

Q.2 a. List the operating characteristics of the following display technologies:

- i. Raster Refresh Systems**
- ii. Plasma Panels**
- iii. LCD**

Answer:

i) A raster scan, or raster scanning, is the rectangular pattern of image capture and reconstruction in television. By analogy, the term is used for raster graphics, the pattern of image storage and transmission used in most computer bitmap image systems. The word *raster* also rastrum, an instrument for drawing musical staff lines. The pattern left by the tines of a rake, when drawn straight, resembles the parallel lines of a raster: this line-by-line scanning is what creates a raster. It's a systematic process of covering the area progressively, one line at a time. Although often a great deal faster, it's similar in the most-general sense to how one's gaze travels when one reads lines of text.

In raster scanning, the beam sweeps horizontally left-to-right at a steady rate, then blanks and rapidly moves back to the left, where it turns back on and sweeps out the next line. During this time, the vertical position is also steadily increasing (downward), but much more slowly – there is one vertical sweep per image frame, but one horizontal sweep per line of resolution. Thus each scan line is sloped slightly "downhill" (towards the lower right), with a slope of approximately $-1/\text{horizontal resolution}$, while the sweep back to the left (retrace) is significantly faster than the forward scan, and essentially horizontal. The resulting tilt in the scan lines is very small, and is dwarfed in effect by screen convexity and other modest geometrical imperfections.

There is a misconception that once a scan line is complete, a CRT display in effect suddenly jumps internally, by analogy with a typewriter or printer's paper advance or line feed, before creating the next scan line. As discussed above, this does not exactly happen: the vertical sweep continues at a steady rate over a scan line, creating a small tilt. Steady-rate sweep is done, instead of a staircase of advancing every row, because steps are hard to implement technically, while steady-rate is much easier. The resulting tilt is compensated in most CRTs by the tilt and parallelogram adjustments, which impose a small vertical deflection as the beam sweeps across the screen. When properly adjusted, this deflection exactly cancels the downward slope of the scanlines. The horizontal retrace, in turn, slants smoothly downward as the tilt deflection is removed; there's no jump at either end of the retrace. In detail, scanning of CRTs is done by magnetic deflection, by changing the current in the coils of the deflection yoke. Rapidly changing the deflection (a jump) requires a voltage spike to be applied to the yoke, and the deflection can only react as fast as the inductance and spike magnitude permit. Electronically, the inductance of the deflection yoke's vertical windings is relatively high, and thus the current in the yoke, and therefore the vertical part of the magnetic deflection field, can change only slowly.

In fact, spikes *do* occur, both horizontally and vertically, and the corresponding horizontal blanking interval and vertical blanking interval give the deflection currents settle time to retrace and settle to their new value. This happens during the blanking interval.

In electronics, these (usually steady-rate) movements of the beam[s] are called "sweeps", and the circuits that create the currents for the deflection yoke (or voltages for the horizontal deflection plates in an oscilloscope) are called the sweep circuits. These create

a sawtooth wave: steady movement across the screen, then a typically rapid move back to the other side, and likewise for the vertical sweep.

Furthermore, wide-deflection-angle CRTs need horizontal sweeps with current that changes proportionally faster toward the center, because the center of the screen is closer to the deflection yoke than the edges. A linear change in current would swing the beams at a constant rate angularly; this would cause horizontal compression toward the center.

ii) A **plasma display** panel (PDP) is a type of flat panel display common to large TV displays 30 inches (76 cm) or larger. They are called "plasma" displays because the technology utilizes small cells containing electrically charged ionized gases, or what are in essence chambers more commonly known as fluorescent lamps.

Advantages

- Picture quality
 - Capable of producing deeper blacks allowing for superior contrast ratio
 - Wider viewing angles than those of LCD; images do not suffer from degradation at high angles like LCDs
 - Less visible motion blur, thanks in large part to very high refresh rates and a faster response time, contributing to superior performance when displaying content with significant amounts of rapid motion (Though newer LCD screens have similar refresh rates, that also introduces the soap opera effect).

Disadvantages

- Picture quality
 - Earlier generation displays were more susceptible to screen burn-in and image retention, recent models have a pixel orbiter that moves the entire picture slower than is noticeable to the human eye, which reduces the effect of burn-in but does not prevent it.
 - Due to the bistable nature of the colour and intensity generating method, some people will notice that plasma displays have a shimmering or flickering effect with a number of hues, intensities and dither patterns.
 - Earlier generation displays (circa 2006 and prior) had phosphors that lost luminosity over time, resulting in gradual decline of absolute image brightness (newer models may be less susceptible to this, having advertised lifespans exceeding 100 000 hours, far longer than older CRT technology)
 - Screen-door effects are possible on screen sizes larger than 127 cm (50 in); the effect is more visible at shorter viewing distances.

Other

- Use more electrical power, on average, than an LCD TV.
- Does not work as well at high altitudes above 2 km due to pressure differential between the gases inside the screen and the air pressure at

altitude. It may cause a buzzing noise. Manufacturers rate their screens to indicate the altitude parameters.

- For those who wish to listen to AM radio, or are amateur radio operators (hams) or shortwave listeners (SWL), the radio frequency interference (RFI) from these devices can be irritating or disabling.

iii) A liquid crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly.

LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and signage. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones, and have replaced cathode ray tube (CRT) displays in most applications. They are available in a wider range of screen sizes than CRT and plasma displays, and since they do not use phosphors, they do not suffer image burn-in. LCDs are, however, susceptible to image persistence.[1]

The LCD screen is more energy efficient and can be disposed of more safely than a CRT. Its low electrical power consumption enables it to be used in battery-powered electronic equipment. It is an electronically modulated optical device made up of any number of segments filled with liquid crystals and arrayed in front of a light source (backlight) or reflector to produce images in color or monochrome. Liquid crystals were first discovered in 1888.[2] By 2008, worldwide sales of televisions with LCD screens exceeded annual sales of CRT units; the CRT became obsolete for most purposes.

b. Explain the following terms with examples:

(i) Interlacing

(ii) Aspect Ratio

Answer:

Interlacing (also known as interleaving) is a method of encoding a bitmap image such that a person who has partially received it sees a degraded copy of the entire image. When communicating over a slow communications link, this is often preferable to seeing a perfectly clear copy of one part of the image, as it helps the viewer decide more quickly whether to abort or continue the transmission.

Interlacing is supported by the following formats, where it is optional:

GIF interlacing stores the lines in the order 0, 8, 16, ... $(8n)$, 4, 12, ... $(8n+4)$, 2, 6, 10, 14, ... $(4n+2)$, 1, 3, 5, 7, 9, ... $(2n+1)$.

PNG uses the Adam7 algorithm, which interlaces in both the vertical and horizontal direction.

TGA uses two optional interlacing algorithms, two-way: 0, 2, 4, ... $(2n)$, 1, 3, ... $(2n+1)$ and four-way: 0, 4, 8, ... $(4n)$, 1, 5, ... $(4n+1)$, 2, 6, ... $(4n+2)$, 3, 7, ... $(4n+3)$.

JPEG, JPEG 2000, and JPEG XR (actually using a frequency decomposition hierarchy rather than interlacing of pixel values)

PGF (also using a frequency decomposition)

Interlacing is a form of incremental decoding, because the image can be loaded incrementally. Another form of incremental decoding is progressive scan. In progressive scan the loaded image is decoded line for line, so instead of becoming incrementally clearer it becomes incrementally larger. The main difference between the interlace concept in bitmaps and in video is that even progressive bitmaps can be loaded over multiple frames.

For example: Interlaced GIF is a GIF image that seems to arrive on your display like an image coming through a slowly-opening Venetian blind. A fuzzy outline of an image is gradually replaced by seven successive waves of bit streams that fill in the missing lines until the image arrives at its full resolution.

Interlaced graphics were once widely used in web design and before that in the distribution of graphics files over bulletin board systems and other low-speed communications methods. The practice is much less common today, as common broadband internet connections allow most images to be downloaded to the user's screen nearly instantaneously, and interlacing is usually an inefficient method of encoding images.[citation needed]

Interlacing has been criticized because it may not be clear to viewers when the image has finished rendering, unlike non-interlaced rendering, where progress is apparent (remaining data appears as blank)

The aspect ratio is the ratio of the width of a shape to its height when the width is larger than the height, i.e. the shape is positioned as a "landscape" rather than "portrait".

The aspect ratio is expressed as two numbers separated by a colon (x:y). A common misunderstanding is that x and y represent actual length and height. This is false; they actually represent the relation between width and height. As an example, 8:5, 16:10 and 1.6:1 are the same aspect ratio.

It may be applied to two characteristic dimensions of a three-dimensional shape.

Q.3 a. Explain Bresenham's Line Drawing Algorithm. Give a step wise solution to digitize the line with endpoints (5, 10) and (10, 20).

Answer:

- 1) Input two line end points and store the left endpoint in (X_0, Y_0)
- 2) Load (X_0, Y_0) in to the frame buffer
- ie) Plot in the first point
- 3) Calculate constants $\Delta x, \Delta y, 2\Delta y, 2\Delta y - 2\Delta x$ and obtain the starting value for the decision parameter as

$$P_0 = 2\Delta y - \Delta x$$
- 4) At each X_k along the line, starting at $k=0$ perform the following test
- ie) If $P_k < 0$ the next point to plot is X_{k+1}, Y_k and

$$P_{k+1} = P_k + 2\Delta y$$
 Otherwise, the next point to plot is
 (X_{k+1}, Y_{k+1}) and
$$P_{k+1} = P_k + 2\Delta y - 2\Delta x$$
- 5) Repeat step 4 Δx times

Example

$$X_1 = 5$$

$$Y_1 = 10$$

$$X_2 = 10$$

$$Y_2 = 20$$

$$\Delta x = 5$$

$$\Delta y = 10$$

$$P = 2\Delta y - \Delta x = 2 \times 10 - 5 = 20 - 5 = 15$$

Increments for the successive decision parameter is

$$2\Delta y = 2 \times 10 = 20$$

$$2\Delta y - 2\Delta x = 2 \times 10 - 2 \times 5 = 20 - 10 = 10$$

If $5 > 10$ (false)

```
X=5
Y=10
Xend =10
Setpixel(5,10)
While 5 < 10
X=6
If (15<0) then ( false)
P += 2dydx
Y=10+1 =11
P= 10 + 2 * (15-10) = 10+10=20
Setpixel ( 6, 11)
Etc
....
Procedure
Procedure linebres(xa , ya ,xb ,yb : integer)
Var
dx, dy, x, y, xEnd, p : integer ;
begin
dx := abs(xa - xb) ;
dy := abs(ya - yb) ;
p := 2* dy - dx ;
if xa > xb then
begin
x := xb ;
y := yb ;
xEnd := xa;
end
else
begin
x := xa ;
y := ya ;
xEnd := xb;
end
setpixel ( x, y, 1);
While x < xEnd do
Begin
x := x + 1 ;
if p < 0 then p := p + 2 * dy;
else
```

- b. Write a procedure of producing Sierpinski Gasket. Explain how to control the Sierpinski Gasket with the Mouse.

Answer: Page no 73 & 92

- Q.4 a. Write a procedure to implement the Cohen Sutherland line clipping algorithm to line clipping for any input pair of line endpoints.

Answer:

```

procedure CohenSutherlandLineClipAndDraw(
    x0,y0,x1,y1,xmin,xmax,ymin,ymax : real ; value:
integer);
{ Cohen-Sutherland clipping algorithm for line P0=(x1,y0)
to P1=(x1,y1)
and clip rectangle with diagonal from (xmin,ymin) to
(xmax,ymax).}
type
    edge = (LEFT,RIGHT,BOTTOM,TOP);
    outcode = set of edge;
var
    accept,done : boolean;
    outcode0,outcode1,outcodeOut : outcode;
    {Outcodes for P0,P1, and whichever point lies outside
the clip rectangle}
    x,y : real;
    procedure CompOutCode(x,y: real; var code:outcode);
    {Compute outcode for the point (x,y) }
    begin
        code := [];
        if y > ymax then code := [TOP]
        else if y < ymin then code := [BOTTOM];
        if x > xmax then code := code +[RIGHT]
        else if x < xmin then code := code +[LEFT]
    end;

begin
    accept := false; done := false;
    CompOutCode (x0,y0,outcode0); CompOutCode
(x1,y1,outcode1);
    repeat
        if(outcode0=[]) and (outcode1=[]) then {Trivial
accept and exit}
            begin accept := true; done:=true end
        else if (outcode0*outcode1) <> [] then
            done := true {Logical intersection is true, so
trivial reject and exit.}
    until done;
end;

```

```

    else
      {Failed both tests, so calculate the line segment
to clip;
      from an outside point to an intersection with clip
edge.}
      begin
        {At least one endpoint is outside the clip
rectangle; pick it.}
        if outcode0 <> [] then
          outcodeOut := outcode0 else outcodeOut :=
outcode1;
          {Now find intersection point;
use formulas  $y=y_0+slope*(x-x_0)$ ,  $x=x_0+(1/slope)*(y-
y_0)$ .}
          if TOP in outcodeOut then
            begin      {Divide line at top of clip
rectangle}
               $x := x_0 + (x_1 - x_0) * (y_{max} - y_0) / (y_1 -
y_0)$ ;
               $y := y_{max}$ 
            end
          if BOTTOM in outcodeOut then
            begin      {Divide line at bottom of clip
rectangle}
               $x := x_0 + (x_1 - x_0) * (y_{min} - y_0) / (y_1 -
y_0)$ ;
               $y := y_{max}$ 
            end
          else if RIGHT in outcodeOut then
            begin      {Divide line at right edge of clip
rectangle}
               $y := y_0 + (y_1 - y_0) * (x_{max} - x_0) / (x_1 -
x_0)$ ;
               $x := x_{max}$ 
            end
          else if LEFT in outcodeOut then
            begin      {Divide line at left edge of clip
rectangle}
               $y := y_0 + (y_1 - y_0) * (x_{min} - x_0) / (x_1 -
x_0)$ ;
               $x := x_{min}$ 
            end;
          {Now we move outside point to intersection point
to clip,
          and get ready for next pass.}
          if (outcodeOut = outcode0) then
            begin

```



```

        x0 := x; y0 := y; CompOutCode(x0,y0,outcode0)
    end
    else
    begin
        x1 := x; y1 := y;
    CompOutCode(x1,y1,outcode1);
    end
    end {subdivide}
until done;
    if accept then MidpointLineReal(x0,y0,x1,y1,value)
{Version for real coordin
ates}
end; {CohenSutherlandLineClipAndDraw

```

Q.5a. Show that a transformation matrix for a reflection about the line $y = x$, is equivalent to a reflection relative to the x - axis, followed by a counter clockwise rotation of 90

Answer:

Reflection about the y axis flips x coordinates while keeping y coordinates the same. The matrix for this transformation is

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The counterclockwise rotation of 90. is accomplished with the transformation matrix

$$\begin{bmatrix} \cos q & -\sin q & 0 \\ \sin q & \sin q & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For the translation sequence

- (1). Reflection relative to y axis
- (2). Counterclockwise rotation of 90.

The transformation matrix is obtained with the concatenation

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

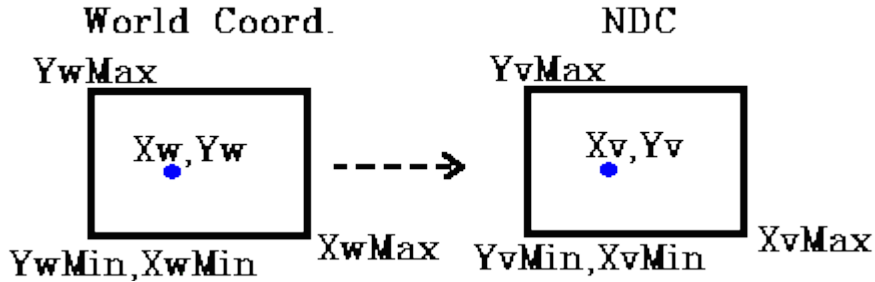
The resultant transformation matrix is equal to the that for the reflection about the line $y = -x$ which is:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

b. Give a brief description of the two dimensional viewing transformation pipeline. Construct the expressions for window to viewport coordinate transformations and explain.

Answer:

We want to map a point from WDC to Ndc, as shown below:



We can see from above that to maintain relative position we must have the following relationship:

$$\frac{X_w - X_{wmin}}{X_{wmax} - X_{wmin}} = \frac{X_v - X_{vmin}}{X_{vmax} - X_{vmin}}$$

$$\frac{Y_w - Y_{wmin}}{Y_{wmax} - Y_{wmin}} = \frac{Y_v - Y_{vmin}}{Y_{vmax} - Y_{vmin}}$$

We can rewrite above as

$$X_v = [(X_{vmax} - X_{vmin}) / (X_{wmax} - X_{wmin})] * (X_w - X_{wmin}) + X_{vmin}$$

$$= S_x * (X_w - X_{wmin}) + X_{vmin} = S_x * X_w + C_x$$

$$\text{where } S_x = (X_{vmax} - X_{vmin}) / (X_{wmax} - X_{wmin})$$

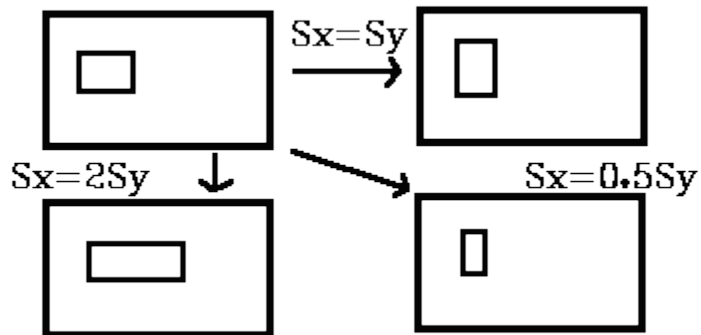
$$C_x = S_x * (- X_{wmin}) + X_{vmin}$$

Similarly for Y

$$Y_v = [(Y_{vmax} - Y_{vmin}) / (Y_{wmax} - Y_{wmin})] * (Y_w - Y_{wmin}) + Y_{vmin}$$

$$= S_y * (Y_w - Y_{wmin}) + Y_{vmin} = S_y * Y_w + C_y$$

Note that S_x, S_y are "scaling" factors. If $S_x = S_y$ the objects will retain same shape, else will be distorted, as shown in the example.



Q.6a. Write a procedure to perform 3-point perspective projection of an object.

Answer:

Three-point perspective is usually used for buildings seen from above (or below). In addition to the two vanishing points from before, one for each wall, there is now one for how those walls recede into the ground. This third vanishing point will be below the ground. Looking up at a tall building is another common example of the third vanishing point. This time the third vanishing point is high in space.

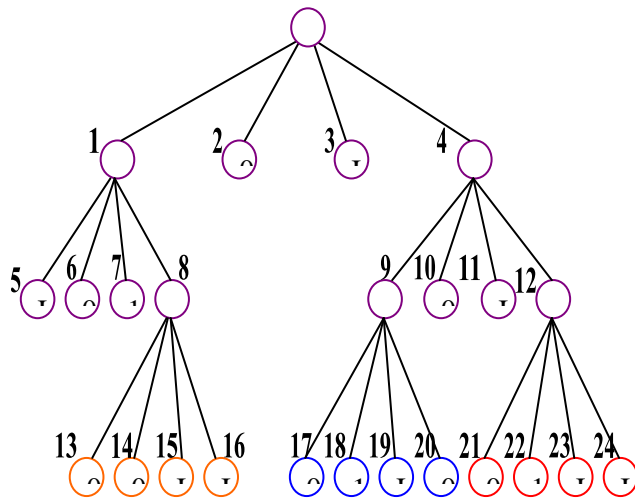
Three-point perspective exists when the perspective is a view of a Cartesian scene where the picture plane is not parallel to any of the scene's three axes. Each of the three vanishing points corresponds with one of the three axes of the scene. One-point, two-point, and three-point perspectives appear to embody different forms of calculated perspective. The methods required to generate these perspectives by hand are different. Mathematically, however, all three are identical: The difference is simply in the relative orientation of the rectilinear scene to the viewer.

b. Develop an algorithm for performing constructive solid geometry modelling using a primitive set of solids defined in octree structures.

Answer:

Divide each region into four quadrants

- Each quadrant is represented by a node.
- If the quadrant is inside the region - the node has a 1.
- If the quadrant is outside - the node has a 0.
- If you can not determine - subdivide further.



For OCTREES - Extend to 3D

- 8 VOXELS (or octants) for each CUBE / CUBOID

- Sub-divide upto the desired "resolution"
- Once an octree is generated - mathematical description is not available.
- Octree models can be used as primitives in Boolean operations.
- Union - Generate Octree for the combined space, and traverse trees.
- Similarly for Intersection and Difference

Q7 a. What is the underlying concept of the Painter's Algorithm for hidden surface removal? Identify two advantages and one disadvantage of the z-buffer algorithm compared to the Painter's algorithm.

Answer:

The idea behind the Painter's algorithm is to draw polygons far away from the eye first, followed by drawing those that are close to the eye. Hidden surfaces will be written over in the image as the surfaces that obscure them are drawn.

The concept is to map the objects of our scene from the world model to the screen somewhat like an artist creating an oil painting. First she paints the entire canvas with a background colour. Next, she adds the more distant objects such as mountains, fields, and trees. Finally, she creates the foreground with "near" objects to complete the painting. Our approach will be identical. First we sort the polygons according to their z-depth and then paint them to the screen, starting with the far faces and finishing with the near faces.

The algorithm initially sorts the faces in the object into back to front order. The faces are then scan converted in this order onto the screen. Thus a face near the front will obscure a face at the back by overwriting it at any points where their projections overlap. This accomplishes hidden-surface removal without any complex intersection calculations between the two projected faces.

The algorithm is a hybrid algorithm in that it sorts in object space and does the final rendering in image space.

The basic algorithm :

1. Sort all polygons in ascending order of maximum z-values.
2. Resolve any ambiguities in this ordering.
3. Scan convert each polygon in the order generated by steps (1) and (2).

The necessity for step (2) can be seen in the simple case shown in Figure [9.2](#).

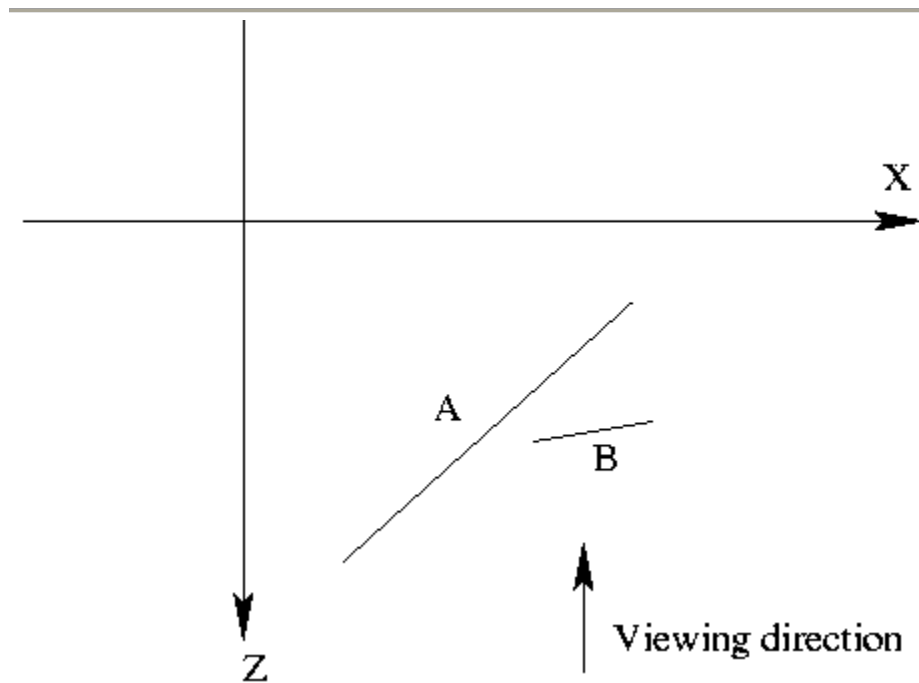


Figure 9.2: Ambiguities in the Painter's algorithm.

B precedes A in order of maximum z but A should precede B in writing order.

At step (2) the ordering produced by (1) must be confirmed. This is done by making more precise comparisons between polygons whose z-extents overlap.

Assume that polygon P is currently at the head of the sorted list, before scan converting it to the frame-buffer it is tested against each polygon Q whose z-extent overlaps that of P. The following tests of increasing complexity are then carried out:

1. If the x-extents of P and Q do not overlap then the polygons do not overlap, hence their ordering is immaterial.
2. If the y-extents of P and Q do not overlap then the polygons do not overlap, hence their ordering is immaterial.
3. If P is wholly on the far away side of Q then P is written before Q.
4. If Q is wholly on the viewing side of P then P is written before Q.
5. If the projections of P and Q do not overlap then the order P and Q in the list is immaterial.

If any of these tests are satisfied then the ordering of P and Q need not be changed. However if all five tests fail then it is assumed that P obscures Q and they are interchanged in the list. To avoid looping in the case where P and Q inter-penetrate Q

must be marked as having been moved in the list.

When polygon Q which has been marked fails all tests again when tested against P then it must be split into two polygons and each of these polygons treated separately. Q is split in the plane of P.

Often the last test will not be done because it can be very complex for general polygons.

The easiest way to achieve hidden-surface removal is to use the depth buffer (sometimes called a z-buffer). A depth buffer works by associating a depth, or distance from the viewpoint, with each pixel on the window. Initially, the depth values for all pixels are set to the largest possible distance, and then the objects in the scene are drawn in any order.

Graphical calculations in hardware or software convert each surface that's drawn to a set of pixels on the window where the surface will appear if it isn't obscured by something else. In addition, the distance from the eye is computed. With depth buffering enabled, before each pixel is drawn, a comparison is done with the depth value already stored at the pixel.

If the new pixel is closer to the eye than what's there, the new pixel's colour and depth values replace those that are currently written into the pixel. If the new pixel's depth is greater than what's currently there, the new pixel would be obscured, and the colour and depth information for the incoming pixel is discarded.

Since information is discarded rather than used for drawing, hidden-surface removal can increase your performance.

b. Write a brief note on Specular Reflection.

Answer: Page no 405 of text book

Q8 b. Using diagrams to illustrate, explain the principle of each of the following anti-aliasing techniques.

(i) Area sampling

(ii) Super-sampling

Answer:

Display primitives generated by the raster algorithms have a jagged, or stairstep appearance because the sampling inter pixel positions. The distortion of information due to low frequency sampling (undersampling) is called aliasing.

A straightforward antialiasing method is to increase sampling rate by treating the screen as if it were covered with a finer grid than is actually available. Then multiple sample points can be used across this finer grid to determine an appropriate intensity level for each screen pixel. This technique of sampling object characteristic at a high resolution and displaying the result at a lower resolution is called supersampling (or postfiltering, since the general method involves computing intensities at subpixel grid

position, then combining the results to obtain the pixel intensities). Displayed pixel positions are spots of light covering a finite area of the screen, and not infinitesimal mathematical points. In the line and fill-area algorithms the intensity of each pixel is determined by the location of a single point on the object boundary. By supersampling, intensity information is obtained from multiple points that contribute to the overall intensity of a pixel.

An alternative to supersampling is to determine pixel intensity by calculating the areas of overlaps of each pixel with the object to be displayed. Antialiasing by computing overlap area is referred to as area sampling (or prefiltering since the intensity of the pixel as a whole is determined without calculating subpixel intensities). Pixel overlap areas are obtained by determining where object boundaries intersect individual pixel boundaries.

Raster objects can also be antialiased by shifting the displayed location of pixel areas. This technique, called pixel phasing, is applied by micropositioning the electron beam in relation to object geometry.

Q9 a Explain the characteristics of a Bezier curve. Describe the concept of degree elevation and give its expression. Also explain the variation diminishing property with an example.

Answer:

Bezier curves are used in computer graphics to produce curves which appear reasonably smooth at all scales (as opposed to polygonal lines, which will not scale nicely). Mathematically, they are a special case of cubic Hermite interpolation (whereas polygonal lines use linear interpolation). What this means is that curves are constructed as a sequence of cubic segments, rather than linear ones. But whereas Hermite interpolating polynomials are constructed in terms of derivatives at endpoints, Bezier curves use a construction due to Sergei Bernstein, in which the interpolating polynomials depend on certain control points. The mathematics of these curves is classical, but it was a French automobile engineer Pierre Bezier who introduced their use in computer graphics. Roughly speaking, to each set of four points P_0, P_1, P_2, P_3 we associate a curve with the following properties:

- * It starts at p_0 and ends at p_3 .
- * When it starts from p_0 it heads directly towards p_1 , and when it arrives at p_3 it is coming from the direction of p_2 .
- * The entire curve is contained in the quadrilateral whose corners are the four given points (their **convex hull**).

One reason these curves are used so much in computer graphics is that they are very efficient to construct, since a simple recursion process means that the basic arithmetic operation needed to build the points along one is just division by two. For this reason, also, the most efficient implementations use **scaled integers** instead of floating point numbers as basic numerical data.

b. Give a short note on Bezier surface. Give a procedure to construct a Bezier surface. Give example.

Answer:

Bézier surfaces are a species of mathematical spline used in computer graphics, computer-aided design, and finite element modeling. As with the Bézier curve, a Bézier surface is defined by a set of control points. Similar to interpolation in many respects, a key difference is that the surface does not, in general, pass through the central control points; rather, it is "stretched" toward them as though each were an attractive force. They are visually intuitive, and for many applications, mathematically convenient.

A given Bézier surface of order (n, m) is defined by a set of $(n + 1)(m + 1)$ control points $k_{i,j}$. It maps the unit square into a smooth-continuous surface embedded within a space of the same dimensionality as $\{ k_{i,j} \}$. For example, if k are all points in a four-dimensional space, then the surface will be within a four-dimensional space.

A two-dimensional Bézier surface can be defined as a parametric surface where the position of a point p as a function of the parametric coordinates u, v is given by:

$$\mathbf{p}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) \mathbf{k}_{i,j}$$

evaluated over the [unit square](#), where

$$B_i^n(u) = \binom{n}{i} u^i (1 - u)^{n-i}$$

is a [Bernstein polynomial](#), and

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

is the [binomial coefficient](#).

Some properties of Bézier surfaces:

A Bézier surface will transform in the same way as its control points under all linear transformations and translations.

All $u = \text{constant}$ and $v = \text{constant}$ lines in the (u, v) space, and, in particular, all four edges of the deformed (u, v) unit square are Bézier curves.

A Bézier surface will lie completely within the convex hull of its control points, and therefore also completely within the bounding box of its control points in any given Cartesian coordinate system.

The points in the patch corresponding to the corners of the deformed unit square coincide with four of the control points.

However, a Bézier surface does not generally pass through its other control points.

Generally, the most common use of Bézier surfaces is as nets of bicubic patches (where $m = n = 3$). The geometry of a single bicubic patch is thus completely defined by a set of 16 control points. These are typically linked up to form a B-spline surface in a similar way as Bézier curves are linked up to form a B-spline curve.

Simpler Bézier surfaces are formed from biquadratic patches ($m = n = 2$), or Bézier triangles.

Bézier surfaces in computer graphics
Ed Catmull's "Gumbo" model, composed from patches

Bézier patch meshes are superior to meshes of triangles as a representation of smooth surfaces, since they are much more compact, easier to manipulate, and have much better continuity properties. In addition, other common parametric surfaces such as spheres and cylinders can be well approximated by relatively small numbers of cubic Bézier patches.

However, Bézier patch meshes are difficult to render directly. One problem with Bézier patches is that calculating their intersections with lines is difficult, making them awkward for pure ray tracing or other direct geometric techniques which do not use subdivision or successive approximation techniques. They are also difficult to combine directly with perspective projection algorithms.

For this reason, Bézier patch meshes are in general eventually decomposed into meshes of flat triangles by 3D rendering pipelines. In high-quality rendering, the subdivision is adjusted to be so fine that the individual triangle boundaries cannot be seen. To avoid a "blobby" look, fine detail is usually applied to Bézier surfaces at this stage using texture maps, bump maps and other pixel shader techniques.

A Bézier patch of degree (m, n) may be constructed out of two Bézier triangles of degree $m+n$, or out of a single Bézier triangle of degree $m + n$, with the input domain as a square instead of as a triangle.

A Bézier triangle of degree m may also be constructed out of a Bézier surface of degree (m, m) , with the control points so that one edge is squashed to a point, or with the input domain as a triangle instead of as a square.

Text Book

**Computer Graphics using GL, F.S. Hill, Jr., II Edition, PHI/Person
Education, 2005**