Robust ellipse detection based on hierarchical image pyramid and Hough transform

Chung-Fang Chien, Yu-Che Cheng, and Ta-Te Lin*

Department of Bio-Industrial Mechatronics Engineering, National Taiwan University,
No. 1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan
*Corresponding author: m456@ntu.edu.tw

Received September 28, 2010; revised December 28, 2010; accepted January 28, 2011; posted February 1, 2011 (Doc. ID 135830); published March 16, 2011

In this research we propose a fast and robust ellipse detection algorithm based on a multipass Hough transform and an image pyramid data structure. The algorithm starts with an exhaustive search on a low-resolution image in the image pyramid using elliptical Hough transform. Then the image resolution is iteratively increased while the candidate ellipses with higher resolution are updated at each step until the original image resolution is reached. After removing the detected ellipses, the Hough transform is repeatedly applied in multiple passes to search for remaining ellipses, and terminates when no more ellipses are found. This approach significantly reduces the false positive error of ellipse detection as compared with the conventional randomized Hough transform method. The analysis shows that the computing complexity of this algorithm is $\Theta(p^{\alpha^2})$, and thus the computation time and memory requirement are significantly reduced. The developed algorithm was tested with images containing various numbers of ellipses. The effects of noise-to-signal ratio combined with various ellipse sizes on the detection accuracy were analyzed and discussed. Experimental results revealed that the algorithm is robust to noise. The average detection accuracies were all above 90% for images with less than seven ellipses, and slightly decreased to about 80% for images with more ellipses. The average false positive error was less than 2%. © 2011 Optical Society of America

OCIS codes: 100.2000, 100.3008, 100.4904, 100.5010.

1. INTRODUCTION

The Hough transform is a feature extraction technique widely used in digital image processing. It is originally a method of locating underlying lines in a digital image by mathematically identifying collinear points [1]. In this process, every candidate pixel in an image is transformed into another parameter space, which is defined by the parametric representation used to describe lines in the image plane. Duda and Hart modified the Hough transform by proposing a method of using a different set of parameters to describe a straight line, and they also suggested that the Hough transform can be extended to arbitrary geometric shapes [2]. The feature extraction process is particularly robust to missing data and tolerant to noise. It has been popularized in the computer vision community and has become a standard tool for the recognition of straight lines, circles, and ellipses since then [3–5].

For a more complex geometric shape, such as an ellipse that needs a five-dimensional parameter space, the Hough transform is usually a time- and memory-consuming task. Many researchers have proposed various approaches for ellipse detection using the Hough transform [6–13]. These efforts on reducing the computational and memory requirements all contribute to improving the performance and efficiency of the Hough transform algorithms. Besides applying the Hough transform approach, other searching-based techniques have also been developed for multiple ellipse detection in real-world images [14–17].

In the field of image representations, pyramid data structures are considered to be an important approach in many applications [18–20]. In the transmission of images through communication links, progressive transmission has been proved to be an efficient way for improving bit rate reductions [21–25]. Conceptually, progressive transmission is based on human image-understanding capabilities. An observer may perceive the general characteristics of a scene or an image in different resolutions and, therefore, one can intuitively extract some main features from a scene even if it is obscure. Therefore, the pyramid structure is suitable for images containing dominantly low-frequency components. Based on the approach of pyramid data structure, researchers have integrated it with the Hough transform algorithm for the detection of some geometric shapes with satisfactory results [26–29]. However, for geometric shapes defined by a higher number of parameters, such as a five-parameter ellipse, the accuracy and computation cost in applying the Hough transform for object detection still need improvement. This is especially true for applications requiring real-time processing. With this rationale, we integrated the concepts of pyramid data structure and iterative search to improve the computational complexity and accuracy of the elliptical Hough transform in this research.

In this paper, we begin with a brief introduction to the elliptical Hough transform. Then the concepts of the image pyramid are presented and integrated with the formulation of the elliptical Hough transform. An analysis on the computing complexity of the new algorithm follows. To test the algorithm, experiments were performed on both real and synthetic images. We investigate the effect of noise-to-signal ratio (NSR) and the size of ellipses on computation time and detection accuracy, and present the experimental results with discussion.

1084-7529/11/040581-09$15.00/0 © 2011 Optical Society of America
2. ELLIPTICAL HOUGH TRANSFORM

A. Hough Transform

The Hough transform is a powerful technique for object detection and is a special case of the discrete Radon transforms [6,30,31]. The transform converts every candidate pixel in an image to another parameter space. The parameter space is defined by the parametric representation for describing lines in the image plane. Duda and Hart modified the original Hough transform by proposing a different set of parameters for describing a straight line [2]. They also suggested that the Hough transform can be extensively applied to arbitrary geometric shapes.

A point in the Cartesian coordinates \((x, y)\) can be translated into the polar coordinates \((\rho, \theta)\) and then a straight line in the polar coordinate system can be represented by a single point in the parameter space using the following parameterized form:

\[
\rho = y \sin \theta + x \cos \theta, \tag{1}
\]

where \(\rho\) is the perpendicular distance from the origin of the Cartesian coordinates to the line and \(\theta\) is the angle with the normal. Collinear points are transformed into sinusoidal curves in the polar plane, which intersect at the point \((\rho, \theta)\). In practice, the transform is implemented by quantizing the parameter space into finite intervals or accumulator cells. Each point is transformed into a curve and the accumulator cells lying on this curve are incremented. The peaks in the final accumulator array represent possible candidates corresponding to the straight lines in the image.

B. Elliptical Hough Transform

For the elliptical Hough transform, we use the following classical definition of an ellipse: the set of points in a plane whose distances from two fixed points in the plane have a constant sum [32]. The two fixed points are the foci of the ellipse. In the Cartesian coordinates, this definition can be expressed as

\[
d_r = \sqrt{(x-x_1)^2 + (y-y_1)^2} + \sqrt{(x-x_2)^2 + (y-y_2)^2}, \tag{2}
\]

where \((x, y)\) are the points on the ellipse, \((x_1, y_1)\) and \((x_2, y_2)\) are the coordinates of the two foci, and \(d_r\) is the sum of distances from the two foci to any point on the ellipse. This set of image points \((x, y)\) can be defined by a relation, \(f\), as

\[
f((x_1, y_1, x_2, y_2, d_r), (x, y)) = \sqrt{(x-x_1)^2 + (y-y_1)^2} + \sqrt{(x-x_2)^2 + (y-y_2)^2} - d_r = 0. \tag{3}
\]

After the polling process, the number of votes is stored in a five-dimensional accumulator array \([x_1][y_1][x_2][y_2][d_r]\). The higher numbers of votes in the accumulator cells correspond to candidate ellipses and they are queued into a candidate list according to their voting counts. It is obvious that, for this five-dimensional parameter space, the Hough transform requires intensive computation and a huge amount of memory for the accumulator cells.

For the inverse transform, the point in the five-dimensional parameter space is transformed back to an ellipse in the image. Assuming that one or more ellipses exist in the image, we may start from the standard equation of an ellipse centered at the origin \((0, 0)\):

\[
\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{4}
\]

where \(x\) and \(y\) represent the Cartesian coordinates of any point on the ellipse, and \(a\) and \(b\) are the major and minor axes of the ellipse, respectively. This is an ellipse whose foci are on the \(X\) axis and the ellipse possesses the following basic properties:

- center-to-focus distance, \(c = \sqrt{a^2 - b^2}\);
- foci, \((\pm c, 0)\); and
- horizontal vertices, \((\pm a, 0)\).

Equation (4) can also be expressed in polar coordinates by substituting \(x = r \cos \theta\) and \(y = r \sin \theta\). After rearrangement, the equation becomes

\[
r = \pm \sqrt{\frac{a^2 b^2}{b^2 \cos^2 \theta + a^2 \sin^2 \theta}} \tag{5}
\]

where \(r\), choosing the positive value, denotes the distance between a point \(A\) on the ellipse and the center \(C\) of the ellipse, and \(\theta\) is the angle between the \(X\) axis and the vector \(\overrightarrow{CA}\). Equation (5) is an extension of the standard parametric Hough transform for ellipses. For an ellipse transformed into the five-dimensional parameter space, the discrete coordinates of the corresponding accumulator cell are \(x_1, y_1, x_2, y_2,\) and \(d_r\). When these values are determined by the voting scheme, the parameters of Eq. (4) can then be calculated:

\[
c = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}, \tag{6}
\]

\[
a = \frac{d_r}{2}, \tag{7}
\]

\[
b = \sqrt{a^2 - c^2} = \sqrt{\left(\frac{d_r}{2}\right)^2 - c^2}, \tag{8}
\]

\[
x = r \cos \theta = \frac{d_r}{2} \sqrt{\frac{d_r^2 - 4c^2}{d_r^2 - 4c^2 \cos^2 \theta}} \tag{9}
\]

\[
y = r \sin \theta = \frac{d_r \sin \theta}{2} \sqrt{\frac{d_r^2 - 4c^2}{d_r^2 - 4c^2 \cos^2 \theta}} \tag{10}
\]

So every point \((x, y)\) on the ellipse can be derived from Eqs. (9) and (10).

However, not every ellipse is as ideal as the standard-form ellipse. To generalize the formulation for ellipses in arbitrary orientation and location, one may consider the ellipse with a rotation angle \(\alpha\) from the \(X\) axis to its major axis counterclockwise. The angle \(\alpha\) may be determined by the coordinates of the two foci:
\[ \alpha = \tan^{-1} \left( \frac{y_2 - y_1}{x_2 - x_1} \right). \]

Finally, by plane coordinate transformation, every point \((x', y')\) of the ellipse represented by \((x_1, y_1, x_2, y_2, d_\alpha)\) parameters can be described as follows:

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} r' \cos \theta \\ r' \sin \theta \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix},
\]

where \((x_0, y_0)\) is the center of the ellipse. After introducing the values determined using Eqs. (9)-(11), the coordinates of all points on the ellipse can be easily derived by substituting \(\theta\) from 0° to 360° in Eq. (12).

3. HOUGH TRANSFORM BASED ON IMAGE PYRAMID

Because the use of the Hough transform to detect ellipses requires a five-dimensional array of accumulator cells, it is necessary to reduce the computing complexity for practical applications. To achieve this aim, we integrated the elliptical Hough transform with the image pyramid data structure and dynamic quantization to relax intensive computation \([6, 18, 31]\). This approach does not require exhaustive search of every pixel in the image. Instead, it performs the subdivision and subsequent “vote counting” only for candidate ellipses with votes in accumulator cells exceeding a preset threshold. Thus it is possible to locate ellipses quickly by detecting apparent ellipses in the low-resolution image of the image pyramid. The successive high-resolution image layer is subsequently searched to obtain more detailed boundary pixels of ellipses when the ellipses are considered existing in the low-resolution image. By using this hierarchical approach, significant reduction in both computational complexity and memory requirement can be achieved. The new algorithm developed in this research was implemented for ellipse detection, but it can be further extended to other geometric shapes. The algorithm consists of three main steps: (1) reducing image resolution, (2) exhaustive searching for ellipses, and (3) increasing image resolution iteratively and relocating ellipses. Figure 1 shows the flowchart of the three-step algorithm, and a more detailed description of the algorithm is as follows.

A. Step 1: Reducing Image Resolution

The algorithm starts with a resolution reduction process to alter the image resolution from the originally higher resolution to a preset low-resolution image. In the experiments of this research, the original image resolution was \(512 \times 512\) pixels and the preset lowest resolution was \(32 \times 32\) pixels, which is the initial image resolution for the exhaustive search for ellipses. In other words, we start the elliptical Hough transform from the low-resolution image of the image pyramid, and, thus, the five-dimensional accumulator cells for voting scheme can be significantly reduced to a reasonable level. The Hough transform was only applied to the boundary pixels of the ellipses to cut down computation time. The down-sampling method of an image is depicted in Fig. 2. From fine to coarse resolution, such as from Figs. 2(a) and 2(b), we merge pixels in both the horizontal and vertical dimensions. In each resolution reduction step, four pixels are merged into one pixel. A parallel weighting factor pyramid, the same size as the image pyramid, is created to record the number of boundary pixels in the four pixels to be merged. The down-sampling process is performed from the \(512 \times 512\) image layer to the \(32 \times 32\) image layer and the number of boundary pixels are accumulated and recorded in each corresponding layer of the weighting factor pyramid. These numbers are later used as the weighting factors for the normalization of vote counts in the subsequent stage of the elliptical Hough transform.

For an original image of \(n \times n\) pixels, the coarsest resolution in the image pyramid is an image of \(1 \times 1\) resolution. If \(n\) is an integral power of 2, \(n = 2^m\), then this pyramid architecture is a stack of \(m\) images of sizes \(2^m \times 2^m, 2^{m-1} \times 2^{m-1}, 2^{m-2} \times 2^{m-2}, \ldots, 2 \times 1 \times 1\). The height of the pyramid, the number of layers, is \(m + 1 = (\log_2 n) + 1\), and the sum of all pixels in the image pyramid is \((4/3)n^2 - 1/3\). To determine whether the pixel is a boundary pixel for the Hough transform, we must finally examine every pixel in the image pyramid in the down-sampling process. Obviously, the complexity at this step for an \(n \times n\) image is \(\Theta(n^2)\).
The votes of all ellipses are accumulated in a five-dimensional array. The votes in each accumulator cell need to be further normalized first by dividing the vote counts by the number of boundary pixels of the corresponding ellipse, and then multiplying them with the corresponding weighting factor stored in the weighting factor pyramid created during the pixel merging process in Step 1. A preset threshold value is employed to pick candidate ellipses corresponding to larger values of normalized vote counts. Then the candidate ellipses are stored in a candidate list and sorted in a descending order according to the normalized vote counts.

In the exhaustive search process, each boundary pixel has voted for many possible ellipses. However, in practice, a boundary pixel does not belong to many ellipses unless it is at the intersection of ellipses. To avoid the effect of multiple voting from a single pixel, the vote counts of the candidate list are reexamined to remove duplicate votes according to the rule that a pixel only casts a vote on the most probable ellipse in the candidate list that corresponds to the candidate ellipse with the highest vote counts. This extra processing on the candidate list of ellipses is helpful in reducing the interference between boundaries of ellipses and the effect is especially obvious when the shape of ellipses is not ideal in an image.

C. Step 3: Increasing Image Resolution Iteratively and Relocating Ellipses

In the previous step, a sorted list of candidate ellipses is obtained in the low-resolution image. The resolution of the image is then increased step by step until the original image resolution is reached. Increasing the image resolution also increases the uncertainty of the positions and dimensions of the detected ellipses that mainly resulted from the quantization error. To relocate the detected ellipses in a higher-resolution image, the information of position and size of the ellipses determined in the low-resolution image need to be used. If a selected ellipse is defined by the discrete parameters \((X_1, Y_1, X_2, Y_2, D_r)\) in a \(k \times k\) image, its reasonable distribution ranges should be

\[
\begin{align*}
X_1 - 0.5 &\leq x_1 < X_1 + 0.5 \\
Y_1 - 0.5 &\leq y_1 < Y_1 + 0.5 \\
X_2 - 0.5 &\leq x_2 < X_2 + 0.5. \\
Y_2 - 0.5 &\leq y_2 < Y_2 + 0.5 \\
D_r - 0.5 &\leq d_r < D_r + 0.5
\end{align*}
\]

where \(X_1, Y_1, X_2, Y_2,\) and \(D_r\) are integers after rounding off real numbers \(x_1, y_1, x_2, y_2,\) and \(d_r,\) respectively. As depicted in Fig. 3(a), the position of a boundary pixel \(P\) in discrete location is exactly on the intersection of horizontal and vertical lines. Its possible distribution in the continuous spatial domain is within the region enclosed by the dashed square in Fig. 3(b). When we increase image resolution from \(4 \times 4\) to \(8 \times 8\) pixels, the possible position of \(P,\) enclosed by the dashed square, is shown in Fig. 3(c). Thus the new parameters for the enlarged ellipse are bounded by

\[
\begin{align*}
2X_1 - 1 &\leq x_1^* < 2X_1 + 1 \\
2Y_1 - 1 &\leq y_1^* < 2Y_1 + 1 \\
2X_2 - 1 &\leq x_2^* < 2X_2 + 1. \\
2Y_2 - 1 &\leq y_2^* < 2Y_2 + 1 \\
2D_r - 1 &\leq d_r^* < 2D_r + 1
\end{align*}
\]
where \( x'_1, y'_1, x'_2, y'_2 \), and \( d' \) are new parameter values in the \( 8 \times 8 \) image, and the integers \( X'_1, Y'_1, X'_2, Y'_2, \) and \( D'_1 \) are round-off values of \( x'_1, y'_1, x'_2, y'_2, \) and \( d' \), respectively. They can be rearranged as

\[
\begin{align*}
X'_1 &\in \{2X_1, 2X_1 \pm 1\} \\
Y'_1 &\in \{2Y_1, 2Y_1 \pm 1\} \\
X'_2 &\in \{2X_2, 2X_2 \pm 1\} \\
Y'_2 &\in \{2Y_2, 2Y_2 \pm 1\} \\
D'_1 &\in \{2D, 2D \pm 1\}
\end{align*}
\]  

(15)

During the up-sampling process, however, there should be only one ellipse inherited from the previous low-resolution image. To relocate an ellipse in the higher resolution image, the Hough transform using Eqs. (9)–(12) needs to be carried out to search the most probable ellipse in the range of parameters defined by Eq. (15). This step can be considered as adjusting the previously searched ellipse to a more precise position and dimension. The location and dimension of the ellipse is slightly "shaken" and "shrunk or enlarged" to fit the newly modified ellipse container.

Assuming that the relocation of an ellipse takes constant computing time \( C_s \) for each boundary pixel, the computation time is \( p_x \times C_s \) for an ellipse with \( p_x \) boundary pixels. Take the case of up-sampling process in Fig. 3(b) and 3(c), for example; the total computation time needs about \( (2p_x) \times 3^5 \times C_s \) to relocate an ellipse in the higher-resolution image. For an image pyramid with its height \( m = \log_2 n \) and assuming that there are \( p_x \) boundary pixels for each ellipse on average, \( p_x \) has the worst computation time of \( \Theta(n) \). So if there are \( N \) ellipses on an image, without loss of generality, the computation time can be calculated as

\[
C_{\text{total}} = N \times p_x \times C_s + N \times (2p_x) \times 3^5 \times C_s + N \times (4p_x) \\
\times 3^5 \times C_s + \ldots + N \times (2^m p_x) \times 3^5 \times C_s
\]

\[
= N \times p_x \times C_s + N \times p_x \times C_s \times 3^5(2^1 + 2^2 + \ldots + 2^m)
\]

\[
= N \times p_x \times C_s + N \times p_x \times C_s \times 3^5 \times 2 \times (2^m - 1)
\]

\[
= N \times p_x \times C_s (1 + 3^5 \times 2^{m+1} - 2 \times 3^5) = \Theta(n) \times \Theta(2^{m+1})
\]

\[= \Theta(n) \times \Theta(2n) = \Theta(n^2). \]  

(16)

Therefore, for this step, the computation time is bounded by \( \Theta(n^2) \).

For the overall three-step algorithm to search for ellipses in an \( n \times n \) image, the computation complexity can be calculated by the sum of these three steps:

\[
T(n) = \Theta(n^2) + \Theta(n^{5/2}) + \Theta(n^2) = \Theta(n^{5/2}).
\]  

(17)

## 4. MULTIPASS DETECTION OF ELLIPSES

We further extend the three-step algorithm to a multipass process for searching ellipses to enhance the performance of ellipse detection. The flowchart of this modified algorithm is shown in Fig. 4. With this strategy, the more outstanding ellipses are selected in the first pass of searching. When the ellipses are recognized, the boundary pixels of the identified ellipses are erased and not to be used in the three-step Hough transform algorithm for the next pass. Consequently, the boundary pixels in the later passes are usually less than the previous pass, and thus the search process will be faster than that in the previous pass. This multipass search process terminates when no more new ellipses are found. The effect of the multipass process is to reduce the interference between boundary pixels of different ellipses. When ellipses are detected in each pass and removed from the image, the detection of less obvious ellipses in the next pass becomes a simpler task without interference from removed pixels not belonging to the remaining ellipses.

## 5. EXPERIMENTS AND ANALYSES

To test the developed elliptical Hough transform algorithm, a software program was implemented using Borland C++ Builder under the Microsoft Windows operating system. The elliptical Hough transform searching for ellipses with this three-step approach has worked well and proved to be successful for detecting real images of vegetable seedling leaves; those results were published elsewhere [33, 34]. In this research, we place emphasis on the formulation of the algorithm and discussion of the performance of the new algorithm. To analyze the computation time and detection accuracy, 500 synthetic binary images containing ellipses of random sizes, position, and orientation were produced. There are 10 sets of images, with 50 images in each set, containing various
numbers of ellipses from one to 10 ellipses in an image. Figure 5 shows two typical images, one with five ellipses and the other with 10 ellipses. The ellipses usually intersect, especially for images with more ellipses. Computation experiments for ellipse detection were carried out with these images on a desktop computer with 2.4 GHz Intel Pentium 4 CPU. The detection accuracy, false positive errors, and computation time were recorded and analyzed. In these experiments, we used a preset threshold value of 4 to pick candidate ellipses to form the candidate list of ellipses in the lowest-resolution pyramid image, which means a possible ellipse is considered when there are at least four votes in an accumulator cell.

A. Experiments on Real Images

Figure 6 shows five real images for demonstrating the performance of ellipse detection using our algorithm. In practice, preprocessing is usually required to segment the objects from the background and extract boundary pixels of ellipse-shaped objects in an image. The images in the central column of Fig. 6 are the processed images following preprocessing of object segmentation, edge detection, and thinning. We used color-based segmentation, and the Kirsch edge detection and Yokoi thinning process in the preprocessing [35]. The processed images containing boundary pixels of objects are input images of the elliptical Hough transform algorithm. The images in the right column of Fig. 6 show the results of ellipse detection. The image containing coins is a typical example representing images with ideal ellipse-shaped objects, while the image of seedling leaves demonstrates the robustness of the elliptical Hough transform algorithm in detecting objects of approximately elliptical shape. The third image demonstrates the accurate detection of ellipse-shaped objects while the two integrated circuit chips were ignored. The fourth image shows the capability of the algorithm in detecting a occluded ellipse. Finally, the bottom image exhibits a more complicated scene with various geometric shape and noise. The two ellipses in the image were correctly detected.

B. Computation Time of the Algorithm

Figure 7 shows the average computation time (in seconds) against the number of ellipses in an image. The multipass approach executed at least one more pass than that in a single pass. However, the difference in computation time is not significant for images with fewer ellipses in both cases. In a simple image, the ellipses can be detected easily in the first pass. Thus, not many pixels remained after removing the determined ellipses on the image. In a more complex image, ellipse detection usually takes several passes. It is reasonable to expect that a multipass search in an image with more ellipses takes more time than an image with fewer ellipses. This can be observed in Fig. 7, in both per-image and per-ellipse cases. For both single-pass and multipass detection, the average computation time per image increases proportionally to the number of ellipses in an image. Computation time per ellipse is relatively constant, ranging from 6.6 to 11.4 s per ellipse for multipass detection. The computation time per ellipse of the single-pass detection decreases slightly as the number of ellipses increases.

<table>
<thead>
<tr>
<th>Original image</th>
<th>After segmentation, edge detection, and thinning</th>
<th>Detected ellipses</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="a" alt="Original image" /></td>
<td><img src="b" alt="After segmentation, edge detection, and thinning" /></td>
<td><img src="c" alt="Detected ellipses" /></td>
</tr>
</tbody>
</table>

Fig. 5. Typical synthetic images containing various numbers of ellipses: (a) image with five ellipses; (b) image with 10 ellipses.

Fig. 6. (Color online) Demonstration of ellipse detection in real images. The images in the left column are the original images. The binary images in the central column are images after segmentation, edge detection, and thinning. The images in the right column show the detected ellipses corresponding to the original images.

Fig. 7. Comparison of computation time for ellipse detection using the single-pass and multipass approaches.
C. Detection Accuracy of the Algorithm

Figure 8 plots the percentage of ellipses correctly found against the number of ellipses contained in each image. For simple images with either one or two ellipses, 100% detection accuracy was achieved using the single-pass approach. However, for more complex images with more than five ellipses, the detection accuracy decreased. The average detection accuracy was about 40% for images with 10 ellipses at the first pass. However, when using the multipass approach, the detection accuracies were significantly improved and all were at a satisfactory high level. The average accuracies were all above 90% for images with fewer than seven ellipses, and slightly decreased to about 80% for images with more ellipses. These experimental results indicate that the multipass search approach can be applied robustly and effectively for images with multiple ellipses. It also reveals the effect of interference among boundaries of multiple ellipses on the detection of ellipses using the Hough transform algorithm. By removing the detected ellipses at each pass, the interference can be reduced for subsequent passes. The detection accuracies of two other ellipse detection algorithms are also presented in Fig. 8 for comparison. The data reported by McLaughlin [10] for the randomized Hough transform (RHT) method and the Hough transform method using geometric symmetry (GSHT) [36] were plotted. For images containing one or two ellipses, the detection accuracies were similar and all above 85%, except for the GSHT method. It is obvious that the multipass approach of our proposed method outperformed the RHT and GSHT methods for images containing more than four ellipses.

Figure 9 shows the average false positive error for each set of images. There was no difference in false positive errors using single-pass or multipass search. For images with few ellipses, false positive detection is rare because only a small number of boundary pixels may contribute to the votes for the Hough transform. An image with more ellipses is more apt to encounter false positives. However, the average false positive error was just 2% for the image set of 10 ellipses in our experiments. The false positive errors reported by McLaughlin for the RHT and GSHT methods was about 2%–4% and 7%–47% for images containing 1–8 ellipses, respectively [10].

D. Analyses of the Effect of Noise

Figure 10 shows the average computation time per ellipse for the five sets of images with different NSRs. In this multipass ellipse detection experiment, it is obvious that more boundary
pixels resulted in more computation time for the detection of ellipses. The number of pixels in the image, including boundary and noise pixels, appears to be the major factor influencing the computation time. Figure 10 also reveals that the computation time per ellipse increases with higher NSR. The increasing trend is approximately proportional to the levels of NSR. The computation time is also proportional to the sizes of ellipse.

Figure 11 shows the average detection accuracy of the multipass ellipse detection algorithm as affected by the levels of NSR for images containing different sizes of ellipses. In this experiment, the detection accuracies were in the range of 70%-100% for images containing ellipses of $d_e$ with 70, 90, and 110 pixels when the NSR was less than 200%. There are no significant differences among the detection accuracies of these cases when the NSR was less than 200%. The detection accuracy drops considerably when NSR reaches a threshold value of approximately 200% for images containing ellipses of $d_e$ with 90 and 110 pixels. The NSR does have a strong effect on the detection accuracy for cases of $d_e$ with 130 and 150 pixels. For these cases, the detection accuracy decreases significantly as NSR increases. At a constant NSR, there appears to be a threshold value of $d_e$ above which the detection accuracy drops significantly. At constant NSR, the number of noise pixels actually increases with ellipse size. The low detection accuracy was also partly due to stronger interference among larger ellipses in the voting scheme of the Hough transform. The experimental result indicates that the algorithm is only robust to a limited level of noise in an image. In this experiment on 10-ellipse images with various levels of NSR, when $d_e$ is less than 110 and NSR is below 200%, the average detection accuracies were stably in the range of 70%-100%. This is a fairly large range and pertinent to most ellipse detection applications. This experimental result demonstrates that the multipass algorithm is robust to noise.

### 6. CONCLUSIONS

In this paper, we present an iterative Hough transform algorithm for ellipse detection based on image pyramid data structure and multipass approach. The computing complexity of this algorithm is $\Theta(n^{0.5})$. This new algorithm has distinctive advantages in that it is easy to understand and implement. By using the multipass three-step elliptical Hough transform, the computation time and memory requirement are significantly reduced. The average detection accuracies were all above 90% for images with less than seven ellipses, and slightly decreased to about 80% for images with more ellipses. The average false positive error was less than 2% for all images tested in our experiments. We also demonstrated that the developed algorithm was very robust even when the NSR in an image is high. The significant reduction of computational load and high detection accuracy achieved by using this algorithm makes the five-dimensional elliptical Hough transform applicable to most practical applications.

### REFERENCES

34. C. F. Chien, “Non-destructive measurement of selected vegetable seedlings using 3D machine vision,” Ph.D. dissertation (Bio-Industrial Mechatronics Engineering Department, National Taiwan University, 2003).