PGScIT (S4) 02
Exam. Code: M4CGR

Computer Graphics

SEMESTER-IV

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

BLOCK-2



KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakati Mahanta, Gauhati University

Prof. (Retd.) Pranhari Talukdar, Gauhati University

Dr. Jyotiprakash Goswami, Assam Engineering College

Prof. Sikhar Kumar Sarma, Gauhati University

Programme Coordinator

Dr. Ridip Dev Choudhury, KKHSOU

Course Coordinator

Dr. Tapashi Kashyap Das, KKHSOU

Dr. Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
8,12	Mrs. Jonalee Barman Kakati,NERIM
9,10,11,13,14	Ms. Mala Ahmed,GIMT
15	Dr.Utpal Barman,Kaziranga University

Editorial Team

Content : Mr. Dipankar Dutta, NERIM,

Structue, Format & Graphics: Harekrishna Deka, KKHSOU

August, 2022
ISBN NO :

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international): http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Headquarters : Patgaon, Rani Gate, Guwahati - 781017
City Office : NH 37,Resham Nagar,Khanapara,Guwahati-781022; Web: www.kkhsou.in

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
Computer Graphics
DETAILED SYLLABUS

		Page No.
UNIT 8 :	**CLIPPING ALGORITHMS** Clipping,Line Clipping :Cohen-Sutherland Algorithm,Cyrus-beck line clipping algorithms,Polygon Clipping ,Sutherland –Hodgeman algorithm	**137-153**
UNIT 9 :	**3-D OBJECT REPRESENTATION** 3-D Object Representation,Polygon surfaces,Quadric surfaces,Spline representation,Basic illumination models,Polygon rendering methods	**154-167**
UNIT 10 :	**BEZIER AND B-SPLINE CURVES** Hermite Curve ,Bezier Curve and B-Spline Curves :Bezier Curve,B-Spline Curve,Bezier and B-spline surfaces:Bezier Surface,B-Spline Surface	**168-177**
UNIT 11 :	**PROJECTIONS** Projection,Projection Classification,Parallel projections,Perspective projections,Hidden Layer Projection	**178-187**
UNIT 12 :	**3-D GEOMETRIC TRANSFORMATIONS** Translation,Rotation,Scaling,Reflection and Shearing transformation, Composite transformation	**188-205**
UNIT 13 :	**3-D VIEWING** Viewing pipeline,Viewing coordinates,View volume ,General projection transforms ,Clipping	**206-215**
UNIT 14 :	**BASICS IN ANIMATION** Animation Basic,Computer Animation,Principles of Animation,Types of Animation,Animation Software and Hardware,Common Terms in Animation	**216-225**
UNIT 15 :	**ANIMATION DESIGNING** Design of Animation sequence,Morphing, Application of Computer Animation, Future of Animation	**226-232**

BLOCK INTRODUCTION

This block comprises the following eight units :

Unit - 8 gives general overview of the concepts on Clipping Algorithms.

Unit - 9 is concerned with 3 - D object trepresentation.

Unit - 10 deals with the concepts of Bezier and B-spline curves.

Unit - 11 concentrates on Projections.

Unit - 12 introduces concepts on 3-D Geometric transformations.

Unit - 13 includes the basic concepts of 3 - D viewing.B

Unit - 14 introduces concepts on Basics in animation.

Unit - 15 includes the basic concepts of Animation designing.

UNIT 8: CLIPPING ALOGRITHMS

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Clipping

8.4 Line Clipping

- 8.4.1 Cohen-Sutherland Algorithm
- 8.4.2 Cyrus-beck line clipping algorithms

8.5 Polygon Clipping

- 8.5.1 Sutherland -Hodgeman algorithm

8.6 Let Us Sum Up

8.7 Further Reading

8.8 Answer to Check Your Progress

8.9 Model Questions

8.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Understand the idea of clipping in graphics.
- Learn about line and polygon clipping.
- Know how the line clipping algorithms works.
- Learn the Sutherland -Hodgeman polygon clipping algorithm

8.2 INTRODUCTION

In the previous unit we have learnt about the 2D viewing which is a transformation process of real world objects into position point which is relative to the viewing volume. The viewing functions were also discussed.

Often a time in computer graphics or a drawing, we may be interested in a portion of the picture or a scene instead of the whole picture. This portion is the region of interest. In such a situation, we need a method to display only those portions of the drawing that are of immediate interest. This is just like looking at a large image through a window. The process of

selecting and viewing the picture is called windowing. Thus, there are two region that we have - visible and non visible. The part we are interested in the picture should be visible. The part that is not the region of interest should not be visible. This is called ***clipping***. In this unit, we will introduce you to clipping in computer graphics. Here we will find how line clipping and polygon clipping are done, along with algorithms.

8.3 CLIPPING

Picture consists of objects having curves and boundaries. Line segments and polygons usually make up the surface curves and the boundaries. Polygons are also formed by line segments. The word 'Clipping' refers to the removal of objects, lines, or line segments that are outside the viewing pane. Comninos [2006] describes clipping in the following, more general, way:

'**Clipping** is a process that subdivides each element of a picture, which is to be displayed, into its visible and invisible parts. In simple words, any process that identifies parts of a picture that lie either inside or outside a specified region is called clipping algorithm. This allows discarding the invisible parts of the picture and keeps only the parts that are to be considered. Clipping may be internal or external. An internal clipping removes parts of a picture outside a given region and an external clipping removes parts inside a region.

The clipping process can be applied to a variety of graphics primitives such as lines, points, polygons and curves. Clipping can be performed with respect to a clipping boundary, which may be a convex or concave polygonal boundary.

The specified region of the scene is called a clip window or simply window. Clipping is used to extract parts of a defined scene for viewing, anti-aliasing lines or object boundaries, identifying visible surfaces in three-dimensional views and many other applications.

The clip window can be a general polygon or a window that have curved boundaries, depending on the type of the application. In this unit,

we would be considering only rectangular shaped windows.

Clipping algorithms can be applied in world coordinates so that the contents of the window interior are mapped to device coordinates.

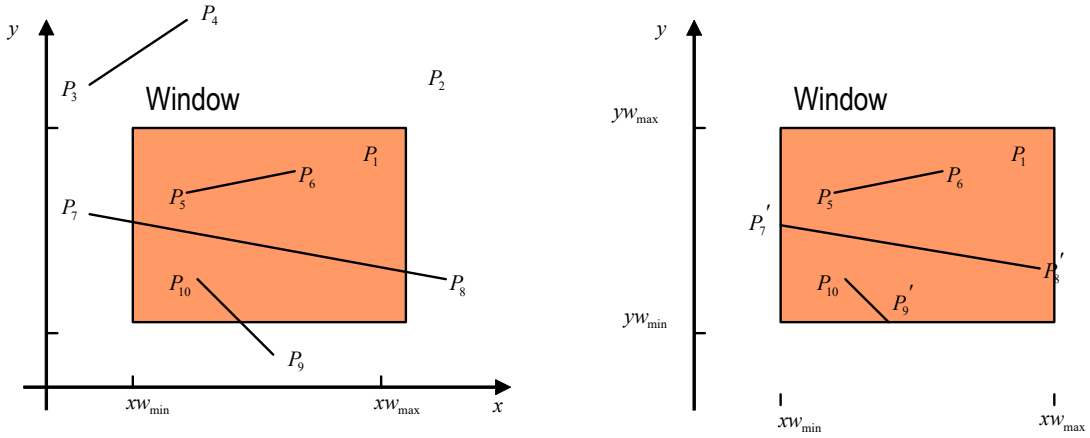
In the following sections we will discuss clipping that are applied to the following primitive types:

- Lines
- Polygon

8.4 LINE CLIPPING

A line clipping is the process of removing portions of line segments outside an area of interest that is, the clip window. The lines that intersect the clip window at some point have some portion of the line inside the clip window. The lines that do not intersect the window at any point are totally inside the window or totally outside the clip window. Thus, any line or a part of it that is outside the viewing area is removed.

To clip a line against a clip window, only the end-points are tested rather than testing all the points on the line. The figure below shows the line clipping against a clip window.



(a) Before Clipping **(b) After Clipping**

A line clipping algorithm has to test the following before clipping a line or any part of the line.

- Whether the line lies completely inside the window.
- Whether the line is completely outside the window

- If the line lies partially inside and partially outside the specified window.

Lines that are not intersecting any border of the clipping window are either completely outside or completely inside the window boundary. The intersecting lines have one segment of the line inside and the rest outside.

The clipping region is commonly called as clipping window, or if the window is a standard rectangle, it is called as clipping rectangle. There are many line clipping algorithms. The Cohen-Sutherland Line clipping algorithm is generally used for line clipping.

8.4.1 COHEN-SUTHERLAND ALGORITHM

This algorithm is one of the simple and most widely used. To clip a line, the end-points of the line are only considered. If both the end-points lie inside the boundary, it is accepted that the line is inside the clip window. Thus there is no clipping procedure to be performed. In the case, where both end-points lie entirely to one side of the clip window, the line must be entirely outside the window. It is trivially rejected and the line needs neither clipping nor is to be displayed. The Cohen-Sutherland algorithm works in two phases:

- It identifies those lines which are intersecting the window boundary
- Perform the clipping operation.

Let P(x,y) be a line. Say (xs,ys) and (xe, ye) be the line end-points. A line is said to be completely inside if the following condition holds:

$$Xw_{min} \leq xs \leq xw_{max}$$

$$Xw_{min} \leq xe \leq xw_{max}$$

and $yw_{min} \leq ys \leq yw_{max}$

$$yw_{min} \leq ye \leq yw_{max}$$

A line is said to be completely outside if the following condition holds :

$$Xs < xw_{min} \text{ and } xe < xw_{min}$$

Or, $Xs > xw_{max}$ and $xe > xw_{max}$

The algorithm assigns an 8 digit binary code to each of the two end-points of the line. This code is called the region code. The region code identifies the location of the point relative to the boundaries of the clipping window. Each bit position in the region code is used to indicate one of the four relative coordinate positions of the point with respect to the clip window: to the left, right, top or bottom.

The figure below shows the binary codes that are assigned to the end-points of a line.

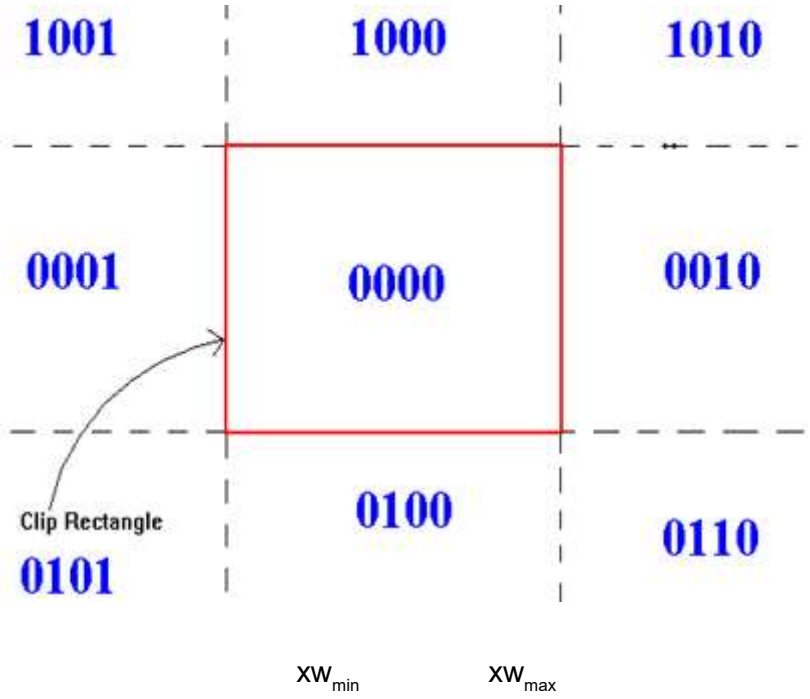
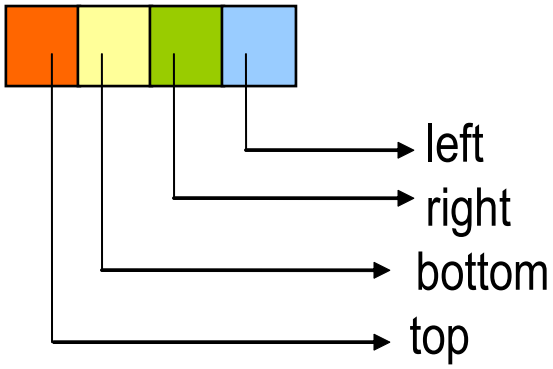


Figure : The binary region codes assigned to the line end-points.

This algorithm has extended the window boundaries to define 9 regions :

Top-left, top-center, top-right, center-left, center, center-right, bottom-left, bottom-center, bottom-right.

These 9 regions are uniquely identified by the given region code. The order --- left, right, bottom, top (LRBT) will be used for the 8-bit region code.



The sequence for reading the code bits is Left, Right, Bottom and Top, that is from the least-significant bit to the most-significant bit.

A value of 1 in any bit position indicates that the point lies in that position; otherwise that bit position is set to a value of 0. For any point P (x,y) :

- Left (first) bit is set to 1 if P lies to left of window, that is, when x < xwmin
- Right (second) bit is set to 1 if P lies to right of window,
- Bottom (third) bit is set to 1 if P lies below window, that is, when y < ywmin
- Top (fourth) bit set is set to 1 if P lies above window, that is, when y > ywmax

If the point P lies within the clip window, then all the bit positions are set to a value 0, that is, 0000.

For a point that lies to the top-left corner of the clip window, the region code is 1001.

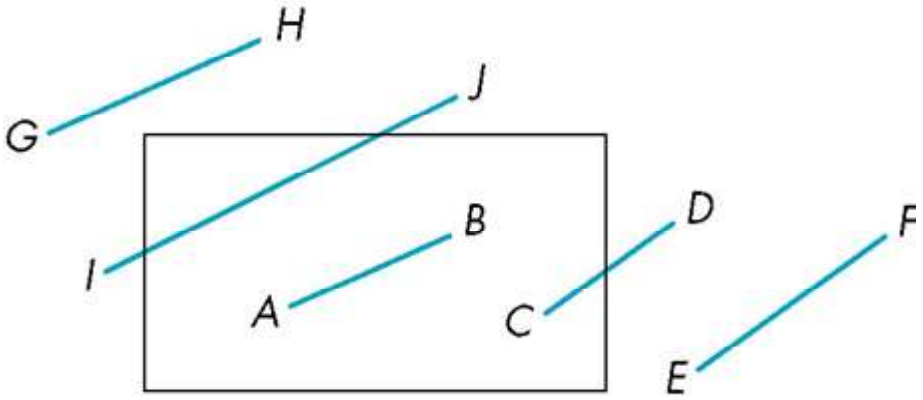
Thus the region codes for the line end-points are established. Using this region code it is now easier to find out which of the lines are inside the clip window and which are not.

- If the region code for both the end-points of the line is 0000, then the line is completely inside the clip window and is accepted.
- If the region codes for both the end-points have a 1 in the same bit position, then the line is completely outside the clip window and it is trivially rejected. For example, for the line where one

end-point is having the region code as 1001 and the other end-point is having the code 0101.

- A logical AND operation can also be performed on the two region codes of the two end-points of the line to check for clipping the lines.

The lines which are not completely inside or completely outside the clip window are tested for intersection with the window boundaries. The figure below shows some lines and the clip window.



Let us consider the five cases above where we have to perform line clipping.

AB : The region code for line A is 0000.
The region code for B is 0000.
So the line AB is accepted as both the region codes are 0000.

CD : C has code 0000.
D has 0010.

The line CD thus neither lie completely inside nor completely outside as one of the end-point lie inside the window and the other lie to the left of the window (as the right bit is 1). The line thus has to be shortened by intersecting with one of the window edge. Compute the code of intersection that is the new end-point of the shortened line segment and then check the end-points (C and new one for D) again.

The location of the bit 1 determines which edge of the window to intersect. If there were two 1's in the region codes of the end-

points then two intersections might have to be done.

EF : The end-point E has the code 0010
F has the code 0010.

As there is a 1 in the same bit position of the two end-points of the line, the line EF lies completely outside the window and is thus discarded.

GH : The region code for point G is 1001
Code for the point H is 1000.

As there is a 1 bit in the same position of G and H, the line is completely outside and is discarded.

IJ : Region code for I is 0001
Region code for J is 1000.

As there is a 1 bit in the left bit position of I, it means that it lies to the left of the clip window.

Again as a 1 exists in the top bit position of J so it lies to the top of the window.

This means that the line IJ might have intersected the window in two different positions. The intersection point of I with the window say I/ is first calculated. So the new line segment is now I/J.

Then the intersection point of J with the window is calculated, say we have J/. So, I/J/ is the new line segment.

The region codes for I/ and J/ is now checked. For I/ it is 0000 and for J/ it is also 0000. so the line I/J/ is accepted and lies inside the clip window.

This algorithm has many advantages. It is the simplest one, and easy to understand and program. It does not have any unnecessary computations to determine fully visible and invisible lines. The memory usage of the algorithm is economical.

However, it is not left without any shortcoming too. For non-trivial cases, finding an intersection point might need to do four times intersection calculation per vector.

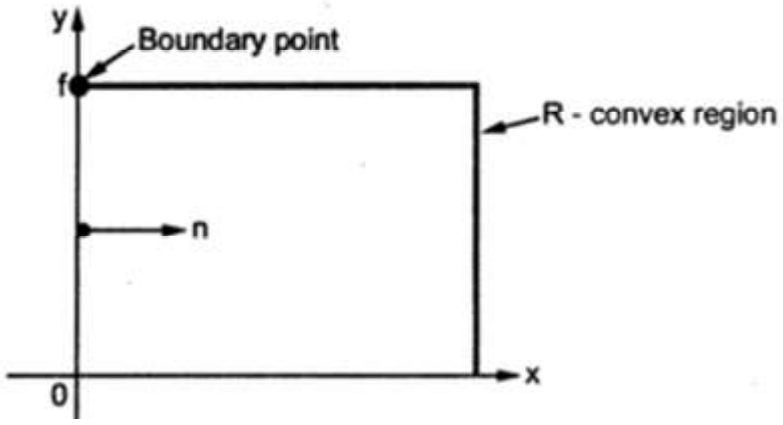
8.4.2 Cyrus-beck Line Clipping Algorithm

The Cyrus-Beck line clipping algorithm is a generalized line clipping algorithm that is applicable to arbitrary convex polygons and not necessarily rectangular window. Not only a rectangular window, but also it works fine with non-rectangular windows. It is more efficient than the Cohen-Sutherland algorithm which uses repetitive clipping. It uses parametric line representation and simple dot products.

The algorithm explained above assumes that the clipping window is a regular rectangle. Cyrus and Beck have developed a generalized line clipping algorithm. It is applicable to an arbitrary convex region. It uses a parametric equation of a line segment to find the intersection points of a line with the clipping edges.

This algorithm generally clips only once or twice unlike the Cohen Sutherland algorithm where the lines are clipped about four times.

The algorithm is applicable to an arbitrary convex region. A parametric equation of a line segment is used in this algorithm to find the intersection points of a line with the clipping edges.

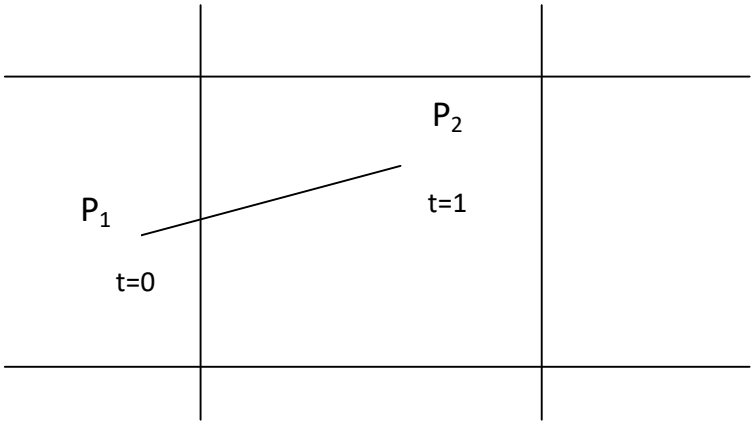


The parametric equation of a line segment from P1 to P2 is

$$P(t) = P_1 + (P_2 - P_1)\, t \qquad ; 0 \le t \le 1$$

$$= (1\text{-}t)\, P_1 + tP_2$$

Where t is a parameter, t=0 at P_1 and t=1 at P_2.



The value of the dot product is calculated so as to find the intersection point with the clipping window. Let PE be a point on the clipping plane E.

Calculate :

if < 0, vector pointed towards interior;

if = 0, vector pointed parallel to plane containing P;

if > 0, vector pointed away from interior.

Here n stands for normal of the current clipping plane (pointed away from interior).

By this we select the point of intersection of line and clipping window where (dot product is 0) and hence clip the line.

The algorithm is as follows:

Step1: Input the line segment P1P2 with end coordinates P1(x1,y1) and P2(x2,y2) respectively.

Step 2: Read the vertex coordinate of the clipping polygon and calculate the direction D = P2 - P1

Step 3: Assign boundary point P to the edge and compute the inner normal n for the edge.

Step 4: Compute lower and upper limits.

If n.D > 0

Else $t_L = -\frac{n - W}{n - D}$

$$t_u = -\frac{n - W}{n - D}, \text{Where } D \neq 0$$

$$t_L = -\frac{n - W}{n - D}, \text{Where } D \neq 0$$

Step 5: Repeat steps 3 to 4 for each edge of the clipping polygon.

Step 6: Find maximum lower limit and minimum upper limit if they do ot satisfy the condition 0 d" t d" 1

Step 7: Compute the intersection points using the maximum lower limit and minimum upper limit in the parametric equation.

Step 8: Draw the line segment from $P(t_L)$ to $P(t_U)$

Step 9: Finish

The Cyrus-Beck algorithm is efficient when many line segments needs to be clipped. The algorithm can also be extended to convex polygon clip windows.



CHECK YOUR PROGRESS-1

Q 1: Fill in the blanks :

a. _____ refers to the removal of objects, lines, or line segments that are outside the viewing pane.

b. Clipping can be performed with respect to a clipping boundary, which may be a _____ or _____ polygonal boundary.

c. Internal clipping removes parts of a picture _____ the window and external clipping removes those parts that are _____ a region.

d. To clip a line against a clip window, only the _____ are tested rather than testing all the points on the line.

e. The region code in the Cohen- Sutherland line clipping algorithm identifies the location of the point relative to the _____ of the clipping window.

f. The _____ line clipping algorithm is more efficient than the Cohen-Sutherland algorithm which uses repetitive clipping.

8.5 POLYGON CLIPPING

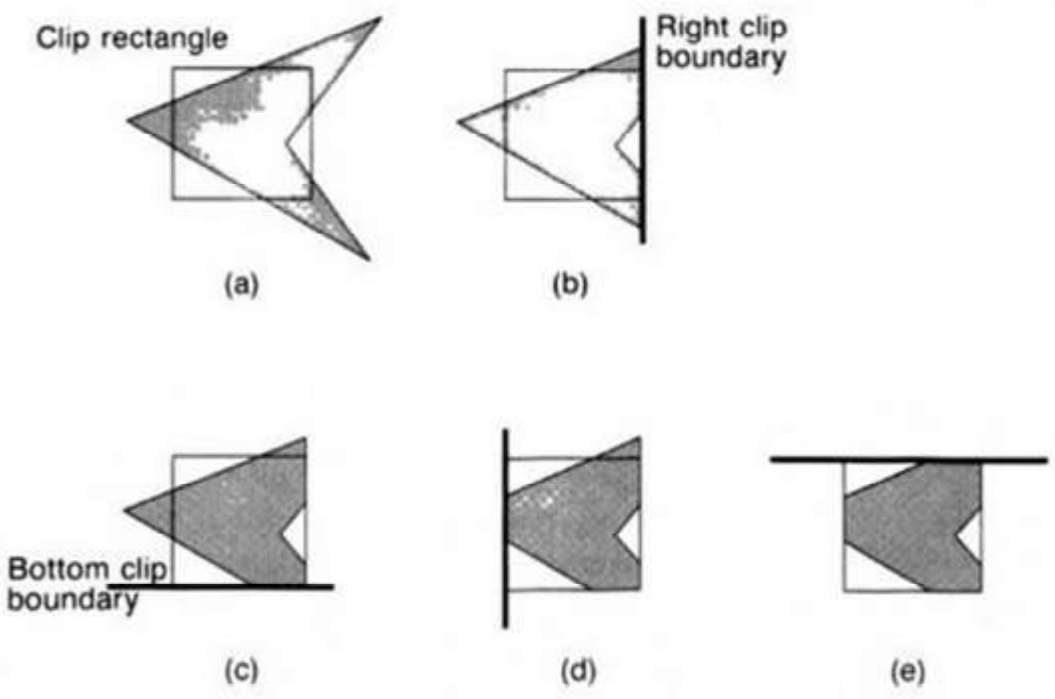
In the previous sections we have seen the line clipping algorithms. In this section we would be learning about the polygon clipping algorithm. A polygon is a closed figure that is bounded by straight lines. Polygons can be concave or convex. Computer graphics deals with both types. Although polygon is a collection of lines, the polygon clipping algorithms and the line clipping algorithms are different. If a line algorithm is used to clip a closed polygon, the original closed polygon becomes one or more open polygons or discrete lines.

In case of polygon clipping, each edge of the polygon must be tested against each edge of the clip rectangle, new edges must be added and existing edges must be discarded, retained or divided. Sometimes, clipping one polygon may result in many polygons.

8.5.1 Sutherland -Hodgman algorithm

There are several polygon clipping algorithms. One of the earliest polygon clipping algorithms is the Sutherland-Hodgeman algorithm. It is relatively simple. The algorithm is efficient in two cases: when the polygon is completely inside the boundaries and other is when it is completely outside.

It is based on clipping the entire polygon against an edge of the window, then clipping the new polygon against the next edge of the window and so on until the polygon has been clipped against all of the four edges. That is, the algorithm starts with an initial set of polygon vertices and clipping against the left window edge to produce a new set of vertices. The new set of vertices is then clipped against the bottom edge, right edge and the top edge respectively. In other words, it uses a divide-and-conquer strategy to solve the problem.



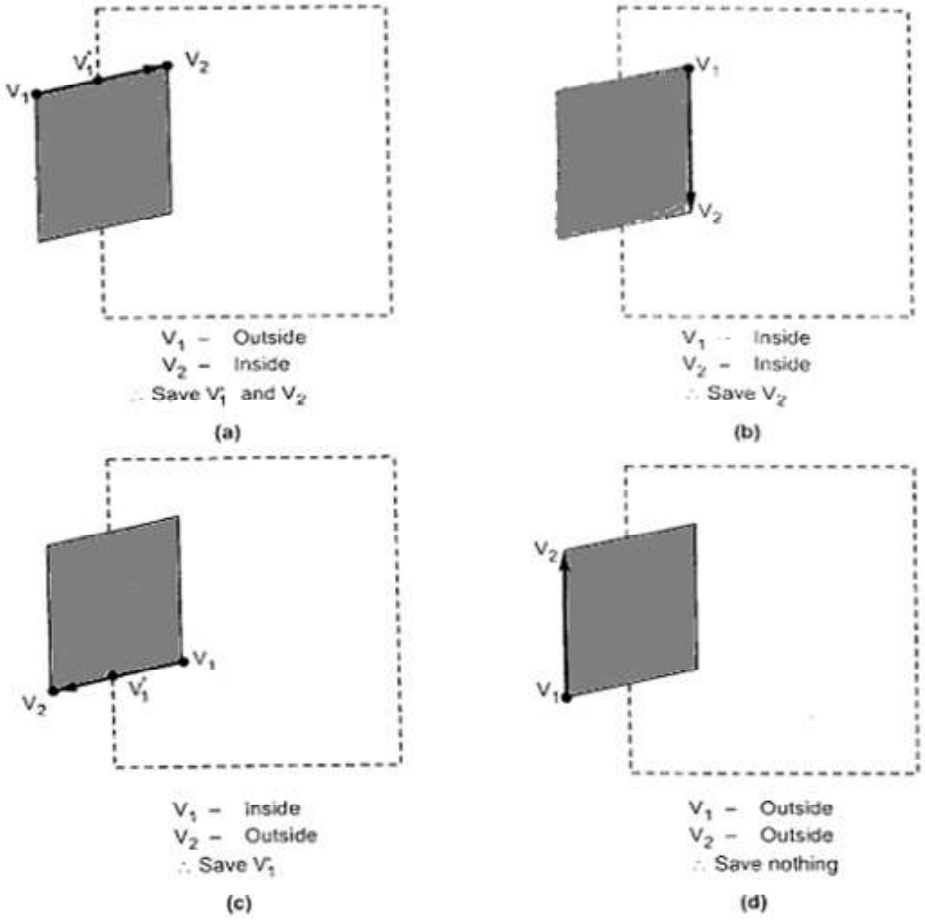
The figure above shows polygon clipping, edge by edge. Figure (a) shows before clipping. The figure (b) shows clipping on the right, (c) Clip on bottom, (d) Clip on left, (e) Clip on the top and the polygon is fully clipped.

All the polygon vertices are processed against each clip rectangle boundary in turn. Beginning with the original polygon, it is clipped. At each step a new set of polygon vertices is generated and passed on to the next window boundary clipper, as shown in the figure above. This is the fundamental idea that is used in this algorithm.

The output of this algorithm is a list of polygon vertices all of which are on the visible side of a clipping plane. This is achieved by processing two vertices of each edge of the polygon around the clip boundary. This results in four possible relationships between the edges and clip boundary.

- If the first vertex of the edge is outside the window boundary and the second vertex of the edge is inside then the intersection point of the polygon edge with the window boundary and the second vertex are added to the output vertex list.

- If both vertices of the edge are inside the window boundary, only the second vertex is added to the output vertex list.
- If the first vertex of the edge is inside the window boundary and the second vertex of the edge is outside, only the edge intersection with the window boundary is added to the output vertex list.
- If both vertices of the edge are outside the window boundary, nothing is added to the output list.





CHECK YOUR PROGRESS-2

Q 2: Fill in the blanks:

a._____ uses a divide-and-conquer strategy.

b. All the _____ vertices are processed against

each clip rectangle boundary in turn.

c. The output of the Sutherland-Hodgman algorithm is a list of polygon vertices all of which are on the visible side of a_____.

d. If both vertices of the edge are inside the window boundary, only the _____ is added to the output vertex list.

e. If both vertices of the edge are outside the window boundary, _____ is added to the output list.



8.6 LET US SUM UP

- Clipping' refers to the removal of parts of a picture that are either inside or outside a specified region.
- Internal clipping removes parts of a picture outside the window; external clipping removes those parts that are inside a region.
- A point P(x,y) is inside the clip window if the point lies between the window boundaries. That is,

$$xw_{min} \leq x \leq xw_{max}$$
$$yw_{min} \leq y \leq yw_{max}$$

- A line clipping algorithm has to test the following before clipping a line or any part of the line.
  a. Whether the line lies completely inside the window.
  b. whether the line is completely outside the window
  c. if the line lies partially inside and partially outside the specified window.
- The algorithm assigns an 8 digit binary code to each of the two end-points of the line. This code is called the region code. The region code identifies the location of the point relative to the boundaries of the clipping window.
- The Cyrus-Beck line clipping algorithm is more efficient than the Cohen-Sutherland algorithm which uses repetitive clipping
- Cyrus-Beck line clipping algorithm generally clips only once or twice unlike the Cohen Sutherland algorithm where the lines are clipped about four times.

- Although the polygon is a collection of lines, the polygon clipping algorithms and the line clipping algorithms are different.
- One of the earliest polygon clipping algorithms is the Sutherland-Hodgeman algorithm.
- The output of Sutherland-Hodgeman polygon clipping algorithm is a list of polygon vertices all of which are on the visible side of a clipping plane.



8.7 ANSWER TO CHECK YOUR PROGRESS

Ans to Q No 1:

a. Clipping
b. Convex, concave
c. Outside, inside
d. end-points
e. boundaries
f. Cyrus-Beck

Ans to Q No 2:

a Sutherland-Hodgman algorithm
b. polygon
c. clipping plane
d. second vertex
e. nothing



8.8 FURTHER READINGS

1) Computer Graphics By Donald Hearn and M. Pauline Baker
2) Fundamentals of Computer Graphics, Third Edition, By Peter Shirley, Steve Marschner
3) Computer graphics: principles and practice By James D. Foley



8.9 MODEL QUESTIONS

Q 1: Explain clipping. What is line and polygon clipping?

Q 2: Discuss Line clipping. Explain the Cohen-Sutherland Algorithm.

Q 3: Discuss Cyrus-beck line clipping algorithm.

Q 4: Explain the Sutherland-Hodgman algorithm for polygon clipping.

*** ***** ***

UNIT 9: 3-D OBJECT REPRESENTATION

UNIT STRUCTURE

9.1 Learning Objectives

9.2 Introduction

9.3 3-D Object Representation

9.4 Polygon surfaces

9.5 Quadric surfaces

9.6 Spline representation

9.7 Basic illumination models

9.8 Polygon rendering methods

9.9 Let Us Sum Up

9.10 Further Reading

9.11 Answer to Check Your Progress

9.12 Model Questions

9.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Learn about 3-D object representation
- Learn about Polygon surfaces
- Learn about Quadric surfaces
- Learn about Basic illumination models
- Learn about Polygon rendering methods

9.2 INTRODUCTION

We have already been acquainted with the idea of different clipping algorithms of line and polygon like Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland -Hodgeman polygon clipping algorithm in the previous unit.

In this unit, we will learn about 3-D Object representation. We will be able to gather knowledge on Polygon surfaces, quadric surfaces, spline representation, basic illumination models, and polygon rendering methods.

In the next unit we will explore the concept of Hermite curve, Bezier curve and B-spline curves, Bezier and B-spline surfaces.

9.3 3-D OBJECT REPRESENTATION

In 3-D representation x, y, and z coordinates are used to represent an object. For 3-D objects, transition, scaling, and rotation are used to represent the object. 3-D object representation is divided into the following two categories:

- **Boundary Representations:** Here, the object representation is possible in the form of polygon surfaces. It is a three-dimensional object model. It is enclosed by surfaces and has its interior and exterior. Example: Polygon surfaces
- **Space-partitioning representations:** It is used to represent the internal features of the polygon which is used to represent a 3D object. In non-overlapping regions, the space is divided.



EXERCISE

1. What are the major division of 3-D object representation?

9.4 POLYGON SURFACES

In this type of representation, the object is represented with a set of polygon surfaces. This is the most commonly used boundary representation of 3D objects. We may not be able to represent all the 3D objects using polygon surfaces. We can represent polygon surfaces using the following three approaches:

- **Polygon Tables:** In this method, the surface is represented using a set of coordinates and attributes. Here the object information is stored by using three tables
  - Vertex table
  - Edge table
  - Polygon-surface table

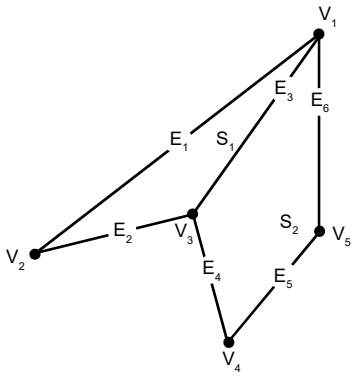


Figure 9.1: Polygon

- **Vertex table:** As shown in figure 9.1, there are five vertices v_1, v_2, v_3, v_4, and v_5. Each vertex stores the information of x, y, and z coordinates. This information can be written using the following vertex table for the five vertices.

Vertex table	
V_1	x_1, y_1, z_1
V_2	X_2, y_2, z_2
V_3	X_3, y_3, z_3
V_4	X_4, y_4, z_4
V_5	X_5, y_5, z_5

Table1: Vertex table of figure 9.1

- **Edge table:** As shown in figure 9.1, there are six edges E_1, E_2, E_3, E_4, E_5, and E_6. The Edge table is used to store the edge information of the polygon. For example, in figure 9.1, the E_1 edge connects the vertices V_1 and V_2. So we can write this information using the edge table for all the edges as given in the following table.

Edge table	
E_1	V_1 , V_2
E_2	V_2 , V_3
E_3	V_1 , V_3
E_4	V_3, V_4
E_5	V_4, V_5
E_6	V_1 ,V_5

Table 2: Edge table of figure 1

- **Polygon-surface table:** As shown in figure 9.1, there are two surfaces S_1 and S_2. It stores the number of surfaces present in the polygon. For example, in figure 9.1 surface S_1 is covered by the edges E_1, E_2, and E_3, and surface S_2 is covered by the edges E_3, E_4, E_5, and E_6. We can write the surface information using the following polygon-surface table.

Polygon-surface table	
S_1	E_1, E_2, E_3
S_2	E_3 , E_4 , E_5 ,E_6

Table 3: Polygon-surface table of figure 9.1

Polygon equation is:

Ax + By + Cz + D = 0

Where, x, y, and z are points on polygon and A, B, C, and D are the constants.

We can represent this equation like the following also:

(A/D)x+ (B/D)y+(C/D)z= -1

Let us assume three vertices are (x1, y1, z1), (x2, y2, z2), and (x3, y3, z3).

Now, if we apply Cramer's rule, we can simplify the above in the following way:

$$A=\begin{bmatrix}1 & y_1 & z_1\\ 1 & y_2 & z_2\\ 1 & y_3 & z_3\end{bmatrix}\quad B=\begin{bmatrix}x_1 & 1 & z_1\\ x_2 & 1 & z_2\\ x_3 & 1 & z_3\end{bmatrix}\quad C=\begin{bmatrix}x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{bmatrix}\quad D=-\begin{bmatrix}x_1 & y_1 & z_1\\ x_2 & y_2 & z_2\\ x_3 & y_3 & z_3\end{bmatrix}$$

If we simplify this, we will finally get the values as

$A= y_1 (y_2z_3-y_3z_2) + y_2(y_3z_1-y_1z_3)+y_3(y_1z_2-y_2z_1)$

$B= x_1 (x_2z_3-x_3z_2) + x_2(x_3z_1-x_1z_3)+x_3(x_1z_2-x_2z_1)$

$C= x_1 (x_2y_3-x_3y_2) + x_2(x_3y_1-x_1y_3)+x_3(x_1y_2-x_2y_1)$

$D= x_1 (y_2z_3-y_3z_2) + x_2(y_3z_1-y_1z_3)+x_3(x_1y_2-x_2y_1)$

This is how we get the constant values of A, B, C, and D.

For any point x, y, or z with parameters A, B, C, and D, we can say that -

- $A_x + B_y + C_z + D \neq 0$, the point is not on the plane.
- $A_x + B_y + C_z + D < 0$, the point is inside the surface.
- $A_x + B_y + C_z + D > 0$, the point is outside the surface.

- **Polygon Meshes:** The objects which we cannot represent with the help of the polygon surfaces can be represented using polygon meshes. We can have two types of representation: i) Triangle representation and ii) Quadrilateral representation
  - **Triangle representation:** Here if the object has 'n' vertices then the object has an (n-1) triangle.
  - **Quadrilateral representation:** Here, the surface is divided in the form of small cubes. If the object has n × m array of vertices it will have (n-1)×(m-1) array of Quadrilaterals



CHECK YOUR PROGRESS

Q 1: How can we represent polygon surface?

.....

9.5 QUADRIC SURFACES

Quadric surfaces are represented in the form of second-degree equations. Using this surface we can represent a complex object. We can represent spheres, ellipsoids, torus, paraboloids, and hyperboloids using quadric surfaces.

- **Sphere:** The equation of the sphere can be represented using the following second-degree equation.

$$x^2+y^2+z^2=r^2$$

where, $x = r\cos\varphi\cos\theta$
$y = r\cos\varphi\sin\theta$
$z = r\sin\varphi$

$$\text{where } \varphi: \text{ Latitude}, -\frac{\pi}{2} \le \varphi \le \frac{\pi}{2}$$
$$\varphi: \text{Longitude}, -\pi \le \theta \le \pi$$

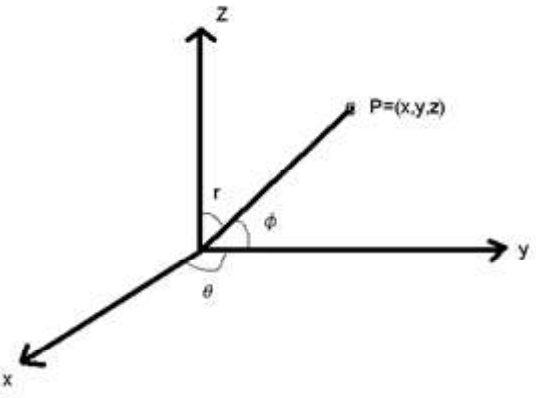


Figure 9.2: Axis angle

- **Ellipsoid:** This is an extension of a spherical surface. This can also be represented in the form of second-degree equations. Here we have three radii, which are perpendicular to each other. The equation for Ellipsoid is given below:

  $(x/r_x)^2+(y/r_y)^2+(z/r_z)^2=1$

  And parametric representation is:

$$x=r_x \cos\phi \cos\theta$$

$$y=r_y \cos\phi \sin\theta$$

$$z=r_z \sin\phi$$

$$\text{where } -\frac{\pi}{2} \le \phi \le \frac{\pi}{2}$$

$-\pi \le \theta \le \pi$ and rx is the radius of the x axis, ry is the radius of the y axis and rz is the radius of z axis.

- **Torus:** A torus is a doughnut-shaped object. It can be generated by rotating a circle, about a specific axis. This can also be represented in the form of second-degree equations.

The equation of Torus is given below:

$$\text{r-v } (x/r_x)^2+(y/r_y)^2+(z/r_z)^2 =1$$

$$x = r_x(r+\cos\phi)\cos\theta$$
$$y = r_y(r+\cos\phi)\sin\theta$$
$$z = r_z \sin\phi$$
$$where -\frac{\pi}{2} \le \phi \le \frac{\pi}{2}$$

$-\pi \leq \theta \leq \pi$ and rx is the radius of x axis, ry is the radius of y axis and rz is the radius of z axis.



CHECK YOUR PROGRESS

Q 2: Define quadric surface.

.....

9.6 SPLINE REPRESENTATION

A spline is a flexible strip used to produce a smooth curve through a designated set of points. These points are termed control points. The equation of the spline curve is cubical. Spline curves are piecewise polynomial curves. These curves are cubic, polynomial parametric. We use Spline curves to create the model of automobile bodies. It is a free-form curve. We join the given control points to form the spline curve. Control points define the shape of a curve. We can manipulate curves with the help of these control points.

There are two types of spline curves. They are as follows:

- **Interpolation spline:** If a curve is passing through all the control points then the curve is termed as interpolation spline curve. Figure 9.2 shows the interpolation spline curve.
- **Approximation spline:** When the polynomials are fitted to the general control point path without necessarily passing through the control points such type of spline curve is termed as approximation spline. Figure 9.3 shows the Approximation spline curve.



Figure 9.3: Interpolation spline Figure 9.4: Approximation spline

Properties of Spline curves: Spline curve has both local as well as global properties. The property that defines spline curve at a particular place only is termed as local property. Example of local property may be continuity of the curve, curvature of the curve. On the other hand, global

properties define the whole curve. For example global properties will be like the curve is open or close.



CHECK YOUR PROGRESS

Q 3: Write the properties of spline curve.

.....

9.7 BASIC ILLUMINATION MODELS

The illumination model is used to calculate the intensity of the light that is reflected at a given point on a surface. The lighting effect depends on the light source, surface, and observer. These factors are explained below:

- **Light Source:** It is the light emitting source. There are three types of light sources:
  - **Point Sources:** this is the source that emits rays in all directions. For example a bulb.
  - **Parallel Sources:** The point source which is far from the surface is termed as parallel source. For example: The sun
  - **Distributed Sources:** The source which has rays originated from a finite area is termed as distributed source. For example: A tubelight.
- **Surface:** When light falls on a surface part of it is reflected and part of it is absorbed. The surface structure, the position of the surface and all the nearby surfaces decide the amount of light reflection and absorption.
- **Observer:** The position and sensor sensitivities of the observer also have a huge impact on the lightning effect. The different types of observer's reflection are explained below:
- **Ambient Illumination:** When source of light is indirect, then it is termed as Ambient Illumination. For example, if an observer is standing on a road and facing a building which has glass exterior and the sun rays are falling on that building and rays are reflecting back from the building and the falling on an object which is under observation then it is termed as Ambient Illumination.

The equation of reflected intensity Iamb of any point is:

$I_{amb}=K_a I_a$

Where, I_a : ambient light intensity

K_a: surface ambient reflectivity, the value of K_a varies from 0 to 1

- **Diffuse Reflection:** On rough or grainy surfaces diffuse reflection occurs. The brightness of a point is dependent on the angle made by the light source and the surface.

  The equation of the reflected intensity I_{diff} of a point is:

  $I_{diff} = K_d I_p \cos(?)=K_d I_p (N.L)$

  Where, I_p: the point light intensity

  K_d: the surface diffuse reflectivity, value of K_d varies from 0 to 1

  N: the surface normal

  L: the light direction

- **Specular Reflection:** When light falls on any shiny surface most of it is reflected back and this reflection is termed as specular reflection.

  Phong Model is an empirical model for Specular Reflection which provides the formula for calculation the reflected intensity I_{spec}:

  $I_{spec}=W(\theta)I_l\cos n(\phi)$

  Where $W(\theta)$:K_s

  L: Direction of light source

  N: Normal to the surface

  R: Direction of reflected ray

  V: Direction of observer

  θ: Angle between L and R

  ϕ: Angle between R and V



CHECK YOUR PROGRESS

Q 4: Define Diffuse Reflection

.....

9.8 POLYGON RENDERING METHODS

IRendering is a process to create realistic image of an object with

the help of computer programs. For rendering techniques, normally two models are used i.e. Illumination Model and Shading Model. We have already discussed about Illumination Model in 9.7. So in this section we will discuss about shading model.

Shading model: The process of altering color of an object or surface in 3-D scene is termed as shading.

There are two types of shading. They are:

- **Constant Intensity Shading:** This shading type is fast. It is also termed as flat shading. It renders an object with polygon surfaces. Here, a single intensity is calculated for each polygon. All points over the surface of the polygon are displayed with the same intensity value. Constant Shading can be useful for quickly displaying the general appearances of the curved surface as shown in the following figure:



Figure 9.5: object A



Figure 9.6: A is rendered with flat shading

- **Gouraud shading:** It is developed by Gouraud. It renders a polygon surface by linear interpolating intensity value across the surface.

  Here each polygon surface is rendered by performing the following calculations:

  - Determining the average unit normal vector at each polygon vertex.
  - Apply an illumination model to each vertex to determine the vertex intensity.
  - Linear interpolate the vertex intensities over the surface of the polygon.

  At each polygon vertex, we obtain a normal vector by averaging the surface normal of all polygons staring that vertex as shown in the following figure:

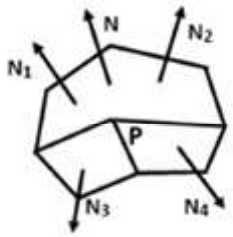


Figure 9.9: the normal vertex at vertex V is calculated as an average of the surface normal for each polygon sharing the vertex

Thus, for any vertex position V, we acquire the unit vertex normal with the calculation

$$Nv = \frac{\sum_{k=1}^{n} Nk}{\left|\sum_{k=1}^{n} Nk\right|}$$

Once we have the vertex normals, we can determine the intensity at the vertices from a lighting model.

- **Phong Shading:** The more accurate method for rendering a polygon surface is to interpolate the normal vector after that apply the illumination model to each surface point. This method is developed by Phong Bui Tuong.

Here a polygon surface is rendered by carrying out the following steps:

- Determine the average unit normal vector at each polygon vertex.
- Linearly interpolate the vertex normals over the surface of the polygon.
- Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points.

Interpolation of the surface normal along a polynomial edge between two vertices as shown in the following figure:

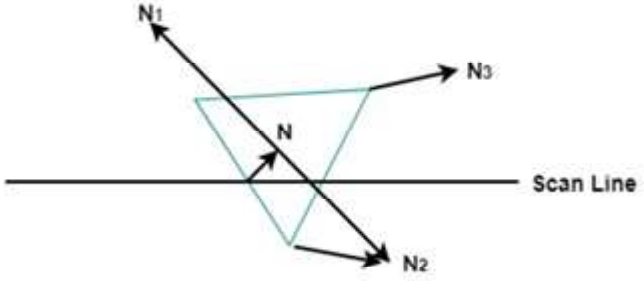


Figure 9.7: interpolation of the surface normal along a polygon edge

$$N = \frac{Y - Y2}{y1 - y2} N_1 + \frac{y1 - y}{y1 - y2} N_2$$

Here, incremental methods are used to evaluate normals between scan lines and along each scan line. At each pixel position along a scan line, the illumination model is applied to determine the surface intensity at that point.



CHECK YOUR PROGRESS

Q 5: Define gouraud shading.

.....



9.9 LET US SUM

- In 3-D, three coordinates x, y and z are used to represent an object. For 3-D image transition, scaling and rotation are needed.
- Quadric surface are described with second-degree equations. They include sphere, ellipsoids, torus, paraboloids and hyperboloids.
- Spline curves are used to create animation paths. We use Spline curves to create the model of automobile bodies and other drawings and fonts. It is a free-form curve.
- The illumination model is used to calculate the intensity of light that is reflected at a given point on the surface.
- There are two polygon rendering methods namely the Illumination model and the Shading model.



9.10 FURTHER READINGS

1) D. Hearn and P.M. Backer, Computer Graphics, Prentice Hall of India, 1986.
2) W. Newman and R.F. Sproul, Principles of Interactive Computer Graphics, McGraw Hill Publication, 1980.
3) W.K. Giloi, Interactive Computer Graphics, Prentice Hall of India, 1978.

9.11 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q No 1: We can represent polygon surfaces using the following different ways:

- **Polygon Tables:** In this method, the surface is specified by the set of vertex coordinates and associated attributes.
- **Polygon Meshes:** 3D surfaces and solids can be approximated by a set of polygonal and line elements. Such surfaces are called polygonal meshes. In polygon mesh, each edge is shared by at most two polygons. The set of polygons or faces, together form the "skin" of the object.

Ans. to Q No 2: Quadric surface are described with second-degree equations. They include sphere, ellipsoids, torus, paraboloids and hyperboloids.

Ans. to Q No 3: Spline curve has both local as well as global properties. The property that defines spline curve at a particular place only is termed as local property.

Ans. to Q No 4: Diffuse reflection occurs on the surfaces which are rough or grainy. In this reflection the brightness of a point depends upon the angle made by the light source and the surface.

Ans. to Q No 5: This Intensity-Interpolation scheme, developed by Gouraud and usually referred to as Gouraud Shading, renders a polygon surface by linear interpolating intensity value across the surface. Intensity values for each polygon are coordinate with the value of adjacent polygons along the common edges, thus eliminating the intensity discontinuities that can occur in flat shading.

9.12 MODEL QUESTIONS

Q 1: Explain the different approaches of 3D object representation with example.

Q 2: Explain the concept of vertex table with the help of an example.

Q 3: What is the use of polygon surface table?

Q 4: What is sphere? How can we represent the radius of sphere in parametric notation?

Q 5: What is Ellipsoid?

Q 6: How Spline is represented?

Q 7: Write about the illumination models.

Q 8: Discuss polygon rendering methods.

*** ***** ***

UNIT 10: BEZIER AND B-SPLINE CURVES

UNIT STRUCTURE

10.1 Learning Objectives

10.2 Introduction

10.3 Hermite Curve

10.4 Bezier Curve and B-Spline Curves

- 10.4.1 Bezier Curve
- 10.4.2 B-Spline Curve

10.5 Bezier and B-spline surfaces

- 10.5.1 Bezier Surface
- 10.5.2 B-Spline Surface

10.6 Let Us Sum Up

10.7 Further Reading

10.8 Answer to Check Your Progress

10.9 Model Questions

10.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Learn about the Hermite curve
- Learn about Hermite matrix
- Learn about Hermite Blending Functions
- Learn about the properties of the Bezier Curve and B-Spline Curves
- Differentiate between Bezier Curve and B-Spline Curves
- Learn about Bezier and B-Spline Surfaces

10.2 INTRODUCTION

We have already been acquainted with the idea of 3-D object representation and these 3-D objects are represented as a collection of surfaces. We have learned about quadric surfaces, spline representation, basic illumination models, and polygon rendering methods.

In this unit, we will learn about the Bezier curve and B-spline curves. Here we will also learn about the Hermite curve, Hermite matrix and Hermite blending function, Bezier and B-Spline Surfaces as well as the properties of Bezier Curve and B-Spline Curves. In the next unit, we will explore the concept of projection and the classification of projection.

10.3 HERMITE CURVE

The Hermit curve is expressed using parametric notation. The joining of multiple curves into a single curve with the help of the endpoints of each curve is termed as Hermite Curve. In the following figure 10.1, the endpoints of two different curves are shown and in figure 10.2 these two curves are joined to form the Hermite curve. Interpolation spline curve and cubic polynomial functions are required for making the Hermite equation.



Figure 10.1: curves



Figure 10.2: Hermite Curve

With the help of the following cubic polynomial, the Hermite curve is represented.

$$x(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$

Here, the value of t ranges from 0 to 1. The polynomial can be specified by the position of, and gradient at, each endpoint of the curve.

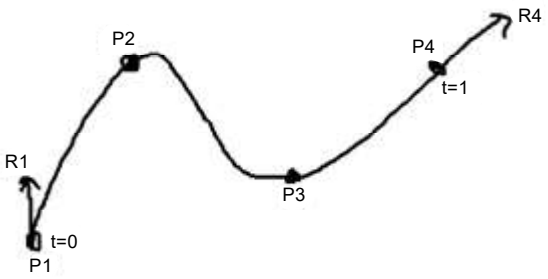


Figure 10.3: Hermite Specifications

In the above figure 10.3, p1 and p2 are the endpoints.

For making Hermite function, we need:

- **First control points, Last control points:** These two points are step position vectors.
- First derivatives of points R_1 and R_4 or P' and P'_4

$R_1 = P'_1$ (1st Derivative of P'_1)

$R_4 = P'_4$ (1st Derivative of P_4)

If we want to make a curve equation we need to make

Q(t)=[x(t) y(t) z(t)], Q(t) denotes equation in three dimension x, y, and z.

Where,

$x(t)= a_x t^3 + b_x t^2 + c_x t + d_x$

$y(t)= a_y t^3 + b_y t^2 + c_y t + d_y$

$z(t)= a_z t^3 + b_z t^2 + c_z t + d_z$

$Q(t)=[t^3 \quad t^2 \quad t \quad 1]_{1x4} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$

We can also write the above equation as

Q (t)= T* C ------- (i)

Or

C=M* G

Where

M= Basic matrix: Provide Blending equation

G= Geometry Vector Matrix

By putting the value of C in equation (i), we will get

$Q(t)= T * M_H * G_H$

M= Basic matrix: Provide Blending equation of Hermite

G= Geometry Vector Matrix of Hermite

$= [t3 \quad t2 \quad t \quad 1] * M_H * G_H$ ------- (ii)

Like equation (ii), we can make the matrix equation of the curve.

$$\begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} * M_H * G_H$$

Or

$$M_H * G_H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} P_1 \\ P_2 \\ R_1 \\ R_4 \end{bmatrix}$$

After inverse comparing both sides

$$M_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

And

$$G_H = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

Now,

$$Q(t) = [t^3 \quad t^2 \quad t \quad 1] * M_H * G_H$$

$$= [t^3 \quad t^2 \quad t \quad 1] * \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

$$= \underbrace{(2t^3-3t^2+1)P_1}_{H_1} + \underbrace{(-2t^3+3t)P_4}_{H_2} + \underbrace{(t^3-2t^2+t)R_1}_{H_3} + \underbrace{(t^3-t^2)R_4}_{H_4} \quad \text{-------(iii)}$$

H_1, H_2, H_3, and H_4 are called Hermite Blending Function.

$Q(t) = H_1P_1+H_2P_4+H_3R_1+H_4R_4$

In equation (iii) we can directly put the value of t, and we can find the equation of the Hermite curve.



EXERCISE

1. Write down the Hermite blending function.

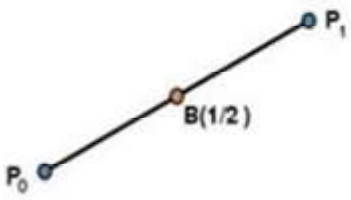
.....

10.4 BEZIER CURVE AND B-SPLINE CURVES

10.4.1 Bezier Curveb

The bezier curve is defined by the control points. It is discovered by

Pierre Bezier. It is a spline curve. It is used in CAD, drawing applications, etc.



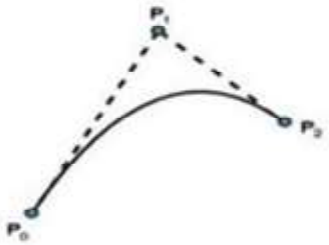




Figure 10.4: Simplest Bezier Figure 10.5: Quadratic Bezier Figure 10.6: Cubic Bezier

In the above figures P0, P1, P2, and P3 are the control points. And the curves formed from these points in the above figures are the Bezier curves.

There are three types of Bezier curves. Namely, Simplest Bezier curve, Quadratic Bezier curve and Cubic Bezier curve. The simplest Bezier curve is the straight line from the point P0 to P1, which connects two endpoints. In figure 10.4 one example of the Simplest Bezier curve is given. A quadratic Bezier curve is determined by three control points. In figure 10.5 a quadratic Bezier curve is shown. And a cubic Bezier curve is determined by four control points. Figure 10.6 shows a cubic Bezier curve. The curve order is always less than the number of control points by one. The Bezier curve always lies inside the convex hull made by the control points.

Properties of Bezier Curve:

- Bezier curve depends on the number of control points.
- Whenever we draw the Bezier curve it may only pass through the endpoints. It may not be all the control points.
- The polynomial equation of a Bezier curve is dependent on the control points. If we have n number of control points, then the degree of the polynomial function is n-1.
- Bezier curve is a global control curve, that is the change of the position of a control point can change the shape of the complete graph. Changing the curve shape locally or a portion of the graph

through the control points is not possible.

Derivation for the quadratic Bezier curve is given below:

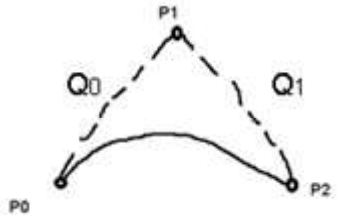


Figure 10.7: Bezier curve having three control points P1, P2 and P3

Bezier curve is a parametric curve.

Let the parameter is u and 0<=u<=1

And Q_0 and Q_1 points are the lines between P0 $\rightarrow$ P1 and P1 $\rightarrow$ P2

Q_0 = (1-u)P0+uP1 ----(i)

Q1 = (1-u)P1+uP2 ------(ii)

Suppose c(u) is the point between $Q_0 \rightarrow Q_1$

c (u)=(1-u) Q_0 + uQ_1 from equation (i) and (ii)

c (u)=(1-u) [(1-u)P0+uP1] + [u(1-u)P1+uP2]

= $(1-u)^2$ P0+u(1-u)P1+u(1-u)P1+u^2P2

= $(1-u)^2$P0+2u(1-u)P1+u^2P2 ----(iii) (for 3 control points)

Equation (iii) is the quadratic equation for the Bezier curve for three control points P0, P1, and P2.

The blending function for the Bezier curve is given below:

Suppose we have n+1 control points positions P_k (x_k y_k z_k) then the

Positional vector P(u) where 0<=u<=1 is given by

$P(u)=\sum_{k=0}^{n} P_k BEZ_{k,n}(u)$ 0<=u<=1

Blending function

$BEZ_{k,n} = c(n,k)u^k(1-u)^{n-k}$

$x(u)= \sum_{k=0}^{n} x_k BEZ_{k,n}(u)$

$y(u)= \sum_{k=0}^{n} y_k BEZ_{k,n}(u)$

$z(u)= \sum_{k=0}^{n} z_k BEZ_{k,n}(u$

10.4.2 B-SPLINE CURVES

The b-Spline curve is locally controlled that is the shape of a portion of the curve can be changed through the control points without affecting the rest of the portion of the graph. Here the degree of the polynomial does not depend on control points. It depends on the order of the polynomial. That is, it is based on the number of control points and if the order varies then the blending function degree will be different.

Properties of B-Spline Curve:

It is made of (n+1) control points and the order of the curve (k).

- It has local control over the curve
- It is used to draw open and closed curves.
- It gives us a polynomial of degree k-1.
- It has n-k+2 segments n: number of control points
- It allows us to change the number of control points without changing the degree of the polynomial.

The mathematical expression of the B-Spline curve using the Blending function is given below:

$$P(u) = \sum_{k=0}^{n} B_{ik}(u) P_k$$

$$P_k(X_k \quad Y_k \quad Z_k)$$

$$X(u) = \sum_{k=0}^{n} B_{ik}(u) X_k$$

Where 0<=u<=n-k+2 and 2<=k<=n+1

K=here k is responsible for segments (number of segments)



CHECK YOUR PROGRESS

Q 1: Define Bezier curve.

.....

Q 2: Write the properties of Bezier curve and B-Spline curve.

.....

10.5 BEZIER AND B-SPLINE SURFACES

10.5.1 BEZIER SURFACE

Bezier surface can be obtained by two sets of orthogonal Bezier curves by specifying an input mesh of control points. The bezier surface has the same properties as the Bezier curve. A cylinder/ cone can be formed from a Bézier surface. The equation of the Bezier surface is parametric and it is formed as the Cartesian product of the blending functions as given below:

$P(u,v)=\sum_{j=0}^{m}\sum_{k=0}^{n} P_{j,k}\, BEZ_{j,m}(v)\, BEZ_{k,n}(u)$

Where, $P_{j,k}$ is the location of (m+1) by (n+1) control points.

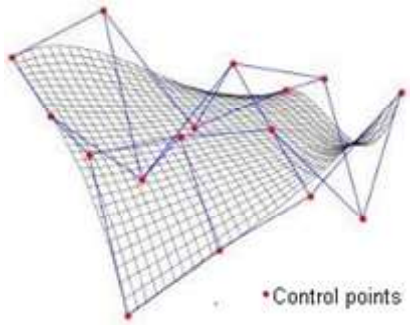


Figure 10.8: Bezier surface

10.5.2 B-SPLINE SURFACES

The b-Spline surface can be similarly formed to the Bezier Spline. The b-Spline surface has the same properties as the B-Spline curve. B-spline surface maps the unit square to a rectangular surface. The vector point function of B Spline can be obtained using the Cartesian product of the B-Spline blending function as given below:

$P(u,v)=\sum_{k1=0}^{n1}\sum_{k2=0}^{n2} P_{k1,k2}\, B_{k1,d1}(u) B_{k2,d2}(v)$

Where $P_{k1,k2}$ are the positions of the (n_1+1) by (n_2+1) control points. And d1 and d2 determine the polynomial degrees to be used.

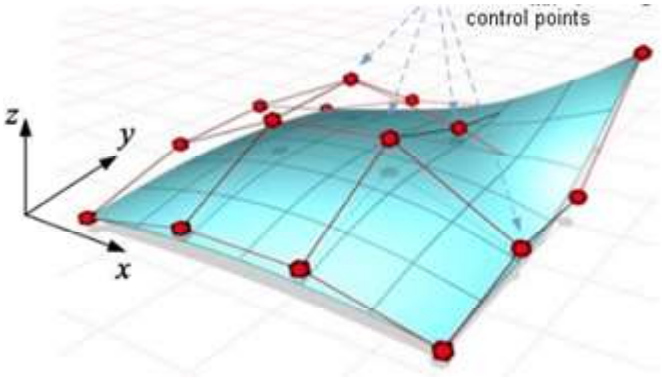


Figure 10.9: B-Spline surface



CHECK YOUR PROGRESS

Q 3: Define B-Spline surface.

.....



10.6 LET US SUM UP

- Hermit curve is expressed in terms of parametric notation i.e. $x(t) = a_3t^3 + a_2t^2 + a_1t + a_0$
- The result of joining different curves is called a Hermite Curve.
- Bezier curves can be generated under the control of other points.
- The Bezier surface is formed as the Cartesian product of the blending functions of two orthogonal Bezier curves.
- B-spline surface maps the unit square to a rectangular surface.



10.7 FURTHER READINGS

1) D. Hearn and P.M. Backer, Computer Graphics, Prentice Hall of India, 1986.
2) W. Newman and R.F. Sproul, Principles of Interactive Computer Graphics, McGraw Hill Publication, 1980.
3) W.K. Giloi, Interactive Computer Graphics, Prentice Hall of India, 1978.

10.8 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q No.1: Bezier curve is defined by the control points. It is discovered by Pierre Bezier. It is a spline curve. It is used in CAD and drawing applications.

$c(u)=(1-u)[(1-u)P0+uP1] + [u(1-u)P1+uP2]$

$= (1-u)^2 P0+u(1-u)P1+u(1-u)P1+u^2P2$

$= (1-u)^2P0+2u(1-u)P1+u^2P2$ ----(i) (for 3 control points)

Equation (i) is the quadratic equation for the Bezier curve for three control points P0, P1, and P2.

Ans. to Q No. 2: Properties of Bezier Curve:

- Bezier curve depends on the number of end control points.
- Whenever we draw the Bezier curve it may only pass through the endpoints. It may not be all the control points.

Properties of B-Spline Curve:

- It is made of (n+1) control points and the order of the curve (k).
- It has local control over the curve

Ans. to Q No. 3: B-Spline surface can be similarly formed like the Bezier Spline. B-Spline surface has same properties like B-Spline curve. B-spline surface maps the unit square to a rectangular surface.

10.9 MODEL QUESTIONS

Q 1: Write the limitation of the Hermite curve.

Q 2: How many points are needed for a piece of Hermite curve?

Q 3: What is the use of the Hermite curve?

Q 4: Write the difference between the Hermite curve and the Bezier curve.

Q 5: Elaborate on the equation of the Bezier curve.

Q 6: Explain the different types of Bezier curves.

Q 7: Explain the concept of Bezier's surface with the help of an example.

Q 8: Derive the Hermite matrix.

Q 9: Write down the properties of the Bezier curve.

Q 10: Write down the properties of the B-Spline surface.

*** ***** ***

UNIT 11: PROJECTIONS

UNIT STRUCTURE

11.1 Learning Objectives

11.2 Introduction

11.3 Projection

11.4 Projection Classification

11.5 Parallel projections

11.6 Perspective projections

11.7 Hidden Layer Projection

11.8 Let Us Sum Up

11.9 Further Reading

11.10 Answer to Check Your Progress

11.11 Model Questions

11.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Learn about Projection and its different types
- Learn about Parallel Projection and its classification
- Learn about Perspective Projection and Vanishing point
- Learn about anomalies in Perspective Projection
- Learn about Hidden Layer Projection and its techniques

11.2 INTRODUCTION

We have already been acquainted with the idea of the Hermite curve, Bezier curve and B-spline curves, Bezier surface, and B-spline surfaces in the previous unit.

In this unit, we will learn about Projection and its classification which is very important. Here we will also learn about hidden layer projection and its techniques. In the next unit, we will explore the concept of 3-D Geometric transformations that is Translation, rotation, scaling, reflection and shear transformations, and composite transformations.

11.3 PROJECTION

The process of representing a 3D object into a 2D medium is termed Projection. All graphics devices are 2D, such as the screen of the monitor, printout of the printer, etc. The projection was invented in 1756 by the Swiss mathematician, engineer, and astronomer Leonhard Euler Around. Projection is nothing but the transformation of an n-dimensional object into the n-1-dimensional object in the projection plane.

Important concepts related to Projection:

- **Projection Plane/ View Plane:** It is the plane on which an object's projection is formed.
- **Center of Projection:** it is the point from which projection is taken.
- **Projectors:** These are the lines coming from the center of the projection and hitting the projection plane after passing through a point in the object.

11.4 PROJECTION CLASSIFICATION

Mainly projection is divided into two types. These categories are Parallel Projection and Perspective Projection.

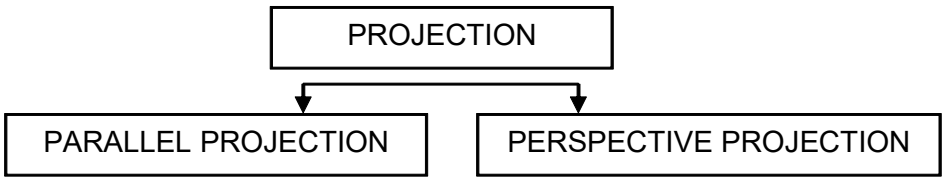


Figure 11.1: Types of Projection

- **Parallel Projection:** If the center of projection is at an infinite distance to the projection plane then it is termed parallel projection. The main types of parallel projection are shown in the following figure 11.2. The detail of parallel projection will be explained in the next section which is 11.5.

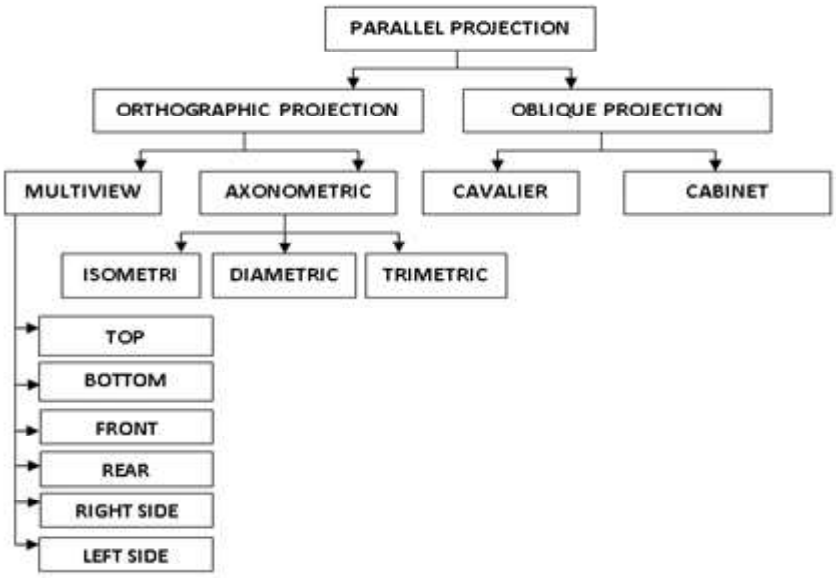


Figure 11.2: Types of Parallel Projection

- **Perspective Projection:** If the center of projection is at a finite distance to the projection plane then it is termed perspective projection. The main types of perspective projection are shown in the following figure. The detail of parallel projection will be explained in the next section which is 11.6.

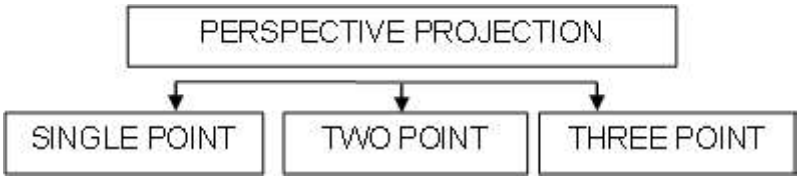


Figure 11.3: Types of Perspective Projection



EXERCISE

1. Define projection.
2. What is the need of projection?

11.5 PARALLEL PROJECTIONS

In parallel projection, coordinate positions are transformed to the view plane along parallel lines. A projection is said to be parallel if the center of the projection is at an infinite distance from the projected plane. Also here the projection lines are parallel to each other and extended from the object and intersect the view plane.

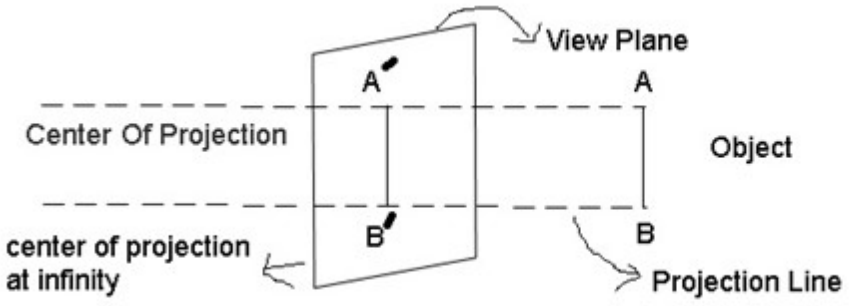


Figure 11.4: Parallel Projection

The major types of parallel projection are shown in figure 11.2. In this section, we will learn about the different types of parallel projection i.e. orthographic projection and oblique projection.

- **Orthographic Projection:** Here the direction of the projection is normal to the projection plane. In orthographic projection, lines are parallel to each other making an angle of 90 degrees with the view plane/ projection plane. There are mainly two types of orthographic projection i.e. a) Multi-view and b) Axonometric. The multi-view projection has 5 types they are they are: i) Top ii) Bottom iii) Rear iv) Right side v) Left side. Whereas Axonometric projection is divided into 3 types, they are i) Isometric, ii) Diametric and iii) Trimetric

By combining multi-view types that are top, bottom, rear, right side, and left side, we can construct the whole object that we want to project.

In axonometric projection, the direction of projection is not parallel to any of the three principal axis x, y, and z. the projection here is done using rotation and translation to manipulate the object such that at least three adjacent faces are shown. In isometric projection under axonometric projection, the direction of the projection makes equal angles with the three principal axis x, y, and z. In diametric projection, the direction of the projection makes an equal angle with exactly two of the principal axis. And lastly, in trimetric projection, the direction of the projection makes unequal angles with the three principal axis x, y, and z.

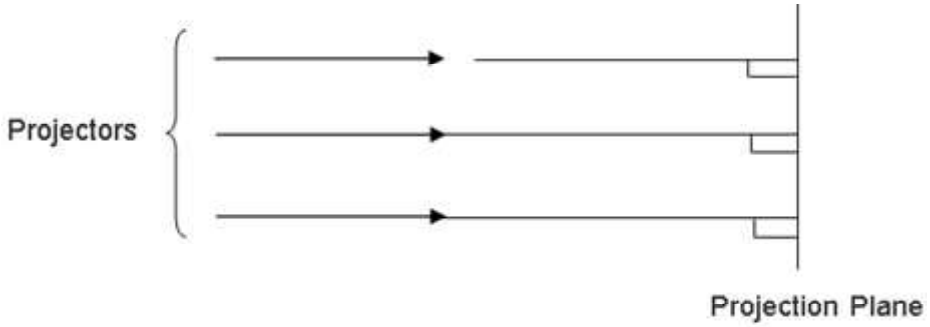


Figure 11.5: Orthographic projection: here projectors are perpendicular to the projection plane.

- **Oblique Projection:** Here, the direction of the projection is not normal to the projection plane. In this projection, we can view the object better than in the orthographic projection. Here projections are non-perpendicular to the view plane. An example of this projection can be seen in the form of the shadow of any object due to sunlight. There are mainly two types of oblique projection. They are a) Cavalier and b) Cabinet.

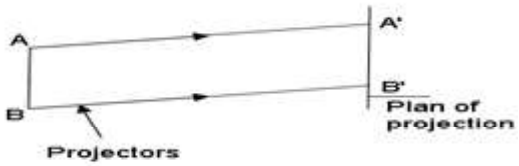


Figure 11.6: Oblique Projection

  - **Cavalier Projection:** if the projected line is making an angle of 45 degrees with the projection plane, as a result, the line of the object length will not change.
  - **Cabinet Projection:** it makes a 63.4-degree angle with the projection plane. Here lines perpendicular to the viewing surface are projected at ½ their actual length.



CHECK YOUR PROGRESS

Q 1: What is isometric projection?

.....

Q 2: Why parallel projection is used?

.....

11.6 PERSPECTIVE PROJECTIONS

In perspective projection, the center of projection is at a finite distance from the projection plane or view plane. The visual effect in perspective projection is similar to the human visual system. Here object appears smaller as the distance from the center of the projection increases. It is difficult to determine the exact size and shape of the object in perspective projection. There are three types of perspective projection. i.e. a) one-point perspective projection, b) two-point perspective projection and c) three-point perspective projection.

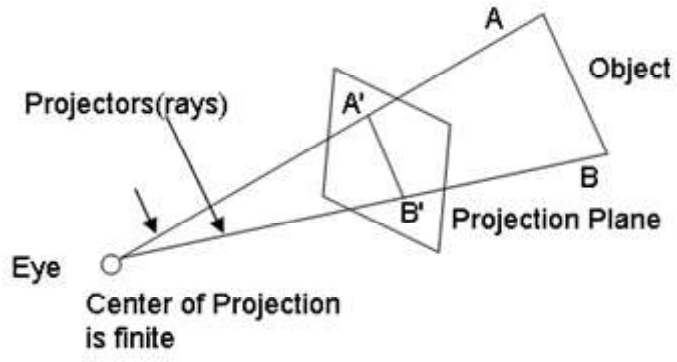


Figure 11.7: Perspective Projection

Vanishing Point: The point where all line meet is termed a vanishing point.

- **One Point:** There is only one vanishing point in one point perspective projection. And it is simple to draw

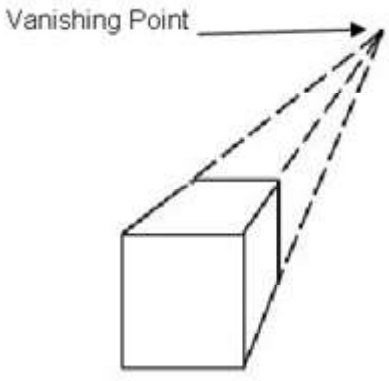


Figure 11.8: One-point perspective projection

- **Two Points:** There are two vanishing points in the two-point perspective projection. One point is in the x-direction and the other is in the y -direction. This gives a better impression of depth.

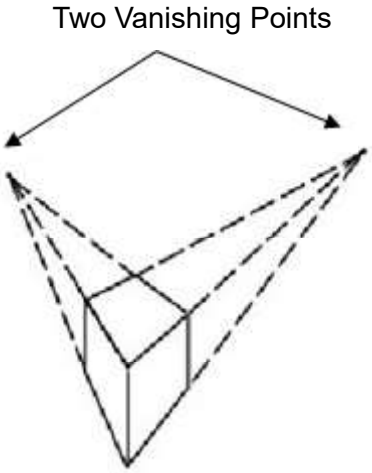


Figure 11.9: Two-point perspective projection

- **Three Points:** There are three vanishing points in the three-point perspective projection. It is most difficult to draw. Here all three principal axes have a finite vanishing point.

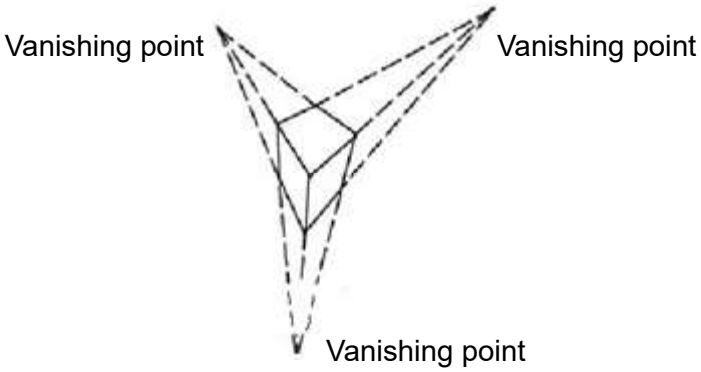


Figure 11.10: Three-point perspective projection



CHECK YOUR PROGRESS

Q 3: Define vanishing point

.....

11.7 HIDDEN LAYER PROJECTION

Hidden layer projection is a process through which surfaces that should not be visible to the user are prevented from being rendered. One of the most challenging problems in computer graphics is the project the hidden parts from images of solid objects.

There are two types of hidden surface detection algorithms: a) Object space methods and

- Image-space methods.
- Object space method: Here the parts of the objects are compared. After the comparison, visible, invisible or hardly visible surface is determined. Normally it determines the visible surfaces or layers.
- Image-space methods: In this method, pixels' position is determined. It is used to locate the visible surface instead of a visible line. Each point is detected for its visibility.

Following are some examples of algorithms used for hidden Layer or surface detection:

- Back Face Removal Algorithm
- Z-Buffer Algorithm
- Painter Algorithm
- Scan Line Algorithm
- Subdivision Algorithm
- Floating horizon Algorithm



CHECK YOUR PROGRESS

Q 4: Define hidden layer projection.

.....

.....



11.8 LET US SUM UP

- Projection is the process of converting a 3D object into a 2D object.
- Parallel Projection is used to display a picture in its true shape and size.
- When projectors are perpendicular to the view plane then is called orthographic projection.
- In perspective projection farther away object from the viewer, small it appears. This property of projection gives an idea about depth.
- In Perspective projection lines of projection do not remain parallel.

- The Center of projection is the location of the eye on which projected light rays converge.
- Hidden layer projection is a process through which surfaces that should not be visible to the user are prevented from being rendered.



11.9 FURTHER READINGS

1) D. Hearn and P.M. Backer, Computer Graphics, Prentice Hall of India, 1986.
2) W. Newman and R.F. Sproul, Principles of Interactive Computer Graphics, McGraw Hill Publication, 1980.
3) W.K. Giloi, Interactive Computer Graphics, Prentice Hall of India, 1978.



11.10 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q No1: Isometric projection is a method for visually representing three-dimensional objects in two dimensions in technical and engineering drawings. It is an axonometric projection in which the three coordinate axes appear equally foreshortened and the angle between any two of them is 120 degrees.

Ans. to Q No 2: 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. Parallel Projection use to display a picture in its true shape and size.

Ans. to Q No 3: The vanishing point is a point at which receding parallel lines seem to meet when represented in linear perspective.

Ans. to Q No 4: Hidden layer projection is a process through which surfaces that should not be visible to the user are prevented from being rendered.



11.11 MODEL QUESTIONS

Q 1: What is projection? Explain with the help of an image.

Q 2: Differentiate between parallel projection and perspective projection.

Q 3: What do you mean by the center of projection?

Q 5: Define view volume.

Q 6: Explain the different techniques for hidden layer projection.

Q 7: What is the use of projection in computer graphics?

Q 8: Define perspective projection and its types.

*** ***** ***

UNIT 12: 3-D GEOMETRIC TRANSFORMATIONS

UNIT STRUCTURE

12.1 Learning Objectives

12.2 Introduction

12.3 Translation

12.4 Rotation

12.5 Scaling

12.6 Reflection and Shearing transformation

12.7 Composite transformation

12.8 Let Us Sum Up

12.9 Further Reading

12.10 Answer to Check Your Progress

12.11 Model Questions

12.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Know about the different 3D Geometric transformations.
- Learn how the transformations are done in graphics
- Know about the composite transformation in computer graphics.

12.2 INTRODUCTION

In the previous unit we have discussed the projections and their classifications. We have also gone through the perspective projections as well as the hidden layer projections.

To create a realistic computer model of the world or anything in it, the model should be 3D as well. Representation of any object in 3D requires the understanding of the shape of the object. In computer graphics, animations are done on the screen using the different transformation techniques. Graphics is now used almost everywhere and it is rapidly growing. The three dimensional (3D) graphics are used extensively by the interactive software in business and education.

In this unit, we will introduce you to the transformation techniques of the pictures and images. Transformations are important part of the process of building an image. Construction of any geometrical object requires the use of geometrical transformations. If an image has two identical or similar parts, then only one of them needs to be constructed. The other part can be obtained by copying the first part and then moving, reflecting, scaling and rotating it to get the required shape, size and orientation. Operations such as moving, reflecting, scaling, rotating an image are called geometric transformations and will be discussed in this unit for three dimensions.

Three-dimensional transformation is similar to 2D transformations, with some minor differences. Here, we have 4 X 4 transformation matrices (in homogeneous coordinate system) instead of 3 X 3.

12.3 TRANSLATION

A translation refers to moving every point at a constant distance in a particular direction. In other words, repositioning an object along a straight line path from one coordinate location to another coordinate location is called translation transformation. t_x and t_y are the translation parameters or translation distances.

A point (x,y,z) is translated to a new location (x', y', z') by adding the translation parameters t_x, t_y and t_z to the original coordinates x, y and z.

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

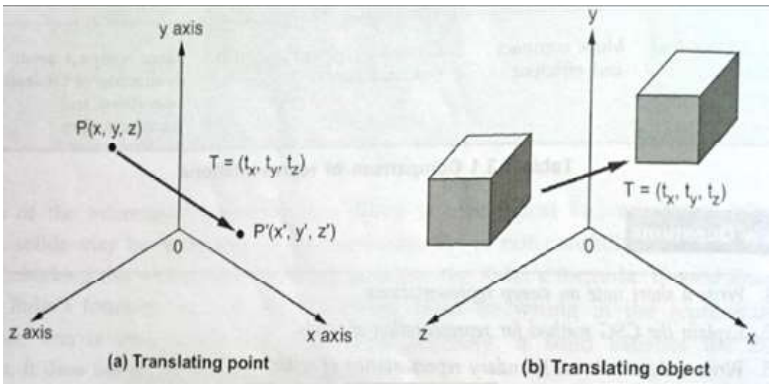


Figure : 3D Translation

Translation in 3D is a simple extension from that in 2D. In matrix form,

$$T(t_x,t_y,t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This can be written as,

P' = T.P

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

- Translation is a rigid body transformation which moves the objects without deformation.
- 3D translation is very similar to 2D translation.
- There are now three coordinates to translate (t_x, t_y, t_z). A 4×4 matrix is used to perform the operation.

12.4 ROTATION

Rotation is movement in circular motion. The rotation transformation describes the motion of a rigid body around a fixed point. In 2-dimension, all rotations are defined about the Z-axis, or an axis parallel to it. However, in 3D, there are three basic rotations, with respect to each of the principle axes X, Y and Z. performing rotation transformation in 3D is much more complex than that of 2D because 3D transformation rotation can be specified around any line in space.

The transformation matrix for rotation about an arbitrary axis (any axis of rotation other than the principle axis) is more complicated.

Coordinate axes rotations

The 2D z-axis rotation equations can be extended to 3D as

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta \quad \text{.....(12.1)}$$
$$z' = z$$

where θ is the rotation angle. Thus, the 3D z-axis rotation can be expressed in homogeneous coordinates as

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

which can be rewritten as

$$P' = R_z(\theta).P$$

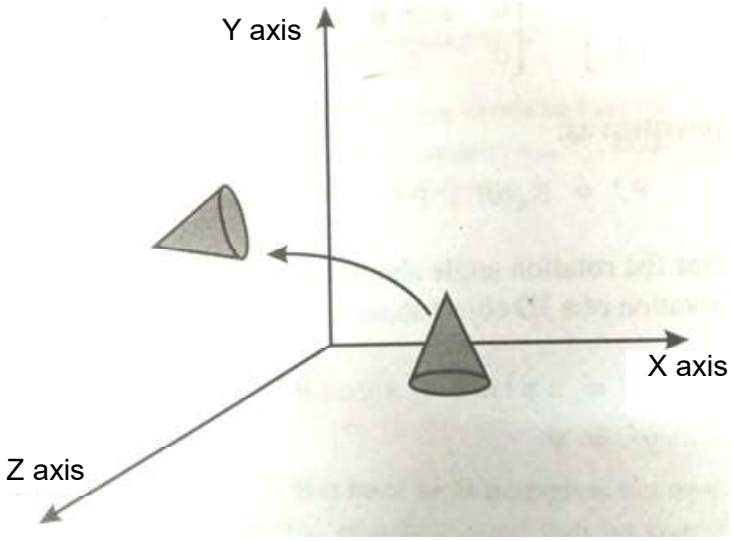


Figure : Rotation of an object about the z-axis

Rotations about the other two axes can be obtained by the following replacement.

X → y → z → x ..... (12.2)

Substituting the (12.2) in the equation 12.1, the equations for the x-axis rotation are obtained as follows:

$$y' = y\cos\theta - z\sin\theta$$
$$z' = y\sin\theta + z\cos\theta \rightarrow (12.3)$$
$$x' = x$$

which can be expressed in homogeneous coordinates as

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

This can be rewritten as,

$$P' = R_x(\theta).P$$

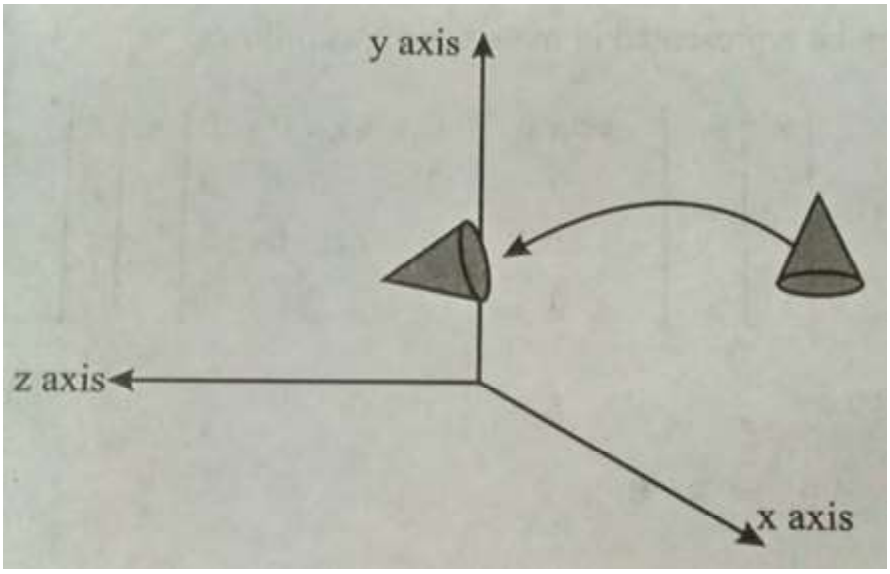


Figure : Rotation of an object about the x-axis

Substituting (12.2) in the equation 12.3, the equations for the y-axis rotation are obtained as follows:

$$\begin{aligned} z' &= z\cos\theta - x\sin\theta \\ x' &= z\sin\theta + x\cos\theta \quad \rightarrow (12.4) \\ y' &= y \end{aligned}$$

which can be expressed in homogeneous coordinates as

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

This can be rewritten as,

$$P' = R_y(\theta) \,.\, \mathrm{P}$$

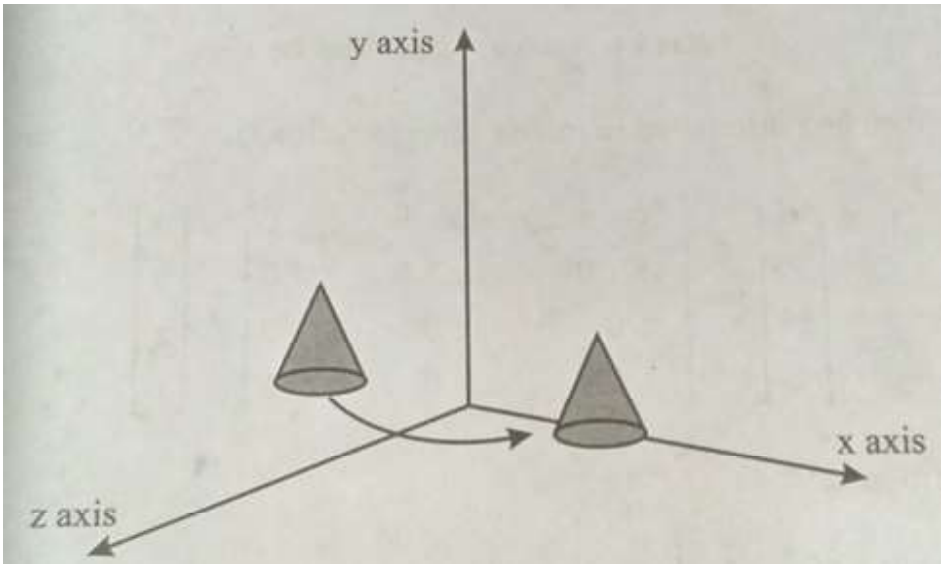


Figure : Rotation of an object about the y-axis

Rotate an object about an axis parallel to one of the coordinate axis:

For rotating an object about an axis parallel to one of the coordinate axis, the following steps needs to be performed.

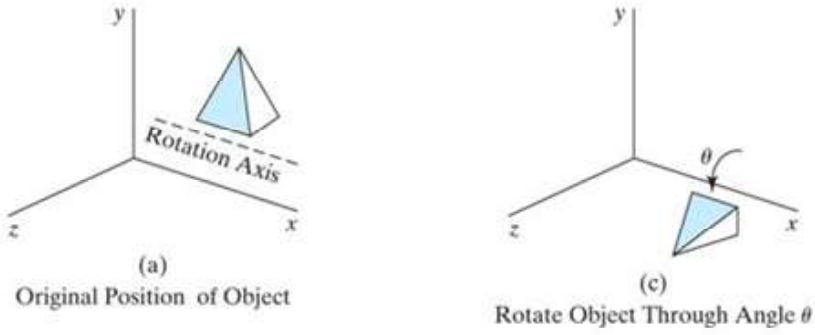
- **Translate :** object is translated in such a way that the rotation axis coincides with the parallel coordinate axis.
- **Rotate :** object is then rotated by the specified angle about that axis.
- **Translate :** the object is finally translated so that the rotation axis is moved backed to the original position.

Mathematically, it can be expressed as,

$P' = T^{-1} \cdot R(\theta).T.P$

where $R(\theta) = T^{-1}.R_x(\theta).T$ is the composite transformation.

The steps are shown in the figure below.



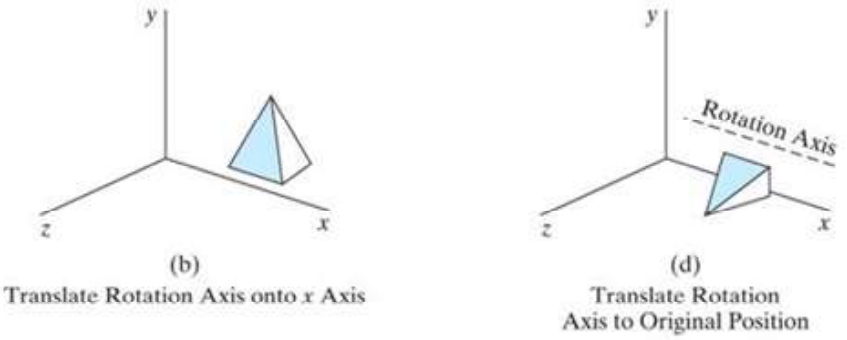
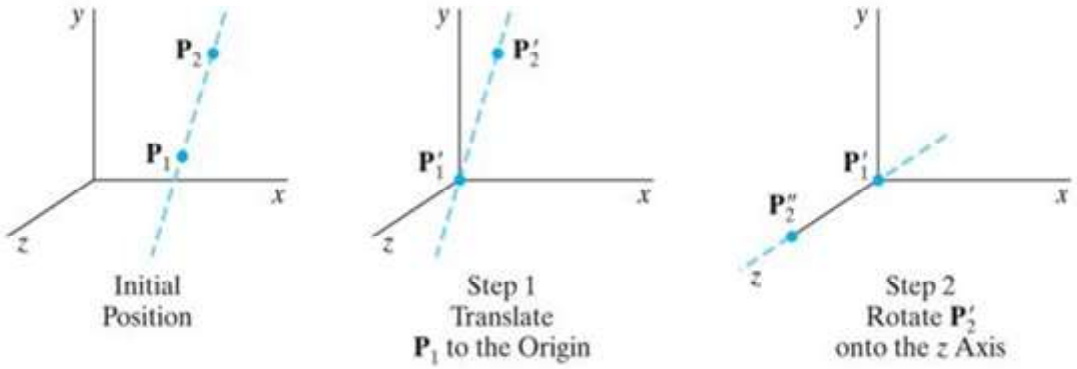


Figure : Rotating an object about an axis parallel to the x-axis

Rotate an object about an axis not parallel to any of the coordinate axis:

For rotating an object about an axis not parallel to any of the coordinate axis, the following steps needs to be performed.

- **Translate:** The object is translated such that the rotation axis passes through the coordinate origin.
- **Rotate :** Object is rotated such that its axis of rotation coincides with any of the coordinate axes
- **Rotate:** It is again rotated by the specified angle about the coordinate axis fro where the axis of rotation has passed.
- **Inverse rotation:** This brings the rotation axis to its original position.
- **Inverse translation:** This is done to bring back the rotation axis to its original position.



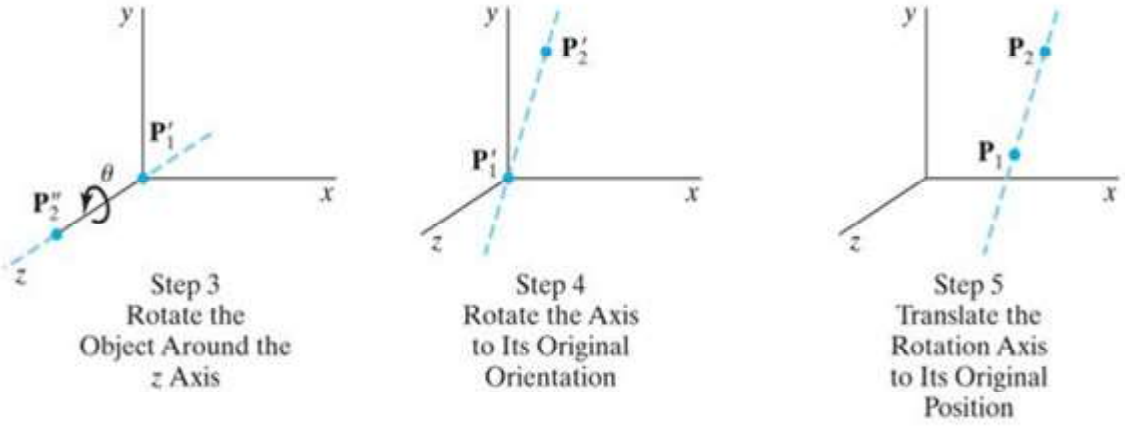


Figure : Rotating an object about an arbitrary axis

12.5 SCALING

Scaling changes the size of an object and repositions the object relative to the coordinate origin. Like 2D scaling, in 3D scaling also the size of an object is changed by compressing or expanding its dimensions. For example, the point P will be scaled from its original position P(x,y,z) to a new position $P'(x', y', z')$ relative to the coordinate origin from the scaling matrix expression.

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

where s_x, s_y and s_z are the scaling parameters that can be assigned any positive values.

$$x' = x.S_x$$
$$y' = y.S_y$$
$$z' = z.S_z$$

Equation 12.5 can also be written as

$$P' = S.P$$

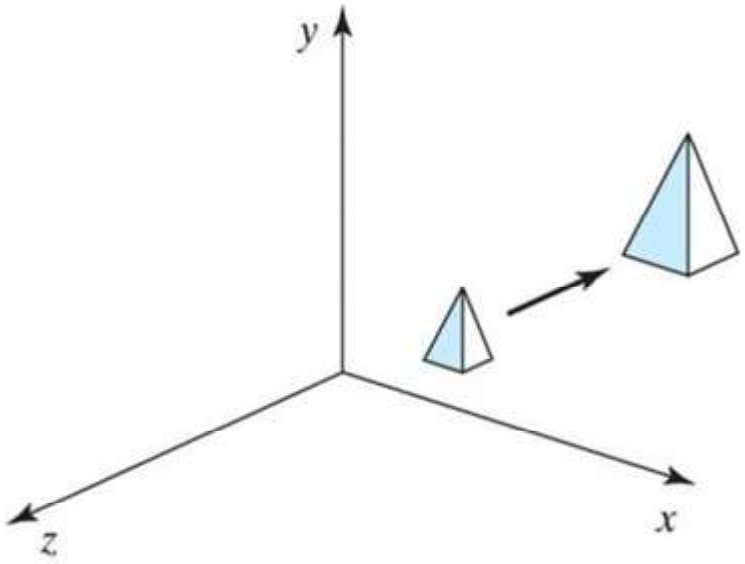


Figure : Scaling an object with the scaling parameters set to 2

If the transformation parameters are not all equal, relative dimensions in the object are changed. The original shape of an object is preserved by uniform scaling that is $s_x = s_y = s_z$. In the figure above, the scaling parameters $s_x = s_y = s_z = 2$. This has increased both the size and the distance of the object by a factor of 2.



CHECK YOUR PROGRESS

Q 1: Fill in the blanks :

a) Repositioning an object along a straight line path from one coordinate location to another coordinate location is called_____.

b) A _____ matrix is used to perform the 3D transformations.

c) In 3D, there are _____ basic rotations, with respect to each of the principle axes X, Y and Z.

d) Rotation transformation in 3D is much more complex than that of 2D because 3D transformation rotation can be specified around any _____ in space.

e) The steps needed to be performed for rotating an object about an axis parallel to one of the coordinate axis are _____, _____ and _____.

f) _____ changes the size of an object and repositions the object relative to the coordinate origin.

g) If the transformation parameters in scaling are not all equal, relative dimensions in the object are _____.

h) Operations such as moving, reflecting, scaling, rotating an image are called _____ transformations.

i) _____ is a rigid body transformation which moves the objects without deformation.

j) The transformation matrix for rotation about an _____ axis is more complicated.

12.6 REFLECTION AND SHEARING TRANSFORMATION

Reflection produces a mirror image of an object. An object is rotated a particular angle about the axis of reflection to get the reflected image.

A 3D reflection can be performed in one of the following ways:

- With respect to a selected reflection axis
- With respect to a selected reflection plane.

When reflection is performed with respect to a selected reflection axis, the object of reflection is rotated by 1800 about that axis. If the reflection is performed with respect to a selected reflection plane then also the object is rotated by 1800. However, the rotation will be performed in 4D space.

Reflection about the xy- plane

Reflection relative to the xy-plane can be obtained by the following set of instructions.

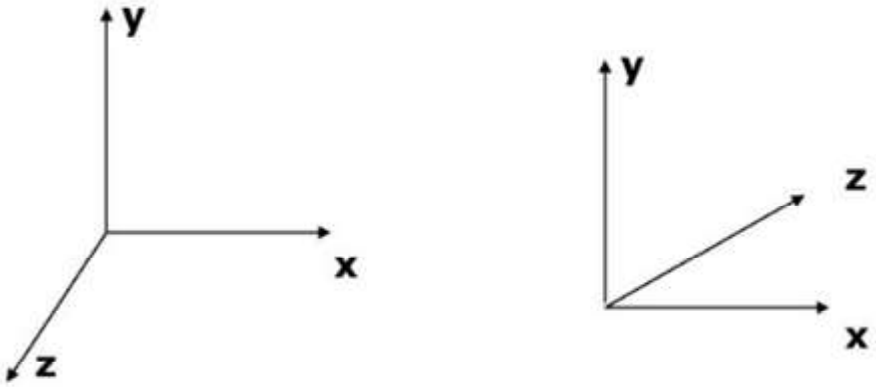
$x' = x \quad y' = y \quad z' = -z$

In homogeneous matrix form,

$$RF_{xy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Reflections about other planes can also be obtained by symmetry.

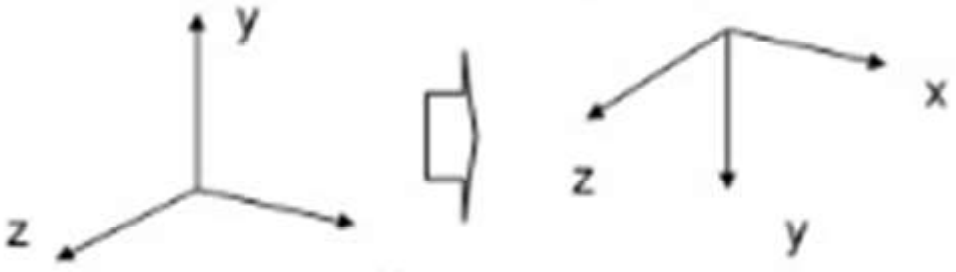
Reflection about xy-plane is a useful reflection as it converts a right-handed coordinate system into a left-handed coordinate system.



Similarly, the transformation matrix representation for the reflection of points with respect to the xz-plane (in which only the sign of the y coordinate is changed) is given as follows:

$$RF_{xy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

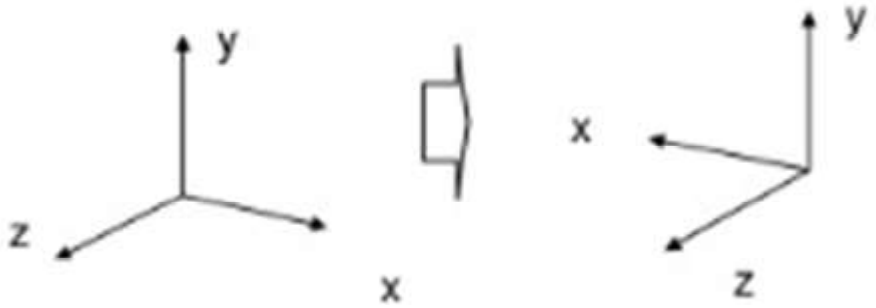
The figure below shows the reflection transformation with respect to the xz-plane.



The transformation matrix representation for the reflection in the yz-plane (where only the sign of the x-coordinate value is changed) is given as follows:

$$RF_{yz} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The figure below shows the reflection transformation with respect to the yz-plane.



3D- SHEARING TRANSFORMATION

This kind of transformation distorts the shape of the objects and they appear to look as if the internal layers are caused to slide over each other. The object is pulled horizontally by different amounts.



Original After Shear

3D shearing transformation is similar to 2D shearing transformation except that in 3D shearing one can shear along the z-axis. Shearing along the x-axis keeps the y-coordinate and the z-coordinate unchanged. Only the x-coordinate gets changed. This makes the object tilt towards the x-direction.

The transformation matrix of 3D shearing along the x-axis is as follows:

$$SH_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where a and b are parameters that can be assigned any real values.

Similarly, the transformation matrix of 3D shearing along the y-axis and the z-axis is as follows:

$$SH_y = \begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$SH_z = \begin{pmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Shearing in the y-axis and the z-axis tilts the object towards the y- and z- directions respectively.

12.7 COMPOSITE TRANSFORMATION

Complex geometric and coordinate transformations can be done from the basic transformations. Composite transformation can be setup as a sequence of transformation using matrix representation. In composite transformations, we multiply the basic matrices. As matrix multiplication is not commutative, the sequence of the matrix is very important. Wrong sequence can give wrong results. The composite transformation is carried out from left to right, where the rightmost matrix is the first transformation to be applied and the leftmost matrix is the last one to be applied. That is, each successive transformation matrix premultiplies the product of the preceding transformation matrices.

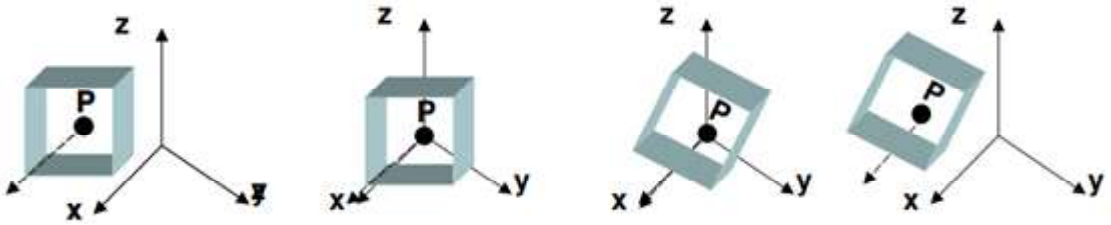
Thus, if we want to apply two transformations to a point at position P, the transformation would be calculated as

$$P' = M_2.M_1.P$$
$$= M.P$$

For example, to rotate about a fixed point, the following transformations are needed to be performed.

- Translate point to origin
- Rotate the object
- Reverse the previous translation.

The figure below shows the composite transformation.



$$\text{Translation}\quad T(-P) = \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Rotation around x}\quad R(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Translation}\quad T(P) = \begin{pmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M = T(P)R(\theta)T(-P)$$



CHECK YOUR PROGRESS

Q 2: Fill in the blanks:

a. Reflection produces a _____ image of an object.

b. If the reflection is performed with respect to a selected reflection plane then the object is rotated by_____.

c. Reflection about xy-plane is a useful reflection as it converts a right-handed coordinate system into a _____

coordinate system.

d. _____ transformation distorts the shape of the objects and they appear to look as if the internal layers are caused to slide over each other.

e. 3D reflection can be performed with respect to reflection axis and reflection _____.

f. Composite transformation can be setup as a _____ of transformation using matrix representation.

g. The composite transformation is carried out from _____ to right.



12.8 LET US SUM UP

- Repositioning an object along a straight line path from one coordinate location to another coordinate location is called translation transformation
- A 4X4 matrix is used to perform the 3D transformation operations.
- In 3D, there are three basic rotations, with respect to each of the principle axes X, Y and Z.
- Scaling changes the size of an object and repositions the object relative to the coordinate origin.
- Like 2D scaling, in 3D scaling also the size of an object is changed by compressing or expanding its dimensions.
- The original shape of an object is preserved by uniform scaling that is sx = sy = sz
- Reflection produces a mirror image of an object.
- When reflection is performed with respect to a selected reflection axis, the object of reflection is rotated by 1800 about that axis.
- If the reflection is performed with respect to a selected reflection plane then also the object is rotated by 1800
- Shear transformation distorts the shape of the objects and they appear to look as if the internal layers are caused to slide over each other.
- 3D shearing transformation is similar to 2D shearing transformation

except that in 3D shearing one can shear along the z-axis.

- Complex geometric and coordinate transformations can be done from the basic transformations.
- Composite transformation can be setup as a sequence of transformation using matrix representation.
- The composite transformation is carried out from left to right, where the rightmost matrix is the first transformation to be applied and the leftmost matrix is the last one to be applied.



12.9 FURTHER READINGS

1) Computer Graphics By Donald Hearn and M. Pauline Baker
2) Computer Graphics, Schaum's Outlines, Plastock and Kalley, McGraw-Hill © 1986
3) Computer graphics: principles and practice by James D. Foley.



12.10 ANSWER TO CHECK YOUR PROGRESS

Ans to Q No 1:

a. Translation
b. 4X4
c. Three
d. Line
e. Translate, rotate, translate
f. Scaling
g. Changed
h. Geometric
i. Translation
j. arbitrary

Ans to Q No 2:

a. mirror.
b. 1800
c. left-handed
d. Shear

e. plane

f. sequence

g. left



12.11 MODEL QUESTIONS

Q 1: Explain translation of an object in 3D space.

Q 2: How can you rotate an object about an axis parallel to one of the coordinate axis?

Q 3: Show the steps to rotate an object about an axis not parallel to any of the coordinate axis.

Q 4: Explain relection and shearing of objects in 3-dimension

Q 5: What is composite transformation?

*** ***** ***

UNIT 13: 3-D VIEWING

UNIT STRUCTURE

13.1 Learning Objectives

13.2 Introduction

13.3 Viewing pipeline

13.4 Viewing coordinates

13.5 View volume

13.6 General projection transforms

13.7 Clipping

13.8 Let Us Sum Up

13.9 Further Reading

13.10 Answer to Check Your Progress

13.11 Model Questions

13.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Learn about Viewing pipeline
- Learn about viewing coordinates
- Learn about view volume
- Learn about general projection transforms and clipping

13.2 INTRODUCTION

We have already been acquainted with the idea of Translation, rotation, scaling, reflection and shear transformations, and composite transformations in the previous unit.

In this unit we will learn about Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping. In the next unit we will explore the concept of Animation Basic, Computer Animation, Principles of Animation, Types of Animation, Animation Software (Maya, etc.) and Hardware, Common Terms in Animation.

13.3 VIEWING PIPELINE

A 3D projection (or graphical projection) is a design technique used to display a three-dimensional (3D) object on a two-dimensional (2D) surface. These projections rely on visual perspective and aspect analysis to project a complex object for viewing capability on a simpler plane. The viewing-pipeline in 3 dimensions is almost the same as the 2D-viewing-pipeline. Only after the definition of the viewing direction and orientation (i.e., of the camera) an additional projection step is done, which is the reduction of 3D-data onto a projection plane:

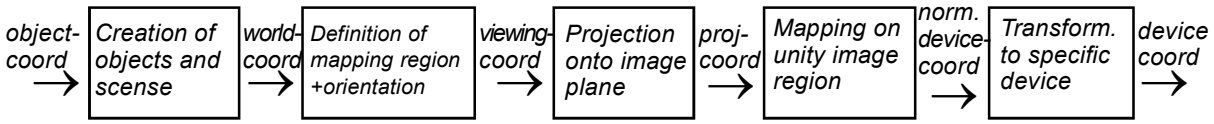


Figure 13.1: Projection step

This projection step can be arbitrarily complex, depending on which 3D-viewing concepts should be used.

13.4 VIEWING COORDINATES

Similar to photography there are certain degrees of freedom when specifying the camera. They are:

- Camera position in space
- Viewing direction from this position
- Orientation of the camera (view-up vector)
- Size of the display window (corresponds to the focal length of a photo-camera) With these parameters the camera-coordinate system is defined (viewing coordinates). Usually the xy-plane of this viewing-coordinate system is orthogonal to the main viewing direction and the viewing direction is in the direction of the negative z-axis.

Based on the camera position the usual way to define the viewing-coordinate system is:

- Choose a camera position (also called eye-point, or view-point).
- Choose a viewing direction = Choose the z- direction of the viewing-coordinates.

- Choose a direction "upwards". From this, the x-axis and y-axis can be calculated: the image-plane is orthogonal to the viewing direction. The parallel projection of the "view-up vector" onto this image plane defines the y-axis of the viewing coordinates.
- Calculate the x-axis as vector-product of the z- and y-axis.
- The distance of the image-plane from the eye-point defines the viewing angle, which is the size of the scene to be displayed.

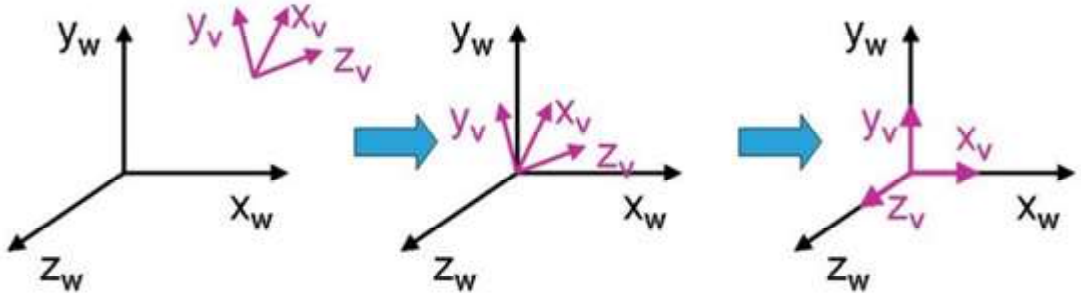


Fig.13.2: Viewing Coordinates



Fig.13.3: World Coordinates

In animations the camera-definition is often automatically calculated according to certain conditions, e.g. when the camera moves around an object or in flight-simulations, such that desired effects can be achieved in an uncomplicated way. To convert world-coordinates to viewing-coordinates a series of simple transformations is needed: mainly a translation of the coordinate origins onto each other and afterwards 3 rotations, such that the coordinate-axes also coincide (two rotations for the first axis, one for the second axis, and the third axis is already correct then). Of course, all these transformations can be merged by multiplication into one matrix, which looks about like this: $M_{WC,\ VC} = R_z \cdot R_y \cdot R_x \cdot T$

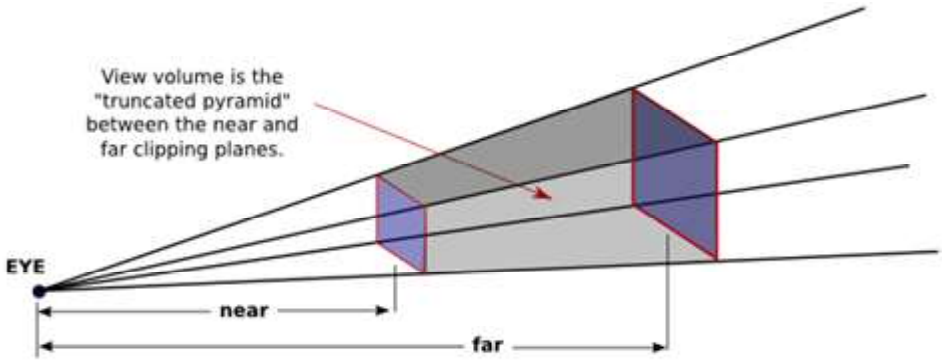


The volume of space that is actually rendered into the image is called the view volume. Things inside the view volume make it into the image; things that are not in the view volume are clipped and cannot be seen. For purposes of drawing, OpenGL applies a coordinate transform that maps the view volume onto a cube.

EXERCISE

1. What is world coordinates?

13.5 VIEW VOLUME



The view volume is bounded by six planes-the four sides plus the top and bottom of the truncated pyramid. These planes are called clipping planes because anything that lies on the wrong side of each plane is clipped away. The projection transformation maps the six sides of the truncated pyramid in eye coordinates to the six sides of the clipping cube in clip coordinates.

3D view volume Define a volume in which objects are visible for parallel projection, a box for perspective projection, a frustum surfaces outside this volume are clipped or removed and not drawn.

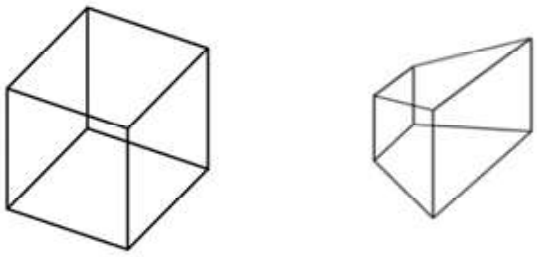


Figure 13.5: objects

CHECK YOUR PROGRESS

Q 1: Define view volume.

.....

.....

13.6 GENERAL PROJECTION TRANSFORMS

Once a 3D model has been completed, its co-ordinates need to be converted to 2 dimensions in order to display the scene on a flat computer monitor or to print it on paper. This process of converting from 3D to 2D is called projection. The visual appearance of a 3D model depends on the position of the viewer (among other things), so this must be taken into account when projecting a model. There are two main types of projection available, parallel and perspective. n three-dimensional viewing, a rectangular view window, or projection window, in the view plane is used to the same effect. Edges of the view window are parallel to the x,y axes, and the window boundary positions are specified in viewing coordinates. The view window can be placed anywhere on the view plane.

The projection matrix is typically a scale and perspective projection. The projection transformation converts the viewing frustum into a cuboid shape. The near end of the viewing frustum is smaller than the far end, which has the effect of expanding objects that are near to the camera. This is how perspective is applied to the scene. In the viewing frustum, the distance between the camera and the origin of the viewing transformation space is defined arbitrarily as D, so the projection matrix looks like the following illustration.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/D \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The viewing matrix translates the camera to the origin by translating in the z-direction by -D. The translation matrix is as follows.

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

Multiplying the translation matrix by the projection matrix (T*P) gives

the composite projection matrix. It looks like the following illustration.

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

The following illustration shows how the perspective transformation converts a viewing frustum to a new coordinate space. Notice that the frustum becomes cuboid and also that the origin moves from the upper-right corner of the scene to the center.

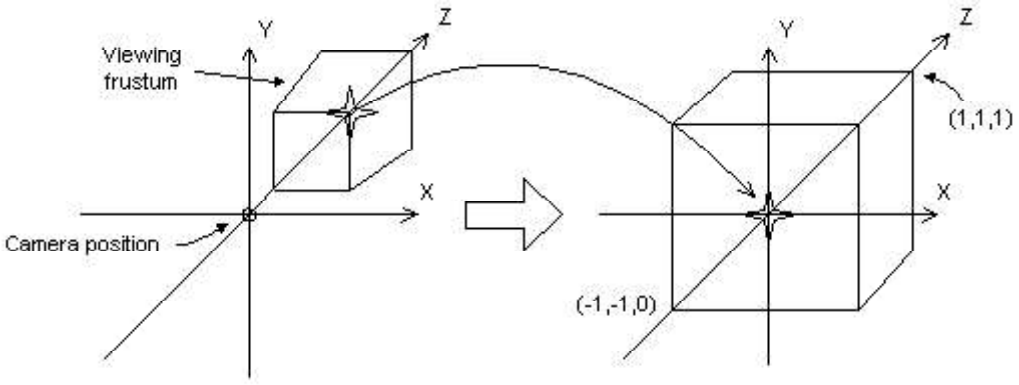


Figure 13.6: Cuboid

In the perspective transformation, the limits of the x- and y-directions are -1 and 1. The limits of the z-direction are 0 for the front plane and 1 for the back plane.

This matrix translates and scales objects based on a specified distance from the camera to the near clipping plane, but it does not consider the field of view (fov), and the z-values that it produces for objects in the distance can be nearly identical, making depth comparisons difficult. The following matrix addresses these issues, and it adjusts vertices to account for the aspect ratio of the viewport, making it a good choice for the perspective projection.

$$\begin{bmatrix} w & 0 & 0 & 0 \\ 0 & h & 0 & 0 \\ 0 & 0 & Q & 1 \\ 0 & 0 & -QZ_n & 0 \end{bmatrix}$$

In the following matrix, Zn is the z-value of the near clipping plane. The variables w, h, and Q have the following meanings. Note that fovw and fovk represent the viewport's horizontal and vertical fields of view, in radians.

$$w = \cot\left(fov_w / 2\right)$$
$$h = \cot\left(fov_k / 2\right)$$
$$Q = \frac{Z_t}{Z_f - Z_x}$$

For your application, using field-of-view angles to define the x- and y-scaling coefficients might not be as convenient as using the viewport's horizontal and vertical dimensions (in camera space). As the math works out, the following two formulas for w and h use the viewport's dimensions, and are equivalent to the preceding formulas.

$$w = \frac{2.Z_x}{V_w}$$
$$h = \frac{2.Z_x}{V_k}$$

In these formulas, Z_n represents the position of the near clipping plane, and the V_w and V_h variables represent the width and height of the viewport, in camera space.



CHECK YOUR PROGRESS

Q 2: What is projection matrix?

.....

.....

13.7 CLIPPING

When we have to display a large portion of the picture, then not only scaling & translation is necessary, the visible part of picture is also identified. This process is not easy. Certain parts of the image are inside, while others are partially inside. The lines or elements which are partially visible will be omitted.

For deciding the visible and invisible portion, a particular process called clipping is used. Clipping determines each element into the visible and invisible portion. Visible portion is selected. An invisible portion is discarded.

Types of Lines:

- Visible: A line or lines entirely inside the window is considered visible
- Invisible: A line entirely outside the window is considered invisible
- Clipped: A line partially inside the window and partially outside is clipped. For clipping point of intersection of a line with the window is determined.

Applications of clipping:

- It will extract part we desire.
- For identifying the visible and invisible area in the 3D object.
- For creating objects using solid modelling.
- For drawing operations.
- Operations related to the pointing of an object.
- For deleting, copying, moving part of an object.

Clipping can be applied to world co-ordinates. The contents inside the window will be mapped to device co-ordinates. Another alternative is a complete world co-ordinates picture is assigned to device co-ordinates, and then clipping of viewport boundaries is done.

Types of Clipping:

- Point Clipping
- Line Clipping
- Area Clipping (Polygon)
- Curve Clipping
- Text Clipping
- Exterior Clipping



CHECK YOUR PROGRESS

Q 3: What is the need of clipping?

.....

.....



13.8 LET US SUM

- A 3D projection is a design technique used to display a three-dimensional object on a two-dimensional surface.
- Viewing Coordinates system, [u, v, w], describes 3D objects with respect to a viewer.
- A viewing plane (projection plane) is set up perpendicular to w and aligned with (u, v).
- The view volume is bounded by six planes-the four sides plus the top and bottom of the truncated pyramid. These planes are called clipping planes because anything that lies on the wrong side of each plane is clipped away
- Once a 3D model has been completed, its co-ordinates need to be converted to 2 dimensions in order to display the scene on a flat computer monitor or to print it on paper. This process of converting from 3D to 2D is called projection.
- Clipping determines each element into the visible and invisible portion. Visible portion is selected.



13.9 FURTHER READINGS

1) D. Hearn and P.M. Backer, Computer Graphics, Prentice Hall of India, 1986.
2) W. Newman and R.F. Sproul, Principles of Interactive Computer Graphics, McGraw Hill Publication, 1980.
3) W.K. Giloi, Interactive Computer Graphics, Prentice Hall of India, 1978.



13.10 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No1: The view volume is bounded by six planes-the four sides plus the top and bottom of the truncated pyramid. These planes are called clipping planes because anything that lies on the wrong side of each plane is clipped away. The projection transformation maps the six sides of the truncated pyramid in eye coordinates to the six sides of the

clipping cube in clip coordinates.

Ans to Q No 2: The projection matrix is typically a scale and perspective projection. The projection transformation converts the viewing frustum into a cuboid shape. The near end of the viewing frustum is smaller than the far end, which has the effect of expanding objects that are near to the camera. This is how perspective is applied to the scene. In the viewing frustum, the distance between the camera and the origin of the viewing transformation space is defined arbitrarily as D, so the projection matrix looks like the following illustration.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/D \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ans to Q No 3: For deciding the visible and invisible portion, a particular process called clipping is used. Clipping determines each element into the visible and invisible portion. Visible portion is selected. An invisible portion is discarded.



13.11 MODEL QUESTIONS

Q 1: Explain 3D viewing.

Q 2: What is the need of viewing coordinates?

Q 3: What is 2D viewing?

Q 4: How can we convert 3D viewing to 2 D viewing?

Q 5: What is the need of view volume?

Q 6: what is the use of clipping? Explain few of its application.

*** ***** ***

UNIT 14: BASICS IN ANIMATION

UNIT STRUCTURE

14.1 Learning Objectives

14.2 Introduction

14.3 Animation Basic

14.4 Computer Animation

14.5 Principles of Animation

14.6 Types of Animation

14.7 Animation Software and Hardware

14.8 Common Terms in Animation

14.9 Let Us Sum Up

14.10 Further Reading

14.11 Answer to Check Your Progress

14.12 Model Questions

14.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Learn about animation and its basic terminologies
- Learn about different principles of animation
- Learn about types of animation
- Learn about software and hardware requirements for animation

14.2 INTRODUCTION

In the previous unit, we studied 3-D viewing. Along with this, we have gone through the topics of viewing pipeline, viewing coordinates, world coordinates, clipping, view volume, and different projection transforms.

In this unit, we will learn about animation and its types along with this we will see the hardware and software requirements of animation and the application of animation. In the next unit, we will explore the concept of animation design and the future of animation.

14.3 ANIMATION BASIC

Computer graphics is all about animation. It is defined as the act of making an object come alive. An animation is an object that is moving into or out of the screen. In animation, a series of images rapidly changed to create an illusion of a moment.

14.4 COMPUTER ANIMATION

Animation is the process of creating an illusion of motion images. All the cartoons and animation images are made from thousands of single images. And add them together and play step by step.

Animation can be used in various fields like education, marketing, and business to make it interesting. Following are some fields where animation can be used:

- Education purpose
- Architecture
- Cartoon
- Games
- Film
- Entertainment
- Advertisement
- Marketing
- Demonstration



EXERCISE

1. Write about the need of animation.

.....

.....

14.5 PRINCIPLES OF ANIMATION

Animators should follow principles to create a good animation. These principles evolved from past animation techniques. In 1981 two bright Disney animators Ollie Johnston and Frank Thomas introduced 12 basic principles

of animation to produce more realistic works. These principles are as follows:

- **Squash and Stretch:** This is the most important principle of animation. Drawing an object gives a sense of weight and volume.
- **Anticipation:** Here, the animator will create a starting scene. This scene shows that something will happen, but suddenly nothing happens.
- **Staging:** The animator creates the scene which attracts the audience.
- **Straight Ahead:** Here all frames are drawn from the beginning to the end and then fill the scene.
- **Flow through and overlapping action:** We can easily describe this principle by considering two objects' actions that have different speeds in a different scenes.
- **Slow in and Slow out:** We can easily describe this principle by considering an object, that has maximum acceleration in between and resisting on the beginning and end.
- **Arc:** All objects in animation will follow some arc in their action.
- **Secondary action:** As with one character's action, the second character's move shows the multiple dimension of animation.
- **Timing:** Perfect timing is important for playing a given action.
- **Exaggeration:** Developing a proper animation style creates extra reality in the animation scene.
- **Solid drawing:** Here objects are created in 3D form to get realistic visualization of the scene.
- **Appeal:** There is no need for the animated character to be the same as any real character. But it seems to be like it creates proper thinking in the audience's mind.



CHECK YOUR PROGRESS

Q 1: Define Animation.

.....

.....

14.6 TYPES OF ANIMATION

There are 4 types of animation. They are as follows-

- Traditional Animation
- Computer Animation
- Kinematics
- Morphing
  - **Traditional Animation:** Traditional animation can also be termed cell animation. Here, the animator has to draw every single frame to create an animated scene by hand.
  - **Computer Animation:** Here, we use computer graphics and animation techniques to draw all the frames to create animation. In computer animation, we do not need to draw each frame by hand. It is a creation of moving images.
  - **Kinematics:** It is the study of movement and motion of structures that have joints. For example, walking man. This type of animation is used in the areas like machines.
  - **Morphing:** It is an effect in which one image is transferred into another. For example, on Facebook, we have filters to check how we will look when we will become old, etc. here the image we will give will be morphed to get the desired result. The morphed images are built at a rate of 8 frames per second with each transition taking a total of 4 seconds.



CHECK YOUR PROGRESS

Q 2: Write about cell animation.

.....

.....

Q 3: What is morphing?

.....

.....

14.7 ANIMATION SOFTWARE AND HARDWARE

Software requirements:

Animators are digital artists. They require the processing of the images using different software. Image editing software examples are given below:

- **3Ds Max (Autodesk):** This animation software tool allows digital artists to produce 3D animations. It includes developing models and rendering them into 2D or 3D images.
- **Photoshop:** Here images are edited. Mainly we modify images in Photoshop. It is developed by Adobe. It is one of the most powerful and widely used photo editing software. It edits raster images.
- **Maya 3D:** This software is used to make an animation or can edit an animated scene. It is used to develop video games, animated films, etc. it runs on Windows, MAC OS, and Linux. It has cached playback which allows the user to check the animation in the viewport. This is created by Alias Systems Corporation. But later Autodesk took all rights.
- **CAD:** It is used to design images like a house or any structure. It creates the design in 2D and 3D. It is used by Construction companies, architects, and civil engineers.
- **CorelDraw:** it is developed by Corel Corporation. It is used for editing vector graphics. Touch-friendly GU interface and multi-monitor are some of the features of CorelDraw.

Hardware requirements for animation:

We will require the following input and output devices for animation:

Input devices:

- Touch panels
- Light pens
- Graphics tablet
- Microphone
- Film recorder
- Mouse
- Joysticks

- Keyboard
- Data gloves
- Image scanner

Output devices:

- Cathode Ray Tube(CRT)
- Vector scan display/ random scan display
- Raster scan display
- Coloured monitors
- LED, LCD, TFT Screen

Processor (CPU)

This is the most important part of our machine. At least a quad-core that operates at 2Ghz is needed that is capable of running a 64-bit operating system.

Graphics card (GPU)

A good GPU will enable the machine to perform modeling, mapping, and animation work in 3D spaces smoothly and with more stability. We should go for a dedicated GPU to have at least 2GB of RAM.

Memory (RAM)

We need a minimum of 8GB RAM.

Storage/Hard Drive (HDD or SSD)

At least 1TB of storage is needed.



CHECK YOUR PROGRESS

Q 4: Write about the minimum software and hardware requirements for animation.

.....

.....

14.8 COMMON TERMS IN ANIMATION

Following are some common terminologies used in animation.

- **Append:** Append is an additional keyframe. It is added to either the end or the start of the animation. And this increases the total duration of the animation.

- **Frame:** A frame is an image. It represents a moment in a particular time. The frames can be connected to other frames to generate a video.
- **Keyframe:** It is a visual waypoint along the animation path scene. When an animation is played, values such as the location of the camera, the current map time, the current map range, and layer transparencies are interpolated between the stored states using a configurable transition type.
- **Maintain speed:** The camera travel speed is maintained when appending keyframes to the animation.
- **Transition time:** It is the period between the keyframes. Usually, the transition time is 3 seconds. But we can set different transition times or the transition time can be automatically calculated using the travel speed of the camera.
- **Transition type:** The type of transition determines how frames are interpolated between keyframes. Normally Fixed, Adjustable, Linear, Hop, Stepped, and Hold transition types are there.
- **Animation Timeline pane:** It contains the animation timeline control, the keyframe gallery, and the overlay timeline if overlays exist for the animation.
- **Animation timeline control:** It displays the full length of the animation, including the timing between individual keyframes.
- **Cluster:** It is a single representative icon on the timeline when keyframes are very close and become automatically grouped.
- **Overlay:** Overlays are on-screen text, image, and shape elements that add detail and information to the animation.
- **Control point:** These points allow the shape of the Bezier curve to be customized.
- **Look at point:** It overrides the interpolation of the camera's heading and pitch between keyframes. It is used across consecutive keyframes to maintain focus on an important feature as the camera moves.



CHECK YOUR PROGRESS

Q 5: Write about the animation timeline control.

.....

.....



14.9 LET US SUM UP

- Animation is the process of creating an illusion of motion images. Here the figures appear as moving images
- Traditional animation can also be referred to as cell animation. This type of animation requires the animator to draw every single frame by hand to create an animated scene.
- Games and cartoons are some of the major examples of animation.
- Append an additional keyframe to your animation. A new keyframe is added to either the end or the start of the animation, and the total duration of the animation increases.
- A single representative icon on the timeline when keyframes are very close or overlapping and become automatically grouped.
- Overlays are on-screen text, image, and shape elements that add detail and information to the animation.



14.10 FURTHER READINGS

1) D. Hearn and P.M. Backer, Computer Graphics, Prentice Hall of India, 1986.
2) W. Newman and R.F. Sproul, Principles of Interactive Computer Graphics, McGraw Hill Publication, 1980.
3) W.K. Giloi, Interactive Computer Graphics, Prentice Hall of India, 1978.



14.11 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: Animation is the process of transforming still images into moving images. When still images are displayed rapidly on media, such as film or video, they create the illusion of motion.

Ans to Q No 2: Traditional animation can also be termed cell animation. Here, the animator has to draw every single frame to create an animated scene by hand.

Ans to Q No 3: It is an effect in which one image is transferred into another. For example, on Facebook, we have filters to check how we will look when we will become old, etc. here the image we will give will be morphed to get the desired result.

Ans to Q No 4:

Hardware requirements are as follows:

- A quad-core that operates at 2Ghz at the very least. One that is capable of running a 64-bit operating system.
- GPU to have at least 2GB of RAM.
- A minimum of 8GB RAM.
- At least 1TB of storage (SSD).

  Following are some examples of software that helps in creating animation:
- FlipBook (Digicel)
- Maya (Autodesk)
- Moho Pro 13 (Smith Micro)
- After Effects (Adobe)

Ans to Q No 5: The animation timeline control displays the full length of the animation, including the timing between individual keyframes. It groups them into a cluster.



14.12 MODEL QUESTIONS

Q 1: Write about the principles of animation.

Q 2: Describe the different types of animation.

Q 3: Define 3D animation.

Q 4: Define frame.

Q 5: Write about five tools that are used for animation creation.

Q 6: What do you mean by look at point?

Q 7: Define transition time.

Q 8: Differentiate between 2D and 3D animation.

Q 9: Give some major applications of animation.

*** ***** ***

UNIT 15: INTRODUCTION TO JAVA

UNIT STRUCTURE

15.1 Learning objectives

15.2 Introduction

15.3 Design of Animation sequence

15.4 Morphing

15.5 Application of Computer Animation

15.6 Future of Animation

15.7 Let Us Sum Up

15.8 Further Reading

15.9 Answer to Check Your Progress

15.10 Model Questions

15.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- About the animation
- About the design of animation sequence
- About morphing
- About computer animation application

15.2 INTRODUCTION

Animation is a method where images are manipulated to appear as moving object. Traditionally, animation are drawn or painted by hand on transparent celluloid sheets. Most of the animations of today's world are made with computer-generated imagery (CGI). A computer can be 2D and 3D animation. A 2D computer animation can be used for stylistic reasons, low bandwidth, or faster real-time renderings. The examples of computer animation are presented below.

- Motion graphics.
- Sprite animations.
- CSS animations.

- JavaScript animations.
- SVG animations.
- WebGL animations.
- UI animations.
- Web-based game animations.
- A cartoon is an example of animated film.

15.3 Design of Animation Sequences

There is a standard approach for carrying out animations. The following four steps are followed in designing an animation sequence:

- **Layout of Storyboard :** It is the rough diagram needed to show what exactly we want to perform. It defines what we are going to do or what are the events going to happen in a sequence of the motion. It includes a list of basic ideas or a set of rough diagrams depending on what type of animation will be produced.
- **Definitions of Object :** The different objects which will be used in the action should be defined with reference to basic shapes, such as rectangle, circle, spline, etc. The designated shape of an object is corresponding to the movement of each object.
- **Specifications of Key-frame :** The starting and the ending point of the motion sequence is the key-frame specification. It defines the explicit drawing of an animation sequence at a certain time.
- **In-between frames Generation :** The frames in the middle of the key-frames, i.e., starting and the ending frames of the motion sequence which helps to create the illusion of motion are called the in-between frames of the film.

  Depending on the application, many other tasks might also be required to carry out an animation design. Many of these functions are nowadays computer generated which are required to produce animations. Some animations might require functions like motion verification, editing, producing and synchronization of sound tracks.



CHECK YOUR PROGRESS

Q 1: True or False

a. Flash is 2d Animation.

b. Maya is a animation software

c. Claymation is stop motion animation

Q 2: What is an object in animation?

.....

.....

Q 3: What is an animation?

.....

.....

15.4 Morphing

Morphing is the technique used to transform one image to another. Morphing is basically a geometric interpolation technique used in motion sequences for the transformation of object shapes smoothly and continuously with no apparent gaps or spaces in-between frames. The morphing process is simple and it consists of not so complicated linear interpolations that produce intermediate forms between two maximal forms. In this technique an image is step by step altered without any observable individual changes during the process in a motion sequence of an animation video or film to create the illusion of movement and changes of shape of objects.

15.5 Application of Computer Animation

- Animation gives life to our imagination. Anything we imagine could be converted into animated artwork and presented. Animation has an important role in modern day education, entertainment, advertisement, scientific visualization, gaming, creative art etc.
- Animation helps teachers to teach and make students learn easily by using computer animated videos and displays of any topic they want.

Animation is used in training centres, schools and colleges.

- Then comes the entertainment industry, where animations play a vital role. Many motion pictures, music videos, television shows; especially children cartoon shows use animation methods.
- In advertisements too, animation is used. Suppose a new mobile phone is getting launched, so to show its functions, its usages, an animation video is often created to attract the customers.
- Scientists use animation videos to show how a particular scientific reaction occurs, for example, reactions of some atoms could be shown through an animated video for the visualization of students and learners.
- Animation also has a vast use in gaming. The characters in computer games and all the graphical events that happen in a computer game are the result of animation. Graphic designers use animation to create their art forms.
- Describing any kind of financial, statistical, mathematical, scientific and economic data is made easy and presentable by animated presentations.

15.6 Future of Animation

The following points can be considered as the future of the animation.

- The animation world is a fast growing field of computer graphics. It is a valuable and easy tool for storytelling and artwork. Animation has the prospect of telling stories in a realistic way.
- Animation, both as an industry and art form is in great demand nowadays. Animated motion pictures capture the attention of audiences.
- Animation helps in communicating information effectively and quickly.
- The animation industry has a bright future since it has been growing day by day with improvement in quality.
- People around the world accept it both as an art form and also as an industry by paying more attention to the animation movies and computer games.

- Animation is getting interesting, popular and widespread even during and after the pandemic.



CHECK YOUR PROGRESS

Q 1: Define two application of animation

.....

.....

Q 2: What is AR?

.....

.....

Q 3: What is VR?

.....

.....



15.7 LET US SUM UP

- Animation is a method where images are manipulated to appear as moving object. Traditionally, animation are drawn or painted by hand on transparent celluloid sheets.
- The examples of animation are presented below.
  - Motion graphics.
  - Sprite animations.
  - CSS animations.
  - JavaScript animations.
  - SVG animations.
  - WebGL animations.
  - UI animations.
  - Web-based game animations.
  - A cartoon is an example of animated film.
- There are four steps for designing an animation
  - Layout of Storyboard :
  - Definitions of Object
  - Specifications of Key-frame

- In-between frames Generation

- Morphing is the technique used to transform one image to another. Morphing is basically a geometric interpolation technique used in motion sequences for the transformation of object shapes smoothly and continuously with no apparent gaps or spaces in-between frames.



15.8 FURTHER READING

1) Computer Animation: Algorithms and Techniques by Book by Rick Parent.



15.9 ANSWER TO CHECK YOUR PROGRESS

Ans to Q No 1: I) True ii) True iii) True

Ans to Q No 2: The different objects which will be used in the action should be defined with reference to basic shapes, such as rectangle, circle, spline, etc. The designated shape of an object is corresponding to the movement of each object.

Ans to Q No 3: Animation is a method where images are manipulated to appear as moving object.

Ans to Q No 4:

- Animation gives life to our imagination. Anything we imagine could be converted into animated artwork and presented. Animation has an important role in modern day education, entertainment, advertisement, scientific visualization, gaming, creative art etc.
- Animation helps teachers to teach and make students learn easily by using computer animated videos and displays of any topic they want. Animation is used in training centres, schools and colleges.

Ans to Q No 5: Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somato sensory and olfactory.

Ans to Q No 6: Virtual reality is a simulated experience that can be similar to or completely different from the real world. Applications of virtual reality include entertainment, education and business.



15.10 MODEL QUESTIONS

Q 1: What do you mean by animation?

Q 2: What is the difference computer animation and animation>

Q 3: What are the design techniques of animation?

Q 4: What is morphing?

Q 5: Explain the computer application of animation.

Q 6: Give some example of computer animation.

Q 7: Explain the AR and VR.

Q 8: Explain the application of AR VR in animation.

*** ***** ***

Name of the Paper: **Course Code:**

Centre for Internal Quality Assurance (CIQA)
Krishna Kanta Handiqui State Open University
City Office, Housefed Complex, Guwahati- 06

Learner's Feedback on Course

Dear Learner,

Regarding the course as mentioned above, we would like to know your opinions and comments so as to improve the quality of self learning materials in future. Please respond to the following statements by ticking the number you feel most reflect your opinion. After completion of the additional comments, please detach the page and send/mail the same to us at the address given below.

The Director, Centre for Internal Quality Assurance, KKHandiqui State Open University
Housefed Complex, Dispur, Guwahati- 781006
(E-mail id: ciqa@kkhsou.in)

1. Approximately how many hours did you spend for studying the units in the course? | | | |
2. Please give your opinions (by √ mark) to the following items based on your reading of the block:

Sl. No.	Statements	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
I.	The SLMs of the course fulfil the learning objectives set out in the beginning of each unit					
II.	The units of the SLM could satisfy my academic needs and expectations					
III.	The Unit writers have excellent knowledge about the course contents					
IV.	Language and contents of the units were presented at a level which I could readily understand					
V.	Ample opportunity for participation in the activities provided in the units					
VI.	Used enough Illustrations (Diagrams, tables etc.) for conceptual clarity					
VII.	Quality of content is engaging, relevant, and up-to-date					
VIII.	The self check questions are very helpful					
IX.	The Possible/model questions and the answers to check my progress have benefited me a lot					

Additional Comments: *(Please feel free to provide your open comments)*

1. Which aspects of the SLM, according to you, worked well?

   ---

2. What sort of changes/improvements do you feel KKHSOU could implement to improve the overall quality of the SLM?

Thank you for taking the time to complete this form.

******