



Last updated: Yeh-Liang Hsu (2010-09-20).

Note: This is the course material for “ME550 Geometric modeling and computer graphics,” Yuan Ze University. Part of this material is adapted from *CAD/CAM Theory and Practice*, by Ibrahim Zeid, McGraw-Hill, 1991. This material is be used strictly for teaching and learning of this course.

Projection of geometric models

1. Homogeneous coordinate

Many graphics applications involve sequences of geometric transformations. An animation, for example, might require an object to be translated and rotated at each increment of the motion. Here we consider how the matrix representations can be reformulated so that such transformation sequences can be efficiently processed.

Translation, scaling, mirroring, and rotation of a point \mathbf{P} are represented by the following equations respectively:

$$\mathbf{P}^* = \mathbf{P} + \mathbf{d} \quad (1)$$

$$\mathbf{P}^* = [\mathbf{S}]\mathbf{P} \quad (2)$$

$$\mathbf{P}^* = [\mathbf{M}]\mathbf{P} \quad (3)$$

$$\mathbf{P}^* = [\mathbf{R}_z]\mathbf{P} \quad (4)$$

While the last three equations are in the form of matrix multiplication, translation takes the form of vector addition. This makes it inconvenient to concatenate transformations involving translation. It is desirable, therefore, to express all geometric transformations in the form of matrix multiplications only. **Representing points by their homogeneous coordinates provides an effective way to unify the description of geometric transformations as matrix multiplications.**

In three-dimensional space, **a point P with Cartesian coordinates (x, y, z) has the homogeneous coordinates (x*, y*, z*, h)**, where h is any scalar factor $\neq 0$. The two types of coordinates are related to each other by the following equations:

$$x = \frac{x^*}{h} \quad y = \frac{y^*}{h} \quad z = \frac{z^*}{h} \quad (5)$$

For example, (1, 2, 3, 1) and (2, 4, 6, 2) are the same point.

Translation can then be represented by matrix multiplication using homogeneous coordinates.

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 & x_d \\ 0 & 1 & 0 & y_d \\ 0 & 0 & 1 & z_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (6)$$

$$\mathbf{P}^* = [\mathbf{D}]\mathbf{P} \quad (7)$$

And the transformation matrices in Equation (2), (3), (4) become

$$[\mathbf{S}] = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$[\mathbf{M}] = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \\ 0 & \pm 1 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$[\mathbf{R}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Expressing positions in homogeneous coordinates allows us to represent all geometric transformation equations as matrix multiplications. For example,

$$\mathbf{P}^* = [\mathbf{D}_2][\mathbf{R}_z][\mathbf{D}_1]\mathbf{P} = [\mathbf{T}]\mathbf{P} \quad (11)$$

where

$$[\mathbf{D}_1] = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$[\mathbf{R}_z] = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$[\mathbf{D}_2] = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$[\mathbf{T}] = [\mathbf{D}_2][\mathbf{R}_z][\mathbf{D}_1] = \begin{bmatrix} \cos \theta & -\sin \theta & x_1(\cos \theta - 1) - y_1 \sin \theta & 0 \\ \sin \theta & \cos \theta & x_1 \sin \theta + y_1(\cos \theta - 1) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

or in a general form,

$$[\mathbf{T}] = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \\ \mathbf{T}_3 & 1 \end{bmatrix} \quad (16)$$

The 3×3 submatrix $[\mathbf{T}_1]$ produces scaling, reflection, or rotation. The 3×1 column matrix $[\mathbf{T}_2]$ generates translation. The 1×3 row matrix $[\mathbf{T}_3]$ produces perspective projection, which will be discussed later.

◇Assignment 1

Create a rectangle in x - y plane in your CAD software; specify the coordinates of the 4 corner points. Assume some parameters and apply the following two-dimensional transformations, translation, rotation, translation, to the object in your CAD software. Draw the object after these transformations.

Use homogenous coordinates to represent the key points of the object you generated. Write a program in Matlab and concatenate the 3 transformations into one transformation matrix as shown in Equation (11)~(15). Calculate the positions of the key points using this transformation matrix. Regenerate the object by connecting the key points after transformation. Does the object match the object you drew in your CAD software? ◇

◇Assignment 2

Use two-dimensional rotation matrices to show that rotation is not commutative, that is,

$$[\mathbf{R}_2][\mathbf{R}_1]\mathbf{P} \neq [\mathbf{R}_1][\mathbf{R}_2]\mathbf{P} \cdot \diamond$$

Homogeneous coordinates was introduced by August Ferdinand Möbius in 1827. They have the advantage that the coordinates of a point, even those at infinity, can be represented using finite coordinates. Often formulas involving homogeneous coordinates are simpler and more symmetric than their Cartesian counterparts. Homogeneous coordinates have a range of applications, including computer graphics and 3D computer vision, where they allow affine transformations and, in general, projective transformations to be easily represented by a matrix.

2. Projections of geometric models

2.1 Perspective projection and parallel projection

Viewing a three-dimensional model is a rather complex process due to the fact that display devices can only display graphics on two-dimensional screens. To define a projection, **a center of projection** and **a projection plane** must be defined.

There are two different types of projections based on the location of the center of projection (or projection reference point) relative to the projection plane, as shown in Figure 1. **If the center is at a finite distance from the plane, perspective projection**

results and all the projectors meet at the center. If, on the other hand, **the center is at an infinite distance, all the projectors become parallel (meet at infinity) and parallel projection results.**

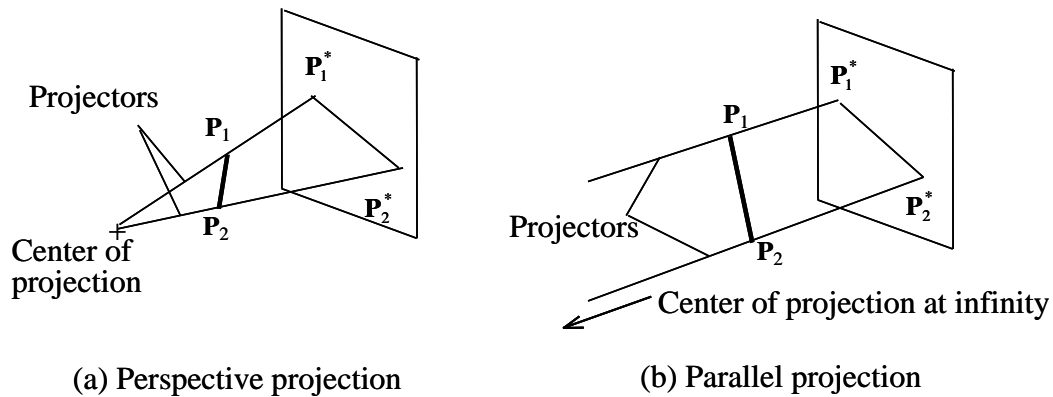


Figure 1. Types of projections

Perspective projection does not preserve parallelism, that is, no two lines are parallel. Perspective projection creates an artistic effect that adds some realism to perspective views. **Size of an entity is inversely proportional to its distance from the center of projection**; that is, the closer the entity to the center, the larger its size is.

Parallel projection preserves actual dimensions and shapes of objects. It also preserves parallelism. Angles are preserved only on faces of the object that are parallel to the projection plane.

◇ Assignment 3

Does your CAD software support perspective projection? If so, build a 3D object to demonstrate the difference between perspective projection and parallel projection. In your CAD software, how do you modify the parameters to change the viewpoint of the perspective projection? ◇

There are two types of parallel projections based on the relation between the direction of projection and the projection plane. If this direction is normal to the projection plane, **orthographic projection** and views result. If the direction is not normal to the plane, **oblique projection** occurs.

There are two types of orthographic projections. The most common type is the one that uses projection planes that are perpendicular to the principal axes of the MCS of the model; that is, the direction of projection coincides with one of these axes. **The multi-view**

projection -- front, top, and right views that are used customarily in engineering drawings belong to this type.

The other type of orthographic projection uses projection planes that are not normal to a principal axis and therefore show several faces of a model at once. This type is called **axonometric projections. They preserve parallelism of lines but not angles.** Axonometric projections are further divided into **trimetric, dimetric and isometric projections.** The isometric projection is the most common axonometric projection. **The isometric projection has the useful property that all three principal axes are equally foreshortened.** Therefore measurements along the axes can be made with the same scale — thus the name: “iso” for equal, metric for measure. In addition, the normal to the projection plane makes equal angles with each principal axis and the principal axes make equal angles (120° each) with one another when projected onto the projection plane.

The oblique projection places the principal face of the object parallel to the plane of the paper, and is often used in freehand sketching. The advantage is that details on the front face of the object retain their true shape. The disadvantage is that oblique projection does not appear realistic.

◇ Assignment 4

Draw a “family tree” for the types of projections discussed above. Use your CAD software to generate all projections in the family tree using the 3D object in Assignment 3. ◇

2.2 Mappings of geometric models to the viewing coordinate system

Mapping of a point (or a set of points) belonging to an object from one coordinate system to another is defined as changing the description of the point (or the set of points) from the first coordinate system to the second one. This is equivalent to transforming one coordinate system to another.

Given the coordinates of a point \mathbf{P} measured in a given xyz coordinate system, find the coordinates \mathbf{P}^* of the point measured in another coordinate system, say $x^*y^*z^*$ such that

$$\mathbf{P}^* = f(\mathbf{P}, \text{transformation parameters}) \quad (17)$$

The mapping parameters describe the relationship between the two systems and consist of the position of the origin and orientation of the $x^*y^*z^*$ system relative to the xyz system.

$$[\mathbf{T}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_d \\ r_{21} & r_{22} & r_{23} & y_d \\ r_{31} & r_{32} & r_{33} & z_d \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} [\mathbf{R}] & \mathbf{d} \\ \hline 0 & 1 \end{bmatrix} \quad (18)$$

where $[\mathbf{R}]$ and \mathbf{d} are the rotational and translational mapping parts of $[\mathbf{T}]$ respectively.

Therefore, Equation (18) gives the position vector of the origin of the xyz system as \mathbf{d} and its orientation as $[\mathbf{R}]$ both measured in the $x^*y^*z^*$ system. The columns of $[\mathbf{R}]$ are the components of the unit vectors of the xyz system (along its axes) measured in the $x^*y^*z^*$ system.

A view has a **viewing coordinate system** (VCS). It is a three-dimensional system with the X_v axis horizontal pointing to the right and the Y_v axis vertical pointing upward, as shown in Figure 2. The Z_v axis defines the viewing direction. The positive Z_v axis has an opposite sense to the viewing direction to keep the VCS a right-handed coordinated system, even though a left-handed system may be more desirable here since its positive Z_v axis is in the direction of the lines of sight emitting from the viewing eye.

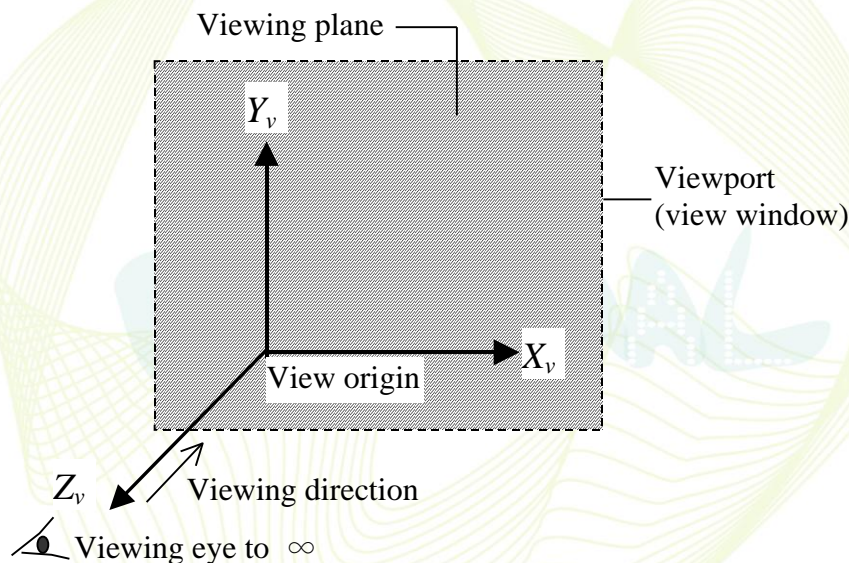


Figure 2. View definition.

To obtain views of a model, the viewing plane, the $X_v Y_v$ plane, is made coincident with the x - y plane of the MCS such that the VCS origin is the same as that of the MCS. Model views now become a matter of rotating the model with respect to the VCS axes until the desired model plane coincides with the viewing plane followed by projecting the model

onto that plane. Thus, **a view of a model is generated in two steps: rotate the model properly and then project it.**

2.3 Orthographic projections

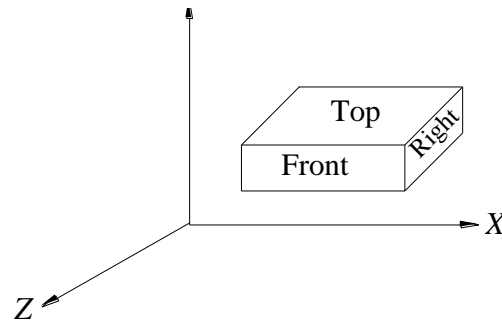
Figure 3 shows the relationship between MCS and VCS. An orthographic projection (view) of a model is obtained by setting to zero the coordinate value corresponding to the MCS axis that coincides with the direction of projection (or viewing) after the model rotation. **To obtain the front view, we only (no rotation is needed) need to set $z = 0$ for all the key points of the model.** Thus, Equation (18) becomes

$$[\mathbf{T}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

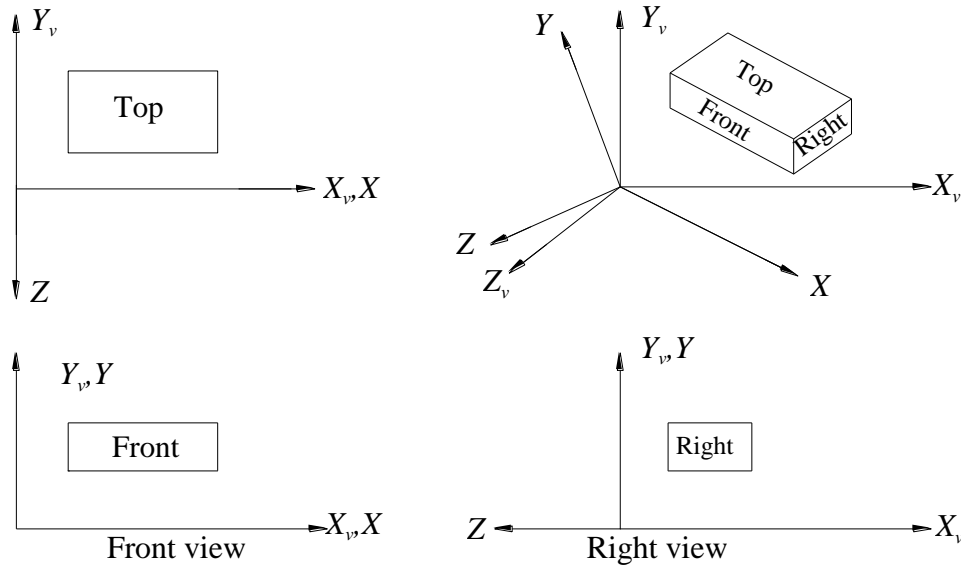
And

$$\mathbf{P}_v = [\mathbf{T}]\mathbf{P} \quad (20)$$

where \mathbf{P}_v is the point expressed in the VCS. For the front view, Equation (20) gives $x_v=x$ and $y_v=y$.



(a) Model views relative to its MCS



(b) MCS and VCS relationship

Figure 3. Relationship between MCS and VCS

For the top view, the model and its MCS are rotated by 90° about the X_v axis followed by setting the y coordinate of the resulting points to zero. The y coordinate is the one to set to zero because the Y axis of the MCS coincides with the projection direction. In this case, $[\mathbf{T}]$ becomes

$$[\mathbf{T}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

And Equation (20) gives $x_v=x$ and $y_v=-z$. If we use the above equation to transform the MCS itself, the X axis ($y=z=0$) transforms to $x_v=x$ and the Y axis ($x=z=0$) transforms to $y_v=-z$.

The right view shown in Figure 3 can be obtained by rotating the model and its MCS about the Y_v axis by -90° and setting the x coordinate to zero. Thus,

$$[\mathbf{T}] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

which gives $x_v = -z$ and $y_v = y$.

Examining Equation (20), (21), and (22) shows that **[T] is a singular matrix with a column of zeros which corresponds to the MCS axis that coincides with the projection or viewing direction.**

To obtain the isometric projection or view, the model and its MCS are customarily rotated an angle $\theta = \pm 45^\circ$ about the Y_v axis followed by a rotation $\phi = \pm 35.26^\circ$ about the X_v axis. In practice, the angle ϕ is taken as $\pm 30^\circ$ to enable the drafting (plastic) triangles in manual construction of isometric views.

$$\mathbf{P}_v = [\mathbf{T}_x][\mathbf{T}_y]\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (23)$$

◇ Assignment 5

Create a 3D block and assign coordinates to all 8 corner points of the block. Write a program in Matlab and use Equation (23) to transform the 8 key points of the block. Setting z coordinates of the key points to zero, then draw the block again. Do you obtain an isometric view of the block? Show your Matlab program too. ◇

2.4 Perspective projections

One common way to obtain a perspective view is to place the center of projection along the Z_v axis of the VCS and project onto the $z_v=0$ or the X_vY_v plane.

A new coordinate system called the eye coordinate system (ECS) is introduced relative to the line of sight. The ECS has an origin located at the same position as the viewing eye. Its X_e and Y_e axes are parallel to the X_v and Y_v axes of the VCS. However, it is a left-handed system.

The transformation matrix of coordinates of points from the VCS to the ECS or vice versa can be written as

$$[\mathbf{T}] = \begin{bmatrix} 1 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & -1 & | & 0 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (24)$$

This matrix simply inverts the sign of the z coordinate. In the orthographic views, the ECS is located at infinity. It is obvious that the ECS can be replaced by the VCS. In this case, points with smaller z values are interpreted as being further from the viewing eye.

One-point projection is shown in Figure 4. From similar triangles, we can the following equations.

$$\frac{y_v}{y} = \frac{d}{d-z} = \frac{1}{1-z/d} \quad (25)$$

$$\frac{x_v}{x} = \frac{r_2}{r_1} = \frac{d}{d-z} = \frac{1}{1-z/d} \quad (26)$$

$$\mathbf{P}_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (27)$$

If this equation is expanded it gives $\mathbf{P}_v = [x \quad y \quad 0 \quad (1-z/d)]^T$.

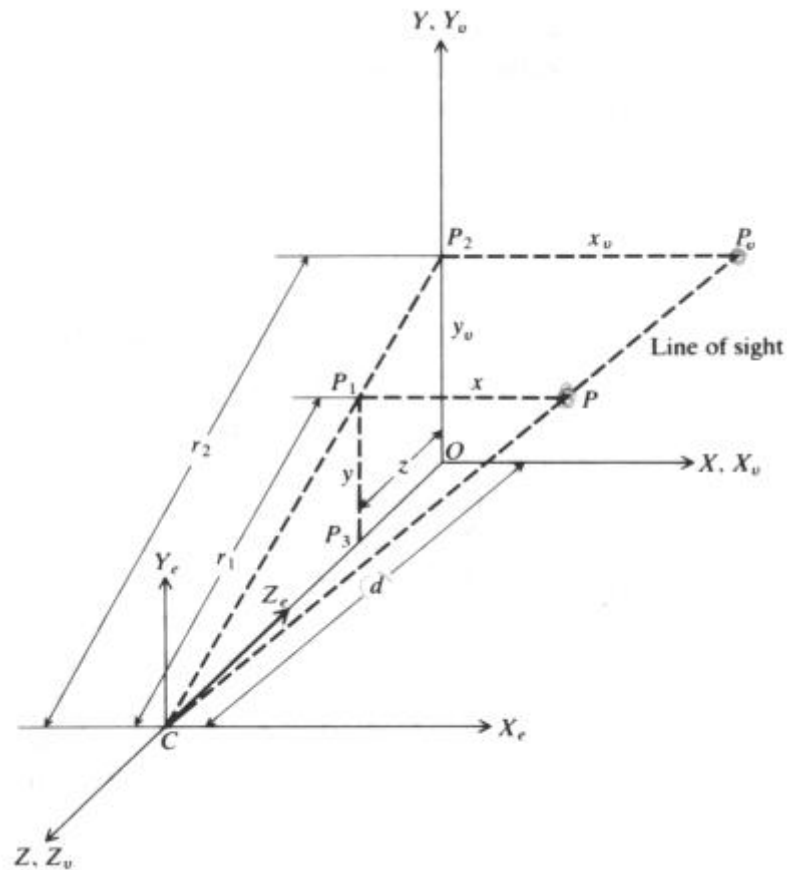


Figure 4. Perspective projection along the Z_v axis

◇ Assignment 6

Use the same 3D block in Assignment 5. Write a program in Matlab to apply Equation (27) to its 8 key points, then create the perspective projection of the block. Change several values of d , how does the perspective projection change? Show your Matlab program too. ◇