Ampliación de Informática Gráfica

Tema 3. Sistemas de Visualización Tiempo-Real. Hardware Gráfico.

Índice del Tema.

- Objetivos y Restricciones para la generación Gráfica 3D. Necesidades Básicas
- Modelo de Procesamiento para gráficos 3D basados en representación poligonal: Tubería gráfica de Polígonos.
- Tubería Gráfica de Vertices:
  - Transformaciones del modelo
  - Transformaciones de la Vista
  - Transfomaciones de Recorte y Proyección
  - Transformaciones de Ventana.
- Tubería Gráfica de Pixels.
  - Recorte o Clipping.
  - Z-Buffer
  - Rellenado.

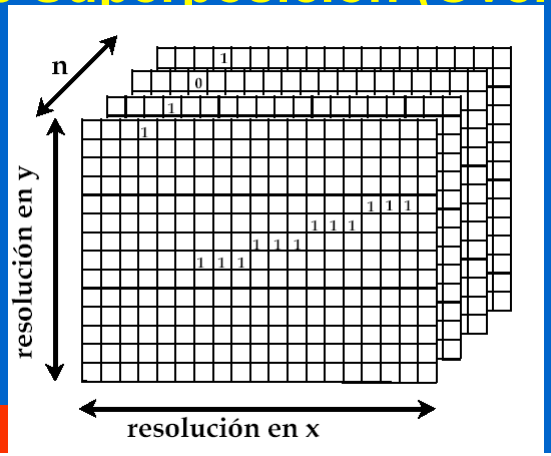
Generación Gráfica Tiempo Real.

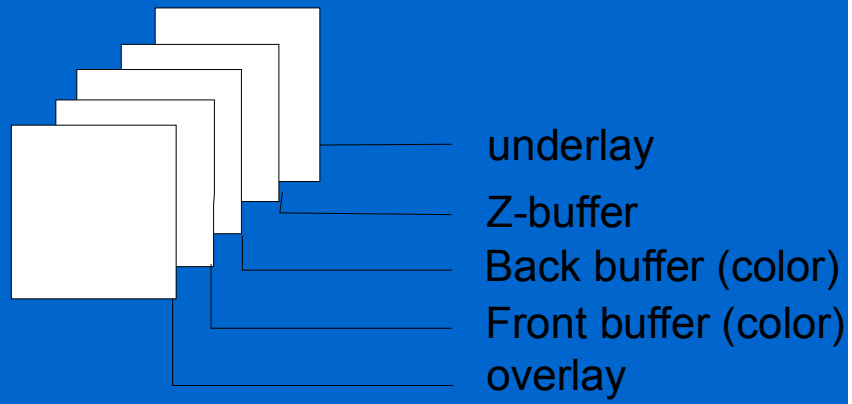
- ***Sistema informático en tiempo real*** **aquel que tiene ciertas restricciones de comportamiento en el tiempo.**
- **En particular, un sistema de síntesis de gráficos 3D en tiempo real, será un sistema que pueda redibujar la escena en un tiempo suficientemente corto**
- **Las características que se pueden destacar en este tipo de sistemas son :**
  - **Trabajan con primitivas poligonales**
  - **Minimizar la cantidad de polígono, permitiendo realismo media la incorporación del pegado de texturas de color**
  - **Linealización de los cálculos. Mayor facilidad para implementación Hardware.**
  - **Eliminación de Superficies ocultas y procesamiento superfluo.**
  - **Memorias especiales para apoyar los algoritmos básicos de rellenado y el procesamiento a nivel de pixels.**

Generación Gráfica Tiempo Real.

Elementos de Memoria Necesarios para la generación Gráfica

- **Memoria de Color: 24 bits (R,G,B)+8 bits (A)**
- **Duplicación de la Memoria de Color: Modo Doble Buffer.**
- **Memoria de Z-buffer: 16-32 bits.**
- **Memoria de Acumulación: 32 bits**
  - **Efectos atmosféricos.**
  - **Funciones de Mezcla.**
  - **Antialiasing.**
- **Planos de Superposición (Overlay y Underlay):16 bits.**





Generación Gráfica Tiempo Real.

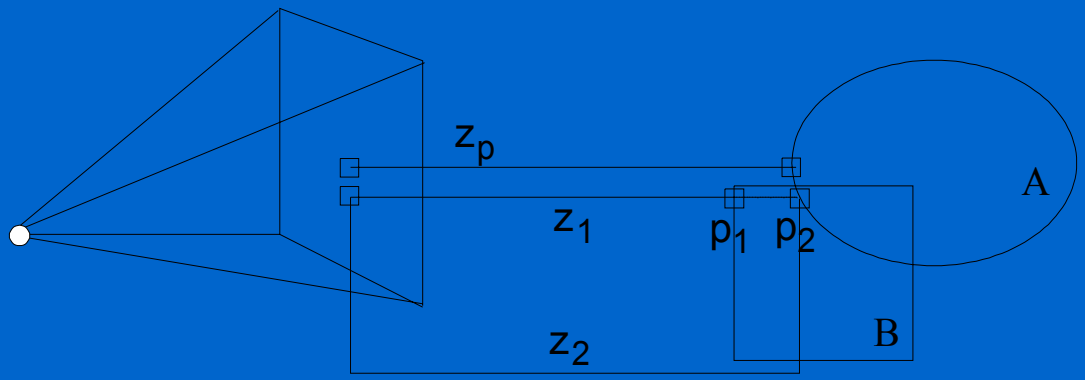
Eliminación de Partes Ocultas: Algoritmo de Z-Buffer.

- **Algoritmo de espacio Imagen.**
- **No necesita orden especial de procesado de la escena.**
- **Aplicable a distintas modelos de representación de escenas.**
- **Memoria adicional necesaria. Implementación hardware sencilla.**
- **Problemas de dentado o *aliasing*.**
- **Perdida de precisión con la distancia.**
- **Lo podemos emplear para mezclar imágenes ya que la información no se pierde.**

Generación Gráfica Tiempo Real.

Algoritmo de Z-Buffer.

```
Inicializar imagen a color fondo
Inicializar zbuffer a z de máximo alejamiento
Para cada polígono
    para cada pixel en la proyección del polígono
    z:=z(x,y)
    si z más cercana que zbuffer(x,y)
            zbuffer(x,y):=z
            escribir pixel (x,y) al color correspondiente
    fin si
    fin para
Fin para
Fin Zbuffer
```

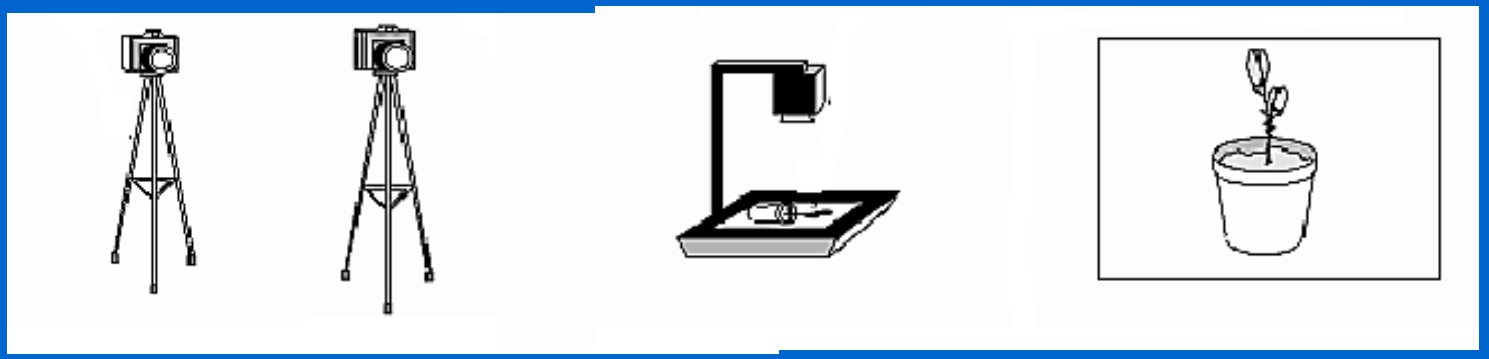


Generación Gráfica Tiempo Real.

- Estructura de procesamiento basada en tubería de polígonos:
  - Procesamiento de vértices. Trabajo realizado en coordenadas 3D.
  - Procesamiento de *pixels*. Trabajos realizados en coordenadas de imagen.

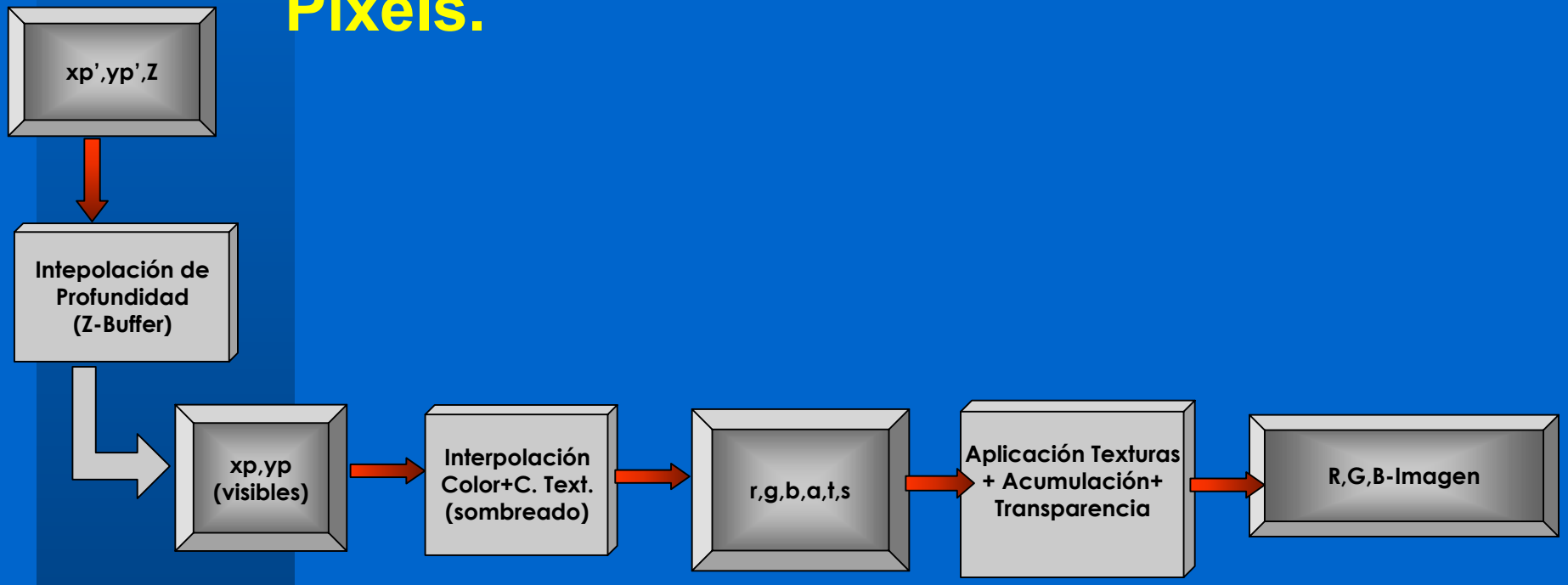
Generación Gráfica Tiempo Real.

Componentes Principales de la Tubería de Vértices.



Generación Gráfica Tiempo Real.

Componentes Principales de la Tubería de Pixels.



Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Colocar los objetos en su posición final en la escena.**
- **Transformar los valore de los vértices y la normales.**
- **Expresión que permita linealizarlas.**
- **Descomponer las transformación general en partes elementales.**
- **Representación homogénea de todas las transformaciones.**
- **Transformaciones afines:**

$$f : R^2 \Rightarrow R^2$$

$$f(u+v) = f(u) + f(v)$$

$$f(\lambda u) = \lambda f(u)$$

Etapas de la *pipeline* de vértices.

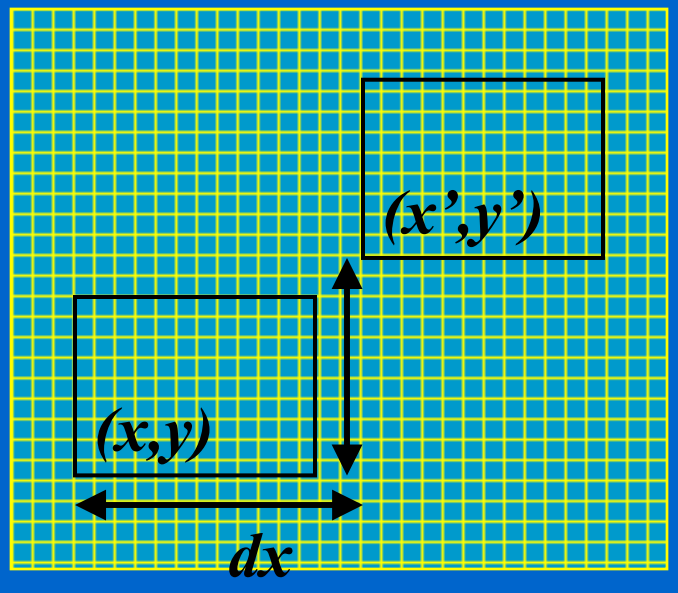
Transformaciones del Modelo

- **Rotación**
  - **Translación**
  - **Rotación**
  - **Escalado**

$$T : R^2 \Rightarrow R^2$$

$$P' = P + T$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$



Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Rotación**

$$x' = r\cos(\alpha + \beta)$$

$$y' = r\sin(\alpha + \beta)$$

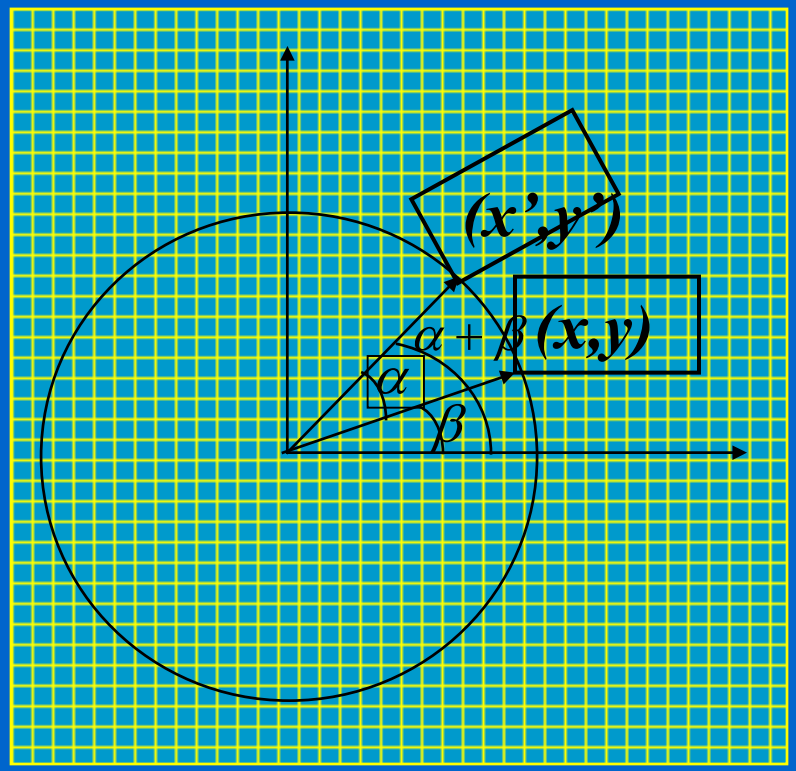
$$x = r\cos\beta$$

$$y = r\sin\beta$$

$$R : R^2 \Rightarrow R^2$$

$$P' = PR$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$



Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Escalado**

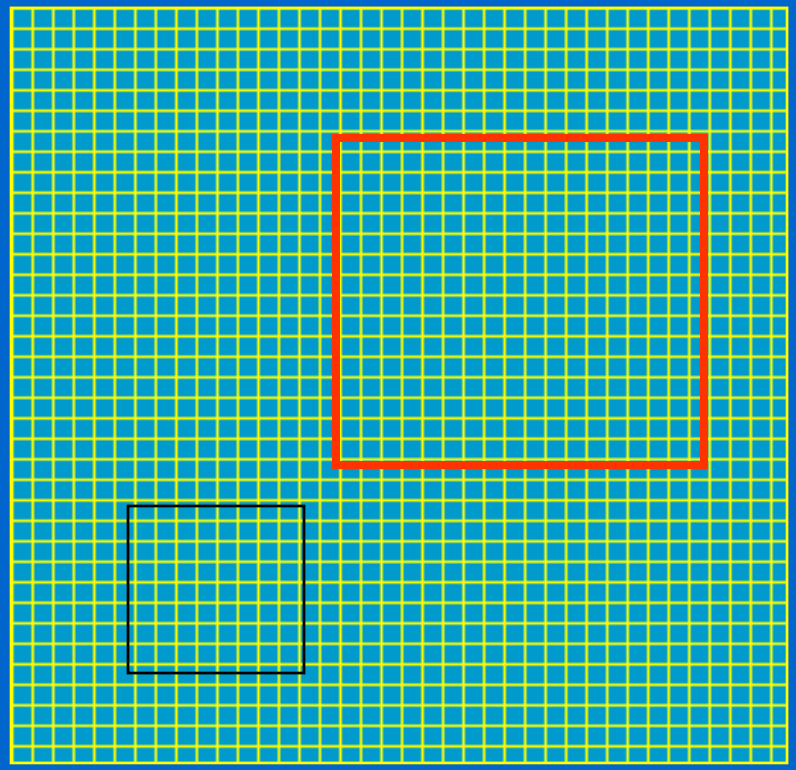
$$x' = xS_x$$

$$y' = yS_y$$

$$R : R^2 \Rightarrow R^2$$

$$P' = PS$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$



Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- Problemas no tenemos una representación homogénea

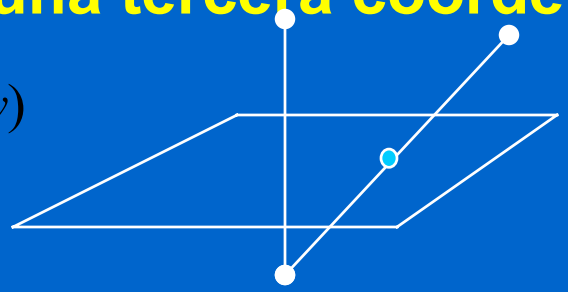
$$P' = P + T$$

$$P' = PR$$

$$P' = PS$$

- Para evitar esto introducimos las coordenadas Homogéneas: ahora para definir un punto añadimos una tercera coordenada ***W***

(x_h, y_h, w)



$(\frac{x_h}{w}, \frac{y_h}{w})$

Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- Transformaciones en coordenadas homogéneas.

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \qquad \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$x' = x + wt_x \Rightarrow \frac{x'}{w} = \frac{x}{w} + t_x$$

$$y' = y + wt_y \Rightarrow \frac{y'}{w} = \frac{y}{w} + t_y$$

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

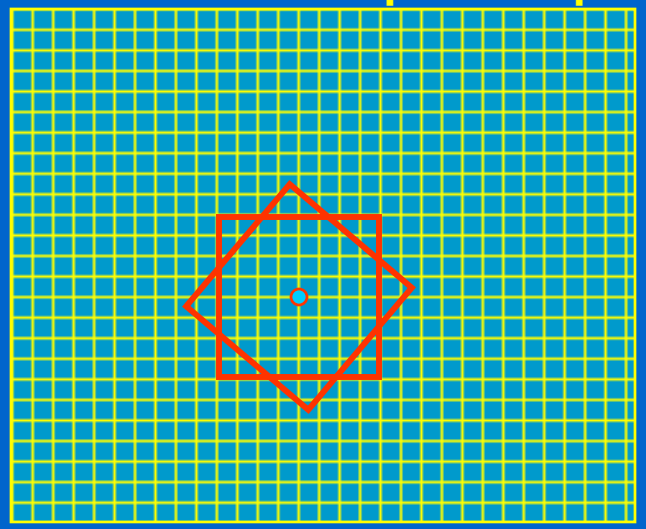
- Ahora ya tenemos una representación homogénea y podemos realizar composición de transformaciones.

Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Ejemplo Rotación alrededor de un punto:**
  - **Trasladar el punto al origen**
  - **Aplicar la rotación.**
  - **Devolver el conjunto a su posición deshaciendo la traslación original.**
  - **Recordar que la composición se realiza con premultiplicación**

$$P' = T_{xy} R_{45} T_{-xy} P$$

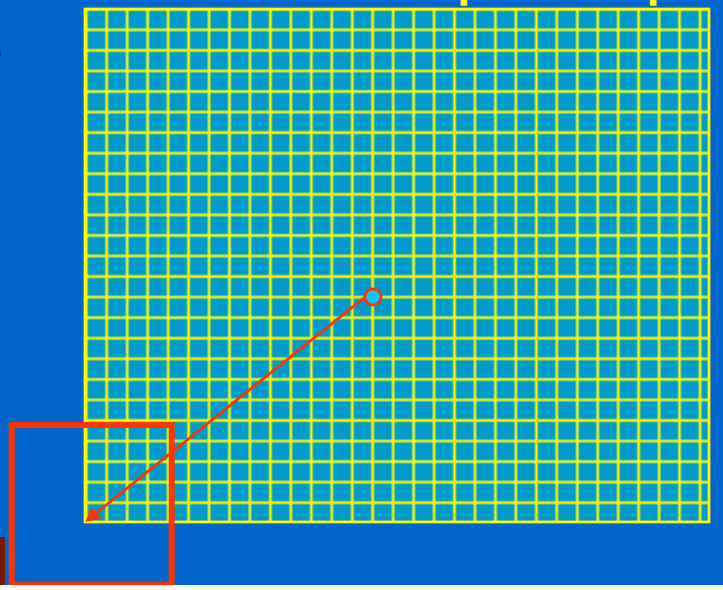


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Ejemplo Rotación alrededor de un punto:**
  - **Trasladar el punto al origen**
  - **Aplicar la rotación.**
  - **Devolver el conjunto a su posición deshaciendo la traslación original.**
  - **Recordar que la composición se realiza con premultiplicación**

$$P' = T_{xy} R_{45} T_{-xy} P$$

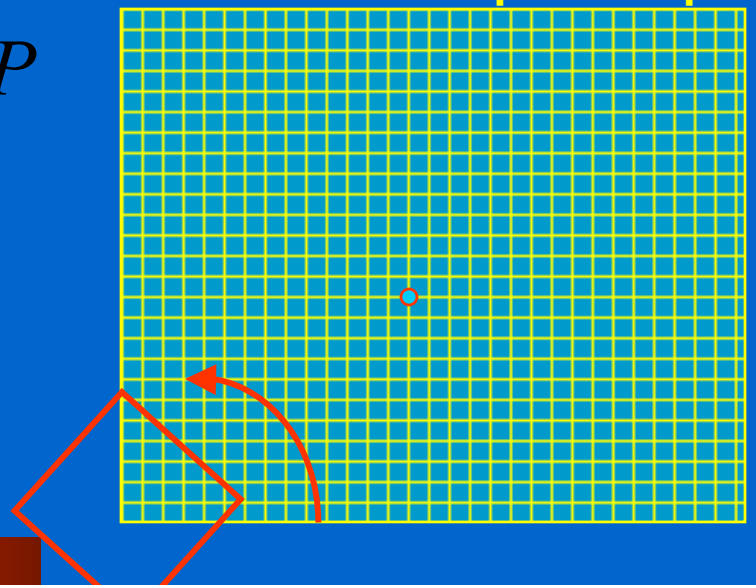


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Ejemplo Rotación alrededor de un punto:**
  - **Trasladar el punto al origen**
  - **Aplicar la rotación.**
  - **Devolver el conjunto a su posición deshaciendo la traslación original.**
  - **Recordar que la composición se realiza con premultiplicación**

$$P' = T_{xy} R_{45} T_{-xy} P$$

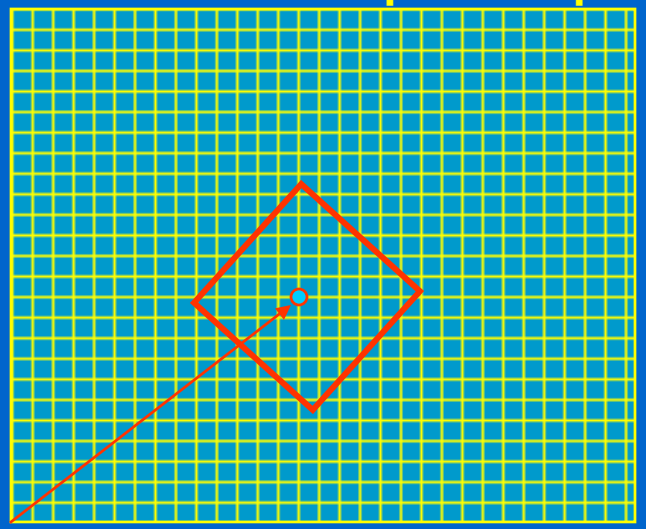


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- **Ejemplo Rotación alrededor de un punto:**
  - **Trasladar el punto al origen**
  - **Aplicar la rotación.**
  - **Devolver el conjunto a su posición deshaciendo la traslación original.**
  - **Recordar que la composición se realiza con premultiplicación**

$$P' = T_{xy} R_{45} T_{-xy} P$$

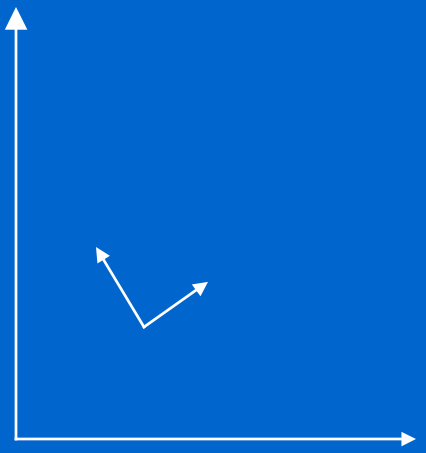


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- Matrices del Solido Rigido, solo contienen rotaciones y translaciones y conservan la ortogonalidad.
- Útiles para cambios de sistema de referencia.

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

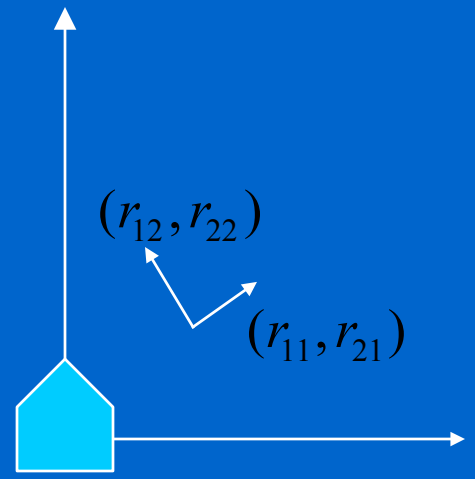


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- Matrices del Solido Rigido, solo contienen rotaciones y translaciones y conservan la ortogonalidad.
- Útiles para cambios de sistema de referencia.

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

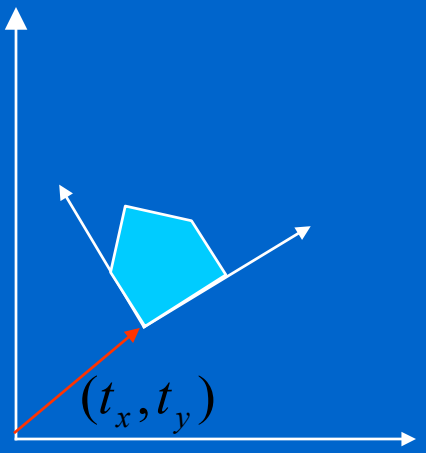


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo

- Matrices del Solido Rigido, solo contienen rotaciones y translaciones y conservan la ortogonalidad.
- Útiles para cambios de sistema de referencia.

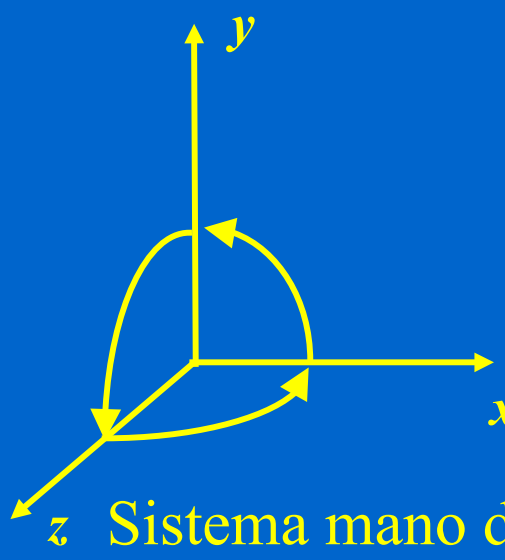
$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

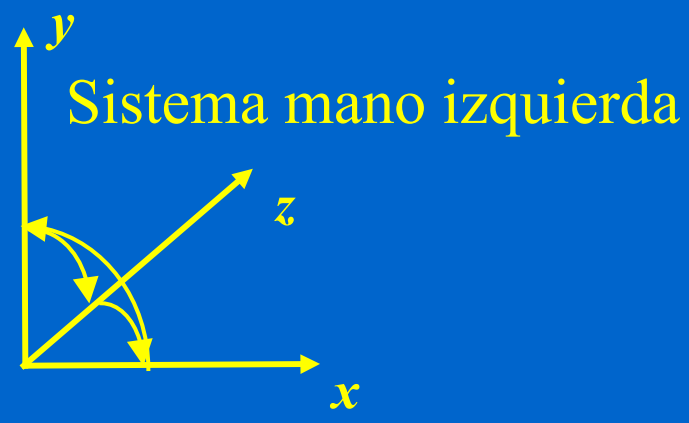


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- Simplemente añadimos una coordenada adicional.
- Consideraciones de sistemas de ejes.





Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- **Transformaciones básicas.**

$$T(t_x,t_y,t_z)=\begin{bmatrix}1&0&0&t_x\\0&1&0&t_y\\0&0&1&t_z\\0&0&0&1\end{bmatrix}$$

$$S(s_x,s_y,s_z)=\begin{bmatrix}S_x&0&0&0\\0&S_y&0&0\\0&0&S_z&0\\0&0&0&1\end{bmatrix}$$

$$T_{SólidoRígido}=\begin{bmatrix}r_{11}&r_{12}&r_{13}&t_x\\r_{21}&r_{22}&r_{23}&t_y\\r_{31}&r_{32}&r_{33}&t_z\\0&0&0&1\end{bmatrix}$$

$$R_z(\theta)=\begin{bmatrix}\cos\theta&\sin\theta&0&0\\-\sin\theta&\cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

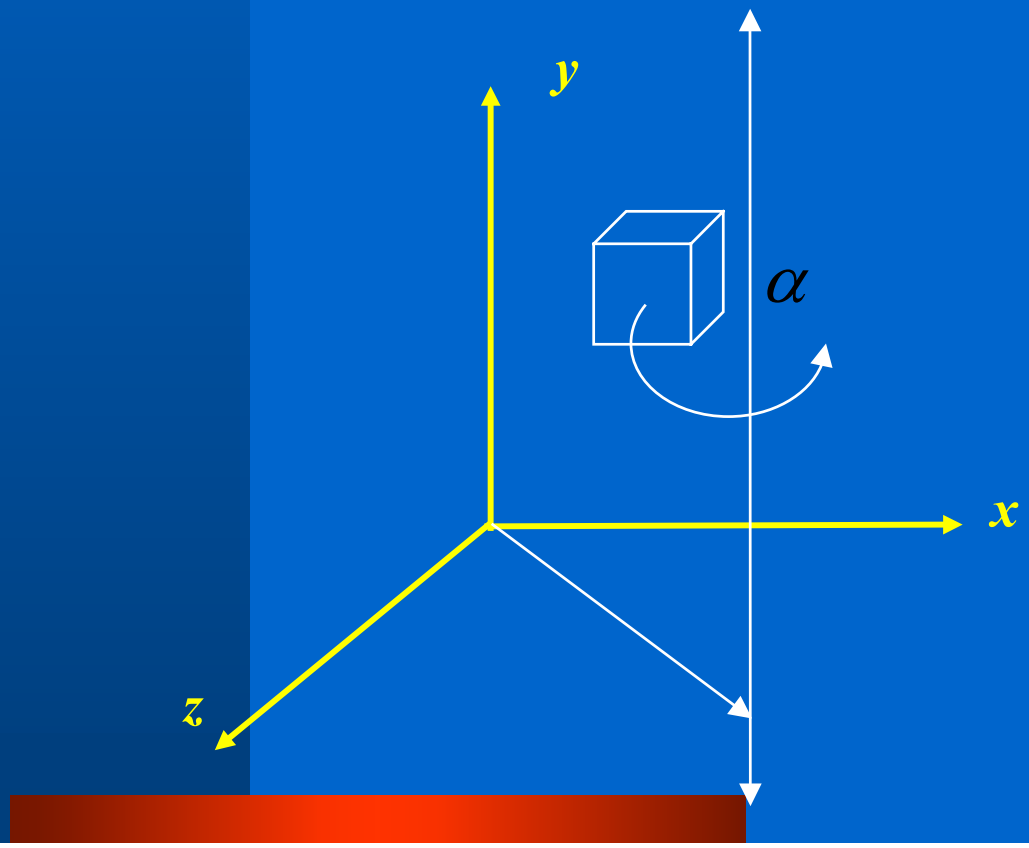
$$R_x(\theta)=\begin{bmatrix}1&0&0&0\\0&\cos\theta&-\sin\theta&0\\0&\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix}$$

$$R_y(\theta)=\begin{bmatrix}\cos\theta&0&\sin\theta&0\\0&1&0&0\\-\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix}$$

Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- Transformaciones Compuesta 3D: Giro alrededor de ejes no principales. Eje paralelo



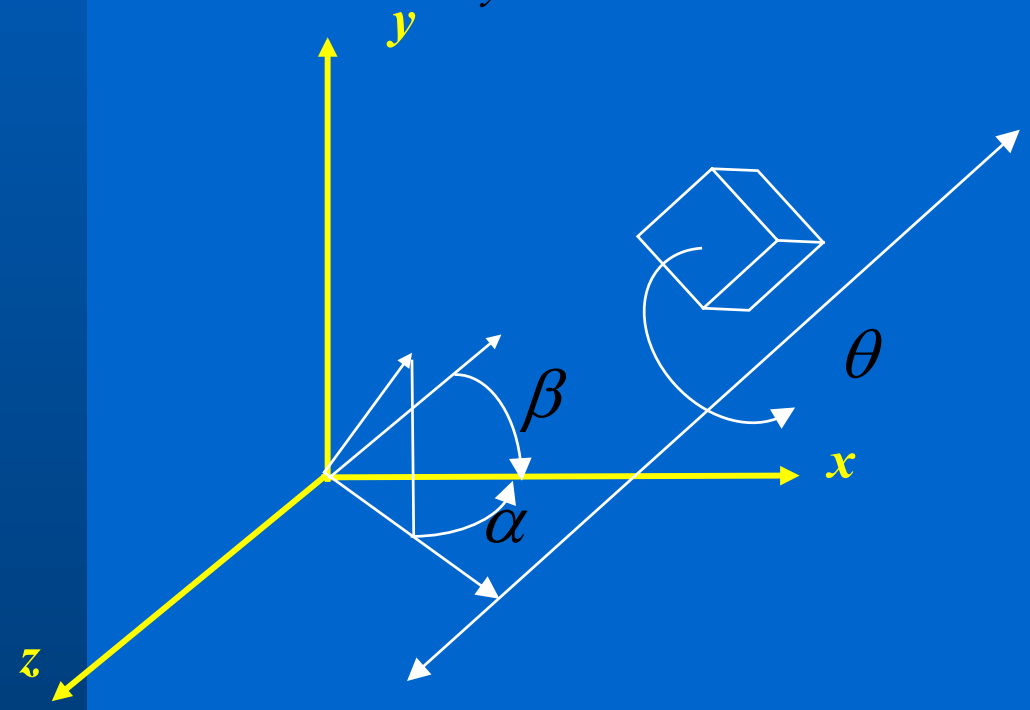
$$P' = T_{xz} R_y(\alpha) T_{-xz} P$$

Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- **Transformaciones Compuesta 3D: Giro alrededor de ejes no principales. Eje Génerico**

$$P' = T_{xz} R_y(\alpha) R_z(\beta) R_x(\theta) R_z(\beta) R_y(\alpha) T_{-xz} P$$

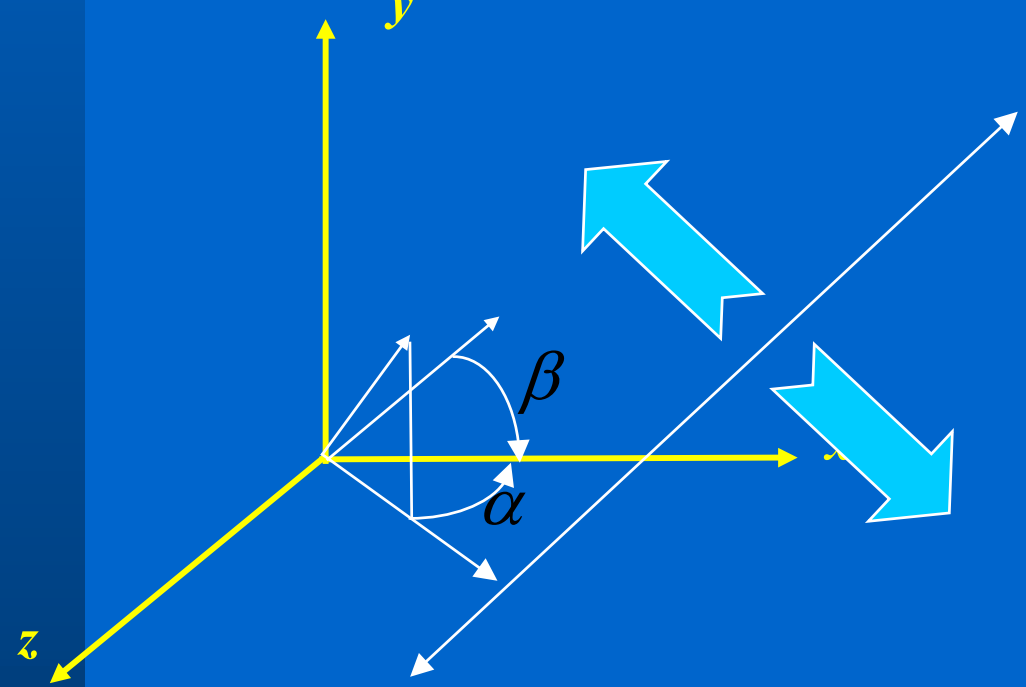


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- Transformaciones Compuesta 3D: Simetrías

$$P' = T_{xz} R_y(\alpha) R_z(\beta) S_y(-1) R_z(\beta) R_y(\alpha) T_{-xz} P$$

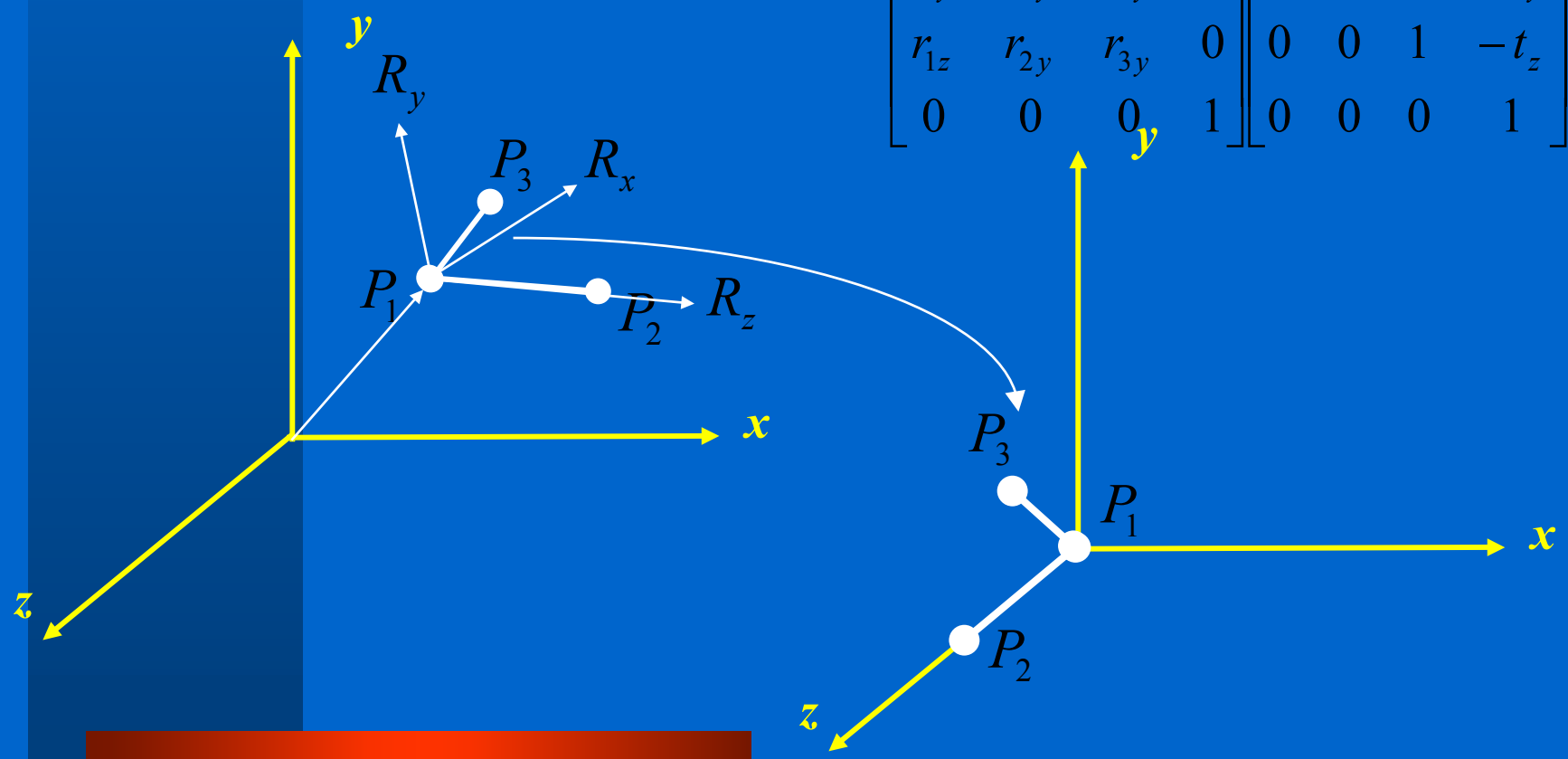


Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- **Transformaciones Compuesta 3D: Alineación de Vectores.**

$$\begin{bmatrix} r_{1x} & r_{2x} & r_{3x} & 0 \\ r_{1y} & r_{2y} & r_{3y} & 0 \\ r_{1z} & r_{2y} & r_{3y} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

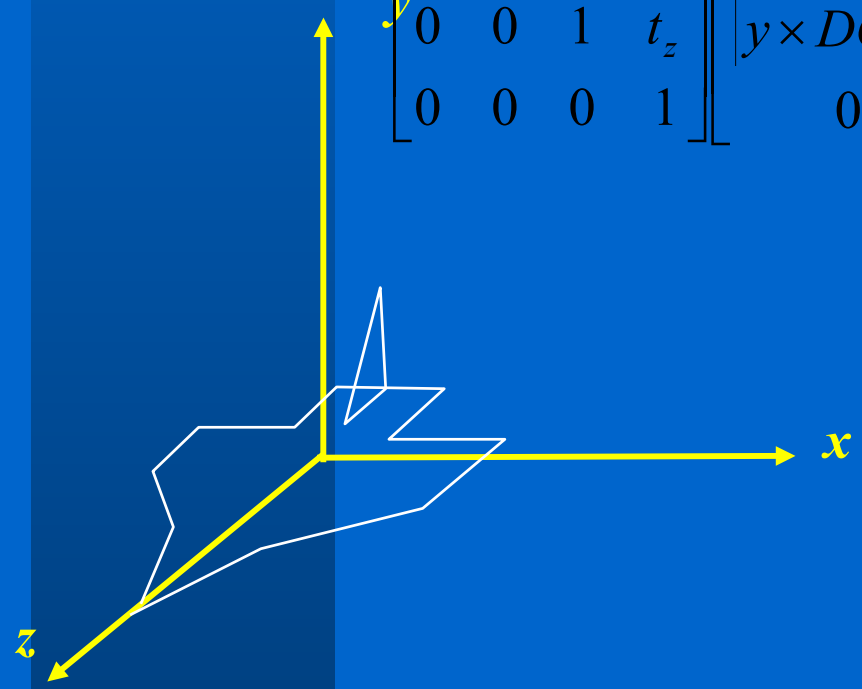


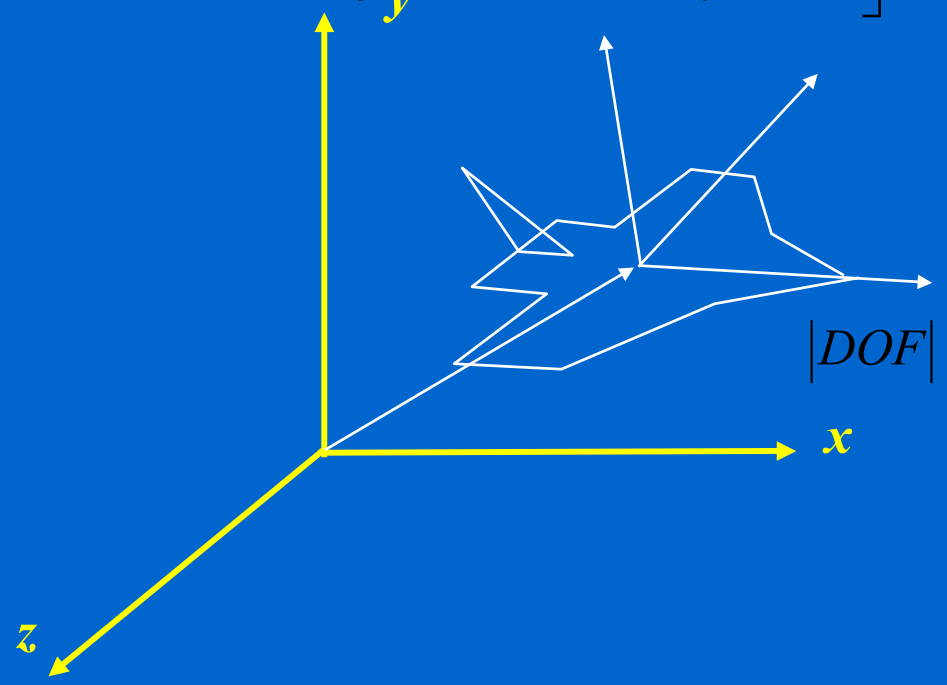
Etapas de la *pipeline* de vértices.

Transformaciones del Modelo:Versión 3D.

- **Transformaciones Compuesta 3D: Alinear un objeto.**

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} |y \times DOF|_x & |DOF \times (y \times DOF)|_x & |DOF|_x & 0 \\ |y \times DOF|_y & |DOF \times (y \times DOF)|_y & |DOF|_y & 0 \\ |y \times DOF|_z & |DOF \times (y \times DOF)|_z & |DOF|_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

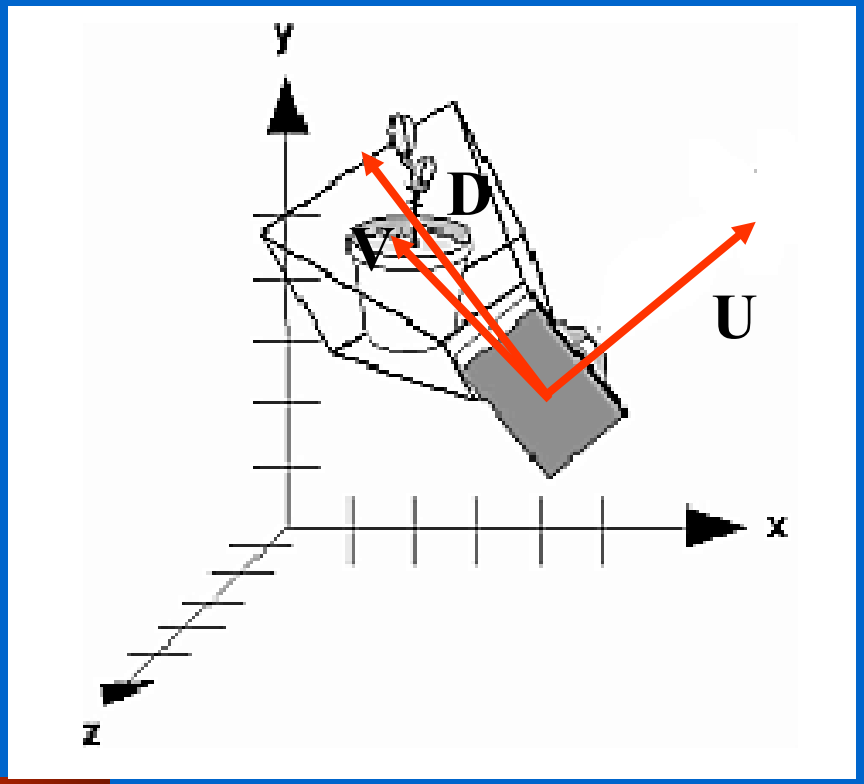




Etapas de la *pipeline* de vértices.

Transformaciones de la Vista

- La posición de la cámara viene definida por los siguientes elementos:
  - Vector de posición
  - Vector de orientación de la lente
  - Vector vertical.

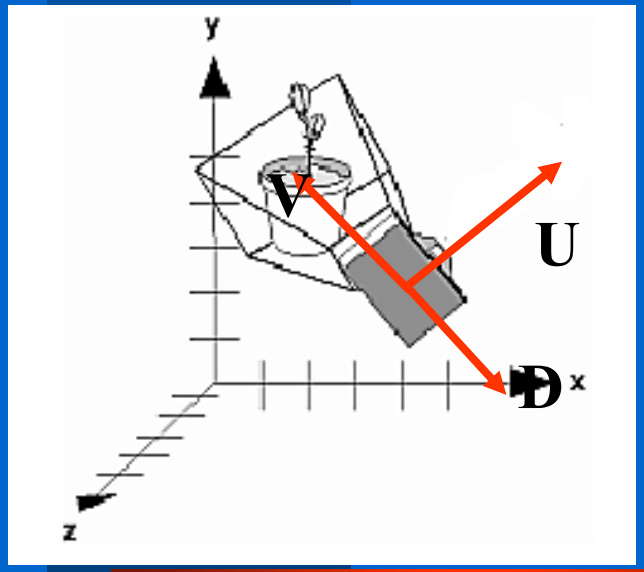


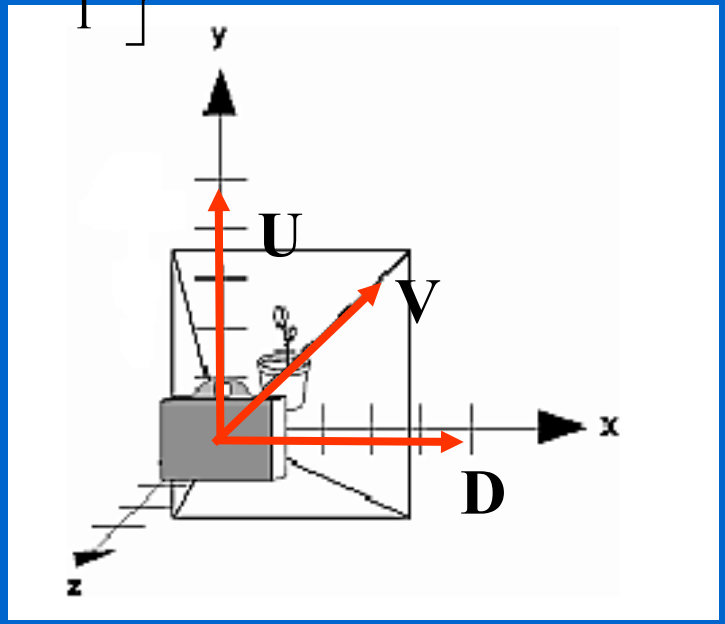
Etapas de la *pipeline* de vértices.

Transformaciones de la Vista

- **Para facilitar operaciones siempre colocamos la cámara en una posición determinada.**

$$\begin{bmatrix} D_x & D_y & D_z & 0 \\ U_x & U_y & U_z & 0 \\ -V_x & -V_y & -V_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

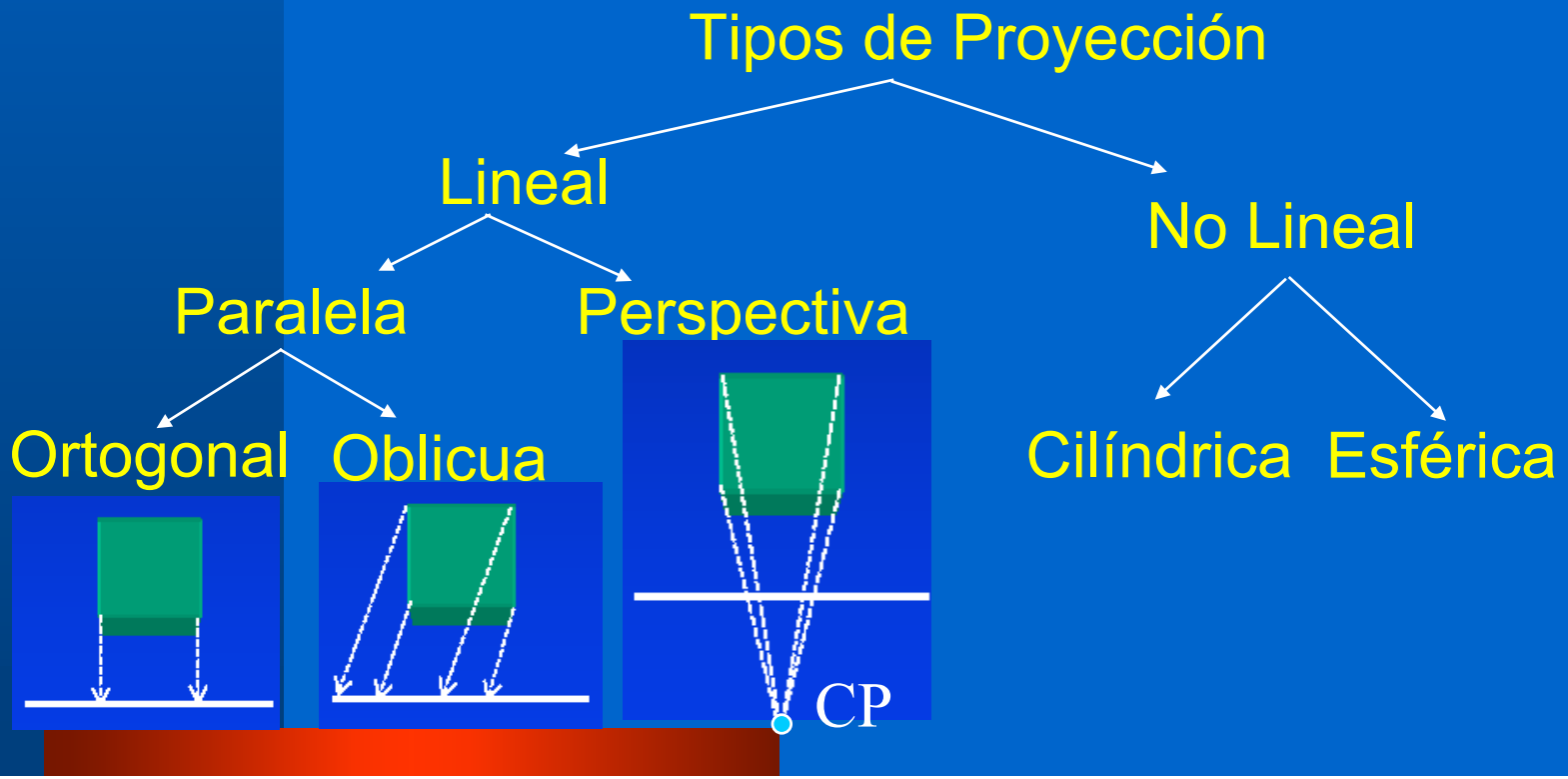




Etapas de la *pipeline* de vértices.

Transformaciones de proyección

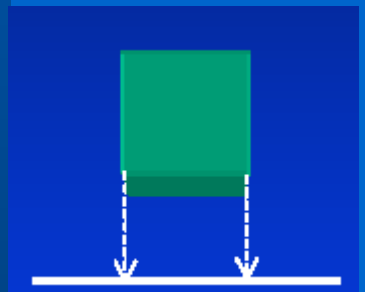
- **En esta punto de la tubería se produce un cambio de coordenadas 3D a coordenadas 2D. También obtenemos para cada vértices un valor adicional referente a la profundidad.**



Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Paralelas

- **Proyección paralela ortogonal.**
  - **Los rayos de proyección son ortogonales al plano de proyección y paralelos entre sí.**
  - **Conserva las coordenadas X e Y**



Plano de Proyección

$$X_p = X_c$$

$$Y_p = Y_c$$

$$Z_p = 0$$

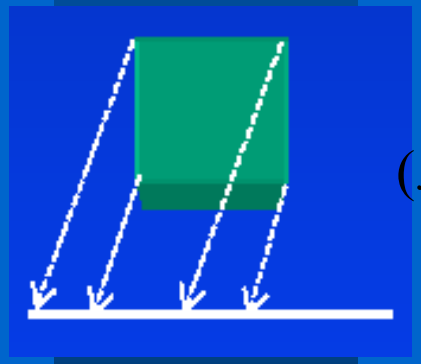
$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$$P_p = M_{Ortho} P_c$$

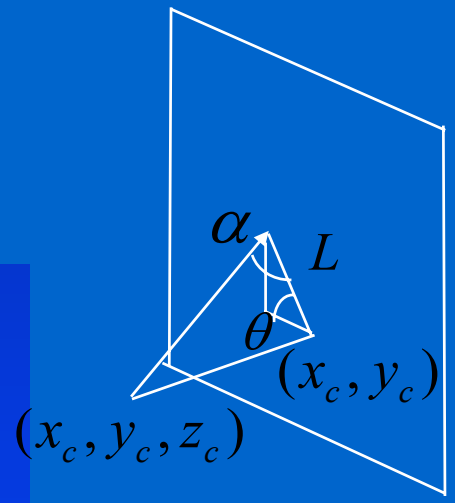
Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Paralelas

- **Proyección paralela Oblicua.**
  - Los rayos de proyección son paralelos entre sí y forma un determinado ángulo vertical y horizontal con el plano de proyección.
  - Caballera, tanα=1
  - Cabinet, tanα=2



Plano de Proyección



$$X_p = X_c + L\cos\theta \rightarrow X_p = X_c + \frac{Z_c}{\tan\alpha}\cos\theta$$

$$Y_p = Y_c + L\sin\theta \rightarrow Y_p = Y_c + \frac{Z_c}{\tan\alpha}\sin\theta$$

$$Z_p = 0$$

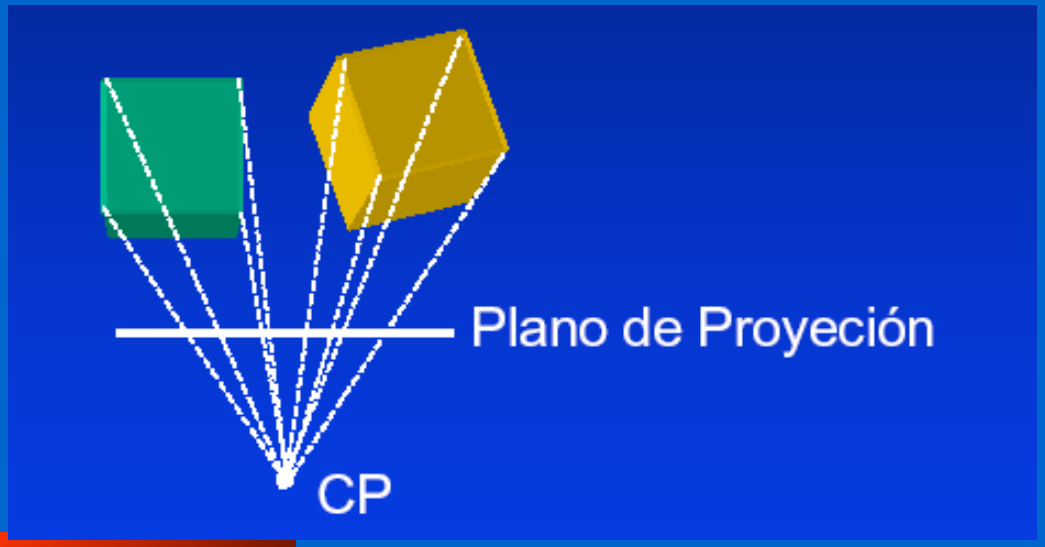
$$\tan\alpha = \frac{Z_c}{L}$$

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L\cos\theta & 0 \\ 0 & 1 & L\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

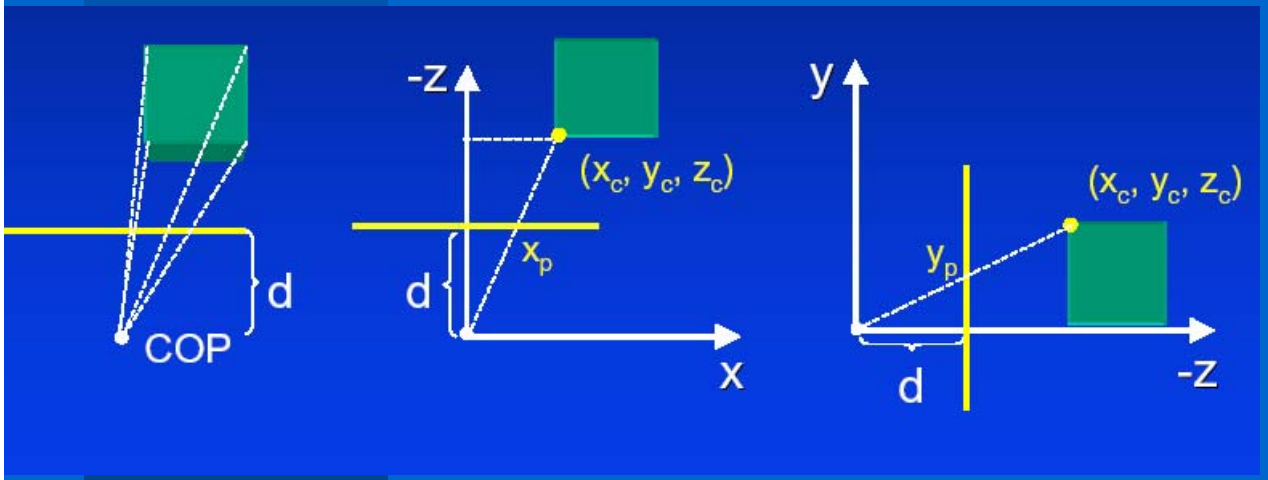
- **Perspectiva Lineal.**
  - Con un solo centro de proyección.
  - Con al plano de proyección las lineas proyectadas se acortan.
  - Simula relativamente bien la visión 3D humana, es más natural que las proyecciones paralelas
  - Solamente las líneas paralelas al plano de proyección conservan el paralelismo al ser proyectadas.



Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Perspectiva Lineal.**
  - **Si proyectamos cada coordenada por separado y utilizamos las propiedades de los triángulos semejantes tenemos.**



$$x_p = \frac{x_c}{-z_c} d$$

$$x_p = \frac{y_c}{-z_c} d$$

$$z_p = -d$$

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ W_p \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$$P_p = M_{Persp} P_c$$

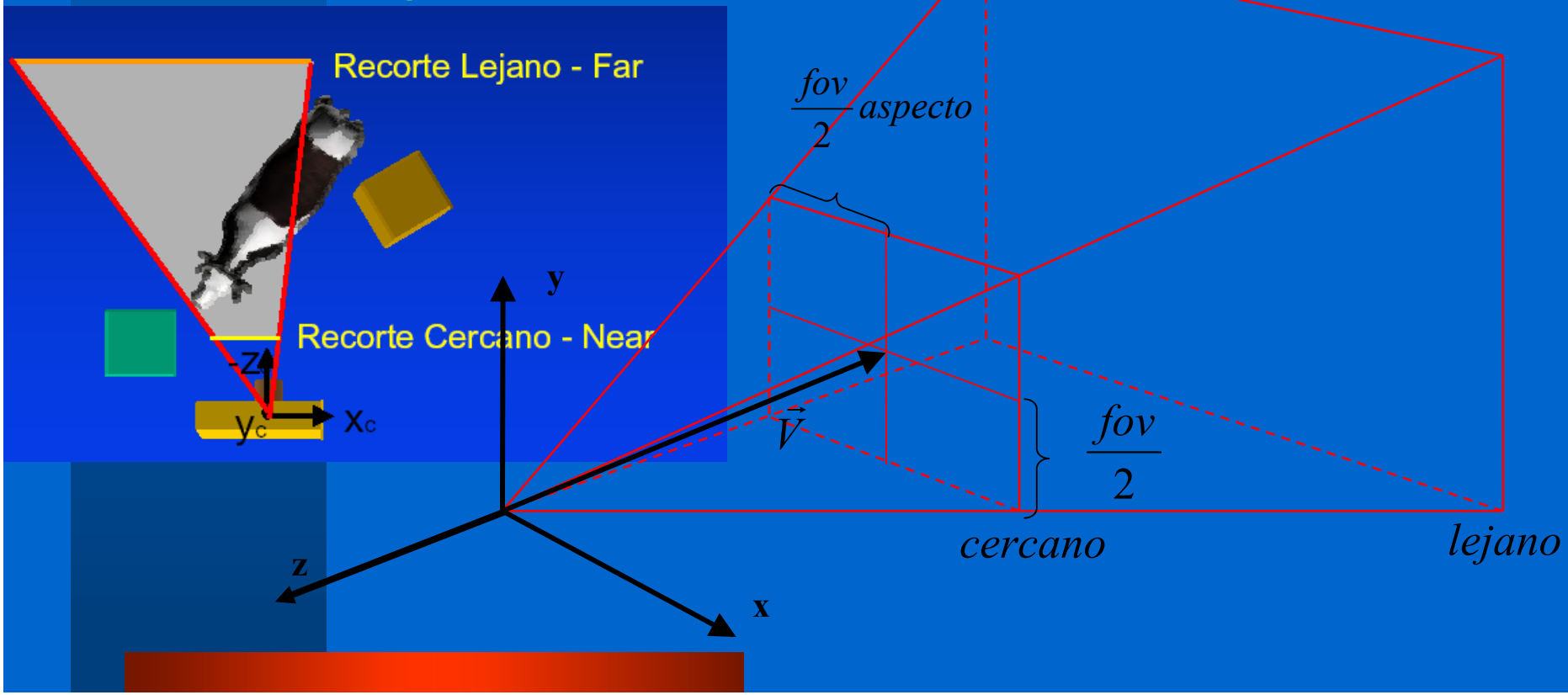
$$W_p = -Z_c / d$$

Pero nada es perfecto estamos mintiendo: d ≠z
Sobre todo cuanto estamos lejos del eje del la pirámide de visión. Esto produce deformaciones sobre todo para ángulos de visión grandes.

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

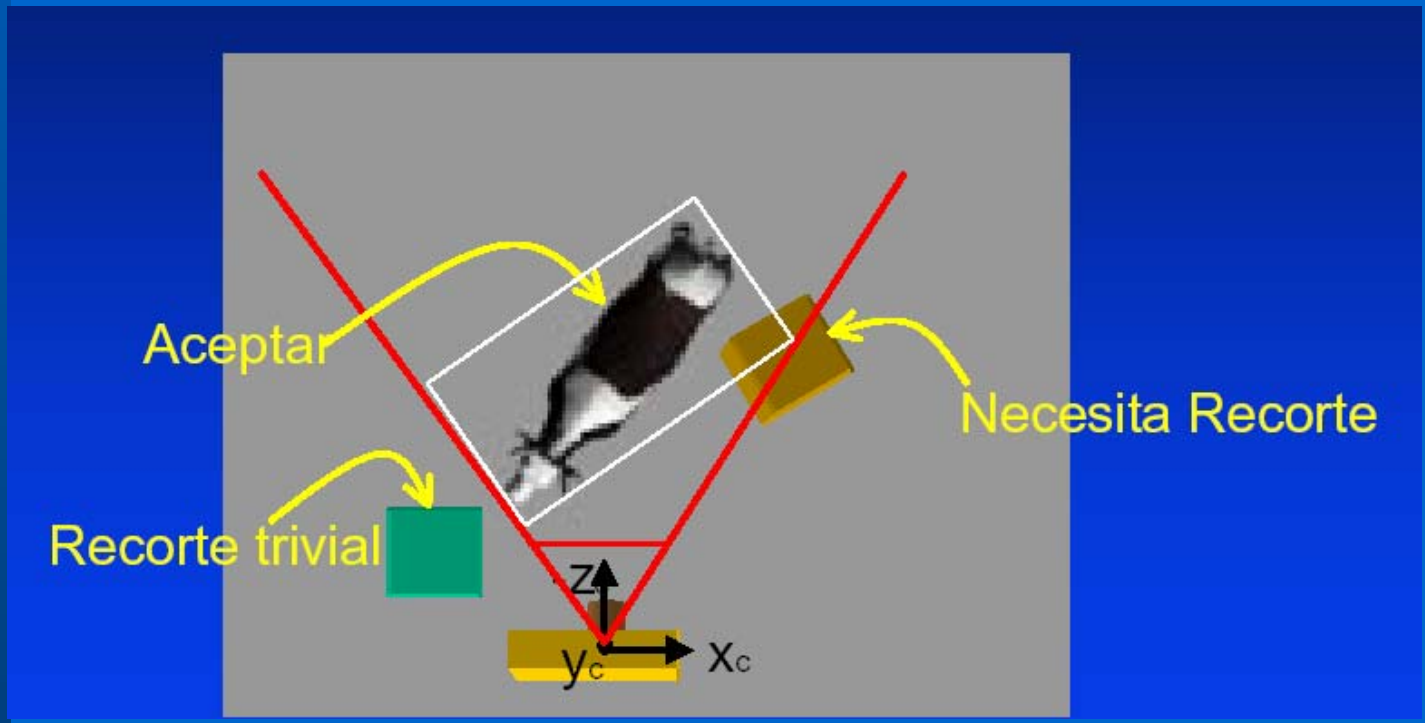
- **Perspectiva Lineal.**
  - Para definir la proyección completa necesitamos algunos datos más. Toda esta información corresponde a la pirámide de visión o *frustrum*
  - Los *frustums* pueden ser simetricos o asimétricos según nos convenga.



Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

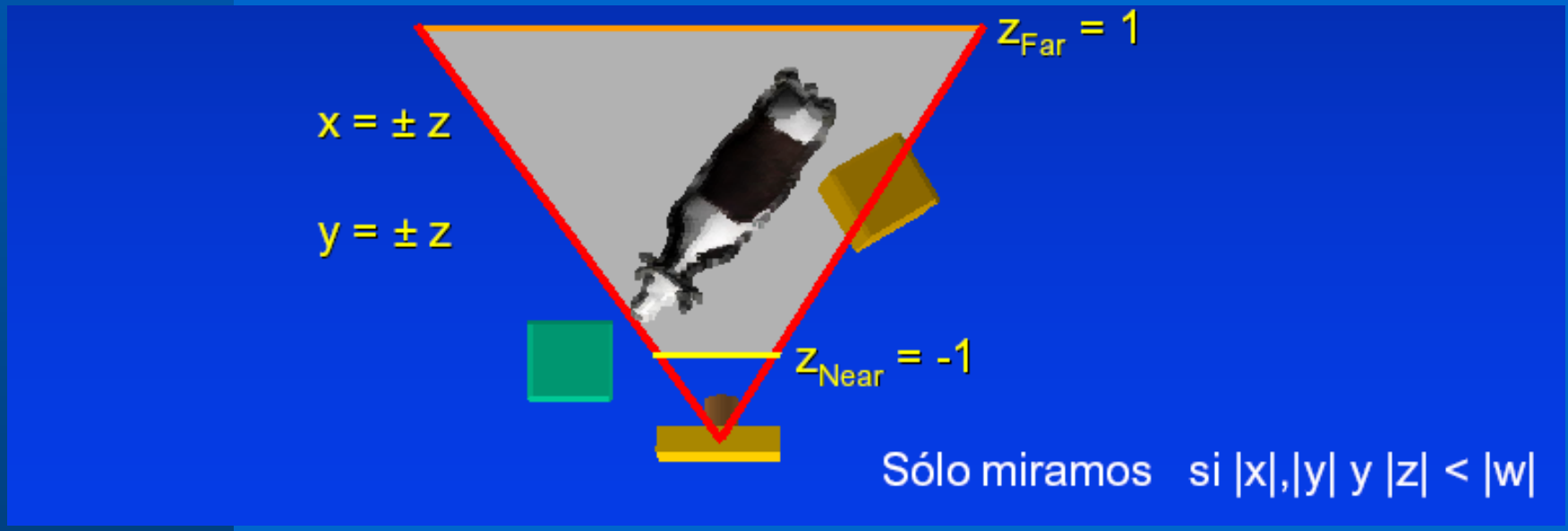
- **Recorte contra la pirámide de visión.**
  - En la imagen final tenemos partes que sobran tenemos que recortarlas



Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

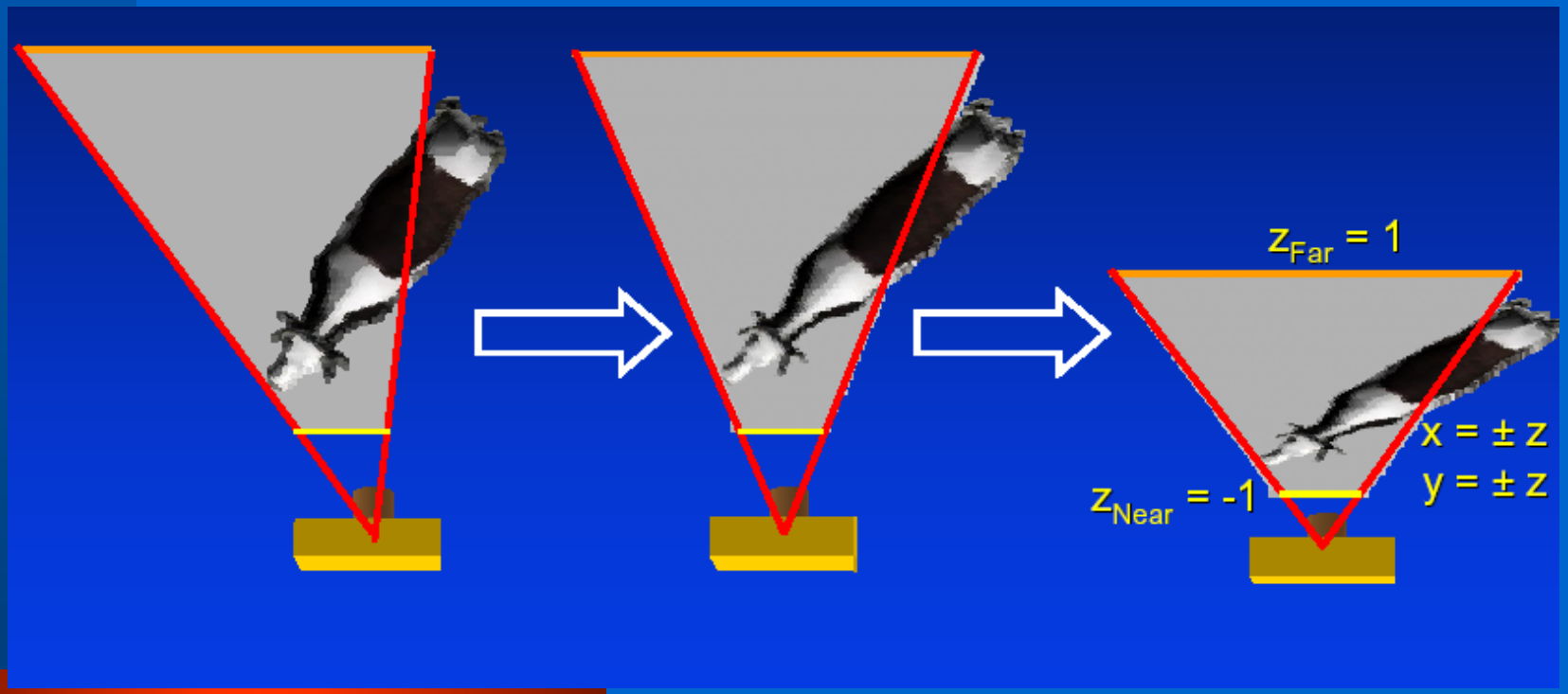
- **Recorte contra la pirámide de visión.**
  - Una manera de simplificar el recorte es utilizar siempre la misma pirámide. Esto es lo que se llama el volumen canónico.



Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

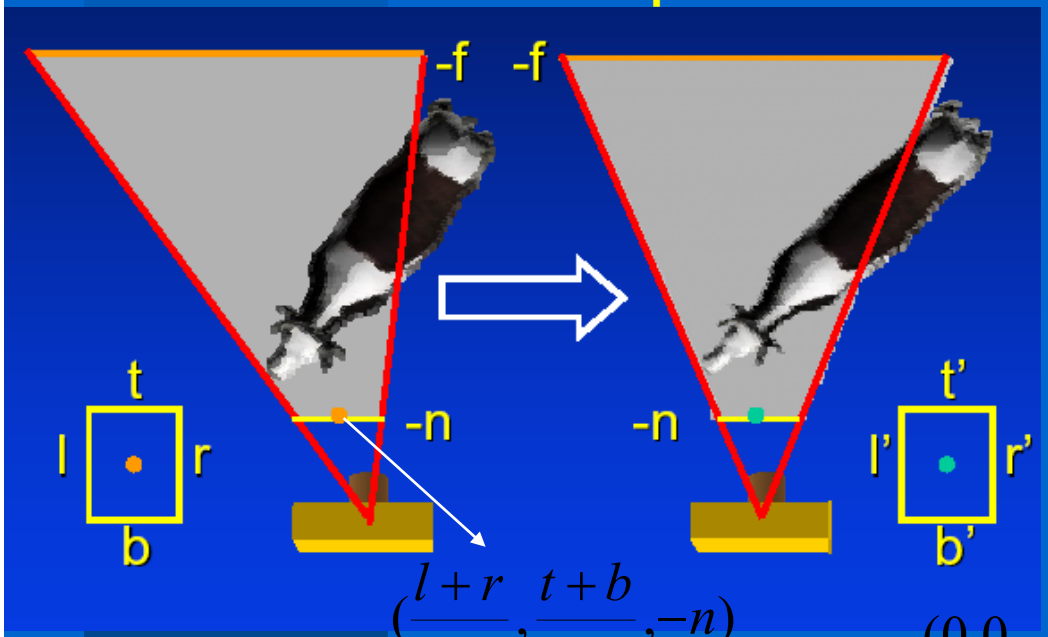
- **Recorte contra la pirámide de visión.**
  - No obstante como queremos libertad a la hora de configurar la lente de nuestra cámara (campo de visual, zoom, etc), conseguir el volumen canónico nos costara realizar algunas transformaciones. Shear y Escalado (XY y Z)



Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Recorte contra la pirámide de visión.**
  - El shear o desplazamiento será necesario en el caso de *frustums*



$$(\frac{l+r}{2}, \frac{t+b}{2}, -n) \qquad (0,0,-n)$$

$$H = \begin{bmatrix} 1 & 0 & g & 0 \\ 0 & 1 & h & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -n \\ 1 \end{bmatrix} = H \begin{bmatrix} (l+r)/2 \\ (t+b)/2 \\ -n \\ 1 \end{bmatrix}$$

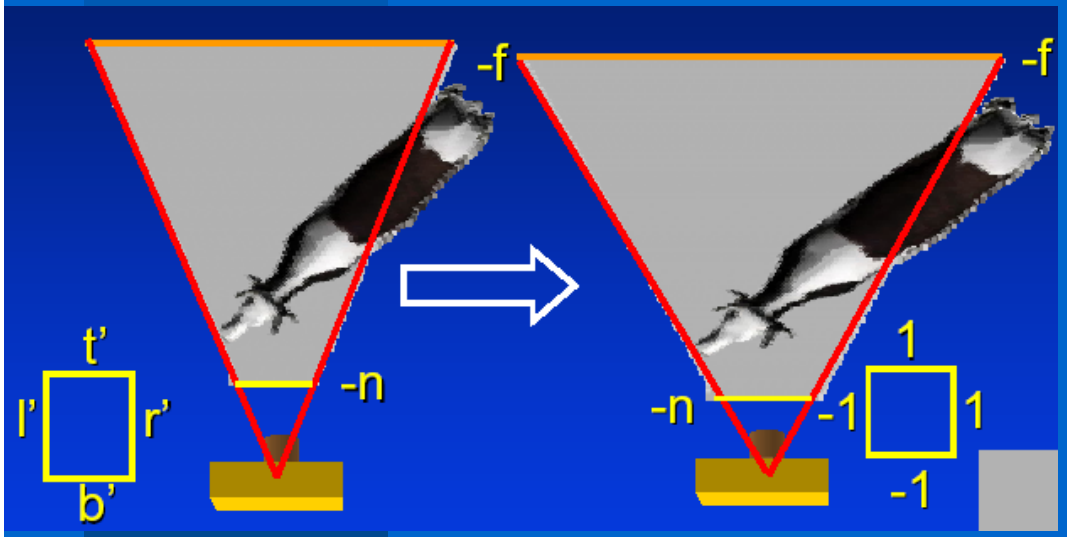
$$g = \frac{l+r}{2n} \qquad h = \frac{t+b}{2n}$$

$$H = \begin{bmatrix} 1 & 0 & \frac{l+r}{2n} & 0 \\ 0 & 1 & \frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Recorte contra la pirámide de visión.**
  - El escalado se tendrá que realizar siempre que no tengamos como ventana de coordenadas –1,-1,1,1



$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S_x = \frac{2}{r-l}n$$

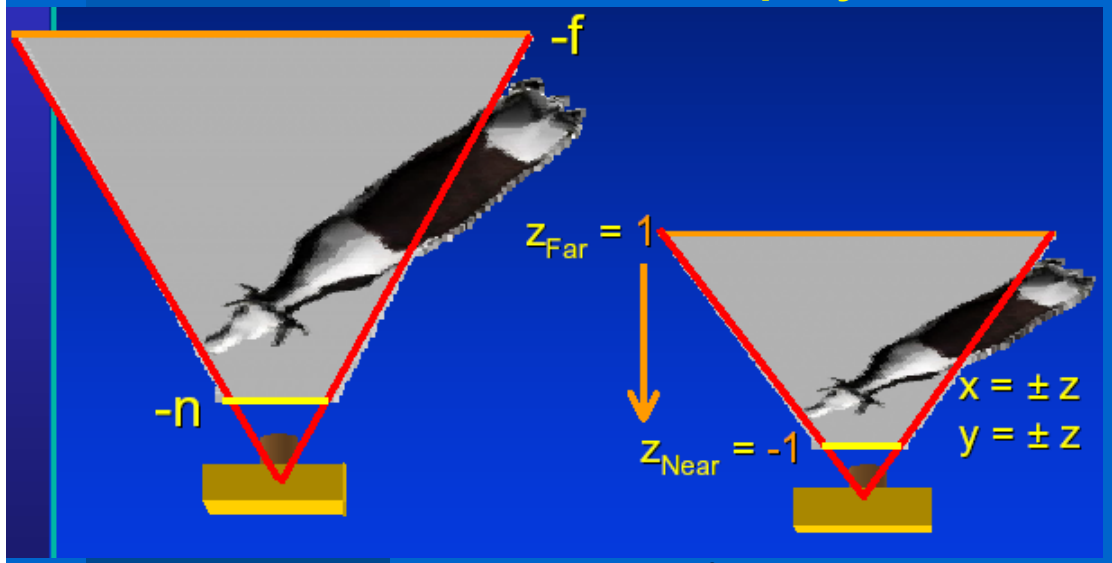
$$S_y = \frac{2}{t-b}n$$

$$S = \begin{bmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Recorte contra la pirámide de visión.**
  - Nos falta la normalización en Z que tratamos aparte por que esta relacionada con la proyección



$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -n \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -f \\ 1 \end{bmatrix}$$

$$\alpha = -\frac{f+n}{f-n}$$

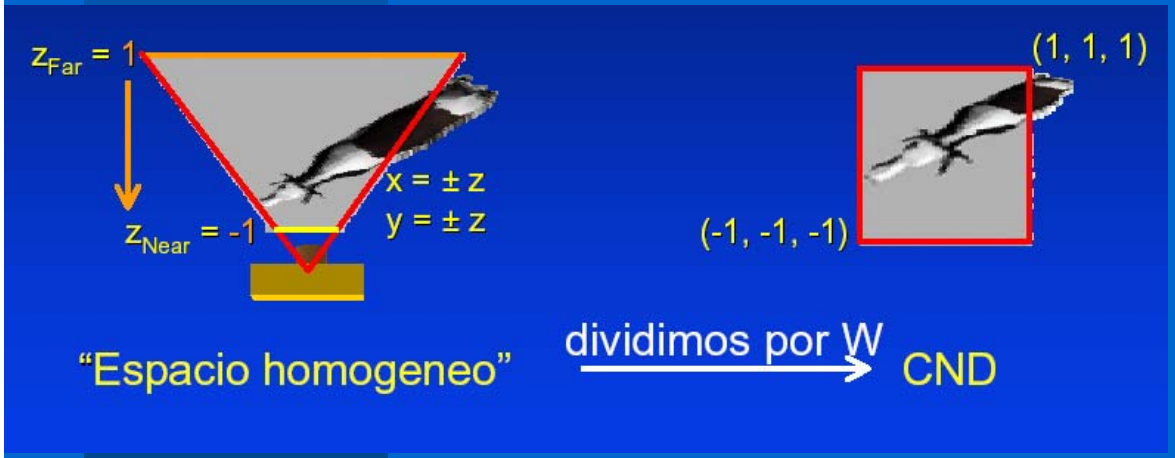
$$\beta = -\frac{2fn}{f-n}$$

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Matriz de proyección total.**
  - Multiplicamos las tres matrices obtenidas.
  - Nos faltaría dividir por W para obtener coordenadas normalizadas del dispositivo.



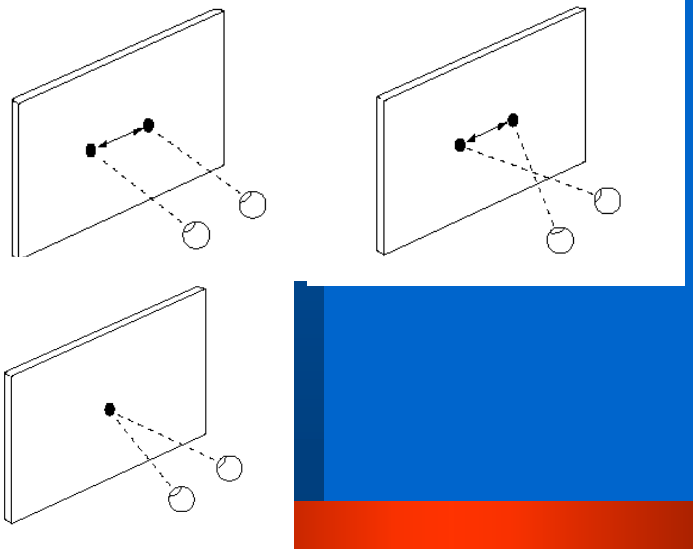
$$P = NSH$$

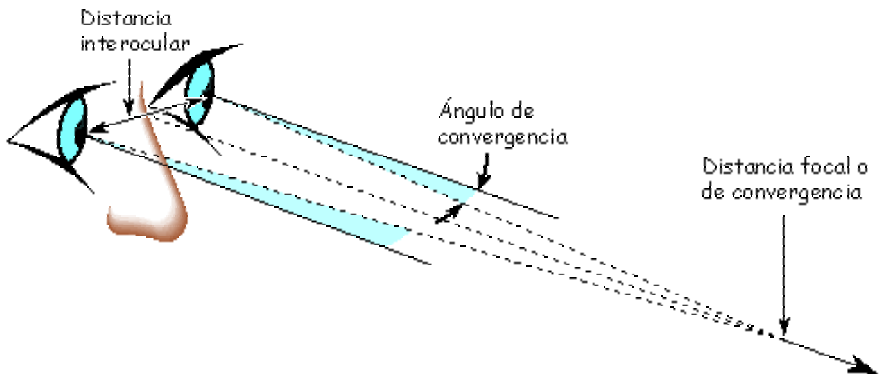
$$P = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Configuraciones de proyección para Visualización Estereoscópica.**
  - Efecto basado en la disparidad binocular: diferencia entre lo que ve el ojo derecho y el izquierdo.
  - Esto depende de:
    - Distancia interocular
    - Distancia focal o de convergencia
  - Cuando llevamos esto a una proyección podemos tener distintos paralajes: positivo, negativo y cero.

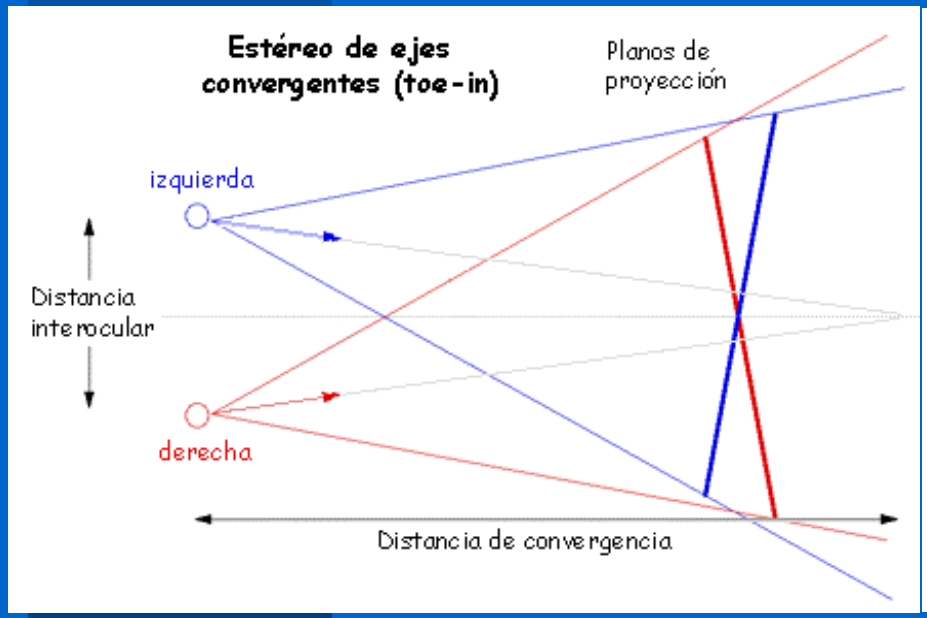


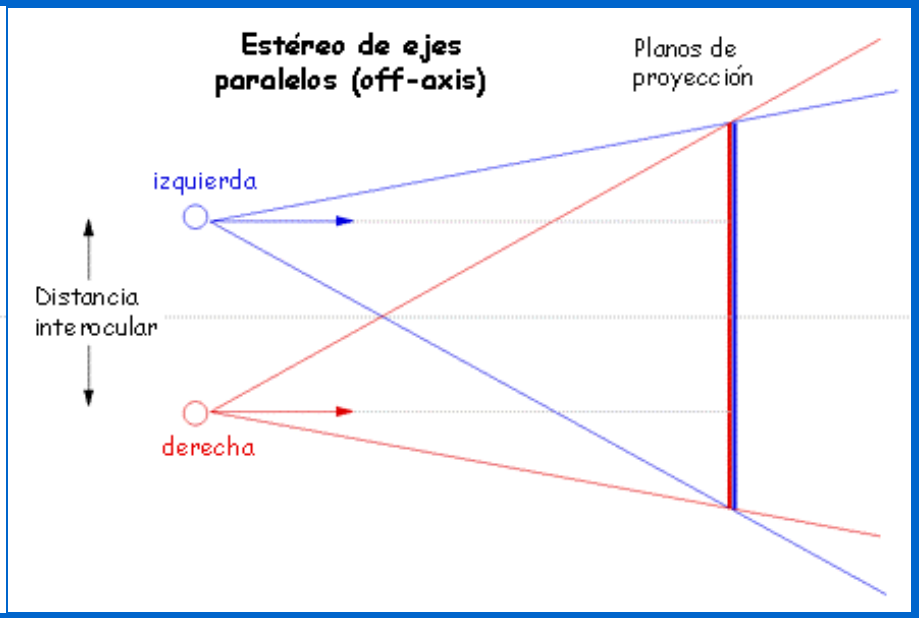


Etapas de la *pipeline* de vértices.

Transformaciones de proyección:Perspectiva

- **Configuraciones de proyección para Visualización Estereoscópica.**
  - **Dos aproximaciones básicas para la generación en gráficos que afecta a la configuración de la pirámide de visión:**
    - **Estéreo de ejes convergentes (TOE-IN)**
    - **Estéreo de ejes paralelos (OFF-AXIS)**





Etapas de la *pipeline* de vértices.

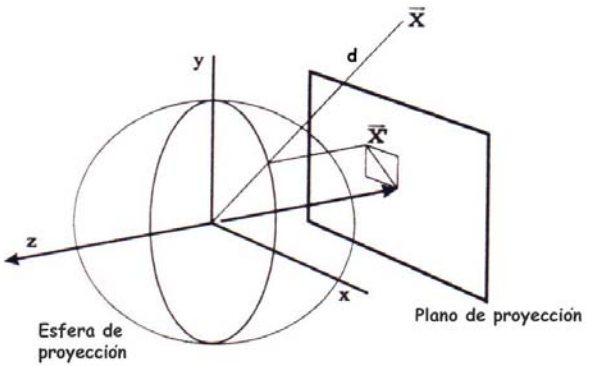
Transformaciones de proyección:Perspectiva

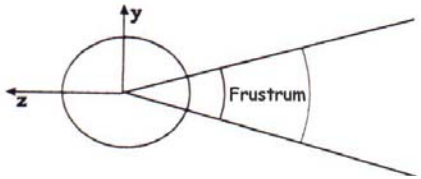
- Ejemplos de estereoscopia con anaglifos

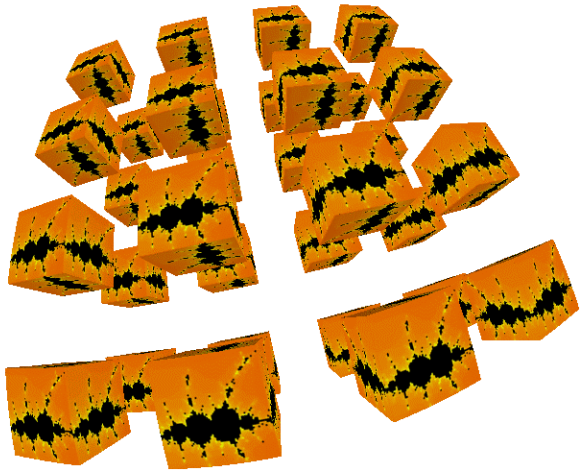
Etapas de la *pipeline* de vértices.

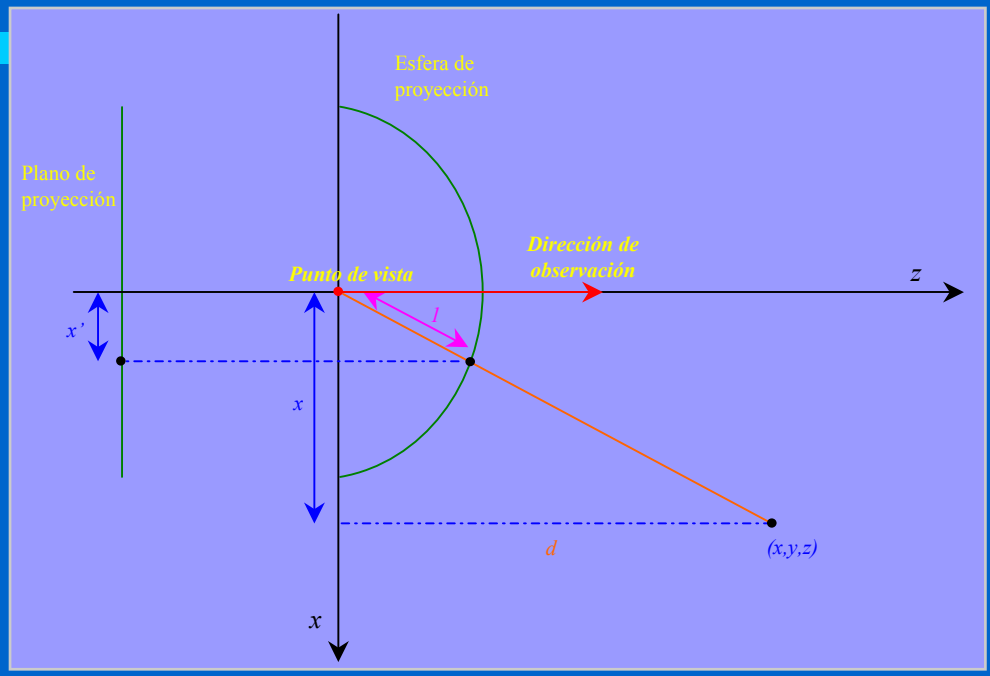
Transformaciones de proyección:no lineales

- Proyección Esférica









$$x' = \frac{x}{d} \qquad x' = \frac{x}{\sqrt{(x^2 + y^2 + z^2)}} \qquad x' = \frac{x}{(d/z)} = \frac{x \cdot z}{d}$$

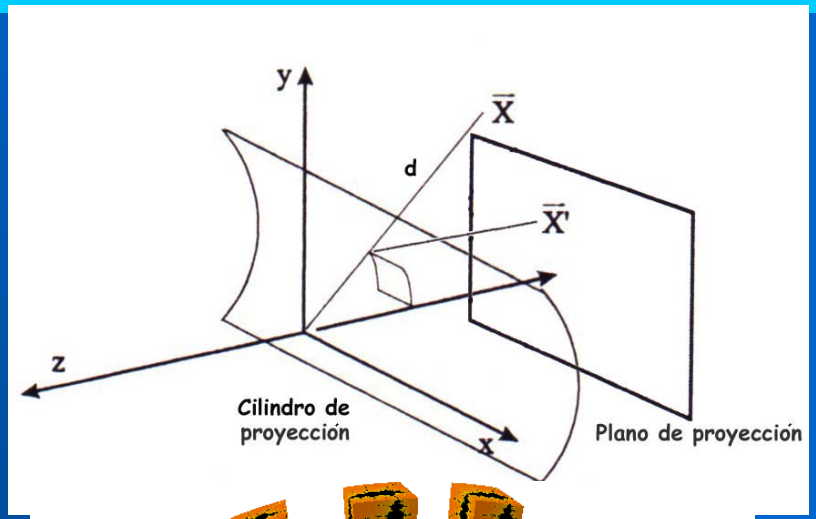
$$y' = \frac{y}{\sqrt{(x^2 + y^2 + z^2)}} \qquad y' = \frac{y}{(d/z)} = \frac{y \cdot z}{d}$$

$$z' = \sqrt{(x^2 + y^2 + z^2)}$$

Etapas de la *pipeline* de vértices.

Transformaciones de proyección:no lineales

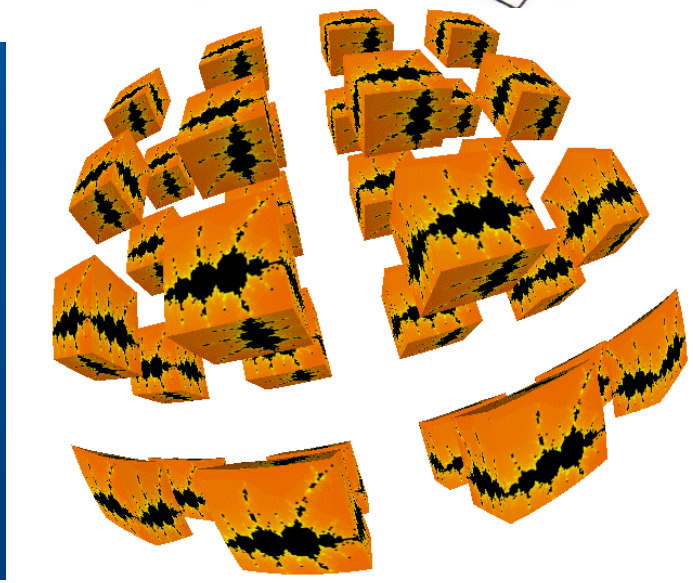
- Proyección Cilíndrica

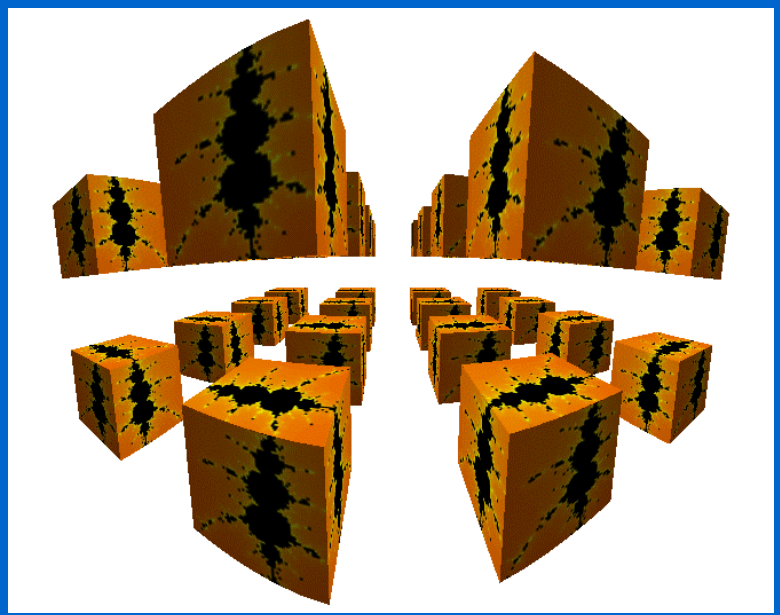


$$x' = \frac{x}{\sqrt{(x^2 + z^2)}}$$

$$y' = \arctan(\frac{y}{z})$$

$$z' = \sqrt{(x^2 + y^2 + z^2)}$$

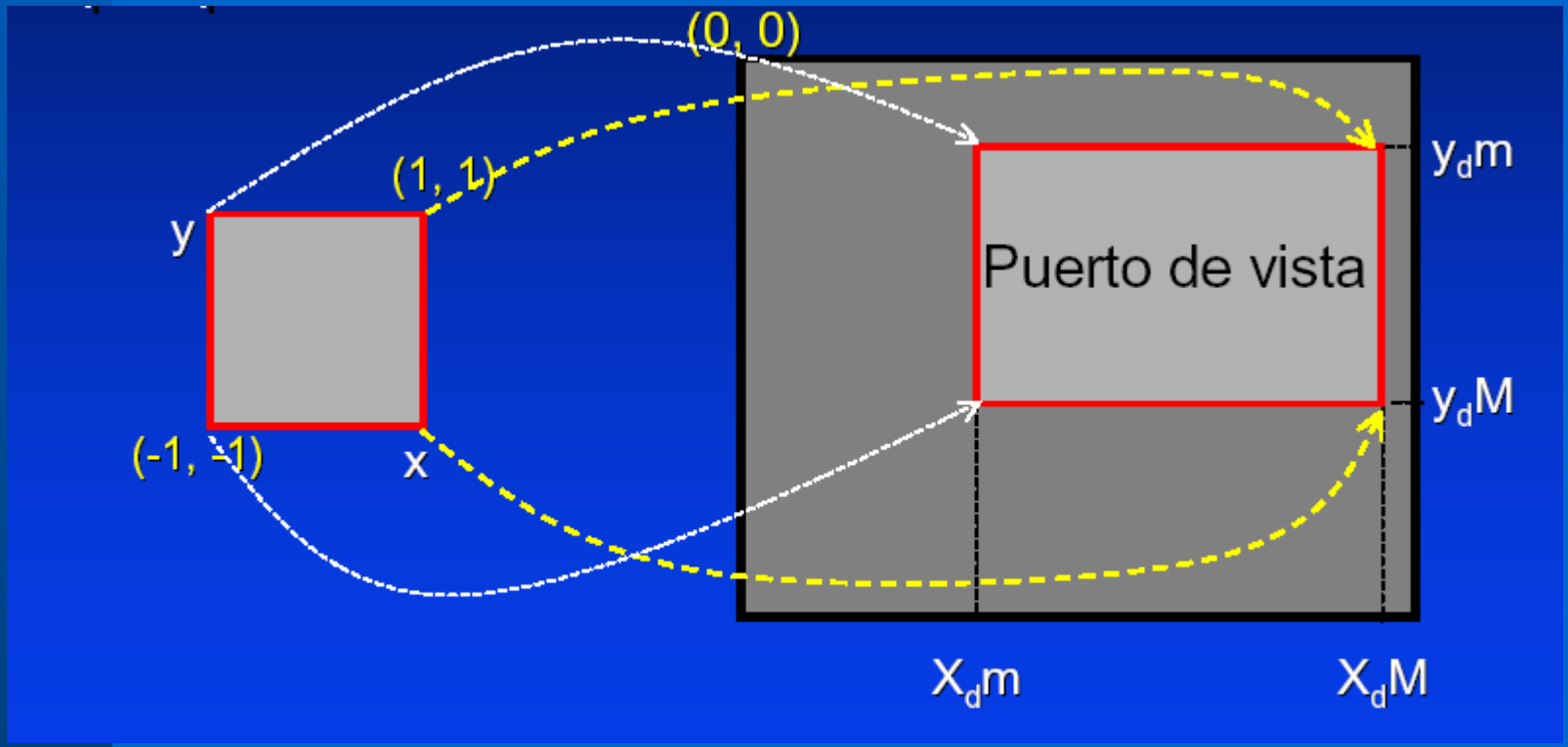




Etapas de la *pipeline* de vértices.

Transformaciones de Ventana.

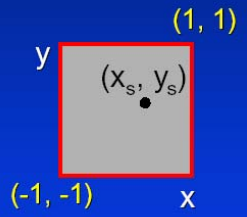
- Ahora tenemos que convertir nuestras coordenados 2D en coordenadas de pixels para esto nos tenemos que ajustar al puerto de vista definido.



Etapas de la *pipeline* de vértices.

Transformaciones de Ventana.

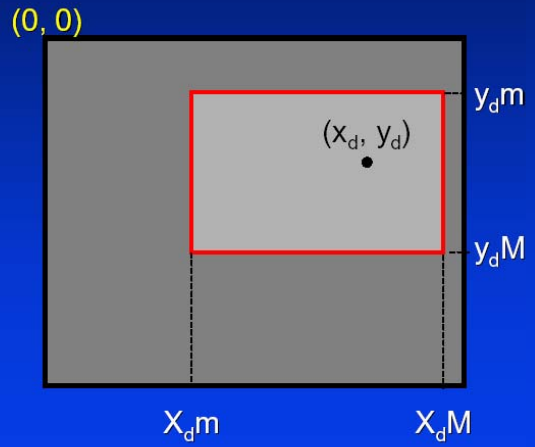
- Esto básicamente consiste en una traslación y en un escalado.

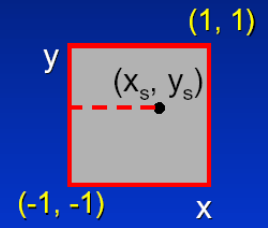


$\Delta x_s = \Delta y_s = 2$

$\Delta x_d = X_dM - X_dm$

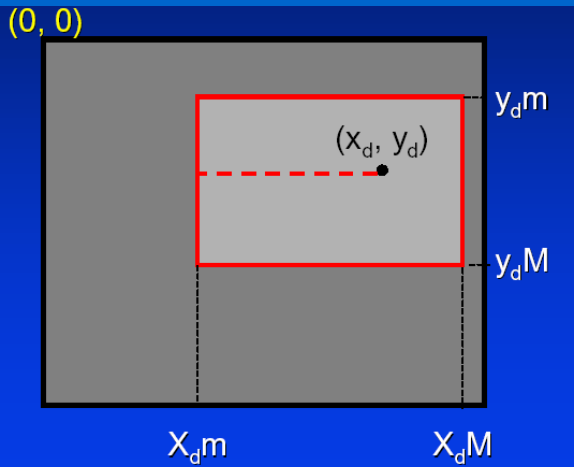
$\Delta y_d = y_dM - y_dm$

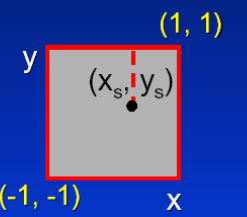




$$\frac{x_s - (-1)}{2} = \frac{x_d - x_d m}{\Delta x_d}$$

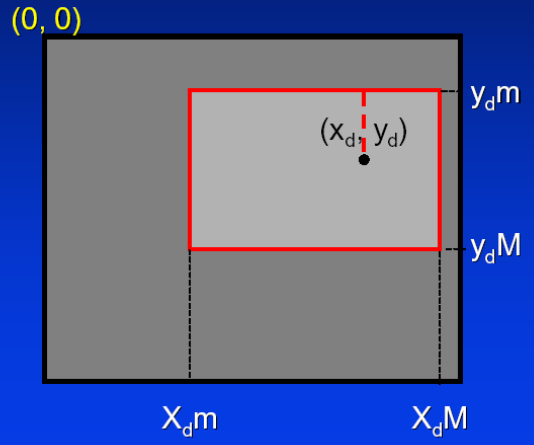
$$x_d = (x_s + 1)\frac{\Delta x_d}{2} + x_d m$$





$$\frac{1 - y_s}{2} = \frac{y_d - y_d m}{\Delta y_d}$$

$$y_d = (1 - y_s)\frac{\Delta y_d}{2} + y_d m$$



$$V = \begin{bmatrix} \frac{\Delta x_d}{2} & 0 & x_d m + \frac{\Delta x_d}{2} \\ 0 & -\frac{\Delta y_d}{2} & y_d m + \frac{\Delta y_d}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Etapas de la *pipeline* de vértices en OpenGL.

Matrices OpenGL

- OpenGL trabaja basicamente con dos matrices:
  - Matriz de la Vista-Modelo.
  - Matriz de Proyección.
- Tenemos que seleccionar sobre que matriz queremos actuar en cada momento:

```
glMatrixMode(GL_PROJECTION/GL_MODELVIEW)
```

- Las matrices OpenGL son vectores de 16 Glfloat. No obstante podemos utilizarlos en forma matricial recordando la forma de indexar.
- Si hacemos algo así:

```
Glfloat  matriz_ogl[4][4];
```

$$matriz\_ogl = \begin{bmatrix} m_o & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{bmatrix}$$

Etapas de la *pipeline* de vértices en OpenGL.

Matrices OpenGL

- Funciones para alteración de matrices
- Trabajo directo

```
glLoadIdentity() // Carga la matriz identidad sobre la
                 // matriz actic C
glLoadMatrix(GLfloat *M)  // Carga la matriz
                          //indicada  sobre la
                         // matriz activa C<-M

glMultMatrix(Glfloat *M) //realiza una
                         //premultiplicación
                         //de la matriz actual por M
                         // C<-CM
```

- Apilar y desapilar la matriz actual. Esto se debe hacer cundo no queremos que las transformaciones sobre un modelo afecten a otros.

```
glPushMatrix()
glPopMatrix()
```

Etapas de la *pipeline* de vértices en OpenGL.

Matrices OpenGL:Vista-Modelo

- Funciones para alteración de matrices del modelo
  - Traslación

```
glTranslatef{v}(Glfloat tx, Glfloat ty, Glfloat tz)
```

  - Rotación

```
glRotatef{v}(Glfloat angulo,Glfloat Vx, Glfloat Vy, Glfloat Vz)
```

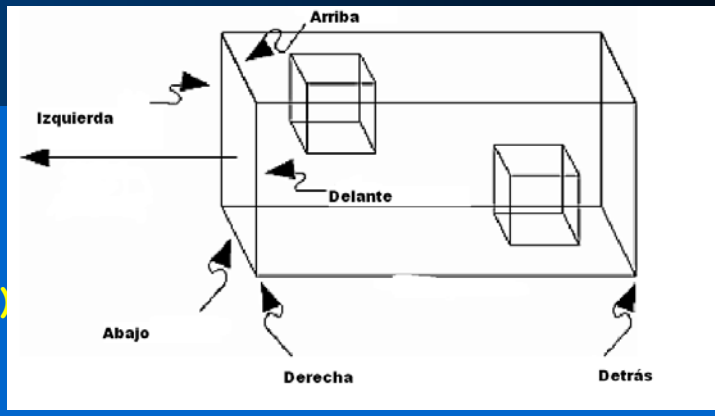
  - Escalado

```
glScalef{v}(Glfloat Sx, Glfloat Sy, Glfloat Sz)
```

- Composición de la matriz de la vista.
  - `gluLookAt(Glfloat Obsx, Glfloat Obsy, Glfloat Obsz,`
    `Glfloat Refx, Glfloat Refy, Glfloat Refz,`
    `Glfloat Vertx, Glfloat Verty, Glfloat Vertz)`

Etapas de la *pipeline* de vértices en OpenGL.

Matrices OpenGL:Proyección



- **Configuración de Proyección Ortogonal**

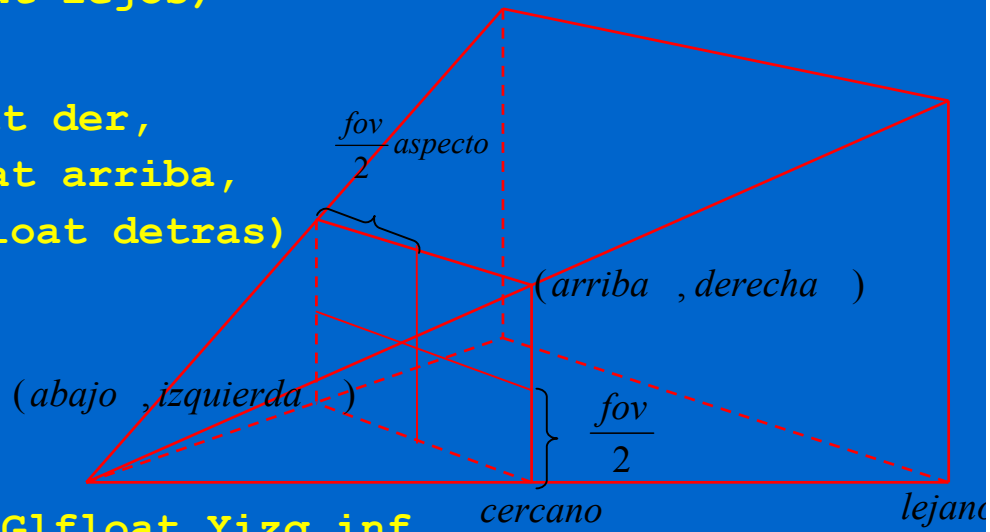
```
glOrtho(Glfloat izq, Glfloat der,
        Glfloat abajo, Glfloat arriba,
        Glfloat delante, Glfloat detras)
```

- **Configuración de la proyección en simétrica**

```
gluPerspective(Glfloat fov, Glfloat aspect,
        Glfloat cerca, Glfloat lejos)
```

- **Configuración general:**

```
glFrustum (Glfloat izq, Glfloat der,
        Glfloat abajo, Glfloat arriba,
        Glfloat delante, Glfloat detras)
```



- **Puerto de visión**

```
glViewport (Glfloat Xizq_inf, Glfloat Yizq_inf,
        Glfloat ancho, Glfloat alto)
```

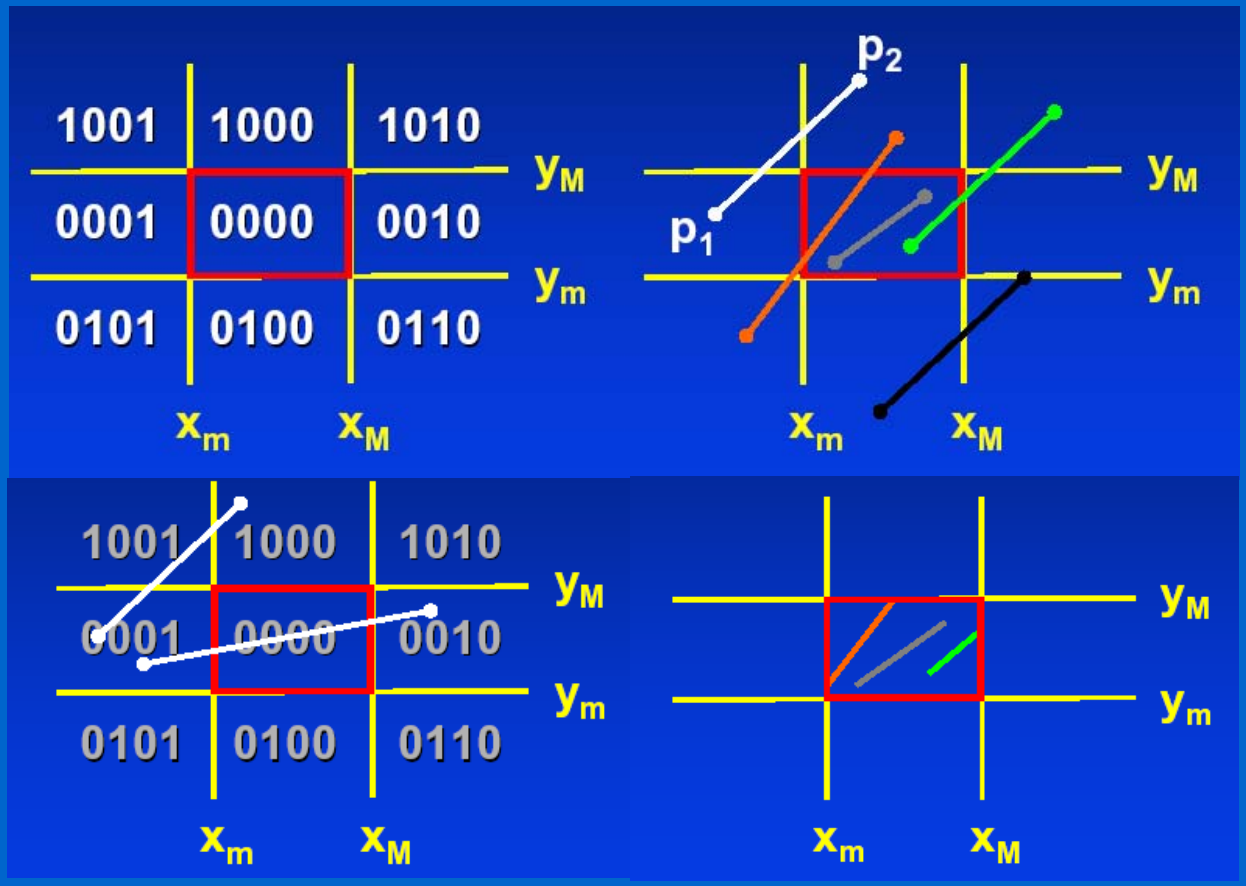
Etapas de la *pipeline* de Vértices

Ejemplo Tiovivo de las Teteras

Etapas de la *pipeline* de Pixels

Recorte Frente a la Ventana

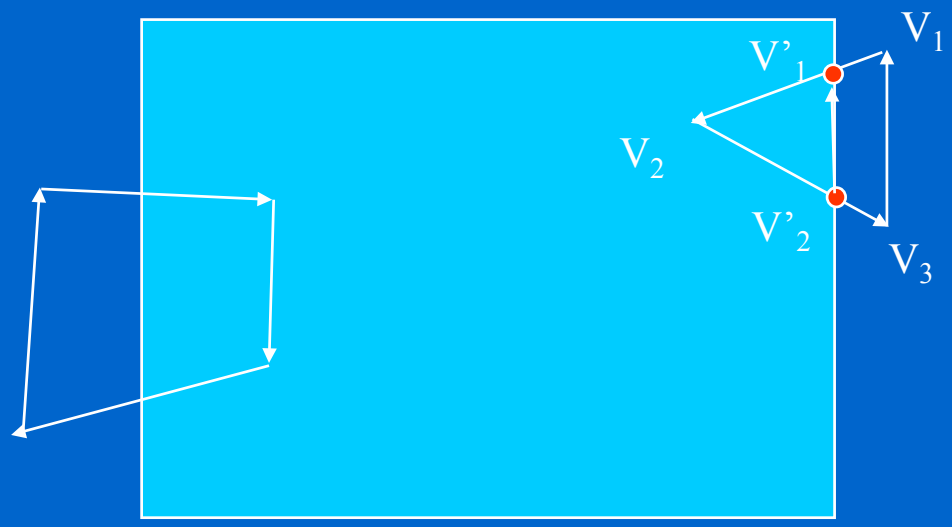
- **Cohen-Sutherland: basado en en códigos binarios y operaciones lógicas. Útil para determinar las líneas a recortar**



Etapas de la *pipeline* de Pixels

Recorte Frente a la Ventana

- **Sutherland-Hodgeman: recorte de polígonos frente a la ventana.**
- **Problemas con los polígonos convexos.**



$V_1V_2 \rightarrow V'_1 \rightarrow V'_1V_2$
$V_2V_3 \rightarrow V'_2 \rightarrow V_2V'_2$
$V'_2V'_1 \rightarrow V'_2V'_1$

Etapas de la *pipeline* de Pixels

- **Las últimas Etapas de la Pipeline de Vértices son:**
  - **Estimación Z-Buffer: Visto.**
  - **Relleno de Color: Visto.**
  - **Transparencia.**
  - **Aplicación de Texturas:Tema7**
- **Otras consideraciones:**
  - **Pipeline de Modo inmediato.**
  - **Pipeline de modo retenido.**