

**Computer Graphics**  
**Prof. Sukhendu Das**  
**Dept. of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**  
**Lecture - 15**  
**Scan Converting Lines, Circles and Ellipses**

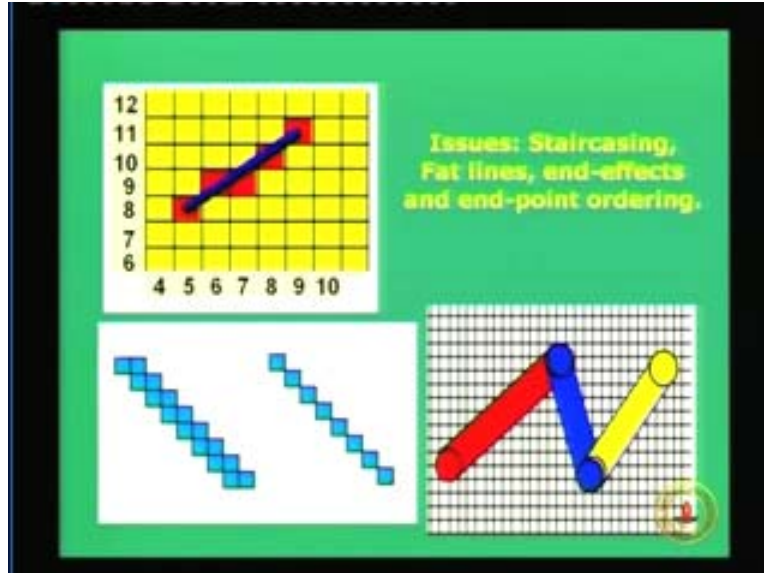
Hello and welcome again to all of you to the lecture sequence and the course on computer graphics. In the last two lectures we have discussed the algorithm for line drawing in which we had seen the Bresenham's algorithm on mid point criteria to evaluate the next mid point and to find out whether the next selected point is among one out of the east or north east pixels to be put on the line. And remember we are working on a discretized environment or digitized space as you may call it and we were also working on the constraint first that the algorithm works points only in the first octant and that is why the choice is between east and north east.

The first octant means that typically we are moving from left bottom of the screen to the right top and the slope of the line is less than 1. And towards the end of the last lecture we had also seen how to take care of the other octants. That means the slope is more than 1 or it is negative or the starting order of the point is different that means if you have to move from the starting point on the top right to the bottom left on the screen and how to draw the line. So we have to swap the points and change the increment from positive y plus 1 to y minus 1 and things like that. So I think you worked out based on the steps I have given to swap the roles from other octants and bring it to the first octant and then apply the same algorithm.

So you need to apply an overhead for the other seven octants and then when the line is simplified to a situation of drawing a line on the first octant only we apply the Bresenham's sequence of steps; the iterative loop from the left bottom to the right top of the screen and the slope less than 1. Towards the very end of the lecture from where I will basically continue today, we had also discussed the situation in brief as to what happens when you try to draw a line on digitized space and what are the problems. And the main problem which came forward with examples which we had shown in the last class and we are going to show today again is the effect of stair casing or aliasing or jaggies we do not have a very smooth line and you typically have a stair casing the effect of the line drawn.

So we start with that point from there onwards or from where we left in the last class and the main three issues which we see here along with that we will see other issues as well today is the effect of stair casing on the left as you see we have taken the same example of a line drawn from a point 5, 8 to 9, 11.

(Refer Slide Time: 3:35)



As in the last class we worked out this example and we came across these three points on the screen and this blue line on the top left basically shows the line which is actually they analog line I would say or the straight line. But the point which where selected by the algorithm or even if you use the round function you get the same sequence of points and you see the effect of stair casing or aliasing as it is called. That is one of the problems we have.

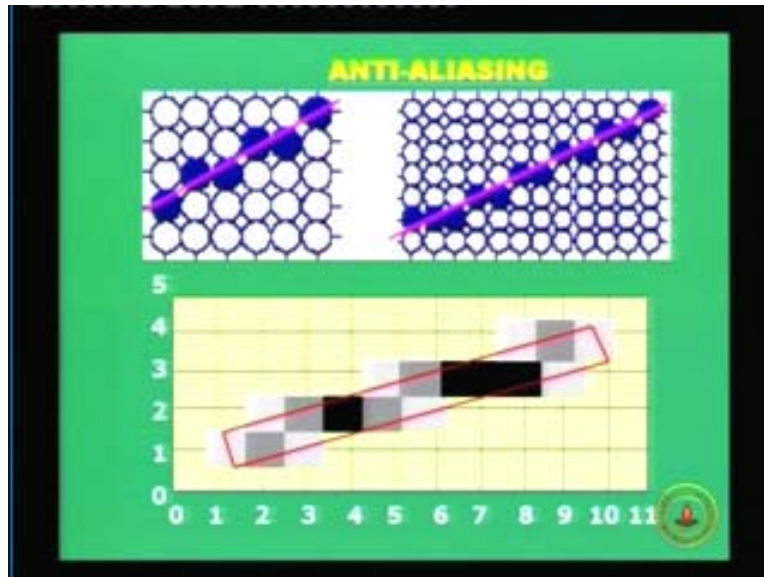
We will not discuss the solution in this section today but will just talk about methods of how to reduce this stair casing effect and the special effects of line drawing which will try give a smooth nature. There are problems when we try to draw flat lines because this line typically as you see here are the blue dotted lines from bottom right to top left is one pixel width but if you need to draw lines which are more than one may be two or three pixels wide and you may get a scenario which is like this which cannot be ideal. That is the problem with the fat lines or thicker lines as they are called.

Then of course you have the problem of n point ordering of the line where these three lines are drawn with three different colors, they are very wide and thick, one is the red line and another is the blue line here and then you have the yellow line. And in such a situation the problem which may come is that what you do with the end point and how do you draw the end point because if you draw the red color line and then top the blue color line and then of course the point comes is the common end junction.

There are two junctions of the lines in this case whether you would like to put a color red or blue in this case or in this case blue or yellow or a combination of that. So that is what you have to decide to maintain the continuity, to maintain the continuity so that the lines look very ideal. These are the three problems which is associated with line and we will take these solutions to these cases specifically in the case of aliasing or the stair casing,

try to see how to solve that. But to give an idea on how to reduce the effect of aliasing, we look in to the next slide where this is the another example of the snap shot taken from one picture where you see two different lines and the effect of stair casing in each.

(Refer Slide Time: 7:03)



As you see here the blue dots are the once which have been chosen by the Bresenham's mid point line algorithm here and this is zoomed in basically to larger dots which have larger resolution of the right hand side, you see a fair set of points selected here and typically as you see almost there are no points which strictly lies on the line. At the maximum you may find one but not strictly. So you will have this effect of stair casing and the question is when you try to see this line in a high resolution screen and zoom into it you will typically see this effect and it may appear very bad to the user. So it may not appear neat and nice in uniform and the thickness may also weary depending upon the spacing of the respective pixels on the screen as you progress along the line.

One way, we will not discuss an algorithm here to reduce the effect of aliasing of what is called the anti-aliasing method. In fact the lot of concepts based and even four year transformed in the spectral domain which can talk about anti-aliasing in signal processing theory. But we will keep ourselves to the computer graphics scenario in digitized space and try to see an example of how to reduce aliasing or reducing the stair casing effect.

One way to do that is to vary the intensity of pixels around the points which have been selected by Bresenham's. Let us look at an example where we see a line is drawn from a pixel 1 1 to a pixel 10 4 and the slope is less than 1 the line is in the first quadrant.

As you can see we are attempting to draw a line which is more than one pixel wide or even it is a one pixel wide what we try to do is that we take a few adjacent pixel with respect to each pixel and then shade those pixels with a lesser color, lesser intensity and

in this case of course we are drawing a dark pixel on a white back ground but it could be the reverse. That means you may have a dark background and draw a white pixel. So whatever the case may be you are talking about the contrast white and black or black and white and in this case we are talking about black and white background so the deepest intensity will be on those pixels with exactly lie on the center of the line.

And as you see that the pixels which are separated out towards the border of the line, they will start having lesser and lesser intensity and in this case of course they will start to become white or other than being dark. And now the question comes is, of course which pixels do you select? That is one question which is ought to be thought of, which we will discuss in special effects towards the end of this course. But suppose the pixel is selected **they able to excess** which have to be shaded with lesser amount of intensity than the one which is actually on the center of the line, the simple method can be used to find out what should be the intensity of the border pixels which are adjacent pixels which are not falling exactly on the center of the line.

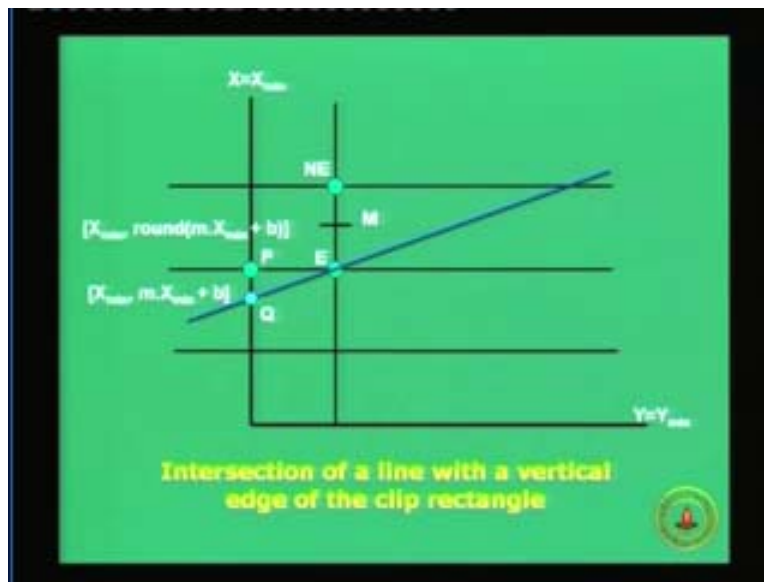
The simple way of trying to find out is the following: actually to visualize that the pixel has a finite square unit area as you see that each pixel is put is shaded by a squared unit area completely dark or grey or lighter ones and depending upon the amount of area coverage. The amount of the area of the pixel squared unit block which is inside the line which is about one or two pixel provide in this case assuming that this red rectangular strip basically shows that the true line which we are trying to draw and we actually just calculate how much of this square area unit of this pixel is lying within this rectangular strip and based on the amount of area which is inside that is the factor which is used to shade the line. So it is simple.

If this is the line which is fairly thick and then there are pixels which are lying within the line which have been selected by Bresenham's and there are a few adjacent pixels and we want to shade those adjacent pixels which are on the border of that line, more pixels are taken with lesser intensity compared to the center one. So the center one which is within that line strip will have maximum area coverage in terms of unit area which is within the line strip that will have the maximum area and that will have the maximum intensity, black or white whatever the case may be. Whereas adjacent pixels will have less coverage in fact that means that the part of the rectangular strip will be overlapping with the square area which is adjacent and we just calculate the amount of square area overlap between the rectangular strip and unit pixel. The larger the area overlaps more is the intensity, the lesser the area overlaps, lesser is the intensity.

So you look back based on this concept as you can see that the pixels which are almost or exactly on the center are going to have darker or deeper intensity or black in this case. And those which are in the border will have lesser amount of overlap and that is what is going to cause them to give lesser intensity. So the area overlap is one case, of course this area overlap function typically when they take in terms of, you can use different types of models in terms of guarsion or rectangular parallelepiped.

We will talk about this in special effects to see how different types of effects of area overlap or even volume overlap can create different types of uniform ness in a line. So this is one of the most standard techniques used to avoid aliasing and we will probably talk about this more in terms of what we call as special effects in terms of additional topics which will be covered at the end. Since today's talk and the last two where on scan line drawing, we will go back to algorithm and of scan line drawings and curves in this course of the lecture. There are few more points which have to be discussed and the next point talks about intersection of a line with a vertical edge of the clip rectangle.

(Refer Slide Time: 11:21)

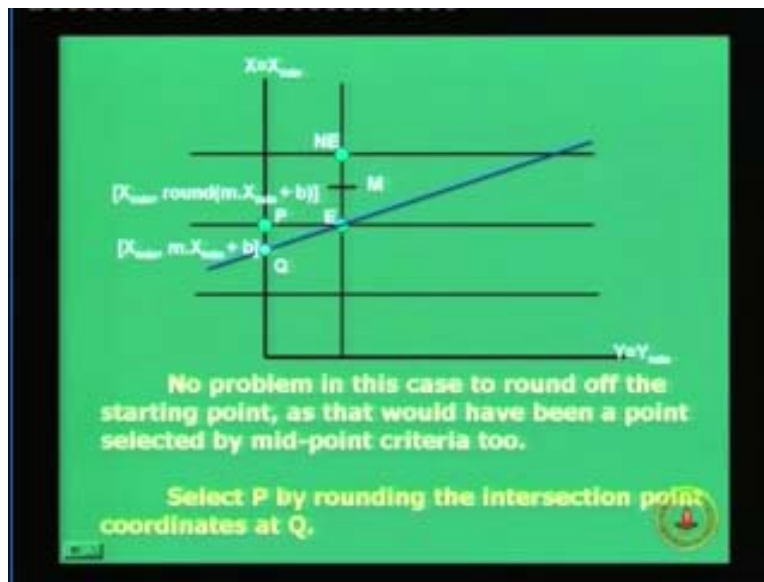


Again I should repeat here that we have not covered the concept of clipping algorithm but let us say in this case I am trying to illustrate a scenario where the blue line is the actual line and then there is a rectangular strip, a border we have seen the left and the bottom part of the rectangular area or coverage which will be clipping that line so we do not need a part of the line which is to the left of the point Q. What is the point Q? Q is the intersection of this blue line with the vertical line  $x$  equal to  $x$  min as you can see here. And we need to clip the line at this point Q and do not show pixels to the left of Q you only need pixels to the right of Q because the line portion of the line which is left hand side of the point Q is outside the clipped area.

This is the clipped area let us say, we are looking into the left edge of the clip rectangle and the bottom edge which is  $y$  equal to  $y$  min of the clip rectangle. And without applying the clipping algorithm for this line and we are trying to visualize what could be the problem if I am trying to clip this line with  $x$  equal to  $x$  min that the left edge is what is bothering us. What will happen is we not only want points to the left of the point Q the pixels to glow but we want the point to start from the pixel p.

So the first point will be the pixel p which we want and the next point will be E and so on. Now what will happen is, instead of Q you can automatically select p because that is the nearest round function. So when actually do a scan line algorithm the point p which is provided in the north east pixel with respect to the previous one will be selected by Bresenham's. So that the point says here that it is no problem in this case to round of the starting point, we are talking of the starting point Q is it? That is not a starting point in the diceretized environment because that is the fractional number M multiplied by x min plus P is the y coordinate of the point p and x equal to x min is the x coordinate of point Q.

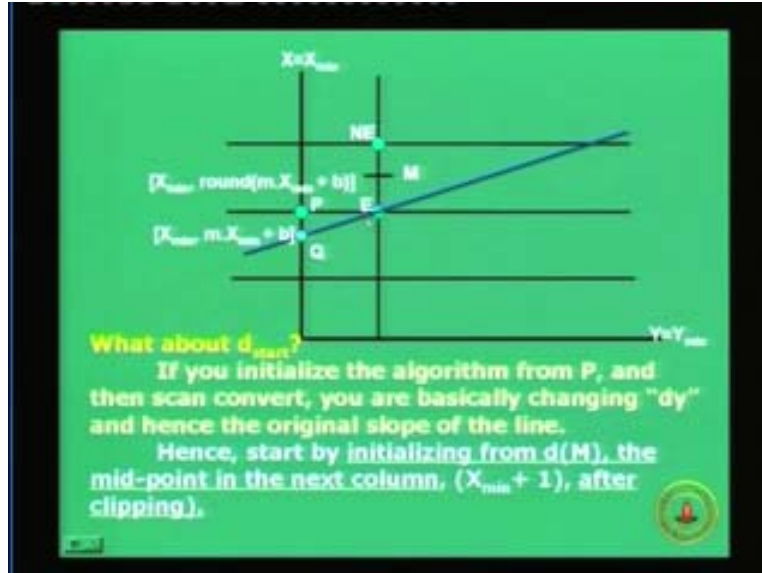
(Refer Slide Time: 13:49)



Few are bothered about y coordinate of point Q which is the point of course. P is the nearest to Q so in terms of Bresenham's mid point criteria the point will lie over the mid point below Q and P will be selected. So let us read the statement, it says that it is not a problem in this case to round off the starting point as that would have been a point selected by the mid point criteria too. So P is selected that is not a problem but the next problem which comes is of course we select P, how to select P? It is very simple, select P by rounding the intersection point coordinates at Q. So evaluate the y coordinate set point Q as given by main terms x min plus b and round that function to give you p so that is straight forward thing. But what is the other problem? What about d start, you remember we had a decision variable d and it needed an initialized condition or a starting value which we call as d start and the d start will go wrong if you initialize the algorithm from P.



(Refer Slide Time: 15:28)



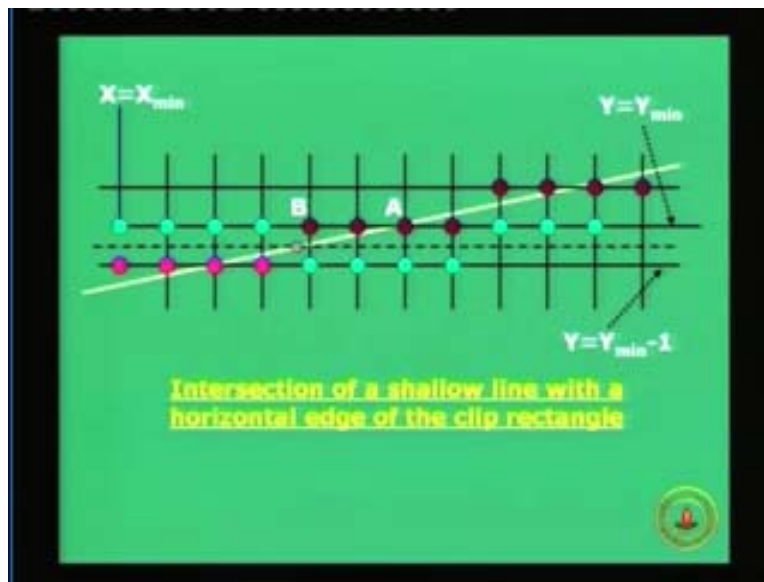
If you initialize the algorithm from P and then scan convert, scan convert means apply the Bresenham's mid point criteria iteratively apply the mid point criteria, look at the next east and north east and so on. But if you start from p instead of Q somehow which is basically going to happen because Q is the floating point, P is an integer number of both x and y coordinates. So if you start from P you are basically changing the initial variable dy, what was dy? Remember  $y_2$  minus  $y_1$  and hence you are adjusting the original slope of the line. So this is a dangerous ploy to make to change the d start from P because we change the line itself.

Instead of the blue line passing through P which was original line, the line will tend to appear to start from P, it may have a slightly different slope and that is the main problem but that could cause some other pixels being selected as you apply the Bresenham's criteria from P onwards. So what you do after P of course P is selected by rounding off but what about the d start. The concept is very simple, start by initializing from d of M so d of d is a variable which is evaluated at the mid point. So do not start from d of p because that will change, start by initializing from d of M that is the mid point criteria in the next column  $x_{min} + 1$  after clipping. So you do a clipping, round off Q to p and then start by initializing d from the mid point criteria.

Start to evaluate it from here at  $x_{min} + 1$  so that will help you to actually get rid of these points here from the key. Actually apply the d of M by using the original values of dx and dy and what will happen in this example as shown in the figure is if you see that the mid point is lying above the line E and E will be the east pixel, it will be the next pixel which we selected and so on. So once the d is initialized correctly at  $x_{min} + 1$  which is the next line passing through the north east and east you do not have a problem of a wrong initialization of d start. So you must be very careful with clippings.

We had seen two problems now, aliasing and we know if not algorithmically but the method to overcome aliasing a little bit. And we have seen how clipping could create problems in Bresenham's and we have to be careful of starting with the  $d$  value in case you are clipping a line. That means you have a line and you are clipping it and you want to start at one point and then go ahead and do not bother about the rest of the line. So the clipping does create a problem and you have to be careful when you start the initial variable  $d$ . Let us look at another example or another problem of drawing a line what we will call as a very shallow line or a line with a very small slope.

(Refer Slide Time: 16:56)



The value of  $M$  is in between 0 and 1 but it is very small approaching 0. It is not equal to 0 but it is a very small value, it could be a fractional of course a fractional number and a very small one. And of course it is absolutely no problem and no restriction of applying Bresenham's algorithm for any value of  $m$  as long as it is 0 and 1. You will get the correct values of the integer pixels along the part from the left hand starting point to the top right as long as the value of  $m$  is between 0 and 1. Even if the value is close to 1 or close to 0 there is absolutely not going to be any problem. But if the line is going to be shallow you do have problems with clipping. In addition to the clipping problem discussed earlier you do have problems here. What is the problem?

Let us look into the slide once again you are talking of intersection of a shallow line with horizontal edge of the clip rectangle. So the clip rectangle is, again the left edge of clip rectangle  $x$  equal to  $x_{min}$  the left edge and the horizontal edge is the  $y$  equal to  $y_{min}$  line.  $y$  is equal to  $y_{min}$  line as marked here, this line in the middle, there are three lines which are shown  $y$  equal to  $y_{min}$ ,  $y$  equal to  $y_{min} + 1$ ,  $y$  equal to  $y_{min} - 1$ . So,  $y$  equal to  $y_{min}$  is the horizontal clip edge of the line and the left edge is  $x$  equal to  $x_{min}$ . So now what we are interested to do is to clip the line and show points of the line which are inside the clip rectangle.



And if you carefully see here, neglect the color for the timing and see the dots and as you can see here that points which are on and within the clip rectangle are marked by this brownish color here. There are four pixels on  $y$  equal to  $y_{\min}$  and four pixels on  $y$  equal to  $y_{\min} + 1$ . Now the pixels which will be selected by the straight line by Bresenham's algorithm without clipping are these four pixels marked in a purple color at  $y$  equal to  $y_{\min} - 1$  that is the bottom horizontal line which is below the horizontal clip line of the horizontal edge of the clip rectangle. So these four pixels would have been selected, then would have selected these four pixels and out of the middle four pixels two are marked as the A and B. We will see why they are significantly marked and those four pixels lie on  $y$  equal to  $y_{\min}$  that is the horizontal edge and  $y$  equal to  $y_{\min} + 1$  will also have four pixels and so on.

So if you are interested in clipping now the problem which will first come is you will be tempted to find the intersection point of the straight line and  $y$  equal to  $y_{\min}$  line and that will give you the point A because the point A is the closest round of integer value as you can see in the figure of the straight line and the horizontal clip line. So you will be tempted to start from A and close sort of the all the other pixels.

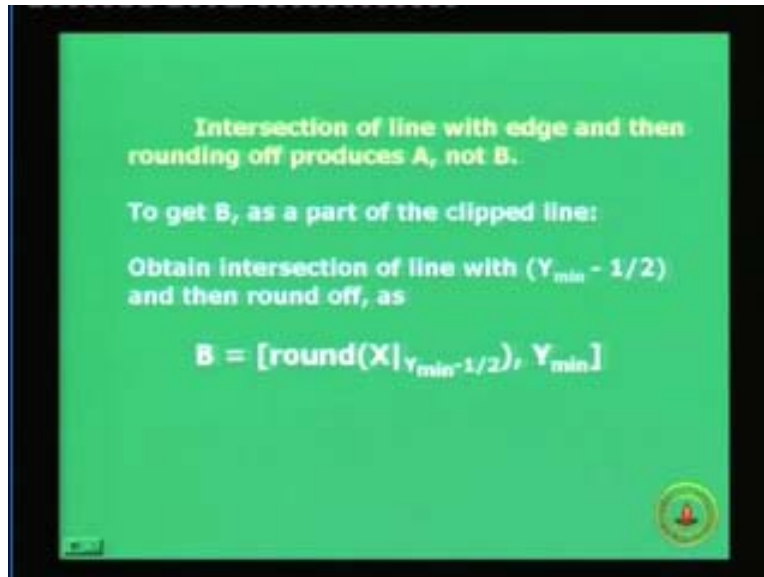
Start from A, have these two pixels and then go on get the four pixels and so on. There are two problems which could occur here, there are two problems, first of all changing the slope of the line and then at the edge what might happen is the line will end abruptly because I would like to have the two pixels left of A along with B which should also be a part of the line because those would have been selected also by Bresenham's and they fall on and within the clip rectangle. Remember I keep repeating on and within the clip rectangle, so all points lifted by Bresenham's which lie also on the horizontal edge must be selected. So the question is how do you select these two pixel to the left of A one of them which is B in fact the B is marked because that is the starting point and it should be the starting point of the line on the bottom horizontal edge of the clip rectangle whereas A is an intersection point.

Automatically A selection is obvious and you might faultily start with A. But ideally, to give the shallow effect of the line you should start with B. So this is the problem of intersection of a shallow line with the horizontal edge of the clip rectangle. And we will try to see how we can avoid these that means we start from B rather than starting from A. The solution is in the next slide, I hope you drawn the figure by now, otherwise please draw it, there are three lines  $y$  equals  $y_{\min}$ ,  $y$  equal to  $y_{\min} + 1$  and  $y$  equal to  $y_{\min} - 1$ .

Do not worry about the left edge which is  $x$  equal to  $x_{\min}$ , we are bothered about the horizontal edge, there are four pixels marked here and  $y$  equal to  $y_{\min} - 1$ , four pixels marked by Bresenham's on  $y$  equal to  $y_{\min}$  itself and that is the horizontal edge of interest and four pixels are also marked and shown here at  $y$  equal to  $y_{\min} + 1$ . And the problem is again I repeat how to start from b rather than starting from A because starting from A is a very obvious choice but not the correct choice, I should repeat it again, obvious choice because that is the intersection point of the line A and the horizontal line  $y$  equal to  $y_{\min}$ . But I do not want to start with A because you will be

throwing of too legitimate and legal points or correct points obtained by Bresenham's on the horizontal clip edge which are to the left of A to be precise and these two points must also be selected along with the other. So all the brown points that is brown colored points on  $y$  equal to  $y_{min} + 1$  and also on the horizontal edge  $y$  equals  $y_{min}$ , all these eight points must be selected, not only the six points starting from A to the right hand side. So in intersection we had the problem of intersection of a line with edge and then rounding off produces A.

(Refer Slide Time: 22:55)



Therefore, if you take the intersection of the line with the horizontal edge and round off the coordinates you are basically starting from A and not B that is the problem and to get rid of this problem, that means to get B and the pixel between A and B as well as a part of the clip line, basically I want to start with the B. And what is basically done is you need to obtain the intersection of line with  $y_{min} - 1/2$  and then round off and then get the coordinates of B. Let us see how it works.

Remember what we have to do, the suggestion says obtain intersection of the line with the horizontal line  $y_{min} - 1/2$ , that  $y_{min} - 1/2$  line should lie between its horizontal line between  $y_{min}$  and  $y_{min} - 1$  and then you need to have an intersection of the line with this  $y_{min} - 1/2$  and then round off. So this is the formula to obtain the  $x$  intersection for  $y_{min}$ . And let us go back to this figure and see what this basically means here. So, intersection of the line with  $y$  equals  $y_{min}$  was giving us A which is correct up to that point but we also need to get the two points to the left so you see here the dotted line, the dashed line between  $y$  equals  $y_{min}$  and  $y$  equal to  $y_{min} - 1$ , this line is  $y_{min} - 1/2$  that is  $y$  equal to  $y_{min} - 1/2$ ,  $y$  coordinates is going up vertically and the dashed line in between is the  $y$  equals  $y_{min} - 1/2$ . If that is the horizontal line which is  $y$  equals  $y_{min} - 1/2$ , intersection of that line with the line being drawn on the screen or the scan line as it is called will give you this intersection

point but the coordinates of that intersection point are going to be floating point numbers. So what you need to do is calculate this in floating point but then round off. You need at least to start with one round function in both these cases. In the previous problem as well and in this current problem you need to evaluate these coordinates and then round off to get the coordinates of B which is going to be your starting point. So B is going to be your starting point.

I repeat the equation again, please note it down, you can get B by rounding of  $x$  at  $y$  min minus  $1$  by  $2$  but not of  $x$  at  $y$  min. If we evaluate  $x$  at  $y$  min from the equation you will start from the point A and not B. But if you use this formula, as you have seen in the figure you will start with the correct point A and also the point pixel A and B as many of them will be selected and this is the starting point of B. So this is the problem and of course the  $d$  start, the initial value of  $d$  must also be followed in a similar manner as I described in the previous problem. You should not change the starting point so that the value of the slope of the line and the  $dy$  both of them change that is not correct.

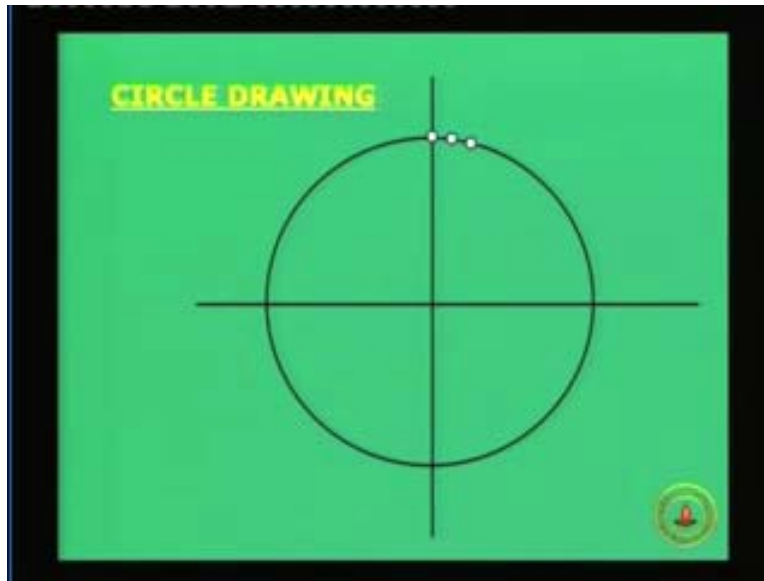
You have to start with the formula which has been given earlier for  $d$  start to basically ensure that you evaluate the  $d$  of the next mid point and start evaluating that is the key idea so do that for  $d$  start. But in addition to  $d$  start you have a problem in some cases specifically for what is called the shallow line, the line with a very small and negligible slope which is almost horizontal but not exactly  $M$  is equal to  $0$  case where it is purely horizontal then of course we do not have a problem but  $M$  is slightly more than  $0$ , it is a small positive value let us say, then you have a problem when you clip that line. If you have no clipping of course there is no problem and Bresenham's will work with any value of  $M$ , small value of  $B$  because we are working in the integer space is absolutely no problem here. But when you clip you have problem with two cases; one is basically the problem that where you start because some points may unintentionally fall outside where they are supposed to be lying on the edge, you start neglecting that if you do not do a correct round function and then the initial value of  $d$  start.

We know how to do the  $d$  start. That means evaluate the value of  $d$  or start the initial value of the  $d$  at the next mid point, that is how you do with the  $d$  and typically we have also seen when you need to start correctly you evaluate by rounding off the value at the previous value of  $y$  min. These are the two special cases which one must take care in terms of clipping for a straight line. And we wind up the scan line algorithm for straight line or line drawing right now. I mean we have discussed the DDA algorithm earlier and then of course the Bresenham's criteria, Bresenham's algorithms steps which is based on integer arithmetic, we have taken an example to illustrate how it works and then of course we discussed the problems of aliasing, jaggies, flat lines, the end point effects and of course the problems of clipping where we discussed the starting value and the clipping problem where some points unintentionally outside the range.

These were in detail, the discussion we had today on scan on algorithm and we move towards the next scan algorithms and curves and we start with the simplest curve which we know is the circle. Of course we will also discuss ellipses but since circle is a special case of an ellipse, if you are able to draw an ellipse then you should be able to draw a

circle. But we will see later on why we are discussing circle specifically, separately again for the first time and then move on to ellipses and whence you have the algorithms for scan line like we have drawn for a straight line. If you have algorithms to draw a circle or an ellipse you should be able to draw any arc. So these are the three types of scan on algorithms, first we discussed line, we have discussed the special effects and we have discussed the problems of intersection, lines with clip edges and all that. Now we move on to circle drawing which is the next stage. We move over to circle drawing and that is how you draw a circle on a screen and we will see an example.

(Refer Slide Time: 28:14)



If you look into the picture you see that you have to have the points on the circumference of the circle and in this case we assume that you have to start with the top most point of the circle. That is the intersection of the circle with the vertical line which is passing through the center of the circle. So we start with that point move to the next and so on. So unlike this scan line for line drawing where we were drawing lines between slope 0 and 1 and that means points where in the first octant.

We remember the octant eight of those which divide the entire space into eight equal parts in 2D space. We can also visualize the octants here in the case of the circle. And if you look at the figure again you should be able to tell me in which octants the points are lying? They are not in the first octant any more, they are lying in the second octant and so we will draw points only in the second octant. And points in the second octant, we will start from the top most point here and keep going, I have drawn three points only and stop at a particular point and the last point in the second octant will have a special property. You can guess easily but I am telling you right now that the coordinates of the last point in the second octant which lies on the circumference of the circle will have its x coordinates equal to y coordinates.

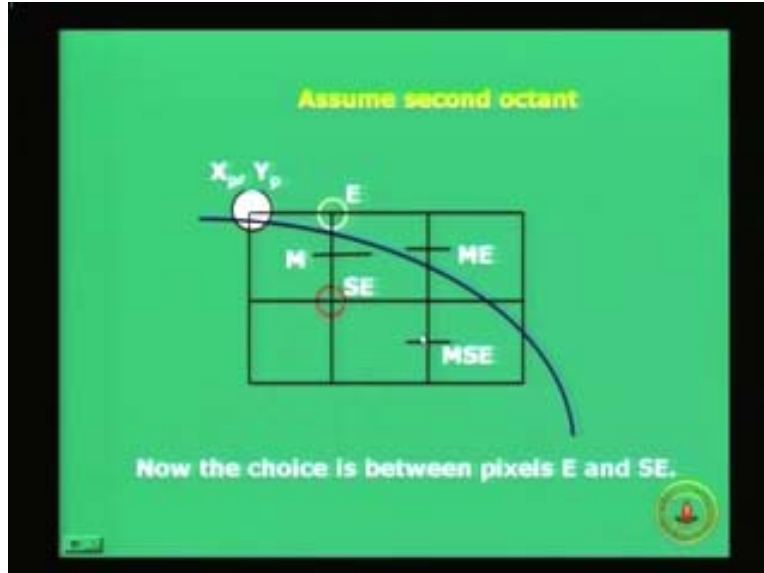
What are the coordinates of the first point? Assuming the center of the circle is lying in the origin, do not worry if it is not lying in the origin. Let us start with assumption in the simplest case where it is lying in the origin. If the origin is 0 0 and the radius is equal to r so the coordinates of the first point will be 0, r so 0, r will be the coordinates of the first point. x coordinate will be 0 and y coordinate will be r. And as you keep moving along the circumference of the circle from the first point to the second point to the third point and so on, what will happen is the x coordinate will start increasing and y coordinate can only decrease.

The y coordinate starts with r and decrease x coordinate starts with 0 and increase and the last point in the second octant will have the scenario where the x coordinate will be equal to y coordinate and that point will be at exactly 45 degree orientation respectively the center of the circle. Now the center of the circle is not in the origin, this is absolutely not a problem, we have already seen lots of transformations in 2D and 3D and we know how to manipulate in figures on the screen and so if it is not on the origin it is absolutely not a problem, actually we shift the origin to the center of the circle or shift the center of circle to the origin, draw the circle and then shift it back.

We know all these manipulations when we studied about transformation. So we apply two constraints here to draw a circle, first of all the origin is the center of the circle and we start with the top most point with coordinates 0, r and move to right along the circumference so we are moving in the second octant and once you have covered the points on the second octant I believe you should be able to visualize already how the other octants symmetrically, how the other points on the other seven octants can easily be obtained.

Of course I will show the algorithm, how to do that as you have done for a scan line when we drew points only in the first octant but actually got the points in the other seven octants also. So we did at swap if you remember, swap and interchange a type of operations, increment and decrement etc. Here in this case if you pick up the points on the circumference which lies on the second octant only, the other octants can also be symmetrically generated because a circle is **ary** symmetric not only in four parts but also in eight parts. Let us see what is happening in the second octant. We are also going to blindly apply the Bresenham's mid point criteria for a case of a circle. So it means the following; if you are working on the second octant and  $X_p$   $Y_p$  is the point of the current iteration which has been selected at the previous stage of iteration let us say, if that is the point which has been selected the next point can only be one out of east or south east depending upon where the line passes.

(Refer Slide Time: 34:35)



If it is to the top of the point M or M is below the circumference of this line, in this case it is a curve but I will call it as a curve, so do not confuse, it is not a straight line but it is a line let us say or an arc but I will use the word arc or line interchangeably. So this line or arc if it is on top of M you select E or if the M is on top of the line passes below M then you select south east, so that is a very simple criteria like Bresenham's case of a line, a value at the mid point with respect to the arc or the line.

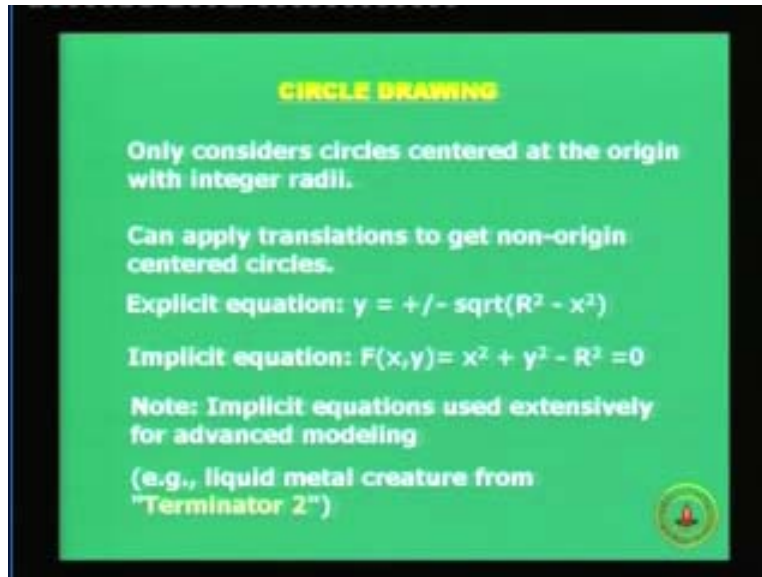
Find out if M is below or above the line and that will tell you whether the line is close to the east or south east. In the case of a line for the first octant the choice was east and north east for you and in this case when the circumference is coming down in the second octant obviously the choice after the current iteration is going to be either east or south east because the curve is going to fall in the second octant. So choice is here east and south east and in the case of a line we had east and north east, here it is falling in the second octant there it was going up in the first octant so we had the east and north east after the current choice. In this case of a second octant after the current choice with the curve is going down along the circumference so you have choose between east and south east for the line. This is what is going to happen in the second octant.

And if we look back into the figure if you have chosen east in the current iteration after you have evaluated the  $X_p, Y_p$  and then you have moved over to select east in this case as the line is above M or M is below the line, then if you have selected E the next mid point criteria where you have to evaluate is this point M of E and mid point after E or if you have selected south east if the line was below M then you would have selected south east and next mid point evaluation in the next iteration could have been mid point of south east or MSE which is this point as marked here. So we know that the choice basically at any stage of iteration is between east and south east and we are working on the second octant. So please you remember that now the choice is between east and south east pixels



at any stage of iteration. So a few points to be considered on circle drawing, we go through few points **tournely** considers circles centered at the origin with integer variables. Let us assume that we have integer radius and the center is at the origin.

(Refer Slide Time: 37:09)



We can apply translations to get non-origin centered circles. We also discussed this that means if the center of the circle is not at the origin you either translate the origin or translate the circle such that the center is shifted to the origin or the origin is shifted to the center of the circle and then every thing falls in place. Well, now we have to look into the equation for a circle.

We look into the slide; an explicit equation as it is called for a circle that means given a particular value of  $x$  if you want to obtain a value of  $y$ . Remember unlike in the case of a line you can have two values of  $y$ . For a given value of  $x$ ,  $R$  is the radius of the circle, is the plus or minus square root of  $R^2 - x^2$ .  $R^2 - x^2$  is the value of  $y$ . Of course in one case it will be two different points of  $y$  or two different octants, that is all.

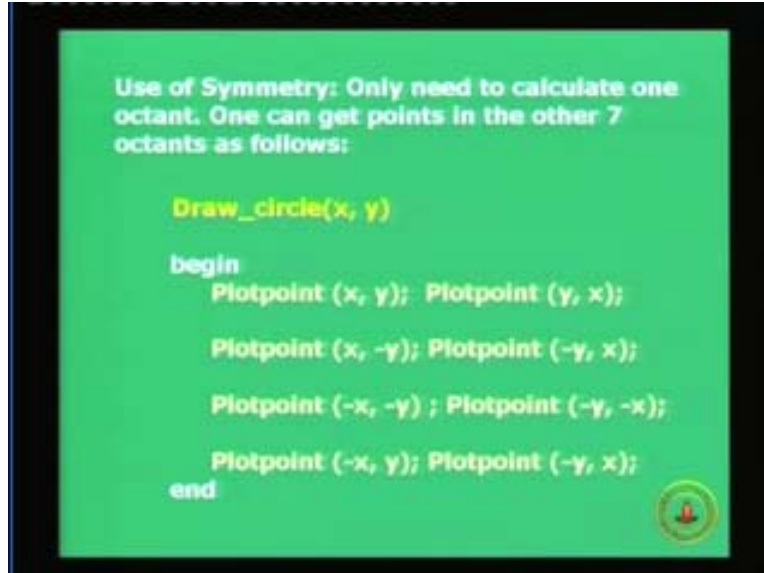
But since it is the mid point criteria we are going to use the functional form  $F(x, y)$  is equal to 0 which is called the implicit equation for a line. The implicit equation of a line in the case of  $F(x, y)$  is equal to 0 for the case of a line also we had seen and now in the case of a circle the implicit equation becomes  $x^2 + y^2 - R^2 = 0$ . The equation of a line is  $x^2 + y^2 = R^2$  like  $mx + y = b$ . So shift everything to the left hand side and make  $y - mx - b = 0$  in the case of a line. In the case of the circle we make  $x^2 + y^2 - R^2 = 0$  so that is the implicit equation of the line.

And implicit equations are used extensively in advanced modeling not only in circle drawing in fact not only for curves but for surfaces in 3D planes and surfaces in 3D and whatever is in the line or a circle or a curve in 2D is basically a non-planar or a planar surface in 3D implicit equations are used extensively for advance modeling in computer graphics. We will come across these equations later on and we also discussed about solid modeling and advanced concepts of modeling. A typical example for using implicit equations for a surface model is a liquid metal creature from the movie Terminator 2.

I hope you have seen it, I did say entertainment games and movies is one very good application or one of the most widely used applications of computer graphics and those who have seen the picture Terminator 2 Judgment Day would have see this liquid metal creature which melts itself into through various part, its physical bodies is done with the help of, this an example of implicit equation. Coming back to circle drawing which you are discussing, if you look into the point here as mentioned in the slide we use the symmetry, what is the symmetry of the circle we are talking about? One only needs to calculate one octant and in this case we have already decided to start with the second octant. Start from the point 0, R and calculate in the second octant and after completing the second octant one can get points in the other seven octants as follows.

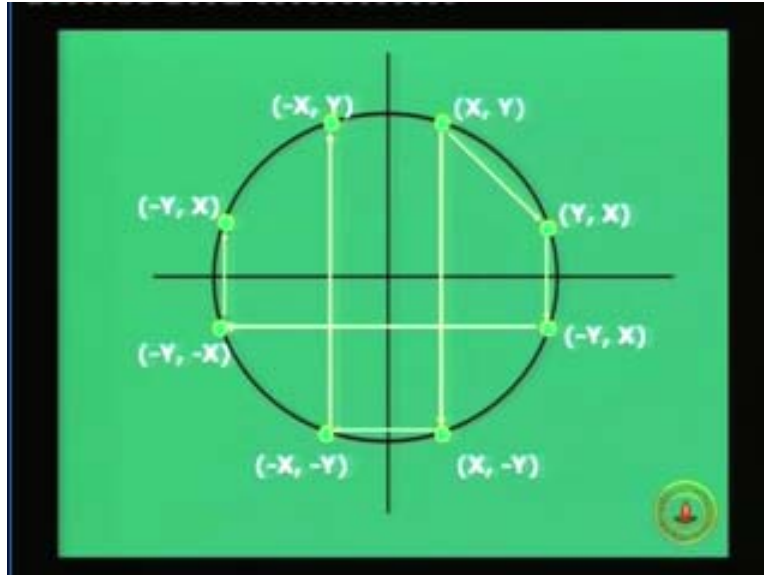
As we did for a line we computed from the first octant when slope was less than 1 and for the case of a line if the line is not in the first octant we knew what we had to do, to shift the line in this case to the first octant. In the case of a circle it is not that. In fact if you want to draw a complete circle, points on the circumference of the circle will be spread over all the eight octants in the 2D space and since there is a lot of symmetry in the way the points in the circles are distributed if you can get 1 by 8<sup>th</sup> of the circumference that means points on the second octant only by symmetry you can obtain the points in the other seven octants. And that is what is going to be discussed here.

(Refer Slide Time: 39:06)



As you can see we have put a small piece of pseudocode for what we call as a draw circle at  $x, y$ . This is not a line circle drawing algorithm but it says that I am using symmetry if I have already obtained a point  $x, y$  using Bresenham's criteria for the second octant then I can get points in the other octant using this set of plot point. Assuming you have a graphical specification for plotting a point at  $x, y$  and the  $x, y$  coordinates of the point on the circumference of the circle along the circle is obtained at the current stage of iteration by the Bresenham's algorithm. All the other seven points can be easily obtained by symmetry. You can see the symmetric nature of the plot point commands where we are either swapping the positions of  $x$  and  $y$  or changing the sign of  $x$  and  $y$  or both in and generating all the other seven different cases for drawing lines.

(Refer Slide Time: 40:07)



This next figure will show you what these basically mean in terms of getting the point. The point  $X, Y$  the first point is shown in a different color then the other seven points  $X, Y$  is obtained say at a current point of iteration in the second octant and after this is obtained you can obtain all the other one two three four five six seven points by symmetry in the other seven octants, from  $XY$  you can go to first  $Y, X$  then change the sign of  $Y$  to get minus  $Y, X$  then change the sign of  $X$  to get minus  $Y, X$  and then finally again change the sign of  $X$  to get minus  $Y, X$ . From  $X, Y$  you can only change of sign of  $Y$  to get  $X$  minus  $Y$  then you change the sign of  $X$  to get minus  $X, Y$  and then finally you change the sign of  $Y$  again to get minus  $X, Y$ .

So the paths are shown to illustrate the sequence but you can pick up any sequence in fact you can absolutely use the plot point commands from  $Xy$  to  $YX$  to minus  $Y, X$  and so on. But this shows the path where the symmetricity is maintained and it shows how symmetric the points are with respect to the horizontal and vertical axis of this four way symmetry as shown and eight way symmetry is also visible here. That means all the other points in the seven octants are obtained after you obtain the point in the first octant only.  $X, Y$  is in the first octant, all the seven point are obtained and I repeat it again, this is a court to obtain the points only in the seven octants, not for the Bresenham's. Bresenham's gives you  $X, Y$  and I repeat again and this is a court for an overhead draw circle routine which will basically plot all the eight points in the screen corresponding to points on the eight octants obtained from the second octant point  $X, Y$ .

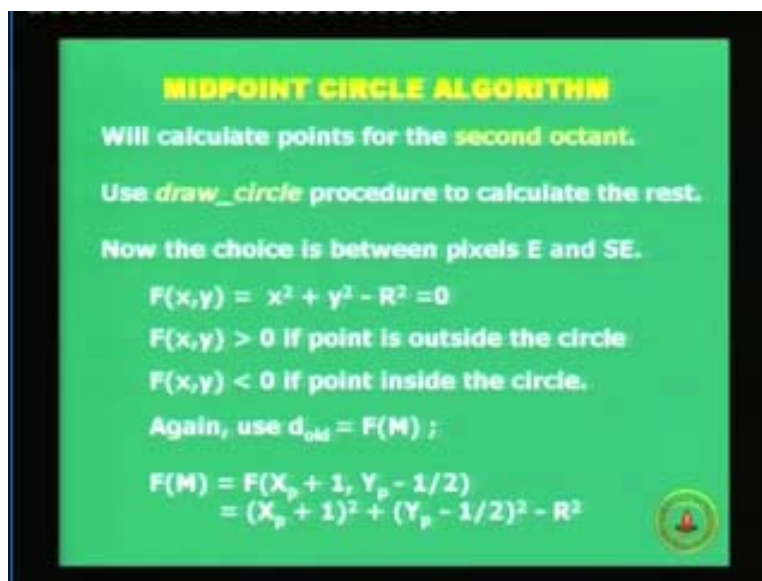
Second octant point is the  $X, Y$  and all the other points in all the other seven octants are obtained using the symmetrical nature. So that is very easy and straight forward and in fact it is much easy to visualize the scenario and also the code as given here in the previous slide that you can see that the flip nature of the  $X$  and  $Y$  or the sign of the  $X$  and  $Y$  or the both are changed as we were talking about and the point here at symmetric is

much easier than the visualization of all the other seven octants in the case of a line, because in the other case of a line basically you had to do a little bit of manipulation juggling with XY coordinates or the positions of the line if the line has more slope more than 1 and the starting order is different, you have to do little bit of adjustment to bring it into this first octant then draw it and then generate the point which will originally generate the points back in the other quadrants.

In this case you just find the points always in the second octant and the other points are drawn using the algorithm as given here. Well we now slowly move actually into the mid point circle algorithm. Let us see how much time we have to get into the criteria if possible today and then continue in the next class. So we will calculate points in the second octant as given here and after a single point in the second octant all the points in the second octant are available, we use the draw circle code which we have given here which produces and calculates the rest of the points using symmetry which we have just seen now.

So points in the second octant have to be drawn from the equation of the circle. And we also know now that the choice is between east and south east pixel because the origin has been shifted to the center of the circle or the center of the circle has been shifted to the origin. So we are always starting from 0, R, keep choosing east and south east.

(Refer Slide Time: 46:23)



**MIDPOINT CIRCLE ALGORITHM**

Will calculate points for the **second octant**.

Use **draw\_circle** procedure to calculate the rest.

Now the choice is between pixels **E** and **SE**.

$$F(x,y) = x^2 + y^2 - R^2 = 0$$

$F(x,y) > 0$  if point is outside the circle  
 $F(x,y) < 0$  if point inside the circle.

Again, use  $d_{mid} = F(M)$  ;

$$F(M) = F(X_p + 1, Y_p - 1/2)$$
$$= (X_p + 1)^2 + (Y_p - 1/2)^2 - R^2$$

Let us look into the implicit equation of a circle it is given as. We have seen that earlier  $F(x, y)$  is equal to  $x$  square plus  $y$  square minus  $R$  square equal to 0. And the mid point criteria we also know from Bresenham's in the case of a line, we apply the same logic if the function value of  $F(x, y)$  is more than 0 the point is outside the circle, if the value of  $F(x, y)$  is less than 0 then the point is inside the circle. This is easy for you to visualize. If we take a circle and take points exactly on the circumference of the circle the value of the

function value  $F(x, y)$  will be equal to 0. So assume you have a circle and the circumference of the circle and when you choose points on the circumference of the circle and substitute into the implicit equation  $F(x, y) = 0$  which is given in this case where a circle is  $x^2 + y^2 - R^2 = 0$ .

So the point is exactly on the circle and the value of  $x^2 + y^2$  will be equal to  $R^2$  which is very natural and the value of  $F(x, y)$  will be equal to 0 that we know. Now if we choose a point within the inside of the circle not on the circumference but within the area enclosed by the circumference what will happen is, sum of  $x^2 + y^2$  will be less than  $R^2$  and hence the value of the function  $F(x, y)$ , you remember the equation as given in the slide,  $x^2 + y^2 - R^2 = 0$  is given, that is the implicit form of the equation of the circle and if I choose a point which is inside the circle the value of  $x^2 + y^2$  is going to be less than  $R^2$  and hence the value of the function will be negative.

However, if I choose a point outside the circle assuming this to be a circle, a small circle and if I choose a point outside the circle outside the circumference or the area of the circle then the value of  $x^2 + y^2$  will be more than  $R^2$  and the value of function will be more than 0. This is the concept similar to the case of a line, you also had a scenario of greater than or equal to 0 depending upon whether the point is above or below the line in the case of a straight line, here it is below or above an arc in fact we logically have to think that this is within the circle or outside the circle so that is the condition.

We look here, we have the functional value  $F(x, y)$  as given here and you remember  $F(x, y)$  will be positive if the point is outside the circle the value of  $F$  will be negative if the point is inside the circle, if the point is exactly on the circumference which we actually have to find out what are those pixels which are the closest on the circle then the value of  $F(x, y)$  is equal to 0. We start straight with the Bresenham's because we know, we already had ideas on what is the decision variable based on the functional value of the expression of the circle. So we straight away know the logic, at each point of iteration starting from the first point what we have to do? We have to evaluate in the next point, take a decision based on the sign, update, choose and again evaluate the next mid point and so on.

So look back, again in the case of mid point criteria for a circle we use  $d_{old}$  equal to  $F$  of  $M$  where  $M$  is the next point. Now what is the next mid point? It is the first mid point which comes. Well given any point  $X_p, Y_p$ , we will see that figure again, we have seen earlier  $X_p, Y_p$  is the current point, the next mid point is  $X_p + 1$  and  $Y_p - 1$  by 2. The first mid point, when the first point which has been selected is 0,  $R$  if you remember, that is the top most point on the circle and the second octant, the next mid point will be plus 1 or minus 1 by 2 is how the mid point will be progressing depending upon east or south east choice of course.

But again if we go back into the slide  $X_p, Y_p$  if it is a current point of interest then the mid point will be  $X_p + 1, Y_p - 1$  by 2 why? It is because the mid point is between east



and south east pixels unlike east and north east in the case of a line. You remember in the case of a line when we had east and north east I keep switching between line and circle here, please do not confuse because I hope I assume that the line is in your mind and you try to map that. In the case of a circle the equations are just different but the concept is exactly similar. In the case of a line when the slope was more than 0 but less than 1 the choice has between east and north east so the next mid point was plus 1 by 2 typically in the case in the choices between east and south east the next mid point will be plus 1 minus 1 by 2.

So  $X_p$   $Y_p$  is the current point of iteration, next mid point is  $X_p$  plus 1  $Y_p$  minus 1 by 2. If you substitute this coordinates in the expression of x square plus y square minus R square equal to 0 you have the d old value as F of M equals  $X_p$  plus 1 whole squared plus  $Y_p$  minus 1 by 2 square minus R square. You can break open these brackets if you want and expand and get the polynomial form of this expression in two variables  $X_p$  and  $Y_p$  but do not do that. Keep that expression of FM and let us look at the next choices. If d is positive or equal to 0, as you see in the slide the choice is d greater than or equal to 0 you choose south east, so your choice is between east and south east and the next mid point based on with the d which is greater than equal to 0 then the choice becomes south east. And the next mid point M new will have to be incremented by plus 1. If you look into the slide, now you will find the next point is incremented by 1 along x and minus 1 along y which gives the d new.

(Refer Slide Time: 51:46)

$d \geq 0$  choose SE ; next midpoint:  $M_{new}$ ;  
 Increment + 1 in X, -1 in y; which gives  $d_{new}$ .  
 $d < 0$  choose E ; next midpoint:  $M_{new}$ ;  
 Increment + 1 in X; which gives  $d_{new}$ .

$$\begin{aligned}
 (\Delta d)_{SE} &= d_{new} - d_{old} \\
 &= F(X_p + 2, Y_p - 3/2) - F(X_p + 1, Y_p - 1/2) \\
 &= 2X_p - 2Y_p + 5;
 \end{aligned}$$

$$\begin{aligned}
 (\Delta d)_E &= d_{new} - d_{old} \\
 &= F(X_p + 2, Y_p - 1/2) - F(X_p + 1, Y_p - 1/2) \\
 &= 2X_p + 3;
 \end{aligned}$$

$$\begin{aligned}
 d_{start} &= F(X_0 + 1, Y_0 - 1/2) = F(1, R - 1/2) \\
 &= 1 + (R - 1/2)^2 - R^2 = 1 + R^2 - R + 1/4 - R^2 \\
 &= 5/4 - R
 \end{aligned}$$

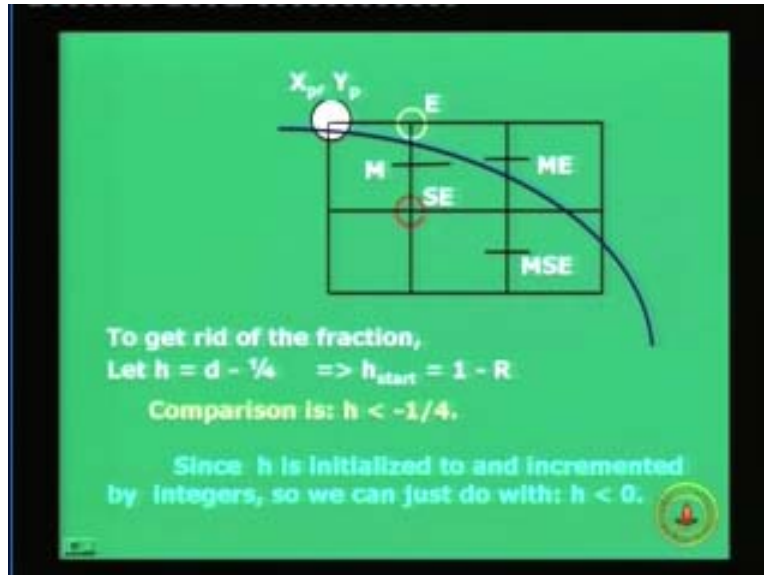
So the d new has to be the new value of the variable d which is evaluated at the next mid point M new. If you choose south east you basically have to increment by plus 1 along x and minus 1 along y to get the d new. What is the next d new? Well we know what is d new and we know what d old is. So what we do is we look at the first order derivative of delta d under the choice that we have chosen south east and that is d new minus d old

given by  $X_p$  plus 2 provided you have chosen south east. So it is plus 1 and then plus 1 again and  $Y_p$  minus 3 by 2 because we discussed about minus 1 by 2 earlier and then minus 1 3 by 2 minus the  $d$  old and the difference will be, if you subtract these two expressions of  $X_p$  plus 2 and  $Y_p$  minus 2 for  $F$  and  $X_p$  plus 1 that is the  $d$  old expression here. The  $d$  new expression here you will get the expression of  $2X_p$  minus  $2Y_p$  plus 5, so that is the increment in delta  $d$  under the condition that you have chosen south east.

Now let see what happens if you choose east. If you choose east under the condition that the value of  $d$  is less than 0 or negative you are choosing east because we will see in the figure because when the value of  $d$  is negative the mid point is basically below the line or the circle is passing over the mid point and closer to east and so you have chosen east so the next mid point  $M$  new must be incremented by 1 along  $x$  and nothing by  $y$  which gives  $d$  new. And the value of  $d$  new under the condition that we have choose in east is given by  $d$  new minus  $d$  old delta  $d$  east the increment provided you have chosen as  $t$  will be given as  $d$  new as  $d$  old is  $X_p$  plus  $2Y_p$  minus 1 by 2 instead of 3 by 2 because you remember in the case of south east you were also decrementing  $y$  by 1 but not in this case any more. Henceforth the difference in these two will be  $F$  of  $X_p$  plus  $2Y_p$  minus 1 by 2 minus the  $d$  old value which was originally computed earlier and the difference in the value of first derivative under the condition that the east is chosen is  $X_p$  twice into  $X_p$  plus 3.

The only point of contention now is what is going to be your  $d$  start. The  $d$  value will be the first mid point and I did say that the first mid point is 1 and  $R$  minus 1 by 2 because the first point is 0,  $R$ , the next mid point is plus 1 and minus 1 by 2. If you substitute in the value of  $F$  you have 5 by 4 minus  $R$ . So, that is the initial value of the  $d$  and of course we do have a problem now because as you can see if  $X_p$   $Y_p$  are integer points then the delta  $d$ , the first increment along for south east and east are all integer quantities.

(Refer Slide Time: 53:54)



If you look into the expression that  $X_p$   $Y_p$  are integers so these two expressions are integers but unfortunately  $d$  start will be a floating point number. This is easily avoided by the following; to get rid of the fraction for  $d$  start we take a new variable  $h$  and define it as  $d$  minus 1 by 4. For  $h$  start the new variable for  $h$  is 1 minus  $R$  because if you take  $h$  is equal to  $d - 1$  by 4 and substitute the value of  $d$  start...., what was  $d$  start in the previous expression? It was 5 by 4 minus  $R$ . So take the value of  $d$  start substitute in  $d$  you will get 1 minus  $R$  and since  $R$  is an integer  $h$  is an integer you keep incrementing  $h$  by the first derivative differential delta  $d$  east and south east.

You can say now the next problem is you have to compare  $h$  with minus 1 by 4 which is a fraction because if  $d$  is equal to 0 that is the decision variable which you are evaluating for the mid point, we are comparing with 0,  $h$  will have a minus  $\frac{1}{4}$ , so you have to compare minus 1 by 4 which is also neither good, why? It is because we volt all integer arithmetic if possible both in terms of incrementing, computation, row round function, initial value also to start an integer and compare with integers in this case which is 0.

So instead of comparing  $h$  with minus 1 by 4 what we do, since  $h$  is initialized to and incremented by integers we can just do this by comparing  $h$  with 0, that 1 by 4 factor will not affect you at all because as if all though  $h$  notionally has to be compared with a floating point number but it will be incremented by integers and decremented by integers only, so you do not compare with floating point numbers, you always compare  $h$  with the integer value with 0.

So we have reduced this  $d$  start depress it by an  $h$  start and it is an integer again and we are comparing only with 0 that we look for the values of  $h$  whether it is positive or negative. It is a new decision variable  $h$ , instead of  $d$  we have an  $h$ . We know the relation between  $h$  and  $d$  so we keep incrementing by delta  $d$  east and south east which are again

integers, start with an integer value, increment with integers, increment or decrement depending upon the sign and the value of delta d and also compare with the value 0. So there is no comparison, no addition, no computation of floating point numbers in this case, it is all integer arithmetic done for a circle, remember although the circle had a non-linear equation unlike a straight line which was a linear form  $y$  equal to  $mx$  plus  $c$  we had a square term in both  $xy$  and  $R$  but still we were able to do with integer arithmetic given with Bresenham's criteria for mid point.

We will stop at this point and continue from here onwards where we have understood to apply the mid point criteria for a circle and we look at the first version of the mid point algorithm for drawing a circle with the starting integer arithmetic and then increment by integers. Thank you very much.