

# A Comparative Study on the Center-based Iterative Hough Transform

Joshua Park<sup>1</sup>

<sup>1</sup>Yongsan International School of Seoul, South Korea

## ABSTRACT

Circle detection is one of the most critical aspects of computer vision and has been widely studied and developed in a variety of ways. The Center-based Iterative Hough Transform (CBIHT) is a method for unassisted multiple circle detection, based upon iterative uses of a center-based voting process to determine the circle's center coordinate. This paper gives a thorough analysis of the CBIHT as well as a comparison with the Standard Hough Transform (SHT) and its well-known variants including the Generalized Hough Transform (GHT) and the Adaptive Hough Transform (AHT). When applied to synthetic and real-life circular images, our accuracy and performance comparison studies show that (i) the CBIHT is more computationally efficient than the SHT's brute-force algorithm; (ii) the CBIHT's center-based voting method has greater resilience to noise than the GHT and AHT's gradient information method; and (iii) the CBIHT's iterative process provides an adaptability and speed in unassisted multiple circle detection similar to that of the AHT; (iv) yet, the CBIHT requires no parameters for circle detection unlike the GHT and the AHT. All in all, a comparison with other methods highlights the aforementioned merit of the CBIHT, proving the CBIHT to be an excellent choice in detecting the circles with noise in real-life images.

## Introduction

Circle detection is one of the most critical aspects of computer vision and has been widely studied and developed in a variety of ways. Initially developed by Duda and Hart (1972) for detection of lines and curves, the Hough Transform is a particular method of circle detection that is especially well known for its resistance to image corruption and noise as well as its ability to detect figures such as lines, circles, and ellipses. This method was named after Peter Hough's patent (Hough, 1962), hence the name "Hough," and became popularized in the field of computer vision through Dana H. Ballard (1981)'s paper titled "Generalizing the Hough transform to detect arbitrary shapes."

The Standard Hough Transform (SHT) (e.g., Kimme *et al.*, 1975) is a brute-force algorithm that is slow in computation yet is extremely accurate. Due to its immense potential, the Hough Transform has been expanded by numerous authors. For example, Kimme *et al.* (1975) in their Generalized Hough Transform (GHT), use the gradient direction of each of the edge points in the images in order to efficiently search for circle's center coordinates. Illingworth and Kittler (1987) in their Adaptive Hough Transform (AHT) develop the gradient direction method further through iterative approach, searching from the lowest resolution to a finer one. Many other authors, including Kim and Kim (2001) in their two-step circle detection algorithm, have also improved the Hough Transform in unique ways. Currently, the Hough Transform is being used in a variety of situations such as retinal detection (Ünver *et al.*, 2019) and medical imaging (Wohlfart, 2003).

The Center-based Iterative Hough Transform (CBIHT), recently introduced by Park (2019), iteratively uses a center-based voting process to determine the circle's center coordinate. Consequently, the CBIHT is able to achieve a speedy, parameter-free and multiple circle detection that is resistant to noise. The goal of this paper is to perform a thorough analysis of the CBIHT as well as a comparison with the SHT and its well-known variants including the GHT and the AHT, in order to highlight the CBIHT's capabilities.

In this paper, we first present a closer explanation of the particular methods utilized by the Standard Hough Transform and its variants followed by a detailed explanation of the method used in the CBIHT. Secondly, we provide empirical analysis of the accuracy and performances of the various methods through the usage of synthetic and real-life images. Finally, we conclude this paper with a summary and closing remarks.

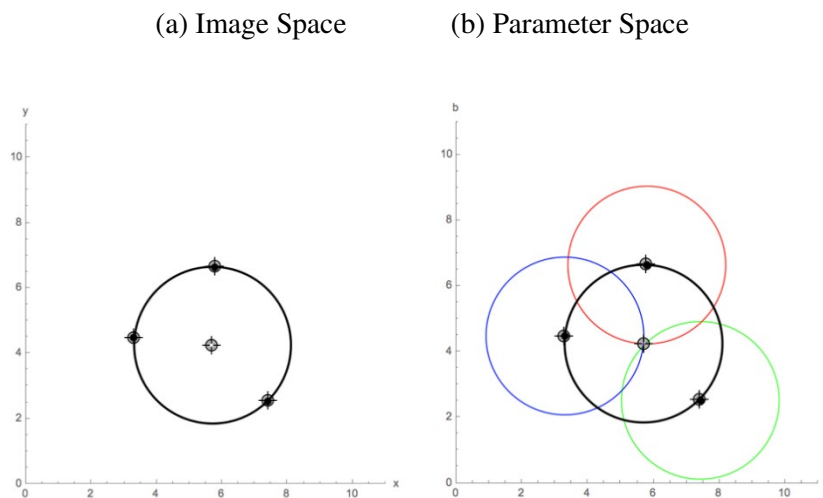
## Methodologies

### The Standard Hough Transform and Its Variants

The Standard Hough Transform (SHT) is a brute-force algorithm as it requires the calculation of every possible circle that passes through each of the edge points using the following parameterization

$$(x - a)^2 + (y - b)^2 = r^2. \quad (1)$$

This method calculates the distance between every pixel in the image and every image point and stores it as a possible circle  $((a, b, r))$ , where  $((a, b))$  stands for the coordinates of the pixel from which the distance is calculated and  $r$  stands for the distance calculated, or the radius of the imagined circle. After plotting these possible circles in parameter space, where the  $x$ ,  $y$ , and  $z$  axes represent  $a$ ,  $b$ , and  $r$  respectively, the value with the highest overlap or “vote” is determined to be the circle in the image. Fig. 1 briefly illustrates this voting process using two comparative images. Fig. 1 (a) displays three dots in image space through which the true circle is shown in black. Fig. 1 (b) shows the same dots in parameter space, drawing out possible centers of circles with a fixed radius in red, blue, and green. The intersection of all three colored circles represents the center of the circle that passes through the three points, as confirmed by the black circle from Fig. 1 (a). The center of the true circle, in this case, receives a vote of 3, making it the value with the highest overlap or vote.



**Figure 1.** Illustration of Voting Process of the Standard Hough Transform in Circles

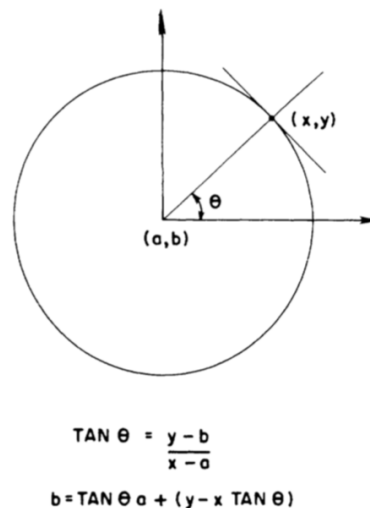
The process of computing all the possible circles takes up a substantial amount of computational power and, as such, the SHT requires an immense amount of time to locate a circle. The already slow speed is further elongated by its voting process which requires three separate parameters  $a$ ,  $b$ , and  $r$ .

The Generalized Hough Transform (GHT) reduces the computational complexity by looking at only one direction (the angle towards the potential center) for each image point. This is done by first locating the circle’s edge

and using the gradient information, *via* the Sobel Operator, to locate a single direction towards the circle's center (Kimme *et al.*, 1975). Within the Sobel Operator, the partial derivative of the edge point is taken, the normal line from the derivative is found, and the angle from the point and the center is shown. Using the  $(x, y)$  coordinates of each image point, along with the gradient direction ( $\theta$ ), the projected center of the circle  $(a, b)$  can be calculated using the following equations.

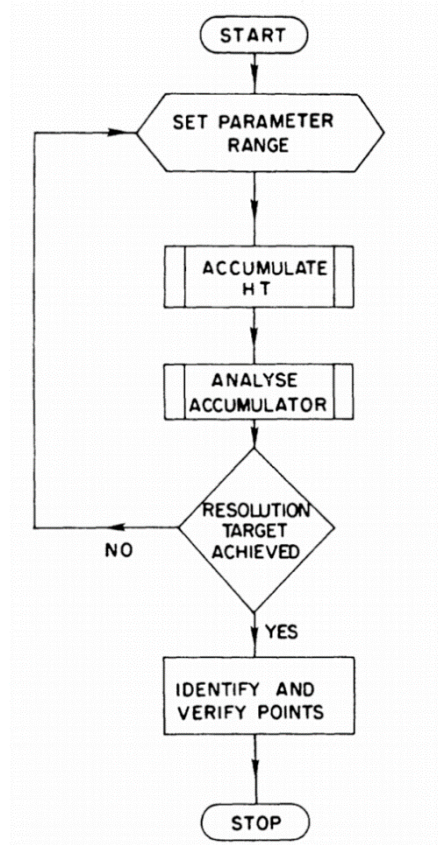
$$\begin{aligned} a &= x - r \cos\theta \\ b &= y - r \sin\theta \end{aligned} \quad (2)$$

For the variable  $r$ , a number of potential radii, specified with the parameters for the minimum radius  $r_{min}$  and the maximum radius  $r_{max}$ , in the calculated gradient direction ( $\theta$ ) are taken into consideration. Fig. 2 shows a graphical illustration of the method used by the GHT in computing the center coordinates by equation (2) using the gradient direction computed at each pixel  $(x, y)$ . As a result of taking a narrower set of potential centers into consideration, the required number of computations and subsequently the computational time is able to be drastically reduced through the GHT.



**Figure 2.** Illustration of Gradient Direction used in Generalized Hough Transform. Reprinted from “The Adaptive Hough Transform”, by J. Illingworth and J. Littler, 1987, *IEEE Trans on Pattern Analysis and Machine Intelligence*, PAMI-9, p. 692. Copyright 1987 by IEEE.

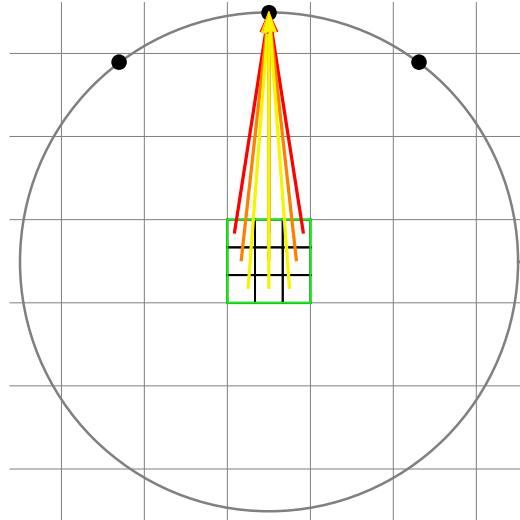
The Adaptive Hough Transformation (AHT) further increases the computational efficiency by introducing the idea of (i) an iterative coarsen-to-fine search as well as (ii) a sequential search for multiple circles by removing the circle found prior to the current search (Illingworth and Kittler, 1987). It is also able to retain the proficiency and adaptiveness of the GHT through utilizing the gradient information employed by the GHT. The specific approach to circle detection taken by the AHT is illustrated in Fig.3. Since it uses the same gradient information, the AHT has the equivalent benefits and drawbacks (as will be discussed in a later section) as the GHT has, with the additional benefit of an increased speed.



**Figure 3.** The Adaptive Hough Transformation Algorithm. Reprinted from “The Adaptive Hough Transform”, by J. Illingworth and J. Littler, 1987, *IEEE Trans on Pattern Analysis and Machine Intelligence, PAMI-9*, p. 692. Copyright 1987 by IEEE.

### The CBIHT Method

The Center-based Iterative Hough Transform, as the name suggests, utilizes a search approach based on refining the position of the located circle in order to pinpoint an accurate position. In this method, the potential candidates for the circle’s center are reduced through an iterative process in which the resolution of the image is increased incrementally from a low-resolution image to its full size (1920 by 1080 pixels for a full HD image) to enable a quick, rough finding of the circle’s center followed by a more precise evaluation. An empirical study by Park (2019) shows that 6-8 iterations seem to be the optimal choice to maximize the speed of the algorithm. At each iteration, the circle’s center is located in the image through a voting method based only on the circle’s radius (using equation (1)), resulting in a successive refinement of the found center. This voting process is illustrated in Fig. 4. After locating the center, all the coordinates in a small square box (of 11 by 11 pixels, *per se*) around the center are considered to be potential candidates for the circle in the finer-resolution image to account for possible error, and each of these points are scaled up to the subsequent image’s resolution. Then, through a calculation involving only the scaled-up points, the candidate closest to the circle’s center is then determined, and the process is repeated.



**Figure 4.** Center-based Voting Process by CBIHT

By only considering a limited number of points to be potential centers, the required number of computations decreases significantly. Paired with the iterative process that disregards all areas known not to contain the center, the CBIHT is able to achieve both the accuracy of the SHT and the speed of the GHT and/or AHT.

The CBIHT includes the same benefits of the AHT with the usage of the iterative and adaptive search process but avoids the drawback of the gradient information (to be discussed in the subsequent section) through a unique center-based voting method.

## Accuracy and Performance Analyses

This section presents experimental tests and empirical data highlighting the contrasting characteristics of the Hough Transform variations and evaluating the performance of the Center-Based Iterative Hough Transform. The experiments, implemented in the language Mathematica (Wolfram, 2003) and OpenCV-Python (O. Team, 2018), were performed on a MacBook Pro with a 2.3 GHz Intel Core i5 processor and a 16 GB 2133 MHz memory.

### Accuracy Comparison

To begin with, the SHT boasts impeccable accuracy due to its aforementioned brute-force approach. As such, it is noted for its resistance to noise and corrupted images and is able to detect the circle, or circles, within almost any given image provided that the computation is feasible. The GHT, as well as the AHT, utilize the gradient information that may potentially be riskier in certain situations or images. The presence of noise or corruption within an image is likely to interfere with the Sobel Operator (Kimme *et al.*, 1975), resulting in a lack of consistency in the angles found by the gradient information (*i.e.* a false direction that does not face the center of the true circle). These erroneous directions will then cause the wrong center candidates to be tallied, ultimately leading to an incorrect circle being located, as pointed by Antolovic (2018) that “Calculating the edge direction is typically rather inaccurate and introduces amplified errors in the location of the circle’s center.” This same issue also appears when attempting an image with multiple or overlapping circles with the GHT or AHT.

On the contrary, the CBIHT is able to successfully retain the complete accuracy of the SHT. As mentioned earlier, the CBIHT uses a method analogous to that of the SHT in which the same voting procedure is carried out with only the most likely candidates for the center; thus, the CBIHT is capable of accounting for images with noise far

better than the GHT/AHT. As the process inherently accounts for potential errors in the previous iteration, the CBIHT is able to perform with the same perks as the SHT while accounting for speed, which will be discussed shortly.

Fig. 5 compares the results of various methods when detecting circles with varying degrees of noise in order to illustrate the robustness, or lack thereof, that each method has in regard to noise. The three different methods are the SHT, GHT, and CBIHT, with the CBHT, a single-step variant of the CBIHT for a better comparison of time.

In detecting an ideal image (Image 1), that is a computer-generated image of a perfectly round circle, all the methods are successful in finding the correct center. All the methods are also successful in detecting the correct circle when computing a synthetic image with noise, a broken-up version of the ideal image with a few added noise dots (Image 2). However, when a broken-up, hand-drawn circle (Image 3) is tested, all of the methods but the GHT are able to locate the correct circle. This failure of the GHT results from the irregular hand-drawn circle not forming a true circle. As a result, the GHT's directional probing is unable to result in an accumulation of the correct center. The susceptibility of the GHT to noise is clear when put in contrast to the robustness of the CBIHT and the brute-force robustness of the SHT.

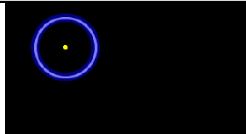
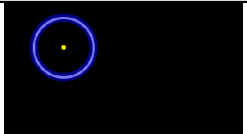
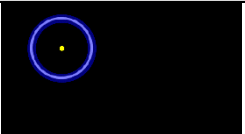
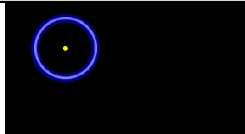

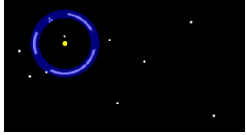






	SHT	GHT	CBHT	CBIHT
Image 1: Ideal im- age				
Image 2: Synthetic image with noise				
Image 3: Hand- drawn ir- regular image				

Figure 5. Circle Detection Results for Images by Various Methods

### Complexity and Performance Analysis

The superiority of the CBIHT can be further demonstrated through an analysis of the complexities of the various algorithms. The Big  $O$  Notation, a method commonly used to classify algorithms on their computation speed or complexity in relation to the input size, can be used to compare the speeds of the SHT, AHT, and CBIHT.

The complexity of the SHT can be expressed as  $O(nrdm^2)$ , which represents the SHT's brute-force approach of calculating all  $n$  edge pixels with  $r$  radii in every  $d$  directions for each of the possible center coordinates  $m^2$ , where  $m^2$  represents the number of center coordinates considered. The GHT further reduces this complexity, as it limits the direction to one (towards the center), resulting in a complexity of  $O(nrm^2)$ . The iterative process utilized in the AHT and CBIHT can be determined using a geometric sequence leading to the following equation

$$\frac{bn}{R^T} = b \Leftrightarrow T = \frac{\log n}{\log R} \propto \log n, \quad (3)$$

where  $T$  is the total number of iterations performed,  $n$  denotes the total expansion of the image in pixels, and  $R$  is the ratio by which the image size is incrementally increased from the smallest base size  $b$  to the largest base size  $bn$ .

Equation (3) can be reduced to the number of edge pixels involved in the calculation as shown, resulting in the entire process being  $O(\log n \cdot rm^2)$ . Finally, the CBIHT is able to simplify the calculation even more through fixing the number of points considered as potential centers, reducing the  $m^2$  into  $m'^2$  where  $m' \ll m$ . As such, the CBIHT can be notated by  $O(\log n \cdot rm'^2)$ .

**Table 1.** Computational Complexity by Various Method

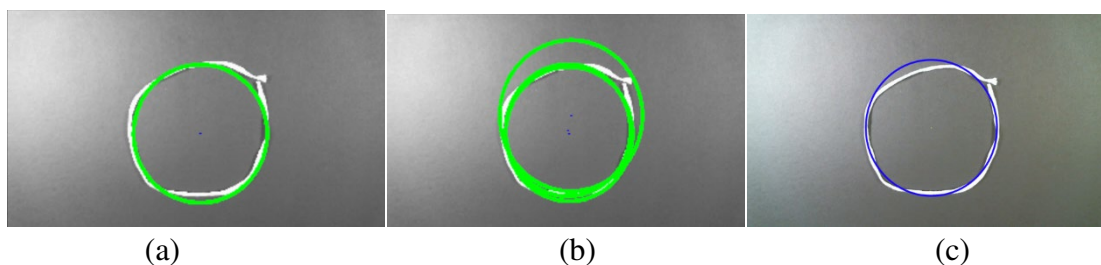
Complexity		SHT	GHT	CBHT	CBIHT
		$O(rdm^2n)$	$O(rm^2n)$	$O(rm'^2)$	$O(rm'^2 \log n)$
Time (s)	Image 1	280.25	37.37	0.17	1.59
	Image 2	223.50	37.74	0.15	1.50
	Image 3	305.59	15.11	0.20	3.02

Table 1 presents a comparison of the efficiency of these algorithms, in terms of both their growth rates and the actual computational times, where the empirical time samples are obtained from the images shown in Fig. 5. As seen in their complexity analyses of the different methods, the growth rate of the computational speeds of the various algorithms is greatest in the SHT and decreases in the order of SHT, GHT and CBHT, and CBIHT. For image 1, the SHT took 280.25 seconds, while the GHT took only 37.37 seconds. The CBHT and CBIHT reduced the time even further, taking 0.17 seconds and 1.59 seconds respectively. A similar trend can be observed for images 2 and 3. From these comparisons, it is clear that the CBHT and CBIHT are much faster than the other algorithms.

While the individual times each algorithm takes for the different images varies, a predominant trend can be observed. As expected, the SHT took the longest time due to its brute-force approach, rendering it hardly applicable in real life. The GHT is able to perform the computations in a shorter amount of time, with its more efficient gradient information method, and the CBHT and CBIHT reduced the time even further. As such, the comparison of computational speeds demonstrates that the CBHT and CBIHT are much faster than the other algorithms.

It is important to note that the CBHT, a single-step version of the CBIHT, has a few significant differences. For a fairer comparison with the GHT, the CBHT was given a range of coordinates that contained the coordinates of the correct circle in advance much like the GHT requires a predicted range of the radii of the circle. As such, the CBHT is able to perform the computations at a much faster speed than the CBIHT, which employs an iterative search procedure and does not rely on any prior information.

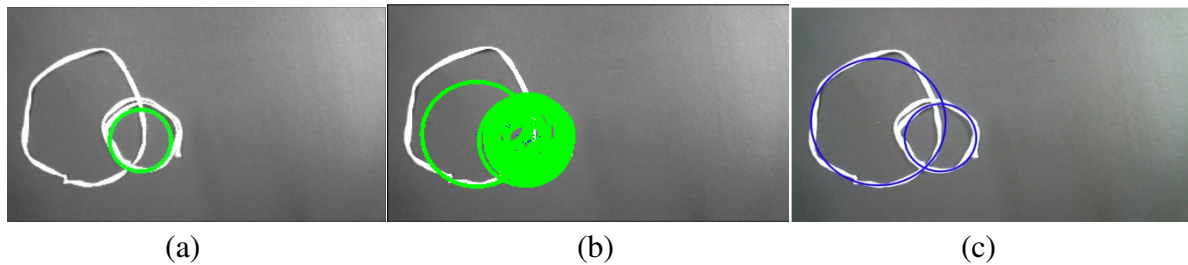
Fig. 6 compares the results of OpenCV's Hough Transform operator when given the correct/ideal parameters and incorrect parameters with the results of the parameter-free CBIHT. The success of OpenCV in Fig. 6 (a) compared to its failure in Fig. 6 (b) underscores the imperative nature of precise pre-determined parameters to the proper function of the GHT. This pattern can be commonly observed in any other image, where the alteration of a single parameter by even 1 causes an error in the detection of the circle. The CBIHT's contrasting lack of necessary parameters (i.e. a data-driven detection) contributes to its superior adaptiveness and potential for real-world applications as seen in Fig. 6 (c).



**Fig. 6.** Comparisons of Circle Detection by OpenCV's GHT versus the CBIHT: (a) Success of OpenCV with correct parameters; (b) Failure of OpenCV with incorrect parameters; (c) Success of CBIHT with no parameters

## Multiple/Overlapping Circle Detection

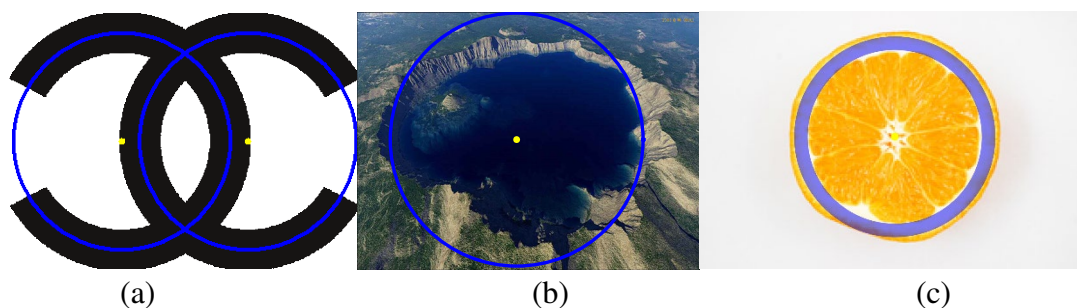
Fig. 7 compares the results of overlapping circle detection with OpenCV's GHT and CBIHT. As seen in Fig. 7 (a) and (b), the GHT in OpenCV's Hough Transform is unable to properly identify both circles simultaneously despite using the most optimal pre-determined parameters through trial and error. The CBIHT, on the other hand, is able to effectively locate both circles as seen in Fig. 7 (c).



**Figure 7.** Comparisons of Overlapping Circle Detection by the CBIHT versus OpenCV's GHT: (a) Failure of OpenCV in detecting the circles with ideal parameters set 1; (b) Failure of OpenCV with ideal parameters set 2; (c) Success of the CBIHT in detecting overlapping circles with no parameters.

## Real-life Applications

Applying this algorithm to a variety of real-life images, the true potential of the CBIHT can be seen through its accurate detection of circles amidst irregularity and noise. In Fig. 8 (a), the successful detection of the logo for Chanel, consisting of two overlapping, incomplete circles, by the CBIHT can be seen. Even more impressively, Fig. 8 (b) and (c) show the CBIHT approximating the ideal circles of immensely noisy images of a crater and an orange.



**Figure 8.** Circle Detection of Real-life Images by CBIHT: (a) Chanel logo with Two Incomplete Circles, (b) Crater Image with Rough/irregular Circular Shape, (c) Sliced Orange

Table 2 compares the computation times each method took in detecting the real-life images shown previously. The overall pattern of computation time remained the same as that shown in table 1, with the SHT taking the longest time, followed by the GHT, CBIHT, and lastly the CBHT. Additionally, the GHT was unable to locate the correct circle for the crater and orange image.



**Table 2.** Computation Time Comparisons for Real-life Images by Various Methods in seconds

	SHT	GHT	CBHT	CBIHT
Crater	13451.92	533.81	0.21	12.42
Orange	19795.40	92.05	0.21	9.07

## Conclusion

This paper presented a thorough analysis of the CBIHT in comparison to the Standard Hough Transformation along with its well-known variants, focusing on the accuracy and performance of each algorithm. When applying these methods to a set of synthetic and real-life images, our study demonstrated that the CBIHT has a greater resilience to noise than the GHT/AHT, a faster computation time than the SHT, and an independency from the pre-determined parameters needed by the GHT/AHT. All in all, the speed and adaptiveness of the CBIHT was able to be highlighted through this study.

## References

- Antolovic, D. (2018). Review of the Hough Transform Method, With an Implementation of the Fast Hough Variant for Line Detection, Technical Report TR663, [Online]. Available: <https://www.cs.indiana.edu/cgi-bin/techreports/TRNNN.cgi?trnum=TR663>. [Accessed 30 11 2018].
- Ballard, D. H. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes, *Pattern Recognition*, 13(2), pp. 111-122.
- Duda, R. and Hart, P. (1972). Use of the Hough Transform to Detect Lines and Curves in Pictures, In *Comm. ACM*. 15(1), pp. 11-15.
- Illingworth, J. and Kittler, J. (1987). The Adaptive Hough Transform, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, PAMI-9(5), pp. 690-699.
- Kim, H-S., Kim, J-H. (2001). Two-step circle detection algorithm from the intersecting chords, *Pattern Recognition Letters*, 22, pp. 787-798.
- Kimme, C., Ballard, B., and Sklansky, J. (1975). Finding Circles by an Array of Accumulators, *Comm. Of the ACM*, 18(2), pp. 120-122.
- O. Team, OpenCV, Intel. (2018). [Online]. Available: <https://opencv.org/>. [Accessed 30 9 2018].
- Park, J. (2019). A Center-based Iterative Hough Transformation for Unassisted Multiple Circle Detection, *Journal of the International STEAM*, 1, 1, 1-6.
- Ünver, H. M., Kökver, Y., Duman, E., Erdem, O. A. (2019). Statistical Edge Detection and Circular Hough Transform for Optic Disk Localization, *Applied Sciences*, 9, 350; doi:10.3390/app9020350.
- Wohlfart, M. (2003). Hough transform applications in Computer Graphics (with focus on medical visualization), unpublished, Spring 2003, [Online]. Available: [https://www.cg.tuwien.ac.at/courses/Seminar/SS2003/Ergebnisse/WohlfartMichael\\_HoughTransform.pdf](https://www.cg.tuwien.ac.at/courses/Seminar/SS2003/Ergebnisse/WohlfartMichael_HoughTransform.pdf). [Accessed 30 11 2018].

Wolfram, S. (1988, June 23). Wolfram Mathematica, Wolfram, [Online]. Available: <https://www.wolfram.com/mathematica>. [Accessed 20 October 2017].