form and vector form of edge mask space \mathcal{Y} interchangeably.

Autoencoder is one of the state-of-the-art unsupervised nonlinear dimension reduction methods. The training of an autoencoder consists of both the pretraining step, which consists of learning a stack of restricted Boltzmann machines

(RBMs) layer by layer, and the fine-tuning step that seeks to minimize the discrepancy between the original data and its reconstruction. For an autoencoder with L layers,

$$y^l = \sigma(W^l y^{l-1} + b^l), \quad (2)$$

where $\sigma(t) = 1/(1 + e^{-t})$, $y^0 = y \in \mathcal{Y}$, $y^L = z \in \mathcal{Z}$, and W^l , b^l , $l = [1 \cdots L]$ are parameters to be learned in training the autoencoder.

Please note that although the transformed data z is used to choose split function $h(x, \theta_n)$ in training the decision tree, only the original edge mask y is stored at leaf nodes for the prediction. Some sample edge masks learned and stored at the leaf nodes are shown in Fig. 2. As one can tell that many edge structures are unique for muscle cell boundaries, which demonstrate the effectiveness of the structured edge detection procedure.

Our proposed structured edge detection algorithm takes a 32×32 image patch as input and generates a 16×16 edge mask around the input's center pixel. The image patch is represented with the same high-dimensional feature used in [2,6], which is effective and computationally efficient. In total, two million samples are randomly generated to train the structured decision random forest, which consists of 8 decision trees. The autoencoder model used in our work consists of an encoder with layers of sizes $(16 \times 16) - 512 - 256 - 30$ and a symmetric decoder. The autoencoder model is trained once offline and applied to all decision trees, and the data compression is only performed at the root node.

3 Hierarchical Image Segmentation

Hierarchical strategy has been successfully applied to image segmentation recently [7,8]. In general, the hierarchical image segmentation consists of two steps: 1) A collection of segmentation candidates is generated by running some existing segmentation algorithms with different parameters. Usually, an undirected graph is constructed from these partition candidates where an edge exists between two overlapping regions; 2) Based on some domain-specific criteria, a maximal weighted independent set (MWIS) of the constructed graph is selected as the final segmentation results. For example, Felzenszwalb's method [9] with multiple levels is used to generate the segmentation candidate pool. An optimal purity cover algorithm is used in [7] to select the most representative regions. In [8], the watershed segmentation method with different thresholds gives a collection of partitions and a conditional random field (CRF) based learning algorithm is utilized to find the best ensembles to present the final segmentation results.

However, the MWIS problem on a general graph is NP-hard and usually difficult to optimize. In this work, an Ultrametric Contour Map (UCM) [3] is first constructed on the edge map generated by the proposed structured edge detection algorithm presented in the previous section, then the pool of segmentation candidates is produced by setting different thresholds of UCM. Because of the nice property of UCM where the segmentation results using different thresholds are nested into one another, we can construct a tree graph for this pool of seg-

mentation candidates. The final step is to solve this tree graph based MWIS problem using dynamic programming.

Given a set of segmentation candidates generated with different thresholds using UCM, an undirected and weighted tree graph, $G = (V, E, w)$, is constructed, where $V = \{v_i, i = 1, 2, \dots, n\}$ represents the nodes with each v_i corresponding to a segmented region R_i . E denotes the edges of the graph. The $w(v_i)$ is learned via a general random decision forest classifier to represent the likelihood of R_i as a real muscle cell. An adjacent matrix $A = \{a_{ij} | i, j = 1, \dots, n\}$ is then built with $a_{ij} = 1$ if $R_i \subset R_j$ or $R_j \subset R_i$, and otherwise 0. Denote $\mathbf{x} \in \{0, 1\}^n$ the indicator vector, where its element is equal to 1 if the corresponding node is selected, otherwise 0. Finally, the constrained subset selection problem is formulated as

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} w^T \mathbf{x}, \text{ s. t. } x_i + x_j \leq 1, \text{ if } a_{ij} = 1. \quad (3)$$

Considering the special tree graph structure, (3) can be efficiently solved via the dynamic programming approach with a bottom-to-up strategy.

Muscle Cell Likelihood Generation. In order to ensure that (3) selects the desired regions, each candidate region (node) must be assigned an appropriate muscle cell likelihood score w . In our work, each candidate region is discriminatively represented with a feature vector that consists of a descriptor to model the convexity of the shape, and two histograms to describe the gradient magnitudes of the pixels on the cell boundary and inside the cell region. These morphological features are proposed based on the following observations: 1) The shape of a muscle cell is nearly convex; 2) The cell boundaries often exhibit higher gradient magnitudes; 3) The intensities within the cell regions should be relatively homogeneous.

4 Experiments

The proposed algorithm is tested using 120 H&E-stained muscle cell images captured at $10\times$ magnification. Each image contains around 200 muscle cells. The images are randomly split into two sets with equal size. The ground truth of each individual muscle cell is manually annotated. To quantitatively analyze the pixel-wise segmentation accuracy, we calculate the precision $P = \frac{|S \cap G|}{|S|}$, recall $R = \frac{|S \cap G|}{|G|}$, and F_1 -score $F_1 = \frac{2 * P * R}{P + R}$, where S denotes the segmentation result and G is the ground truth.

Both the proposed structured edge detection algorithm and the hierarchical image segmentation method are evaluated and compared with three state-of-the-art methods: 1) Isoperimetric graph partition (ISO) [10] that produces high quality segmentations as a spectral method with improved speed and stability, 2) global probability of boundary detector (gPb) [3] that is widely used in natural image segmentation, and 3) a deep convolutional neural network (DCNN) method that is widely adopted as recent state-of-the-art for detection and segmentation. The trained DCNN model consists of six learned layers including

three convolutional layers with filter sizes (4×4 , 3×3 , and 2×2), and three fully connected layers with neuron numbers (500, 250, and 2). Each convolutional layer has 20 output maps and is followed by a max pooling layer with pooling size 2×2 . Rectifier activation $f(x) = \max(0, x)$ is applied to all parameter layers except the last fully connected layer (output layer) where a 2-way softmax nonlinear activation is used. In total, two million image patches with size 51×51 , half positive and half negative, are randomly generated from 60 images to train the DCNN model. The training of DCNN is carried out using open source tool Caffe [11].

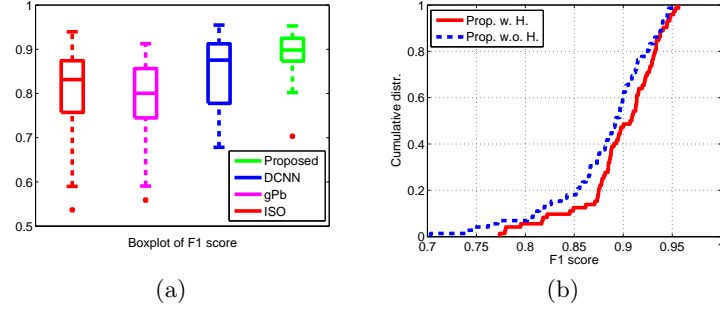


Fig. 3. The comparative segmentation results of the proposed muscle image segmentation algorithm.

Experimental Results for Structured Edge Detection. For fair comparison, UCM is first constructed for edge detection based methods including gPb [3], DCNN, and the proposed structured edge detection algorithm. The segmentation is done by changing thresholds of UCM. Different segmentation results are generated for each algorithm. Only the results with highest F_1 -score are reported. Similar to ISO, multiple segmentations are produced by varying the isoperimetric parameter and the best result is reported. The comparative boxplots of F_1 -scores are shown in Fig. 3 (a). We can see that our structured edge detection based segmentation algorithm outperforms other methods [10], [3] and DCNN for muscle specimens. The quantitative comparative results are shown in Table 1, where we report the average and standard variance of F_1 -score, precision, and recall.

It is worth to note that the proposed structured edge detection algorithm indeed performs better than DCNN for the muscle image dataset. One reason is that we do not have sufficient training data to learn a DCNN model that is sufficient to capture all the edge variations. A larger DCNN model with more training data might be able to achieve better performance. With respect to speed, the current DCNN model takes 28 seconds to generate an edge map for segmentation on an image with size 1024×768 , which is much slower than our structured edge detection (less than 1 second).

Experimental Results for Hierarchical Segmentation. In this part, five different segmentations are produced by setting thresholds 0.9, 0.8, 0.6, 0.4, and

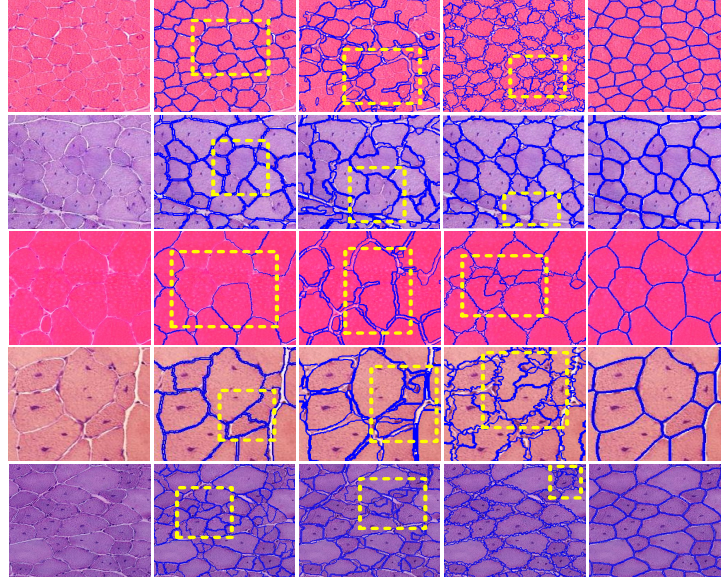


Fig. 4. Some qualitative segmentation results. From left to right, the columns denote the original image patch, the segmentation results using ISO [10], gPb [3], DCNN, and the proposed method. The yellow rectangles highlight some regions with segmentation errors.

Table 1. The pixel-wise segmentation accuracy

Method	F ₁ -score		Prec.		Rec.	
	mean	std	mean	std	mean	std
ISO[10]	0.8050	0.0993	0.8988	0.0589	0.7429	0.1369
gPb[3]	0.7904	0.0780	0.9123	0.0515	0.7011	0.0962
DCNN	0.8388	0.1073	0.9464	0.0411	0.7666	0.1441
Prop. w.o. H.	0.8815	0.0523	0.8861	0.0551	0.8783	0.0587
Prop. w. H.	0.8974	0.0422	0.9078	0.0433	0.8888	0.0543

0.2 of UCM constructed from the proposed edge detection algorithm. The segmentation with threshold 0.6 achieves highest F_1 -score. The hierarchical segmentation is achieved by solving Equation (3). In this text, we use Prop. w.o. H. to denote the proposed structured detection based segmentation algorithm without using hierarchical segmentation, and Prop. w. H. to denote the hierarchical segmentation results. The comparative results are shown in Fig. 3 (b), where the cumulative distributions $F(x) = Pr(t \leq x)$ with respect to F_1 -scores are plotted. By comparing their cumulative distributions in Fig. 3, we can see that the proposed hierarchical segmentation consistently achieves better results than single segmentation with even fine-tuned parameter. In addition, as shown in Table 1, the hierarchical segmentation also gives the lowest variance with respect to F_1 score, which indicates strong robustness. The proposed hierarchical

segmentation method is also more practical since it is not always possible or convenient to fine-tune the parameter at runtime for new testing images.

Based on our experimental results, we conclude that: 1) The proposed structured edge detection algorithm outperforms the state-of-the-art edge detection methods including gPb[3] and DCNN for muscle images. 2) The proposed hierarchical segmentation method consistently boosts the overall segmentation accuracy and also provides strong robustness. For better illustration, several qualitative segmentation results are presented in Fig. 4.

5 Conclusion

In this paper, we have proposed a structured edge detection based hierarchical segmentation algorithm for muscle image segmentation. Compared with traditional edge detection methods, the proposed structured edge detection algorithm can better capture the inherent muscle image edge structures such that it can accurately and efficiently detect the muscle cell boundaries. We have experimentally demonstrated that the proposed hierarchical segmentation method exceeds the performance of many state of the art segmentation methods even with fine-tuned parameters.

References

1. Kotschieder, P., Buló, S.R., Bischof, H., Pelillo, M.: Structured class-labels in random forests for semantic image labelling. In: ICCV, 2011, IEEE (2011) 2190–2197
2. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV, 2013, IEEE (2013) 1841–1848
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *PAMI* **33**(5) (2011) 898–916
4. Chen, Y.T., Yang, J., Yang, M.H.: Extracting image regions by structured edge prediction. In: WACV, 2015, IEEE (2015) 1060–1067
5. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786) (2006) 504–507
6. Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR, 2014, IEEE (2014) 328–335
7. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. *PAMI* **35**(8) (2013) 1915–1929
8. Uzunbaş, M.G., Chen, C., Metaxas, D.: Optree: A learning-based adaptive watershed algorithm for neuron segmentation. In: MICCAI 2014. Springer (2014) 97–105
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *IJCV* **59**(2) (2004) 167–181
10. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. *PAMI* **28**(3) (2006) 469–475
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, ACM (2014) 675–678