Unit-4

Three Dimensional Transformations:

Methods for geometric transforamtions and object modelling in 3D are extended from 2D methods by including
the considerations for the z coordinate.
Basic geometric transformations are: Translation, Rotation, Scaling

Basic Transformations

Translation

We translate a 3D point by adding translation distances, tx, ty, and tz, to the original coordinate position (x,y,z):
x' = x + tx, y' = y + ty, z' = z + tz
Alternatively, translation can also be specified by the transformation matrix in the following formula:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Exercise: translate a triangle with vertices at original coordinates (10,25,5), (5,10,5), (20,10,10) by
tx=15, ty=5,tz=5. For verification, roughly plot the x and y values of the original and resultant triangles, and imagine the locations of z values.
Scaling With Respect to the Origin
We scale a 3D object with respect to the origin by setting the scaling factors sx, sy and sz, which are
multiplied to the original vertex coordinate positions (x,y,z):
x' = x * sx, y' = y * sy, z' = z * sz
Alternatively, this scaling can also be specified by the transformation matrix in the following formula:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \begin{pmatrix} s_X & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_Z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Exercise: Scale a triangle with vertices at original coordinates (10,25,5), (5,10,5), (20,10,10) by s_X=1.5, s_y=2, and s_Z=0.5 with respect to the origin. For verification, roughly plot the x and y values of the original and resultant triangles, and imagine the locations of z values.

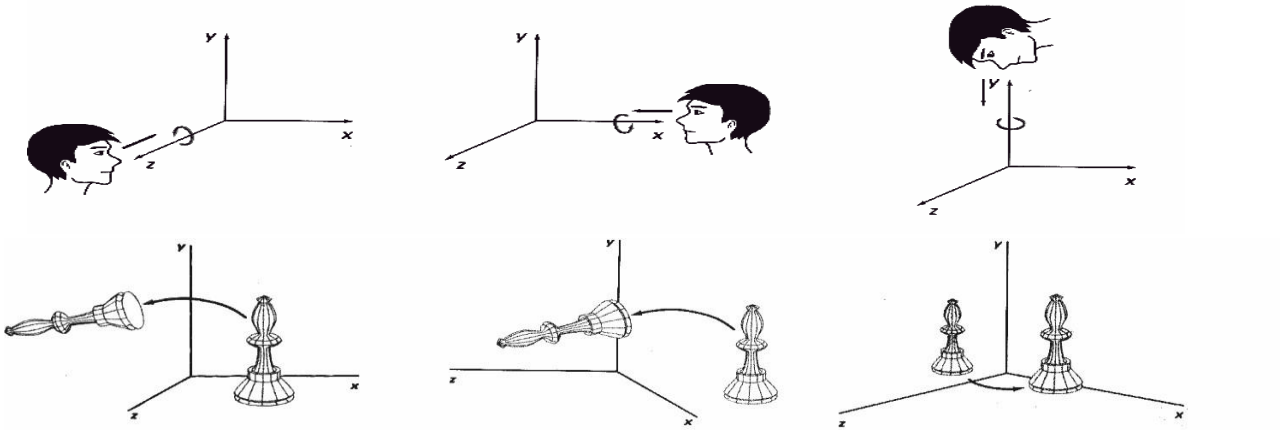
Scaling with respect to a Selected Fixed Position

Exercise: What are the steps to perform scaling with respect to a selected fixed position? Check your answer with the text book.

Exercise: Scale a triangle with vertices at original coordinates (10,25,5), (5,10,5), (20,10,10) by s_X=1.5, s_Y=2, and s_Z=0.5 with respect to the centre of the triangle. For verification, roughly plot the x and y values of the original and resultant triangles, and imagine the locations of z values.

Coordinate-Axes Rotations

A 3D rotation can be specified around any line in space. The easiest rotation axes to handle are the coordinate axes.



Z-axis rotation: x' = x cos ? - y sin ?,

y' = x sin ? + y cos ?, and

z' = z

write matrix for z- axis rotation

X-axis rotation:

y' = y cos ? - z sin ?,

z' = y sin ? + z cos ?, and

x' = x

write matrix for x- axis rotation

Y-axis rotation:

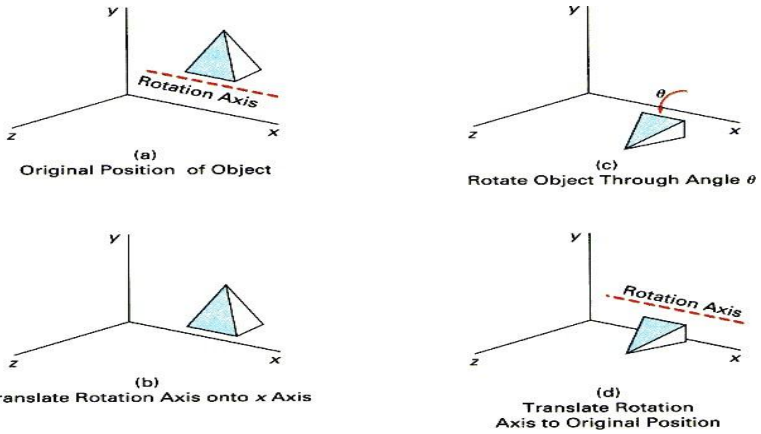
z' = z cos ? - x sin ?,

x' = z sin ? + x cos ?, and

y' = y

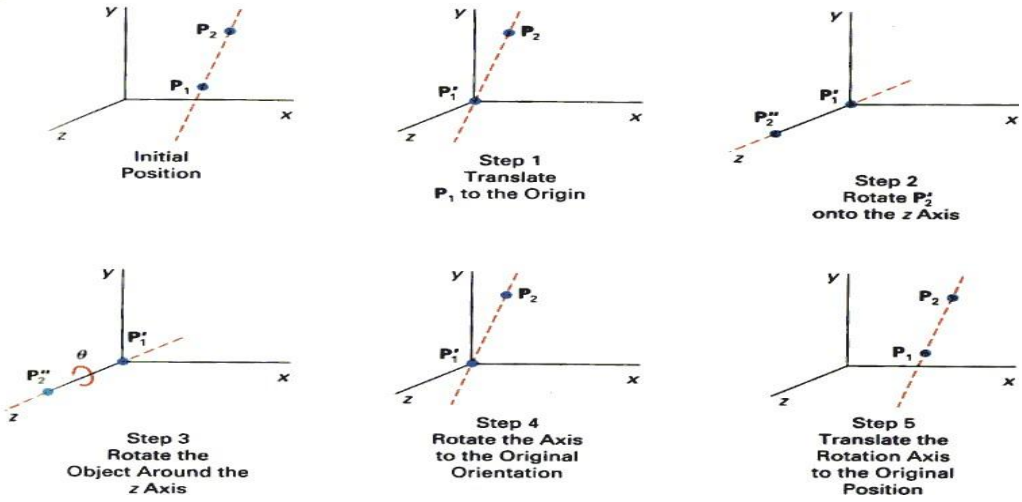
write matrix for y- axis rotation

3D Rotations About an Axis Which is Parallel to an Axis



Step 1. Translate the object so that the rotation axis coincides with the parallel coordinate axis.
Step 2. Perform the specified rotation about that axis.
Step 3. Translate the object so that the rotation axis is moved back to its original position.

General 3D Rotations



Step 1. Translate the object so that the rotation axis passes through the coordinate origin.
Step 2. Rotate the object so that the axis of rotation coincides with one of the coordinate axes.
Step 3. Perform the specified rotation about that coordinate axis.
Step 4. Rotate the object so that the rotation axis is brought back to its original orientation.
Step 5. Translate the object so that the rotation axis is brought back to its original position.

Three-Dimensional Viewing

Viewing in 3D involves the following considerations:

- We can view an object from any spatial position, eg. In front of an object, Behind the object, In the middle of a group of objects, Inside an object, etc.
- 3D descriptions of objects must be projected onto the flat viewing surface of the output device.
- The clipping boundaries enclose a volume of space

Viewing Pipeline

Modelling Coordinates

⇓

Modelling Transformations

⇓

World Coordinates

⇓

Viewing Transformation

⇓

Viewing Coordinates

⇓

Projection Transformation

⇓

Projection Coordinates

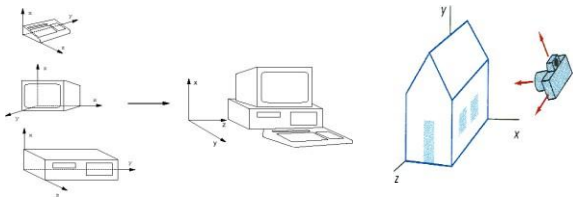
⇓

Workstation Transformation

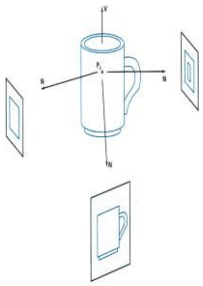
⇓

Device Coordinates

Explanation



Modelling Transformation and Viewing Transformation can be done by 3D transformations. The viewing-coordinate system is used in graphics packages as a reference for specifying the observer viewing position and the position of the projection plane. Projection operations convert the viewing-coordinate description (3D) to coordinate positions on the projection plane (2D). (Usually combined with clipping, visual-surface identification, and surface-rendering)Workstation transformation maps the coordinate positions on the

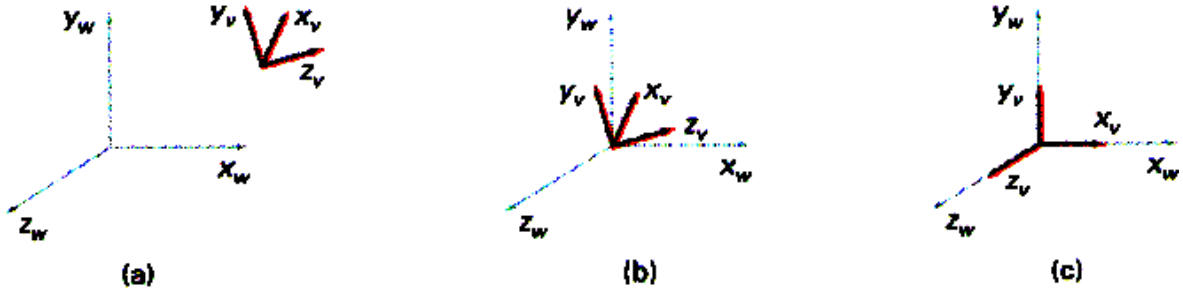


projection plane to the output device

Viewing Transformation

Conversion of objection descriptions from world to viewing coordinates is equivalent to a transformation that superimposes the viewing reference frame onto the world frame using the basic
geometric translate-rotate operations:

1. Translate the view reference point to the origin of the world-coordinate system.
2. Apply rotations to align the xv, yv, and zv axes (viewing coordinate system) with the world xw, yw,
zw axes, respectively.

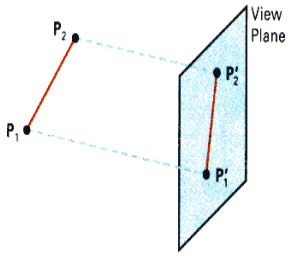


Projections

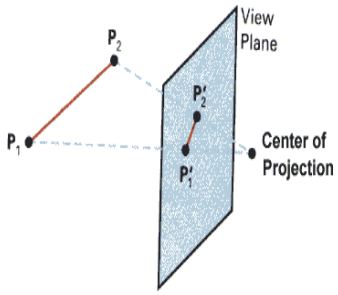
Projection operations convert the viewing-coordinate description (3D) to coordinate positions on the
projection plane (2D). There are 2 basic projection methods:

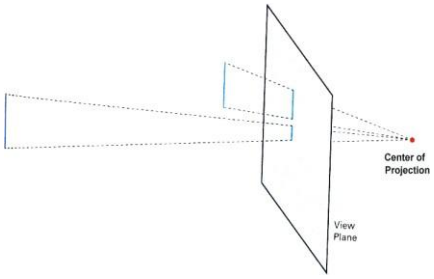
1. Parallel Projection transforms object positions to the view plane along parallel lines.

A parallel projection preserves relative proportions of objects. Accurate views of the various sides of
an object are obtained with a parallel projection. But not a realistic representation



2. Perspective Projection transforms object positions to the view plane while converging to a center
point of projection. Perspective projection produces realistic views but does not preserve relative proportions. Projections of distant objects are smaller than the projections of objects of the same size that are closer to the
projection plane.

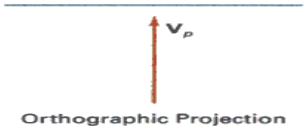




Parallel Projection

Classification:

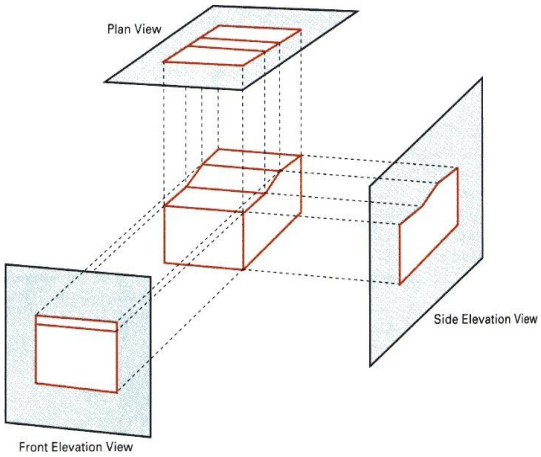
Orthographic Parallel Projection and Oblique Projection:

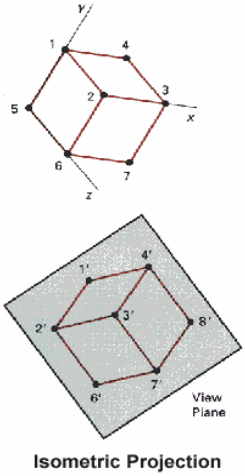


Orthographic parallel projections are done by projecting points along parallel lines that are perpendicular to the projection plane.

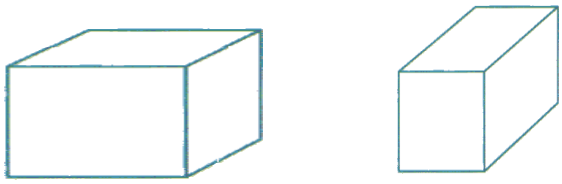
Oblique projections are obtained by projecting along parallel lines that are NOT perpendicular to the

projection plane.Some special Orthographic Parallel Projections involve Plan View (Top projection), Side Elevations, and Isometric Projection:





The following results can be obtained from oblique projections of a cube:



Perspective Projection

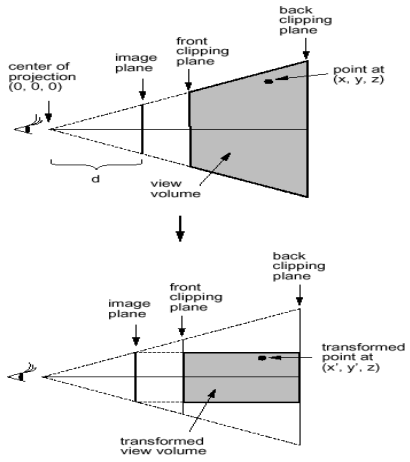
Perspective projection is done in 2 steps: Perspective transformation and Parallel projection. These

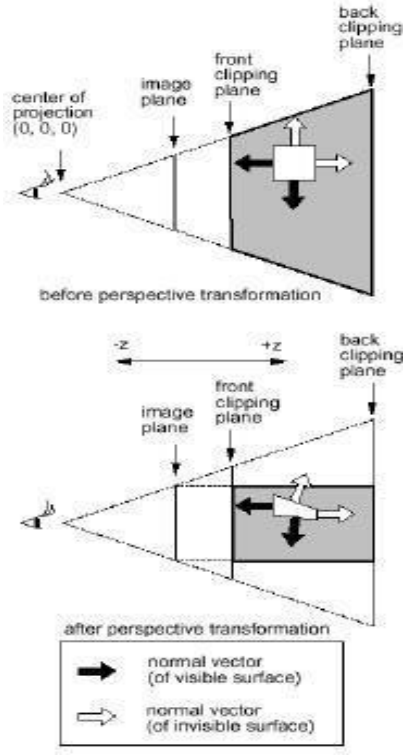
steps are described in the following section.

Perspective Transformation and Perspective Projection To produce perspective viewing effect, after Modelling Transformation, Viewing Transformation is carried out to transform objects from the world coordinate system to the viewing coordinate system. Afterwards, objects in the scene are further processed with Perspective Transformation: the view volume in the shape of a

frustum becomes a regular parallelepiped. The transformation equations are shown as follows and are applied to every vertex of each object:
x' = x * (d/z),
y' = y * (d/z),
z' = z

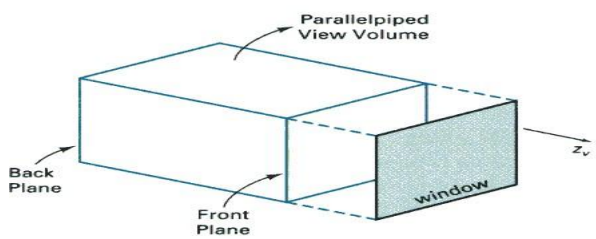
Where (x,y,z) is the original position of a vertex, (x',y',z') is the transformed position of the vertex,
and d is the distance of image plane from the center of projection.
Note that:
Perspective transformation is different from perspective projection: Perspective projection projects a
3D object onto a 2D plane perspectively. Perspective transformation converts a 3D object into a deformed 3D object. After the transformation, the depth value of an object remains unchanged. Before the perspective transformation, all the projection lines converge to the center of projection.
After the transformation, all the projection lines are parallel to each other. Finally we can apply parallel projection to project the object onto a 2D image plane. Perspective Projection = Perspective Transformation + Parallel Projection



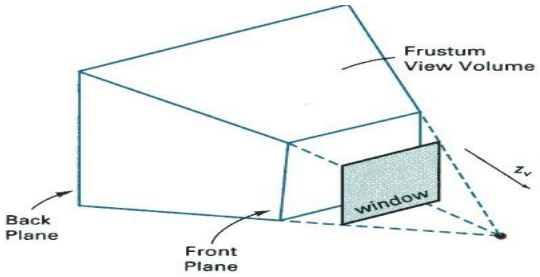


View Volumes

View window - A rectangular area in the view plane which controls how much of the scene is viewed.
The edges of the view window are parallel to the xv and yv viewing axes. View volume - formed by the view window and the type of projection to be used. Only those objects within the view volume will appear in the generated display. So we can exclude objects that are beyond the view volume when we render the objects in the scene. A finite view volume is obtained by bounding with front plane and back plane (or the near plane and the far plane). Hence a view volume is bounded by 6 planes => rectangular parallelepiped or a frustum, for parallel projection and perspective projection respectively. Some



Parallel Projection



Perspective
Projection

Some facts:
Perspective effects depend on the positioning of the center point of projection. If it is close to the view plane, perspective effects are emphasized, ie. closer objects will appear larger than more distant
objects of the same size. The projected size of an object is also affected by the relative position of the object and the view plane.

'Viewing' a static view:
The view plane is usually placed at the viewing-coordinate origin and the center of projection is positioned to obtain the amount of perspective desired.

'Viewing' an animation sequence:
Usually the center of projection point is placed at the viewing-coordinate origin and the view plane is
placed in front of the scene. The size of the view window is adjusted to obtain the amount of scene
desired. We move through the scene by moving the viewing reference frame (ie. the viewing coordinate system).

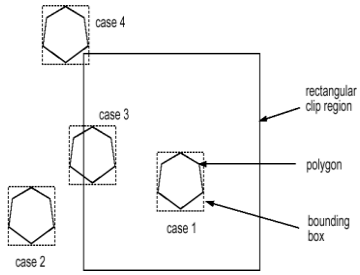
Some facts:
Perspective effects depend on the positioning of the center point of projection. If it is close to the view plane, perspective effects are emphasized, ie. closer objects will appear larger than more distant
objects of the same size. The projected size of an object is also affected by the relative position of the object and the view
plane.

'Viewing' a static view:
The view plane is usually placed at the viewing-coordinate origin and the center of projection is positioned to obtain the amount of perspective desired.

'Viewing' an animation sequence:
Usually the center of projection point is placed at the viewing-coordinate origin and the view plane is
placed in front of the scene. The size of the view window is adjusted to obtain the amount of scene
desired. We move through the scene by moving the viewing reference frame (ie. the viewing coordinate system).

Clipping

The purpose of 3D clipping is to identify and saveall surface segments within the view volume for display on the output device. All parts of objects that are outside the view volume are discarded. Thus the computing time is saved. 3D clipping is based on 2D clipping. To



understand the basic concept we consider the following algorithm:

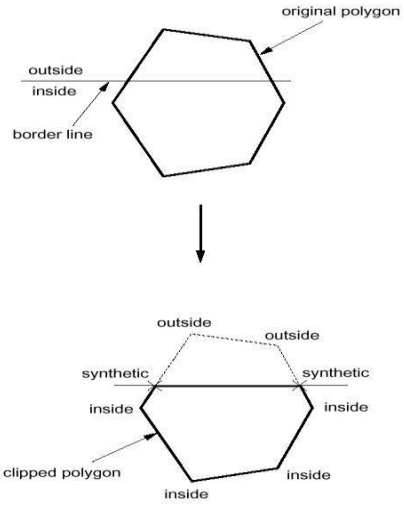
Polygon Clipping

Assuming the clip region is a rectangular area,

1. The rectangular clip region can be represented by xmin, xmax, ymin and ymax.
2. Find the bounding box for the polygon: ie. the smallest rectangle enclosing the entire polygon.
3. Compare the bounding box with the clip region (by comparing their xmin, xmax, ymin and ymax).
4. If the bounding box for the polygon is completely outside the clip region (case 2), the polygon is

outside the clip region and no clipping is needed.

5. If the bounding box for the polygon is completely inside the clip region (case 1), the polygon is



inside the clip region and no clipping is needed.

6. Otherwise, the bounding box for the polygon overlaps with the clip region (cases 3 and 4) and the

polygon is likely to be partly inside and partly outside of the clip region. In that case, we clip the polygon against each of the 4 border lines of the clip region in sequence as follows:

Using the first vertex as the current vertex. If the point is in the inside of the border line, mark it as 'inside'. If it is outside, markit as 'outside'. Check next vertex. Again mark it 'inside' or 'outside' accordingly. Compare the current and the next vertices. If one is marked 'inside'

and the other 'outside', the edge joining the 2 vertices crosses the border line. In this case, we need to calculate where the edge intersects the border (ie. intersection between 2 lines). The intersection point becomes a new vertex. We mark it 'synthetic'. Now we set the next vertex as the current vertex and the followingvertex as the next vertex, and we repeat the same operations until all the edges of the polygon have been considered. After the whole polygon has been clipped by a border, we throw away all the vertices marked 'outside' while keeping those marked as 'inside' or 'synthetic' to create a new polygon.

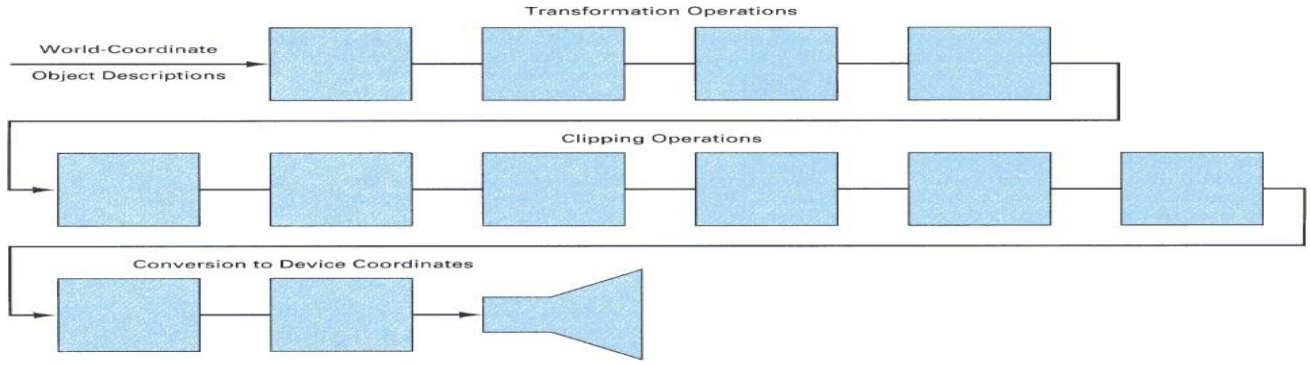
We repeat the clipping process with the new polygon against the next border line of the clip region.

7. This clipping operation results in a polygon which is totally inside the clip region.

Hardware Implementations

Most graphics processes are now implemented in graphics chip sets. Hardware systems are now designed to transform, clip, and project objects to the output device for either 3D or 2D applications.

In a typical arrangement, each of the individual chips in a chip set is responsible for geometric transformations, projection transformation, clipping, visible-surface identification, surface-shading
procedure, octree representation processing, or ray-tracing etc., in a pipe-line way.



A hardware implementation of three-dimensional viewing operations using 12 chips for the coordinate transformations and clipping operations.