



SIMULACIÓN Y PLANIFICACIÓN QUIRÚRGICA

Tema 3 – Modelos 3D: intro. y visualización



Francisco Morán Burgos
Grupo de Tratamiento de Imágenes

Dpto. Señales, Sistemas y Radiocomunicaciones
E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid

Índice

- Introducción
 - Modelado (*model[ing]*) vs. visualización (*rendering*)
 - Tipos de coordenadas
 - *Rendering pipeline*
 - Taxonomía de proyecciones
 - Perspectiva histórica de dispositivos de visualización

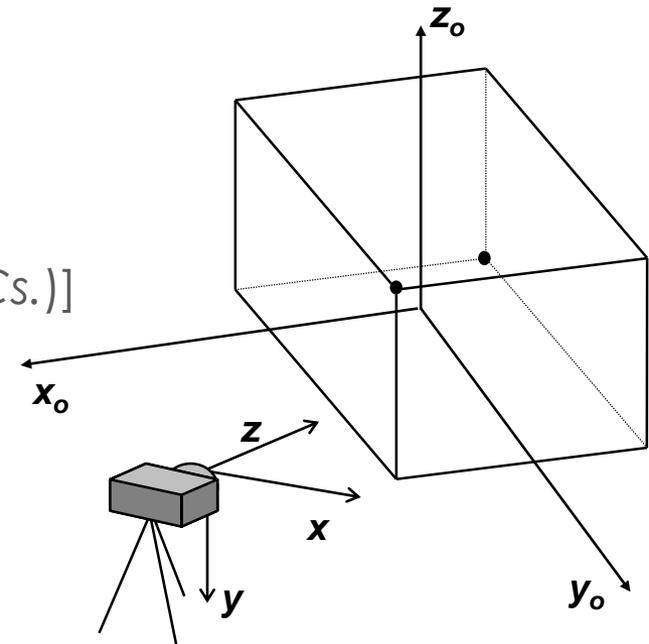
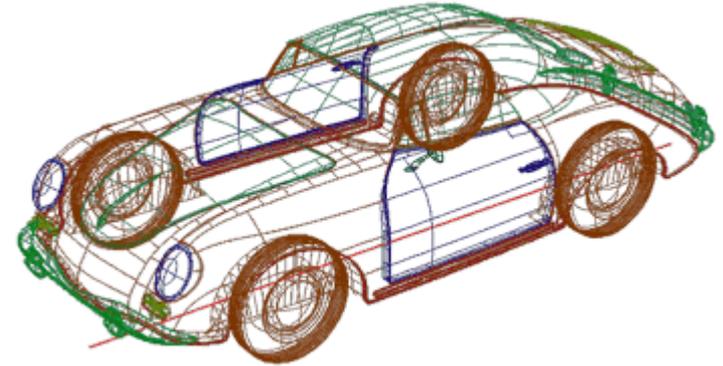
- Visualización
 - Eliminación de superficies ocultas
 - Iluminación y “sombreado” (interpolación del color)
 - Lujo: transparencias, sombras, reflexiones, refracciones, etc.
 - Filtrado para reducir *aliasing*

Introducción: modelado vs. visualización

- ▣ Modelado (*model[ing]*) ⇒ **tema 4**
 - ▣ Descripción de cada objeto 3D: forma y apariencia
 - ▣ Descripción de escena 3D: distintos objetos, luces, cámara...
 - ▣ [Descripción de animación]
 - ▣ [Descripción de interactividad (entre objetos y con usuario)]
 - ▣ Proceso muy laborioso para humanos
- ▣ Visualización (*rendering*)
 - ▣ Generación de imagen 2D a partir de modelo de escena 3D
 - ▣ Aproximación de leyes físicas: modelos de iluminación, reflexión, refracción...
 - ▣ Eliminación de partes ocultas, texturado, interpolación de color...
 - ▣ Aproximación de más leyes físicas: dinámica, biomecánica...
 - ▣ Proceso muy laborioso para máquinas

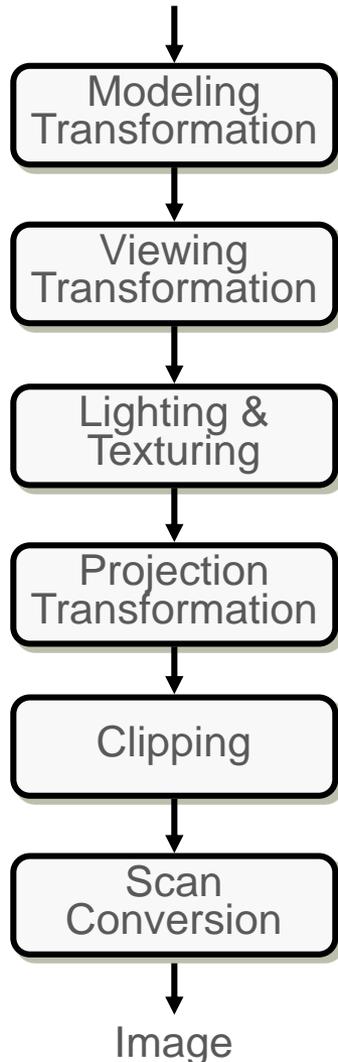
Introducción: tipos de coordenadas

- MCs (*Modelling Coordinates*)
 - 3D, alta precisión
 - Referencial LOCAL para cada objeto
- WCs (*World Coordinates*)
 - 3D, alta precisión
 - Referencial ABSOLUTO...
 - [... ligado a cámara \Rightarrow CCs (*Camera Cs.*)]
- DCs (*Device Coordinates*)
 - 2D, baja resolución
 - Referencial de la imagen



Introducción: *rendering pipeline*

3D Geometric Primitives



Transform into 3D world coordinate system

Transform into 3D camera coordinate system
Done with modeling transformation

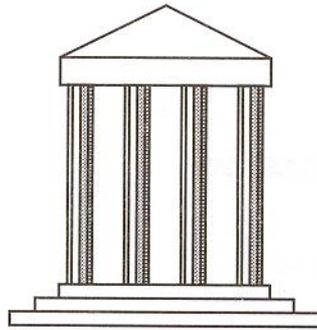
Illuminate according to lighting and reflectance
Apply texture maps

Transform into 2D screen coordinate system

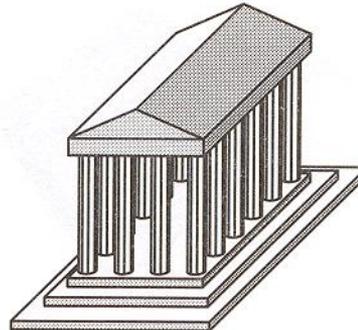
Clip primitives outside camera's view

Draw pixels (incl. HSR, shading, texturing...)

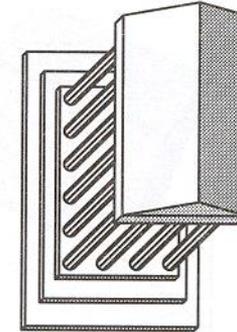
Introducción: zoo de proyecciones



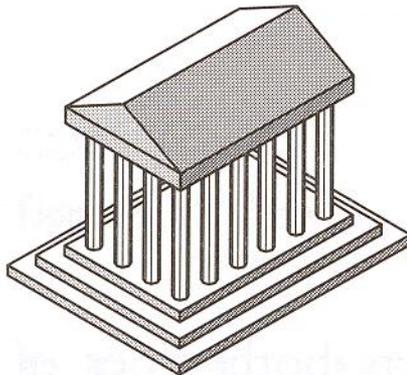
Front elevation



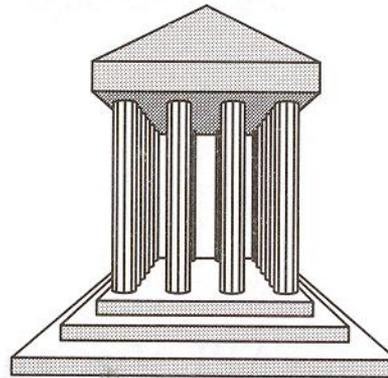
Elevation oblique



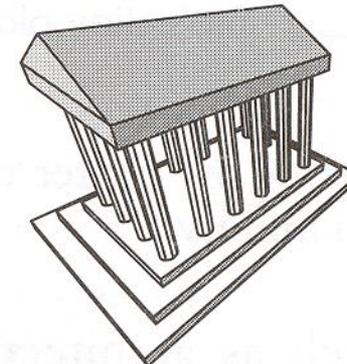
Plan oblique



Isometric



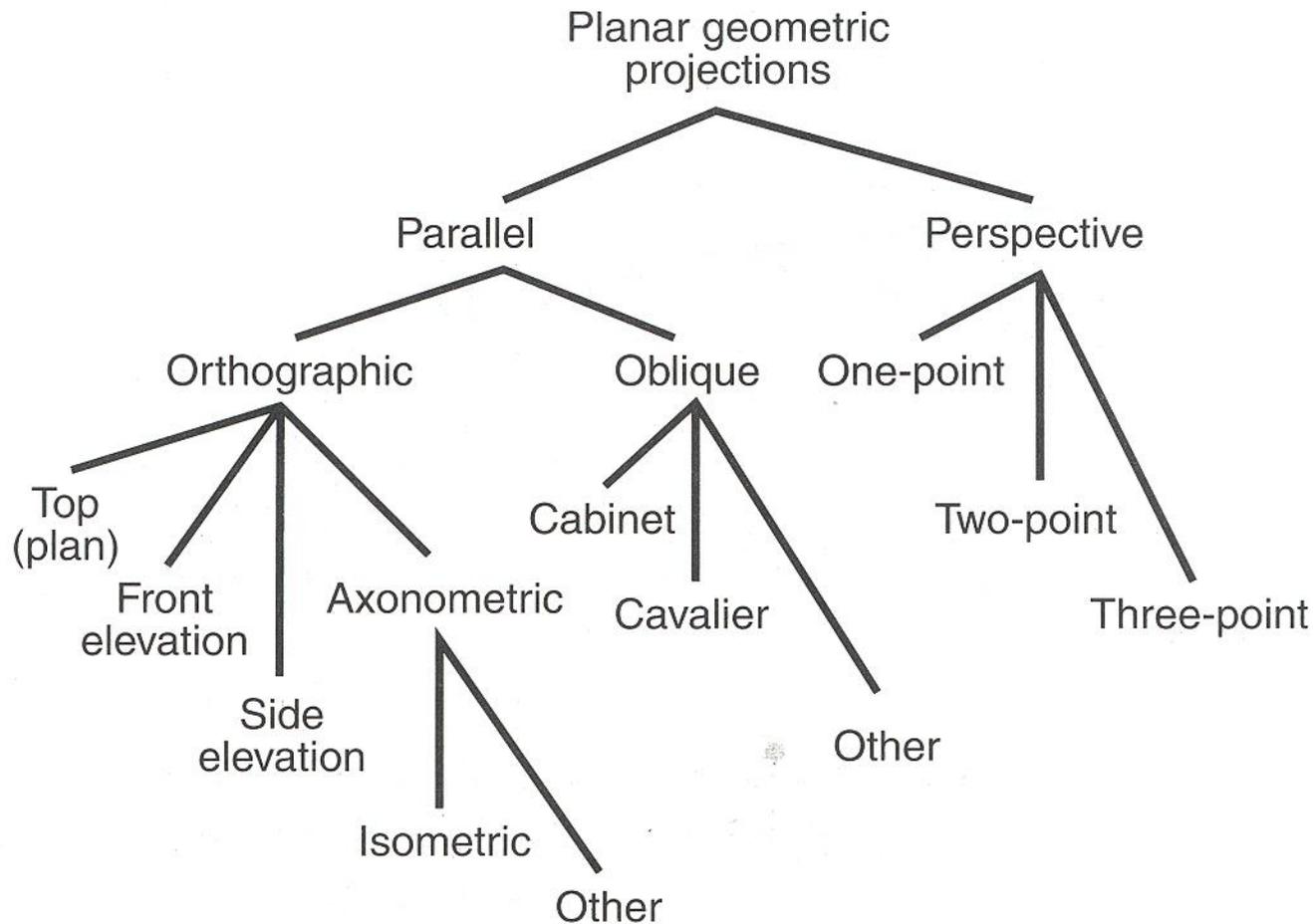
One-point perspective



Three-point perspective

Imágenes y taxonomía tomadas de "biblia" de Gráficos [3D] de J.D. Foley et al.:
Computer Graphics: Principles and Practice (2nd ed. in C), Addison-Wesley, 1997

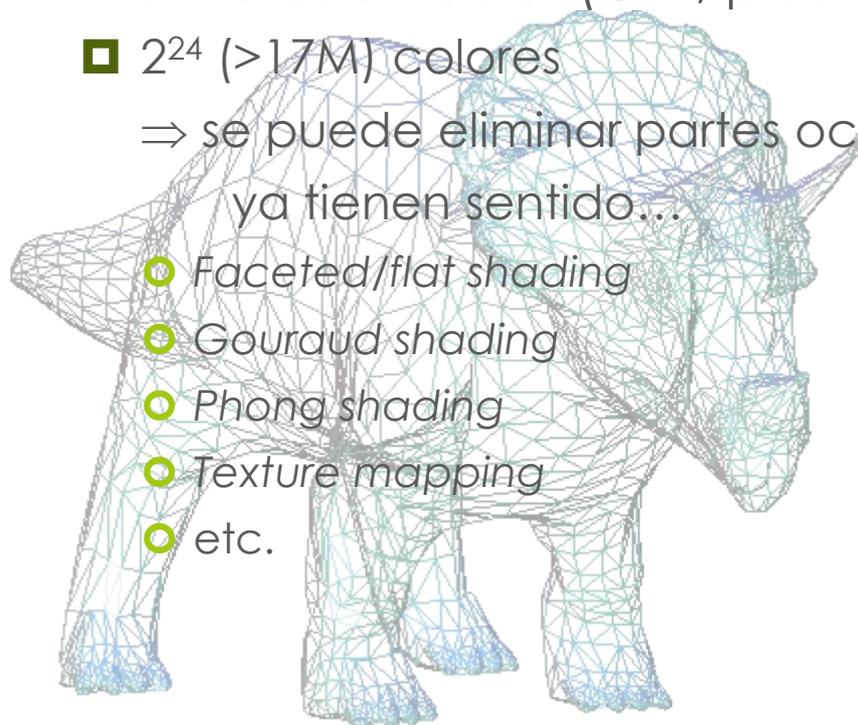
Introducción: zoo de proyecciones



Imágenes y taxonomía tomadas de "biblia" de Gráficos [3D] de J.D. Foley et al.: *Computer Graphics: Principles and Practice (2nd ed. in C)*, Addison-Wesley, 1997

Introducción: perspectiva histórica

- CRT (*Cathode Ray Tube*) de fósforo monocromo
 - 1! tono de verde o ámbar (negro en impresoras)
 - ⇒ imposible dar “sensación 3D”: sólo cabe *wireframe rendering*
- Monitores en color (CRT, plasma, LCD/TFT, LED, etc.)
 - 2^{24} (>17M) colores
 - ⇒ se puede eliminar partes ocultas y ya tienen sentido...

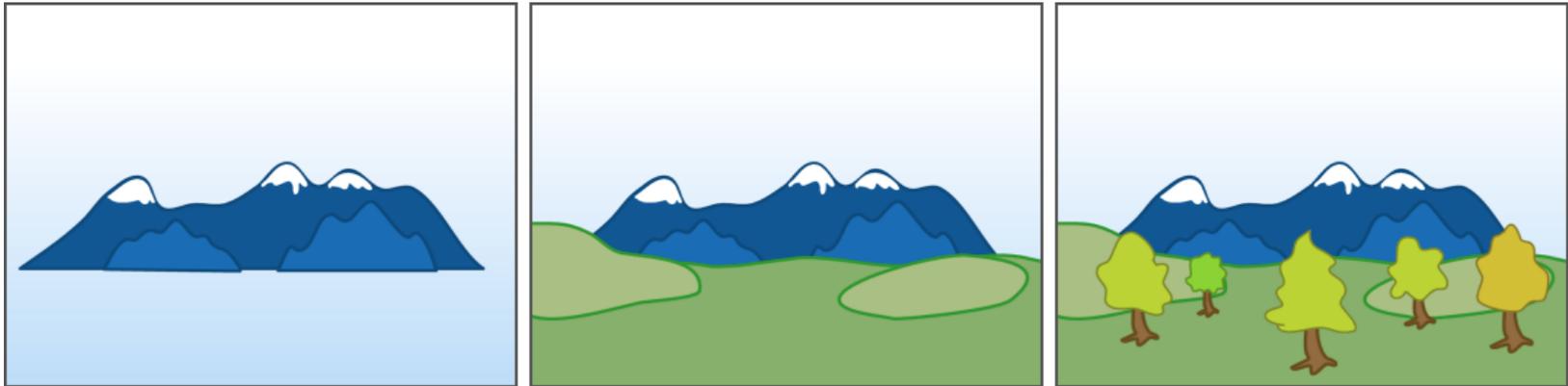


- *Faceted/flat shading*
- *Gouraud shading*
- *Phong shading*
- *Texture mapping*
- etc.

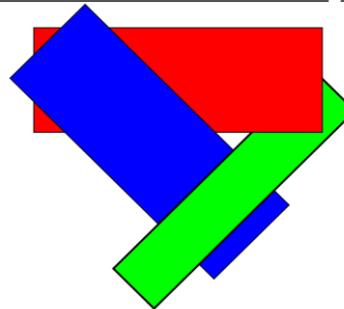


Visualización: HSR (1/2)

- ▣ Problema: HSR (*Hidden Surface Removal*)
- ▣ Soluciones
 - ▣ Algoritmos de “precisión de imagen” (ej: *ray tracing*)
 - ▣ Algoritmos de “precisión de objeto”
 - Ordenación de primitivas previa a pintado (ej: algoritmo del pintor)



○ Pb: no siempre posible!



Visualización: HSR (2/2)

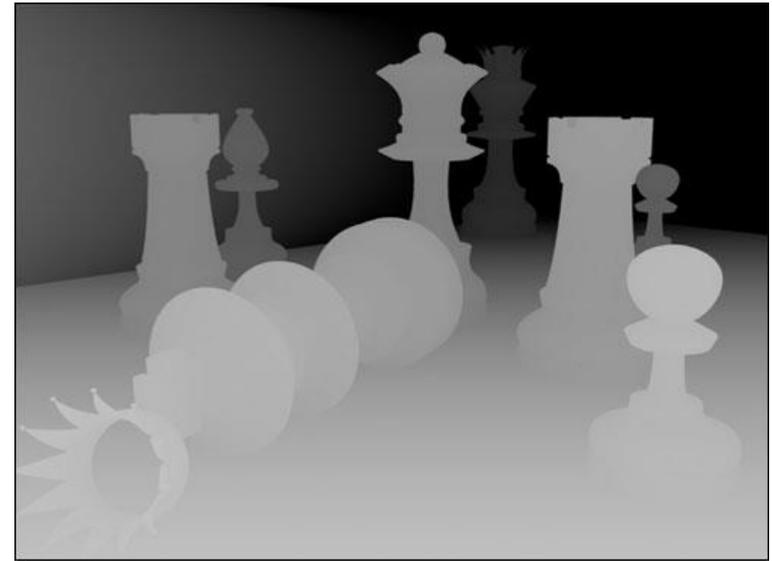
■ Soluciones

■ Algoritmos de “precisión de objeto”

- [...] *and the winner is...*
- *z-buffer* (memoria adicional para almacenar profundidad)
⇒ Implementado en tarjetas gráficas con aceleración HW



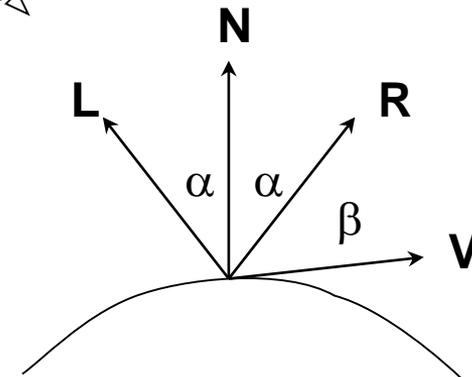
frame buffer



z/depth-buffer

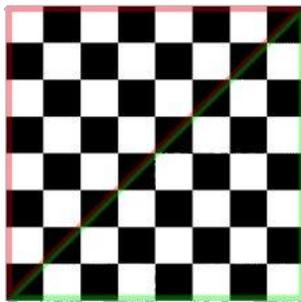
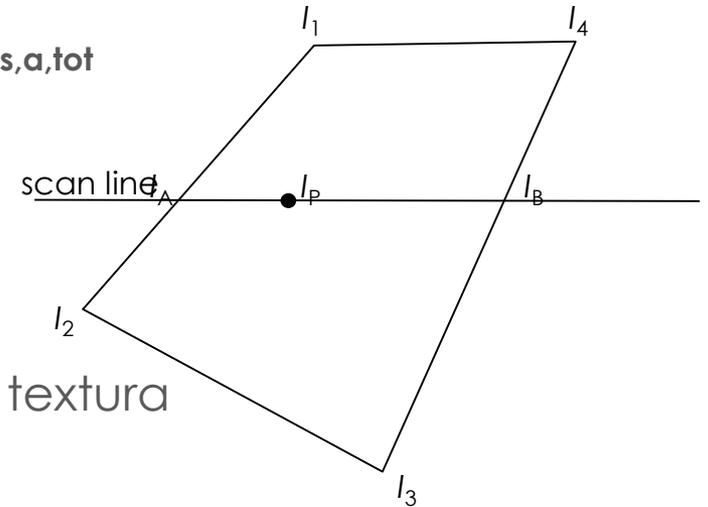
Visualización: iluminación

- Luces direccionales vs. posicionales
- Importancia de normales
- Reflejo difuso: intensidad $I_d = (I_R; I_G; I_B)$
 - Material mate (ej: tiza, papel no satinado) \Rightarrow coeficiente k_d (color)
 - Modelo Lambert: $I_d = k_d I \cos \alpha$ [= $k_d I (\mathbf{L} \cdot \mathbf{N})$ si $\|\mathbf{L}\| = \|\mathbf{N}\| = 1$]
- Reflejo especular
 - Material brillante (ej: metal, plástico)
 - Modelo Phong
 - Coeficientes k_s y n (color e intensidad brillo)
 - $I_s = k_s I (\cos \beta)^n$ [= $k_s I (\mathbf{R} \cdot \mathbf{V})^n$ si $\|\mathbf{R}\| = \|\mathbf{V}\| = 1$]
- Reflejo total
 - Txapuza: luz ambiente (suma de otras contribuciones), I_a
 - $I_{tot} = I_d + I_s + I_a$

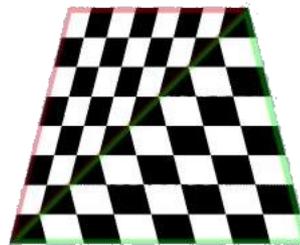


Visualización: sombreado

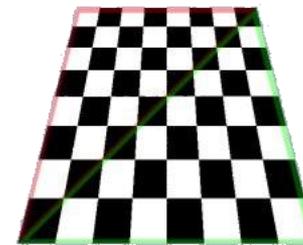
- $\odot j \odot$: Interpolación color (*shading*), no sombras (*shadows*)
 - Técnica original de Gouraud interpola $I_{d,s,a,tot}$
 - Interpolación 2D
 - Técnica de Phong interpola L, N, R, V
 - (y sólo después $\alpha, \beta, I_{d,s,a,tot}$)
- Técnicas similares para coordenadas de textura
 - P_b : *warping*



Flat



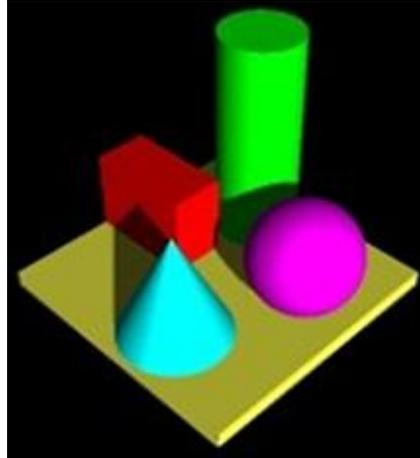
Affine



Correct

Visualización: lujo

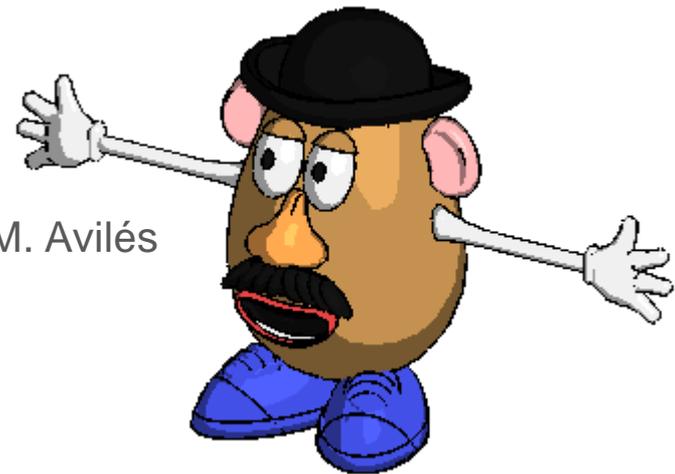
- Transparencias
- Sombras
- Reflexiones
- Refracciones
- etc., etc.



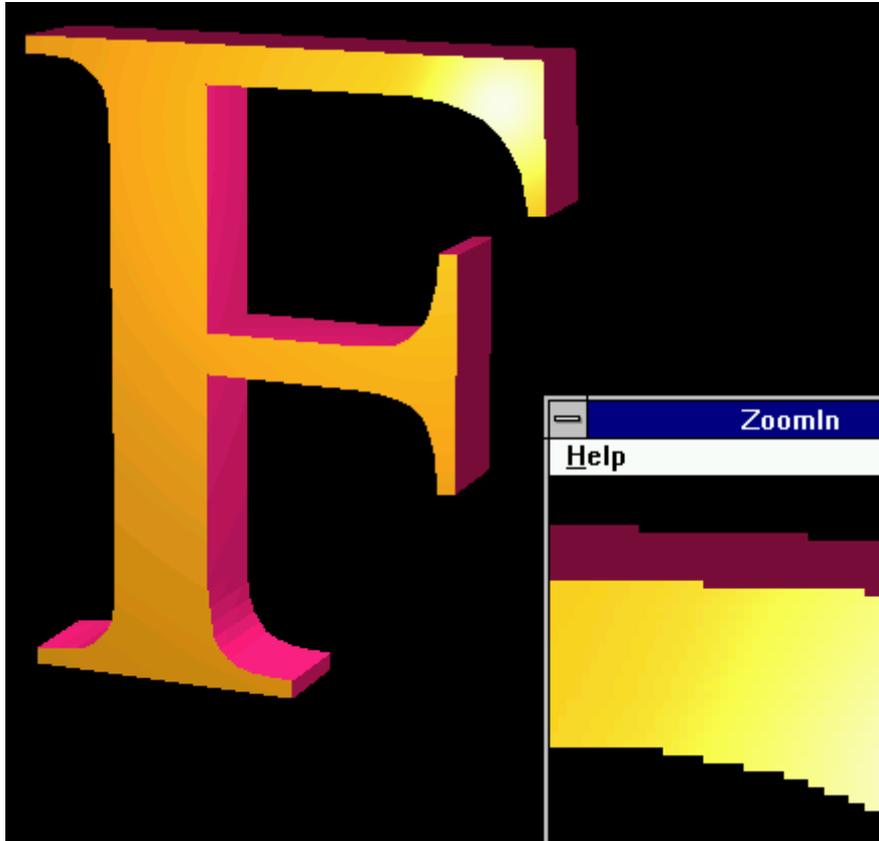
- NB: también “antilujo”: NPR (*Non-Photorealistic Rendering*)
 - Ejemplo: *cartoon shading*



© M. Avilés



Visualización: filtrado (*anti-aliasing*)



Bibliografía

- M. F. Cohen, J. R. Wallace: *Radiosity and Realistic Image Synthesis*, Academic Press, 1993
- **J. D. Foley, A. van Dam et al.: *Computer Graphics: Principles and Practice (2nd ed. in C)*, Addison-Wesley, 1997 [NB: ∃ 3rd ed. 2013]**
- A. S. Glassner (ed.): *Graphics Gems*, Academic Press, 1990 [primer libro de una serie de cinco]
- J. Kessenich et al.: *OpenGL Programming Guide (9th ed.)*, Addison-Wesley, 2016
- R. Hall: *Illumination and Color in Computer Generated Imagery*, Springer-Verlag, 1988
- G. Wolberg: *Digital Image Warping*, IEEE Computer Society Press, 1990
- Wikipedia, claro...