doi:10.1017/S1431927616006036
Microsc. Microanal. 22 (Suppl 3), 2016

Automated GPU-Accelerated Segmentation of Volumetric Fiber Networks Using a Predictor-Corrector Algorithm.

Pavel A. Govyadinov[1], David Mayerich[1]

[1.] University of Houston, Department of Electrical and Computer Engineering, Houston, Texas, USA.

High-throughput microscopy techniques allow three-dimensional imaging of whole organ tissue samples at sub-cellular spatial resolutions. Filaments are common structures in biomedical tissue samples. However, filaments are particularly difficult to image and segment, due to the need for a high spatial sampling rate, large volume size, and good image quality. The most common examples of filament networks are neuronal and microvascular. These structures are composed of cellular components that form complex interconnected networks of fibers of varying thickness. Such structures often compose < 6% of the entire volume, however appropriate sampling generally requires maximizing the sampling rate between fibers, and therefore producing extremely large data sets. We present an automated predictor-corrector algorithm for segmentation of interconnected networks that utilizes graphics processing units (GPUs) to achieve high speeds.

Knife-edge scanning microscopy is an innovative technique that allows for digitization of volumes at diffraction-limited (≈500nm) sampling resolutions [1]. This tissue preparation and imaging process provides a high signal-to-noise ratio, effective sampling rate, and fast imaging speed. The segmentation algorithm is performed in two parts, using a predictor-corrector similar to that described in previous work [2, 3] and a novel method for branch detection. The algorithm starts with a single seed point and assumes that all of the fibers are interconnected. The predictor-corrector uses step-wise template matching to update three variables, direction, position and size (D_i, P_i, M_i). We sample the volume along some constant user-defined number of direction vectors and advance along the new direction (D_{i+1}) with the lowest cost. The new position is then corrected by sampling the volume along a plane perpendicular to (D_{i+1}) direction vector and choosing the one with the lowest cost (P_{i+1}), as well as sampling templates of different size to match the size of the traced vessel (M_{i+1}) ending a simple step. This process is continued until we reach the edge of the volume or the optimal cost value exceeds a user define threshold. The cost function is defined as the sum of absolute differences between the sampled volume and the optimal template (a straight line) and a single sample two squares perpendicular to one another directed along D_i, centered at P_i as demonstrated in Fig. 1.

Following a single step we perform branch detection by sampling a cylindrical surface from the data of length proportional to a single step-size, radius M_i, directed along D_i and centered at P_i consisting of 8x16 pixels. The cylinder is unwrapped and blob-detection ran using GPU-based iterative voting [4] with five iterations each detected point is then stored in a list as a possible seeds for future iterations. We continue segmenting using the predictor-corrector until the list of possible seeds is empty.

During runtime the dataset is stored in texture memory on the GPU, hence the majority of the algorithm is executed on the GPU eliminating the common CPU-to-GPU data transfer bottleneck. In addition to the speed gained through parallel processing, data analysis is limited to regions near the filament surface, reducing the amount of unnecessary memory fetches and processing. Additionally the correction step allows us to significantly reduce the number of samples necessary for prediction.

A GPU-dependent implementation in addition to fast spatial caching offers several key advantages: low cost and portability. Commonly, segmentation algorithms increase execution speed by utilizing high performance computing resources such as clusters and supercomputers [2]. This algorithm is portable and works with any CUDA capable GPU. While the underlying problem is highly parallel and the performance would improve with every additional GPU available to the system, up to 3 kernels of the algorithm can execute simultaneously in a single GPU with 4GB of available memory.

A single 512x512x512 data volume is segmented in ≈ 221 seconds. Each iteration of find-cost executes in 1.8ns and each iteration of branch detection executes in 5.1ms and branch detection takes 6.1ms per step. The rest of the time is spent on sampling the texture.

In conclusion we propose a fast automated segmentation algorithm that does not require parallel computing resources and functions well on affordable systems. One possible improvements can be made by performing branch detection once per fiber instead of once per step. This would significantly reduce run-time and allow the algorithm to segment the 512^3 volume in under 60 seconds.

References:

[1] D Mayerich, L Abbott, B McCormick, Journal of Microscopy **231**, p. 134-143.
[2] K Al-Kofahi, et al., IEEE Trans. Information Technology in Biomedicine **6(2)**, 171-187.
[3] D Mayerich, J Keyser, IEEE Trans. on Visualization and Computer Graphics **14(4)**, p. 670-681.
[4] B Parvin et al., IEEE Trans. on Image Processing **16(3)**, p. 615-623.

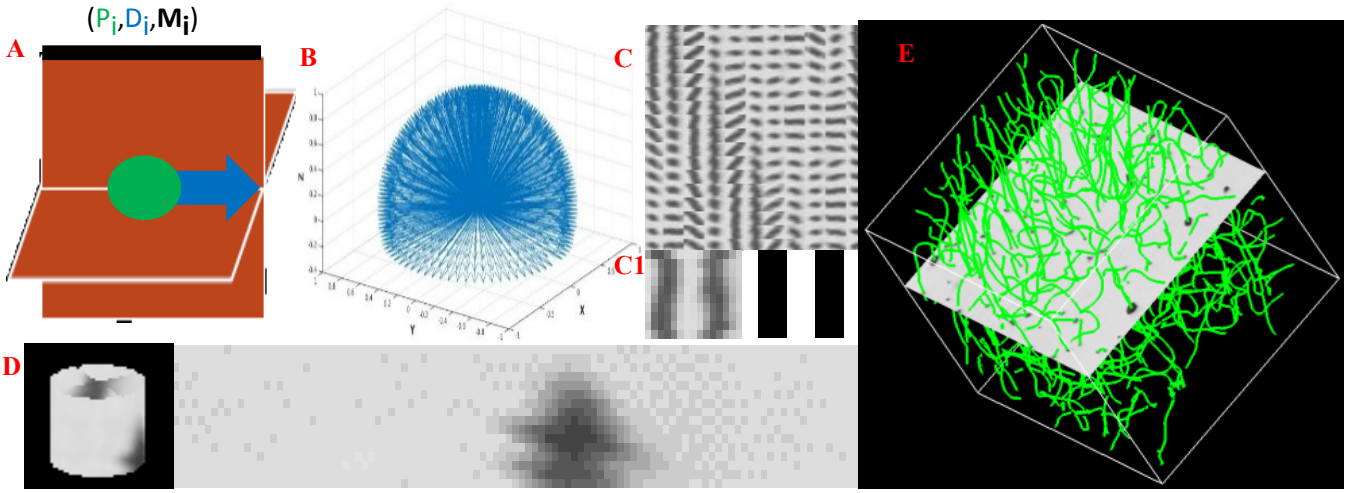

Figure 1. (A) A single sample represented by two perpendicular square planes around a center point. (B) A set of such samples are projected around a central direction, and projected into 2D memory on the GPU (C) and the new direction corresponds to the sample with the lowest cost; based on the difference image of a sample (C1-left) and a generic template (C1-right) as a new direction completing the ***prediction*** step.(D) A cylindrical shape is sampled around the optiomal direction and blob detection is ran on an unwrapped cylinder in order to detect branching fibers(seeds). (E) Execution is terminated when there are no seeds to process, resulting in a complete skeleton of the fibers in the volume.

https://doi.org/10.1017/S1431927616006036 Published online by Cambridge University Press