International Journal Of Engineering Research And Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 13, Issue 11 (November 2017), PP.01-07

An Improved Hough Transform for Circle Detection

Nitin Bhatia
Department of Computer Science, DAV College, Jalandhar 144008, India
Corresponding Author:Nitin Bhatia

Abstract:-Digital image processing relies heavily on detection of various objects. Detection of geometric objects like lines and circles helps image understanding. In this paper, an improved version of Hough transform for circle detection in digital images is presented. Hierarchical Hough transform has been modified to achieve better computational efficiency. An important property of square root of integers has been examined and applied to work in the favour of algorithm. The experimental analysis shows considerable improvement in speed of the algorithm without compromising the accuracy.

Keywords:-Circle detection, Hough transform, hierarchical Hough transform, Automatic thresholding.

Date of Submission: 01-11-2017 Date of acceptance: 09-11-2017

I. INTRODUCTION

Curve detection in digital images has a variety of applications. Line and circle detection are two of the many different geometrical objects that can be detected from images. Circular objects can be located in a digital image with the help of circle detection [1]-[4]. Most popular technique to detect circles from images is Hough transform [5]. In literature, many variations of Hough transform are available. Fast Hough transform or hierarchical Hough transform is one such variation [6]. Randomised Hough transform uses randomised image pixels to vote for Hough space cells or bins instead of using all pixels, thus reducing time required [7]. Fuzzy Hough transform uses fuzzy logic to decide the membership of an image pixel to a Hough bin [8].

Apart from techniques based on Hough transform, other techniques have also been proposed [9]-[12]. Mingzhu and Huanrong [9] give a simple method of choosing a pixel and then finding two other pixels, one in horizontal and other in vertical direction, and fit a circle to these three points. The efficiency of the method depends on the selection of initial pixel. Chen et al. [10] propose an algorithm in which edge image is divided into several sub-images by the properties of circle, and then the parameters of circles are calculated using the point pairs chosen from the sub-images followed by a data merging method to process the parameters. Ayala-Ramirez et al. [11] present a circle detection method based on genetic algorithms. The method does not work well while detecting small circles from digital images. Cuevas et al. [12] propose a method based on Electro-magnetism optimization, which is a nature inspired technique to detect circles. The technique requires a number of parameters to be set manually to detect circles. We propose a method based on hierarchical Hough transform. The main feature of this method is its speed. The proposed method shows its robustness to the presence of noise as well. We conduct experiments on a large number of images most of which are natural images and the accuracy of the method is observed to be the same as that of conventional Hough transform. The method shows very high efficiency in terms of time requirements. The comparisons are made with techniques given by [12] and [13].

II. HOUGH TRANSFORM

Hough transform continues to be one of the landmark algorithm for curve detection in digital images. It works on the basis of voting in the Hough parameter space, a three dimensional space in case of circles. The local maxima is then selected as the bin which corresponds to circles in the image space.

A. Classical Hough Transform

Classical or standard Hough transform choses the simplistic approach of using equation of circle as $(x-h)^2-(y-k)^2-r^2=0$. Using all image points as (x, y) and voting for each bin as (h, k, r). This minimalistic approach is time consuming.

B. HierarchicalHoughTransform

The main drawback of circular Hough transform is its slow speed due to large number of computations. Speed inefficiency of CHT is dealt with using a hierarchical algorithm to work down the Hough space from coarser level to finer level [10], [11]. The Hough space used by the technique is dynamically quantized. It reduces the time taken as well as the memory needed to store the accumulated votes during the transform. The process starts with considering the whole HT space as a single accumulator cell which is assumed to attract number of votes equal to the number of edge pixels in the image space. Whenever, an accumulator cell is voted by number of edge pixels larger than a predefined threshold value, the accumulator cell is sliced into octants and each octant is tested further. The subdivision of Hough space is depicted by Fig. 1.

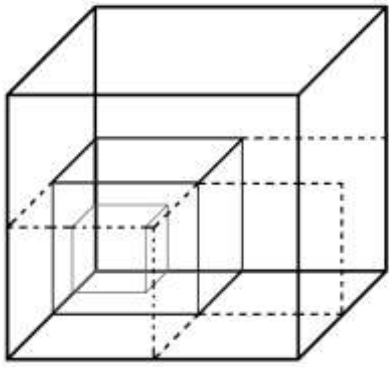

Fig. 1. Hierarchical subdivision of Hough space.

The octants failing to attract enough votes are barred from further contests. The process reaches the unit cell in $\log_2 L$ moves. Therefore, considering the number of edge pixels as E_I, the complexity reduces to $O(E_I \log_2 L)$. In the next section, we show that the proposed technique has an even further reduced time complexity.

III. THE PROPOSED METHOD

A. Elimination of Edge Pixels at Coarser Level

The boundary of a right circular cone centered at (h,k) with radius r is given as

$$f(h,k,r) = (h-x_0)^2 + (k-y_0)^2 - r^2 = 0 \quad (1)$$

At any level, the Hough space cubic cell can have lower left corner as (h_1,k_1,r_1) and upper right corner as (h_2,k_2,r_2). The boundary plans of the cubic cell are defined by

$$h-h_1=0 \qquad h-h_2=0$$
$$k-k_1=0 \qquad k-k_2=0$$
$$r-r_1=0 \qquad r-r_2=0$$

If the axis r is represented vertically, then the top boundary plane is $r-r_2=0$. This plane intersects with the cube, represented by Eq. 1, to produce a circle given in Eq. 2.

$$(h-x_0)^2 + (k-y_0)^2 - r_2^2 = 0 \quad (2)$$

This circle is centered at (x_0, y_0) and has radius r_2 as given by Fig. 2. This circle cannot intersect the square or rectangle that represents the top of the cubic Hough space cell if it lies above, below, to the right of or to the left of the top, bottom, right or left boundaries, respectively.

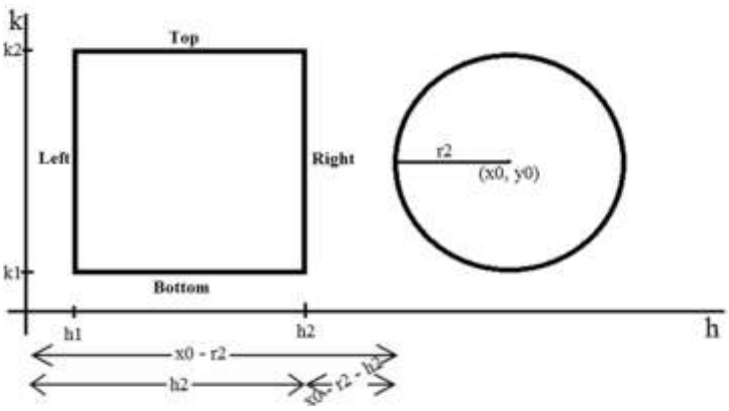

Fig. 2. Description of Step I

Therefore, at a coarser level, an edge pixel (x_0, y_0) does not vote for a cubic cell defined by $[(h_1, k_1, r_1) - (h_2, k_2, r_2)]$ if the following conditions are not met:

- Circle lies to the right of the right boundary: $x_0 - r_2 - h_2 > 0$
- Circle lies to the left of the left boundary: $x_0 + r_2 - h_1 < 0$
- Circle lies above the top boundary: $y_0 - r_2 - k_2 > 0$
- Circle lies below the bottom boundary: $y_0 + r_2 - k_1 < 0$

Elimination of edge pixels at a coarser level saves substantial amount of time. For example, in one of the sample image, the time saved by this step is 40%.

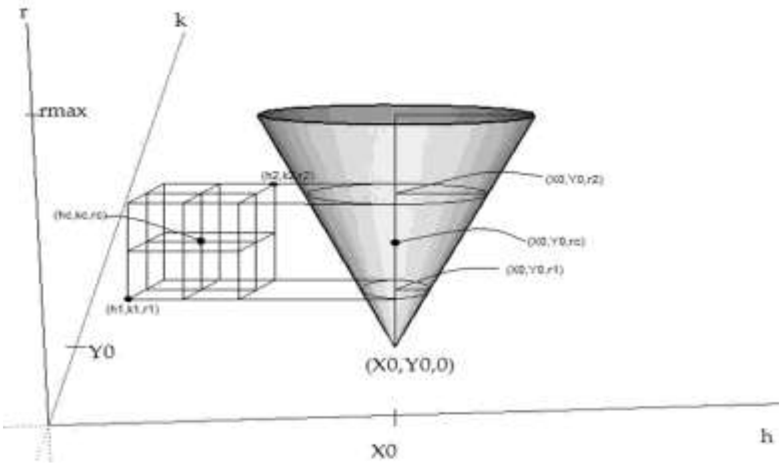

Fig. 3. A view of the Hough space for circle parameters.

B. Elimination of Edge Pixels at a Finer Level

The edge pixels that pass the decision test, as described by previous step, are further tested at a finer level. Instead of testing all of eight vertices of the Hough space cubic cell only the center of the cell is tested against the right circular cone generated by the edge pixel. A Hough space cubic cell defined by opposite vertices $(h_1, k_1, r_1) - (h_2, k_2, r_2)$ has its center point given by (h_c, k_c, r_c) where $h_c = (h_1 + h_2)/2$, and so on. Fig. 3 shows a Hough space cell against a right circular cone generated by one edge pixel (x_0, y_0). The vertical line representing the centre axis of cone centered at $(x_0, y_0, 0)$ is given by the equations $h = x_0, k = y_0$. The distance of this line from the cell centre (h, k) is

$$d = \sqrt{(h_c - x_0)^2 + (k_c - y_0)^2} \quad (3)$$

The width of the cubic cell is $w = \dfrac{\sqrt{2}N}{2^{level+1}}$.

Circumradius of the circumsphere to the Hough cubic cell is $w' = \dfrac{\sqrt{2}N}{2^{level+1}} \times \dfrac{1}{\sqrt{2}} = \dfrac{N}{2^{level+1}}$, level is the current level of Hough space precision.

Circumsphere lies entirely outside the cone if $d + w' \geq r_1$

Circumsphere lies entirely inside the cone if $d - w' \leq r_2$

Equation (3) involves the square root operation which is time consuming Instead of using a relatively slow inbuilt sqrt(N) function, we pre-compute and store the rounded off square root values in an array. Let us examine an important property of rounded off square root for integers.

C. Square Root Property of Integers

Theorem: An integer k is rounded off square root for all the integers in the interval (k2-k, k2+k].

Proof: Let k be the rounded off square root value of integer N then,

$$k \le \sqrt{N} + 0.5 < k + 1$$

$$\Rightarrow k - 0.5 \le \sqrt{N} < k + 0.5$$

$$\Rightarrow k^2 - k + 0.25 \le N < k^2 + k + 0.25$$

$$\Rightarrow k(k-1) + 0.25 \le N < k(k+1) + 0.25$$

$$\Rightarrow k(k-1) + 1 \le N < k(k+1)$$

$$\Rightarrow k^2 - k + 1 \le N < k^2 + k$$

$$\Rightarrow k^2 - k < N < k^2 + k$$

□

The algorithm used to pre-compute the rounded off square root values is given here. The tradeoff for using the pre-computed square root values is the additional space requirement of R words.

The Algorithm: Rounded off square root computation

```
Set Maxradius to √2×N , Square_root(0) to 0 and counter to 1
For I = 1 to Maxradius
        For J = I to 2I
                Set Square_root(counter) to I
                Increment counter by 1
end for
end for
```

The algorithm stores rounded off square root values for integers from 0 upto Maxradius.

Table 1: Rounded off square root values.

N	Square root value ($\sqrt{N}$)	Rounded off square root value int($\sqrt{N}$ + 0.5)		
1	1	1	}	1 occurs 1 X 2 = 2 times
2	1.414	1		
3	1.732	2	}	2 occurs 2 X 2 = 4 times
4	2	2		
5	2.236	2		
6	2.449	2		
7	2.645	3	}	3 occurs 3 X 2 = 6 times
. . .	. . .	. . .		
19	4.3589	4	}	4 occurs 4 X 2 = 8 times
20	4.472	4		
.	.	.		

D. Automatic Thresholding for Voting

Efficiency and accuracy of the Hough transform technique depends on the selection of a threshold value. In case of HHT, a Hough space cell is further divided into octants and the octants which attract votes above a pre-defined threshold value are further investigated. The threshold value is given such that the digital circles to be detected must have pixels larger than or equal to the threshold. Images have circles varying in size. Having a common threshold value does not suffice. Ideally, the threshold value must be radius dependant. For a radius dependant threshold, considering an edge image after applying Sobel's operator must have a threshold

$4\sqrt{2}r-8$, where r is the radius of the circle. For the edge pixels after pixel reduction, the threshold value must be set to $8r(1-\frac{1}{\sqrt{2}})$ which is explained as follows.

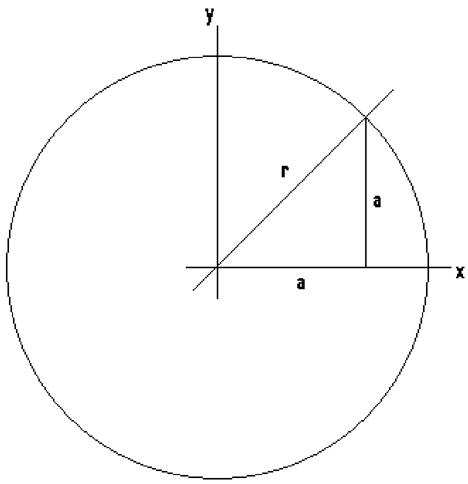

Fig. 4. Calculation of radius dependant threshold

A digital circle with radius r has 8-way symmetry and in an octant, the maximum number of pixels will be $a=r/\sqrt{2}$. Thus, the complete circle will have $8r/\sqrt{2}$ pixels, but 8 pixels will repeat in the process of computation. Thus, the maximum number of pixels a digital circle can have is $T=4\sqrt{2}r-8$. The actual threshold however, will be much less than $4\sqrt{2}r-8$ due to the edge pixel reduction process. This is explained as follows. We compute the number of pixels that contribute towards the actual threshold, say, T' in an octant. Starting from $y=r$ to $y=r/\sqrt{2}$, which belongs to the octant 1, (as shown in Fig. 4), only $r-r/\sqrt{2}$ pixels contribute towards T'. Thus, the total value of T' for the complete circle will be $8\left(\left(r-r/\sqrt{2}\right)-1\right)$ after reducing the repeating pixels on the circle during the scanning process. The percentage reduction in the threshold is $T_p=\frac{T-T'}{T}\times 100$. which is $\frac{200r\left(\sqrt{2}-1\right)}{\sqrt{2}r-2}\%$. Typically, for $r=10,20,30,40,50$ the values of $T_p=68.22746\%, 63.03596\%, 61.47668\%, 60.72562\%, 60.28372\%$, respectively.

We compare the proposed method to one of the work by Cuevas et al. [12] theoretically and to Chiu and Liaw [13] by using experiment analysis. The proposed method is automatic as it automatically calculates the value of threshold. The method given in [12] requires manual adjustment of various parameters for accurate detection of circles. Moreover, the technique presented is based on an algorithm that has complexity $O(E_I^2 C)$ where E_I is the number of edge pixels and C is a predefined constant value. Therefore, the actual complexity of the algorithm is $O(E_I^2)$. It also suffers from a drawback that it requires large number of floating point operations and also calls random function to generate a set of three elements to be used as initial population for detection task. In contrast, the proposed technique has a low complexity with $O(E_I \log_2 L)$. For Hough space with dimensions 1024x1024x1024, the number of times the algorithm needs to iterate is $10E_I$. Apart from that, the pixel reduction technique reduces the parameter E_I to a great extent. Our experiments show that E_I is reduced to an average 12% of original E_I, hence reducing the number of iterations to an extremely low count.

IV. EXPERIMENTAL ANALYSIS

A. Execution Efficiency

An extensive experimentation using 100 different images (natural as well as synthetic) has been performed (results are shown for 25 only). Table 2 shows the execution time comparison of proposed technique with technique given by [13].

Table 2: Execution time comparison.

Image Name	**Execution Time (sec.)**		**[13] vs. HHT %**
	Circle detection by [13]	**Proposed Hierarchical HT**	
Img1	0.0865	0.063	27.1466
Img2	0.0828	0.078	5.7425
Img3	1.5780	1.375	12.8623
Img4	0.2143	0.062	71.0733
Img5	1.0078	0.734	27.1674
Img6	0.5867	0.313	46.6518
Img7	0.1296	0.098	24.3769
Img8	0.5230	0.374	28.4873
Img9	0.4056	0.387	4.5830
Img10	0.1054	0.068	35.4618
Img11	0.1053	0.088	16.4245
Img12	0.1598	0.075	53.0790
Img13	0.1288	0.091	29.3374
Img14	1.7950	0.2313	10.5070
Img15	0.3726	0.1033	37.9267
Img16	0.1816		43.1148
Img17	0.7781	0.3138	**59.6723**
Img18	0.3611	0.2186	39.4684
Img19	0.2402	0.1155	51.9198
Img20	0.2980	0.2091	29.8406
Img21	0.4211	0.3323	21.0819
Img22	0.8739	0.7022	19.6505
Img23	1.1989	1.0173	15.1488
Img24	0.8327	0.5647	32.1820
Img25	1.9383	0.7897	59.2583

The proposed technique shows 59.67% improvement of speed.

B. Robustness to Noise

Robustness to noise is a prime requirement for the circle detection methods. The proposed method works well even in the presence of noise. We added varying amount and type of noise to the images containing circular objects. The experiments are conducted using noisy images and results are presented. Fig. 5 shows an image containing circles along with 3% and 5% added noise and their corresponding results.

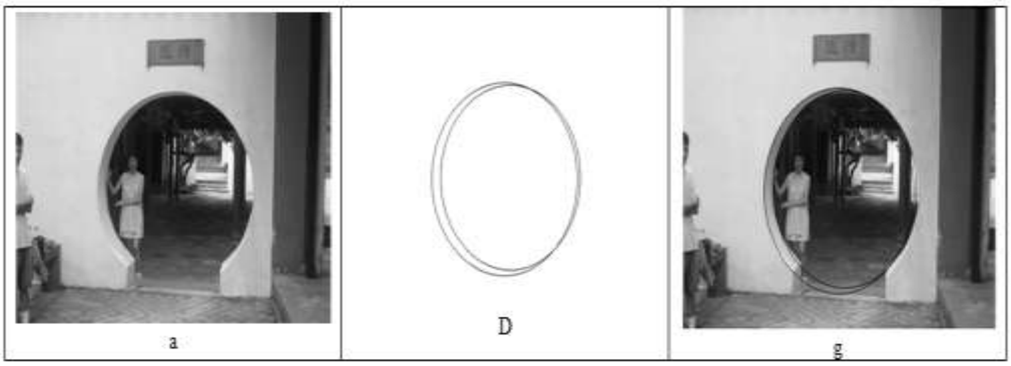

Fig. 5. (a) A synthetic image containing two circles. (b) Added 3% salt noise. (c) Added 5% salt noise. (d) Result of Proposed method on Fig. 5a. (e) Result of Proposed method on Fig. 5b. (f) Result of Proposed method on Fig. 5c. (g) Fig. 5d superimposed on Fig. 5a. (h) Fig. 5e superimposed on Fig. 5b. (i) Fig. 5f superimposed on Fig. 5c.

V. CONCLUSIONS

The proposed technique uses some very peculiar properties of hierarchical Hough transform and suggests improvement over the existing technique, thereby achieving a significant improvement in execution time taken. The improved version is showing a maximum of 59.67% of time improvement. The proposed method thus proves itself a very important technique in circle detection. In future, the similar technique can also be applied to other curve detection methods to achieve improvements.

REFERENCES

[1]. H. Yazid, H. Yazid, M. Harun, S. Mohd, A.A. Mohamed, M. Rizon, S. Sayuti, Circular discontinuities detection in welded joints using Circular Hough Transform, NDT & E International 40 (2007) 594-601.

[2]. C.-P. Chang, J.-C. Lee, Y. Su, P.S. Huang, T.-M. Tu, Using empirical mode decomposition for iris recognition, Conput. Stand. Interfaces 31 (4) (2009) 729-739.

[3]. N.V. Huan, H. Kim, A novel circle detection method for iris segmentation, in: Proceedings of Congress on Image and Signal Processing, Sanya, China, 2008, pp: 620-624.

[4]. N. Bhatia, M. Chhabra, Improved Hough transform for fast Iris detection, in: Proceedings of 2nd International Conference on Signal Processing Systems, Dalian, China, 2010, pp: V1-172 - V1-176.

[5]. R. Duda, P. Hart, Use of Hough transform to detect lines and curves in pictures, Communications of the ACM 15 (1975) 11-15.

[6]. N. Guil, E.L. Zapata, Lower order circle and ellipse Hough transform, Pattern Recognition, 30 (10) (1997) 1729-1744.

[7]. C. Xing, J. Wang, An improved method for circle detection", ICIRA 2008, Part I, LNAI 5314, pp. 463–468, 2008.

[8]. J.H. Han, L.T. Koczy, T. Poston, Fuzzy Hough transform. In: Proceedings of the 2nd International Conference on Fuzzy Systems, 1993, pp. 803–808.

[9]. Z. Mingzhu, C. Huanrong, A new method of circle's center and radius detection in image processing, in: Proceedings of the IEEE Int. Conf. on Automation and Logistics, China, 2008. pp. 2239-2242.

[10]. G. Chen, J. Ji, L. Sun, A novel circle detector based on sub-image point pairs, in: Proceedings of the IEEE Int. Conf. on Information and Automation, China, 2008. pp. 661-666.

[11]. V. Ayala-Ramirez, C.H.G.-Capulin, A.P.-Garcia, R.E.S.-Yanez, Circle detection on images using genetic algorithms, Pattern Recognition Letters 27 (2006), 652-657.

[12]. E. Cuevas, D. Oliva, D. Zaldivar, M.P.-Cisnero, H. Sossa, Circle detection using electro-magnetism optimization, Inform. Sci. (2011), doi:10.1016/j.ins.2010.12.024.

[13]. S.-H. Chiu, J.-J. Liaw, An effective voting method for circle detection, Pattern Recognition Letters 26, (2005) 121-133.

Nitin Bhatia. "An Improved Hough Transform for Circle Detection." International Journal Of Engineering Research And Development , vol. 13, no. 11, 2017, pp. 01–07.