International Journal of Computer Applications in Engineering Sciences
[VOL I, ISSUE II, JUNE 2011] [ISSN: 2231-4946]

Application of Hough Transform for finding Parametric Curves

Love Jain#, Devesh Mahor*

Institute of Engineering and Technology, J K Lakshmipat University, Jaipur
* *M.M.E.C., M. M. University, Mullana, Ambala*

#lovejain.eck@gmail.com
*mahor.devesh@gmail.com

***Abstract*— The Hough Transform has been used to characterize analytic features. It was first applied to the recognition of straight lines, and later extended to circles, ellipses and arbitrary shaped objects. The Hough Transform (HT) has long been recognized as a technique of almost unique promise for shape and motion analysis in images containing noisy, missing and extraneous data. Its main disadvantage is the computational and storage requirements increase as a power of the dimensionality of the curve. It is not difficult to implement Circular Hough Transform (CHT) algorithm on modern personal computer. The initial work is done on line detection and then this work is generalized for detection of circle in gray level images. The Hough transform gives better result in case of noisy images. In this paper, the algorithms for line and circle detection are proposed and the results are demonstrated.**

***Keywords*— Hough Transform, CHT, Binary image, image processing, image space.**

I. INTRODUCTION

Hough transform is a technique to detect straight line segments in a given binary image. To achieve this, for the first time in 1962, Hough proposed a method which provided a robust technique to identify the parameters of straight line edges in an image space [1].

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. It is used to detect straight lines within a two dimensional binary image.

By Hough Transform It is possible to find all kind of shapes that can be mathematical expressed, for instance lines, circles and ellipses, but only straight lines will be considered here.

II. HOUGH TRANSFORM FOR LINE

The problem of determining the location and orientation of straight lines in images arises in many diverse areas of image processing and computer vision,

Since straight lines occur in various natural and man-made objects and many complex objects can often be identified by their distinct combination of linear features [5].

The expression for any straight line in which ‘m’ is the slope and “ c” is where the line intersects the y-axis can be set as:

$$y = m x + c \quad (1)$$

Now if we reverse our variables and look instead at the values of (m, c) as a function of the image point coordinates (x, y), then Equation 1 becomes:

$$c = y - m x \quad (2)$$

Now, consider two pixels P1 and P2, which lie on the same line in the (x, y) space. For each pixel, we can represent all the possible lines through it by a single line in the (m, c) space. Thus a line in the (x, y) space that passes through both pixels must lie on the intersection of the two lines in the (m, c) space, which represent the two pixels. This means that all pixels which lie on the same line in the (x, y) space are represented by lines which all pass through a single point in the (m, c) space, see Fig. 1 and Fig. 2.

The problem by represent a line as a point in the (m, c) parameter-space is that both m and c goes toward infinity when the line becomes more and more vertical, and thereby the parameter space becomes infinity large.

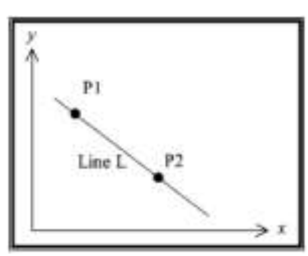

Fig. 1 Points on the same line [5]

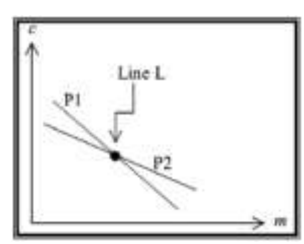

Fig. 2 The mapping of P1 and P2 from cartesian space to the (m, c) space [5]

The problem by represent a line as a point in the (m, c) parameter-space is that both m and c goes toward infinity when the line becomes more and more vertical, and thereby the parameter space becomes infinity large. Therefore it is desirable to find other expression of the line with some parameters that have limited boundaries. It is done by using an angle and a distance as parameters, instead of a slope and an intersection. The technique was improved by Duda et al in 1972 by replacing the slope-intercept equation by the normal equation:

$$\rho = x\cos\theta + y\sin\theta \qquad (3)$$

where ρ is the length of the normal from the line being detected to the origin and θ is the angle between the normal and the positive x-axis.

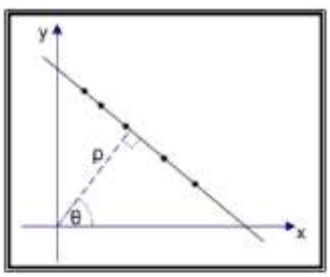

Fig. 3 The straight line [8]

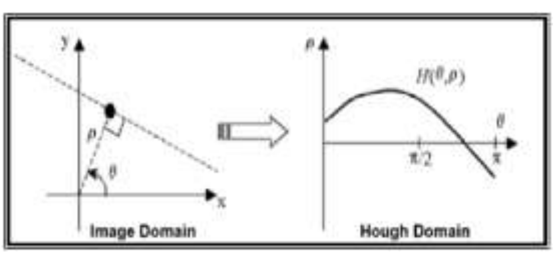

Fig. 4 ρ and θ representation of a straight line [8]

Contrary to when the parameters are m and c, the values that ρ and θ can have are limited to: $\theta \in [0,180]$ in degrees or $\theta \in [0, \pi]$ in radians, and $\rho \in [-D, D]$ where D is the diagonal of the image. A line can then be transformed into a single point in the parameter space with the parameters θ and ρ; this is also called the Hough space. If, instead of a line, having a pixel in an image with the position (x, y), infinity many lines can go through that single pixel. By using Equation 3 all these lines can be transformed into the Hough space which gives a sinusoidal curve that is unique for that pixel. Doing the same for another pixel, gives another curve that intersect the first curve in one point, in the Hough space. This point represents the line, in the image space, that goes through both pixels. This can be repeated for all the pixels on the edges, in an edge detected image.

A. Hough Algorithm For Line Detection

The Pseudo code of the Hough Transform for straight lines can be given as:

- load image
- find the edges in the image
- for all pixels in the image

  if the pixel (x, y) is an edge

  for all the theta angles

  calculate rho for the pixel (x, y) and the angle(theta)

  increment that position (rho, theta) in Accum
- show the Hough space
- find the highest value in the accumulator
- draw the line with the highest value in the input image

III. Hough Transform for Circle

The location of round objects is important in many areas of image analysis, but it is especially important in industrial applications such as automatic inspection and assembly. Since round picture objects form a special category of their own, efficient algorithms should be available for analyzing digital images containing them.

The circle is simpler to represent in parameter space, compared to the line, since the parameters of the circle can be directly transfer to the parameter space. The equation of a circle is

$$r^2 = (x-a)^2 + (y-b)^2 \qquad (4)$$

As it can be seen the circle got three parameters, r, a and b. Where a and b are the center of the circle in the x and y direction respectively and where r is the radius. The parametric representation of the circle is

$$x = a + r\cos\theta \qquad (5)$$

$$y = b + r\sin\theta \qquad (6)$$

Thus the parameter space for a circle will belong to R^3 whereas the line only belonged to R^2. As the number of parameters needed to describe the shape increases as well as the dimension of the parameter space R increases so do the complexity of the Hough transform. Therefore is the Hough transform in general only considered for simple shapes with parameters belonging to R^2 or at most R^3. In order to simplify the parametric representation of the circle, the radius can be held as a constant or limited to number of known radii.

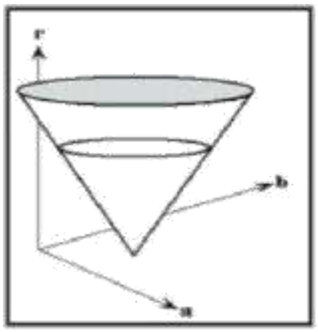

Fig. 5 The parameter space used for CHT [9]

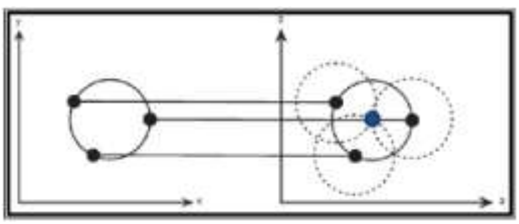

Fig. 6 A Circular Hough transform from the x, y-space (left) to the parameter space (right) [9]

A. Algorithm

The algorithm for Circular Hough Transformation can be summarized to:

1. Find edges
2. //HOUGH BEGIN
3. For each edge point
   Draw a circle with center in the edge point with radius r and increment all coordinates that the perimeter of the circle passes through in the accumulator.
4. Find one or several maxima in the accumulator
5. //HOUGH END
6. Map the found parameters (r, a, b) corresponding to the maxima back to the original image

IV. MATLAB RESULTS

The various images have been demonstrated here below. The Hough Transform is very effective in case of noisy images.

The results have been shown for Hough Transform for line and circle detection.

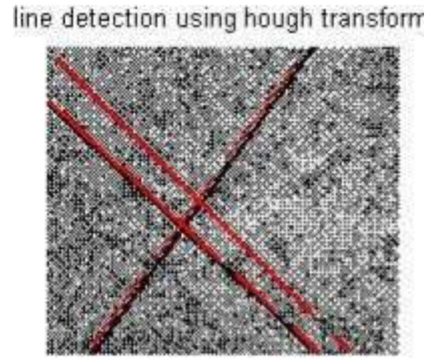

Fig. 7 Edge detection results for noisy image using Hough Transform

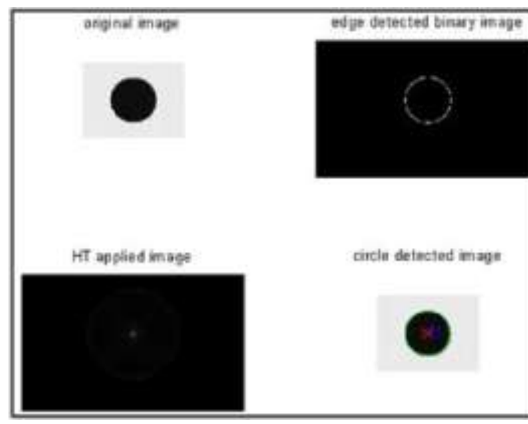

Fig. 8 Results for image im_1.jpg

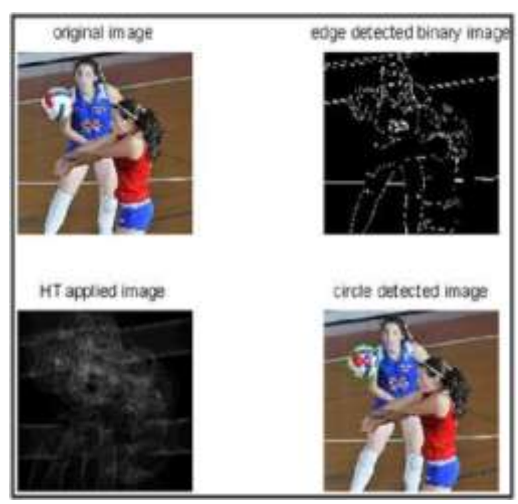

Fig. 9 Results for image im_2.jpg

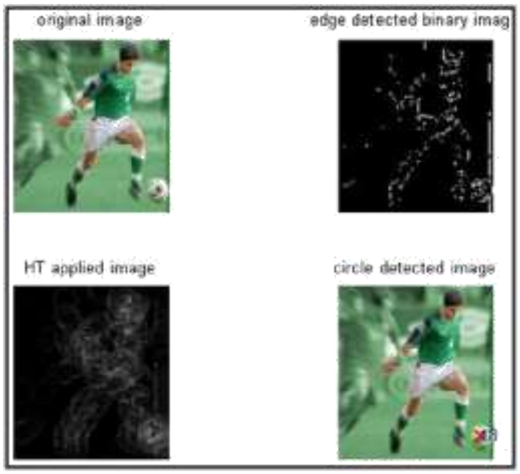

Fig. 10 Results for image im_3.jpg

TABLE 1
ANALYSIS FOR DIFFERENT IMAGES

Image	Image size	Radius range	(a,b)	r	Time(s)
im_1.jpg	189x301x3	28-32	(151,82)	31	5.7974
im_2.jpg	300x300x3	25-29	(69,85)	27	9.5962
im_3.jpg	295x249x3	16-20	(229,253)	17	6.678

V. CONCLUSIONS

The concept of Hough states that converting the image from one parameter space to another and finds the co-ordinates of maxima in another parameter space. The secondary parameter space co-ordinates gain mapped into primary space domain. For the case of circle primary domain is (x, y) and secondary is (r, a, b).

So, we will have to think about a such a robust algorithm that require less computations and then it can be implemented on Hardware for real time applications like replacing the football with sponsor's logo in live or recorded match.

REFERENCES

[1] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Communication of the ACM, Vol. 15, pp. 11–15.

[2] Carolyn Kimme, Dana Ballard, Jack Sklansky, "*Finding Circles by an Array of Accumulators*", Communication of the ACM, Vol. 18, Number 2, Feb 1975.

[3] Mohamed Roushdy, "*Detecting Coins with different Radii based on Hough Transform in Noisy and Deformed Image*", GVIP Journal, Vol. 7, Issue 1, Apr 2007, pp 25-29.

[4] E. Davis, "*Machine Vision: Theory, Algorithms, Practicalities*", Elsevier, ISBN 0-12-206093-8, 2005.

[5] Shih-Hsuan Chiu, Jiun-Jian Liaw, "*An Effective Voting Method for Circle Detection*", Conference on Computer Vision and Image Processing (CVGIP), 2003, pp 756-762.

[6] R. C. Gonzalez, R. E. Woods, "*Digital Image Processing*", Prentice Hall, 2007. ISBN 0-13-168728-X.

[7] http://www.physics.nyu.edu/grierlab/idl_html_help /h17.html, Dated: Oct 2009.

[8] Simon Just Kjeldgaard Pedersen, "*Circular Hough Transform*", Aalborg University, Vision, Graphics, and Interactive Systems, 2007.