KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

Computer Graphics

(ICS – 535)

Term Project Report

|Performance And Accuracy Improvements In Ellipse Drawing Algorithms

SUBMITTED TO

Dr. Muhammad Sarfraz

BY

Syed Manzoor Hussain

Student id # 220518

Abstract

The paper presents three existing algorithms for ellipse drawing, Desilva's algorithm, Kappel's algorithm, Van-Aken's algorithm, and the cases in which they fail are cited and two new algorithms for ellipse drawing are presented which provide solutions to the cases in which the three mentioned algorithms fail. Bresenhams ellipse drawing algorithm has also been implemented and this project aims to present an analysis of these six algorithms on the basis of the comparisons that are being made in each algorithm, i.e., the number of times loops are executed, and also the number of time decision variables, d1 and d2 are calculated (which are used to plot the next pixel) for each of the algorithms.

1. Introduction

The traditional approach to draw ellipses has been to rely on slope calculations. The well-known problem with this approach is that the slopes are not exact but merely approximations on the neighboring grid points. Three algorithms will be considered briefly. The first one, given by Desilva combines the original development by pitteway and later by Van-Aken and Kappel.. It calculates the slope at next midpoint rather that nearest grid points. The algorithm developed by Kappel calculates the slope components on the nearest grid points.. The third one developed by Van-Aken adapts a different approach for handling octant change, it calculates both decision variables d1 and d2 in a single procedure for the whole arc and hence no slope approximation is necessary.

In summary this report presents the cases in which the three existing algorithms fail and the reasons as to why such failures occur will be briefly discussed. The next approach will be to present two algorithms for ellipse. The aim of the first algorithm is accuracy as well as efficiency for centered, axes aligned ellipses. The second algorithm focuses on accuracy, symmetry and potential for generalizations. An efficiency analysis will demonstrate the merits of the algorithms, the analysis based on the comparisons that are being made in each algorithm, i.e., the number of times loops are executed, and also the number of time decision variables, d1 and d2 are calculated (which are used to plot the next pixel) for each of the algorithms.

This Report is organized in 5 sections, the first is introduction, the second section gives the basic construction of the algorithms, the section 3 discusses the cases of failures, and the section 4 discusses the two proposed algorithms and section 5 gives the analysis of these algorithms. Finally there is conclusion

2. The Construction of the Existing Algorithms: -

The equation of the ellipse positioned on the origin of the coordinate space is

$$x^2/a^2 + y^2/b^2 = 1 \quad \text{(figure 1)}$$

The ellipse is draw taking the following idea: -Starting from x=0and y=b and in clockwise direction.

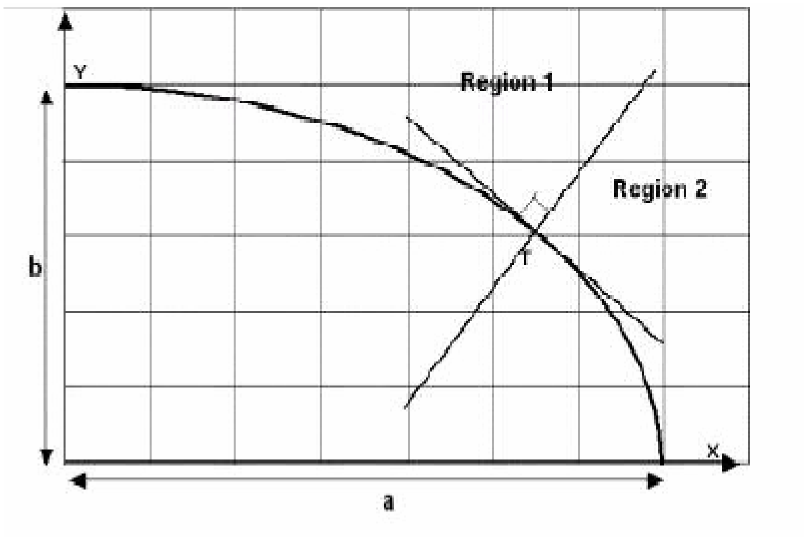


Figure 1. The curve is divided by normal to the tangent at T. The slope in region 1 is greater than -1 whereas in region 2 it is less than –1.

In Region 1: - x is monotonically increasing while y remains unchanged or decrement downward. i.e, $dy/dx >=1$ or $dy >= -1$ since $dx=1$ (increasing in unit steps) the next pixel is plotted at (x+1 , y -0.5).

In Region 2: - y is monotonically decreasing while x may remain same or incrementing the clockwise direction. i.e, $dy/dx <=-1$ or $dy = -1$ and $dx >=1$, and the next pixel is plotted at (x+ 0.5 , y-1).Octant change procedure telling the difference between region 1 and 2 can be accomplished as $dy/dx= -(b^2x / a^2y)$. This is calculated from the ellipse equation $b^2x^2 + a^2y^2 – a^2b^2$ .At the boundary between region 1 and 2 $dy/dx =-1$ and therefore $b^2x = a^2y$, thus we move out of region 1 . The difference in Desilvas algorithm when compared to kappels and van-akens algorithm is that they start ellipse drawing in anti –clockwise direction. Van –Akens algorithm calculates and includes decision parameters d1 and d2 in the procedure for region 1 rather than calculating the slope.

3. Cases of Failures

Three cases of failures are considered and the reasons for the failures are cited.

CASE 1: - This occurs for all the three algorithms, and it occurs when a/b or b/a (where a and b are major and minor axes respectively) are very large. The algorithms halt prematurely leaving the last pixel or last few pixels, undetected.
Ex: - when a =10, b=1 or when a=40, b=2. This is shown in figure 2

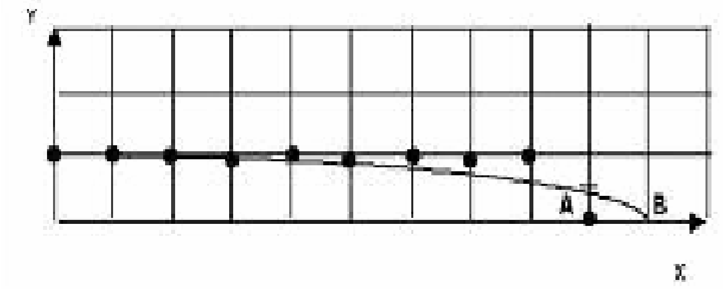


Figure 2. On arriving at A, the algorithms (D, K and A) halt, leaving pixel B undetected. For a larger than a certain value more pixels are missed.

Reason: - This happens because of the terminating conditions in WHILE loop for region 2, because as the procedure for region 2 is called, Y is already zero and the algorithm does not enter loop2.
Remedy: - We can add a simple FOR loop just after the second WHILE loop, thus forcing the algorithm to continue moving along x (or y) axis and assigning zeroes to y (or x) until x=a (or y=b).
CASE 2:- The algorithm enters octant 2 when it should not enter (i.e., it enters prematurely). This is shown in fig 3.

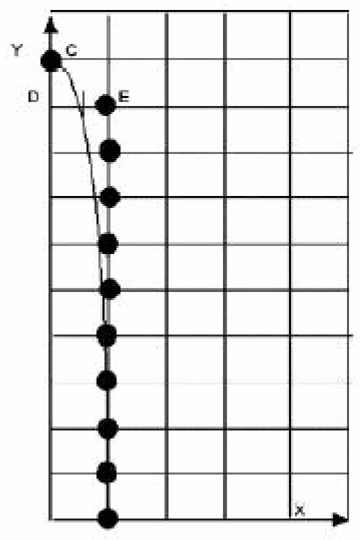
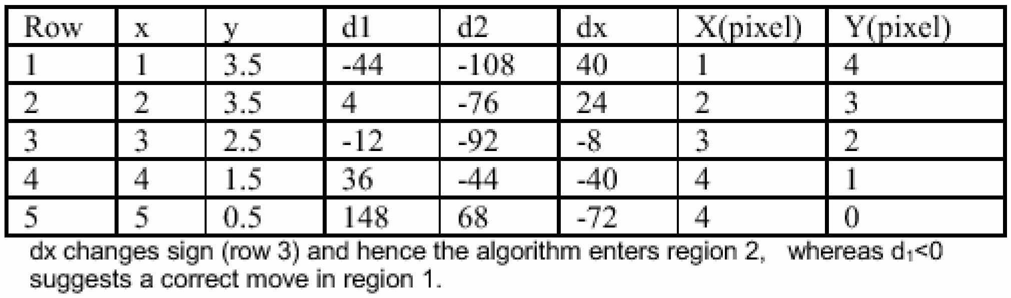


Figure 3. Starting at C, version K misses D and picks E.

Reason: - The reason is that it is due to sensitivity of the octant change conditions in sharp corners.
EX:- b =10 and a=1 the algorithm picks a different pixel than desired .
Remedy: - This is purely due to inaccuracy of slope approximations. Kappels algorithm performs correctly in this case. But there is no definite solution for best slope approximation.

CASE 3: - This occurs for smoothly varying curves, the Kappels algorithm fails in this case .To illustrate the problem, here is the sample output by desilvas algorithm, for a = 4 and b = 4 ,we see in row 3 dx is changing from positive to negative and hence the procedure for region 2 is called for. But we also observe in row 3 that d1 is still negative and therefore next point is (x+1,y) but procedure for region 2 forces a wrong move by selecting (x+1,y-1) as shown in table below,



Row	x	y	d1	d2	dx	X(pixel)	Y(pixel)
1	1	3.5	-44	-108	40	1	4
2	2	3.5	4	-76	24	2	3
3	3	2.5	-12	-92	-8	3	2
4	4	1.5	36	-44	-40	4	1
5	5	0.5	148	68	-72	4	0

dx changes sign (row 3) and hence the algorithm enters region 2, whereas $d_1<0$ suggests a correct move in region 1.

Reason: - This is due to the sensitivity of the octant change procedure where the slopes are calculated at next midpoint rather than nearest grid point. Kappels algorithm is safer as it calculates slope at grid points.

The later two cases of failures cannot be fixed but require radical and an entirely different approach in octant change detection procedures. One way to overcome these failures is to divide the curve into two regions separated by a point, say T where dy/dx=1 and scan convert them as separate arcs. The first algorithm, which is implemented, will present this.

4. **Proposed algorithms: -** Two algorithms have been proposed, which are built on the standard algorithm (midpoint) with a little advancement.

Algorithm 1: -
The main goal of the presentation of this algorithm is to find a way other than the traditional approach of relying on slope calculations for detecting octant changes. For this the first quarter of the ellipse is divided into two arcs as before but the coordinates of the tangent will be used as the dividing point between the two regions. The x-coordinate of the region is calculated as $X_t = a^2/\text{sqrt}\ (a^2+b^2)$ where t is the tangent, so while $x < X_t$ region 1 is drawn and while y>0 region 2 is drawn.

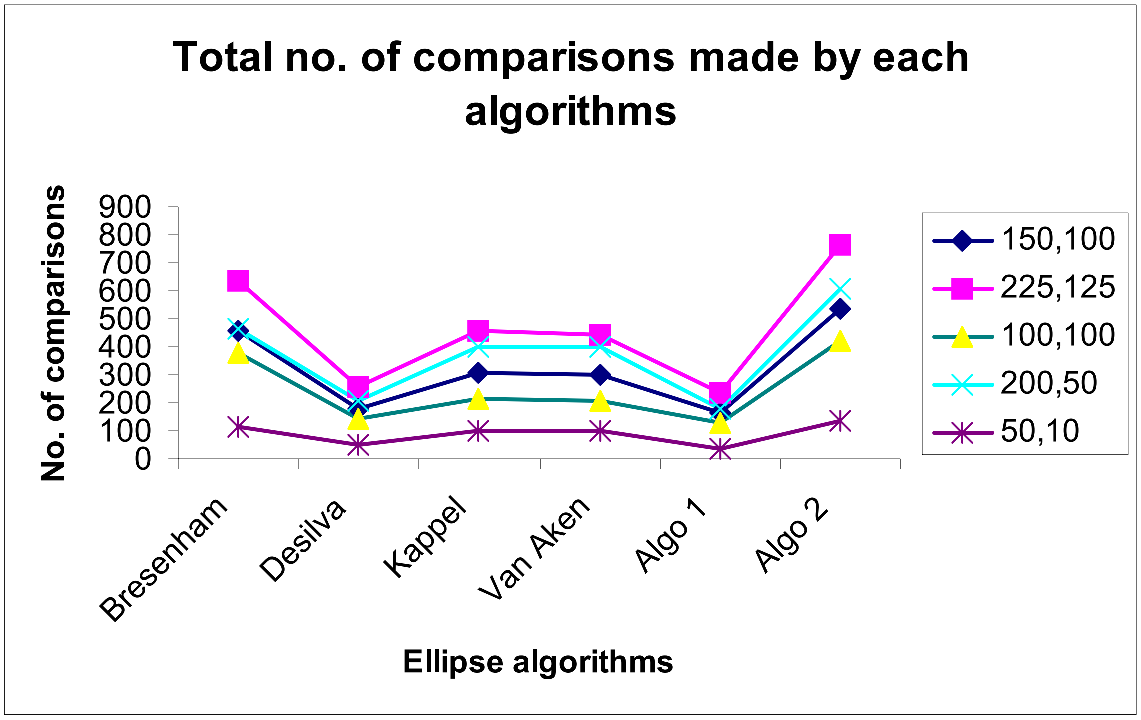
Algorithm 2: - This algorithm is based on the Van-Aken algorithm i.e., to calculate both decision variables d1 and d2 simultaneously and include them in a single procedure. As in the previous algorithm (algorithm 1) no

slope calculation is done. The candidate pixels are decided based on the sign of both decision variables simultaneously, Hence there are a lot of comparisons being made in this algorithm as will be proved in the analysis part.

5. Project implementation and Analysis

The implementation of all the six algorithms is in the Appendix. An analysis of the algorithms have been made, the following table shown the total number of comparisons each algorithm is making, the first row represents the major and minor axes respectively.

	150,100	225,125	100,100	200,50	50,10
Bresenham	460	639	382	462	112
Desilva	182	259	143	208	53
Kappel	307	456	214	402	103
Van Aken	298	445	205	397	100
Algo 1	162	233	127	179	39
Algo 2	535	766	421	607	139

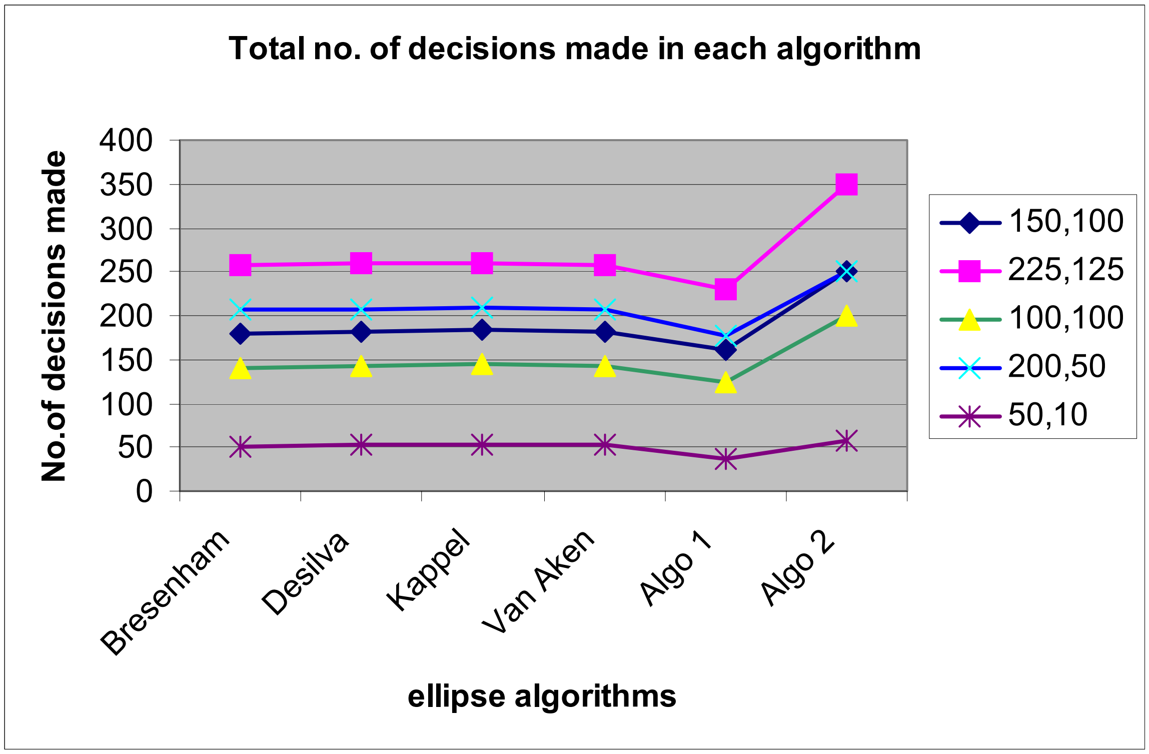


The result from the above graph suggests that Desilva and algorithm1 makes the least number of comparisons than Van–Aken and Kappel followed by Bresenham. The maximum numbers of comparisons are being made by algorithm 2.

Another analysis is made on the basis of the number of time decision variables, d1 and d2 are calculated for each of the algorithms. This comparison is shown in the table below.

	150,100	225,125	100,100	200,50	50,10
Bresenham	180	257	141	206	51
Desilva	182	259	143	208	53
Kappel	183	260	144	209	54
Van Aken	181	258	142	207	52
Algorithm 1	160	231	125	177	37
Algorithm 2	250	350	200	250	57

This is shown Graphically as follows:-



Again here the least number of decisions are made by algorithm 1 whereas Bresenham, Desilva, Van-Aken and Kappel algorithms make an equal number of decisions. Algorithm 2 is making the maximum numbers of decisions as evident from the above graph.

Conclusion: - It can be concluded from the above graphs that algorithm 1 and Desilvas algorithm gives the best performance than the other algorithms followed by Bresenham, Kappel and Van-Aken. The algorithm 2 makes a lot of comparisons and decisions than other algorithms.

References: -

1. Computer Graphics, *Hearn and Baker*