Neurite Detection using CUDA

GPU accelerated biological imaging for High-Content Analysis

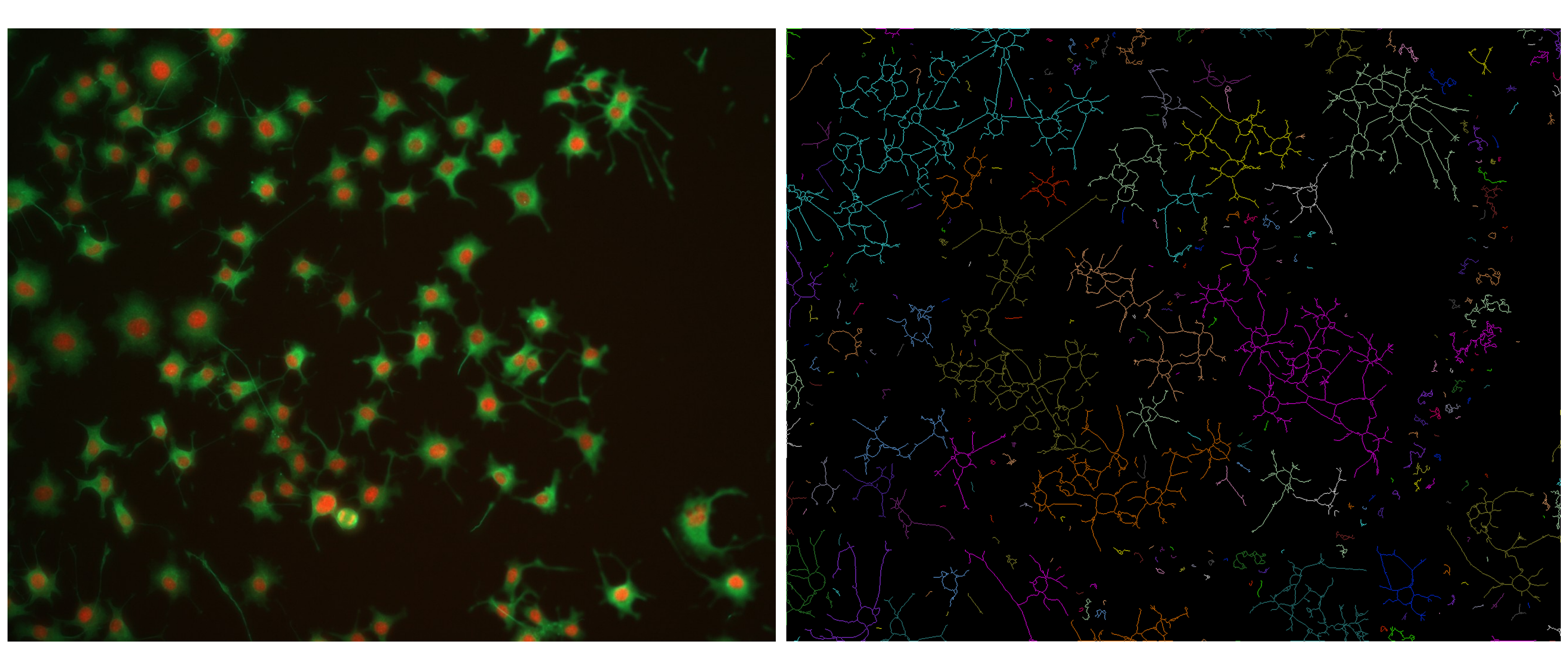
Luke Domanski, Changming Sun, Dadong Wang, Pascal Vallotton, John Taylor
CSIRO Mathematics, Informatics & Statistics - Biotech Imaging

Introduction

The analysis of microscopic neurite structures in images is an important task in biological imaging, particularly for studying the effects of lead compounds on brain diseases or the regeneration of brain cells after trauma. In High-Content Analysis (HCA) hundreds to thousands of microscopy images are processed during automated experiments. The speed at which image processing is performed in these situations can greatly affect the overall workflow throughput. Here we report some early results on GPU acceleration of the Neurite Detection module in our groups' HCA-Vision[1] software for high-content analysis.

Neurite Detection and Analysis

Neurites are long thin tree structures which mediate communication between neurons. Finding and masking neurites is the first step in a broarder task of quantifying neurite structure, e.g. density, length, and branching, and can be performed through a process of linear feature detection[1,2]. Poor uptake of fluorescent dyes into some portions of the neurites and noise in the recorded image can result in many gaps in the detected features, requiring a gap filling process to join appropriate segments. These two processes can be computationally expensive for large or complex images.



Neurons courtesy of Dr. Xiaokui Zhang, Helicon Therapeutics, Inc., USA, with detected neurites (right)

Figure 1 shows the performance breakdown for the Neurite Detection process on a CPU for images with a range of sizes and complexities. The linear feature detection filter (LFD) and gap filling steps consume 76%-79% of the time, while the remainder is consumed by utility functions such as component labeling and mask skeletonization. As a first step, we focused our attention on parallelizing LFD and gap filling on the GPU, as they take the majority of the execution time.

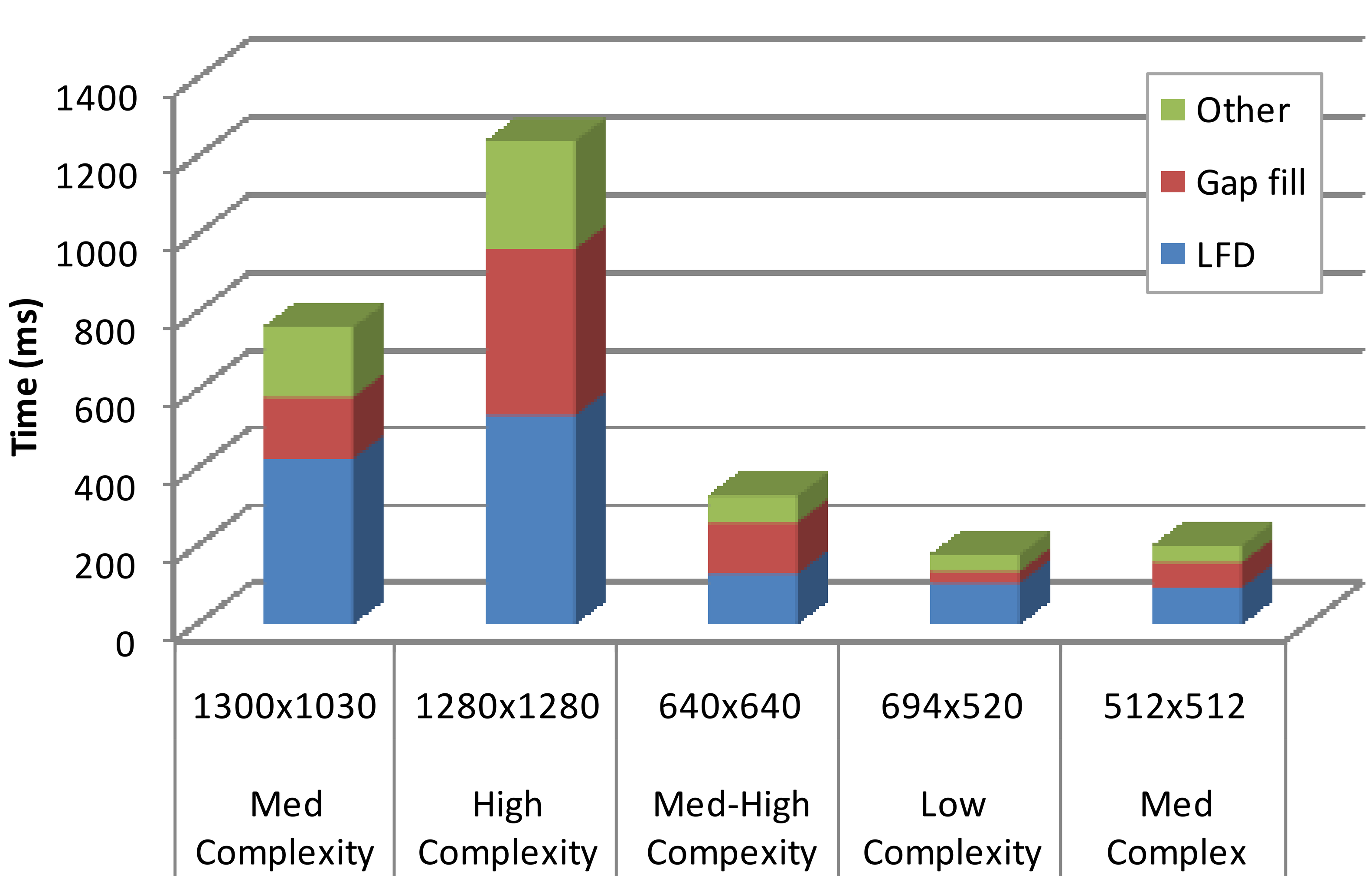
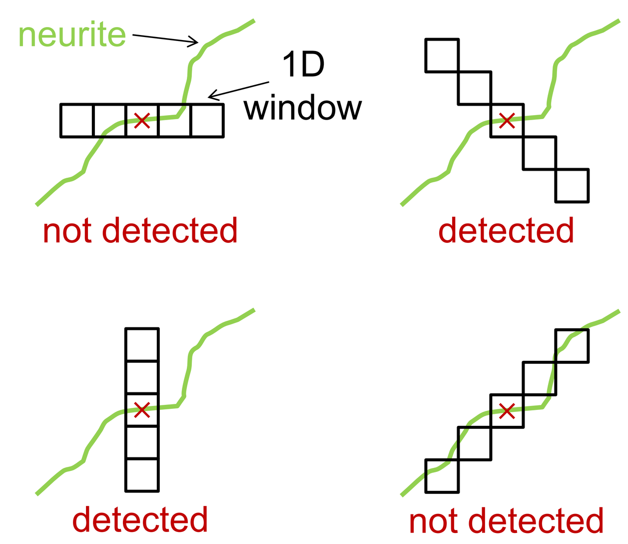


Figure 1: Performance breakdown for neurite detection. The most time-consuming steps are LFD and gap filling, all other utility functions combined consume less than 24% of the time.

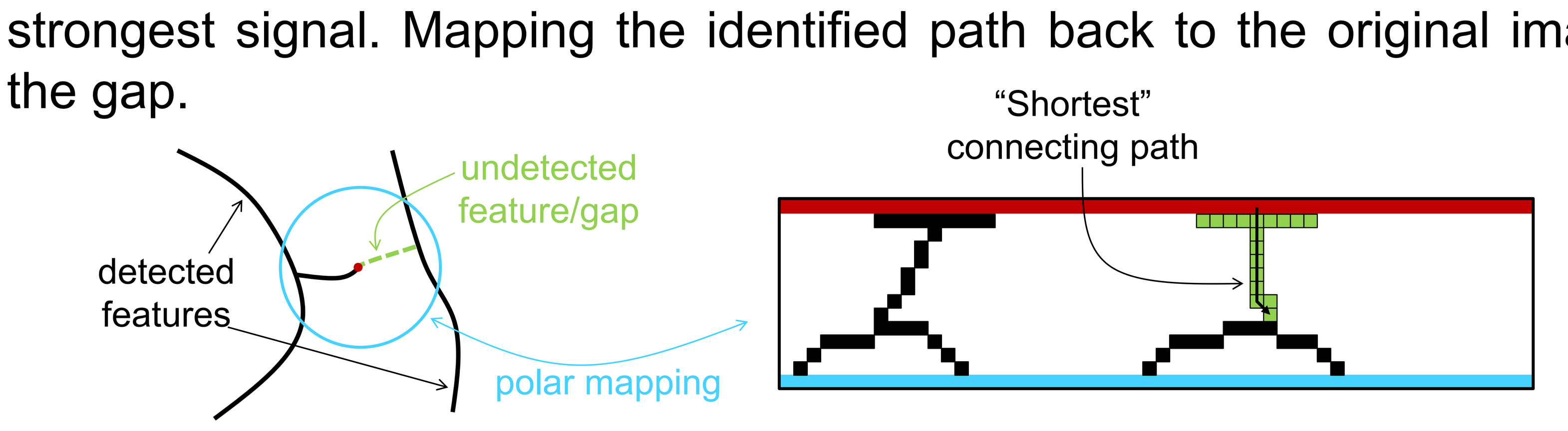
Parallel Algorithm

LFD performs a 1D neighborhood filter at various orientations around a pixel to identify intensity maxima which are indicative of linear feature cross sections. A subsequent symmetry filter over the same neighbourhood(s) ensures that the intensity profile around a maxima is in fact a peak and not a step. A balance is struck between noise resilience and identifying features in close proximity, by applying a smaller secondary LFD and symmetry filter near features indentified with a larger filter window.



Such neighbourhood filters are embarrassingly parallel & straightforward to implement on a GPU by having one thread perform the neighbourhood calculation per output pixel. Efficiency comes from performing the smaller window filter as a subtask of the larger filter, avoiding redundant global memory transfers and filter calculations.

Gap Filling proceeds by performing a directed shortest path search across polar images generated locally at each feature endpoint in the LFD output to see if the endpoint can be connected with another feature along the path of strongest signal. Mapping the identified path back to the original image fills the gap.



Multiple endpoints are addressed simultaneously on the GPU by independent blocks of threads which cooperatively process their assigned endpoint. For both polar image generation and shortest path cost calculation the threads calculate pixel values for an entire polar image row in parallel before proceeding to the next row. The disconnected feature point with lowest path cost is found by the block using a parallel minima reduction[3] over the polar image. Polar image generation is accelerated using texture fetches.

Results

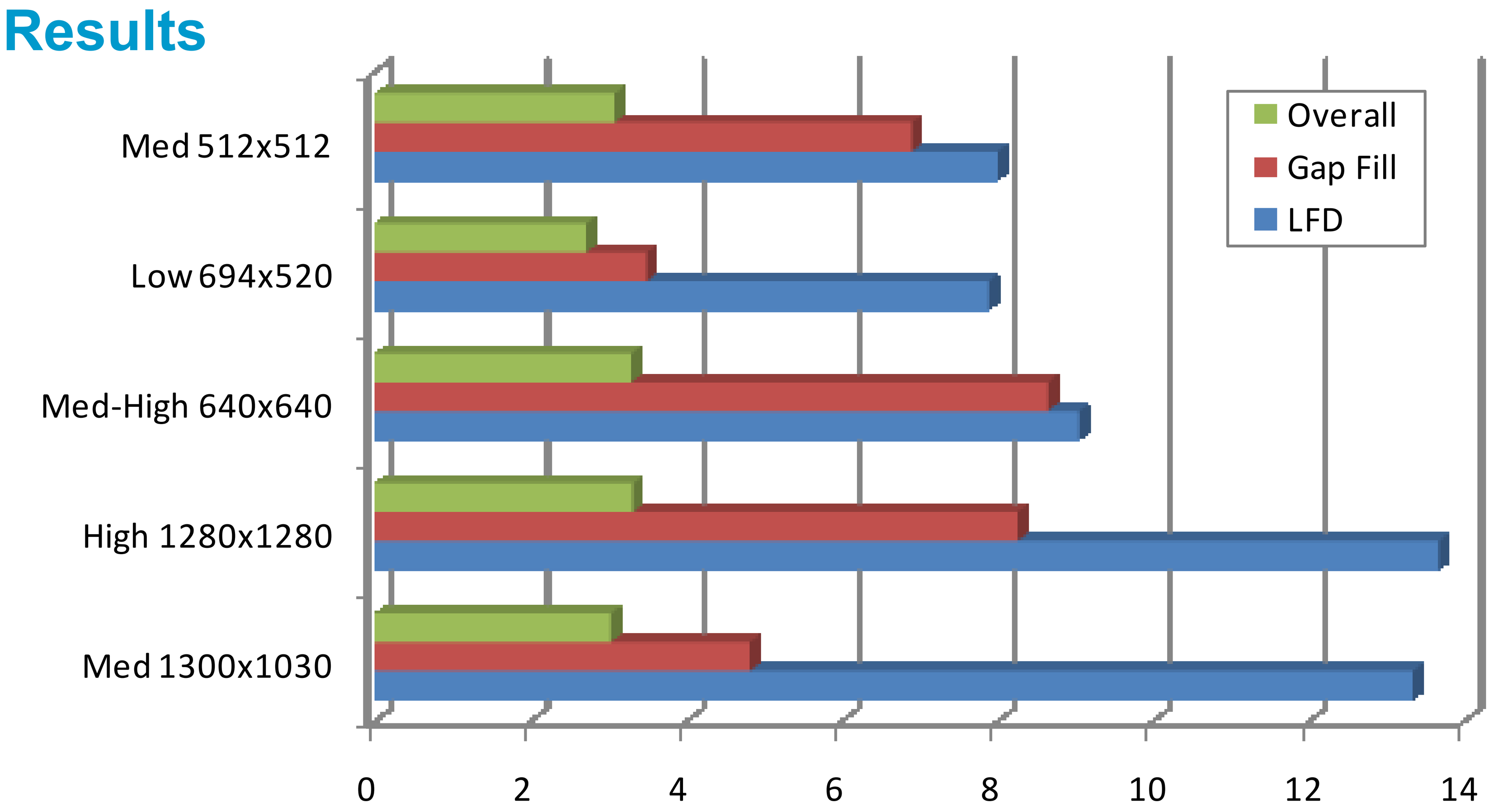


Figure 3: Speedup of the GPU accelerated LFD and gap filling steps in isolation for various images, as well as the overall speed up of the entire process.

The LFD and gap filling steps were sped up by **8-13.6x** and **3.4-8.6x** respectively on a GeForce GTX260 GPU for the tested images. This provides up to a 3.3x speedup for the entire process, including the serial utility functions, which is about 70% of the maximum overall 4.7x speedup obtainable according to Amdahl's law by accelerating LFD and gap filling. Future work will focus on accelerating the component labeling and skeletonization in neurite detection to gain a greater overall speedup, as well as quantitative analysis of the identified neurites in HCA-Vision.

References

1. Vallotton, P. *et al.* (2007), 'Automated Analysis of Neurite Branching in Cultured Cortical Neurons Using HCA-Vision', *Cytometry, Part A 71A(10), 889—895.*
2. Sun, C. & Vallotton, P. (2009), 'Fast Linear Feature Detection Using Multiple Directional Non-Maximum Suppression', *Journal of Microscopy 234(2), 147--157.*
3. Harris, M. (2007), 'Optimizing Parallel Reduction in CUDA', NVIDIA, CUDA SDK Whitepaper,

Acknowledgements

This work was primarily undertaken as part of the Computational and Simulation Sciences Platform within CSIRO.

www.csiro.au/science/CSS.html

Further information

contact: Luke Domanski
phone: (+61) 2 9325 3247
email: luke.domanski@csiro.au
web: www.cmis.csiro.au/IAP/
www.csiro.au

CSIRO