ME759 Final Project Proposal

Project Title: Image Stitching

Link to git repo for the project:
https://euler.wacc.wisc.edu/iyelurwar/imagestitching.git

Problem statement:

1. Parallelize image stitching algorithm using OpenMP (in Multicore) and Cuda (for GPUs).
2. If time permits, also implement using CUB and Thrust.
3. Draw a comparison and bottleneck analysis for all versions of the algorithm.

Motivation/Rationale:

Image stitching algorithms are widely used in generating a panoramic image from multiple non-panoramic images. When implemented sequentially, the application can be painfully slow in case of large inputs. However, this kind of application does have the potential for a SIMD application where we can work on multiple pixels in parallel. Given the immense hardware resources (multi-cores and GPUs) at our disposal, we intend to utilize this potential parallelism inherent in the algorithm.

Explain how you contemplate going about it: First, we need to understand the image stitching algorithm and sequential implementation of the algorithm. We need to get OpenCV + Cuda working on Euler. We will then use OpenMP to implement the parallel CPU version and CUDA to implement the parallel GPU version. The algorithm will be using OpenCV libraries. If required we will be using CUB and Thrust libraries for the parallel GPU implementation.

ME759 aspects the proposed work draws on:

1. OpenMP Parallel Programming and NUMA aspects
2. GPU Computing with CUDA
3. CUDA Profiler and optimization
4. CPU and GPU Memory Hierarchy

Team member[s]:

· **Name**: Prathamesh Patel
· Email: prathamesh.patel@wisc.edu
· Home department + advisor : CS + No advisor

· Student's role in the project: Handling the infrastructure and OpenMP implementation

· **Name**: Rahul Singh
· Email: rsingh76@wisc.edu
· Home department + advisor: ECE + Prof. Joshua San Miguel
· Student's role in the project: Part of CUDA implementation along with analysis

· **Name**: Ishan Yelurwar
· Email: yelurwar@wisc.edu
· Home department + advisor: ECE + Prof. Joshua San Miguel
· Student's role in the project: Part of CUDA implementation along with analysis

Deliverables: Source Code for sequential, parallel CPU and GPU versions, Final Report with analysis of the performance on CPU and GPU along with various optimizations and Sample output and inputs.

How you will demonstrate what you accomplished:

1. Speed-up of all implementations (OpenMP (Parallel on CPU), Cuda (GPU), if possible CUB and Thrust on GPUs) compared to baseline sequential implementation.
2. A final report comparing and analyzing the performance variations and explaining the reasoning.

Milestone:
1. Get OpenCV + CUDA working on Euler.
2. Understanding and executing the native algorithm.
3. Implementing the parallel OpenMP version along with basic analysis.
4. If time permits start working on the parallel CUDA implementation.