Point to Ellipse Distance: A Binary Search Approach*

Zhikai Wang† http://www.heteroclinic.net

Montreal, Quebec, Canada e-mail: wangzhikai@yahoo.com *url:*

Abstract: We give an algorithm to compute the shortest distance from an outside point to an ellipse. The ellipse should be transformed to the orthogonal form. We confine the computation in a single quadrant which is bounded by two orthogonal unit vectors. We locate a special normal vector by iterations. At each iteration, our searching space is halved. So the algorithm is is $\mathbf{O}(\log n)$. We also give a way to span an ellipse shape with a generating set. This method avoids using trigonometric functions and offers level of detail setting.

Keywords and phrases: composite number, factoring.

Contents

1. Introduction

An ellipse is a convex shape. An ellipse in *orthogonal* position follows

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{1.1}$$

where $x, y, a, b \in$ R and we take $a > 0, b > 0$. We also have the *generic* form of an ellipse

$$c_1x^2 + c_2y^2 + c_3xy + c_4x + c_5y + c_6 = 0 \tag{1.2}$$

for c_1, c_2, c_3, c_4, c_5 and $c_6 \in$ R. The transformation between the two forms is widely discussed, for example in [1]. So we take the ellipse in form of Equation (1.1). Now, the question is that, given a point outside an ellipse, how to get the shortest distance to the ellipse?

*This latex file is built as per latex template at http://www.e-publications.org. Any dispution upon the usage of the above template, please contact the author.

†Zhikai Wang graduated from computer science department of Concordia University, Canada with a master degree.

A classical way is to use Newton's method and solve and quadratic equation of order four. In [3], the distance is approximated by projecting points to the unit circle and an affine transformation back. Here, we would rather propose a new approach than do a survey discussing various methods. In addition, we give a complexity expectation of our algorithm.

2. Span a 2D Ellipse with a Generating Set

Let $\mathbf{e}_1 = (1,0)$, $\mathbf{e}_2 = (0,1)$; $\{\mathbf{e}_1, \mathbf{e}_2\}$ is the *standard basis* for $\mathbf{F}^2$. An ellipse can be spanned by a generating set that are the linear combinations of $\mathbf{e}_1$ and $\mathbf{e}_2$. Let the generating set have l levels, we have $n+1$ generating vectors, $n = 2^{l+1}$, starting from $\mathbf{g}_0$ to $\mathbf{g}_n$. We let $\mathbf{g}_n \equiv \mathbf{g}_0$. At the first level, we have five vectors $\mathbf{g}_0 \equiv (1,0)$, $\mathbf{g}_{n/4} \equiv (0,1)$, $\mathbf{g}_{2n/4} \equiv (-1,0)$, $\mathbf{g}_{3n/4} \equiv (0,-1)$ and $\mathbf{g}_{4n/4} \equiv \mathbf{g}_0$. From the second level on, at a given level i, we add the same number $m = 2^i$ of new generating vectors as one level upper. Each new vector is added between the two adjacent upper level vectors. The new vector is the vector sum of them then normalized. That is

$$\begin{aligned} \mathbf{g}_{\frac{n}{2m}+\frac{jn}{m}} &\equiv \mathbf{g}_{\frac{jn}{m}} + \mathbf{g}_{\frac{(j+1)n}{m}}, \\ \mathbf{g}_{\frac{n}{2m}+\frac{jn}{m}} &\equiv \mathbf{g}_{\frac{n}{2m}+\frac{jn}{m}} / |\mathbf{g}_{\frac{n}{2m}+\frac{jn}{m}}|. \end{aligned} \tag{2.1}$$

Starting from $(0,0)$, suppose a ray along direction of $\mathbf{g}$, for $\mathbf{g} \equiv (g_x, g_y)$ and $\mathbf{g}$ is normalized. The intersection with the ellipse is point $\mathbf{P} \equiv (p_x, p_y)$ and t is the distance from $\mathbf{P}$ to $(0,0)$. We have

$$t = \sqrt{1/(g_x^2/a^2 + g_y^2/b^2)} \tag{2.2}$$

and

$$\mathbf{P} \equiv (tg_x, tg_y). \tag{2.3}$$

In such a way, we can get the corresponding point of each *ray*, starting from the origin and along the direction of each generating vector. Linking the points sequentially as a loop, we get an ellipse. Figure (2.3) shows some ellipses generated by this method using a MATLAB script.

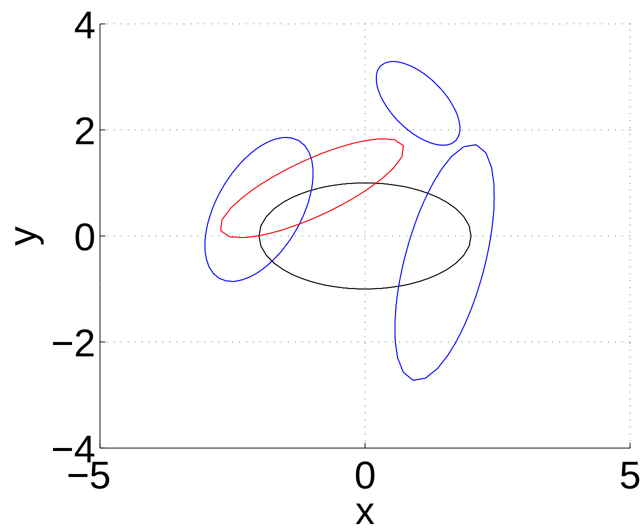

Fig 1. *Some ellipses spanned by the generating set.*

3. Computing the Shortest Distance

Here, we give the following algorithm

Algorithm pt2ellipse($\mathbf{E}$,$\mathbf{Q}$)
Input: $\mathbf{E}$, an ellipse in the orthogonal form; $\mathbf{Q}$, a point outside the ellipse.
Output: $\mathbf{P}$, a point on the ellipse; d is the shortest distance and $d = \|\mathbf{PQ}\|$.
Decide the quadrant $\mathbf{Q}$ is in thus two bounding unit vector $\mathbf{g}_1$, $\mathbf{g}_2$, counter-clockwise.
Let $\mathbf{g}_3 \equiv \mathbf{g}_1 + \mathbf{g}_2$, $\mathbf{g}_3 \equiv \mathbf{g}_3/|\mathbf{g}_3|$.
Along $\mathbf{g}_3$, find a point $\mathbf{P}'$ on $\mathbf{E}$.
Compute the normal vector $\mathbf{v}$ of $\mathbf{P}'$.
while $\mathbf{Q}$ is not on $\mathbf{v}$ from $\mathbf{P}'$ **do**
  if $\mathbf{Q}$ is to the left of $\mathbf{v}$ from $\mathbf{P}'$ **then**
    $\mathbf{g}_1 \equiv \mathbf{g}_3$.
  else
    $\mathbf{g}_2 \equiv \mathbf{g}_3$.
  end if
  Let $\mathbf{g}_3 \equiv \mathbf{g}_1 + \mathbf{g}_2$, $\mathbf{g}_3 \equiv \mathbf{g}_3/|\mathbf{g}_3|$.
  Along $\mathbf{g}_3$, find a point $\mathbf{P}'$ on $\mathbf{E}$.
  Compute the normal vector $\mathbf{v}$ of $\mathbf{P}'$.
end while
$\mathbf{P} \equiv \mathbf{P}'$; $d = \|\mathbf{PQ}\|$.
return a structure containing $\mathbf{P}$ and d.

It is simple to deal with the degenerated cases that $\mathbf{Q}$ happens to on the ray along the coordinate axes. For a point (x, y) on an ellipse, the normal vector is $(x/a^2, y/b^2)$. We also use the following proposition.

Proposition 1. If we have a directed line through two points $\mathbf{p}$ and $\mathbf{q}$. Let $\mathbf{p} \equiv (p_x, p_y)$, $\mathbf{q} \equiv (q_x, q_y)$, and $\mathbf{r} \equiv (r_x, r_y)$. We define a matrix

$$\mathbf{D} \equiv \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}. \tag{3.1}$$

If $det(\mathbf{D}) > 0$ then $\mathbf{r}$ lies left of the line. If $det(\mathbf{D}) < 0$ then $\mathbf{r}$ lies right of the line. If $det(\mathbf{D}) = 0$ then $\mathbf{r}$ lies on the line.

This proposition is quoted from [2]. The proof is simple and omitted. In the above algorithm, we see in each iteration the searching space is halved. Thus the algorithm is $\mathbf{O}(\log n)$. We also illustrate this algorithm in Figure(2). In the figure, we try to compute the shortest distance from point $(4.0, 5.0)$ to ellipse $\frac{x^2}{3.75^2} + \frac{y^2}{1.25^2} = 1$. After six iterations, $\mathbf{P'}$ converges to $\mathbf{P}$ shown as the small red circle on the ellipse. $\mathbf{Q}$ is shown as the red circle outside the ellipse. The short blue segments show the normal vectors on the ellipse.

We also give the numerical results in Table(1). The second initial condition

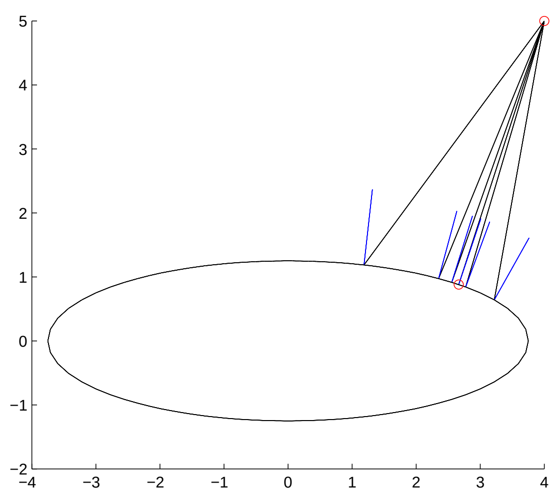

FIG 2. *An example computational result.*

uses the approximation method of []. For such initial condition, we can save some number of iterations. The distance computed singlely with the approximation method of [] is 4.351877438. We can compare this result to our results. The computation is done by MATLAB scripts. The accuracy is set by MATLAB

$\lvert det\ \mathbf{D}\rvert_{max}$	$n = \frac{1}{\lvert det\ \mathbf{D}\rvert_{max}}$	$log(n)$	Initial condition			
			$\mathbf{g}_3 \equiv (1/2, 1/2)$		$\mathbf{g}_3 \equiv (\frac{a\cdot\mathbf{Q}_x}{\lvert\mathbf{Q}\rvert}, \frac{b\cdot\mathbf{Q}_y}{\lvert\mathbf{Q}\rvert})$	
			Iterations	Distance	Iterations	Distance
1e-1	1e+1	1	6	4.331434802	4	4.331305541
1e-2	1e+2	2	8	4.331308858	4	4.331305541
1e-3	1e+3	3	13	4.331304615	12	4.331304617
1e-4	1e+4	4	16	4.331304614	15	4.331304614
1e-5	1e+5	5	19	4.331304614	17	4.331304614
1e-6	1e+6	6	23	4.331304614	22	4.331304614

TABLE 1
$\lvert det\ \mathbf{D}\rvert_{max}$ vs. iteration times

command 'digits(10)'. It shows the time complexity of our algorithm is $\mathbf{O}(\log n)$.

Our algorithm may be applied to an n-dimensional. Suppose a point $\mathbf{Q}$ in n-dimensional space and a algebraic body

$$\frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} + \cdots + \frac{x_n^2}{a_n^2} = 1, \quad \text{for } x_1,\ x_2,\ \cdots, x_n \text{ and } a_1,\ a_2,\ \cdots, a_n \in \mathbf{R}, \tag{3.2}$$

we can divide the n-space into 2^n subspace each bounded n-hypersurface. It is easy to find which subspace $\mathbf{Q}$ is in. Adding a auxilary ray, the subspace can be divided further into n sub-subspace. We can see Proposition (1) is in fact a kind of ray-tracing method. Such tracing can be introduced to an n-dimensional point to a hypersurface. A normal vector passing $\mathbf{Q}$ can finally be obtained by dividing and tracing.

4. Conclusions

We give a method to span an ellipse using a generating set. In computer graphics, we offer a new way to draw level of detail of a circle, an ellipse, a sphere, an ellipsoid *etc.*. We also give an algorithm to locate the shortest distance from a point to an ellipse. The algorithm is $\mathbf{O}(\log n)$. We show this result in our numerical experiments. The scripts used for this text can be obtained from the authors.

References

[1] Stephen H. Friedberg, Arnold J. Insel, and Lawrence E. Spence. *Linear Algebra*, Fourth Edition, Prentice, New Delhi, 2007.
[2] Mark de Berg, marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational geometry*, Second Revised Edition, Springer, 2000.
[3] Karel Zimmermann and Tomas Svoboda. *Approximation of Euclidean Distance of Point from Ellipse*. Czech Technical University, Prague, Czech Republic.
ftp://cmp.felk.cvut.cz/pub/cmp/articles/zimmerk/Zimmermann-TR-2005-23.pdf