Object Detection using Circular Hough Transform

Introduction to the Hough Transform

Fatoumata Dembele
dembelef@msu.edu
ECE 448 – Design Team 3

Executive Summary

One of the major challenges in computer vision is determining the shape, location, or quantity of instances of a particular object. An example is to find circular objects from an input image. While numerous feature extraction techniques are openly available for circle detection, one of the most robust and commonly used methods is the Circular Hough Transform (CHT). The goal of this application note is to provide the reader with an understanding of the operations behind a CHT. First an overview of the Hough Transform will be given, then an explanation of how a Hough Transform can be used to implement a Circular Hough Transform. Finally, a discussion on limitations of the Hough Transform to accurately detect edges points in noisy images will be given.

Table of Contents

Keywords:

Hough Transform (HT), Generalized Hough Transform (GHT), Circular Hough Transform (CHT), edges.

Introduction:

One on the most challenging tasks in Computer Vision is feature extraction in images. Usually objects of interest may come in different sizes and shapes, not pre-defined in an arbitrary object detection program. A solution to this problem is to provide an algorithm than can be used to find any shape within an image then classify the objects accordingly to parameters needed to describe the shapes. A commonly used technique to achieve this is the Hough Transform. Invented by Richard Duda and Peter Hart in 1992, the HT was originally meant to detect arbitrary shapes of for different objects. The Hough Transform was later extended to only identify circular objects in low-contrast noisy images, often referred to as Circular Hough Transform.
This method highly depends on converting gray-scale images to binary images using edge detection techniques such as Sobel or Canny. The goal of this technique is to find irregular instances of objects within a pre-defined set of shapes by a voting procedure. In the following, we will look at the Hough Transform, Circular Hough Transform, and their limitations.

Hough Transform

Generalized Hough Transform:

The GHT is a modified version of the HT that not only searches for analytically defined shapes, but also arbitrary shapes (shapes that cannot be defined by an analytical equation).
This method uses the principle of template matching, which relies on detecting smaller elements matching a template image.

Theory:

The HT can be described as a transformation of a point in a 2 dimensional region to a parameter space, dependent on the shape of the objects to be identified. The basic functionality of the HT is to detect straight lines. A straight line in the x,y-plane is described by:

$$y = m*x + b \tag{1}$$

This line is represented in the Cartesian coordinate system by its parameters b and m where m is the slope and b is the intercept. Due to the fact that perpendicular lines to the x-axis can give unbounded values for parameters m and b (b and m rises to infinity), lines are parameterized in terms of theta θ and r such that:

$$r = x*\cos(\theta) + y*\sin(\theta), \quad \text{for } \theta \in [0, \pi] \tag{2}$$

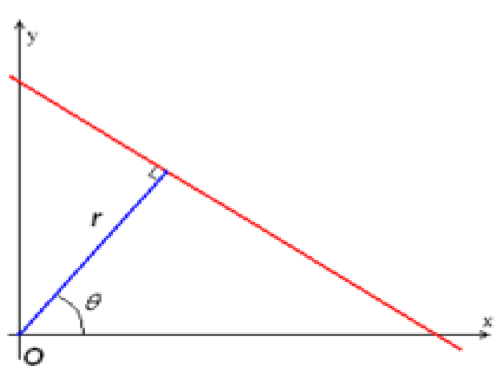


Figure 1

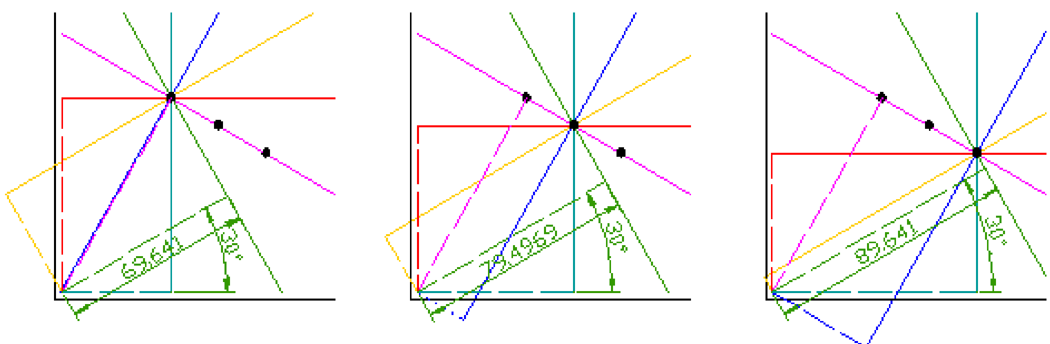
where r is the distance between the line and the origin, θ is the angle. Thus, given x and y, every line passing through point (x, y) can uniquely be represented by (θ, r). Both θ and r have finite sizes. The distance r will have the maximum value of two times the diagonal of the image.

An edge detector is commonly used to provide a set of points that represents the boundaries in the image space.
Equation (1) corresponds to a sinusoid curve in the (r, θ) plane unique to that point. If several points are located on the same line, they produce sinusoids crossing at the parameters for that line.

Implementation:

An array, called an accumulator is used to detect the existence of a line y = m*x +b. In other words, it counts the votes for sets of parameter values. For instance, the HT detecting straight lines has two unknown parameters m and b. The HT then proceeds to a voting in order to determine which pixel is likely to be an edge on the image. If there is enough evidence that a point might be an edge, then increment the parameter values corresponding to the lines that would cause this line to appear in the edge image (Luke Fletcher, 2).

The following is an illustrated example.



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4

Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	-19.5

Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	-39.6

Figure 2 ***(Source: Wikipedia online Hough Transform)***

Each point in the Figure 2 is a point in the edge image.
Consider drawing several lines going through each data point.
A line, starting from the origin and perpendicular to each solid line is then drawn, the length and angle of each dashed line is then measured and stored in the tables in Figure 2.

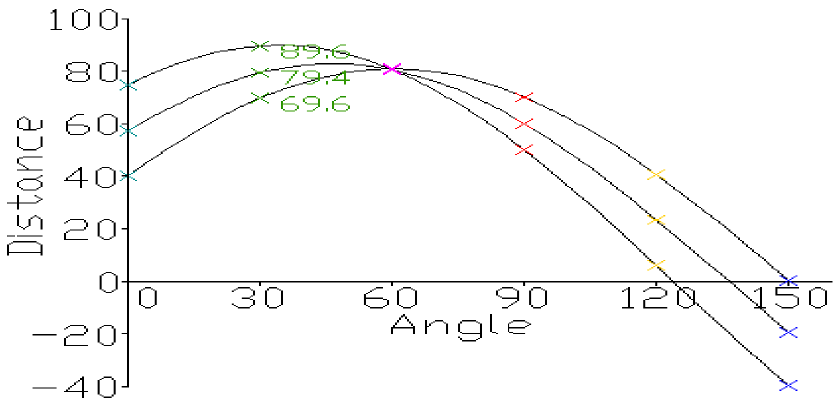
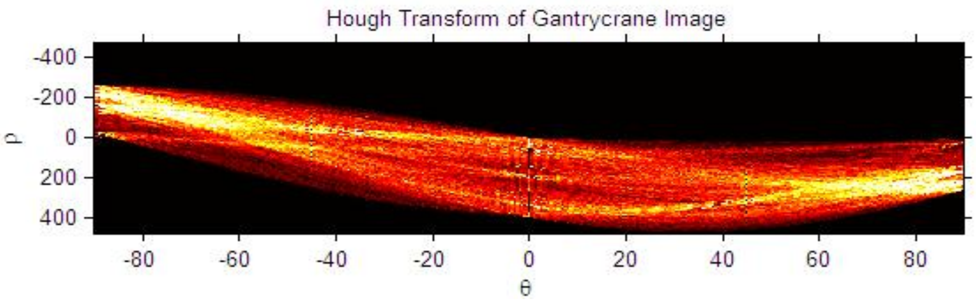


Figure 3 *(Source: Wikipedia online Hough Transform)*

The graph is Figure 3 shows how different lengths are related to different angles. The point (purple) where the curves intersect provides the distance and angle from all 3 points located on the same line. The value of the parameters (angle and length) of the line is then added to the array for future reference.

Matlab example:

Below is an example of applying the Hough transform on a Gantry crane image.



Circular Hough Transform

Implementation:

Unlike the linear HT, the CHT relies on equations for circles. The equation of the a circle is,

$$r^2 = (x - a)^2 + (y - b)^2 \quad (3)$$

Here a and b represent the coordinates for the center, and r is the radius of the circle.
The parametric representation of this circle is

$$x = a + r*\cos(\theta)$$
$$y = b + r*\sin(\theta) \quad (4)$$

In contrast to a linear HT, a CHT relies on 3 parameters, which requires a larger computation time and memory for storage, increasing the complexity of extracting information from our image.
For simplicity, most CHT programs set the radius to a constant value (hard coded) or provide the user with the option of setting a range (maximum and minimum) prior to running the application.

For each edge point, a circle is drawn with that point as origin and radius r. The CHT also uses an array (3D) with the first two dimensions representing the coordinates of the circle and the last third specifying the radii. The values in the accumulator (array) are increased every time a circle is drawn with the desired radii over every edge point. The accumulator, which kept counts of how many circles pass through coordinates of each edge point, proceeds to a vote to find the highest count. The coordinates of the center of the circles in the images are the coordinates with the highest count.

Geometric Modeling:

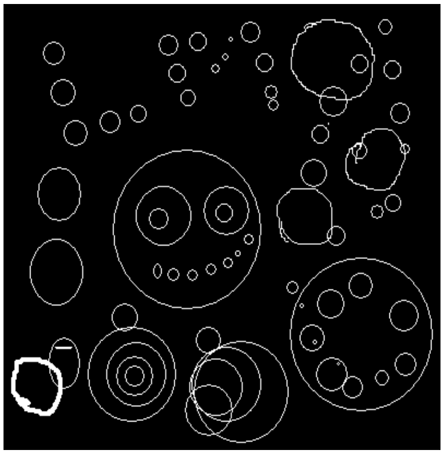


Figure 4

Input image – after edge detection

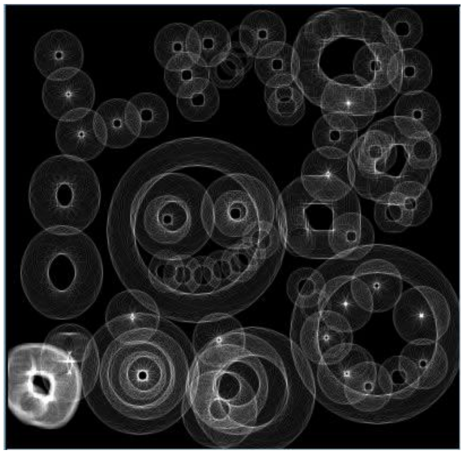


Figure 5

Image after voting

Matlab Example:

The following code was writen in Matlab to detect multiple disks in an image using the HT.

Steps:

1. Turn the colored image into grayscale

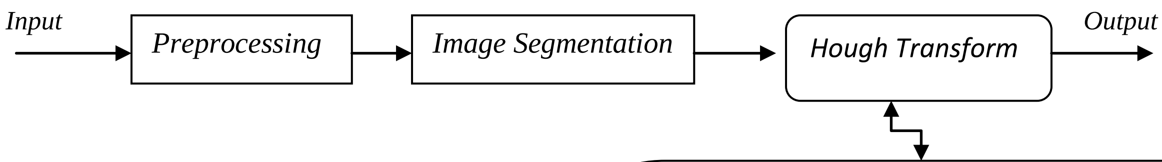
2. Create a 3D Hough array (accumulator) with the first two dimensions representing the coordinates of the circle origin and the third dimension represents the radii

3. Detect edges using the Canny edge detector. For each edge pixel (point), increment the corresponding elements in the Hough array.

4. Collect candidate circles, then delete similar circles.

5. Draw circles around coins.

Flow chart:



```
% Detect edge pixels using Canny edge detector.
Adjust the lower and/or
% upper thresholds to balance between the
performance and detection quality.
% For each edge pixel, increment the corresponding
elements in the Hough
% array. (Ex Ey) are the coordinates of edge pixels
and (Cy Cx R) are the
% centers and radii of the corresponding circles.
edgeim = edge(im, 'canny', [0.15 0.2]);
[Ey Ex] = find(edgeim);
[Cy Cx R] = find(Rmap);
for i = 1:length(Ex);
  Index = sub2ind(size(hough), Cy+Ey(i)-1,
Cx+Ex(i)-1, R-minR+1);
hough(Index) = hough(Index)+1;
End

% Collect candidate circles.
% Due to digitization, the number of detectable
edge pixels are about 90%
% of the calculated perimeter.
twoPi = 0.9*2*pi;
circles = zeros(0,4);     % Format: (x y r t)
for radius = minR:maxR    % Loop from minimal to
maximal radius
  slice = hough(:,:,radius-minR+1);  % Offset by
minR
  twoPiR = twoPi*radius;
  slice(slice<twoPiR*thresh) = 0;  % Clear pixel
count < 0.9*2*pi*R*thresh
  [y x count] = find(slice);
  circles = [circles; [x-maxR, y-maxR,
radius*ones(length(x),1), count/twoPiR]];
end
```

Limitations

Dependence of data quality:

The performance of the HT is highly dependent on the results from the edge detector. The input image must be carefully chosen for greater edge detection.
Using the HT on noisy images reduces the efficiency of the algorithm due to some edge points being missed because they were not defines as edges.

Object size/position constraints:

The CHT depends on a pre-define value of the circles' radius. A Program that entirely depends on user inputs is not practical for general use.
The position of the objects relative to the image plays an important role on how accurately they can be detected. For example, running the CHT on a two pictures of the same coins may hold different results depending on the distance between the coins and camera at the moment the picture was taken.

Conclusion

This paper introduced the concept of Hough Transform and Circular Hough Transform, and how they are used in object detection. Although it is the commonly preferred method for circular object detection, the HT in general has several limitations making it challenging to detect anything other than lines and circles. This is especially the case when more parameters are needed to describe shapes, this add more complexity thus decreasing the votes within the accumulator.

References

Just Kjeldgaard Pedersen, Simon. “Circular Hough Transform.” Aalborg University, Vision, Graphics, and Interactive Systems. November 2007. <http://www.cvmt.dk/education/teaching/e07/MED3/IP/Simon_Pedersen_CircularHoughTransform.pdf> (11/15/2010)

Epitropou, Victor. “Enhanced (Circular) Hough Transform”. DUTh Circle Detection. 2005-2006 <http://www.papamarkos.gr/uploaded-files/Papaparkos/demos/GHT/Help/EHT.htm> (11/15/2010)

MathWorks. “Hough Transform.” 2010. <http://www.mathworks.com/help/toolbox/images/ref/hough.html>

Kolas . “Hough transform.” Wikipedia. < http://en.wikipedia.org/wiki/File:Hough_transform_diagram.png> (11/18/2010)