

## APÉNDICES

### APÉNDICE A.- Transformaciones de Objetos 3D

Las transformaciones en 3D pueden ser representadas por matrices de 4X4 para representar las coordenadas homogéneas de los puntos en tres dimensiones. De esta manera, en lugar de representar un punto (X, Y, Z), se representa como (X, Y, Z, W). La representación estándar es con  $W = 1$ ,

El sistema de coordenadas en 3D con el sistema de la mano derecha es:

Para el eje en:	Dirección de rotación positiva
X	Y a Z
Y	Z a X
Z	X a Y

La matriz de rotación sobre el eje X es la siguiente

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

La matriz de rotación sobre el eje Y es la siguiente

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

La matriz de rotación sobre el eje Z es la siguiente

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

Por rapidez, cuando se requiere multiplicar un dato en los tres ejes se puede pre multiplicar la matriz en los tres ejes:

$$R_x(\alpha) R_y(\beta) R_z(\delta) = \begin{pmatrix} c[\beta]*c[\delta] & -(c[\beta]*s[\delta]) & s[\beta] & 0 \\ c[\delta]*s[\alpha]*s[\beta]+c[\alpha]*s[\delta] & c[\alpha]*c[\delta]-s[\alpha]*s[\beta]*s[\delta] & -(c[\beta]*s[\alpha]) & 0 \\ -(c[\alpha]*c[\delta]*s[\beta]+s[\alpha]*s[\delta]) & c[\delta]*s[\alpha]+c[\alpha]*s[\beta]*s[\delta] & c[\alpha]*c[\beta] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

donde

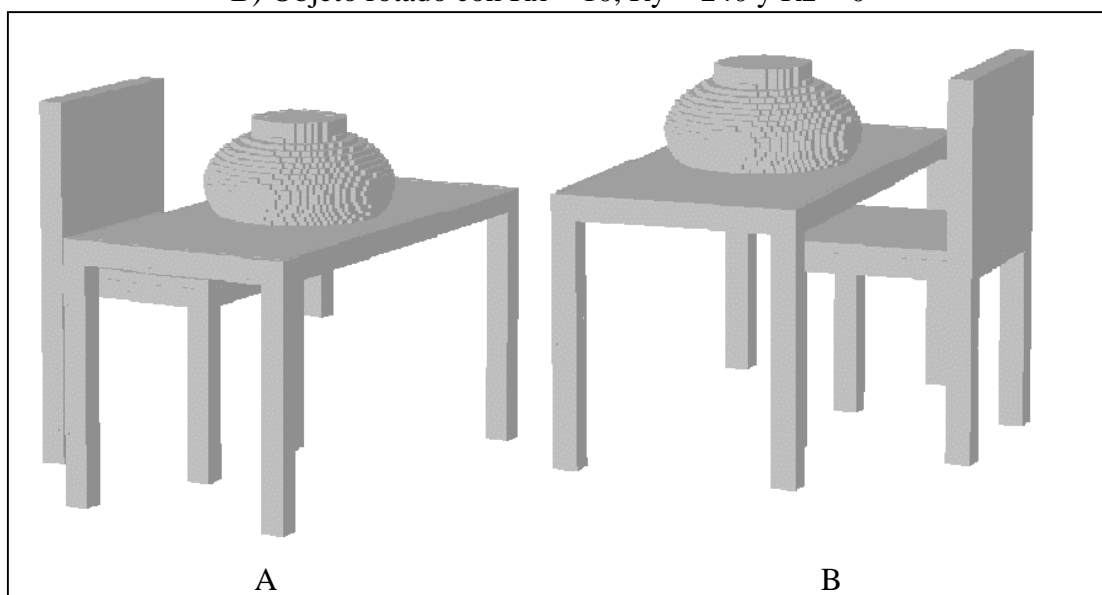
c es coseno

s es seno

Esta matriz de rotación sobre los tres ejes al ser multiplicada por los datos, en este caso, los vértices de cada octante del objeto tridimensional, proporcionan las nuevas coordenadas, es decir el octante rotado.

En la figura A.1 se muestra un objeto con dos rotaciones diferentes, permitiendo ver desde dos puntos de vista al mismo objeto.

**Figura A.1** A) Objeto rotado con  $R_x = 10$ ,  $R_y = 40$  y  $R_z = 0$ .  
B) Objeto rotado con  $R_x = 10$ ,  $R_y = 240$  y  $R_z = 0$



"Elaboración propia"

La matriz de translación es:

$$\text{Trans}(X,Y,Z) = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

donde:

$T_x$  es Translación en X

Ty es Translación en Y

Tz es Translación en Z.

Para mostrar la perspectiva de un punto dado P(X, Y, Z):

$$\text{Persx} = (X * \text{distancia}) / (\text{distancia} - Z)$$

$$\text{Persy} = (Y * \text{distancia}) / (\text{distancia} - Z)$$

Donde:

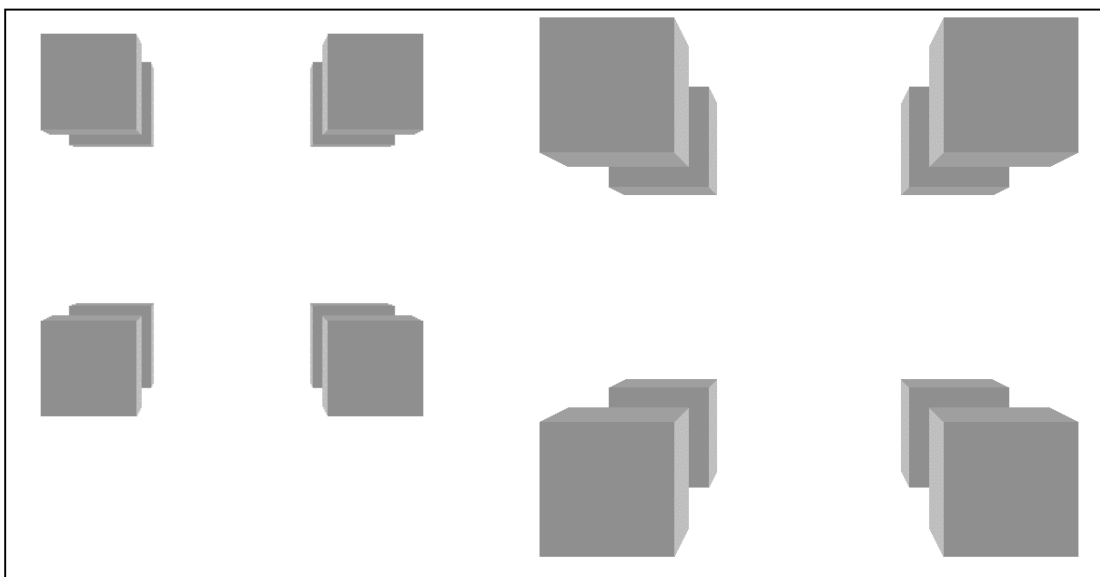
Persx es la perspectiva en X

Persy es la perspectiva en Y

distancia es la distancia entre el punto de vista del usuario y el punto P.

La matriz de translación en los tres ejes al ser multiplicada por los datos, en este caso, los vértices de cada octante del objeto tridimensional, proporcionan las nuevas coordenadas, es decir el octante trasladado. En la figura A.2 se muestra un objeto desde dos distancias diferentes en Z,

**Figura A.2** Un objeto visualizado con una translación en Z



"Elaboración propia"

## APÉNDICE B.- Visualización de octantes

Para visualizar un octante rotado se utiliza la matriz inversa  $(R_x(\alpha) R_y(\beta) R_z(\delta))^{-1}$

$$(R_x(\alpha) R_y(\beta) R_z(\delta))^{-1} =$$

$$\begin{pmatrix} c[b]*c[d] & c[d]*s[a]*s[b]+c[a]*s[d] & -(c[a]*c[d]*s[b])+(s[a]*s[d]) & 0 \\ -(c[b]*s[d]) & c[a]*c[d]-(s[a]*s[b]*s[d]) & (c[d]*s[a])+(c[a]*s[b]*s[d]) & 0 \\ s[b] & -(c[b]*s[a]) & c[a]*c[b] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

la cuál se multiplica por la matriz

$$\begin{pmatrix} 0 \\ 0 \\ d \\ 1 \end{pmatrix}$$

Dando un resultado

$$\begin{pmatrix} c[b]*c[d] & c[d]*s[a]*s[b]+c[a]*s[d] & -(c[a]*c[d]*s[b])+(s[a]*s[d]) & 0 \\ -(c[b]*s[d]) & c[a]*c[d]-(s[a]*s[b]*s[d]) & (c[d]*s[a])+(c[a]*s[b]*s[d]) & 0 \\ s[b] & -(c[b]*s[a]) & c[a]*c[b] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ d \\ 1 \end{pmatrix} =$$

Resultado =

$$\begin{pmatrix} \text{Resulx} \\ \text{Resuly} \\ \text{Resulz} \\ 1 \end{pmatrix}$$

Con el resultado de esta multiplicación se puede conocer que cara del octante se visualiza.

A continuación se muestra una subrutina, la cual determina las caras vistas de un octante.

```
Caras(int [] c,int [] Octante,int ancho)
{
    c[0]=c[1]=c[2]=7;
    Si(resulz<=Octante[z])                c[0] = atras
    Si no: Si(resulz>=Octante[z]+ancho)    c[0] = frente
    Si(resuly<=Octante[y])                c[1] = abajo
    Si no: Si(resuly>=Octante[y]+ancho)    c[1] = arriba
    Si(resulx<=Octante[x])                c[2] = izquierda
    Si no: Si(resulx>=Octante[x]+ancho)    c[2] = derecha
}
```

La siguiente subrutina muestra el recorrido del árbol octal de un objeto definido en CSG

```
recorrer(Primitiva p,Octante,int ancho,int precision)
{
    int [] orden=new int[8];
    int [] S=new int[3];
    Si (octante_Color = negro)
        Visualizar_Caras(Octante,ancho);
    Si no: Si(octante_Color = gris && ancho>=precision)
    {
        ancho=ancho/2;
        for(int i=0;i<3;i++)
        {
            S[i]=(int)(resul[i]-(Octante[i]+ancho/2));
            Si(S[i]<0) S[i]=0;
            Si no: Si(S[i]>0) S[i]=1;
        }
    }
}
```

```

    recorrido(orden,4*S[2]+2*S[1]+S[0]);
    for(int k=0;k<8;k++)
    {
        v=Desplazar(mundo,orden[k],ancho);
        recorrer(g,p,v,ancho,precision);
    }
} //Si no:
} //recorrer

```

Las subrutinas recorrido y Desplazar son utilizadas en la subrutina recorrer.

```

recorrido(int []orden,int op)
{
    Si(op = 0)          orden = {7,6,5,3,4,2,1,0};
    Si no: Si(op = 1)   orden = {6,7,4,2,5,3,0,1};
    Si no: Si(op = 2)   orden = {5,4,7,1,6,0,3,2};
    Si no: Si(op = 3)   orden = {4,5,6,0,7,1,2,3};
    Si no: Si(op = 4)   orden = {3,2,1,7,0,6,5,4};
    Si no: Si(op = 5)   orden = {2,3,0,6,1,7,4,5};
    Si no: Si(op = 6)   orden = {1,0,3,5,2,4,7,6};
    Si no: Si(op = 7)   orden = {0,1,2,4,3,5,6,7};
}

```

```

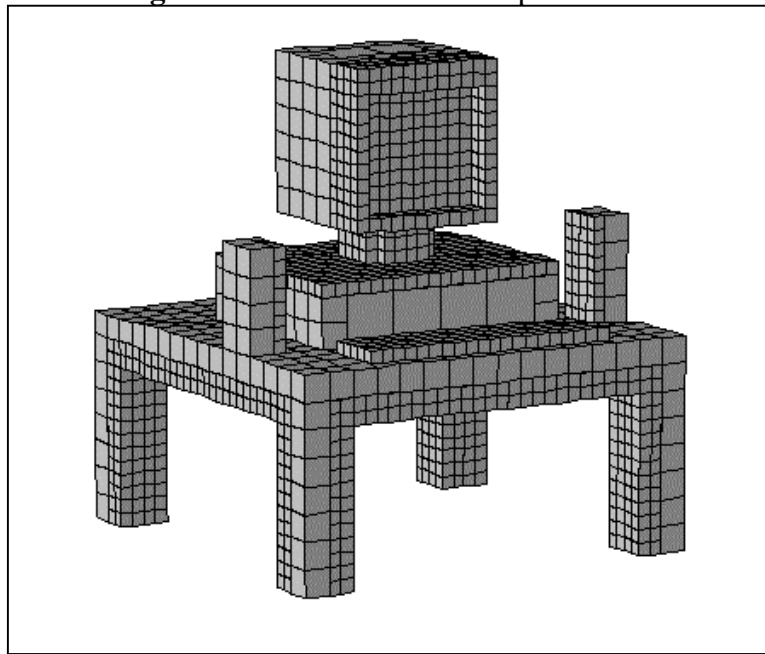
Desplazar(int [] coordenadas,int K,int dist)
{
    int [] vector= new int[3];
    vector[0]=coordenadas[0];
    vector[1]=coordenadas[1];
    vector[2]=coordenadas[2];
    Si(K%2==1) vector[0]=coordenadas[0]+dist;
    K=(int)(K/2);
    Si(K%2==1) vector[1]=coordenadas[1]+dist;
    K=(int)(K/2);
    Si(K%2==1) vector[2]=coordenadas[2]+dist;
    K=(int)(K/2);
    return(vector);
}

```

}  
}

En la Figura B.1 se muestra un objeto, en el cual se visualizan las izquierda, frontal y la cara superior de cada octante. Al hacer el recorrido del árbol octal de la forma 1, 0, 3, 5, 2, 4, 7, 6, los octantes de atrás se van tapando.

**Figura B.1** Recorrido de atrás para adelante



"Elaboración propia"