Name of Faculty:Dr.Rachana Dubey

Dr.Megha Kamble

Designation:Professor

Department:Computer Science and Engg

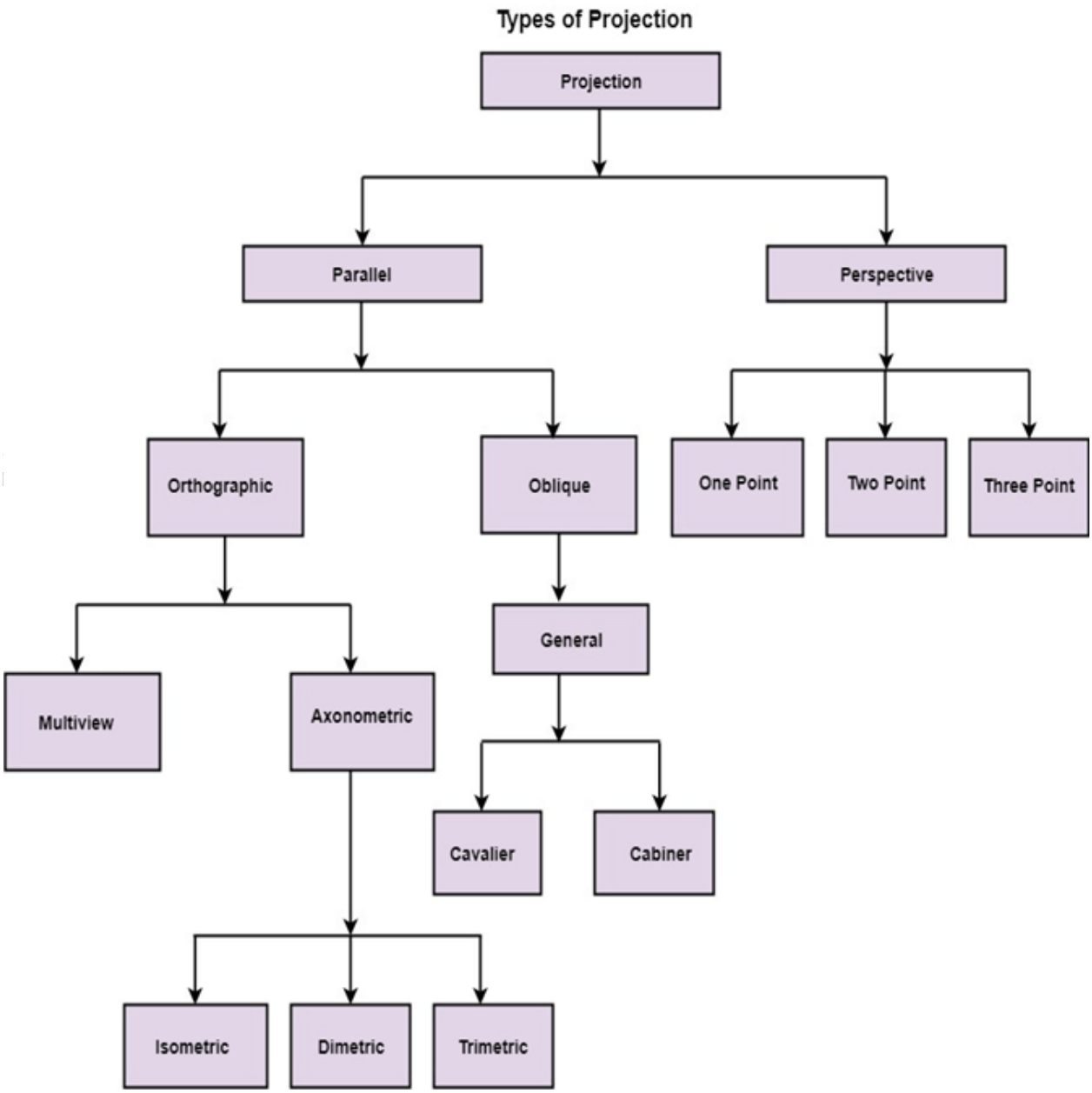
Subject: Computer Graphics and Visualisation

Unit: 3

Topic:3D Projection Transformation

LNCT GROUP OF COLLEGES "WORKING TOWARDS BEING THE BEST"

Projection. It is the process of converting a 3D object into a 2D object. It is also defined as mapping or transformation of the object in **projection** plane or view plane. The view plane is displayed surface.



Perspective Projection

In perspective projection farther away object from the viewer, small it appears. This property of projection gives an idea about depth. The artist use perspective projection from drawing three-dimensional scenes.

Two main characteristics of perspective are vanishing points and perspective foreshortening. Due to foreshortening object and lengths appear smaller from the center of projection. More we increase the distance from the center of projection, smaller will be the object appear.

Vanishing Point

It is the point where all lines will appear to meet. There can be one point, two point, and three point perspectives.

One Point: There is only one vanishing point as shown in fig (a)

Two Points: There are two vanishing points. One is the x-direction and other in the y -direction as shown in fig (b)

Three Points: There are three vanishing points. One is x second in y and third in two directions.

In Perspective projection lines of projection do not remain parallel. The lines converge at a single point called a center of projection. The projected image on the screen is obtained by points of intersection of converging lines with the plane of the screen. The image on the screen is seen as of viewer's eye were located at the centre of projection, lines of projection would correspond to path travel by light beam originating from object.

Important terms related to perspective

1. **View plane:** It is an area of world coordinate system which is projected into viewing plane.
2. **Center of Projection:** It is the location of the eye on which projected light rays converge.
3. **Projectors:** It is also called a projection vector. These are rays start from the object scene and are used to create an image of the object on viewing or view plane.

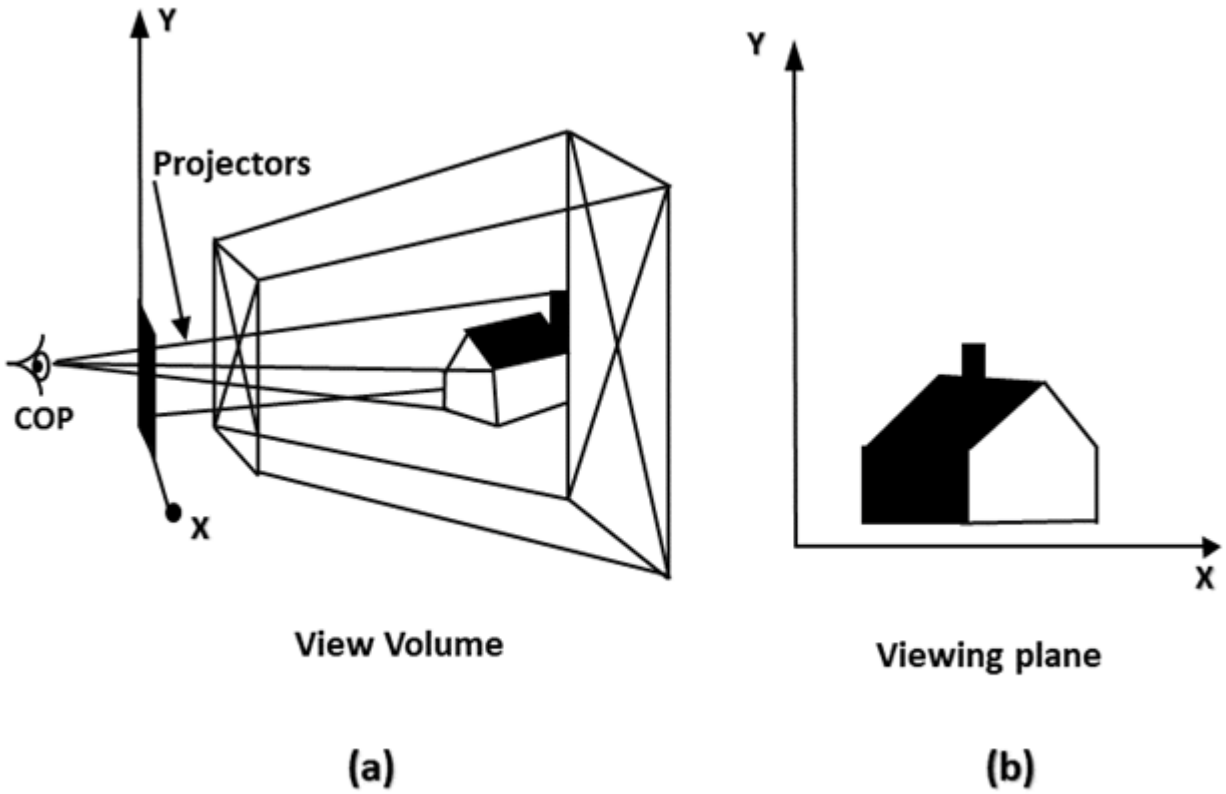
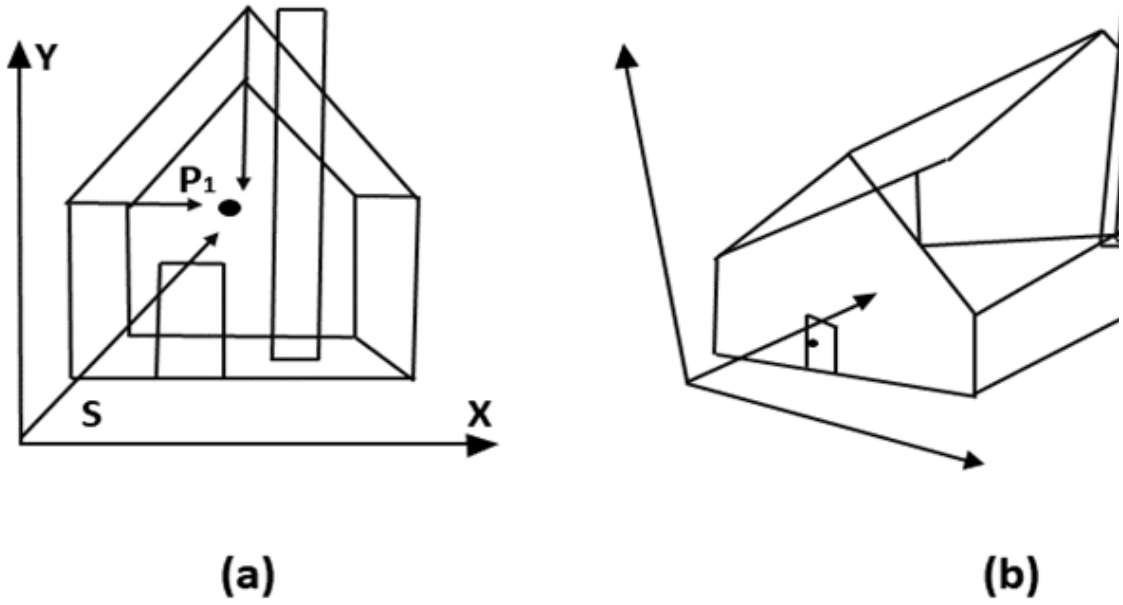


Fig: Prespective Projection

Anomalies in Perspective Projection

It introduces several anomalies due to these object shape and appearance gets affected.

1. **Perspective foreshortening:** The size of the object will be small of its distance from the center of projection increases.
2. **Vanishing Point:** All lines appear to meet at some point in the view plane.
3. **Distortion of Lines:** A range lies in front of the viewer to back of viewer is appearing to six rollers.

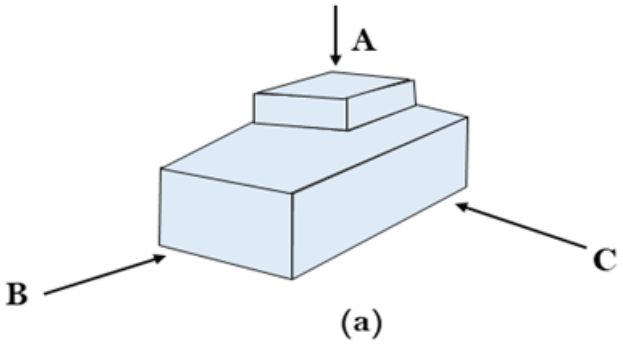


Foreshortening of the z-axis in fig (a) produces one vanishing point, P_1. Foreshortening the x and z-axis results in two vanishing points and Adding a y-axis foreshortening adds vanishing point along the negative y-axis.

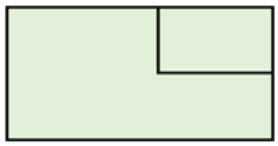
Parallel Projection

Parallel Projection use to display picture in its true shape and size. When projectors are perpendicular to view plane then is called **orthographic projection**. The parallel projection is formed by extending parallel lines from each vertex on the object until they intersect the plane of the screen. The point of intersection is the projection of vertex.

Parallel projections are used by architects and engineers for creating working drawing of the object, for complete representations require two or more views of an object using different planes.



(a)



Parallel projection from top i.e. A direction

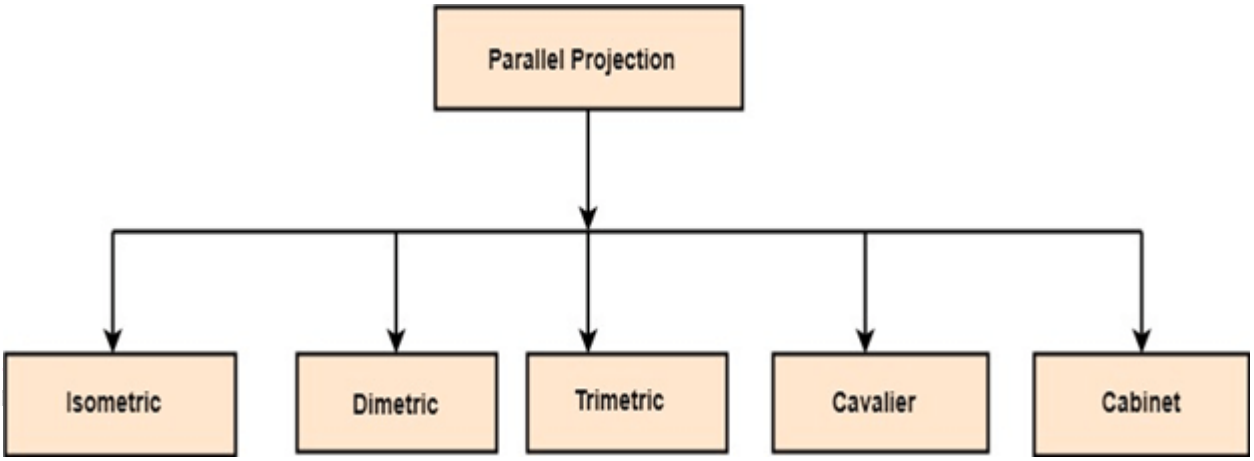
(b)

Parallel projection from top i.e. B direction

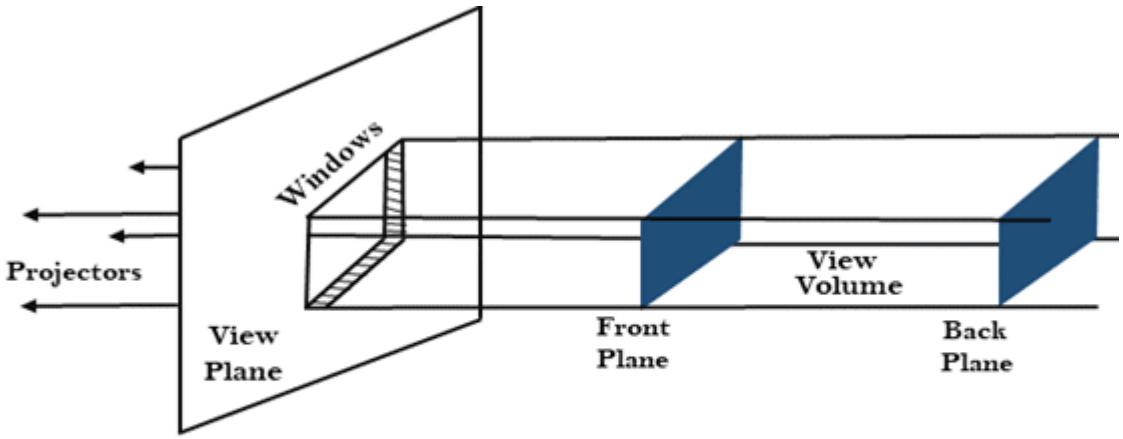
(c)

Parallel projection from top i.e. C direction

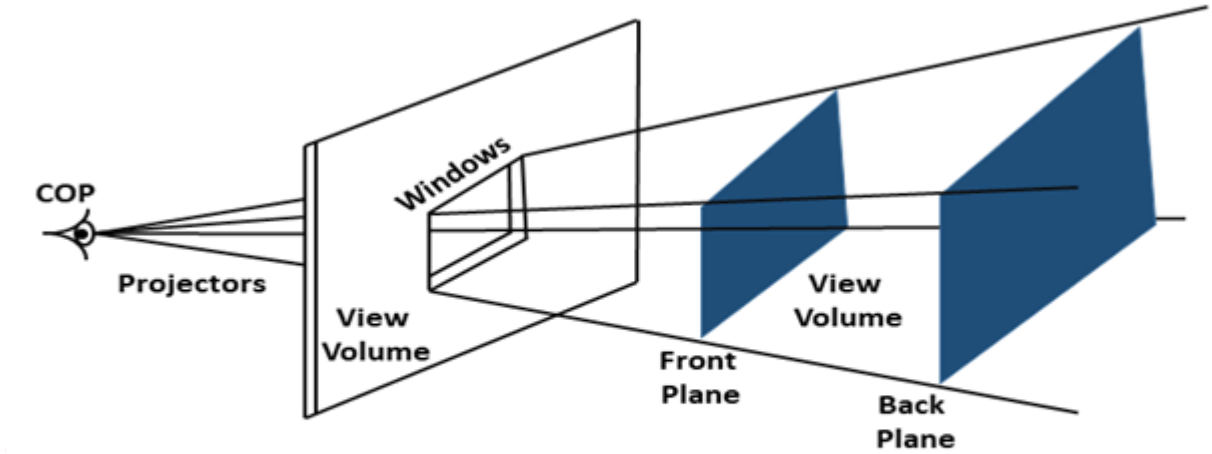
(d)



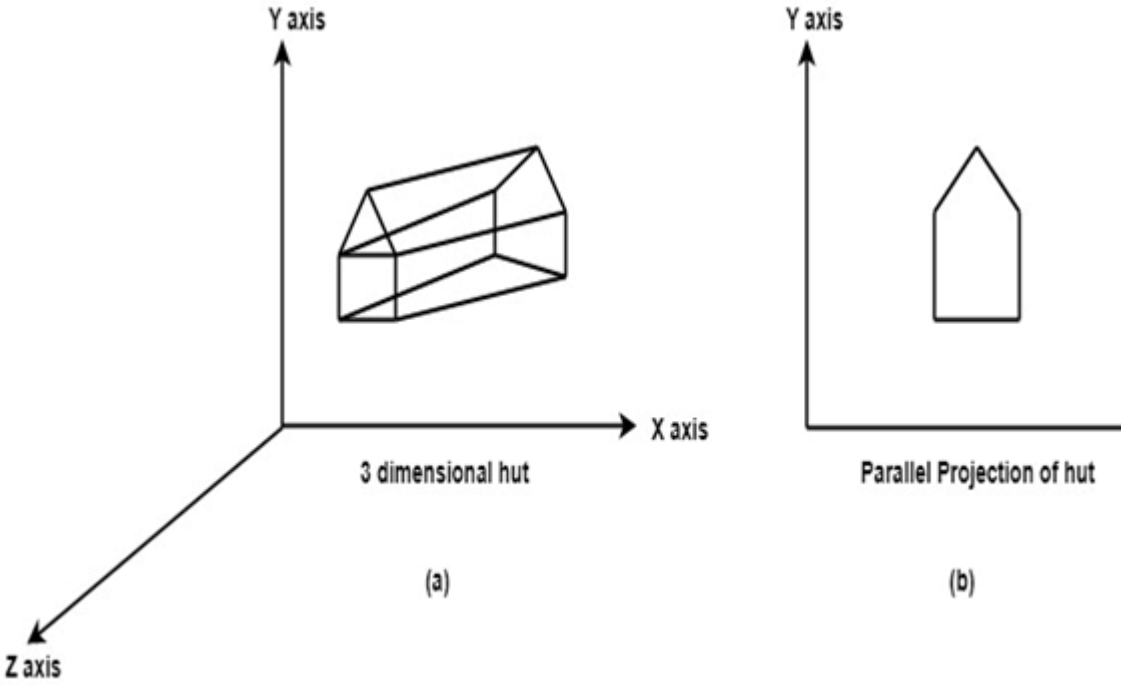
1. **Isometric Projection:** All projectors make equal angles generally angle is of 30°.
2. **Dimetric:** In these two projectors have equal angles. With respect to two principle axis.
3. **Trimetric:** The direction of projection makes unequal angle with their principle axis.
4. **Cavalier:** All lines perpendicular to the projection plane are projected with no change in length.
5. **Cabinet:** All lines perpendicular to the projection plane are projected to one half of their length. These give a realistic appearance of object.

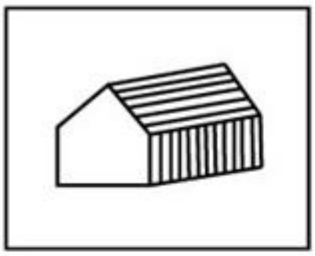


(a) Viewing Volume in orthographic projection

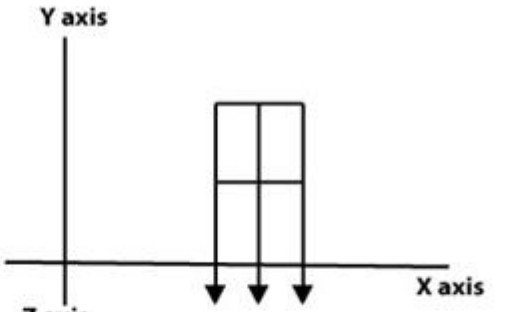


(b) Viewing volume in perspective projection

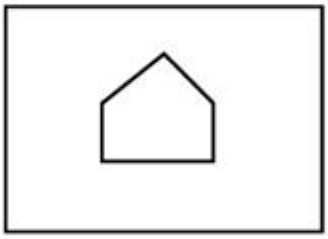




(a) Object original

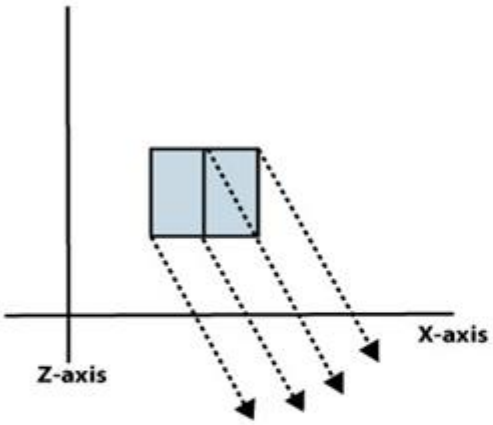


Center of projection is from front
(b)

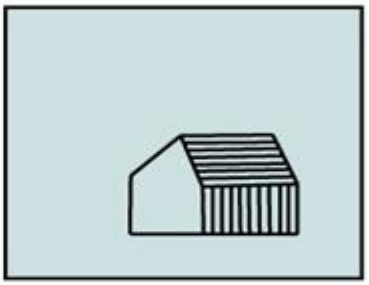


projected object
(c)

Fig (a) shows original object. Fig (b) shows object when projection is taken. Fig (c) gives projected object.



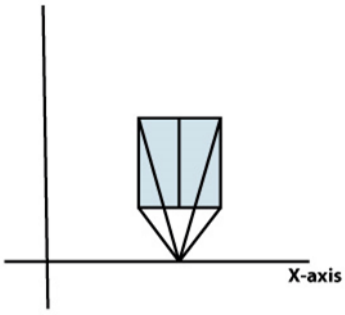
(d)



(e)

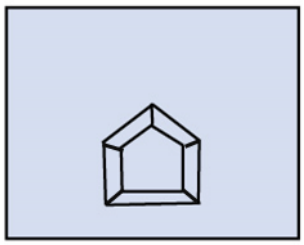
Fig (d) changes the direction of projection

Fig (e) shows object after changing direct projection.

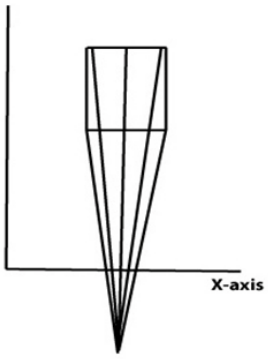


Center of projection meet at a chosen point

(a)



(b)



(a)

(b)

Name of Faculty:Dr.Rachana Dubey
Dr.Megha Kamble
Designation:Professor
Department:Computer Science and Engg

Subject: Computer Graphics and Visualisation

Unit: 3&4

Topic:Visualisation

Sub topics: Hidden surface removal

Illumination Model

Scalar Visualisation

Vector Visualisation

Color Model

LNCT GROUP OF COLLEGES "WORKING TOWARDS BEING THE BEST"

Hidden Surface Removal

This technique of "**hidden surface elimination**" may be done by extending the pixel attributes to include the "depth" of each pixel in a scene, as determined by the object of which it is a part.

1. One of the most challenging problems in computer graphics is the removal of hidden parts from images of solid objects.
2. In real life, the opaque material of these objects obstructs the light rays from hidden parts and prevents us from seeing them.
3. In the computer generation, no such automatic elimination takes place when objects are projected onto the screen coordinate system.
4. Instead, all parts of every object, including many parts that should be invisible are displayed.
5. To remove these parts to create a more realistic image, we must apply a hidden line or hidden surface algorithm to set of objects.
6. The algorithm operates on different kinds of scene models, generate various forms of output or cater to images of different complexities.
7. All use some form of geometric sorting to distinguish visible parts of objects from those that are hidden.
8. Just as alphabetical sorting is used to differentiate words near the beginning of the alphabet from those near the ends.
9. Geometric sorting locates objects that lie near the observer and are therefore visible.
10. Hidden line and Hidden surface algorithms capitalize on various forms of coherence to reduce the computing required to generate an image.
11. Different types of coherence are related to different forms of order or regularity in the image.
12. Scan line coherence arises because the display of a scan line in a raster image is usually very similar to the display of the preceding scan line.
13. **Frame coherence** in a sequence of images designed to show motion recognizes that successive frames are very similar.
14. **Object coherence** results from relationships between different objects or between separate parts of the same objects.
15. A hidden surface algorithm is generally designed to exploit one or more of these coherence properties to increase efficiency.

16. Hidden surface algorithm bears a strong resemblance to two-dimensional scan conversions.

Types of hidden surface detection algorithms

1. Object space methods
2. Image space methods

Object space methods: In this method, various parts of objects are compared. After comparison visible, invisible or hardly visible surface is determined. These methods generally decide visible surface. In the wireframe model, these are used to determine a visible line. So these algorithms are line based instead of surface based. Method proceeds by determination of parts of an object whose view is obstructed by other object and draws these parts in the same color.

Image space methods: Here positions of various pixels are determined. It is used to locate the visible surface instead of a visible line. Each point is detected for its visibility. If a point is visible, then the pixel is on, otherwise off. So the object close to the viewer that is pierced by a projector through a pixel is determined. That pixel is drawn is appropriate color.

These methods are also called a **Visible Surface Determination**. The implementation of these methods on a computer requires a lot of processing time and processing power of the com The image space method requires more computations. Each object is defined clearly. Visibility of each object surface is also determined.

Differentiate between Object space and Image space method

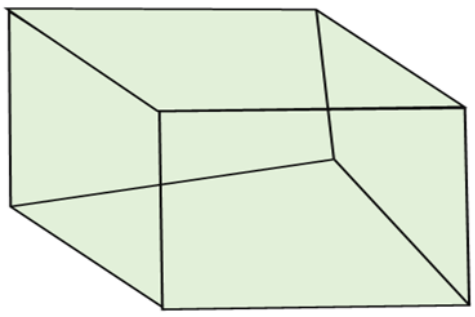
Object Space	Image Space
1. Image space is object based. It concentrates on geometrical relation among objects in the scene.	1. It is a pixel-based method. It is concerned with the final image, what is visible within each raster pixel.
2. Here surface visibility is determined.	2. Here line visibility or point visibility is determined.
3. It is performed at the precision with which each object is defined, No resolution is considered.	3. It is performed using the resolution of the display device.
4. Calculations are not based on the	4. Calculations are resolution base, so

resolution of the display so change of object can be easily adjusted.	the change is difficult to adjust.
5. These were developed for vector graphics system.	5. These are developed for raster devices.
6. Object-based algorithms operate on continuous object data.	6. These operate on object data.
7. Vector display used for object method has large address space.	7. Raster systems used for image space methods have limited address space.
8. Object precision is used for application where speed is required.	8. There are suitable for application where accuracy is required.
9. It requires a lot of calculations if the image is to enlarge.	9. Image can be enlarged without losing accuracy.
10. If the number of objects in the scene increases, computation time also increases.	10. In this method complexity increase with the complexity of visible parts.

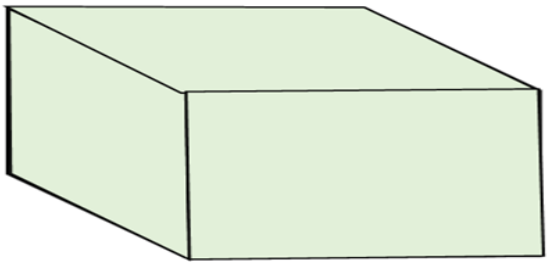
"WORKING TOWARDS BEING THE BEST"

Similarity of object and Image space method

In both method sorting is used a depth comparison of individual lines, surfaces are objected to their distances from the view plane.



Object with hidden line



Object when hidden lines removed

Coherence

It is used to take advantage of the constant value of the surface of the scene. It is based on how much regularity exists in the scene. When we moved from one polygon of one object to another polygon of same object color and shearing will remain unchanged.

Types of Coherence

1. Edge coherence
2. Object coherence
3. Face coherence
4. Area coherence
5. Depth coherence
6. Scan line coherence
7. Frame coherence
8. Implied edge coherence

1. Edge coherence: The visibility of edge changes when it crosses another edge or it also penetrates a visible edge.

2. Object coherence: Each object is considered separate from others. In object, coherence comparison is done using an object instead of edge or vertex. If A object is farther from object B, then there is no need to compare edges and faces.

3. Face coherence: In this faces or polygons which are generally small compared with the size of the image.

4. Area coherence: It is used to group of pixels cover by same visible face.

5. Depth coherence: Location of various polygons has separated a basis of depth. Depth of surface at one point is calculated, the depth of points on rest of the surface can often be determined by a simple difference equation.

6. Scan line coherence: The object is scanned using one scan line then using the second scan line. The intercept of the first line.

7. Frame coherence: It is used for animated objects. It is used when there is little change in image from one frame to another.

8. Implied edge coherence: If a face penetrates in another, line of intersection can be determined from two points of intersection.

Algorithms used for hidden line surface detection

1. Back Face Removal Algorithm
2. Z-Buffer Algorithm
3. Painter Algorithm
4. Scan Line Algorithm
5. Subdivision Algorithm
6. Floating horizon Algorithm

Back Face Removal Algorithm

It is used to plot only surfaces which will face the camera. The objects on the back side are not visible. This method will remove 50% of polygons from the scene if the parallel projection is used. If the perspective projection is used then more than 50% of the invisible area will be removed. The object is nearer to the center of projection, number of polygons from the back will be removed.

It applies to individual objects. It does not consider the interaction between various objects. Many polygons are obscured by front faces, although they are closer to the viewer, so for removing such faces back face removal algorithm is used.

When the projection is taken, any projector ray from the center of projection through viewing screen to object pieces object at two points, one is visible front surfaces, and another is not visible back surface.

This algorithm acts a preprocessing step for another algorithm. The back face algorithm can be represented geometrically. Each polygon has several vertices. All vertices are numbered in clockwise. The normal M_1 is generated a cross product of any two successive edge vectors. M_1represent vector perpendicular to face and point outward from polyhedron surface

$N_1=(v_2-v_1)(v_3-v_2)$
If $N_1.P \geq 0$ visible
$N_1.P<0$ invisible

Advantage

1. It is a simple and straight forward method.
2. It reduces the size of databases, because no need of store all surfaces in the database, only the visible surface is stored.

Back Face Removed Algorithm

Repeat for all polygons in the scene.

1. Do numbering of all polygons in clockwise direction i.e.
   v_1 v_2 v_3.....v_z
2. Calculate normal vector i.e. N_1
   $N_1=(v_2-v_1)*(v_3-v_2)$
3. Consider projector P, it is projection from any vertex
   Calculate dot product
   Dot=N.P
4. Test and plot whether the surface is visible or not.
   If Dot ≥ 0 then
   surface is visible
   else
   Not visible

Z-Buffer Algorithm

It is also called a **Depth Buffer Algorithm**. Depth buffer algorithm is simplest image space algorithm. For each pixel on the display screen, we keep a record of the depth of an object within the pixel that lies closest to the observer. In addition to depth, we also record the intensity that should be displayed to show the object. Depth buffer is an extension of the frame buffer. Depth buffer algorithm requires 2 arrays, intensity and depth each of which is indexed by pixel coordinates (x, y).

Algorithm

For all pixels on the screen, set depth [x, y] to 1.0 and intensity [x, y] to a background value.

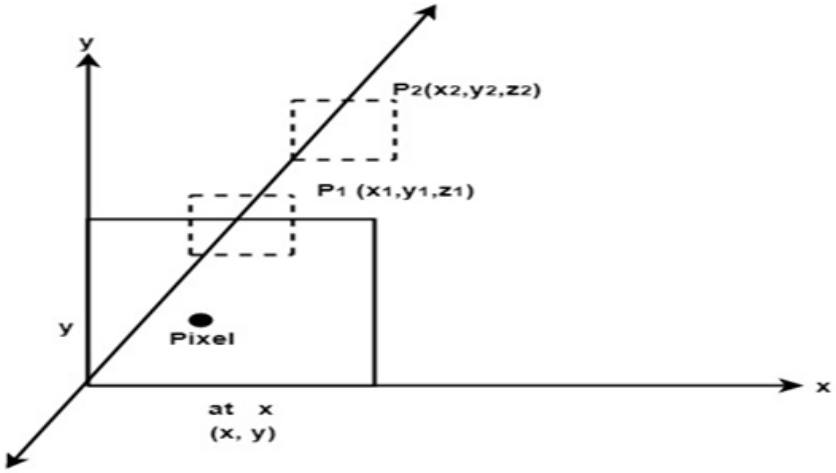
For each polygon in the scene, find all pixels (x, y) that lie within the boundaries of a polygon when projected onto the screen. For each of these pixels:

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

(b) If z < depth [x, y], this polygon is closer to the observer than others already recorded for this pixel. In this case, set depth [x, y] to z and intensity [x, y] to a value corresponding to polygon's shading. If instead z > depth [x, y], the polygon already recorded at (x, y) lies closer to the observer than does this new polygon, and no action is taken.

3. After all, polygons have been processed; the intensity array will contain the solution.

4. The depth buffer algorithm illustrates several features common to all hidden surface algorithms.



5. First, it requires a representation of all opaque surface in scene polygon in this case.

6. These polygons may be faces of polyhedral recorded in the model of scene or may simply represent thin opaque 'sheets' in the scene.

7. The IInd important feature of the algorithm is its use of a screen coordinate system. Before step 1, all polygons in the scene are transformed into a screen coordinate system using matrix multiplication.

Limitations of Depth Buffer

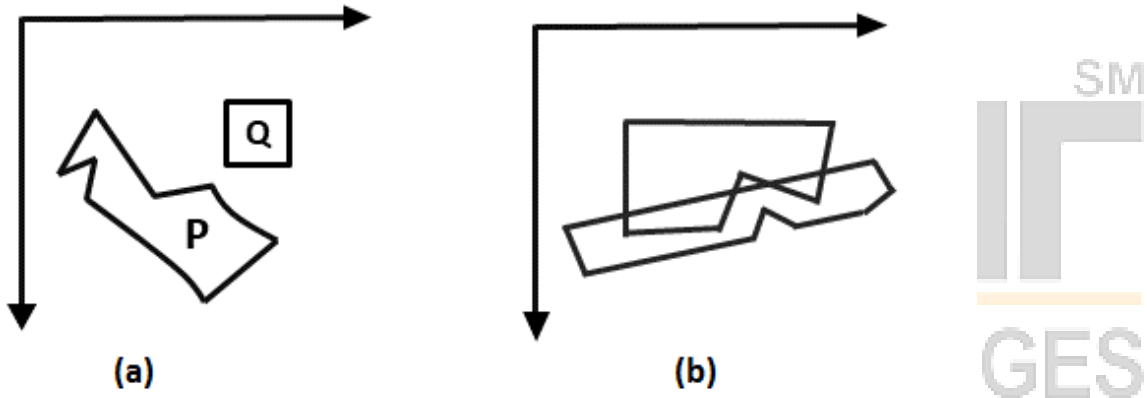
1. The depth buffer Algorithm is not always practical because of the enormous size of depth and intensity arrays.
2. Generating an image with a raster of 500 x 500 pixels requires 2, 50,000 storage locations for each array.
3. Even though the frame buffer may provide memory for intensity array, the depth array remains large.
4. To reduce the amount of storage required, the image can be divided into many smaller images, and the depth buffer algorithm is applied to each in turn.
5. For example, the original 500 x 500 faster can be divided into 100 rasters each 50 x 50 pixels.
6. Processing each small raster requires array of only 2500 elements, but execution time grows because each polygon is processed many times.
7. Subdivision of the screen does not always increase execution time instead it can help reduce the work required to generate the image. This reduction arises because of coherence between small regions of the screen.

Painter Algorithm

It came under the category of list priority algorithm. It is also called a **depth-sort algorithm**. In this algorithm ordering of visibility of an object is done. If objects are reversed in a particular order, then correct picture results.

Objects are arranged in increasing order to z coordinate. Rendering is done in order of z coordinate. Further objects will obscure near one. Pixels of rear one will overwrite pixels of farther objects. If z values of two overlap, we can determine the correct order from Z value as shown in fig (a).

If z objects overlap each other as in fig (b) this correct order can be maintained by splitting of objects.



"WORKING TOWARDS BEING THE BEST"

Depth sort algorithm or painter algorithm was developed by Newell, sancha. It is called the painter algorithm because the painting of frame buffer is done in decreasing order of distance. The distance is from view plane. The polygons at more distance are painted firstly.

The concept has taken color from a painter or artist. When the painter makes a painting, first of all, he will paint the entire canvas with the background color. Then more distance objects like mountains, trees are added. Then rear or foreground objects are added to picture. Similar approach we will use. We will sort surfaces according to z values. The z values are stored in the refresh buffer

Steps performed in-depth sort

1. Sort all polygons according to z coordinate.
2. Find ambiguities of any, find whether z coordinate overlap, split polygon if necessary.
3. Scan convert each polygon in increasing order of z coordinate.

Painter Algorithm

Step1: Start Algorithm

Step2: Sort all polygons by z value keep the largest value of z first.

Step3: Scan converts polygons in this order.
Test is applied

1. Does A is behind and non-overlapping B in the dimension of Z as shown in fig (a)
2. Does A is behind B in z and no overlapping in x or y as shown in fig (b)
3. If A is behind B in Z and totally outside B with respect to view plane as shown in fig (c)
4. If A is behind B in Z and B is totally inside A with respect to view plane as shown in fig (d)

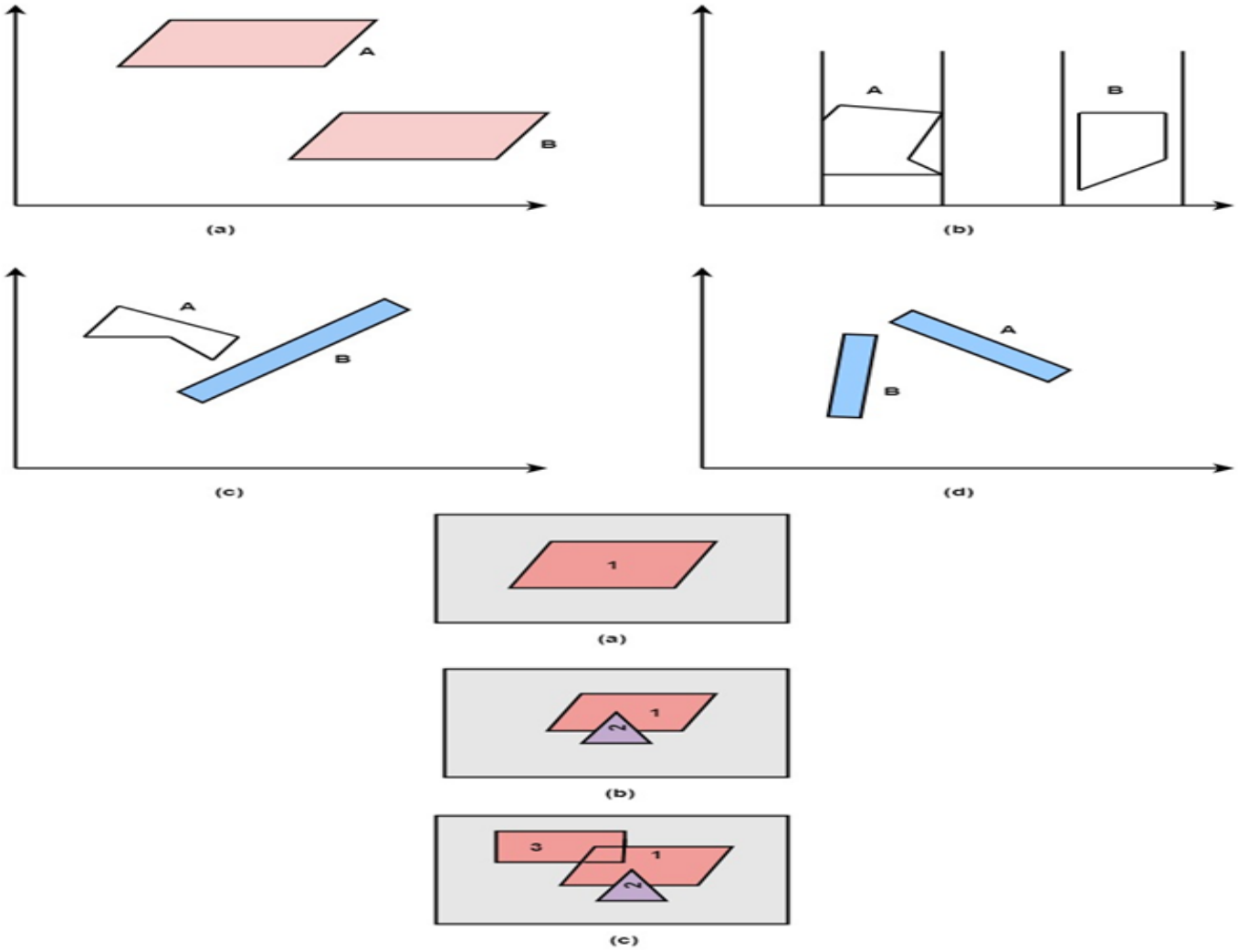
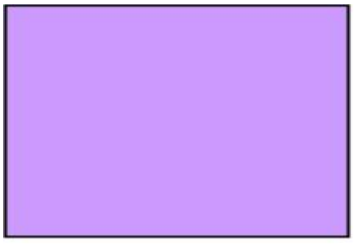


Figure showing addition of surface one by one and then painting done using painter algorithm

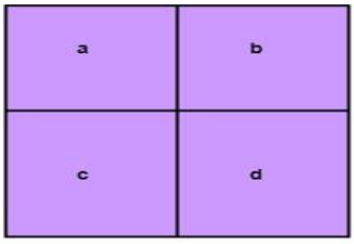
Area Subdivision Algorithm

It was invented by John Warnock and also called a Warnock Algorithm. It is based on a divide & conquer method. It uses fundamental of area coherence. It is used to resolve the visibility of algorithms. It classifies polygons in two cases i.e. trivial and non-trivial.

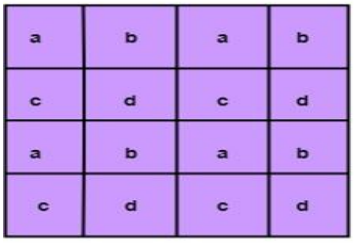
Trivial cases are easily handled. Non trivial cases are divided into four equal subwindows. The windows are again further subdivided using recursion until all polygons classified trivial and non trivial.



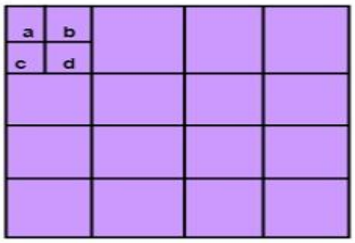
Original area
(a)



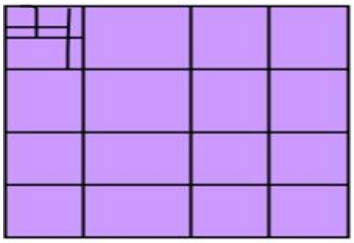
First division or subdivision of area
(b)



Second division
(c)



Third subdivision
(d)



Fourth subdivision
(e)

Classification of Scheme

It divides or classifies polygons in four categories:

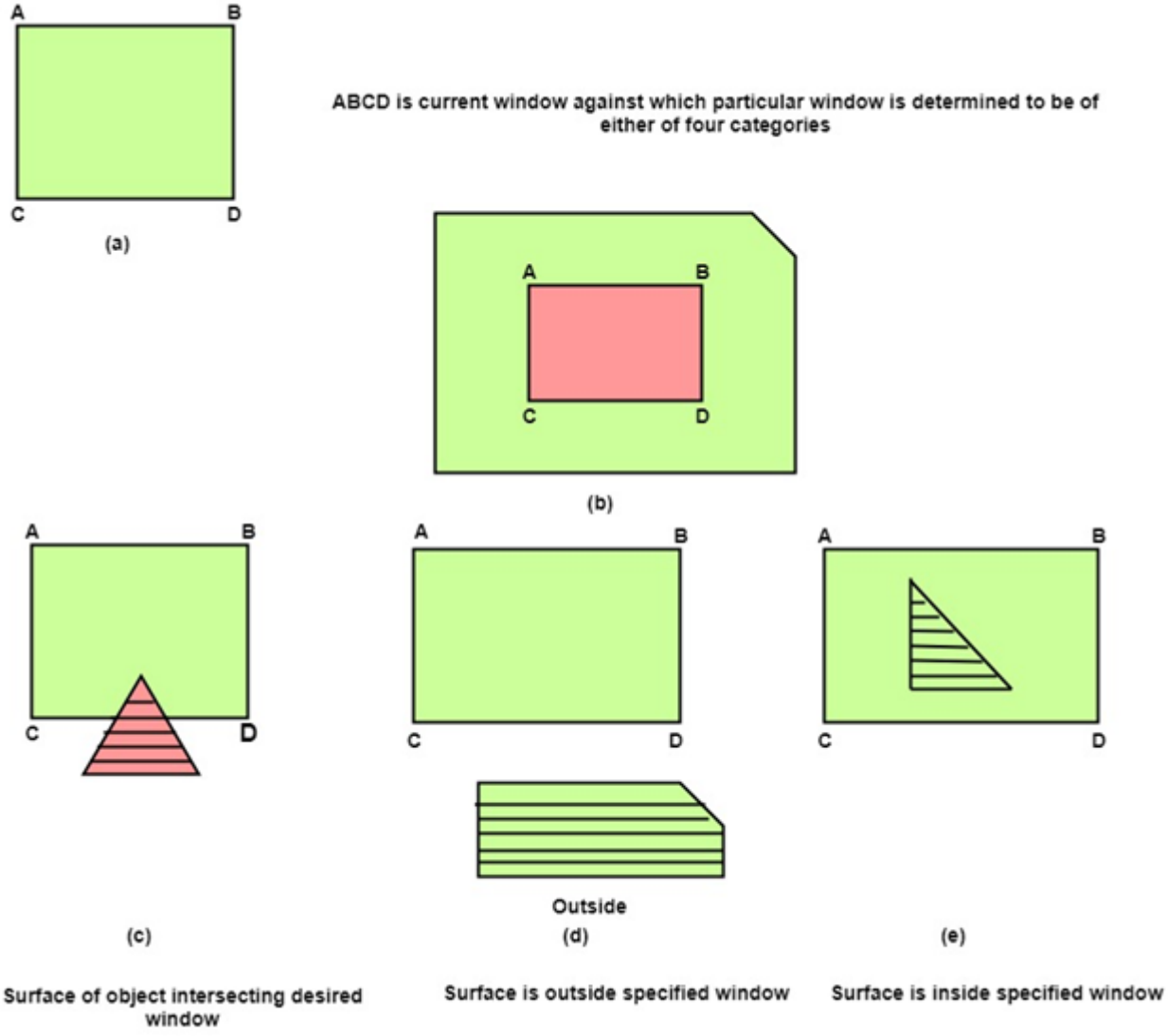
1. Inside surface
2. Outside surface
3. Overlapping surface
4. Surrounding surface

1. Inside surface: It is surface which is completely inside the surrounding window or specified boundary as shown in fig (c)

2. Outside surface: The polygon surface completely outside the surrounding window as shown in fig (a)

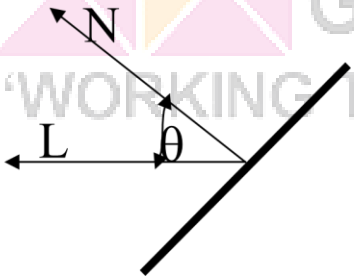
3. Overlapping surface: It is polygon surface which completely encloses the surrounding window as shown in fig (b)

4. Overlapping surface: It is surface partially inside or partially outside the surface area as shown in fig (c)



Illumination Model

- A surface point could be illuminated by
  - ***local illumination***, light directly emitted by light sources
  - ***global illumination***, light reflected from and transmitted through its own and other surfaces
- ***Illumination models*** express the factors which determine the surface color at a given point on a surface and compute the color at a point in terms of both local and global illumination
- ***Diffuse reflection*** also called ***Lambertian reflection*** is a characteristic of dull, matte surfaces like chalk
- Amount of light reflected from a point on the surface is equal in all directions
- The brightness depends on the angle between the direction to the light source L and the surface normal N
- The brightness is independent of the viewing direction
- IL(d)IL(a)+IL(spec)+IL(Trans)



$$I = I_p k_d \overline{N} \bullet \overline{L}$$

- The ambient light is: $I = I_a k_a$
- I_a is the intensity of ambient light

  and k_a is the materials *ambient reflection coefficient* ranging from 0 to 1
- By adding the ambient term to the diffuse reflection the illumination equation is

$$I = I_a k_a + I_p k_d (\overline{N} \bullet \overline{L})$$

Diffuse reflaction is constant over each surface independent of the viewing direction. The intensity at any pt on the surface is given by ambdiff = $k_d 1_a$

Where k_d is diffuse reflecting 1_a intensity of ambient light diffuse reflection is because of ambient light source as well as any other single light source.

Specular reflection is direction dependent when we look at illiminated shiny surface we see a highlight or bright spot t certin viewing direction. This phenomenon is know as specular reflection is given by Ispec = w (Q)Ii cos ^{as} ϕ

When as specular reflection parameter which is determined by type of the surface for shiny surface, it is having large value and for dull surface it is equal to 1.

Is angular direction and according to phong's model this will very as per angle of incidence

w (q) is specular reflection function diffuse reflection model is necessary for illumination of the surface of any type specular reflection is reqmired typically for shiny surfaces. This is large at specific point but diffuse selection is cast over the surface.

Objects other then ideal reflectors exhibit specular reflections over a finite of viewing position around vector R , specular reflection range and dull surfaces have wider reflection range and this range calculation model is developed by phong specular reflection model.

The intensity of specular reflection depends on material properties of the surface and angle of incidence. We can approxemate model as monocromatic specular intensity variations using specular reflection coethicient w (θ) of surface.

W(q) varies in sangel of 0 to 90° As angle of incidence increases w (θ)tends to increase.

The variation of specular intensity with angle of incidence is describe as freshel's laws o reflection.

Ispec = w(θ) I, cos ^{ns} ϕ

Where I_1 is intensity of light source and is veining angle selective to specular reflection R

At q = 90° w (q) = 1 and all incident light is reflected

V unit vector is viewing direction

R specular refln direction

Cos f is dot product or U.R. Ispec = $R_s 1_i$ (V.R) ^{n}S

Where K_s is specular refl ^{n} coefficient

- In case of multiple light sources, the terms for each light source are summed. So for ***m*** light sources the illumination equation is

$$I = I_a k_a + \sum_{i \le i \le m} I_{pi}[k_d(\bar{N} \bullet \bar{L}_i) + k_s(\bar{R}_i \bullet \bar{V})^n]$$

Due to transperancy effect, illumination is

$I=(1-kt)I_{refl}$ + Kt I_{trans}

- Kt -> Transparency Coefficient
- I_{trans} -> trasmitted intensity through surface

Local Illumination Considers light sources and surface properties only

$$I_\lambda = I_a k_a + \sum_{p=1}^{lights} \frac{I_p}{(s_o + k)} [k_d (\bar{N} \bullet \bar{L}) + k_s (\bar{V} \bullet \bar{R})^n] + I_t k_t$$

Application of Illumination model

Gourad Shading Model

Gourad shading model is based on intensity interpolation scheme for rendering polygon surfaces. Intensity values for each polygon are matched with values of adjacent polygons.

The steps for rendering are as:-

1.determine average unit normal vector as each vertex

1.Apply illumination model to calculate vertex intensity.

2. Linearly interpolate the vertex intensities phong shading applies more accurate method for rerdering . It interpolate normal vectors and then apply illumination model.

Phong Shading Model

A polygon surface is rendered using following steps:-

1.determine the average unit normal vector at each vertex

2. Linearly interpolate the vertex normal vector the surface

3. Apply illumination model to calculate pixel intensities at points intensity calculation using an approximation vectors at each point along the scan line produces more accurate results than direct interpolation in gourad shading . It produces more realistic specular high lights.

Mach Band Effect

Highlights on the surface are displayed with anomalous shapes and liner intensity inter polation can cover bright or dark intensity streaks called much bands, appearing on surface.

Constant shading model generates constant intensity throughout the surface. But there is problem of intensities discontinuities for smooth surface display this is not useful for smooth surface display limer interpolation is useful.

Mach band effect is more problematic in case of gourad shading here as in case of phong shading model intensity calculations using an approximated normal vector produces more accurate results than the direct interpolation phong shading model requires more calculation is the only problem. Mach band effect is considerably decreased in case of phong shading than gourad shading or constant shading model

Gourad shading model is based on intensity interpolation method. Calculates different intensities for different pixels on surface area. Low intensity light areas and bands are created producing mach band effect.

Highlights on the surface displayed with anomalous shapes and can not create smooth fading effect.

Phong shading model is based or normal vector interpolation on it user more accurate method for rendering polygon surface. Same surface is subdivided into greater no. of polygon faces and normal vectors are interpolated.

Mathematically interpolated normal vector is going to generate accurate intensities calculations than direct interpolation.

This reduces mach band effect and try to produce more realistic appearance.

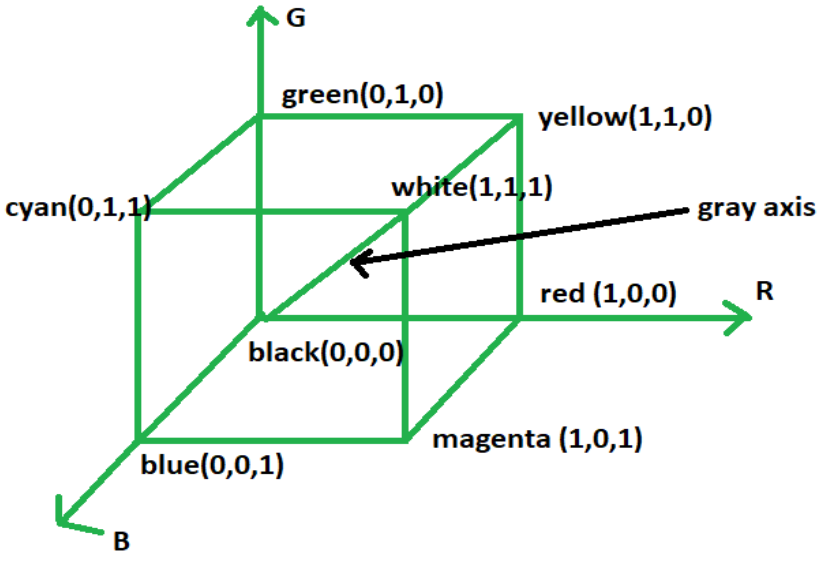
Phong shading increases calculatiues as compased to gourad shading and also difficult to implement it taken around six or seven times longer then gourad shading model.

Color model

The RGB color model is one of the most widely used color representation method in computer graphics. It use a color coordinate system with three primary colors:

```
R(red), G(green), B(blue)
```

Each primary color can take an intensity value ranging from 0(lowest) to 1(highest). Mixing these three primary colors at different intensity levels produces a variety of colors. The collection of all the colors obtained by such a linear combination of red, green and blue forms the cube shaped RGB color space.



0

The corner of RGB color cube that is at the origin of the coordinate system corresponds to black, whereas the corner of the cube that is diagonally opposite to the origin represents white. The diagonal line connecting black and white corresponds to all the gray colors between black and white, which is also known as **gray axis**.
In the RGB color model, an arbitrary color within the cubic color space can be specified by its color coordinates: (r, g.b).

Example:

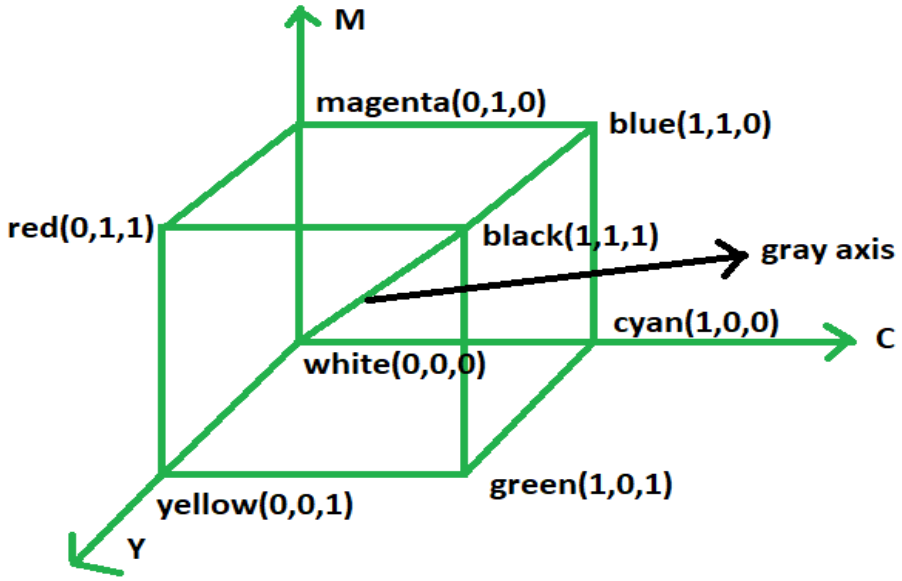
```
(0, 0, 0) for black, (1, 1, 1) for white,
(1, 1, 0) for yellow, (0.7, 0.7, 0.7) for gray
```

Color specification using the RGB model is an **additive process**. We begin with black and add on the appropriate primary components to yield a desired color. The concept RGB color model is used in **Display monitor**. On the other hand, there is a complementary color model known as **CMY color model**. The CMY color model use a **subtraction process** and this concept is used in the **printer**.
In CMY model, we begin with white and take away the appropriate primary components to yield a desired color.

Example:
If we subtract red from white, what remains consists of green and blue which is cyan. The coordinate system of CMY model use the three primaries' complementary colors:

```
C(cray), M(magenta) and Y(yellow)
```

GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"



The corner of the CMY color cube that is at (0, 0, 0) corresponds to white, whereas the corner of the cube that is at (1, 1, 1) represents black. The following formulas summarize the conversion between the two color models:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \qquad \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Difference between additive color model and subtractive color model:-

ADDITIVE COLORS
VERSUS
SUBTRACTIVE COLORS

Criterion	Additives	Subtractive
Definition	Occurs with the simultaneous action of various color 'irritants' on the eye.	Not mixing of color 'irritants', but creating color by subtraction
Basic colors	red, green, and blue	cyan, magenta and yellow
Color combinations	green + red = yellow, blue + red = magenta, blue + green = cyan	yellow + magenta = red, yellow + cyan = green, magenta + cyan = blue
Systems	RGB	CMYK

DB Difference Between.net

Generic XYZ Color model

The three standard primaries are imaginary colors. The set of CIE primaries is referred as XYZ color model where X,Y,Z three vectors represented by 3-D additive color space.

Any color C_x is XX+YY+ZZ where X,Y,Z represents amounts of standard primaries.

To normalize the amount is against luminance (X+Y+Z) Normalized amount are calculated as.

X= X/(X+Y+Z) Y=Y/(X+Y+Z)

Z= Z/(X+Y+Z)

Where X+Y+Z=1

Parameters X and Y are called as chromaticity values. A complete description of a color is with three values X,Y and Y

` X=(X/Y)Y Z(Z/Y)Y

Where Z = 1-X-Y

Any color can be obtained by using the two values X and Y in hue and saturation.

CIE chromaticity diagram is represented as tongue shaped curve points along the curve are the "pure" colors in electromagnetic spectrum auording to wavelength from the red end to violet end of the spectrum.

Point C in diagram corresponds to the white light position Illuminate CF is standard approx . for average day light luminance values are not available in chromacity diag. The chromacity diagram is useful for comparing color garents for different primaries

1.Identifying complementary colors

2.Determining dommant wavelength and purity of a given color

Color gamats are presented on the chromaticity diagram as straight line segments or as polygons.

Global illumination

– The notion that a point is illuminated by more than light from local lights; it is illuminated by all the emitters and reflectors in the global scene

Rendering equation

- Jim Kajiya (Current head of Microsoft Research) developed this in 1986
- I(x, x') is the total intensity from point x' to x
- g(x, x') = 0 when x/x' is hidden and $1/d^2$ otherwise (d = distance between x and x')
- e(x, x') is the intensity emitted by x' to x
- r(x, x',x'') is the intensity of light reflected from x'' to x through x'
- S is all points on all surfaces
- The light that hits x from x' is the direct illumination from x' and all the light reflected by x' from all x''
- To implement:
  - Must handle recursion effectively and shadowing model effectively
  - Must support diffuse and specular light

Computer Visualisation

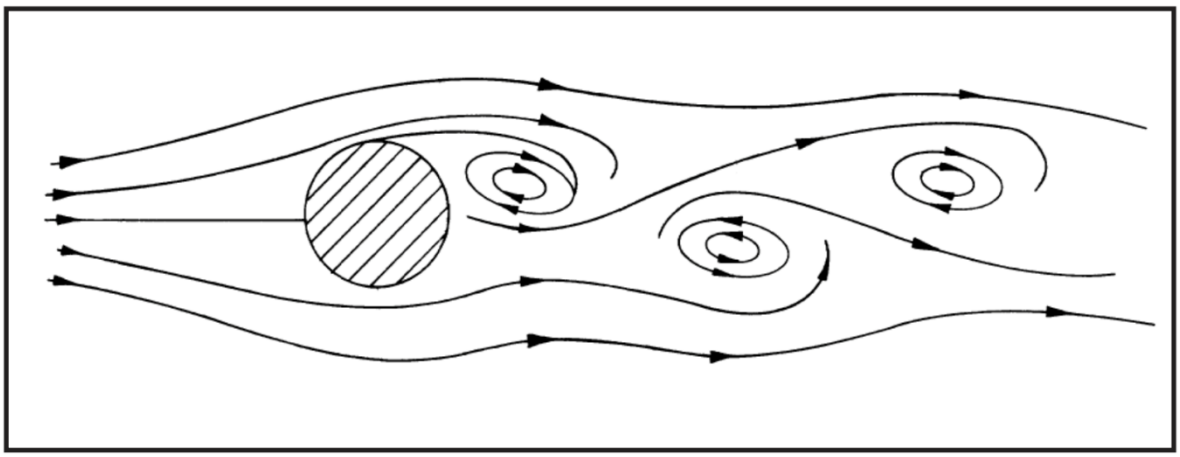
Visualization is the process of representing data graphically and interacting with these representations in order to gain insight into the data. Traditionally, **computer graphics** has provided a powerful mechanism for creating, manipulating, and interacting with these representations

Scientific visualisation is gaining importance in all domains of medical science, data analytics, computer vision, neural network, deep learning, weather and satellite imagery. It draws from and contributes to work in algorithms, human perception, art, **animation**, **computer** vision, and image processing.

With visualisation, Computer graphics technology enables creation of synthetic 3D or 2D environments and user interfaces for different industries: entertainment (movies, video games), simulation (medical, military), scientific visualization and consumer electronics (mobile phones, media players).

Although in the real world, space and time are continuous, in a numerical simulation we generally use discrete space and time to describe the system. Commonly, simulation output contains a set of sample points, data values associated with each sample and an interpolation rule. Generally, output data can be categorized by sampling patterns, or by the dimension of data values associated with each sample point.Typical sampling patterns include regular (Cartesian) grid, curvilinear(structured) grid and unstructured (irregular) grid.

For ex. How to create following diagram in computer graphics and visualisation



Dimensionality of value

Data can be classified as scalar, vector, or tensor field by the dimension of associated value with each sample.

A scalar field associates a scalar to every sample point, and is often used to indicate some physical properties such as temperature, pressure and density. For a vector field, the quantity is specified by a vector, giving a direction as well as a magnitude, such as velocity and magnetic field.

Interactive visualization of the resulting time-varying data allow scientists to freely explore how selected properties of the data change over time, and gain insight into the relationships between values that might not be apparent in the raw data and this is where visualisation helps the researchers.

For ex. For physics

The resulting data sets share some common characteristics with data from other scientific simulations: • Data are sampled on a curvilinear grid.

• For any time instant, several physical properties are measured: –

Scalar field: density, temperature.

Vector field: velocity, magnetic field. • Data are captured at thousands of time steps

Basic unit of visualisation is voxel.

A **voxel** represents a value on a regular grid in three-dimensional space. As with pixels in a **2D** bitmap, **voxels** themselves do not typically have their position (coordinates) explicitly encoded with their values. ... **Voxels** are frequently used in the **visualization** and analysis of medical and scientific data (e.g. GIS).

2D scalar visualisation techniques

1. **Color mapping** is a common **scalar visualization** technique that **maps scalar** data to **colors**, and displays the **colors** on the computer system. The **scalar mapping** is implemented by indexing into a **color** lookup table. **Scalar** values then serve as indices into this lookup table. The lookup table holds an array of **colors**.
2. **Isosurface** extraction is a powerful tool for investigating volumetric **scalar** fields. An isosurface in a **scalar** volume is a **surface** on which the data value is constant, separating regions of higher and lower value.
3. The characteristic of **direct volume rendering** is the **direct** mapping of voxels on pixels of a **2D** image plane. It allows for the "global" representation integrating physical characteristics, but prohibits interactive display due to its numerical complexity in general.
4. **Ray casting** is the use of **ray**–surface intersection tests to solve a variety of problems in 3D **computer graphics** and computational geometry. The term was first used in **computer graphics** in a 1982 paper by Scott Roth to describe a method for

rendering constructive solid geometry models. It computes 2D images from 3D volumetric data sets (3D scalar fields).

2D 3D Vector Visualisation

You can **visualize** a **vector field** by plotting **vectors** on a regular grid, by plotting a selection of streamlines, or by using a gradient color scheme to illustrate **vector** and streamline densities. You can also plot a **vector field** from a list of **vectors** as opposed to a mapping

Vector fields represent fluid flow (among many other things). They also offer a way to visualize functions whose input space and output space have the same dimension.

Vector notation: V=i+j+k

i is the unit vector in the x-direction

- j is the unit vector in the y direction
- k, is the unit vector in the z-direction

Methods used

1. Streamline: A **streamline** is a line that is tangent to the velocity **vector** at every point. **Streamlines** are traditionally used. in flow **visualizations** as they give the direction and orientation of the flow at each point along the line. In the. simplest case, we define a stationary 2D **vector field** as a map v : R.
2. Loopable animation
3. Generating a Random Velocity Field
4. Vector Glyphs : **Glyphs** are graphical objects that are used to represent information at points within a model.
5. **Non-spatial data** (also called attribute or characteristic **data**) is that information which is independent of all geometric considerations. o For example, a person's height, mass, and age are **non-spatial data** because they are independent of the person's location. It is necessary to create a communication channel that could quickly and efficiently transfer the information from the **data** to the user. By using visual elements like charts, graphs, and maps, **data visualisation** is an accessible way to see and understand trends, outliers, and patterns in **data**.
6. **A data set** consisting of two or more than two variables is referred to as **multivariate dataset**. It can be visualised with scatterplots, bar plots visualization techniques.and structured and unstructured techniques of graphs and trees.
7. **Perceptual issues** With the increase in the amount and dimensionality of scientific data collected, new approaches to the design of displays of such data have become essential. The designers of visual and auditory displays of scientific data seek to harness perceptual processes for data exploration. In these cases, even with good aesthetic qualities and good data, the graph will be confusing or misleading because of how people perceive and process what they are looking at. It is important to understand that these elements, while often found together, are distinct from one another.

8. **Congnitive Foundation:** The theoretical **cognitive** process of **visualization**. is directly related to creative visualisation. In relation to **cognition**, learning involves the internal (psychological) and external (physical) domains. Learning therefore involves processing of information as a way of interaction between these two domains. The cognitive aspect is different worldviews or frameworks serve as the foundation of understanding visual perception and cognition. Perception and cognition can be assumed as computations properties of interaction between active agents and lies in creataive tools for visualization.

Evaluating data visualization system is difficult however some pattern based approach is required for evaluation. It can be broadly classified as

Quantitative evaluation focuses on collecting performance measurements, for example on time and errors, that can be analyzed using statistical methods. Qualitative evaluation, on the other hand, collects more in-depth and free-form data, such as observations, notes, transcripts, etc, and is often used for more exploratory or explanatory purposes.

The various patterns seen in visualisation are:

- Exploration: Patterns concerned with exploring the design space of the evaluation study. Are we using the right independent and dependent variables? Are we confident that the study is appropriate? Are we asking the right questions?
- Control: Mechanisms for controlling an evaluation study design to achieve high internal validity
- Generalization: Is this study grounded in real-world practices How trustworthy are these results?
- Validation: Finding the right balance, calibration, and parameters for an evaluation to save time, resources, and money. Are we testing the right thing in the right way?
- Presentation: Reporting the results of an evaluation correctly and economically. Are the results presented in a way where they can be easily understood?

LNCT GROUP OF COLLEGES "WORKING TOWARDS BEING THE BEST"

Bibliography

Donald Hearn and M.P. Becker "Computer Graphics" Pearson Pub.

Schaum's series " Computer Graphics" TMH

Theories in Information Visualization: What, Why and How, Zhicheng Liu* and John Stasko§

Visualisation research papers