Progress Report

Milestone: indicate what will be accomplished by April 23 milestone (9 pm).
1. A complete sequential program of panoramic stitching.
2. Complete parallel section profiling to identify potential parallel speedups.
3. Start some parallel implementation and evaluation.

Part 1. A complete sequential program of panoramic stitching

Part 1.1. Installing dependencies
OpenCV 3.4.1 (with 3-rd library) and CUDA 10.0 are dependencies used for this project. I have configured denpendencies on both local computer and Euler. Installation Instructions are provided as shell script:

install_euler.sh
install_pc.sh

Part1.2. Sequential programs: Stitcher.cpp, stitch_seq.cpp, stitch_binary.cpp
Stitcher.cpp: Stitcher class that is able to do panorama stitching on two images
stitch_seq.cpp: Stitch multiple images in a sequential manner. For example let plus sign be stitch operation, img_panorama = ((img1 + img2) + img3) + imag4
stitch_binary.cpp: Does exactly the same as stitch_seq.cpp. However it stitches multiple image following using divide and conquer method. E.g. img_panorama = ((img1 + img2) + (img3 + img4))
stitch_seq_opt.cpp: Optimized sequential reduction algorithm. Reduces number of feature detection & matching, homography computation. More details will be included in final report.

All programs stitches images horizontally.

Part 2. Program profiling

For panorama stitcher, I have identified dependencies in the program and start search parallel programming alternatives to replace functions like feature detection, feature matching, homography computation, etc.

For panorama stitching with more than two images, this problem is essentially a reduction problem. However it is not as trivial as sum reduction, because different reduction method would results in different complexities in feature detection & matching and perspective warping. More details will be included in final report.

Part 3. Parallel program: Stitcher_CUDA.cpp

“stitcher_cuda.cpp” includes feature detection & matching, perspective warping with GPU. Still working on optimizing RANSAC homograph matrix computation with GPU. Partially completed multiple image stitching with CUDA.