

Clase Auxiliar 4 CC52B Computación Gráfica

Transformaciones 3D – Proyecciones 3D

Auxiliar: Alvaro Neira
Fecha: 29.Abril.2005

Transformaciones 3D

$$P' = MP$$

Coordenadas homogéneas $\begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} \rightarrow P = \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} \quad h = 1$

Composición de transformaciones $M = M_n M_{n-1} \dots M_2 M_1$

$$S(Sx, Sy, Sz) = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T(Tx, Ty, Tz) = \begin{bmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rx(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Ry(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$Ref_x = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Ref_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Ref_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Ref}_{\text{origen}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Sh} = \begin{bmatrix} 1 & S_{xy} & S_{zx} & 0 \\ S_{yz} & 1 & S_{yz} & 0 \\ S_{zx} & S_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Positivo es CCW, centrado en el origen.

Problema 1 (P2, C1, 2003/02)

La función

static void dibCirc(Matrix4d T)

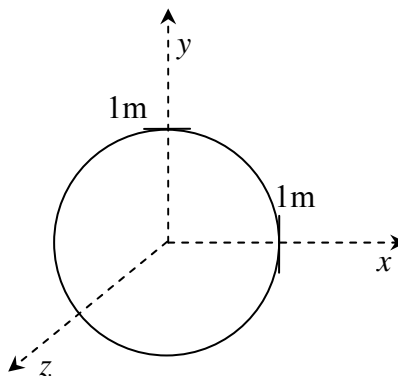
Recibe una Matriz de 4x4 elementos, que corresponde a una transformación.

Lo que hace este método es aplicar la transformación T a una circunferencia de 1m de radio y la muestra en la pantalla (el eje de la circunferencia coincide con el eje z).

Por ejemplo, si se llama a

$$\text{dibCirc} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

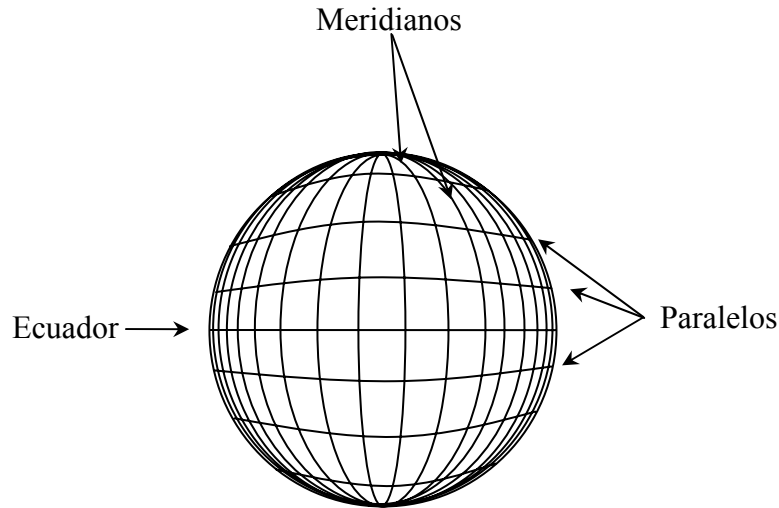
El resultado será una circunferencia de radio 1m situada como en la figura siguiente:



Se pide:

Utilizando la función `dibCirc(Matrix4d T)` mostrar en pantalla el wireframe de un globo terráqueo (de radio 1m), donde se distingan los meridianos y paralelos. En el globo dibujado deben distinguirse 35 meridianos y 7 paralelos (22,5°, 45°, 67,5° en latitud Norte y Sur, además del Ecuador).

Nota: se recomienda centrar el globo en (0,0,0).



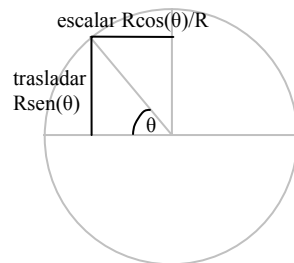
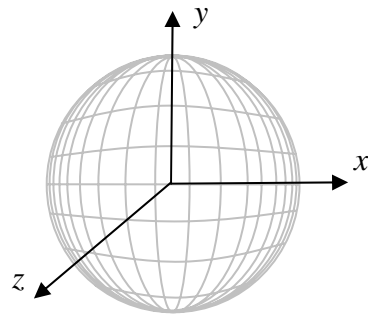
Solución:

Para dibujar los meridianos, se dibuja primero el círculo sin transformar y luego se rota sucesivamente en $360^\circ/18$ con respecto al eje y y se vuelve a dibujar:

```
for( $\theta = 0^\circ$ ;  $\theta < 180^\circ$ ;  $\theta += 360^\circ/18$ ) {
    dibCirc( $R_y(\theta)$ )
}
```

Para dibujar los paralelos:

1. Rotar en 90° con respecto al eje x
2. Escalar $\cos \theta$
3. Trasladar $\sin \theta$ en eje y
4. Variar θ entre $-67,5^\circ$ y $67,5^\circ$ con incremento de $22,5^\circ$.



```
for( $\theta = -67,5^\circ$ ;  $\theta \leq 67,5^\circ$ ;  $\theta += 22,5^\circ$ ) {
    dibCirc( $T(0, \sin \theta, 0)S(\cos \theta, \cos \theta \cos \theta)R_x(90^\circ)$ )
}
```

$T(0, \sin \theta, 0)S(\cos \theta, \cos \theta \cos \theta)R_x(90^\circ)$:

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

Entonces, dejando todo en pseudo-código:

```
for(θ = 0°; θ < 180°; θ += 360°/18){ //Meridianos
    dibCirc(
        [
            [cos θ  0  sen θ  0]
            [ 0      1      0      0]
            [-sen θ  0  cos θ  0]
            [ 0      0      0      1]
        ]
    )
}
for(θ = -67,5°; θ <= 67,5°; θ += 22,5°){ //Paralelos
    dibCirc(
        [
            [cos θ  0      0      0]
            [ 0      0     -cos θ  sen θ]
            [ 0      cos θ  0      0]
            [ 0      0      0      1]
        ]
    )
}
```

Generalizando para M meridianos, P paralelos, un globo de radio R centrado en x, y, z:

```
for(θ = 0°; θ < 180°; θ += 360°/M){ //Meridianos
    dibCirc(
        T(x,y,z) S(R,R,R)
        [
            [cos θ  0  sen θ  0]
            [ 0      1      0      0]
            [-sen θ  0  cos θ  0]
            [ 0      0      0      1]
        ]
    )
}
for(θ = -90° + 180°/(P+1); θ < 90°; θ += 180°/(P+1)){ //Paralelos
    dibCirc(
        T(x,y,z) S(R,R,R)
        [
            [cos θ  0      0      0]
            [ 0      0     -cos θ  sen θ]
            [ 0      cos θ  0      0]
            [ 0      0      0      1]
        ]
    )
}
```

Proyecciones 3D

Problema 2

(Alfredo Cofré)

Se le entrega una línea en 3D que va desde $P1 = (6,10,3)$ a $P2 = (-3,-5,2)$ y una pirámide de visión definida por los planos limitados por $z = x$, $z = -x$, $z = y$, $z = -y$. El plano de proyección se encuentra en $z = 1$. El VUP se orienta a lo largo de la recta $x = y$. Projete la línea en el plano y luego haga clipping sobre la ventana de proyección. ¿Cuáles son los puntos extremos de la línea sobre la ventana de proyección?

Solución:

La situación se describe en la figura 2.4. Dado que usaremos transformaciones 3D, convertimos los vectores a coordenadas homogéneas

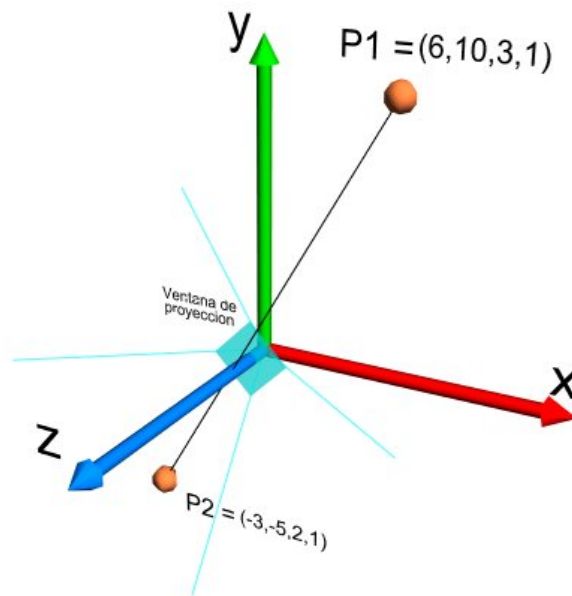


Figura 2.4

En primer lugar, debemos trasladar el origen de la pirámide de proyección hacia el origen del sistema de coordenadas global. En este caso, no es necesario, puesto que en nuestro problema ambos puntos coinciden. Luego, debemos rotar los puntos de la escena hasta que el vector normal a la ventana de proyección VPM sea colinear con el eje Z. Tampoco esto es necesario en este caso. A continuación, debemos rotar nuevamente de modo que el vector que indica la parte superior de la ventana de proyección se alinee con el eje Y (figura 2.5)

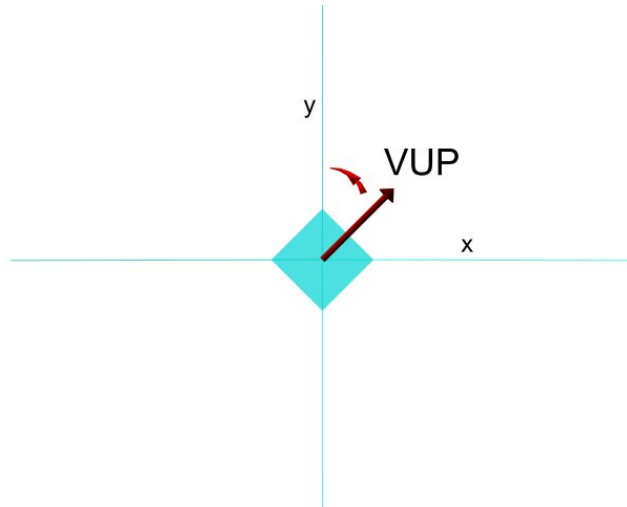


Figura 2.5: Rotación de VUP

Debemos rotar la ventana de proyección (y los elementos de la escena) en $-\pi/4$ c/r al eje Z. Al aplicar la matriz de rotación R a P1 y P2, tenemos que:

$$P1' = P1 * R = \begin{bmatrix} 6 & 10 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.71 & 0.71 & 0 & 0 \\ -0.71 & 0.71 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -2.84 & 11.36 & 3 & 1 \end{bmatrix}$$

$$P2' = P2 * R = \begin{bmatrix} -3 & -5 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0.71 & 0.71 & 0 & 0 \\ -0.71 & 0.71 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.42 & -5.68 & 2 & 1 \end{bmatrix}$$

A continuación, debemos normalizar la ventana de proyección de modo que sea de 1X1. Para esto, escalamos por la matriz S en una escala de $\frac{1}{\sqrt{2}}$, puesto que en estos momentos la ventana es de $\sqrt{2} \times \sqrt{2}$:

$$P1' = P1' * \begin{bmatrix} 1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 8 & 2.1 & 1 \end{bmatrix}$$

$$P2' = P2' * \begin{bmatrix} 1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -4 & 1.4 & 1 \end{bmatrix}$$

Ahora, proyectamos. Primero usamos el teorema de Thales para encontrar la intersección de las rectas que van de $P1'$ y $P2'$ al origen con el plano de proyección (figura 2.6):

$$\frac{P1'(z)}{P1'(x)} = \frac{P1''(z)}{P1''(x)} \Rightarrow \frac{2.1}{-2} = \frac{1}{P1''(x)} \Rightarrow P1''(x) = -0.95$$

$$\frac{P2'(z)}{P2'(x)} = \frac{P2''(z)}{P2''(x)} \Rightarrow \frac{1.4}{1} = \frac{1}{P2''(x)} \Rightarrow P2''(x) = 0.71$$

$$\frac{P1'(z)}{P1'(y)} = \frac{P1''(z)}{P1''(y)} \Rightarrow \frac{2.1}{8} = \frac{1}{P1''(y)} \Rightarrow P1''(y) = 3.81$$

$$\frac{P2'(z)}{P2'(y)} = \frac{P2''(z)}{P2''(y)} \Rightarrow \frac{1.4}{-4} = \frac{1}{P2''(y)} \Rightarrow P2''(y) = -2.86$$

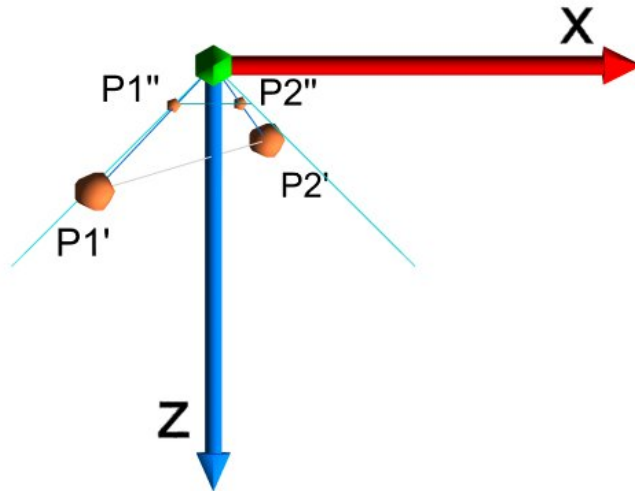


Figura 2.6: Proyección en el plano XZ

Entonces, $P1'' = (-0.95, 3.81, 1, 1)$ y $P2'' = (0.71, -2.86, 1, 1)$. Ahora hacemos clipping sobre la ventana de proyección (figura 2.7):

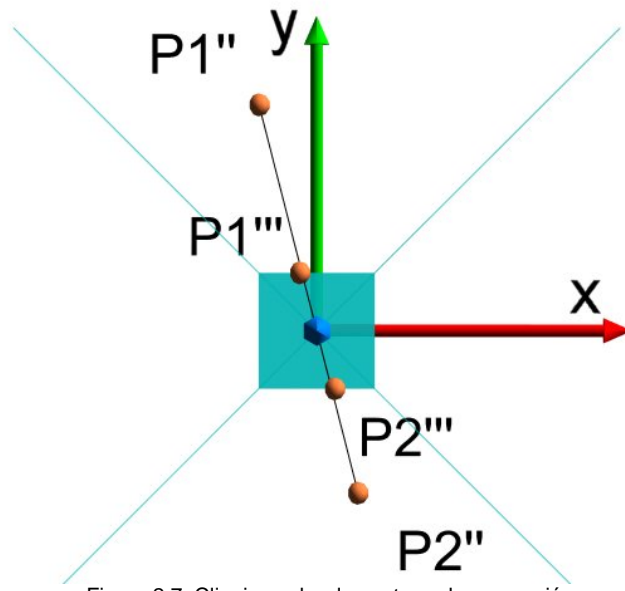


Figura 2.7: Clipping sobre la ventana de proyección

Trivialmente, vemos que debemos hacer clipping sobre $y = 1$ y $y = -1$. En el caso general, debemos usar Cohen-Sutherland u otro algoritmo de clipping. Las ecuaciones paramétricas de la recta que va entre $P1'''$ y $P2'''$ son:

$$\begin{aligned}x &= -0.95 + 1.66 t \\y &= 3.81 - 6.64 t\end{aligned}$$

$$\text{Si } y = 1, \Rightarrow 1 = 3.8 - 6.64 t \Rightarrow -(1+3.8)/6.64 = t = 0.421$$

$$\Rightarrow x = -0.95 + 1.66 * 0.42 = -0.25$$

$$\text{Si } y = -1, \Rightarrow -1 = 3.8 - 6.64 t \Rightarrow -(-1+3.8)/6.64 = t = 0.723$$

$$\Rightarrow x = -0.95 + 1.66 * 0.72 = 0.25$$

En la ventana de proyección (2D), la recta está limitada por los puntos $P1''' = (-0.25, 1)$ y $P2''' = (0.25, -1)$.