INTERACTIVE EXPLORATION OF MICROSTRUCTURAL FEATURES IN GIGAPIXEL MICROSCOPY IMAGES

Hsueh-Chien Cheng *Antonio Cardone* *Amitabh Varshney*

Department of Computer Science and UMIACS, University of Maryland
College Park, Maryland, USA

ABSTRACT

Modern imaging technologies enable the study of microstructural features, which require capturing the finest details in high-resolution gigapixel images. Nevertheless, the resolution disparity between gigapixel images and megapixel displays presents a challenge to effective visual analysis because subtle texture differences are hardly perceivable at coarser resolutions. In this paper, we present a hierarchical segmentation technique based on the joint distribution of intensity and noise-resistant local binary patterns to differentiate subtle microstructural textures across various scales. The coarse-to-fine segmentation procedure subdivides each parent segment into texturally-distinct child segments at progressively higher resolutions. The hierarchical structure of segments allows creating intermediate segmentation results interactively. Based on the intermediate results, we highlight regions with texture differences using distinct colors, which provide salient visual hints to users despite the current viewing resolution. Our new technique has been validated on large microscopy images and shows promising results.

Index Terms— Image segmentation, gigapixel images

1. INTRODUCTION

As technology pushes the limits of image resolution, the widening gap between image and screen resolutions has become an impediment to visual analysis of high-resolution images. For example, visually inspecting a 281-gigapixel microscopy image [1] on an ordinary four-megapixel screen would take tremendous effort, especially in an exploratory analysis when viewers have yet to decide which specific targets to focus on before exploring the image. The analysis of gigapixel images therefore requires human-computer collaboration, especially in non-trivial analyses that are laborious even for moderate-sized images.

Many examples of such demanding analyses involve texture detection, which is challenging because of the complex nature of textures. Besides the complexity in modeling textures computationally [2], visually inspecting textures in large images is also difficult because textures can easily become indistinguishable after the high-frequency components are removed by downsampling. Analysts are therefore faced with two conflicting operations: 1) zooming in to specific regions of the image for the high-resolution details required in texture analysis, and 2) zooming out for a global view that puts the target microstructures in a better context [3, 4].

Although the conflicts caused by resolution disparity aggravate image understanding, additional visual cues such as color [3] and depth [5] can enhance human perception of images. In the past, specialized lighting functions [6] and clustering techniques [7] have been used to assign colors and opacities that highlight brain microstructures in volumetric data. Studies in visual psychophysics have also revealed that human eyes are naturally more efficient in processing colors than textures [8]. Therefore, in this work, we use colors to highlight texture differences and facilitate the search for distinct textures and microstructures in large images. Whereas the resolution disparity problem is inevitable when exploring large images, texture differences that are less apparent at a given resolution stand out more easily because the color-based highlights are more salient to human eyes. We address the following challenges. First, conducting texture analysis on gigapixel images is extremely computationally expensive. Second, the textures of interest are not discernible at all scales. The need for finding and visualizing texture differences across scales in large gigapixel images calls for an efficient segmentation procedure and a flexible interaction component to enable manual exploration of gigapixel images.

We use the joint distribution of intensity and noise-resistant local binary pattern (NRLBP) [9] to characterize textures in the image (Section 2.1). Based on the joint intensity-NRLBP distribution, we apply an efficient hierarchical segmentation algorithm to identify regions with different textures at various scales (Section 2.2). Segments are colored differently to allow visual detection even when zooming out. Based on the color-based highlights, users can decide whether they need to split a segment further (Section 2.3). Such a user-mediated approach may significantly reduce the time needed to locate regions with texture differences, reveal ones that could otherwise be overlooked, and lead to a more thorough exploration.

2. METHODOLOGY

The computation and memory required to segment an image grow significantly with the number of pixels. For example, in our preliminary experiments, we found that F&H [10] and FSEG [11] terminated prematurely when segmenting gigapixel images because of insufficient memory when allocating memory space for an excessive number of vertices and edges (F&H) or a large high-dimensional feature vector for all pixels (FSEG). Because segmenting a gigapixel image based on pixel-level representations is infeasible, in this paper we reduce the memory usage by partitioning a segment at progressively higher resolutions and modeling the image as a graph of superpixels.

Similar to previous research on gigapixel images [3, 12, 13], we use a Gaussian pyramid as the multiscale image representation. A tile-based storage divides each pyramid level into a set of rectangular tiles of size 1024×1024, allowing efficient loading of relevant regions in the image pyramid at various scales. In addition to browsing gigapixel images, our tool allows exploring the segmentation results (the color highlights) hierarchically and interactively.

2.1. Joint intensity-NRLBP histogram

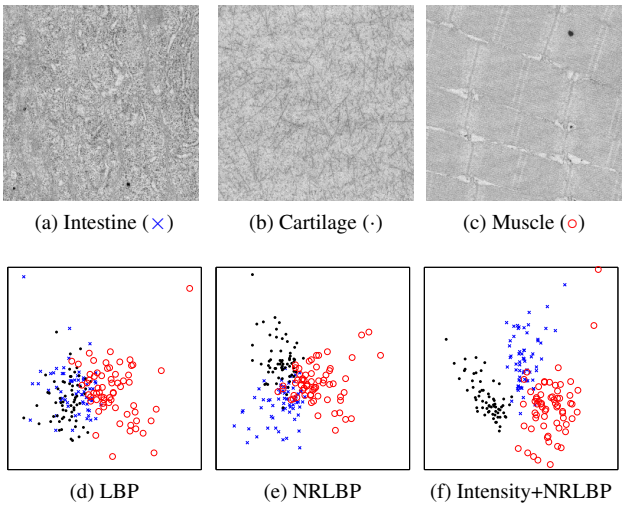
We use the joint histogram of intensity and noise-resistant local binary pattern (NRLBP) [9], which is a variant of local binary pattern (LBP) [14, 15], to identify texture differences. An LBP is a binary string in which each bit encodes the sign of the difference between the intensity value of the center pixel and the intensity value of a neighboring pixel. Nevertheless, this binary encoding is prone to noise that causes unstable sign of intensity differences. Local ternary pattern (LTP) [16] alleviates the sensitivity to noise by a ternary encoding that uses the third state to represent situations when the intensity difference is less than a predefined threshold. NRLBP [9] treats the third state of LTP as a wild card that can map to either zero or one. Each ternary encoding of LTP therefore corresponds to a specific set of conventional uniform LBPs [15] after mapping the wild cards to zero or one.

The LBP-based features describe textures based on intensity differences instead of actual intensity values. Nevertheless, intensity values are important attributes in many applications, including the analysis of microscopy images in this work because tissues of different types may have distinct intensity values. By combining intensity with NRLBP, we use the joint distribution of intensity and NRLBP to improve the descriptive power. The joint histogram divides the intensity range $[0, 255]$ into eight non-overlapping bins and uses a separate bin for each pattern of NRLBP. We use the chi-square distance $\chi^2(x, y)$ to compare two normalized intensity-NRLBP joint histograms x and y; $\chi^2(x, y)$ is defined as:

$$\chi^2(x, y) = \sum_{i,j} \frac{(x_{i,j} - y_{i,j})^2}{x_{i,j} + y_{i,j}}, \qquad (1)$$

where $x_{i,j}$ and $y_{i,j}$ are the values of bin (i, j) in x and y, respectively.

Fig. 1 shows the power of the intensity-NRLBP joint histograms by visualizing the result of multidimensional scaling (MDS) [17] with the intestine (Fig. 1a), cartilage (Fig. 1b), and muscle (Fig. 1c) tissue samples. Each sample of size 256×256 is divided into 8×8 blocks (each of size 32×32), which are projected to a two-dimensional space based on the pairwise chi-square distances computed using LBP (Fig. 1d), NRLBP (Fig. 1e), and the joint distribution of intensity and NRLBP (Fig. 1f). By modeling noise explicitly, NRLBP separates the blocks of different tissue types better than LBP. Using the joint distribution of intensity and NRLBP further separates the blocks into three distinguishable clusters, one for each tissue type.



(a) Intestine (×) (b) Cartilage (·) (c) Muscle (○)

(d) LBP (e) NRLBP (f) Intensity+NRLBP

Fig. 1. The two-dimensional projections (d–f) show the distribution of sample blocks of (a) intestine, (b) cartilage, and (c) muscle tissues. The projections are obtained by MDS analysis based on the pairwise distances computed using (d) LBP, (e) NRLBP, and (f) joint distribution of intensity and NRLBP.

2.2. Unsupervised graph-based segmentation

For large images, working directly at the pixel level is impractical because of the computational burden. Instead, we perform the actual texture segmentation at the superpixel level. Because the number of superpixels is two orders of magnitude smaller than the number of pixels, the computational burden becomes less of an issue.

We use the simple linear iterative clustering (SLIC) [18] to efficiently segment each image tile into superpixels of size about 500 pixels. The SLIC generates regular-shaped superpixels and therefore avoids the bias introduced by the irregular distribution of nodes in a graph, which may diminish the quality of graph-based segmentation techniques [19]. We set the compactness to ten to generate results with a balanced shape

regularity and boundary accuracy. The superpixel extraction is repeated for each pyramid level independently.

Given the superpixels, we first construct a superpixel adjacency graph G and then partition G into disjoint connected components. Each connected component corresponds to a segment that differs from its neighboring segments in texture. An edge in G is assigned a weight representing the texture dissimilarity (Eqn. 1) between the corresponding pair of superpixels. Because a tile-based image representation is used here, constructing G requires iterating over all the tiles, connecting vertices within the same tile as well as vertices that lie on the boundary of adjacent tiles.

After constructing G, we use a modified version of the efficient graph-based segmentation (F&H) [10], originally formulated to work at the pixel level, to partition G. Starting from having each vertex being its own connected component, the algorithm examines all edges in a non-decreasing order with respect to edge weights. Given that the two end nodes of the edge under inspection belong to the connected components C_1 and C_2, these two connected components are merged if that edge has a weight less than or equal to a threshold $\min\{\text{diff}(C_1)+\tau(|C_1|), \text{diff}(C_2)+\tau(|C_2|)\}$. Here $\text{diff}(C)$ denotes the internal difference of C, calculated as the largest edge weight in the minimum spanning tree of C. Function $\tau(|C|) = \frac{k}{|C|}$, in which k is a constant and $|C|$ denotes the number of vertices (superpixels) in the connected component C; $\tau(|C|)$ decreases as connected component C grows. Setting k to a larger value leads to larger segments consisting of more superpixels.

2.2.1. Hierarchical segmentation

The hierarchical segmentation procedure starts from the coarsest resolution and gradually moves to finer resolutions (i.e. from top to bottom level in the Gaussian pyramid). We first create a separate graph for each segment obtained in the previous (low resolution) level and then apply the aforementioned graph-based segmentation procedure to subdivide that segment. The hierarchical approach establishes the parent-child relationships between segments, which can be represented as a tree of segments across scales.

We make an additional modification before applying the graph-based segmentation hierarchically. In the original F&H segmentation [10], the image size determines the size of the input graph. However, in the hierarchical setting described here, the size of graph depends on the size of segment (i.e. the number of superpixels) in the previous level. Small values of the constant k in $\tau(|C|)$ usually lead to over-segmentation of large segments, and large values of k lead to under-segmentation of small segments. To alleviate this problem, we set $k = k_{\text{H}} \times \log|G|$, where $|G|$ denotes the segment size (i.e. the number of vertices in the input graph) and k_{H} is a constant ($k_{\text{H}} = 0.6$ in this paper); this way the value of k is adjusted dynamically according to the size of the parent segment.

Because superpixels are generated independently before applying the hierarchical segmentation, the boundaries of segments at a previous (low resolution) level may not align perfectly with the superpixels in the following (high resolution) level. We address this inconsistency using a majority vote to assign superpixels in the child level to a parent segment.

2.3. User interaction

After building the segment tree, each intermediate segmentation consists of the segments that correspond to the leaf nodes in the pruned segment tree. We use a hierarchical exploration strategy similar to Ip *et al.* [20] and build a tool that allows users to unfold a leaf node by clicking on the corresponding segment. The newly-added child nodes of that unfolded leaf node become the leaf nodes in the updated pruned tree, leading to the subdivision of the clicked segment. The colors of segments reflect the changes made to the tree interactively.

3. EXPERIMENTS AND RESULTS

Our method was applied to the zebrafish embryo image [1] obtained through the Journal of Cell Biology (JCB) DataViewer. This image consists of eleven types of tissues: brain, cartilage, eye, intestine, liver, muscle, notochord, olfactory pit, pancreas, pronephric duct, and yolk. We scaled down the image from its original resolution of 281 gigapixels to 6.04 gigapixels to reduce processing time.

Fig. 2 shows an example of the hierarchical segmentation. For simplicity, only part of the whole segment tree is shown. In the first row the root node, which represents the whole image, unfolds into node 5 and node 6 that correspond to segments 5 and 6 in the second row. Segment 5 further subdivides into child segments including segments 26 and 34 in the third row. In the segment tree, each node (except for the root) corresponds to a segment obtained through the graph-based segmentation. The tree edges between nodes 5 and 26 and nodes 5 and 34 represent the parent-child relationships between them.

The user interaction facilitates the reconfiguration of the segment tree following the steps from the first row to the third row in Fig. 2. Initially, the segment tree contains only one (root) node, which corresponds to the yellow segment. Upon clicking the yellow segment, the root node unfolds into its child nodes (e.g. nodes 5 and 6). The two nodes correspond to two segments (blue and dark slate gray) in the second row; the dark slate gray segment contains eye tissues. Another click on the blue segment in the second row further subdivides the blue segment into a few smaller child segments (e.g. the red and brown segments in the third row). The click triggers the unfolding of node 5 into its children nodes 26 and 34 in the segment tree. If the users keep unfolding node 34 (and its child), they can eventually reach the visualization shown in the first column of Fig. 3, which separates the muscle tissue

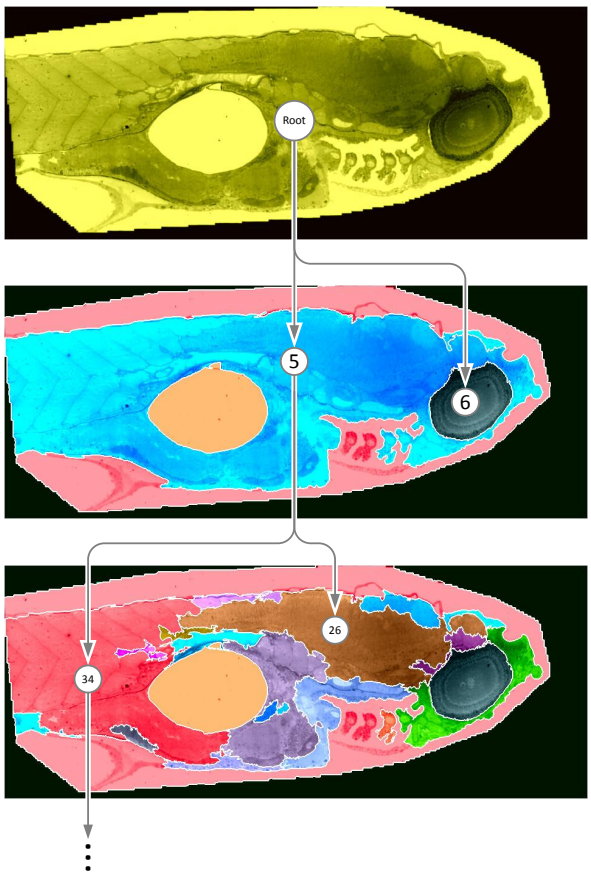


Fig. 2. With our visualization tool, clicking the yellow segment in the first row subdivides the top-right region of the zebrafish embryo into segment 5 (blue) and segment 6 (dark slate gray). Another click on segment 5 subdivides segment 5 into segment 26 (brown) and segment 32 (red).

from the intestine tissue.

Many tissues in the image, such as the muscle, intestine, and pronephric duct tissues (first column of Fig. 3), are indistinguishable at low resolution. Users can hardly detect the subtle differences in texture because the high-frequency information is missing after zooming out. Our method, by enabling user intervention, provides corresponding color-based highlighting such that the muscle-intestine tissue boundaries are easily observable. The muscle tissues can also be split into several regions separated by a thin dark boundary (second column of Fig. 3).

The total processing time of the image applying the hierarchical segmentation is 98.27 minutes. Over half of the time (50.94 minutes) was spent on constructing the graph, which involves creating the joint intensity-NRLBP histograms and calculating the edge weights (χ^2 in Eqn. 1). The processing time reported here is based on an Intel Xeon 2.6GHz CPU. It has been shown that the calculation of LBP and χ^2 can be substantially accelerated on a GPU [21]. We expect a similar

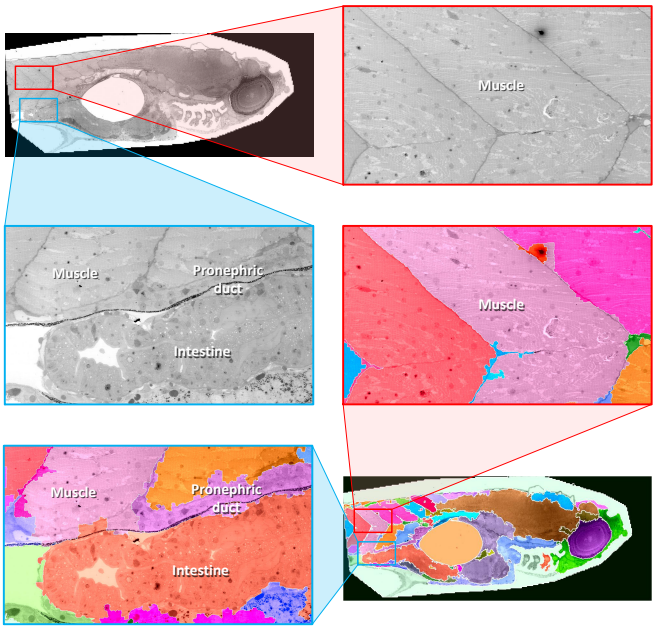


Fig. 3. The texture differences in the down-sampled grayscale images are more distinguishable with color highlights. First column shows the boundary between the muscle, intestine, and pronephric duct tissues; second column shows the boundary between muscle tissues.

degree of acceleration when we migrate the calculation from CPU to GPU.

4. CONCLUSIONS

The disparity between image size and screen resolution poses new challenges to the analysis of subtle microstructures. This work presents a method to assist exploration and analysis of gigapixel images by highlighting regions with different textures. We use a multiscale hierarchical image segmentation algorithm to identify regions that differ in texture based on the joint distribution of intensity and noise-resistant local binary pattern. These segments across resolutions are organized into a tree to facilitate the exploration of intermediate segmentation results interactively. The texture-based color highlights provide visual feedback to assist users in determining which parts of the image need more detail. We are extending this work to images obtained from other modalities.

5. ACKNOWLEDGEMENTS

This work has been supported in part by the NSF Grants 14-29404, 15-64212, NIST Grant #70NANB15H329, the State of Maryland's MPower initiative, and the NVIDIA CUDA Center of Excellence. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the research sponsors.

6. REFERENCES

[1] F. G. A. Faas, M. C. Avramut, B. M. van den Berg, A. M. Mommaas, A. J. Koster, and R. B. G. Ravelli, "Virtual nanoscopy: Generation of ultra-large high resolution electron microscopy maps," *The Journal of Cell Biology*, vol. 198, no. 3, pp. 457–469, June 2012.

[2] M. Tuceryan and A. K. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, pp. 207–246. World Scientific, 1998.

[3] C. Y. Ip and A. Varshney, "Saliency-assisted navigation of very large landscape images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1737–1746, 2011.

[4] R. A. Ruddle, R. G. Thomas, R. Randell, P. Quirke, and D. Treanor, "The design and evaluation of interfaces for navigating gigapixel images in digital pathology," *ACM Transactions on Computer-Human Interaction*, vol. 23, no. 1, pp. 5:1–5:29, Jan. 2016.

[5] S. Bista, Í. L. L. da Cunha, and A. Varshney, "Kinetic depth images: Flexible generation of depth perception," *The Visual Computer*, pp. 1–13, May 2016.

[6] S. Bista, J. Zhuo, R. P. Gullapalli, and A. Varshney, "Visualization of brain microstructure through spherical harmonics illumination of high fidelity spatio-angular fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2516–2525, Dec. 2014.

[7] S. Bista, J. Zhuo, R. P. Gullapalli, and A. Varshney, "Visual knowledge discovery for diffusion kurtosis datasets of the human brain," in *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, pp. 213–234. Springer, Cham, 2015.

[8] H.-C. Nothdurft, "The role of features in preattentive vision: Comparison of orientation, motion and color cues," *Vision Research*, vol. 33, no. 14, pp. 1937–1958, Sept. 1993.

[9] J. Ren, X. Jiang, and J. Yuan, "Noise-resistant local binary pattern with an embedded error-correction mechanism," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4049–4060, Oct. 2013.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[11] J. Yuan, D. Wang, and A. Cheriyadat, "Factorization-based texture segmentation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3488–3497, Nov. 2015.

[12] P. Bajcsy, A. Vandecreme, J. Amelot, J. Chalfoun, M. Majurski, and M. Brady, "Enabling stem cell characterization from large microscopy images," *Computer*, vol. 49, no. 7, pp. 70–79, July 2016.

[13] R. Marée, L. Rollus, B. Stévens, R. Hoyoux, G. Louppe, R. Vandaele, J.-M. Begon, P. Kainz, P. Geurts, and L. Wehenkel, "Collaborative analysis of multi-gigapixel imaging data using Cytomine," *Bioinformatics*, vol. 32, no. 9, pp. 1395–1401, May 2016.

[14] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[15] T. Ojala, M. Pietikäinen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[16] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[17] J. B. Kruskal and M. Wish, *Multidimensional Scaling*, Sage Publications, Beverly Hills, 1978.

[18] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sŭsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[19] C. Cigla and A. A. Alatan, "Efficient graph-based image segmentation via speeded-up turbo pixels," in *Proceedings of IEEE International Conference on Image Processing*, 2010, pp. 3013–3016.

[20] C. Y. Ip, A. Varshney, and J. JaJa, "Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2355–2363, 2012.

[21] S. Yi, I. Yoon, C. Oh, and Y. Yi, "Real-time integrated face detection and recognition on embedded GPGPUs," in *Proceedings of IEEE Symposium on Embedded Systems for Real-Time Multimedia*, Oct. 2014, pp. 98–107.