COURSE CODE :- MCS-053 NAME- AKANKSHA SAXENA

COURSE TITLE :- COMPUTER GRAPHICS AND MULTIMEDIA

ASSIGNMENT NUMBER :- MCA (V)/053/Assignment/15-16

Ques 1

Solution : (i) Painting and Drawing

Drawing - is a software application means using tools that create "objects", such as squares, circles, lines or text, which the program treats as discrete units. If you draw a square in PowerPoint, for example, you can click anywhere on the square and move it around or resize it. It's an object, just like typing the letter "e" in a word processor. i.e a drawing program allows a user to position standard shape which can be edited by translation, rotation and scaling operations on these shapes.

Painting - functions, on the other hand don't create objects. If you look at a computer screen, you'll see that it's made up of millions of tiny dots called pixels. You'll see the same thing in a simpler form if you look at colour comics in the Sunday newspaper - lots of dots of different colour ink that form a picture. Unlike a drawing function, a paint different colour ink that form a picture. Unlike a drawing function, a paint function changes the colour ~~ink that~~ of individual pixels based on the tools you choose.

(i) Raster Scan and Random Scan

In a Random scan system, the Display buffer stores the picture information. Further, the device is capable of producing pictures made up of lines but not of curves. Thus, it is also known as "Vector display device or line display device or Calligraphic display device".

In a raster scan, the frame stores the picture information which is the plane (with m rows and n columns)

(ii) Simulation and animation

1 Computer simulation is discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer and analysing the execution output. Simulation model on a digital computer, and analysis the execution output. Simulation embodies the principle of "learning by doing" - to learn about the system we must first first build a model of some sort and the operate the model

2 Animation is a time based phenomenon for imparting visual changes in any scene according to any time sequence, the visual changes could be incorporated through translation of object, scaling of object, or change in colour, transparency, surface texture etc. whereas Graphics does not contain dimension of time

Graphics + Dimension of time = Animation

(iv) Visualization and Image processing

It is difficult for the human brain to make sense out of the large ~~amount~~ volume of numbers produced by a scientific computation. Numerical and statistical methods

methods are useful for solving the problem. Visualization techniques are another approach for interpreting large data sets, providing insights that might be missed by statistical methods. The picture they provide are vehicle for thinking about data.

Modern digital technology has made it possible for the manipulation of multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The analysis and manipulation of a digitized image, especially in order to improve its quality.

(6) DDA (Digital Differential Analyser) Algorithm

Standard algorithms are available to determine which pixels provide the best approximation to the desired line, one such algorithm is DDA algorithm.

Advantages

1. It is the simplest algorithm and it does not require special skills for implementation.
2. It is faster method for calculating pixel positions than the direct use of equation $y = mx + b$. It eliminates the multiplication in the equation by making use of raster characteristics, so that appropriate increments are applied in x or y direction to find the pixel positions along the line path.

Disadvantages

1. Division logic is needed which switches it towards hardware logic.
2. Floor integer values are used in place of normal integer values which may different result.
3. Floating point arithmetic is needed thus it is time consuming.

Line segment with end points (2,3) and (9,8)

Sol We know the general equation of line is given by

$y = mx + c$ where $m = (y_1 - y_0/(x_1 - x_0))$

given $(x_0, y_0) = (2,3)$; $(x_1, y_1) = (9,8)$

$\Rightarrow m = \frac{y_1 - y_0}{x_1 - x_0} = \frac{8-3}{9-2} = \frac{5}{7}$ i.e. $0 < m < 1$

$\Rightarrow c = y_1 - mx_1 = 8 - \frac{5}{7} \times 9 = \frac{56-45}{7} = \frac{11}{7}$

So by equation of line $(y = mx + c)$ we have

$y = \frac{5}{7}x + \frac{11}{7}$

DDA Algorithm for two case:	
Case 1: $m < 1$	$x_i + 1 = x_i + 1$
	$y_i + 1 = y_i + m$
Case 2: $m > 1$	$x_i + 1 = x_i + (1/m)$
	$y_i + 1 = y_i + 1$

As $0 < m < 1$ so according to DDA algorithm case 1

$x_i + 1 = x_i + 1$ $\quad$ $y_i + 1 = y_i + m$

given $(x_0, y_0) = (2,3)$

1) $x_1 = x_0 + 1 = 3$

$y_1 = y_0 + m = 3 + \frac{5}{7} = \frac{21+5}{7} = \frac{26}{7} = 3\frac{5}{7}$

put pixel (x_0, round y, colour)

i.e. put on 3, 5

2) $x_2 = x_1 + 1 = 3 + 1 = 4$

$y_2 = y_1 + m = \left(\frac{26}{7}\right) + \frac{5}{7} = \frac{31}{7} = 4\frac{3}{7}$

put on (4,3)

3) $x_3 = x_2 + 1 = 4 + 1 = 5$

$y_3 = y_2 + m = \frac{31}{7} + \frac{5}{7} = \frac{36}{7} = 5\frac{1}{7}$ put on (5,1) //ly go on till (9,8) is reached.

(c) Sutherland - Hodgman Algorithm

When we try to clip the polygon under consideration with any rectangle window, then, we observe that the coordinate of the polygon vertices satisfies one of the four cases listed in the table, and further it is to be noted that this procedure of clipping can be simplified by clipping the polygon edgewise and not the polygon as a whole. This decomposes the bigger problem into a set of subproblems, which can be handled separately as per the cases listed. Actually this table describes the cases listed.

Cases	V_i	V_{i+1}	Output Vertex
A	Inside	Inside	V_{i+1}
B	Inside	Outside	V'_i i.e. intersect of polygon & window edge
C	Outside	Outside	None
D	outside	Inside	V'_i ; V_{i+1}

pseudocode for clipping polygon into a window

Define Variable

int VertexArray is the array of input polygon vertices
outVertexArray is the array of output polygon vertices
Nin is the number of entries in inVertexArray
Nout is the number of entries in outVertexArray
n is the number of entries in outVertexArray
n is the number of edges of the clip polygon
ClipEdge[x] is the x^{th} edge of clip polygon defined by a pair of vertices
s, p are the start and end point respectively of current polygon edge

i is the intersection point with a clip boundary
j is the vertex loop counter

Define functions

AddNewVertex(newVertex, Nout, outVertexArray)
: Adds new vertex to outVertexArray and then updates Nout.

InsideTest(testVertex, clipEdge[x])
: check whether the vertex lies inside the clip edge or not; return TRUE if inside else FALSE

Intersect(first, second, clipEdge[x])
: clip polygon edge (first, second) against clipEdge[x], outputs the intersection point.

: begin main

{ x = 1
while(x ≤ n) : Loop through all n clip edges
{ : Flush the out Vertex Array
Nout = 0
s = inVertexArray [Nin] : Start with last vertex in VertexArray
for j = 1 to Nin do : loop through Nin number of polygon vertices (edges)
{ p = inVertexArray[j]
if InsideTest(p, clipEdge[x]) = = TRUE then : Case A and D
if InsideTest(s, clipEdge[x]) = = TRUE then : Case A
AddNewVertex(p, Nout, outVertexArray)
else
i = Intersect(s, p, clipEdge[x]) : Case D
AddNewVertex(i, Nout, outVertexArray)
AddNewVertex(p, Nout, outVertexArray)
end if
else : ie if InsideTest(p, clipEdge[x]) = = FALSE (Case 2 and 3)
if InsideTest(s, clipEdge[x] = = TRUE then : Case B
{

Insects (s, p, clipEdge[x])
Add New Vertex (i, Nout, outVertexArray)
end if — : No action for case C
s = p — : Advance to next pair of vertices
j = j + 1
end if — : end {for}
}
x = x+1 — : Proceed to the next ClipEdge[x+1]
Nin = Nout
inVertexArray = outVertexArray — : The output vertex array for the current clip edge becomes the input vertex array for the next clip ed
} : end while
}: end main

Ques 2

(a) Homogeneous Coordinate systems

Let P(x,y) by any point in 2-D Euclidean (Cartesian) system.
In Homogeneous Coordinate system, we add a third coordinate to a point. Instead of (x,y), each point is represented by a triple (x,y,H) such that $H \neq 0$; with the condition that

$(x1, y1, H1) = (x2, y2, H2) \leftrightarrow x1/H1 = x2/H2 \; ; \; y1/H1 = y2/H2$.

Here if we take H=0, then we have point at infinity ie generation of horizons.
Thus (2,3,6) and (4,6,12) are the same points are represented by different coordinate triples ie each point has many different Homogeneous coordinate representation.

2-D Euclidian System	Homogeneous Coordinate System
Any point (x,y) ⟶	(x,y,1)
	If (x,y,H) be any point in HCS (such that $H \neq 0$); Then $(x,y,H) = (x/H, y/H, 1)$
(x/H, y/H) ⟵	(x,y,H)

Triangle ABC coordinates $A(0,0)$, $B(5,8)$, $C(4,2)$, clockwise rotation 45° about the origin, then translate in direction of vector $(1,0)$

$$[ABC] = \begin{bmatrix} 0 & 0 & 1 \\ 5 & 8 & 1 \\ 4 & 2 & 1 \end{bmatrix}$$

Suppose the rotation is made in counter clockwise direction. Then, the transformation matrix for rotation, R_{45° in terms of homogeneous system is given by:

$$R_{45^\circ} = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ & 0 \\ -\sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the Transformation matrix, T_v, where $v = 1I + 0J$ is:

$$T_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

where t_x and t_y is the translation factors in the x and y directions respectively.

i) Now rotation followed by translation can be computed as:

$$R_{45} \cdot T_v = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

So the new coordinates $A'B'C'$ of a given triangle ABC can be found as: $[A'B'C'] = [ABC] \cdot R_{45^\circ} \cdot T_v$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 5 & 8 & 1 \\ 4 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1 & 0 & 1 \end{bmatrix} = [illegible]$$

$$= \begin{bmatrix} 1 & 0 & 1 \\ -3/\sqrt{2}+1 & 13/\sqrt{2} & 1 \\ 2/\sqrt{2}+1 & 6/\sqrt{2} & 1 \end{bmatrix} \quad \text{— I}$$

implies that the given triangle $A(0,0)$, $B(5,8)$, $C(4,2)$ be transformed into

$A'(1,0)$ $B'\left(-\frac{3}{\sqrt{2}}+1, \frac{13}{\sqrt{2}}\right)$ and $C'\left(\frac{2}{\sqrt{2}}+1, \frac{6}{\sqrt{2}}\right)$, respectively as

Similarly, we can obtain the translation followed by rotation transformation as:

$$T_v . R_{45°} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 1 \end{bmatrix}$$

And hence, the new coordinates $A'B'C'$ can be computed as

$$[A'B'C'] = [ABC] . T_v R_{45°}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 5 & 8 & 1 \\ 4 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1 \\ -2/\sqrt{2} & 14/\sqrt{2} & 1 \\ 3/\sqrt{2} & 7/\sqrt{2} & 1 \end{bmatrix} \quad \text{— (II)}$$

Thus in this case, the given triangle $A(0,0)$, $B(5,8)$ & $C(4,2)$ are transformed into $A''(1/\sqrt{2}, 1/\sqrt{2})$, $B''(-2/\sqrt{2}, 14/\sqrt{2})$ $C''(3/\sqrt{2}, 7/\sqrt{2})$

By (I) and (II), we see that the two transformation do not commute

(b) 4 vertices of polygon $A(0,0)$, $B(3,0)$, $C(3,3)$, $D(0,3)$

(i) translate 2 units along x-axis

We can present the given polygon using homogeneous coordinates of vertices as

$$\begin{array}{c} A \\ B \\ C \\ D \end{array} \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix}$$

Translation factor $t_x = 2$ & let $t_y = 3$.

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

The transformation matrix for translation

$$T_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{bmatrix}$$

New object point coordinates are:

$$[A'B'C'D'] = [ABCD] \cdot T_v$$

$$\begin{array}{c} A' \\ B' \\ C' \\ D' \end{array} \begin{bmatrix} x_1' & y_1' & 1 \\ x_2' & y_2' & 1 \\ x_3' & y_3' & 1 \\ x_4' & y_4' & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \\ 5 & 6 & 1 \\ 5 & 9 & 1 \\ 2 & 5 & 1 \end{bmatrix}$$

Thus $A'(x_1', y_1') = (2,3)$

$B(x_2', y_2') = (5,6)$

$C(x_3, y_3) = (5,9)$

$B(x_4, y_4) = (2,5)$

(ii) xy shear about the origin.

Let $a=2$ and $b=3$.

We represent polygon ABCD, in matrix form, using homogeneous coordinates of vertices as

$$\begin{bmatrix} A & 0 & 0 & 1 \\ B & 3 & 0 & 1 \\ C & 3 & 3 & 1 \\ D & 0 & 3 & 1 \end{bmatrix}$$

The matrix of x-shear is:

$$Sh_x(a) = \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So the new coordinates A'B'C'D' of the x-sheared object ABCD can be found as: $[A'B'C'D'] = [ABCD] \cdot Sh_x(a)$

$$[A'B'C'D'] = \begin{bmatrix} A & 0 & 0 & 1 \\ B & 3 & 0 & 1 \\ C & 3 & 3 & 1 \\ D & 0 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 6 & 1 \\ 9 & 9 & 1 \\ 6 & 3 & 1 \end{bmatrix}$$

Thus $A' = (0,0)$, $B' = (3,6)$, $C' = (9,9)$, $D' = (6,3)$

(b) Similarly the effect of shearing in the y direction can be found as $[A'B'C'D] = [A\ B\ C\ D] \cdot Sh_y(b)$

$$[A'B'C'D'] = \begin{bmatrix} A & 0 & 0 & 1 \\ B & 3 & 0 & 1 \\ C & 3 & 3 & 1 \\ D & 0 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 9 & 1 \\ 9 & 12 & 1 \\ 6 & 3 & 1 \end{bmatrix}$$

Thus, $A' = (0,0)$, $B' = (3,9)$, $C' = (9,12)$, $D' = (6,3)$

(c) Finally the effect of shearing in both direction is $[A'B'C'D'] = [A, B, C, D] \cdot Sh_{xy}(a,b)$

$$[A'B'C'D'] = \begin{bmatrix} A & 0 & 0 & 1 \\ B & 3 & 0 & 1 \\ C & 3 & 3 & 1 \\ D & 0 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 9 & 1 \\ 9 & 12 & 1 \\ 6 & 3 & 1 \end{bmatrix}$$

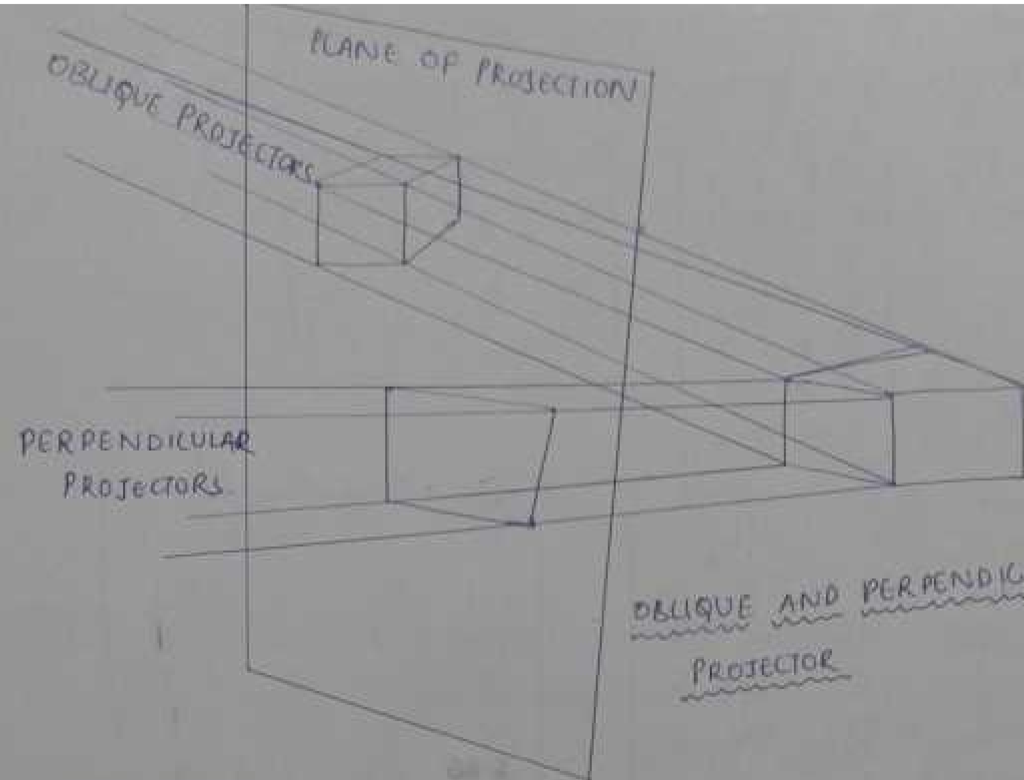
Thus $A' = (0,0)$ $B' = (3,9)$, $C' = (9,12)$, $D' = (6,3)$

(c) Projection

A transformation which maps 3-D objects onto 2-D screen, we call it projections. Here, 2-D screen is known as Plane of projection or view plane, which constitutes the display surface. The mapping is determined by projection rays called the projectors, Geometric projections of objects are formed by the intersection of lines with a plane called plane of projection / view plane. Projectors are lines from an arbitrary point, called centre of projection (COP), through each point in an object

We have two types of projections namely Perspective projection and parallel projection. This categorisation is based on the fact whether rays coming from the object converge at the centre of projection, then this projection is known as perspective projection, otherwise it is parallel projection. In the case of parallel projection the rays from an object converge at infinity unlike perspective projection where the rays from an object converge at a finite distance (called COP).

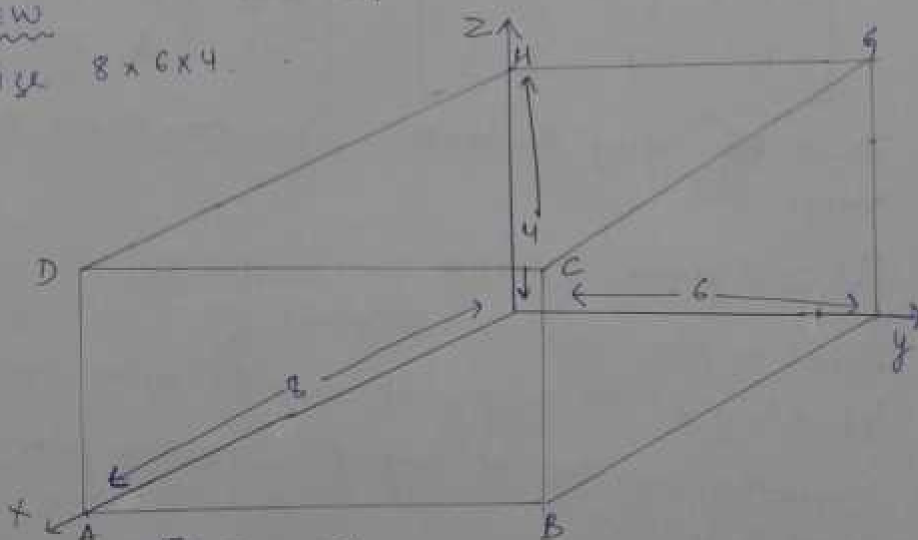
Parallel projection is further categorised into Orthographic and Oblique projection. Parallel projection can be categorized according to the angle that the direction of projection makes with the projection plane if the direction of projection of rays is perpendicular to the projection plane then this parallel projection is known as orthographic projection and if the direction of projection of rays is not perpendicular to the projection plane then this parallel projection is known as Oblique projection. The orthographic projection shows only the front face of the given object, which includes only two dimensions: length and width. The oblique projection, on the other hand, shows the front surface and the top surface which includes three dimensions: length, width and height. Therefore, an oblique projection is one way to show all three dimension of an object in a single view.



OBLIQUE AND PERPENDICULAR PROJECTOR

ISOMETRIC VIEW

of cuboid size $8 \times 6 \times 4$



$$V = [A\,B\,C\,D\,E\,F\,G\,H] = \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{bmatrix} 8 & 0 & 0 & 1 \\ 8 & 6 & 0 & 1 \\ 8 & 6 & 4 & 1 \\ 8 & 6 & 4 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 6 & 0 & 1 \\ 0 & 6 & 4 & 1 \\ 0 & 0 & 4 & 1 \end{bmatrix}$$

To draw an Isometric projection, we find the image coordinate of a given cuboid as follows:

$$P' = V \cdot P_{iso} = \begin{bmatrix} 8 & 0 & 0 & 1 \\ 8 & 6 & 0 & 1 \\ 8 & 6 & 4 & 1 \\ 8 & 0 & 4 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 6 & 0 & 1 \\ 0 & 6 & 4 & 1 \\ 0 & 0 & 4 & 1 \end{bmatrix} \cdot \begin{bmatrix} -2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} =$$

$$\Rightarrow \begin{bmatrix} -6 & -8 & -8 & 3 \\ -22 & 4 & -14 & 3 \\ -26 & 0 & -6 & 3 \\ -20 & -12 & 0 & 3 \\ 0 & 0 & 0 & 3 \\ -6 & 12 & -6 & 3 \\ -10 & 8 & 2 & 3 \\ -4 & -4 & 8 & 3 \end{bmatrix} = \begin{bmatrix} -2 & -2.66 & -2.66 & 1 \\ -7.33 & 1.33 & -4.66 & 1 \\ -8.66 & 0 & -2 & 1 \\ -6.66 & -4 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ -2 & 4 & -2 & 1 \\ -3.33 & 2.66 & 0.66 & 1 \\ -1.33 & -1.33 & 2.66 & 1 \end{bmatrix}$$

Thus by using the matrix, we can draw an isometric view.

(d) $z=2$ plane, projection point is $(0,0,10)$

Plane of projection : $z = 2$ (given)

Let $P(x,y,z)$ be any point in the space. We know the parametric equation of line AB, starting from A and passing through B is

$P(t) = A + t.(B-A),\ 0 < t < \infty$

So that parametric equation of a line starting from $E(0,0,10)$ and passing through $P(x,y,z)$ is :

$E + t(P-E),\ 0 < t < \infty$

$= (0,0,10) + t[(x,y,z) - (0,0,10)]$

$= (0,0,10) + [t.x, t.y, t.(z-10)]$

$= [t.x + t.y + t.(z-10) + 10]$ Assume

point p' is obtained when $t = t^*$

$\therefore P' = (x', y', z') = [t^*x, t^*y, t^*(z-10)+10]$

Since, P' lies on $z=2$ plane, so

$t^*(z-10)+10 = 2$ must be true; $t^* = \frac{-8}{z-10}$

$$P' = (x', y', z') = \left[\frac{-8x}{z-10}, \frac{-8y}{z-10}, 2\right]$$

$$= \left[\frac{-8x}{z-10}, \frac{-8y}{z-10}, \frac{2z-20}{z-10}\right]$$

In homogeneous coordinate system

$$P' = (x', y', z', 1) = \left[\frac{-8x}{z-10}, \frac{-8y}{z-10}, \frac{2z-20}{z-10}, 1\right] = (-8x, -8y, 2z-20, z-10) \quad \text{— (1)}$$

In matrix form

$$(x', y', z', 1) = (x, y, z, 1) = \begin{bmatrix} -8 & 0 & 0 & 0 \\ 0 & -8 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & -2 & -10 \end{bmatrix} \quad \text{— (2)}$$

Thus, equation (2) is the required perspective transformation from (0, 0, 10) view point.

Vanishing point

The vanishing point is that point at which parallel lines appear to converge and vanish. A practical example is a long straight railroad track

Ques 3

(a) Bezier Curves

Bezier curves are used in computer graphics to produce curves which appear reasonably smooth at all scales. This spline approximation method was developed by French engineer Pierre Bezier for automobile body design. Bezier spline was designed in such a manner that they are useful and convinent for curve and surface design, and are easy to implement. Curves are trajectories of moving points. We will specify them as functions assigning a location of that moving point (in 2D or 3D) to a parameter i.e parametric curves.

Curves are useful in geometric modeling and they should have a shape which has a clear and intuitive relation to the path of the sequence of control points. One family of curves satisfying this requirement are Bezier curve.

The Bezier curve require only two end points and other points that control the endpoint tangent vector.

$P_0(0,0)$, $P_1(2,5)$, $P_2(5,9)$, $P_3(10,20)$

We know the cubic Bezier curve is

$$P(u) = \sum_{i=0}^{3} p_i B_{3,i}(u)$$

$$\Rightarrow P(u) = p_0(1-u)^3 + 3p_1 u(1-u)^2 + 3p_2 u^2(1-u) + p_3 u^3$$

$$P(u) = (0,0)(1-u)^3 + 3(2,5)u(1-u)^2 + 3(5,9)u^2(1-u) + (10,20)u^3$$

We choose different values of u from 0 to 1.

$$u = 0.2 : P(0) = (0,0)(1-0.2)^3 + 3(2,5)(0.2)(1-0.2)^2 + 3(5,9)(0.2)^2 (1-0.2) + (10,2)(0.2)^3.$$

$= (0,0)(0.2)^3 + (2,5)(2.4) + (5,9)(2.4) + (10,2)(0.8)$

$= (0,0) + (4.8, 12.0) + (12.0, 21.6) + (8, 1.6)$

$= (24.8, 35.2) = P(0.2)$

$u = 0.4$: $P(0.4) = (0,0)(1-0.4)^3 + 3(2,5)(0.4)(1-0.4)^2 + 3(5,9)(0.4)^2(1-0.4) + (10,2)(0.4)^3$

$= (0,0)(0.4)^3 + (2,5)(0.192) + (5,9)(0.192) + (10,2)(0.64)$

$= (0,0) + (0.384, 0.96) + (0.96, 1.728) + (0.64, 0.128)$

$P(0.4) = (1.984, 2.816)$

$u = 0.6$: $P(0.6) = (0,0)(1-0.6)^3 + 3(2,5)(0.6)(1-0.6)^2 + 3(5,9)(0.6)^2(1-0.6) + (10,2)(0.6)^3$

$= (0,0)(0.6)^3 + (2,5)(0.648) + (5,9)(0.648) + (10,2)(0.28)$

$= (0,0) + (1.296, 3.24) + (3.24, 5.832) + (2.16, 0.432)$

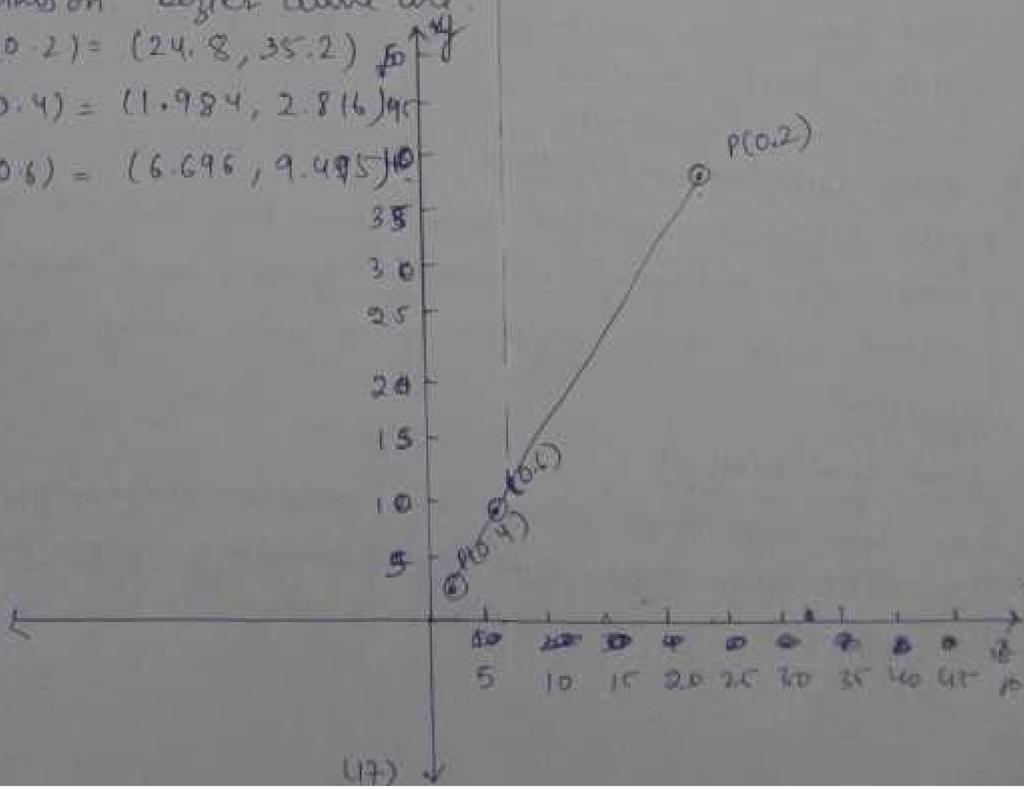
$P(0.6) = (6.696, 9.495)$

Three points on bezier curve are

$P(0.2) = (24.8, 35.2)$

$P(0.4) = (1.984, 2.816)$

$P(0.6) = (6.696, 9.495)$



(b) Visible Surface-Detection

Visibility tests try to identify the visible surface or visible edges that are visible from a given view point. Visibility tests are performed by making uses of either
(i) Object-space or (ii) image-space (iii) both object-space and image spaces

Object-space approaches use the directions of a surface normal wrt a viewing direction to detect a back face. Image-space approaches utilize two buffers: one for storing the pixel intensities and another for updating the depth of the visible surfaces from the view plane. A method, which uses both object-space and image-space, utilize depth of sorting of surfaces. The methods in this category also use image-space for conducting visibility tests. While making visibility tests, ~~coherenco~~ coherency property is utilized to make the method very fast.
There are three methods of detecting visible surfaces
- Depth-buffer method
- Scan-line method
- Area subdivision method

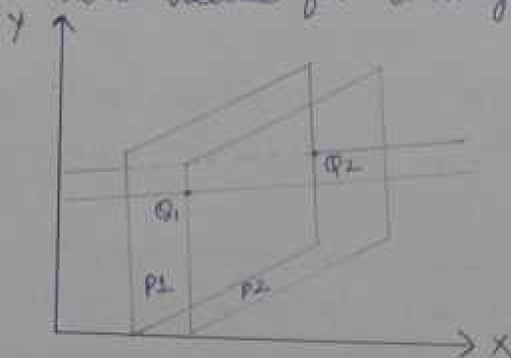
Depth-buffer method and Scan-line method come under the category of image-space, and area-subdivision method uses both object-space and image-space approach.

Scan-line Method

Scan-line algorithm solves the hidden-surface problem, one scan-line at a time, usually processing scan lines from the bottom to the top of the display.

Assumptions : 1. Plane of projection is $z=0$ plane.
2. Orthographic parallel projection.
3. Direction of projection, $d = (0,0,-1)$
4. Objects made up of polygon faces.

The scan-line algorithm is a one-dimensional version of the depth-Buffer. We require two arrays, intensity [x] & depth [x] to hold values for a single scan-line.



Here at Q_1 and Q_2 both polygons are active (i.e. sharing). Compare the z-values at Q_1 for both the planes (P_1 & P_2). Let $z_1^{(1)}$, $z_1^{(2)}$ be the z-value at Q_1, corresponding to P_1 & P_2 polygon respectively.

Similarly

$z_2^{(1)}$, $z_2^{(2)}$ are the z-values at Q_2, corresponding to P_1 & P_2 polygon respectively.

Case 1 : $\left.\begin{array}{l} z_1^{(1)} < z_1^{(2)} \\ z_2^{(1)} < z_2^{(2)} \end{array}\right\} \rightarrow$ Q_1, Q_2 is filled with the color of P2.

Case 2 : $\left.\begin{array}{l} z_1^{(2)} < z_1^{(1)} \\ z_2^{(2)} < z_2^{(1)} \end{array}\right\} \rightarrow$ Q_1, Q_2 is filled with the color of P2.

Case 3 : Intersection is taking place

In this case we have to go back pixel by pixel and determine which plane is closer. Then choose the color of the pixel.

Algorithm (scan-line):

For each line perform step (1) through step (3).

1) For all pixels on a scan-line, set depth [x] = 1.0 (max value) & Intensity [x] = background-color.

2) For each polygon in the scene, find all pixels on the current scan-line (say S1) that lies within the polygon. For each of these x-values:

(a) calculate the depth z of the polygon at (x, y).

(b) if z < depth [x], set depth [x] = z & intensity corresponding to the polygon's shading.

3) After all polygons have been considered, the values contained in the intensity array represent the solution and can be copied into a frame-buffer.

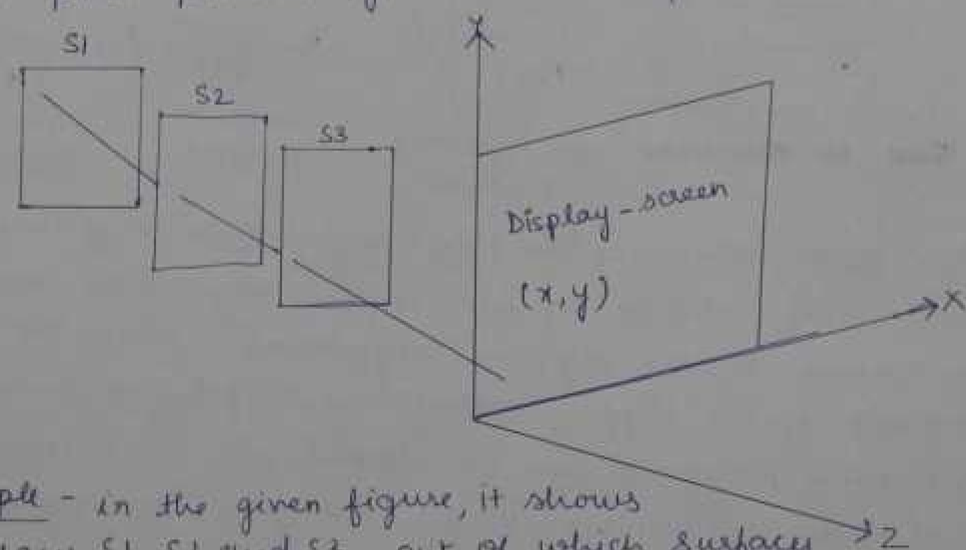
Ex Given two triangles P with vertices P1 (100, 100, 50), P2 (50, 50, 50) P3 (150, 50, 50) and q with vertices Q1 (40, 80, 60), q2 (70, 70, 50), Q3 (10, 75, 70), determine which triangle should be painted first using the scan-line method.

Sol In the scan-line method, two triangles P and Q are tested for overlap in xy-plane. Then they are tested for depth overlap. In this question, there is no overlap in the depth. But P and Q have overlap in xy-plane. So the Q is painted first followed by P.

In Z-buffer algorithm every pixel position on the projection plane considered for determining the visibility of surfaces w.r.t this pixel. On the other hand in scan-line method all surfaces intersected by a scan line are examined for visibility. The visibility test in Z-buffer method involves the comparison of depths of surfaces w.r.t a pixel on the projection plane. The surface closest to the pixel position is considered visible. The visibility test in scale-line method compares depth calculations for each overlapping surface to determine which surface is nearest to the view-plane so that it is declared as visible.

(c) Depth-buffer (or Z-buffer) Method

Depth-buffer method is a fast and simple technique for identifying visible surface. This method also referred to as the z-buffer method, since object depth is usually measured from the view plane along the z-axis of a viewing system. This algorithm compares surface depths at each pixel position (x, y) on the view plane.

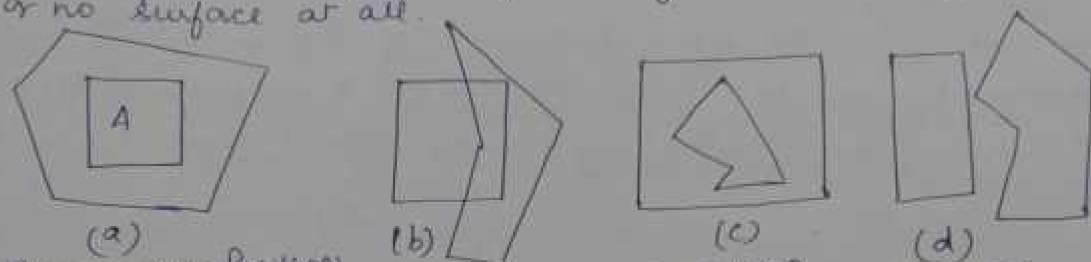


For example - in the given figure, it shows three surfaces S1, S2 and S3, out of which surfaces S1 has the smallest z-value at (x, y) position. So surfaces S1 is visible at that position. So its surface intensity value at (x, y) is saved in the refresh-buffer.

Here the projection is orthographic and the projection plane is taken as the xy-plane. So each (x, y, z) position on the polygon surfaces corresponds to the orthographic projection point (x, y) on the projection plane. Therefore, for each pixel position (x, y) in the view plane, object can compared by z-values, as shown in the figure.

(b) Area-Subdivision method

This method is essentially an image-space method but uses object space operations reordering of surfaces according to depth. This method takes advantage of area-coherence in a sense by locating those view areas that represent part of a single surface. In this method we successively subdivide the total viewing area, usually a rectangular window, into small rectangles until each small area is the projection of part of a single visible surface or no surface at all.



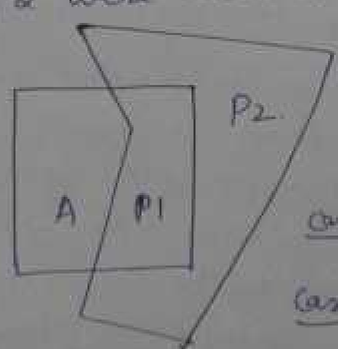
(a) SURROUNDING POLYGON

(b) INTERSECTING POLYGON

(c) CONTAINED POLYGON

(d) DISJOINT POLYGON

The classification of the polygons within a picture is the main computational expense of the algorithm and is analogous to the clipping algorithms. With the use of any one of the clipping algorithms, a polygon is category 2 (intersecting polygon) can be clipped into a contained polygon and a disjoint polygon in (e). Therefore, we could proceed as if category 2 were eliminated.



No further division of a specified area are needed if one other condition is true

<u>Case 1</u>: All polygons are disjoint from area

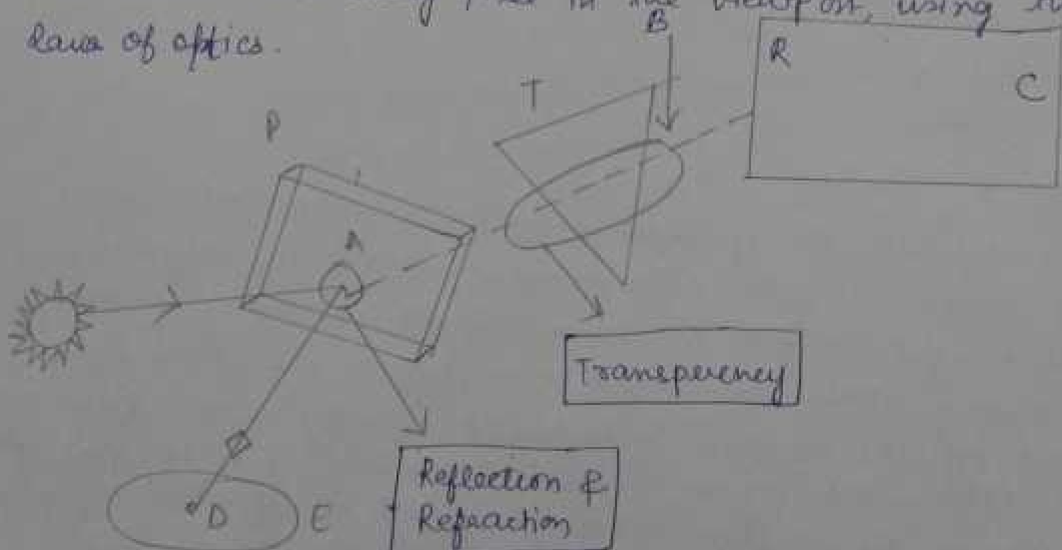
<u>Case 2</u>: Exactly one polygon faces, after projection, intersecting or contained in the square area

case 3: There is a single surrounding polygon, but no intersecting or contained polygons.

case 4: More than one polygon is intersecting, contained in, or surrounding the area, but one is a surrounding polygon that is in front of all the other polygons

(iii) Basic-ray tracing algorithm

It is also describe the concept behind anti-alaising, a method for improving the realism of an image by smoothing the jagged edges caused by the digital nature of computer displays. The hidden-surface removal is the most complete and most versatile method for display of objects in a realistic fashion. The concept is simply to take one ray at a time, emanating from the viewer's eye and reaching out to each and every pixel in the viewport, using the laws of optics.



illustrates some of the general principles of ray tracing.

(v) Equation of a plane that passes through point $P(0,0,0)$ and normal to point is $\bar{N}(1,0,-1)$?

Sol Given $\bar{N}(1,0,-1)$ and $P(0,0,0)$; equation of plane = ?

Say $P'(x,y,z)$ be another point on the plane then

the line $\overrightarrow{PP'} = (x-0, y-0, z-0) = x\hat{i} + y\hat{j} + z\hat{k}$

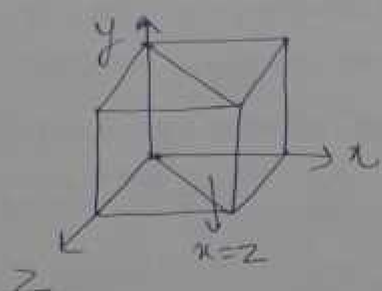
now determine the dot product of PP' and normal N

$$\overline{PP'} \cdot \bar{N} = 0 \Rightarrow n_1 x + n_2 y + n_3 z - (x_0 n_1 + y_0 n_2 + z_0 n_3) = 0$$

$$1 \cdot x + 0 \cdot y + (-1) \cdot z - (0+0+0) = 0$$

$$x - z = 0 \rightarrow \text{plane equation}$$

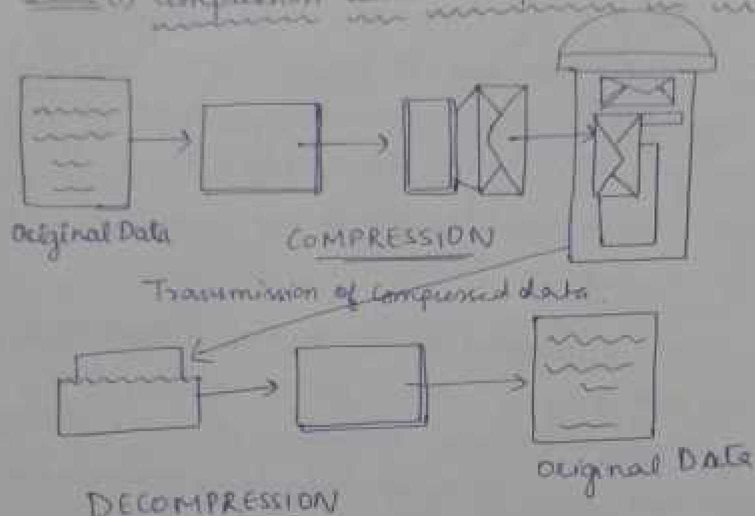
$\Rightarrow x = z$ is the required plane



(vi) Diffuse Reflection

It is a characteristic of light reflected from a dull, non-shiny surface. Objects illuminated solely by diffusely reflected light exhibit an equal light intensity from all viewing directions. That is in diffuse reflection light incident on the surface is reflected equally in all directions and is attenuated by an amount dependent upon the physical properties of the surface. Since the light is reflected equally in all the directions the perceived illumination of the surface is not dependent on the position of observer. Diffuse reflection models the light reflecting properties of matt surfaces i.e surfaces that are rough or grainy which tend to scatter the reflected light in all directions. The scattered light is called diffuse reflection

Ques 4 (i) Compression and Decompression in Digital Video



Compression is a reversible conversion of data to a format that requires fewer bits, usually performed so that the data can be stored or transmitted more efficiently. The size of data is compressed from (C) relative to original size (O) is known as the compression ratio ($R = C/O$). If the inverse of the process, decompression produces an exact replica of the original data then the compression is lossless.

Compression is analogous to folding a letter before placing it in small envelope so that it can be transported more easily and cheaply (as in figure). Compressed data, like the folded letter, is not easily read and must be decompressed or unfolded to restore it to its original form.

Compression techniques used for digital video can be categorised into three main groups:

- General purpose compression techniques can be used for any kind of data.
- Intra-frame compression techniques work on images. Intra frame compression is compression applied to still images. This technique can be applied to individual frames of a video sequence.

(ii) Hypertext and hypermedia

Hypertext :- Hypertext is conceptually, the same as a regular text - it can be stored, read, searched or edited with an important difference : hypertext is text - with pointers to other text. The browsers let you deal with the pointers in a transparent way - select the pointer, and you are presented with the text that is pointed to.

Hypermedia - Hypermedia is a ~~subset~~ superset of hypertext. Hypermedia documents contain links not only to other pieces of text, but also to other forms of media - sounds, images & movies. Images themselves can be selected into link to sounds or documents. Hypermedia simply combines hypertext and multimedia.

(iii) Types of Bitmap and Vector graphics

Types of Bitmap graphics

Their are four main categories

(1) Line art : These are the images that contain only two colours, usually black and white.

(2) Grayscal images :- images, which contain various shades of gray as well as pure black and white.

(3) Multitones : Such images contain shades of two or more colors.

(4) Fuel color images :- The color information can be described using a number of colour spaces : RGB, CMYK for instance.

Types of Vector graphics

- EPS : the most popular file format to exchange vector drawing although EPS - files can also contain bitmap data.
- PDF :- versatile file format that contains just about any type of data including complete pages, not yet widely used to exchange just images
- PICT :- can contain both bitmap and vector data but mainly used on Macintosh computer

(v) Gif and Jpeg images

Graphic Interchange Format (GIF)

GIF is an efficient means to transmit images across data networks. In the early 1990s the original designers of the world wide web adopted GIF for its efficiency and widespread familiarity. The overwhelming majority of images on the web are now in GIF format, and virtually all web browsers that support graphics can display GIF files. GIF files incorporate a compression scheme to keep file sizes at a minimum, and they are limited to 8-bit colour palettes.

JPEG graphics

The other graphic file format commonly used on the web to minimise graphics file sizes is the joint photographic Experts Group (JPEG) compression scheme. Unlike, GIF graphics, JPEG images are full-colour images. JPEG images find great acceptability among photographers, artists, graphic designers, medical imaging specialists, art historians and other groups for whom image quality is paramount and where colour fidelity cannot be compromised.

(iv) Ray Tracing

Ray Tracing - Ray tracing is an exercise performed to attain the realism in a scene. In simple terms ray tracing is a global illumination based rendering method used for producing views of a virtual 3-dimensional scene on a computer. Ray tracing is closely allied to, and is an extension of ray-casting, a common hidden-surface removal method. It tries to mimic actual physical effects associated with the propagation of light. One of the prime advantage of raytracing method is, it makes use of actual physics and mathematics behind light.

~~Q. Ray Casting~~

Ray Casting

Ray casting is a method in which the surfaces of objects visible to camera are found by throwing rays of light from the viewer into scene. The idea behind ray casting is to shoot rays from the eye, one per pixel and find the closest object blocking the path of that ray - think of an image as a screen -door with each square in the screen being a pixel. This is then the object the eye normally sees through the pixel. Using the material properties and the lights in the scene, this algorithm can determine the shading of objects.