

# Parallel Object Detector with Generalized Hough Transform

## 15618 Project Proposal

Yijie Chen(yijieche), Yiliu Xu(yiliux)

November 9, 2022

### 1 URL

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

### 2 SUMMARY

We are going to implement a parallel object detector using generalized Hough Transform on GPU.

### 3 BACKGROUND

Hough Transform is an image processing technique used for detecting parametric shapes, such as lines and circles. A generalized version of Hough Transform proposed by Ballard is able to detect any arbitrary shapes, even ones that cannot be described by equations, by encoding the input template shape and mapping the raw image into a Hough Transform space. We believe an efficient parallel implementation for generalized Hough detector will be valuable in a wide range of areas, such as medical image processing, and robot vision.

The pipeline of implementing a Generalized Hough Detector is as follows:

- Apply a derivative filter (e.g. Sobel filter) on both the template image and raw image to detect all edge pixels
- Encode all edge points of the template shape into an R-table
- Apply Generalized Hough Transform on the raw image, accumulate the vote of each edge pixel into a 4D accumulator matrix  $(x_c, y_c, s, \theta)$
- Traverse the accumulator matrix and find the local maxima

## 4 THE CHALLENGE

We aim to construct a 4D parameterized Hough space (x, y, scale, rotation) to detect arbitrary input shapes. Above all, generalized hough algorithm requires more computation and processing for non-analytic curves than simple lines or circles. Besides, compared with the detection with invariant scale and rotation in 2D space, this detection problem adds two more degree of computation complexity. In addition, it requires a large memory capacity to store the 4D accumulator. It's challenging to design and deploy CUDA program to achieve high performance, memory efficiency and reduced communication.

## 5 RESOURCES

The starter C++ code will be structured from this github repo and Assignment 2. Note that the reference code uses OpenCV to compute image gradient, but we will write the convolution filter from scratch in order to parallelize the code using CUDA.

Some other research papers we will look into:

Ballard, Dana H. "Generalizing the Hough transform to detect arbitrary shapes."

Chen, Su, and Hai Jiang. "Accelerating the hough transform with CUDA on graphics processing units."

## 6 GOALS AND DELIVERABLES

75%: Implement a working version of CUDA Generalized Hough detector with known scale and rotation (2D accumulator matrix)

100%: Implement a working version of CUDA Generalized Hough detector with arbitrary scale and rotation. 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.

## 7 PLATFORM CHOICE

We choose to implement this project using C++ on CUDA, as it's an image processing problem that fits well with GPU's feature of massively parallel execution.

## 8 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.

11/27 - 12/03 Benchmark and optimize the parallel version.

12/04 - 12/08 Write report and prepare for poster session.