
Visualización y Transformaciones de Proyección para Computación Gráfica y Modelamiento

Prof. María Cecilia Rivara
mcrivara@dcc.uchile.cl
2011-2

Contenido

- Visualización 3D: conceptos y elementos importantes
- Proyecciones geométricas planas: clasificación y conceptos
- Volumen de la vista y ventana de visualización
- Deducción de matrices de proyección
- Comentarios sobre implementación: transformaciones de normalización
- Transformaciones de proyección en OpenGL

Proceso visualización 3D



Necesitamos

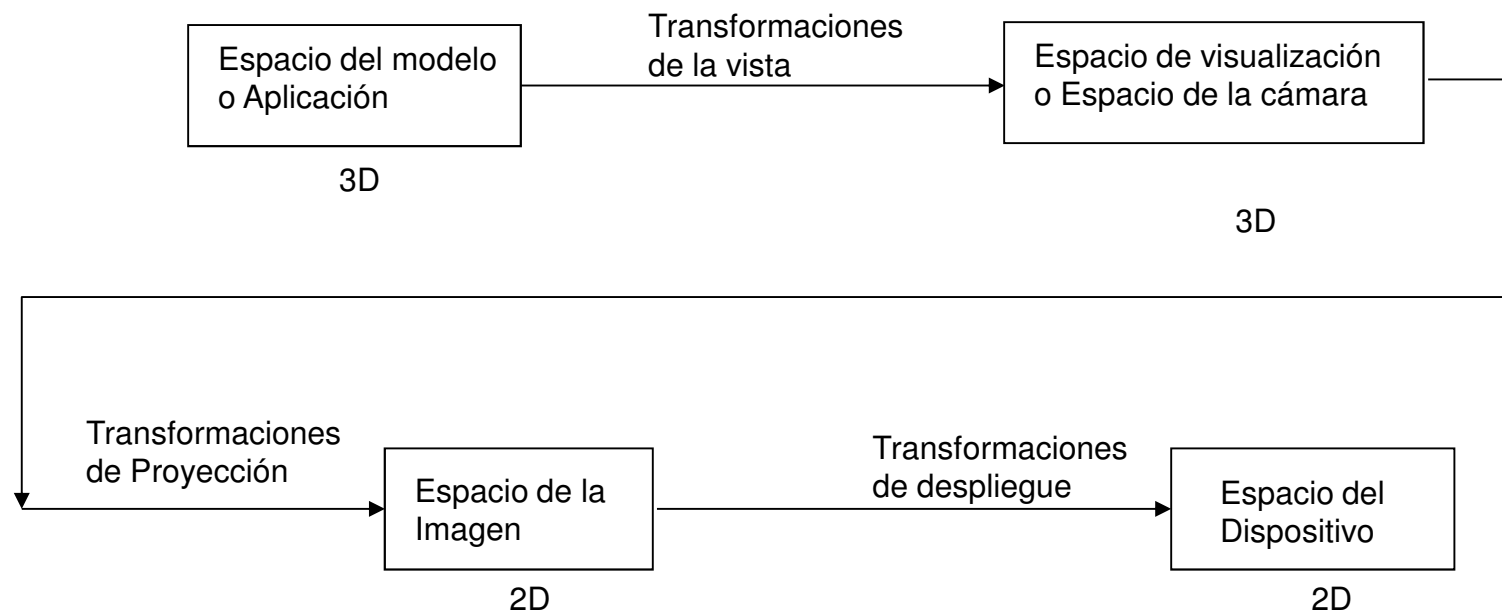
- Proyecciones: transforman objetos 3D en proyecciones en plano 2D
- Volumen de la vista
- Plano de proyección que se reduce a ventana (viewport en el dispositivo)
- Clipping (recorte) contra volumen de la vista

Visualización 3D → 2D

Rendering (pintado) de la imagen

- Es el proceso completo que permite pasar de la escena modelada en el espacio 3D de la aplicación al espacio raster (discretizado) del dispositivo.
- Se obtiene así la imagen de la escena en el dispositivo.
- El proceso de rendering se describe mediante una línea de flujo (rendering pipeline) que describe la secuencia de métodos, técnicas y algoritmos necesarios para obtener la imagen

Flujo del rendering simplificado



Elementos útiles en proyección del mundo 3D en CG

- Transformaciones de proyección. Son herramientas matemáticas que permiten pasar de la escena del 'mundo' en 3D a la ventana de visualización 2D en un dispositivo / monitor / pantalla / papel.
- Window: ventana rectangular de visualización en coordenadas del mundo
- Viewport (ventana rectangular de visualización en la pantalla del dispositivo gráfico). de lados paralelos a los lados de la pantalla.

Clipping (Recorte)

- Clipping (recorte). Concepto importante que se traduce en algoritmos / técnicas que permiten restringir los objetos / datos en 2D o, 3D, a la porción de éstos que son 'visibles' en un viewport o en un 'volumen de la vista' de la escena en 3D.
- El volumen de la vista contiene el pedazo de la escena 3D cuya proyección será visible en la ventana de visualización (corresponde a la vista de una cámara) y después en el viewport. Clipping es un concepto clave para reducir el trabajo computacional.
- Clipping también se usa en algoritmos de eliminación de caras no visibles para recortar un polígono contra polígono.

Proyecciones Geométricas Planas

(materia presentada según Voley-Van Dam et al)
Referencia muy importante en CG

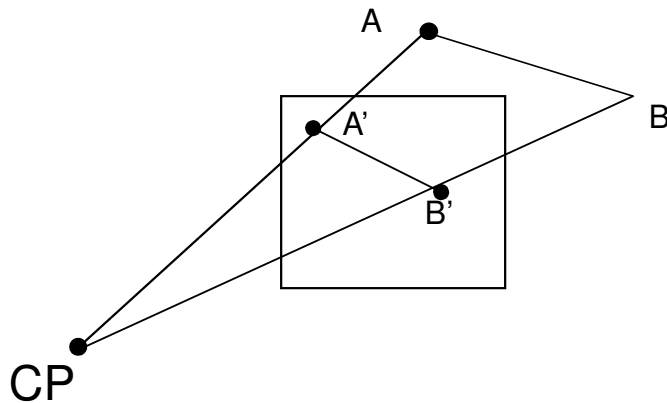
Proyecciones Geométricas Planas

Conceptos: proyectores rectos, centro de proyección, plano de proyección

Proyecciones

Perspectiva

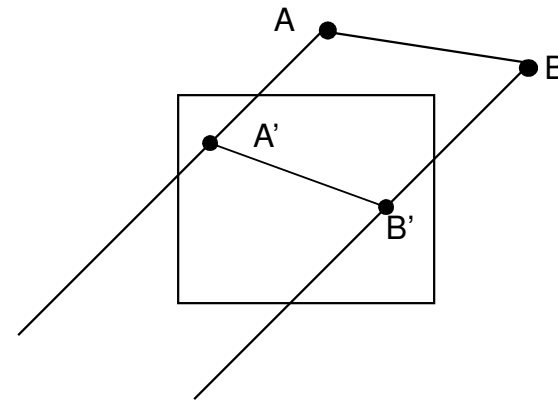
Centro de proyección a distancia finita del plano de proyección



(proyectores convergen)

Paralela

Centro de proyección a distancia infinita del plano de proyección



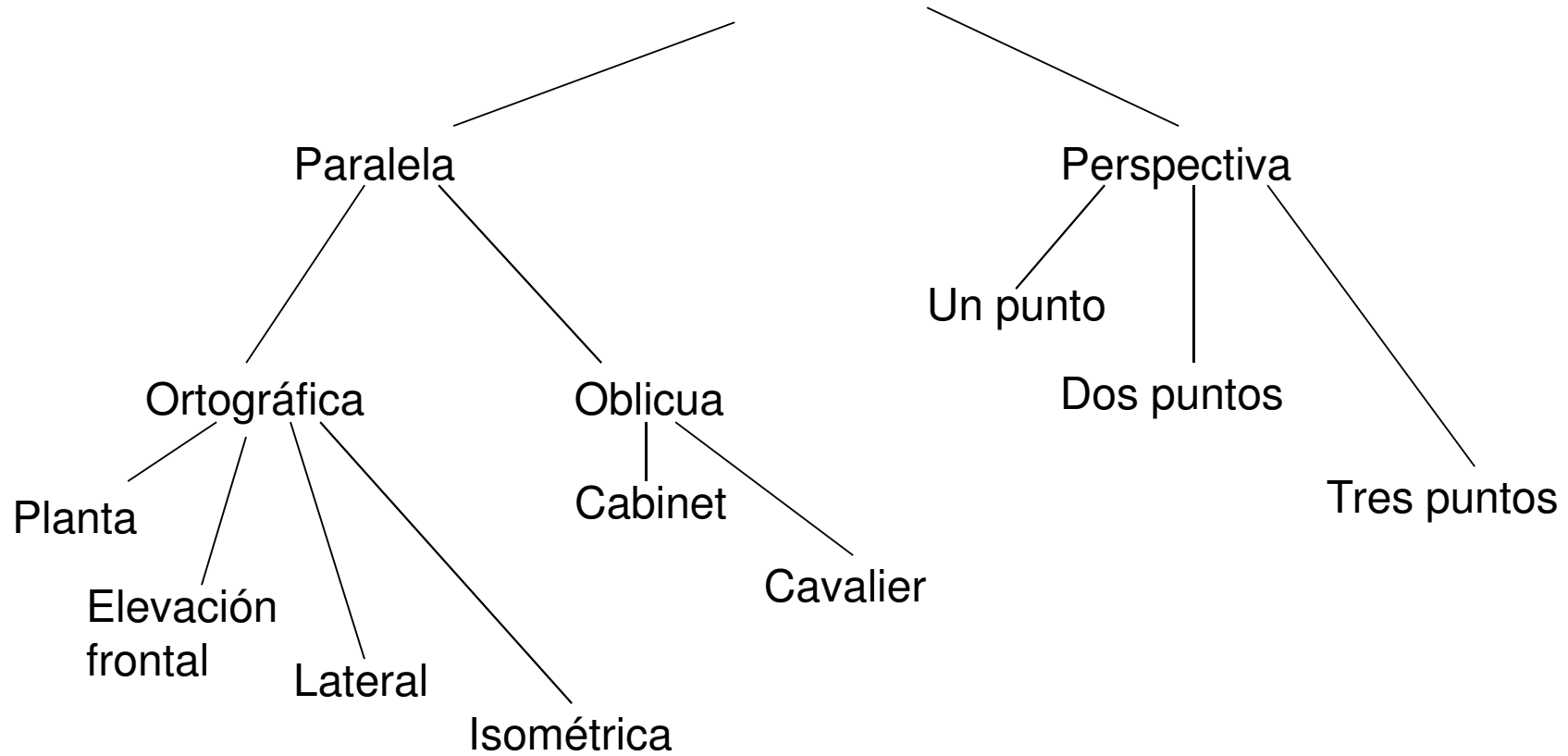
CP en el infinito
(proyectores paralelos)

Proyección de Perspectiva

- El tamaño del objeto varía inversamente con la distancia del objeto al centro de proyección.
- Objetos parecen más realistas
- No es útil para almacenar forma y medidas exactas de los objetos. Las líneas paralelas en general no se mantienen paralelas.
- Proyecciones de líneas paralelas que no son paralelas al plano de proyección convergen en un punto de anulación (vanishing point)

Clasificación

Proyecciones Geométricas Planas



Proyección Perspectiva (clasificación)

- Un punto. El plano de proyección corta un eje principal (plano de proyección paralelo a un plano del sistema de coordenadas).
- Dos puntos. Plano de proyección corta dos ejes principales.
- Tres puntos Plano de proyección corta los tres ejes principales.

Ejes principales son ejes del sistema de coordenadas. En aplicaciones de ingeniería se alinean con caras importantes del objeto

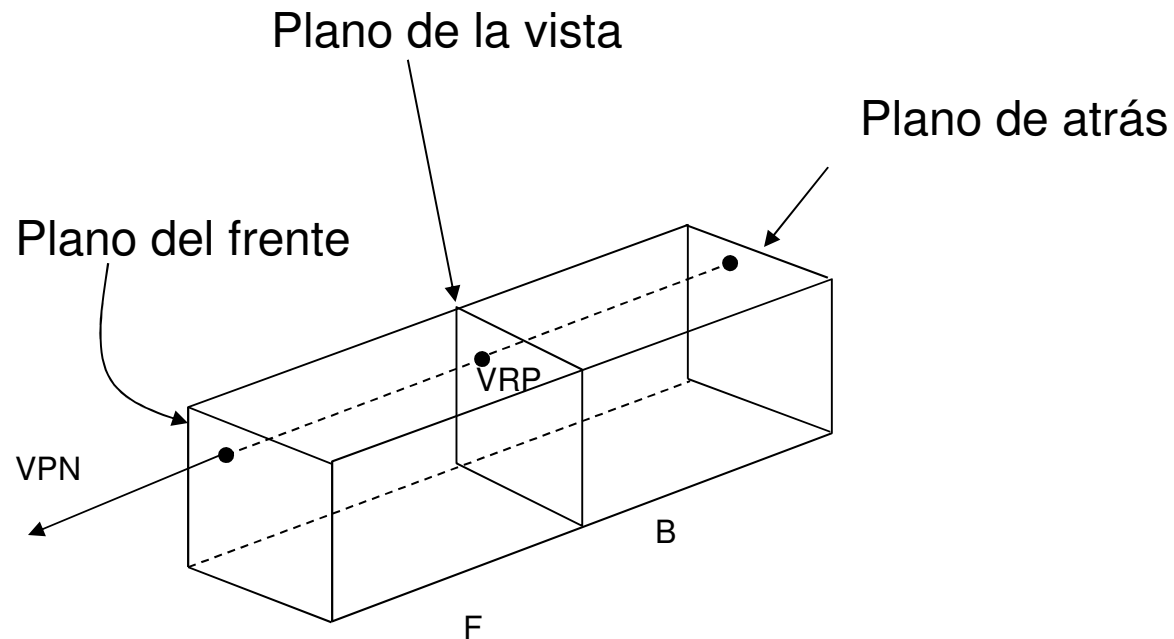
Proyección Paralela

- Ortográfica. Dirección de proyección es normal al plano de proyección
 - Planta, elevación, lateral son usadas en ingeniería. Plano de proyección es perpendicular a eje principal
 - Isométrica. Normal al plano de proyección forma ángulos iguales con ejes principales. Vistas más reales
- Oblicua. Los proyectores no son normales al plano de proyección. Plano de proyección es normal a un eje principal
 - Cavalier. Proyectores forman ángulo de 45° con plano de proyección
 - Cabinet. Proyectores forman ángulo de $\arctg(2) = 63.4^\circ$

Volumen de la vista en CG 3D

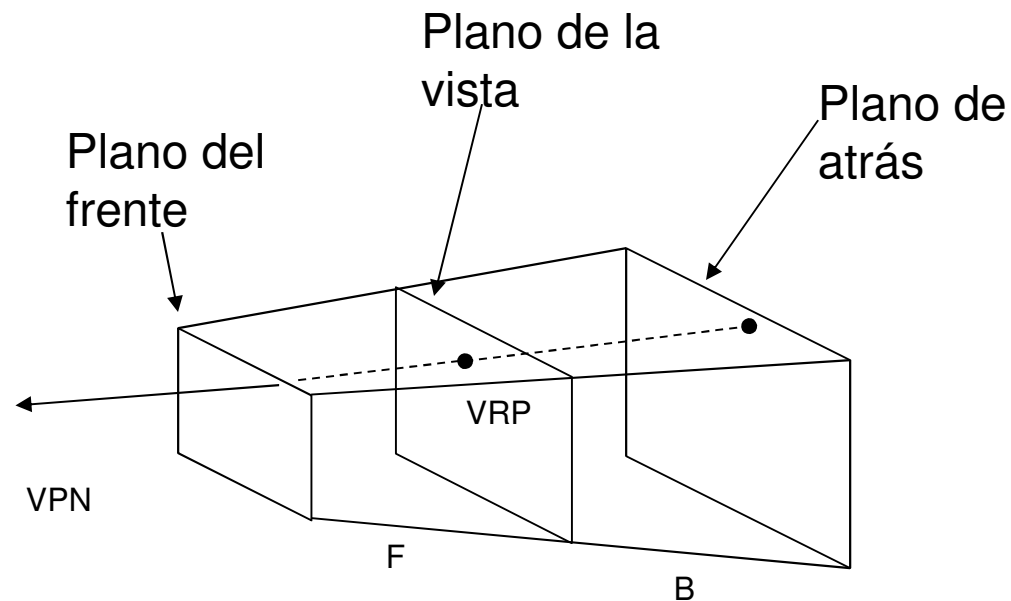
- Identifica la parte visible de la escena
- Depende la transformación de proyección utilizada

Volumen de la vista proyección paralela ortográfica



Volumen de la vista truncado para proyección paralela ortográfica

Volumen de la vista proyección perspectiva



Volumen de la vista truncado para proyección en perspectiva

Plano de visualización en 3D

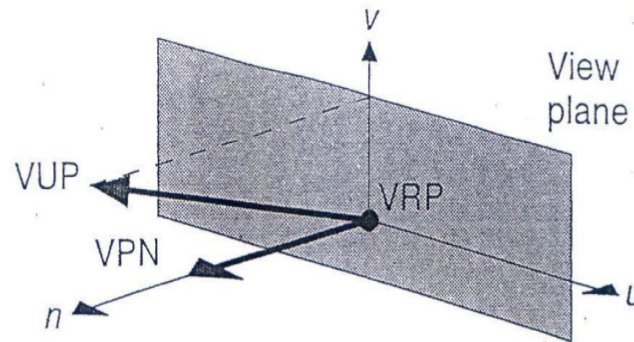


Fig. 6.14 The view plane is defined by VPN and VRP; the v axis is defined by the projection of VUP along VPN onto the view plane. The u axis forms the right-handed viewing reference-coordinate system with VPN and v .

Ventana de Visualización en 3D

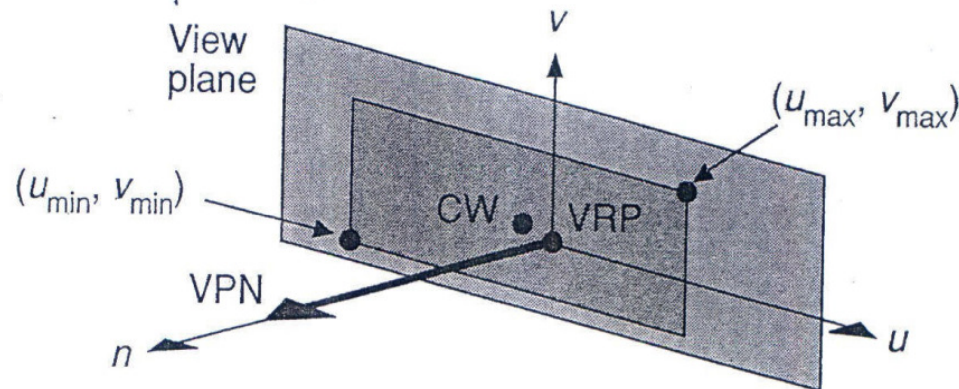
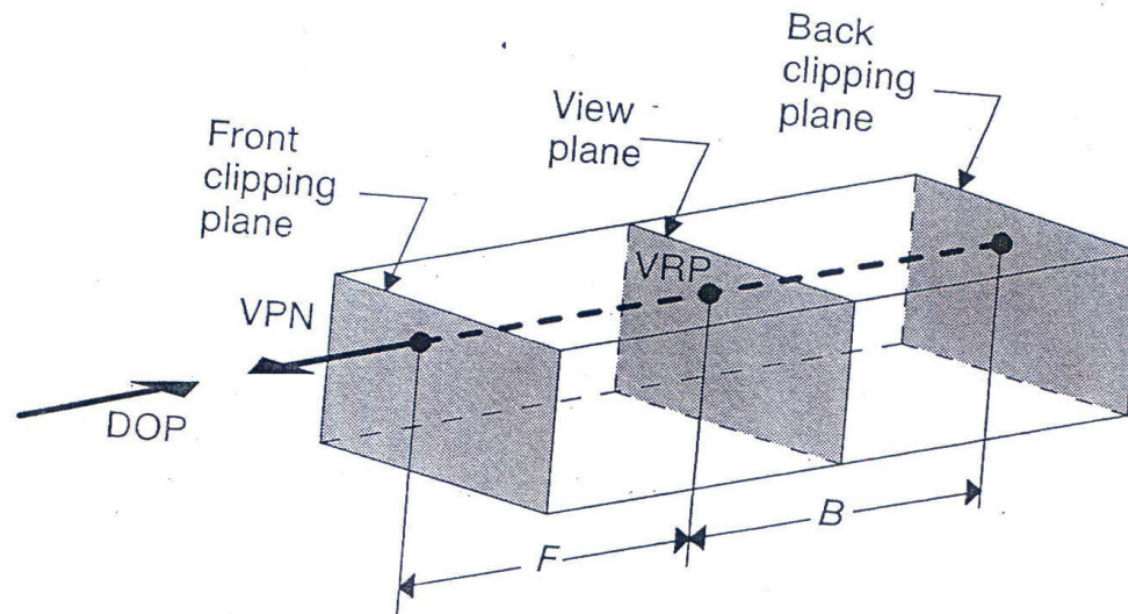


Fig. 6.15 The view reference-coordinate system (VRC) is a right-handed system made up of the u , v , and n axes. The n axis is always the VPN. CW is the center of the window.



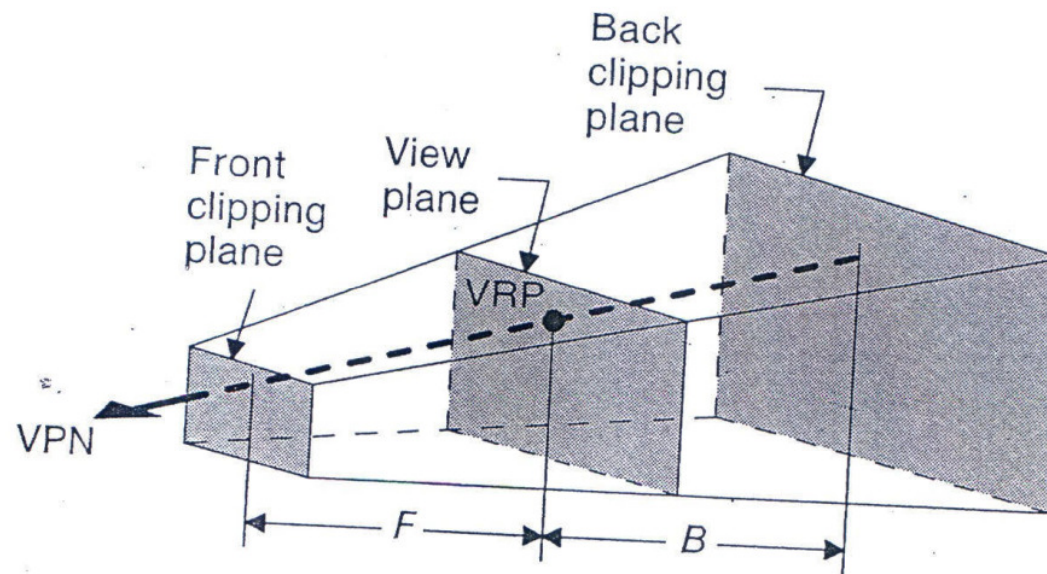
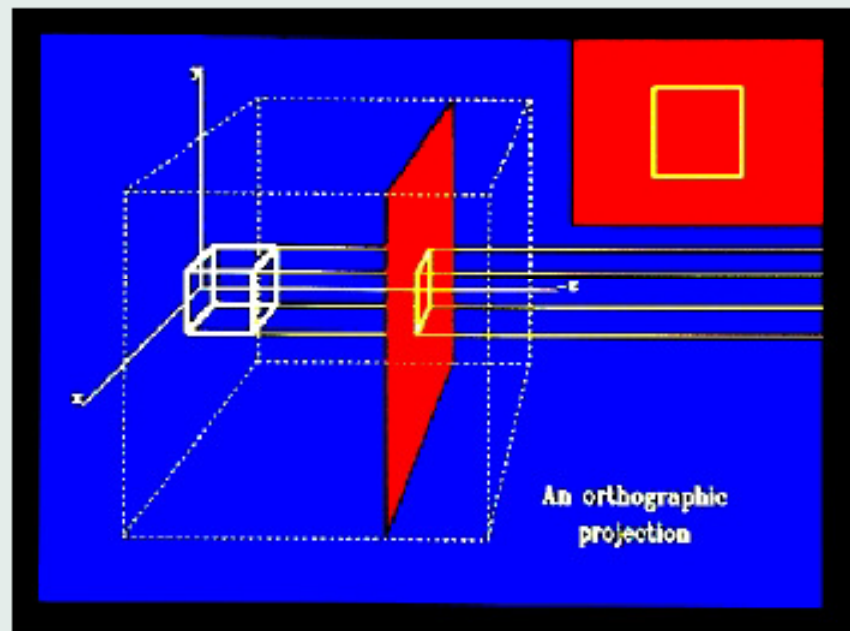


Fig. 6.21 Truncated view volume.

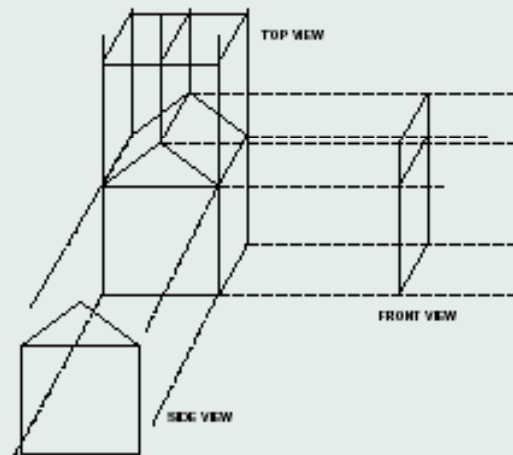
Orthographic Projection



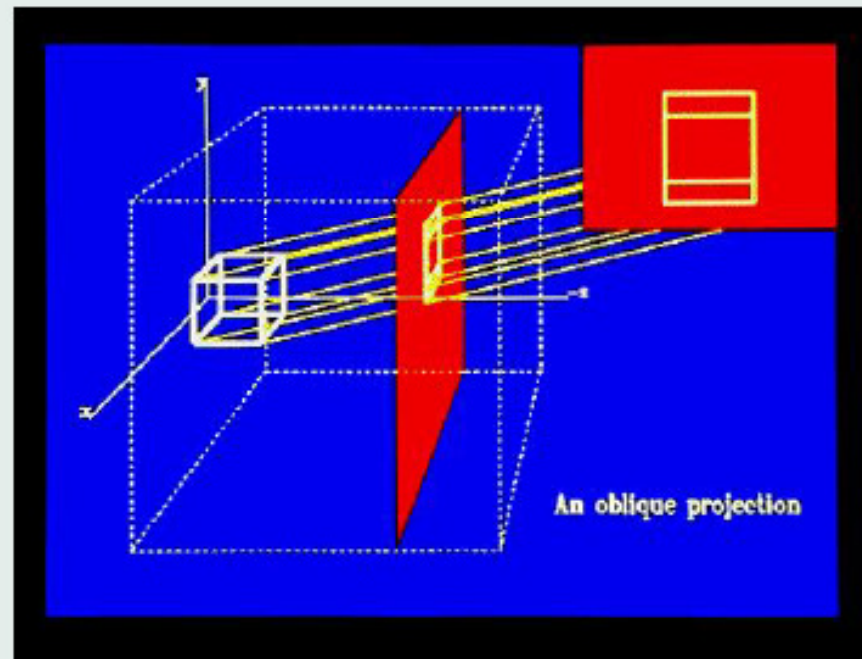
- Projectors are orthogonal to the projection plane

Multi-View Orthographic Projection

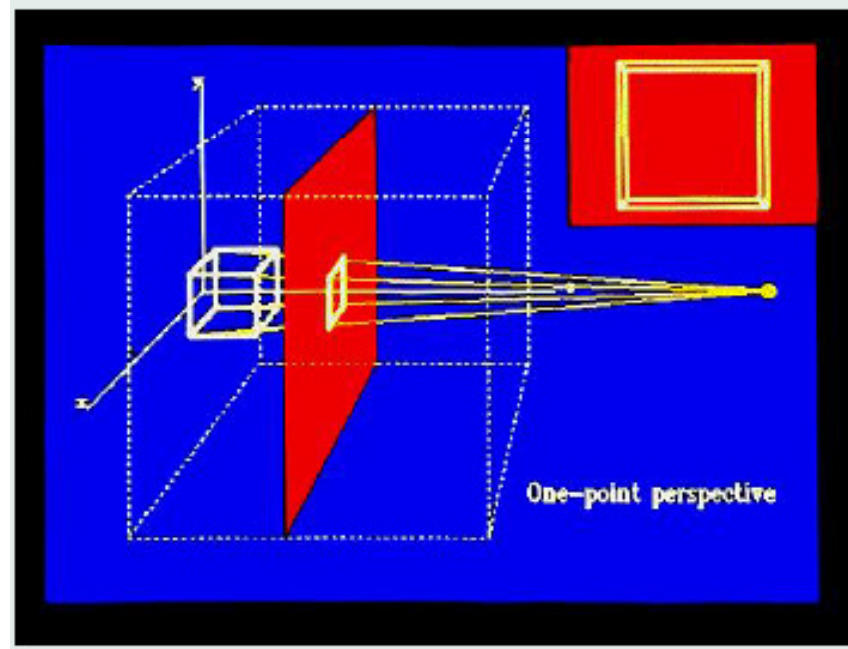
- projectors are also aligned with principal axes, common in architectural drawings (Top, Side and Front Views)



Oblique Projections



Perspective Projections: 1-Point Perspective



Plano de proyección y sistema de coordenadas

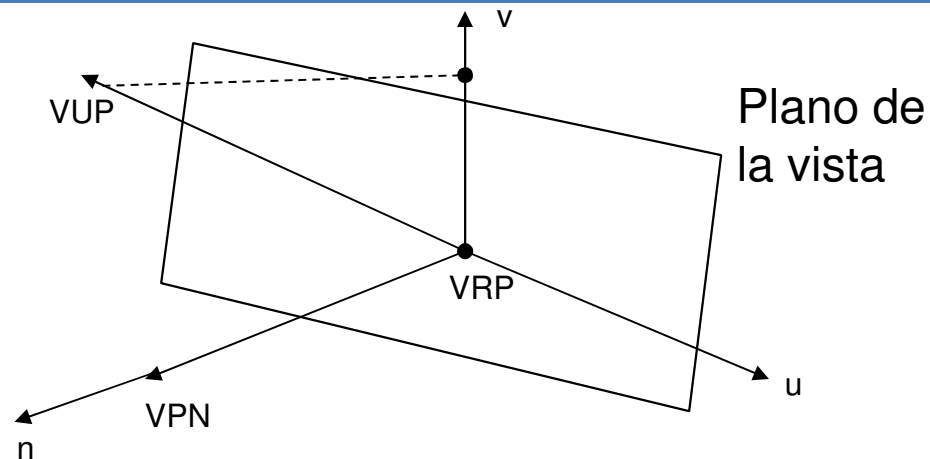
En CG en abstracto, según convenciones Foley-van Dam et al

Plano de Proyección

- VRP (view reference point) Punto sobre el plano
- VPN (view plano normal) Normal al plano

VRC Sistema de coordenadas de referencia de la vista (u, v, n)

- VRP: Origen del sistema
- VPN: Un eje del sistema (n)
- VUP (view up vector): Determina el eje de dirección hacia arriba.
La proyección de VUP sobre el plano de la vista define el eje v .



Ventana de visualización. Definida por límites $(u_{\min}, v_{\min})$ $(u_{\max}, v_{\max})$.

No necesariamente centrada en CW (centro de la ventana)

Proyección. Se define mediante PRP (projection reference point) e indicador del tipo de proyección

- Si es perspectiva PRP es el centro de proyección
- Si es paralela, la dirección de proyección (DOP) va desde PRP a CW

Deducción de una matriz de perspectiva simétrica simple (volumen de la vista simétrico)

Perspectiva: plano de proyección en $z=d$

$$M_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}.$$

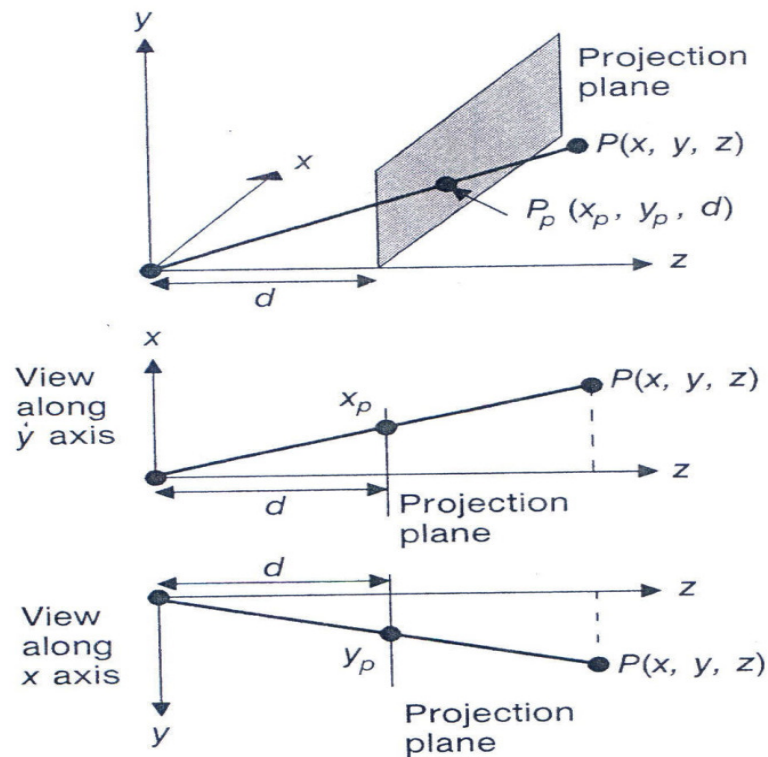


Fig. 6.42 Perspective projection.

Triángulos semejantes

$$\frac{x_p}{d} = \frac{x}{z} \Rightarrow x_p = \frac{x}{z/d}$$

$$\frac{y_p}{d} = \frac{y}{z} \Rightarrow y_p = \frac{y}{z/d}$$

Matricialmente

$$M_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

Multiplicando M_{per} por $P = [x \ y \ z \ 1]^T$

$$M_{\text{per}} P = [x \ y \ z \ z/d]^T \quad w = z/d \neq 1$$

Luego para recuperar el punto en $\mathbb{R}^3$ es necesario dividir por w .

$$\text{Luego } (x_p, y_p, z_p) = \left(\frac{x}{x/d}, \frac{y}{z/d}, 1 \right)$$

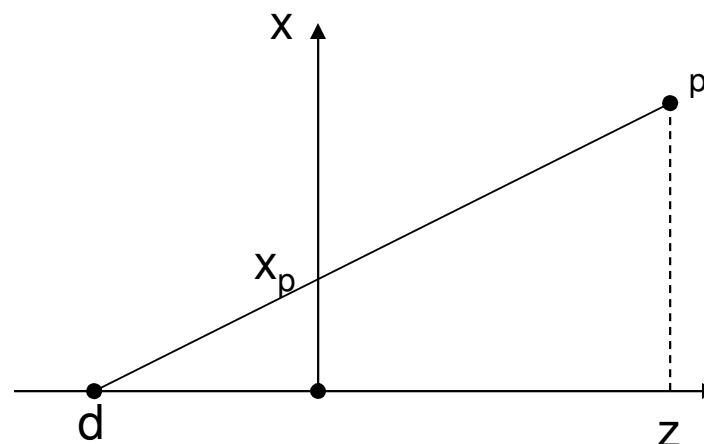
Perspectiva (Otra deducción)

plano de proyección $z = 0$, centro de proyección en $z = -d$

$$\frac{x_p}{d} = \frac{x}{z+d} \quad \frac{y_p}{d} = \frac{y}{z+d}$$

$$x_p = \frac{x}{z/d+1} \quad y_p = \frac{y}{z/d+1}$$

$$M'_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$



En este caso d puede tender a infinito

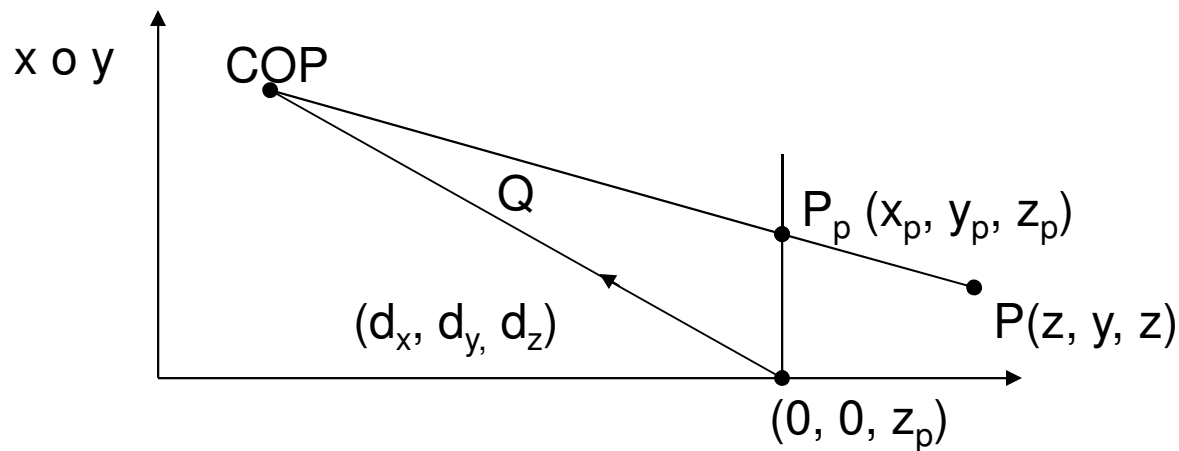
Proyección Paralela Ortográfica

Plano de proyección en $z = 0$

dirección de proyección $\equiv$ normal al plano de proyección

$$M_{\text{ort}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Deducción general que integra proyecciones en paralela y perspectiva



- plano de proyección $z = z_p$
- centro de proyección COP a distancia Q del punto $(0, 0, z_p)$
- dirección desde $(0, 0, z_p)$ a COP dado por vector de dirección normalizado (d_x, d_y, d_z)

Ec. paramétrica línea recta de COP a P

$$P_L(t) = COP + t (P - COP) \quad 0 \leq t \leq 1$$

Además $COP = (0, 0, z_p) + Q (d_x, d_y, d_z)$

Luego para P' cualquiera sobre la línea (ecuación paramétrica)

$$x' = Qd_x + (x - Qd_x) t$$

$$y' = Qd_y + (y - Qd_y) t$$

$$z' = (z + Qd_z) + (z - (z_p + Qd_z)) t$$

P_p se encuentra en la intersección de $P_L(t)$ con plano de proyección

$z = z_p$. Luego haciendo $z' = z_p$ y resolviendo para t

$$t = \frac{z_p - (z_p + Qd_z)}{z - (z_p + Qd_z)}$$

Luego:

$$x_p = \frac{x - z \frac{d_x}{d_z} + z_p \frac{d_x}{d_z}}{D}$$

$$y_p = \frac{y - z \frac{d_y}{d_z} + z_p \frac{d_y}{d_z}}{D}$$

$$z_p = \frac{-z \frac{z_p}{Qd_z} + \frac{z_p^2}{Qd_z} + z_p \frac{Qd_z}{Qd_z}}{D}$$

$$\text{donde } D = \frac{z_p - z}{Qd_z} + 1$$

(se trabajó sobre identidad $z_p = z_p$ para tener el mismo denominador D)

$$M_{\text{general}} = \begin{bmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p \frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p \frac{d_y}{d_z} \\ 0 & 0 & -\frac{z_p}{Qd_z} & \left(\frac{z_p^2}{Qd_z} + z_p \right) \\ 0 & 0 & -\frac{1}{Qd_z} & \left(\frac{z_p}{Qd_z} + 1 \right) \end{bmatrix}$$

Incluye M_{per} M'_{per} M_{ort}

	z_p	Q	(d_x, d_y, d_z)
M_{ort}	0	∞	(0, 0, -1)
M_{per}	d	d	(0, 0, -1)
M'_{per}	0	d	(0, 0, -1)

Plano de la vista

- Puede ubicarse alternativamente sobre el plano del frente o fuera del volumen de la vista.
- Los distintos sistemas gráficos usan convenciones levemente distintas.
- En OpenGL el plano de la vista corresponde al plano del frente

Proyecciones en OpenGL

Función glFrustum

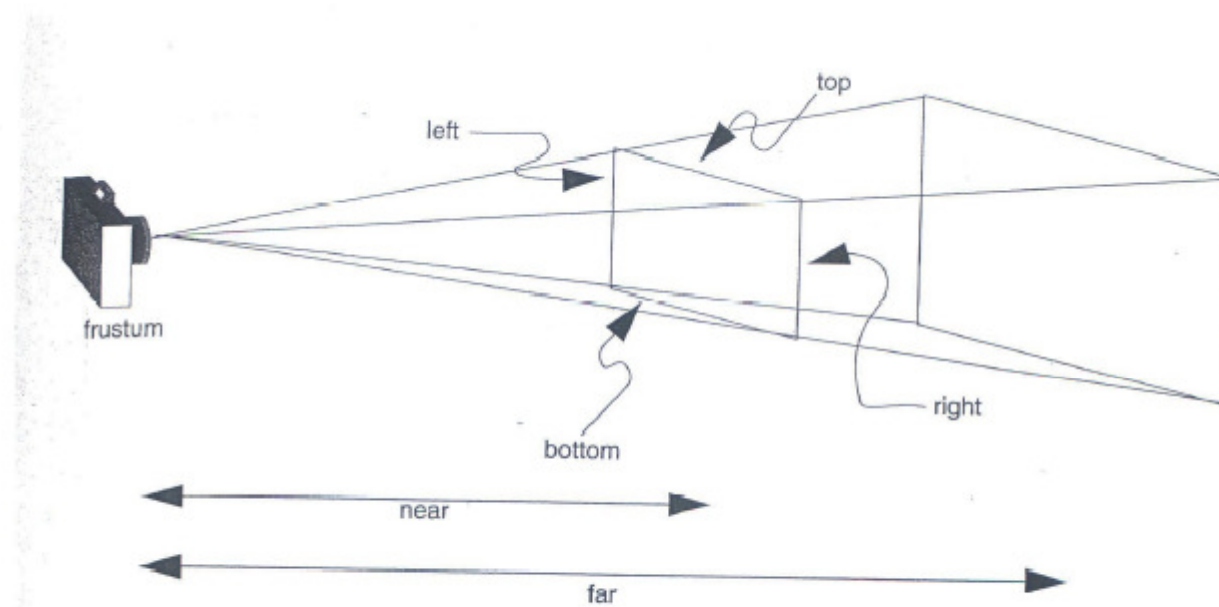


Figure 3-13 Perspective Viewing Volume Specified by `glFrustum()`

Función glFrustrum

(muy simple y con poco control)

- Frustrum es el volumen de la vista
- Near y far son distancias desde el punto de vista a los planos (de clipping) de adelante y atrás.
- Coordenadas (left, bottom, -near), (right, top -near) define la ventana de la vista.
- No necesariamente define volumen de la vista simétrico.

Función gluPerspective

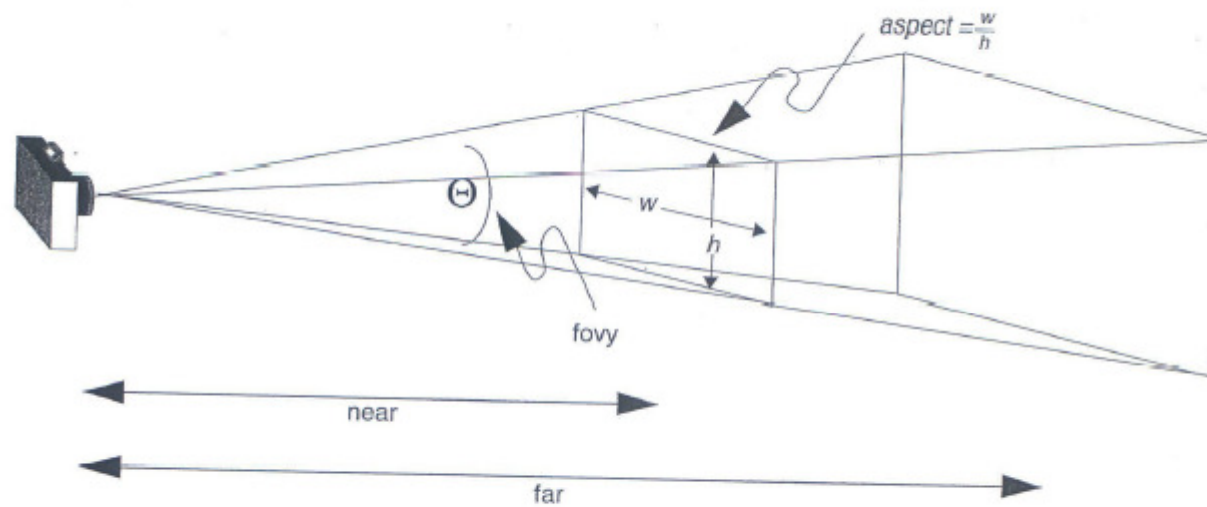


Figure 3-14 Perspective Viewing Volume Specified by `gluPerspective()`

Función gluPerspective (avanzado)

➤ Volumen de la vista simétrico

➤ Parámetros

- fovy: ángulo del campo de vista en plano x-z. Varía entre 0° y 180° .
- aspect: razón de aspecto de la ventana de la vista. Ancho/altura.
- near y far distancias desde el punto de vista a los planos de clipping de adelante (near) y atrás (far) a lo largo del eje z. Son valores positivos.

Proyección paralela ortográfica

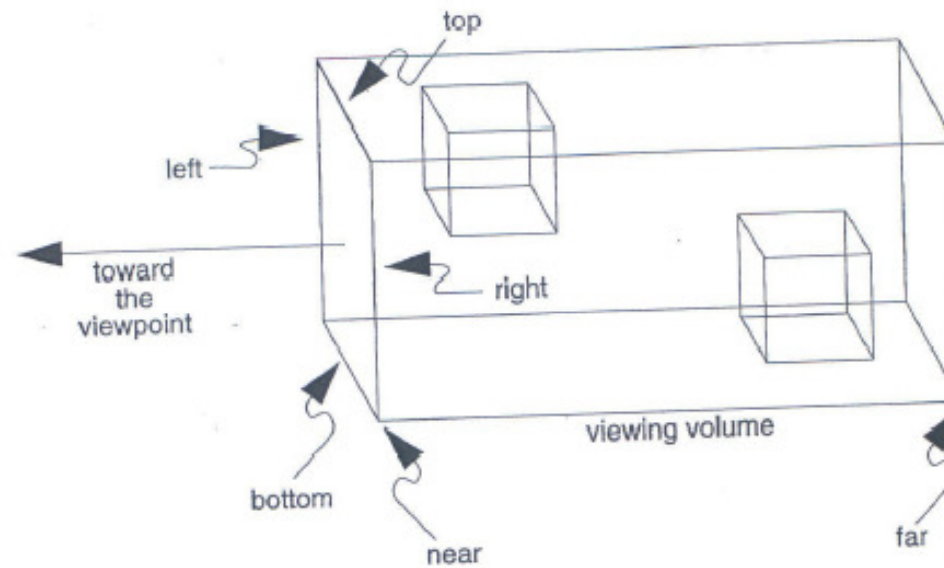


Figure 3-15 Orthographic Viewing Volume

Even after you've aimed the camera in the correct direction and you can see your objects, they might appear too small or too large. If you're using `gluPerspective()`, you might need to alter the angle defining the field of view by changing the value of the first parameter for this command. You can use trigonometry to calculate the desired field of view given the size of the object and its distance from the viewpoint: The tangent of half the desired angle is half the size of the object divided by the distance to the object (see Figure 3-19). Thus, you can use an arctangent routine to compute half the desired angle. Example 3-3 assumes such a routine, `atan2()`, which calculates the arctangent given the length of the opposite and adjacent sides of a right triangle. This result then needs to be converted from radians to degrees.

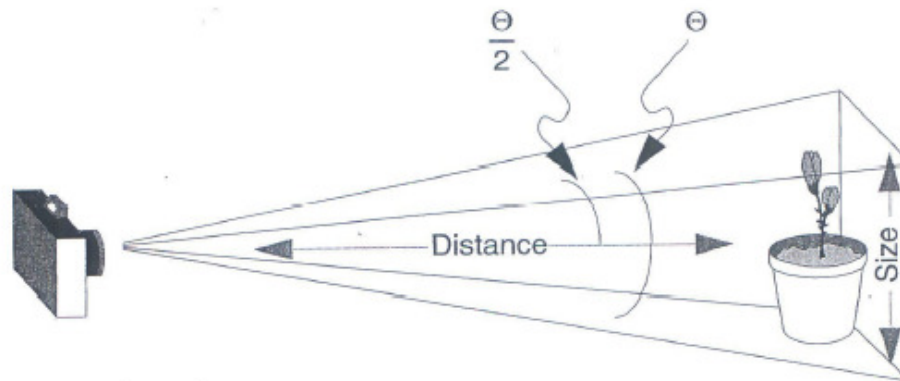


Figure 3-19 Using Trigonometry to Calculate the Field of View

Example 3-3 Calculating Field of View

```
#define PI 3.1415926535

double calculateAngle(double size, double distance)
{
    double radtheta, degtheta;

    radtheta = 2.0 * atan2 (size/2.0, distance);
    degtheta = (180.0 * radtheta) / PI;
    return (degtheta);
}
```

Of course, typically you don't know the exact size of an object, and the distance can only be determined between the viewpoint and a single point in your scene. To obtain a fairly good approximate value, find the bounding box for your scene by determining the maximum and minimum x , y , and z coordinates of all the objects in your scene. Then calculate the radius of a bounding sphere for that box, and use the center of the sphere to determine the distance and the radius to determine the size.

For example, suppose all the coordinates in your object satisfy the equations $-1 \leq x \leq 3$, $5 \leq y \leq 7$, and $-5 \leq z \leq 5$. Then the center of the bounding box is (1, 6, 0), and the radius of a bounding sphere is the distance from the center of the box to any corner—say (3, 7, 5)—or

$$\sqrt{(3-1)^2 + (7-6)^2 + (5-0)^2} = \sqrt{30} = 5.477$$

If the viewpoint is at (8, 9, 10), the distance between it and the center is

$$\sqrt{(8-1)^2 + (9-6)^2 + (10-0)^2} = \sqrt{158} = 12.570$$

The tangent of the half angle is 5.477 divided by 12.570, which equals 0.4357, so the half angle is 23.54 degrees.

Remember that the field-of-view angle affects the optimal position for the viewpoint, if you're trying to achieve a realistic image. For example, if your calculations indicate that you need a 179-degree field of view, the viewpoint must be a fraction of an inch from the screen to achieve realism. If your calculated field of view is too large, you might need to move the viewpoint farther away from the object.