# Object Recognition based on Generalized Hough Transformation Proposal


Zhongyao Chu
NYU
zc1213@nyu.edu

Hupo Tang
NYU
ht1073@nyu.edu



## Abstract

*Hough-Transformation[2] is universally used to detect complicated shapes like circle, ellipses, etc. In the real life, objects have far more complicated shapes which may consist of several simple shapes. Generalized-Hough-Transformation[1] proposed by Ballard offers such a method by which object in real life can be detected via a pre-defined template and several matching search in the parameter space. This project is to implement Generalized Hough Transformation and, if time allowing, looking for improvements on the origin Generalized-Hough-Transformation.*


## 1. Introduction

Shapes like circles squares, etc. , which have well-defined algebra equations containing several parameters, can be detected using Hough-Transformation. In a more complicated situation, rigid objects with various outlines is hard to define using equations. Generalized-Hough-Transformation[1] uses gradients and orients vectors (from the origin) to search for corresponding shapes in parameter space, then decide the final object via a majority vote based on the gradients of each points we need to consider.

## 2. Methods

The methods of this project is firstly based on the paper of Hough-Transformation[2] and Generalized-Hough-Transformation[1]. The implementing programming language is python, with several required packages revealed in the README.md file in the github-repo in the section Code.

The implementation of this project is expected to be realized by merely basic mathematical packages like numpy and scipy, without any further image processing libraries.

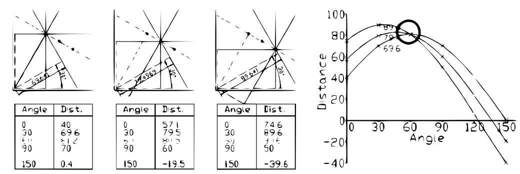


Figure 1. The basic hough-transformation. Lines could be found in two parameters can be detected in the parameter space in the right graph.

### 2.1. Basic Hough-Transformation

For shape like lines and circles, we use the standard equations definition of these graphics to search for peak in the parameter space. An example is displayed in **Figure.1**.

### 2.2. Generalized Hough Transformation

There are two possible difficulties may occur in the above Hough transform method:
(a) the shape has to be described by an equation
(b) the number of parameters $n$ (dimensions of the parameter space) may be high.
Thus, a more general way to detect arbitrary shapes is needed.

#### 2.2.1 Build a table for the given shape

Prepare a table with $k$ entries each indexed by an angle $\phi_i$ ( $i = 1, \cdots, K$) which increases from 0 to 180 degrees with increment $180/k$, where $k$ is the resolution of the gradient orientation (see below). Define a reference point $(x_c, y_c)$ somewhere inside the 2D shape (e.g., the gravitational center). For each point $(x, y)$ on the boundary of the shape, find two parameters :

$$\begin{cases} r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \beta = tan^{-1}\ (y - y_c)/(x - x_c) \end{cases}$$

and the gradient direction $\angle G$. Add the pair $(r, \beta)$ to the table entry with its $\phi$ closest to $\angle G$.

Then, prepare a 2D Hough array $H(x_c, y_c)$ initialized to 0.

| | | | | |
|---|---|---|---|---|
| $\phi_1 = 0$ | $(r, \beta)_{1,1}$ | $(r, \beta)_{1,2}$ | $\cdots$ | $(r, \beta)_{1,n_1}$ |
| $\phi_j$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $\phi_1 = \pi$ | $(r, \beta)_{1,1}$ | $(r, \beta)_{1,2}$ | $\cdots$ | $(r, \beta)_{1,n_1}$ |

#### 2.2.2 Detection of the shape and its locations in image

For each image point $(x, y)$ with $|G(x, y)| > T_s$, find the table entry with its corresponding angle $\phi_j$ closest to $\angle G(x, y)$. Then for each of the $n_j$ pairs $(r, \beta)_i$ ( $i = 1, \cdots, n_j$) in this table entry, find

$$\begin{cases} x_c = x + r\ cos\ \beta \\ y_c = y + r\ sin\ \beta \end{cases}$$

Then, increment the corresponding element in the H array by 1:

$$H(x_c, y_c) = H(x_c, y_c) + 1$$

All elements in the H table satisfying $H(x_c, y_c) > T_h$ represent the locations of the shape in the image.

It is desirable to detect a certain 2D shape independent of its orientation and scale, as well as its location. To do so, two additional parameters, a scaling factor $S$ and a rotational angle $\theta$, are needed to describe the shape. Now the Hough space becomes 4-dimensional $H(x_c, y_c, S, \theta)$. The detection algorithm becomes the following:

For each image point $(x, y)$ with $|G(x, y)| > T$, find the proper table entry with $\phi_j = \angle G(x, y)$. Then for each of the $n_j$ pairs $(r, \beta)_i$ ( $i = 1, \cdots, n_j$) in this table entry, do the following for all $S$ and $\theta$: find

$$\begin{cases} x_c = x + r\ S\ cos(\beta + \theta) \\ y_c = y + r\ S\ sin(\beta + \theta) \end{cases}$$

and increment the corresponding element in the 4D H array by 1:

$$H(x_d, y_c, S, \theta) = H(x_c, y_c, S, \theta) + 1$$

All elements in the H table satisfying $H(x_c, y_c, S, \theta) > T_h$ represent the scaling factor $S$, rotation angle $\theta$ of the shape, as well as its reference point location $(x_c, y_c)$ in the image.

### 2.3. Data

The model within this very project doesn't require extra data for training or processing.

## 3. Grouping

**This final project of course CS-GY 6643 Computer Vision is proposed and expected to be co-finished by student Zhongyao Chu (netID zc1213) and Hupo Tang(netID ht1073).**

## 4. Code Repo

The code of this project is stored and managed in the github-repo: `https://github.com/edwardchor/OR-GHT.git`. This repo will be published after the second milestone of this project.

## References


[1] D. H. Ballard. Generalizing hough transformation to detect arbitrary shapes. 1979. 1

[2] P. Hough. Method and means for recognizing complex patterns. 1962. 1