2-D Geometrical Transforms and Viewing
Part-3

Content:

1. Concept of projection
2. Terms Related to Projection
3. Categories of projection
4. Vanishing points
5. Graphical Kernel System(GKS)

CONCEPT OF PROJECTION

It is a process of representing a three dimensional object or scene into two dimensional medium. That, is projection is nothing but a shadow of the object.

IMPORTANT TERMS RELATED TO PROJECTION

Centre of Projection: The point from where projection is taken. It can either be light source or eye position.

Projection Plane: The plane on which projection of the object is formed.

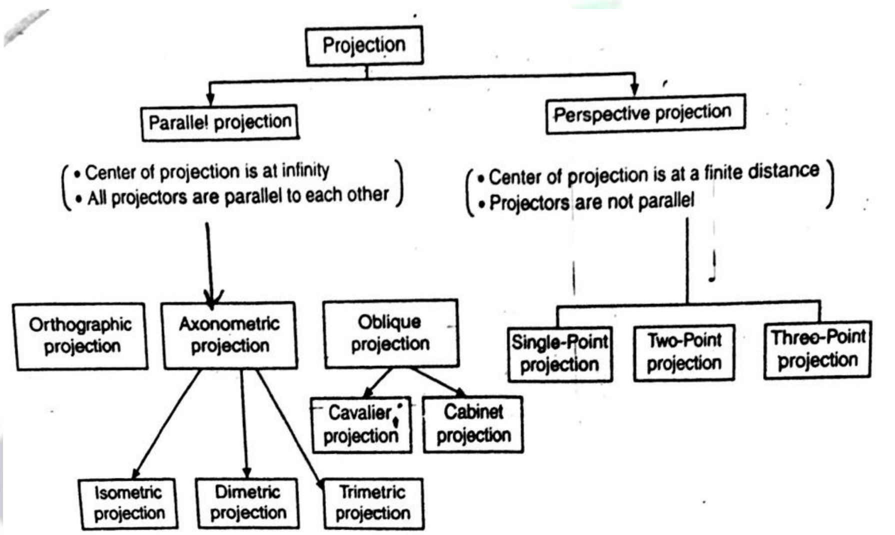
Projection: Lines emerging from center of projection and hitting the projection plane after passing through a point in the object to be projected.

Thus, the plane geometric projection or simply projection of the objects are formed by the intersection of lines called projectors, on a plane called the projection plane. Projectors are lines from an arbitrary point called the center of projection, through each point in an object. If the center of projection is at a finite distance in three dimensional space, the result is a **perspective projection.** if the center of projection is at infinity, all the projectors are parallel and the result is a **parallel projection.** thus, projections can be divided into two basic classes: **perspective** and **parallel.**

a. **Parallel projection:** In a parallel projection, coordinate positions are transformed to the view plane along parallel lines. These are **linear** transforms that are useful in blueprints to produce scale drawings of three-dimensional objects.

b. **Perspective Projection:** For a perspective projection object positions are transformed to the view plane along lines that converge to a point called to **center of projection.** The projected view an object is determined by calculating the intersection of the projection lines with the view plane.

CATEGORIES OF PROJECTION



PARALLEL PROJECTION

A parallel projection is formed by extending parallel lines from each vertex of the object until they intersect the plane of the screen. The point of intersection is the projection of the vertex. Then we connect the projected vertices by line segments which correspond to connection on the original object.

The plane on which we are taking projection is called as view plane. If we want to represent a 3D, object on 2D plan, the most simple way is to discard the z-coordinates. Our special case of discarding the z-coordinates is the case when the screen or viewing surface is parallel to the xy plane and the line of projection are parallel to the z-axis.

A parallel projection preserves relative properties of objects in x and y direction.

ORTHOGRAPHIC PROJECTION

It is a part of the parallel projection in which the center of projection lies at infinity. In orthographic projection, the projectors are perpendicular to the projection plane and hence true size and shape of a single plane force of an object is obtained.

Orthographic projections do not change the length of line segments which are parallel to projections plane. Other lines are projected with reduced length.

It is commonly used for engineering drawing. It is the projection on one of the coordinate planes i.e. x = 0, y = 0 or z = 0.

The matrix for projection onto the x= 0 plane is

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

In this matrix, we see that the first column, i.e. the x-column is all zero. The effect of the transformation is to see the x-co-ordinate of the position vector to zero. Similarly the matrixes for projection onto the y = 0 and z = 0 plane are

$$P_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } P_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

AXONOMETRIX PROJECTIONS

They are orthographic projections in which the direction of projection is not parallel to any of the three principal axis. The construction of an axonometric projection is done by using rotation and translation of manipulate the object such that atleast three adjacent face are shown. The result is projected at infinity onto one of the co-ordinate planes (usually z = 0 plane) from the centre of projection. An axonometric projection shown its true shape only when a face is parallel to the plane of projection. Here the parallel lines are equally fore shortened.

The sub-categories of axonometric projections are:

a. **Isometric Projection:** The direction of projection makes equal angles with all three principal axis.
The problem associated in *diametric projection* is removed in isometric projection. So, the most commonly used axonometric projection is isometric projection. In isometric projection all three foreshortening factors are kept equal. The projection plane makes equal angles with every principal axis.

b. **Diametric Projection:** The direction of projection makes unequal angels with exactly two of the principle axis.
In diametric projection, two foreshortening factor are equal and third arbitrary. It is constructed by a rotation about a y-axis though an angle ϕ followed by a rotation about the x-axis through an angle θ and projection at infinity onto the z = 0 plane i.e.

c. **Trimetric Projection:** The direction of projection makes unequal angels with the three principal axis.
It is formed by arbitrary rotation in arbitrary order about any or all of the co-ordinate axis followed by parallel projection onto the z = 0 plane. For each projected principal axis, the foreshortening ratios are different.

OBLIQUE PROJECTION

When the angle between the projectors and the plane of projection is not equal to 90^o then the projection is called oblique projection. Thus, projection are not-perpendicular to view plane. This type of projection is seen in form of shadow of any object due to sunlight. Thus in this type of projection normally the shadow is displayed and body is not displayed.

Some common subcategories of oblique projections are:

a. Cavalier projection
b. Cabinet projection

The Cavalier Projection: Is obtained when the angle between the oblique projection and the plans of projection is 45^o and the foreshortening factors for all three principal directions are equal. In cavalier projection the resulting figure is thicker.

It is a form of oblique projection in which the projection lines are presumed to make a 45^o horizontal angle with the plane of projection.

The Cabinet Projection: is used to correct he distortion that is produced by cavalier projection. As oblique projection for which the foreshortening factor for edge perpendicular to the plane of projection is one-half, is called a cabinet projection. For a cabinet projection, the angle between the projectors and the plans of projection is $\cot^{-1}(1/2) = 63.43^o$.

VANISHING POINTS

Perspective projection generates realistic views but does not preserve relative proportions of object dimensions. Projections of distant objects are smaller than the projections of objects of the same size that are close to the projection plane (or center of projection). This feature of perspective projection is known as **perspective foreshortening.**

Another characteristic feature of perspective projection is the illusion that after projection certain set of parallel lines appear to meet at some point on the projection plane. These points are called **Vanishing points.** Each such set of projection parallel lines will have a separate vanishing point and in general, a scene can have any number of vanishing points depending on how many sets of parallel lines there are in the same.

The set of lines that are parallel to one of the principal axis of an object is referred to as a **principal vanishing point or axis vanishing point.** We control the number of principal vanishing points (one, two or three) with the orientation of the projection plane and perspective projections are accordingly classified as

- One-point projection
- Two-point projection
- Three-point projection

Thus, depending on the number of principal axis intersecting the plane of projection, the number of principal vanishing points on that plane of projection vary from a minimum of one to a maximum of three.

Single point perspective transformation: One-point perspective projection occurs when only one principal axis intersects the plane of projection. there are three types of single point perspective transformation.

- **When projectors are located at x-axis, it is given by**

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix}1 & 0 & 0 & p\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

- **When projectors are located at y-axis, it is given by**

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **When projectors are placed at z-axis, it is given by**

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Two-point perspective transformation: Two point or two principal vanishing point perspective projection occurs when the plane of projection intersects exactly two of the principal axis.

The single point perspective transformation unfortunately does not provide an adequate view of the 3D shape of an object so, we develop some more complex perspective transformation i.e. two point perspective transformation as

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Three-point perspective transformation: It occurs when the projection plane intersects all three of the principal axis i.e. no principal axis is parallel to the projection plane. The requirement is to reconstruct the shape of a 3-D object, because to draw a cube it is necessary to know the coordinate of the three shape of a 3-Dobject, because to draw a cube it is necessary to know the coordinate of the three face. The matrix representation of three-point perspective transformation is

$$\begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

GRAPHICAL KERNEL SYSTEM (GKS)

WHAT IS GKS (THE GRAPHICAL KERNEL SYSTEM)

It is a document produced by the International Standards Organization (ISO), which defines a common interface to interactive computer graphics for application programs. It has been designed by a group of experts representing the national standards institutions of most major industrialized countries. The standard gives functional specifications for some 200 subroutines which perform graphics input and output in a device independent way. Application program can thus move freely between different graphics devices and different host computers. For the firs t time graphics programs have become genuinely portable.

GKS PRIMITIVES

The objective of the GKS, is the construction and manipulation of picture (in a way the does not depend on the computer or graphical device being used) such pictures differ from simple line graphs (to illustrate experimental results, for example) to, engineering drawings, to integrated circuit layouts (using color to differentiate between layers), to image representing medial data (from computerized tomographic (CT) scanners) or astronomical data (from telescopes) in gray scale or color. Every picture must be described to GKS, so that they may be drawn.

In GKS, pictures are considerd to be constructed from a number of basic building blocks. These basic building blocks, or primitives as they are called, are of a number of types each of which can be used to describe a different component of a picture. The five main primitives in a GKS are:

1. **Polyline:** It draws a sequence of connected line segments.
2. **Polymarker:** It marks a sequence of points with the same symbol
3. **Fill area:** which displays a specified area.
4. **Text:** which draws a string of characters.
5. **Celi array:** which displays an image composed of a variety of a clolors or gray scales

Polylines

The function for drawing line segments is called polyline. It function takes an array of X-Y coordinates and draws line segments connecting them. The attributes control the appearance of a polyline are:

- Linetype, It controls whether the polyline is drawn as a solid, dashed, dotted, or dash-dotted.
- Linewideth scale factor, It controls how thick the line, is.
- Polyline color index, It controls what color the line, is.

The main line drawing primitive of GKS is the polyline, which is generated by calling the function:

POLYLINE (N, XPTS, YPTS)

Where XPTS and YPTS are arrays giving the N points (XPTS(I), YPTS(I) to (XPTS(N), YPTS(N)). They generated contain of N-1 line segments joining adjacent points starting with the first point and ending with the last.

Why was polyline chosen as the basic line drawing primitive in GKS? If we examine actual graphical devices, we can see that there are many ways of describing line segments. Incremental plotters require each individual increment o the approximated line segment to be specified. Other graphical devices rely on the concept of a current point only one end of each line segment need be specified and a line segment is drawn from the current point to the specified end point, which them itself becomes the current point. Yet other graphical devices expect a connected sequence of line segments to be specified.

Polymarkers

Instead of drawing lines through a set of points, we may wish just to mark the set of points, GKS provides the primitive **polymarker** to do just this. The GKS polymarker function allows us to draw marker symbols contered at coordinate points that we specify.

A polymarker is generated by the function.

POLUMAKRER (N, XPTS, YPTS)

Where the arguments are the same as for the polyline function, namely XPTS and YPTS are arrays giving the N points (XPTS(I), YPTS(I) to (XPTS(N), YPTS(N)).

The attributes that control the appearance of polymarkers are:

- Marker, it specifies one of five standardized symmetric characters to be use for the marker. The five nature are dot, plus, asterisk, circle, and cross.
- Marker size scale factor, it controls how large each marker is (except for the dot marker).
- Polymarker color index, it specifies what color the marker is.

Text

It has a text primitive, which is used to title pictures or place labels on them as appropriate. Invoking the function may generate a text string:

TEXT (X, Y, STRING)

Where (X, Y) is the text position and STRING is a string of characters.

Text is more complicated than the other primitives that we have examined. Everybody is used to good quality text in books whether it is the printed text of the books itself or text within the context of diagrams. It is printed at different sized, in different fonts, at different orientations and at different spacing. Graphics devices, are often not good at text; indeed some are not capable of text at all. Those do often only have a restricted number of sizes. One or perhaps tow orientations, and a single font. The GKS text function allows us to draw a text string at a specified coordinate position. The attributes control the appearance of text are:

- Text font and precision, it specifies what text font should be used for the characters and how precisely their representation should adhere to the settings of the other text attributes.
- Character expansion factor, it controls the height-to-width ratio of each plotted character.
- Character spacing, it specifies how much additional white space should be in serted between characters in a string.
- Text color index, it specifies what color the text string should be.
- Character height, it specifies how large the characters should be.
- Character up vector, it specifies at what angle the text should be drawn.
- Text path, it specifies in what direction the text should be written (right, left, up or down)
- Text alignment, it specifies vertical and horizontal centering options for the text string.

FILL AREA

GKS provides a fill area function to satisfy the application needs which can use the varying device capabilities. Defining an area is a fairly simple extension of defining a polyline. An array of points is specified which defines the boundary of the area. If the area is not closed (i.e. the first point is not the same as the last point), the boundary is the polyline defined by the points but extended to join the last point to the first point. Invoking the function may generate a fill area:

Fill area (N, XPTS, YPTS)

Where as usual XPTS and YPTS are arrays giving the N points (XPTS(I), YPTS(I) to (XPTS(N), YPTS(N)).

The GKS fill area function allows us to specify a polygonal shape of an area to be filled with various interior style. The attributes that control the appearance of fill areas area:

- Charge area interior style, which specifies how the polygonal area should be filled with solid colors or various hatch patterns, or with nothing, that is, a line is drawn to connect the points of the polygon, so we get only a border.
- Charge area style index. If the fill area style is hatch, this index specifies which hatch pattern is to be used: horizontal lines; vertical lines; left slant lines; right slant lines horizontal and vertical line; or left slant and right slant lines.

Charge area color index, which specifies the color of the fill patterns or solid areas.