B.Tech.

Fifth Semester Examination

Computer Graphics (CSE-303-F)

Note : Attempt any ***five*** questions. All questions carry equal marks.

Q. 1. (a) Discuss the applications of computer graphics.

Ans. Applications of Computer Graphics :

1. User Interfaces : Most applications that run on personal computers and workstations, & even those that run on terminals attached to time-shared computers & network compute servers, have user interfaces that rely on desktop window systems to manage multiple simultaneous activities, & on paint-and-click facilities to allow user to select menu items, icons & objects on the screen; typing is necessary only to input text to be stored and manipulated. Word-processing, spreadsheet and desktop-publishing programs are typical applications that take advantage of such user-interface techniques.

2. (Interactive) Plotting in Business, Science and Technology : The next most common use of graphics today is probably to create 2D and 3D graphs of mathematical, physical and economic functions; historgrams, bar & pie charts; task-scheduling charts; inventory and production charts; & the like.

3. Office Automation and Electronic Publishing : The use of graphics for the creation and dissemination of information has increased enormously since the advent of desktop publishing on personal computers. Many organizations whose publications used to be printed by outside specialists can now produce printed materials in hours. Office automation and electronic publishing can produce both traditional printed (hard copy) documents & electronic (soft copy) documents that contain text, tables, graphs, & other forms of drawn or scanned in graphics.

4. Computer-Aided Drafting and Design : In CAD, interactive graphics is used to design components & systems of mechanical, electrical, electromechanical, and electronic devices, including structures such as buildings, automobile bodies, very large-scale-integrated (VLSI) chips, optical systems, & telephone & computer networks.

5. Simulation and Animation for Scientific Visualization and Entertainment : Computer-produce animated movies & displays of the time-varying behaviour of real & simulated objects are becoming increasingly popular for scientific and engineering visualization.

6. Art and Commerce : Computer graphics is used to produce pictures that express a message & attract attention. Personal computers and teletext and videotext terminals in public places such as museums, transportation-terminals, supermarkets & hotels as well as in private homes, offer much simpler but still informative protures that let users orient themselves, make choices or even "teleshop" and conduct other business transactions.

Q. 1. (b) Discuss DDA line drawing algorithm.

Ans. DDA Line Drawing Algorithm :

DDA Algorithm : The digital differential analyzer (DDA) is a scan conversion line algorithm based on calculating either Δy or Δx, using

$$\Delta y = m\Delta x \;\; \& \;\; \Delta x = \frac{\Delta y}{m}$$

Case I : Consider first a line with +ve slope. If the slope is less than or equal to 1, we sample at unit x

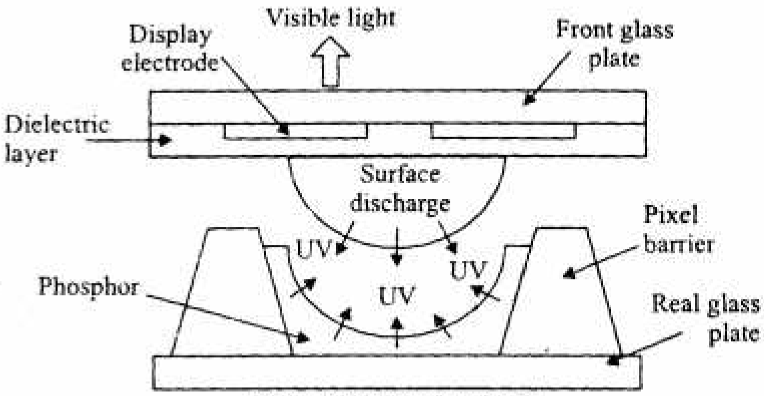


Fig. Plasma display panel

Q. 2. (a) Write steps required to draw a line from point (x_1, y_1) to (x_2, y_2) using Bresenham's line drawing algorithm.

Ans. There are following steps required to compute pixel positions for drawing a line,

(i) Compute initial values :

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

$$d = \text{in } c_1 - \Delta x$$

$$inc_2 = 2(\Delta y - \Delta x)$$

$$inc_1 = 2\Delta y$$

(ii) Set (x, y) equal to lower left-hand end point and x end equal to the largest value of x. If $\Delta x < 0$, then $x = x_2$, $y = y_2$, $x_{end} = x_1$

If $\Delta x > 0$, then $x = x_1$, $y = y_1$, $x_{end} = x_2$

(iii) Plot a point at the current (x, y) coordinates.

(iv) If $x = x_{end}$ then stop.

(v) If $d < 0$, then $d = d + inc_1$ otherwise $d = d + inc_2$ and then $y = y + 1$

(vi) $x = x + 1$ and plot the point at (x, y)

(vii) Goto step (iv).

Q. 2. (b) What is meant by antialiasing? Explain various methods used to develop antialiasing routines.

Ans. Antialiasing : Antialiasing is one which compensate the consequences of undersampling process. The displays which allows setting pixel to grey levels between blacks and white, provides a means to reduce the effect of aliasing and it uses the grey levels to gradually turn off the pixels in a row as it gradually turn on the pixels in the next.

Basically, there are two methods of antialiasing :

(i) Post-filtering (ii) Pre-filtering

(i) Post-Filtering : In this process sample rate is increased and this is accomplished by increasing the resolution of the raster. However there is a limit to the ability of CRT raster scan devices to display very fine rasters.

This limit suggests that the raster to be calculated at higher resolution and displayed at lower resolution, using some type of averaging to obtain the pixel attributes at the lower resolution. Actually the concept of filtering originates from the field of signal processing. This technique is called post filtering.

There are two main post-filtering techniques :

(a) Super sampling (b) Low-pass filtering.

(ii) Pre-Filtering : In this method, a pixel is treated as a finite area rather than as a point and this technique basically, works on the true signal in the continuous space to drive proper values for individual pixels There is one most popular pre-filtering technique, is termed as area sampling.

Area Sampling : We superimpose a pixel grid pattern onto the continuous object definition. For each pixel area that intersects the object, we calculate the percentage of overlap by the object. This percentage determines the proportion of the overall intensity value of the corresponding pixel that is due to the object's contribution. In other words, the higher the percentage of overlap, the greater influence the object has on the pixel's overall intensity value.

Q. 3. (a) Explain the difference between parallel and perspective projection.

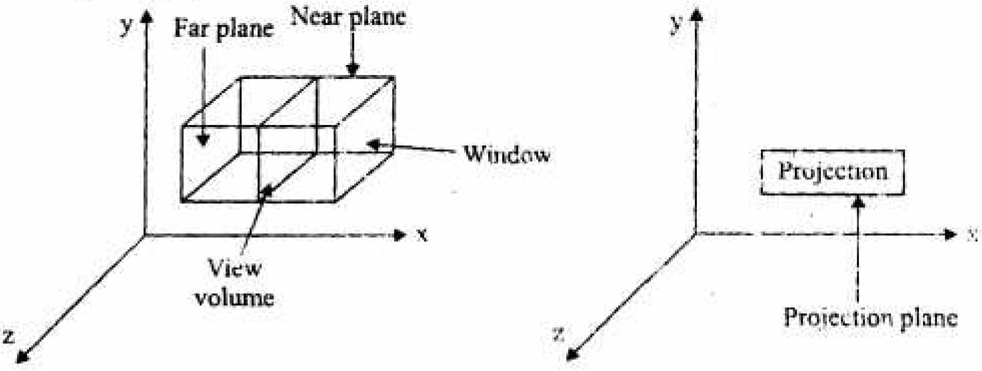
Ans. Parallel Projections : These are linear transform that are useful in blueprints, schematic diagrams etc.

In parallel projection, we have two types of projection named :

(i) Oblique (ii) Orthographic

Parallel projection is that, when we discuss the three dimensional object but our viewing surface is only two dimensional, if we projected the 3D object, we obtain our viewing surface onto the two dimensional screen. This can be done by the parallel projection.

In this type of projection line are parallel.



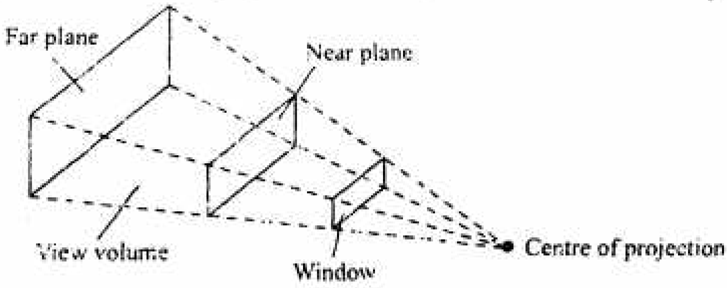
Perspective Projection : These are non-linear transforms. Perspective projections can be implemented with a matrix in projective space followed by the homogeneous coordinate. This is very useful in architectural rendering, realistic views etc.

In this we have 3 types :

(i) 1pt (ii) 2pt (iii) 3pt

In a perspective projection, the object is very far from the viewer and it seems very small. This provide the viewer with a depth one, an indication of which portions of the image correspond to parts of the object which are close or far away.

In such projection, the lines of projection are not parallel.



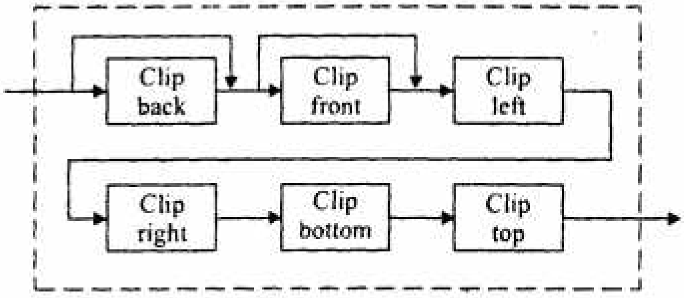
Q. 3. (b) Explain briefly :

(i) 3D-clipping

(ii) 3D-scaling

(iii) 3D-translation

Ans. (i) 3D-Clipping : In 3D geometry the clipping is very analogous to that of 2D geometry with minor differences. Three dimensional clipping algorithms are direct adaptations of the 2D Cohen Sutherland, Sutherland Hodgman & midpoint subdivision algorithms. The modifications necessary arise from the fact that we are now clipping lines against the six faces of the view volume, which are planes, as opposed to the four edges of the 2D windows, which are lines.



Clipping process

(ii) 3D Scaling : Using scaling transformation, we can change the size of any graphics object. To do this, we have to multiply the coordinates by suitable scaling factors. Here the matrix form of equation can be given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

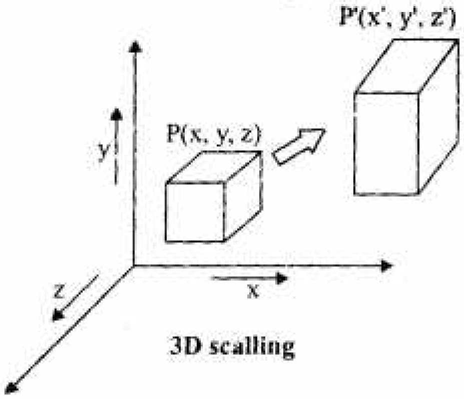
Or $$P' = S.P.$$

$P \rightarrow$ Point on 3D object having coordinate (x, y, z)

$P' \rightarrow$ After scaling, shifted new point (x', y', z')

$S \rightarrow$ Scaling matrix

$$x' = x.S_x, \quad y' = y.S_y, \quad z' = z.S_z$$



3D scalling

(iii) 3D Translation : Translation means to shift and re-positioning the object along a straight line path from one coordinate location to other. It can be given as

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$P(x, y, z) \rightarrow$ Original coordinates

$P'(x', y', z') \rightarrow$ New coordinates

Or, $$P' = T.P$$

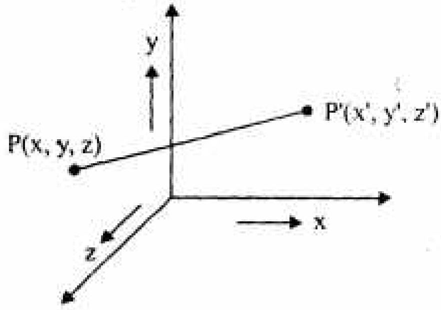
$T \rightarrow$ Translation matrix.

Again the equation can also be represented as a set of three equations as given below

$$x' = x + t_x$$
$$y' = y + t_y$$
$$z' = z + t_z$$

This situation is shown in the figure as shown alongside

So, to do translation of a 3D object we would transform each component of the coordinate separately.



Q. 4. Using any of the line-clipping algorithm, obtain the visible portion of the following line segment :

(i) $P_1 = (0.5,\ 0.4)$ and $P_2 = (1.6,\ 0.7)$

(ii) $P_1 = (0.4,\ -1.6)$ and $P_2 = (0.4,\ 2.7)$

(iii) $P_1 = (0.4,\ -1.6)$ and $P_2 = (0.8,\ -2.7)$

(iv) $P_1 = (-2,\ -3)$ and $P_2 = (3,\ 5)$

For the window $(x_{w\,min},\ y_{w\,min}) = (0,\ 0)$ and $x_{w\,max},\ y_{w\,max} = (1,\ 1)$

In each case determine the end points of the clipped line segment.

Ans. The region code for point $(x,\ y)$ are as follows :

$$\text{Bit 1} = \text{sign}(y - y_{w\,max}) = \text{sign}(y-1)$$
$$\text{Bit 2} = \text{sign}(y_{w\,min} - y) = \text{sign}(-y)$$
$$\text{Bit 3} = \text{sign}(x - x_{w\,max}) = \text{sign}(x-1)$$
$$\text{Bit 4} = \text{sign}(x_{w\,min} - x) = \text{sign}(-x)$$

Here sign
$$(a) = \begin{cases} 1 & \text{if } a \text{ is positive} \\ 0 & \text{otherwise} \end{cases}$$

By Cohen-Sutherland Algorithm

(i) $P_1 = (0.5,\ 0.4)$ and $P_2 = (1.6,\ 0.7)$:

So,
$$P_1(0.5,\ 0.4) \rightarrow 0000$$
$$P_2(1.6,\ 0.7) \rightarrow 0010$$

Parametric form of the line P_1P_2

$$\begin{cases} x = x_1 + t(x_2 - x_1) \\ y = y_1 + t(y_2 - y_1) \end{cases}$$

Or,
$$x = 0.5 + t(1.1)$$
$$y = 0.4 + t(0.3)$$

Since line P_1P_2 crosses the line $x = 1$, then

$$y = y_1 + \left(\frac{a - x_1}{x_2 - x_1}\right)(y_2 - y_1)$$
$$= 0.4 + \left(\frac{1 - 0.5}{1.1}\right)(0.3)$$
$$= 0.57$$

So intersection point $I(1,\ 0.57)$

Region code for point I, $\quad I \rightarrow 0000$

Since both the points P_1 & I have region code 0000, therefore line is visible and having endpoints $P_1(0.5, 0.4)$ and $I = (1, 0.57)$. **Ans.**

(ii) $P_1 = (0.4, -1.6)$ and $P_2 = (0.4, 2.7)$:

So $P_1 = (0.4, -1.6) \rightarrow 0100$

$P_2 = (1.4, 2.7) \rightarrow 1000$

Since logical AND of $P_1 P_2$ is 0000, so this line is candidate of clipping.

This line intersects two lines $x = 0$ & $y = 1$. Parametric form of the line $P_1 P_2$

$$x = 0.4 + t(0) = 0.4$$

$$y = -1.6 + t(4.3)$$

For $y = 0 \Rightarrow y_1 = 0$

& $$y = 0.4 + \left(\frac{0-1.6}{2.7+1.6}\right)(0.4 - 0.4)$$

$$= 0.4$$

Intersection point $I(0.4, 0)$

Region code for $I \rightarrow 0000$, since (0000) AND (1000) is (0000). Now P_2 is outside the window. We clip P_1, I. Since line segment IP_2 intersect the $y = y_{max} = 1$

$$y_{I_1} = 1$$

& $$x_{I_1} = 0.4 + \left(\frac{1-0}{2.7-0}\right)(0.3)$$

$$x_{I_1} = 0.4$$

Region code for intersection point $I_1(0.4, 1) \rightarrow 0000$. So clip the line segment $I_1 P_2$ (do not display).

Since both end points of II_1 have region code (0000), therefore completely inside the window and has the endpoints (0.4, 0) and (0.4, 1). **Ans.**

(iii) $P_1 = (0.4, -1.6)$ and $P_2 = (0.8, -2.7)$:

Region code for $P_1 = (0.4, -1.6) \rightarrow 0100$

$P_2 = (0.8, -2.7) \rightarrow 0100$

(0100) AND (0100) = 0100 (which is non-zero).

Therefore line is not visible.

(iv) $P_1 = (-2, -3)$ and $P_2 = (3, 5)$:

Region code for $P_1 = (-2, -3) \rightarrow 0101$

$P_2 = (3, 5) \rightarrow 1010$

(0101) AND (1010) $\Rightarrow$ (0000) which is zero i.e., this is candidate for clipping.

Equation of line segment $P_1 P_2 = y + 3 = \frac{8}{5}(x + 2)$

Or $$-5y + 8x + 1 = 0$$

Equation of four window boundary

$$x = 0, \quad y = 0, \quad x = 1 \ \& \ y = 1$$

Find the intersection points of line P_1P_2 with all the boundaries and if intersection points lies between the end points of corresponding boundary then line segment crosses the window.

$$I_1 = (0, .2)$$
$$I_2 = (-0.125, 0)$$
$$I_3 = (1, 1.4) \ \&$$
$$I_4 = (0.5, 1)$$

Here I_1 & I_2 satisfy the condition for crossing the window.

Region code for $\quad I_1 = (0, 0.2) \rightarrow 0000$

$$I_2 = (0.5, 1) \rightarrow 0000$$

Because both the end points I_1 & I_2 have the region code zero. Therefore I_1I_2 is completely inside the window. Clip the rest portion of the line i.e.,

$$P_1I_1 \ \& \ I_2P_2 \quad \textbf{Ans.}$$

End points for the line segment

$$I_1 = (0, 0.2) \quad \& \quad I_2 = (0.5, 1)$$

Q. 5. (a) Explain depth buffer method for hidden surface detection.

Ans. Depth Buffer Method : The z-buffer or depth buffer, is one of the simple of the visible surface or hidden surface algorithms and it is an image space algorithm.

The z-buffer is a simple extension of the frame buffer idea. A frame buffer is used to store the attributes (intensity or shade) of each pixel in the image space. The z-buffer is a separate depth buffer used to store the z co-ordinate, or depth, of every visible pixel in the image space. In use the depth or z-value of a new pixel to be written to the frame buffer is compared to the depth of that pixel stored in the z-buffer. If the comparison indicates that the new pixel is in front of the pixel stored in the frame buffer, then the new pixel is written to the frame buffer and the z-buffer updated with new z-value. If not, no action is taken.

Conceptually, the algorithm is a search over x, y for the largest value of $z(x, y)$. This algorithm is frequently implemented for polygonally represented scenes and also applicable for any object for which depth and shading characteristics can be calculated. Scenes can contain mixed object types and may be of any complexity. It has basically two main disadvantages :

(i) It requires larger storage.

(ii) Difficulty & expense of implementing antialiasing transparency and transparency effects.

Algorithms : (i) Set the frame buffer to the background intensity or colour.

(ii) Set the z-buffer to the minimum z-value.

(iii) Scan convert each polygon in arbitrary order.

(iv) For each pixel (x, y) in the polygon, calculate the depth $z(x, y)$ at that pixel.

(v) Compare the depth $z(x, y)$ with the value stored in the z-buffer at that location, z-buffer (x, y).

(vi) If $z(x, y) >$ z buffer (x, y) then write the polygon attributes (intensity, colour, etc) to the frame buffer and replace z-buffer (x, y) with $z(x, y)$. Otherwise no action is taken

Q. 5. (b) Explain Painter's algorithm.

Ans. Painter's Algorithm : Painter's algorithm basically perform the following two steps :

(i) Surfaces are stored in order of decreasing depth.

(ii) Surfaces are scan converted in order, starting with the surface of greatest path

The painter's algorithm is very analogous to creating an oil painting. In creating an oil painting, an artist first paints the background colours, text, the most distant objects added, then the nearer objects and so forth. At the final step, the foreground objects are painted on the canvas over the background and other objects that have been painted on the canvas. Each layer of paint covers up the previous layer. Using a similar technique, we first sort surfaces according to their distance from the viewplane. The intensity values for the farthest surface are then entered into the refresh buffer. Taking each succeeding surface in turn, we "paint" the surface intensities onto the frame buffer over the intensities of previous processed surfaces.

Algorithm :

Step 1 : The surfaces are ordered according to the largest z value present on each surface.

Step 2 : Comparison is made with the largest or greatest depth(d) with the depths of the other surfaces to find the over-lapping surfaces, if present. If no such overlap in depths are found, 'd' is scan converted to scan converted to 'D'.

Step 3 : For the surface the intensity value is entered into the refresh buffer.

Step 4 : Repeat the process for all the surfaces present.

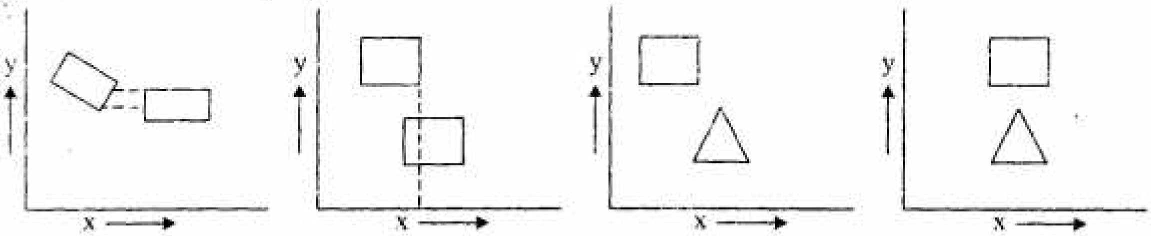
Step 5 : If the depth overlapping is observed, further tests are required. These include :

(i) Whether the bounding rectangles in the XY plane overlapped or not.

(ii) Whether one surface is completely behind the overlapping surface relative to the viewing position or not.

(iii) Whether the overlapping surface is completely in front of the surface relative to the viewing position.

(iv) Whether the projections of the 2 surfaces overlap on the view plane or not.



(Overlapping in y dir.) (Overlapping in x dir.) (Not overlapping) (Completely overlapped)

Q. 6. (a) Describe the properties of B-spline curves.

Ans. Properties of B-Spline Curves : (i) The sum of all B-spline blending functions, if curve lies within the convex hull, is always 1, and all are always positive

$$\sum_{i=0}^{n} N_i, p(u) = 1$$

(ii) Any curve having $n+1$ control points, has $n+1$ blending functions.

(iii) This curve has degree $(d) = \text{order } (P) - 1$ and continuity C^{P-2} over range of u.

(iv) Each piece of the B-spline is influenced by P control points.

(v) Any control point can affect all the sections i.e., it has numerical instability.

(vi) The range of parameter u is divided into $n+P$ sub-intervals by the $n+P+1$ values specified in the knot vectors (i.e., $m = n+P$).

(vii) For knot vector T, resulting B-spline curve is defined only in the interval from the knot values u_{P-1} to u_{n+1}.

(viii) The degree of a B-spline polynomial is independent on the number of control points with certain limits.

(ix) B-splines have local control over the shape of the curves.

Q. 6. (b) What are the basic differences between B-spline and Bezier curves?

Ans. In Bezier curves, each of the control points affect every point on the surface. Bezier curves are not very good at providing localized control.

B-spline curves can be considered to be a superset of Bezier curves. They can do everything that a bezier curve can do, plus more! One of the benefits of B-spline over bezier curves is that B-splines provide localized control. They provide a mechanism for controlling the extend of influence of each control point across the surface.

Another advantage of B-splines is that they provide a mechanism for controlling the smoothness of a curve or surface.

In Bezier curves the 'order' of curve is always equal to the no. of control points which means that for a fixed no. of control points the smoothness of Bezier curve is going to remain the same.

On the other hand the B-spline curves give you the ability to change the order of the curve or surface independent of the no. of control points.

Q. 7. (a) How can an image be described as a function?

Ans. The image of a subset of a function's domain under (or through) the function is the set of all outputs obtained when the function is evaluated at each element of the subset. The inverse image or preimage of a particular subset S of the codomain of a function is the set of all elements of the domain that map to the members of S.

Image and inverse image may also be defined for general binary relations, not just functions.

The word "image" is used in three related ways. In these definitions, $f : X \to Y$ is a function from set X to set Y.

Image of an Element : If x is a member of X, then $f(x) = y$ (the value of f when applied to x) is the image of x under f.y is alternatively known as the output of f for argument x.

Image of a Subset : The image of a subset $A \subseteq X$ under f is the subset $f[A] \subseteq Y$ defined by (in set-builder notation) :

$$f[A] = \{y \in Y \mid y = f(x) \text{ for some } x \in A\}$$

When there is no risk of confusion, $f[A]$ is simply written as $f(A)$. This convention is a common one; the intended meaning must be inferred from the context. This makes the image of f a function whose domain is the power set of X (the set of all subsets of X), and whose codomain is the power set of Y. See notation below.

Image of a Function : The image $f[X]$ of the entire domain X of f is called simply the image of f.

Inverse Image : "Preimage" redirects here. For the cryptographic attack on hash functions, see preimage attack. Let f be a function from X to Y. The preimage or inverse image of a set $B \subseteq Y$ under f is the subset of X defined by

$$f^{-1}[B] = \{x \in X \mid f(x) \in B\}$$

The inverse image of a singleton, denoted by $f^{-1}[\{y\}]$ or by $f^{-1}[y]$, is also called the fibre over y or the level set of y. The set of all the fibres over the elements of Y is a family of sets indexed by Y. This leads to the notion of a fibred category.

Again, if there is no risk of confusion, we may denote $f^{-1}[B]$ by $f^{-1}(B)$, and think of f^{-1} as a function from the power set of Y to the power set of X. The notation f^{-1} should not be confused with that for inverse function. The two coincide only if f is a bijection.

Notation for Image & Inverse Image : The traditional notations used in the previous section can be confusing. An alternative is to give explicit names for the image and preimage as functions between powersets :

Arrow Notation :

$$f^{-1} : P(X) - P(Y) \text{ with } f^{-1}(A) = \{f(a) \mid a \in A\}$$

$$f^{-1} : P(Y) - P(X) \text{ with } f^{-1}(B) = \{a \in X \mid f(a) \in B\}$$

Star Notation :

$$f^{*} : P(X) \to P(Y) \text{ instead of } f^{-}$$

$$f^{*} : P(Y) \to P(X) \text{ instead of } f^{-}$$

Q. 7. (b) What are the various techniques used for image processing?

Ans. Image Processing Techniques : With the use of image enhancement techniques, the difference in sensitivity between film and RTR can be decreased. A number of image processing techniques, in addition to enhancement techniques, can be applied to improve the data usefulness. Techniques include convolution edge detection, mathematics, filters, trend removal, and image analysis. Enhancement programs make information more visible :

(i) Histogram Equalization : Redistributes the intensities of the image of the entire range of possible intensities (usually 256 gray-scale levels).

(ii) Unsharp Masking : Subtracts smoothed image from the original image to emphasize intensity changes.

Convolution programs are 3-by-3 masks operating on pixel neighbourhoods :

(i) Highpass Filter : Emphasizes regions with rapid intensity changes.

(ii) Lowpass Filter : Smoothes images, blurs regions with rapid changes.

Math processes programs perform a variety of functions :

(i) Add Images : Adds two images together, pixel-by-pixel.

(ii) Subtract Images : Subtracts second image from first image, pixel by pixel.

(iii) Exponential or Logarithm : Raises to power of pixel intensity or takes log of pixel intensity. Non-linearly accentuates or diminishes intensity variation over the image.

Noise filters decrease noise by diminishing statistical deviations :

(i) Adaptive Smoothing Filter : Sets pixel intensity to a value somewhere between original value and mean value corrected by degree of noisiness. Good for decreasing statistical, especially single-dependent noise.

(ii) Median Filter : Sets pixel intensity equal to median intensity of pixels in neighbourhood. An excellent filter for eliminating intensity spikes.

Edge detection programs sharpen intensity-transition regions :

(i) First Difference : Subtracts intensities of adjacent pixels. Emphasizes noise as well as desired changes.

(ii) Sobel Operator : 3-by-3 mask weighs inner pixels twice as heavily as corner values. Calculates intensity differences.

(iii) Morphological Edge Detection : Finds the difference between dilated (expanded) and eroded (shrunken) version of image.

Image analysis programs extract information from an image :

(i) Gray-scale Mapping : Alters mapping of intensity of pixels in file to intensity displayed on a computer screen.

(ii) Image Extraction : Extracts a portion or all of an image and creates a new image with the selected area.