Quantization-free parameter space reduction in ellipse detection

Kuang Chung Chen

A Thesis

in

The Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Information Systems Security) at

Concordia University

Montréal, Québec, Canada

September 2009

© **Kuang Chung Chen, 2009**

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-63052-5
Our file Notre référence
ISBN: 978-0-494-63052-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Quantization-free parameter space reduction in ellipse detection

Kuang Chung Chen

Ellipse modeling and detection is an important task in many computer vision and pattern recognition applications. In this thesis, four Hough-based transform algorithms have been carefully selected, studied and analyzed. These techniques include the Standard Hough Transform, Probabilistic Hough Transform, Randomized Hough Transform and Directional Information for Parameter Space Decomposition. The four algorithms are analyzed and compared against each other in this study using synthetic ellipses. Objects such as noise have been introduced to distract ellipse detection in some of the synthetic ellipse images. To complete the analysis, real world images were used to test each algorithm resulting in the proposal of a new algorithm.

The proposed algorithm uses the strengths from each of the analyzed algorithms. This new algorithm uses the same approach as the Directional Information for Parameter Space Decomposition to determine the ellipse center. However, in the process of collecting votes for the ellipse center, pairs of unique edge points voted for the center are also kept in an array. A minimum of two pairs of edge points are required to determine the ellipse. This significantly reduces the usual five dimensional array requirement needed in the Standard Hough Transform. We present results of the experiments with synthetic images demonstrating that the proposed method is more effective and robust to noise. Real world applications on complex real world images are also performed successfully in the experiments.

Acknowledgements

Most importantly, I wish to thank my advisor Nizar Bouguila for providing me with this great opportunity. I have really learned a lot and I really appreciate that. I would also like to thank Dr. Bouguila for his encouragement, support and guidance when I needed it the most.

Additionally, I would like to thank Professor Djemel Ziou from the University of Sherbrooke for recommending using the Hough Transform when we started working on the problem of eye detection and modeling. Without his coaching on the Hough Transform, our research direction would have been significantly different.

Finally, I would like to thank my wife, I-Chun, for always believing in me, and my daughter Ashlin, for opening up a new chapter in my life.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Introduction

Detecting elliptical shapes accurately and immediately has been a major issue in computer vision and pattern recognition. These elliptical shapes are commonly viewed in real world scenes. Some of these elliptical shapes can be found in the human body, such as the head [7], eyes [18] and lips [29]. In the biometric world, the eye is often used as a method of personal identification [10]. In driver vigilance and fatigue monitoring systems, the gaze of an eye can provide us with information about the attention a driver is placing on the road [13] [18] [2] [14]. The appearance of elliptical forms increases on an image through the perspective projection of 3-Dimensional circular or elliptical features. This application includes road sign detection [25] and cancerous cell counting [23].

Some of the methods currently applied to detect ellipses are based on symmetry [28] [17]. However, since ellipses of a given image are rarely in perfect condition, the symmetry is hard to apply. Often, segments of an ellipse are missing and symmetry can't be used. Other methods of ellipse detection include random sampling [20] [22]. These methods have an advantage where memory consumption is kept to the minimum. These methods perform well in a real environment where ellipse edge presence is high [19]. However, in real world images, where the number of ellipse edges is low, sampling the correct ellipse edges becomes more difficult [19]. This difficulty is compounded when random sampling edges must also satisfy a specific edge combination. In other words, either the sampled ellipse edges must not be too close to each other or only a specific combination of ellipse edges can be used to locate the ellipse parameters accurately. These

edge point combinations are normally close to the major and minor axis of the ellipse and far apart from each other in order to obtain the best accurate results for ellipse parameters. If they are not, the accuracy of the ellipse detection is compromised. Another issue is that random sampling of edges cannot produce consistent results. This is why a specific number of trials are necessary to produce the final average result, even if it is for the same image. But, if a consistent result is required in a probabilistic model, more trials are needed [16].

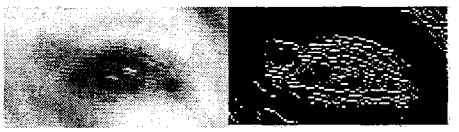
To produce a consistent result in terms of ellipse detection (success/failure) and speed, Hough-based transforms can be used. These methods are recognized as a powerful tool in shape detection, or analysis, with good results in a noisy and occluded environment. Although the technique has a straight forward computation, its primary limitations include lengthy computational time and massive storage requirements. These limitations are barely noticeable with today's high power computing, especially for line detection. For a complex shape, such as ellipse, the computation slowness and five dimensional storage requirements are still obvious.

Since the introduction of the Hough Transform, many improvements have been made . These improvements have targeted performance, software methods, probabilistic models and parallel processing. In the case of performance, the algorithms must perform relatively well in a noisy environment. There are in general two types of noise: random noise and correlated noise. Random noise occurs randomly in the image. As for correlated noise, it only happens when two features are being placed together to form a third feature. Concerning software methods, algorithms have been rewritten efficiently for specific hardware. One of the most popular methods is the lookup table. Using this method, a number of calculations can be performed in advance and stored in an array index . This method speeds up the calculation time, but uses an enormous amount of memory.

Other methods based on Hough transforms vary from edge directional information to parameter space decomposition. Edge directional information can be used to limit the input range of the parameter space. Meanwhile, in the parameter space decomposition, the parameter space can be reduced into separate stages. Each stage passes the result into the next. In the case of probabilistic models, algorithms seek to reduce the number of redundancies using a smaller sample population size. A comprehensive survey of Hough-based transform algorithms can be found in [16].

(a) Image (b) Edges

Figure 1.1: Edges missing in ellipse



(a) Eye (b) Edges

Figure 1.2: An eye and its edges

1.2 Algorithm Motivation

Detecting ellipses accurately and quickly is difficult in many computer vision applications. Accurate detection is generally compromised due to the presence of noise, partially hidden ellipse or poor threshold values in edge detection, as illustrated in Figure 1.1.

Additionally, overlapped ellipse formation interferes with ellipse accuracy detection. This overlapping adds complexity to the problem. As an example, an image of an eye is represented in Figure 1.2a and its edges are represented by Figure 1.2b.

As illustrated in Figures 1.3a and 1.3b, extra edges create additional potential ellipses or features in the image. These ellipses or features can be overlapped one on top of the others. This overlapping is difficult to be spotted even by the naked eyes as illustrated in Figure 1.2b.

1.3 Background

An ellipse located at the origin with no axis of rotation can be described as follows:

(a) (b)

Figure 1.3: Overlapping ellipses

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

where a is semi-major axis and b is semi-minor axis.

However, when the axes of ellipse are rotated on an angle ϕ, the equations of orientation for the semi-major and semi-minor axes become:

$$\begin{aligned} x' &= x\cos\phi + y\sin\phi \\ y' &= -x\sin\phi + y\cos\phi \end{aligned} \tag{2}$$

Substituting the previous two equations into equation 1, a new ellipse equation located at the origin with axis orientation can be obtained by:

$$[\frac{x\cos\phi + y\sin\phi}{a}]^2 + [\frac{-x\sin\phi + y\cos\phi}{b}]^2 = 1 \tag{3}$$

If a translation is performed on the ellipse to (x_0,y_0), equation 3 becomes:

$$[\frac{(x-x_0)\cos\phi + (y-y_0)\sin\phi}{a}]^2 + [\frac{-(x-x_0)\sin\phi + (y-y_0)\cos\phi}{b}]^2 = 1 \tag{4}$$

In the following sections, we will present in details the Standard Hough Transform [4], Probabilistic Hough Transform [15], Randomized Hough Transform [23] and finally, Parameter Space Decomposition [1].

1.3.1 Standard Hough Transform (SHT)

As stated previously, detecting elliptical shape is a difficult problem in computer vision. This difficulty is primarily due to the variables that describe an ellipse. In the case of an ellipse, there are five variable

parameters: center (x_0,y_0), semi-major axis a, semi-minor axis b and orientation of the ellipse angle ϕ as illustrated in Figure 1.4.

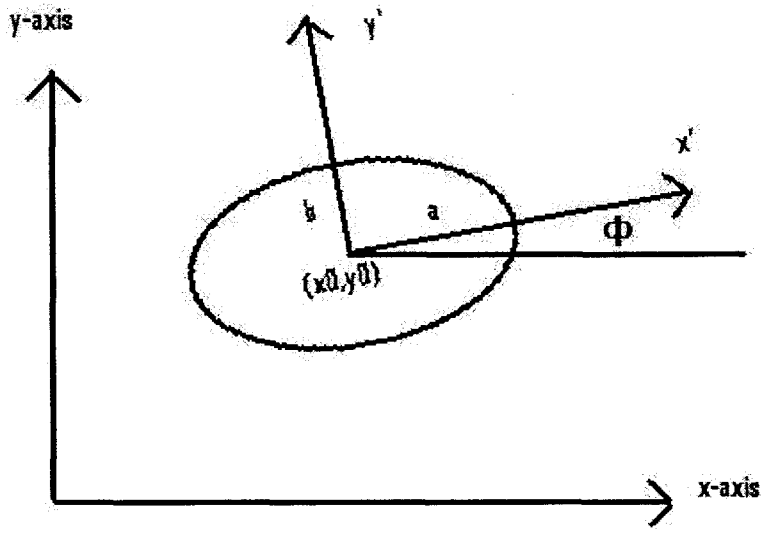


Figure 1.4: Ellipse parameters

A popular method used to detect ellipses is the Hough Transform. The Standard Hough Transform was first proposed in 1962 [11] and later improved by Duda [4]. The basic concept behind the Standard Hough Transform in ellipse extraction is to define a two dimensional image space mapping in the xy-plane with a five dimensional parameter space mapping in the $x_0y_0ab\phi$-plane. All possible ellipses in the parameter space can be generated from a given point in the image space. This process is then repeated for every other points in the image space. The parameter set that appears most frequently indicates the possible presence of an ellipse. The counting frequency of a parameter set is constructed using an accumulator cell. This counting technique is often referred to as a voting scheme where each accumulator cell represents a parameter set

is normally quantized and is initially set to zero. The collection of all the accumulator cells is called the accumulator. An example of an accumulator, taking into account the ellipse in Figure 1.5 is presented in Figure 1.6.

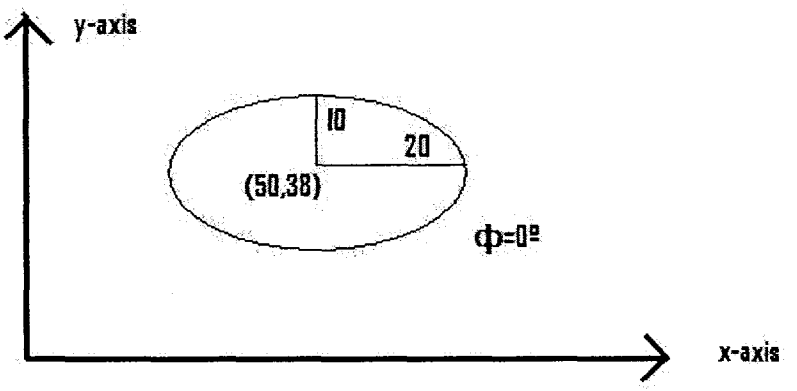


Figure 1.5: Ellipse example

1.3.2 Probabilistic Hough Transform (PHT)

The Probabilistic Hough Transform was proposed in 1991 by Kiryati [15]. The idea is to reduce the number of points across the image evenly as illustrated in Figures 1.7a and 1.7b. After reducing the number of points in the image, the Standard Hough Transform is used. This reduction of points reduces the computational time. The improvement in the Probabilistic Hough Transform can be explained as follows. Normally, in the Standard Hough Transform, there are two phases. The first phase consists of the generation of all the possible ellipse parameter sets using every point in the image. The second phase consists of searching for the highest accumulator cell. Analyzing these two phases, the first phase clearly takes more time. Using only a small percentage of points to compute the first phase clearly reduces the overall computing time.

Figures 1.8 and 1.9 demonstrate an example of the Probabilistic Hough Transform. The comparison between figures 1.6 and 1.9 demonstrate that figure 1.9 is only a scaled down version of Figure 1.6. The overall shape of the histogram is retained. Additionally, according to Kiryati [15], the percentage of points required from the original image falls in the range of 5% to 15%.

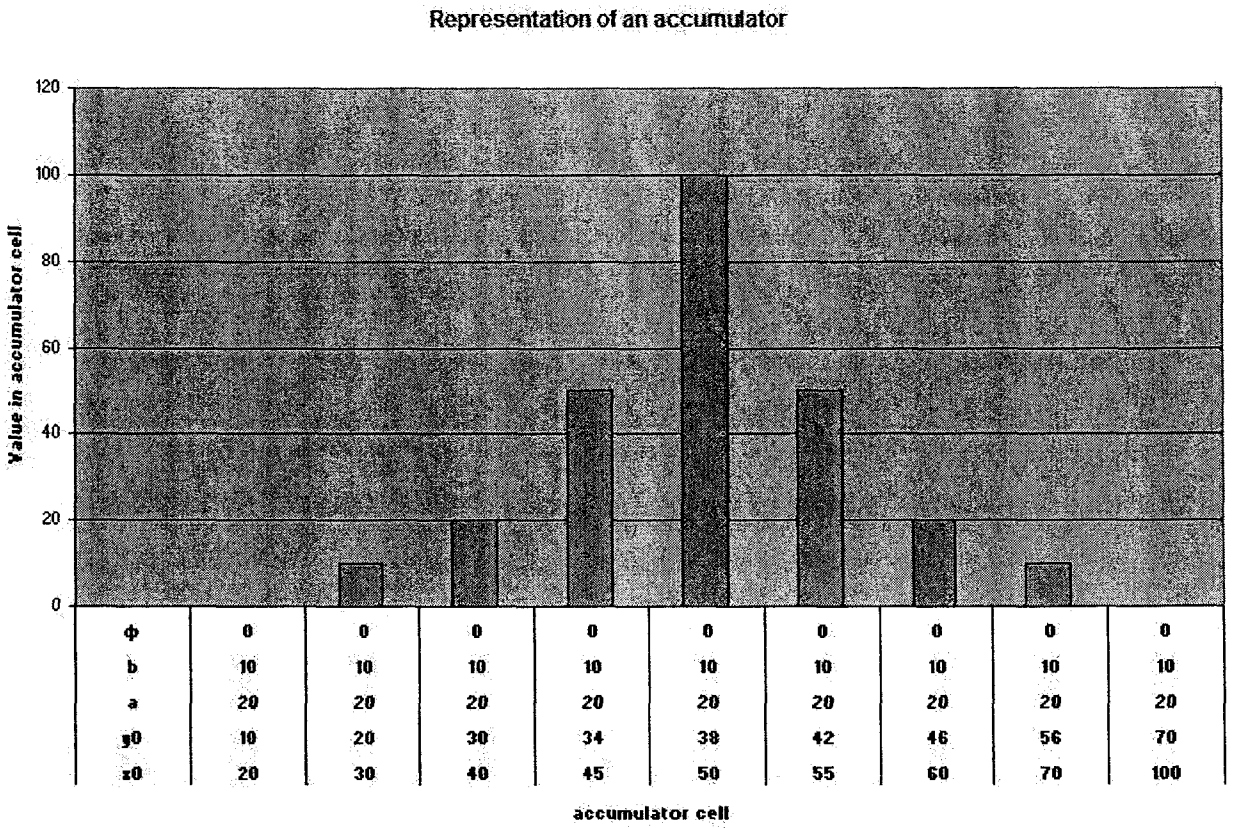


Figure 1.6: Accumulator

1.3.3 Randomized Hough Transform (RHT)

The Randomized Hough Transform was originally proposed for curves that can be only expressed by a linear equation. Curves that are nonlinear cannot be applied directly. Reference [31] provides a brief summary of how to apply a Randomized Hough Transform in linear curve while reference [23] details how to apply it to an ellipse.

The algorithm described by McLaughlin [23] is as follows. Select any of the three point's $p_1(x_1,y_1)$,$p_2(x_2,y_2)$ and $p_3(x_3,y_3)$ randomly from the image space and estimate a tangent line at each of the selected points using the neighboring points, as illustrated in Figure 1.10. Take any of the two selected points, in our first case p_1 and p_2, and locate their midpoint m_{12} and the intersection of their tangent lines t_{12}, as described in Figure

(a) All edges (b) 30 of edges

Figure 1.7: PHT and edges

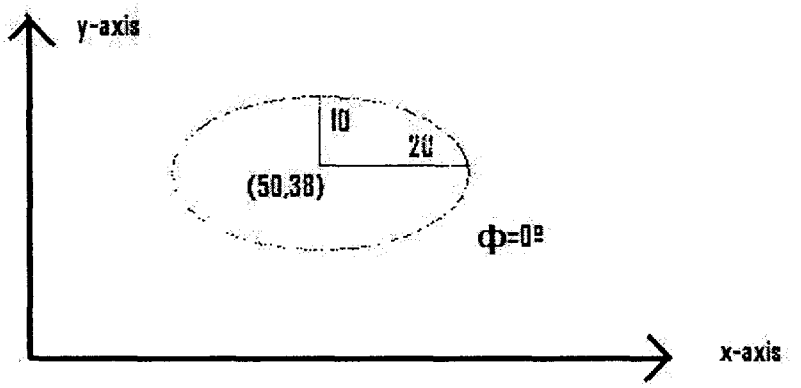


Figure 1.8: Ellipse example with Probabilistic Hough Transform

1.10. The center (x_0,y_0) will lie on the line l_{12} that passes through points t_{12} and m_{12} as described by Yuen et al [32].

Similarly, take points p_2 and p_3 and locate their midpoint, m_{23}. The center of the ellipse is situated on the line l_{23} that passes through t_{23} and m_{23}. The approximate center of the ellipse (x_0,y_0) is situated on the intersection of the lines l_{12} and l_{23} as described in Figure 1.10.

The ellipse equation used by McLaughlin is:

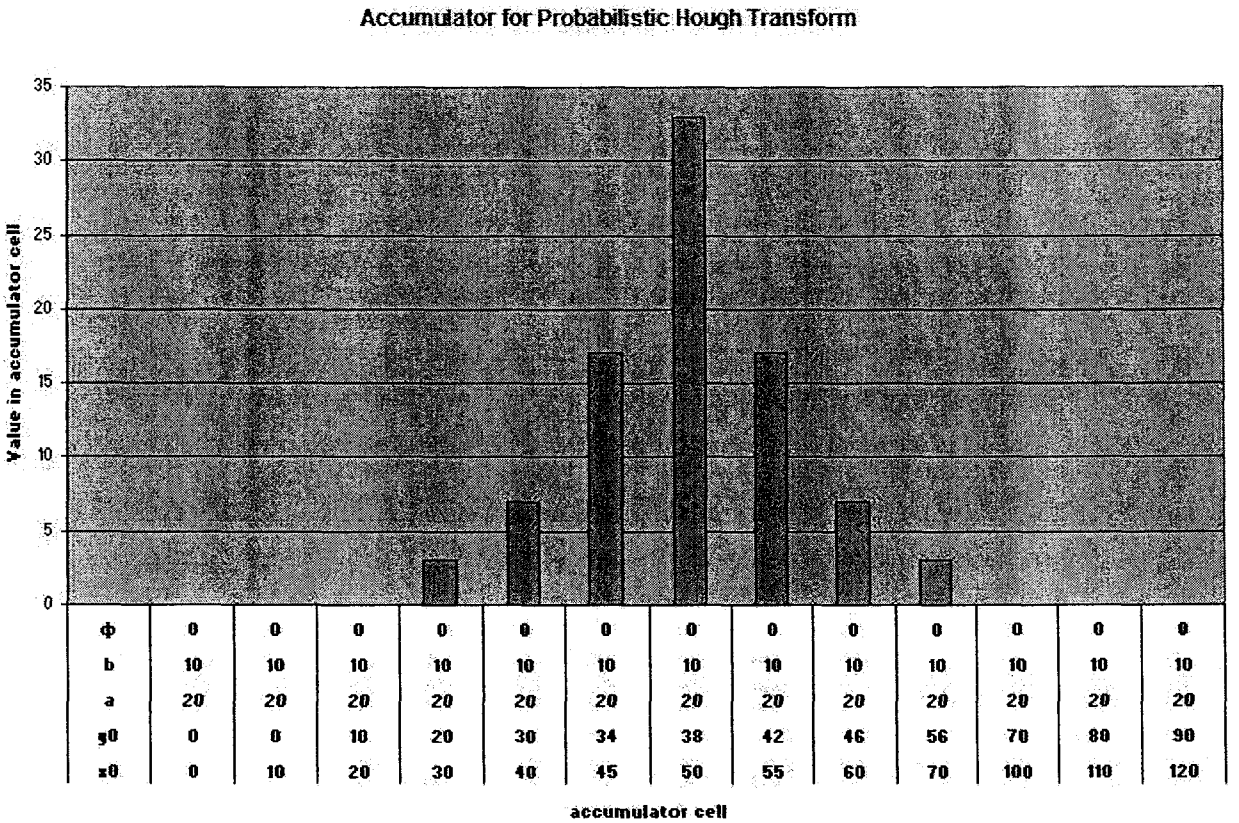


Figure 1.9: Accumulator for Probabilistic Hough Transform

$$A(x - x_0)^2 + 2B(x - x_0)(y - y_0) + C(y - y_0)^2 = 1 \tag{5}$$

where

$$AC - B^2 > 0 \tag{6}$$

and A, B and C are coefficients.

For simplicity, the center of the ellipse is translated to the origin. Thus,

$$Ax^2 + 2Bxy + Cy^2 = 1 \tag{7}$$

Substituting the coordinates of the points p_1, p_2 and p_3 into equation 5, a three linear equation system is obtained.

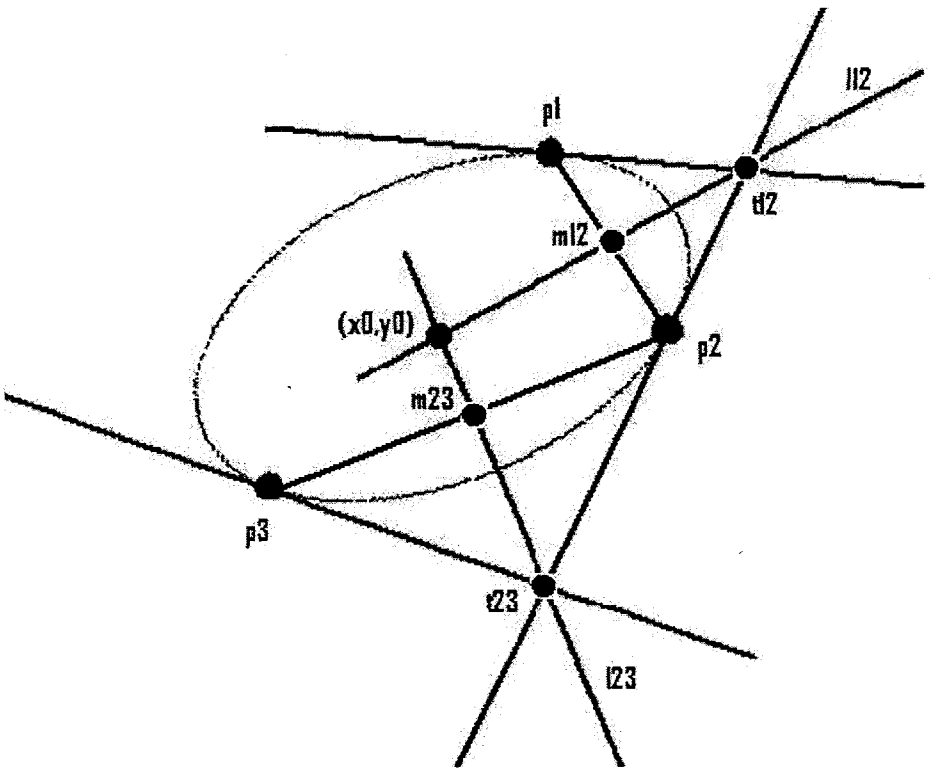


Figure 1.10: Create tangent lines at point p_1, p_2 and p_3 and find their midpoints m_{12} and m_{23}.

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ x_2^2 & 2x_1y_2 & y_2^2 \\ x_3^2 & 2x_1y_3 & y_3^2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{8}$$

Once elliptical (x_0,y_0,A,B,C) quadratic equations shown in equation 8 are solved, a translation back to the original parameter form is required (x_0,y_0,a,b,ϕ) as demonstrated by Inverso [12]. Store the value of the ellipse parameters in the accumulator set, if the accumulator set does not exist previously in the accumulator. If the accumulator set does exist, combine it with the existing accumulator set found in the accumulator.

1.3.4 Parameter Space Decomposition (PSD)

The idea behind this transform is to use two independent accumulators to gather evidence. The first accumulator is used to gather evidence regarding the ellipse center and the second accumulator is used to gather evidence about a, b and ϕ.

This transform focuses on locating a map using two selected points, p_1 and p_2, to find the ellipse center (x_0,y_0). Once the ellipse center (x_0,y_0) has been located, it can be used as an input to gather additional evidence regarding parameters a, b and ϕ.

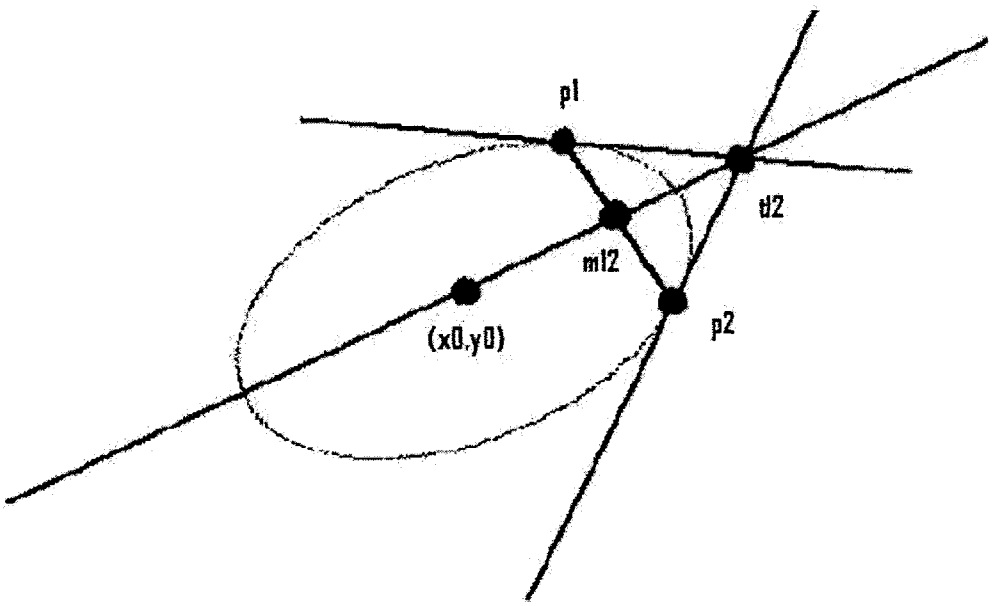


Figure 1.11: Geometrical relationship between the two randomly selected points and their tangent line

Take any of the two selected points, in our first case p_1 and p_2, and locate their midpoint m_{12} and the intersection of their tangent lines t_{12}, as illustrated in Figure 1.11. Let L_1 be a line that passes through point p_1 and t_{12}. Let L_2 be a line that passes through points p_2 and t_{12}. Then, equations L_1 and L_2 are described as the following, where $slope_1$ and $slope_2$ are the slopes of L_1 and L_2, respectively.

$$\begin{aligned} L_1 &= slope_1 x + c_1 \\ L_2 &= slope_2 x + c_2 \end{aligned} \tag{9}$$

where c_1 and c_2 are constants.

$$\begin{aligned} slope_1 &= \frac{y_t - y_1}{x_t - x_1} \\ slope_2 &= \frac{y_t - y_2}{x_t - x_2} \end{aligned} \tag{10}$$

Solving c_1 and c_2 in Equation 9 using point t_{12}, it can be determined that

$$\begin{aligned} c_1 &= y_t - \frac{y_t - y_1}{x_t - x_1} x_t \\ c_2 &= y_t - \frac{y_t - y_2}{x_t - x_2} x_t \end{aligned} \tag{11}$$

The equations of L_1 and L_2 are as follows:

$$\begin{aligned} L_1 &= slope_1(x - x_t) + y_t \\ L_2 &= slope_2(x - x_t) + y_t \end{aligned} \tag{12}$$

By using points p_1 and p_2 in Equation 12

$$\begin{aligned} y_2 &= slope_1(x_2 - x_t) + y_t \\ y_1 &= slope_2(x_1 - x_t) + y_t \end{aligned} \tag{13}$$

Solving Equation 13 for x_t and y_t yields:

$$\begin{aligned} x_t &= \frac{y_2 - y_1 - slope_2 x_2 + slope_1 x_1}{slope_1 - slope_2} \\ y_t &= \frac{-slope_2 y_1 + slope_1 y_2 + slope_1 slope_2 (x_1 - x_2)}{slope_1 - slope_2} \end{aligned} \tag{14}$$

Let the midpoint m_{12} of p_1 and p_2 be defined as (x_m, y_m), the slope for m_{12} and t_{12} can be defined by $slope_3$ as follows:

$$slope_3 = \frac{y_t - y_m}{x_t - x_m} \tag{15}$$

Let x_m and y_m be defined as

$$\begin{aligned} x_m &= \frac{x_2 + x_1}{2} \\ y_m &= \frac{y_2 + y_1}{2} \end{aligned} \tag{16}$$

Using x_m and y_m from Equation 16 and y_t and x_t from Equation 14 and substituting into Equation 15, it can be found that:

$$slope_3 = \frac{AC + 2BD}{2A + BC}$$

where

$$A = y_1 - y_2$$

$$B = x_1 - x_2$$

$$C = slope_1 + slope_2$$

$$D = slope_1 * slope_2$$

$$slope_1 = \frac{y_1 - y_t}{x_1 - x_t}$$

$$slope_2 = \frac{y_2 - y_t}{x_2 - x_t}$$

Therefore, the equation of line that passes through the ellipse center and midpoint m_{12} can be expressed as:

$$y_0 = y_m + \frac{AC + 2BD}{2A + BC}(x_0 - x_m) \tag{17}$$

1.4 Contributions

The contributions of this thesis to the literature will be many. A thorough analysis of the various Hough Transform based algorithms for ellipse detection will be provided. The focus will primarily be on the Standard Hough Transform, Probabilistic Hough Transform, Randomized Hough Transform and Parameter Space Decomposition. After this thorough analysis, a new proposed method will be provided.

This thesis is organized as the follows: Chapter 1 provides essential information about the concepts and definitions that will be referenced throughout the thesis. It also provides information about the different types of Hough Transforms. In chapter 2, synthetic images will be used to validate how quickly and accurately ellipses are being detected by each algorithm. Real world images will be introduced afterwards.

And a discussion will follow after the experiments. In chapter 3, a new method is proposed by taking into account the strengths and weaknesses of each Hough-based transform. Comparisons between the new proposed method and other Hough-based transforms will follow afterwards. Chapter 4 concludes the thesis and provides recommendation for future work.

CHAPTER 2

Experimental results using synthetic and real world images

2.1 Description of the experiments

The main purpose of this chapter is to analyze each algorithm through experiments. The experiments were designed to bring out the best and the worst of each algorithm under different circumstances. It is important in our case, to determine how each algorithm deteriorates in terms of detection and calculation. The experiments were first conducted using synthetic ellipses. Additional objects were placed in afterwards to disturb ellipse detection. Finally, noise was scattered and added to the image. In the Probabilistic Hough Transform, only 20% of the edge points were taken. In the Randomized Hough Transform, each trial only had 200 samples. In each sample, three points were given the chance to vote. There were a total of 10 trials. The emphasis was placed mainly on the detection and the calculation time. The experiments were conducted using a DELL laptop D500 centrino 1.3GHz with 1.5Gigabytes of memory running on a Windows XP 32bit and Matlab 2008a system.

2.2 Experiment using single synthetic ellipse

The first experiment was carried out using synthetic ellipses. Figure 2.1 illustrates an example. The ellipse center location (x_0,y_0), orientation ϕ, semi-major axis a and semi-minor axis b vary from picture to picture. There were in total ten images used in the first experiment. On average, there were a total of 77 edge points

in the ellipse. In this experiment, all the algorithms performed relatively well for the ten images provided. As expected, the Standard Hough Transform was the slowest and the parameter space decomposition was the quickest. The results of the first experiment are summarized in Table 2.1

Figure 2.1: Single synthetic ellipse

Table 2.1: Experiment1 - Single synthetic ellipse detection.

	SHT	PHT	RHT	PSD
Number of Trials	10	10	10	10
Number of successful ellipse detections	10	10	10	10
Percentage of successful rate	100%	100%	100%	100%
Average time in seconds	500	100	42	2

2.3 Experiment using single synthetic ellipse with a rectangle

In the second experiment, an object of rectangular shape was added to the image as illustrated in Figure 2.2.

Figure 2.2: Experiment2 - Single synthetic ellipse and a single rectangle

Ellipse and rectangle location, shape and orientation were modified in every image. There were a total of ten images used in the second experiment. On average, there were, in total, 168 edge points in each image. 77 belonged to the ellipse and 91 belonged to the rectangle on average. This translates to over 50% of the edges that do not belong to the ellipse. In the second experiment, the Standard Hough Transform and

Directional information for parameter space decomposition had the highest accuracy in the ten images provided. However, due to the extra involved edge points in the image, the average calculation has been almost doubled. As expected, the Standard Hough Transform was the slowest and the Directional information for parameter space decomposition was the quickest. It should be noted that the accuracy of the Randomized Hough Transform and Probabilistic Hough Transform have deteriorated slightly. The calculation time for the Randomized Hough Transform has been kept constant, since only a fixed number of samples were used and the number of trials remained the same. The results of the second experiment can be seen in Table 2.2

Table 2.2: Experiment2: Single synthetic ellipse detection using figure 2.2.

	SHT	PHT	RHT	PSD
Number of Trials	10	10	10	10
Number of times of successful ellipse detection	10	7	5	10
Percentage of successful rate	100%	70%	50%	100%
Average time in sec	900	180	42	2

2.4 Experiment using synthetic ellipse with a rectangle, triangle, and a letter T

In experiment 3, two additional objects were added. These objects consisted of a triangle and a letter T. The purpose of adding these two objects was to distract further each ellipse detection algorithm. The images used were similar to those in Figure 2.3.

Figure 2.3: Experiment3 - Single synthetic ellipse, rectangle, triangle and a letter T.

The location, shape and orientation of each ellipse, rectangle, triangle and letter T were changed randomly in every image. There were a total of ten images used in the third experiment. On average, there were,

in total 227 edge points in each image. 77 belonged to the ellipse and 150 belonged to the rest of the objects. Over $\frac{2}{3}$ of the edge points did not belong to the ellipse. These extra edge points in the image tripled the original calculation compared to experiment 1. The Standard Hough Transform and Directional information for parameter space decomposition had the highest accuracy in the ten images provided. As expected, Standard Hough Transform was the slowest and the Directional information for parameter space decomposition was the quickest. It should be noted that the accuracy of Randomized Hough Transform and Probabilistic Hough Transform have further deteriorated. The results of the third experiment are summarized in Table 2.3

Table 2.3: Experiment3 - Single synthetic ellipse detection using figure 2.3.

	SHT	PHT	RHT	PSD
Number of Trials	10	10	10	10
Number of successful ellipse detection	10	6	2	10
Percentage of successful rate	100%	60%	20%	100%
Average time in seconds	1220	244	42	2

2.5 Experiment using synthetic ellipse with a rectangle, triangle, and a letter T, with one percent noise

In experiment 4, random noise was added to the images to further distract each algorithm from detecting the ellipse as illustrated in Figure 2.4).

Figure 2.4: Experiment4 - Single synthetic ellipse, rectangle, triangle and a letter T with one percent noise.

Adding the 1% noise has made the ellipse detection more difficult and the computation longer. There were a total of ten images used in the fourth experiment. On average, there were a total of 594 edge points in the image. Only 77 edge points belonged to the ellipse. Over 87% of the edge points did not belong to the ellipse. The Standard Hough Transform and Directional information for parameter space decomposition had the highest accuracy in ellipse detection in the ten images provided. Computational time was longer than ever. Both the Randomized Hough Transform and Probabilistic Hough Transform performed poorly in the fourth experiment. The summary of the fourth experiment is illustrated in Table 2.4.

Table 2.4: Experiment4 - Single synthetic ellipse detection using figure 2.4 with one percent noise.

	SHT	PHT	RHT	PSD
Number of Trials	10	10	10	10
Number of successful ellipse detection	7	1	0	10
Percentage of successful rate	70%	50%	0%	100%
Average time in seconds	3290	630	42	5

2.6 Summary of the four experiments

As percentage of edge points of the ellipse diminished, the probabilistic models were found to perform poorly. This can be explained by the difficulty encountered in the random sampling. Sampling the right ellipse edge points was more difficult than ever, because of the low ellipse edge points presence. This observation can be used to explain the results provided in Table 2.5.

To detect an ellipse, edge points belonging to the ellipse need to be used. Random sampling edge points belonging to the ellipse do not guarantee an accurate detection. For instance, this is true given that the edge points belonging to the ellipse were chosen in close proximity to each other as illustrated in Figure 2.5.

Given that edge points belonging to the ellipse were chosen in close proximity to each other as seen in Figure 2.5, the partial ellipse detection can be observed in Figure 2.6. This phenomenon can be explained by the lack of information about the ellipse itself such as the ellipse center, semi-major axis, semi-minor

Table 2.5: Noise and edge influence in the experiments.

	Experiment1	Experiment2	Experiment3	Experiment4
Number of Edge points belonging to ellipse	74	74	74	74
Total number of edge points	74	168	227	594
% of edge points belonging to ellipse	100%	44%	32%	12%
% of successful ellipse detection in SHT	100	100	100	70
% of successful ellipse detection in PHT	100	70	60	50
% of successful ellipse detection in RHT	100	50	20	0
% of successful ellipse detection in PSD	100	100	100	100

Figure 2.5: Close edge selected for the random sampling.

axis and ellipse orientation. Therefore, ideally, to raise the accuracy of ellipse detection, the points selected should be equidistant apart from each other in the perimeter of the ellipse. This ensures accurate ellipse detection.

Another observation made while performing these four experiments in the Standard Hough Transform and Directional information for parameter space decomposition was the computational time. As more and more edge points were found in the image, the computation time increased linearly. When the number of edge points doubled in the image, the computational time also doubled.

2.7 Experiment using real world images

The images used in this experiment can be divided into four categories: a normal eye illustrated in Figures 2.7a-2.10a, an eye wearing glasses as illustrated in Figures 2.11a-2.14a and an eye with a different

Figure 2.6: Ellipse formation using close edge points

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.7: Results of each algorithm for eye 1

orientation as illustrated in Figures 2.15a-2.18a.

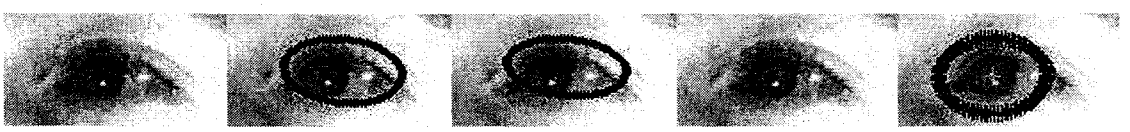
The results of each algorithm can be seen side-by-side to each other. Figures 2.7b-2.18b show the results for the Standard Hough Transform. Figures 2.7c-2.18c show the results for the Probabilistic Hough Transform. Figures 2.7d-2.18d shoe the results for the Randomized Hough Transform. And finally, Figures 2.7e-2.18e show the results for Parameter Space Decomposition.

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.8: Results of each algorithm for eye 2

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.9: Results of each algorithm for eye 3



(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.10: Results of each algorithm for eye 4

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

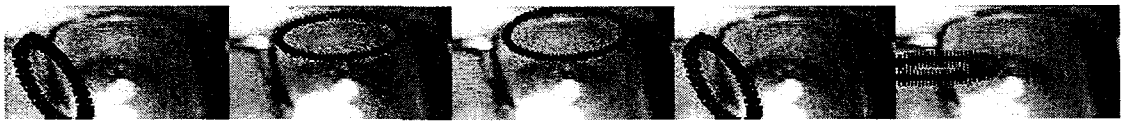
Figure 2.11: Results of each algorithm for eye 5

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.12: Results of each algorithm for eye 6

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.13: Results of each algorithm for eye 7

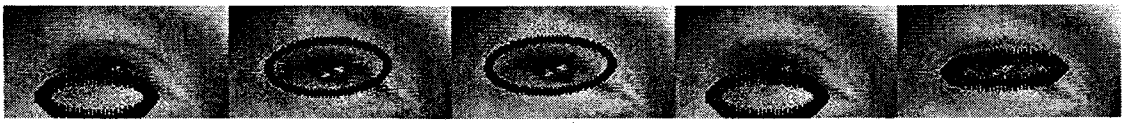


(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.14: Results of each algorithm for eye 8

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.15: Results of each algorithm for eye 9



(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.16: Results of each algorithm for eye 10

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.17: Results of each algorithm for eye 11

(a) Original (b) SHT (c) PHT (d) RHT (e) PSD

Figure 2.18: Results of each algorithm for eye 12

In Table 2.6, the calculation time required for each image has been recorded. The average calculation time for each algorithm is also provided at the end. The Standard Hough Transform had the slowest average calculation time. The Parameter Space Decomposition had the quickest. The Randomized Hough Transform was omitted, since, in most pictures, no ellipse was found.

Looking through Figures 2.7-2.18, it was found that the Standard Hough Transform and Probabilistic Hough Transform had provided reasonable accuracy. In the Probabilistic Hough Transform, the result was impressive, considering that only a fraction (20%) of the edge points was used. In the Randomized Hough Transform, the result of the ellipse detection has been quite disappointing. This was due to the fact that there were too many edge points not pertaining to the ellipse. In most cases, no ellipse was found. The Randomized Hough Transform provided the poorest detection/accuracy. The Parameter Space Decomposition was found to have the quickest calculation and best accuracy.

Table 2.6: Calculation time required in the SHT, PHT, RHT and PSD for eye1-12

Image	SHT	PHT	RHT	PSD
Eye1	614sec	123sec	386sec	4.07sec
Eye2	610sec	126sec	416sec	4.67sec
Eye3	630sec	145sec	357sec	4.1sec
Eye4	615sec	130sec	360sec	4.95sec
Eye5	1211sec	219sec	130sec	7.72sec
Eye6	834sec	156sec	99sec	5.11sec
Eye7	938sec	178sec	105sec	5.69sec
Eye8	741sec	140sec	98sec	4.17sec
Eye9	588sec	138sec	92sec	4.53sec
Eye10	724sec	173sec	100sec	5.66sec
Eye11	627sec	148sec	95sec	4.84sec
Eye12	638sec	166sec	88sec	4.60sec
Average time	730sec	154sec	193sec	5sec

2.8 Comparison and discussion of the Hough based algorithms

In Table 2.7, major iterations were calculated in BigOh notation for each algorithm. The major iterations include computing the Hough Transform and locating the highest accumulated value in the accumulator. Memory requirements for each algorithm were also calculated.

Table 2.7: Number of major iterations and memory requirements for the Standard Hough Transform, Probabilistic Hough Transform, Randomized Hough Transform and Parameter space decomposition

	Big Oh notation in terms of major iteration loop	Accumulator storage requirement
Standard Hough Transform	$M[a]_i[b]_j[x_0]_k[\phi]_v$ + $[a]_i[b]_j[x_0]_k[\phi]_v$ where M is the total number of edge points	$[a]_i[b]_j[x_0]_k[y_0]_u[\phi]_v$
Probabilistic Hough Transform	$m[a]_i[b]_j[x_0]_k[\phi]_v$ + $[a]_i[b]_j[x_0]_k[\phi]_v$ where $m < M$ and m is the total randomly selected point and M is the total number of edge points	$[a]_i[b]_j[x_0]_k[y_0]_u[\phi]_v$
Randomized Hough Transform	$P\left[\begin{pmatrix} M \\ 3 \end{pmatrix} + \begin{pmatrix} M \\ 3 \end{pmatrix}\right]$ where P is a percentage of points combination	$P\begin{pmatrix} M \\ 3 \end{pmatrix}$
Parameter space decomposition	$P\left[\begin{pmatrix} M \\ 2 \end{pmatrix}\right][x_0]_k$ + $[a]_i[b]_j[\phi]_v$ + $[x_0]_k[y_0]_u$ +$[a]_i[b]_j[\phi]_v$ where P is a percentage combination of edges	$[x_0]_k[y_0]_u$+$[a]_i[b]_j[\phi]_v$

Figure 2.19: A normal eye

In Figure 2.19, there were in total 454 edge points. Only 133 edge points belonged to the eye contour. There were 321 irrelevant points, meaning more than 70% of the points were not needed. These irrelevant edge points contributed to unnecessary calculation in the Standard Hough Transform. Eliminating these irrelevant edge points sped up Standard Hough Transform calculation, at the same time as removing the false ellipse detection.

Additionally, these irrelevant edge points also played an important role in the Randomized Hough Transform. As the number of irrelevant edge points increased, the precision and the successful ellipse detection rate for the Randomized Hough Transform decreased. This was due to the fact that there were too many irrelevant edge points in the image. Therefore, randomly selecting points belonging to ellipse became more difficult. Hence, the Randomized Hough Transform has more difficulty converging to a possible ellipse

parameter set.

Using eye contour edge points only improves part of the computational time. The other part of the improvement comes from how each of the accumulated cell was constructed. Here is an example:

Let $[a]_i$ be all the possible values of the semi-major axis, let $[b]_j$ be all the possible values of the semi-minor axis, let $[x_0]_k$ be all the possible values of the ellipse center in x-component, let $[y_0]_u$ be all the possible values of the ellipse center in y-component and let $[\phi]_v$ be all the possible values of the ellipse orientation.

A normal accumulator can be constructed as follows:

$$Accumulator[a]_i[b]_j[x_0]_k[\phi]_v$$

Where

$$[a]_{i+1} - [a]_i = 1$$
$$[b]_{j+1} - [b]_j = 1$$
$$[x_0]_{k+1} - [x_0]_k = 1$$
$$[\phi]_{v+1} - [\phi]_v = 1$$

Having the accumulator cell spaced at 1 for every Hough Space parameter x_0-y_0-a-b-ϕ provides the following number of loop operations.

$M[a]_i[b]_j[x_0]_k[\phi]_v + [a]_i[b]_j[x_0]_k[\phi]_v$

where M is the number of edge points. The first term $M[a]_i[b]_j[x_0]_k[\phi]_v$ is for collecting votes and the second term $M[a]_i[b]_j[x_0]_k[\phi]_v$ is for searching for the highest accumulator cell with the most accumulated votes.

If the accumulator cell is spaced at 0.5 for every Hough Space parameter, as illustrated next:

$$[a]_{i+1} - [a]_i = 0.5$$

$$[b]_{j+1} - [b]_j = 0.5$$

$$[x_0]_{k+1} - [x_0]_k = 0.5$$

$$[\phi]_{v+1} - [\phi]_v = 0.5$$

Then the number of iteration loops increases by 16.

$16M[a]_i[b]_j[x_0]_k[\phi]_v + 16[a]_i[b]_j[x_0]_k[\phi]_v$

Hence, the accuracy of each accumulator cell plays an important role in the computational time of any Hough based Transform.

In the Probabilistic Hough Transform, every edge point contribution to the final voting accumulator set depends upon whether the edge point is being selected to vote. This is a good way to reduce unwanted edge points since over 70% of the edge points were irrelevant. In our Probabilistic Hough Transform experiment, we randomly selected 10% of the edge points or 45 points from Figure 2.19. Using only 14 edge points in the eye contour, the Probabilistic Hough Transform was able to accurately determine an ellipse. The only drawback with the Probabilistic Hough Transform is that if points were not selected carefully, the ellipses that were not the highest peaks will be chosen. Similarly, if there are too many edge points not belonging to the ellipse in the image, locating the ellipse becomes more difficult.

In the Randomized Hough Transform, the primary iteration loop can be found only through the different combination set of the 3-point selection process in *p1*, *p2* and *p3*. Selecting points belonging to the ellipse has turned out to be difficult using Figure 2.19 since there are $\begin{pmatrix} 454 \\ 3 \end{pmatrix}$ or 15493203 combinations in total and only 383306 combinations of 3-points belonging to the ellipse. Only 2.5% of the 3-point combinations belong to the ellipse. 4532969 iterations can be removed if eye contour points are selected, or over than 97.5% of the iteration is unnecessary. 4532969 accumulator cells can be removed with a reduction of memory consumption by 97.5%. Locating the maximum accumulated cell is also quicker, as a result.

CHAPTER 3

New proposed algorithm

3.0.1 Introduction

Many real world images contain elliptical shapes. Elliptical shape detection has always been a key problem in computer vision and pattern recognition, especially for real-time applications like human face detection [26], iris detection [30] and driving assistance [9] [6].

Most ellipse detection approaches falls under four categories: Symmetry-based [28] [17], Random sampling [3], Genetic algorithm [27] and Hough-based transform [4] [15] [23] [1]. In symmetry-based detection, the ellipse geometry is taken into account. The most common elements used in ellipse geometry are the ellipse center and axis. Using these elements and edges in the image, the ellipse parameters can be found. In random sampling methods, samples must satisfy a time bound and also a given probability density based on a particular geometry. After sampling, the edge points are used to determine ellipse parameters [3]. In Genetic algorithms, the points in the image are divided into smaller subsets. The random samples taken from the subset are used to create a particular ellipse parameter set. A cost function is used to evaluate the presence of an ellipse by counting the number of points in the proximity of the ellipse perimeter. High points presence close to the ellipse perimeter indicates the presence of an ellipse. This cost function acts as an equivalent of an accumulator in the Hough transform. Two subsets (parents) are chosen to evoluate/crossover/mutate based on their cost function value to produce two additional subsets (offsprings). These offsprings are used to replace their parents in the subsets. A solution is found when all subsets converge to a particular ellipse parameter set. Hough-based transform is easy to implement. It uses edges in the image along with different ellipse parameters to create votes for the accumulator. The highest accumulated cell indicates the presence

of an ellipse.

Hough-based transforms vaguely fall under three different variations. There is the classic standard Hough transform [4]. It is a robust method used to detect ellipses. However, due to the heavy computational time and large amount of storage area requirement, ellipse detection under the standard Hough transform is impractical. Moreover, it may result in innacurate estimates especially in the case of noisy images. The second type is the probabilistic Hough transform [15]. This algorithm seeks to reduce the number of redundant votes in the accumulator. This reduction is accomplished by reducing the edges across the image evenly. The third type is parameter space decomposition [24] [8]. In this method, calculations are split into multiple stages where output in the accumulator in the first stage is used as an input in the next stage. Furthermore, improvement in the original parameter space decomposition is accomplished by the edge directional information level [1]. A comprehensive survey of different Hough-based transforms can be found in [16].

3.0.2 Recent works

Recent developments on the Hough-based transform in [33] have proposed removing spurious points using edge curvature convexity. Edges that do not converge to the surrounding group edges curvature are then removed. This process dramatically reduces the number of edges to be used in any Hough-based transform. Keeping only edges belonging to the curvatures should help locating the ellipse parameters values accurately. In the method proposed in [33], ellipse detection consists then of four stages: removing spurious edge points in the first stage, locating the ellipse center using an accumulator (x_0,y_0) in the second stage, locating the ratio N of semi-minor axis b over the semi-major axis a and the tangent of ellipse orientation $\tan(\phi)$ using a second accumulator (N,K) in the third stage, where $K = \tan(\phi)$. Rewriting the second accumulator in terms of a, b and ϕ, the new second accumulator becomes $(a/b,\tan(\phi))$. Finally, using a/b and $\tan(\phi)$ in the third stage, combined with the ellipse center (x_0,y_0) to indirectly locate a, b and ϕ through N and K in the fourth stage. This method suffers mostly from input error propagation, where inaccurate results are passed on to detect a, b and ϕ due to the error introduced by indirect quantization. The main source of inaccuracy is situated in the second accumulator (N,K) where ellipse parameters a, b and ϕ are not quantized directly. Ellipse orientation is computed from K. Since K is a quantized value or more precisely

an approximation, the orientation ϕ will be erroneously calculated. For instance, given the real value of K to be 1.5, the theoretical ellipse orientation value ϕ yields 56.31 degrees. If the quantization of 1 is to be used between accumulator cells, the original K will be rounded off to 2. This will yield 63.43 degrees. Narrowing down the quantization cells will only increase the additional memory requirement for the accumulator. If the direct parameter quantization of 1 was applied to the ellipse orientation, the ellipse orientation will only be off by 1. In other words, 56.31 degrees will be rounded off to 56 degrees. By the same explanation, b will be inaccurately calculated given that a is provided and N is retrieved from the highest accumulator cell. Finally, it can be seen in the experimental results in [33] that the values of the semi-major axis and semi-minor axis fall short in terms of accuracy.

Other recent developments have focused on improving Randomized Hough transform (RHT). The authors in [19] demonstrated that RHT is easily influenced by noise in the image, so an iterated algorithm is proposed by narrowing down a search area in each iteration to remove noise. By focusing on a specific area, the edges belonging to the ellipse have a higher percentage chance of being selected randomly. Higher selection of ellipse edges leads to higher ellipse detection. However, if ellipse edges were not included in the first iteration, ellipse detection becomes difficult. In addition, what happens when noise and other unrelated edges are actually inside the ellipse? Zooming-in to a specific area will not help reduce unrelated edges or noise. Therefore, narrowing down a specific area might not work as originally intended. Furthermore, at each iteration, votes are reset and recollected. This leads to the same combination of edges voting multiple times throughout the iterations. Other proposed algorithms involve extracting line segments at the pixel level [21] and then linking together all the potential line segments to form arc segments. Arc segments of the same ellipse are then grouped together. This method falls short in terms of detecting small ellipses, since the accuracy of multiple arcs cannot be guaranteed.

In general, most Hough-based transforms face parameter quantization inaccuracy. This inaccuracy not only affects the ellipse parameter values but also disperses votes into other accumulator cells. Therefore, to improve ellipse accuracy and memory consumption, quantization must be avoided and original edges should be used instead to compute ellipse parameters. However, quantization inaccuracy is not the only issue that faces Hough-based transforms. The computation slowness for Hough-based transforms is caused by the number of different combinations in the parameter space and the number of edges to be used in the image

space. This different combinations adds extra memory requirement in the accumulator.

In this paper, we propose a method to minimize the number of edges to be used in the image space and at the same time to eliminate the use of parameter space. This improves memory consumption and ellipse detection time. The proposed method uses the approach in [1] to find the ellipse center using different combinations of two points with different values of x_0 in the image space. These two points were separated by a distance apart as described in [1]. However, instead of searching and solving for the rest of the ellipse parameters, the proposed method stores edges that voted for the ellipse center. Direct least square fitting of ellipse proposed in [5] is used to fit the stored edges. A two-dimensional parameter space will be required to store the ellipse location and one additional three-dimensional array to store edges that have voted for the ellipse center. Using only pairs of edges and the ellipse center, the recreation of the original ellipse can be accomplished and ellipse parameters determined using the Direct Least Square Ellipse fitting. The proposed method requires only a two edge accumulator voting system.

The rest of this chapter is organized as follows. In the next section, new proposed algorithm is presented. Section 3 is devoted to the experimental results comparison and analysis. Section 4 contains some concluding remarks.

3.1 Quantization-free parameter space reduction(QFPSD)

The algorithm consists of 5 stages:

- Use two different points, $p_1(x_1, y1)$ and $p_2(x_2, y2)$ with different values of x_0 to locate the ellipse center (x_0, y_0) as described by [1]. Store p_1 and p_2 if they are equidistant to the ellipse center and increase the accumulator for the ellipse center. Otherwise, disregard them. Only a small sample of equidistant pairs is required.

- Locate the highest accumulated ellipse center cell and retrieve the edge points, that voted for the ellipse center.

- Use p_1, p_2 and the ellipse center to project the new additional points p_1' and p_2'.

- Using the Euclidean distance between ellipse points and the ellipse center, the normal distribution can be used to model the presence probability of ellipse edge points.
- Use filtered scattered edges to fit into Direct Least Square Ellipse fitting.

These operations are described in details in each of the following subsection stages.

3.1.1 Locating the ellipse center using two points

The procedure used to locate the ellipse center is the same as described in [1] (see figure 3.1). The ellipse center can be found using the following equation:

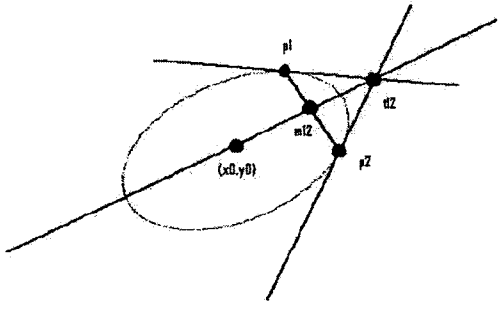


Figure 3.1: Geometrical relationship between the two randomly selected points and their tangent line

$$y_0 = y_m + \frac{AC + 2BD}{2A + BC}(x_0 - x_m) \tag{1}$$

where

$$A = y_1 - y_2$$

$$B = x_1 - x_2$$

$$C = slope_1 + slope_2$$

$$D = slope_1 slope_2$$

$$slope_1 = \frac{y_1 - y_t}{x_1 - x_t}$$

$$slope_2 = \frac{y_2 - y_t}{x_2 - x_t}$$

$$x_m = \frac{x_2 + x_1}{2}$$

$$y_m = \frac{y_2 + y_1}{2}$$

3.1.2 Equidistant points to the ellipse center

In our algorithm, only the pair of edge points that are equidistant to the ellipse, as illustrated in Figure 3.2, are stored in a three-dimensional array and allowed to vote. All other pairs are disregarded and their votes are also being omitted.

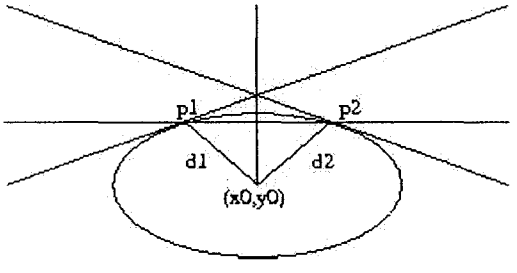


Figure 3.2: p_1 and p_2 equidistant to the ellipse center (x_0, y_0).

If the goal is to store as few points as possible, then only two pairs of unique edge points are required for a perfect half ellipse. Throughout our experiments, it has been proven that a single pair was more than enough to detect the ellipse. This is because by having the ellipse center and four edge points, it is possible to solve the ellipse equation. This pair of points should be located with an angle of $\frac{\pi}{4}$ from any of the ellipse axis.

Normally, in real world images where ellipses are never perfect, a tolerance parameter has to be introduced to detect the elliptical form. This tolerance value can be described as the following where d_1 and d_2 are distances from ellipse edges to the ellipse center.

$$tolerance = |d_1 - d_2|$$

The tolerance value will be zero for a perfect ellipse. For a real world elliptical shape, the tolerance value will be higher than zero. This value can be adjusted depending on the image. Furthermore, not all pairs of equidistant points are needed. Only a small portion of the pairs is needed. Pairs of points can be sampled at a specific interval. This is equivalent as to reduce the number of votes to a smaller scale in the Probabilistic Hough transform.

3.1.3 Locate the ellipse center with the highest accumulated cell and retrieve edges that have voted for the ellipse center

In this stage, there are two main tasks. The first is to locate the ellipse center in terms of the xy coordinate with the highest accumulated cell by traversing a two-dimensional array: $[x_0]_k[y_0]_u$. Once the location of the ellipse center is found, the second task consists of using the location of the ellipse center (x_0,y_0) to retrieve the edges that have voted for the center. These edges can be stored in a three-dimensional array, as illustrated by the following:

$$[x_0]_k[y_0]_u[numberOfEdges]$$

3.1.4 Points projected using p_1, p_2 and ellipse center

Using the original pair of points p_1, p_2 and the ellipse center (x_0,y_0), additional points can be generated. The purpose of generating additional points is to save unnecessary storage space and add ellipse curve projection. This technique is illustrated in Figure 3.3.

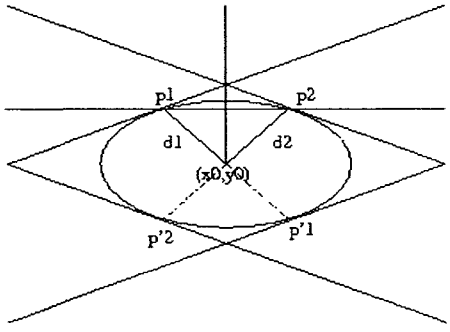
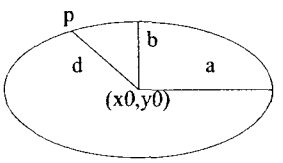


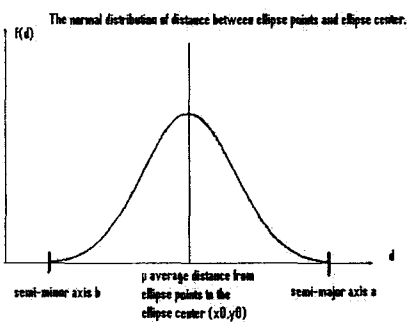
Figure 3.3: Projected points from p1 and p2.

3.1.5 Normal distribution

Using the Euclidean distance d between the ellipse center and ellipse points, the presence of ellipse points can be modeled using the Normal distribution, as illustrated in Figure 3.4. In this stage, the goal is to remove a small quantity of points that do not belong to the ellipse. Points that are beyond the semi-minor and semi-major axis or points that are too close to the ellipse center are removed. Using a plus or minus one standard deviation, most points beyond the bell curve are rejected.



(a) Ellipse



(b) Normal distribution

Figure 3.4: Normal distribution of distance between ellipse points and ellipse center

3.1.6 Direct Least Square Ellipse fitting

In this stage, the fitting of scattered data into the ellipse is performed using [5]. These scattered data are obtained after ellipse points are modeled using Normal distribution. The main idea is to minimize the sum of the distances of the scattered data from the ellipse curve. This method is especially efficient when data or spurious points to be fitted are in imperfect condition, as illustrated in figure 3.5a. The result of the scattered data fitting can be seen in figure 3.5b.

(a) Noisy ellipse (b) Ellipse fitting using Direct least square

Figure 3.5: Direct least square fitting for ellipse

3.2 Experiments

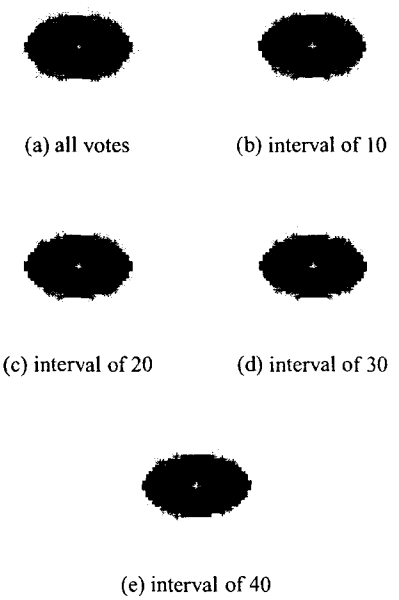
The experiments were conducted using a Dell D500 Centrino 1.3GHz with 1.5Gigabytes of memory running Matlab 2008a under Windows XP 32bit. The experiments were divided into two primary stages: validating the correctness of the new proposed algorithm using synthetic images and testing the algorithm in real world environment. In the validation process, random noise was added to the images to disturb ellipse detection.

Unfortunately, synthetic images can only validate the correctness of the algorithm. In real world images, elliptical shapes are often never in perfect condition. Therefore, the new algorithm must be adaptable to poor elliptical shape. Not having a perfect ellipse is just one of many issues that can arise when detecting elliptical shape. In some cases, elliptical shapes might have other obstructing objects inside itself or partially hidden by the elliptical object itself. To conclude our real world experiments, the new algorithm was tested on traffic sign detection and eye detection. Both of these applications are primarily used in the automotive industry for real-time application. In the traffic sign detection, the detector can warn the driver against danger ahead if the driver has not being diligent on the road. Similarly, in respect of eye detection, the machine can warn the driver if not enough attention has been put on the road, or if fatigue is taking over.

The experiments in the real world environment are carried out with comparisons against other Hough-based transforms, such as the Standard Hough transform (SHT), Probabilistic Hough transform (PHT), Randomized Hough transform (RHT) and Parameter Space decomposition (PSD).

3.2.1 Experiments with synthetic images

Since the proposed algorithm relied on storing edges that voted for the ellipse center, in the next experiment, a demonstration revealing that not all edges are required is given in Figure 3.6. The basic idea is to store edges at a particular interval. In other words, stored edges are never close to each other. An interval of 50 has been used to demonstrate the accuracy of the ellipse detected in Figure 3.7. Figure 3.8 demonstrates that a single pair of edges might be enough to adequately locate the ellipse parameters. Figure 3.9 demonstrates the efficiency of each filter (equidistant and normal distribution) working side-by-side to remove unwanted non-pertinent edges. Overlapping synthetic ellipses were tested in Figure 3.10. Visually fitted ellipse theoretical parameter values and the detected ellipse parameter values using the proposed algorithm for figures 3.9 and 3.10 can be seen in tables 3.1 and 3.2.



(a) all votes (b) interval of 10

(c) interval of 20 (d) interval of 30

(e) interval of 40

Figure 3.6: Sampling interval for the pairs of points

(a) interval of 50 (b) interval of 50 result

Figure 3.7: Accuracy of ellipse using an interval of 50

(a) one pair vote (b) one pair vote with projected points

(c) one pair vote with projected points result

Figure 3.8: Accuracy of ellipse detection using one pair of points

(a) all votes (b) equidistant votes

(c) applying normal distribution (d) direct least square fitting

Figure 3.9: Experimental results to demonstrate both filters

Table 3.1: Theoretical values and QFPSR in figure 3.9d.

Image	Methods	x_0	y_0	a	b	ϕ
Figure 3.9d	Visually fitted	33.8829	37.0013	13.4027	9.5305	-0.0128
	QFPSR	35.8783	37.0828	14.5824	9.9159	-0.0072

(a) Occluded ellipse 1 (b) Occluded ellipse 2

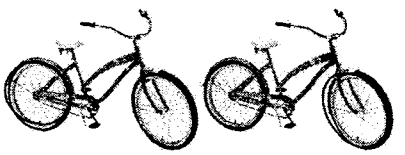
Figure 3.10: Occluded ellipses

Table 3.2: Theoretical values and QFPSR in figure 3.10.

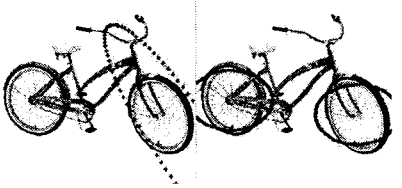
Image	Methods	x_0	y_0	a	b	ϕ
Figure 3.10a	Visually fitted	42.1244	41.9764	17.6388	9.2941	-0.0007
	QFPSR	42.9877	41.6621	16.4914	10.4560	-0.1108
Figure 3.10b	Visually fitted	46.4839	27.4097	16.7158	9.7942	1.5663
	QFPSR	47.0384	25.5777	13.8856	10.3807	1.45

Since elliptical objects are found everywhere, elliptical shapes are put to test in Figures 3.11 and 3.12.

Clearly, in the results on both cases, our algorithm has performed relatively well with more precision. Edges not pertaining to the ellipse were filtered out using equidistant condition and normal distribution.



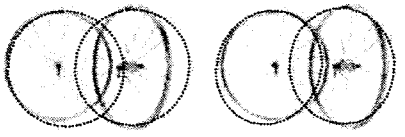
(a) SHT (b) PHT



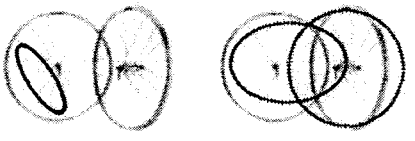
(c) RHT (d) PSD

(e) QFPSR

Figure 3.11: Pink bicycle result



(a) SHT (b) PHT



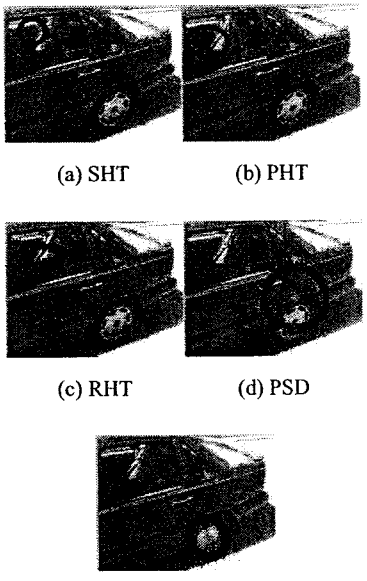
(c) RHT (d) PSD

(e) QFPSR

Figure 3.12: Wheels result

3.2.2 Experiments on elliptical shapes in real world environment

In this section, our algorithm is tested in real world environment. Side-by-side results are illustrated in Figures 3.13, 3.14, 3.15 and 3.16

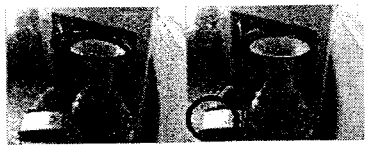


(a) SHT (b) PHT

(c) RHT (d) PSD

(e) QFPSR

Figure 3.13: Left rear of a red volvo 850



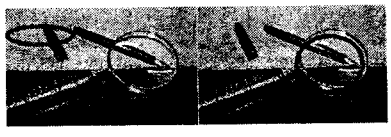
(a) SHT (b) PHT

(c) RHT (d) PSD

(e) QFPSR

Figure 3.14: Antique vase

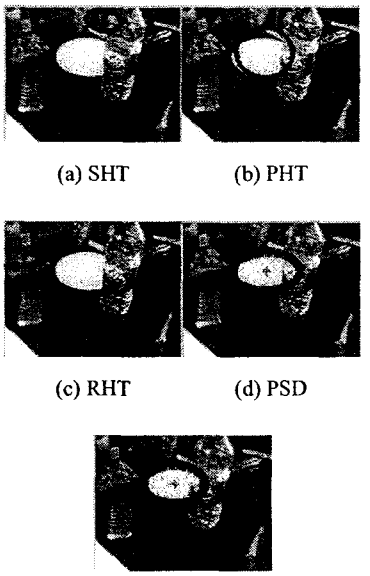
(a) SHT (b) PHT



(c) RHT (d) PSD

(e) QFPSR

Figure 3.15: Magnifying glass



(a) SHT (b) PHT

(c) RHT (d) PSD

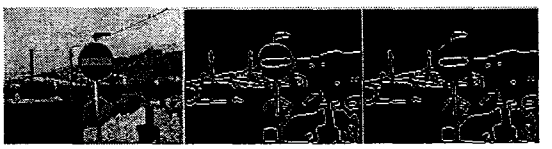
(e) QFPSR

Figure 3.16: Hidden plate

3.2.3 Real world application: traffic sign detection

Detecting the appropriate road warning sign can help drivers reduce accidents on the road. Different road signs are classified with different geometry shapes. Special signs that might require strict attention for drivers are "No Entrance", as illustrated in Figure 3.17, "No right turn", as illustrated in Figure 3.19, "No left turn", as illustrated in Figure 3.21 and "No U turn", as illustrated in Figure 3.23. Comparison of the results can be seen in Figures 3.18, 3.20, 3.22 and 3.24.

The main purpose of this experiment was to test accuracy. To measure how accurate our algorithm is comparing to other Hough-based transforms, ellipses were visually fitted to create theoretical values. Only the highest accumulator cell was retrieved in this experiment. A summary of the experimental results can be found in Table 3.3. Through experimental results, it can be seen that our proposed algorithm had the best result in terms of accuracy and execution time.



(a) Original image (b) Edges (c) Visually fitted

Figure 3.17: Traffic sign 1

(a) SHT (b) PHT (c) RHT

(d) PSD (e) QFPSR

Figure 3.18: Traffic sign 1 results

(a) Original image (b) Edges (c) Visually fitted

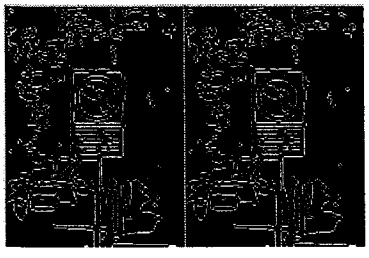
Figure 3.19: Traffic sign 2

(a) SHT (b) PHT (c) RHT

(d) PSD (e) QFPSR

Figure 3.20: Traffic sign 2 results

(a) Original image



(b) Edges (c) Visually fitted

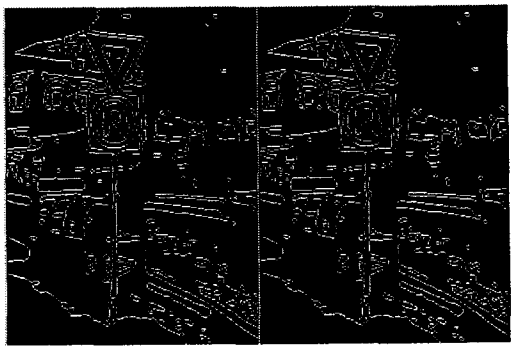
Figure 3.21: Traffic sign 3

(a) SHT (b) PHT (c) RHT

(d) PSD (e) QFPSR

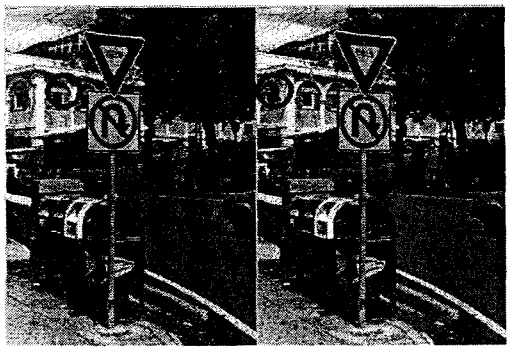
Figure 3.22: Traffic sign 3 results

(a) Original image

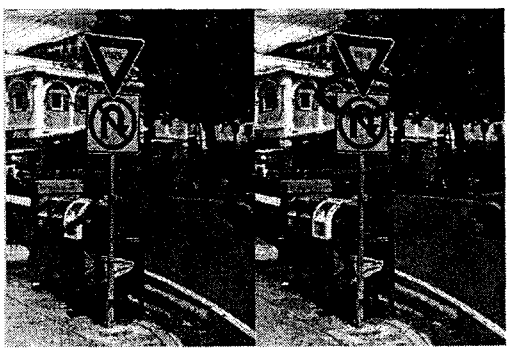


(b) Edges (c) Visually fitted

Figure 3.23: Traffic sign 4



(a) SHT (b) PHT



(c) RHT (d) PSD

(e) QFPSR

Figure 3.24: Traffic sign 4 results

Table 3.3: Value comparison of theoretical, SHT, PHT, RHT, PSD and QFPSR

Image	Methods	x_0	y_0	a	b	φ	time in sec
Traffic Sign 1	Visually fitted	81.2850	47.5423	15.6248	16.4715	-0.0348	N/A
	SHT	79	41	11	10	-0.0873	9271
	PHT	15	73	12	11	-0.0768	1854
	RHT	66	89	14	4	0	7835
	PSD	82	47	30	3	-0.0175	38
	QFPSR	81.5134	47.2630	15.8250	17.1777	-0.9379	27
Traffic Sign 2	Visually fitted	102.7677	39.0177	17.2039	16.1118	-0.0181	N/A
	SHT	92	46	15	13	-0.2618	8164
	PHT	17	76	14	10	-0.0192	1633
	RHT	68	94	61	9	1	744
	PSD	102	38	24	18	-0.0175	23
	QFPSR	101.3711	38.1045	15.3174	17.5491	0.8238	9
Traffic Sign 3	Visually fitted	84.9302	83.1512	19.8230	19.9808	-0.1555	N/A
	SHT	30	68	11	10	-0.3491	12470
	PHT	21	167	14	13	0	2495
	RHT	25	173	47	33	2	225
	PSD	85	114	30	4	-0.0175	107
	QFPSR	83.6187	81.6666	18.2584	17.7879	-0.2095	82
Traffic Sign 4	Visually fitted	97.9580	102.8233	21.5531	20.9893	1.5606	N/A
	SHT	52	74	15	14	-0.3491	36710
	PHT	16	74	15	14	-0.1920	7342
	RHT	1	162	81	45	1	205
	PSD	94	76	30	3	-0.0175	223
	QFPSR	96.9702	102.7695	17.7194	20.2579	-0.4772	198

In reviewing our experimental results, I found that our algorithm has achieved results very close to the theoretical values. Most ellipse parameters values were found within 1%. This 1% error can be attributed to human error since the theoretical results were created visually. In addition, the algorithm was the fastest to finish up the ellipse detection.

3.2.4 Real world application: eye detection

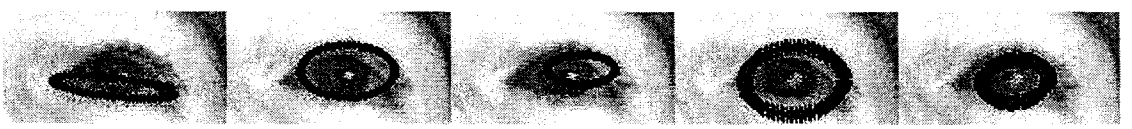
Eye detection presents a good scenario to test the proposed algorithm, since the eye does not have a perfect elliptical shape. Additionally, pupils, irises and sclera generate unwanted disturbance edges. In some of the pictures, eyeglasses frames were also included. Figures 3.25 to 3.36 were used to demonstrate the results of the proposed algorithm against other Hough-based algorithms. These pictures are truly challenging, since the upper left/right curvatures and lower left/right curvatures of the eyes do not necessary reflect to the same ellipse parameters. Locating the best fit ellipse to the eye can be a difficult task. The calculation time for each image is illustrated in Table 3.4.

(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.25: Results of each algorithm for eye 1

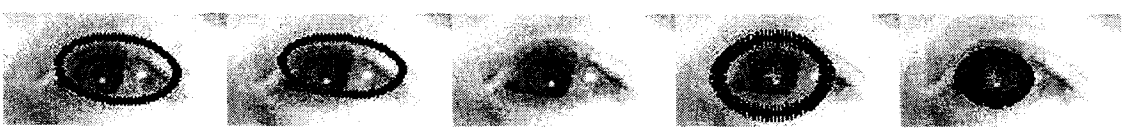
(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.26: Results of each algorithm for eye 2



(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.27: Results of each algorithm for eye 3

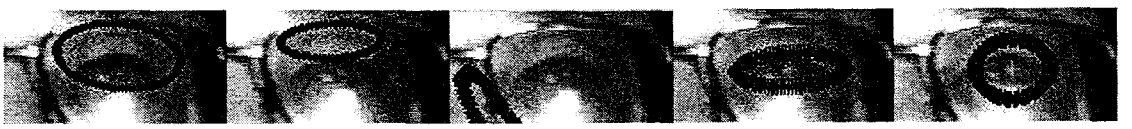


(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.28: Results of each algorithm for eye 4

(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.29: Results of each algorithm for eye 5



(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.30: Results of each algorithm for eye 6

(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

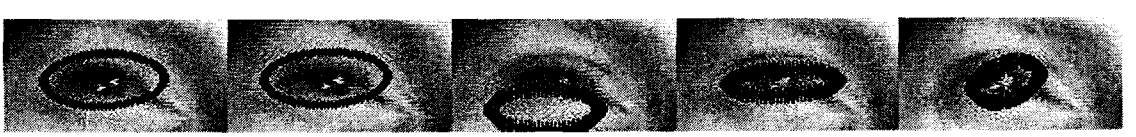
Figure 3.31: Results of each algorithm for eye 7

(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.32: Results of each algorithm for eye 8

(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.33: Results of each algorithm for eye 9

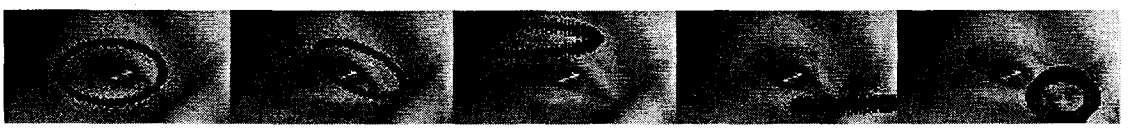


(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.34: Results of each algorithm for eye 10

(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.35: Results of each algorithm for eye 11



(a) SHT (b) PHT (c) RHT (d) PSD (e) QFPSR

Figure 3.36: Results of each algorithm for eye 12

Table 3.4: Calculation time required in SHT, PHT, RHT, PSD and QFPSR for eye1-12

Image	SHT	PHT	RHT	PSD	QFPSR
Eye 1	614 sec	123 sec	386 sec	4.07 sec	2.31 sec
Eye 2	610 sec	126 sec	416 sec	4.67 sec	2.74 sec
Eye 3	630 sec	145 sec	357 sec	4.1 sec	2.27 sec
Eye 4	615 sec	130 sec	360 sec	4.95 sec	2.89 sec
Eye 5	1211 sec	219 sec	598 sec	7.72 sec	4 sec
Eye 6	834 sec	156 sec	456 sec	5.11 sec	2.38 sec
Eye 7	938 sec	178 sec	500 sec	5.69 sec	2.47 sec
Eye 8	741 sec	140 sec	398 sec	4.17 sec	1.98 sec
Eye 9	588 sec	138 sec	400 sec	4.53 sec	2.28 sec
Eye 10	724 sec	173 sec	501 sec	5.66 sec	3.01 sec
Eye 11	627 sec	148 sec	402 sec	4.84 sec	2.60 sec
Eye 12	638 sec	166 sec	412 sec	4.60 sec	2.49 sec
Average time	730sec	154sec	432 sec	5 sec	2.62 sec

3.2.5 Analysis of Quantization-free parameter space reduction

In the new proposed algorithm, there were basically two filters to remove unwanted non-ellipse pertinent edges: one filter focus on the individual edge-pair level and the other focus on the group edge level. In the first filter, individual pairs of edges were targeted and rejected if the equidistant condition was not met. In the second filter, edges that were too far away compared to the average distance to the ellipse center were

completely removed in order to improve ellipse accuracy.

Additionally, the new proposed algorithm did not add any major loops to the elliptical shape detection. The complexity stayed at $cO(n^2)$ where c is a fraction. In terms of memory requirements, only a two dimensional accumulator was needed to store the ellipse center. A three dimensional array was needed to store the edge points from the image. Using the Direct Least Square fitting of the ellipse, the algorithm itself could validate the ellipse center (x_0, y_0) and eliminate errors in the parameter space quantization. In an elliptical shape where the ellipse is not perfect, the Direct Least Square fitting of the ellipse can be used to best describe the ellipse by minimizing the distance between ellipse noise edges.

When comparing the new proposed algorithm with directional information parameter space decomposition, the new proposed algorithm had few advantages. In the proposed algorithm, once the ellipse center was located, ellipse parameters were found by fitting into the Direct least square method. Meanwhile for parameter space decomposition, additional combinations of ellipse parameters were required to locate the rest of the parameter values. The number of additional loops was proportional to the number of ellipses found.

The new proposed method also eliminates input error propagation in the original Parameter Space Decomposition. This input error propagation affects values of semi-major axis a, semi-minor axis b and angle ϕ of the ellipse orientation. Additionally, the values of semi-major axis a, semi-minor axis b and angle ϕ of the ellipse orientation were quantized. In the current proposed algorithm, no ellipse parameters were quantized.

In terms of accuracy, unlike any Hough-based transform, the new proposed algorithm uses original edges. This translates into higher accuracy, as illustrated in the experiments. The accumulator for the ellipse center was only used to locate and collect evidence in order to decide which edges were needed in order to be fed into direct least square method. Therefore, no parameters of the ellipse were actually quantized. Quantized accumulator cells only led to approximated values for the ellipse parameters.

Additionally, to detect the ellipse, the range of values of the semi-major axis a, semi-minor axis b and angle ϕ of ellipse orientation must be given the opportunity to vote. If, for a particular reason, the range of these values were skipped, accuracy on the ellipse is compromised, or failure in ellipse detection will occur.

In any Hough-based transform, the local maxima in the accumulator are often surrounded by other

closed values, if they are not equal to the local maxima itself, as illustrated in Figure 3.37. This phenomenon occurs when ellipse edges are noisy. Because Direct Least square methods have been used to locate ellipse parameters, the center local maxima are truly found.

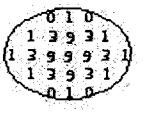


Figure 3.37: Multiple local maxima

CHAPTER 4

Conclusions

The most popular method for ellipse detection in digital imagery is currently the Hough Transform. The weaknesses in Hough-based transforms include inaccuracy due to quantization and memory consumption due to five-dimensional parameter space. As a result, we have proposed a new approach for ellipse detection. Unlike any of the previous Hough-based transforms, where five-dimensional parameters space is required, in the proposed algorithm, the ellipse parameter values were located by reconstructing the original ellipse using a pair of edges and the ellipse center. This method has demonstrated that a five-dimensional parameter space is no longer required by an ellipse. Through rigorous testing using a large number of experimental results involving synthetic images and different real world applications, such as traffic sign detection and eye detection, our proposed algorithm has obtained higher accuracy, lower memory consumption and quicker calculation times.

Unlike previous generations of Hough-based transforms, the focus was based on locating quantized parameter values, the new algorithm focused on the edges themselves. Any edges that have the potential of being part of an ellipse were retained for locating the ellipse parameter values. Because focus was placed on the edges themselves, accuracy on the ellipse has greatly improved. Also, since only edges belonging to the ellipse were kept, due to the filters equidistant and normal distribution, the proposed algorithm was really robust to noise. Furthermore, the new algorithm, did not add any additional major iterations to the original parameter space decomposition algorithm. The complexity was kept to $cO(n^2)$ where c is a fraction. As for the memory consumption for storing the edges, only a small portion was required. Results have been proven to be quite accurate.

In the future works, additional improvement can be done regarding the number of edges to be stored for the potential ellipse based on the ellipse center. Ideally, edges should be stored evenly across the ellipse perimeter. This additional improvement can improve greatly the detected ellipse parameters. Additionally, effective edges storage can improve greatly memory consumption. Furthermore, technically speaking, having two edges p_1 and p_2, the projected edges p_1' and p_2' and the ellipse center, ellipse equation should be solvable without the need of Direct Least Square method.

List of References

[1] Alberto S. Aguado, Eugenia, and Mark S. Nixon. On using directional information for parameter space decomposition in ellipse detection. *Pattern Recognition*, 29(3):369–381, March 1996.

[2] J. Batista. A drowsiness and point of attention monitoring system for driver vigilance. In *Intelligent Transportation Systems Conference*, pages 702 – 708, 2007.

[3] Y.Y. Tsai C.M. Wang, N.C. Hwang and C.H. Chang. Ellipse sampling for monte carlo applications. *Electronics Letters*, 40(1):21–22, 2004.

[4] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[5] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[6] M.A.; Martm-Gorostiza-E. Garcia-Garrido, M.A.; Sotelo. Fast traffic sign detection and recognition under changing lighting conditions. *Intelligent Transportation Systems Conference*, 17-20:811 – 816, 2006.

[7] Nikos Grammalidis and Michael G. Strintzis. Head detection and tracking by 2-d and 3-d ellipsoid fitting. In *CGI '00: Proceedings of the International Conference on Computer Graphics*, page 221, Washington, DC, USA, 2000. IEEE Computer Society.

References

[8] N. Guil and E. Zapata. Lower order circle and ellipse hough transform. *Pattern Recognition*, 30:1729–1744, 1997.

[9] F. Fraunhofer IDMT-Ilmenau Hardzeyeu, V. Klefenz. On using the hough transform for driving assistance applications. In *4th International Conference on Intelligent Computer Communication and Processing*, pages 91–98, 2008.

[10] Ric Heishman, Zoran Duric, and Harry Wechsler. Using eye region biometrics to reveal affective and cognitive states. *Computer Vision and Pattern Recognition Workshop*, 5:69, 2004.

[11] P.V.C. Hough. Method and means for recognizing complex patterns. *U.S. Patent 3069654*, 1962.

[12] S. A. Inverso. Ellipse detection using randomized hough transform. *http://www.saminverso.com/res/vision/EllipseDetection.pdf*, May 2006.

[13] Qiang Ji and Xiaojie Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, 2002.

[14] M. Imran Khan and A. Bin Mansoor. Real time eyes tracking and classification for driver fatigue detection. In *ICIAR '08: Proceedings of the 5th international conference on Image Analysis and Recognition*, pages 729–738, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[16] V.F. Leavers. Survey: Which hough transform? *CVGIP image understanding*, 58(2):250–264, September 1993.

[17] Yiwu Lei and Kok Cheong Wong. Ellipse detection based on symmetry. *Pattern Recognition Letters*, 20(1):41–47, 1999.

[18] Xia Liu, Fengliang Xu, and K. Fujimura. Real-time eye detection and tracking for driver observation under various light conditions. *Intelligent Vehicle Symposium, 2002. IEEE*, 2:344–351 vol.2, 2002.

[19] Wei Lu and Jinglu Tan. Detection of incomplete ellipse in images with strong noise by iterative randomized hough transform (irht). *Pattern Recognition*, 41(4):1268–1279, 2008.

[20] E. Lutton and P. Martinez. A genetic algorithm for the detection of 2d geometric primitives inimages. *Proceedings of the 12th IAPR International Conference on Computer Vision and Image Processing, Pattern Recognition.*, 1:526–528, 1994.

[21] F. Mai, Y. S. Hung, H. Zhong, and W. F. Sze. A hierarchical approach for fast and robust ellipse extraction. *Pattern Recognition*, 41(8):2512–2524, 2008.

[22] Tom Mainzer. Genetic algorithm for traffic sign detection. *Applied Electronic*, 2002.

[23] Robert A. Mclaughlin. Randomized hough transform: improved ellipse detection with comparison. *Pattern Recognition Letters*, 19(3-4):299–305, 1998.

[24] Derek Pao, H. F. Li, and R. Jayakumar. A decomposable parameter space for the detection of ellipses. *Pattern Recognition Letters*, 14(12):951–958, 1993.

[25] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani. Robust road sign detection and recognition from image sequences. *Intelligent Vehicles '94 Symposium, Proceedings of the*, pages 278–283, 1994.

[26] A. Pietrowcew. Face detection in colour images using fuzzy hough transform. *Opto-Electronics Review*, 11(3):247–251, 2003.

[27] G. Roth and M. D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(9):901–905, 1994.

[28] Chun ta Ho and Ling-Hwei Chen. A fast ellipse/circle detector using geometric symmetry. *Pattern Recognition*, 28(1):117–124, 1995.

[29] Yong-Hui Huang Bao-Chang Pan Sheng-Lin Zheng Jianjia Pan Yuanyan Tang. Lip-reading detection and localization based on two stage ellipse fitting. In *Wavelet Analysis and Pattern Recognition, 2008. ICWAPR '08. International Conference on*, volume 1, pages 168–171, Hong Kong, 2008.

References

[30] Klaus Toennies, Frank Behrens, and Melanie Aurnhammer. Feasibility of hough-transform-based iris localisation for real-time-application. *Proceedings on 16th International Conference on Pattern Recognition*, 2:1053–1056, 2002.

[31] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: randomized hough transform (rht). *Pattern Recognition Letters*, 11(5):331–338, 1990.

[32] H. K. Yuen, J. Illingworth, and J. Kittler. Detecting partially occluded ellipses using the hough transform. *Image and Vision Computing*, 7(1):31–37, 1989.

[33] Si-Cheng Zhang and Zhi-Qiang Liu. A robust, real-time ellipse detector. *Pattern Recognition*, 38(2):273–287, February 2005.