COMPUTER GRAPHICS

UNIT-1

OverviewofComputerGraphics

ApplicationofComputerGraphics

Computer-AidedDesign forengineeringandarchitecturalsystemsetc.
Objects maybe displayed in a wireframe outline form. Multi-window environment is alsofavoredforproducingvarious zoomingscales andviews.
Animationsareusefulfortestingperformance.

PresentationGraphics
Toproduceillustrationswhichsummarizevariouskindsofdata.Except2D,3Dgraphicsaregood tools for reportingmore complexdata.

ComputerArt
Painting packages are available. With cordless, pressure-sensitive stylus, artists canproduce electronic paintings which simulate different brush strokes, brush widths, andcolors. Photorealistic techniques, morphing and animations are very useful in commercialart.For films,24frames persecondarerequired.Forvideomonitor,30framespersecondarerequired.

Entertainment
Motionpictures,Musicvideos, andTVshows, Computergames

EducationandTraining
Training with computer-generated models of specialized systems such as the training ofshipcaptains and aircraft pilots.

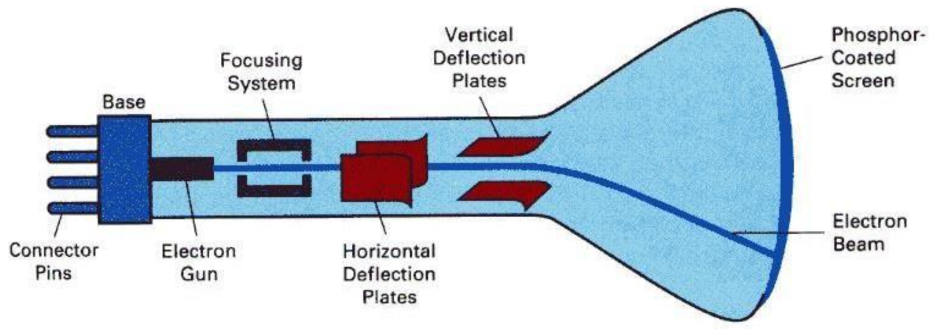
Visualization
Foranalyzingscientific,engineering,medicalandbusinessdataorbehavior.Convertingdatato visual form canhelp to understand massvolumeof data veryefficiently.

ImageProcessing
Imageprocessingis toapplytechniques tomodifyorinterpretexistingpictures.Itiswidelyused in medical applications.

GraphicalUserInterface
Multiplewindow,icons,menusallow acomputer setuptobeutilizedmoreefficiently.

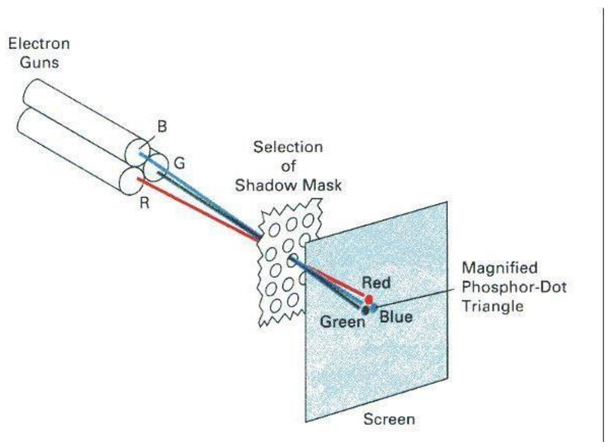
VideoDisplaydevices

Cathode-RayTubes(CRT)-stillthemostcommon videodisplaydevicepresently



ElectrostaticdeflectionoftheelectronbeaminaCRT

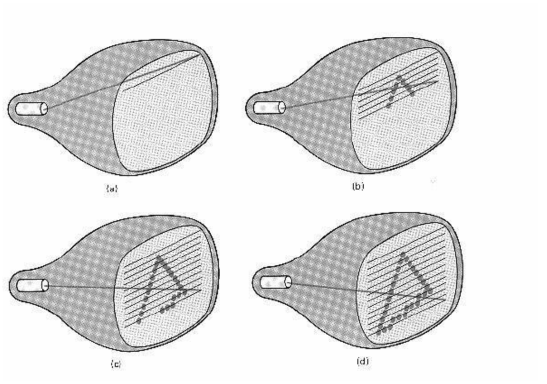
An electron gun emits a beam of electrons, which passes through focusing and deflectionsystems and hits on the phosphor-coated screen. The number of points displayed on a CRT isreferred to as r**esolutions** (eg. 1024x768). Different phosphors emit small light spots ofdifferent colors, which can combine to form a range of colors. A common methodology forcolorCRT displayis the**Shadow-mask**meth



Illustrationof ashadow-maskC

The light emitted by phosphor fades very rapidly, so it needs to redraw the picture repeatedly.Thereare2 kindsofredrawingmechanisms: Raster-Scanand Random-Scan

Raster-Scan



The electron beam is swept across the screen one row at a time from top to bottom. As itmovesacrosseachrow,thebeamintensity isturnedonandofftocreateapatternofilluminated spots. This scanning process is called refreshing. Each complete scanning of ascreenis normallycalleda**frame**.

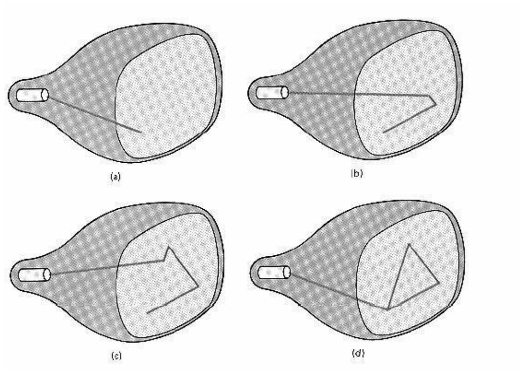
The refreshing rate, called the**framerate**, is normally 60 to 80 frames per second, ordescribedas 60 Hz to 80 Hz.

Picture definition is stored in a memory area called the**framebuffer**. This frame bufferstores the intensity values for all the screen points. Each screen point is called a **pixel** (pictureelement).

On black and white systems, the frame buffer storing the values of the pixels is called a**bitmap**. Each entry in the bitmap is a 1-bit data which determine the on (1) and off (0) of theintensityof thepixel.

On color systems, the frame buffer storing thevalues of thepixels is calleda**pixmap**(Though nowadays many graphics libraries name it as bitmap too). Each entry in the pixmapoccupies a number of bits to represent the color of the pixel. For a true color display, thenumber of bits for each entry is 24 (8 bits per red/green/blue channel, each channel 2^8=256levelsofintensityvalue,ie.256voltagesettingsforeach ofthered/green/blueelectron guns).

Random-Scan(VectorDisplay)



The CRT's electron beam is directed only to the parts of the screen where a picture is to bedrawn. The picture definition is stored as a set of line-drawing commands in a refresh displayfileor arefresh buffer inmemory.

Random-scan generally have higher resolution than raster systems and can produce smoothlinedrawings, however it cannot displayrealisticshaded scenes.

DisplayController

For a raster display device reads the frame buffer and generates the control signals for thescreen, ie. the signals for horizontal scanning and vertical scanning. Most display controllersinclude a **color map** (or video look-up table). The major function of a color map is to provideamappingbetween the input pixel valueto theoutput color.

Anti-Aliasing

On dealing with integer pixel positions, jagged or stair step appearances happen veryusually.Thisdistortionofinformationduetoundersamplingiscalledaliasing.A numberofant aliasingmethods havebeen developed to compensate this problem.

Onewayis to displayobjects at higher resolution. However thereisalimit to how bigwecanmaketheframe bufferand still maintainingacceptablerefresh rate.

DrawingaLineinRasterDevices

DDAAlgorithm

In computer graphics, a hardware or software implementation of a digital differential analyzer(DDA) is used for linear interpolation of variables over an interval between start and end point.DDAs are used for rasterization of lines, triangles and polygons. In its simplest implementationthe DDA Line drawing algorithm interpolates values in interval [(xstart, ystart), (xend, yend)] bycomputing for each xi the equations $x_i = x_{i-1} + 1/m$, $y_i = y_{i-1} + m$, where $\Delta x = x_{end} - x_{start}$ and$\Delta y = y_{end} - y_{start}$ and $m = \Delta y/\Delta x$.

The dda is a scan conversion line algorithm based on calculating either dy or dx. A line issampledatunitintervalsinonecoordinate andcorrespondingintegervaluesnearestthelinepath

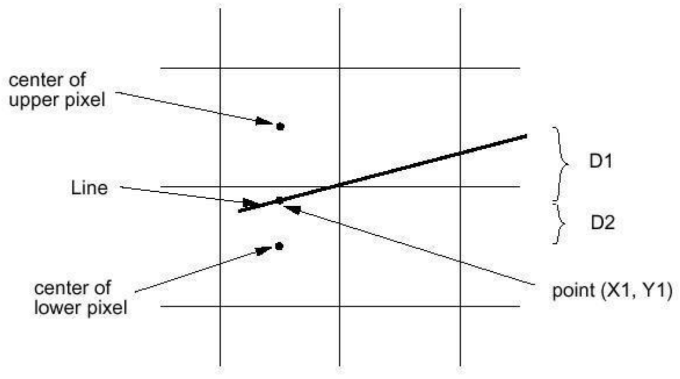
aredeterminedforother coordinates.
Consideringalinewithpositive slope,iftheslope isless than or equalto 1,wesample at unitxintervals(dx=1)and compute successiveyvaluesas

Subscriptktakesinteger values startingfrom0,for the1stpointand increasesbyuntilendpointisreached.yvalueis rounded offtonearest integer tocorrespond toascreenpixel.
Forlineswith slope greaterthan 1,wereversethe roleof x andyi.e. wesampleatdy=1andcalculate consecutivexvalues as

Similarcalculationsarecarried outtodeterminepixelpositionsalongaline withnegativeslope.Thus, if the absolute value of the slope is less than 1, we set dx=1 ifi.e. the starting extremepointis at

Thebasicconceptis:

- Alinecanbespecifiedintheform:
  y=mx+c
- Letm bebetween 0to 1,then theslopeoftheline is between0 and45 degrees.
- For the x-coordinate of the left end point of the line, compute the corresponding y valueaccording to the line equation. Thus we get the left end point as (x1,y1), wherey1 maynotbe an integer.
- Calculate the distance of (x1,y1) from the center of the pixel immediately above it and call itD1
- Calculate the distance of (x1,y1) from the center of the pixel immediately below it and call itD2
- IfD1issmallerthanD2,itmeansthatthelineiscloserto theupperpixelthanthelowerpixel,then, weset theupper pixel toon; otherwise weset thelower pixel toon.
- Thenincreatementxby1andrepeatthesameprocessuntilxreachestherightendpointofthe line.
- Thismethod assumes thewidth ofthelineto be zero



Bresenham'sLineAlgorithm

This algorithm is very efficient since it use only incremental integer calculations. Instead ofcalculatingthenon-integralvalues of D1andD2 fordecisionofpixel location,itcomputesavalue,p, which is defined as:

p=(D2-D1)*horizontal lengthoftheline
ifp>0,it meansD1 issmallerthan D2,andwecandeterminethepixel locationaccordingly

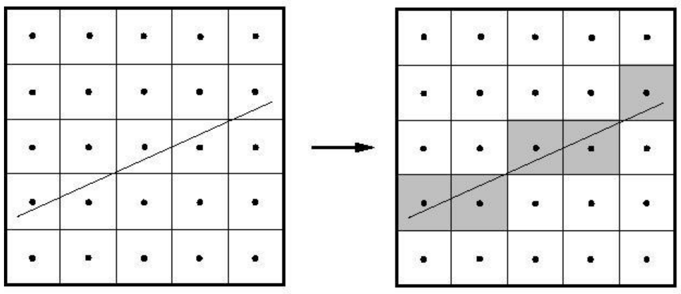
However,the computation ofpis veryeasy:
Theinitial valueofpis 2 *vertical height of theline-horizontal lengthof theline.

Atsucceedingxlocations,ifp hasbeen smallerthan0, then,weincrementpby2*verticalheight of the line, otherwise we increment p by 2 * (vertical height of the line - horizontallengthof the line)

Allthecomputationsare onintegers.Theincrementalmethod isappliedto

```
voidBresenhamLine(int x1,int y1,intx2,int y2)
{ int x, y, p, const1, const2; /*
    initializevariables */ p=2*(y2-y1)-(x2-
    x1);const1=2*(y2-y1); const2=2*((y2-
    y1)-(x2-x1));

    x=x1;y=
    y1;
    SetPixel(x,y);
    while(x<xend){x++;
        if(p<0)
        {p=p+const1;
        }
        else
        {y++;
            p=p+const2;
        }
        SetPixel(x,y);
    }
}
```



Bitmap

- Agraphicspatternsuchasaniconoracharactermaybeneededfrequently,ormayneedto be re-used.
- Generatingthe patterneverytime when neededmaywaste alot of processingtime.
- Abitmapcanbeusedtostoreapatternandduplicateittomanyplacesontheimageoron the screen with simple copyingoperations.

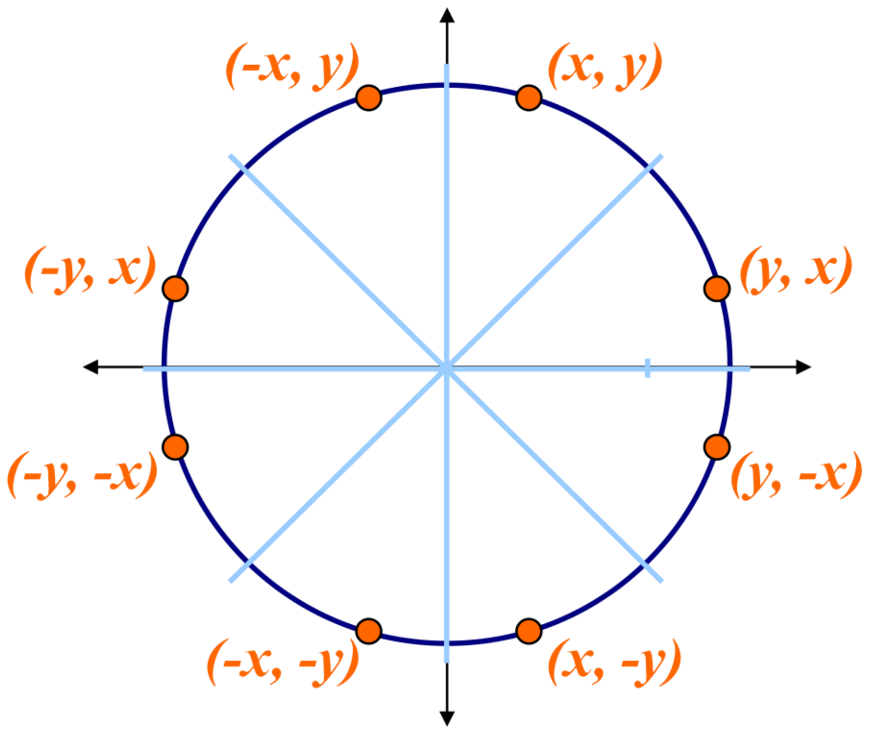
MidPointcircleAlgorithm

However,unsurprisinglythisisnotabrilliantsolution!
Firstly, the resulting circle has large gaps where the slope approaches the verticalSecondly,thecalculations arenot veryefficient

Thesquare(multiply)operations

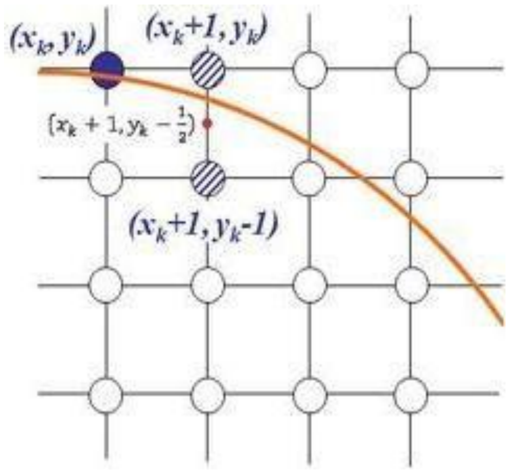
Thesquare root operation– tryreallyhard toavoid these!

Weneed amore efficient, moreaccuratesolution.
Thefirstthingwecannoticetomakeourcircledrawingalgorithm moreefficientisthat circlescentredat (0, 0) have*eight-waysymmetry*



Similarly to the case with lines, there is an incremental algorithm for drawing circles – the *mid-point circlealgorithm*
Inthe mid-point circlealgorithm weuseeight-waysymmetryso onlyevercalculate thepointsforthe topright eighth of acircle,andthen usesymmetrytogettherest ofthe points

Assume that we havejustplottedpoint(x_k,y_k)
Thenextpointisachoice between $(x_k+1, y_k)$and(x_k+1,y_k-1)



Wewould liketo choose

the point that is nearest
totheactual circle
Sohow do wemakethischoice?

Let’sre-jigthe equationofthe
circleslightlytogiveus:Theequation evaluates as follows:

$$f_{circ}(x,y) = x^2 + y^2 - r^2$$

$$f_{circ}(x,y)\begin{cases} < & 0, \\ = & 0,0, \\ > & \end{cases}$$

<0if(x,y)isoutsidethecircleboundary
=0if (x,y)isonthecircleboundary
>0if (x,y)isinsidethecircleboundary

Byevaluatingthis functionat themidpointbetween the candidate pixelswecan makeourdecision

Assuming we have just plotted the pixel at (x_k,y_k) so we need to choose between (x_k+1,y_k) and(x_k+1,y_k-1)
Ourdecisionvariablecanbedefinedas:

$$p_k = f_{circ}(x_k + 1, y_k - \tfrac{1}{2})$$
$$= (x_k + 1)^2 + (y_k - \tfrac{1}{2})^2 - r^2$$

If p_k<0the midpointis insidethecircle and andthepixelat$y_k$is closerto
thecircleOtherwisethemidpoint is outsideandy_k-1 is closer

To ensure things are as efficient as possible we can do all of our calculations incrementallyFirstconsider:

$$p_{k+1} = f_{circ}\left(x_{k+1}+1, y_{k+1} - \frac{1}{2}\right)$$

$$= [(x_k + 1) + 1]^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2$$

$$p_{k+1} = p_k + 2(x_k + 1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1$$

where$y_{k+1}$is either$y_k$ory_k-1 dependingon thesign of p_k

Thefirstdecisionvariableisgiven as:

$$p_0 = f_{circ}(1, r - \frac{1}{2})$$

$$= 1 + (r - \frac{1}{2})^2 - r^2$$

$$= \frac{5}{4} - r$$

Thenifp_k<0then thenext decisionvariable isgiven as:

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

If p_k>0 thenthedecision variableis:

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_k + 1$$

Input radius r and circle centre (x_c, y_c), then set the coordinates for the first point on thecircumferenceofacirclecentred on theorigin as:

$$(x_0, y_0) = (0, r)$$

- Calculatetheinitialvalueof thedecisionparameteras:

$$p_0 = \frac{5}{4} - r$$

- Starting with $k = 0$ at each position x_k, perform the following test. If $p_k < 0$, the next pointalongthecirclecentred on$(0, 0)$is$(x_k+1, y_k)$and:

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

Otherwise the nextpointalongthecircleis$(x_k+1, y_k-1)$and:

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

Determinesymmetrypointsin theotherseven octants

Move each calculated pixel position (x, y) onto the circular path centred at (x_c, y_c) to plot thecoordinatevalues:

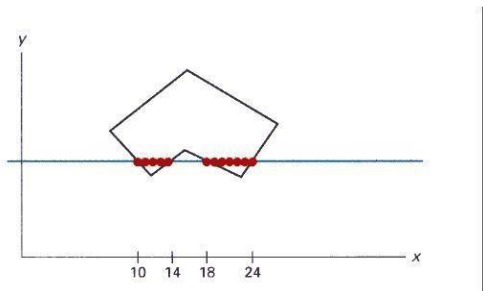
$$x = x + x_c \quad y = y + y_c$$

Repeatsteps3 to5until$x >= y$

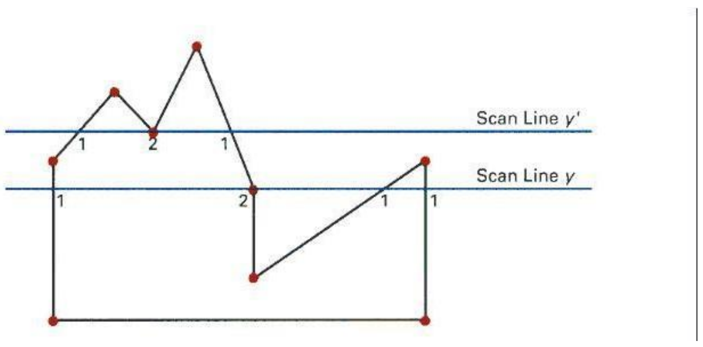
To see the mid-point circle algorithm in action lets use it to draw a circle centred at (0,0) withradius10

Scan-LinePolygonFillAlgorithm

- Basic idea: For each scan line crossing a polygon, this algorithm locates the intersectionpoints of the scan line with the polygon edges. These intersection points are shorted fromleftto right.Then,wefillthe pixels betweeneachintersection pair.

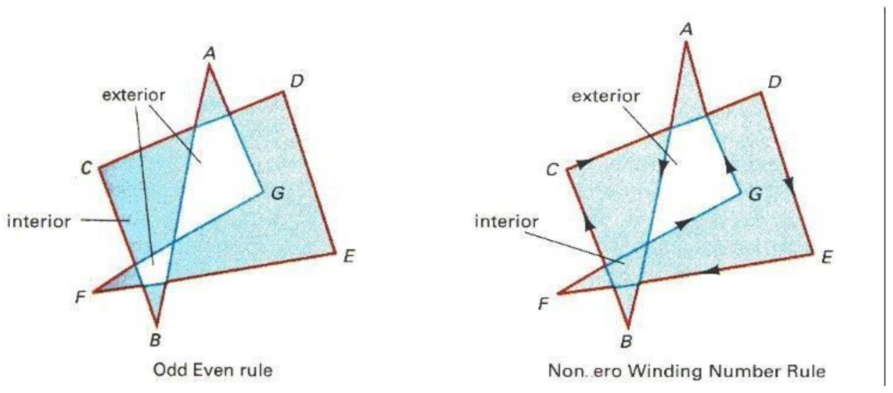


- Some scan-line intersection at polygon vertices require special handling. A scan linepassingthroughavertexasintersectingthepolygontwice.Inthiscasewemayormaynot add 2 points to the list of intersections, instead of adding 1 points. This decisiondepends on whether the 2 edges at both sides of the vertex are both above, both below, orone isabove and one is belowthe scan line. Only for the case if bothareabove or botharebelow thescan line, then wewill add 2 points.



Inside-OutsideTests:

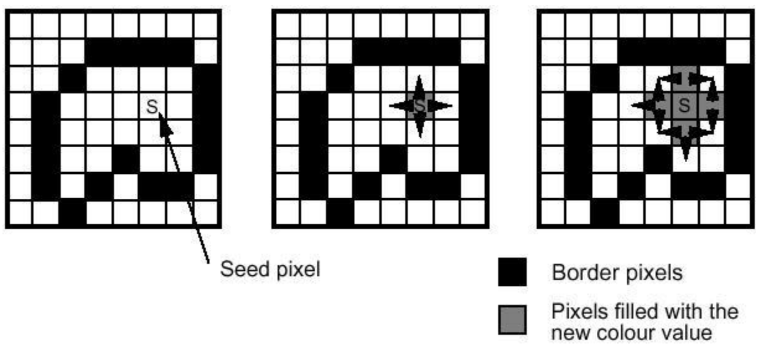
- The above algorithm only works for standard polygon shapes. However, for the caseswhich the edge of the polygon intersects, we need to identify whether a point is an interioror exterior point. Students may find interesting descriptions of 2 methods to solve thisproblem in manytext books: odd-even rule and nonzero windingnumber rule.



Boundary-FillAlgorithm

- Thisalgorithmstartsatapointinsidearegionandpainttheinterioroutwardtowardstheboundary.
- Thisisasimplemethodbutnotefficient:1.Itisrecursivemethodwhichmayoccupyalargestack sizein themain memory.

```
voidBoundaryFill(intx,inty,COLORfill,COLORboundary)
{                    COLOR
    current;current=GetPixel(
    x,y);
    if(current<>boundary)and(current<>fill)then{SetPixel(x,y,fill);
        BoundaryFill(x+1,y,fill,boundary);Bo
        undaryFill(x-
        1,y,fill,boundary);BoundaryFill(x,y+1
        ,fill,boundary);BoundaryFill(x,y-
        1,fill,boundary);
    }
}
```



- Moreefficientmethodsfillhorizontalpixelspandsacrossscanlines,insteadofproceedingto neighboringpoints.
-

Flood-FillAlgorithm

- Flood-FillissimilartoBoundary-Fill.ThedifferenceisthatFlood-FillistofillanareawhichInot defined byasingle boundarycolor.

```
voidBoundaryFill(intx,inty,COLORfill,COLORold_color)
{   if   (GetPixel(x,y)==   old_color)
    {SetPixel(x,y,fill);
        BoundaryFill(x+1,y,fill,boundary);Bo
        undaryFill(x-
        1,y,fill,boundary);BoundaryFill(x,y+1
        ,fill,boundary);BoundaryFill(x,y-
        1,fill,boundary);
    }
```