ALGORITHMS FOR THE PARALLEL CONSTRUCTION OF GRAPHS DERIVED FROM IMAGE-LIKE DATA

Abhinav Sinha
sinha45@purdue.edu
Purdue University

Advised by: Prof. David Gleich
dgleich@purdue.edu
Purdue University

January 2021

In Imaging Science, graphs derived from images and image-like data are commonly used to effectively analyze and visualize these data. For a three-dimensional image, the construction of these graphs involves creating a node for each voxel (or, region representing multiple voxels) and then attaching these nodes with weighted edges for each nearby-similar region. The notion of distance and similarity varies depending on the application. Practical applications can easily have millions of nodes and 10 million to a few billion edges, with regions as far away as 50 to 100 voxels. Even though the computation is highly parallelizable, building these graphs from images (especially three-dimensional images) is slow, often taking hours for a simple implementation to build a single graph. In this research project, we aim to accelerate this construction step through GPUs, with an ideally transparent and extensible implementation in Julia.

In the first phase, we will characterize the runtime of a simple initial implementation; the corresponding computational results will serve as a correct "reference" that will be accelerated throughout the course of this research. In this stage, it is crucial to properly identify and rationalize the time and memory sinks of the construction process. The next phase will be the development of a parallel implementation using straight-forward techniques, such as threading, distributed computing, or multiprocessing. Finally, we will apply learned techniques and performance optimizations from the previous phases to develop a complete GPU implementation. As a part of the desire for transparent algorithms, we will explore differences between implementations in C++ and Julia throughout these phases. This comparison will include determining fair trade-offs among ease of implementation, flexibility, modularity, portability, and potential performance gains. Further analysis may include determining if hardware produces appreciable gains in the runtime of these algorithms (although, at this point, we believe it unlikely). The final step will be to test the efficacy of the GPU implementation by applying these algorithms in conjunction with established algorithms as part of a larger problem. In other words, we aim to run a few localized algorithms on the resulting graphs. Localized algorithms do not require the entire graph to be constructed at once, but instead request "pieces" of the graph as needed. Alternatively, we may plan to run simple correlation clustering methods on these graphs as well.

Ultimately, by the end of this project, we will provide a complete implementation for constructing these image graphs in parallel on CPUs and GPUs. There will also be a report on our algorithmic choices for parallelization on CPUs and GPUs, and any results on the more advanced problems.