Subpixel Detection of Circular Objects Using Geometric Property

Wen-Yen Wu, and Wen-Bin Yu

I. INTRODUCTION

Detecting circular objects is an important task in various industrial applications. Most algorithms for detecting and locating circular objects are based on Hough transformation (HT) or its variants [1, 3, 6, 7-19, 22-23]. In general, the HT methods employ a 3D accumulator array by mapping the x-y coordinates to the corresponding (x_c, y_c, r) parameter space. A peak finding procedure is then applied to detect circles.

Some non-HT algorithms have been proposed to solve the problems caused by HT-based algorithms. Wojcik [21] used graph to represent objects and a template matching technique is used to recognize the circular objects. Chen and Lin [4] used the orthogonal circular detector to estimate the parameters of circles. However, only isolated circles were tested in their study. Chen and Lee [5] used the geometric property of a circle to find some pixels on the circle and then fitted these pixels to get the center and radius of the detected circular object.

In this paper, we propose an efficient technique to identify the diameter as well as the pixels on the circumference of a circle by using the basic geometric properties of a circle. The result is of subpixel accuracy. The proposed method is presented in section II. The experimental results of the proposed method to various types of circular objects are illustrated in section III. Discussion and conclusion can be found in the final section.

II. THE PROPOSED METHOD

Some basic geometric properties are first discussed. Based on the properties, we develop a simple method for circular object detection.

Theorem 1. Let ∠BAC=0 (see Fig. 1), then the cosine value of 0 can be defined as follow:

\[
\cos \theta = \frac{u_1v_1 + u_2v_2}{\sqrt{u_1^2 + u_2^2} \sqrt{v_1^2 + v_2^2}},
\]

where \((u_1, u_2) = (x_2-x_1, y_2-y_1)\) and \((v_1, v_2) = (x_3-x_1, y_3-y_1)\)

Fig. 1 The computation of cosine: points A, B, and C

Theorem 2. As seen in Fig. 2, the segment BC is a diameter of the circle, if and only if, ∠BAC is equal to π/2.

Theorem 3. Let BC be a diameter of a circle (see Fig. 2). A is a point on the circle, if and only if, ∠BAC is equal to π/2.
Theorem 4. As shown in Fig. 3, BC is a diameter of a circle and h is the height from A to BC. Furthermore, if a, b, and c are the chord length of BC, AC, and AB respectively, then \( ah = bc \).

For a horizontal diameter, i.e. \( y_2 = y_3 \), it implies that \( h = y_2 - y_1 = y_3 - y_1 \) and \( a = x_3 - x_2 \) (assume that \( y_3 > y_1 \) and \( x_3 > x_2 \)), Eq. (1) can be formulated as:

\[
\cos \theta = \frac{(x_2 - x_1)(x_3 - x_1) + (y_2 - y_1)(y_3 - y_1)}{bc} = \frac{(x_2 - x_1)(x_3 - x_1) + (y_2 - y_1)^2}{ah} \tag{2}
\]

For a vertical diameter, that is, \( x_2 = x_3 \), similarly, we have

\[
\cos \theta = \frac{(x_2 - x_1)^2 + (y_2 - y_1)(y_3 - y_1)}{(x_2 - x_1)(y_3 - y_2)} \tag{3}
\]

A. Finding the Diameter

To identify pixels on a circle, it is important to find the diameter correctly. In Fig. 4, for each object we first find two pixels \( A_1(x_1, y_1) \) and \( A_2(x_4, y_4) \) vertically, and then the two other pixels \( B(x_2, y_2) \) and \( C(x_3, y_3) \) can be found by the corresponding mid-horizontal chord, i.e. the horizontal line \( y = (y_1 + y_4)/2 \). Therefore, \( y_2 = y_3 = (y_1 + y_4)/2 \). Because a circle is symmetric, the line segment connected by B and C is the candidate diameter of a circle. Suppose that \( A_1 \) or \( A_2 \) is one pixel on a circle. Next, we can test whether \( BC \) is a diameter by Theorem 2. That is, \( BC \) is a diameter, if and only if \( \angle B A_1 C \) or \( \angle B A_2 C \) is equal to \( \pi/2 \).

Therefore, if \( \cos \theta_1 \) or \( \cos \theta_2 \) equals to 0, where \( \theta_1 = \angle B A_1 C \) and \( \theta_2 = \angle B A_2 C \), then \( BC \) will be considered as a diameter of a circle. However, the condition of \( \cos \theta = 0 \) may be never hold due to the digitization error. Thus, the condition \( \cos \theta = 0 \) is replaced by \( |\cos \theta| < \epsilon \).

B. Detecting Pixels on a Circle

Once a diameter has been identified by the above searching strategy, one can subsequently look for all the candidate circular pixels. We first search for these pixels in pairs in the perpendicular direction corresponding to the circular diameter \( BC \) from B to C. For each pair of pixels, we can determine...
whether they are pixels on a circular object by Theorem 3 (see Fig. 6). Again, the condition of \( \cos \theta = 0 \) must be relaxed by setting an allowance as described in the last section.

**Fig. 6 Looking for circular pixels in pairs from \((A_1, A_2)\) to \((A_{21}, A_{22})\) with diameter BC**

However, some objects may result in false diameter detection as seen in Fig. 7. In order to avoid false detections, it is necessary to develop a criterion to judge whether the detected object is a circular object or not. Since the number of circular pixels is proportional to the diameter, a threshold \( n_t \) depending on the length of the detected diameter is specified. If the number of the detected circular pixels is larger than the threshold, the object is considered as a circle; otherwise, the object is not a circle. For example, only a few of pixels will be identified as the circular pixels in Fig. 7. Therefore, it is not a circle. In addition, for a broken or noisy circular object, the threshold should be decreased, because the number of circular pixels is less than that of a perfect one.

**Fig. 7 An example of objects results in false diameter detection**

**C. Estimating Parameters of the Circular Objects**

In this section, the procedure of estimating the parameters of the circle is demonstrated. Once the circular pixels were identified and the number was larger than a preset threshold, it is necessary to locate the circular object. The Thomas and Chan’s approach [20] is used to estimate the center \((x_c, y_c)\) and radius \(r\). Let \((\hat{x}_c, \hat{y}_c, \hat{r})\) be the estimate of \((x_c, y_c, r)\). For a set of \(n\) circular pixels, \((x_i, y_i)\) for \(i=1, 2, \ldots, n\), the formula of estimating the center and radius is given as:

\[
\begin{align*}
\hat{x}_c &= \frac{c_1b_2 - c_2b_1}{a_1b_2 - a_2b_1} \quad (4) \\
\hat{y}_c &= \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1} \quad (5) \\
\hat{r} &= \left( \frac{1}{n} \left( \sum x_i^2 - 2x_c \sum x_i + n\xi_c^2 + \sum y_i^2 - 2y_c \sum y_i + n\eta_c^2 \right) \right)^{1/2} \quad (6)
\end{align*}
\]

where

\[
\begin{align*}
a_1 &= 2(\sum x_i^2 - n\xi^2) \quad (7) \\
b_1 &= 2(\sum x_i \sum y_i - n\xi\eta) \quad (8) \\
a_2 &= 2(\sum x_i \sum y_i - n\xi\eta) = b_1 \quad (9) \\
b_2 &= 2(\sum y_i^2 - n\eta^2) \quad (10) \\
c_1 &= \sum x_i^2 - n\xi^3 + \sum x_i \sum y_i^2 - n\xi^2 \eta \\
c_2 &= \sum x_i^2 - n\xi^3 + \sum y_i \sum x_i^2 - n\eta^2 \xi \\
\end{align*}
\]

Considering the efficiency of computation, we can reduce the number of circular pixels in the least squares estimation. When locating circular pixels in pairs along the perpendicular direction of the diameter from B to C, instead of moving pixel by pixel, it can be done by jumping \(\alpha\)-pixels in each step. The setting of \(\alpha\) depends on the length of the diameter. The longer the length of the diameter is, the larger the \(\alpha\) can be. Both of the circular pixels identification and the parameter estimation time will be about \(1/\alpha\) of the original processing time. Besides, the threshold \(n_t\) must be reduced to \(n_t/\alpha\) at the same time.

**Fig. 8 Circle: continuous vs. digital**

The modified coordinates are closer to the true coordinates of a continuous circle than that of the original coordinates. In addition, they are of subpixel accuracy. They are adjusted by the following rules.

\[
\begin{align*}
\end{align*}
\]
D. Circle Detection Algorithm

In summary, the proposed circular object detection method can be stated explicitly as the following steps.

Step 1. Acquire the image of the object and convert it to binary image.

Step 2. Repeat Steps 3 to 6 until all objects have been processed.

Step 3. Find a horizontal, vertical, or diagonal diameter using the modified version of Theorem 2. If there is no diameter found, go to Step 2.

Step 4. Identify the circular pixels of the detected diameter using the modified version of Theorem 3.

Step 5. If the number of circular pixels is larger than a threshold $n_t$, a circle has been detected. Otherwise, go to Step 2.

Step 6. Modified the x-y coordinates of the circular pixels by Eq. (13) and Eq. (14). Next, estimate the parameters of the circle by Eq. (4) to Eq. (6).

III. EXPERIMENTAL RESULTS

For a perfect circular object, by experience, we can use 160% of the length of the detected diameter as threshold. That is, the value of $n_t$ is set to $1.6 \times \left| \overline{BC} \right|$, where $\overline{BC}$ is the length of the diameter of the object. On the other hand, if the object is semi-circular, then the threshold should be reduced to 80% of the diameter; that is, $n_t = 0.8 \times \left| \overline{BC} \right|$. In general, it is sufficient to use about $40$ pixels in the parameter estimation. In order to use about $40$ pixels in the parameter estimation for a perfect circle, the jumping interval $\alpha$ is set to the smallest integer that is larger than or equal to $\frac{\left| \overline{BC} \right|}{20}$.

In order to verify the proposed method, it has been tested on 400 synthetic images. Each of the synthetic images includes five perfect objects and four half-broken circular objects with random radius varying from 3 to 50 pixels. One of the images including five types of circular objects is presented in Fig. 9. Further, in order to access the ability of the proposed method under noisy condition, two additional experiments have been conducted. We repeat the first experiment by adding noise the testing images with signal to noise ratio (S/N) 20 and 50, respectively.

Since the centers and radii of the testing circles are known. The errors can be accessed. Let $\left( \hat{x}_c, \hat{y}_c, \hat{r} \right)$ be the estimate of the real parameter $\left( x_c, y_c, r \right)$.

The experimental results are shown in Table I. All of the circular objects were successfully detected. For noise-free images, the average errors of $x_c$, $y_c$, and $r_c$ are very small (within 0.10 pixels) for all the five types of circular objects. Especially for perfect circular objects, the average $x_c$ and $y_c$ are even smaller i.e.- about 0.01 pixels. In addition, the average accuracy level of radius approximates to 0.05 pixels for perfect circular object. For half-broken circular objects under the four directions, the accuracy level can reach 0.10 pixels. Further, the new method can detect very small circular objects. For a synthetic image, it can even detect circular objects with radius of 3 pixels.

The experimental results are shown in Table I. All of the circular objects were successfully detected. For noise-free images, the average errors of $x_c$, $y_c$, and $r_c$ are very small (within 0.10 pixels) for all the five types of circular objects. Especially for perfect circular objects, the average $x_c$ and $y_c$ are even smaller i.e.- about 0.01 pixels. In addition, the average accuracy level of radius approximates to 0.05 pixels for perfect circular object. For half-broken circular objects under the four directions, the accuracy level can reach 0.10 pixels. Further, the new method can detect very small circular objects. For a synthetic image, it can even detect circular objects with radius of 3 pixels.

The experimental results are shown in Table I. All of the circular objects were successfully detected. For noise-free images, the average errors of $x_c$, $y_c$, and $r_c$ are very small (within 0.10 pixels) for all the five types of circular objects. Especially for perfect circular objects, the average $x_c$ and $y_c$ are even smaller i.e.- about 0.01 pixels. In addition, the average accuracy level of radius approximates to 0.05 pixels for perfect circular object. For half-broken circular objects under the four directions, the accuracy level can reach 0.10 pixels. Further, the new method can detect very small circular objects. For a synthetic image, it can even detect circular objects with radius of 3 pixels.

For the images with S/N=20, the average errors are all smaller than 0.10 pixels for the perfect circular objects. The average errors of x-coordinate and y-coordinate are less than 0.23 for the semi-circles. In addition, for the images with S/N=50, the average errors of x-coordinate, y-coordinate, and radius are less than 0.90 pixels for the perfect circles. The errors are less than 1.50 pixels for the broken circular objects. The results indicate that the proposed method is robust for noise.

Another testing image that contains various types of objects is shown in Fig. 10. They include some near half-broken,
occluded, and perfect circular objects as well as other shaped objects. The detected circles (dash-lined) are superimposed into the original image and the detected centers have been marked as small crosses. As it can be seen, the proposed method can successfully locate all the circular objects.

![Fig. 10 Results of detecting circular objects, where the detected circles are presented by dash-line and the centers are marked as crosses](image)

IV. CONCLUSION

A new technique for circular object detection and location is proposed. Instead of transforming pixels into its parameter space as suggested by the HT approaches, we make use of the geometric properties of a circle to find the diameter and circumference pixels and estimate the parameters of the circles. In comparison with the HT approaches, the proposed method has the advantages of rather small storage required, high speed, and high accuracy. Further, the new method does not suffer from the problems of clustering and peak finding involved in HT-based methods. The experimental results also indicated that incomplete, occluded, as well as tiny circular objects can also be successfully detected and located by the proposed method. Further, the proposed method is robust to noisy environment. Overall, the proposed is effective and efficient in detecting circular objects.

REFERENCES