Opportunity and Significance

Computer vision based particle tracking is widely used in fluid dynamics for particle image velocimetry [1], and more recently in life sciences for cell cytometry [2,3]. However, the throughput is generally too low to implement real time tracking. We are developing a high-speed droplet/particle tracking software which achieves ~3000 fps by leveraging high performance graphics processing units (GPUs). It is 249X faster than our previous DMV algorithm implemented on a CPU using OpenCV, and 498X faster than our initial implementation in MATLAB.

This technology could be used for rapid imaging-based cell cytometers and cell sorters, which are being increasingly used for cancer diagnostics.

Technical Objectives

To implement a GPU based accelerated image processing algorithm that can enable rapid analyses using parallel processing. Image processing inherently requires computations on many pixels, and therefore parallel processing can achieve substantial gains compared to single-core processors. This should substantially increase performance resulting in a real time system that can help analyze particles, droplets and cells for cancer diagnostics.

Related Work and State of Practice

Droplet Morphometry and Velocimetry (DMV) is a freely available object tracking tool developed by our group [4], and is currently used in >50 labs in 16 countries worldwide. DMV identifies moving droplets in videos, tracks them over multiple frames, and reports >21 characteristics including trajectory, velocity, size, orientation, deformation, spacing, etc. However, the throughput of the original DMV algorithm, implemented in Matlab, was only 6 frames per second on a full HD video (1920x1080). Low frame rates preclude high-speed applications like imaging flow cytometry and morphometric feedback control.

Technical Approach, Accomplishments and Results

The performance gains are achieved by implementing the DMV algorithm on a GPU so that multicore processing may be leveraged (Fig. 1). Image processing inherently requires computations on many pixels, and therefore parallel processing can achieve substantial gains compared to single-core processors. We implemented our algorithm in C++ using software libraries from OpenCV [6] and NVIDIA CUDA [7], a compute architecture for parallel computing.

We compared the performance of the three implementations of the DMV software: 1) The initial MATLAB-based version [4], 2) the C++/OpenCV implementation using a CPU [5], and 3) The GPU-based implementation reported here. We deployed all three algorithms on a Macbook Pro with a 2.8 GHz i7 CPU, 16 GB of RAM, and an NVIDIA GeForce GTX 750m GPU. We then compared the processing times on a full HD video (1920x1080) of a droplet generator with 32 droplets. The GPU implementation increases performance by a factor of ~498X compared MATLAB. The overall throughput of the GPU implementation is 948.86 frames per second (fps), compared to 6.3 fps in MATLAB, and 12.25 fps on the CPU.

The GPU version was further implemented on a computer with an i7-8700K 4GHz processor and NVIDIA Titan XP GPU and it was able to analyze at 2885 Frames per second. Compared to other reported droplet trackers like ADM [8], our system is ~30X faster.

Next Steps for Development and Test

GPU acceleration opens the possibility of real-time image cytometry of droplets, cells, and particles. In the future, we would like to add a microfluidic sorting module and sort cancer cells based on morphological characteristics.

Commercialization Plan & Partners

Currently a prototype system is being developed. We have partnered with Wayne State University’s School of medicine for early clinical validation.

Upon further clinical validation, we are hoping to acquire the NIH SPIR funding for commercialization.

References