Final Project Checkpoint

Urvi Agrawal and Sophie Smith
urvia@andrew.cmu.edu
sophiesm@andrew.cmu.edu
11/18/2019

Current Progress

Given the our recent change in topic, we are progressing well towards our end goals. We have currently implemented the sequential algorithm for Lucas-Kanade object tracking (with the exception of data loading/storing as will be elaborated). Throughout this project, we've had multiple setbacks, none of which are detrimental to our overall performance. The main setback we have with our performance is the change in topic. As discussed in a meeting with the Professor, we realized after discussing with our Security Professor, that there wasn't much room for parallelization with our previous topic idea. Given that, we decided to switch project domains to computer vision where we decided to parallelize the Lucas-Kanade object tracking algorithm. This set us back approximately 1.5 weeks in the project, but we've set out some clear goals to get caught up, including dedicating some additional time to the project over the upcoming school break. After we concluded on switching the project, we encountered another setback with packages to use in the project. Initially, we wanted to use OpenCV as a framework to help with matrix computations and reading and loading image data for input and eventual display. After experimenting for multiple hours and looking at previous projects where OpenCV was used, we realized this was not the best approach for our project. The setup to get OpenCV on the GHC machines is unintuitive and requires a significant amount of storage. After even downloading relevant packages, there is no clear setup to integrate the package with use in our project. We found similar results across the two groups who previously used OpenCV for their 15-418 projects. In their reports, they discussed spending hours or weeks simply trying to integrate OpenCV into their project. Given this, we decided to analyze how useful the package even is for our project and determine if it's necessary for our overall implementation. After considering the algorithm, we realized it's really only useful for loading the data from images and producing the output images, not actually for the algorithm itself. Thus, we decided to omit using this package and instead we plan to store the pixel representations of images in text files (similar to the input for nbody homework problems) and we can load the data through file reading operations instead. Because of this process, we were set back in configuring the data loading and storage steps, however, we were still able to implement the remainder of the sequential algorithm.

As discussed in the previous paragraph, we had some setbacks both in terms of project topic and packages to use. Despite these, we made a significant amount of progress with implementing the sequential algorithm. Omitting the image loading and storing steps, we implemented all other parts of the sequential algorithm including matrix operations and the convergence loop. All that is left to get this working is to perform image preprocessing to represent the image pixels in another format which we can load into the algorithm and perform object tracking on. We believe in the next couple of days we'll be able to finish this step, thus, completing the sequential portion of the project overall.

Goals and Deliverables

Our goals and deliverables remain the same for the final poster presentation and formal report. Given the sequential algorithm that we've implemented, the bare minimum we wish to complete for the final project is to parallelize this using the three models discussed during class: CUDA, Open MP, and MPI. Given these optimizations, we will measure performance and timing/speedup results

and analyze why the performance is as observed. The final demonstration will include these results presented on a paper including various graphs displaying the quantitative measurements. Ideally, we will also have a computer on display to show the visual results of our tracking and emphasize the real-world impact our optimizations can have.

Detailed Schedule

November 18 - 20

(Both) Finish the sequential algorithm. While the sequential version of the algorithm is implemented, we need to finish the process of loading image frames and displaying tracking results across the frames. This is a minor part of the overall implementation, so we definitely think it's feasible within this time span.

November 21 - 24

(Both) Measure performance results and timing measurements on the sequential implementation of the algorithm. Include these results in the writeup.
(Urvi) Begin to restructure the code for GPU applications using CUDA. This will require significant restructuring, so this will most likely take longer than this time period.
(Sophie) Parallelize the sequential algorithm using shared address space abstraction. This will be completed using Open MP. We're anticipating this step to not require a significant amount of restructuring or implementation modifications, thus, we believe it's feasible in this time span.

November 25 - 27

(Sophie) Measure performance of shared address space and perform further optimizations to improve the parallelism and speedup. Incorporate the results for this model in the formal report. Additionally, begin with the message passing implementation.
(Urvi) Continue working on the CUDA implementation.

November 28 - December 1

(Urvi) Finish the CUDA implementation of Lucas-Kanade. Perform all required measurements for analysis and incorporate these results into the final formal report.
(Sophie) Complete message passing implementation and measure performance. Include these results in the writeup as well.

December 2 - 4

(Both) Formalize the writeup with all previously computed results. Begin to design the final presentation. Additionally, work on our extended goal which requires restructuring the pipeline to accomplish more parallelism.

December 4 - 8

(Both) Finish the extended goal, formal report and poster. Prepare for the presentation.

Preliminary Results

At this current point, we don't have preliminary results. As described above, currently we have implemented the sequential algorithm, disregarding the initial image loading and saving steps. Once we add these two steps, we should be able to display results and measure performance metrics about the sequential implementation, after which we will update the webpage with results. The goal is to complete this within the next couple of days.

Main Concerns

We don't have any main concerns at the moment. Due to switching projects, we feel slightly behind our peers, however, given our planning we should be able to complete all desired goals by the final project deadline. We both have a significant amount of free time soon, especially given the Thanksgiving vacation, so we plan to put in extra time then to catch up on the project.