Computer Graphics

Viewing (Projections)

Viewing (Projections)

- Def #1: a technique (or process) used to transform a 3D object into a 2D plane.

- Def #2: a mapping of points P (x,y,z) onto its image P’ (x’,y’,z’) in the projection plane or view plane.

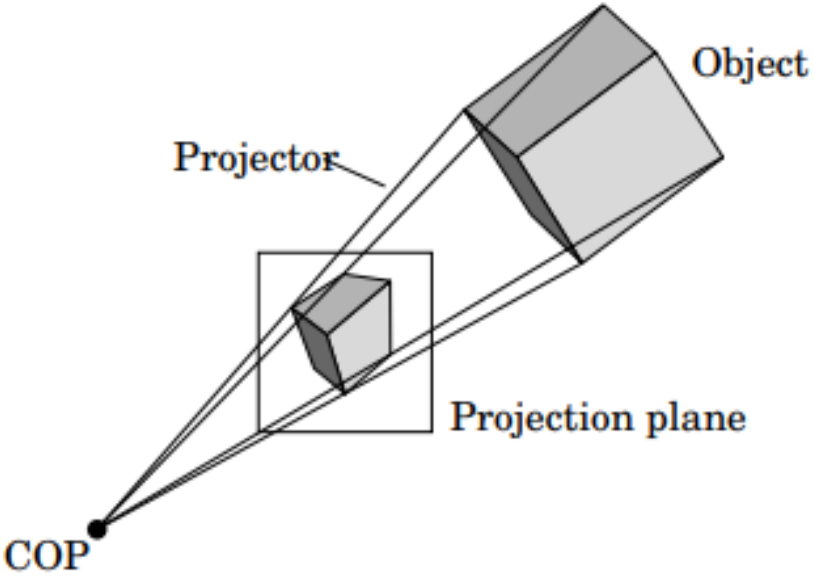
- two types of viewing: classical and computer.

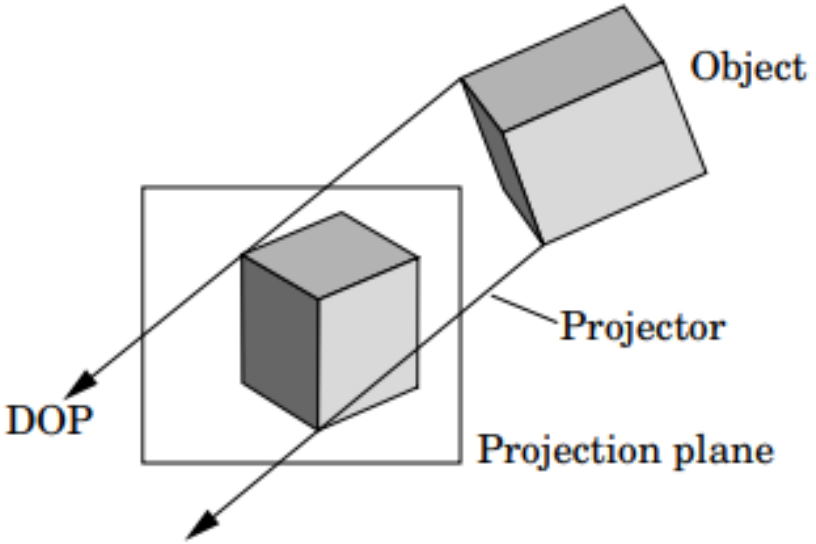
- existed before computers (photography, animation, architectural drawings, mechanical designs, cartography…).

- these jobs formerly done by hand drawing are now done with the aid of computer graphics.

- both types of viewing requires three basic elements: (one or more) object(s), a viewer (with a projection surface), and light sources (projectors from the object(s) to the projection surface).

Classical and Computer Viewing





- centre of projection (COP): centre of the lens or eye, or (in computer graphics) the origin of the camera frame. The projectors meet at the COP.

- the viewer can be at an infinite distance from the objects. As we move the COP to infinity, the projectors become parallel and it can be replaced by a direction of projection (DOP).

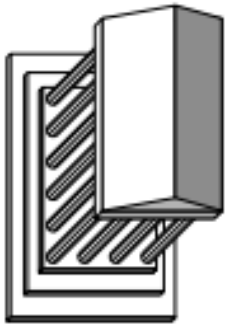
- perspective views: views with a finite COP, parallel views: with a COP at infinity.

Classical Viewing

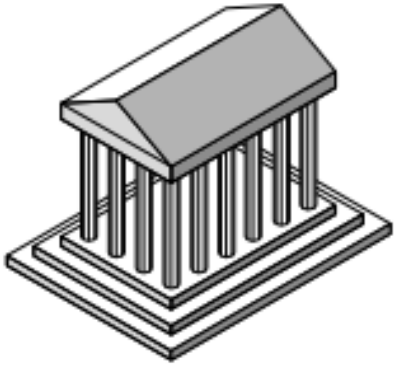
- there is an underlying notion of a principal face.
- typical in architecture, where objects are often buildings with planar faces: front, back, top, bottom, right, and left.
- such objects have three orthogonal directions associated with them.

Front elevation

Elevation oblique

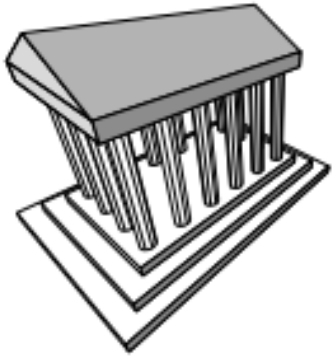


Plan oblique



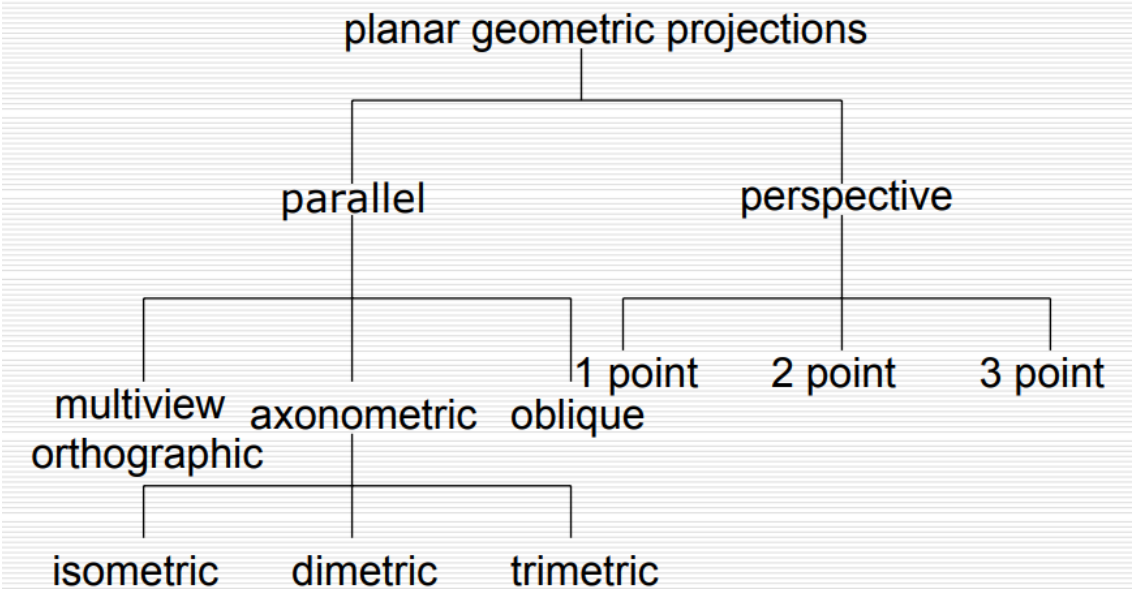
Isometric

One-point perspective



Three-point perspective

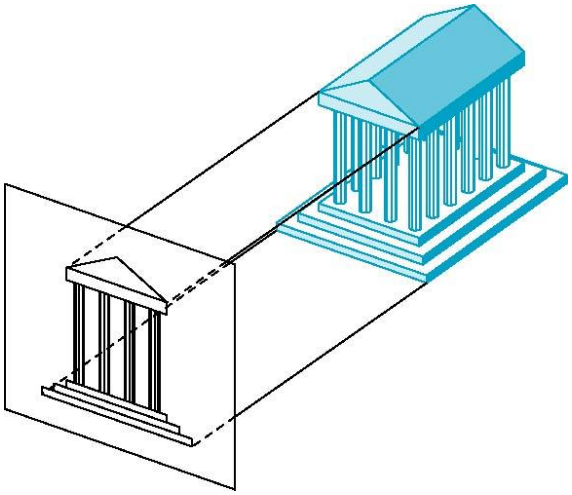
Projection Classification



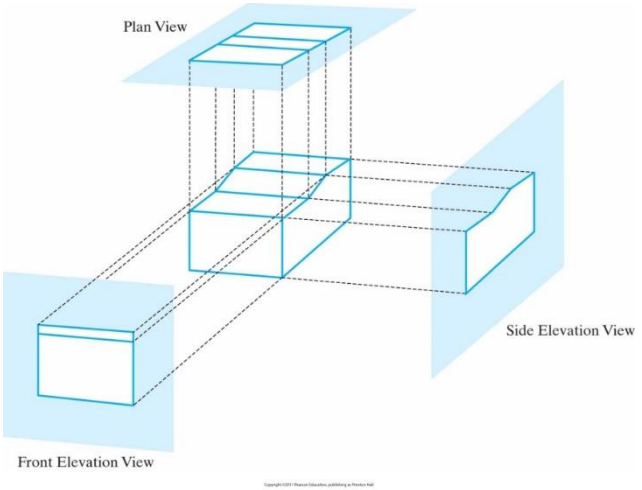
Taxonomy of Planar Geometric Projections

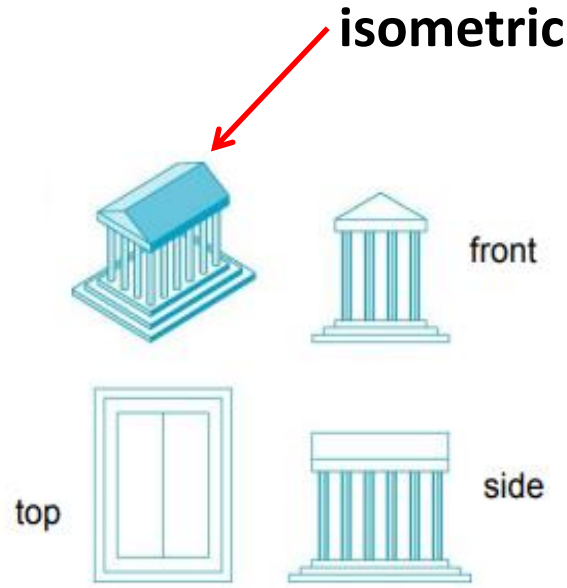
- two main classes: parallel and perspective.
- parallel projection: the projection lines are parallel to each other. The lines that are parallel to the object are also parallel to the drawing.
- the view is less realistic: no foreshortening and amount of distortion within the object is the least.
- good for accurate measurements.

Orthographic (Orthogonal) Projections

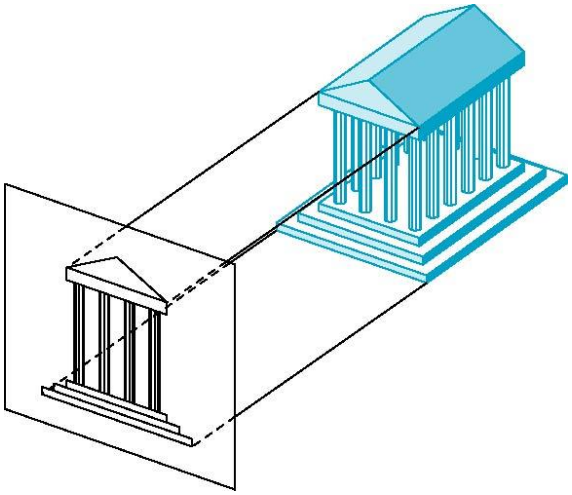


- first (type of) parallel projection.
- DOP is perpendicular to the projection plane.
- projection plane is placed parallel to one of the principal faces of the object: Elevations (front, side) and plan view (top).
- includes these three views and so it is called multiview orthographic projection.

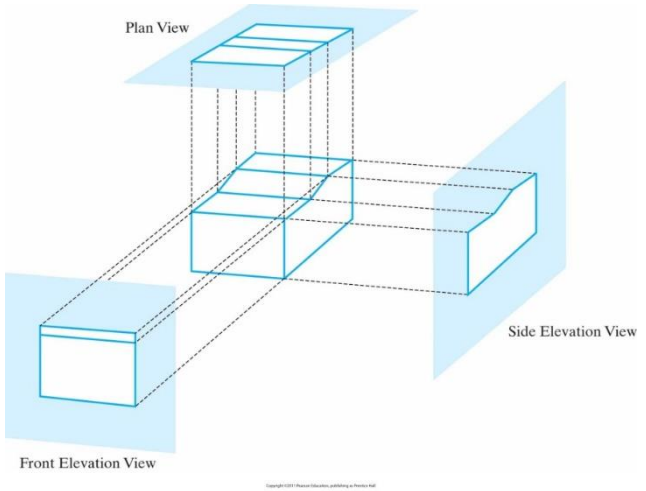


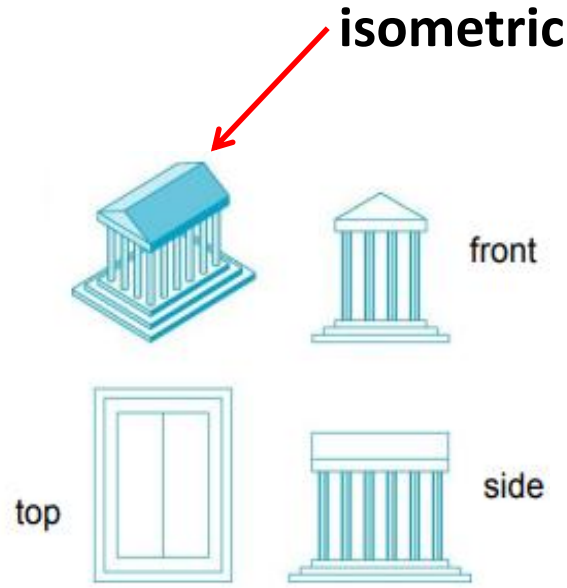


Orthographic (Orthogonal) Projections

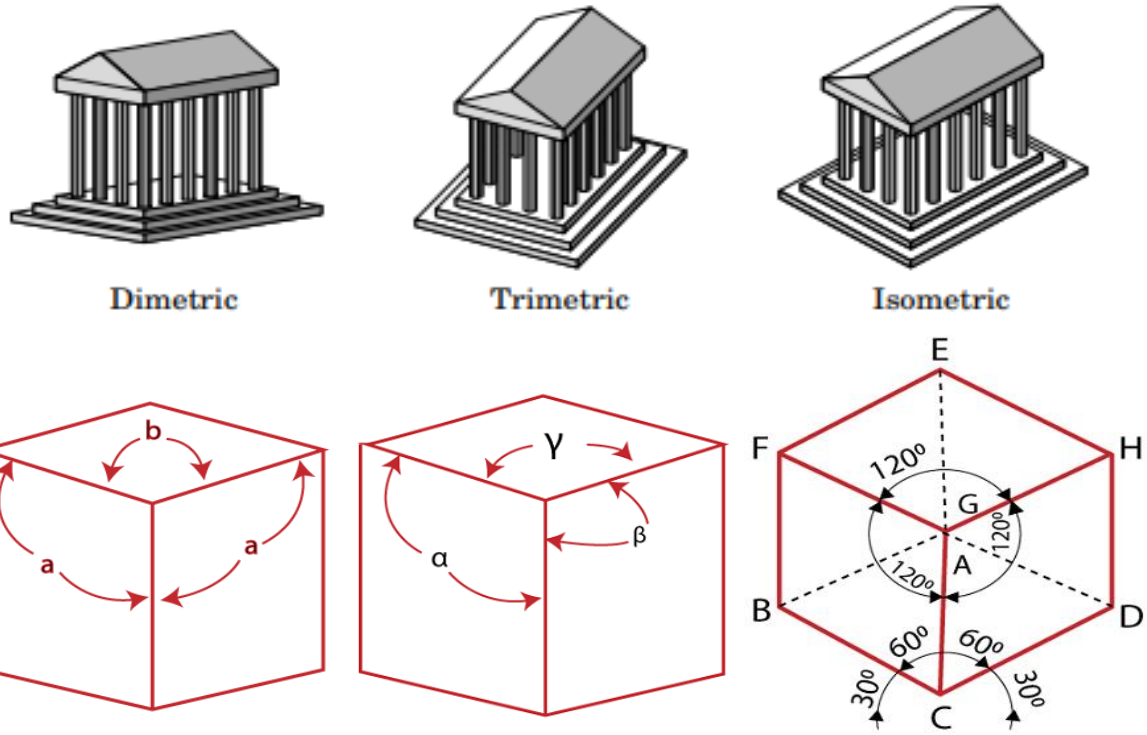


- Advantages:
  * preserves both distances and angles.
  * shapes are preserved.
  * can be used for measurements (e.g. building plans and manuals).
- Disadvantage:
  * many surfaces hidden from view, so cannot see what object really looks like (the isometric projection is added, isometric doesn't belong to this type, it is axonometric).





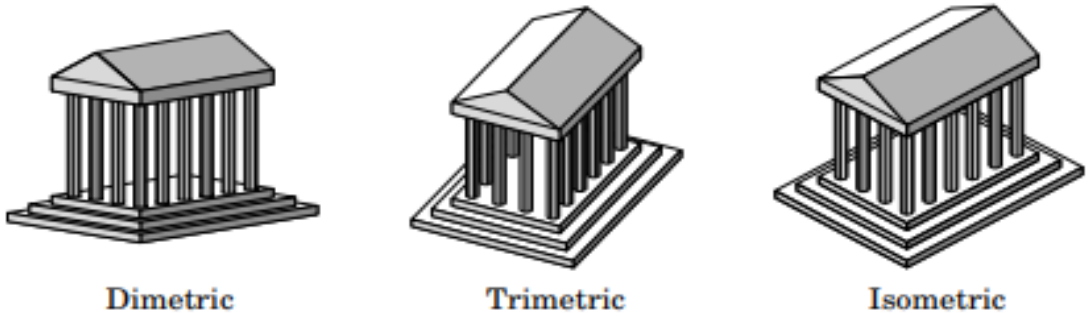
Axonometric Projection



- the projection plane can have an arbitrary orientation.
- when an orthogonal projection shows more than one face (i.e., when image plane is not parallel with one of the faces), we have an axonometric projection.
- it is classified by how many angles of a corner of a projected cube are the same.

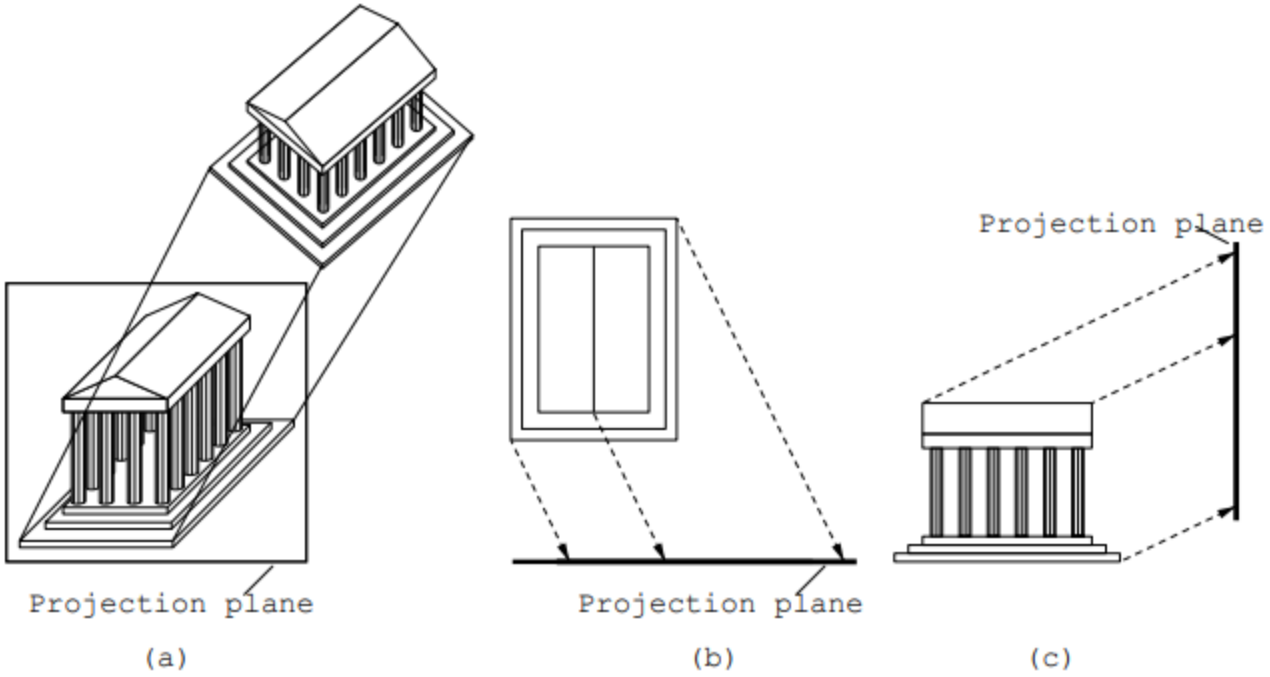
none: trimetric; two: dimetric; three: isometric.

Axonometric Projection



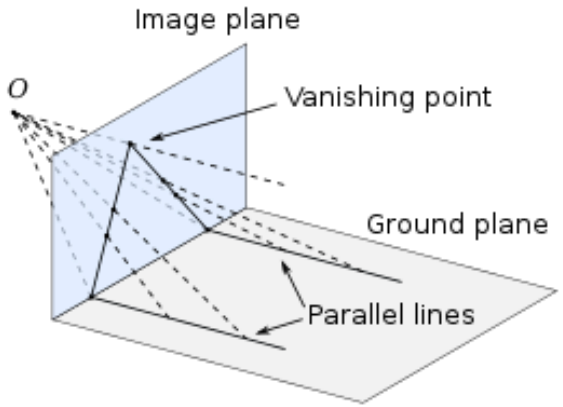
- a drafting style, a 2D object is drawn to look 3D.
- the horizontal edges of the object are drawn at 30^o or 150^o, all vertical lines at 90^o (remain vertical).
- advantages and disadvantages:
  * lines are scaled (foreshortened) but the scaling factors can be found.
  * lines preserved but angles are not.
  * we can see three principal faces of a box-like object.
  * some optical illusions possible (parallel lines appear to diverge).
  * doesn't look real for large objects (far objects are scaled the same as near objects).
  * used in CAD applications.

Oblique Projection



- (a) the construction, (b) top view, and (c) side view.
- the projection plane is not orthogonal to the projectors: projection lines are still parallel with each other but they are not perpendicular to the image plane.
- the projection plane is “skew”, is mostly used in technical drawing.
- can pick the angles to emphasize a particular face.
- angles in faces parallel to projection plane are preserved while we can still see “around ” side.
- cannot be created with simple camera; possible with bellows camera or special lens (architectural).

Perspective Projections



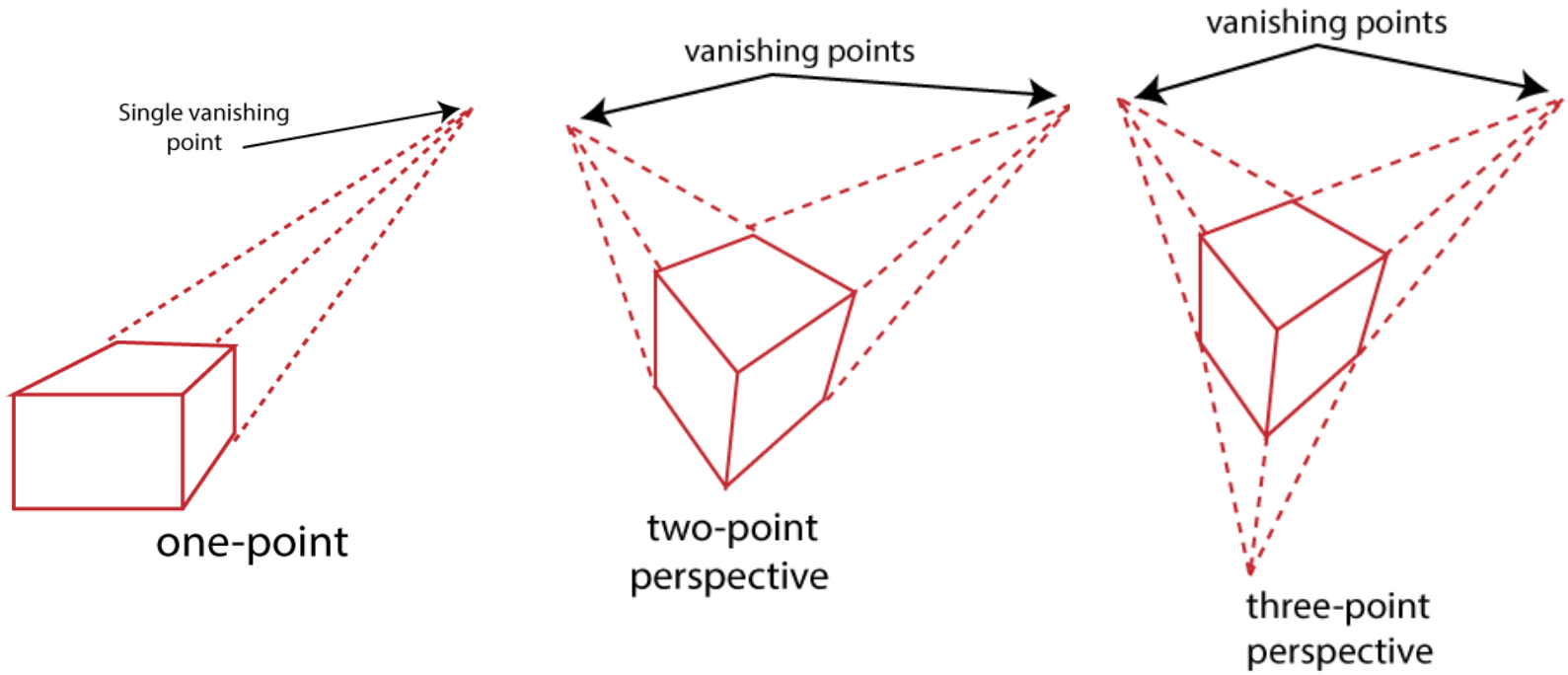
- the image of an object gets smaller when it is moved away from the viewer. Therefore it looks more natural.

- projection lines converge at a point: center of projection.

- parallel lines (on the object) converge at a single point (in the projection): vanishing point.

- drawing simple perspectives by hand, uses these vanishing point(s).

Perspective Projections



- One Point perspective: contains only one vanishing point on the horizon line. is mostly used to draw the images of roads, railway tracks, and buildings.
- Two Point perspective: “Angular Perspective”, contains two vanishing points on the line. The main use is to draw the two corner roads.
- Three-Point perspective: contains three vanishing points. Two points lie on the horizon line, and one, above or below the line. When we see an object from above, than the third point is below the ground. If we see an object from the below, than the third point is in the space above. It is mainly used in skyscraping.

Perspective Projections

(a) (b) (c)

Classical perspective views. (a) Three-point. (b) Two-point. (c) One-point.

- objects, further from viewer are projected smaller than the same sized objects closer to the viewer: diminution.

- some characteristics:

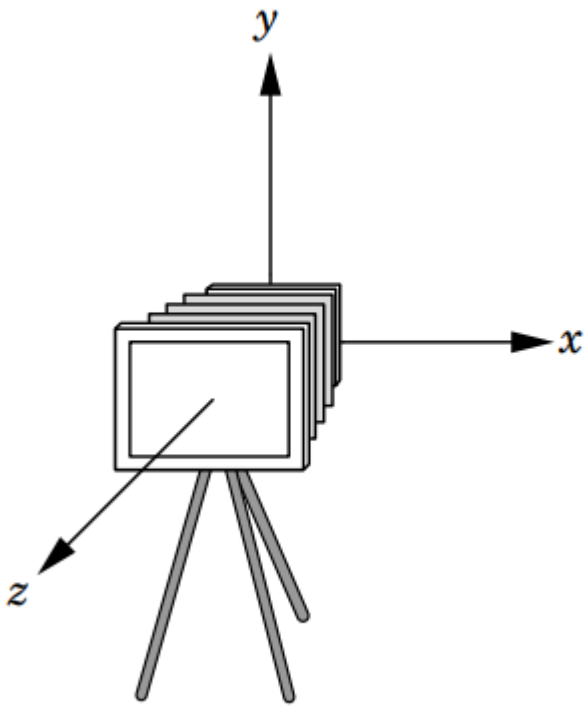
 * looks realistic.

 * equal distances along a line are not projected into equal distances: nonuniform foreshortening.

 * angles preserved only in planes parallel to the projection plane.

 * it is more difficult to construct by hand than parallel projections, but not more difficult by computers.

Arranging the Camera



Viewing in practice:

- how do we position (arrange) the camera with respect to the objects? There are two points of view:

(1) we move the camera with respect to the objects.

(2) we move objects relative to a fixed camera.

- these two reciprocal movements will result in the same image.

- these movements are controlled by the model-view matrix.

- model-view matrix represents a transformation between the world and the camera coordinate system.

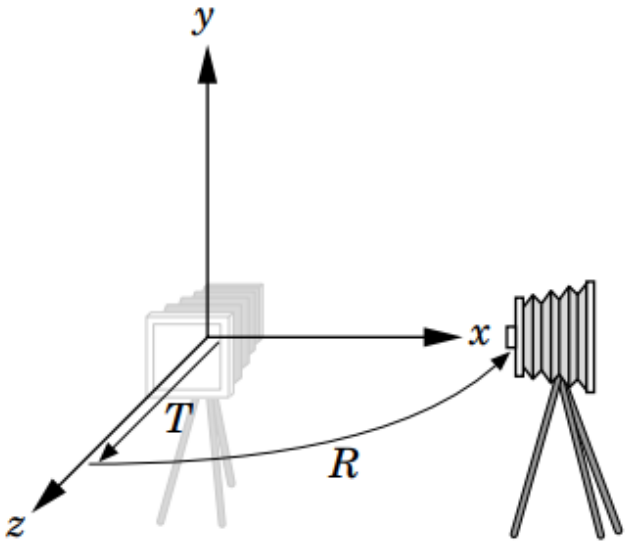
- initially and by default, the model-view matrix is set to the identity.

- e.g.: objects are typically placed near the origin. So to view them, we need to move it towards the origin, we need to move the camera back along the z axis.

The correct transformation is;

translation of the camera through the vector (0, 0, d) or translation of objects through the vector (0, 0, −d)

Arranging the Camera



- now view the objects from the positive x axis. (Objects are again centered around the origin)

- first move the camera backwards, along the positive z axis (distance d) and then rotate the camera around the y axis by 90^o or

- first rotate the objects the y axis by -90^o , and then translate through (0,0,−d)

M (model-view matrix) = R.T

M is applied to each object vertex p

Arranging the Camera

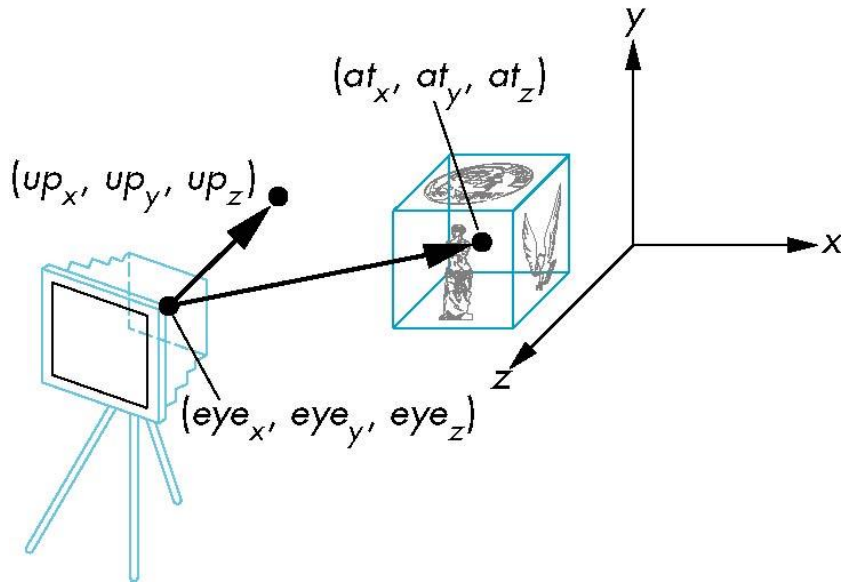
```
m = lookAt(eye, at, up)
```

$$n = at - eye$$

Normalize n

$$u = \frac{up \times n}{|up \times n|}$$

$$v = \frac{n \times u}{|n \times u|}$$



- to set up the camera, use functions like ‘lookAt’ (provided in most graphics libraries).
- lookAt requires two points and one vector:

1) the location of the camera (eye)

2) the location of a point along it’s line of sight (at)

3) what direction is upwards (up vector)

- up vector describes the roll of the camera (which point is "up" in the camera's orientation).
- eye will be used for translation.
- n is a vector in the direction that camera is looking at (z direction for the camera).
- normalize n and find u and v.
- u, v and n vectors are the x, y and z directions for the camera (can be used to create the rotation matrix).

Orthogonal Projection in WebGL

- default projection is an orthogonal projection (defined with the cube centered at origin that has a side length of 2).
- we can set a different view volume with the 'ortho' function.
- to form the required transformation matrix:

(1) translate the cube (the center of our volume is at the center of the cube (origin)),

(2) scale the cube (get the cube with side length 2)

var N = ortho(left, right,
bottom, top,
near, far);

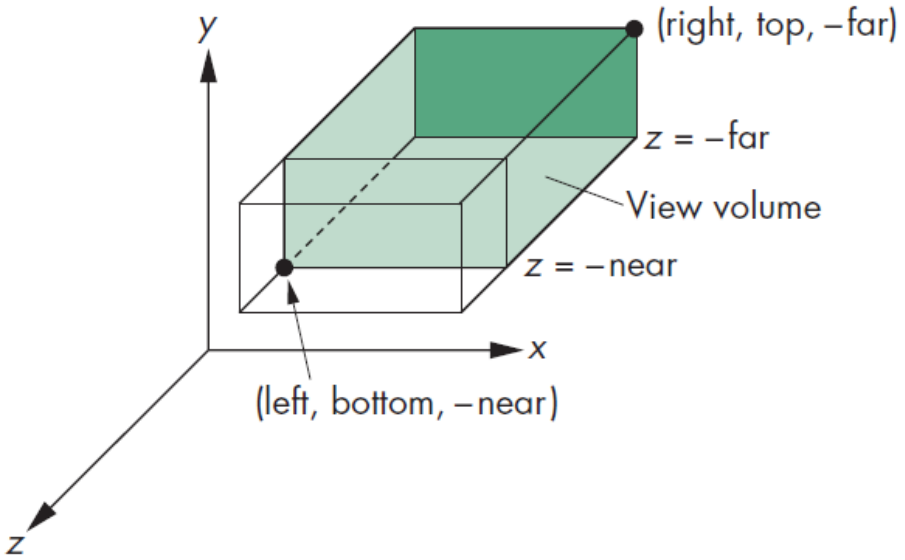


FIGURE 5.22 Orthographic viewing.

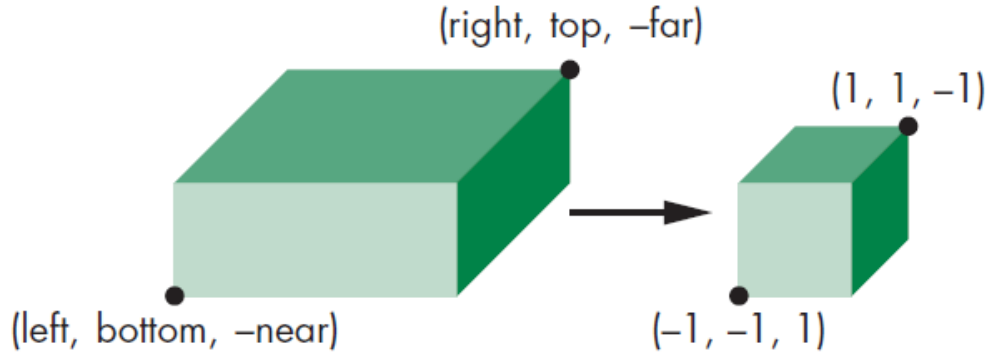


FIGURE 5.25 Mapping a view volume to the canonical view volume.

Orthogonal Projection in WebGL

- Translation

$$T = T\left(-\frac{\text{right} + \text{left}}{2}, -\frac{\text{top} + \text{bottom}}{2}, \frac{\text{far} + \text{near}}{2}\right)$$

- Scale

$$S = S\left(\frac{2}{\text{right} - \text{left}}, \frac{2}{\text{top} - \text{bottom}}, \frac{2}{\text{near} - \text{far}}\right)$$

- Together

CTM

$$ST = \begin{bmatrix} \frac{2}{\text{right} - \text{left}} & 0 & 0 & -\frac{\text{left} + \text{right}}{\text{right} - \text{left}} \\ 0 & \frac{2}{\text{top} - \text{bottom}} & 0 & -\frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\ 0 & 0 & -\frac{2}{\text{far} - \text{near}} & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

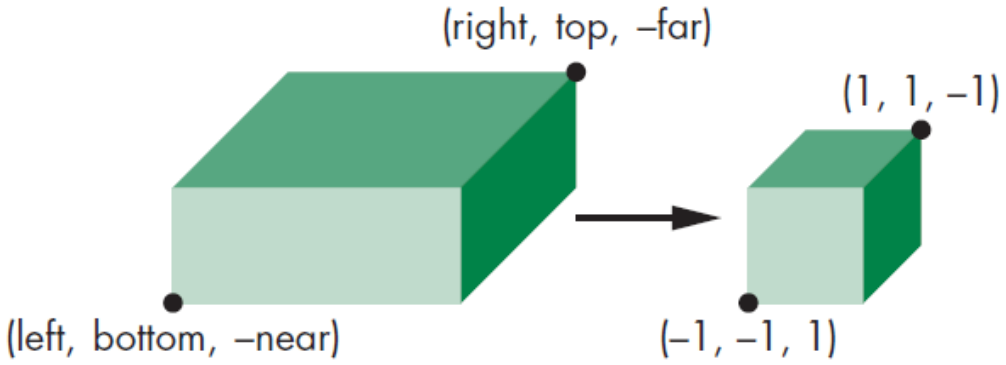


FIGURE 5.25 Mapping a view volume to the canonical view volume.

Perspective Projection in WebGL

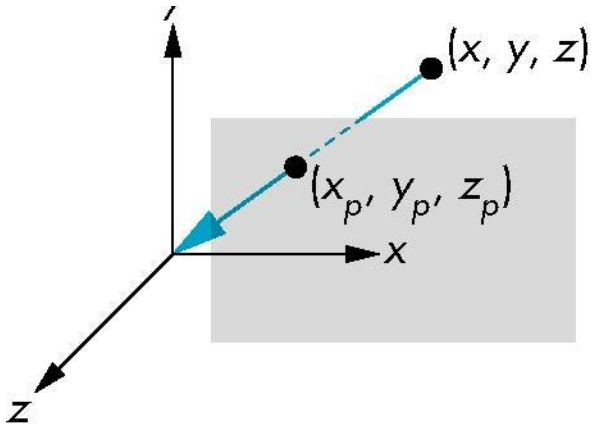
Suppose the camera is at its default position at the origin, pointing along the negative z axis. How do we compute a perspective projection?

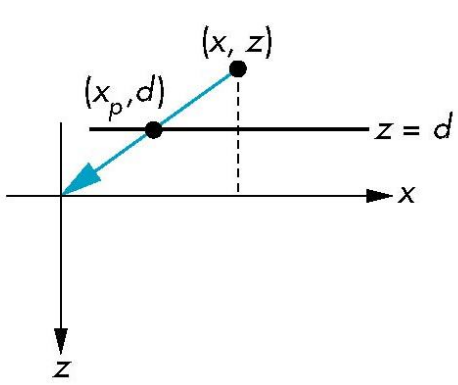
- camera placement can be done with the 'lookAt' function (same as before).

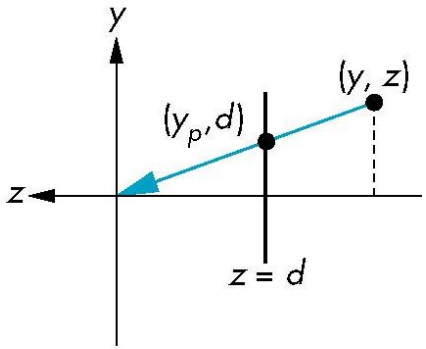
- camera is at the origin, projection plane is at z=d (d<0).

- projection coordinates of point (x,y,z) will be (xp,yp,zp) (*calculation using similar triangles).*

$$x_p = dx/z, \qquad y_p = dy/z, \qquad z_p = d$$







Perspective Projection in WebGL

- Consider

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

- This will transform a point (x,y,z,1) to (x,y,z,z/d)
- When we divide all coordinates with the value of w, we get

  (x/z/d, y/z/d, d, 1)

which is the projection point

- 4 x 4 transformation matrices (note that the last row will be used).

- M is the transformation matrix

Perspective Projection in WebGL

- our camera can see a wider or more narrow area (not everything will be projected).
- define the field of view for perspective projection: use frustum or perspective functions.

frustum (left,right,bottom,top,near,far)

or

perspective (fovy,aspect,near,far)

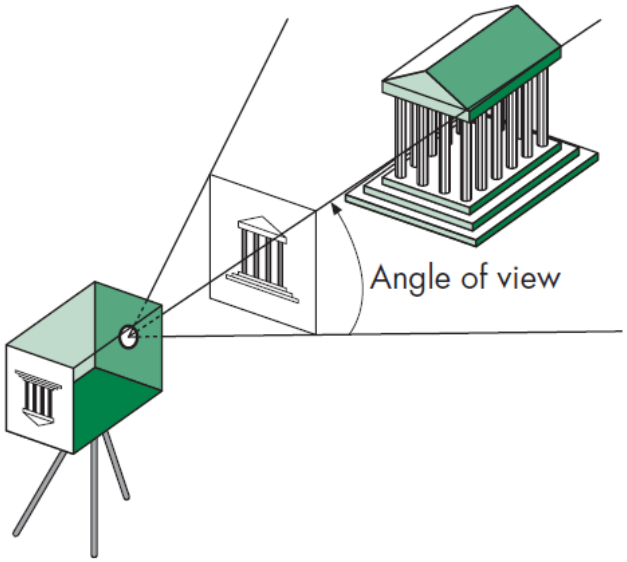
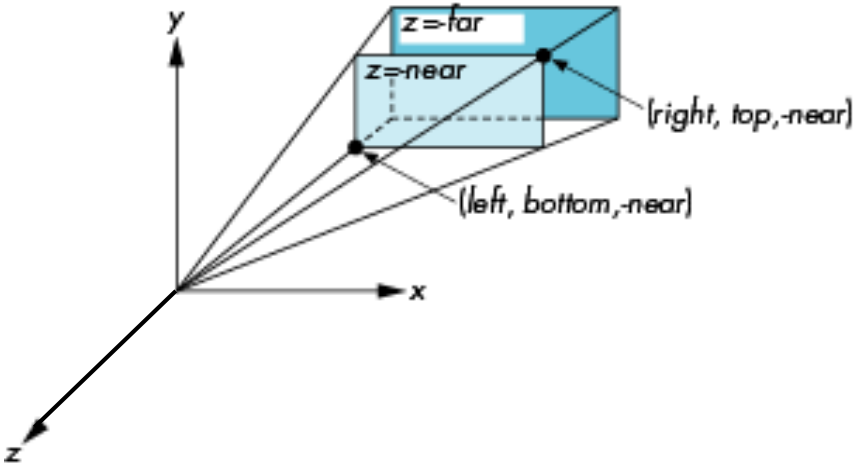


FIGURE 5.34 Specification of a view volume.



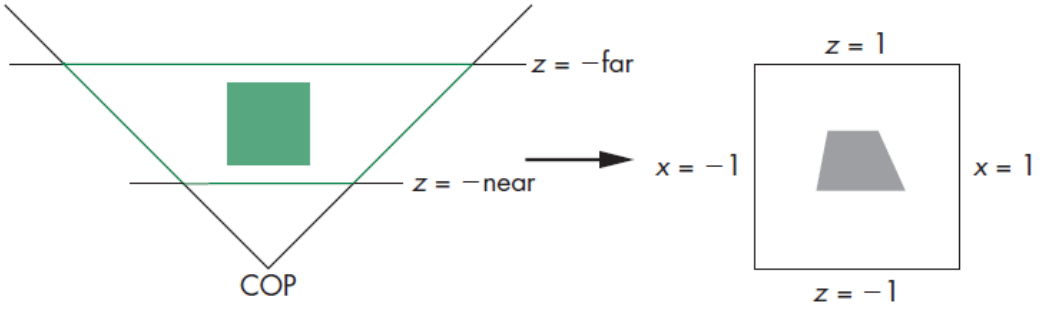


FIGURE 5.39 Perspective normalization of view volume.