# A Hierarchical Pyramid Elliptical Hough Transform

*Chung-Fang Chien*[1], *Ta-Te Lin*[1], *and Chiou-Shann Fuh*[2]

[1]Department of Bio-Industrial Mechatronics Engineering, National Taiwan University
d86631001@ntu.edu.tw, m456@ntu.edu.tw
[2]Department of Computer Science and Information Engineering, National Taiwan University
fuh@csie.ntu.edu.tw

## ABSTRACT

In this research we develop a new ellipse detection algorithm. This algorithm is based on the applications of divide-and-conquer Hough transform and pyramid-based image data structure. Firstly, the exhaustive searching on a reduced resolution image is made by elliptical Hough transform. Then the processing gradually increases image resolution and updates candidate ellipses till the final resolution reached. After removing the determined ellipses, the algorithm searches for remaining ellipses over and over and terminates at no more ellipses being found. By using this algorithm we successfully achieve satisfactory results. The average computation time for searching each ellipse is about 6 to 11.3 seconds, and the accuracy rate is about 100% to 88.5%, on a 512×512 image with one to 10 ellipses. The average false alarm rate about searching ellipses is far beneath 2%, in the image with no more than 10 ellipses. This algorithm demonstrates highly robust and effective results.

**KEYWORDS:** Ellipse, Hough Transform, Pyramid Image

## 1. INTRODUCTION

The Hough Transform [11] is a widely used method of parameter extraction. The Hough transform is originally a method of locating underlying lines in a digital image by identifying collinear points mathematically. Every candidate pixels in an image will be transformed into another parameter space. The parameter space is defined by the parametric representation used to describe lines in the image plane. Duda and Hart [6] modified the Hough transform by proposing the use of different set of parameters to describe a straight line. They also suggested that the Hough transform could be extended to arbitrary shapes. This technique has long been widely used to characterize shape detection from contours. But for a more complex shape, such as an ellipse, that the Hough transform needs the requirement of a five-dimensional parameter space is obviously. The Hough transform is always a time- and storage-consuming task. Many researchers have provided some approaches to aim at the ellipse searching by using Hough transform [1, 9, 12, 13, 17, 18, 19, 21, 24, 27]. Those efforts for reducing the computational and memory requirements all contribute to improve the performance and efficiency. Anyway, ellipse detection is still one of the key problems in image processing.

In the field of image representations pyramid data structures are considered to be an important class of approaches [22, 28]. Moreover, in the transmission of images through communication links, progressive transmission has been proved to be an efficient way on improving bit rate reductions [8, 10, 20, 25, 29]. In fact, progressive transmission uses the image understanding capabilities of the human being. An observer may know the general characteristics of the image that he is looking for. Usually, he could thus make a more detailed forecast about the image than he really got in fact. In general, this pyramid structure is very suitable with images which contain dominant low-frequency components.

Some researchers have developed a pyramid-based Hough transform algorithm to detect some selected geometric shapes fast and efficiently [2, 7, 14]. They seem to achieve satisfactory results on detecting straight edges. Thus in this research, we developed a new algorithm based on pyramid image and Hough transform to fast and efficiently detect 5-parameter ellipses.

## 2. ELLIPTICAL HOUGH TRANSFORM

### 2.1. Hough Transform

The Hough Transform [11] is originally a method of locating underlying lines in a digital image by identifying collinear points mathematically. Though this transform is a special case of the Radon transform [5, 16], this technique is powerful for multi-dimensional pattern detection and parameter extraction, anyway [18]. Hough transform applies every candidate pixel in an image to transform into another parameter space. The parameter space is defined by the parametric representation used to describe lines in the image plane.

Duda and Hart [6] modified the Hough transform by proposing the use of a different set of parameters to describe a straight line. They also suggested that the Hough transform could be extended to apply to arbitrary shapes. If a point (*x,y*) on a generally known Cartesian coordinates that could be translated to a polar coordinates and the point could be represented as ($\rho$,$\theta$). Then a straight line in the polar coordinates system could be expressed by a single point in the parameter space as:

$$\rho = y\sin\theta + x\cos\theta \qquad (1)$$

where $\rho$ is the perpendicular distance from the origin of the coordinates to the line. The straight line in X-Y plane and in parameter space is shown in Fig. 1 (A) and (B). All points corresponding with this equation are lying on the line. This transform method is widely applied later for many types of geometric shapes not limited in line finding, but also in searching circles, ellipses and arbitrary shapes [1, 15, 27].

### 2.2. Elliptical Hough Transform

This research uses the definition of ellipse by Thomas and Finney: An **ellipse** is the set of points in a plane whose distances from two fixed points in the plane have a constant sum [26]. The two fixed points are the foci of the ellipse. And this definition can be expressed as (2):

$$d_r = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2} + \sqrt{(x_i - x_2)^2 + (y_i - y_2)^2} \qquad (2)$$

where $x_i$ and $y_i$ are the set of points on the ellipse, ($x_1$, $y_1$) and ($x_2$, $y_2$) are the two foci and $d_r$ is sum of the distance from the two foci to any point on the ellipse. A set of image points (*x*, *y*) which lies on an ellipse can be defined by a relation, *f*, as:

$$f((x_1,y_1,x_2,y_2,d_r),(x,y)) = \sqrt{(x-x_1)^2+(y-y_1)^2} + \sqrt{(x-x_2)^2+(y-y_2)^2} - d_r = 0 \qquad (3)$$

By using the polling search process, the higher values of voting can be recognized as more possible ellipses. The individual voting value is stored into 5-dimensional accumulator array [$x_1$][$y_1$][$x_2$][$y_2$][$d_r$], correspondingly. Those ellipses with higher value of voting will be queued into a candidate list according to their voting ratio, number of votes divided by number of ellipse periphery pixels. That is to say, for this five-parameter transform space, the transformation requires intensive computation and a large amount of memory.

In the operation on back-projection of the five-dimensional parameter space to a set of images, assumed the ellipse is existed, we may start from the standard-form equation (4) an ellipse centered at the origin (0,0):

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad (4)$$

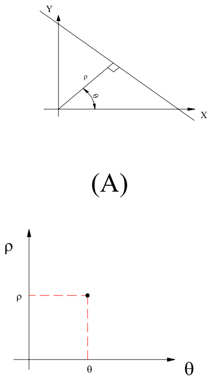
where *x* and *y* represent X-Y Cartesian coordinates of any point on the ellipse, *a* and *b* are the major and minor axis of the ellipse, respectively. This is an ellipse whose foci are on the *X-axis*. Basic properties about this ellipse are:

Center-to-focus distance: $c = \sqrt{a^2 - b^2}$

Foci: $(\pm c, 0)$

Vertices: $(\pm a, 0)$

Fig. 1. Line equation $\rho = y\sin\theta + x\cos\theta$ . (A) The normal parameters for a line in a Cartesian coordinates. (B) Representation in parameter space.



(A)

(B)

Equation (4) can also be expressed in polar coordinates by substituting $x = r{\cdot}cos\theta$ and $y = r{\cdot}sin\theta$. After rearranging, the equation can be derived as:

$$r = \pm\sqrt{\frac{a^2 b^2}{b^2\cos^2\theta + a^2\sin^2\theta}} \qquad (5)$$

where *r* (choosing the positive value) denotes the distance between a point *A* on the ellipse and the central locus point *C* of the ellipse, and $\theta$ is the angle between the *X-axis* and the vector $\overline{CA}$ . Equation (5) is an extension of the standard parametric Hough transform for ellipses. For a determined ellipse from above voting strategy, the corresponding values of the accumulator cell are: $x_1$, $y_1$, $x_2$, $y_2$ and $d_r$. The equations can be derived for the parameters of the ellipse:

$$c = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{2} \qquad (6)$$

$$a = \frac{d_r}{2} \qquad (7)$$

$$b = \sqrt{a^2 - c^2} = \sqrt{(\frac{d_r}{2})^2 - c^2} \qquad (8)$$

$$x = r * \cos\theta = \frac{d_r}{2}\sqrt{\frac{{d_r}^2 - 4c^2}{{d_r}^2 - 4c^2\cos^2\theta}} \qquad (9)$$

$$y = r*\sin\theta = \frac{d_r \sin\theta}{2}\sqrt{\frac{d_r^{\,2}-4c^2}{d_r^{\,2}-4c^2\cos^2\theta+d_r^{\,2}\sin^2\theta}} \tag{10}$$

So every point P($x$, $y$) on the ellipse can be derived from equations (9) and (10). But not every ellipse is as ideal as the standard-form ellipse. In general, the ellipse may not be centered on origin (0, 0) and its major axis is neither horizontal nor vertical. If the rotation angle α is counterclockwise from X-axis to its major axis, the angle α may be calculated as:

$$\alpha = \tan^{-1}\left(\frac{y_2-y_1}{x_2-x_1}\right) \tag{11}$$

Finally, every point P($x'$, $y'$) on X-Y plane for an ellipse with ($x_1, y_1, x_2, y_2, d_r$) parameter can be described as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}\begin{bmatrix} r*\cos\theta \\ r*\sin\theta \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{12}$$

After introducing the values in equations (9), (10) and (11) to (12), all points on the ellipse can be easily derived by computing $\theta$ from 0° to 359°.

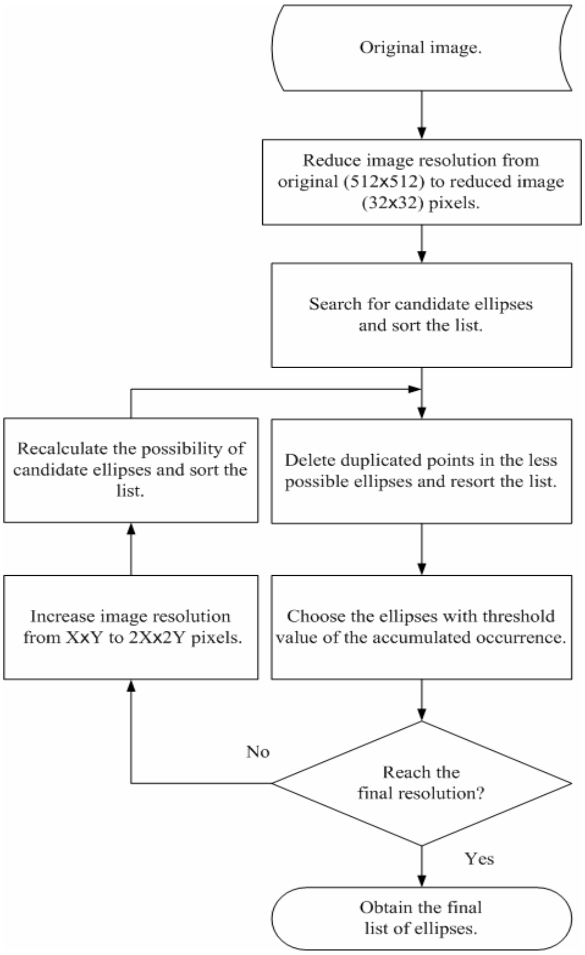
## 3. HIERARCHICAL PYRAMID IMAGE

From above description, the application of the Hough transform to detect ellipses requires a five-dimensional array of accumulators. To reduce computational complexity, there are many software solutions have been provided in the past. This paper describes a divide-and-conquer method based on pyramid image and dynamically quantization to reduce image resolution [16, 18, 23]. This method does not require the search of every pixel on image. It performs the subdivision and subsequent “vote counting” only on ellipses with votes exceeding a pre-given threshold. So it is feasible to localize edges quickly by finding apparent edges in the upper levels of the image pyramid. The succeeding lower layer is then searched to localize fitting better edges after increasing image resolution for each time. By using this hierarchical approach, the data processing can be led into a significant reduction in both computational complexity and memory requirement. This new algorithm developed in this research is mainly implemented with the elliptical Hough transform. The application of this algorithm consists of three steps: (1) reducing image resolution, (2) exhaustive searching for ellipses, and (3) increasing image resolution iteratively and searching for the most moderate ellipses. Fig. 2 shows the flow chart of the 3-step algorithm. And more detailed description is as bellow:

Step 1: Reducing image resolution

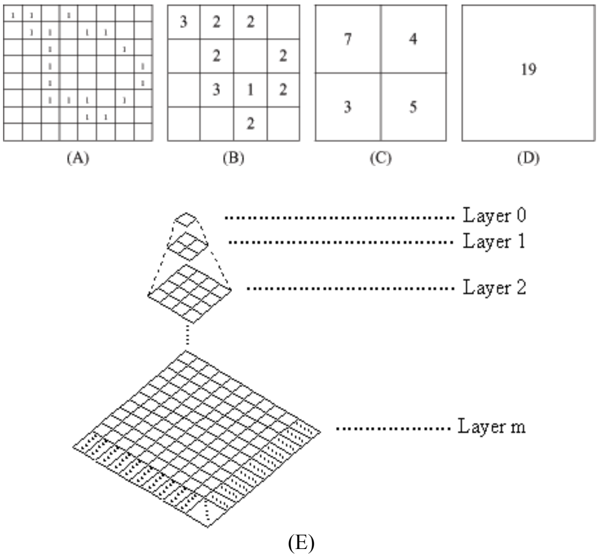
At first, the 512×512 image is saved and a lower resolution image is generated from the original resolution image. In this research we set the lowest resolution image to 32×32 pixels, the initial condition of exhaustive searching for ellipses. While reducing image resolution, the number of the original image object pixels is accumulated in corresponding cell on upper layer. Operating elliptical Hough transform only via object boundary pixels benefits in decreasing the computation. The method of data abstract in down sampling image is described in Fig. 3. From fine to coarse procedure, (A) to (D), we merge both horizontal and vertical dimensions, which yields 4 data squares into one square (of size 2×2). The number for each data square is sum of the corresponding original previous level object pixels. The fine-to-coarse operation may be named as bottom-up procedure.

Fig. 2. Flowchart of the elliptical Hough transform to search for ellipses in an image.


Original image.
Reduce image resolution from original (512x512) to reduced image (32x32) pixels.
Search for candidate ellipses and sort the list.
Recalculate the possibility of candidate ellipses and sort the list.
Delete duplicated points in the less possible ellipses and resort the list.
Increase image resolution from XxY to 2Xx2Y pixels.
Choose the ellipses with threshold value of the accumulated occurrence.
No
Reach the final resolution?
Yes
Obtain the final list of ellipses.

The original image resolution is $n$×$n$, the finest one, and the coarsest image is 1×1 size. If n is an integral power of 2, $n$=$2^m$, then this pyramid architecture will be a stack of m arrays of images of sizes $2^m\times2^m$, $2^{m-1}\times2^{m-1}$, $2^{m-2}\times2^{m-2}$,…, 2×2, 1×1. The height of the pyramid, the number of levels, is $m+1 = (\log_2 n)+1$. Sum of all the series of pyramid image cells are $(4/3)n^2-1/3$. In practice, it is not necessary to consider each cell during searching. So the total number of cells needed to be computed is far less than (4/3)$n^2$, in fact. However in determining the pixel is object pixel or not, we must examine every image pixel, $n\times n$ pixels. Obviously, the complexity of this stage for an n×n image is $\Theta(n^2)$.

Fig. 3. Image down sampling from fine to coarse resolutions, (A) level $i$ (original image 8×8), (B) level $i$+1 (4×4), (C) level $i$+2 (2×2) and (D) level $i$+3 (1×1). The number in each square is the sum of original object pixels. (E) Presentation of pyramid image data structure.



Step 2: Exhaustive searching for ellipses

Follows by the previous step all the object pixels have been recorded in a temporary array. Then we try every object pixels for two undetermined loci ($x_1$, $y_1$) and ($x_2$, $y_2$) of a possible ellipse on that image. The sum of distance $d_r$ can be determined from equation (2). In a $k\times k$ image, the value of $x_1$, $y_1$, $x_2$, $y_2$ all are in the range from 0 to k. Then the computational effort of standard Hough transform for ellipse may be represented as $m_k\times C'_k\times k^4$, where $m_k$ is the number of contour object pixels, $C'_k$ is a computing constant. Though usually $m_k$ is far less than $k$, $m_k$ is varied accompany with the image resolution. So $m_k$ has a worst-case running time of $\Theta(k)$. Now the expression for computing complexity in exhaustive searching for ellipse can be derived as $C'_k\times k^5$. In the brute-force searching under the lower resolution image, such as 1×1, 2×2, and 4×4, the searching seems to be not necessary. So we start the searching from the resolution 32×32. For most cases, we may start from the middle layer, $k = 2^{\left\lfloor \frac{m}{2} \right\rfloor}$ or $k = 2^{\left\lfloor \frac{m}{2} \right\rfloor +1}$. When n and m grows larger, we may only consider $k = 2^{\frac{m}{2}}$ and assume $\left\lfloor \frac{m}{2} \right\rfloor = \frac{m}{2}$. The complexity of the brute-force searching is evidently bounded by $\Theta((2^{m/2})^5) = \Theta((2^m)^{5/2}) = \Theta(n^{5/2})$. The possibilities of all ellipses are accumulated in a five-dimensional array in the form of Possibility($i$)=Array[$X_1$][$Y_1$][$X_2$][$Y_2$][$D_r$]. A threshold value is set to filter out the less possible ellipses according to the occurrence of boundary. Then the candidate ellipses are determined and ordered descendant by the value of the contour edge occurrence that is the sum of original object pixels divided by the corresponding ellipse perimeter, expressed as Possibility($i$). In this step, after exhaustive searching by introducing all loci set [$X_1$][$Y_1$][$X_2$][$Y_2$], each object pixel is inevitable to vote for many possible ellipses. But it is illogical that a boundary pixel belongs to many objects. So after exhaustive searching, the candidate array must be reordered for one-man one-vote. Whenever a pixel belongs to a prior rank or determined ellipse, it should be removed and not to vote again in the same voting stage. Then the edge occurrence is recalculated and the rank is reordered for all candidate ellipses after that ellipse.

Step 3: Increasing image resolution iteratively and searching for the most moderate ellipses

In the previous step, a sorted candidate ellipses array is obtained in the lower-resolution image. The resolution of the image is then operated increasingly step by step until a predefined condition or the original image resolution reached. Increasing the image resolution also increases the uncertainty of the positions and dimensions of ellipses that is mainly resulted from quantization error. To get the most optimal candidate ellipse, a candidate ellipse is supposed to be searched in the neighborhood for each different image resolution stage. If a selected ellipse has parameter [$X_1$][$Y_1$][$X_2$][$Y_2$][$D_r$] in an image of resolution $k\times k$, in previous stage, its reasonable distribution ranges should be:

$$\begin{cases} X_1-0.5 \le x_1 < X_1+0.5 \\ Y_1-0.5 \le y_1 < Y_1+0.5 \\ X_2-0.5 \le x_2 < X_2+0.5 \\ Y_2-0.5 \le y_2 < Y_2+0.5 \\ Dr-0.5 \le d_r < Dr+0.5 \end{cases} \tag{13}$$

where $X_1$, $Y_1$, $X_2$, $Y_2$, and $Dr$ are integers after rounding off real number $x_1$, $y_1$, $x_2$, $y_2$, and $d_r$, the real value, respectively. To make a clarified demonstration we illustrate with a digitized 2D image as shown in Fig. 4. The position of an object pixel $P$ in digital location is exactly on the intersection of horizontal and vertical lines as shown in Fig. 4(A). Its possible distribution in continuous spatial domain is within the region enclosed by dashed square as in Fig. 4(B). Then we increase image resolution from 4×4 to 8×8 pixels. Now the possible position of $P$, enclosed by dashed square, is shown in Fig. 4(C). And the newly ellipse parameters displayed in equation (13) may be deduced as:

$$\begin{cases} 2X_1\text{-}1 \le x'_1 < 2X_1+1 \\ 2Y_1\text{-}1 \le y'_1 < 2Y_1+1 \\ 2X_2\text{-}1 \le x'_2 < 2X_2+1 \\ 2Y_2\text{-}1 \le y'_2 < 2Y_2+1 \\ 2Dr\text{-}1 \le d'_r < 2Dr+1 \end{cases} \tag{14}$$

where $x'_1$, $y'_1$, $x'_2$, $y'_2$, and $d'r$ are new parameter values in resolution 8×8. And the integers $X'_1$, $Y'_1$, $X'_2$, $Y'_2$, and $D'r$ are rounding off $x'_1$, $y'_1$, $x'_2$, $y'_2$, and $d'r$ , respectively. They are ranged variously in:

$$\begin{cases} X'_1 \in \{2X_1, 2X_1 \pm 1\} \\ Y'_1 \in \{2Y_1, 2Y_1 \pm 1\} \\ X'_2 \in \{2X_2, 2X_2 \pm 1\} \\ Y'_2 \in \{2Y_2, 2Y_2 \pm 1\} \\ D'r \in \{2D_r, 2D_r \pm 1\} \end{cases} \tag{15}$$

However, it should be only one ellipse inherited from previous upper level image. So the ellipse must be searched and chose for the most outstanding candidate in these variables. This step can be considered as adjusting the temporarily previous searched ellipse to more precise positions and dimensions. The locations and dimensions of the ellipse are slightly "shaken" and "shrunk and enlarged" to fit the newly modified ellipse container. The succeeding ellipses parameters are calculated by substituting the 5 parameters, from the previous step, into equations (9) to (12). Then we select the most probable candidate as the winner. This ellipse optimization takes constant computing time $C_s$, without lost generality. If an ellipse contour has $n_p$ pixels, it takes computing time $n_p \times C_s$. The computation obviously needs about $(2n_p) \times 3^5 \times C_s$ for one ellipse in progressing into succeeding level image, such as from Fig. 4(B) to 4(C). The same as before, the height of the pyramid is $m=\log_2 n$. It is assumed that there are $n_p$ pixels on each ellipse periphery in average. And $n_p$ is varied accompany with the image resolution, $n_p$ has a worst-case running time of $\Theta(n)$. So if there were $n_0$ ellipses on an image, without loss generality, the computation can be calculated as:

$$\begin{aligned} & C_{total} \\ & = n_0 \times n_p \times C_s + n_0 \times (2n_p) \times 3^5 \times C_s + n_0 \times (4n_p) \times 3^5 \times C_s + \ldots. + n_0 \times (2^m n_p) \times 3^5 \times C_s \\ & = n_0 \times n_p \times C_s + n_0 \times n_p \times C_s \times 3^5 (2^1 + 2^2 + \ldots. + 2^m) \\ & = n_0 \times n_p \times C_s + n_0 \times n_p \times C_s \times 3^5 \times 2 \times (2^m - 1) \\ & = n_0 \times n_p \times C_s (1 + 3^5 \times 2^{m+1} - 2 \times 3^5) \\ & = \Theta(n) \times \Theta(2^{m+1}) = \Theta(n) \times \Theta(2n) = \Theta(n^2) \end{aligned} \tag{16}$$

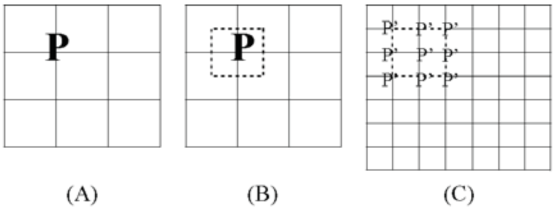
The iterative search step ends at the resolution of 512×512 pixels in this research. So the computation is strictly bounded by $\Theta(n^2)$.

In this research, the 3-step algorithm to search for ellipse, the computing complexity can be calculated by sum of these three steps:

$$T(n) = \Theta(n^2) + \Theta(n^{5/2}) + \Theta(n^2) = \Theta(n^{5/2}) \tag{17}$$

Thus the average computing complexity of this new algorithm is $\Theta(n^{5/2})$.

Fig. 4. (A) A pixel *P* in digitized image with resolution 4×4. (B) Pixel *P*'s probable positions range in continuous space. (C) Pixel *P*'s probable positions in digitized image with resolution 8×8.



## 4. EXPERIMENTS AND ANALYSES

In this research, the image-processing program is developed using Borland C++ Builder under the Microsoft Windows operating system. All data are operated on an Intel Pentium 4 Celeron 2.4GHz PC. Elliptical Hough transform searching for ellipses with this 3-step approach is performed well. It has been proved successful for detecting real-world images with vegetable seedlings ellipse-like leaves [3, 4]. In this research we mainly devote on the discussions of standard ellipses for verifying the performance and feasibility of this 3-step hierarchical pyramid elliptical Hough transform algorithm. Two-hundred images are randomly produced, twenty of which contained one ellipse, twenty with two ellipses, etc. up to twenty images with ten ellipses. Figure 5 shows two typical images, one with 5 ellipses and the other with 10 ellipses. The ellipses often intersected especially on more than 3 or 4 ellipses images. The missing or false alarm errors in more ellipses images are inevitable. We further extend the 3-step approach to a multi-pass method for searching ellipses to enhance the performance. This modified algorithm for detecting ellipses is operated as shown in Figure 6. With this strategy, the more outstanding ellipses are picked out in fore coming pass. As long as the ellipse is recognized, the object pixels belong to that ellipse will be erased and not to vote again. So the pixels in afterward passes needed to be considered will be less than previous pass usually. And the searching process will be speeded up than previous pass. This multi-pass searching method terminates at no more ellipses being found.

Fig. 5. Typical test images, (A) 5 ellipses image, (B) 10 ellipses image.

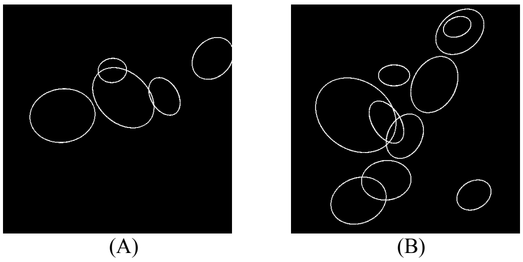
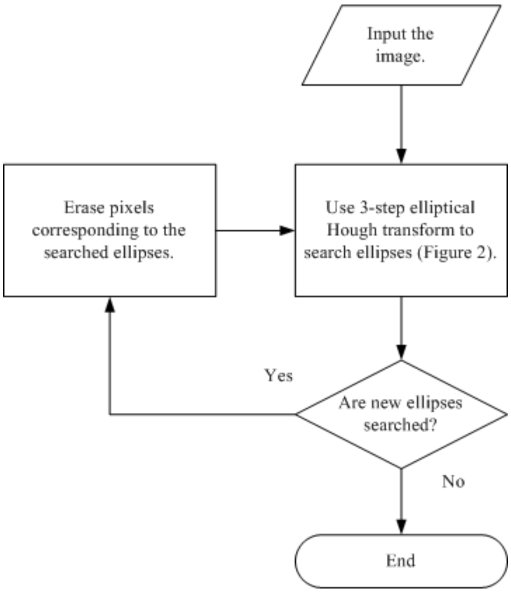


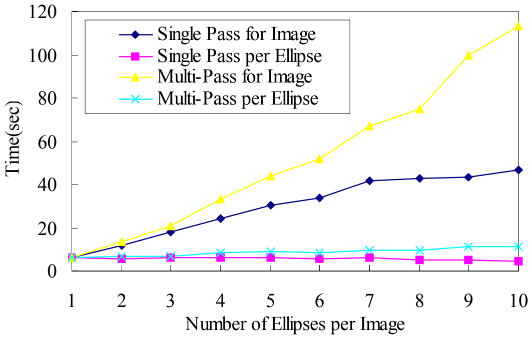
Fig. 6. Flow chart of the multi-pass elliptical Hough transform algorithm for detecting ellipses.


Input the image.
Erase pixels corresponding to the searched ellipses.
Use 3-step elliptical Hough transform to search ellipses (Figure 2).
Yes
Are new ellipses searched?
No
End

### 4.1. Computation time of the algorithm

Figure 7 shows the average computation time (in seconds) both per image and per ellipse against the number of ellipses in the image. The multi-pass approach executes more one pass than single pass at least. However, in fewer ellipses image the difference in time cost is not significant. On a simple image the ellipse is easy to be recognized on single pass or the first pass. There are not too many object pixels remained after removing the ellipse(s) on the image. The execution after the first pass just searches few pixels that belong to some part of ellipse segment. So the null search does not take much time, whereas the difference in time cost on more ellipses image becomes significant. In a more complex image, the ellipse searching usually takes several passes. So it is reasonable that multi-pass searching on more ellipses image take more time than fewer ellipses image. It can be observed in Figure 7 both in image and per ellipse case, the tendency is towards upward slightly. The computation time is from about 6 to 11.3 seconds per ellipse, from one ellipse image to 10 ellipses image. But in the single pass the cost time becomes less from 7 ellipses image to 10 ellipses image. Especially the decline tendency is obvious in per ellipse analysis. Too many object pixels to search for ellipses. It makes confuse for voting and missing real ellipses in low resolution stage. As long as the ellipse searching fails in lower resolution image, the ellipse is surely lost during that pass. And it is no way to find it in higher resolution stage during that pass. This low rate of successful searching, such as 5 ellipses are found on a 10-ellipse image, takes less time than it should be is doubtless. So it is easy to find that time cost per ellipse is disproportional with the number of ellipses in image.

Fig. 7. Average computation time in seconds.



### 4.2. Accuracy of the algorithm

Figure 8 plots the percentage of ellipses correctly found against the number of ellipses in the image. Single pass approach in simple image, such as 1 or 2 ellipses image, the accuracy is satisfied. However, in more complex image, such as more than 5 ellipses image, the accuracy tendency is declined obviously. Especially in a 10 ellipses image, the accuracy is only about 40%. In the multi-pass approach the accurate ratio always keep in a satisfied high level. The accuracies all are about 90% except in 9 ellipses image, about 80%. The tendency of the multi-pass approach accuracy is downward slightly in more ellipses image. It is obvious that this algorithm can be applied robust and highly effective on a complex image. Figure 9 shows the total numbers of false alarm, false ellipses, found in each set of images. Usually, the false alarms may be noisy in lower resolution stage. When searching process proceeds into higher resolution, the false alarm ellipse difficult to receive high voting rate. And on a simpler image, the object pixels are not so many to support for any false ellipse. So the error for false alarm in this research, under the 10 ellipses image, is not significant. Figure 9 displays this benefit about false alarm. Figure 9 shows the false alarm error, the same for single pass and multi-pass. Though more ellipses image is apt to encounter the false alarm, the error is just 2% even in 10 ellipses image.

Fig. 8. Percentage accuracy with multiple ellipses.

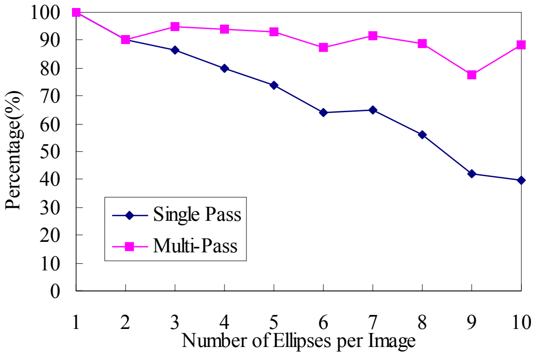
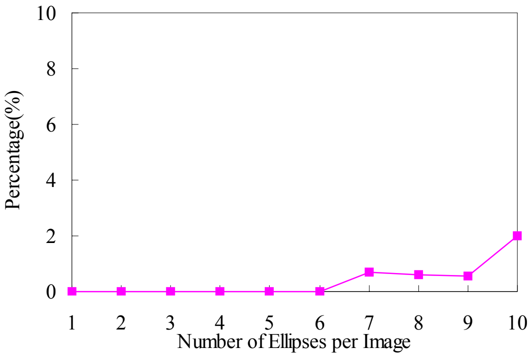


Fig. 9. Percentage of false alarms.



## 5. CONCLUSIONS

In this research we successfully develop a hierarchical pyramid elliptical Hough transform algorithm for searching ellipses in 512×512 resolution image. The computing complexity of this algorithm is $\Theta$ ($n^{5/2}$). Apparently this new approach has very distinctive and outstanding advantages. It is easy to understand and implement with. And it is proved to be efficient and robust. By using multi-pass 3-step elliptical Hough transform, the computation time is from about 6 to 11.3 seconds per ellipse, on one ellipse image to 10 ellipses image. And the accuracies all are about 90% except in 9 ellipses image, about 80%. The significant reduction of computational load and high successful rate by using this algorithm makes the 5-dimensional elliptical Hough transform applicable in most practical applications.

## 6. REFERENCES

[1] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition* 13(2), pp. 111-122, 1981.

[2] G. Bongiovanni, C. Guerra, S. Levialdi, "Computing the Hough Transform on a Pyramid Architecture," *Machine Vision and Applications* 3, pp. 117-123, 1990.

[3] C.F. Chien, and T.T. Lin, "Leaf Area Measurement of Selected Vegetable Seedlings Using Elliptical Hough Transform," *Transactions of the ASAE* 45(5), pp. 1669-1677, 2002.

[4] C.F. Chien, *Non-Destructive Measurement of Selected Vegetable Seedlings Using 3D Machine Vision*, Ph.D. diss., Bio-Industrial Mechatronics Engineering Dept., National Taiwan Univ., Taipei, Taiwan, 2003.

[5] S.R. Deans, "Hough Transform from the Radon Transform," *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-3, pp. 185-188, 1981.

[6] R.O. Duda, P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun. ACM* 15 (1), pp. 11-15, 1972.

[7] C. Espinosa, M.A. Perkowski, "Hierarchical Hough Transform Based on Pyramidal Architecture," *IPCCC '92*, pp. 0743-0750, 1992.

[8] M. Goldberg, L. Wang, "Comparative Performance of Pyramid Data Structures for Progressive Image Transmission," *IEEE Transactions on Communications* 39 (4), pp. 540-548, 1991.

[9] N. Guil, and L. Zapata, "Lower Order Circle and Ellipse Hough Transform," *Pattern Recognition* 30(10), pp. 1729-1744, 1997.

[10] W.D. Hofmann, D.E. Troxel, "Making Progressive Transmission Adaptive," *IEEE Transactions on Communications* 34 (8), pp. 806-813, 1986.

[11] P.V.C. Hough, "Method and Means for Recognizing Complex Patterns," *U.S. Patent* 3069654, 1962.

[12] J. Illingworth, J. Kittler, "The Adaptive Hough Transform," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* VOL. PAMI-9, NO. 5, pp. 690-698, 1987.

[13] J. Illingworth, J. Kittler, "A Survey of the Hough Transform," *CVGIP* 44, pp. 87-116, 1988.

[14] J.M. Jolion, A. Rosenfeld, "An O(log n) Pyramid Hough Transform," *Pattern Recognition Letters* 9, pp. 343-349, 1989.

[15] C. Kimme, D. Ballard and J. Sklansky, "Finding Circles by An Array of Accumulators," *Commun. ACM* 18(2), pp. 120-122, 1975.

[16] V.F. Leavers, "Which Hough Transform?" *CVGIP: IMAGE UNDERSTANDING* 58 (2), pp. 250-264, 1993.

[17] Y. Lei, K.C. Wong, "Ellipse Detection Based on Symmetry," *Pattern Recognition Letters* 20, pp. 41-47, 1999.

[18] H. Li, M.A. Lavin, R.J. Le Master, "Fast Hough Transform: A Hierarchical Approach," *CVGIP* 36, pp. 139-161, 1986.

[19] R.A. McLaughlin, "Randomized Hough Transform: Improved Ellipse Detection with Comparison," *Pattern Recognition Letters* 19, pp. 299-305, 1998.

[20] P. Meer, “Stochastic Image Pyramids,” *Computer Vision, Graphics and Image Processing* 45, pp. 269-294, 1989.

[21] H.K. Muammar, M. Nixon, “Tristage Hough Transform for Multiple Ellipse Extraction,” *IEE proceedings-e* 138 (1), pp. 27-35, 1991.

[22] M.A. Rashwan, M.S. Elsherif, A.M. Elsayad, “Pyramid Data Structures for On-Line Image Progressive Transmission,” 36th., *Mid-West Symposium on Circuits and Systems*, pp. 103-106, 16-18 Aug. 1993.

[23] J.A. Richards, X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, third ed., Springer Verlag, Berlin, 1999.

[24] A.A. Sewisy, F. Leberl, “Detection Ellipses by Finding Lines of Symmetry in the Images via An Hough Transform Applied to Straight Lines,” *Image and Vision Computing* 19, pp. 857-866, 2001.

[25] K.R. Sloan, S.L. Tanimoto, “Progressive Refinement of Raster Images,” *IEEE Transactions on Computers* C-28 (11), pp. 871-875, 1979.

[26] G.B. Thomas, R.L. Finney, *Calculus and Analytic Geometry*, ninth ed., Addison-Wesley Publishing Company, Inc., Reading, MA, 1996.

[27] S. Tsuji, F. Matsumoto, “Detection of Ellipses by A Modified Hough Transformation,” *IEEE Transactions on Computers* C-27(8), pp. 777-781, 1978.

[28] L. Uhr (Ed.), *Parallel Computer Vision*, Academic Press, Boston, 1987.

[29] L. Wang, M. Goldberg, “Progressive Image Transmission Using Vector Quantization on Images in Pyramid Form,” *IEEE Transactions on Communications* 37 (12), pp. 1339-1349, 1989.