

¡Proyecciones y la Cámara Sintética!

Dr. Arno Formella

Informática Gráfica

Departamento de Informática, Universidad de Vigo

30 de abril de 2003

Lo que os espera

- introducción con dibujitos
- clasificación de las proyecciones
- un poco de la matemática detrás
- la cámara sintética
- OpenGL

algunos dibujos están sacados de la Red (Scott D. Anderson,
<http://http://cs.wellesley.edu/~cs307/>)

Objetivo

- visualizar entornos **3D** en dispositivos **2D**
- ejemplo de la vida real: cámara fotográfica
- solamente la **geometría**
- obviamente: hay que **reducir** el número de coordenadas

2

Sacar una foto: real y sintético

real	sintético
posicionar el tripodo y orientar la cámara	definir la geometría de la cámara sintética
seleccionar lente, enfoque, zoom	definir parámetros de proyección
elegir tamaño de la foto	definir tamaño de la ventana

3

El caso mas simple

4

Dualidad

- en vez de *mover* la cámara
- podemos mover los objetos
- o al revés
- (*mover*. trasladar, rotar, escalar, distorcionar ...)
- se aplica la **transformación inversa**

5

Cada cosa su nombre

Centro de Proyección: o observador, o posición de la cámara

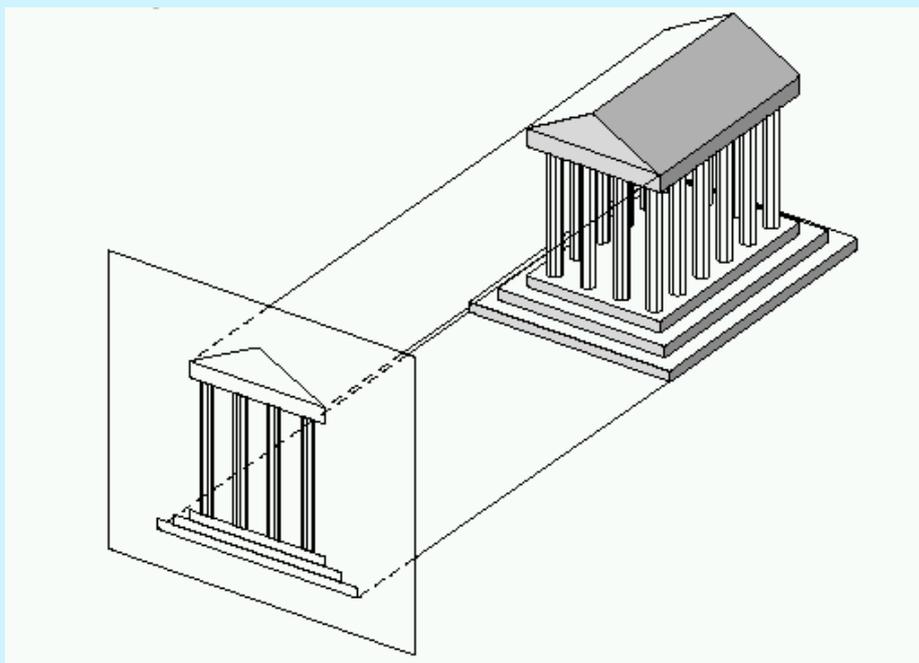
Proyectores: líneas rectas (virtuales) desde puntos de los objetos al centro de proyección

Plano de Proyección: superficie plano en el cual se realiza la proyección (intersecciones con los proyectores)

cosa divertida: para dibujar eso,
¡necesito una proyección ya!

6

Ejemplo: Proyección paralela



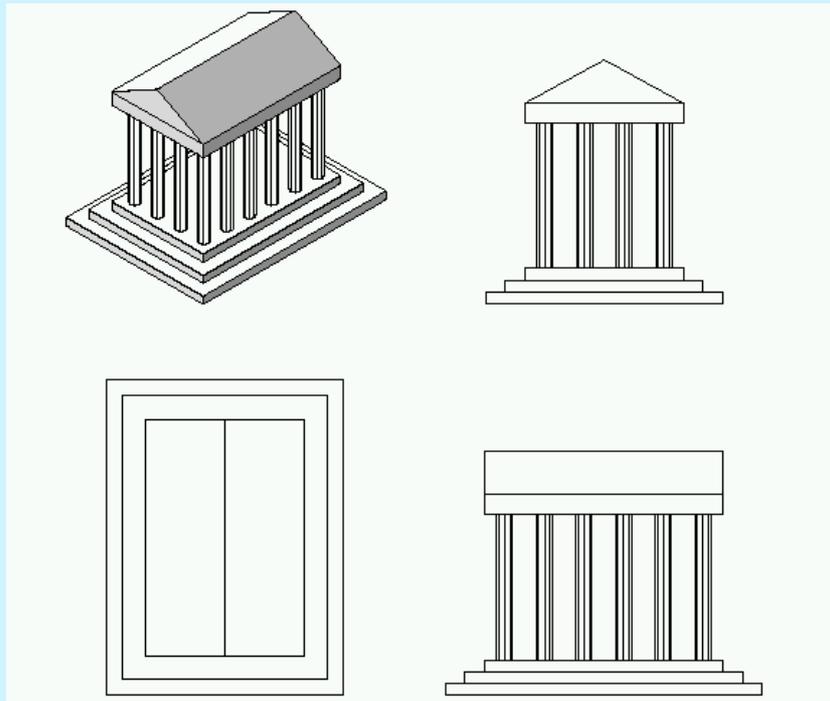
7

Clasificación de las proyecciones

- paralela
 - ortográfica
 - dimétrica, trimétrica, isométrica
 - oblicuo
- perspectiva
 - un punto, dos puntos, tres puntos

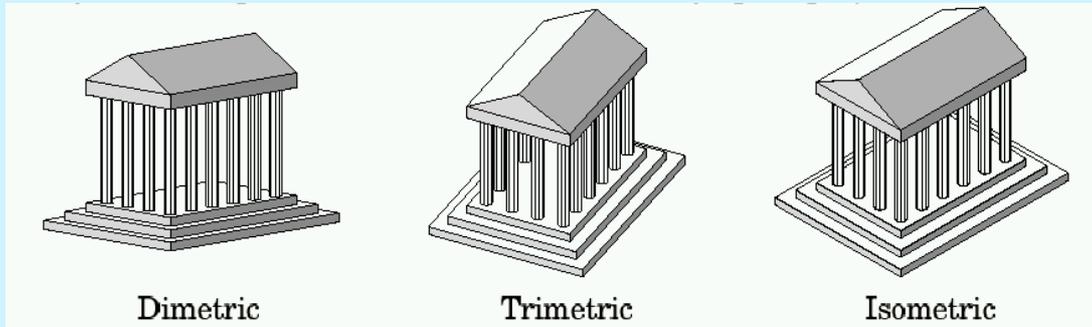
8

Ejemplos: Proyecciones usados en CAD



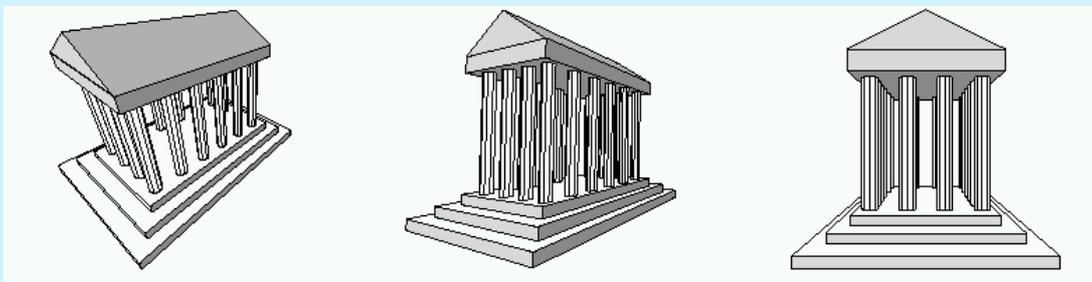
9

Ejemplos: Proyecciones ortográficas



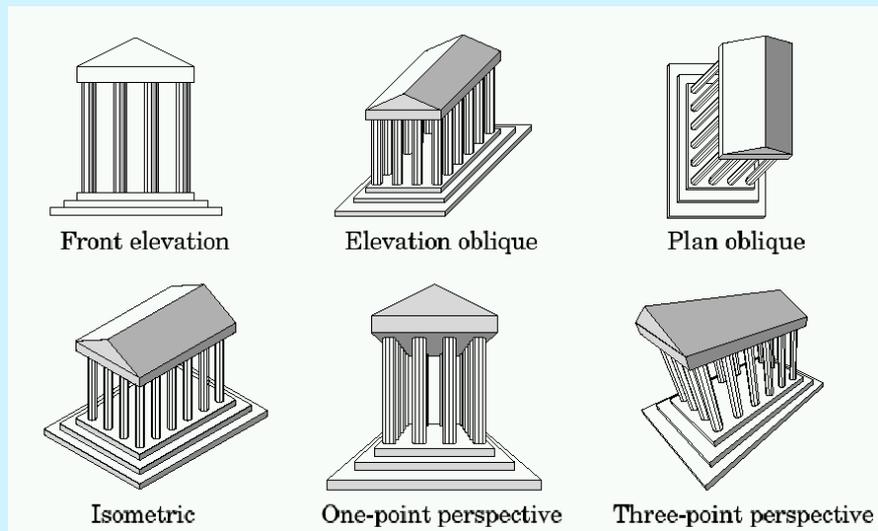
10

Ejemplos: Proyecciones perspectivas



11

Ejemplos: más Proyecciones



12

Características I (de V)

proyecciones paralelas:

- el centro de proyección está en el **infinito**
- los proyectores son **paralelas**
- si intersectan el plano de proyección
 - con ángulo **recto** (son perpendiculares): proyección **ortográfica**
 - con ángulo **oblicuo** (no son perpendiculares): proyección **oblicuo**

13

Características II (de V)

proyecciones paralelas:

- de todos los planos en el entorno que son paralelas al plano de proyección se mantienen **las distancias y los ángulos**
- las rectas paralelas en el entorno **permanecen paralelas** en la proyección
- se pueden realizar **mediciones exactas** con un escalado uniforme para cada eje

14

Características III (de V)

proyecciones paralelas:

- por eso, muy usados en
 - diseños de ingeniería
 - planos de arquitectura
 - CAD
- la visualización no se percibe como muy realista

15

Características IV (de V)

proyecciones perspectivas:

- mantienen
 - solamente los ángulos,
 - solamente en los planos paralelos al plano de proyección
 - solamente en la proyección perspectiva de un punto
- la visualización se percibe como más realista
- el tamaño de los objetos **disminuye** con la distancia

16

Características V (de V)

proyecciones perspectivas:

- existen hasta tres puntos en el plano de proyección donde se intersectan los paralelos a los ejes, llamados: **puntos de fuga** (en el sistema de coordenadas de la cámara)

17

Especificación de la cámara

Pizarra: camara

18

Especificación de la cámara

- se necesita especificar
 - la posición
(en el caso paralelo: infinito (o cualquier punto)),
 - la dirección de vista y
 - la orientación
- y el plano de proyección

19

Colocación del Plano de la Proyección

proyección paralela:

- se puede colocar el plano de proyección de tal manera
 - que intersecta un eje
 - que intersecta dos ejes (dimétrica)
 - que intersecta tres ejes (trimétrica, isométrica)
- del sistema de coordenadas de la cámara

20

Colocación del Plano de la Proyección

proyección perspectiva:

- se puede colocar el plano de proyección de tal manera
 - que intersecta un eje
 - que intersecta dos ejes
 - que intersecta tres ejes
- del sistema de coordenadas de la cámara

21

Recordamos: Cadena de Visualización

coordenadas	(dim)	uso
del objeto	(3D)	modelado
del mundo	(3D)	modelado/recorte
de la cámara	(3D)	recorte/caras ocultas
del plano de proyección	(2D)	imagen
de la vista	(2D)	visualización

22

Volumen de la Vista

- confinamos el entorno en un volumen

- recorte es más fácil

- elegimos un volumen
 - que permite operaciones **eficientes** y
 - que esté **asociado** con la cámara/tipo de proyección

23

Volumen de la vista: Frustum

Pizarra: Frustum

24

Especificación del frustum: paralelo

Pizarra: Frustum paralelo

25

Especificación del frustum: perspectivo

Pizarra: Frustum perspectivo

26

Especificación del frustum: más humano

Pizarra: Frustum humano

27

Especificación del frustum: más humano

razón de aspecto: proporción entre anchura y altura

campo de visión: ángulo de abertura en dirección arriba

distancias: al plano cercano y al plano lejano

proyección perspectiva simétrica de un punto

28

Coordenadas homogéneas

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \iff \begin{pmatrix} x/w \\ y/w \\ z/w \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \iff \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \xrightarrow{w \rightarrow 0} \begin{pmatrix} \infty \\ \infty \\ \infty \end{pmatrix}$$

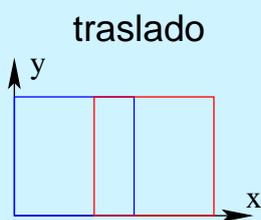
29

un poco de Matemáticas

y de Informática: OpenGL

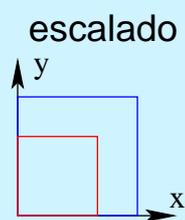
30

Recordamos: Transformaciones I (de II)



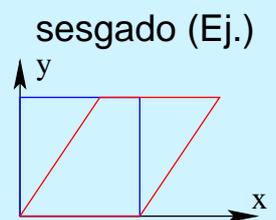
$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

`glTranslate(tx, ty, tz)`



$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

`glScale(sx, sy, sz)`

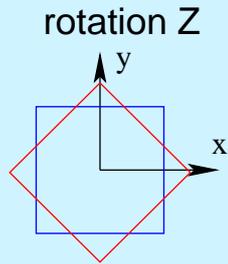


$$\begin{pmatrix} 1 & v_{xy} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & v_{xz} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

no hay
`glMultMatrix(...)`

31

Recordamos: Transformaciones II (de II)



$$\begin{pmatrix} \cos \phi_z & -\sin \phi_z & 0 & 0 \\ \sin \phi_z & \cos \phi_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

`glRotate(phi_z, 0, 0, 1)`

rotation Y

$$\begin{pmatrix} \cos \phi_y & 0 & \sin \phi_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi_y & 0 & \cos \phi_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

`glRotate(phi_y, 0, 1, 0)`

32

Recordamos: Transformaciones III (de III)

rotation X

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi_x & -\sin \phi_x & 0 \\ 0 & \sin \phi_x & \cos \phi_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

`glRotate(phi_x, 1, 0, 0)`

rotation general

$$\begin{pmatrix} u_{00} & u_{01} & u_{02} & 0 \\ u_{10} & u_{11} & u_{12} & 0 \\ u_{20} & u_{21} & u_{22} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ortonormal

no hay

`glMultMatrix(...)`
`glRotate(phi, ux, uy, uz)`

como transformar una rotación con matriz en una rotación con eje es un poco más complicado (se necesita quaterniones)

33

Sistema de Coordenadas de la Cámara

Pizarra: Cámara

34

Sistema de Coordenadas de la Cámara

- U' se puede especificar con cualquier punto que no esté en dirección V desde P
- $U = V \times (U' - P) \times V \quad N = L \times U$
- entonces, si normalizamos todos los vectores, (P, V, U, N) forma un sistema de coordenadas **ortonormal**

35

Coordenadas: Mundo \implies Cámara I (de III)

- perspectiva: se traslada para que la cámara se encuentre en el origen

$$T = \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

36

Coordenadas: Mundo \implies Cámara II (de III)

- se rota para que los ejes del mundo estén alineados con los ejes de la cámara

$$R = \begin{pmatrix} n_x & n_y & n_z & 0 \\ u_x & u_y & u_z & 0 \\ -v_x & -v_y & -v_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- la transformación final es:

$$M = R \cdot T$$

37

Coordenadas: Mundo \implies Cámara III (de III)

hace OpenGL automático

38

Frustum en OpenGL

- paralelo: `glOrtho(left, right, bottom, top, near, far)`
- perspectiva: `glFrustum(left, right, bottom, top, near, far)`
pero es solo proyección perspectiva de un punto
- alternativa: `gluPerspective(...)`
pero es solo proyección perspectiva simétrica de un punto

39

Frustum: Volumen canónico

- para que se puede realizar el recorte más fácil
- se transforma el frustum en un volumen canónico
- que es en OpenGL siempre un cubo
- entre las coordenadas -1 y 1 en todos los ejes
- (del sistema de coordenadas de la cámara)

40

Matrices de proyección de OpenGL

$$M_{par} = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

41

Esquema de una matriz de proyección de tres puntos

$$P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/d_x & 1/d_y & 1/d_z & 1 \end{pmatrix}$$

42

Transformación a la ventana

- del plano de proyección hay que **transformar** hacia la ventana
- se realiza con una simple transformación: **traslado** y escalado
- si no se quiere distorción: hay que **mantener razón de aspecto** de la proyección y razón de aspecto de la ventana
- `glViewport(x,y,w,h)` (en coordenadas del dispositivo)
- la transformación a píxeles realiza OpenGL automática

43

Las matrices de OpenGL

- OpenGL mantiene varias matrices
 - del modelo (MODELVIEW)
 - de la proyección (PROJECTION)
 - de las texturas (TEXTURE)
- cada tipo tiene asociado una **pila**

44

Las pilas de matrices de OpenGL

- `glLoadIdentity()` sustituye la cima con I
- `glPushMatrix()` duplica la cima
- `glPopMatrix()` elimina la cima
- `glLoadMatrix()` sustituye la cima
- `glMultMatrix()` multiplica la cima

45

Sacar una foto: real y sintético

real	sintético	OpenGL/glut
posicionar el tripodo y orientar la cámara	definir posición, punto de vista, y dirección arriba	<code>gluLookAt()</code>
seleccionar lente, enfoque, zoom	definir parámetros de proyección	<code>gluPerspective()</code>
elegir tamaño de la foto	definir tamaño de la ventana	<code>glViewport()</code>

es perspectiva con un punto

46

Efectos especiales

- profundidad de campo
- distorsión de los proyectores
- distorsión del plano de proyección

47

Conclusión

¡Id a las Prácticas!