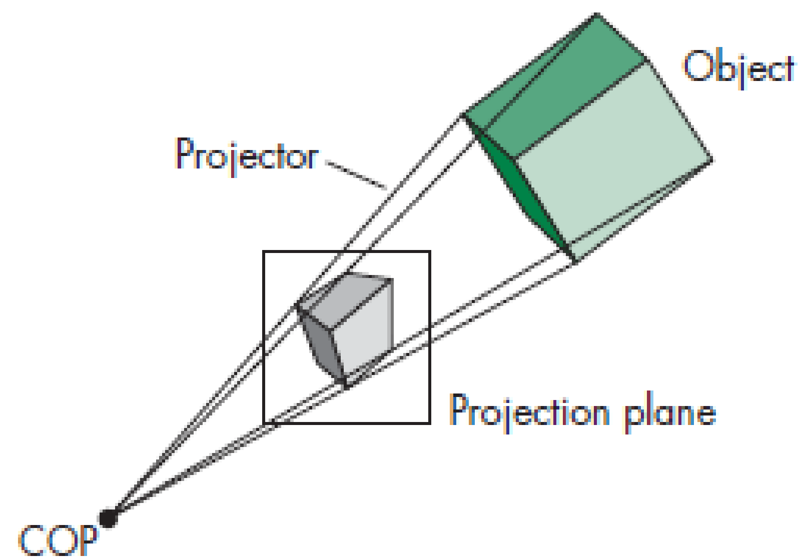
COSC 414/519I: Computer Graphics

2023W2

Shan Du

Classical and Computer Viewing

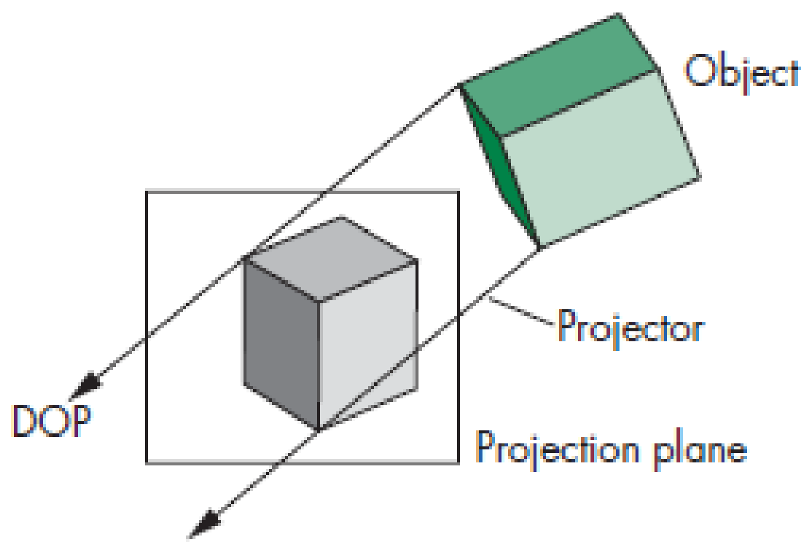
- In the synthetic-camera model, we have objects, a viewer, projectors and a projection plane.
- The projectors meet at the center of projection (**COP**).
- It is the origin of the **camera frame** for perspective views.



Viewing.

Classical and Computer Viewing

- If we move the COP to infinity, it means the viewer is infinitely far from the objects, the projectors become parallel and the COP can be replaced by a direction of projection (**DOP**).



Movement of the center of projection (COP) to infinity.

Classical and Computer Viewing

- Views with a finite COP are called perspective views; views with a COP at infinity are called parallel views. For parallel views, the origin of the camera frame usually lies in the projection plane.
- Most modern APIs support both parallel and perspective viewing. The class of projections produced by these systems is known **planar geometric projections** because the projection surface is a plane and the projectors are lines.

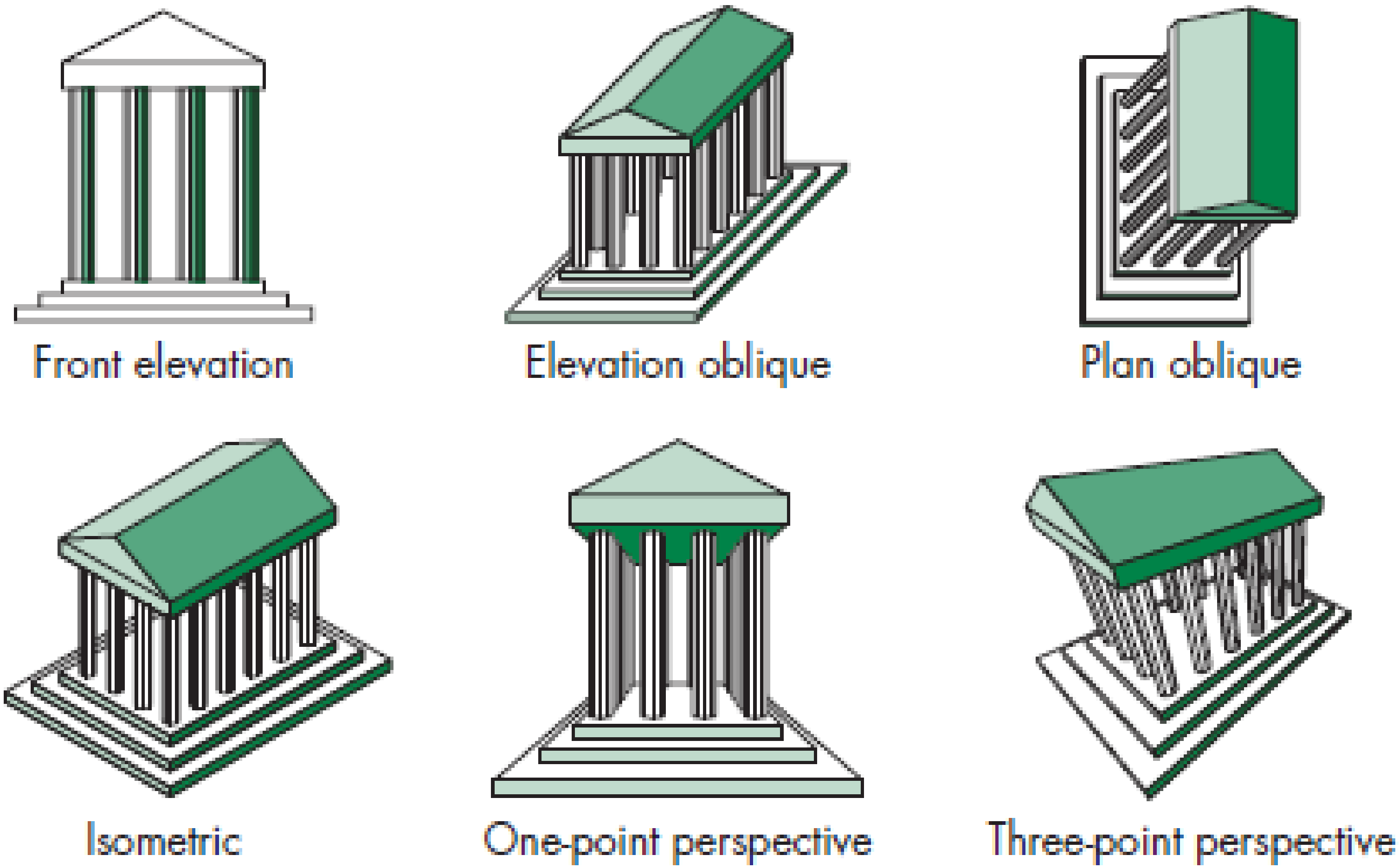
Classical and Computer Viewing

- Both perspective and parallel projections preserve lines; but they do not, in general, preserve angles.
- Although computer graphics systems have two fundamental types of viewing, classical graphics appears to permit a host of different views ranging from multiview orthographic projections to one-, two-, and three-point perspectives.

Classical and Computer Viewing

- Classical Viewing
  - Each classical view is determined by a specific relationship between the objects and the viewer.
  - In classical viewing , there is an underlying notion of a **principal face**.
  - For a rectangular object, there are natural notions of the front, back, top, bottom, right, and left faces.
  - In addition, many real-world objects have faces that meet at right angles; thus, such objects often have three orthogonal directions associated with them.

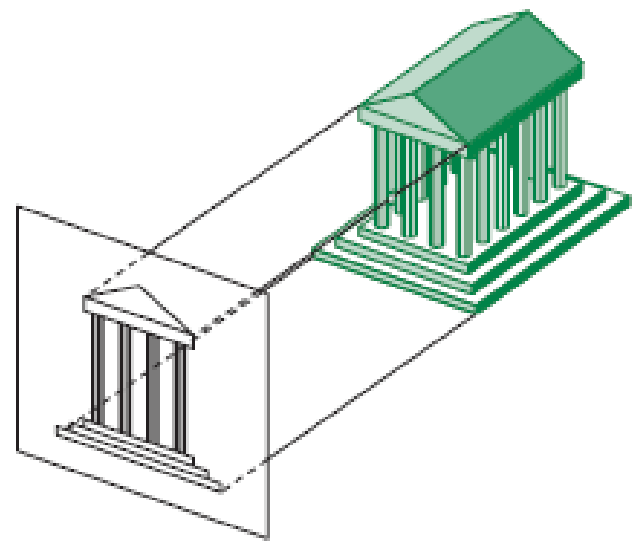
Classical and Computer Viewing



Classical views.

Classical and Computer Viewing

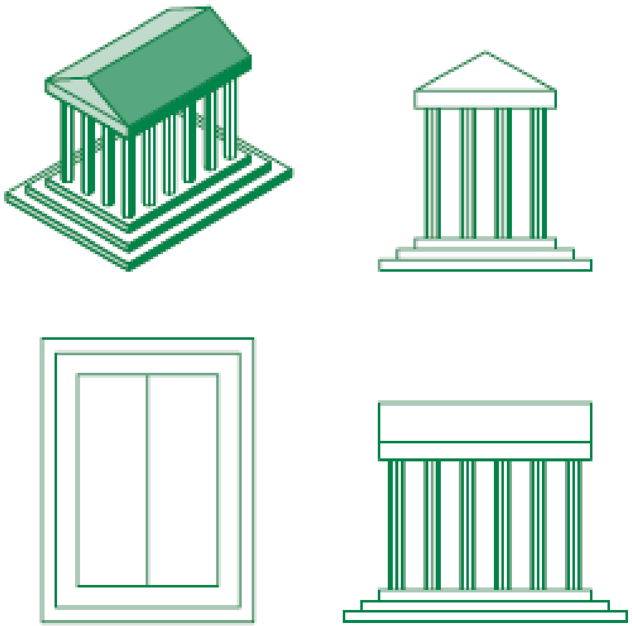
- Orthographic Projections
  - In all orthographic (or orthogonal) views, the projectors are perpendicular to the projection plane.
  - In multiview orthographic projection, we make multiple projections, in each case with the projection plane parallel to one of the principal faces of the object.



Orthographic projection.

Classical and Computer Viewing

- Orthographic Projections
  - Usually, we use three views – such as the front, top, and right – to display object.
  - A viewer usually needs more than two views to visualize what an object looks like.



Temple and three multiview orthographic projections.

Classical and Computer Viewing

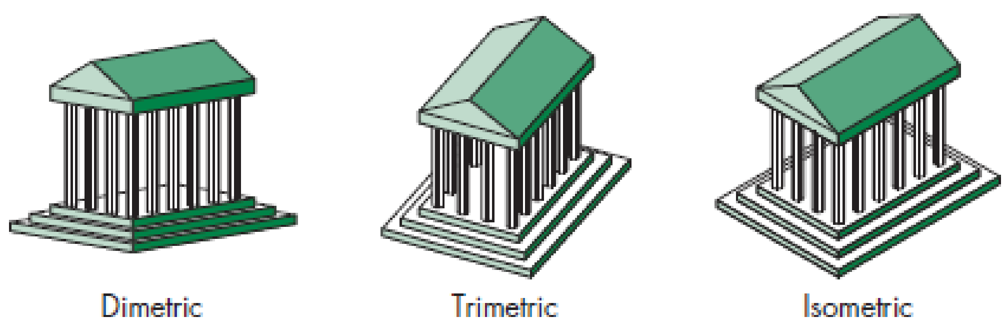
- Orthographic Projections
  - This type of view is that it preserves both distances and angles in faces parallel to the view plane.
  - Because there is no distortion of either distance or shape in images of these faces, multiview orthographic projections are well suited for working drawings.

Classical and Computer Viewing

- Axonometric Projections
  - If we want to see more principal faces of a box-like object in a single view, we must remove one of the restrictions.
  - In axonometric projection, the projectors are still orthogonal to the projection plane, but the projection plane can have any orientation with respect to the object.

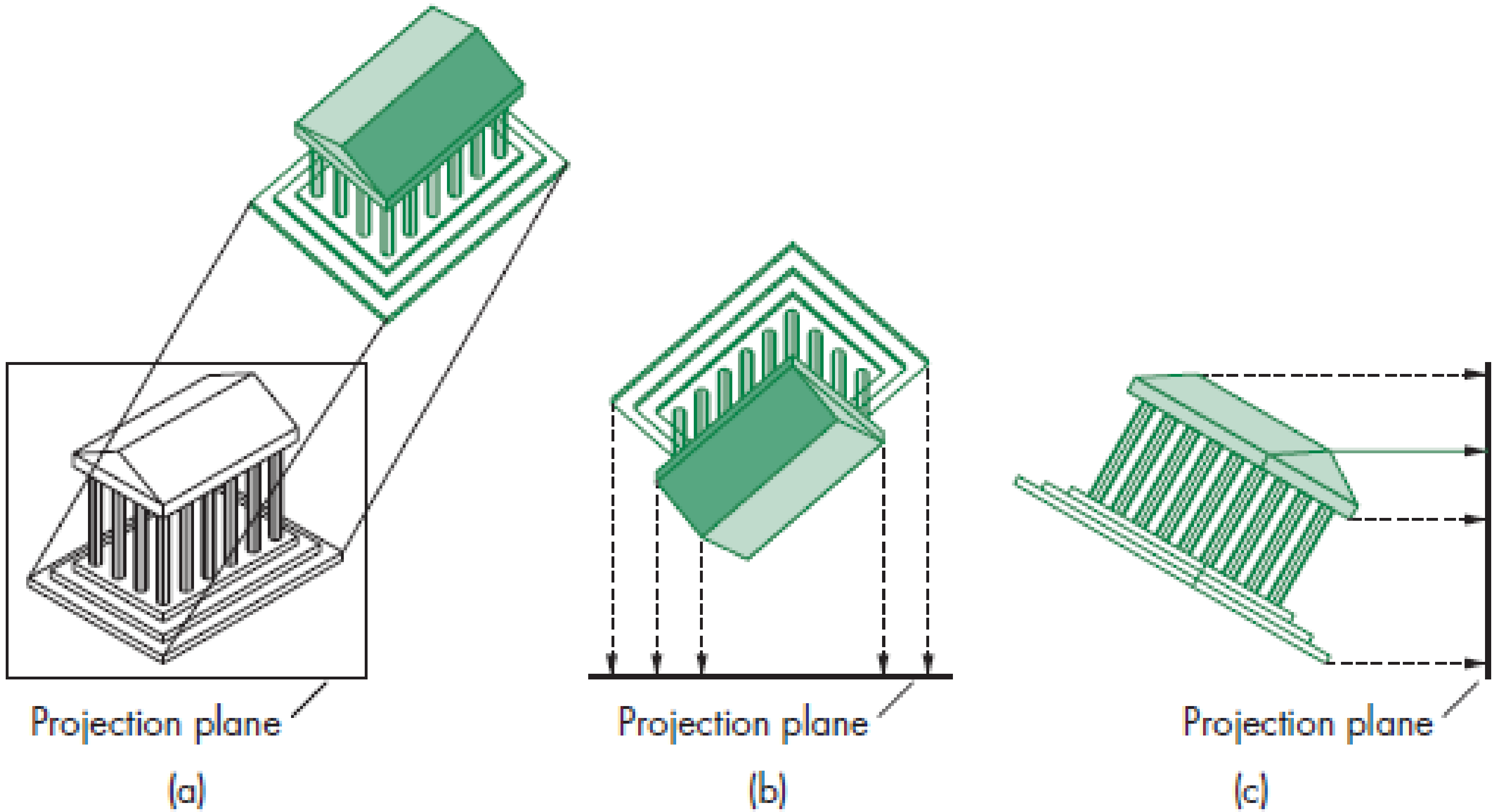
Classical and Computer Viewing

- Axonometric Projections
  - If the projection plane is placed symmetrically with respect to the three principal faces that meet at a corner of our rectangular object, then we have an isometric view.
  - If the projection plane is placed symmetrically with respect to two of the principal faces, then the view is dimetric.
  - The general case is a trimetric view.



Axonometric views.

Classical and Computer Viewing



Axonometric projections. (a) Construction of trimetric-view projections. (b) Top view. (c) Side view.

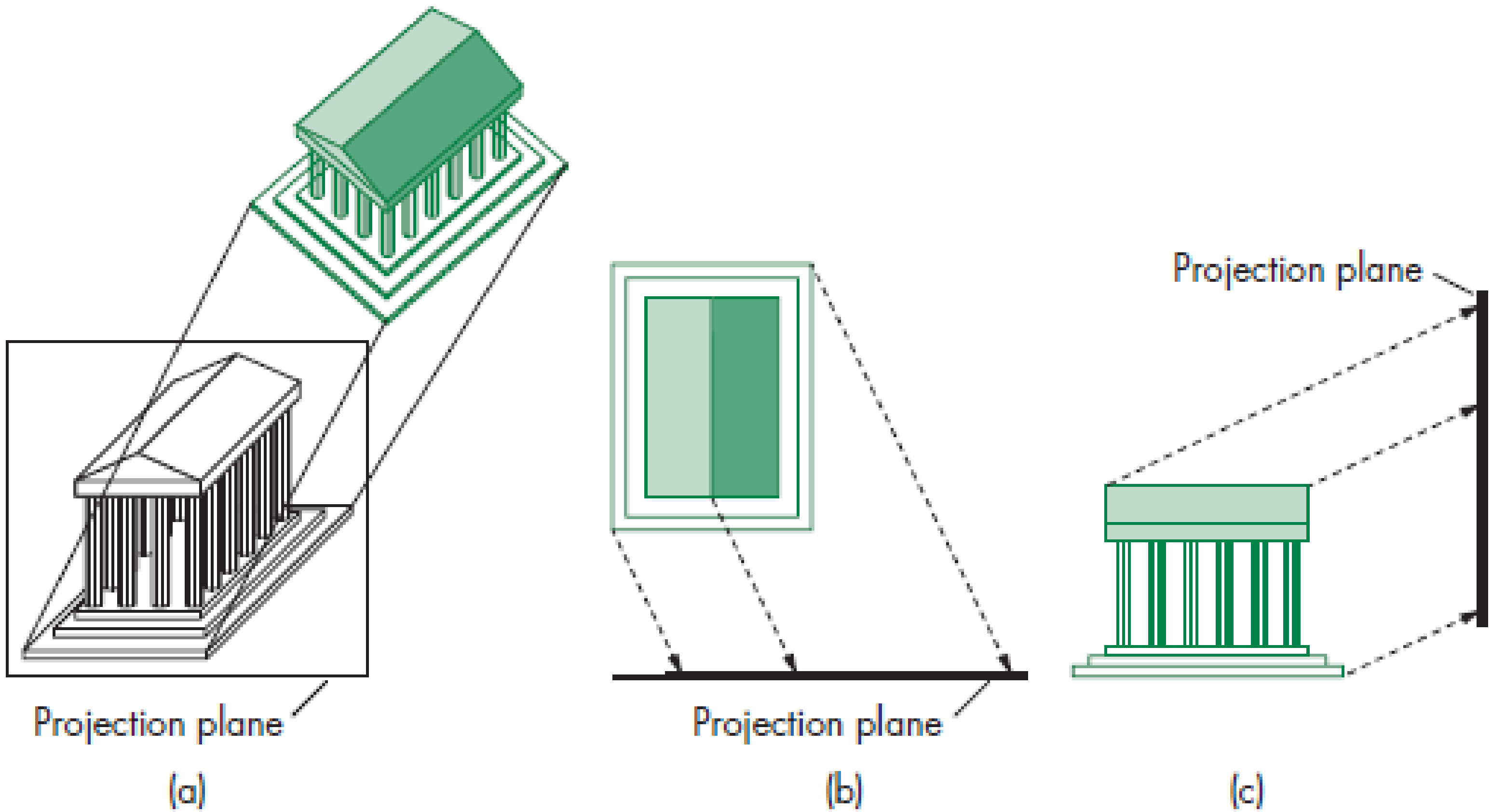
Classical and Computer Viewing

- Axonometric Projections
  - In an isometric view, a line segment's length in the image space is shorter than its length measured in the object space.
  - This foreshortening of the distance is the same in the three principal directions.
  - Parallel lines are preserved in the image, but angles are not. A circle is projected into an ellipse.
  - Axonometric views are used extensively in architectural and mechanical design.

Classical and Computer Viewing

- Oblique Projections
  - It is the most general parallel views. We allow the projectors to make an arbitrary angle with the projection plane.

Classical and Computer Viewing



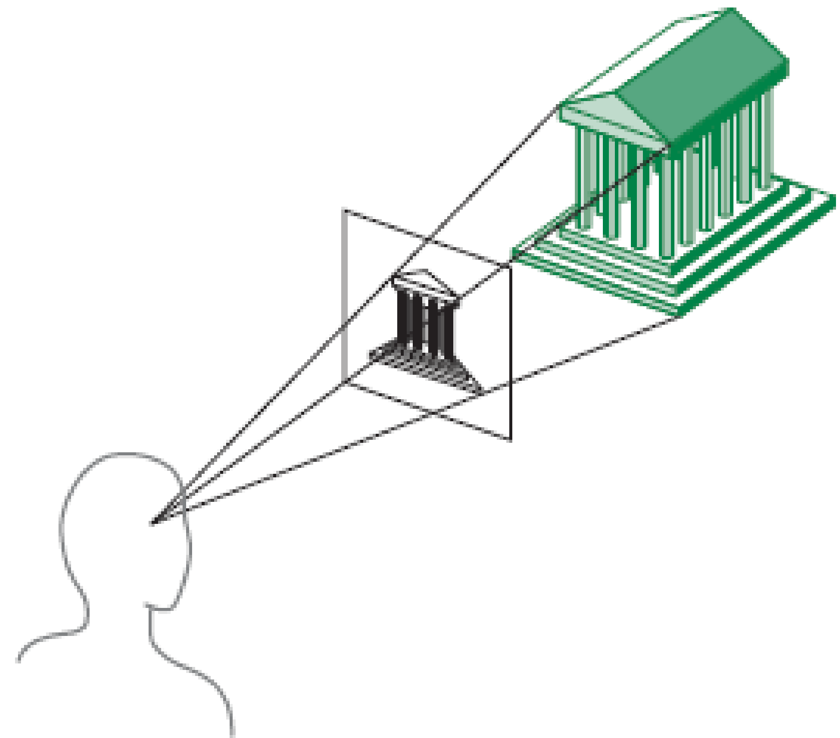
Oblique view. (a) Construction. (b) Top view. (c) Side view.

Classical and Computer Viewing

- Perspective Viewing
  - All perspective views are characterized by diminution of size.
  - When objects are moved farther from the viewer, their images become smaller.
  - The major use of perspective views is in applications such architecture and animation, where it is important to achieve natural-looking images.

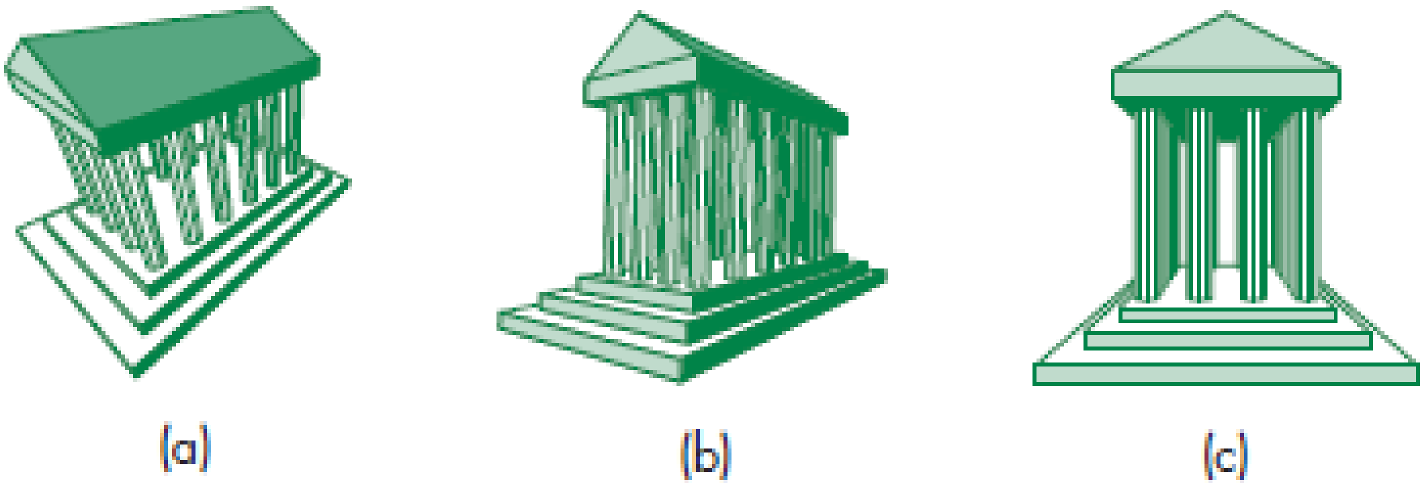
Classical and Computer Viewing

In the classical perspective views, the viewer is located symmetrically with respect to the projection plane and is on the perpendicular from the center of projection.



Perspective viewing.

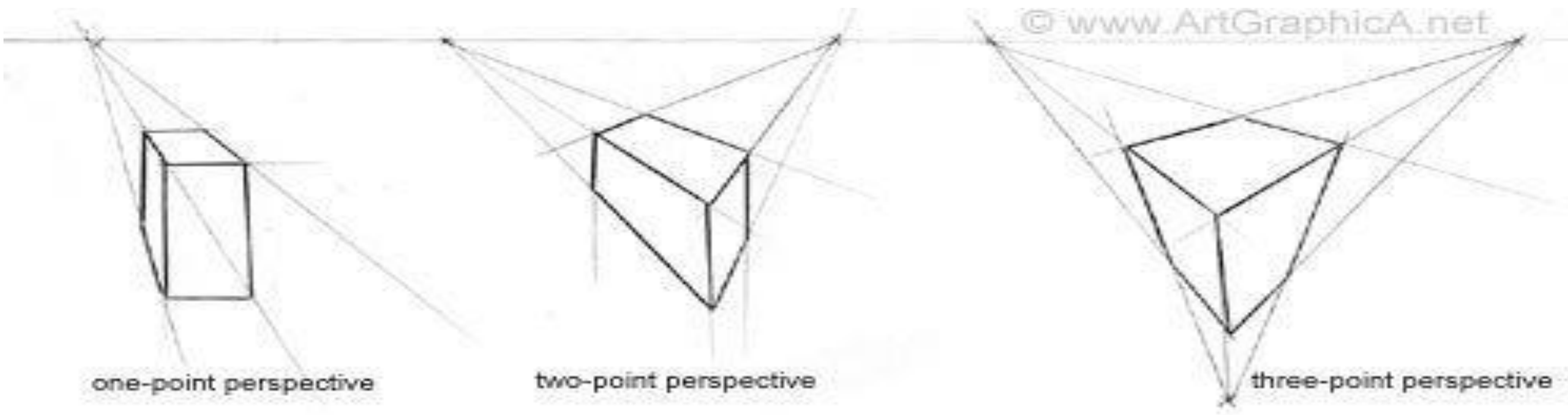
Classical and Computer Viewing



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

Classical and Computer Viewing

- In perspective view, parallel lines converge to a finite vanishing point.



Viewing with a Computer

- All the classical views are based on a particular relationship among the objects, the viewer, and the projectors.
- In computer graphics, we stress on the independence of the object specifications and camera parameters.
- Hence, to create one of the classical views, the application program must use information about the objects to create and place the proper camera.

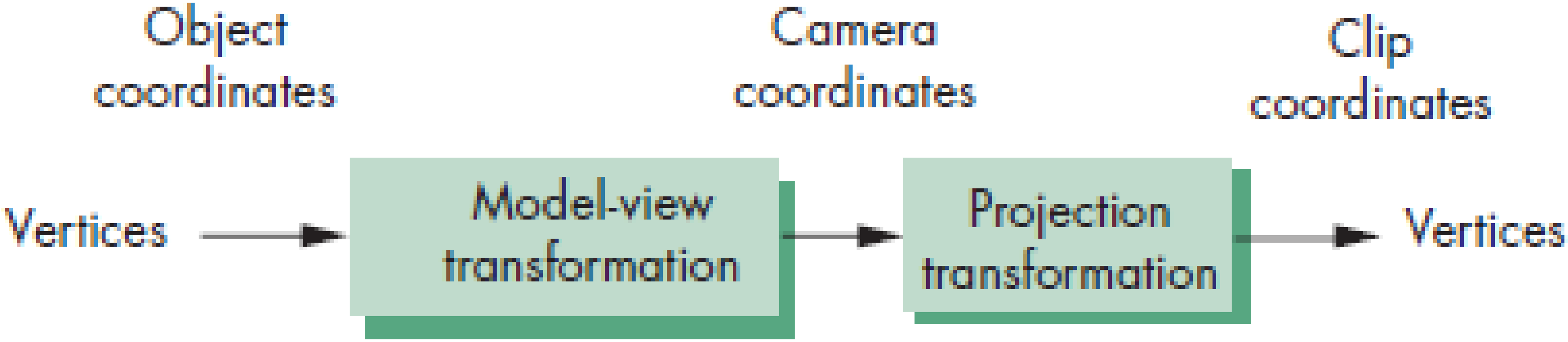
Viewing with a Computer

- Using WebGL, we will have many options on how and where we carry out viewing.
- All our approaches will use the powerful transformation capabilities of the GPU.
- Every transformation is equivalent to a change of frames, we can develop viewing in terms of the frames and coordinate systems.
- In particular, we will work with object coordinates, camera coordinates, and clip coordinates.

Viewing with a Computer

- First, we must position and orient the camera. This operation is the job of the model-view transformation.
- After vertices pass through this transformation, they will be represented in eye or camera coordinates. The second step is the application of the projection transformation. This step applies the specified projection – orthographic or perspective – to the vertices and puts objects within the specified clipping volume into the same clipping cube in clip coordinates.

Viewing with a Computer

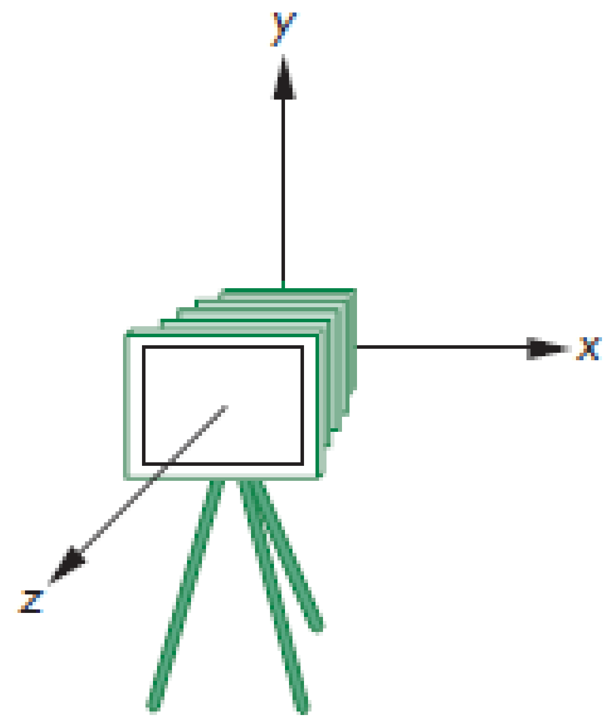


Viewing transformations.

The current transformation matrix (CTM) will be the product of two matrices: the model-view matrix and the projection matrix.

Positioning of the Camera

- Positioning of the Camera Frame
  - Initially, we start with the model-view matrix set to an identity matrix, so the camera frame and the object frame are identical. The camera is initially pointing in the negative *z* direction.

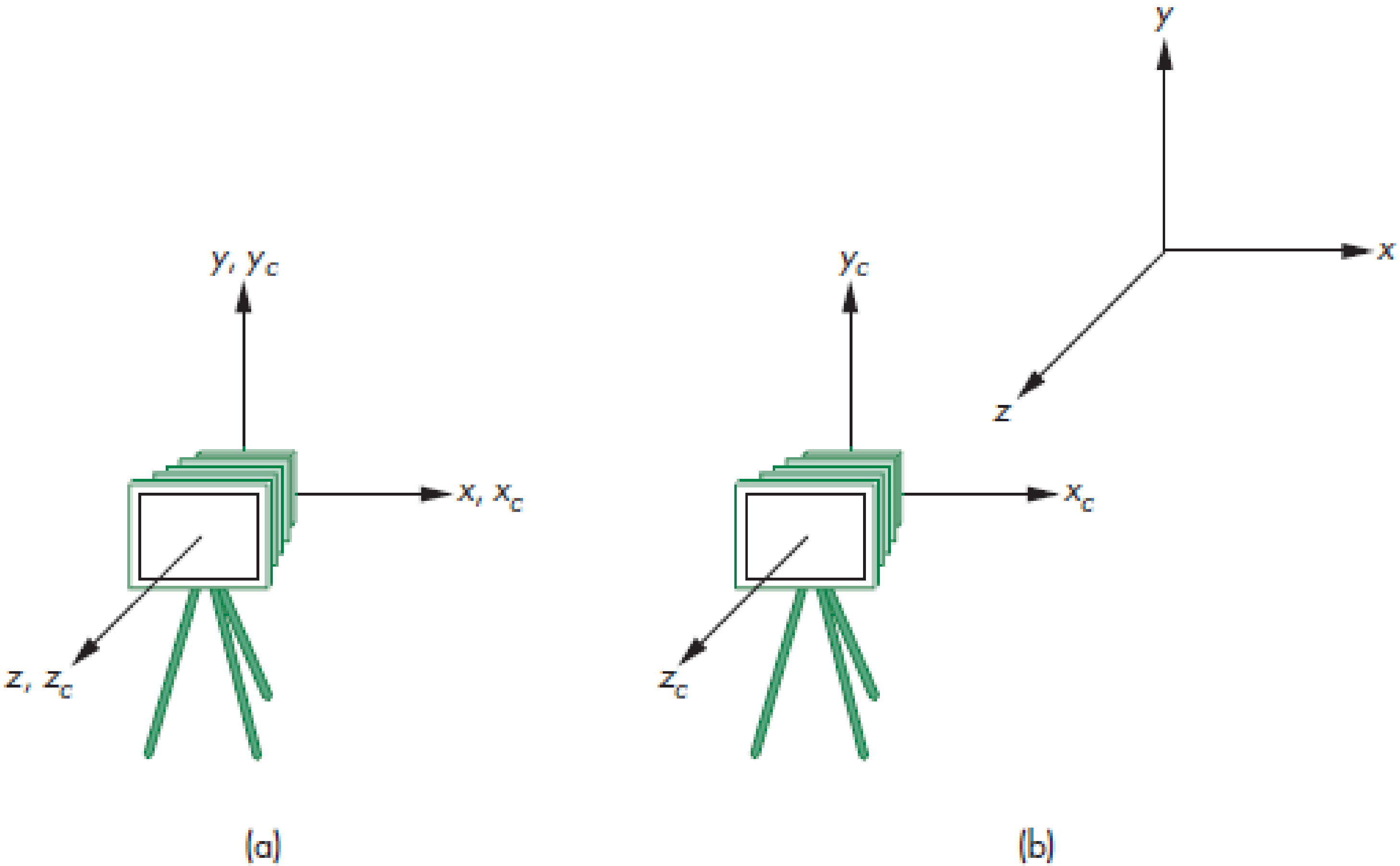


Initial camera position.

Positioning of the Camera

- Positioning of the Camera Frame
  – In most applications, we model our objects as being located around the origin, so a camera located at the default position with the default orientation does not see all the objects in the scene.
  – Thus, we must move the camera away from the objects that we wish to have in our image.

Positioning of the Camera



Movement of the camera and object frames. (a) Initial configuration. (b) Configuration after change in the model-view matrix.

Positioning of the Camera

- Positioning of the Camera Frame
  - We have initially specified several objects by specifying all vertices and putting their positions into an array.
  - We start with the model-view matrix set to an identity matrix.
  - Changes to the model–view matrix move the object frame relative to the camera and affect the camera’s view of all objects defined afterwards.

Positioning of the Camera

- Consider an object centered at the origin. The camera is in its initial position, also at the origin, pointing down the negative z-axis.
- Suppose that we want an image of the faces of an object that point in the positive z direction.
- We must move the camera away from the origin.
- If we allow the camera to remain pointing in the negative z direction, then we want to move the camera backward along the positive z-axis, and the proper transformation is

Positioning of the Camera

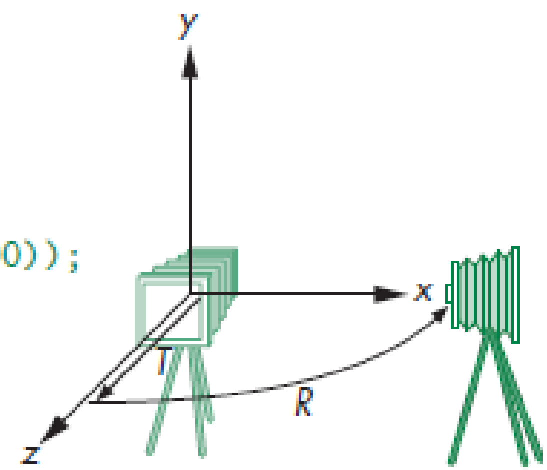
$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ where d is a positive number.

- WebGL views the camera as fixed, and thus the objects are repositioned and reoriented by the model-view matrix.

Positioning of the Camera

- Suppose that we want to look at the same object from the positive x-axis. Now, not only we have to move away from the object, but also we have to rotate the camera about the *y*-axis.

```
modelViewMatrix = mult(translate(0, 0, -d), rotateY(-90));
```



Positioning of the camera.

Positioning of the Camera

- Rotating a cube is equivalent to rotating the frame of the cube with respect to the frame of the camera; we could achieve the same by rotating the camera relative to the cube.