ELLIPSE DETECTION WITH HOUGH TRANSFORM IN ONE DIMENSIONAL PARAMETRIC SPACE

Alex Yong Sang Chia, Maylor K. H. Leung

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798

How-Lung Eng, Susanto Rahardja

Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613

ABSTRACT

The main advantage of using the Hough Transform to detect ellipses is its robustness against missing data points. However, the storage and computational requirements of the Hough Transform preclude practical applications. Although there are many modifications to the Hough Transform, these modifications still demand significant storage requirement. In this paper, we present a novel ellipse detection algorithm which retains the original advantages of the Hough Transform while minimizing the storage and computation complexity. More specifically, we use an accumulator that is only one dimensional. As such, our algorithm is more effective in terms of storage requirement. In addition, our algorithm can be easily parallelized to achieve good execution time. Experimental results on both synthetic and real images demonstrate the robustness and effectiveness of our algorithm in which both complete and incomplete ellipses can be extracted.

***Index Terms*—** Ellipse detection, Hough transform, Shape recognition.

1. INTRODUCTION

Extracting elliptical objects from digital images is of fundamental importance in shape recognition [1]. One of the best known methods in extracting ellipses from images is the Hough Transform (HT). The key concept behind the standard HT in extracting ellipses is to define a mapping between the two dimensional image space and the five dimensional parameter space. These five parameters are the coordinates of the center point of the ellipse, the lengths of the major and minor axes of the ellipse and the orientation of the major axis with respect to the *x*-axis. Each data point of the image space is map onto specific cells of the five dimensional accumulator, whereby the associated parameters of the specific cells are chosen such that the curve defined by these parameters passes through the data point. In this aspect, the data points can be seen as voting for the parameters of the ellipses found in the image. The votes in the cells are then accumulated. After all data points of the image have been considered, the local maxima of the accumulator correspond to the parameters of the ellipses that are detected in the image.

The main advantage of the HT in extracting ellipse is its robustness against discontinuous or missing data points. This is because the HT does not require the connectivity of all the contour points of an ellipse. Owing to this, the HT is well suited to detect ellipses in the presence of moderate noise or in images having a cluttered background. Unfortunately, the requirement for the five dimensional accumulator places huge computational and storage constrains and hence precludes practical applications. In light of this, many algorithms have been developed to retain the original advantages of the HT while minimizing the computational complexity. For example, S. Tsuji and F. Matsumoto [2] first decomposed the five dimensional parameter space and then used the symmetrical properties of the ellipse to reduce the computation complexity. More recently, N. Guil and E. L. Zapata [3] proposed the Fast Ellipse Hough Transform which achieved better execution time. However, these methods require accurate calculation of the gradients and tangents of the edge pixels. As such, the detection accuracy of ellipses using such methods will be adversely affected by the image noise.

To avoid the use of the gradient and tangent information, Xu *et al*. [4] developed the Randomized Hough Transform. In this method, three non collinear edge pixels are randomly selected and then used to vote on the parameters of the ellipse. This method was further extended in [5] in which four random pixels are selected at each iteration. Unfortunately, the accuracy and speed of these randomized HT algorithms is dependant on the number of edge pixels. Furthermore, there is an additional difficulty in the accurate estimation of the probability which is inherent in these two methods.

Many other methods which exploit the geometrical symmetry of the ellipse have also been proposed to avoid the calculation of the gradient and tangent of the edge pixels. For example in [6], C. T. Ho and L. H. Chen used the global geometric symmetry to locate all possible parameters of the ellipses. This idea was extended in [7] in which the parameter space is reduced by using two symmetric axes. Although these methods are able to extract ellipses accurately, they still require high computational complexity. Furthermore, despite using the geometric properties of the ellipse, these methods still demand at least a two dimensional accumulator.

The key contribution of this paper is to present a novel and robust ellipse detection algorithm which uses an accumulator that is only *one* dimensional. As a result of this reduction in accumulator size, our algorithm is more efficient in term of storage requirement than the current generation of HT based ellipse detection algorithms. Another distinctive aspect of our algorithm is that we avoid the calculation of the gradients and tangents of the edge pixels. Thus our algorithm is more robust to image noise. As an initial preview, we also point out here that our algorithm can be easily parallelized to achieve good execution time. We are aware of only one other ellipse detection algorithm that uses an accumulator that is one dimensional [8]. However, we point out here that there are marked differences between our work and theirs. More specifically, we exploit the foci of an ellipse to derive *precisely* the parameters of a hypothetical ellipse. On the other hand, in [8], Y. Xie and Q. Ji *estimate* the half-length of the minor axis of a hypothetical ellipse to approximate the other parameters of the ellipse. In this aspect, the parameters of the ellipses are more accurately calculated by our method. In addition, in this work, we determine the validity of a hypothetical ellipse by relating the circumference of the ellipse to the number of pixels that vote for the ellipse. In contrast, Y. Xie and Q. Ji used the local maxima of the accumulator to determine a hypothetical ellipse's validity. Finally, experimental results on both synthetic and real images demonstrate the efficiency and robustness of our algorithm, in which we are able to extract both complete and incomplete ellipses when the end points of the major axes are available. In contrast, Y. Xie and Q. Ji are not able to detect incomplete ellipses present in their real images.

2. INTRODUCTORY EXAMPLE

Consider the ellipse shown in Fig. 1a. Let point o be the center position of the ellipse and (o_x, o_y) denotes the coordinates of point o. Let α and β be defined as the half-lengths of the major and minor axes respectively. We denote the angle the major axis made with the x-axis as θ. For any arbitrary ellipse, if we know the values of $\{o_x,\ o_y,\ \alpha,\ \beta,\ \theta\}$, we can completely define this ellipse.

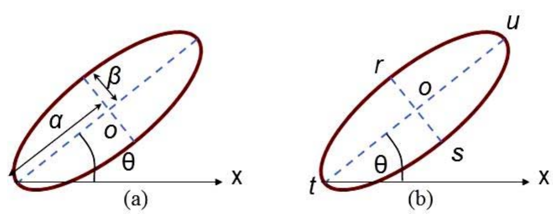

Fig. 1. a) Arbitrary ellipse in which point o denotes the center position of the ellipse, α and β denote the half-lengths of the major and minor axes respectively and θ defines the angle the major axis made with the x-axis. b) Another representation of the ellipse of Fig. 1a in which $l_{t,u}$ and $l_{r,s}$ denote the major and minor axes respectively.

We next consider the same ellipse now presented in Fig. 1b. Let $l_{p,q}$ denotes a line segment whose end points are point p and point q. In this case, $l_{r,s}$ and $l_{t,u}$ denote the minor and major axes of the ellipse respectively. Suppose the edge points t and u of this ellipse is available from the image. Therefore by using (t_x, t_y) and (u_x, u_y), we will be able to derive the values of $\{o_x,\ o_y,\ \alpha,\ \theta\}$ (See Sect. 3). Following this, since only the half-length of the minor axis β is unknown, we can use the remaining edge points of the image to vote for β. Consequently, since we need to vote on only the half-length of the minor axis, we will thus require a one dimensional accumulator.

3. DERIVATION OF ELLIPSE PARAMETERS

Consider an arbitrary ellipse shown in Fig. 2. We denote the foci of the ellipse as points w and v and the center position of the ellipse as point o. Given the values of (t_x, t_y) and (u_x, u_y), we can readily derive $\{o_x,\ o_y,\ \alpha,\ \theta\}$ as follows:

$$o_x = \frac{t_x + u_x}{2} \quad (1)$$

$$o_y = \frac{t_y + u_y}{2} \quad (2)$$

$$\alpha = \frac{\sqrt{(u_x - t_x)^2 + (u_y - t_y)^2}}{2} \quad (3)$$

$$\theta = \tan^{-1}\left(\frac{u_y - t_y}{u_x - t_x}\right) \quad (4)$$

We proceed to derive the half-length of the minor axis β. Let point k be an arbitrary point on the contour of the ellipse. Since

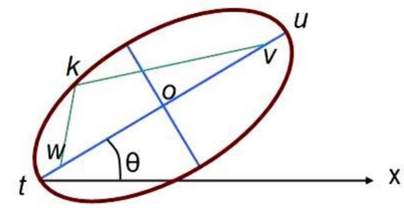

Fig. 2. a) Arbitrary ellipse in which points w and v denote the foci of the ellipse and point o denotes the center position of the ellipse.

points w and v are the foci of the ellipse, therefore the sum of the lengths of line segments $l_{w,k}$ and $l_{k,v}$ can be calculated by equ. (5)

$$\sqrt{(k_y - w_y)^2 + (k_x - w_x)^2} + \sqrt{(k_y - v_y)^2 + (k_x - v_x)^2} = 2\alpha \quad (5)$$

where

$$w_x = o_x - \cos|\theta|\sqrt{\alpha^2 - \beta^2} \quad (6)$$

$$w_y = o_y - \sin|\theta|\sqrt{\alpha^2 - \beta^2} \quad (7)$$

$$v_x = o_x + \cos|\theta|\sqrt{\alpha^2 - \beta^2} \quad (8)$$

$$v_y = o_y + \sin|\theta|\sqrt{\alpha^2 - \beta^2} \quad (9)$$

Therefore given an arbitrary point on the contour of the ellipse, we can derive the value of β using eqns. (5)-(9) as follows:

$$\beta = \sqrt{\frac{\alpha^2\delta^2 - \alpha^2\gamma^2}{\alpha^2 - \gamma^2}} \quad (10)$$

where

$$\delta = \sqrt{(k_y - o_y)^2 + (k_x - o_x)^2} \quad (11)$$

$$\gamma = \sin|\theta|\,(k_y - o_y) + \cos|\theta|\,(k_x - o_x) \quad (12)$$

4. ALGORITHM DESCRIPTION

The above derivations provide the fundamental framework of our ellipse detection algorithm. We first consider every pair of edge pixels as possible end points of the major axis of a hypothetical ellipse. Using eqns. (1)-(4), we can then calculate the values of $\{o_x,\ o_y,\ \alpha,\ \theta\}$ of the hypothetical ellipse. Following that, all other edge pixels will be used to vote on the half-length of the minor axis β of this hypothetical ellipse. Although we can use the local maxima of the accumulator to determine the possible half-lengths of the minor axes, we propose here a refinement that is able to better integrate the number of edge pixels that vote for the hypothetical ellipse with the number of edge pixels that is required to define a complete ellipse.

For each hypothetical ellipse, we use the calculated values of α and β to compute the circumference of the ellipse [9]. An ellipse is detected if the number of edge pixels that vote for this ellipse is greater than $RelativeVote_{min}\times$ *Circumference of Ellipse*, where $0 < RelativeVote_{min} \leq 1$. In this aspect, setting $RelativeVote_{min}$

$= 1$ will give detection of an ellipse only if all the contour points of the ellipse can be found. In our experiments on synthetic and real images, since we are dealing with incomplete ellipses, we set $RelativeVote_{min}$ to be between 0.2 and 0.45.

We point out here that our algorithm can be readily parallelized to obtain good execution time. In our parallel implementation, each slave processor investigates the feasibility of the parameters of a subset of hypothetical ellipses. The complexity of our algorithm is dominated by the voting phase and has a worst case runtime performance of $O(\frac{n^3}{p})$ where n and p denote the number of edge points and the number slave processors respectively. This complexity can be reduced in future by using feature points selected from a curve instead of using all edge points as in our current implementation. Algorithm 1 and 2 detailed the pseudo codes for the master and slave processors respectively.

Input : A two dimensional array, *EdgePixels*, holding the coordinates of the edge pixels.

Output : A five dimensional array, *EP*, holding the values $\{o_x,\ o_y,\ \alpha,\ \beta,\ \theta\}$ of the detected ellipses.

α_{min} = Minimum half-length of the major axis;
RelativeVote$_{min}$ = Relative minimum number of votes required for the assumed ellipse;

• **Initialization**(*EdgePixels*)

numEdgePixels = Number of edge pixels;
count = 0;
for *k=0* ***to*** *numProcessors-1* **do**
  send(&k, &EdgePixels, P$_k$, data$_{tag}$);
  count++;
end

• **Retrieve Ellipse Parameters**(*EdgePixels*, *EP*)

while *count>0* **do**
  recv(&slave, &NewEP, P$_{any}$, result$_{tag}$);
  EP = append(EP, NewEP);
  count- -;
  if *k<numEdgePixels-1* **then**
    send(&k, &EdgePixels, P$_{slave}$, data$_{tag}$);
    count++;
    k++;
  else
    send(&k, &EdgePixels, P$_{slave}$, terminator$_{tag}$);
  end
end

Algorithm 1: Pseudo codes for the master processor.

5. EXPERIMENTAL RESULTS

In our first experiment, we test our ellipse detection algorithm on six synthetic images. We extract the edge pixels with the Canny edge operator. In the first test, we detect the ellipses in an image after extraneous edge pixels have been introduced and correct edge pixels removed from the image. As such, the results of this test provide a good indication of the robustness of our ellipse detection algorithm. These distorted images are shown in Fig. 3a, b and c. In Fig. 3a and b, we manually remove some edge pixels from the complete ellipses. In Fig. 3c, we add salt and pepper noise to the complete ellipse. 5% of the pixels are affected after the adding of the salt and pepper noise. In the second test, we include multiple overlapping

• **Extract Ellipse Parameters**()

recv(&startEdgeIdx, &EdgePixels, P$_{master}$, any$_{tag}$);
while *any$_{tag}$ == data$_{tag}$* **do**
  t = EdgePixels [startEdgeIdx];
  for *i = startEdgeIdx+1* ***to*** *numEdgePixels-1* **do**
    u = EdgePixels [i];
    Clear the one dimensional *accumulator*, *A*, and the array *circumference of assumed ellipse*, *CE*;
    Set points *t* and *u* as the end points of the major axis of a hypothetical ellipse;
    Calculate the values of $\{o_x,\ o_y,\ \alpha,\ \theta\}$ from eqns. (1)-(4);
    if $\alpha \leq \alpha_{min}$ **then**
      continue;
    end
    for *every other edge pixels k* **do**
      $D_{o,k} = \sqrt{(c_x - k_x)^2 + (c_y - k_y)^2}$;
      if $D_{o,k} > \alpha$ **then**
        continue;
      end
      Calculate the value of β from eqn. (10);
      Accumulator[β] += 1;
      CE[β] = Circumference of ellipse with parameters $\{o_x, o_y, \alpha, \beta, \theta\}$;
    end
    Accumulator -= CE$\times$ RelativeVote$_{min}$;
    $\{\beta_i\}$ = find(Accumulator>0);
    NewEP = append(NewEP, $\{o_x, o_y, \alpha, \theta, \{\beta_i\}\}$);
  end
  send(&myid, &NewEP, P$_{master}$, result$_{tag}$);
  recv(&startEdgeIdx, &EdgePixels, P$_{master}$, any$_{tag}$);
end

Algorithm 2: Pseudo codes for the slave processors.

ellipses in the image (Fig. 3 d, e and f). This simulates to an extent the effects of a cluttered environment.

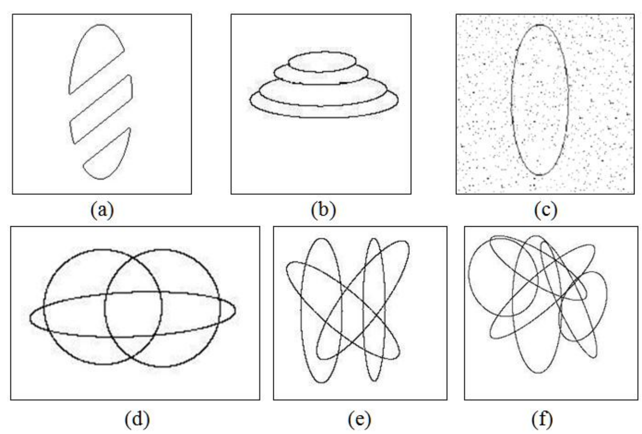

Fig. 3. Synthetic images.

The results of our ellipse detection algorithm on the synthetic images are summarized in Table 1. We evaluate the error $\hat{e}$ of a parameter by dividing the absolute difference between the true and detected values of a parameter by the true value. As seen from Table 1, our algorithm is able to achieve good detection accuracy; the maximum $\hat{e}$ for the images which contain extraneous edge pixels is 1.15% while that for the images which contain multiple overlapping ellipses is 1.07%.

Test	**Percentage error** $\hat{e}$				
	o_x	o_y	α	β	θ
Test 1	0.27%	1.15%	0.34%	0.64%	0.00%
Test 2	0.47%	0.31%	0.78%	1.07%	0.50%

Table 1. Percentage error of our detected ellipse parameters as compared to the true ellipse parameters of the synthetic ellipses.

In our second experiment, we perform ellipse detection on four real images shown in Fig. 4a-d. The edge maps for these images are extracted with the Canny edge operator and are shown in Fig. 4e-h. For the sake of the following discussions, we superimpose the detected ellipses onto the edge maps and show them in Fig. 4i-l.

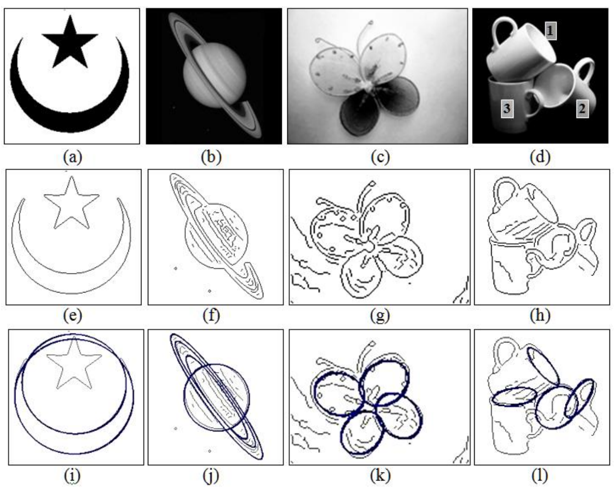

Fig. 4. a)-d) Real images. e)-h) Edges maps that act as input into our ellipse detection algorithm. i)-l) The ellipses which are detected are superimposed onto the edge maps.

Fig. 4a shows an image of a crescent and a star. As seen from its edge map in Fig. 4e, the inner and outer arcs of the crescent can be seen as belonging to the contours of two incomplete ellipses. The result of our algorithm is shown in Fig. 4i. As observed, our algorithm not only correctly detected these two ellipses but also fitted them to the crescent with good accuracy. Fig. 4b shows the photograph of Saturn [10]. The corresponding edge map is shown in Fig. 4f in which we observe that the contours of the ellipses present in this image are not complete. For example, there are missing edge pixels in the arcs of the ring. Despite this, we are still able to extract four perceptually more significant ellipses in the image (Fig. 4j). These detected ellipses correspond to the contour of the planet and the inner and outer arcs of the ring.

Fig. 4c shows the image of a butterfly. Unlike the previous two test images in which the location of the ellipses can be visually detected, in this image the presence of the ellipses is not apparent (Fig. 4g). Nevertheless, our ellipse detection algorithm is still able to fit four ellipses to the four wings of the butterfly. Analysis of the edge map shows that this is because the wing has a curve contour. As such, our algorithm sees the wing as part of an incomplete ellipse.

Of particular interest is the result of our ellipse detection algorithm on the image in Fig. 4d which shows three cups in an arbitrary arrangement. We show the corresponding edge map in Fig. 4h in which the outlines of two ellipses belonging to the mouth of cups 1 and 2 can be seen. As observed in Fig. 4l, our algorithm correctly extracts these two ellipses. More importantly, in the original image of Fig. 4d, we observe that the mouth of cup 3 is partially occluded by the body of cup 1. Despite this, since the location of the major axis of the ellipse can be detected by our algorithm, therefore we are able to determine a perceptually agreeable position for the location of the mouth of cup 3. The result in Fig. 4l also shows the limitation of our algorithm in which the edge points of the major axis must be present in order for our algorithm to detect the ellipse: Consider the handle of cup 1 in which visual inspection shows that the handle corresponds to the contour of an incomplete ellipse. However, since the edge points defining the major axis for this ellipse cannot be located, we are thus not able to fit an ellipse to the handle. However, our algorithm fits an ellipse to the handle of cup 2. Analysis of the edge map shows that although one end point of the major axis for this ellipse is derived correctly from the handle of cup 2, the other end point is derived wrongly from the mouth of cup 2. In this aspect, this ellipse has been erroneously detected.

6. CONCLUSION AND FUTURE WORK

We present a novel ellipse detection algorithm based on HT which uses an accumulator that is only one dimensional. We also show how our algorithm can be easily parallelized to achieve good execution time. We are aware of only one other ellipse detection algorithm that uses a one dimensional accumulator [8]. However, their algorithm is not able to detect incomplete ellipses present in real images. In contrast, experimental results demonstrate the robustness and effectiveness of our algorithm in which *both* complete and incomplete ellipses are accurately detected in our synthetic and real images when the end points of the major axes are available. In the future, we plan to extend this work by locating the major axis of the ellipse from a smaller set of feature points selected from a curve.

7. REFERENCES

[1] E. R. Davies, *Machine Vision : Theory, Algorithms, Practicalities*, Morgan Kaufmann Publishers Inc., 2004.

[2] S. Tsuji and F. Matsumoto, “Detection of ellipses by a modified hough transformation,” *IEEE Transactions on Computers*, pp. 777–781, 1978.

[3] N. Guil and E. Zapata, “Lower order circle and ellipse hough transform,” *Pattern Recognition*, pp. 1729–1744, 1997.

[4] L. Xu, E. Oja, and P. Kultanen, “A new curve detection method: randomized hough transform,” *Pattern Recognition Letters*, pp. 331–338, 1990.

[5] T. C. Chen and K. L. Chung, “An efficient randomized algorithm for detecting circles,” *Computer Vision and Image Understanding*, pp. 172–191, 2001.

[6] C. T. Ho and L. H. Chen, “A high-speed algorithm for elliptical object detection,” *IEEE Transactions on Image Processing*, pp. 547–550, 1996.

[7] C. T. Ho and L. H. Chen, “A fast ellipse/circle detector using geometric symmetry,” *Pattern Recognition*, pp. 117–124, 1995.

[8] Y. Xie and Q. Ji, “A new efficient ellipse detection method,” *International Conference on Pattern Recognition*, pp. 957–960, 2002.

[9] Weisstein and Eric W., http://mathworld.wolfram.com/Ellipse.html, Accessed on January 2007.

[10] California Institute of Technology, http://pds.jpl.nasa.gov/planets/welcome.htm, Accessed on January 2007.