La cámara

P. J. Martín, A. Gavilanes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid

- La cámara se define con el comando:

```
gluLookAt(eyeX, eyeY, eyeZ, lookX, lookY, lookZ, upX, upY, upZ);
```

  donde `eye=(eyeX, eyeY, eyeZ)` es el punto donde se encuentra la cámara, `look=(lookX, lookY, lookZ)` es el punto al que mira la cámara, y `up=(upX, upY, upZ)` es el vector que indica cómo está orientada la cámara.

- Las coordenadas de `eye`, `look` y `up` se expresan usando el sistema de referencia global.
- La matriz de vista se coloca ejecutando:

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(eyeX,eyeY,eyeZ,lookX,lookY,lookZ,upX,upY,upZ);
```

- La matriz de vista pasa de coordenadas locales a coordenadas de cámara.

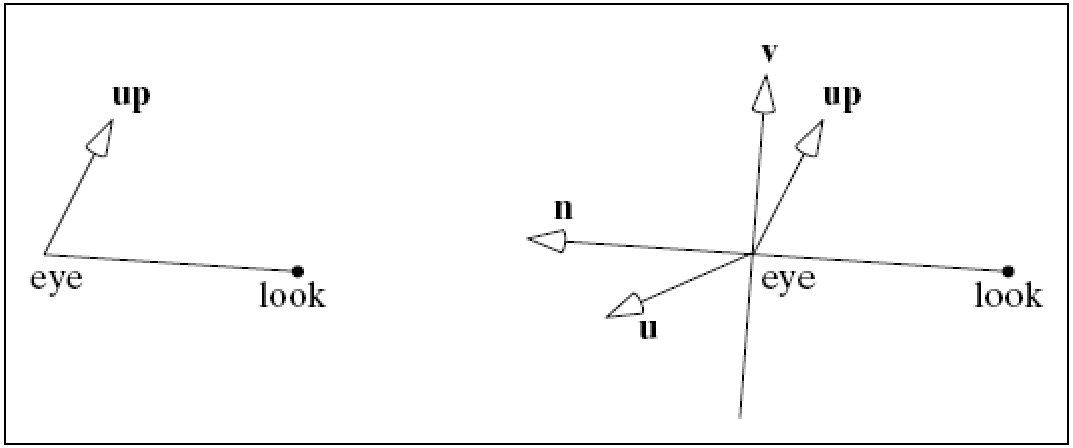
- A partir de `eye`, `look` y `up` se determina el sistema de coordenadas de la cámara:

  n=(eye-look).normalizar()

  u=(**up**×**n**).normalizar()

  v=**n**×**u**

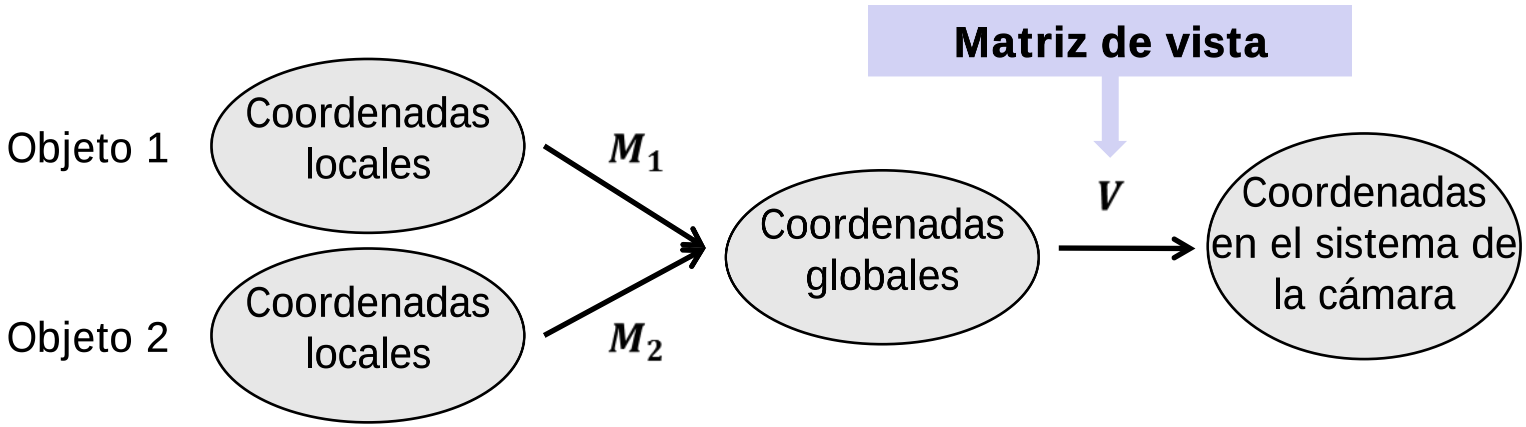
  Origen=eye



- La matriz que pasa del sistema de la cámara al sistema global es entonces:

$$\begin{pmatrix} u_x & v_x & n_x & e_x \\ u_y & v_y & n_y & e_y \\ u_z & v_z & n_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

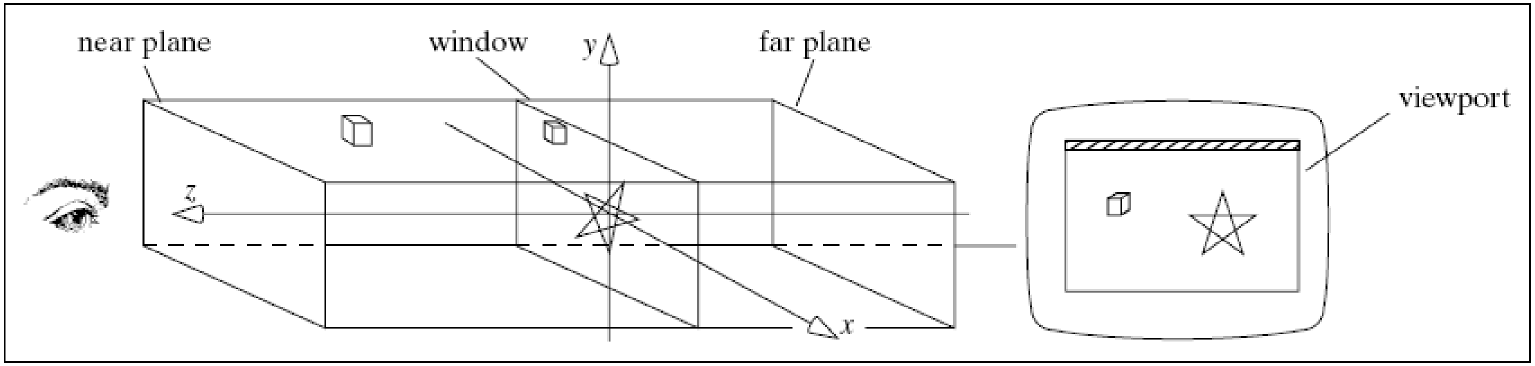
- Pero a la hora modelar interesa pasar de coordenadas globales a coordenadas en el sistema de la cámara:



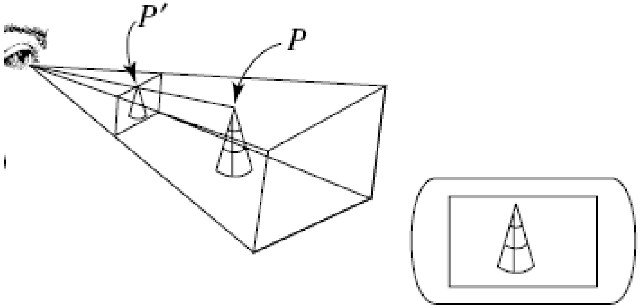
- La matriz que pasa del sistema global al sistema de la cámara es entonces:

$$V = \begin{pmatrix} u_x & v_x & n_x & e_x \\ u_y & v_y & n_y & e_y \\ u_z & v_z & n_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} u_x & u_y & u_z & -\overrightarrow{Oe} \cdot u \\ v_x & v_y & v_z & -\overrightarrow{Oe} \cdot v \\ n_x & n_y & n_z & -\overrightarrow{Oe} \cdot n \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Es la parte de la escena que es visible por la cámara.
- El volumen de vista es proyectado y la proyección es mostrada en el puerto de vista.
- En OpenGL, hay dos formas predefinidas de proyectar el volumen de vista: la proyección ortogonal



y la proyección perspectiva.

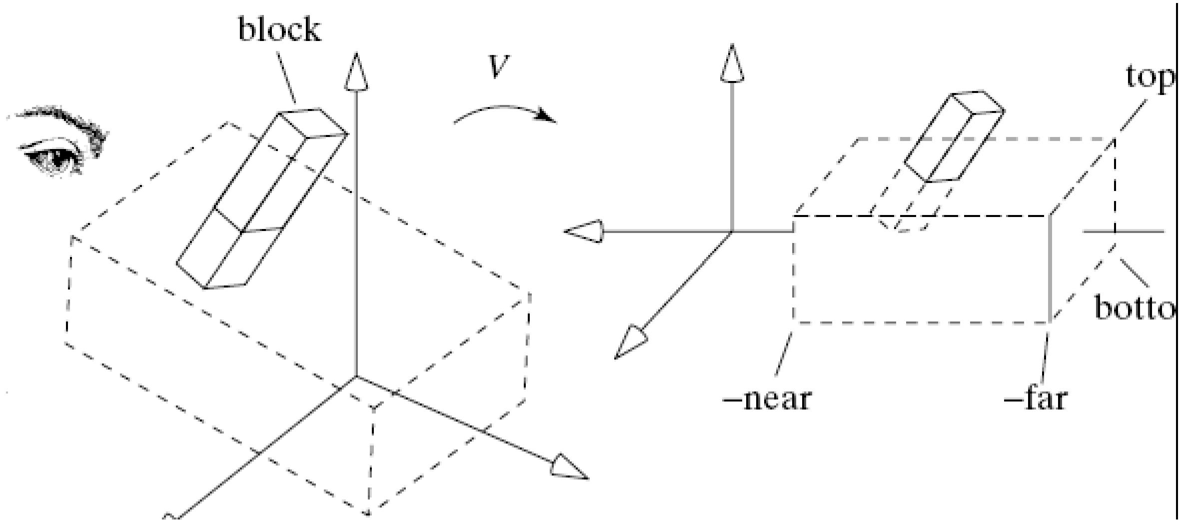


- La proyección ortogonal se obtiene con el comando:

```
glOrtho(left, right, bottom, top, near, far);
```

donde `left, right, bottom, top` son coordenadas en los ejes `U y V` de la cámara, respec., y `near, far` son las distancias de la cámara al plano cercano y lejano, respec., medidas en la parte negativa del eje `N`.

- Los límites del volumen de vista se expresan en el sistema de la cámara.
- En la proyección ortogonal, el volumen de vista es un paralelepípedo.



- La matriz de proyección es almacenada en OpenGL con el bloque de comandos:

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(left, right, bottom, top, near, far);
```

- El comando:

```
gluOrtho2D(left, right, bootom, top);
```

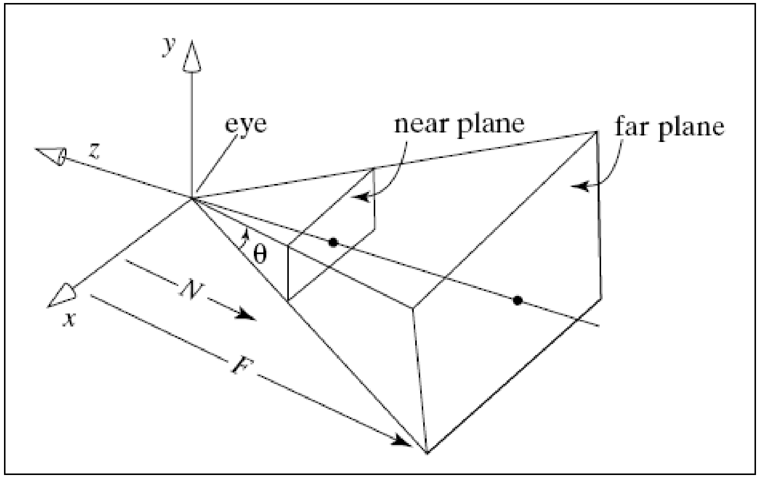
equivale a `glOrtho(left, right, bottom, top, -1, 1);`

- La proyección perspectiva se obtiene con alguno de los comandos siguientes:

```
glFrustum(left, right, bottom, top, near, far);
gluPerspective(fovy, aspect, near, far);
```

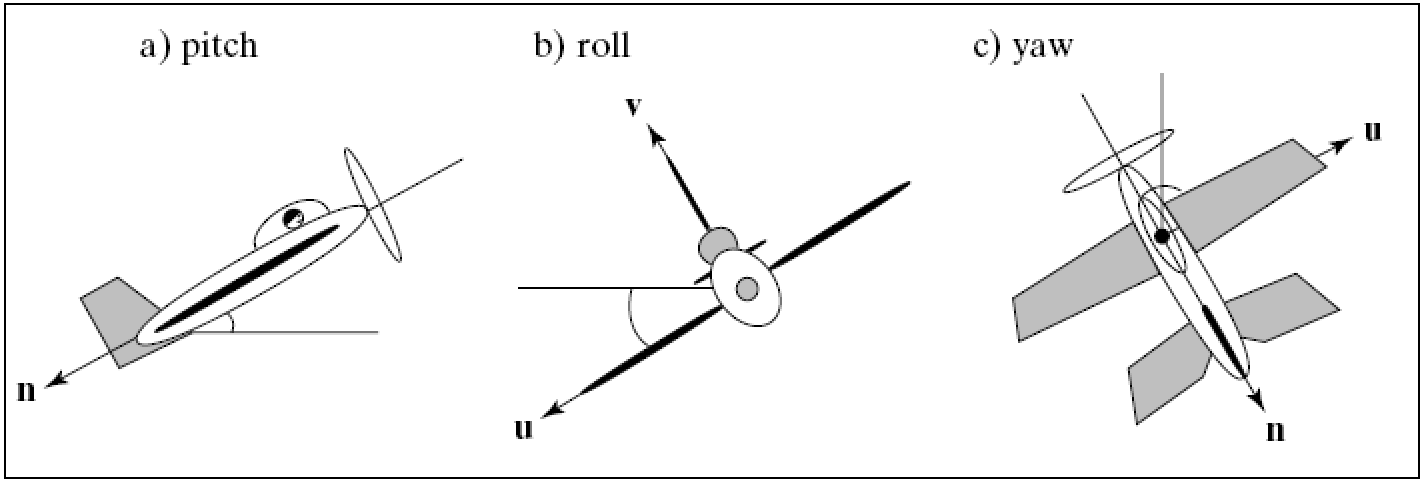
donde `left, right, bottom, top, near, far` se definen como en la proyección ortogonal; `fovy` es la apertura en el eje `V`, y `aspect` es la proporción de las dimensiones del plano cercano (=plano lejano).

- En la proyección perspectiva, el volumen de vista es una pirámide truncada cuyo ápice es el ojo de la cámara. En el caso del segundo comando, la pirámide es simétrica con respecto a los ejes `U y V` de la cámara.



- Zoom por variación de los parámetros `top` o `bottom` en la proyección ortogonal.
- Zoom por variación del parámetro `fovy` en la proyección perspectiva.
- Acercamiento de la cámara sobre el eje `N` de la cámara, en la proyección ortogonal.
- Acercamiento de la cámara sobre el eje N de la cámara, en la proyección perspectiva.

- Rotaciones con respecto a los ejes del sistema de coordenadas de la cámara.



- El movimiento `pitch` es una rotación con respecto al eje `U`
- El movimiento `yaw` es una rotación con respecto al eje `V`
- El movimiento `roll` es una rotación con respecto al eje `N`

- Atributos:
  - Los que permiten definir la orientación y el sistema de coordenadas de la cámara

    ```
    PV3D *eye, *look, *up, *u, *v, *n;
    ```
  - Los que permiten definir las dimensiones del volumen de vista

    ```
    Gldouble left, right, top, bottom,
             near, far, fovy, aspect;
    ```
- Constructora:
  - 1.- Recibe el valor para `eye,look,up` de los parámetros
  - 2.- Define la matriz de vista con el comando `gluLookAt(…)`
  - 3.- Da valor a las dimensiones del volumen de vista
  - 3.- Define la matriz de proyección con el comando respectivo
- Una vez construida, la cámara se encarga de ejecutar `gluLookAt() y glOrtho()` en función de sus atributos.

- Llevaremos a cabo los movimientos de la cámara moviendo la cámara, no moviendo la escena.
- Los movimientos se pueden realizar de dos formas:
    I. Calculando los nuevos valores `eje`, `look` y `up` en coordenadas globales, y ejecutando luego `gluLookAt(…)`
        - Rotar la cámara alrededor del eje `X`: `look` y `up` no cambian, `eye` recorre los puntos de una circunferencia alrededor del eje `X.`
        - Mostrar la vista cenital de la escena: `look` no cambia, `eye` se sitúa sobre la escena y `up` se define para que la cámara pueda mostrar la escena.

II. Transformando el sistema de referencia de la cámara (mediante M):

1. Calcular las coordenadas del nuevo sistema de coordenadas de la cámara en función del sistema de cámara original:

$$\boldsymbol{u}' = Mu \quad \boldsymbol{v}' = Mv \quad \boldsymbol{n}' = Mn \quad \boldsymbol{e}' = Me$$

2. Expresar el nuevo sistema de cámara en coordenadas globales:

$$\boldsymbol{u}' = V^{-1}Mu \quad \boldsymbol{v}' = V^{-1}Mv$$
$$\boldsymbol{n}' = V^{-1}Mn \quad \boldsymbol{e}' = V^{-1}Me$$

$$V^{-1} = \begin{pmatrix} u_x & v_x & n_x & e_x \\ u_y & v_y & n_y & e_y \\ u_z & v_z & n_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. Calcular la matriz de vista correspondiente:

$$\boldsymbol{V}' = \begin{pmatrix} \boldsymbol{u'}_x & \boldsymbol{v'}_x & \boldsymbol{n'}_x & \boldsymbol{e'}_x \\ \boldsymbol{u'}_y & \boldsymbol{v'}_y & \boldsymbol{n'}_y & \boldsymbol{e'}_y \\ \boldsymbol{u'}_z & \boldsymbol{v'}_z & \boldsymbol{n'}_z & \boldsymbol{e'}_z \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{u'}_x & \boldsymbol{u'}_y & \boldsymbol{u'}_z & -\overrightarrow{\boldsymbol{Oe'}} \cdot \boldsymbol{u'} \\ \boldsymbol{v'}_x & \boldsymbol{v'}_y & \boldsymbol{v'}_z & -\overrightarrow{\boldsymbol{Oe'}} \cdot \boldsymbol{v'} \\ \boldsymbol{n'}_x & \boldsymbol{n'}_y & \boldsymbol{n'}_z & -\overrightarrow{\boldsymbol{Oe'}} \cdot \boldsymbol{n'} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{pmatrix}$$

4. Cargar la matriz resultante como la nueva matriz de vista:

```
glMatrixMode(GL_MODELVIEW);
glLoadMatrixf(V'); //16 datos enumerados por columnas
```

1. El nuevo eje $\boldsymbol{u}'$ expresado en el sistema de la cámara es:

$$\boldsymbol{u}'_{\boldsymbol{camera}} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ 0 \end{pmatrix}$$

2. El nuevo eje $\boldsymbol{u}'$ expresado en coordenadas globales es entonces:

$$\boldsymbol{u}'_{\boldsymbol{global}} = \begin{pmatrix} u_x & v_x & n_x & e_x \\ u_y & v_y & n_y & e_y \\ u_z & v_z & n_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix} u'_{camera} = \begin{pmatrix} u_x\cos\theta + v_x\sin\theta \\ u_y\cos\theta + v_y\sin\theta \\ u_z\cos\theta + v_z\sin\theta \\ 0 \end{pmatrix}$$

Del mismo modo se calculan el resto de elementos del nuevo sistema:

$$\boldsymbol{v}'_{\boldsymbol{global}} = \begin{pmatrix} -u_x\sin\theta + v_x\cos\theta \\ -u_x\sin\theta + v_x\cos\theta \\ -u_x\sin\theta + v_x\cos\theta \\ 0 \end{pmatrix} \quad \boldsymbol{n}'_{\boldsymbol{global}} = \begin{pmatrix} \boldsymbol{n_x} \\ \boldsymbol{n_y} \\ \boldsymbol{n_z} \\ \boldsymbol{0} \end{pmatrix} \quad \boldsymbol{e}'_{\boldsymbol{global}} = \begin{pmatrix} \boldsymbol{e_x} \\ \boldsymbol{e_y} \\ \boldsymbol{e_z} \\ \boldsymbol{0} \end{pmatrix}$$

3. A partir del sistema de la cámara $\{\boldsymbol{u}'_{\boldsymbol{global}}, \boldsymbol{v}'_{\boldsymbol{global}}, \boldsymbol{n}'_{\boldsymbol{global}}, \boldsymbol{e}'_{\boldsymbol{global}}\}$, se calcula la matriz de vista correspondiente.

- La cámara se transforma usando su propio sistema de referencia (=> movimiento de la cámara en primera persona).
- Transformación del sistema de referencia típicas:
  - ✓ Rotación sobre algunos de sus ejes: pitch (U), yaw (V) y roll (N).
  - ✓ Traslación a lo largo del eje N ($t = (0,0,k,0)$ en coordenadas de cámara).
- A partir del sistema de cámara actual $\{u, v, n, e\}$, se calcula el nuevo sistema de cámara $\{u', v', n', e'\}$. La siguiente tabla sintetiza el resultado de las cuentas explicadas anteriormente. Siempre se usan coordenadas globales.

Transformación del sistema de la cámara	u'	v'	n'	e'
Rotación sobre U	u	$\cos\theta\, v + \sin\theta\, n$	$-\sin\theta\, v + \cos\theta\, n$	e
Rotación sobre V	$\cos\theta\, u - \sin\theta\, n$	v	$\sin\theta\, u + \cos\theta\, n$	e
Rotación sobre N	$\cos\theta\, u + \sin\theta\, v$	$-\sin\theta\, u + \cos\theta\, v$	n	e
Traslación de vector $(0,0,k,0)$	u	v	n	$e + kn$