Parallel Object Detector with Generalized Hough Transform
15618 Project Milestone Report

Yijie Chen(yijieche), Yiliu Xu(yiliux)

November 30, 2022

1 URL

https://github.com/YijieChen720/Parallel-Object-Detector-with-Generalized-Hough-Transform

2 SUMMARY

So far we implemented a sequential version of Generalized Hough Transform Detector. It reads a template image (contains a single shape) and a source image (one / multiple template shapes under rotation / translation with noise), and output a image with all template shapes marked.
We are still developing the parallel version on CUDA. We designed 3 different parallel strategies for the accumulation phase. We implemented all the parallel code and have a working version. We are still actively debugging the code to handle some memory allocation errors. (Due to some technique issue, we lost access to GHC CUDA hardware for one week).

3 PRELIMINARY RESULTS

Below are sample test image from the sequential implementation. Figure 1 (560x558) shows a template image, and Figure 2 (1880x1202) shows a source image. By running our code, all the template shapes will be identified and marked on the output image in red (Figure 3).

In addition, we implemented a naive parallel version, and two improved versions. The parallel code still has some issue regarding CUDA memory allocation. But the results (Figure 4) look promising on small-size test image (270x338).

Note that 3 times were recorded: total is overall runtime of processing template and source; process template is the preparation phase; process source accumulate is the step to run the 4D accumulator. All run time values are in millisecond.

Figure 1: Template image

Figure 2: Source image

Figure 3: Output image

	sequential		naïve		improved 1		improved 2	
	runtime	speedup	runtime	speedup	runtime	speedup	runtime	speedup
total	194.199	1	16.527	11.751	11.482	16.913	8.712	22.291
process template	2.349	1	6.974	0.337	6.966	0.337	7.034	0.334
process source & accumulate	173.299	1	9.522	18.200	4.488	38.615	1.732	100.057

Figure 4: Preliminary result

As out next steps, we will prioritize the parallelization of processing template, as we are curious why this image processing step do not achieve expected speedup. We will also debug the code and fix the memory allocation error, which will allow us to run a large test image.

4 UPDATED GOALS AND DELIVERABLES

75%: Implement a working version of CUDA Generalized Hough detector and try our best to optimize it.
Comment: We updated this 75% goal to be our original 100% goal, because we are making good progress implementing the CUDA code. The remaining parts are debugging and benchmarking. In case there's a subtle bug that we cannot find in one of our parallel strategies, we hope to have at least one working version.

100%: Implement multiple working versions of CUDA Generalized Hough detector with arbitrary scale and rotation. Conduct a detailed analysis on each parallel strategy. Note each part of the pipeline will need to be highly parallelized.

125%: Optimize the Generalized Hough detector by using Halide and improve detector's performance.
Comment: We will still try our best to achieve this. But given the technical difficulty we experienced related with GHC machines which delayed our development progress, we are on a tight schedule if we plan for this.

5 PLAN FOR POSTER SESSION

We plan to show (1) a demo on how our detector works with test images (2) some performance analysis graphs

6 UPDATED SCHEDULE

11/06 - 11/12 Conduct literature review to better understand the problem. Environment setup on GHC machine. Implement the sequential version.

11/13 - 11/19 Design parallel strategies for each component in the pipeline.

11/20 - 11/26 Implement a parallel version in CUDA: Image Processing (Yiliu), Accumulator (Yijie)

11/27 - 11/30 Merge code and debug. Time each part of code. Generate a computationally expensive testcase. (Yiliu, Yijie)

12/1 - 12/03 If we are able to achieve expected speedup, start analyzing and comparing different parallelization approaches; otherwise keep debugging and optimizing our solution. (Yiliu, Yijie)

12/04 - 12/08 Write report and prepare for poster session. (Yiliu, Yijie)