Vistas 3D

Los objetos se definen en el marco de referencia, luego se cambian mediante transformaciones afines al marco de modelo (coordenadas mundiales), y luego al marco de vista (coordenadas de vista). Se distingue entre objetos e imágenes físicas y sintéticas.

Especificación de la Cámara

Los objetos existen en un espacio independiente del observador (viewer).
Para especificar objetos sencillos (pimitivas gráficas, como línea, cuadrado) se pueden utilizar vértices.
CAD tiene como objetivo proveer una forma fácil de construir modelos sintéticos del mundo.
Cualquier sistema gráfico debe proveer medios para formar imágenes de objetos.

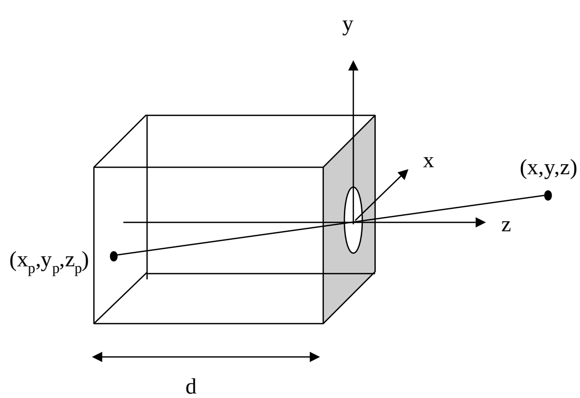
Generar una vista es parecido a tomar una fotográfica. El tipo y el tamaño del lente de la cámara determinan la imagen final. OpenGL ubica la cámara en el origen del marco mundial apuntando en la dirección negativa de *z* (el marco de vista utiliza un sistema de coordenadas de mano izquierda, a diferencia del marco de modelo que utiliza un sistema de coordenadas de mano derecha). Si la matriz modelo-vista es la matriz identidad, el marco de la cámara (vista) y el marco mundial son idénticos. Si se mueve la cámara en relación al marco mundial, los vértices aún serán especificados en el marco mundial, pero la vista será de acuerdo a la nueva posición de la cámara. La matriz modelo-vista especifica la relación entre el marco de modelo y el marco de vista.

El observador puede ser humano, una cámara, o un digitalizador:

- En un humano la imagen se forma atrás del ojo.
- En una cámara se forma en el film.

El observador al igual que los objetos se ubican en 3D para generar la imagen en 2D, siendo esta la esencia de formar imágenes.

Cámara Sencilla (Pinhole)

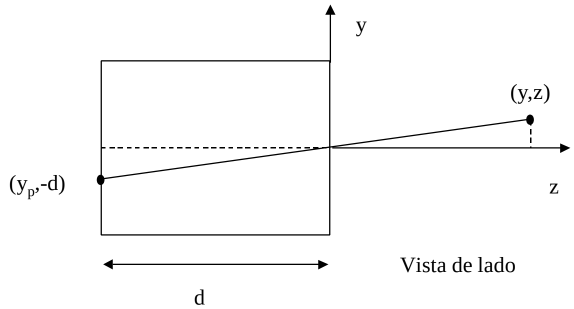


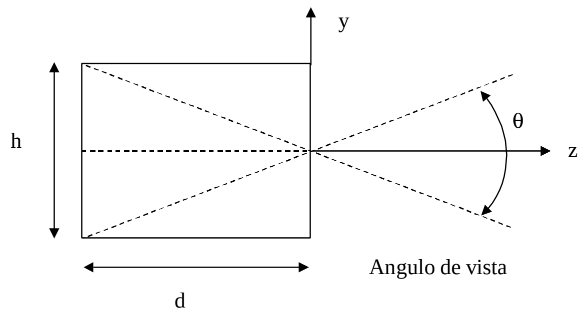
El origen del sistema de coordenadas de la cámara está en el centro del hoyo se supone que es lo suficientemente pequeño para que sólo pase un rayo de luz emanando de un punto.
La película de la cámara se encuentra del lado opuesto del hoyo y a una distancia *d* (*z* = -*d*).
Las coordenadas en el film se pueden calcular por simple triangulación:

$$x_p = -\frac{x}{z/d} \qquad y_p = -\frac{y}{z/d} \qquad z_p = -d$$

El punto (x_p,y_p,z_p) se conoce como la **proyección** de (x,y,z).
El color del punto en el plano de la película será el color del punto original.

El **campo** u **ángulo de vista** de la cámara es el ángulo del objeto más grande que se pueda obtener en la película:





Si *h* es la altura de la cámara, el ángulo de vista *q* es:

$$q = 2\tan^{-1}\frac{h}{2d}$$

La cámara ideal tiene una **profundidad de campo** infinita: Cada punto dentro del campo de vista está enfocado. La imagen de un punto es un punto. Las desventajas de esta cámara son:

- solo admite un rayo de luz una fuente de punto dada la pequeñez del hoyo, o sea que casi no entra luz en la cámara
- no se puede ajustar diferentes ángulos de vista

Reemplazando el hoyo por un lente resuelve ambos problemas:

- el lente recibe mayor cantidad de luz, cuanto más grande el lente mayor la cantidad de luz
- escogiendo un lente con la longitud focal apropiada (equivalente a escoger el valor de *d*), se puede obtener el ángulo de vista deseado (hasta 180 grados).

Sin embargo, los lentes no tienen una profundidad de campo infinita, o sea, objetos a diferentes distancias del lente no estarán todos enfocados.

Estos modelos de sistemas de imágenes ópticos es la base del **modelo de cámara sintético** utilizado en las gráficas por computadora de tres dimensiones.

La línea que une un punto de un objeto al punto correspondiente en la imagen, pasando por el centro del lente, se conoce como **proyector**
El centro del lente se conoce como el **centro de proyección**.
Todos los proyectores emanan del centro de proyección.
El plano de la película frente al lente se conoce como el **plano de proyección**.
El tamaño de una imagen se expresa por el ángulo de vista, correspondiente al **rectángulo de recorte** o **ventana de recorte** (**clipping**) en el plano de proyección.

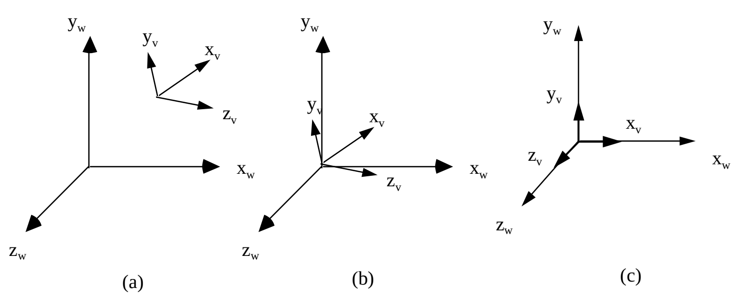
Este rectángulo actúa como una ventana a través de la cual un observador, ubicado en el centro de proyección ve el mundo.
Dada la ubicación del centro de proyección, la ubicación y orientación del plano de proyección, y el tamaño del rectángulo de recorte, se puede determinar que objetos aparecerán en la imagen.

Transformación del marco mundial al marco de vista

Una alternativa sería transformar el marco del mundo en relación al de vista. La transformación de las descripciones de objetos de coordenadas mundiales a coordenadas de vista equivale a una transformación que superpone el marco de referencia de vista sobre el marco mundial al utilizar operaciones geométricas básicas de traslación y rotación. Esta secuencia de transformaciones es:

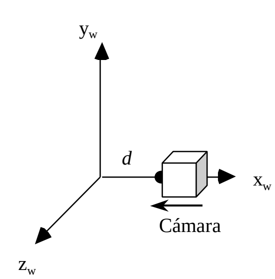
1. Se traslada el punto de referencia de vista al origen del sistema de coordenadas mundiales. Si el punto de referencia de vista se especifica en una posición mundial (x_0, y_0, z_0), se traslada este punto al origen mundial con la matriz de transformación $T(-x_0, -y_0, -z_0)$.
2. Se aplican rotaciones para alinear los ejes x_v, y_v y z_v con los ejes mundiales x_w, y_w y z_w, de manera respectiva. La secuencia de rotación puede implicar hasta tres rotaciones del eje de coordenadas, según la posición que se seleccione para ***N***. En general, si ***N*** no esta alineado con ninguno de los ejes de coordenadas, podemos superponer los sistemas de vista y mundial con la secuencia de transformaciones $R_x \cdot R_y \cdot R_z$. Es decir, primero giramos alrededor del eje mundial de x_w para trasladar z_v al plano $x_w z_w$. De este modo efectuamos una rotación con respecto del eje y x_w para alinear los ejes de y_w y y_v. Además, si el sistema de referencia de vista es de lado izquierdo, también es necesaria una reflexión de uno de los ejes de vista (e.g. el eje z_v).
3. Se traslada el punto de referencia de vista del origen a su posición original, $T(x_0, y_0, z_0)$.

La siguiente figura ilustra la secuencia general de transformaciones de traslación y rotación.



Entonces se aplica la matriz de transformación compuesta en las descripciones de coordenadas mundiales para transformarlas a coordenadas de vista.

Por ejemplo, si originalmente la cámara está ubicada en el origen del marco mundial viendo en la dirección negativa de *z*, y se desea ubicar la cámara en el eje *x* viendo en la dirección negativa de *x* y ubicando la cámara en *x*=*d*, como se muestra en la figura:



Se llevarían a cabo las siguientes transformaciones del marco mundial en OpenGL en relación a la posición de la cámara:

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glRotatef(90.0, 0.0, 1.0, 0.0);
glTranslatef(-d, 0.0, 0.0);
```

El orden es primero rotación (z a x) y luego traslación por $-d$, dado que mover la cámara es como mover el mundo en la dirección opuesta.

Otro método para generar la matriz de transformación y rotación consiste en calcular los vectores unitarios *uvn* y formar la matriz de rotación compuesta de manera directa. Dados los vectores $\boldsymbol{N}$ y $\boldsymbol{V}$, se calculan estos vectores unitarios como

$$\boldsymbol{n} = \boldsymbol{N} / |\boldsymbol{N}| = (n_1,\ n_2,\ n_3)$$

$$\boldsymbol{u} = \boldsymbol{V}\times\boldsymbol{N} / |\boldsymbol{V}\times\boldsymbol{N}| = (u_1,\ u_2,\ u_3)$$

$$\boldsymbol{v} = \boldsymbol{n}\times\boldsymbol{u} = (v_1,\ v_2,\ v_3)$$

Este método también ajusta en forma automática la dirección de $\boldsymbol{V}$ de modo que $\boldsymbol{v}$ sea perpendicular a $\boldsymbol{n}$.
La matriz de rotación compuesta para la transformación de vista es

$$R = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

que transforma $\boldsymbol{u}$ en el eje mundial de x_w, $\boldsymbol{v}$ en el eje de y_w *y* $\boldsymbol{n}$ en el eje de z_w. Esta matriz lleva a cabo de manera automática la reflexión necesaria para transformar un sistema de vista de lado izquierdo en un sistema mundial de lado derecho.

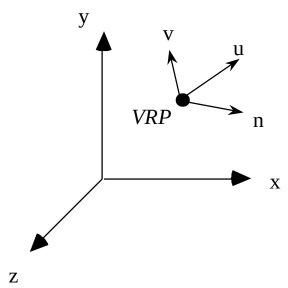
La transformación completa de coordenadas mundiales a coordenadas de vista se obtiene como el producto de matriz

$$M_{VC\leftarrow WC} = R \cdot T^{-1}$$

Especificación de Vista

Por lo general, en los distintos APIs (GKS-3D, PHIGS, OpenGL) se tiene un enfoque diferente para la especificación de la vista (posición de la cámara), la cual se hace mediante la especificación directa del marco de vista (**transformación de normalización**) y no mediante transformaciones afines básicas desde el marco mundial.

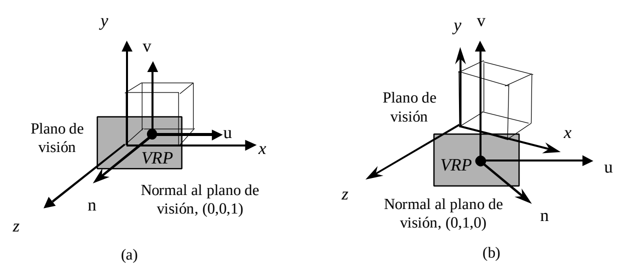
- **Sistema de coordenadas de** (**referencia**) **vista** o **sistema u-v-n** (**VRC - Viewing-Reference Coordinate**) **-** se establece el sistema de coordenadas en referencia a la cámara.
- **Matriz de Orientación de Vista -** es la matriz de transformación para el cambio de marcos (modelo-vista).
- **Punto de referencia de vista** (**VRP - View Reference Point**) **-** establece el marco de referencia de las coordenadas de vista, seleccionado el origen del sistema de coordenadas de vista.



En OpenGL, la cámara está centrada en el origen del marco mundial (VRP = (0,0,0)), mirando en la dirección negativa de *z*. El usuario asigna el VRP mediante:

```
set_view_reference_point(x, y, z);
```

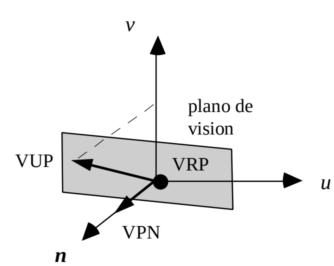
- **Plano de visión** o **proyección** (**View Plane**) **-** normalmente se define perpendicular al eje de vista de *n* y corresponde al plano de la película en la cámara.
- **Vector normal del plano de visión** (**VPN - View-Plane Normal**) **-** especifica la dirección positiva para el eje de vista *n* y la orientación del plano de visión.



En OpenGL se establece la dirección para *n* mediante la siguiente función

```
set_view_plane_normal(nx, ny, nz);
```

- **Vector de vista hacia arriba** (**VUP - View-Up Vector**) **-** indica la dirección positiva del eje de *v* del plano de vista, al especificar un vector *VUP*. Puede ser tedioso determinar la dirección para *VUP* que sea exactamente perpendicular a *n*. Por tal razón, por lo general los procedimientos de vista ajustan la orientación definida para *VUP* por el usuario, para obtener la proyección *v* en un plano perpendicular al vector normal *n*, como se ilustra en la siguiente figura. Se puede determinar que el vector de vista hacia arriba tenga cualquier dirección conveniente, en tanto que no sea paralelo a *n*.

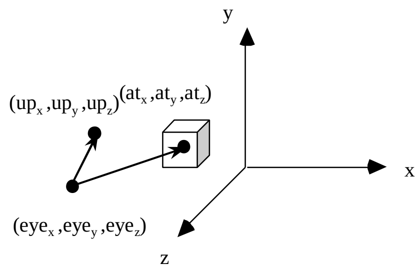


En OpenGL se establece *VUP* mediante la siguiente función

```
set_view_up(upx, upy, upz);
```

Al utilizar los vectores *n* y *v*, se puede calcular un tercer vector *u*, perpendicular tanto a *n* como a *v*, mediante el producto cruz de vectores.

El uso de VRP, VPN y VUP es una de tantas maneras que un API tiene de especificar la posición de la cámara. En muchas situaciones. Un método alterno mas directo se muestra en la siguiente figura:



La cámara se apunta a un punto. La ubicación de la cámara se llama el **punto del ojo** (**eyepoint**), y se puede especificar en el marco mundial, al igual que el punto donde apunta la cámara. Estos puntos determinan un VPN y VRP, solo se requiere añadir el VUP. OpenGL ofrece la función de utilería

```
gluLookAt(eyex, eyey, eyez, atx, aty, atz, upx, upy, upz);
```

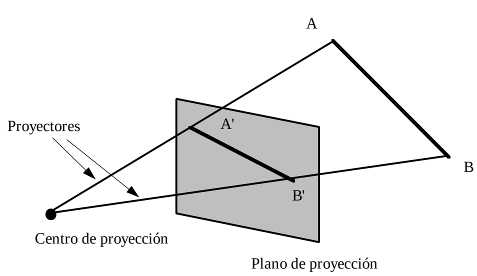
La función altera la matriz modelo-vista para una cámara que apunte a lo largo de la línea especificada.

Proyecciones

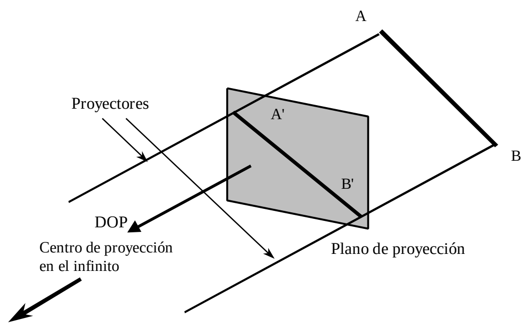
En general, las proyecciones transforman puntos en un sistema de coordenadas de dimensión *n* hacia puntos en un sistema de coordenadas de dimensión menor que *n*, en particular de interés aquí es la proyección de 3 a 2 dimensiones.

- La proyección de un objeto se define por rayos de proyección, llamados **proyectores**, que emanan del **centro de proyección (COP - Center of Projection**) o **punto de referencia de proyección** (**PRP - Projection Reference Point)**, pasando a través de cada punto del objeto, e intersectando el plano de proyección para formar la proyección. El centro de proyección puede estar a una distancia **finita** o **infinita** del plano de proyección. El COP corresponde al centro del lente en la cámara, o en el ojo, y en los sistemas de gráfica por computadora, es el origen del **marco de la cámara**.
- La clase de proyecciones tratadas aquí se conocen como **proyecciones geométricas planas**, ya que la proyección es sobre un plano y no sobre una superficie curva, y usa proyectores rectos y no curvos. Muchas proyecciones cartográficas no son planas o geométricas, como el mapa del mundo.
- Las proyecciones se pueden dividir en dos clases básicas: **perspectiva** y **paralela**. Se distinguen en la relación del centro de proyección al plano de proyección. Ambas proyecciones preservan líneas, pero por lo general no los ángulos. Según el centro de proyección se aleja, los proyectores se aproximan a líneas paralelas.

  Si la distancia del plano de proyección al centro de proyección es **finita**, entonces la proyección es **perspectiva**, como se muestra en la siguiente figura.



Si la distancia entre el centro de proyección y el plano de proyección es **infinita**, los proyectores se hacen paralelos, y la proyección es **paralela**, estableciéndose una **dirección de proyección** (**DOP - Direction of Projection**), como se muestra en la siguiente figura:



El centro de proyección es un punto con coordenadas homogéneas de tipo
centro proyección = (x, y, z, 1).
La dirección de proyección es un vector, que se puede computar restando dos puntos
dirección proyección = (x, y, z, 1) **-**(x', y', z', 1) = (a, b, c, 1)

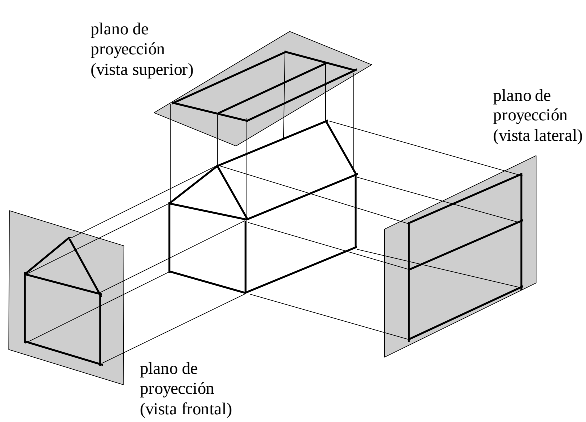
Proyección Paralela

La proyección paralela es menos realista ya que no hay acortamiento de perspectiva, aunque pueden haber diferentes acortamientos constantes sobre cada eje. La proyección puede usarse para medidas exactas, y las líneas paralelas se mantienen como tal. Como en la proyección de perspectiva, los ángulos se preservan solo sobre las caras del objeto paralelas al plano de proyección. Las proyecciones paralelas se categorizan en dos tipos, dependiendo de la relación entre la dirección de proyección y la normal al plano de proyección.

Ortogonal

La proyección **ortográfica** es la proyección ortogonal básica, y se caracteriza en que los proyectores son perpendiculares al plano de proyección, por lo cual la dirección de proyección es normal al plano de proyección.

La proyección **ortográfica multivista** define proyecciones donde el plano de proyección es paralelo a una de las caras principales del objeto. Por lo general, se despliegan al menos tres vistas: **frontal**, **superior** (**top**) y **lateral**.
La siguiente figura muestra la construcción de estas tres proyecciones; utilizadas a menudo en dibujos de ingeniería para mostrar partes de máquinas, ensamblados, y construcciones, ya que se puede medir de ellos las distancias y los ángulos, sin distorsión a los objetos.



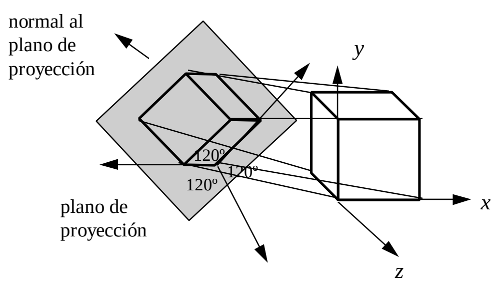
Como cada proyección muestra solo una cara de un objeto, es difícil de deducir la naturaleza tridimensional del objeto proyectado, incluso si varias proyecciones del mismo objeto se estudian simultáneamente.

Axonométrica

La proyección **axonométrica** usa planos de proyección que no son normales a un eje principal y por lo tanto muestra varias caras de un objeto a la vez. Se preserva el paralelismo de las líneas, pero no los ángulos. Existe un **acortamiento** (**foreshortening**) de las distancias con diferentes escalas para líneas no paralelas a un eje principal.

Isométrica

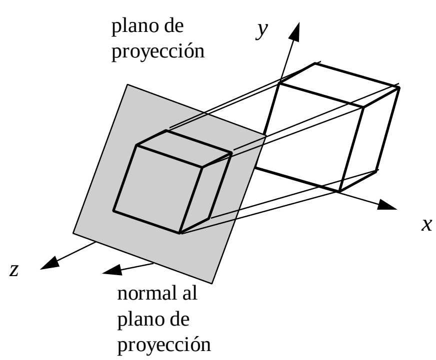
La proyección **isométrica** es una proyección axonométrica, donde la normal al plano de proyección (y por lo tanto la dirección de proyección) tiene ángulos iguales con cada eje principal.
Si la normal al plano de proyección es (d_x, d_y, d_z), entonces se requiere $|d_x| = |d_y| = |d_z|$ o $\pm d_x = \pm d_y = \pm d_z$.
Existen solo ocho direcciones (una para cada octante) que satisfacen esta condición).
La siguiente figura muestra la construcción de una proyección isométrica a lo largo de una de estas direcciones (1, -1, -1).



La proyección isométrica tiene la propiedad que sus tres ejes principales se acortan igualmente, permitiendo medidas a la misma escala a lo largo de los ejes (de aquí su nombre: *iso* para igual, *métrico* para medida). Además, las proyecciones de los ejes principales hacen ángulos iguales de 120º entre si.

Oblicua

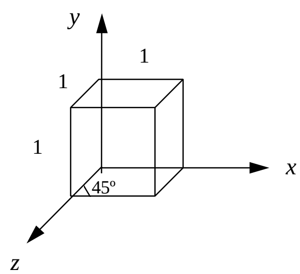
La proyección **oblicua** tiene que la dirección de proyección y la normal al plano de proyección pueden tener un ángulo arbitrario entre si. En una proyección oblicua, se preservan los ángulos en planos paralelos al plano de proyección. Las otras caras del objetos proyectadas permiten que se midan las distancias a lo largo de los ejes principales, pero no los ángulos. Las proyecciones oblicuas se utilizan mucho aunque son las mas difíciles de dibujar. También son poco naturales, ya que las cámaras y el ojo humano tienen una relación fija con el plano de imagen; por lo general el lente es paralelo al plano, generando vistas en perspectiva. La siguiente figura muestra la construcción de una proyección oblicua.



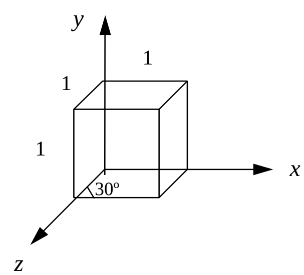
Dos proyecciones frecuentemente usadas son **caballera** (**cavalier**) y **gabinete** (**cabinet**).

Caballera

La proyección **caballera** define una dirección de proyección que no es perpendicular al plano de proyección donde la proyección de una línea perpendicular al plano de proyección tiene el mismo largo que la propia línea, o sea, no hay acortamiento.
Las siguientes figuras muestra varias proyecciones caballeras del cubo unitario al plano (*x*,*y*), formando un ángulo α con la horizontal, típicamente de 30º o 45º.
La figura (a) tiene dirección de proyección de $\left(\sqrt{2}/2, \sqrt{2}/2, -1\right)$.
La figura (b) tiene dirección de proyección de $\left(\sqrt{3}/2, 1/2, -1\right)$.



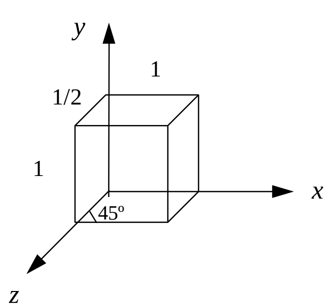
(a)



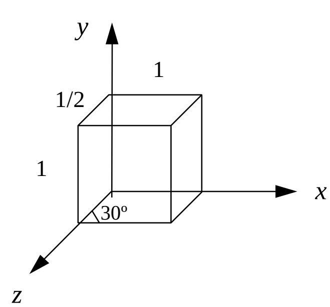
(b)

Gabinete

La proyección **gabinete** define una dirección de proyección que no es perpendicular al plano de proyección, donde la proyección de una línea perpendicular al plano de proyección tiene la mitad de su largo actual.
Las proyecciones de gabinete son un poco mas realistas que las caballeras, ya que el acortamiento por un medio va mas de acuerdo con las experiencias visuales.
Las siguientes figuras muestra varias proyecciones gabinete del cubo unitario al plano (*x*,*y*), formando un ángulo α con la horizontal, típicamente de 30º o 45º.
La figura (a) tiene dirección de proyección de $\left(\sqrt{2}/4, \sqrt{2}/4, -1\right)$.
La figura (b) tiene dirección de proyección de $\left(\sqrt{3}/4, 1/4, -1\right)$.

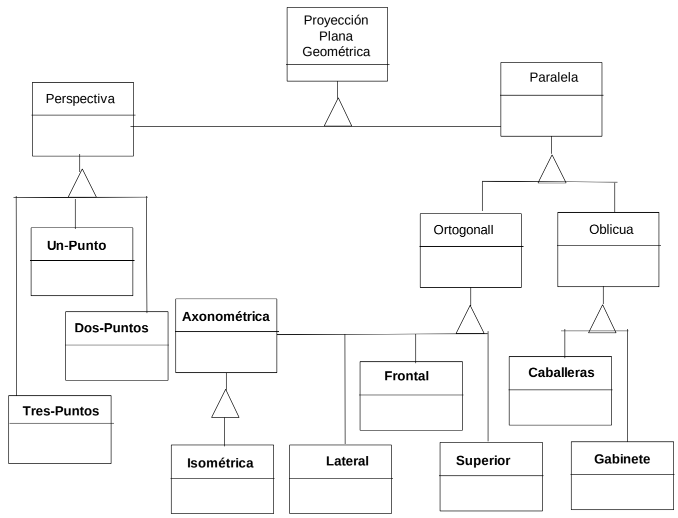


(a)



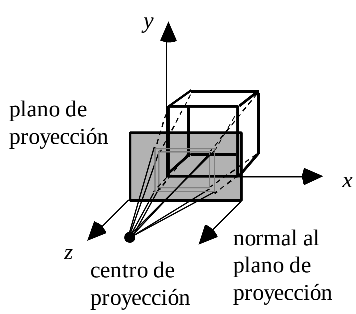
(b)

La siguiente figura muestra la relación lógica entre varios tipos de proyecciones.



Proyección Perspectiva

El efecto visual de una proyección de perspectiva es similar a la de los sistemas fotográficos y del sistema visual humano, y se le conoce como **acortamiento de perspectiva** (**perspective foreshortening**). El tamaño de la proyección de perspectiva de un objeto varía inversamente con la distancia del objeto al centro de proyección. Según se alejan los objetos, las proyecciones se vuelven mas pequeñas.
Aunque la proyección de perspectiva se vea realista, no es muy útil para obtener la forma exacta y medidas del objeto; las distancias no pueden obtenerse de la proyección, los ángulos se preservan solo sobre las caras del objeto paralelos al plano de proyección, y las líneas paralelas en general no se proyectan como tal. La siguiente figura muestra la construcción de una perspectiva, definiendo el centro de proyección con algunos de los proyectores y el plano de proyección cortando el eje de *z*.

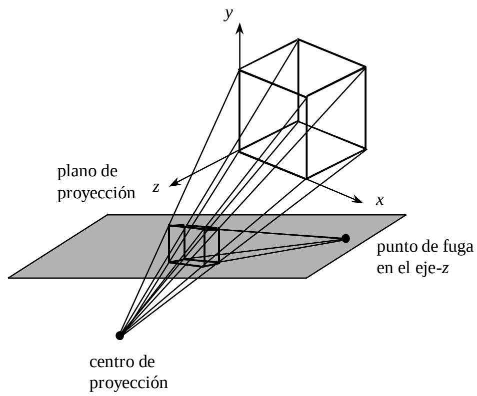


Las líneas paralelas que no son paralelas al plano de proyección convergen en un **punto de fuga** (**vanishing point**). Las líneas paralelas se juntan solo en el infinito, por lo cual se puede ver el punto de fuga como la proyección de un punto en el infinito. Existen potencialmente un número infinito de puntos de fuga, uno para cada una de la infinidad de direcciones en que se puede orientar una línea.
Las proyecciones perspectivas se clasifican principalmente en **perspectivas de un**, **dos** y **tres puntos**, dependiendo de cuantos de los tres ejes principales de un objeto son paralelos al plano de proyección, y el punto de fuga se conoce como **punto de fuga de eje** (**axis vanishing point**).

- Si dos ejes son paralelos, se obtiene **perspectiva de un punto** (eje *z*)
- Si un eje es paralelo se, se obtiene **perspectiva de dos puntos** (eje *z* y, eje *x* o *y*)
- Si ningún eje es paralelo, se obtiene **perspectiva de tres puntos** (eje *z*, eje *x* y *y*)

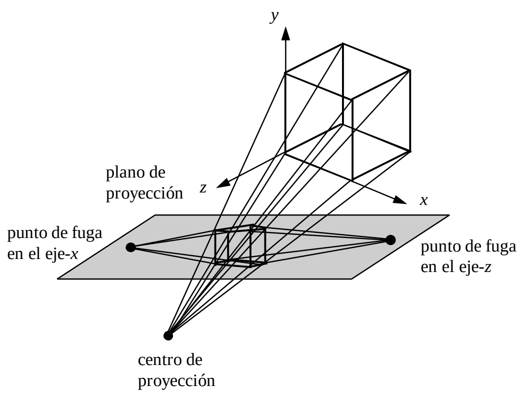
Un punto de fuga

Si el punto de fuja es en el eje *z*, se tiene un **punto de fuga de eje** (**axis vanishing point**), y las líneas paralelas a los ejes *x* y *y* se mantienen paralelas. La siguiente figura muestra una proyección en perspectiva de un punto de fuga.



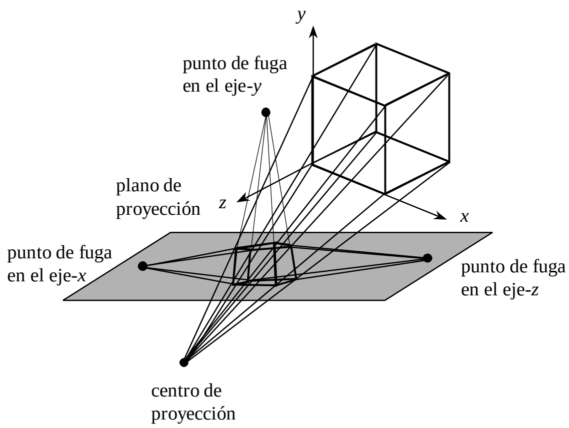
Dos puntos de fuga

La siguiente figura muestra la construcción de una perspectiva de dos puntos, donde las líneas paralelas al eje de y no convergen en la proyección.



Tres puntos de fuga

Perspectivas de tres puntos no tienen ningún eje paralelo a los ejes principales.



Matemáticas para proyecciones

Se presenta las matemáticas básicas para proyecciones geométricas planas. Cada una de las proyecciones se define como una matriz de 4x4, siendo esto conveniente ya que la matriz de proyección se puede componer con las matrices de transformación, permitiendo representar las operaciones de transformación y proyección como una sola matriz. Se considera que el plano de proyección es normal al eje de z, ubicado en z=d en la proyección perspectiva y en z=0 en la proyección paralela

Paralela Ortogonal

Una proyección paralela ortográfica, con el plano de proyección en z_{vp} se describe mediante la siguiente matriz de proyección.

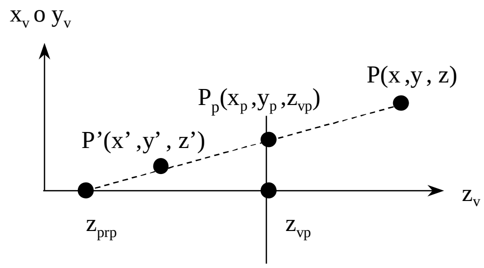
$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & z_{vp} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Si z_{vp} = 0 la matriz de proyección se daría por

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspectiva

Para calcular las transformaciones, establecemos el punto de referencia de proyección en la posición (centro de proyección) z_{prp} a lo largo del eje de z_v y situamos el plano de visión en z_{vp} como se muestra en la siguiente figura (COP corresponde al centro del lente y está delante del plano de proyección correspondiente a la película de la cámara).



Podemos expresar las posiciones de coordenadas (x_{prp}=0, y_{prp}=0)a lo largo del proyector en forma paramétrica como

$$x' = x_{prp} + (1-u)x = x - xu$$
$$y' = y_{prp} + (1-u)y = y - yu$$
$$z' = z_{prp} + (1-u)z = z - (z - z_{prp})u$$

El parámetro u toma valores de 0 a 1 y la posición de coordenadas (x', y', z') representa cualquier punto a lo largo del proyector. Cuando u = 0 corresponde a la posición P=(x, y, z). En el otro extremo de la línea, u = 1 y corresponde al punto de referencia de proyección (0,0, z_{prp}). En el plano de visión, $z' = z_{vp}$ y se puede despejar la ecuación de z para el parámetro de u en esta posición a lo largo del proyecto

$$u = (z - z_{vp})/(z - z_{prp})$$

Al sustituir este valor de u en las ecuaciones para x' y y', se obtiene las ecuaciones de transformación de perspectiva

$$x_p = x - x(z - z_{vp})/(z - z_{prp}) = x(z_{vp} - z_{prp})/(z - z_{prp}) = x(d_p/(z_{prp} - z))$$
$$y_p = y - y(z - z_{vp})/(z - z_{prp}) = y(z_{vp} - z_{prp})/(z - z_{prp}) = y(d_p/(z_{prp} - z))$$

donde $d_p = z_{prp} - z_{vp}$ es la distancia del plano de visión desde el punto de referencia de proyección (distancia entre el lente y la película).

Al utilizar una representación de coordenadas homogéneas tridimensionales, se puede expresar la transformación de proyección en perspectiva en forma de matriz (M_{per}) como

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} d_p/(z_{prp} - z) & 0 & 0 & 0 \\ 0 & d_p/(z_{prp} - z) & 0 & 0 \\ 0 & 0 & 0 & z_{vp} \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

La división por z describe el acortamiento uniforme; cuanto mas lejos los objetos del centro de proyección, mas pequeños se ven. Nótese, que según d_p tiende a infinito en la matriz de proyección de perspectiva, la matriz de proyección se convierte en la ortogonal paralela:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} d_p/(z_{prp} - z) & 0 & 0 & 0 \\ 0 & d_p/(z_{prp} - z) & 0 & 0 \\ 0 & 0 & 0 & z_{vp} \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

dado que la fracción $d_p/(z_{prp}-z)$ se vuelve ∞/∞.

La matriz de proyección de perspectiva se puede modificar de la siguiente manera:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}\left(z_{prp}/d_p\right) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

donde, el factor homogéneo es

$$h = (z_{\text{prp}} - z)/\, d_p = (z_{\text{prp}} - z)/(z_{prp} - z_{vp})$$

y las coordenadas de proyección en el plano de visión se calculan a partir de las coordenadas homogéneas como

$x_p = x_h /h$
$y_p = y_h /h$
$z_p = z_h /h$

El punto de referencia de proyección no tiene forzosamente que localizarse a lo largo del eje de z_v. Se puede seleccionar cualquier coordenada (x_{prp}, y_{prp}, z_{prp}) para el punto de referencia de proyección.

- En el caso que el punto de referencia de proyección esté situado en el origen de las coordenadas de vista, $z_{prp} = 0$ y $z_{vp} = d$,

  $$h = (z_{\text{prp}} - z)/\, d_p = (z_{\text{prp}} - z)/(z_{prp} - z_{vp}) = z/d$$

  $x_p = x(d / z) = x/ z / d$
  $y_p = y(d / z) = y/ z / d$

  La matriz de proyección seria:

  $$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

  donde

  $$[x_h \;\; y_h \;\; z_h \;\; h]^T = [x \;\; y \;\; z \;\; z/d]^T$$

  Dividiendo por $h = z/d$ y sacando la cuarta coordenada para volver a 3D, tenemos:

  $$(x_h/h,\, y_h/h,\, z_h/h) = (x_p,\, y_p,\, z_p) = (x/z/d,\, y/z/d,\, d)$$

- Si $z_{prp} = -d$ y $z_{vp} = 0$ entonces

  $$h = (z_{\text{prp}} - z)/\, d_p = (z_{\text{prp}} - z)/(z_{prp} - z_{vp}) = (-d-z)/-d = (d+z)/d$$

  $x_p = x(d/(d+z)) = x/(\, z/d +1)$
  $y_p = y(d/(d+z)) = y/(\, z/d +1)$

  Y la matriz sería

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

donde

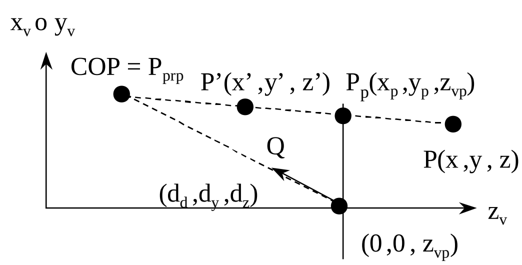
$$[x_h \ y_h \ z_h \ h]^T = [x \ y \ 0 \ z/d+1]^T$$

Dividiendo por $h = z/d+1$ y sacando la cuarta coordenada para volver a 3D, tenemos:

$$(x_h/h, y_h/h, z_h/h) = (x_p, y_p, z_p) = (x/z/d, y/z/d, 0)$$

Matriz de Proyección General

Para una generalización, donde la dirección de proyección no tiene que ser a lo largo del eje de z, se integra ambas matrices de transformación en solo una, a partir de la siguiente figura.



plano de visión

Se describe la proyección de un punto general $P=(x,y,z)$ sobre un plano de proyección $P_p = (x_p, y_p, z_p)$, donde el plano de proyección es perpendicular al eje de z y está a una distancia zv_p del origen, y el centro de proyección (COP) está a una distancia Q del punto $(0, 0, z_{vp})$. La dirección de $(0, 0, z_{vp})$ a COP se da por el vector de dirección normalizado (d_x, d_y, d_z). P_p está sobre la línea COP - P, que puede especificarse de manera paramétrica como

$$P_p = \text{COP} + u(P - \text{COP}), \quad 0 \le u \le 1$$

Esta ecuación se puede escribir como ecuaciones separadas sobre la línea para un punto arbitrario $P'=(x',y',z')$, con COP = $(0, 0, z_{vp}) + Q(d_x, d_y, d_z)$, dando

$$x' = Q\,d_x + (x - Q\,d_x)\,u$$
$$y' = Q\,d_y + (y - Q\,d_y)\,u$$
$$z' = (z_{vp} + Q\,d_z) + (z - (z_{vp} + Q\,d_z))\,u$$

Se define la proyección P_p del punto P, en la intersección de la línea entre COP y **P** con el plano de proyección, sustituyendo $z' = z_{vp}$ en la ecuación anterior

$$u = (z_{vp} - (z_{vp} + Q\,d_z)) / (z - (z_{vp} + Q\,d_z))$$

Sustituyendo el valor de u obtenemos

$$x_p = Q\,d_x + (x - Q\,d_x)\,(z_{vp} - (z_{vp} + Q\,d_z)) / (z - (z_{vp} + Q\,d_z))$$
$$y_p = Q\,d_y + (y - Q\,d_y)\,(z_{vp} - (z_{vp} + Q\,d_z)) / (z - (z_{vp} + Q\,d_z))$$

o

$$x_p = \frac{x - z\dfrac{d_x}{d_z} + z_{vp}\dfrac{d_x}{d_z}}{\dfrac{z_{vp} - z}{Qd_z} + 1} \qquad y_p = \frac{y - z\dfrac{d_y}{d_z} + z_{vp}\dfrac{d_y}{d_z}}{\dfrac{z_{vp} - z}{Qd_z} + 1}$$

Multiplicando por la identidad $z_p = zv_p$ se obtiene

$$z_p = z_{vp}\frac{\dfrac{z_{vp} - z}{Qd_z} + 1}{\dfrac{z_{vp} - z}{Qd_z} + 1} = \frac{-z\dfrac{z_{vp}}{Qd_z} + \dfrac{z_{vp}^2 + z_{vp}Qd_z}{Qd_z}}{\dfrac{z_{vp} - z}{Qd_z} + 1}$$

Estas ecuaciones pueden escribirse como una matriz general de proyección organizada para que la ultima fila produzca el denominador común, el cual es la coordenada homogénea:

$$M_{gen} = \begin{bmatrix} 1 & 0 & -d_x/d_z & z_{vp}d_x/d_z \\ 0 & 1 & -d_y/d_z & z_{vp}d_y/d_z \\ 0 & 0 & -z_{vp}/Qd_z & z_{vp}^2/Qd_z + z_{vp} \\ 0 & 0 & -1/Qd_z & z_{vp}/Qd_z + 1 \end{bmatrix}$$

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_x/d_z & z_{vp}d_x/d_z \\ 0 & 1 & -d_y/d_z & z_{vp}d_y/d_z \\ 0 & 0 & -z_{vp}/Qd_z & z_{vp}^2/Qd_z + z_{vp} \\ 0 & 0 & -1/Qd_z & z_{vp}/Qd_z + 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

donde

$$h = \frac{z_{vp} - z}{Qd_z} + 1$$

La matriz general se especializa en las matrices de proyección previamente derivadas:

Proyección	z_{vp}	**Q**	**[d_x d_y d_z]**
Ortogonal	0	∞	[0 0 -1]
Perspectiva	*d*	*d*	[0 0 -1]
Perspectiva	0	*d*	[0 0 -1]
Caballera	0	∞	[cos a sin a -1]
Gabinete	0	∞	[(cos a)/2 (sin a)/2 -1]

En todos estos casos el plano de proyección es perpendicular al eje de z.

- **Ortogonal -** El caso de la matriz ortogonal se obtiene

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

- **Perspectiva -** En una proyección perspectiva, $Q \neq \infty$, y la distancia entre el centro de proyección y el plano de proyección es $d = z_{vp} - z_{prp}$.

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & z_{vp}/d & -z_{vp}^2/d + z_{vp} \\ 0 & 0 & 1/d & -z_{vp}/d + 1 \end{bmatrix}$$

El punto de fuga de una proyección de perspectiva se calcula multiplicando el punto en el infinito sobre el eje de z, representado en coordenadas homogéneas como $[0\ \ 0\ \ 1\ \ 0]^T$, por M_{ger}.

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_x/d_z & z_{vp} d_x/d_z \\ 0 & 1 & -d_y/d_z & z_{vp} d_y/d_z \\ 0 & 0 & -z_{vp}/Qd_z & z_{vp}^2/Qd_z + z_{vp} \\ 0 & 0 & -1/Qd_z & z_{vp}/Qd_z + 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -d_x/d_z \\ -d_y/d_z \\ -z_{vp}/Qd_z \\ -1/Qd_z \end{bmatrix}$$

Tomando este producto y dividiendo por h se obtiene

$x = x_h/h = -d_x/d_z\ (-Q\ d_z) = Q\ d_x$
$y = y_h/h = -d_y/d_z\ (-Q\ d_z) = Q\ d_y$
$z = -z_{vp}/Qd_z\ (-Q\ d_z) = z_{vp}$

Escogiendo un punto de fuga particular (x,y), y dado que se conoce la distancia Q al centro de la proyección, estas ecuaciones definen de forma única $[\ d_x\ d_y\ d_z\ ]$, ya que $\sqrt{d_x^2 + d_y^2 + d_z^2} = 1$.

- **Caballera -** Una proyección caballera sobre el plano xy, con un ángulo a, resulta en

$$M_{cab} = \begin{bmatrix} 1 & 0 & -\cos a/-1 & 0 \\ 0 & 1 & -\sin a/-1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cos a & 0 \\ 0 & 1 & \sin a & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Gabienete -** Una proyección caballera sobre el plano xy, con un ángulo a, resulta en

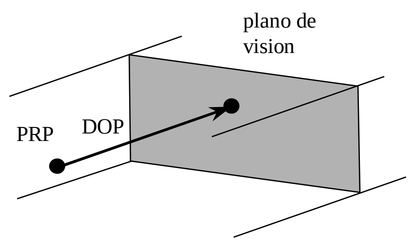
$$M_{gab} = \begin{bmatrix} 1 & 0 & -\cos a/2/-1 & 0 \\ 0 & 1 & -\sin a/2/-1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cos a/2 & 0 \\ 0 & 1 & \sin a/2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Volúmenes de Vista

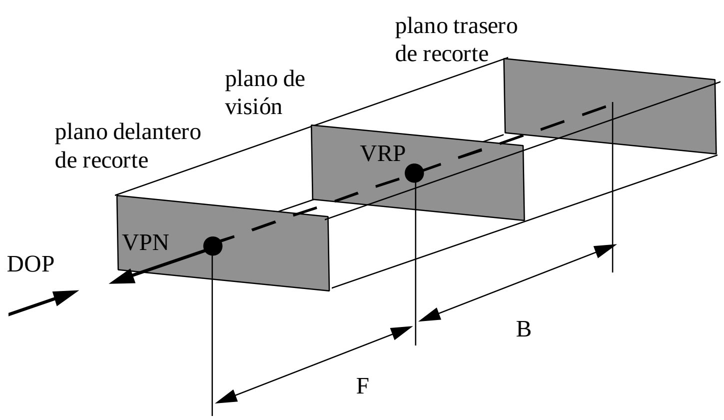
En analogía a la cámara, el tipo de lente que se utiliza es un factor que determina el porcentaje de la escena que se capta en la película, el **ángulo de vista**. Un lente gran-angular capta una parte mas amplia de la escena que un lente regular. Se establecer un **volumen de vista** (**view volume**) que especifica que objetos en la escena aparecerán y cuales serán recortados fuera de ella. Solo aquellos objetos en el volumen de vista aparecerán en le despliegue, limitando la porción del mundo que se recortara y proyectara sobre el plano de vista.

Paralela

Para proyecciones paralelas, el volumen de vista es un paralelepipedo infinito con los lados paralelos a la dirección de proyección.



Se obtiene un volumen de vista finito al limitar la extensión del volumen en la dirección de *n*, mediante uno o dos planos de fronteras adicionales. Se limita el volumen de vista con un **plano de recorte de frontal** (**Front Clipping Plane**) y un **plano de recorte posterior** (**Back Clipping Plane**), también llamados plano cercano (**Hither**) y lejano (**Yon**), que son planos paralelos al plano de vista en posiciones especificas z_{front} y z_{back}; normales al plano de visión. Ambos planos deben situarse en el mismo lado del punto de referencia de proyección y el plano posterior debe estar mas distante del punto de proyección que el plano frontal.

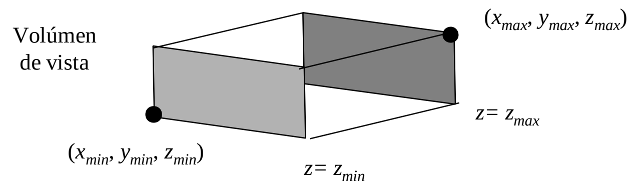


Los planos se especifican por las **distancias de frente**, positivas o negativas, (*F* **- Front distance**) y la **distancia de atrás** (*B* **- Back distance**) relativa a VRP a lo largo de VPN (normal al plano), con distancias positivas en la dirección de VPN.

En OpenGL, la única función paralela de vista ofrecida es la ortográfica:

```
glOrtho(xmin, xmax, ymin, ymax, zmin, zmax);
```

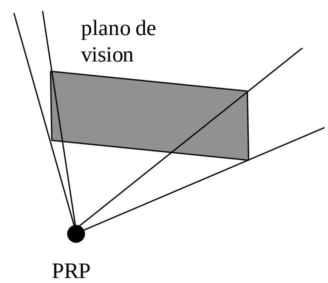
Los parámetros están definidos en la siguiente figura:



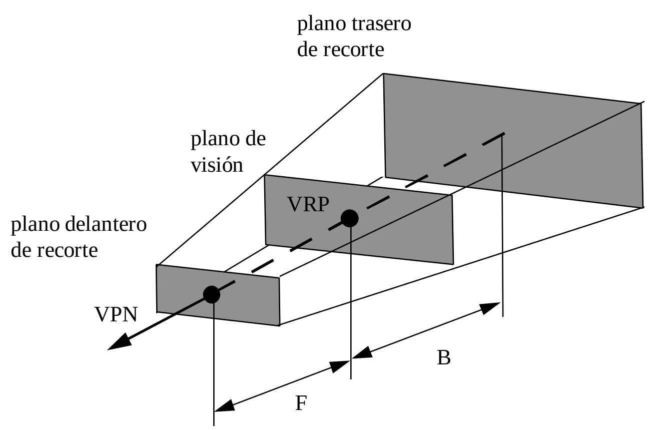
Los planos delanteros y traseros deben ser paralelos al plano z=0 (aunque no hay restricción en que sean sus distancias positivas exclusivamente). Como la cámara apunta en la dirección negativa de z, los planos de recorte delantero y trasero están en $z=z_{min}$ y $z=z_{max}$, respectivamente.

Perspectiva

Para una proyección perspectiva el volumen de vista es la pirámide semi-infinita con el ápice en PRP y los aristas pasando por las esquinas de la ventana.



Las posiciones detrás del centro de proyección no se incluyen en el volumen de vista y por lo tanto no se proyectan. En realidad, nuestros ojos ven un volumen de vista cónico; sin embargo, un volumen de vista piramidal es mas fácil de analizar matemáticamente, y es consistente con el concepto de un puerto de vista rectangular. La siguiente figura muestra la pirámide truncada (**frustum -** tronco).

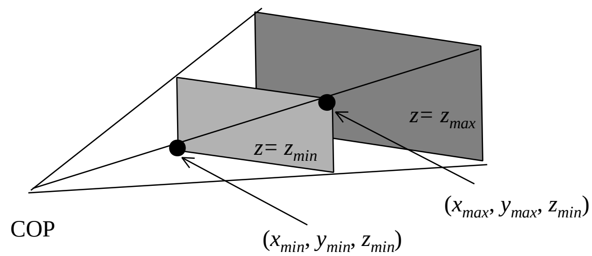


Limitando de esta forma el volumen de vista puede ser útil para eliminar objetos ajenos, permitiendo concentrar en los objetos necesarios. Un objeto muy distante del centro de proyección se proyecta sobre la superficie de vista como una mancha sin forma distinguible. También, un objeto muy cercano al centro de proyección pudiera extenderse a través de toda la ventana, sin una estructura discernible. Especificando un volumen de vista puede aparentemente eliminar tales problemas.

Existen dos funciones en OpenGL para especificar vista perspectiva, directamente al cargar la matriz de proyección, o aplicando transformaciones básicas. La analogía a la vista de la cámara es la función

```
glFrustum(xmin, xmax, ymin, ymax, zmin, zmax);
```

Los parámetros se muestran en la siguiente figura:



Las distancias de los planos delanteros y traseros deben ser positivos y medidos del COP a estos planos, los cuales deben ser paralelos al plano *z*=0. Como la cámara apunta en la dirección negativa de *z*, los planos de recorte delantero y trasero están en $z=z_{min}$ y $z=z_{max}$, respectivamente. Esta matriz multiplica a la matriz actual, por lo cual debe primero seleccionarse el modo. Una secuencia típica es:

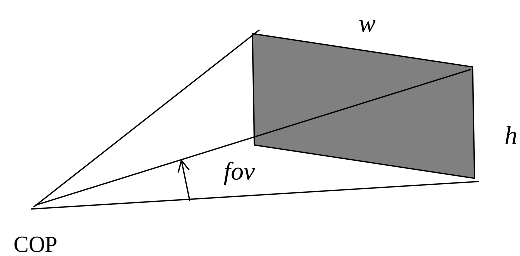
```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(xmin, xmax, ymin, ymax, zmin, zmax);
```

El tronco resultante no tiene que ser simétrico con respecto al eje de *z*.

En muchas aplicaciones es natural especificar el ángulo o campo de vista. Sin embargo, si el plano de proyección es perpendicular, en lugar de cuadrado, se verían dos ángulos de vista diferentes, desde arriba o de lado. En OpenGL, la función es:

```
gluPerspective(fovy, aspect, near, far);
```

la cual permite especficar el campo de vista en la dirección *y* (*vup*) y la relación de aspecto, ancho dividido por altura, del plano de proyección, como se muestra en la siguiente figura:



Los planos *near* y *far* se especifican como en `glFrustum`. Esta matriz también altera la matriz actual, por lo cual debe seleccionarse el modo de matriz apropiado, y reinicializarse si es necesario. (El aspecto es *w*/*h*.)

Volumen de Vista Canónico

Los **volúmenes de vista canónicos** se definen para simplificar las ecuaciones de recorte y para proveer consistencia entre proyecciones paralelas y en perspectiva.

Paralela

Para proyecciones ortográficas el volumen de vista de recorte más sencillo es un cubo centrado en el origen con los lados definidos por lo seis planos:

$x = -1, x = 1, y = -1, y = 1, z = -1, z = 1,$

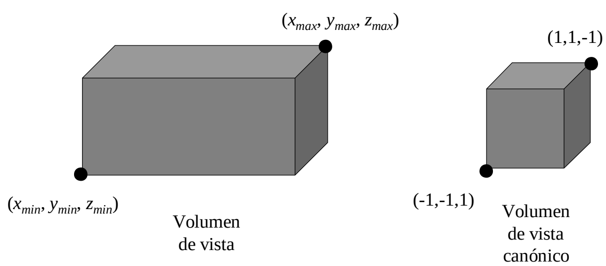
Este es el volumen de vista canónico en OpenGL, que de manera explícita, se pudiera especificar mediante:

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
```

Si se especifica un volumen de vista arbitrario:

```
glOrtho(xmin,xmax,ymin,ymax,zmin,zmax);
```

la matriz de proyección que OpenGL genera convierte automáticamente los vértices especificados en el programa, a vértices dentro del volumen canónico, mediante escala y traslaciones, como se muestra en la siguiente figura:



La transformación de normalización N_{par} se deriva de manera general. incluyendo una transformación shear que causa que la dirección de proyección en coordenadas de vista sea paralela a z, en el caso de proyecciones oblicuas. Al incluir este recorte (shear), se puede proyectar sobre el plano z=0 simplemente haciendo que todas las z se igualen a 0. Si la proyección paralela es ortográfica, el componente de recorte (shear) de la transformación de normalización se vuelve la identidad.

La serie de transformaciones que hace N_{par} es la siguiente

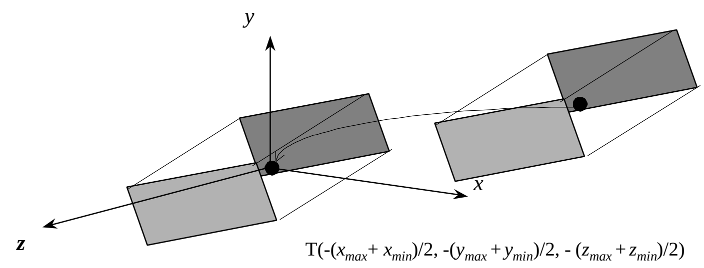
1. Trasladar el centro del volumen al origen.
2. Rotar VRC de manera que el eje n (VPN) sea el eje z, el eje u sea el eje x, y el eje v sea el eje y.
3. Recortar (shear) de manera que la dirección de proyección sea paralela al eje z.
4. Trasladar y escalar al volumen de vista canónico de proyección paralela.

- **Paso 1**

Se mueve el centro del volumen de vista especificado al centro del volumen de vista canónico (el origen) mediante una traslación. Se define la traslación T como

$$\mathbf{T}(-(x_{max}+x_{min})/2, -(y_{max}+y_{min})/2, -(z_{max}+z_{min})/2)$$

El resultado de esto se muestra en la siguiente figura:



- **Paso 2**

Se usa la propiedad de matrices ortogonales especiales. Los vectores fila de la matriz de rotación son los vectores unidad que se rotan por R para orientarse como los ejes x, y, y z. VPN se rota para orientarse como el eje z

$$R_z = \text{VPN}/|\text{VPN}|$$

El eje u, el cual es perpendicular a VUP y VPN y es por lo tanto el producto cruz del vector unidad a lo largo de VUP y R_z (que tiene la misma dirección que VPN), se rota para orientarse en el eje x

$$R_x = \text{VUP}\times R_z / |\text{VUP}\times R_z|$$

De forma similar, el eje v, el cual es perpendicular a R_x y R_z, se rota para orientarse en el eje y

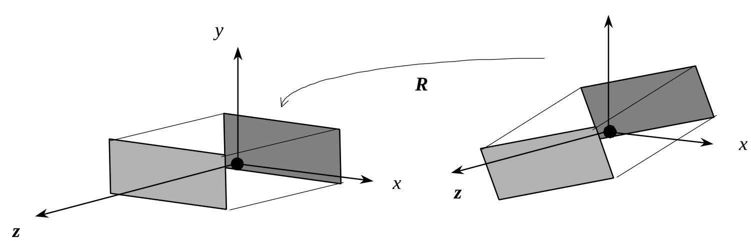
$$R_y = R_x \times R_z$$

Por lo tanto, la matriz de rotación es

$$R = \begin{bmatrix} r_{1x} & r_{2x} & r_{3x} & 0 \\ r_{1y} & r_{2y} & r_{3y} & 0 \\ r_{1z} & r_{2z} & r_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde r_{1x} es el primer elemento de R_x, etc.

El resultado de esto se muestra en la siguiente figura:

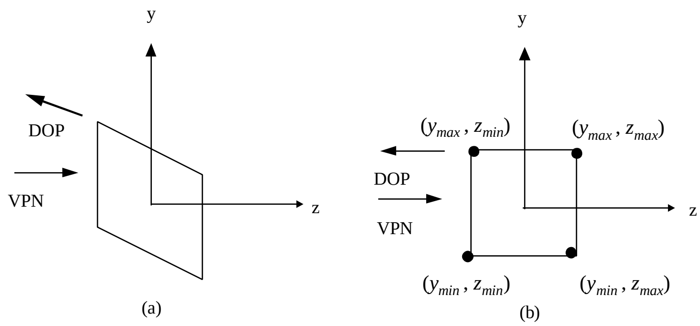


- **Paso 3**

El volumen de vista para una proyección oblicua tiene los planos de recorte cercano y lejano paralelos al plano de vista, y los planos de recorte derecho, izquierdo, superior e inferior, paralelos a la dirección de proyección (DOP). Por lo tanto, se debe recortar (shear) el volumen de vista a lo largo del eje z de manera que todos sus planos sean normales a uno de los ejes principales de coordenadas. Esto se hace determinando el recorte (shear) a ser aplicado a la dirección de proyección y hacer a DOP coincidente con el eje z, donde

$$DOP = \begin{bmatrix} dop_x \\ dop_y \\ dop_z \\ 0 \end{bmatrix}$$

La siguiente figura muestra el DOP especificado, y el DOP' transformado



El recorte (shear) puede lograrse con la matriz de recorte (x,y), con los coeficientes shx_{par} y shy_{par},

$$SH_{par} = SH_{xy}\,(shx_{par}, shy_{par}) = \begin{bmatrix} 1 & 0 & shx_{par} & 0 \\ 0 & 1 & shy_{par} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Este recorte no afecta a z, mientras se le agrega los términos $z \cdot shx_{par}$ y $z \cdot shy_{par}$ a x y y, respectivamente.
Se quiere obtener shx_{par} y shy_{par} de manera que

$$DOP' = [0\ 0\ dop_z\ 0]^T = SH_{par} \cdot DOP$$

Haciendo las multiplicaciones de la ecuación anterior seguida por manipulaciones algebraicas muestra que la igualdad ocurre si

$$dop'_x = 0 = dop_x + shx_{par}\, dop_z$$
$$dop'_y = 0 = dop_y + shy_{par}\, dop_z$$

por lo cual los parámetros de recorte son

$$shx_{par} = -dop_x / dop_z, \qquad shy_{par} = -dop_y / dop_z$$

Nótese que, para una proyección ortogonal, $dop_x = dop_y = 0$, por lo cual $shx_{par} = shy_{par} = 0$, y la matriz de recorte se reduce a la identidad. El resultado es

$$SH_{par} = \begin{bmatrix} 1 & 0 & -dop_x/dop_z & 0 \\ 0 & 1 & -dop_y/dop_z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los límites del volumen de vista después de haber aplicado estas tres transformaciones son

$x_{min} \le x \le x_{max}$, $y_{min} \le y \le y_{max}$, $z_{min} \le z \le z_{max}$,

- **Paso 4**

Finalmente se transforma el volumen de vista recortado (sheared) en el volumen de vista canónico. Se logra esto trasladando el centro frontal del volumen de vista, de la ecuación anterior, al origen, luego escalando al tamaño 2x2x2 del volumen de vista canónico. Las transformaciones son

$$T_{par} = T(-(x_{max}+x_{min})/2,\ -(y_{max}+y_{min})/2,\ -(z_{max} + z_{min})/2)$$

$$S_{par} = S(2/(x_{max} - x_{min}),\ 2/(y_{max} - y_{min}),\ 2/(z_{max} - z_{min}))$$

En resumen,

$$N_{par} = S_{par} \cdot T_{par} \cdot SH_{par} \cdot R \cdot T$$

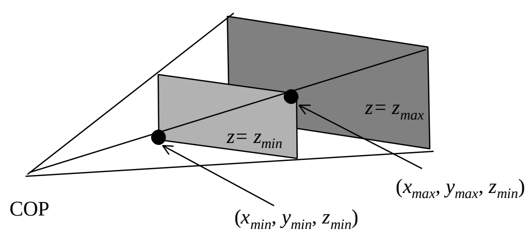
N_{par} transforma un volumen de vista arbitrario de proyección paralela a un volumen de vista canónico de proyección paralela, y por lo tanto permite que las primitivas de salida se recorten contra un volumen de vista canónico de proyección paralela.

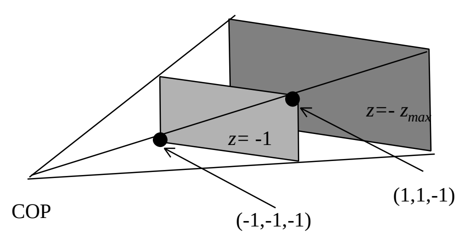
Perspectiva

La transformación de normalización para proyecciones perspectivas resulta en un volumen de vista canónico, teniendo como forma la pirámide truncada con ápice en el origen definido por

$$x = z,\ x = -z,\ y = z,\ y = -z,\ z = -1,\ z = z_{max}$$

Un volumen de vista general se transforma en el volumen de vista canónico, como se muestra en la siguiente figura:





La transformación de normalización N_{per} se deriva para transformar el volumen de vista en el volumen de vista canónico. Después de aplicar N_{per}, se recorta contra este volumen canónico y los resultados se proyectan sobre el plano de vista usando M_{per}.

La serie de transformaciones que hace N_{per} es la siguiente

1. Trasladar el centro de la ventana al origen.

2. Rotar VRC de manera que el eje *n* (VPN) sea el eje *z*, el eje *u* sea el eje *x*, y el eje *v* sea el eje *y*.
3. Trasladar de manera que el centro de proyección (COP), dado por PRP, este en el origen.
4. Recortar (shear) de manera que la línea central del volumen de vista sea el eje *z*.
5. Escalar de manera que el volumen de vista sea el volumen de vista canónico de perspectiva, la pirámide derecha truncada definida por los seis planos.

- **Pasos 1 y 2**

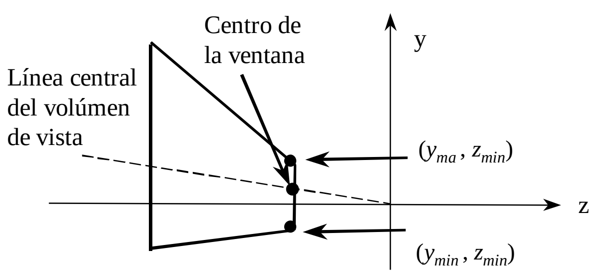
Son los mismo que la proyección paralela: $R \cdot T$.

- **Paso 3**

Se hace una traslación del centro de proyección (COP o PRP) al origen, como se requiere para el volumen de vista canónico de perspectiva. La traslación es simplemente T(-PRP), donde PRP = (prp_u, prp_v, prp_n).

- **Paso 4**

Se convierte el frustum arbitrario a un frustum simétrico con lados de 45 grados. El proceso es similar a la conversión de una vista paralela oblicua a una vista ortogonal. Primero se hace un recorte (shear) para convertir el frustum asimétrico a uno simétrico. Para computar el recorte (shear) se examina la siguiente figura la cual muestra una vista de lado del volumen de vista después de las transformaciones 1 a 3.



Nótese que la línea central del volumen de vista, la cual va del origen (COP) al centro de la ventana, no es la misma que el eje -*z*, debiéndose recortar (shear) para transformar el centro de la línea en el eje -*z*.

La matriz de recorte (shear) es SH_{par}, la misma que para la proyección paralela.

$$SH_{\text{par}} = SH_{xy}\,(shx_{\text{par}}, shy_{\text{par}}) = \begin{bmatrix} 1 & 0 & shx_{par} & 0 \\ 0 & 1 & shy_{par} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Este recorte no afecta a *z*, mientras se le agrega los términos $z \cdot shx_{\text{par}}$ y $z \cdot shy_{\text{par}}$ a *x* y *y*, respectivamente.
Se quiere obtener shx_{par} y shy_{par} de manera que

$$\text{DOP'} = [0\ 0\ dop_z\ 0]^{\text{T}} = SH_{\text{par}} \cdot \text{DOP}$$

El ángulo de recorte (shear) se determina según la correspondencia del punto $((x_{\min}+x_{\max})/2,\ (y_{\min}+y_{\max})/2,\ z_{\min})$ a $(0,0,\ z_{\min})$.

$$dop_x = (x_{\min}+x_{\max})/2 - 0$$
$$dop_y = (y_{\min}+y_{\max})/2 - 0$$

$dop_z = z_{\min} - 0$

Haciendo las multiplicaciones de la ecuación anterior seguida por manipulaciones algebraicas muestra que la igualdad ocurre si

$dop'_x = 0 = dop_x + shx_{\mathrm{par}}\, dop_z$
$dop'_y = 0 = dop_y + shy_{\mathrm{par}}\, dop_z$

por lo cual los parámetros de recorte son

$shx_{\mathrm{par}} = -dop_x / dop_z = -(x_{\min}+x_{\max})/2\ z_{\min}$
$shy_{\mathrm{par}} = -dop_y / dop_z = -(y_{\min}+y_{\max})/2\ z_{\min}$

El recorte (shear) requerido es

$\mathbf{SH}_{\mathrm{par}} = \mathbf{SH}_{xy}(-(x_{\min}+x_{\max})/2z_{\min}, -(y_{\min}+y_{\max})/2z_{\min})$

El resultado es

$$SH_{\mathrm{par}} = \begin{bmatrix} 1 & 0 & -dop_x/dop_z & 0 \\ 0 & 1 & -dop_y/dop_z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(x_{\min}+x_{\max})/2z_{\min} & 0 \\ 0 & 1 & -(y_{\min}+y_{\max})/2z_{\min} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los puntos resultantes se calculan de la siguiente forma:

$$\begin{bmatrix} 1 & 0 & -(x_{\min}+x_{\max})/2z_{\min} & 0 \\ 0 & 1 & -(y_{\min}+y_{\max})/2z_{\min} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\min} \\ y_{\min} \\ z_{\min} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{\min}-(x_{\min}+x_{\max})/2 \\ y_{\min}-(y_{\min}+y_{\max})/2 \\ z_{\min} \\ 1 \end{bmatrix} = \begin{bmatrix} (x_{\min}-x_{\max})/2 \\ (y_{\min}-y_{\max})/2 \\ z_{\min} \\ 1 \end{bmatrix}$$

El frustum resultante está descrito por los planos

$x = \pm(x_{\max}-x_{\min})/2\ z_{\min}$
$y = \pm(y_{\max}-y_{\min})/2\ z_{\min}$
$z = z_{\min}$
$z = z_{\max}$

Después de aplicar el recorte (shear), el volumen de vista está centrado sobre el eje z.

- **Paso 5**

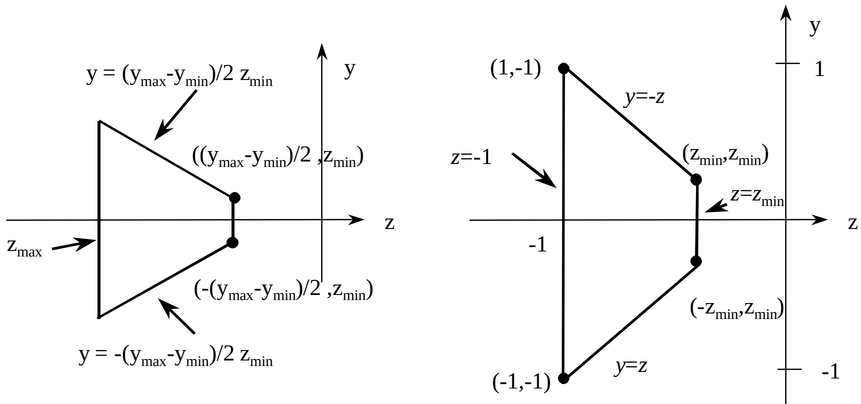
El siguiente paso es escalar los lados del frustum resultante anterior a

$x = \pm z$
$y = \pm z$

y el plano cercano a

$z_{\min} = -1$

El volumen de vista canónico, se muestra en la siguiente figura.



(a) antes de escalar (b) después de escalar

La matriz de escala requerida es

$$\mathbf{S}(2\,z_{min}/(x_{max}-x_{min}),\ 2\,z_{min}/(y_{max}-y_{min}),\ -1/\,z_{min})$$

Nótese que esta transformación se determina de manera única sin referencia a la ubicación del plano lejano $z = z_{max}$, ya que en tres dimensiones una transformación afín se determina por los resultados de la transformación sobre cuatro puntos. En este caso, los puntos son los cuatro vértices donde los lados del frustum intersectan el plano cercano.

En resumen, la transformación de vista normalizante que toma el volumen de vista de proyección perspectiva al volumen de vista canónico de proyección de perspectiva es

$$N_{per} = S_{per} \cdot SH_{per} \cdot T(\text{-PRP}) \cdot R \cdot T(\text{-VRP})$$

Similarmente, recuérdese la transformación de vista normalizante que toma el volumen de vista de proyección paralela al volumen de vista canónico de proyección paralela es

$$N_{par} = S_{par} \cdot T_{par} \cdot SH_{par} \cdot R \cdot T(\text{-VRP})$$

Estas transformaciones ocurren en espacio homogéneo.

Vista del Cubo

El cubo del capítulo anterior se definió rotando en el origen utilizando una proyección ortográfica. Se extiende el ejemplo con vista de perspectiva y se permite al observador mover la cámara al presionar *x*, *X*, *y*, *Y*, *z* y *Z* en el teclado, aunque la cámara siempre apunta al centro del cubo. La función `gluLookAt` provee una manera sencilla de reposicionar y reorientar la cámara.

Los cambios sobre el programa previo son menores. Se define un arreglo `viewer[3]` para guardar la posición de la cámara. Su contenido se altera mediante la función callback de teclado `keys`:

```
void keys(unsigned char key, int x, int y)
{
      if(key == 'x') viewer[0]-= 1.0;
      if(key == 'X') viewer[0]+= 1.0;
      if(key == 'y') viewer[1]-= 1.0;
      if(key == 'Y') viewer[1]+= 1.0;
      if(key == 'z') viewer[2]-= 1.0;
```

```
        if(key == 'Z') viewer[2]+= 1.0;
        display();
    }
```

La función de despliegue llama a `LookAt` usando `viewer` para la posición de la cámara y usa el origen para la posición "at". El cubo se rota, como antes, según la entrada del ratón. Nótese que el orden de las llamadas de función en `display` que alteran la matriz modelo-vista:

```
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(viewer[0],viewer[1],viewer[2],
        0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    glRotatef(theta[0], 1.0, 0.0, 0.0);
    glRotatef(theta[1], 0.0, 1.0, 0.0);
    glRotatef(theta[2], 0.0, 0.0, 1.0);
    colorcube();
    glFlush();
    glutSwapBuffers();
}
```

Se usa el callback `reshape` para especificar el lente de la cámara mediante `glFrustum`:

```
void myReshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    if(w<=h) glFrustum(-2.0, 2.0, -2.0 * (GLfloat) h/ (GLfloat) w,
        2.0* (GLfloat) h / (GLfloat) w, 2.0, 20.0);
    else glFrustum(-2.0, 2.0, -2.0 * (GLfloat) w/ (GLfloat) h,
        2.0* (GLfloat) w / (GLfloat) h, 2.0, 20.0);
    glMatrixMode(GL_MODELVIEW);
}
```

Fuera de la especificación adicional del callback de teclado en `main`, el resto del programa es el mismo que el anterior. Se debe notar el efecto de mover la cámara, el lente, y los lados de la columna de vista.

Tubería de Vista

La tubería de vista completa se muestra a continuación