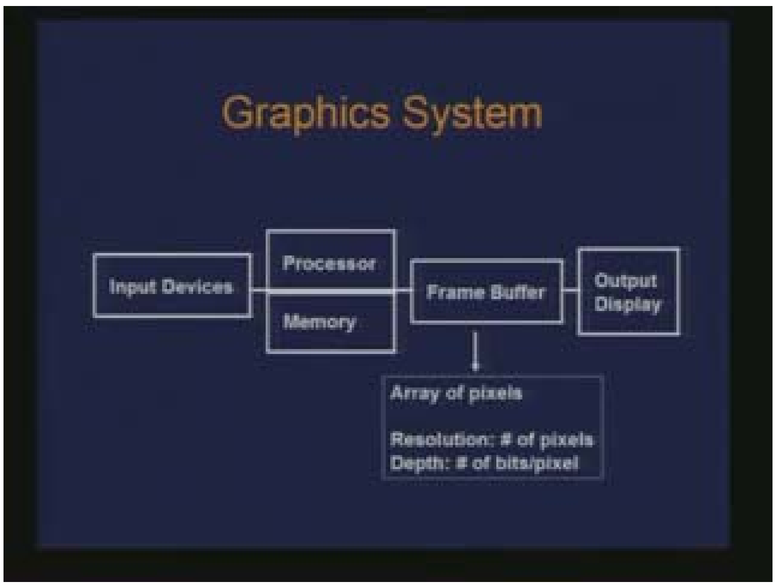
Introduction to Computer Graphics
Dr. Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture - 2
Raster Graphics

Last time we discussed about a couple of basic overview of graphics as a system. One was the rendering pipeline where we looked at the various processing units which are required from converting a three dimensional scene to a 2D image. We also looked at various components in the graphics system which is also shown here. Basically we have the input devices, the processor which is the CPU and now-a-days we also have the GPUs, the different kinds of memory so this whole thing is basically the processing hardware.

(Refer Slide Time 1:08)



In conjunction to that we have the frame buffer which basically acts as an interface between the processing unit and the output display in this frame buffer what we have is a discrete the grid representation of the raster or the screen. So we can basically conceive this to be an array of pixels where we de the spatial resolution of the pixels in area as a number of pixels and the depth in terms of the number of bits per pixel. So we actually closely looked at the interface between the frame buffer and the output display in particular CRT last time. And we also looked at the possibility of enhancing the depth by introducing lookup table which increases the resolution and depth of pixels. Here is the list of devices which possibly can be used for input devices.

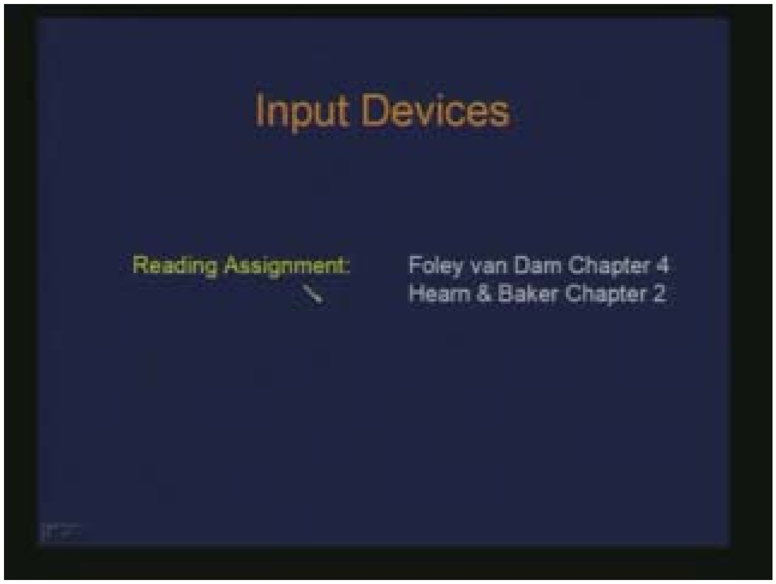
(Refer Slide Time 3:20)

One can actually think of input devices in two ways. One is the physical devices. Physical meaning we observe the devices of the kind keyboard, mouse, tablet etc. These are the physical devices which we use for giving input to the system and there is also a notion of logical devices. Logical devices are basically pertaining to the kind of function which we want to associate as an interface to the program. It is a high level interface with the user program. Typically if you look at the logical devices we generally use are locator that is we want to locate a position of a point so it is a locator device. And the typical physical device which is associated to the locator device as a logical device is a tablet. A tablet is nothing but a pad we have a stylus which des some sort of cross here on the top of that tablet and thereby gives the position of a point on that pad. So, one can also use other devices but tablet seems to be some kind of a natural choice for locator device.

Similarly valuator is a device which gives or returns a floating value. This is some thing similar to the regulator we use to control the speed of the fan. This is why we are using a valuator for. It just returns a real value which we can use for controlling the program.

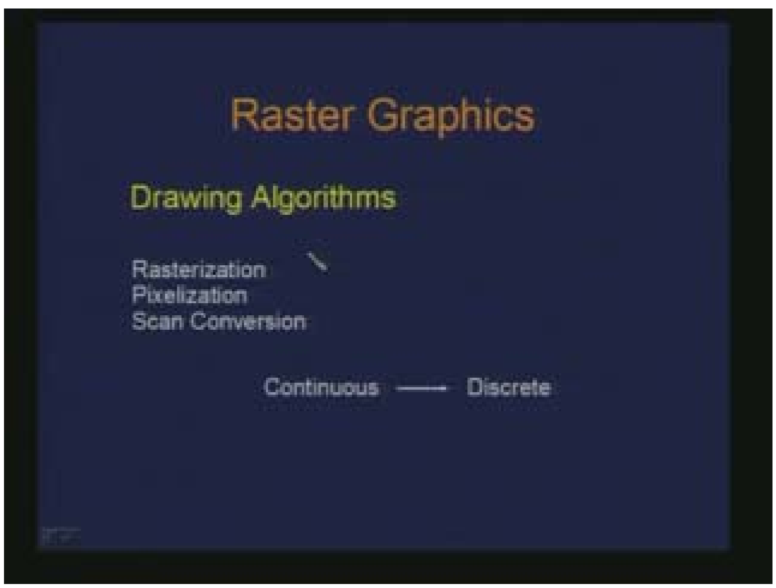
Then there is a suspect device or a selection device so one of the natural choices for a pick device could be a light pen. This is popularly seen now as a device which is associated to the pdis. Then we have a choice device. So these are the device like buttons or the function keys so they just make a choice of connection on and off. And then there is a string device which enables you to give an input as alpha numeric input in the form of a string. So the natural choice for such a device is the keyboard. And there is also a set of 3D input devices like a space ball or a 3D mouse or a data glove which can also be used for the purpose of giving input or manipulating the program in a graphics system.

(Refer Slide Time 6:32)



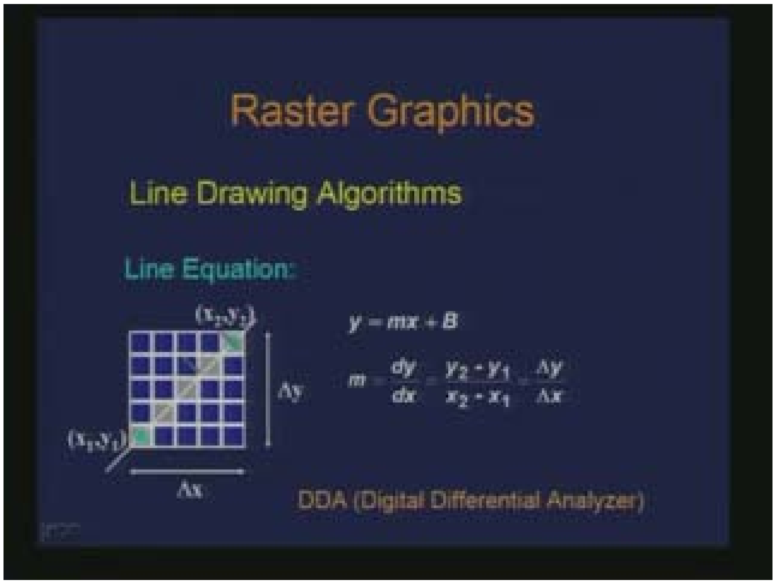
Reading assignment: Foley Van Dam book chapter IV or Hearn and Baker chapter II basically covers the graphics hardware as a system and elaborates further on work we have discussed in terms of graphics system.

(Refer Slide Time 7:00)



Raster graphics process: if you recall the rendering pipeline the last processing unit was rasterization where we wanted to convert the 2D scene which was mostly available in a continuous representation to a discrete representation which is in terms of pixels or the rasters of the display devices. That process is rasterization or pixelization or we also refer to as scan conversion. This process will be looked at in terms of drawings algorithms for various primitives. So we consider the simplest primitive first which is drawing a line. So we have a line as a continuous line given to us and we would like to map it to a raster device which is a collection of pixels.

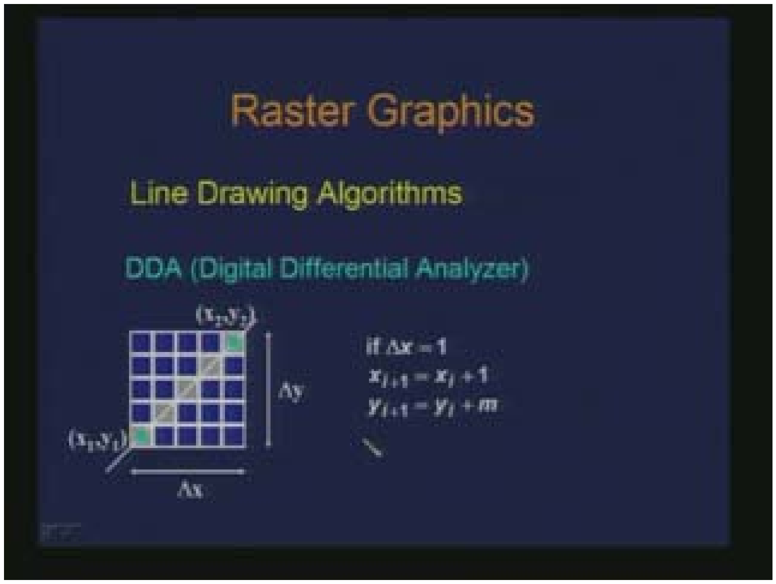
(Refer Slide Time 8:11)



When we treat this line drawing algorithm we have the specification of the line in terms of the two end points x_1y_1 and x_2y_2 so these are the two end points given to us which in turn des the line. So we can always represent this line in the slope and intercept form which is y is equal to mx plus b where m can be computed as dy by dx given as y_2 minus y_1 by x_2 minus x_1 which is nothing but delta y delta x. So now given the slope and the initial point x_1y_1 one can think of devising an algorithm to draw the line intercepted by x_1y_1 x_2y_2. So, if you traverse from x_1y_1 in the direction of the slope and the pixel you encounter you basically draw them or paint them. This is also called as digital differential analyzer as DDA coming from its application in the earlier mechanical engineering or other engineering disciplines. All we are looking at is this differentiation in the form of dy by dx which is given as the slope of the line and we want to use this information to be able to plot the line.

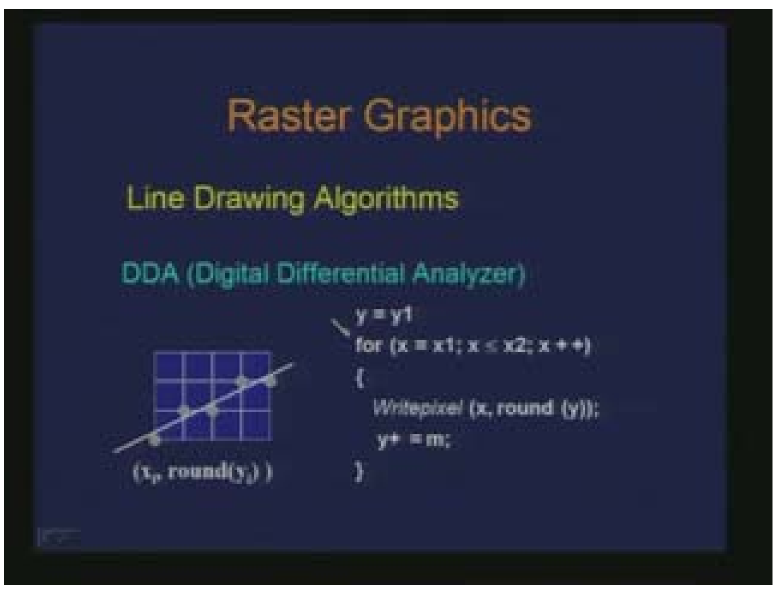
For any point x_iy_i I can write line passing through this point. The next point which could let say xi plus 1 y_i plus 1 this also satisfies the equation of the line and now this xi plus 1 basically can be written in terms of xi plus delta x so my point y_i plus 1 can be written in this form. Therefore if you look at the process which is involved in the drawing of a line basically given this point is already drawn or plotted you would like to answer the question whether to plot this point or that point and that is basically determined by this m.

(Refer Slide Time 11:09)



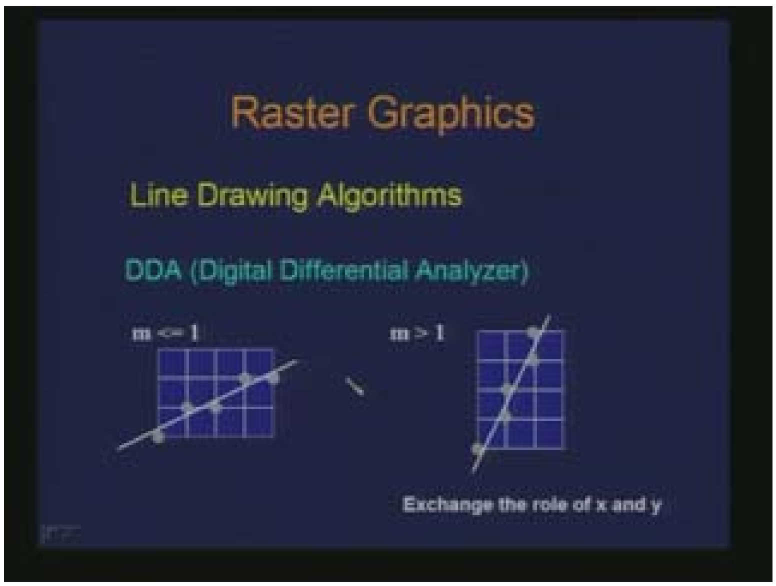
So now if I assume delta x where I make an increment in x by 1 which is valid in the case when we are doing the rasterization we go one pixel by one pixel then all we are saying is that the x is going to be incremented by 1. Therefore xi plus 1 is equal to xi plus 1 this is the delta x and y_i plus 1 would be nothing but y_i plus m. So this is straight coming from the equation of the line. Now I can think of devising a complete algorithm where I start from point.x_1y_1 and I can compute m and knowing m I can decide what pixels to draw. So the algorithm would be something like this.

(Refer Slide Time 12:23)



You have y is equal to y_1 to start with and we traverse from x_1 to x_2 that covers the entire range of the line and we assume that m is less than 1 therefore we are talking when the line is in the first octant. So we change x by 1 every time and decide what point to plot. That means whether to increment y or not depending on what m is. So what we are saying is, we write pixel at position x and take the round of y where y gets incremented by m. This is the simple application of the equation of the line. Therefore at any instance of time for the plot of the line we have xi round y_i. This involves an operation like a round which may not be a desirable operation you want to have. It requires floating point or semantics for instance.

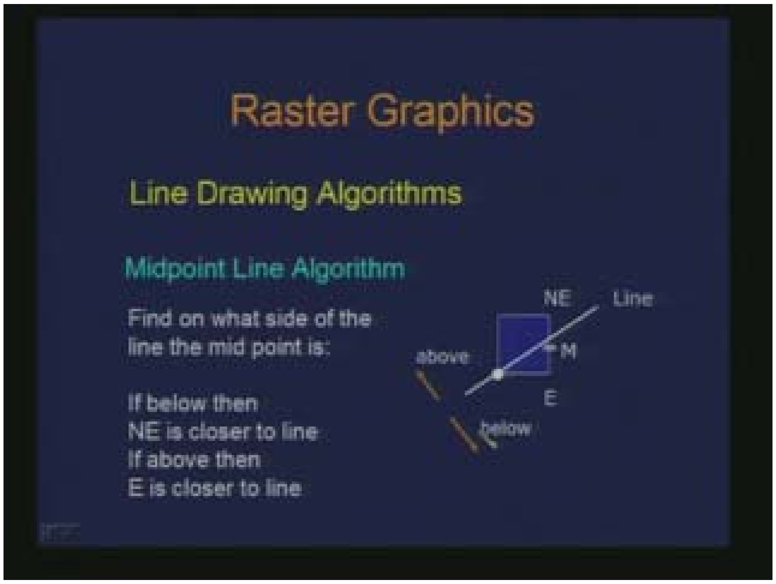
(Refer Slide Time 13:56)



So the other thing is that as far as treating the various octants the algorithm we had given for m is less than or equal to 1 that is the line is in first octant. Now if I have the line in second octant that means m is greater than 1 and the line looks like something like this so all I need to do is exchange the role of x and y. So the algorithm basically does not change and all I need to do is change the role of x and y. I can treat line in any octant. I have an algorithm for a line drawing in any octant. This was the basic idea of using the information of the slope to be able to plot this line and as I said that this required some sort of round operation which one may want to avoid.

Now the question is can we avoid such an operation? If you look at the problem in the following fashion that is, given this point already plotted and again if we are talking about the line to be in the first octant then we are trying to answer the question that when I increase x by 1 whether to plot this point or that point, these are the only two candidates which one can we plot once we have already drawn this point. So, if I device a mechanism by which I can decide in some efficient way to decide whether this point is to be drawn or that point has to be drawn that is what I am looking at and let us say we want to avoid floating point or a symmetric. So, now the question is can we do it?

(Refer Slide Time 16:45)

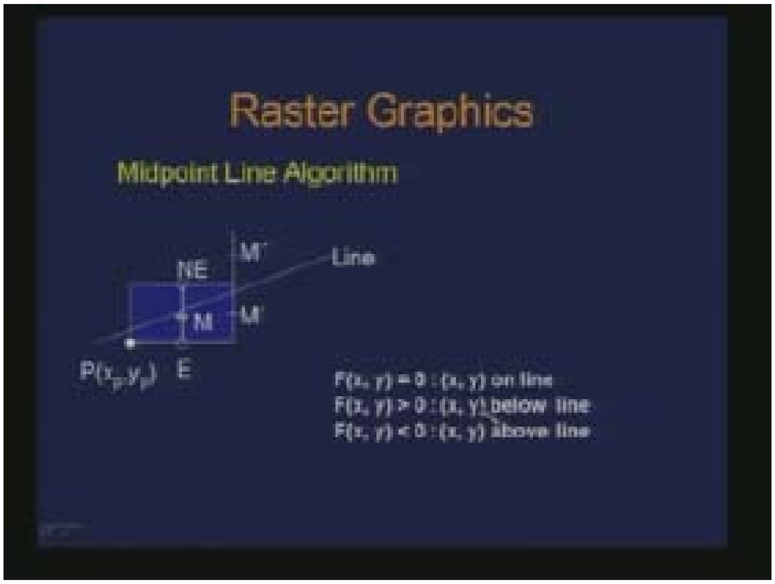


We assume that this is the point which has been already plotted. Now we have to answer the question whether I am going to plot this point E which is east of this point or I am going to plot this which is the north east point of this point and I am trying to look at an efficient method by which I take this decision whether to plot this or this one. Now I assume that the line is in the first octant so now if I consider the mid point which is between E and NE the location of this point M with respect to the line does it decade to me whether I should consider the point E or I should consider the point N because if you see this point M if it turns out to be below this line then this NE point needs to be run because it indicate that NE is actually closer to the line.

Basically we are trying to look at whether this point is closer or that point is closer. So, if m on this side below the line then NE is the point to be plotted. Similarly if M is above the line in that case E is to be plotted because that is closer to the line. Then it boils down to answering the question that whether the mid point m lies below the line or it lies above the line. That is all we are trying to answer. And if we are able to answer this question efficiently then we have a way to decide which point to plot next.

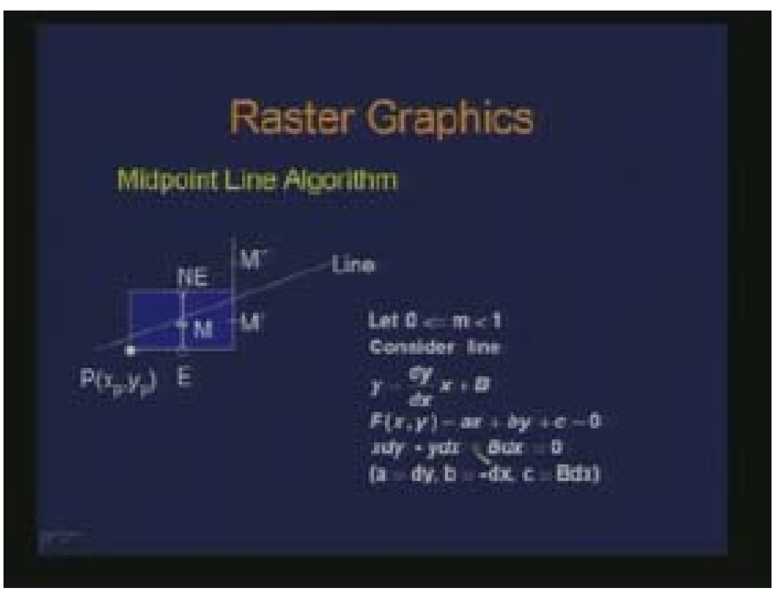
In fact this is a very powerful thing. If you try to further extend this idea of looking at which side of the line the point is in 3D and if I consider a plane instead of this line then we are looking at the situation whether a particular point is in the front of the plane or at the back of the plane and that is a very useful tool to be able to answer many questions in the context of computer graphics. So it is a very useful technique and often we actually talk in terms of these half planes so half planes are suggesting you are on one half space of the plane. Now let us see how we device this.

(Refer Slide Time 20:28)



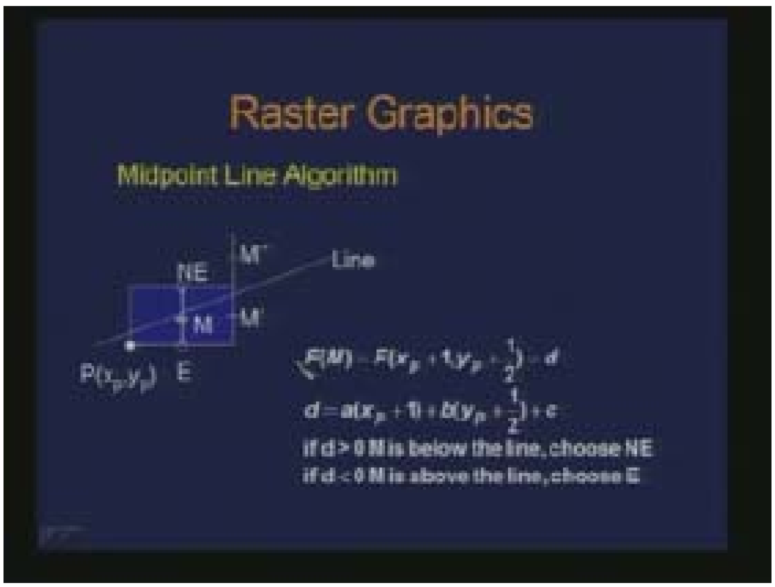
If I consider the line as some implicit function something like F(x, y) is equal to 0 and then if it turns out that the evaluation of this F(x, y) gives me the indication whether the point is below the line or above the line or even on the line. For instance when F(x, y) goes to 0 it suggests that x, y is on the line it satisfies that function. If F(x, y) is greater than 0 then x, y is below the line. Similarly if F(x, y) is less than 0 then x, y is above the line. So all it boils down to is basically the evaluation of this function F(x, y) for x, y. If I have an implicit representation of a line then all I need to do is evaluate that as a function and get the answer whether that gives you greater than 0 or less then 0 and that gives me the answer whether the particular point is below the line or above the line.

(Refer Slide Time 21:56)



Now if we consider an implicit representation of the line which is nothing but F(x, y) is equal to ax plus by plus c goes to 0 so that is an implicit form of the line. Again if I consider the line in the first octant thereby m is between 0 and 1 and the slope and interception representation of the line is y is equal to dy by dx x plus B then i can map the coefficients a b c here. It is xdy minus ydx plus bdx is equal to 0 that gives me a is equal to dy, b is equal to minus dx and c is equal to bdx so these two are basically equivalent. I can always write line in either of the form. This actually gives me the evaluation of the coefficients a, b, c which I can compute from the input of the line. For instance, if I am given the two end points of the line I should be able to compute the necessary coefficients which I am using for the evaluation of F(x, y). That is what precisely is helping for now.

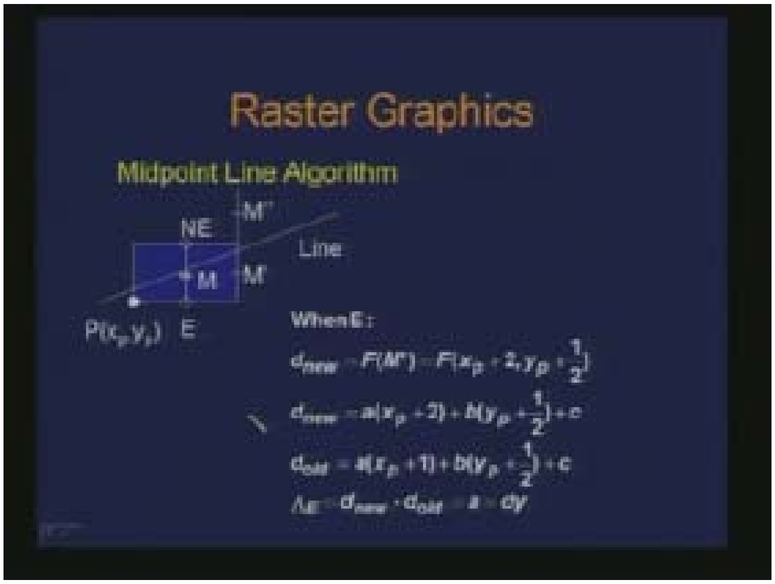
(Refer Slide Time 23:56)



Therefore, once I have this implicit representation now the question becomes that I would perform an evaluation of this F which is to see whether the mid point m is below the line or above the line. So basically that is to say that I have a decision variable d which is nothing but substituting the corresponding x, y values for the point m that is xp plus 1 and yp plus 1 by 2 if I was given that the point xpyp is the point which has been plotted last. so as far as the decision variable d is concerned now again going back to the implicit representation of the line as F(x, y) I can have substituting the xy as xp plus 1 and yp plus 1 by 2 respectively I get d is equal to a xp plus 1 plus b(yp plus 1 by 2) plus c.

Now if I see that d is greater than M then d is greater than 0. If I see that d is greater than 0 then M is below the line then I would choose the north east point to be plotted which is this point. Therefore M is below the line I choose this point. Similarly when d is less than 0 that is M is above the line then I will choose E point to be plotted here. Basically evaluation of the sign of d determines whether I should plot E or NE.

(Refer Slide Time 26:13)



A further development of the algorithm is shown here. after doing the evaluation of FM which gave you the result whether to plot E or NE once you have decided whether you are going to plot E or NE then you are going to do further evaluation for either M' or M''. If you have plotted E then the mid point to be considered is going to be M' and if you have plotted NE then the mid point which needs to be consider is M'' and that is what I try to do.

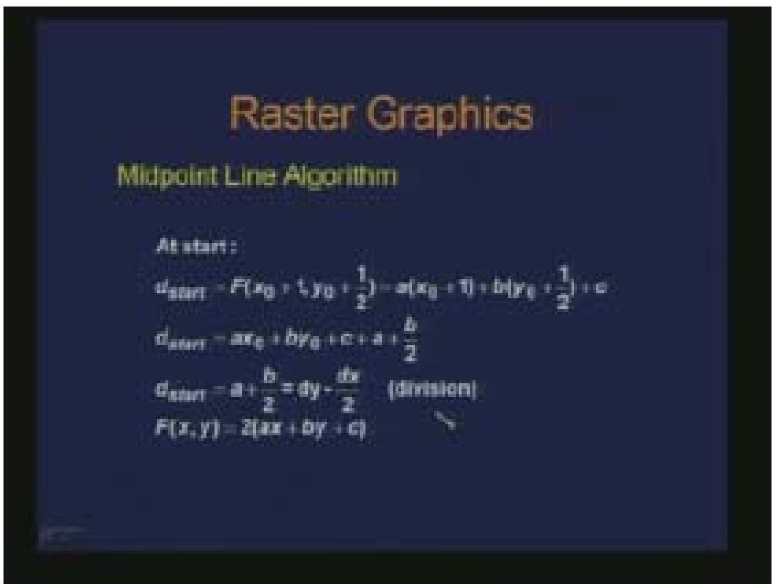
I compute the new d the new decision variable which is the evaluation of F(x, y) for M' which is basically M' is nothing but with respect to xpyp it is xp plus 2 in x and yp plus 1 by 2 so that is what I do. I substitute xy as xp plus 2 and yp plus1 by 2 in my implicit form of the line and compute the d new that is what d new is and I had computed earlier d old which was the evaluation of M as this axp plus 1 plus b(yp plus ½) plus c.

Now if I take the difference of d new and d old it gives me a delta or a change in d when I had considered that E was to be drawn. So when E was to be drawn the change in d is given as d new minus d old which is nothing but a or dy. So, what is the motivation of computing this delta? The motivation of computing this delta is primarily to see the updation of d. So you would be basically looking at the new instance of d every time you have plotted a point as what change has happen to d and that would depend on the fact whether you have chosen to draw the east point or the north east point. So this delta basically gives you the change in d.

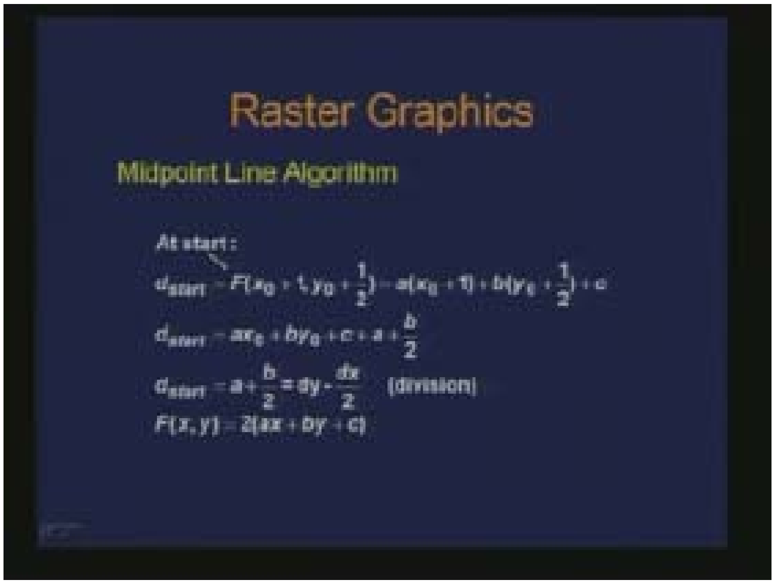
Similarly I can conduct the computation when I had plotted NE so instead of E when I have plotted NE what will be the change in d? That is what I do. So, if the point NE was plotted the new d involves the evaluation of F(x, y) for M double prime so this is an evaluation of m double prime where I substitute for x as xp plus 2 and for y(yp plus 3 by 2) so we are looking again with respect to this xpyp that is where my m double prime is.

So, if I substitute this as 2F(x, y) I get d new as this; axp plus t2 plus b(yp plus 3 by 2) plus c and the d old as earlier computed was a xp plus 1 plus b(yp plus 1 by 2) plus c, again I take the difference to obtain the delta which is the change in d when NE was chosen the point to be plotted last. So this delta again gives me an updation of d. If I had plotted the point NE what is the change in d I would have? So this delta NE gives me d new minus d old which is nothing but a plus b and which again can be written in terms of dy minus dx. So basically I have set the ways to do the computation of d which I call it as the decision variable and also the respective changes I would have to do depending on whether I have plotted the east point or the north east point last. That basically sets up whatever is required for the algorithm of the line drawing. So the entire algorithm may look like something like this. Actually before we do this we need to worry about what happens at the start.

(Refer Slide Time 32:15)

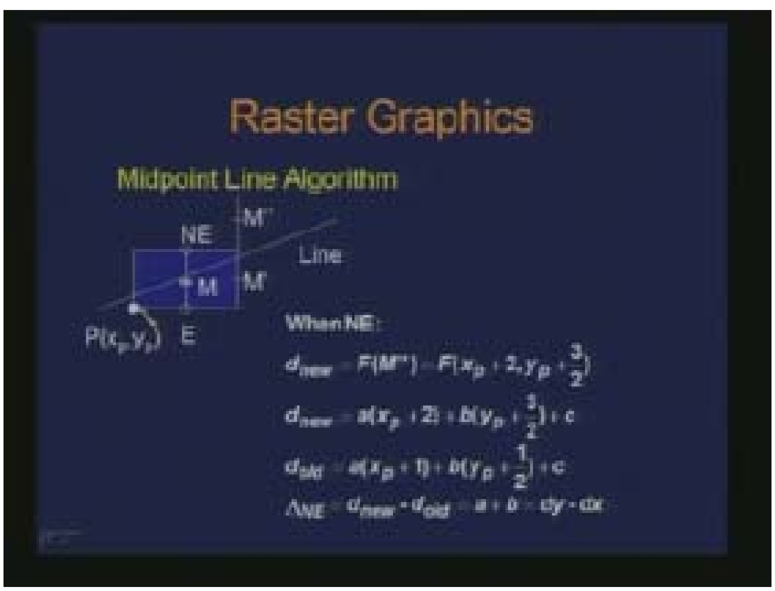


(Refer Slide Time 32:18)



The start of this algorithm basically means I have plotted or I know that a point lies as an end point to the line so I need to do in an evaluation of what will happen to the d which is my decision variable when I start which basically boils down to doing the computation of F at x_0 plus 1 and y_0 plus ½.

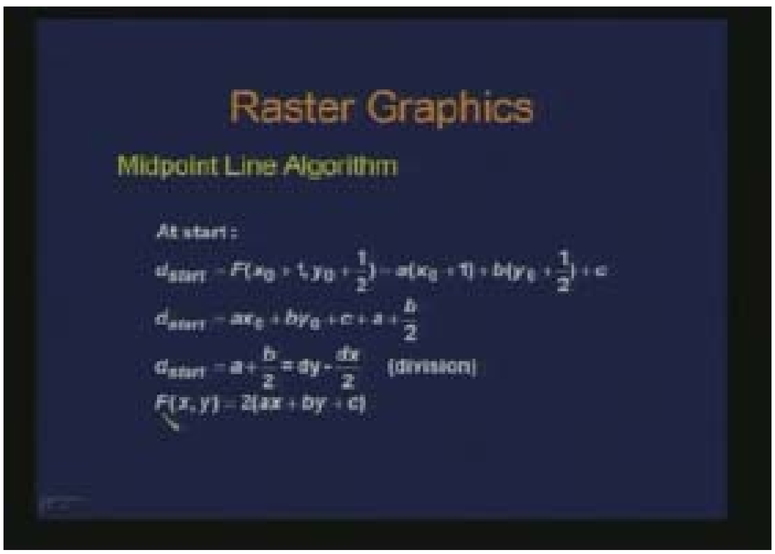
(Refer Slide Time 33:06)



Actually if you see at this point d a plotted point and this is x_0y_0 so d start would be the evaluation of this point and that is what we are trying to do here. So the d start is basically evaluation of x_0 plus 1 y_0 plus 1 by 2 which comes as this ax_0 plus 1 plus $b(y_0$ plus 1 by 2) plus c which is the implicit form of the line. Now if you further simplify this you get d start as ax_0 plus by_0 plus c plus a plus b by 2. Now if you observe this term ax0plus by_0 plus c this is nothing but the equation of the line we had ax plus by plus c for the point x_0y_0 and the fact that x_0y_0 lies on the line this whole expression becomes 0 because it is on the line therefore it satisfies that F(x, y) is equal to 0. Now d start is nothing but a plus b by 2.

Now if I substitute for a and b in terms of dy and dx from the slope interceptor representation of the line then I have this as dy minus dx by 2. So, what you observe here is that when I compute d start which is now dy minus dx by 2 there is a division operation so dx by d2. So this may further require floating point operation if I do not do any thing about it. So if I want to avoid this division what should I do? If you look at the evaluation where d is coming from is basically from F(x, y) which was ax plus by plus c.

(Refer Slide Time 35:40)



Now instead of considering just ax plus by plus c if I consider two times ax plus by plus c it does not change anything. So the benefit of that is I would avoid this division by 2 and do the corresponding changes to whatever we had computed. for instance, the delta and the ds keeping in mind the new evaluation function as twice ax plus by plus c because evaluation of this or F(x, y) as axy plus by plus c does not really change in terms of what we are deciding for the decision variable and the corresponding delta of those decision variables. T

Therefore now with this d start computed and the redefinition of f(x, y) the entire algorithm would look like this. So what do I have? I have dx computed from the exponent x_2 which are the two end points x of the two end points similarly dy as y_2 minus y_1 so dx and dy is basically computed directly from x_1y_1 and x_2y_2.

The d start we have just seen is 2dy minus dx so without multiplying by 2 it was dy minus dx by 2 so we further normalize this by multiplying by 2 and we get d start as 2dy minus dx. And the corresponding deltas so the delta E which was earlier computed as dy and now with the modified f(x, y) as 2(dx plus dy plus c) gives you the delta Es 2dy. Similarly delta NE is now 2(dy minus dx). So correspondingly we change all the evaluations so delta E and delta NE also change. Now if the start point is x_1y_1 then I assign x as x_1 and y as y_1 and then span all the values of x to be able to plot the line. So here I write pixel x, y so write pixel is nothing but a command for drawing a point at that particular location so it is either plot pixel or write pixel or whatever you want to use and then I run into this loop from x less than x_2.

Remember we are looking at line in the first octant so that is why I am changing x and then decide whether to change y or not. Now while x is less than x_2 I have the decision variable d and all I need to do is look at the sign of that d and see whether it is less than or equal to 0 or greater than 0. So this equality sign can be either accommodated in this or in the else part so I could have also have written as d less than 0 and then the else part would have taken greater than or equal to 0 but I just need to be consistent.

Therefore let us say I include this equality in this. That means whenever it happens to be equal I plot accordingly the east point. So if I have the d less than or equal to 0 the update to d is through delta E that is I just change in x so the next point to be plotted is just forward in x no change in y so it is the east point which I plot and accordingly I update the d it is just by a constant term. And if it is else that means I have to plot the north east point wherein I would need to change x as well as y so I increment x by y and I also increment y by y and accordingly I change the decision variable d with the delta value as delta NE. Now you can see and appreciate that why did we required this d delta E and delta NE.

So this was primarily to do the updates of d appropriately. And it turns out that these delta E and delta NE are constants and do not depend on x and y when you are running in this loop. So this way I can basically plot the entire line between x_1y_1 and x_2y_2. And you also observe that the computation which it needs to perform is all integer arithmetic so there is no floating point operation needed. So this particular algorithm which is called as midpoint line algorithm midpoint because of the fact that midpoint is assisting in the process of evaluating what point to draw in the line. Or this is very close to the Bresenham's algorithm which is very popularly known in computer graphics. So Bresenham's algorithm and midpoint line algorithm are very similar.

So what we have seen here is drawing of a line. One can now think of doing the drawing of other primitives which are slightly more complex. For instance one can talk about drawing a circle, drawing an ellipse and things like that. Next time what we are going to look at is how we plot a circle keeping the frame work of midpoint evaluation. So what we have done in the case of line drawing we are trying to ask a certain question in terms of which point to plot by doing an evaluation of a midpoint so can we use the similar idea and do the plot of the circle.