Informática Gráfica

Texturado, iluminación y render de imágenes sintéticas

Diego Gutiérrez
Francisco José Serón

Informática Gráfica

Texturado, iluminación y render de imágenes sintéticas

Diego Gutiérrez
Francisco José Serón

Edita:
Departamento de Informática e Ingeniería de Sistemas
C/María de Luna 1, Campus Río Ebro, Edificio Ada Byron
50018 Zaragoza (España)

ISBN:

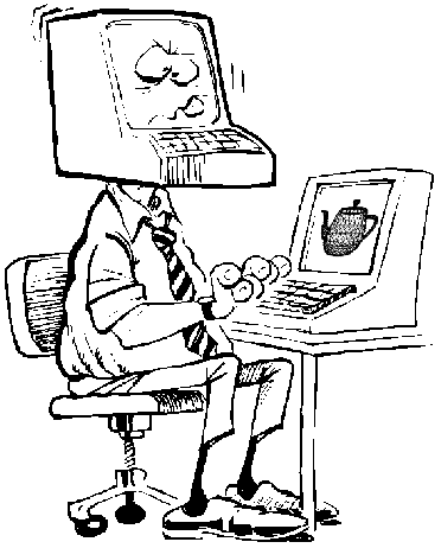
Depósito Legal:

Imprime:

Tabla de contenidos

Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

GIGA



Introducción

Diego Gutiérrez y Francisco J. Serón

Introducción

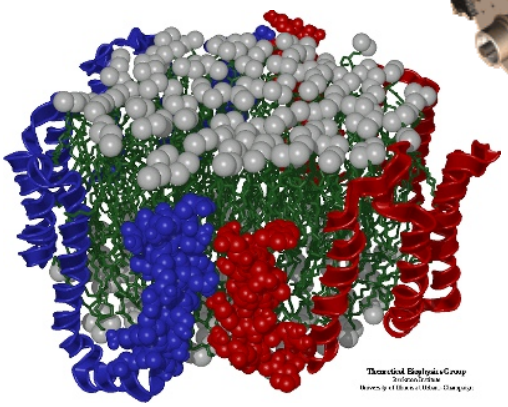
- La informática gráfica es una tecnología para presentar información
- *Si puedes imaginarlo, puedes hacerlo con CG*
- Pensad en todos los ámbitos en que está presente...

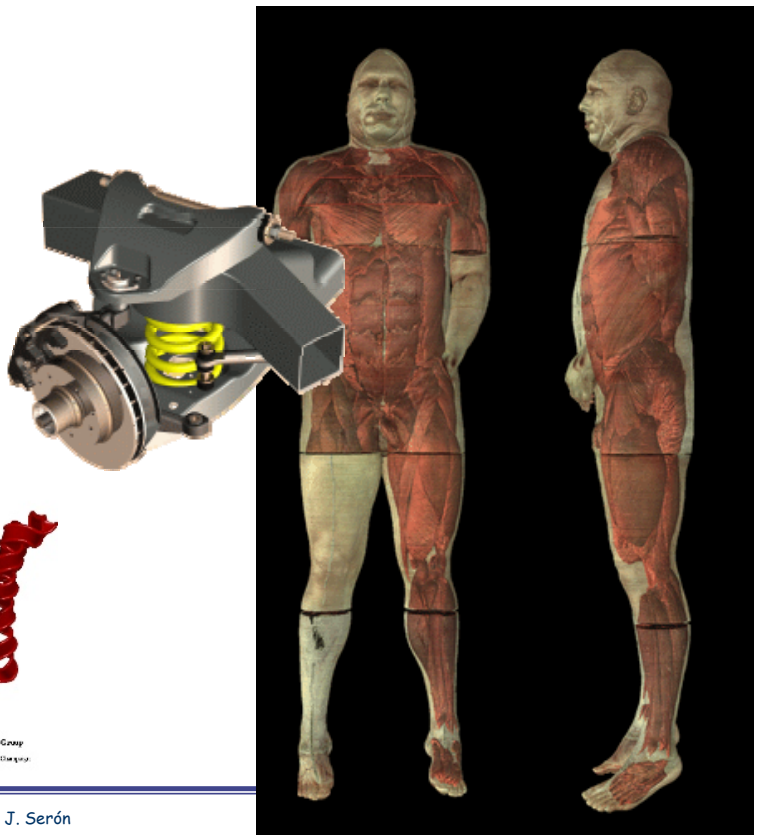
Introducción

- Este curso es la excusa para perfecta para...
  - Ir al cine
  - Jugar al ordenador
  - Ver mucha televisión
  - ...

- Y si alguien pregunta, decidles que estáis estudiando!

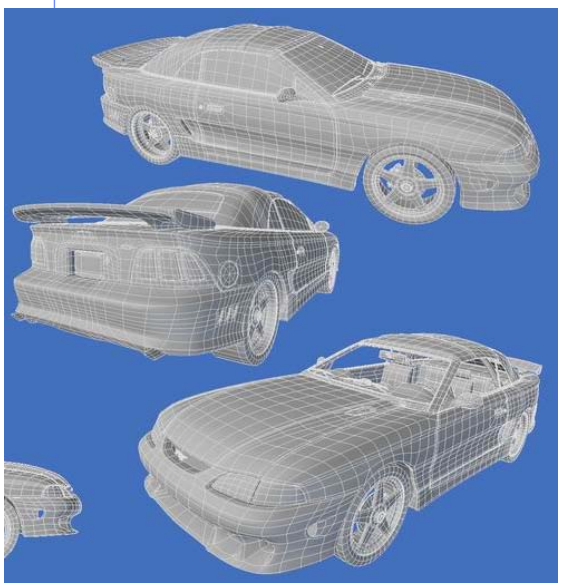
Ambitos





Introducción

- Dada una escena 3D... ¿cómo obtenemos una imagen realista?

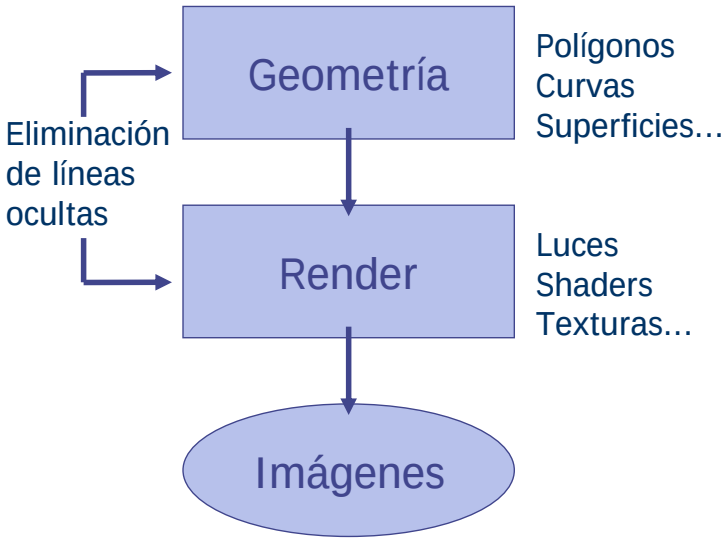


Introducción

- Necesitamos:
  - Descripción de la escena:
    - Forma de los objetos
    - Localización y orientación
    - Propiedades de las superficies
    - Fuentes de luz
    - ...
  - Información de la visibilidad:
    - Qué puntos son visibles?
    - Cómo están iluminados?
    - ...
- Fundamentalmente nos ocuparemos de la última pregunta

Introducción

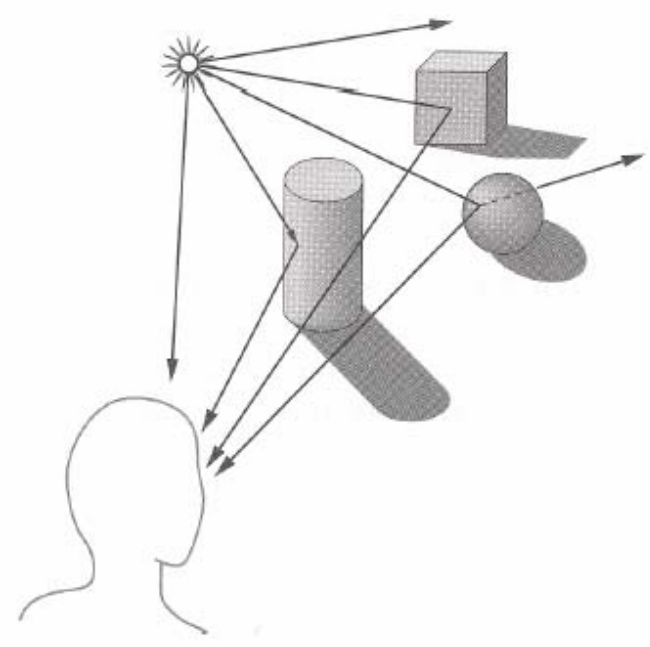
- Rendering pipeline (sobresimplificado!!!)



Introducción

- Una vez hemos elegido una vista de nuestra escena, el objetivo es saber cómo están iluminados los puntos visibles
- Dos fuentes de luz para cada punto:
  - Luz directa
  - Luz indirecta
- ... y dos tipos de algoritmos:
  - Algoritmos de iluminación local
  - Algoritmos de iluminación global (GI)

Introducción



Introducción

- Dos enfoques para afrontar el problema:
  - Render heurístico: usa todos los trucos que puedas con tal de que la imagen quede aparentemente bien (*top-down*)
    - Interpolación
    - Gouraud
    - Phong
  - Render basado en la física: modela los procesos físicos que determinan el transporte de luz en la escena y deja que la imagen aparezca (*bottom-up*)
    - Trazado de rayos (Ray tracing)
    - Radiosidad
    - Mapas de fotones (Photon mapping)
- Los físicos son más caros computacionalmente, pero dan resultados más exactos (más realismo)

Introducción

- Otra clasificación de algoritmos de rendering, según el espacio donde trabajan:
  - Espacio imagen: primero averiguamos qué puntos se ven y luego cómo están iluminados
    - Ray tracing
  - Espacio objeto: primero averiguamos cómo está iluminada la escena y luego qué puntos se ven
    - Radiosidad
  - Mixtos: algunos algoritmos bidireccionales
    - Photon mapping

Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

Modelos de Iluminación Local

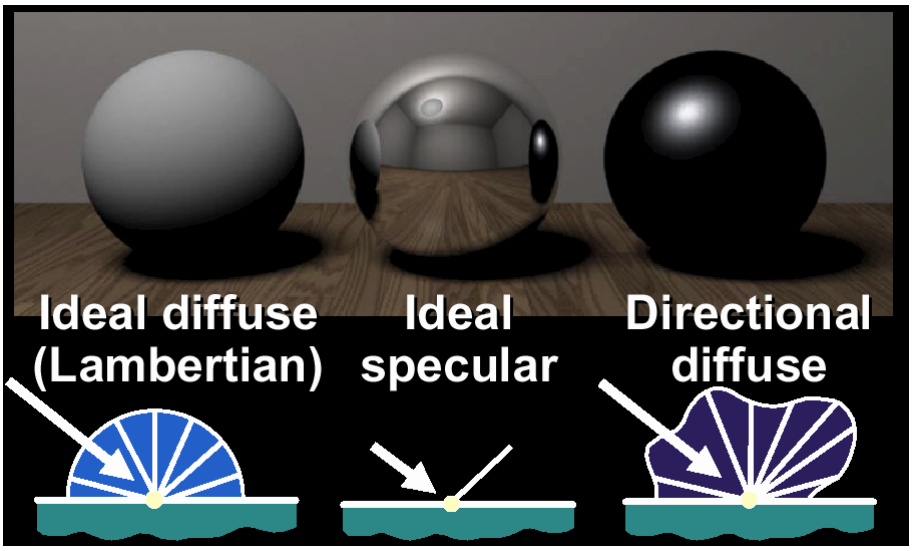
Diego Gutiérrez y Francisco J. Serón

Indice

Introducción

- Comenzamos con los **modelos de iluminación local**
- En estos modelos, sólo se calcula la luz que proviene directamente de las fuentes de luz
- Percibimos los objetos gracias a la **reflexión** de la luz en ellos (especular, difusa)
- Aunque el color es un aspecto muy importante…
  - En la definición de las fuentes de luz
  - En las propiedades del material
  - En la generación de efectos
- …no lo incluiremos en la discusión por claridad

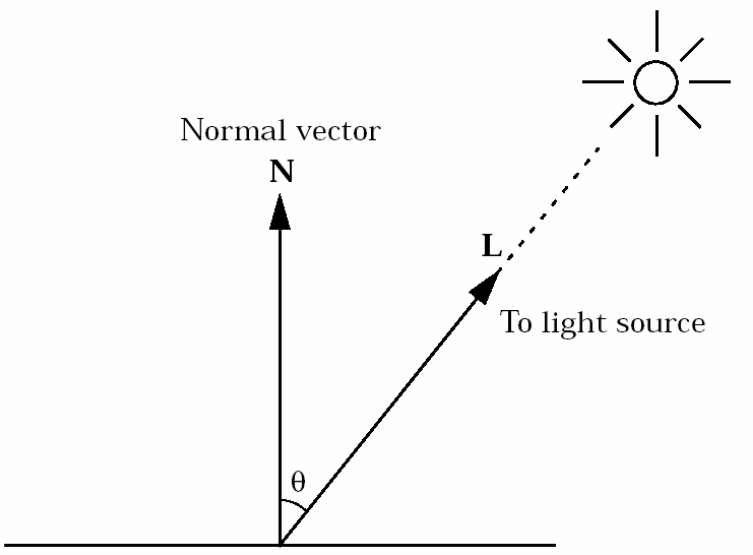
Tipos de reflexiones



- En la vida real todos los materiales son, en mayor o menor medida, difusos direccionales (menos los espejos perfectos)

Orientación de las superficies

- **N** y **L** son vectores unitarios
- **Theta** es el ángulo de incidencia

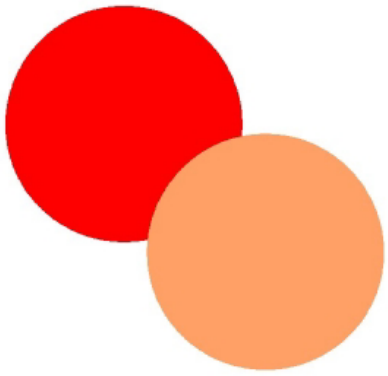


Un modelo sencillo de iluminación

- Consideremos sólo **iluminación ambiental:**
  - Uniforme desde todas las direcciones (intensidad I_a)
  - Proviene de múltiples reflexiones
  - $I = I_a K_d$
  - K_d mide la reflectividad de una superficie ante luz difusa (ambiental). Llamado *coeficiente de reflexión difusa*
  - K_d tiene valores entre 0 y 1
- Problemas?

Un modelo sencillo de iluminación

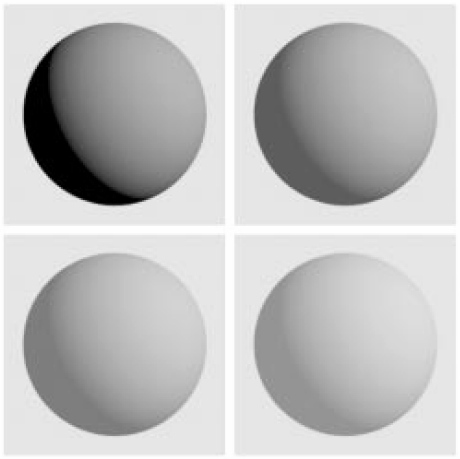
- Al usar la ecuación anterior, cada punto del objeto poseerá la misma intensidad!
- Perdemos la tridimensionalidad



- Son esferas o círculos?

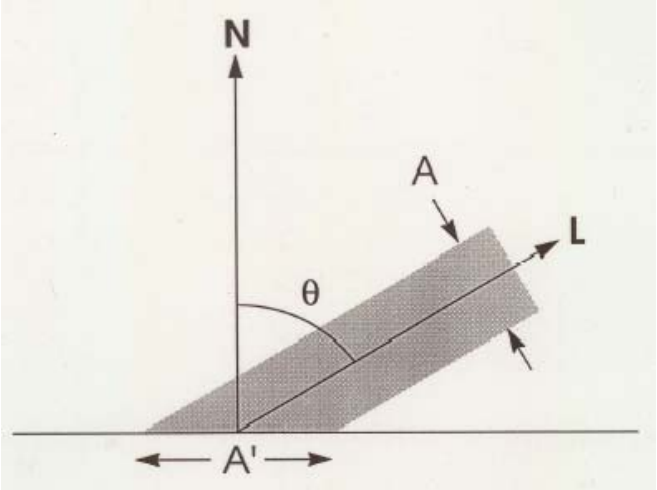
Un modelo sencillo de iluminación

- Ejercicio: escribir 1000 veces las siguientes frases:
  - La iluminación ambiental es un truco rastrero y vil
  - La iluminación ambiental es más mala que el tabaco
  - La iluminación ambiental no es mi amiga!



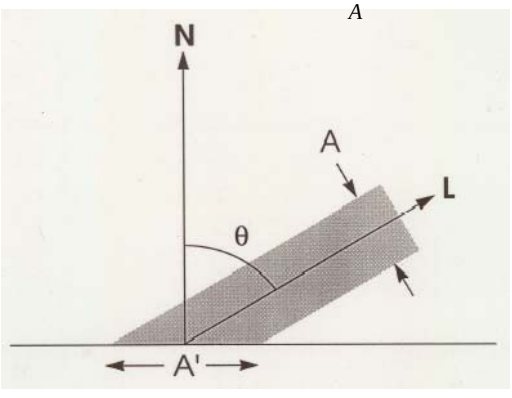
Ley del coseno de Lambert

- La intensidad de la luz reflejada por una superficie depende de la intensidad de la luz recibida, que a su vez depende de la fuente de luz y la orientación de la superficie respecto a ella



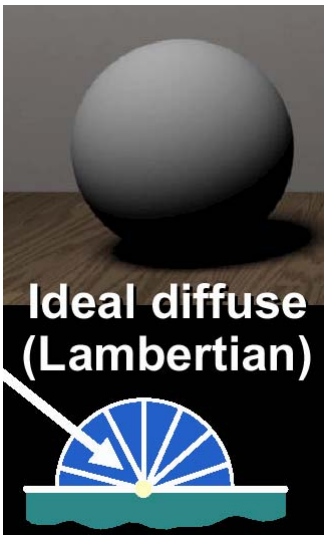
Ley del coseno de Lambert

- Intensidad de luz I_p de área A
- La luz se dispersa sobre un área A'
- $A'=A/\cos\theta$
- $A=A' \cos\theta$
- Luego la intensidad efectiva en la superficie es $I_e=I_p\cos\theta$

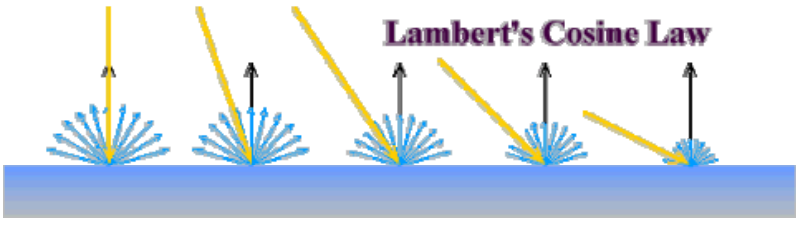


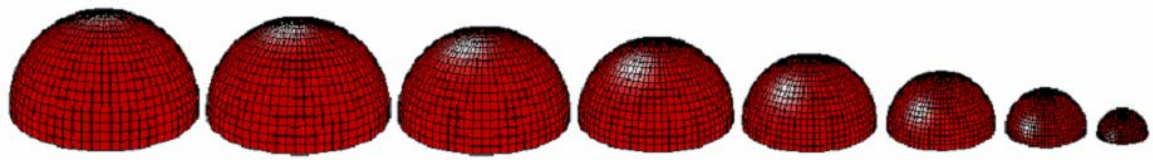
Ley del coseno de Lambert

- ¿Por qué nos quemamos más los hombros en verano?
- Los difusores ideales se comportan según esta ley (materiales *Lambertianos*)
- La intensidad reflejada es **independiente de la dirección de visualización**
- Sólo depende de la **orientación de la superficie respecto a la luz**



Ley del coseno de Lambert



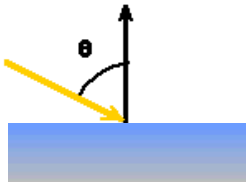


GIGA
Diego Gutiérrez y Francisco J. Serón

Reflexión difusa

- Podemos calcular la intensidad de la luz reflejada de manera difusa como:

$$I_{diffuse} = k_d I_{light} \cos\theta$$



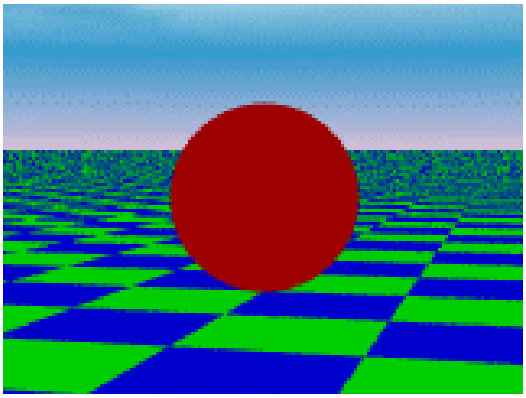
- Como **N** y **L** son vectores unitarios:

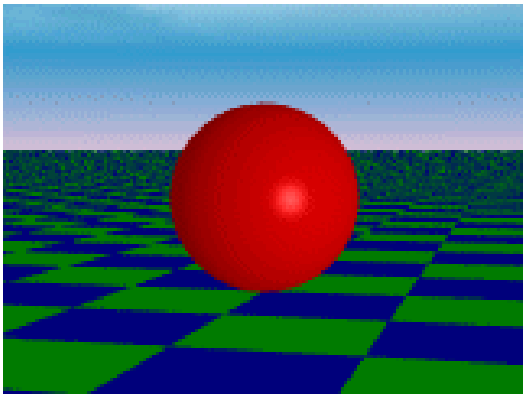
$$\cos\theta = \mathbf{N} \bullet \mathbf{L}$$

- La intensidad de la luz reflejada de manera difusa es entonces:

$$I_{diffuse} = k_d I_{light} \left(\bar{n} \cdot \bar{l}\right)$$

Reflexión difusa





Un segundo modelo de iluminación

- Consideramos ahora la componente ambiental más la difusa (proviniente de una fuente puntual)

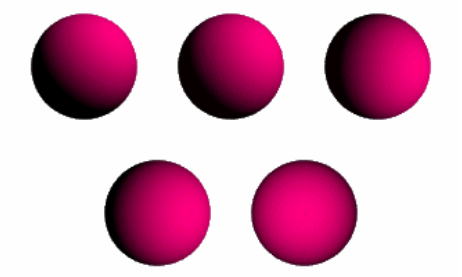
$$I = ambient + diffuse$$

$$I = k_d I_a + k_d I_p (\mathbf{N} \bullet \mathbf{L})$$

- Para la componente difusa, cuando el producto escalar es negativo (ángulos mayores de 90º), la superficie no "ve" la luz y la energía reflejada es cero

Un segundo modelo de iluminación

- Ejemplos de esferas lambertianas con fuentes de luz en distintos ángulos:



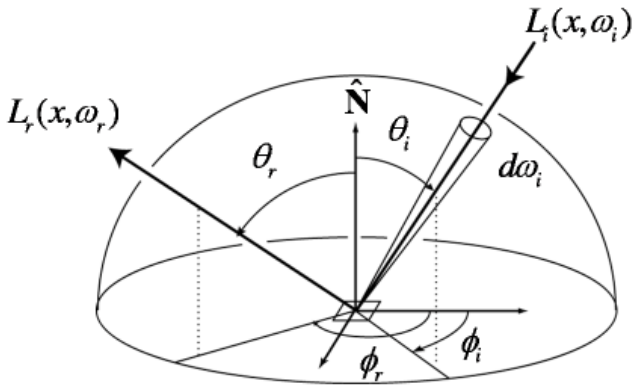
- ¿Por qué se usan esferas para tests de iluminación y sombreado?

Reflexión especular

- El modelo Lambertiano sólo permite modelar objetos mate
- Las superficies brillantes ofrecen reflejos especulares (*highlights*) bajo determinados ángulos de visión
- Ojo con la terminología! (reflejo especular vs. *highlight*)
- Luego la posición del *highlight es* una función del punto de vista de la escena: introducimos la **dependencia de la dirección de visualización**
- Los *highlights* ocurren sobre un estrecho rango de ángulos
- El color del *highlight* es generalmente el mismo que el de la lulz que lo provoca
- Los espejos son un ejemplo de reflexión especular perfecta: ángulo de incidencia igual a ángulo de reflexión

Reflexión especular

- Ecuación de la reflexión

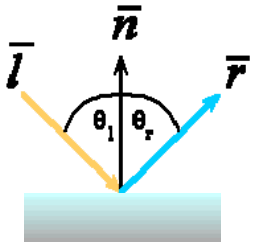


$$L_r(x,\omega_r) = \int_{H^2} f_r(x,\omega_i \rightarrow \omega_r) L_i(x,\omega_i) \cos\theta_i \, d\omega_i$$

Reflexión especular

- El rayo, la normal y el rayo reflejado pertenecen al mismo plano
- Ley de Snell:

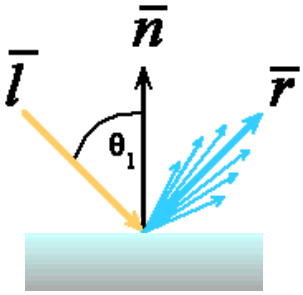
$$n_i \sin\theta_i = n_r \sin\theta_r$$

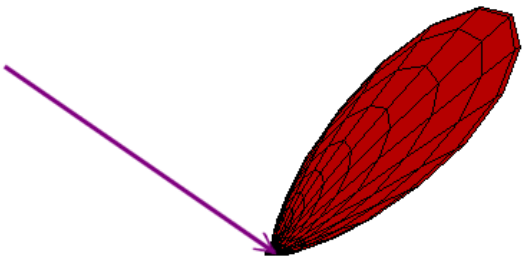


- La reflexión especular cumple que $\theta_i = \theta_r$

Reflexión especular

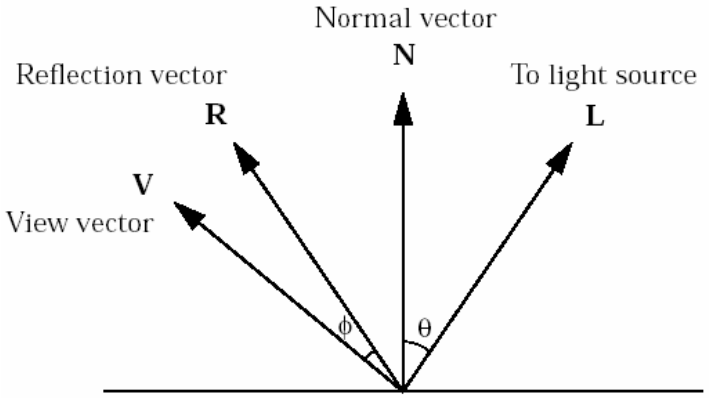
- Ningún material real es especular perfecto
- Imperfecciones en la superficie hacen que los rayos reflejados se dispersen a partir de la dirección teórica





- Introducimos ahora un modelo empírico para calcular el reflejo especular: **Phong**

Modelo de iluminación de Phong



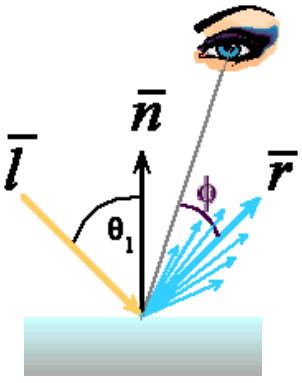
- Introducimos más notación:
  - **N** es la normal a la superficie
  - **L** es la dirección hacia la fuente de luz
  - **V** es la dirección hacia el punto de vista
  - **R** es la dirección de la reflexión especular ideal
  - Como veremos, la intensidad de la reflexión especular depende del ángulo entre **V** y **R**

Modelo de iluminación de Phong

- Aviso de antemano: no confundir con el *modelo de interpolación de Phong*
- El modelo de iluinación de Phong es heurístico, sin base física...
- ...y uno de los más utilizados en CG

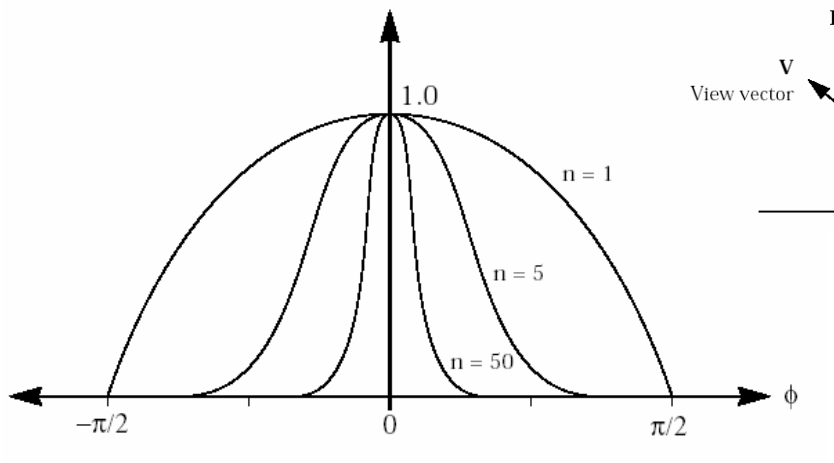
$$I_{specular} = I_{light}(\cos\phi)^{n_{shiney}}$$

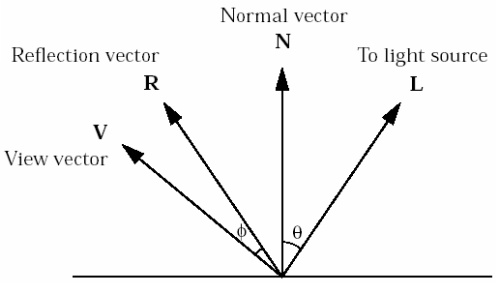
- El término *cos* ϕ es máximo cuando la superficie se mira desde la dirección especular, y cae a 0 cuando se mira a 90º de esa dirección. El término n_{shiny} controla ese decaimiento



Modelo de iluminación de Phong

- La n (o n_{shiny}) depende de la superficie:
  - Reflector ideal: n=*infinito*
  - Reflector casi nulo (una esponja): n=1
  - En la práctica n[1..200]

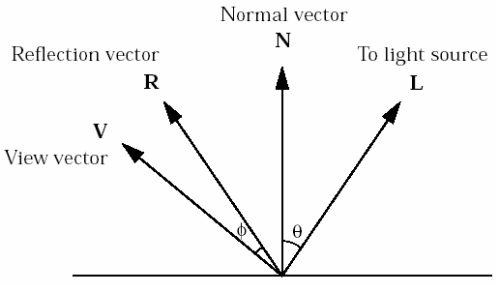




Modelo de iluminación de Phong

- Podemos evaluar el *cos* ϕ vectorialmente:
  - *cos* ϕ = **R·V** (ya que ambos vectores son unitarios)
- La reflexión especular también depende de θ
- Luego tenemos que:

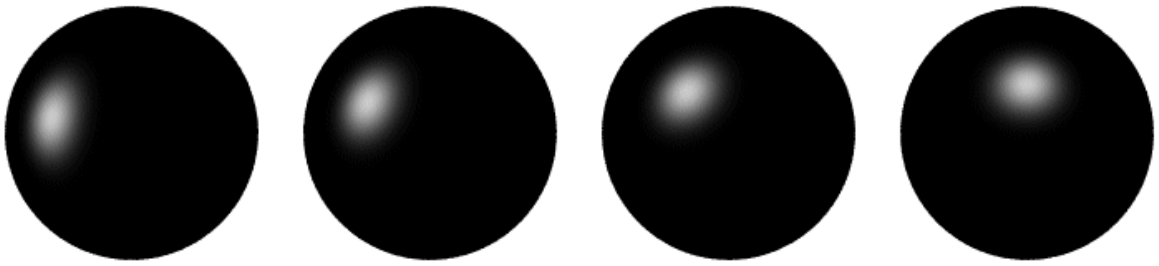
$$I_s \propto W(\theta)\ (\mathbf{R} \bullet \mathbf{V})^n$$



- En la práctica, se reemplaza $W(\theta)$ por K_s
- K_s es el **coeficiente de reflexión especular**
- K_s[0..1]

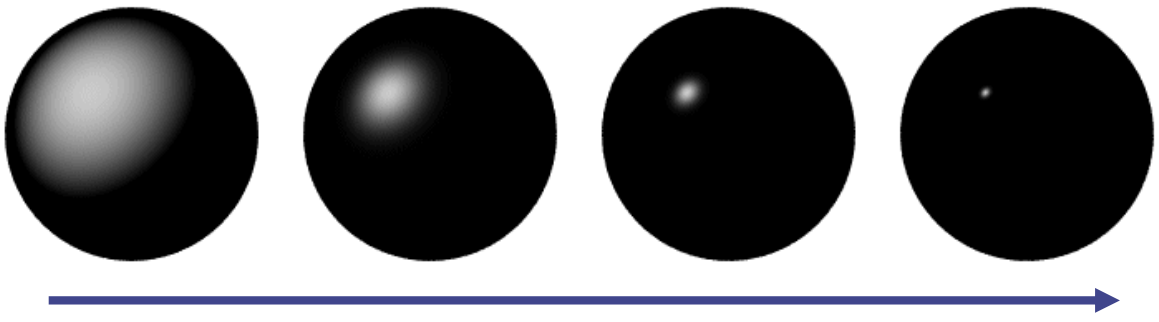
Modelo de iluminación de Phong

- Reflexiones especulares (modelo de Phong) con *n* cte. y variando la posición de la fuente de luz



Modelo de iluminación de Phong

- Reflexiones especulares (modelo de Phong) aumentando *n* y manteniendo la posición de la fuente de luz fija



Tercer modelo de iluminación

- Teníamos:

$$I = ambient + diffuse$$

$$I = k_d I_a + k_d I_p (\mathbf{N} \bullet \mathbf{L})$$

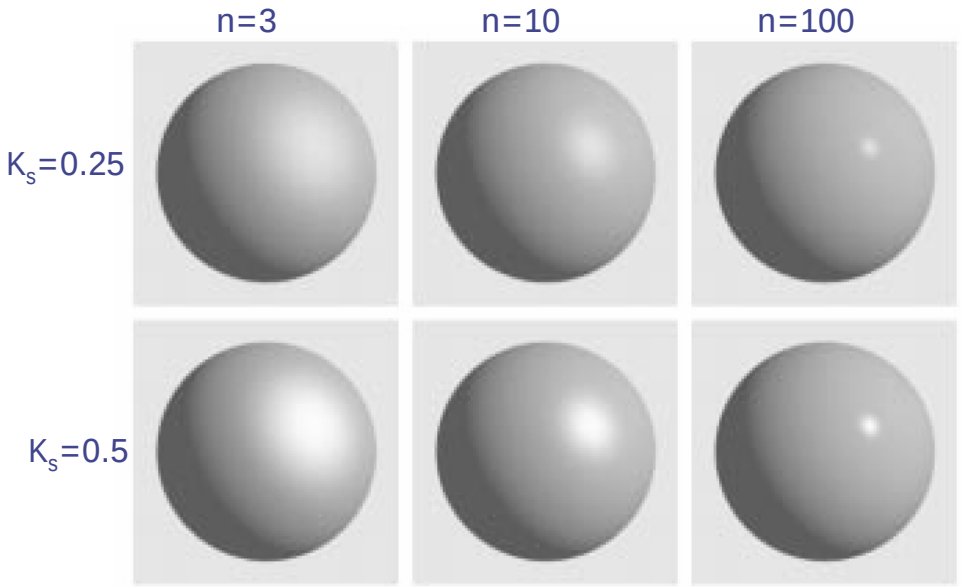
- Seguimos completando el modelo:

$$I = ambient + diffuse + specular$$

$$I = k_d I_a + I_p \left[k_d (\mathbf{N} \bullet \mathbf{L}) + k_s (\mathbf{R} \bullet \mathbf{V})^n \right]$$

Tercer modelo de iluminación

- Ejemplos:
  - K_d=0.4 I_a=0.2I_p



Tercer modelo de iluminación

- La intensidad de luz de una fuente decae con la distancia al cuadrado:
  - $I = I_p/(4\pi d^2)$
- Esto suele sustituirse por la siguiente fórmula experimental:
  - $I = I_p/(d+d_0)$

...donde d_0 es una constante determinada experimentalmente

- De nuevo un truco empírico (*hack*) sustituye a un modelo del fenómeno físico
- Hay muchas “constantes determinadas experimentalmente” en CG

Cuarto modelo de iluminación

- Por modelos será: introducimos la atenuación de la luz con la distancia y tenemos:

$$I = k_d I_a + \frac{I_p}{d+d_0}\left[k_d(\mathbf{N}\bullet\mathbf{L}) + k_s\cos^n\phi\right]$$

- O en forma vectorial (vectores normalizados!)

$$I = k_d I_a + \frac{I_p}{d+d_0}\left[k_d(\mathbf{N}\bullet\mathbf{L}) + k_s(\mathbf{R}\bullet\mathbf{V})^n\right]$$

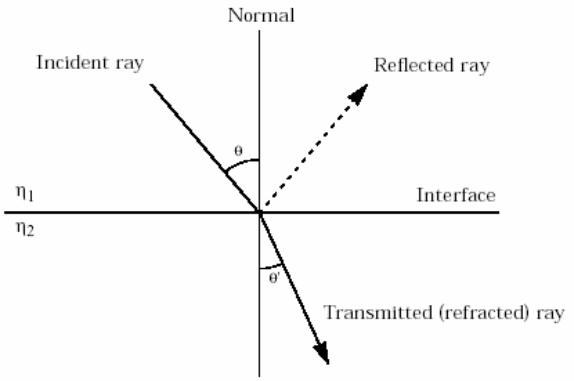
- Si hay más de una luz:

$$I = \text{ambient term} + \sum_{i=1}^{n}\text{diffuse term}_i + \sum_{i=1}^{n}\text{specular term}_i$$

Transparencia

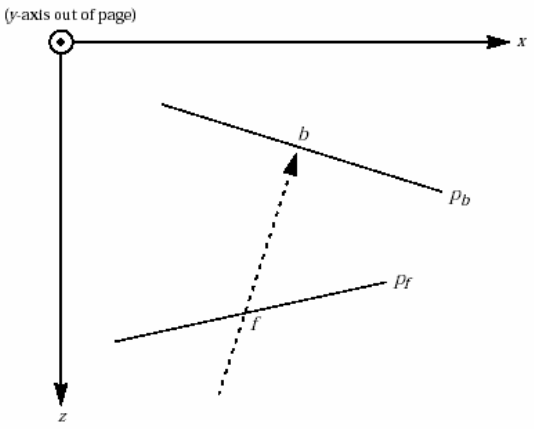
- Recordemos la Ley de Snell:

$$n_i \sin\theta_i = n_r \sin\theta_r$$



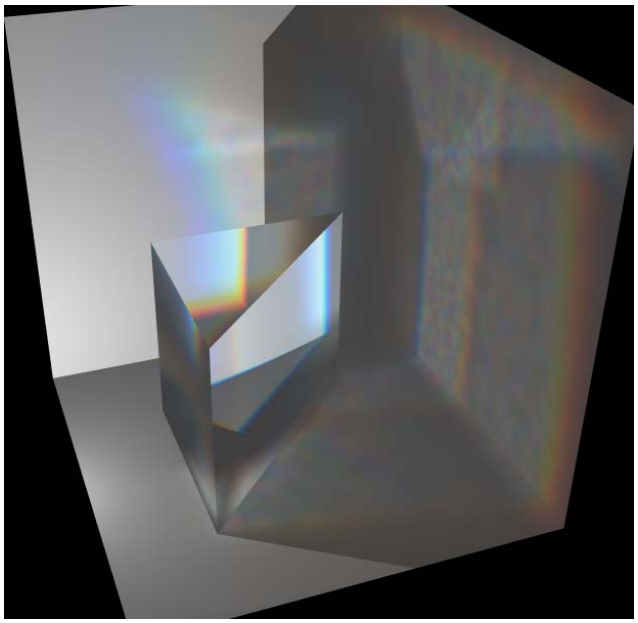
Transparencia

- Transparencia sin refracción:
  - *t* es el **coeficiente de transmisión**
  - Objeto opaco: t=1; Objeto transparente: t=0
  - *t* puede ser función del color (ventana de una iglesia)
- En el ejemplo, la intensidad observada sería:
  - $I = tI_f + (1-t)I_b$



Dispersión

- El índice de refracción *n* es f(λ): **dispersión**



Color

- De momento no hemos considerado el color, sólo hemos hablado de intensidades de luz
- El modelo de color más utilizado en CG es el RGB (lo cual no implica que sea el mejor, ni siquiera el segundo mejor…)
- *Desacoplamos entonces la solución en tres longitudes de onda linearmente independientes*

Qué?

Color

- Desacoplamos entonces la solución en tres longitudes de onda linearmente independientes:
  - R,G,B
  - Para el Rojo:

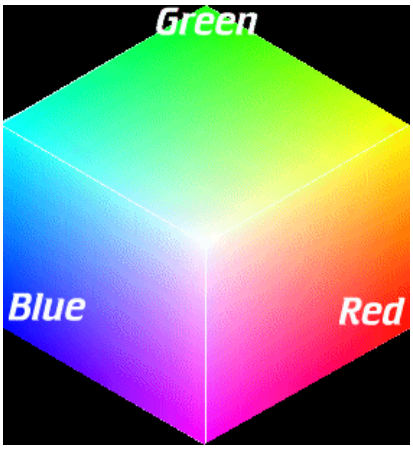
$$I_R = k_{dR} I_{aR} + \frac{I_{pR}}{d + d_0}\left[k_{dR}(\mathbf{N} \bullet \mathbf{L}) + k_s(\mathbf{R} \bullet \mathbf{V})^n\right]$$

  - Se ve cómo mantenemos la suposición de que el *highlight* especular es del color de la fuente
- Modelos espectrales más complejos dan mejores soluciones:

$$I_\lambda = k_{d\lambda} I_{a\lambda} + \frac{I_{p\lambda}}{d + d_0}\left[k_{d\lambda}(\mathbf{N} \bullet \mathbf{L}) + k_s(\mathbf{R} \bullet \mathbf{V})^n\right]$$

Color

- Al trabajar en RGB, los valores obtenidos [0..255] pueden mapearse directamente a los canales RGB de un monitor
- ¿Y qué pasa si sobrepasamos el valor máximo?
- *Reproducción de tono*



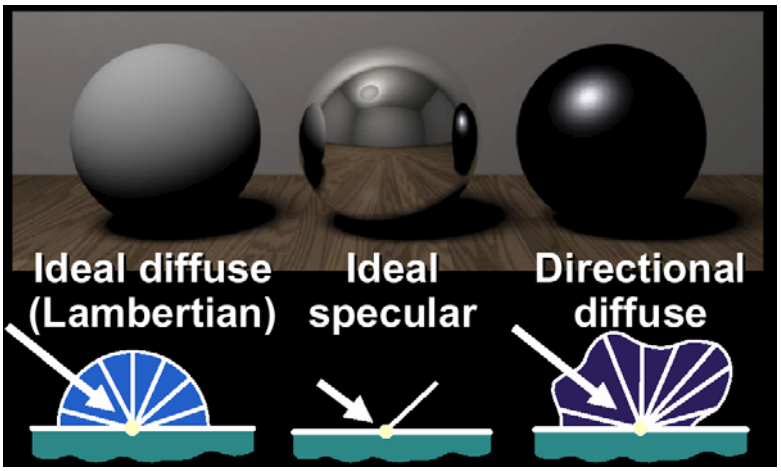
Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

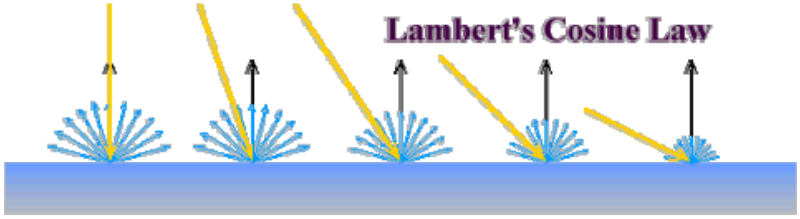
Modelado Visual y Animación

Modelos de Iluminación Local (in a nutshell)

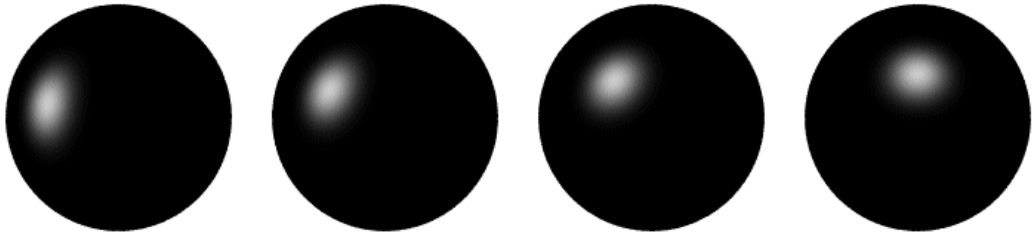
Diego Gutiérrez y Francisco J. Serón

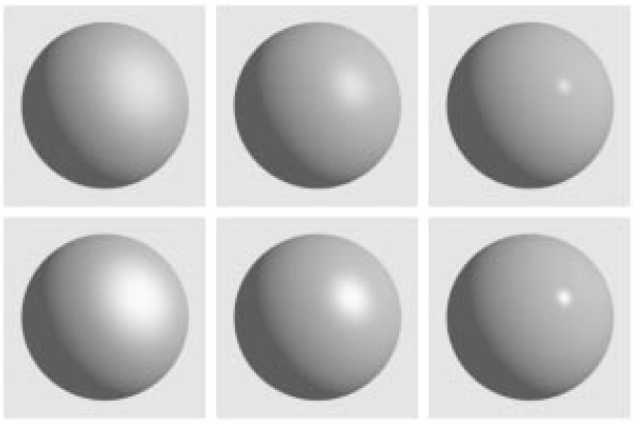
Iluminación local (in a nutshell)



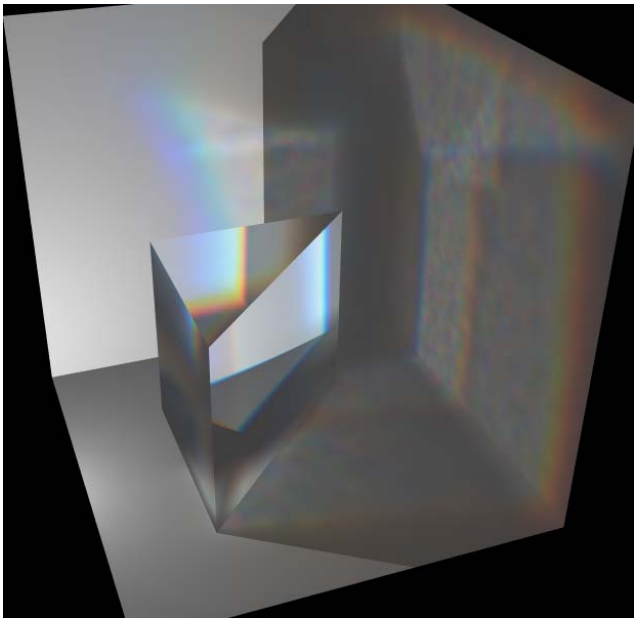


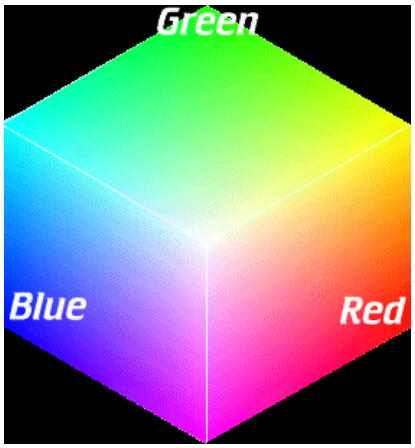
Iluminación local (in a nutshell)



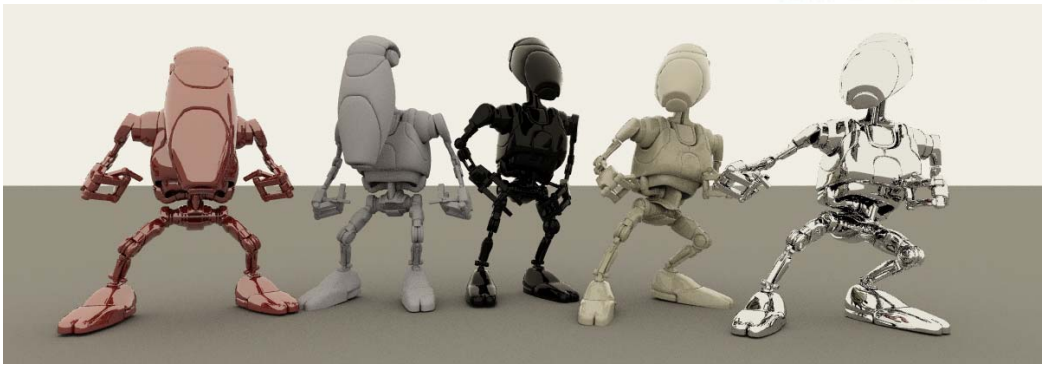


Iluminación local (in a nutshell)





Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza



Shading

Diego Gutiérrez y Francisco J. Serón

Indice

- Sombreado de polígonos
- Sombreado plano
- Sombreado de Gouraud
- Sombreado de Phong
- *Cheap Phong*
- Coste computacional

Sombreado de polígonos

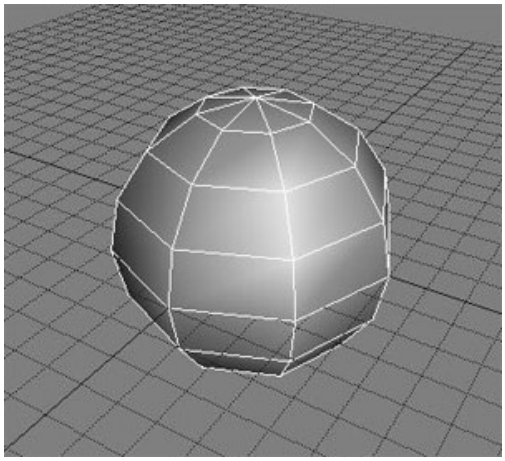
- *Shading* = sombreado

- La mayoría de renderizadores trabajan sobre polígonos
- La mayoría de los objetos se representan como polígonos
- La mayoría de los algoritmos están pensados para polígonos
- Los aceleradores gráficos trabajan sobre polígonos

- Conclusión:

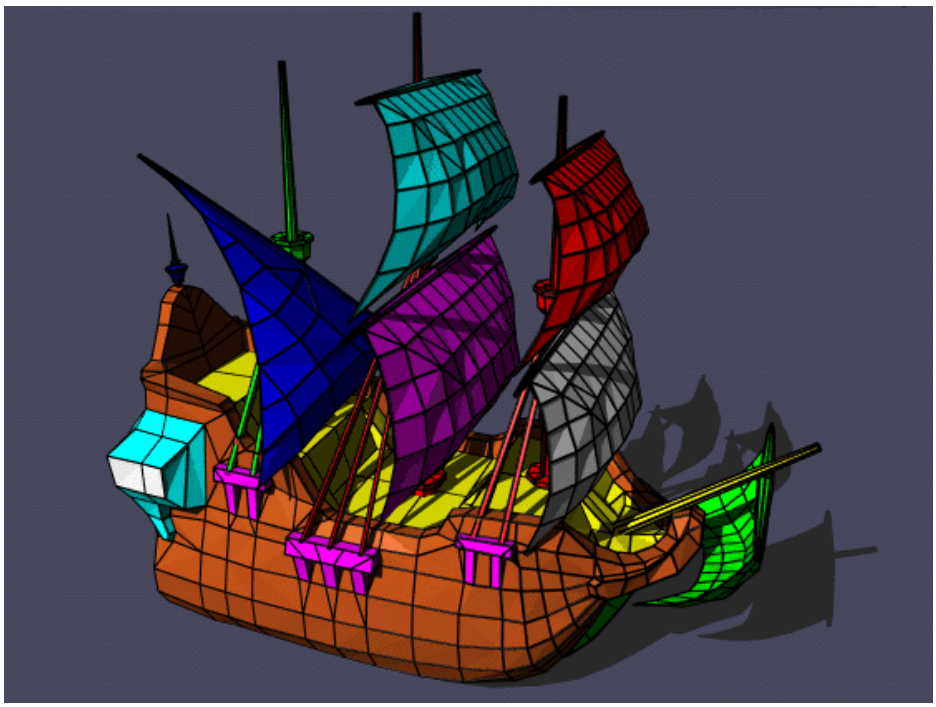
Los polígonos son tus amigos

Sombreado de polígonos

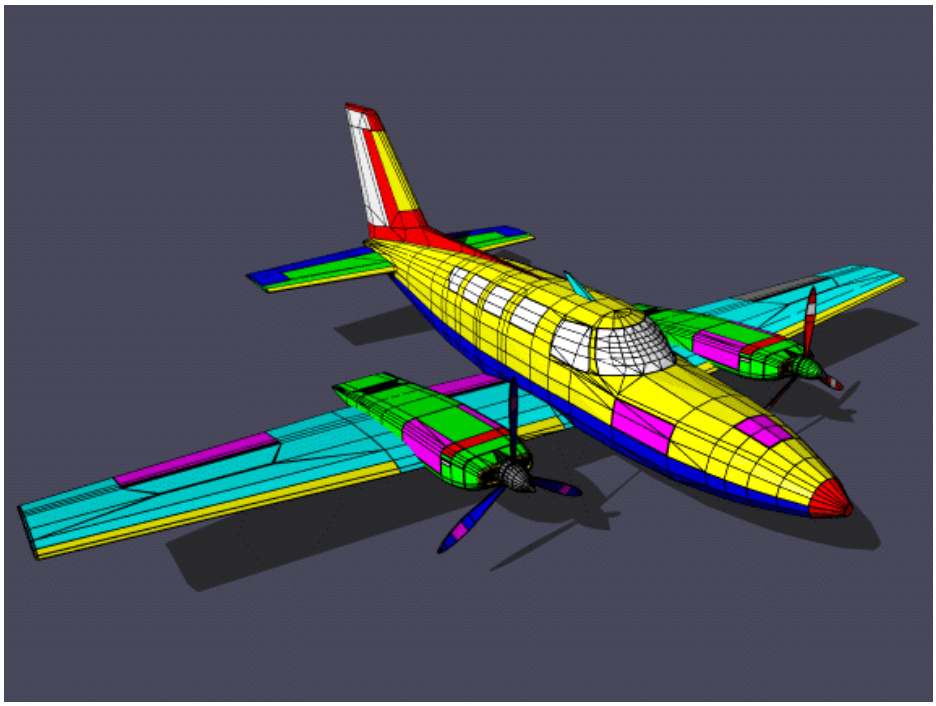


GIGA Diego Gutiérrez y Francisco J. Serón

Sombreado de polígonos



Sombreado de polígonos



GIGA
Diego Gutiérrez y Francisco J. Serón

Sombreado de polígonos

- Los modelos de iluminación que hemos desarrollado calculan la intensidad de la luz **en un punto aislado** de la superficie
- ¿Cómo lo calculamos **para todo el polígono**?

- Diferentes enfoques:
  - Repetimos el cálculo para todos los puntos del polígono (innecesario, poco práctico, nada inteligente)
  - *Flat shading:* calculamos en un punto del polígono (generalmente el centroide) y con el valor obtenido pintamos todo el polígono
  - *Gouraud y Phong:* calculamos en puntos clave del polígono (vértices) e interpolamos

Flat shading (sombreado constante)

- En inglés, a.k.a *constant shading*
- El método más rápido y sencillo

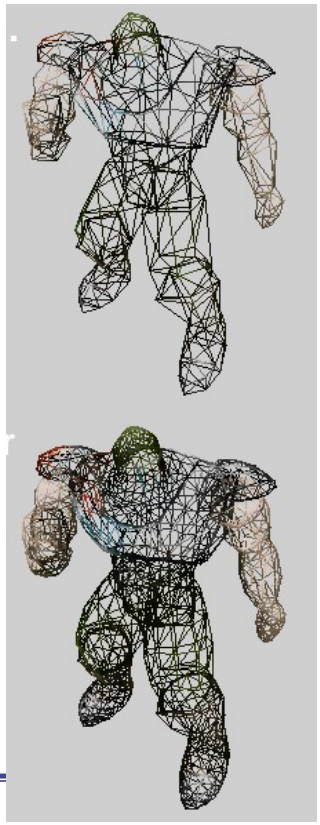
- El método menos realista
- *No pain, no gain*

- El centroide (o centro de gravedad) no tiene por qué caer en el polígono (centroide de un donuts?)

$$centroid = \frac{1}{vertices} \sum_{i=1}^{vertices} \overline{p}_i$$

Flat shading (sombreado constante)

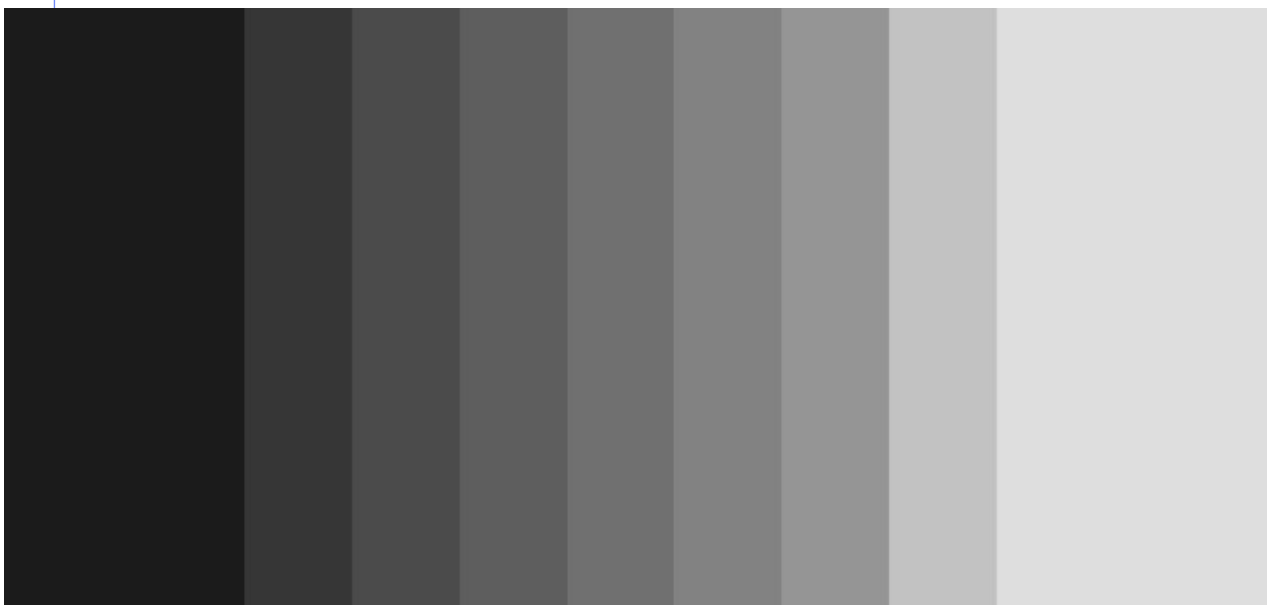
- En realidad, para fuentes puntuales, la dirección de la fuente de luz no es constante sobre todo el polígono
- Para reflejos especulares, la dirección del ojo tampoco
- Esto hace que la estructura poligonal interna se haga evidente
- Podemos refinar más la malla para disimularla un poco



Mach banding

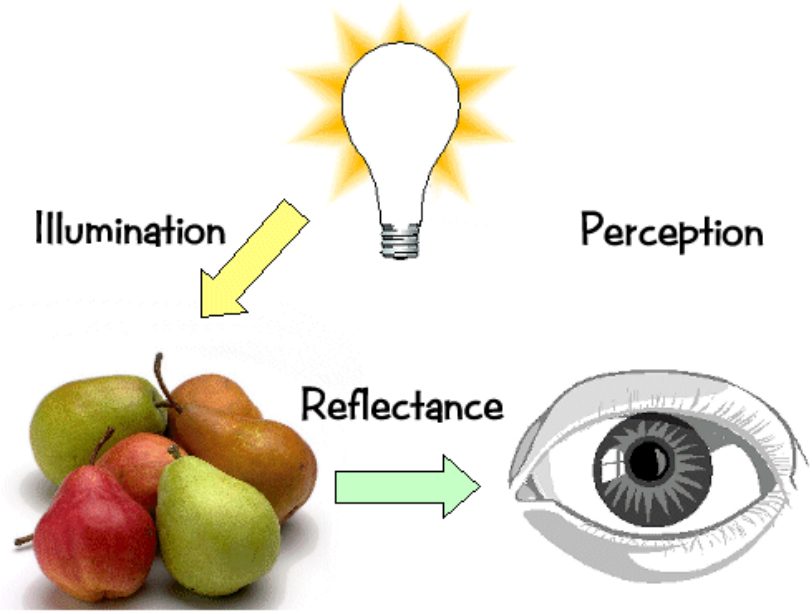
- Además, aparece el problema del *Mach banding*
  - El sistema visual es muy bueno detectando aristas... aunque éstas no existan!
  - Discontinuidades C1
  - Esto hace que las diferencias de sombreado entre polígonos parezcan todavía mayores
  - Y aunque usáramos el horriblemente ineficiente método de calcular en cada punto del polígono, el resultado sería pobre

Mach banding



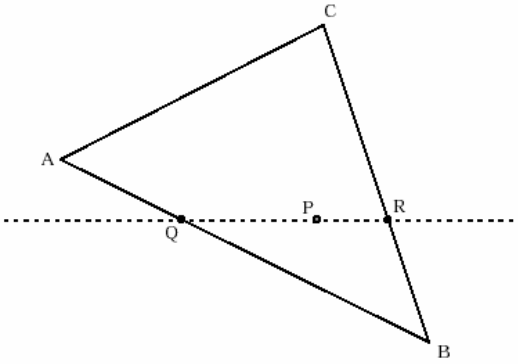
Mach banding

- (Un ejemplo más de que no todo es interacción luz-materia...)



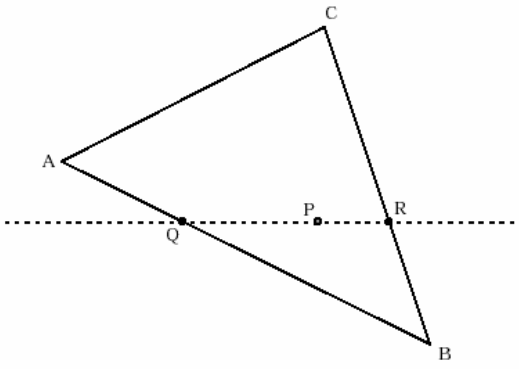
Interpolación

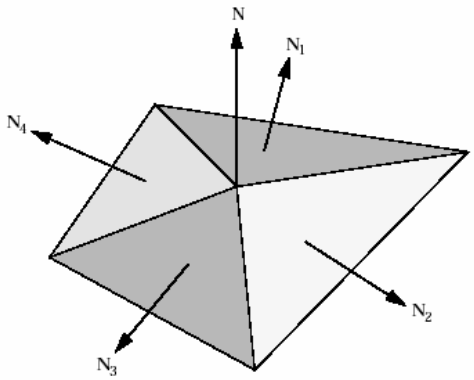
- Para eliminar estos efectos de *banding* y discontinuidad en el sombreado, vamos a aplicar distintas técnicas de interpolación
- El problema a resolver es: dado un polígono y una *scanline,* determinar la intensidad de un punto P interior



Sombreado de Gouraud

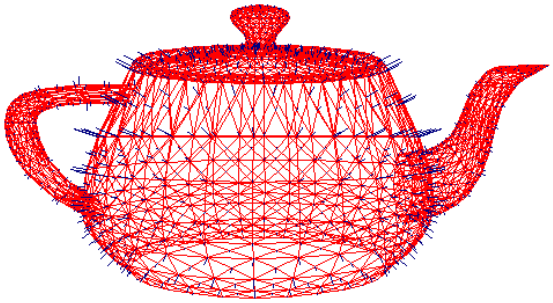
- *Gouraud shading* se basa en interpolar las intensidades obtenidas en cada vértice
- Primero calculamos esa intensidad en A,B,C
  - Necesitamos calcular la normal en cada vértice
  - La aproximamos promediando la normal a la superficie en los polígonos que comparten ese vértice

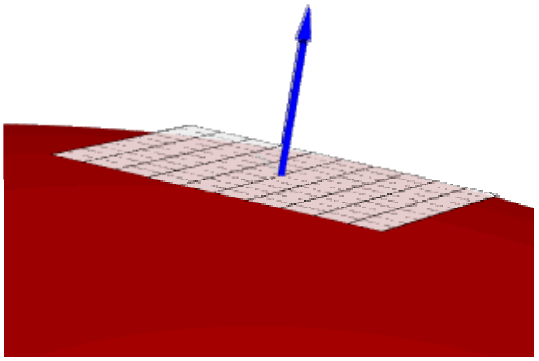




Sombreado de Gouraud

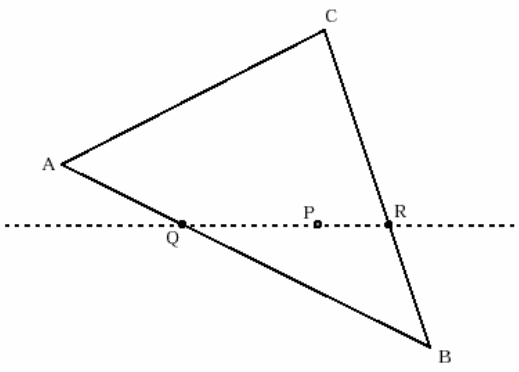
- Equivale a hallar la normal a la superficie *real* que aproximan los polígonos, asumiendo que éstos son una aproximación *piecewise* de la superficie real (C0)
- Las normales proporcionan información sobre el plano tangente a la superficie en cada punto (C1)
- Las normales en los vértices se usan sólo para iluminación (no *backface culling* u otros cálculos geométricos)





Sombreado de Gouraud

- Primero calculamos la intensidad en Q y R (intersección del *scan-line* con el polígono)
- Y luego interpolamos entre I_Q e I_R para obtener I_p



$$I_Q = uI_B + (1-u)\,I_A \quad \text{where } u = \frac{AQ}{AB}$$

$$I_R = wI_B + (1-w)\,I_C \quad \text{where } w = \frac{CR}{CB}$$

$$I_P = vI_R + (1-v)\,I_Q \quad \text{where } v = \frac{QP}{QR}$$

Sombreado de Gouraud

- Pequeño *hack*: usar cálculo incremental para acelerar. Para dos píxeles p_1 y p_2 situados sobre v_1 y v_2 respectivamente en la *scan-line,* tenemos que: $I_{p_2} = v_2 I_R + (1 - v_2) I_Q$

and

$$I_{p_1} = v_1 I_R + (1 - v_1) I_Q$$

Subtracting I_{p_1} from I_{p_2} gives

$$I_{p_2} = I_{p_1} + (I_R - I_Q)(v_2 - v_1)$$

Writing $(I_R - I_Q)$ as ΔI and $(v_2 - v_1)$ as Δv, we have

$$I_{p_2} = I_{p_1} + \Delta I \Delta v$$

- ...con lo que sólo hace falta calcular ΔI y Δv una vez por *scanline*, reduciendo los cálculos por pixel a una suma

Sombreado de Gouraud

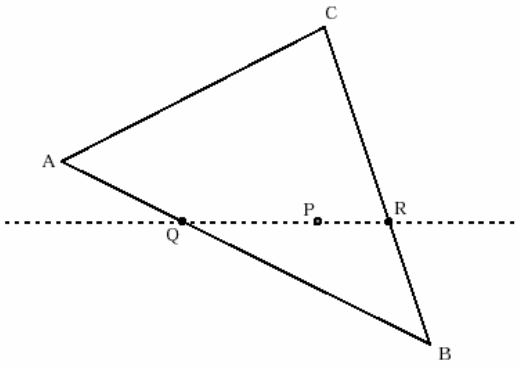
- Las discontinuidades entre polígonos adyacentes desaparecen (la intensidad “al final de un polígono” a la que llegamos interpolando será la intensidad “al comienzo del polígono siguiente”

Sombreado de Gouraud

- Problemas de Gouraud:
  - Las aristas tienden a desaparecer
  - Los reflejos especulares pueden perderse si caen dentro de un solo polígono
  - Como hemos visto, las bandas de Mach no se eliminan del todo
- Para “recuperar” una arista que queremos que sea visible, calculamos dos normales al vértice, una para cada lado. Cada normal se halla promediando las normales a la superficie de los polígonos del lado correspondiente

Sombreado de Phong

- No confundir con el modelo de iluminación de Phong
- Interpolamos linealmente las normales de la superficie a lo largo del polígono (no las intensidades como Gouraud)
- Aparte de eso, el concepto es similar:



$$\mathbf{n}_Q = u\mathbf{n}_B + (1-u)\,\mathbf{n}_A$$
$$\mathbf{n}_R = w\mathbf{n}_B + (1-w)\,\mathbf{n}_C$$
$$\mathbf{n}_P = v\mathbf{n}_R + (1-v)\,\mathbf{n}_Q$$

Sombreado de Phong

- Una vez interpoladas las normales, se aplica el modelo de iluminación con la normal interpolada, para hallar la intesidad de luz en el punto requerido

- Apariencia más real. Los reflejos especulares se capturan mejor. Las bandas de Mach se reducen
- Requiere más cálculos (un cálculo de iluminación completo por cada interpolación)

Cheap Phong

- O **interpolación mediante el producto escalar**
- Es un compromiso entre interpolación de intensidades (Gouraud) y de normales (Phong)
- Las magnitudes a interpolar entre los extremos de una *scanline* son los productos escalares **N·L** y $(\mathbf{R}\cdot\mathbf{V})^n$

Coste computacional

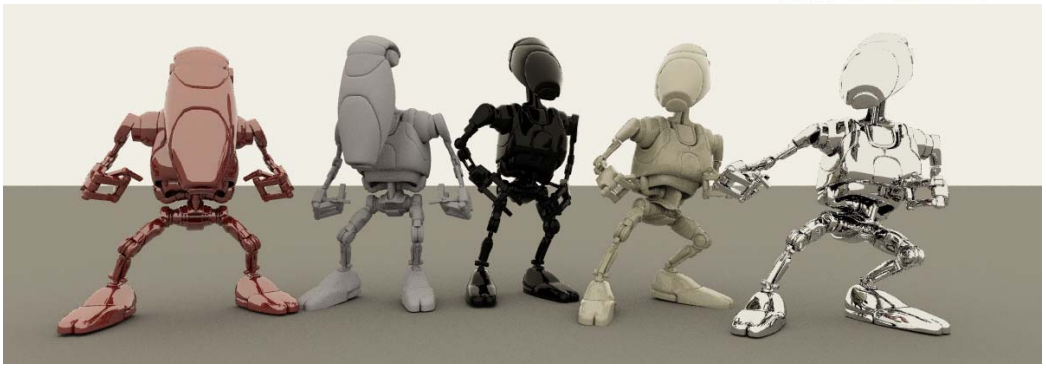
- Cálculos por polígono.

$$I = k_d I_a + \frac{I_p}{d + d_0}\left[k_d(\mathbf{N} \bullet \mathbf{L}) + k_s(\mathbf{R} \bullet \mathbf{V})^n\right]$$

- Flat shading:
  - Una sola vez
- Gouraud
  - $\Sigma_{scanlines}$pixels por scanline (Todo el modelo)
- Cheap Phong
  - $\Sigma_{scanlines}$pixels por scanline (**N·L** y **R·V**)
  - $\Sigma_{scanlines}$pixels por scanline (Todo el modelo)
- Phong
  - $\Sigma_{scanlines}$pixels por scanline (**N**)
  - $\Sigma_{scanlines}$pixels por scanline (**N·L** y **R·V**)
  - $\Sigma_{scanlines}$pixels por scanline (Todo el modelo)

Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

GIGA



Modelado Visual y Animación

Shading (in a nutshell)

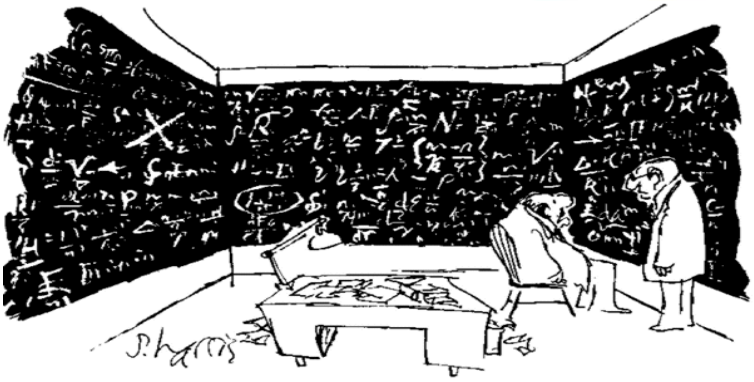
Diego Gutiérrez y Francisco J. Serón

Shading (in a nutshell)

Shading (in a nutshell)

- Hemos obtenido un modelo local de iluminación que incluye:
  - Fuentes puntuales
  - Reflexión difusa
  - Reflexión especular
- El modelo no incluye:
  - Fuentes de área
  - Reflexiones indirectas (especular-especular; especular-difuso; difuso-especular; difuso-difuso)
- Estas reflexiones indirectas se simulan con la iluminación ambiental (puagh!)
- Los modelos poligonales pueden simular la apariciencia de superficies curvas
- Hemos modelado el color con tres muestras (RGB)

Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza



"Whatever happened to *elegant* solutions?"

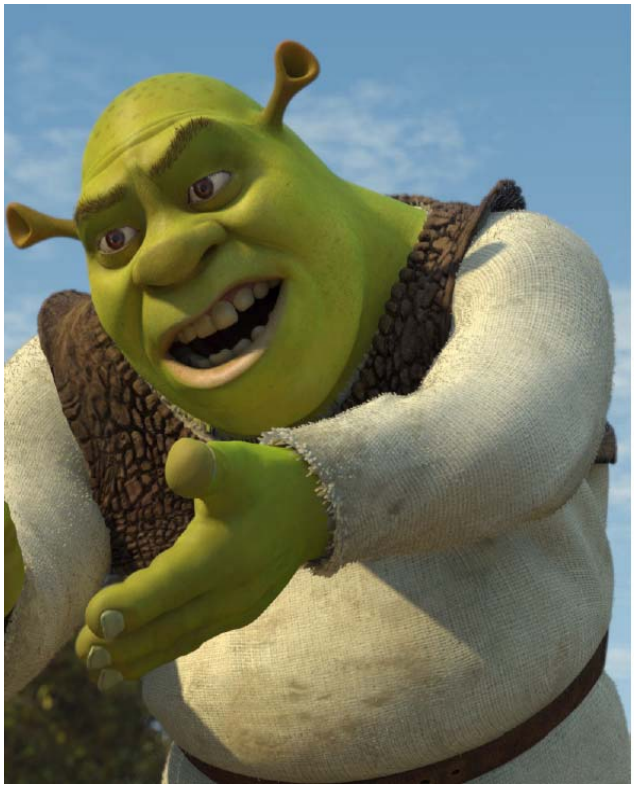
Iluminación Global

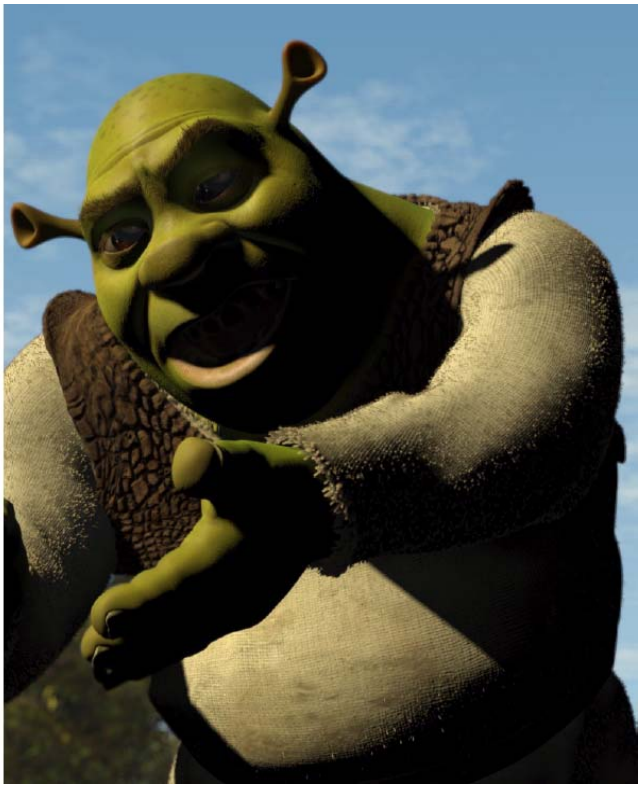
Diego Gutiérrez y Francisco J. Serón

Indice

- Qué es Iluminación Global
- Introducción a varios algoritmos de IG

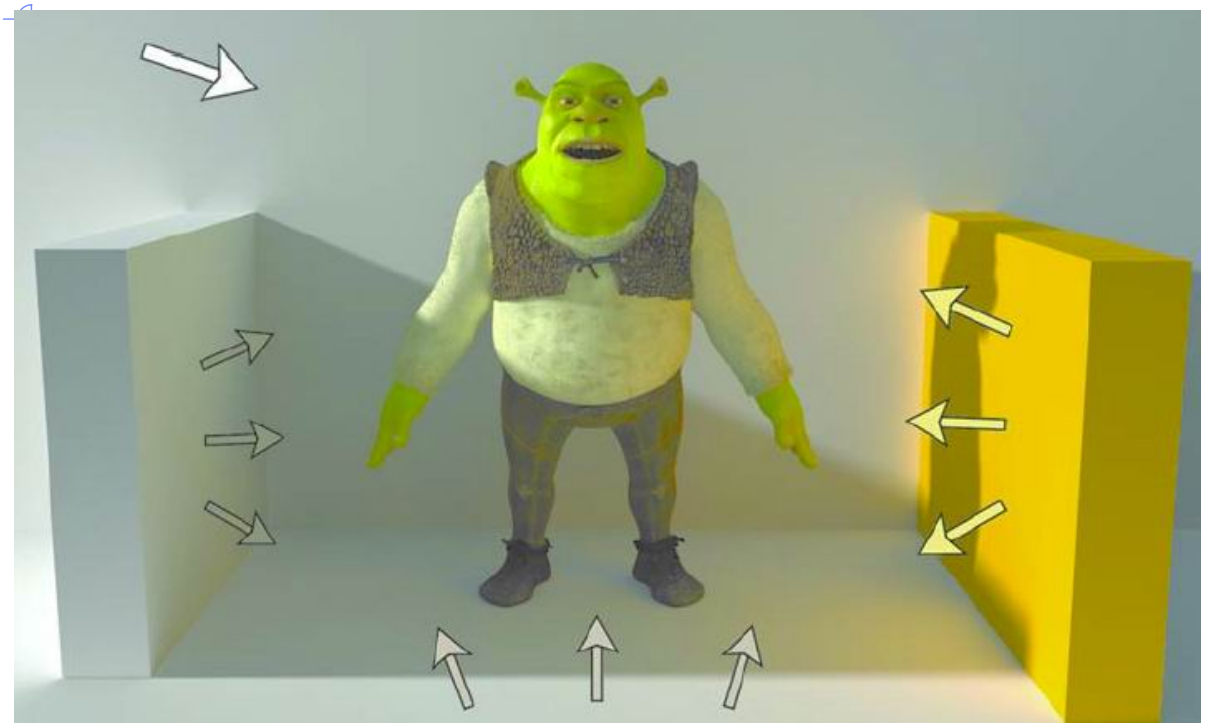
Qué es Iluminación Global



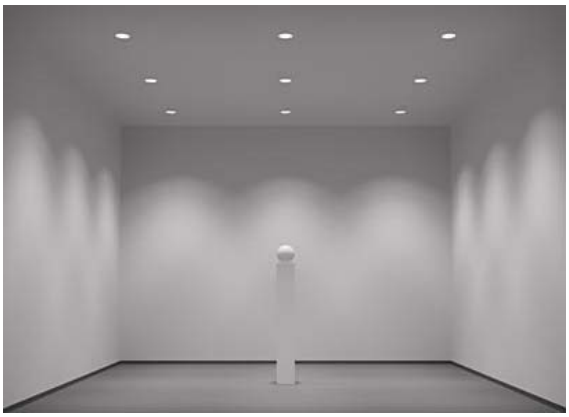


Qué es Iluminación Global

Qué es Iluminación Global



Qué es Iluminación Global



GIGA
Diego Gutiérrez y Francisco J. Serón

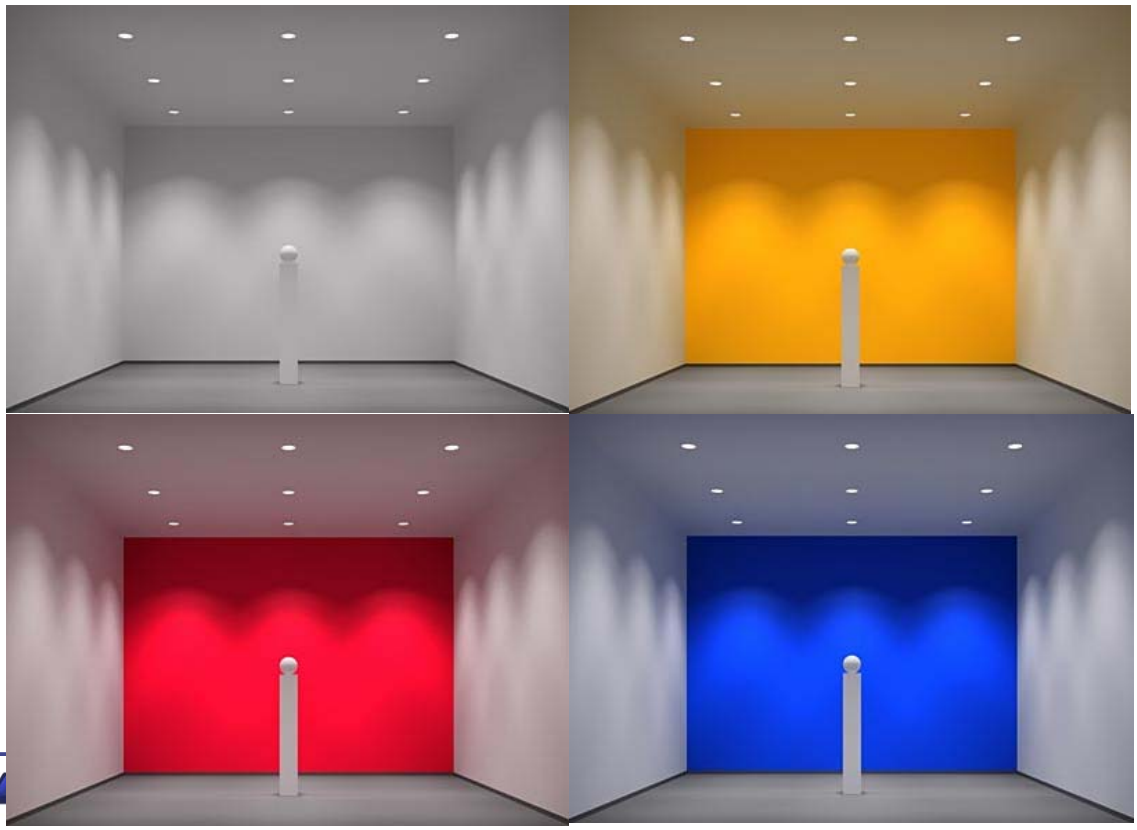
Qué es Iluminación Global

Qué es Iluminación Global

GIGA
Diego Gutiérrez y Francisco J. Serón

Qué es Iluminación Global

Qué es Iluminación Global



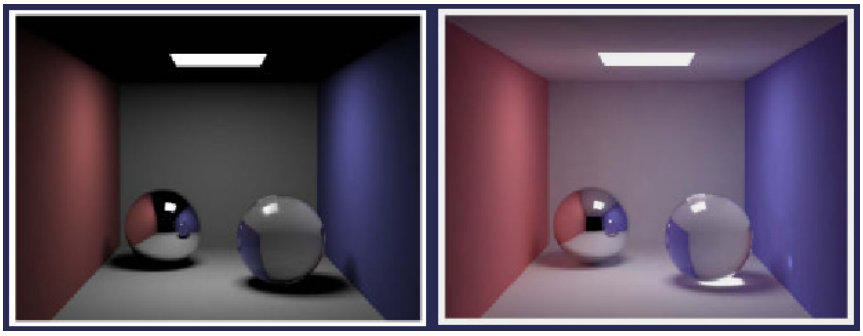
Qué es Iluminación Global

- Monte Carlo = RUIDO!

Qué es Iluminación Global

- Ejemplo con photon mapping [Wann Jensen]

Qué es Iluminación Global



- Iluminación directa
  - Fuentes de luz discretas
  - Cálculos de iluminación eficientes basados en vectores de luz y superficies (trucos!)
- Iluminación indirecta
  - Luz reflejada, dispersada, cáusticas...
  - Cálculos basados en la física del transporte de luz en la escena
  - BRDF's (*Bidireccional Reflective Distribution Function*)

Qué es Iluminación Global

- (Esto es una cáustica)

Luz indirecta

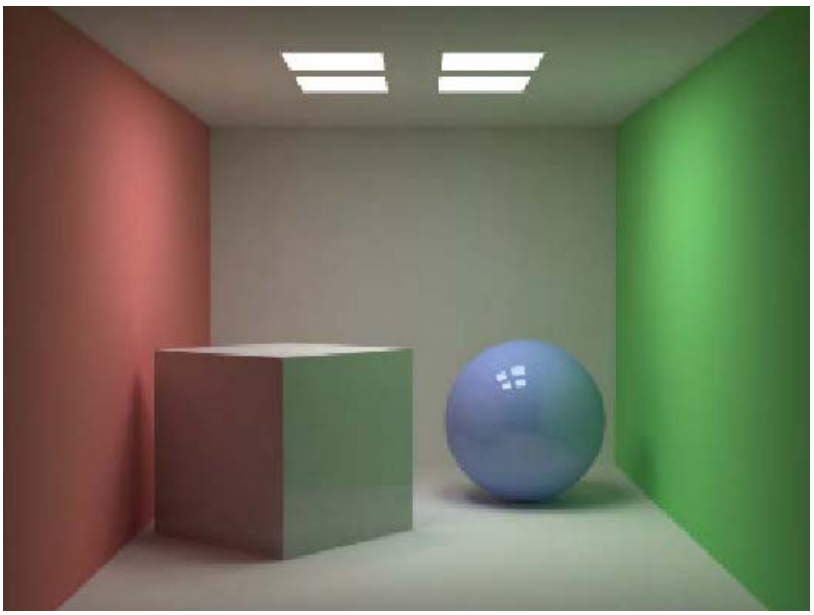
- Museo de Arte Moderno de San Francisco

Luz indirecta

- *Color bleed*

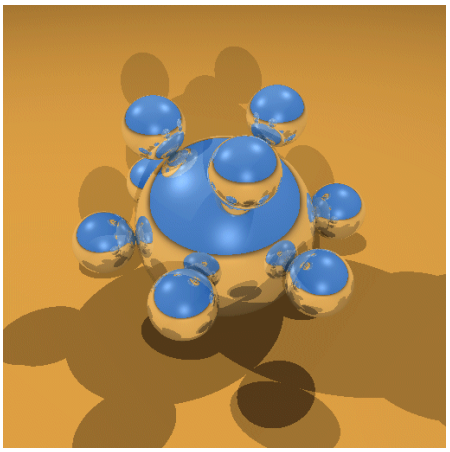
Luz indirecta

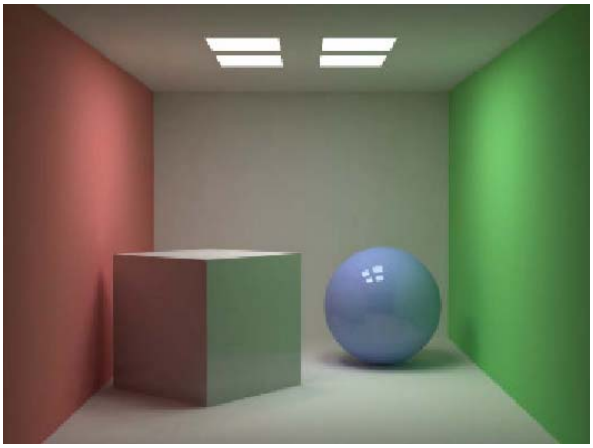
- *Color bleed*



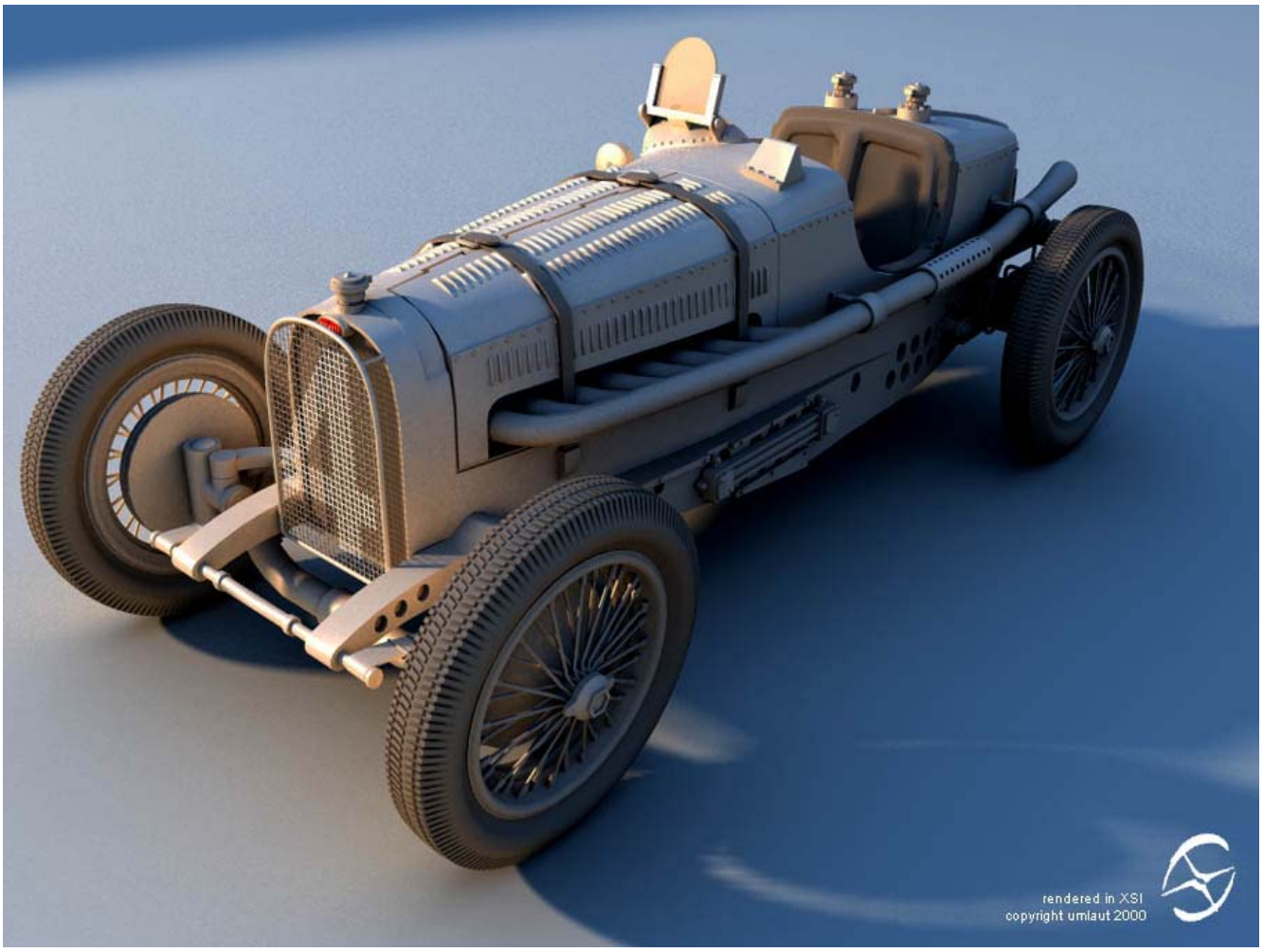
Luz indirecta

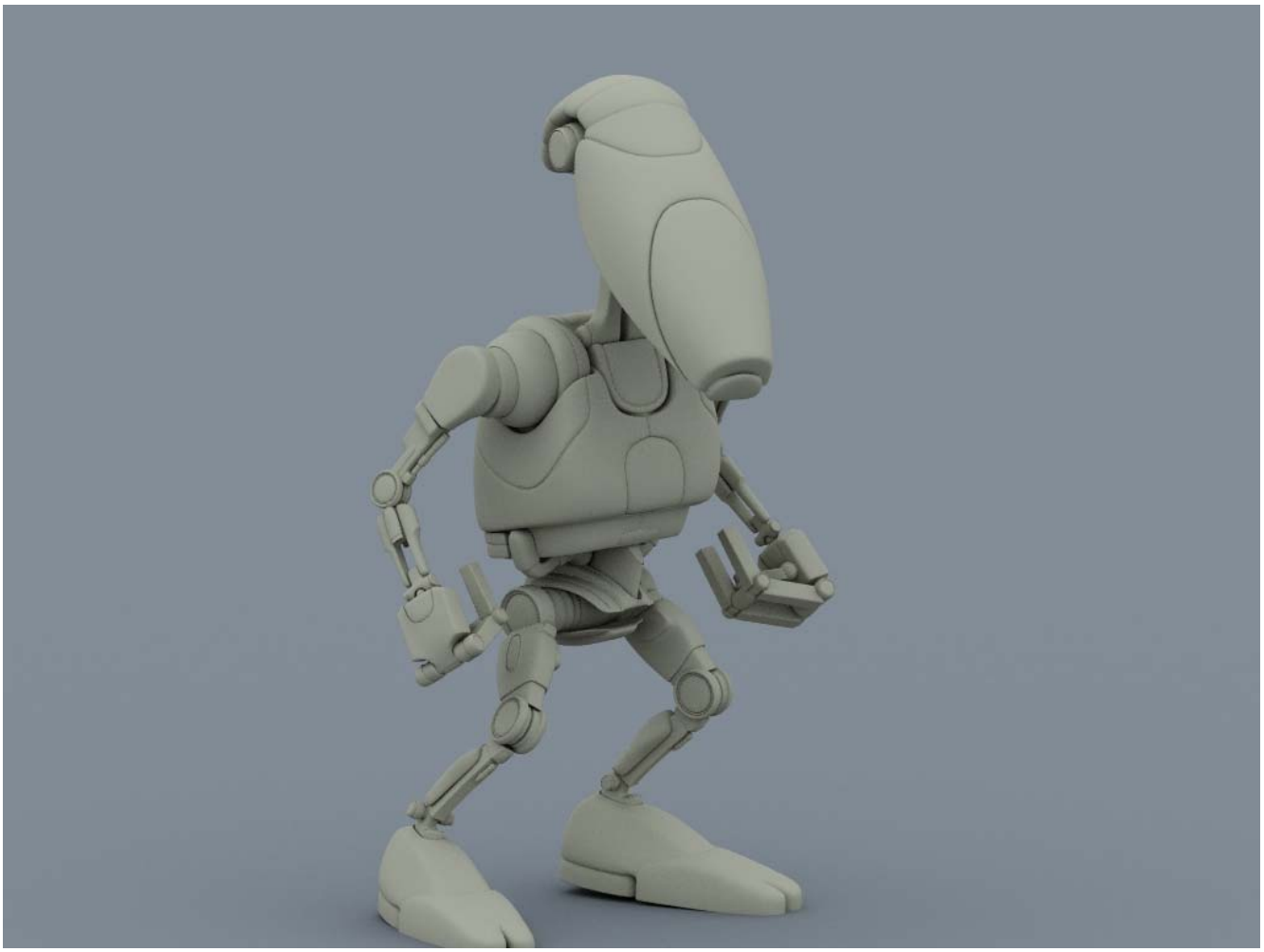
- Comparando con modelos locales, la mejora es evidente





- Decid adiós a las bolas de navidad de los años 80...


rendered in XSI
copyright umlaut 2000




www.panzar.com

Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

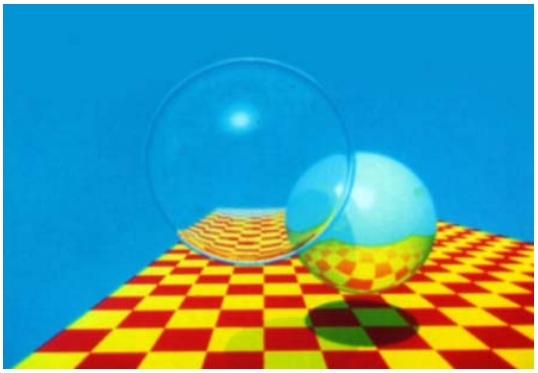
GIGA

Ray Tracing

Diego Gutiérrez y Francisco J. Serón

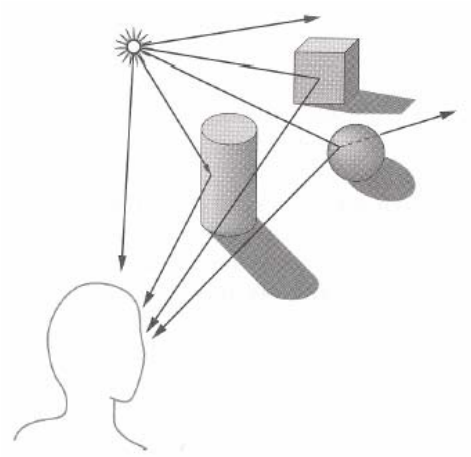
Indice

- Ray tracing: la idea
- Esquema del algoritmo
- Ejemplos
- Características
- Procedimiento
- Sombras
- Estructura de árbol
- Intersecciones



Ray tracing: la idea

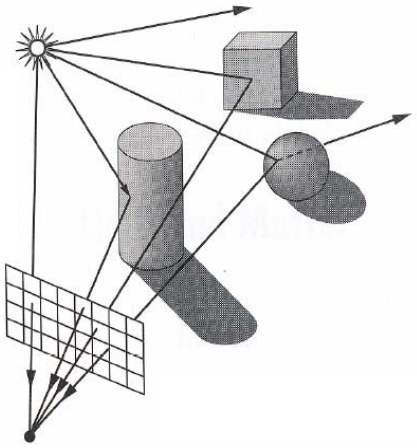
- En la vida real: de la luz al objeto al ojo



- En un trazador de rayos: del ojo al objeto a la luz
- ¿Por qué?

Ray tracing: la idea

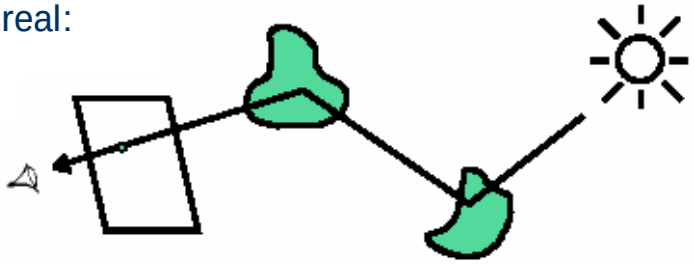
- El trazar rayos desde las fuentes del luz al ojo es altamente ineficiente!



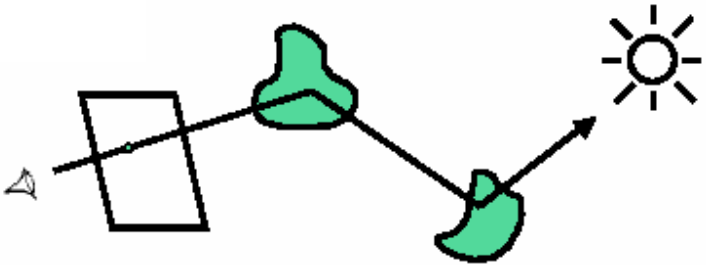
- Por eso se denominó originalmente **trazado de rayos inverso**

Ray tracing: la idea

- Fenómeno real:

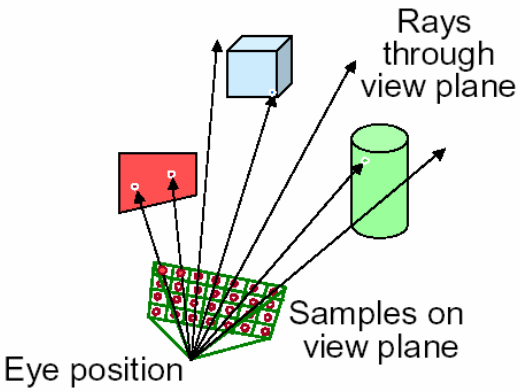


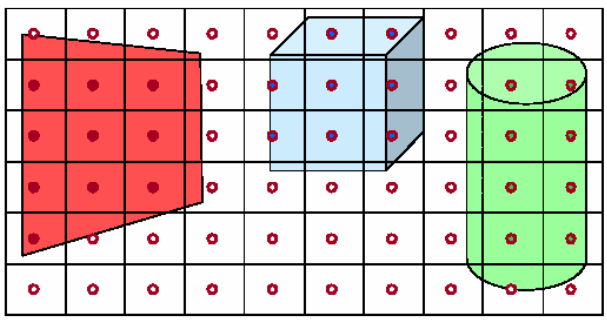
- Ray tracing:



Esquema del algoritmo

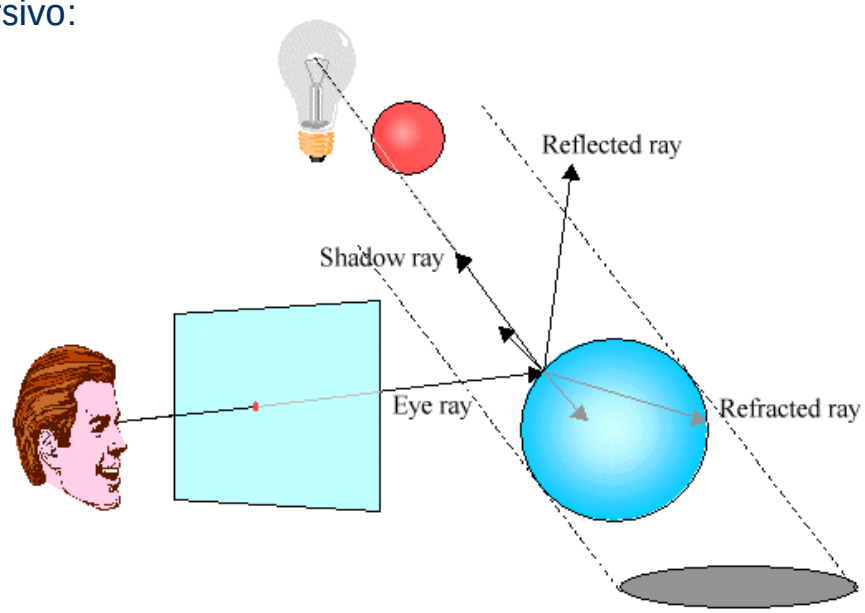
- Empecemos por un algoritmo más sencillo: **ray casting**





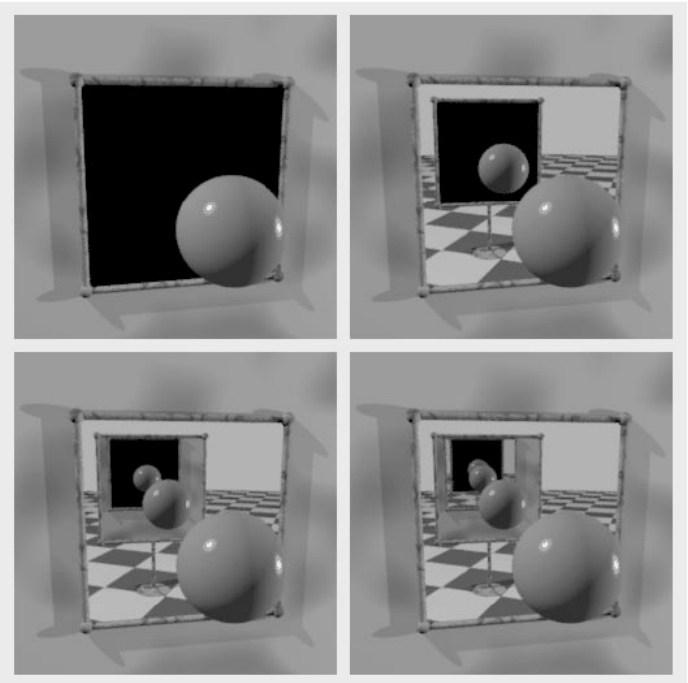
Esquema del algoritmo

- Ray tracing puede entonces considerarse como un ray casting recursivo:



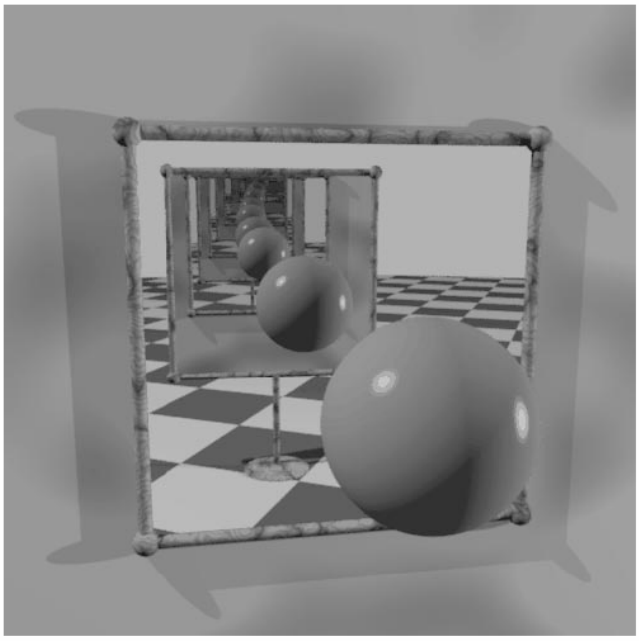
Esquema del algoritmo

- El nivel de recursividad puede afectar al resultado!
- En el ejemplo, el nivel de recursividad es 0,1,2 y 3 respectivamente



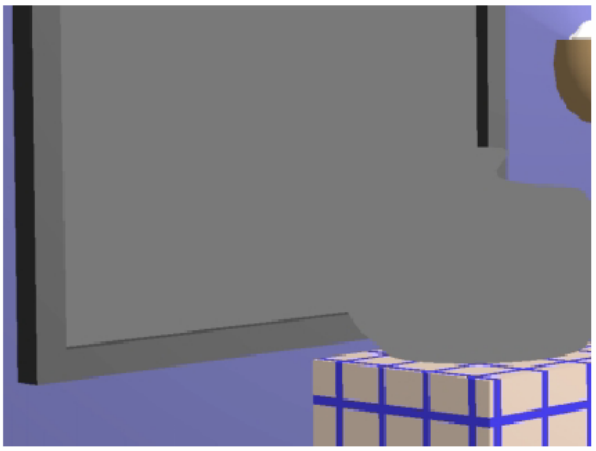
Esquema del algoritmo

- La recursividad infinita es imposible



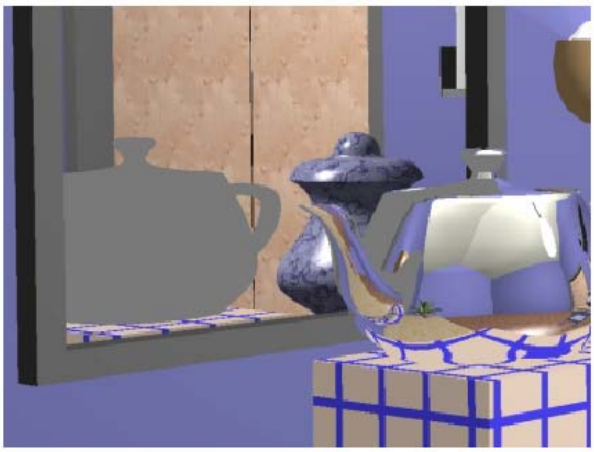
Otro ejemplo

- Profundidad 1 (=recursividad 0)



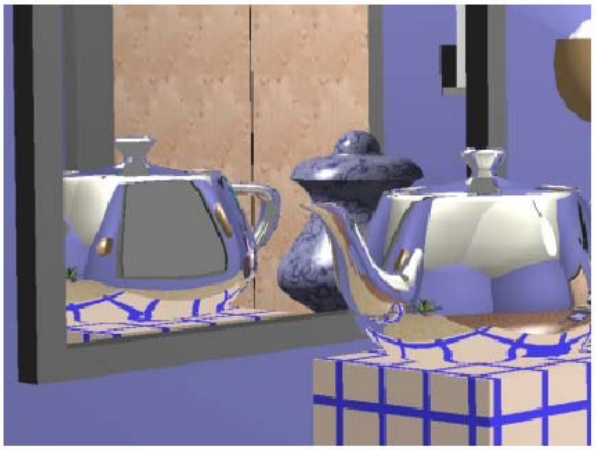
Otro ejemplo

- Profundidad 2 (=recursividad 1)



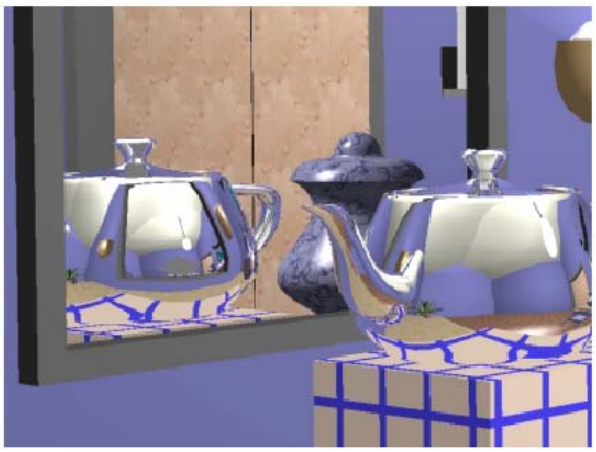
Otro ejemplo

- Profundidad 3 (=recursividad 2)



Otro ejemplo

- Profundidad 4 (=recursividad 3)



Características

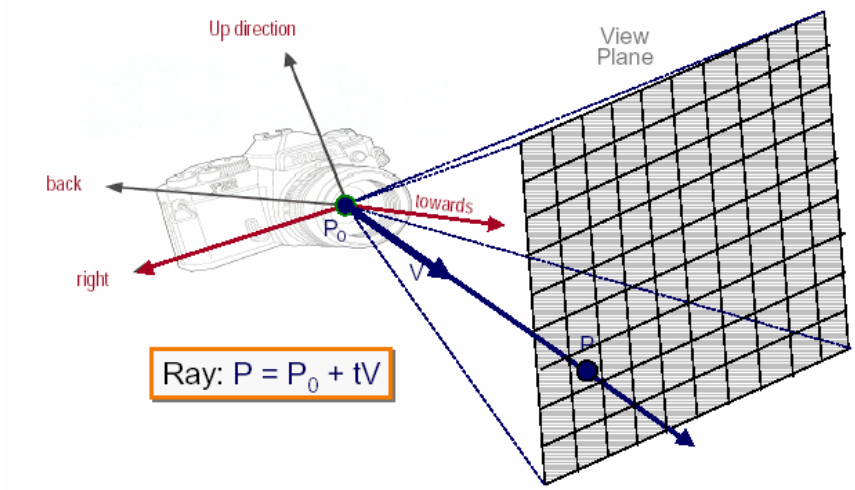
- Sigue la energía de la luz a través de rayos
- Es un procedimiento recursivo
- Depende de la posición del observador
- Aplicable a diferentes tipos de superficies (esferas, conos...), no sólo polígonos

Características

- **Ventajas:**
  - Reflejos
  - Transparencias y refracciones
  - Sombras
  - Diseño conceptualmente sencillo
- **Inconvenientes:**
  - No permite una solución completa al problema de la iluminación global
  - Utiliza aproximaciones a la solución real

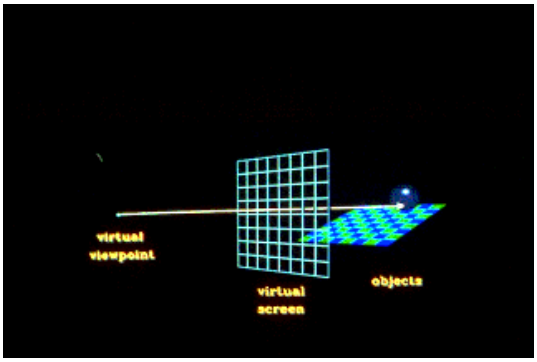
Procedimiento (I)

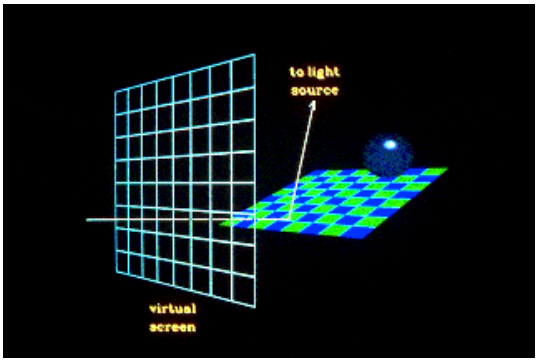
- Los rayos se lanzan **desde el punto de vista:**
  - El plano de visualización se discretiza en una matriz de pixels
  - Se lanza un rayo primario (*primary ray, a.k.a. eye ray*) a través de cada pixel



Procedimiento (II)

- Se encuentra la **intersección** del rayo primario con el objeto más cercano al punto de vista
  - Ese objeto representa la superficie visible a través de ese pixel
  - (Acabamos de dar con un algoritmo de eliminación de superficies ocultas)

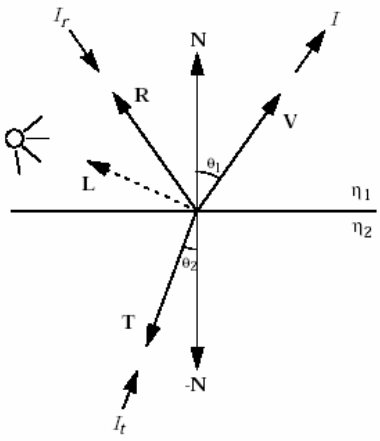




Procedimiento (III)

- Ahora calculamos la **intensidad de luz** en el punto de intersección de la superficie
  - Esta intensidad tendrá una componente de iluminación local más una componente global (energía reflejada y energía transmitida)

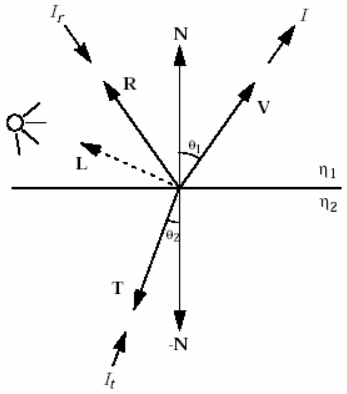
$$I = k_d I_a + k_d \sum_{i=1}^{m} I_{p_i}\left[(\mathbf{N} \bullet \mathbf{L}_i) + k_s (\mathbf{R} \bullet \mathbf{V})^n\right] + k_r I_r + k_t I_t$$



Procedimiento (III)

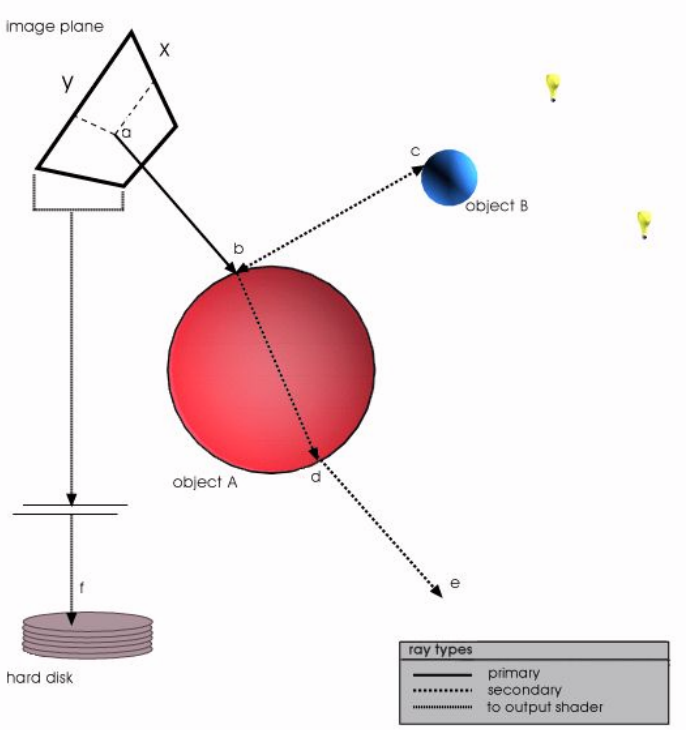
- K_r es el **coeficiente global de reflexión especular** (generalmente igual a K_s)
- K_t es el **coeficiente global de transmisión especular**
- I_r e I_t son las intensidades en las direcciones **R** y **T**

$$I = k_d I_a + k_d \sum_{i=1}^{m} I_{p_i}\left[(\mathbf{N} \bullet \mathbf{L}_i) + k_s (\mathbf{R} \bullet \mathbf{V})^n\right] + k_r I_r + k_t I_t$$



Procedimiento (IV)

- Los términos globales se obtienen lanzando **rayos secundarios** desde la intersección
  - Se calculan las nuevas intersecciones de estos rayos secundarios con los objetos de la escena
  - Se calculan en los nuevos puntos la iluminación
- Proceso **recursivo**



Recursividad

- Como hemos visto, el proceso podría repetirse recursivamente hasta el infinito (teóricamente)
- Necesidad de cortar en algún momento! (profundidad de rayos)

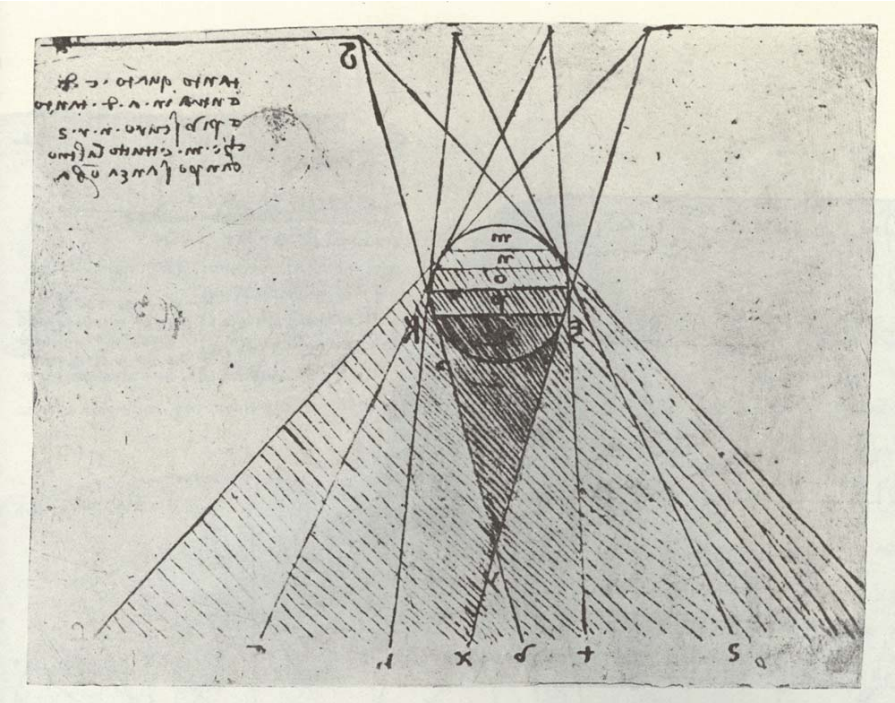
- K_r=0 implica superficie difusa, no hay rayo reflejado
- K_t=0 implica superficie opaca, no hay rayo transmitido

- *Muchos renderizadores comerciales sólo hacen trazado de rayos en los pixels en los que es necesario*

Sombras

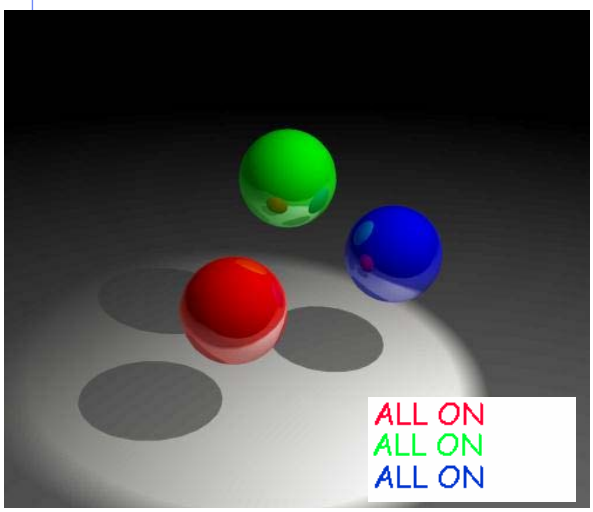
- ¿Qué sucede si un objeto B se interpone entre el punto de intersección del objeto A con el rayo y la fuente de luz?
  - Surgen las **sombras**
- Para calcularlas, lanzamos **rayos de sombra** (*shadow rays*)
  - Origen: intersección rayo-objeto
  - Dirección: hacia la fuente de luz
- Testeamos la intersección de este rayo con los objetos
  - Si encontramos una intersección y el punto de intersección está antes de la luz, el punto origen está en sombra
- Si hay muchas fuentes de luz, se lanza un rayo de sombra a cada una

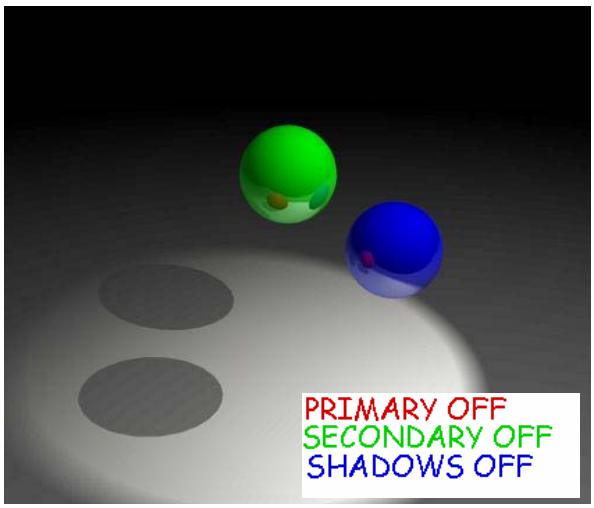
Sombras



Rayos primarios, secundarios y de sombra

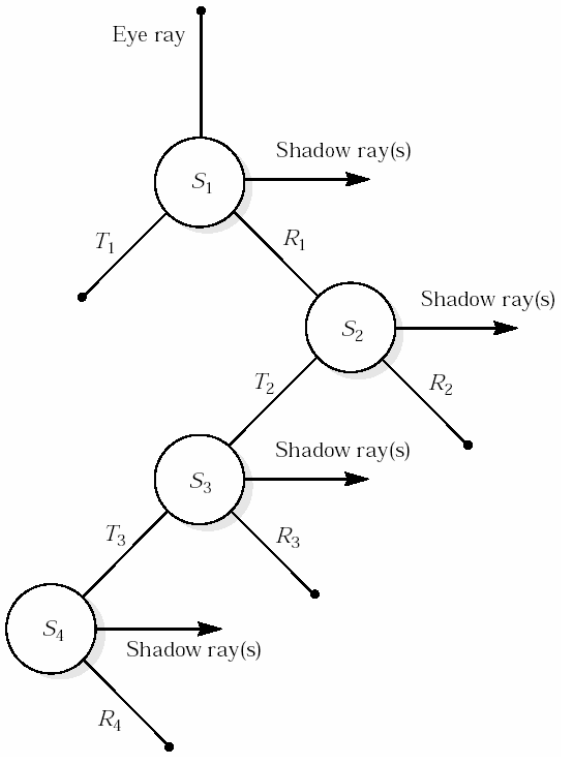
- Podemos conseguir curiosos efectos haciendo invisibles los objetos a algún tipo de rayo:





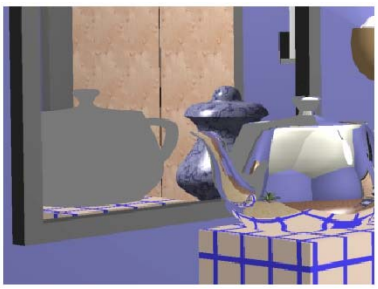
Estructura de árbol

- El proceso recursivo de ray tracing puede visualizarse como un árbol de rayos



Estructura de árbol

- Necesitamos especificar la profundidad máxima
  - Si no, perderemos mucho tiempo en seguir rayos cuya contribución a la imagen es despreciable
  - Pero una profundidad escasa causará errores en la imagen
- Para controlar la profundidad:
  - Si el rayo abandona la escena, debe devolver un valor de *background*
  - Fijar una profundidad máxima absoluta
  - Fijar una profundidad máxima adaptativa, según un umbral de la intensidad devuelta por un rayo secundario



Intersecciones con objetos

- El verdadero cuello de botella del ray tracing
- En principio, cada rayo generado debe ser testeado contra *todos* los objetos de la escena
- Si se encuentran intersecciones, cuál es la más cercana?
- Necesitamos algoritmos eficientes de intersección
  - Con esferas, conos, polígonos, cajas, cilindros, toros...

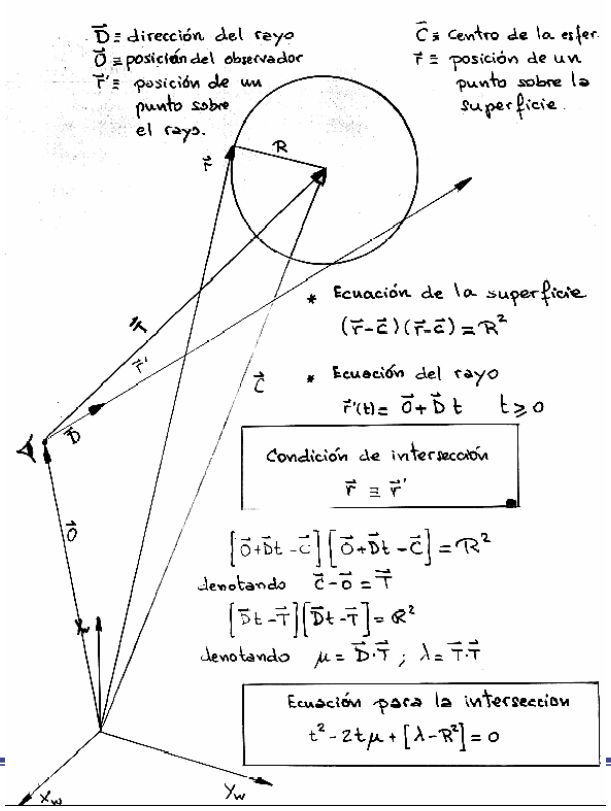
- Brainstorm: ¿se os ocurre alguna forma de evitar testear cada rayo contra *todos* los objetos de la escena, o al menos de hacerlo de una forma eficiente?

Intersección rayo-esfera

- Definimos el rayo primario de manera paramétrica:
  - $\mathbf{r}(t)=\mathbf{O}+\mathbf{D}t$
  - $\mathbf{r}(t)$ es el vector posición de un punto p
  - $\mathbf{O}$ es el vector posición del origen del rayo
  - $\mathbf{D}$ es el vector unitario en la dirección del rayo
  - t es un parámetro real que representa la distancia recorrida a lo largo del rayo
  - Buscamos puntos de intersacción con valores de t positivos

Intersección rayo-esfera

- Esquema:



Intersección rayo-esfera

$$t^2 - 2t\mu + [\lambda - R^2] = 0$$

- Resolviendo para t (ecuación de segundo grado), obtenemos
- Obtenemos:

$$t = \mu \pm \sqrt{\gamma}$$
$$\gamma = \mu^2 - \lambda + R^2$$

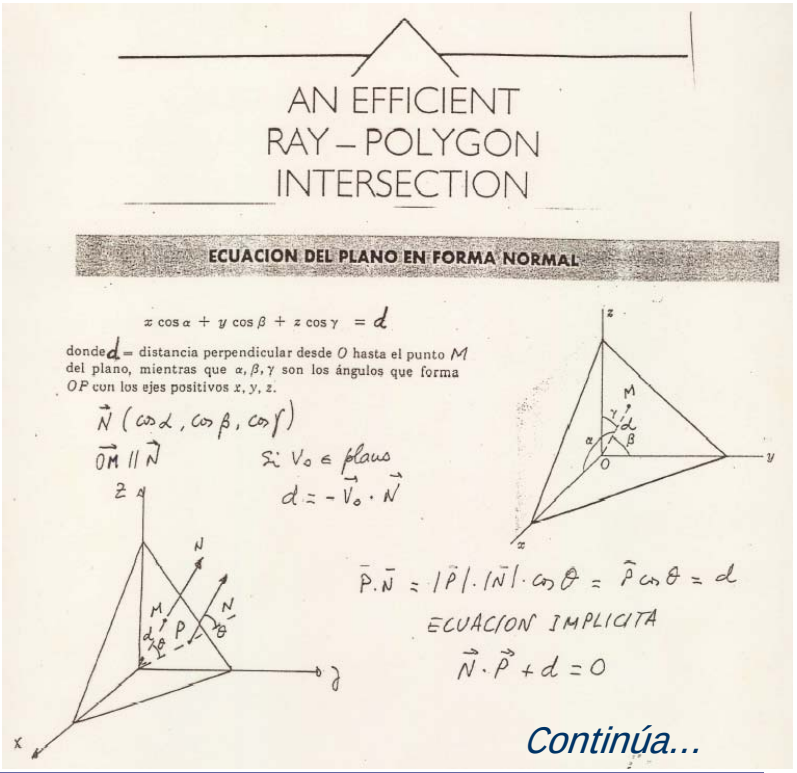
- $\gamma<0$: no hay solución; el rayo no intersecciona con la esfera
- $\gamma=0$: una solución; el rayo “roza”
- $\gamma<0$: dos soluciones: el rayo atraviesa la esfera

Intersección rayo-esfera

- Para hallar los puntos de intersección, sustituimos t en la ecuación del rayo por los valores obtenidos

Intersección rayo-polígono

- [Graphic Gems]



Intersec

- [Graphic Gems]

First Step: Intersecting the Embedding Plane

This step is common with the other intersection algorithms but can be presented again. A polygon is described by its vertices V_i ($i \in \{0, \ldots, n-1\}$, $n \geq 3$). Let x_i, y_i, and z_i the coordinates of the vertex V_i. The normal of the plane containing the polygon, $\mathbf{N}$, is computed with the cross product:

$$\vec{N} = \overrightarrow{V_0V_1} \times \overrightarrow{V_0V_2}.$$

For each point P of the plane, the quantity $P \cdot \mathbf{N}$ is constant. This constant value is computed by the dot product $d = -\vec{V_0} \cdot \vec{N}$. The implicit representation of the plane,

$$\vec{N} \cdot \vec{P} + d = 0, \qquad (1)$$

is computed once, and then stored in the polygon description.

Let the parametric representation of the ray be

$$r(t) = \vec{O} + \vec{D}t = \vec{P} \qquad (2)$$

The evaluation of the parameter t corresponding to the intersection point can be obtained using the equations (1) and (2):

$$t = -\frac{d + \vec{N} \cdot \vec{O}}{\vec{N} \cdot \vec{D}}.$$

This calculation requires 12 floating operations and three tests:

- If polygon and ray are parallel ($\vec{N} \cdot \vec{D} = 0$), the intersection is rejected.
- If the intersection is behind the origin of the ray ($t \leq 0$), the intersection is rejected.
- If a closer intersection has been already found for the ray ($t > t_{ray}$), the intersection is rejected.

Continúa...

Intersección rayo-polígono

- [Graphic Gems]

Second Step: Intersecting the Polygon

A parametric resolution is now presented. This solution is based on triangles. If a polygon has n vertices ($n > 3$), it will be viewed as a set of $n - 2$ triangles. For this reason, the resolution is restricted to convex polygons. The point P (see Fig. 1) is given by

$$\overrightarrow{V_0P} = \alpha\overrightarrow{V_0V_1} + \beta\overrightarrow{V_0V_2}.$$

The point P will be inside the triangle ($\triangle V_0 V_1 V_2$) if

$$\alpha \geq 0, \beta \geq 0, \text{ and } \alpha + \beta \leq 1.$$

Equation (4) has three components:

$$\begin{cases} x_P - x_0 = \alpha(x_1 - x_0) + \beta(x_2 - x_0) \\ y_P - y_0 = \alpha(y_1 - y_0) + \beta(y_2 - y_0) \\ z_P - z_0 = \alpha(z_1 - z_0) + \beta(z_2 - z_0). \end{cases}$$

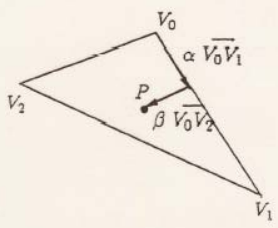


Figure 1. Parametric representation of the point P.

A solution exists and is unique.

Continúa...

Intersección rayo-polígono

- [Graphic Gems]

To reduce this system, we wish to project the polygon onto one of the primary planes, either xy, xz, or yz. If the polygon is perpendicular to one of these planes, its projection onto that plane will be a single line. To avoid this problem, and to make sure that the projection is as large as possible, we find the dominant axis of the normal vector and use the plane perpendicular to that axis. As in Snyder and Barr (1987), we compute the value i_0,

$$i_0 = \begin{cases} 0 & \text{if } |N_x| = \max(|N_x|, |N_y|, |N_z|) \\ 1 & \text{if } |N_y| = \max(|N_x|, |N_y|, |N_z|) \\ 2 & \text{if } |N_z| = \max(|N_x|, |N_y|, |N_z|). \end{cases}$$

Consider i_1 and i_2 (i_1 and $i_2 \in \{0, 1, 2\}$), the indices different from i_0. They represent the primary plane used to project the polygon. Let (u, v) be the two-dimensional coordinates of a vector in this plane; the coordinates of $\overrightarrow{V_0P}$, $\overrightarrow{V_0V_1}$, and $\overrightarrow{V_0V_2}$, projected onto that plane, are

$$u_0 = P_{i_1} - V_{0_{i_1}} \qquad u_1 = V_{1_{i_1}} - V_{0_{i_1}} \qquad u_2 = V_{2_{i_1}} - V_{0_{i_1}}$$

$$v_0 = P_{i_2} - V_{0_{i_2}} \qquad v_1 = V_{1_{i_2}} - V_{0_{i_2}} \qquad v_2 = V_{2_{i_2}} - V_{0_{i_2}}$$

Continúa...

Intersección rayo-polígono

- [Graphic Gems]

Equations 5 then reduce to

$$\begin{cases} u_0 = \alpha . u_1 + \beta . u_2 \\ v_0 = \alpha . v_1 + \beta . v_2 \end{cases}.$$

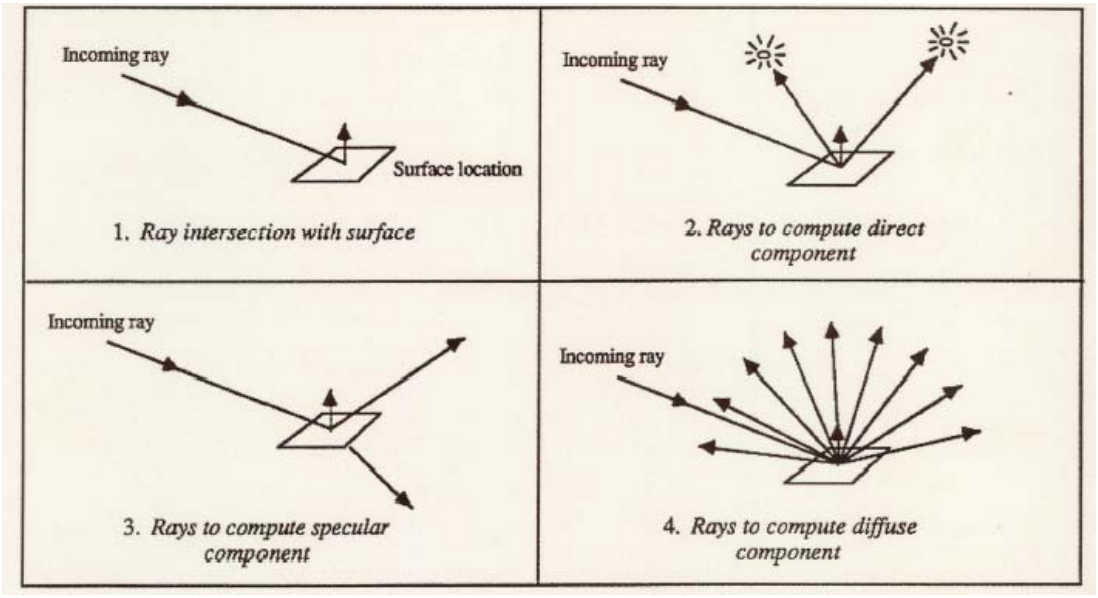
The solutions are

$$\alpha = \frac{\det\begin{pmatrix} u_0 & u_2 \\ v_0 & v_2 \end{pmatrix}}{\det\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix}} \quad \text{and} \quad \beta = \frac{\det\begin{pmatrix} u_1 & u_0 \\ v_1 & v_0 \end{pmatrix}}{\det\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix}}.$$

The interpolated normal from the point P may be computed by

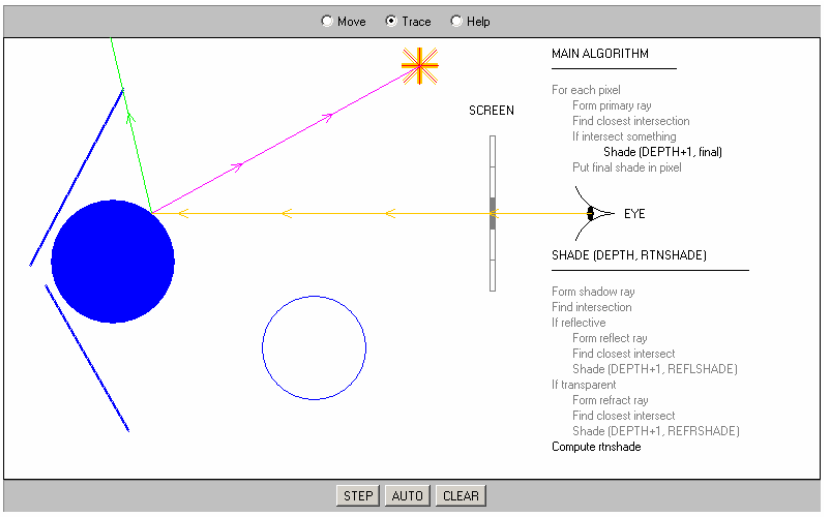
$$N_P = (1 - (\alpha + \beta))N_0 + \alpha N_1 + \beta N_2.$$

Los 4 pasos del ray tracing



Ray tracing demo

- http://www.siggraph.org/education/materials/HyperGraph/raytrace/rt_java/raytrace.html



- Yan Liu & Dr. G. Scott Owen (Georgia State University)

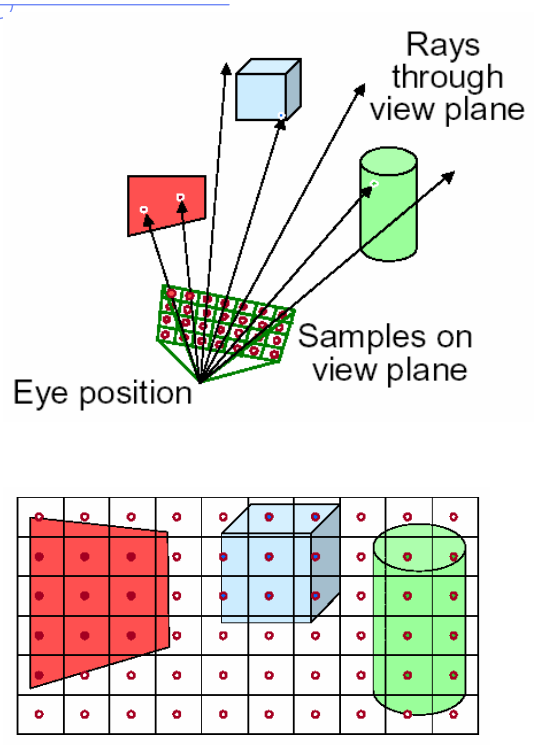
Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

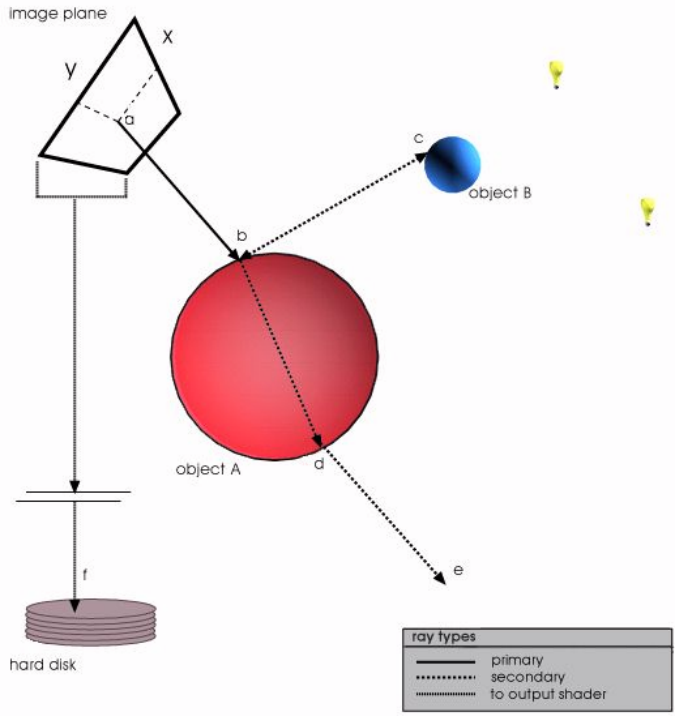
Modelado Visual y Animación

Ray Tracing (in a nutshell)

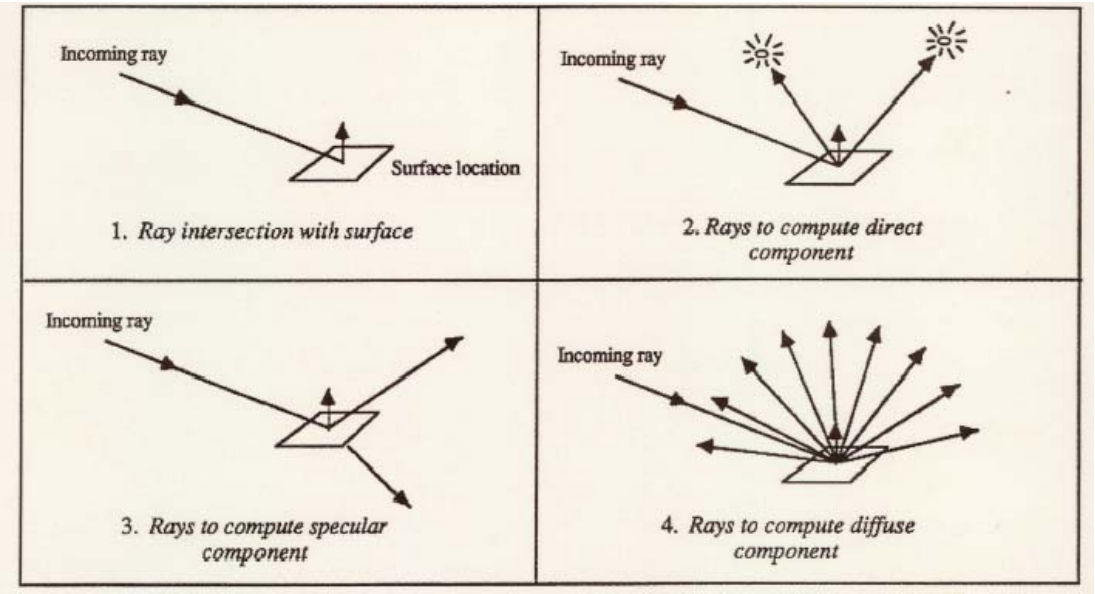
Diego Gutiérrez y Francisco J. Serón

Ray tracing (in a nutshell)





Ray tracing (in a nutshell)



Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

GIGA

Ray Tracing Reloaded
(Técnicas de aceleración)

Diego Gutiérrez y Francisco J. Serón

Introducción

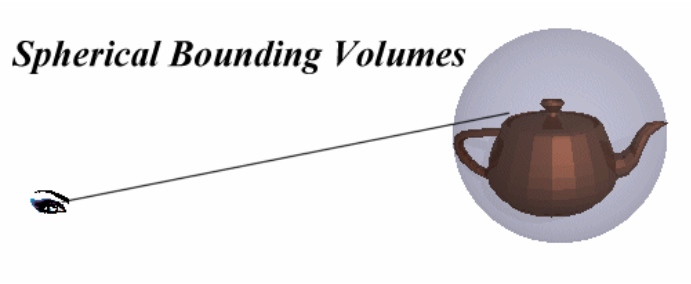
- Ray tracing, tal cual, puede ser un proceso muy lento
- Aproximadamente, el 90% del tiempo de cálculo se gasta en las intersecciones
  - Cuantos más objetos hay en la escena, peor
  - Cuanto más grande es la imagen, peor (más rayos primarios)
- Técnicas de aceleración:
  - Limitar el número de tests de intersección
  - Retrasar un test caro hasta que uno más barato haya probado su necesidad
  - ...
- Bounding volumes, subdivisión espacial
- Además: light buffers...

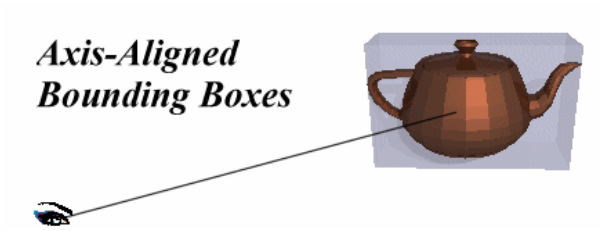
Bounding volumes

- Concepto: encerrar un objeto complicado en una geometría más simple.
  - Si el rayo no intersecta ni siquiera esa geometría envolvente, su contenido puede ser ignorado
  - Si el rayo intersecta esa geometría envolvente, puede que intersecte el objeto encerrado, o puede que no
- *Cuanto más cercano sea el volumen envolvente al objeto real, mejor*
- Los primeros *bounding volumes* utilizados en CG fueron esferas, debido a su fácil algoritmo de intersección y a que una sola basta para encerrar un objeto
- Sin embargo no dan un ajuste demasiado bueno

Bounding volumes

- Actualmente se usan mucho cajas alineadas con los ejes
- Una jerarquía de cajas es aún mejor





Bounding volumes

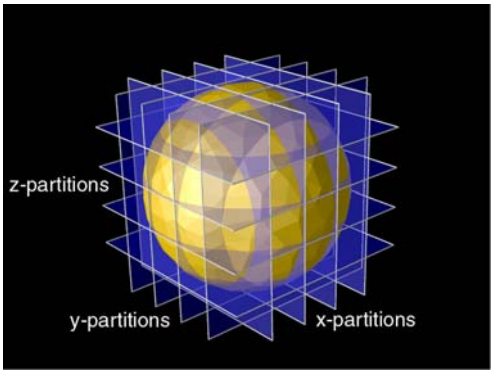
- Primero se comprueba el test de intersección con el *bounding volume*
- Si existe intersección, entonces se comprueba el objeto encerrado
- Si hay una estructura jerárquica de *bounding volumes,* se va profundizando en ella durante los tests de intersección sencillos

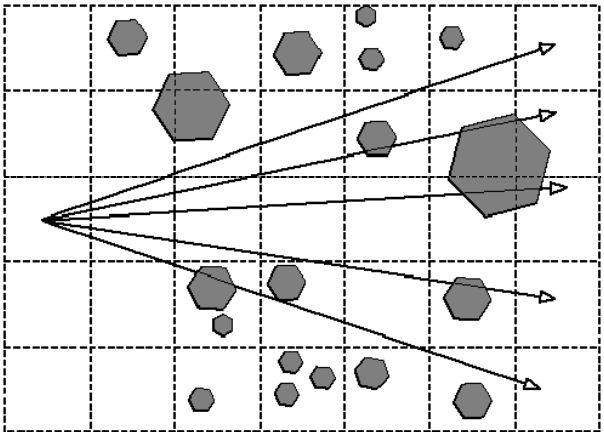
Subdivisión espacial

- Funcionan en base a la idea de *voxels:*
  - Estos voxels ocupan todo el espacio de la escena
  - No se superponen unos sobre otros
  - No tienen por qué contener completamente ningún objeto particular
- Dos tipos de subdivisión espacial:
  - Uniforme (todos los voxels son del mismo tamaño)
  - No uniforme (octrees, jerarquía de voxels...)

Subdivisión espacial

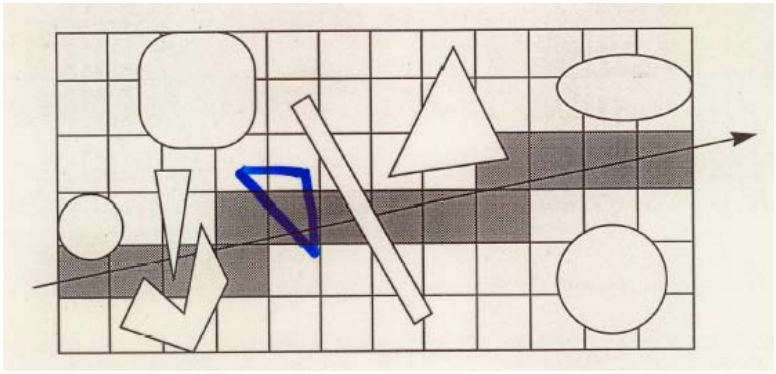
- Idea general:
  - Subdividir el espacio en regiones
  - Hacer una lista con los objetos en cada región
  - Sólo miramos las regiones por las que pasa el rayo
  - Evitar computar dos veces una intersección si el objeto cae en dos o más regiones





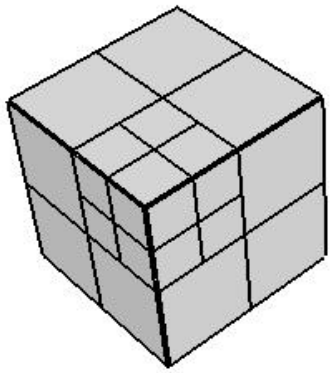
Subdivisión espacial

- ¿Qué ganamos?:
  - Tras hallar la primera intersección, todos los objetos con los que intersecta el rayo en su camino no son calculados
  - En el ejemplo, sólo 3 voxels se han tenido en cuenta para la intersección
- Básicamente, lo que hacemos es testear intersecciones sólo en las proximidades del rayo

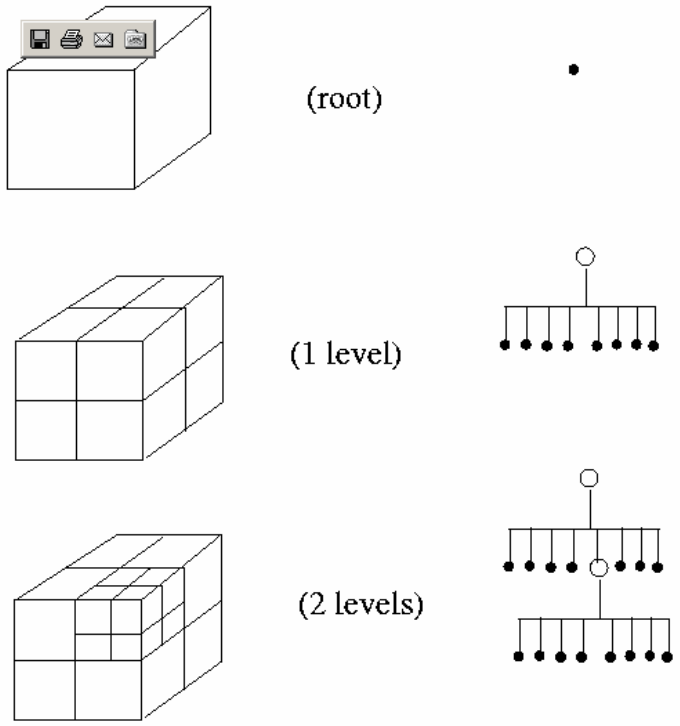


Subdivisión espacial no uniforme: *octrees*

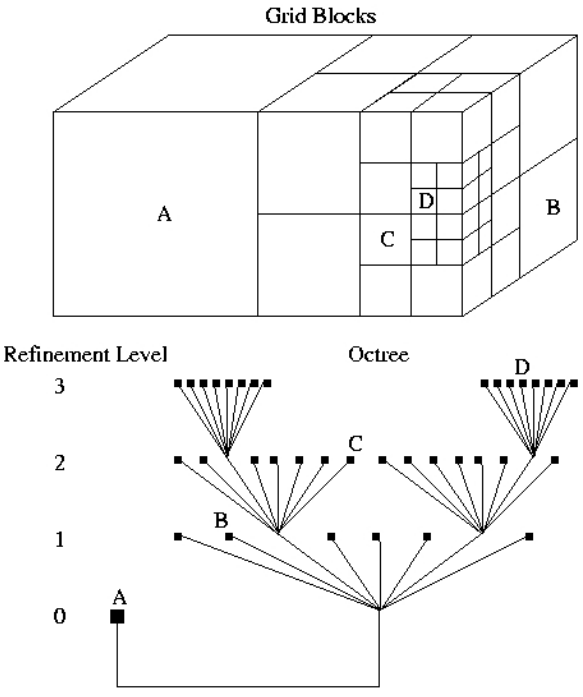
- Arbol jerárquico de voxels de distintos tamaños
  - Refleja la distribución de los objetos en la escena
- Cómo se construye:
  - Rodear la escena con un cubo (bounding box)
  - Subividir en 8 cubos iguales
  - Guardar una lista con los objetos en cada cubo
  - Si en un cubo el número de objetos es mayor que un umbral, subdividir de nuevo
- La subdivisión sólo ocurre donde hay muchos objetos

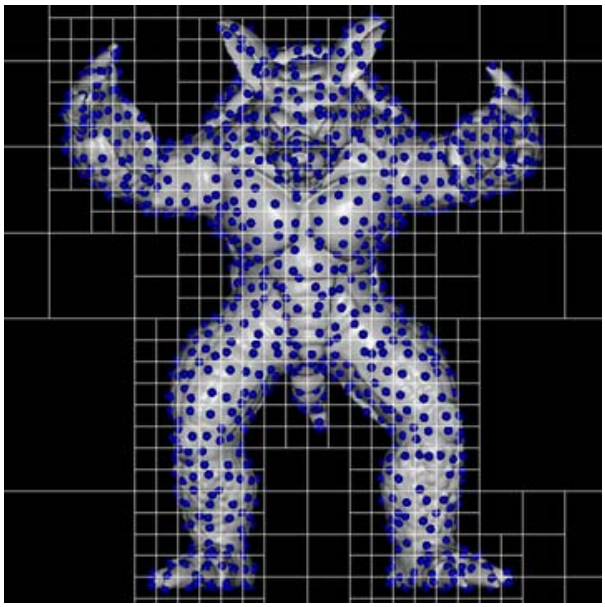


Subdivisión espacial no uniforme: *octrees*

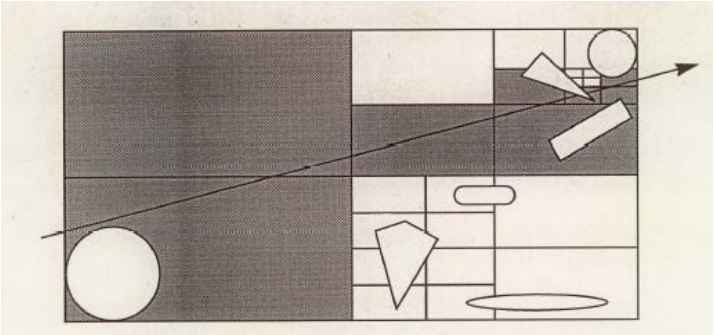


Subdivisión espacial no uniforme: *octrees*





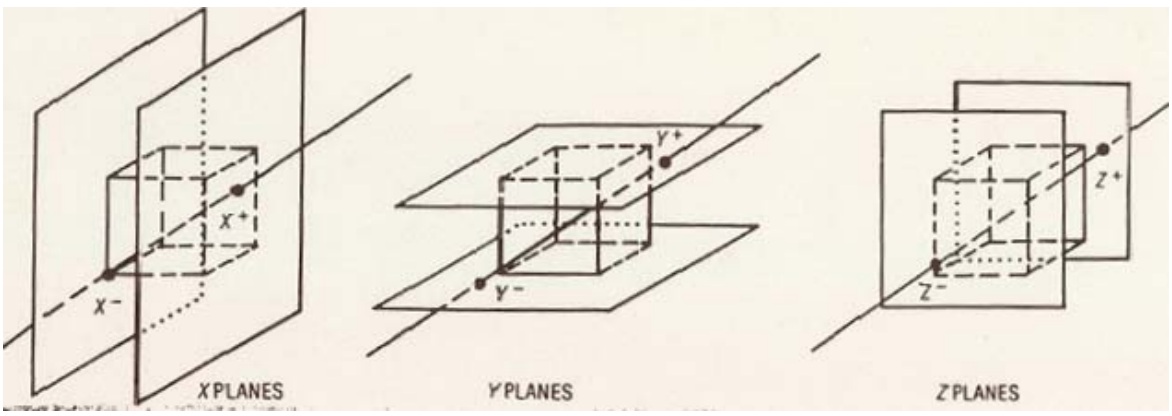
Subdivisión espacial no uniforme: *octrees*



- Sólo 6 voxels son considerados para dar con la intersección
- Sólo 3 tests de intersección son realizados
- El rayo atraviesa grandes zonas vacías de la escena muy rápidamente
- El trabajo útil sólo se realiza en zonas de alta densidad de objetos

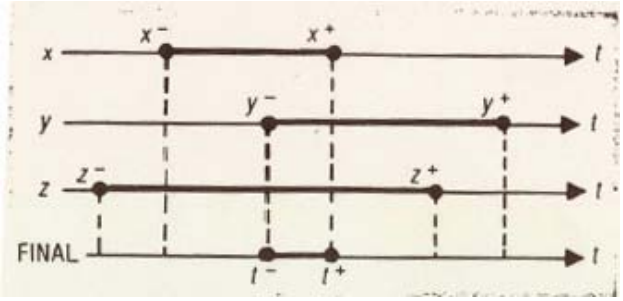
Subdivisión espacial no uniforme: *octrees*

- Una vez sabemos en qué voxel estamos… ¿cómo calculamos el segmento de rayo dentro del voxel?
  - Debemos encontrar el mayor valor de *t* que pueda adoptar el rayo en el nodo actual
  - Para ello primero intersectamos el rayo con los 6 planos delimitantes del voxel…



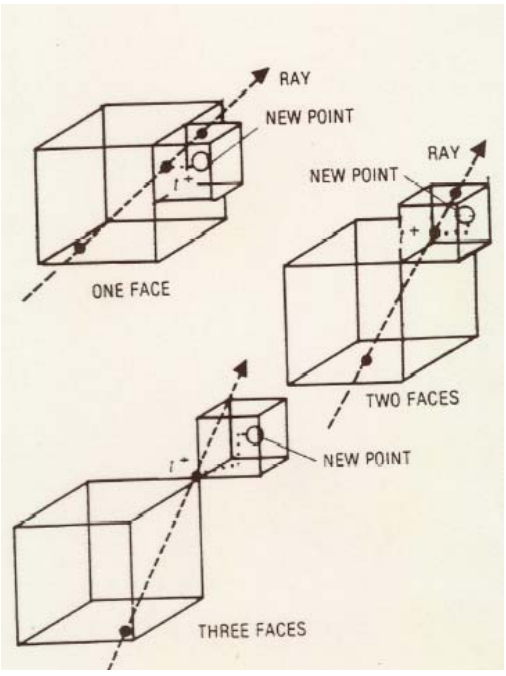
Subdivisión espacial no uniforme: *octrees*

- Una vez sabemos en qué voxel estamos… ¿cómo calculamos el segmento de rayo dentro del voxel?
  - …y luego hallamos la intersección de los tres rangos de *t* obtenidos



Subdivisión espacial no uniforme: *octrees*

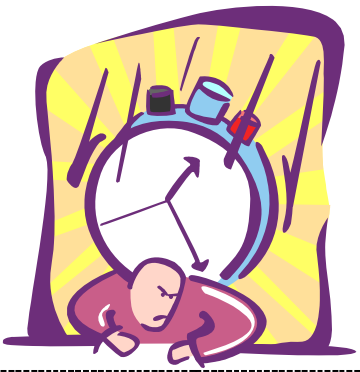
- Para hallar el siguiente voxel, debemos encontrar un punto que sepamos que va a estar en ese nuevo voxel:
  - Avanzamos una distancia *minLen/2* perpendicular a cada cara en la que caiga t^+

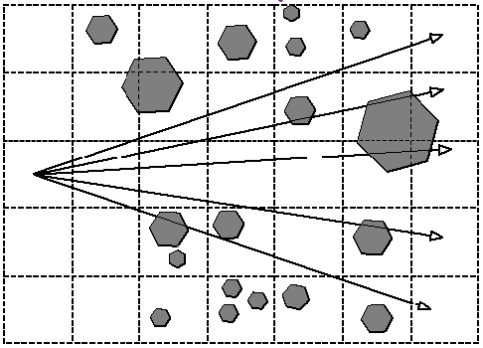


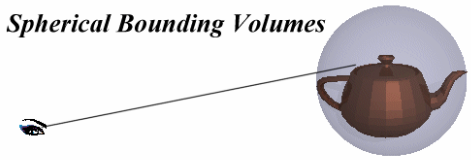
Conclusiones

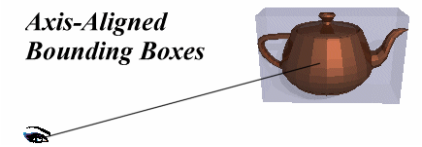
- El *trade-off* entre la resolución de cada voxel y el tiempo empleado en recorrerlos debe ser ajustado en función de la escena...
- ...ya sea subdivisión uniforme o no uniforme
- **Ejemplo:** Una subdivisión uniforme de una escena con 10.000 objetos tardó 15 minutos en renderizarse... Y 40 días sin voxelizar!

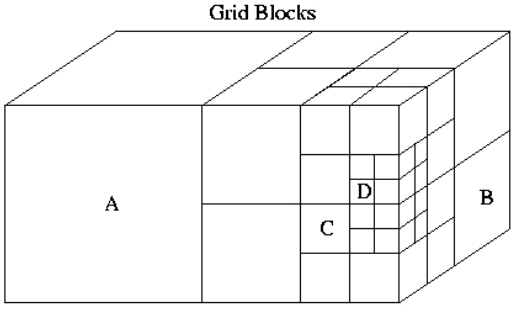
Ray Tracing Reloaded in a nutshell



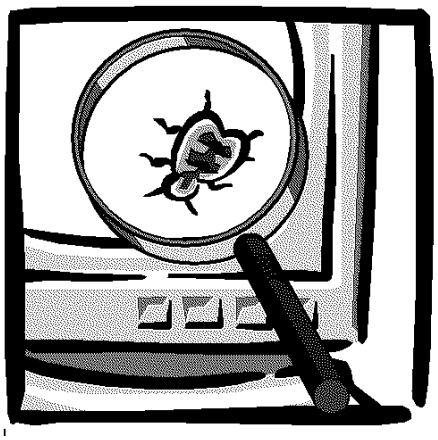








Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza



Sampling y aliasing

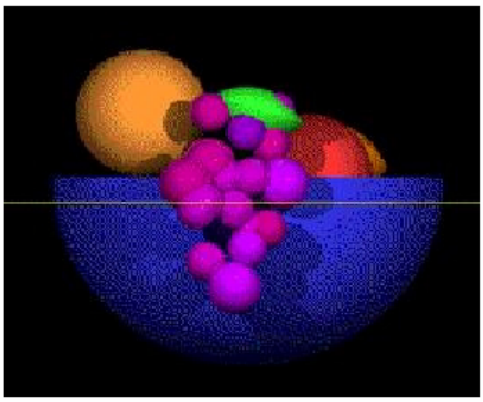
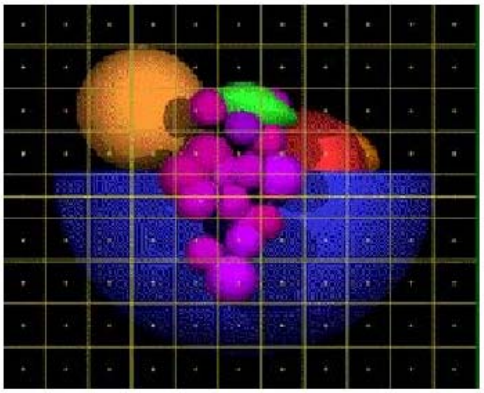
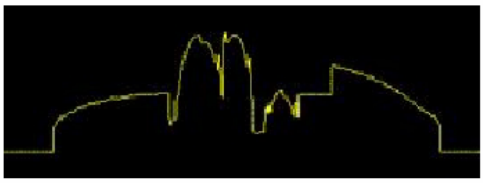
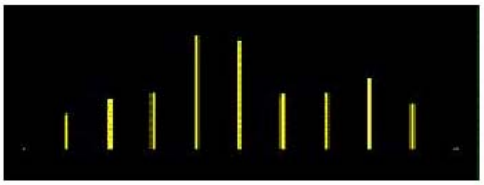
Diego Gutiérrez y Francisco J. Serón

Indice

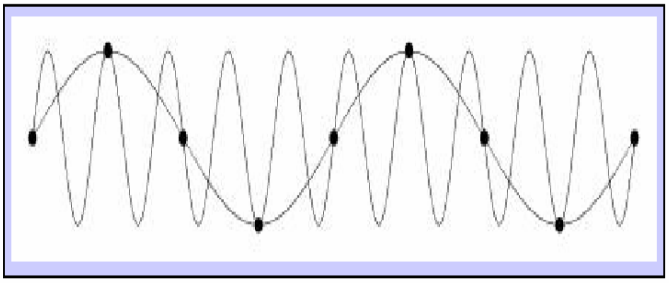
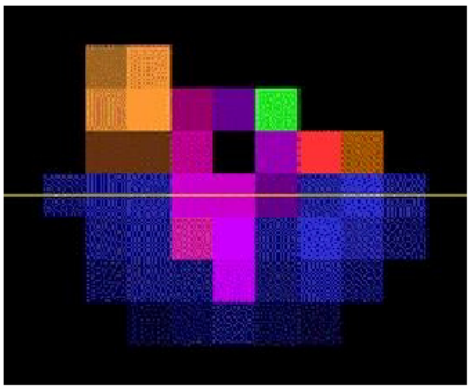
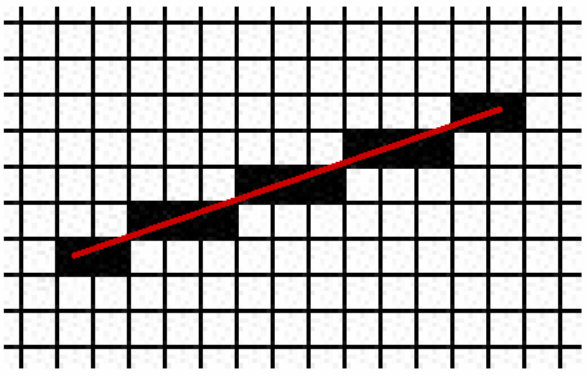
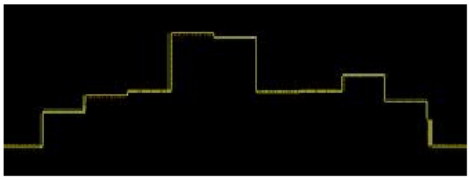
- Qué es el *aliasing*
- Mundo real vs. mundo digital
- Sampling
- Antialiasing uniforme
- Antialiasing adaptativo

Qué es el *aliasing*

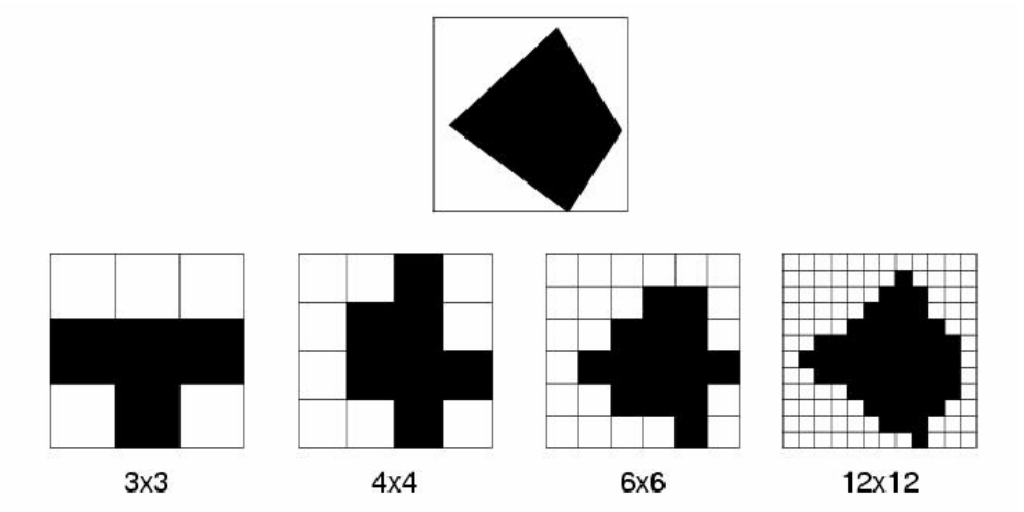
- La reconstrucción de una señal muestreada nos da una solución falsa (*alias*), diferente a la señal continua original

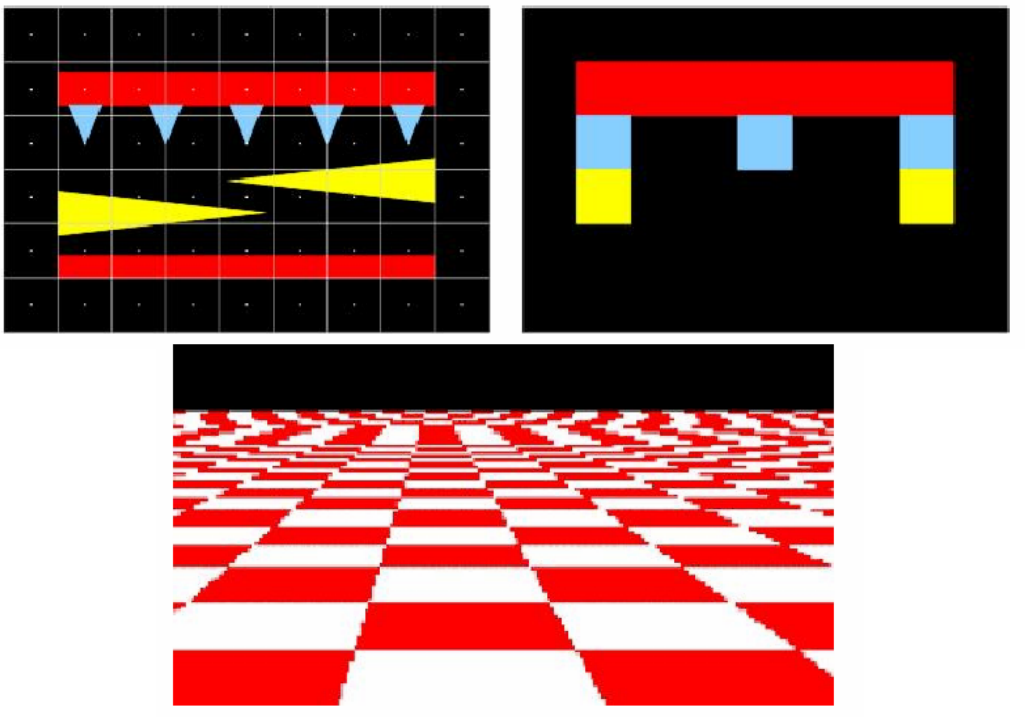
Qué es el *aliasing*

Qué es el *aliasing*



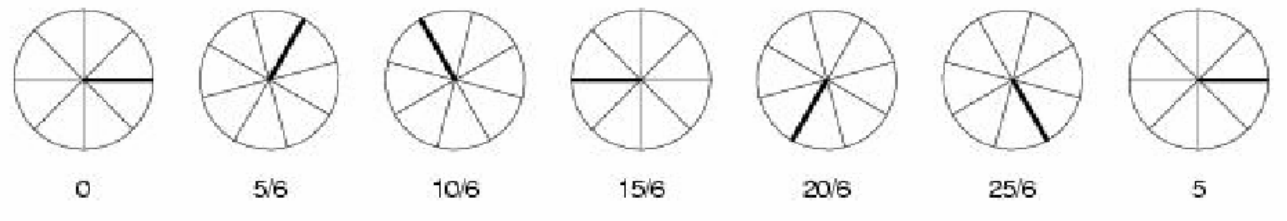
Qué es el *aliasing*



GIGA
Diego Gutiérrez y Francisco J. Serón

Qué es el *aliasing*

- Los ejemplos vistos hasta ahora eran aliasing espacial. Existe también el aliasing temporal (o el efecto de la rueda que gira hacia atrás en las películas)



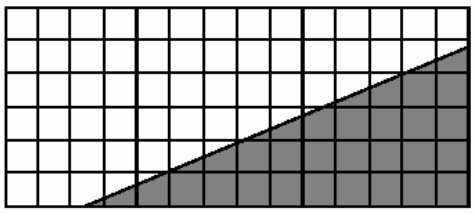
Mundo real vs. Mundo digital

- En el mundo real, la mayoría de las cosas son continuas
- En un computador, **todo es discreto**
- **Muestrear:** mapear una función continua a una discreta

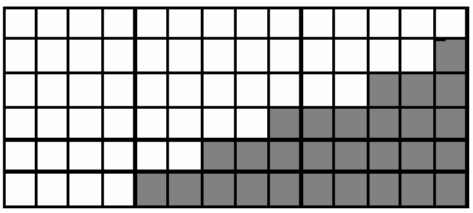
- La mayoría de las funciones que manejamos son analíticas, y pueden por tanto expresarse como una serie de relaciones algebraicas
- Las imágenes, sin embargo, no pueden ser representadas de una manera tan sencilla, en general
- En vez de eso, usamos funciones tabuladas

- Luego... ¿cómo llenamos la tabla?

Mundo real vs. Mundo digital



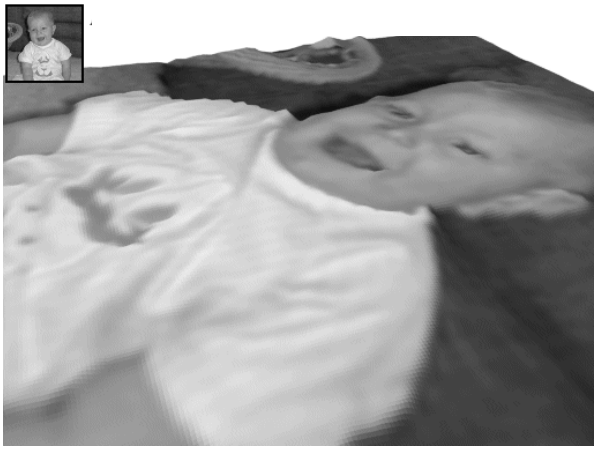
(a) reality



(b) image

Sampling

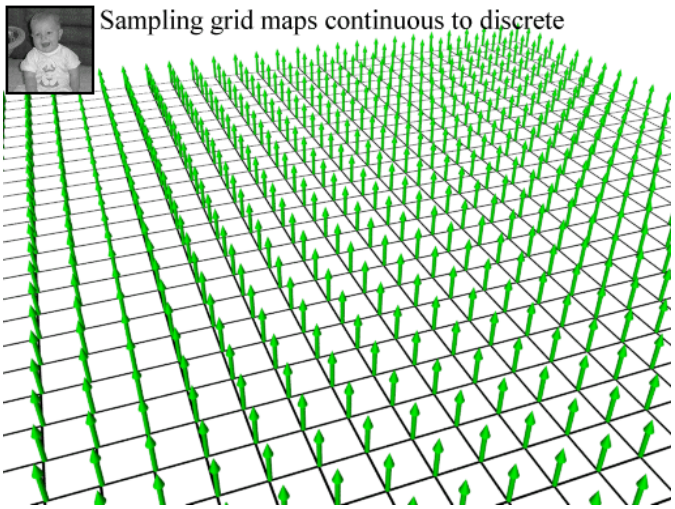
- Una imagen es, por lo general, una función continua I(x,y), que da un valor de intensidad para cada coordenada (x,y)



Sampling

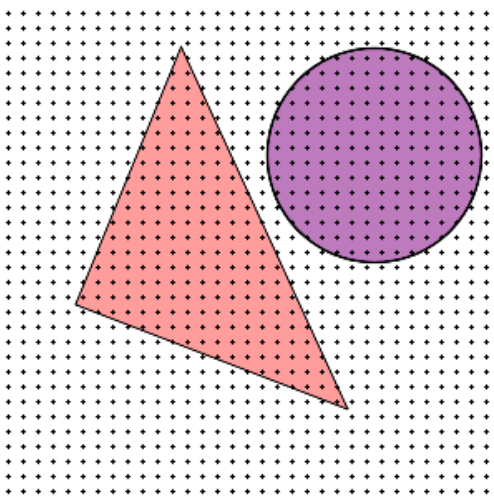
- La forma más común de muestrearla es multiplicarla por una parrilla de muestreado (*sampling grid*) compuesta por funciones delta de Kronecker distribuidas uniformemente

$$\delta(x,y) = \begin{cases} 1, & (x,y) = (0,0) \\ 0, & \text{otherwise} \end{cases}$$



Sampling

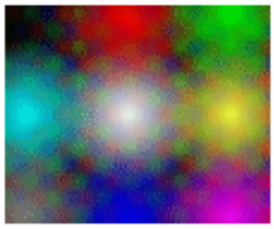
- Cuando esta multiplicación se lleva a cabo, obtenemos un conjunto de valores discretos (los pixels de la imagen)
- Estos se almacenan en memoria como representación de la función continua de la que provienen

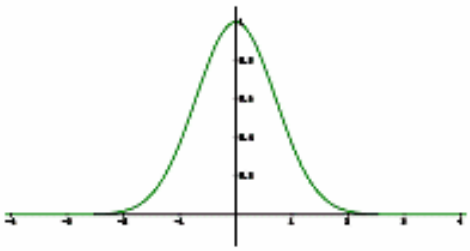


¿Qué es un pixel?

- Por lo que hemos dicho, un pixel es un punto
- No es un pequeño cuadradito, ni un círculo
- Es un punto, sin dimensiones
- Entonces... ¿cómo lo vemos?

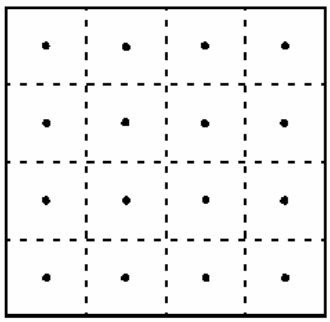
- Gracias a que los dispositivos de visualización no son perfectos, la intensidad de cada punto en realidad decae con una gaussiana... cobrando dimensión de área

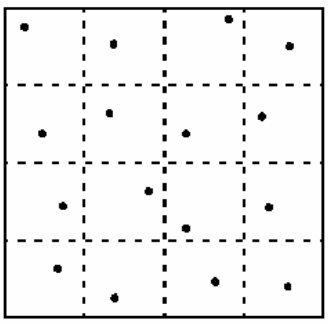




Antialiasing

- El antialiasing intenta reducir los efectos del aliasing
- La forma más habitual es el **supersampling**
  - *Uniform supersampling*
  - *Jittered supersampling*

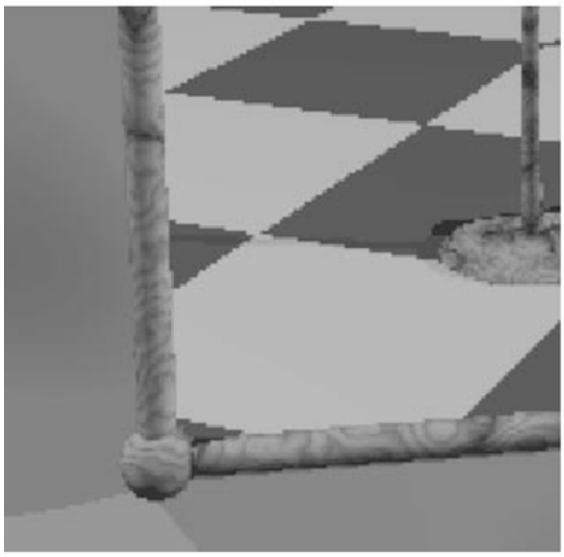


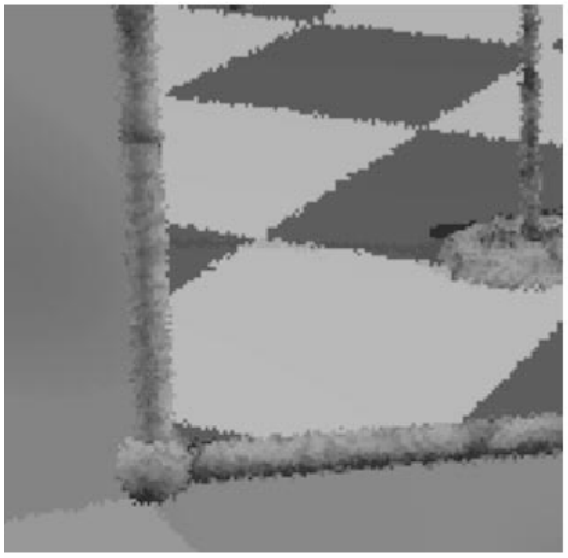


- En general, es recomendable el *jittered*

Antialiasing

- Supersampling uniforme vs *jittered*



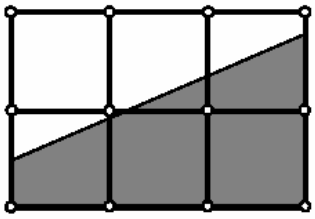


Antialiasing

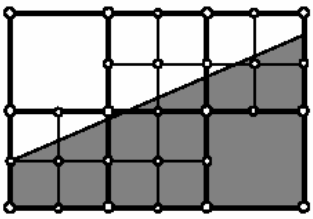
- Antialiasing adaptativo:
  - Supersampling en cada pixel: altamente ineficiente!
  - El aliasing sólo es visible en zonas de alto contraste
  - Los esquemas adaptativos localizan los pixels donde el antialiasing es más necesario
- Ejemplo de esquema de antialiasing adaptativo:
  - 1. Lanzar rayos a las esquinas de cada pixel
  - 2. Para cada pixel, examinar la diferencia entre los valores obtenidos
  - 3. Si esta diferencia es mayor que un umbral, dividir el pixel en cuatro subpixels y repetir el proceso
  - (si trabajamos en RGB, basta que uno de los valores supere el umbral para que se subdivida otra vez)

Antialiasing

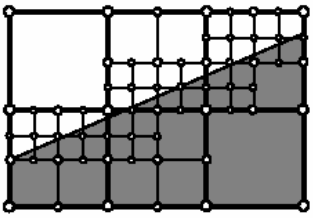
- Antialiasing adaptativo



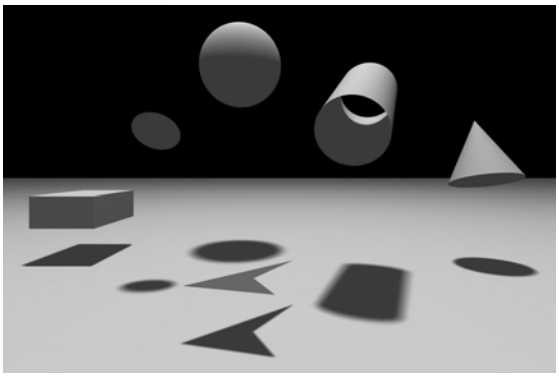
(a)



(b)

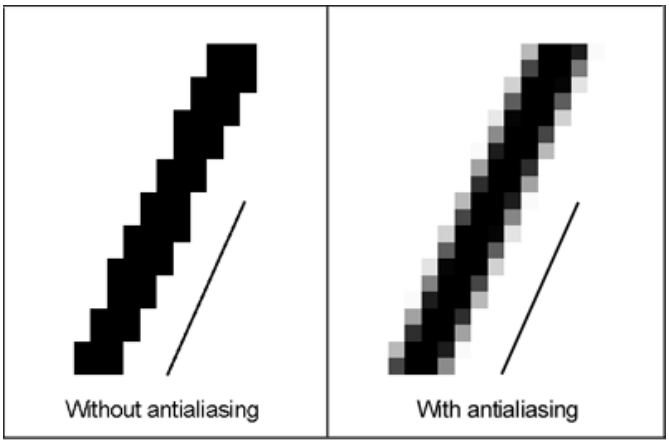


(c)

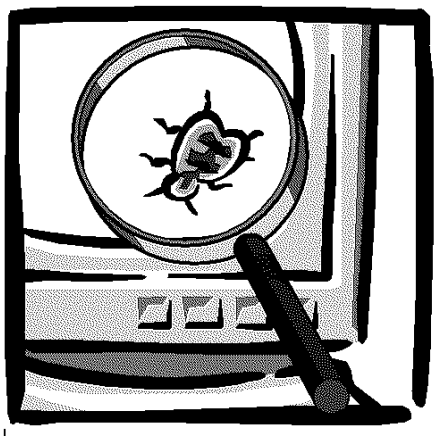


Antialiasing

- El proceso de reconstrucción de las muestras tomadas se llama **filtrado:**
  - Caja
  - Gaussiano
  - Cónico
  - …



Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

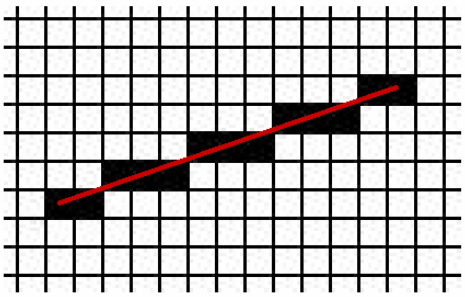


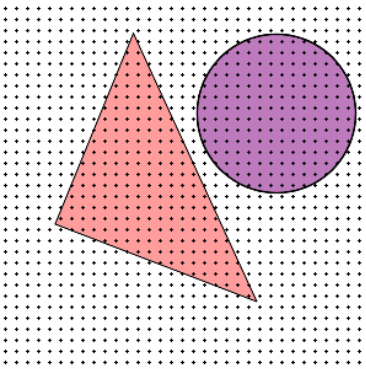
Modelado Visual y Animación

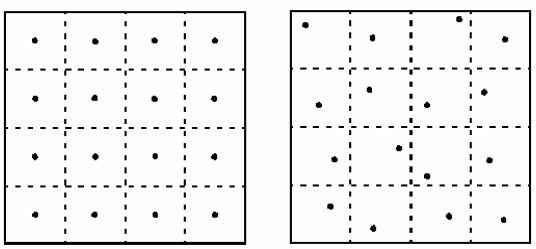
Sampling y aliasing (in a nutshell)

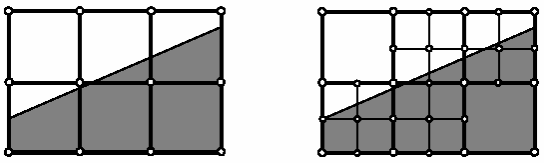
Diego Gutiérrez y Francisco J. Serón

Sampling y aliasing (in a nutshell)

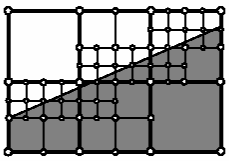








(a) (b)



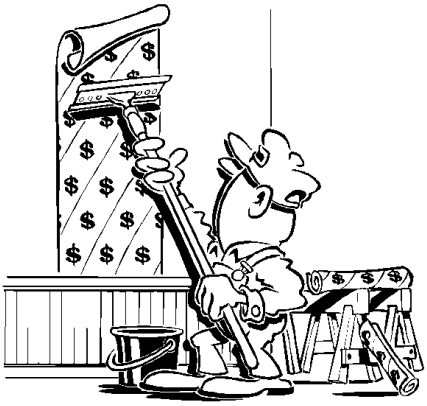
(c)

Bibliografía

J. Foley, A. vanDam, S. Feiner and J. Hughes, Computer Graphics: Principles and Practice. 2nd. ed. Addison-Wesley, 1990.

A. Watt and M. Watt, Advanced Animation and Rendering Techniques: Theory and Practice. Addison-Wesley, 1992.

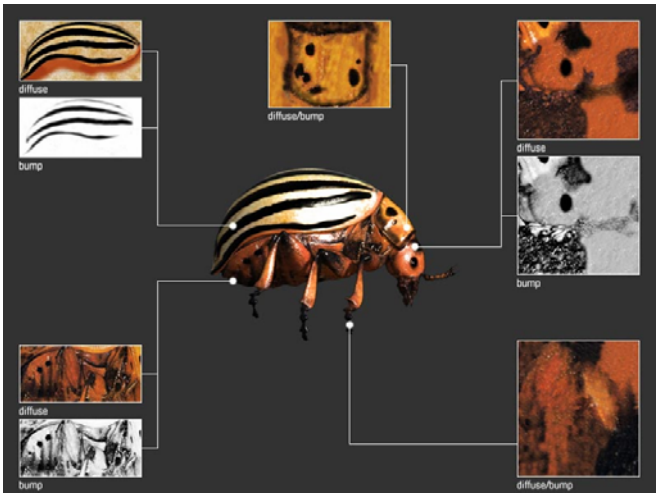
Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza



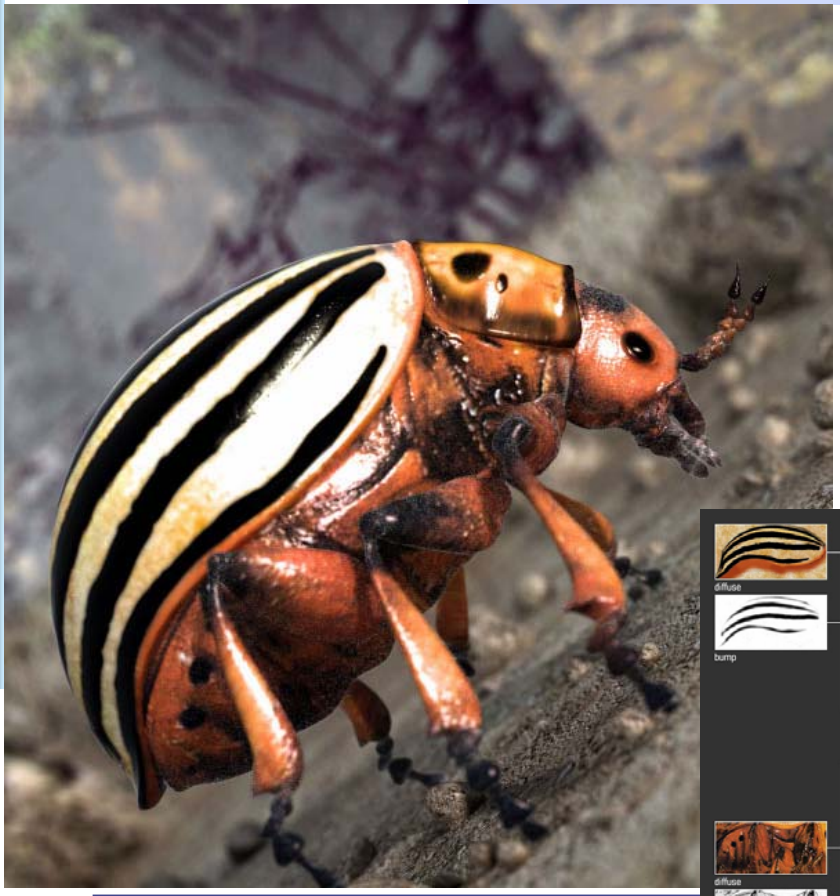
Texturas

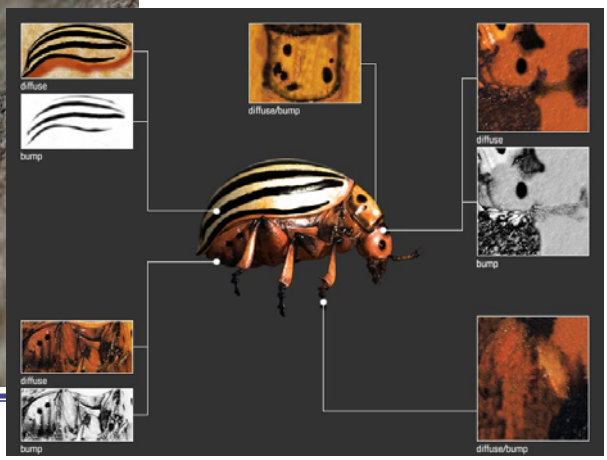
Diego Gutiérrez y Francisco J. Serón

Indice



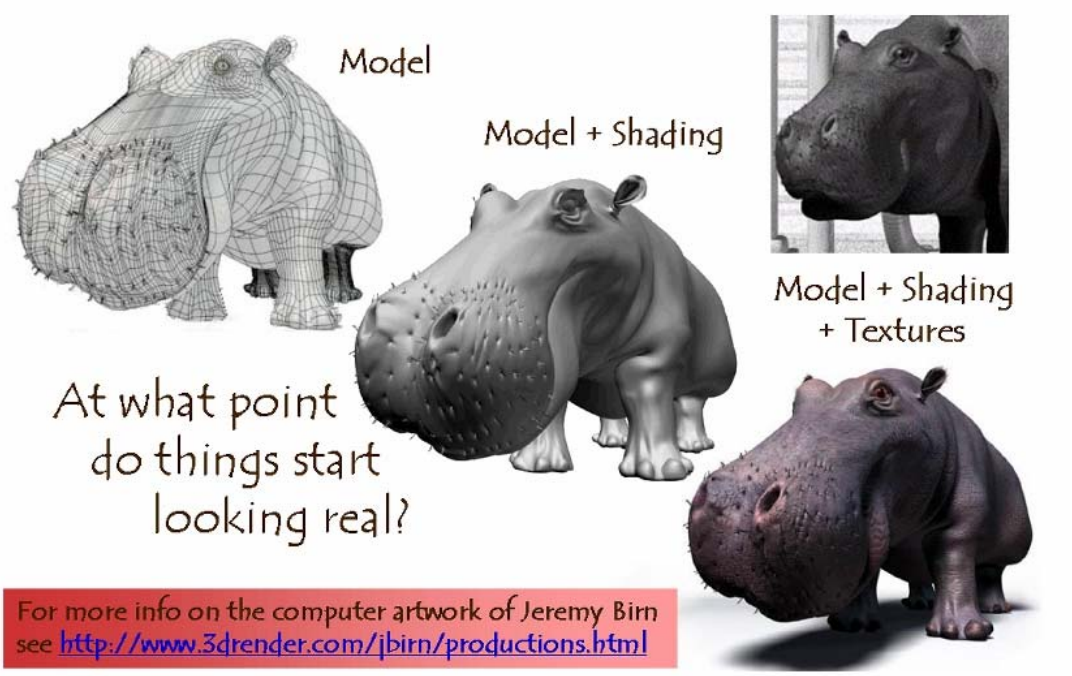
- Aumentando el realismo
- El concepto
- Tipos de texturas
- Mapeado bidimensional
  - Planar, cilíndrico, esférico
- Repetición de texturas
- Superficies paramétricas
- Texturas 3D
- *Bump* y *displacement mapping*
- Mapas de entorno
- Antialiasing y *mipmapping*





Diego Gutiérrez y Francisco J. Serón

Aumentando el realismo



Aumentando el realismo

- Motivación: dejar atrás el aspecto de material plástico del modelo de Phong

- (Los juegos se basan muchísimo en eso!)

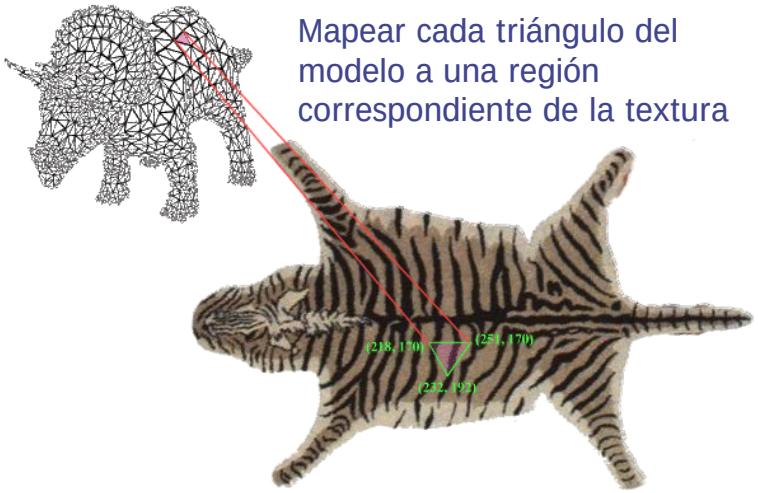
Aumentando el realismo

Aumentando el realismo

Aumentando el realismo

El concepto

- El concepto es muy sencillo:



Mapear cada triángulo del modelo a una región correspondiente de la textura

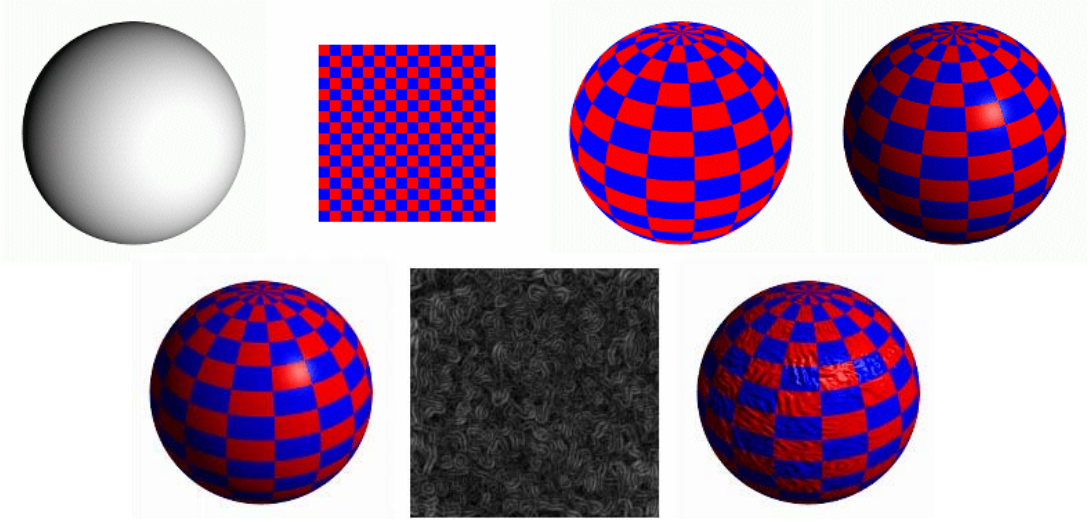
En la fase de rasterización, interpolar los colores de la textura

El concepto

- Las posibilidades son ilimitadas:

$$I_{total} = k_a I_{ambient} + \sum_{i=1}^{lights} I_i \left(k_d \left(\hat{N} \cdot \hat{L} \right) + k_s \left(\hat{V} \cdot \hat{R} \right)^{n_{shiney}} \right)$$

Phong's Illumination Model

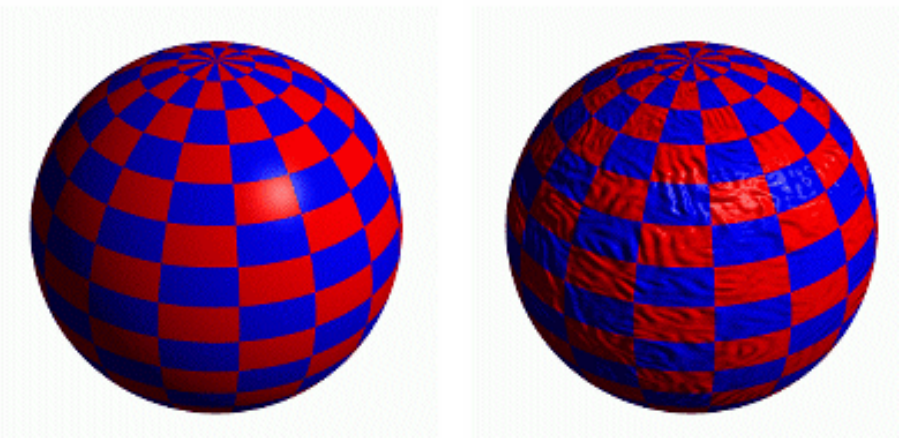


Tipos de texturas

- Vamos a considerar dos tipos de texturas de superficie:
  - Patrones, o texturas de color: “pegamos” el patrón sobre una superficie suave, pero ésta sigue pareciendo suave. Este el proceso comúnmente conocido como **mapeado de texturas** (*texture mapping*)

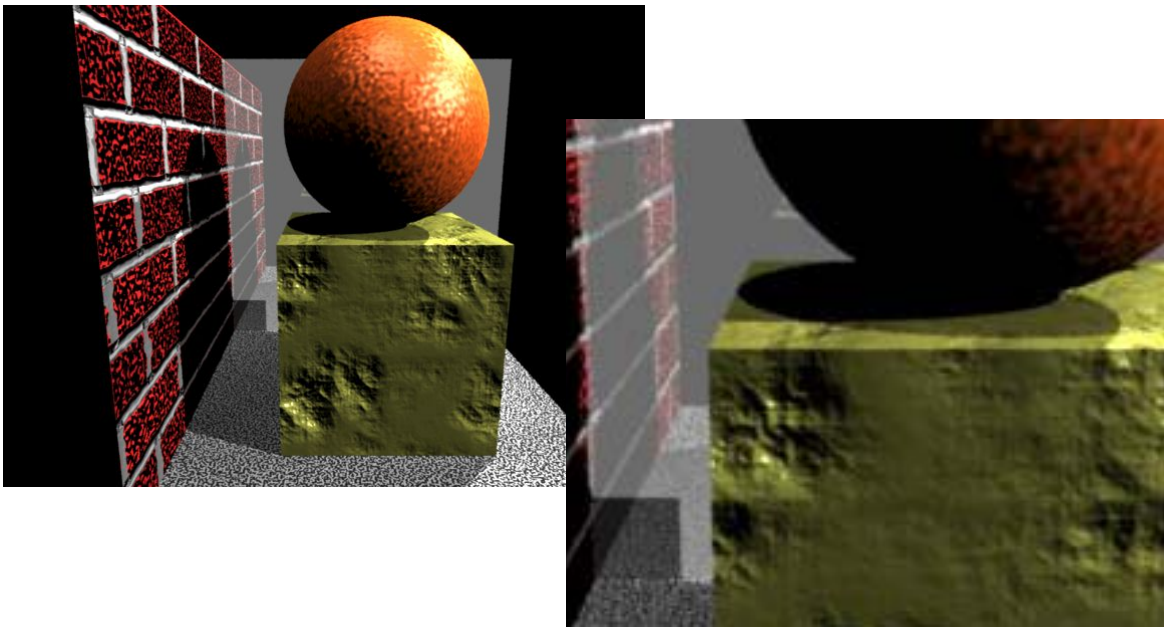
Tipos de texturas

- Vamos a considerar dos tipos de texturas de superficie:
  - Rugosidad: utilizando una función de perturbación que *aparentemente* modifique la geometría de la superficie (*bump mapping;* no confundir con los mapas de desplazamiento o *displacement maps*)



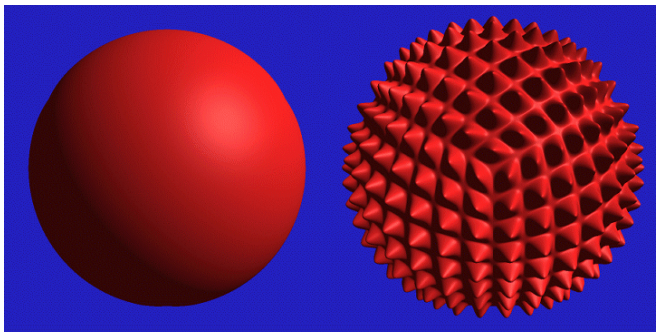
Tipos de texturas

- Y hemos dicho **aparentemente!**



Tipos de texturas

- Los mapas de desplazamiento modifican realmente la geometría



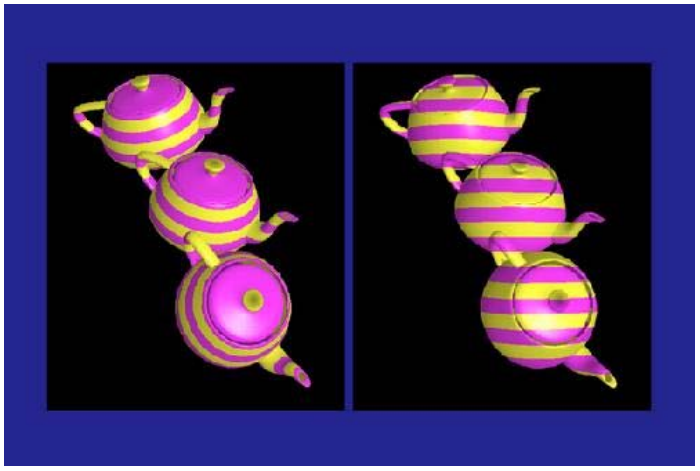
- Objetos caros como demonios!
- La geometría debe ser desplazada antes de determinar la visibilidad
- No es fácil de implementar en un trazador de rayos (*Geometry Caching for Ray-Tracing Displacement Maps,* Matt Pharr and Pat Hanrahan)

- Hay otros tipos: transparencia, *IK-weights, property guiders* (el pelo de Might Joe Young...), mapas de especularidad...

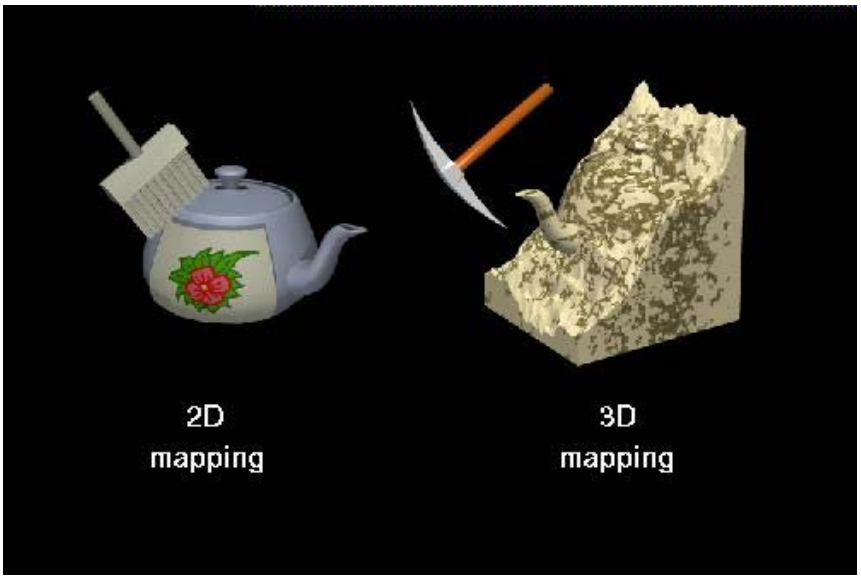
Tipos de texturas

Coordenadas del mundo o del objeto

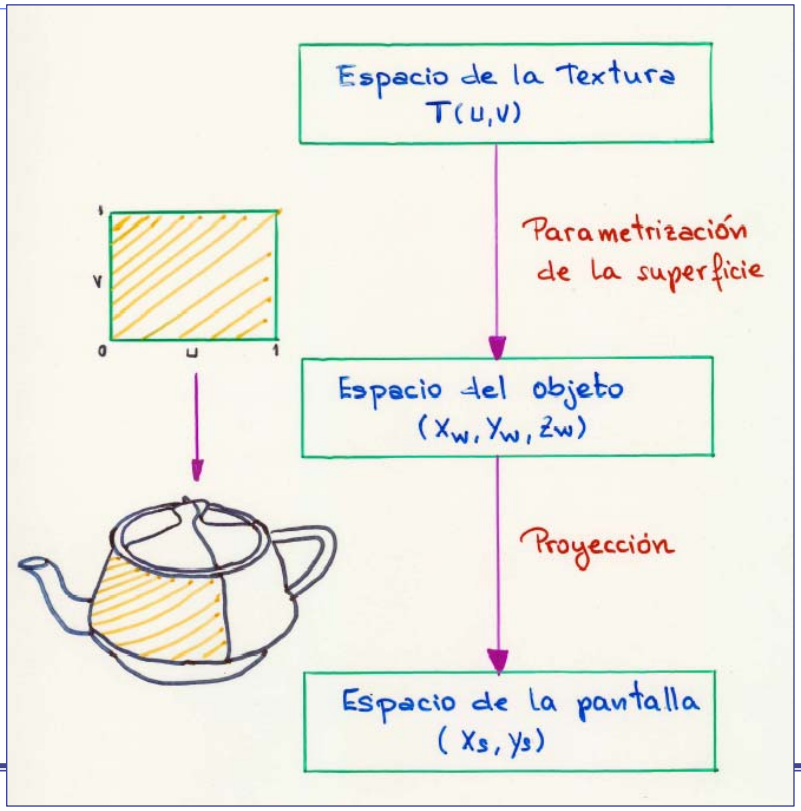
- Casi todos los algoritmos de mapeado usan coordenadas del objeto
- Objeto: el patrón es solidario al objeto mientras éste se mueve
- Mundo: el patrón se desliza sobre el objeto si éste se mueve



2D o 3D

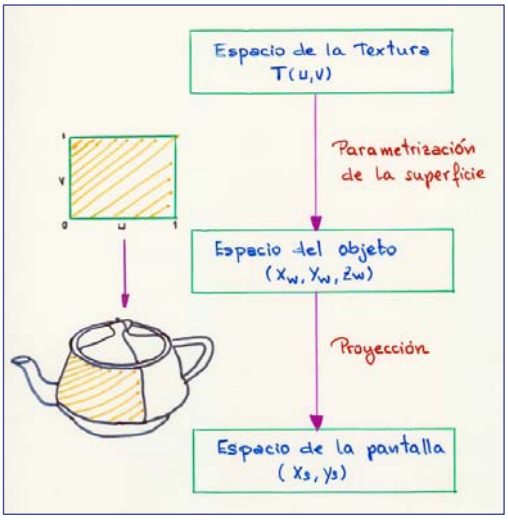


Mapeado bidimensional



Mapeado bidimensional

- Si los objetos están modelados mediante superficies paramétricas, el proceso es posible
- Si son mallas poligonales, el proceso no está definido

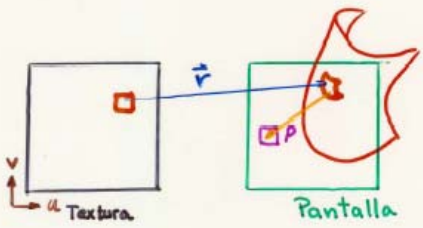


Mapeado bidimensional

. Para dibujar en pantalla

$S \equiv \vec{r}(u,v) = x(u,v)\,\vec{i} + y(u,v)\,\vec{j} + z(u,v)\,\vec{k}$

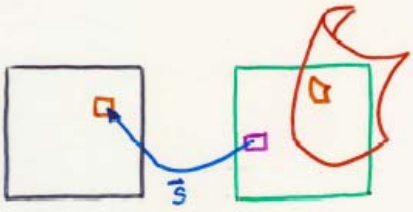
$P \equiv$ proyección del punto (x,y,z) en la pantalla.



Mapeado bidimensional

. Para utilizar una textura

Dado un punto de pantalla obtener las coordenadas en el espacio paramétrico

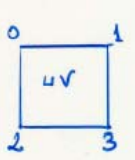


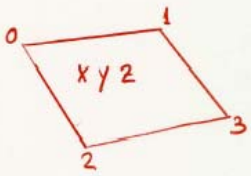
S = Transformación no Lineal compuesta por transformaciones inversas a las anteriores.

Mapeado bidimensional

- Proceso de dos pasos:

Consideremos el problema de una textura cuadrada que se aplica en un cuadrilátero plano situado en el espacio.

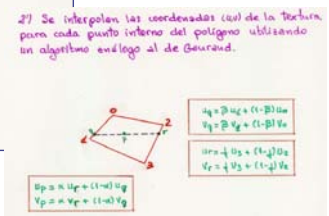




1º) Se establece la correspondencia inicial

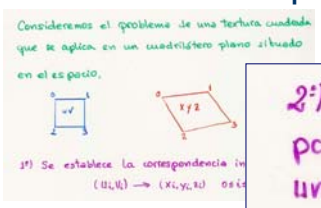
$(U_i, V_i) \longrightarrow (X_i, Y_i, Z_i) \quad 0 \le i \le 3$

2º) Se interpolan las coordenadas (u,v) de la textura para cada punto interno del polígono utilizando un algoritmo análogo al de Gouraud.

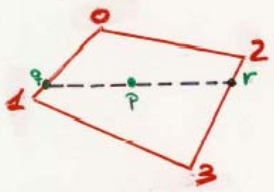


Mapeado bidimensional

- Proceso de dos pasos:



2º) Se interpolan las coordenadas (u,v) de la textura para cada punto interno del polígono utilizando un algoritmo enálogo al de Gouraud.



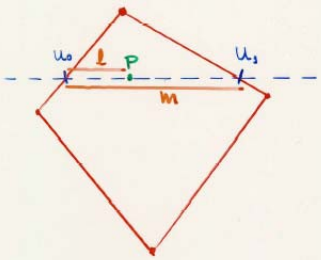
$$u_q = \beta u_1 + (1-\beta) u_0$$
$$v_q = \beta v_1 + (1-\beta) v_0$$

$$u_r = \tfrac{1}{4} u_3 + (1-\tfrac{1}{4}) u_2$$
$$v_r = \tfrac{1}{4} v_3 + (1-\tfrac{1}{4}) v_2$$

$$u_p = \alpha u_r + (1-\alpha) u_q$$
$$v_p = \alpha v_r + (1-\alpha) v_q$$

Mapeado bidimensional

- Interpolación lineal:



Esquema de interpolación: $u_p = \alpha u_1 + (1-\alpha) u_0$

$\alpha = \frac{l}{m}$ $\quad 0 \le \alpha \le 1$

Interpolación lineal: $u_p = a_1 p + a_0 \quad 0 \le p \le 1$

esta interpolación debe verificar

$u_p = u_0$ cuando $p = 0$

$u_p = u_1$ cuando $p = 1$

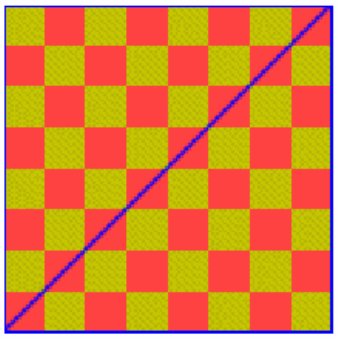
por lo tanto

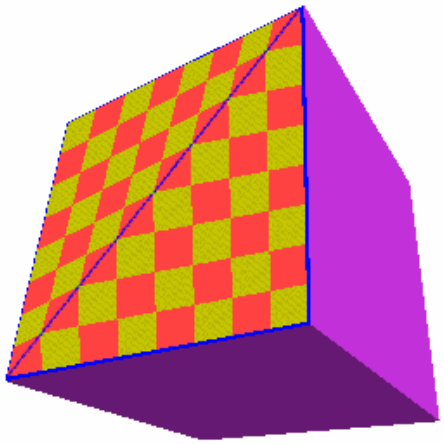
$$\left.\begin{array}{l} u_0 = a_0 \\ u_1 = a_1 + a_0 \end{array}\right\} \begin{array}{l} a_0 = u_0 \\ a_1 = u_1 - u_0 \end{array} \Rightarrow u_p = (u_1 - u_0) p + u_0$$

¡Ambas expresiones coinciden!

Efecto de la proyección

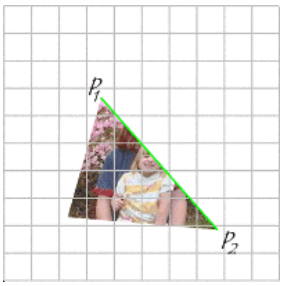
- Hummm..... Algo no funciona aquí...

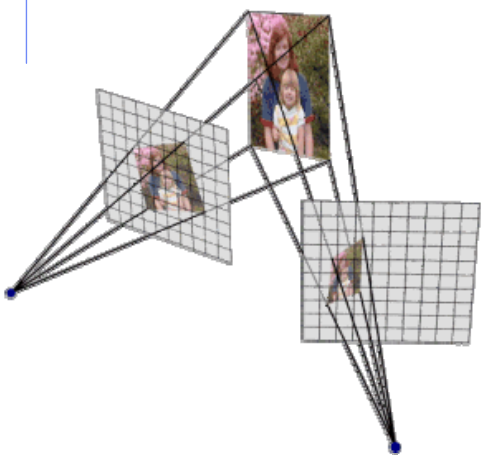


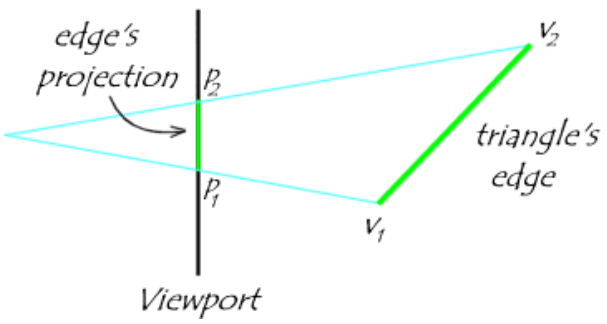


Efecto de la proyección

- Consideremos una arista de un triángulo; esa arista y su proyección pertenecen al mismo plano:

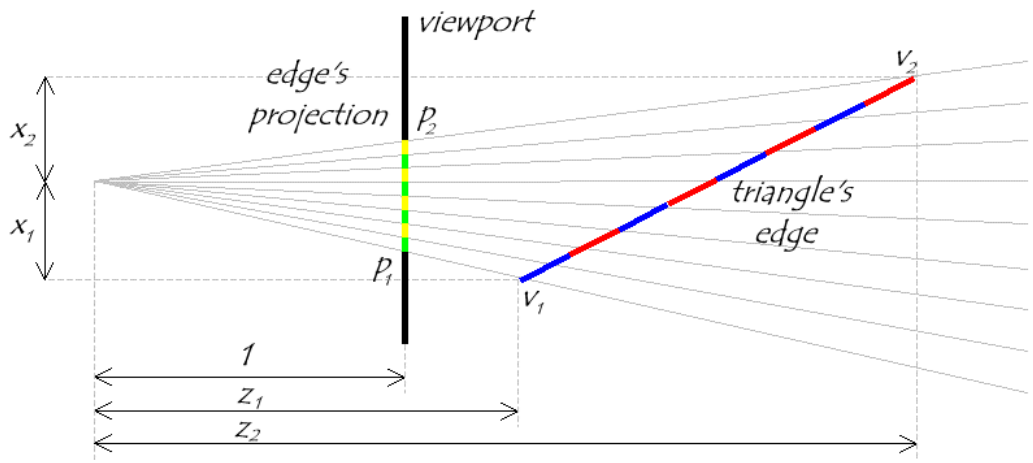




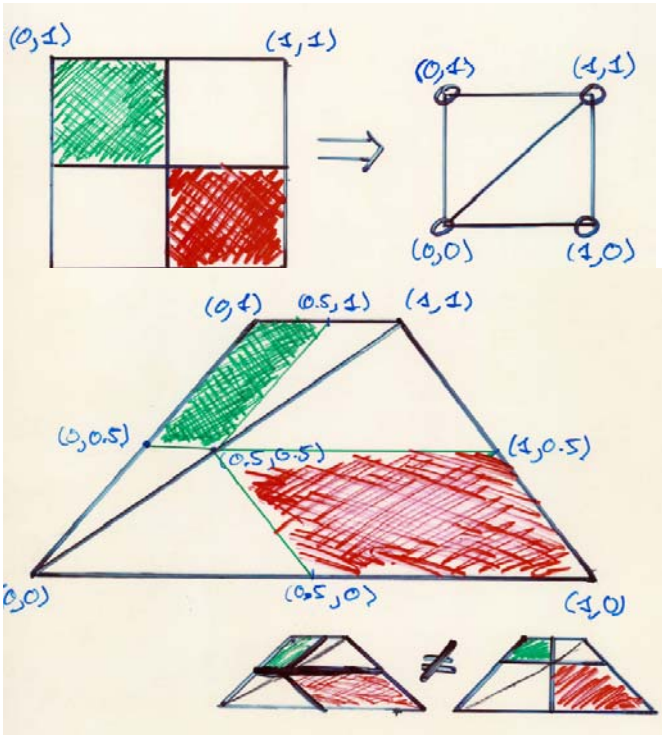


Efecto de la proyección

- Se ve cómo intervalos iguales en la proyección no corresponden con intervalos iguales en la arista real

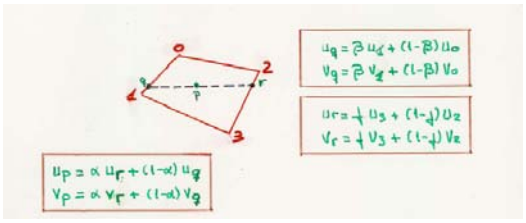


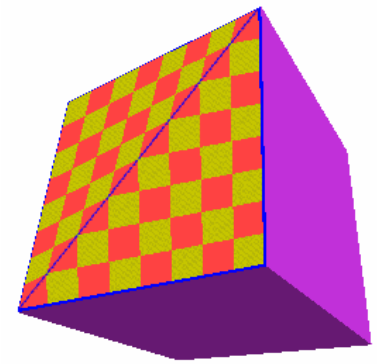
Efecto de la proyección

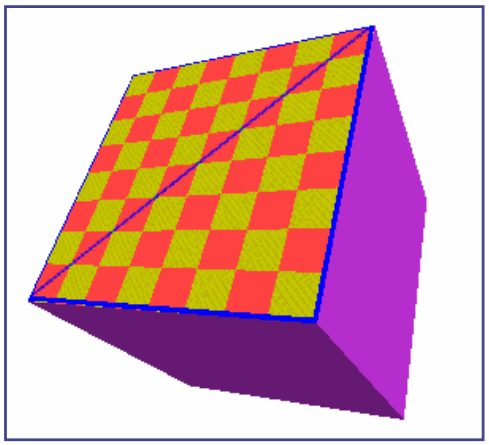


Efecto de la proyección

- Solución: coordenadas homogéneas
  - 1. Interpolación lineal para u
  - 2. Interpolación lineal para v
  - 3. Interpolación lineal para w
  - $u_f=u/w$ $v_f=v/w$







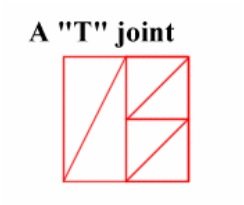
UN MOMENTO!!!!

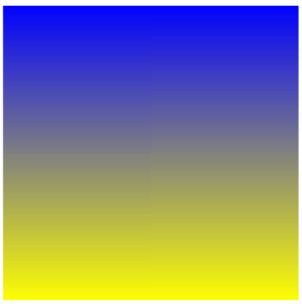
- Habíamos dicho que “interpolamos utilizando un algoritmo análogo al de Gouraud…
- … Y hemos visto que ofrecía resultados erróneos
- ¿Significa eso que Gouraud está mal?

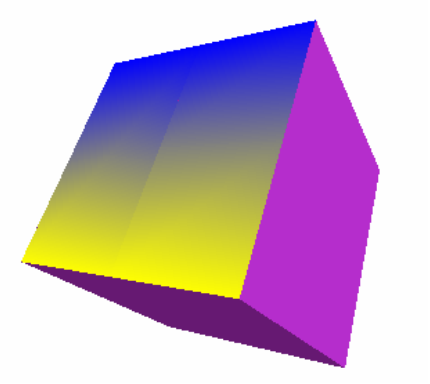
SI

UN MOMENTO!!!!

- El sombreado de Gouraud es erróneo. Lo que pasa es que la transición de colores es suave, y como además no sabemos el color que debería salir (sólo nos importa que salga algo bonito)...
- A veces el error se hace aparente:
  - Cuando cambiamos de nivel de detalle
  - En junturas-T

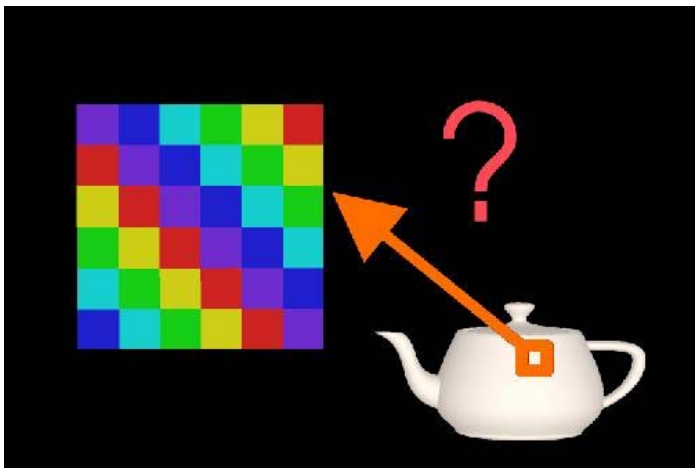






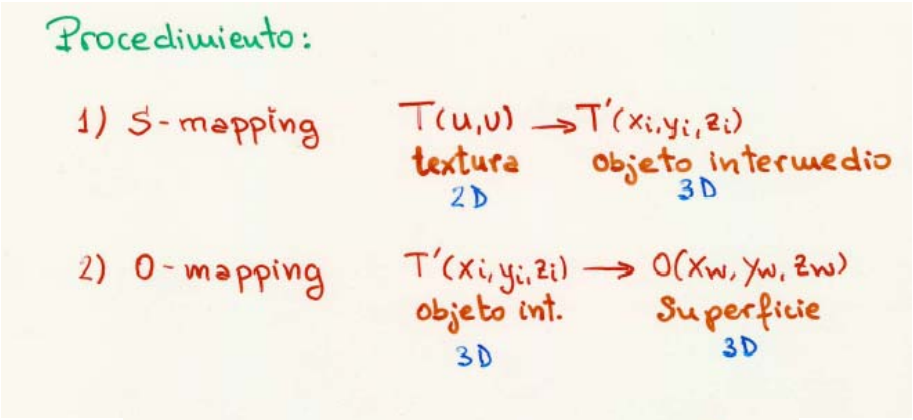
Mapeado bidimensional

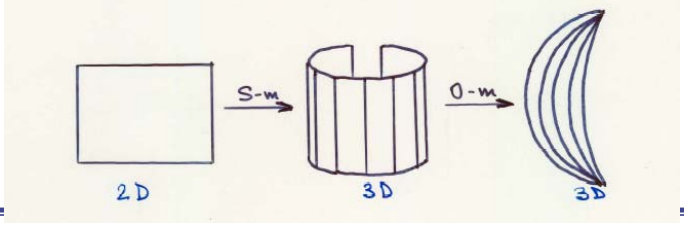
- ¿Cómo pegamos la imagen en el objeto?
- Para cada pixel, debemos responder a la pregunta: ¿a qué parte de la textura miro para coger el color?



Mapeado bidimensional

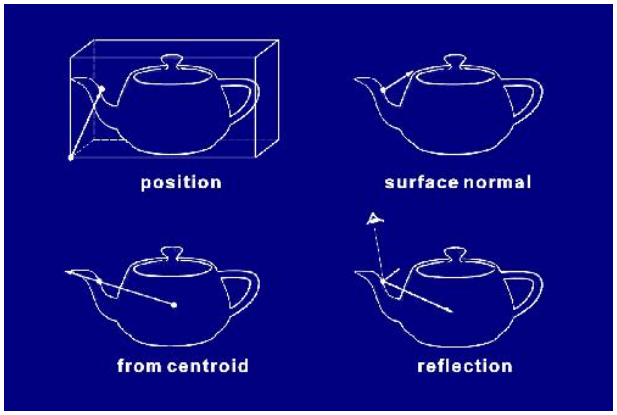
- La textura se suele aplicar sobre una superficie intermedia antes de aplicarse sobre el objeto (*two-part mapping*)





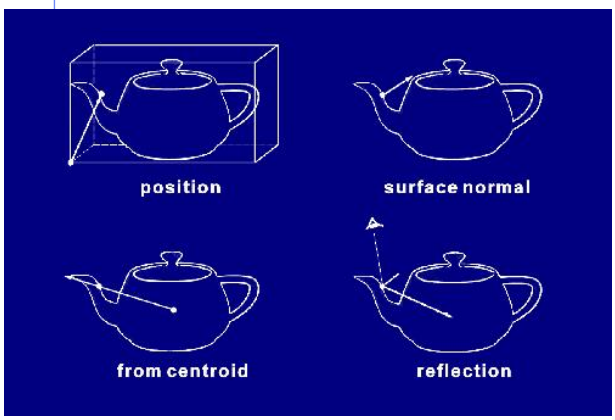
Mapeado bidimensional

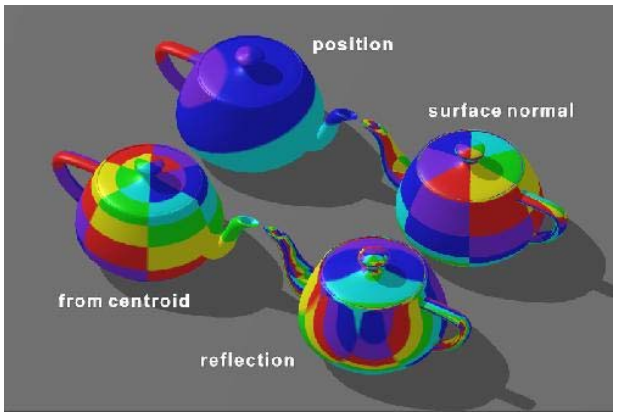
- Para el mapeado S:
  - Plano
  - Cilindro
  - Esfera
  - Cubo
- Para el mapeado O:



Mapeado bidimensional

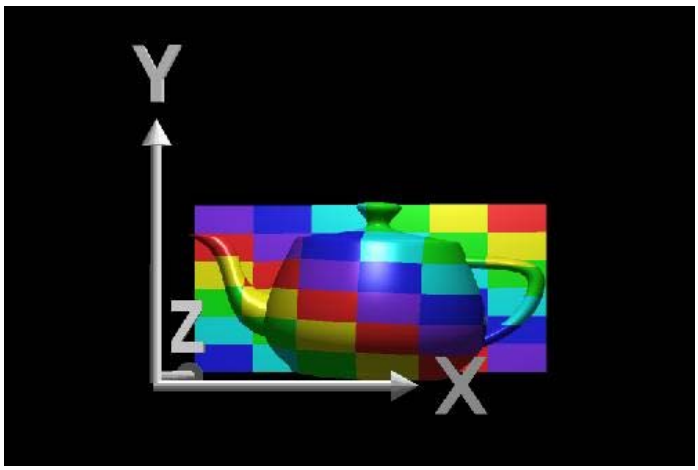
- Lo más usual es tomarlo relativo al centro de masas del objeto, pero hay más posibilidades
- Por ejemplo, para el caso del mapeado cilíndrico, las diferentes opciones nos darían los siguientes resultados:





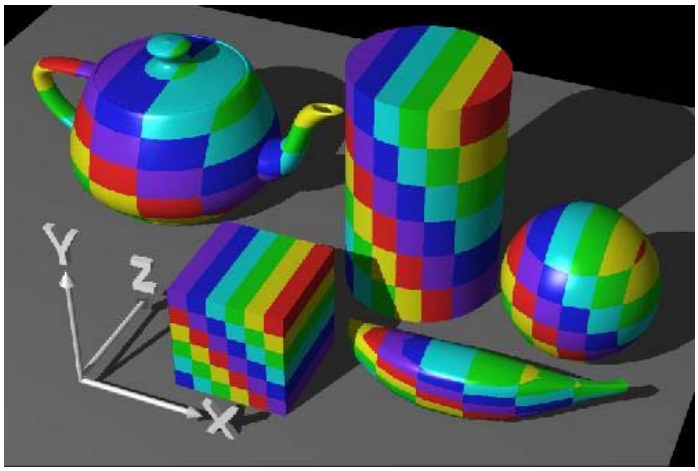
Mapeado planar

- Cogemos un valor (x,y,z) del objeto y descartamos una componente) lo cual nos deja coordenadas 2D (coordenadas planares). Usamos esas coordenadas para buscar el color en la textura



Mapeado planar

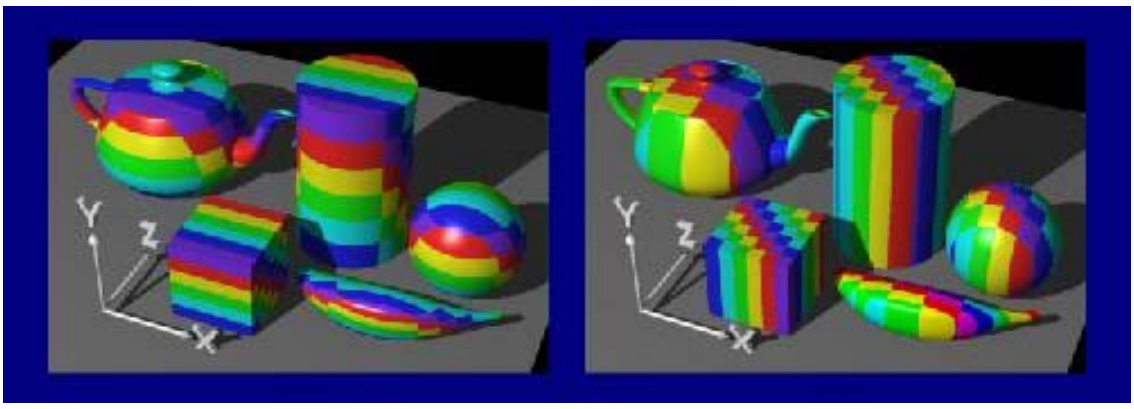
- Presenta serios problemas...



- El color en el objeto no varía a lo largo de la coordenada descartada!

Mapeado planar

- ¿Qué componentes se habían descartado en estos dos dibujos?



Mapeado cilíndrico

- (o etiqueta sobre bote de tomate)
- Convertimos un valor (x,y,z) en coordenadas cilíndricas (r, theta, altura). En mapeado de texturas, sólo nos interesan theta y la altura
- Para hallar el color, theta se convierte a un valor de x, y la altura a un valor de y

Mapeado cilíndrico

- parametric representation of a cylinder (θ, z)

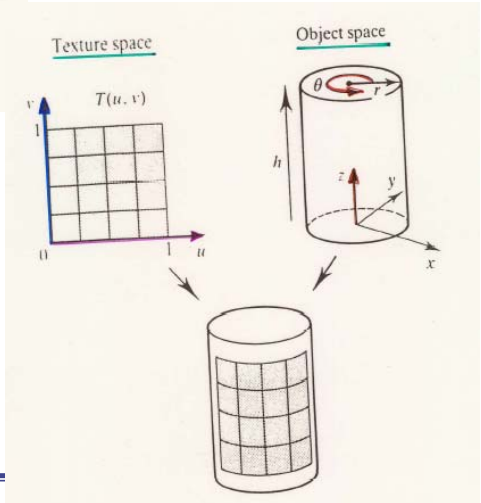
  $(r\cos\theta, r\sin\theta, hz)$ $\quad 0 \le \theta \le 2\pi \quad 0 \le z \le 1$

- cylindrical mapping $\quad (0,0)$ position of texture

  $(u,v) = \left(\frac{\theta}{2\pi}, z\right) \quad u,v \in [0,1]$

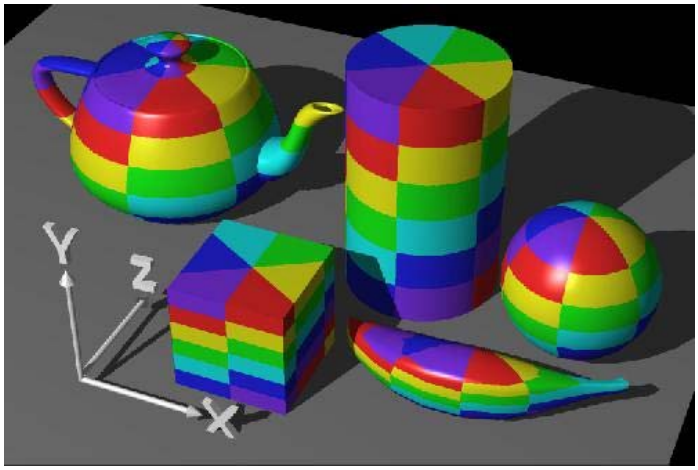
- cylindrical mapping $\quad (\theta_0, z_0)$ position of texture

  $(u,v) = \left[\frac{(\theta-\theta_0)}{2\pi}, z-z_0\right] \quad u,v \in [0,1]$



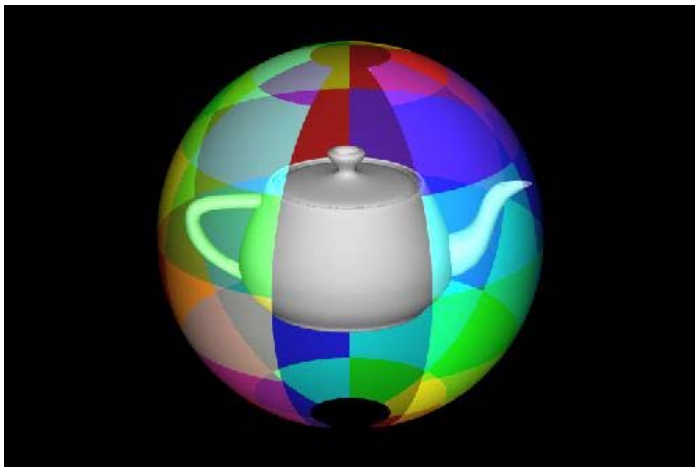
Mapeado cilíndrico

- Efectos de “trozos de tarta” en las coordenadas t_{max} y t_{min} (siendo t[x,y,z] el eje del cilindro usado para mapear)



Mapeado esférico

- Convertimos un valor (x,y,z) en coordenadas esféricas (r, theta, phi). En mapeado de texturas, sólo nos interesan theta (latitud) y phi (longitud)
- Para hallar el color, theta se convierte a un valor de x, y phi a un valor de y



Mapeado esférico

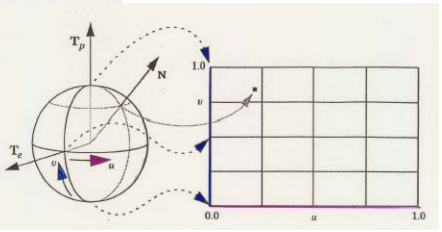
- The local coordinate system (origin at sphere's centre) is defined by two unit vectors,
  - $\vec{T}_p$ which point from the origin to the north pole
  - $\vec{T}_e$ " " " " " " " equator

- Spherical mapping

$$v = \frac{\phi}{\pi} \quad \text{where } \phi \text{ is a latitudinal parameter}$$

$$\phi = \cos^{-1}(-\vec{T}_p \cdot \vec{N})$$

$$\begin{cases} u = \dfrac{\cos^{-1}[(\vec{T}_e \cdot \vec{N})/\sin\phi]}{2\pi} & \text{if } (\vec{T}_p \wedge \vec{T}_e)\cdot\vec{N} \geq 0 \\ u = u - 1 & \text{if } (\vec{T}_p \wedge \vec{T}_e)\cdot\vec{N} < 0 \end{cases}$$



Mapeado esférico

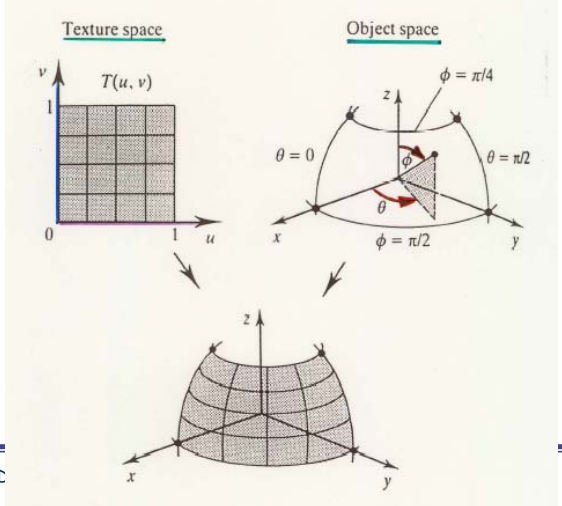
- parametric representation of a part of a sphere (θ, ϕ)

$$(r\cos\theta\sin\phi,\ r\sin\theta\sin\phi,\ r\cos\phi) \quad 0 \leq \theta \leq \pi/2$$

$$\frac{\pi}{4} \leq \phi \leq \pi/2$$

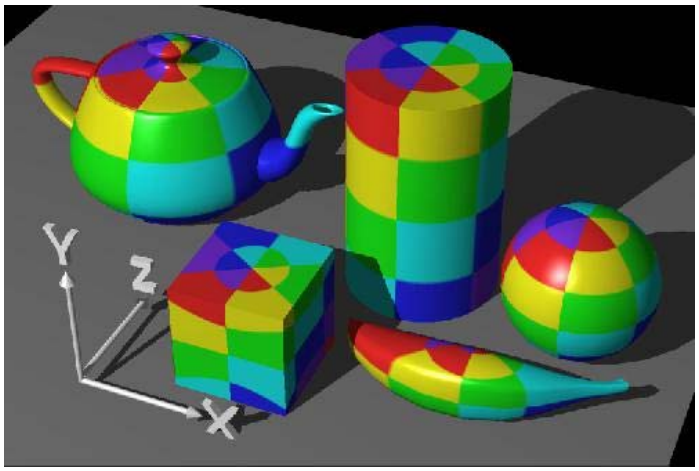
- spherical mapping

$$(u,v) = \left[\frac{\theta}{\pi/2},\ \frac{(\pi/2) - \phi}{\pi/4}\right] \qquad u,v \in [0,1]$$



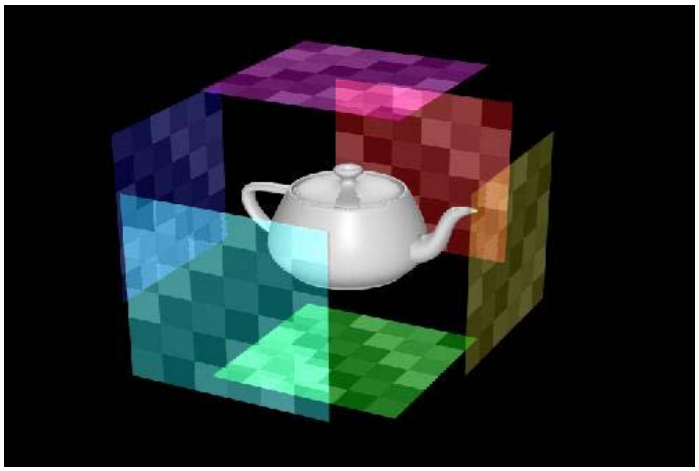
Mapeado esférico

- Los polos de la esfera de mapeado están sobre el eje Y. En los “polos” del objeto, aparecen de nuevo los efectos de “trozo de tarta”, aunque la singularidad aquí es más severa. El mapeado esférico estira la textura en el ecuador y la encoge al acercarnos a los polos.



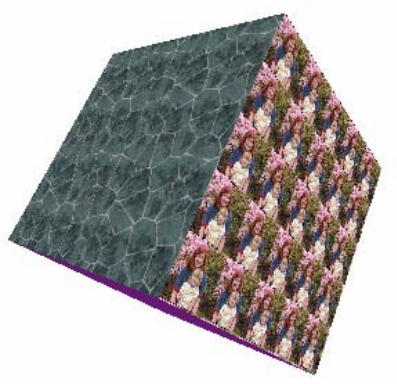
Mapeado en caja

- Similar a seis mapeados planares



Repetición de texturas

- a.k.a. *tileado de texturas*
- La textura ha de ser tileable (photoshop)
- Aún así, ojo con los patrones: el ojo humano es increíblemente bueno detectándolos!

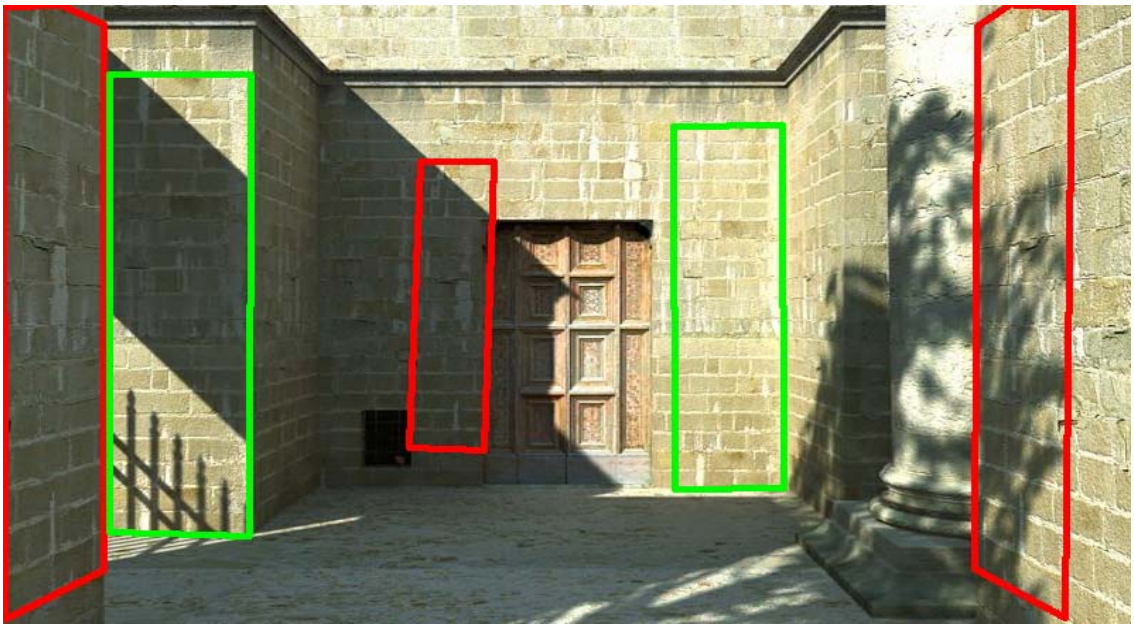


Repetición de texturas

- Pensad en formas inteligentes de romper los patrones!

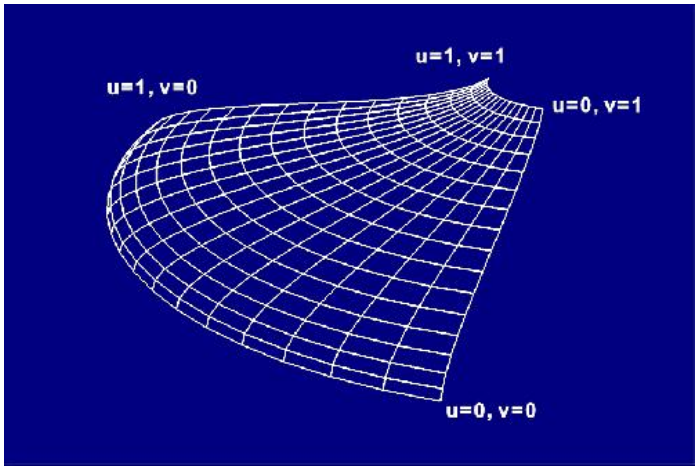
Repetición de texturas

- Pensad en formas inteligentes de romper los patrones!



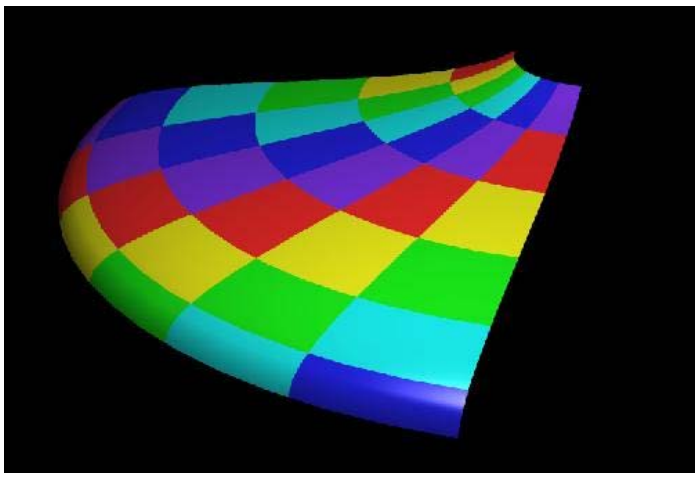
Superficies paramétricas

- (u,v) son las coordenadas paramétricas normalizadas de la superficie
- Multiplicando estas coordenadas por la resolución en pixels de la textura, obtenemos las coordenadas del pixel de textura (texel) correspondiente



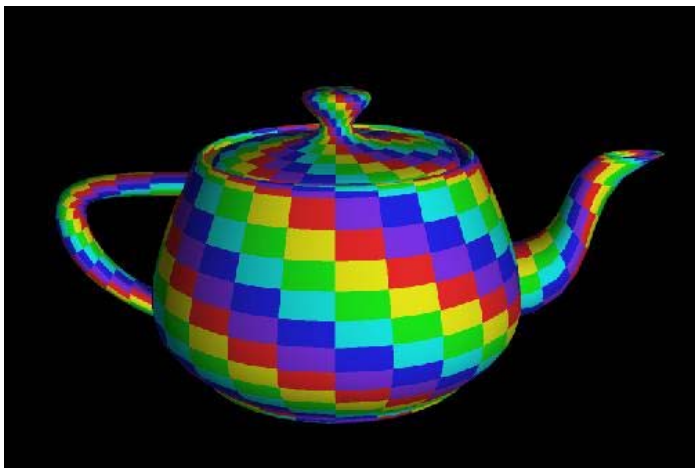
Superficies paramétricas

- (u,v) son las coordenadas paramétricas normalizadas de la superficie
- Multiplicando estas coordenadas por la resolución en pixels de la textura, obtenemos las coordenadas del pixel de textura (texel) correspondiente



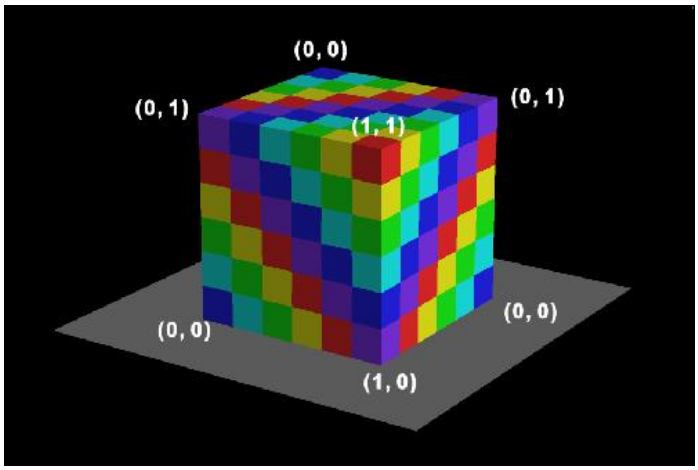
Superficies paramétricas

- Esta tetera contiene 32 parches (*patches*) paramétricos, a cada uno de los cuales se ha asignado una copia de la textura



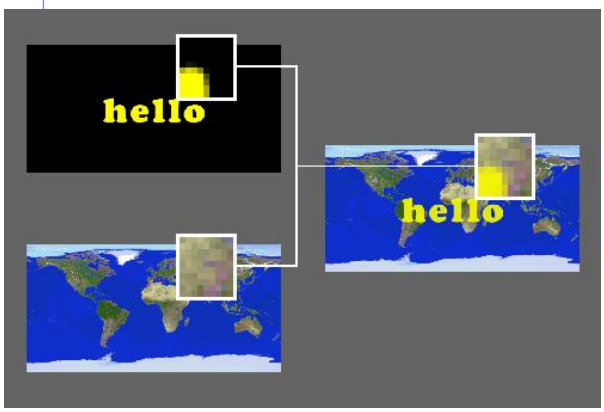
Superficies paramétricas

- Podemos usar la misma técnica a superifices no paramétricas, sin más que asignar coordenadas (u,v) [0..1] a cada vértice.



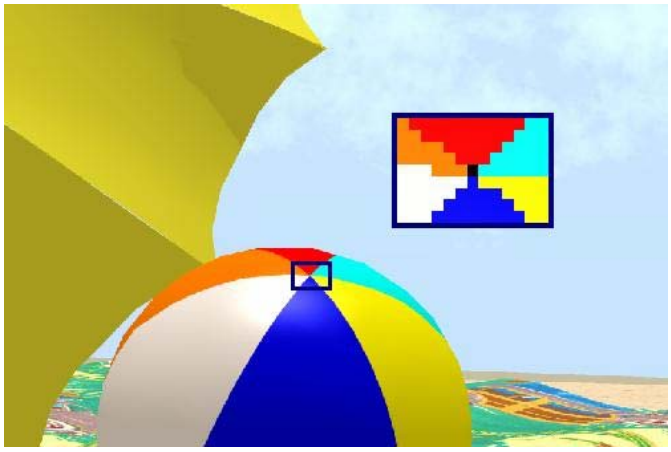
El canal alfa

- El canal alfa (RGBA) indica el grado de transparencia del pixel
- Permite (entre otra cosas) utilizar una textura encima de otra y combinarlas



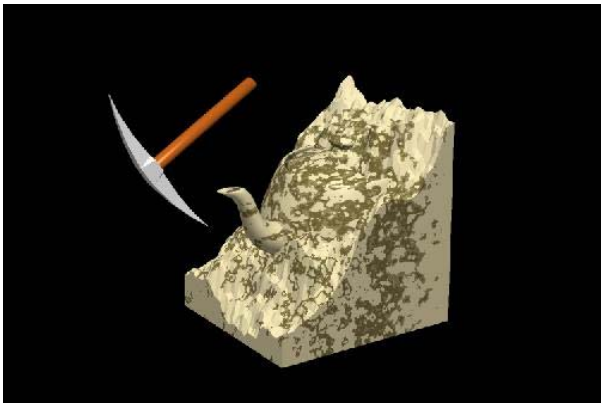
Aviso a los programadores

- El mapeado de texturas 2D no está exento de singularidades matemáticas
- En el mapeado esférico, todos los puntos y_{min} e y_{max} de la textura acaban colapsando en los polos
- Un programa debe estar preparado para manejar estos eventos...



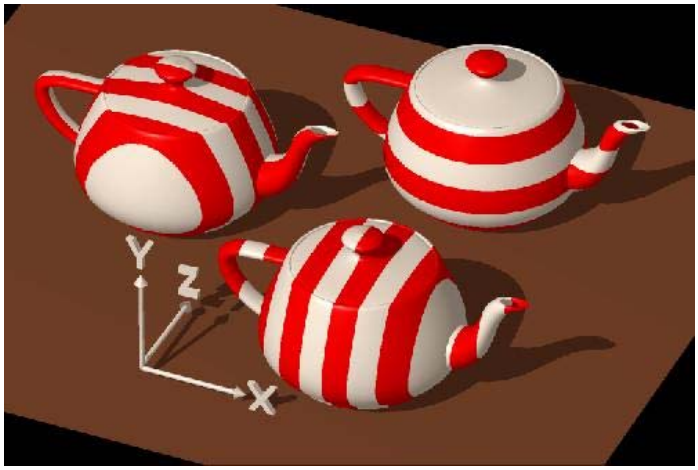
Texturas 3D

- El mapeado de texturas 3D usa directamente las coordenadas (x,y,z) del objeto para calcular directamente el color (similar a esculpir el objeto a partir de un bloque sólido)
- Generalmente no se define explícitamente un valor para cada (x,y,z), sino que se obtiene evaluando una función; por eso suelen llamarse **texturas procedurales**



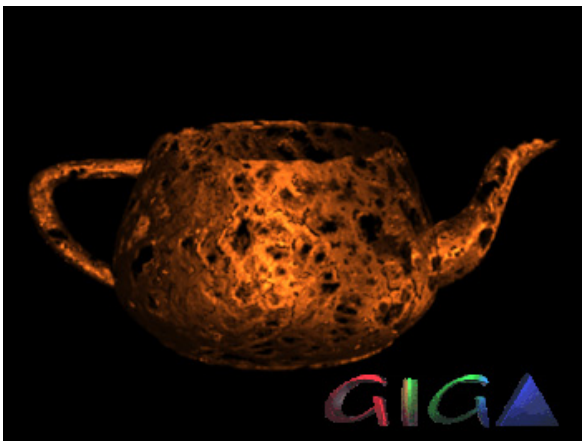
Texturas 3D

- Por ejemplo, para mapear la tetera de la izquierda hemos usado un procedimiento que hallaba la parte entera de la coordenada z de cada punto; si era par, pintamos de rojo, y si era impar, de blanco



Texturas 3D

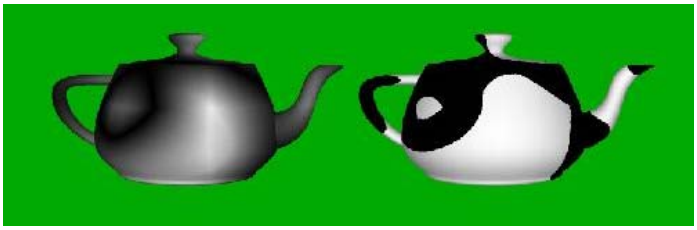
- Mármol, madera...
- Procedimientos similares pueden aplicarse durante el modelado



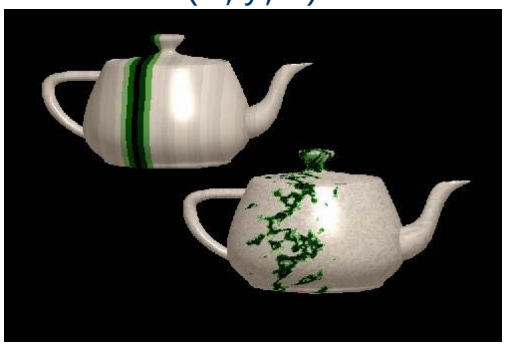
Texturas 3D

- Algunos ejemplos:
  - `gray = noise(x,y,z)`

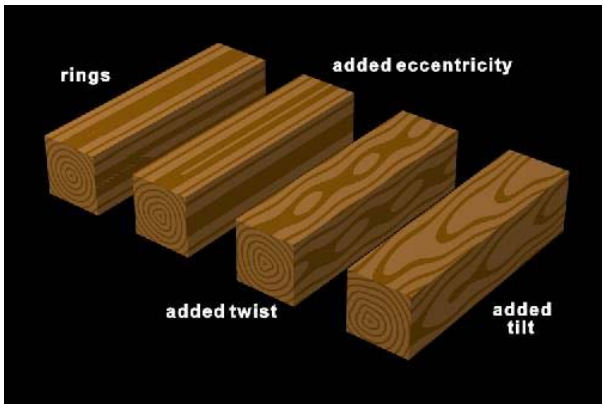
```
if(gray > threshold)
  choose white
else choose black
```



  - Mármol a partir de `noise(x,y,z)`

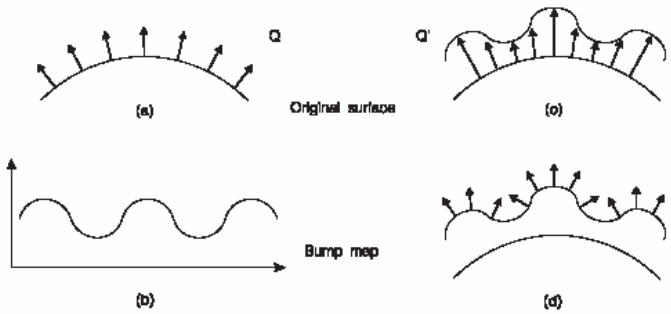


  - Madera a partir de círculos concéntricos



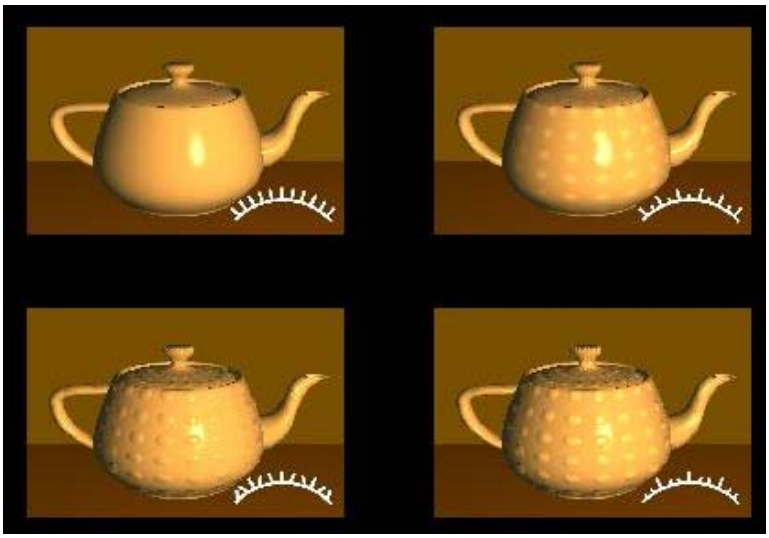
Bump mapping

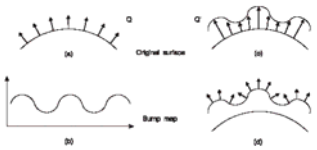
- Introducido por Blinn (1978)
- Se altera la normal a la superficie antes de los cálculos de sombreado



Bump mapping

- Introducido por Blinn (1978)
- Se altera la normal a la superficie antes de los cálculos de sombreado



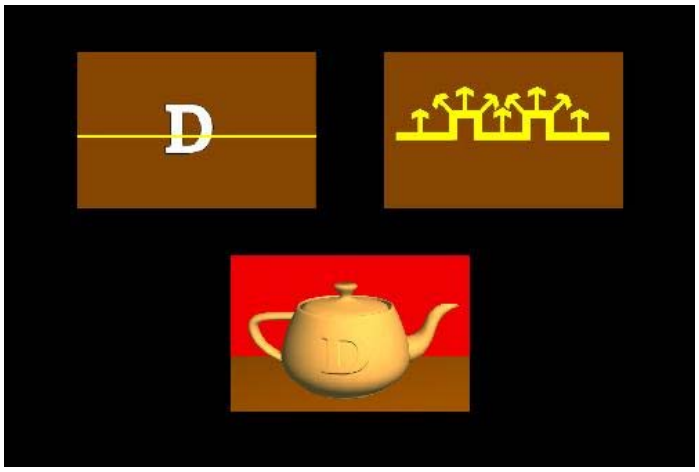


Bump mapping

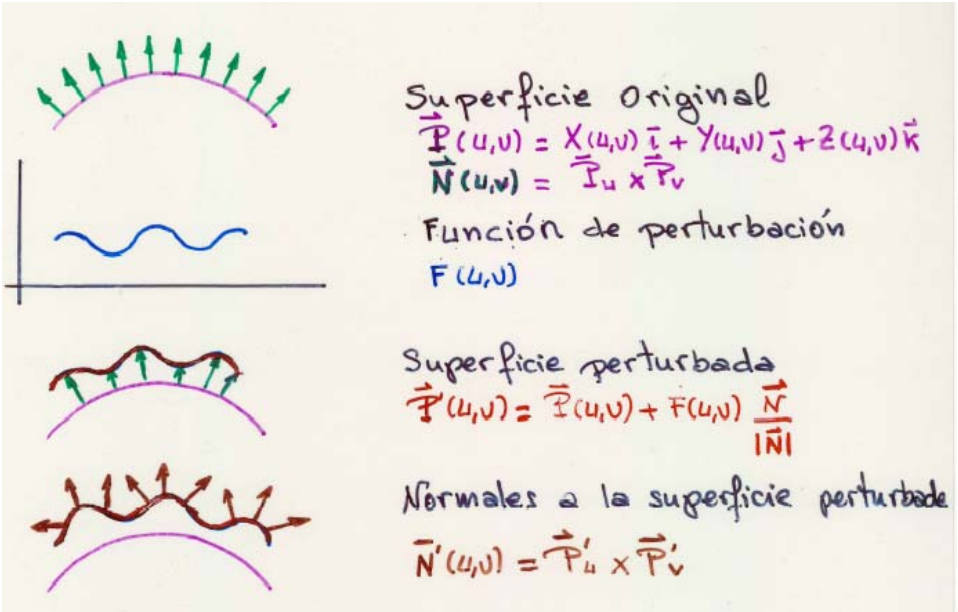
- Recuerda: el *bump mapping* no altera la superficie de los objetos
- Unicamente **engaña** al algoritmo de *shading* para crear superifices interesantes

Bump mapping

- Recuerda: el *bump mapping* no altera la superficie de los objetos
- Unicamente **engaña** al algoritmo de *shading* para crear superifices interesantes



Bump mapping



Bump mapping

$* \quad \vec{N}'(u,v) = \vec{P}'_u * \vec{P}'_v \quad$ siendo $\quad \vec{P}'(u,v) = \vec{P}(u,v) + F(u,v)\frac{\vec{N}}{|\vec{N}|}$

$$\vec{P}'_u = \vec{P}_u + F_u \cdot \frac{\vec{N}}{|\vec{N}|} + F\left(\frac{\vec{N}}{|\vec{N}|}\right)_u$$

$$\vec{P}'_v = \vec{P}_v + F_v \cdot \frac{\vec{N}}{|\vec{N}|} + F\left(\frac{\vec{N}}{|\vec{N}|}\right)_v$$

Suponiendo que F es una perturbación "pequeña"

$$\vec{P}'_u \simeq \vec{P}_u + F_u \cdot \frac{\vec{N}}{|\vec{N}|}$$

$$\vec{P}'_v \simeq \vec{P}_v + F_v \cdot \frac{\vec{N}}{|\vec{N}|}$$

Bump mapping

Por lo tanto

$$\vec{N}' = \vec{P}'_u \times \vec{P}'_v \simeq \left(\vec{P}_u + F_u \cdot \frac{\vec{N}}{|\vec{N}|}\right) \times \left(\vec{P}_v + F_v \cdot \frac{\vec{N}}{|\vec{N}|}\right)$$

$$\vec{N}' \simeq \vec{P}_u \times \vec{P}_v + (\vec{P}_u \times \vec{N}) \frac{F_v}{|\vec{N}|} + (\vec{N} \times \vec{P}_v) \frac{F_u}{|\vec{N}|} + \cancel{(\vec{N} \times \vec{N}) \frac{F_u F_v}{|\vec{N}|\,|\vec{N}|}}$$

Con lo cual

$$\vec{N}' \simeq \vec{N} + \vec{D} \quad \text{siendo} \quad \vec{D} = \left[(\vec{N} \times \vec{P}_v) F_u - (\vec{N} \times \vec{P}_u) F_v\right] / |\vec{N}|$$

Obsérvese que lo interesante de la función $F(u,v)$ es el valor de sus derivadas F_u y F_v.

Bump mapping

La expresión $\vec{N}' \simeq \vec{N} + \vec{D}$ no es independiente del cambio de escala del objeto.

Si la superficie sufre un cambio de escala de factor "s", entonces $\vec{N}$ sufre un cambio de factor "s^2" y $\vec{D}$ de factor "s".

Con objeto de que $\vec{D}$ sea invariante bajo cambio de escala se sugiere

$$\vec{D}' = a\,\vec{D}\,\frac{|\vec{N}|}{|\vec{D}|}$$ obsérvese que $|\vec{D}'| = a\,|\vec{N}|$

siendo

$$a = \left(F_u^2 + F_v^2\right)^{1/2}$$

Bump mapping

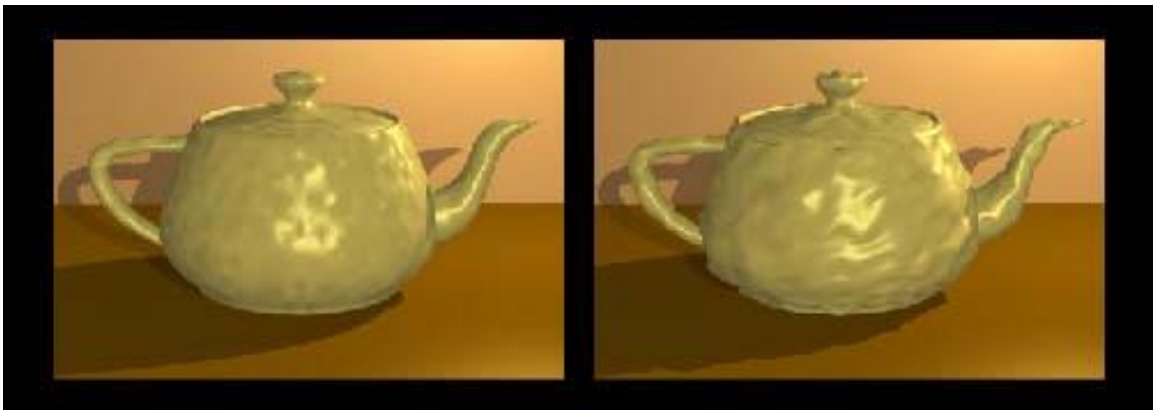
Obsérvese que el valor de a es $\frac{|\vec{D}'|}{|\vec{N}|}$, si la superficie $\vec{P}(u,v) = u\,\vec{i} + v\,\vec{j} + 0\,\vec{k}$ entonces $\vec{N} = (0,0,1)$

$\vec{D} = (-F_u, -F_v, 0)$

$$\left.\begin{aligned} \vec{D}' &= a\,(-F_u, -F_v, 0)\,\frac{1}{\sqrt{F_u^2 + F_v^2}} \\ |\vec{D}'| &= a\,1 \end{aligned}\right\} \quad a = \left(F_u^2 + F_v^2\right)^{1/2}$$

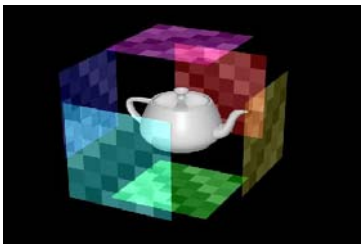
Bump vs. Displacement

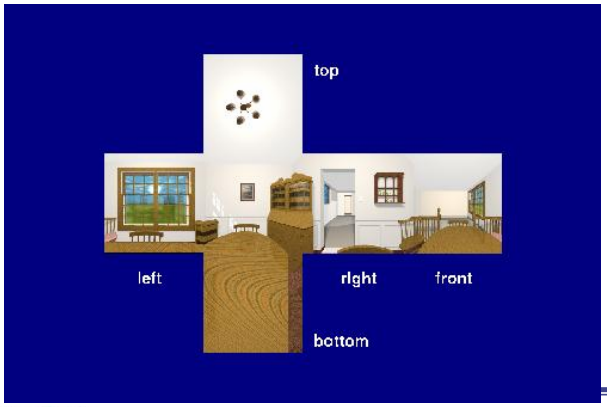
- El mapa de desplazamiento (Cook, 1984) sí modifica la superficie real
- Qué tetera tiene *bump* y cuál *displacement*?



Mapas de entorno

- a.k.a *environment maps* (Blinn y Newell, 1976). Efectos de reflejos sin necesidad de *ray tracing*
- Podemos usar un *z-buffer* en vez de trazar rayos





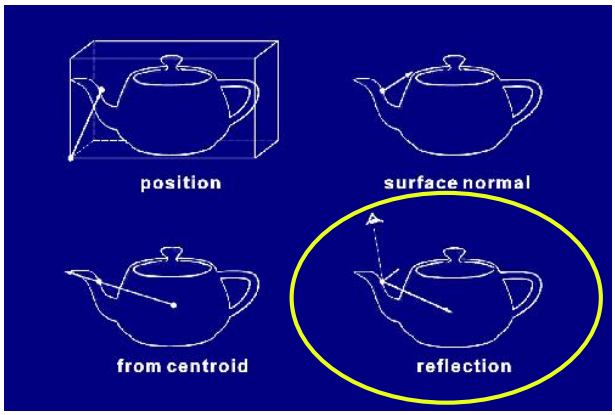
(un paréntesis)

- **<parentesis>**
  - No todo es *ray-tracing* en informática gráfica!!

- **</parentesis>**

Mapas de entorno

- ¿Recordáis que para elegir un punto teníamos diferentes opciones?

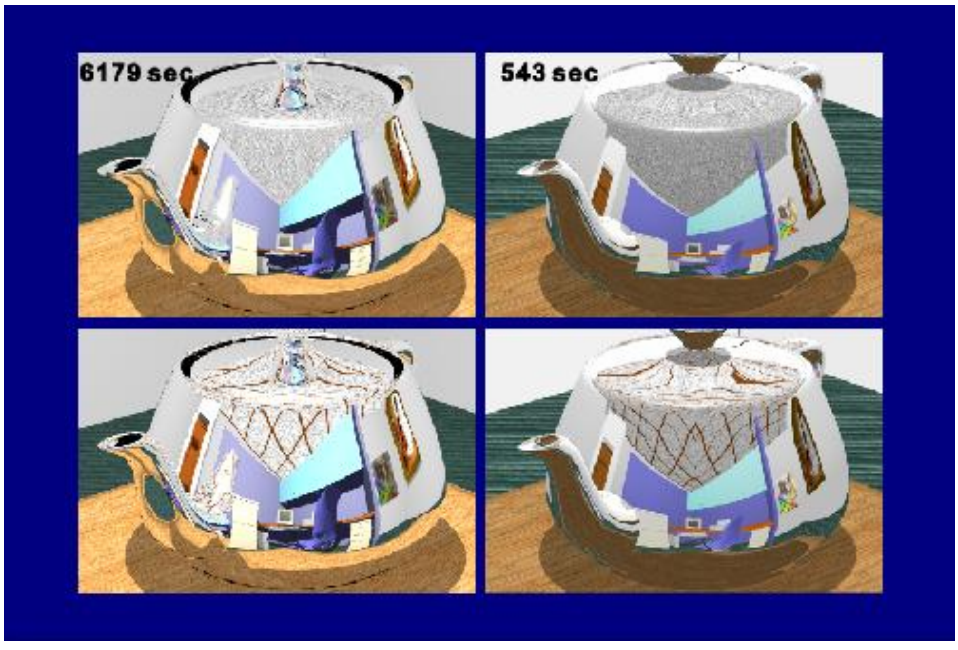


Mapas de entorno

- Trazado de rayos vs. mapa de desplazamiento: ¿veis alguna diferencia?

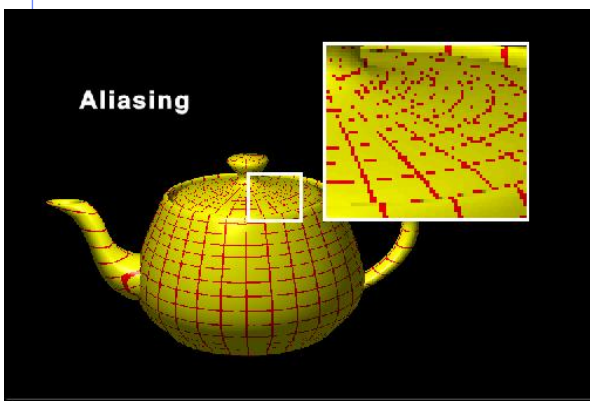
Mapas de entorno

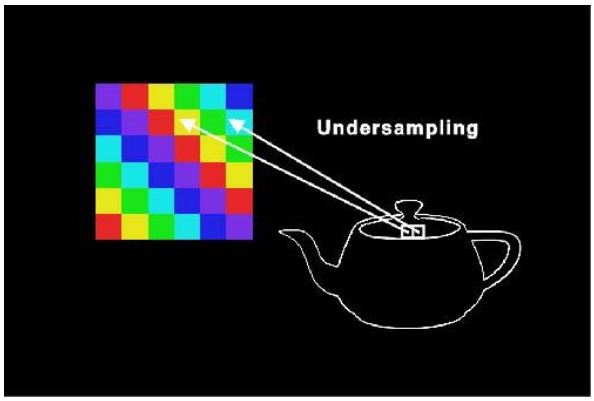
- Coste vs. resultado



Aliasing

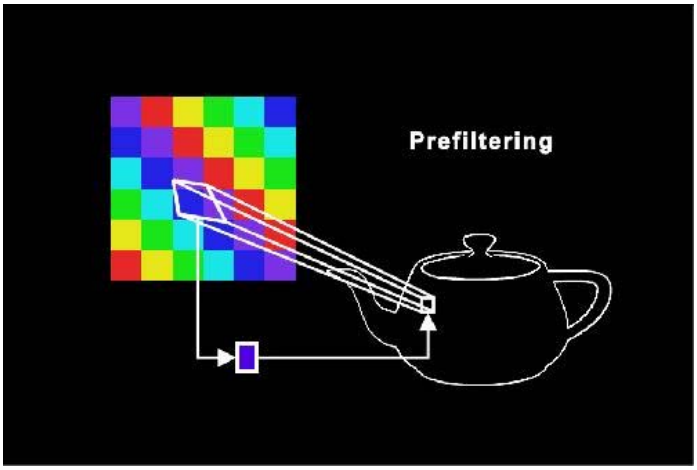
- El aliasing en las texturas es debido a que dos pixels adyacentes del objeto pueden no mapearse con dos pixels adyacentes de la textura





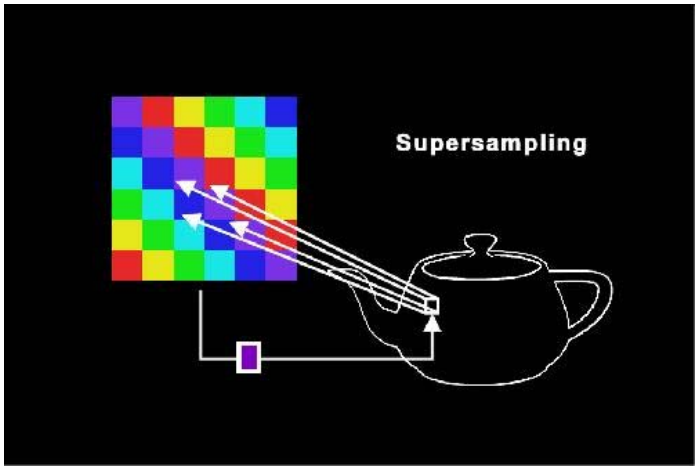
Antialiasing

- **Prefiltrado:** (Catmull, 1978)
- Trata los pixels como áreas
- Promedia los valores de todos los pixels de la textura (texels) que caen dentro del área proyectada

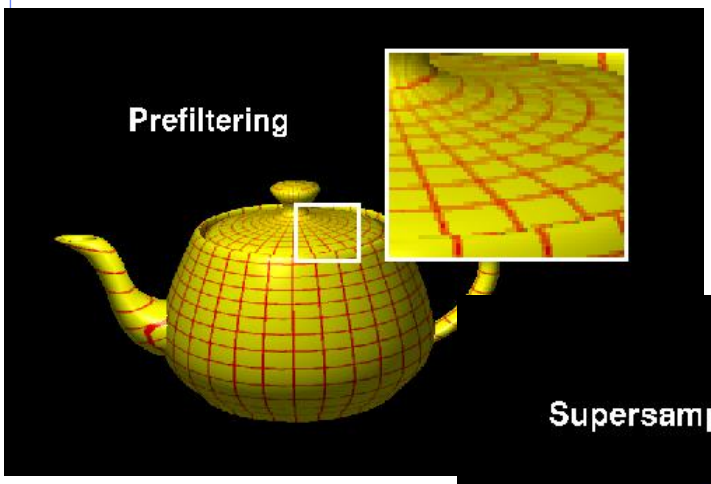


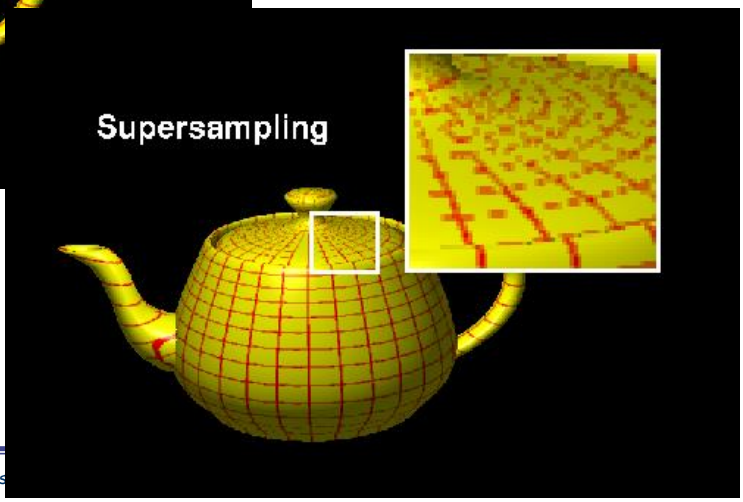
Antialiasing

- **Supersampling:** similar, pero se promedian varios valores dentro del área proyectada (Crow, 1981).
- En el ejemplo, se escogen los valores de las esquinas



Antialiasing

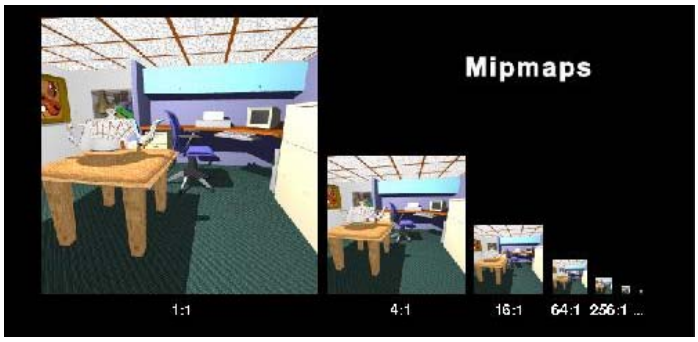




- ¿Cuál te parece mejor? ¿Por qué?

Mipmapping

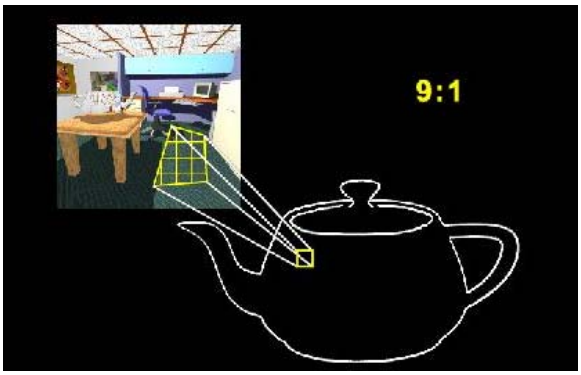
- El antialiasing es un método computacionalmente caro
- Mipmapping se basa en precalcular algunos de los colores promedio de la textura (Williams, 1983)



- El proceso acaba con una textura de un pixel, que contiene el color promedio de toda la textura original

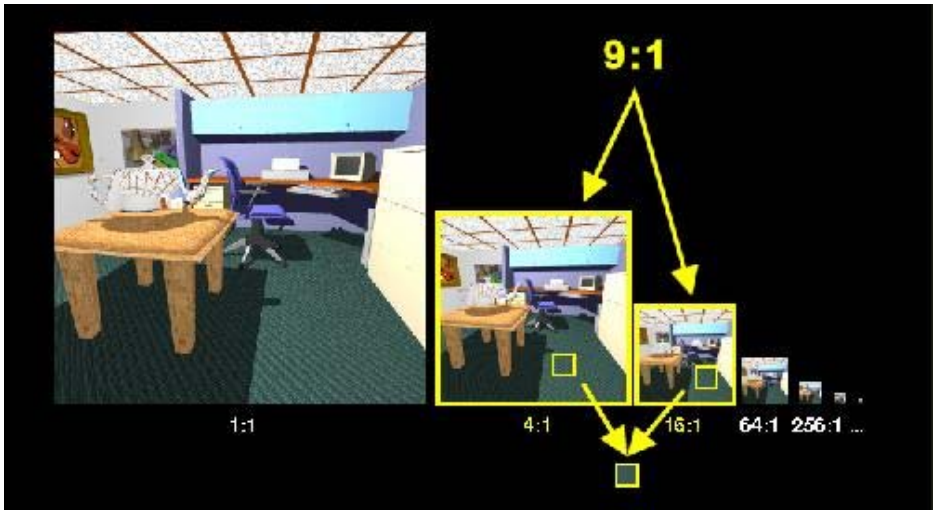
Mipmapping

- Durante el mapeado de la textura:
  - Se mapea el área de cada pixel sobre la textura original
  - Se calcula el número de texels del área de la textura
  - Ese número indica el número de texels que influirán en el color final del pixel
  - En el ejemplo, tenemos que 9 texels se mapearán en un pixel del objeto, luego el ratio texels-pixels es 9:1



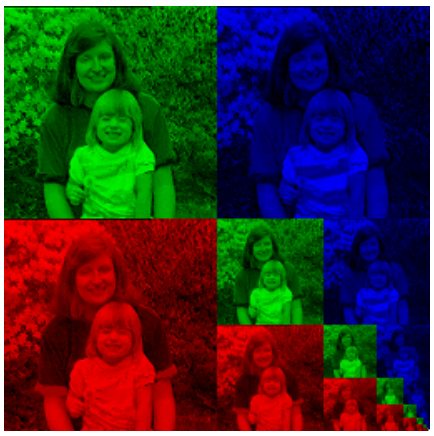
Mipmapping

- Para calcular el color final, se encuentran las dos versiones de la textura cuyos ratios más se aproximen al ratio texel-pixel obtenido
- Buscamos los colores en los mapas correspondientes y los promediamos



Mipmapping

- La figura muestra una forma eficiente de almacenar un mipmap en memoria (overhead=1/3)

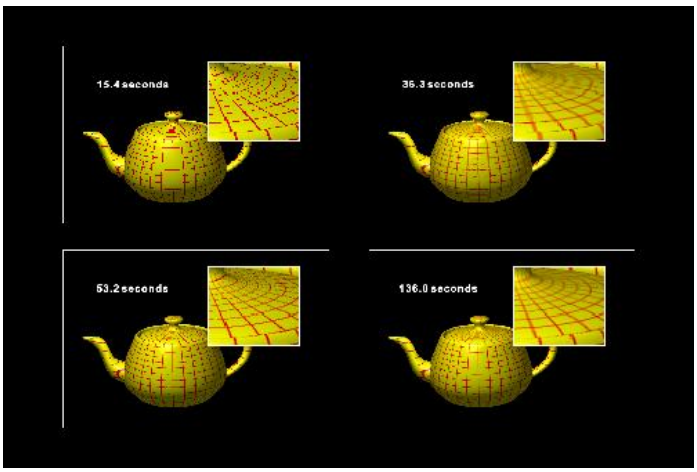


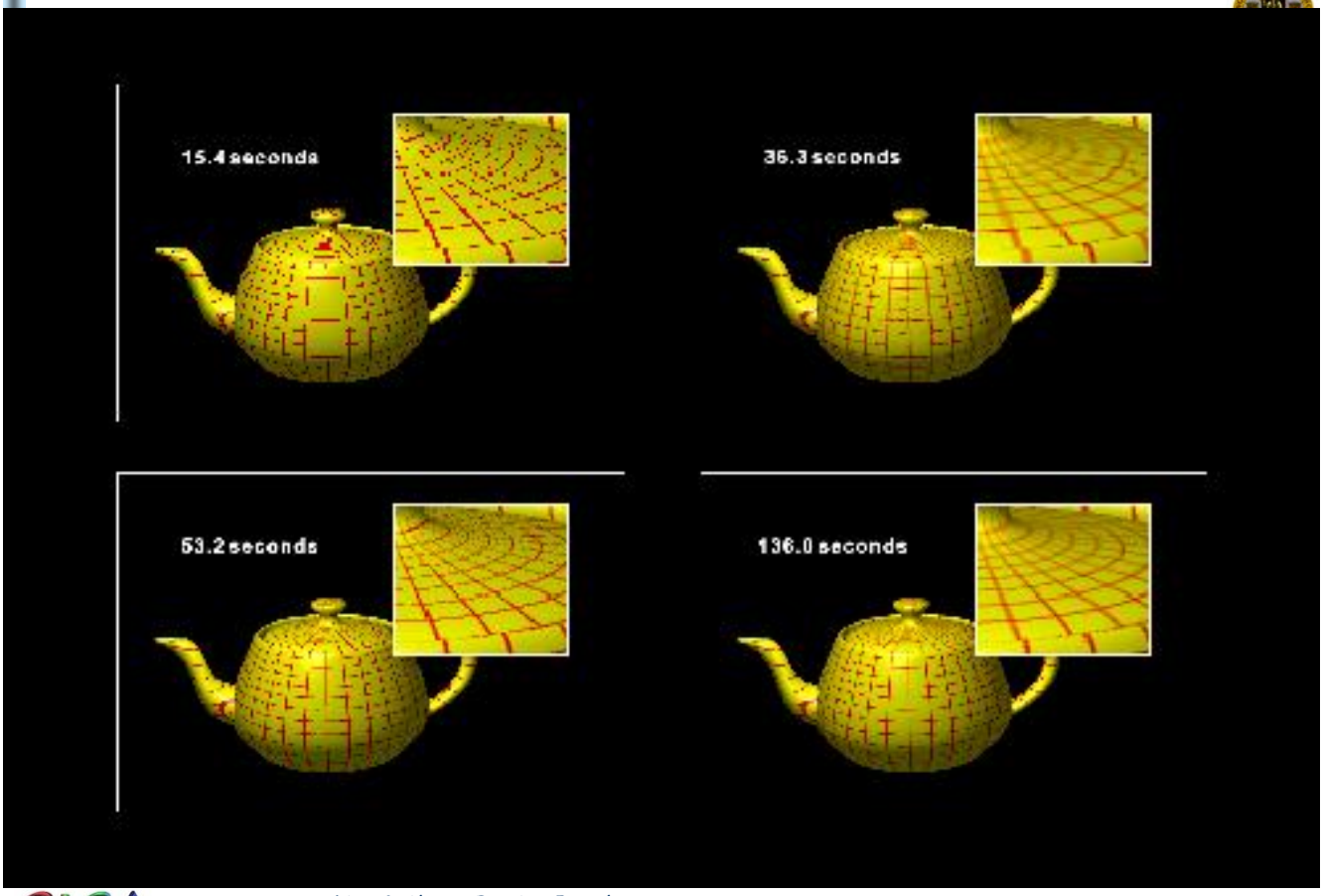
Mipmapping

- Los resultados del mipmapping son evidentes: eliminan altas frecuencias allí donde esperamos encontrar bajas frecuencias

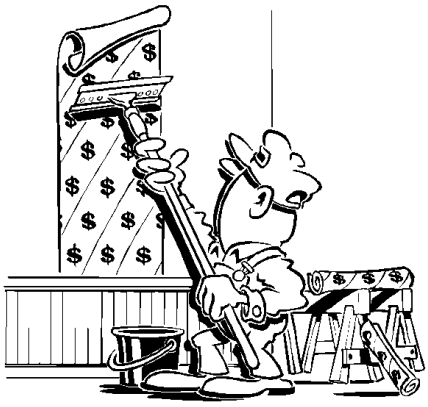
Mipmapping

- Arriba a la izda, sentido horario: sin antialiasing, con mipmapping, supersampling con 9 texels por pixel, y supersampling con 9 texels por pixel más mipmapping





Grupo de Informática Gráfica Avanzada
Universidad de Zaragoza

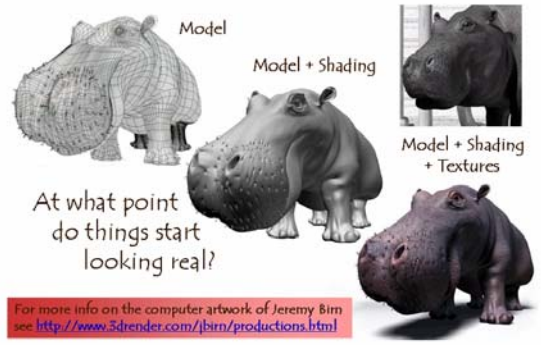


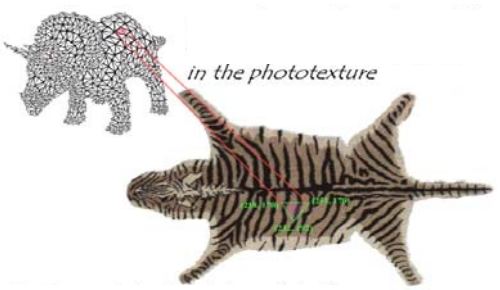
Modelado Visual y Animación

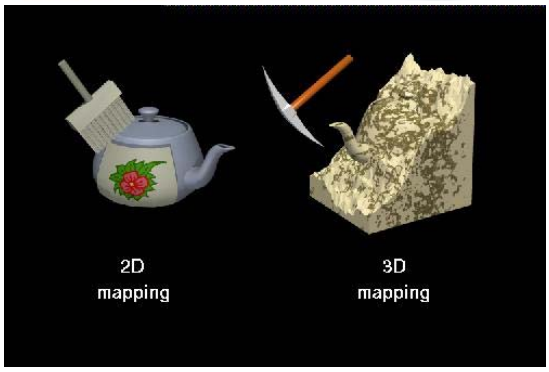
Texturas (in a nutshell)

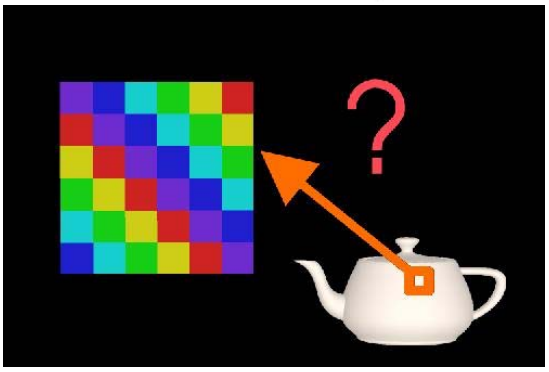
Diego Gutiérrez y Francisco J. Serón

Texturas (in a nutshell)

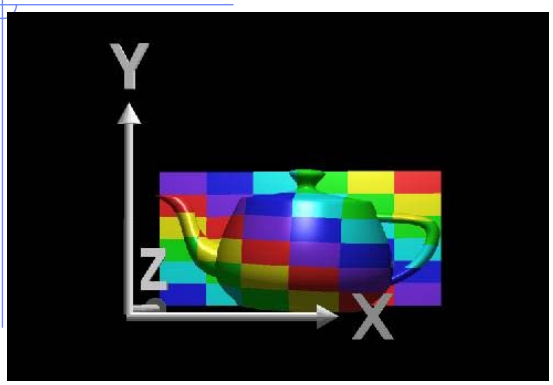


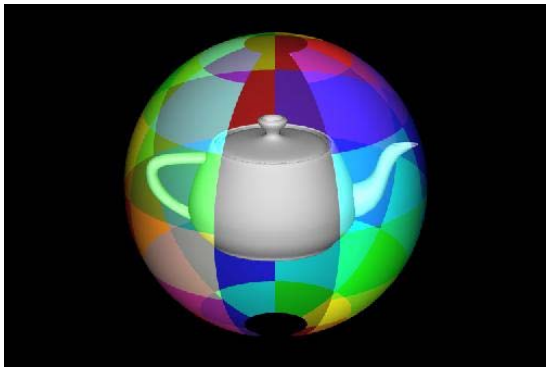


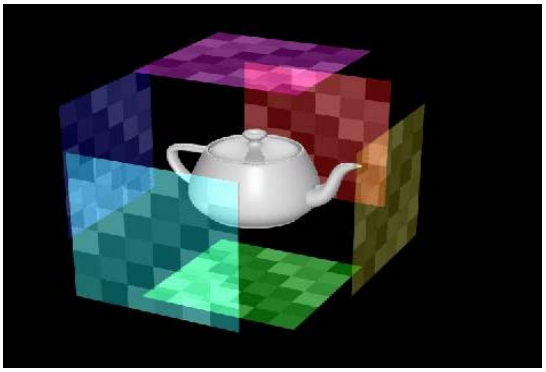




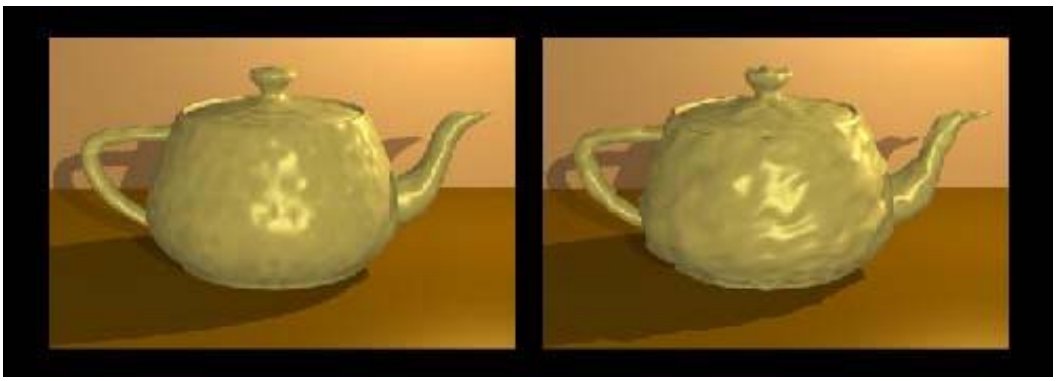
Texturas (in a nutshell)

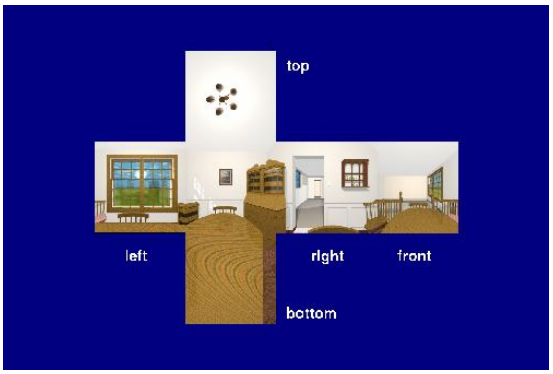




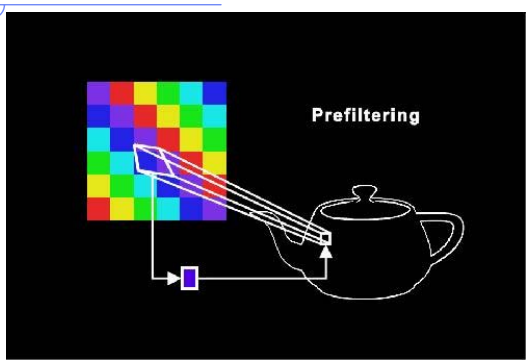


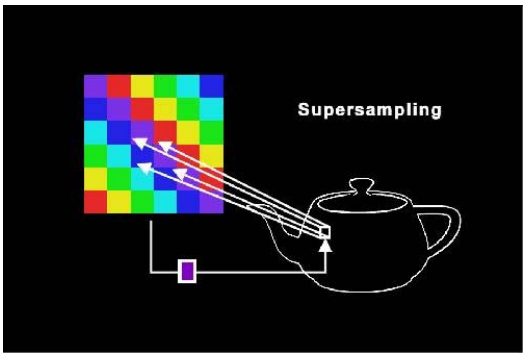
Texturas (in a nutshell)

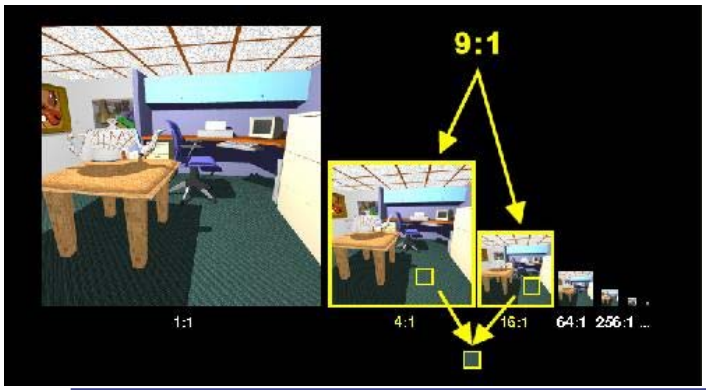


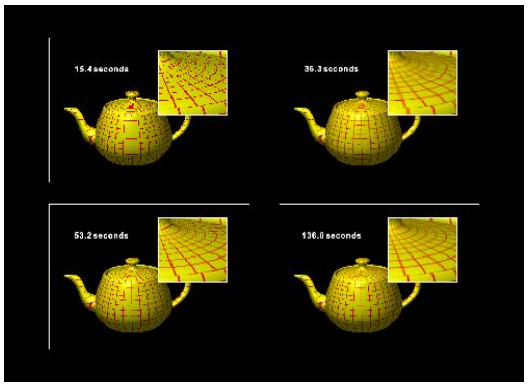


Texturas (in a nutshell)









Bibliografía

(Blinn, 1978) J. Blinn, Simulations of Wrinkled Surfaces. Computer Graphics 12(3) August 1978, 286-292.

(Blinn and Newell, 1976) J. Blinn and M. Newell, Texture and Reflection in Computer Generated Images. Communications of the ACM 19(10) October 1976, 542-546.

(Catmull, 1974) E. Catmull, A Subdivision Algorithm for Computer Display of Curved Surfaces. PhD thesis, Department of Computer Science, University of Utah, December 1974.

(Catmull, 1978) E. Catmull, A Hidden-Surface Algorithm with Anti-Aliasing. Computer Graphics 12 (3) August 1978, 6-10.

(Cook, 1984) R. Cook, Shade Trees. Computer Graphics 18 (3) July 1984 223-231.

(Crow, 1981) F. Crow, A Comparison of Antialiasing Techniques. IEEE Computer Graphics and Applications. 1 (1) January 1981, 40-49.

(Peachey, 1985) D. Peachey, Solid Texturing of Complex Surfaces. Computer Graphics 19 (3) July 1985, 279-286.

(Perlin, 1985) K. Perlin, An Image Synthesizer. Computer Graphics 19 (3) July 1985, 287-296.

(Williams, 1983) L. Williams, Pyradmial Parametrics. Computer Graphics 17 (3) July 1983, 1-11.

D. Ebert, F. K. Musgrave, D. Peachey, K. Perlin and S. Worley, Texturing and Modeling: A Procedural Approach. Academic Press, 1994.

J. Foley, A. vanDam, S. Feiner and J. Hughes, Computer Graphics: Principles and Practice. 2nd. ed. Addison-Wesley, 1990.

A. Watt and M. Watt, Advanced Animation and Rendering Techniques: Theory and Practice. Addison-Wesley, 1992.

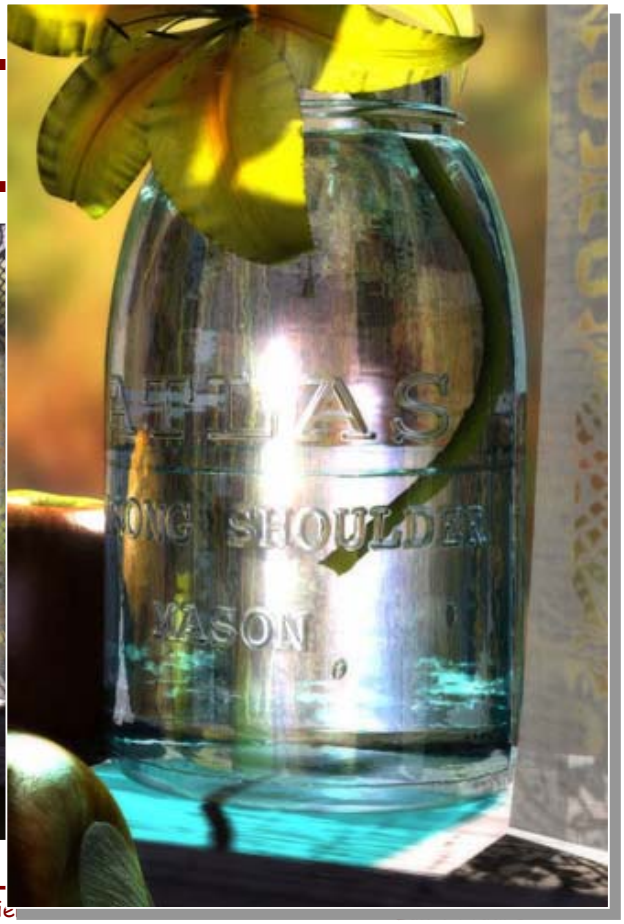
Luz Virtual

Introducción

Luz Virtual

Introducción

Luz Virtual



Luz Virtual

Todo lo que debes saber...

Aprender a iluminar es aprender a observar

Luz Virtual

Introducción

- El término “iluminación” en el ámbito de la informática gráfica suele incluir conceptos tales como la descripción de las características de un material y su respuesta ante la luz (shaders).
- Estas notas tratan el tema de la iluminación desde un punto de vista más artístico, en términos de diseño de ambientes y colocación de luces, siempre orientado al mundo digital.
- No existen reglas fijas: todo lo que se diga aquí debe utilizarse únicamente como guía. El objetivo es enseñar a plasmar sobre el ordenador la imagen que el artista ha visualizado mentalmente.
- Las “reglas” deben servir a la imagen, no someterla.

Luz Virtual

Objetivos de la iluminación

- Crear una ilusión tridimensional a partir de una imagen plana.
- Producir un impacto visual atrayente.
- Crear una ambientación adecuada.
- Dotar de personalidad a los distintos elementos y situaciones.
- Encauzar la atención de la audiencia hacia los puntos más relevantes.
- Ocultar defectos de modelado, animación...
- Integrar y dar coherencia a la historia.
- En definitiva, hacer la historia (o imagen estática), fácilmente legible, atractiva y, dentro de su contexto, creíble.

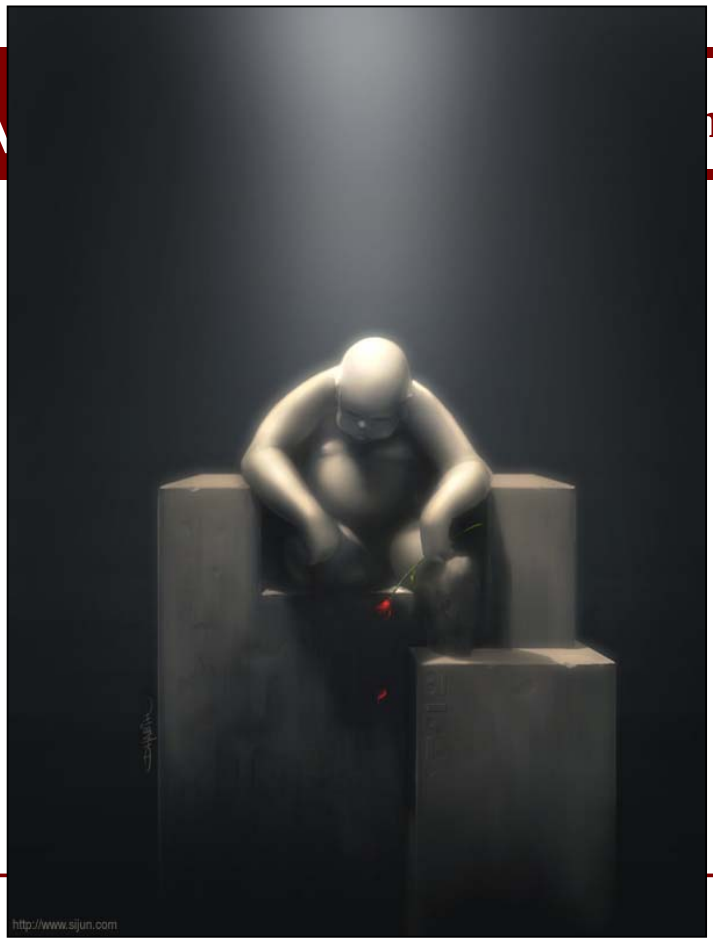
Luz Virtual

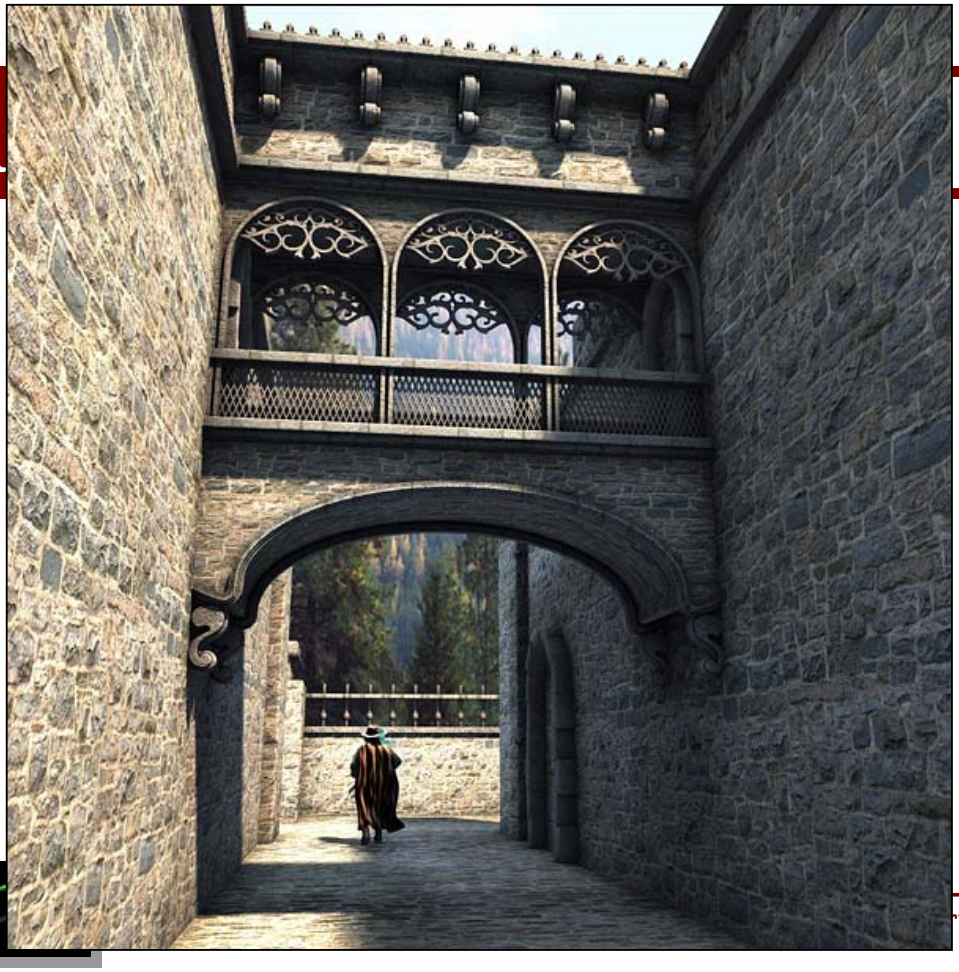
Encauzar la atención

- Aspecto especialmente importante en el ámbito de la animación por ordenador.
- Lo primero que detecta el ojo humano en una escena es (sin orden fijo):
  - La figura humana
  - El movimiento
  - El color rojo
- Hay que saber utilizar esto para favorecer la legibilidad de una escena (un personaje irrelevante pasando por detrás de nuestro héroe NO PUEDE ir vestido de rojo...)

Luz V

nción





Luz Virtual

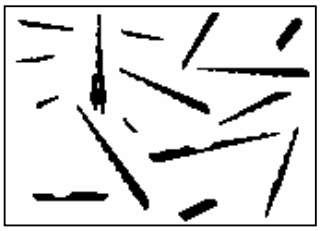
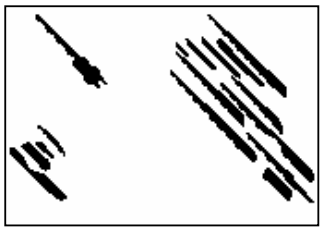
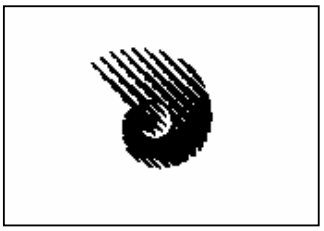
Composición

- El término composición se utiliza para describir una serie de principios visuales relacionados.
- La iluminación debe partir de una buena composición, o tratar de corregirla si esta es defectuosa.
- Los principios que se van a presentar actúan de manera conjunta, aunque se expliquen separadamente.
- Dependiendo de muchos factores, uno será más fuerte que otro en un momento dado.
- Una vez más, las reglas no son fijas.

Luz Virtual

Composición

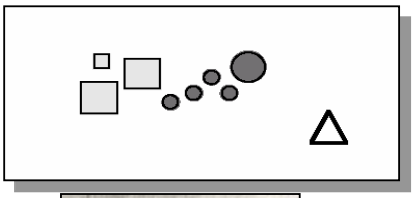
- La teoría de la Gestalt está basada en la noción de que tenemos un deseo intrínseco por la unidad y la armonía. El principio básico de la teoría es que vemos el conjunto antes que las partes que lo componen, y este conjunto es además superior a la suma de sus partes
- El ojo y el cerebro buscan organizar las imágenes. Si no lo consiguen, aparecen caóticas

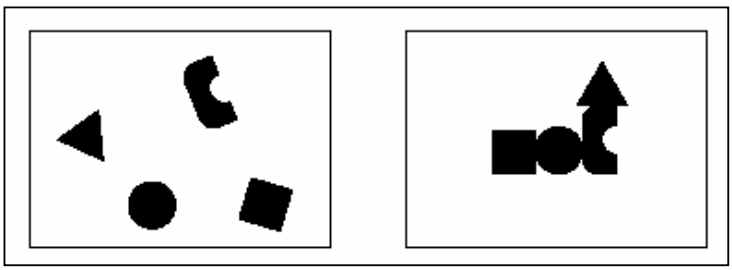
Luz Virtual

Composición

- Unidad y armonía:



- El ojo perderá demasiado tiempo intentando “ordenar” una escena sin unidad ni armonía. Por otra parte, una escena demasiado “ordenada” será visualmente poco atractiva.



Luz Virtual

Composición

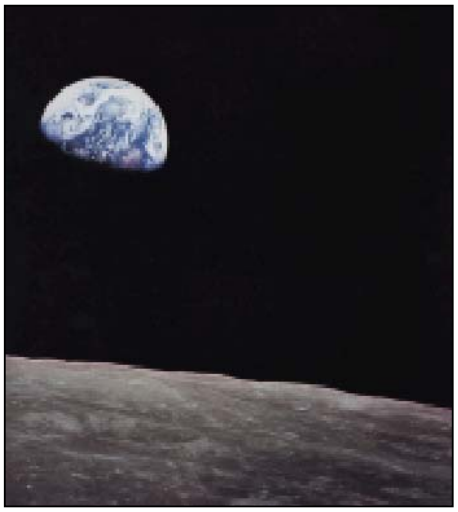
Luz Virtual

Composición

Luz Virtual

Composición

- Formas familiares:

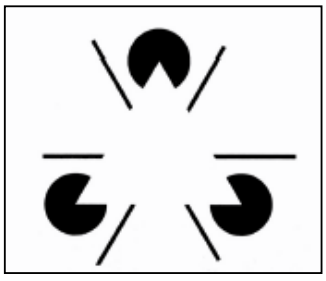


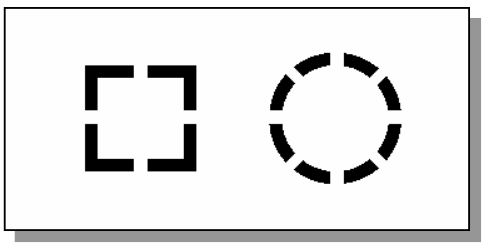
- El ojo busca interpretar el mundo a través de formas familiares. Una escena que carezca de ellos será más difícil de leer y desorientará al espectador.

Luz Virtual

Composición

- Contornos imaginarios :





- Tenemos la habilidad de completar imágenes (de nuevo buscando unidad, armonía y formas reconocibles!)

Luz Virtual

Composición

- Contornos imaginarios:

- Estrechamente relacionados con el reconocimiento de formas

Luz Virtual

Composición

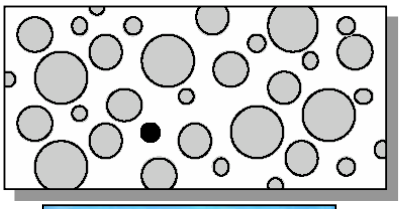
- Eliminación:

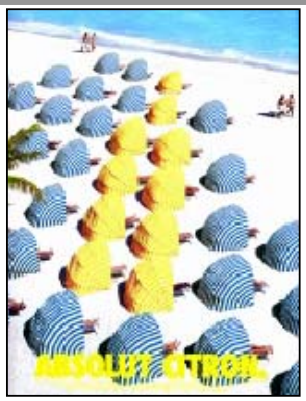
- Consiste en deshacerse de información no esencial, para realzar más lo que dejamos (Walt Disney…)

Luz Virtual

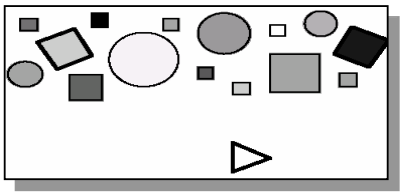
Composición

- Foco: énfasis





- Una imagen necesita un punto focal donde centrar la atención. Si no, obtenemos una imagen visualmente "plana". O no.



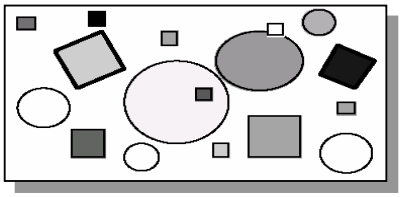
Luz Virtual

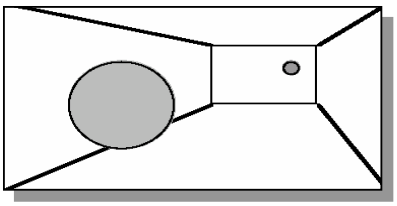
Composición

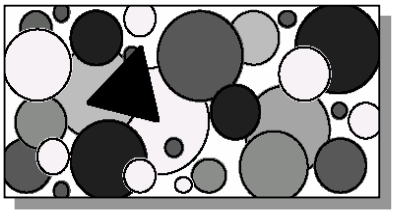
Luz Virtual

Composición

- Foco: énfasis







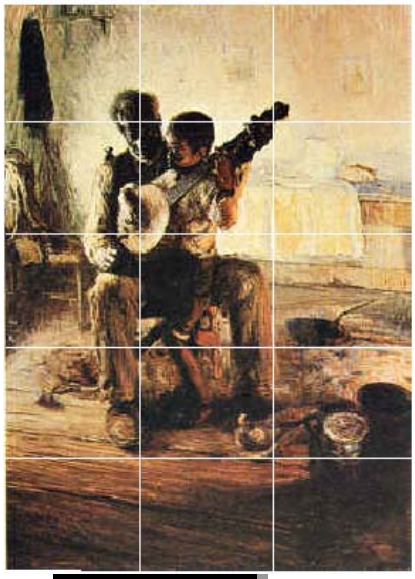
Luz Virtual

Composición

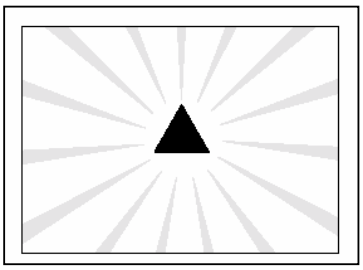
- Foco: énfasis
- Enfasis por reconocimiento y movimiento.

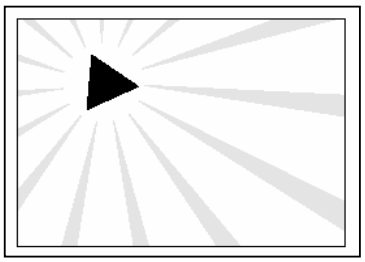
Composición

- Foco:



- Inconscientemente, dividimos la imagen en cuatro cuadrantes: si el foco no está en el centro, es mejor que caiga en uno de ellos. (regla de los treses y cincos)





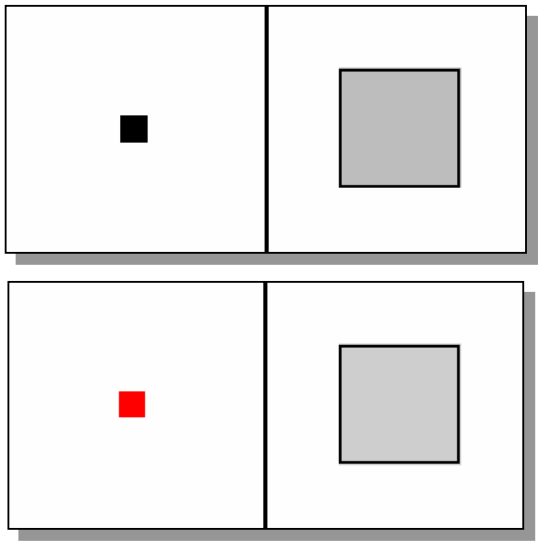
Composición

- Foco:
- Reconstruimos las escenas a base de recorrer la imagen en busca de puntos focales (2º)

Luz Virtual

Composición

- Equilibrio:



- Dividimos la imagen en un eje horizontal y otro vertical
- Por forma, color, contraste, peso aparente...
- Un objeto desequilibrado llama la atención y es visualmente inquietante (mansión encantada en una película de terror...).

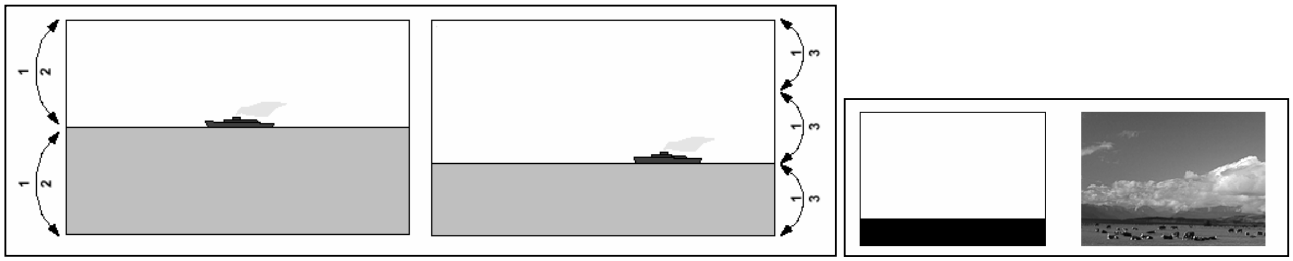
Luz Virtual

Composición

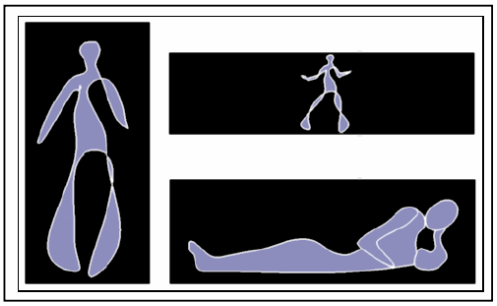
Luz Virtual

Composición

- Equilibrio:



- Formato: transmiten diferentes sensaciones. Rebelarse contra los formatos “standard”!



Luz Virtual

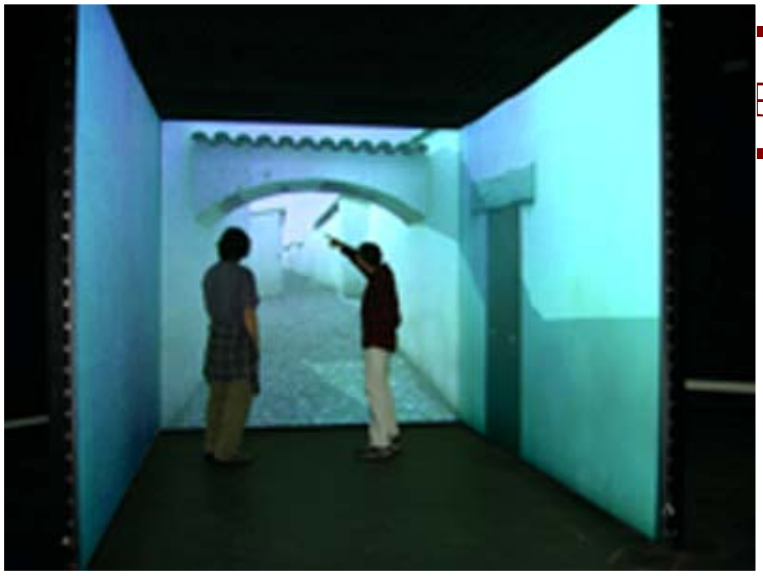
Formato

Luz Virtual

Formato

Luz Virtual

Formato

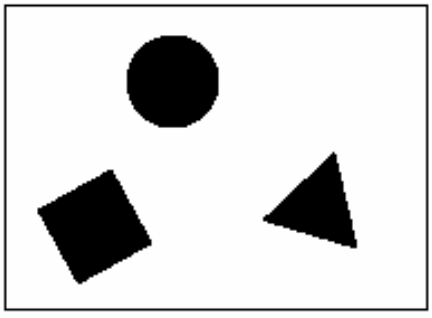


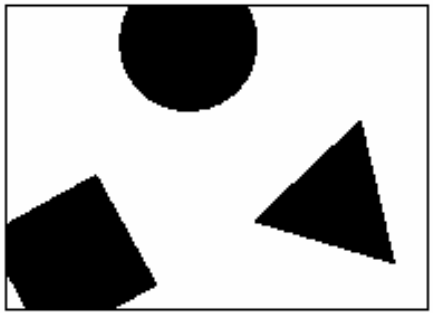
Formato

Luz Virtual

Composición

- Encuadre: imágenes que “se salen” de los límites del cuadro crean un espacio más allá del encuadre. La composición queda más plana si nada se sale del cuadro

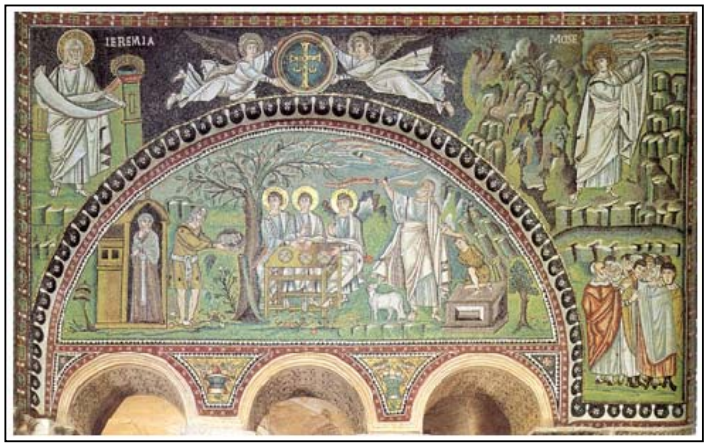


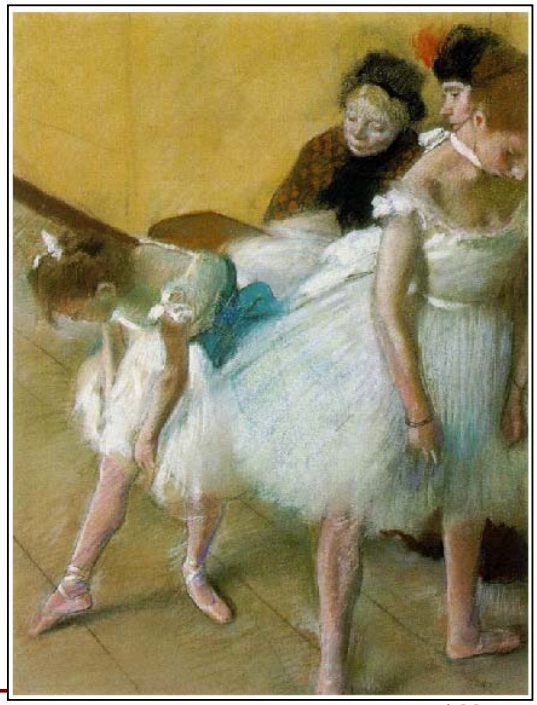


Luz Virtual

Composición

- Encuadre





Luz Virtual

Composición

- No siempre es aconsejable un perfecto equilibrio

- IZDA: perfecto equilibrio, ausencia de diagonales…
- DCHA: la tensión provocada por el movimiento potencial del esqueleto es acentuada por el desequilibrio vertical de la imagen

Luz Virtual

Composición

Luz Virtual

Composición

Luz Virtual

Composición

Luz Virtual

Composición

- Una pequeña prueba…

Luz Virtual

Composición

- Líneas y contornos: líneas reales, implícitas y psicológicas.

Luz Virtual

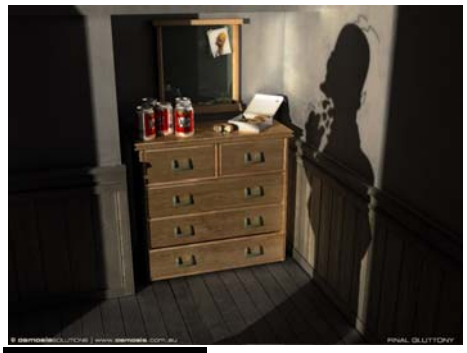
Composición

- Jugando con luces y sombras podemos añadir más líneas a una escena que las puramente geométricas.
- Las sombras *sugieren*

Luz Virtual

Composición

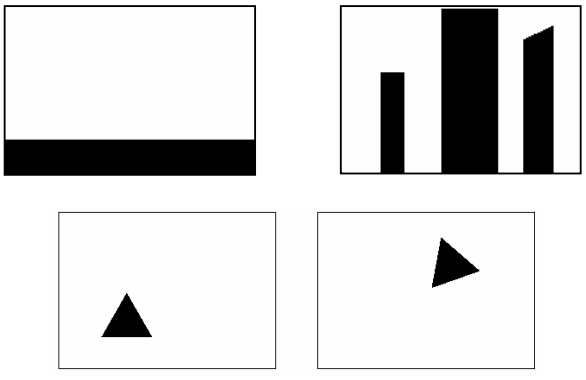
- Jugando con luces y sombras podemos añadir más líneas a una escena que las puramente geométricas.
- Las sombras *sugieren*



Luz Virtual

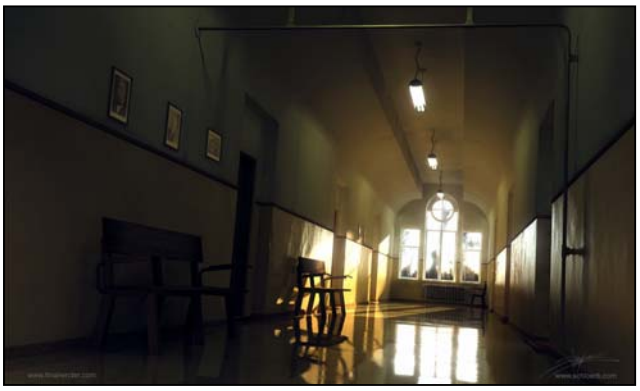
Composición

- Líneas horizontales: estabilidad.
- Líneas verticales: movimiento potencial.
- Líneas diagonales: inestabilidad, movimiento y profundidad.



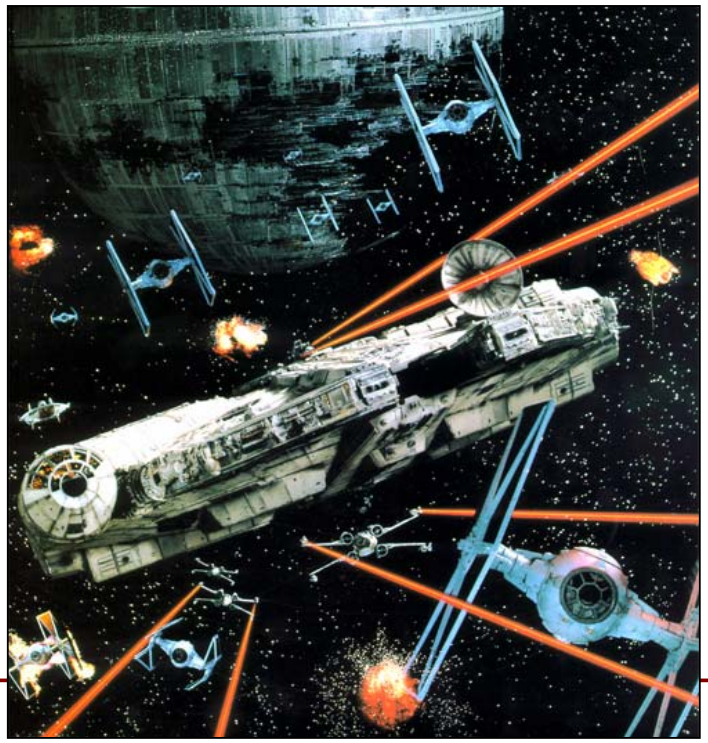
Luz Virtual

Composición



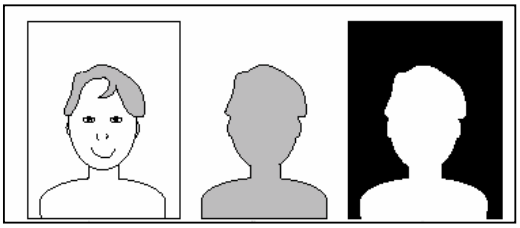
Luz Virtual

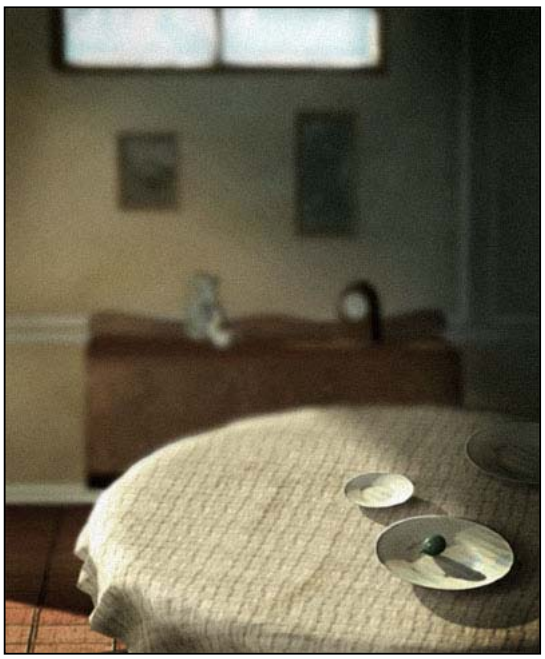
Composición



Composición

- Formas: reconocimiento de formas familiares. División en zonas positivas (primer plano), y negativas (fondo).



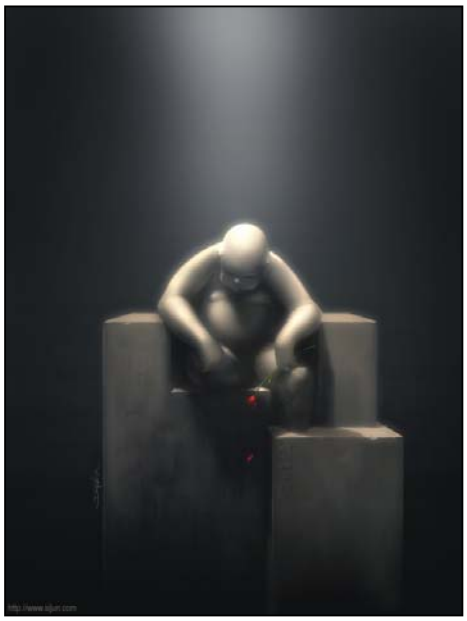


El papel del color

- Dotar a una imagen, escena, animación o película de emoción implica seleccionar una paleta de colores adecuada (Marea Roja, El Quinto Elemento, El Ultimo Hombre...).
- Aunque es algo muy subjetivo, existen algunas generalizaciones: colores fríos y cálidos; los saturados resaltan, los apagados se funden con el fondo.
- Rojo (enfado, pasión, violencia); verde (seguridad, equilibrio); azul (orden, paz); violeta (misticismo, magia); amarillo (felicidad, hambre, timidez); naranja (color festivo); marrón (responsabilidad, pobreza); rosa (carácter femenino, salud); negro (formal, malvado, misterioso); blanco (virginal, inocente); gris (indiferencia, tristeza)...

Luz Virtual

El papel del color



Luz Virtual

Algoritmos de iluminación

- Primero hay que entender cómo interacciona la luz con la materia (reflexiones especular y difusa).
- Scanline
- Z-buffer
- Trazado de rayos (ray tracing)
- Radiosidad
- Trazado de rayos + radiosidad
- Radiancia
- Cone tracing, path tracing
- Photon mapping

Luz Virtual

Algoritmos de iluminación

Luz Virtual

Algoritmos de iluminación

- La iluminación es otra herramienta más para conseguir imágenes realistas
- Una iluminación incorrecta no producirá la sensación adecuada
- Punto de vista artístico y algorítmico! El mayor problema es que los modelos de iluminación digitales sólo ofrecen una burda y lenta aproximación de la compleja fenomenología de la interacción real de la luz con la materia.

Algoritmos de iluminación

Luz Virtual

Todo lo que debes saber...

Aprender a iluminar es aprender a observar

Luz Virtual

Introducción

- El término “iluminación” en el ámbito de la informática gráfica suele incluir conceptos tales como la descripción de las características de un material y su respuesta ante la luz (shaders).
- Estas notas tratan el tema de la iluminación desde un punto de vista más artístico, en términos de diseño de ambientes y colocación de luces, siempre orientado al mundo digital.
- No existen reglas fijas: todo lo que se diga aquí debe utilizarse únicamente como guía. El objetivo es enseñar a plasmar sobre el ordenador la imagen que el artista ha visualizado mentalmente.
- Las “reglas” deben servir a la imagen, no someterla.

Luz Virtual

Introducción

Luz Virtual

Objetivos de la iluminación

- Crear una ilusión tridimensional a partir de una imagen plana.
- Producir un impacto visual atrayente.
- Crear una ambientación adecuada.
- Dotar de personalidad a los distintos elementos y situaciones.
- Encauzar la atención de la audiencia hacia los puntos más relevantes.
- Ocultar defectos de modelado, animación...
- Integrar y dar coherencia a la historia.
- En definitiva, hacer la historia (o imagen estática), fácilmente legible, atractiva y, dentro de su contexto, creíble.

Objetivos de la iluminación

- Ejemplos: ocultar defectos de modelado o animación, dar coherencia a la imagen…

Personalidad

- Iluminación baja con sombras definidas: maldad, algo criminal.
- Iluminación baja con sombras difusas: sensualidad.
- Luz desde arriba: vértigo si el personaje mira hacia abajo; esperanza si mira hacia arriba.
- Iluminación ilógica: se usa para narrar sueños o alucinaciones.
- Queramos o no, reaccionamos emocionalmente a casi todo lo que vemos. Algunas reacciones son casi genéticas (oscuridad), otras dependen de la cultura (color rojo), y otras son de índole completamente personal. Un buen diseño de iluminación debe saber aprovechar estas reacciones sutiles para guiar al espectador durante la historia.

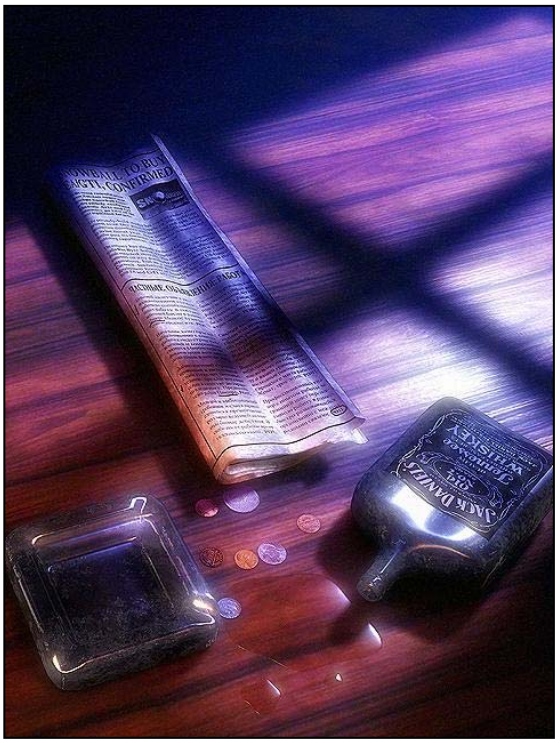
Luz Virtual

Personalidad

Luz Virtual

Personalidad

- Colores cálidos y sombras duras



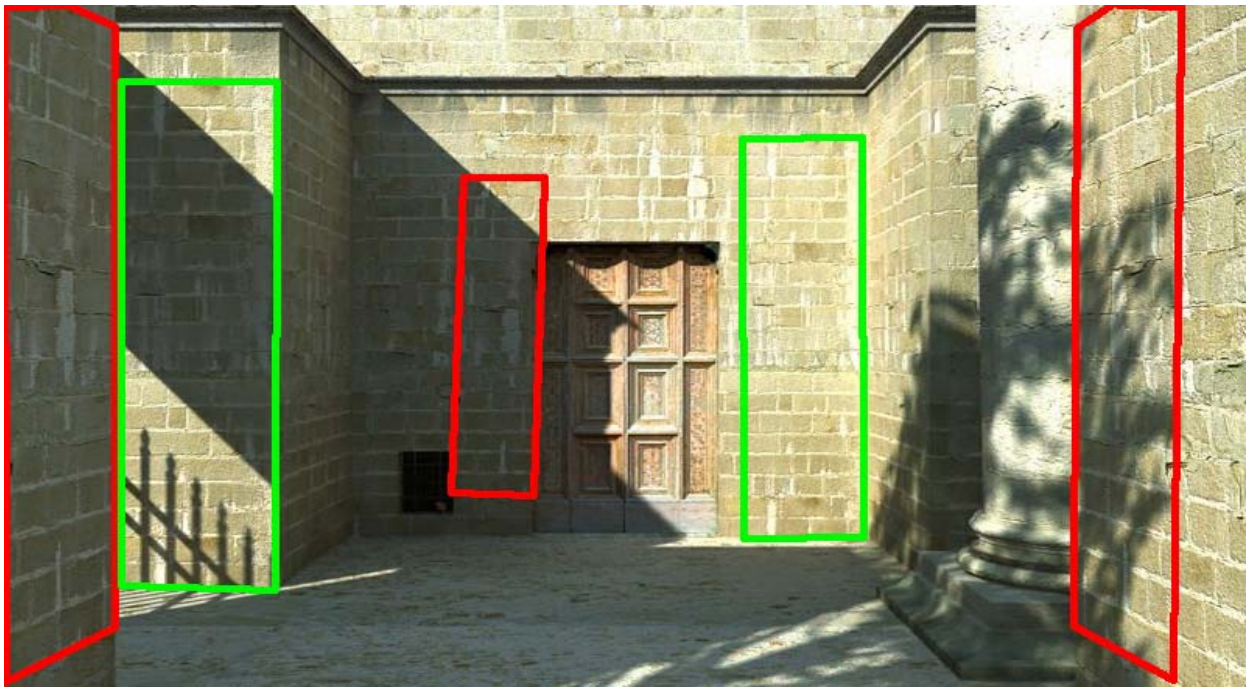
- Colores fríos y sombras suaves

Personalidad

Objetivos de la iluminación

Luz Virtual

Objetivos de la iluminación

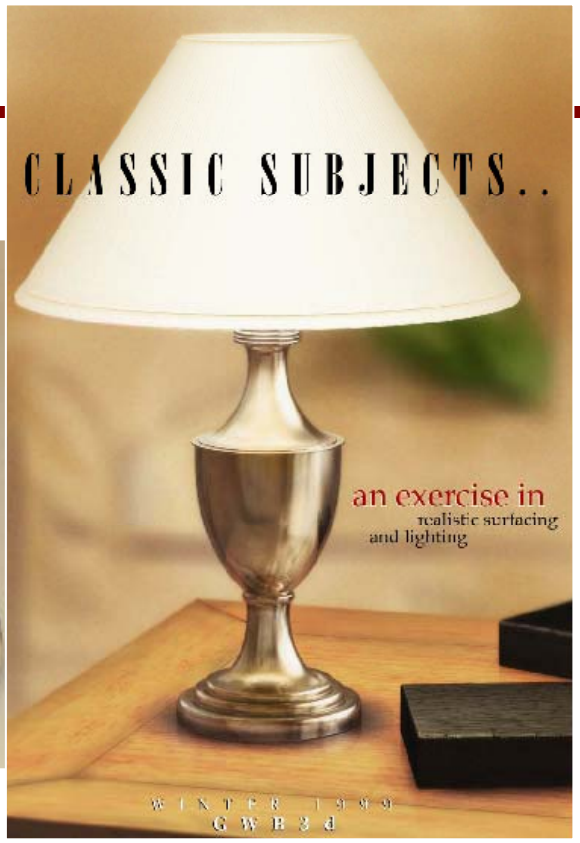


Luz Virtual

Iluminación en CG

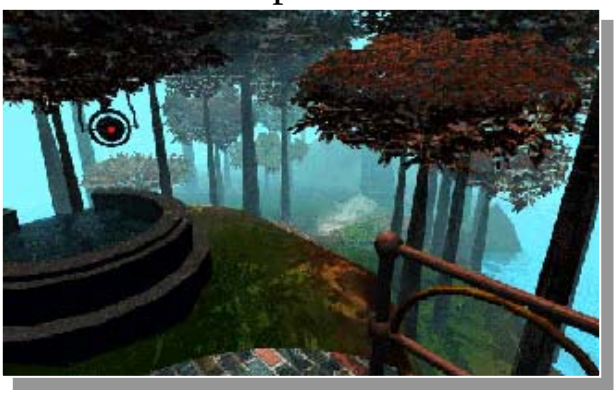
- El mayor problema es que los modelos de iluminación digitales sólo ofrecen una burda y lenta aproximación de la compleja fenomenología de la interacción real de la luz con la materia.
- El objetivo es concentrarse en problemas y sus soluciones, no explicar qué hace cada comando del menú de un programa determinado.
- Es muy fácil echar la culpa a nuestro renderizador, diciendo que no es muy bueno, que no da más de sí...
- Por ejemplo, el rederizador de 3DS Max no es nada del otro mundo...


FIDO


CLASSIC SUBJECTS..
an exercise in
realistic surfacing
and lighting

Iluminación en CG

- Un buen artista debe saber continuar donde su software acaba, explorando caminos que ni los propios programadores del soft habían sospechado.



- (MYST; Strata Vision 3D, Mac)

Creando profundidad

- La plataforma final no es más que una superficie 2D sobre la que se proyecta luz. Cualquier sensación de tridimensionalidad no es más que una ilusión.
- Planos de luz: Conjunto de objetos o personajes paralelos al plano de la cámara, que son iluminados como una unidad, contrastando con el resto de planos.
- Por ejemplo, una escena de interior podría estar formada por cuatro planos de luz: el paisaje visto a través de la ventana, la pared de la habitación, los personajes en la habitación y un objeto (un jarrón...) en primer plano.

Luz Virtual

Creando profundidad

- El jarrón podría ser sólo una silueta oscura; los personajes estar bien iluminados; la pared sumida en semisombra y el paisaje fuertemente iluminado por la luz del sol.
- Podemos añadir más sensación de profundidad variando la temperatura de la luz en cada plano (personajes con luz cálida y luz más fría para la pared).
- Como toque final, podemos desenfocar tanto el jarrón como el paisaje al fondo

Luz Virtual

Creando profundidad

Luz Virtual

Creando profundidad

Luz Virtual

Creando profundidad

Luz Virtual

Creando profundidad

- Volumen y espacio: Podemos acentuar el volumen aparente de los objetos o personajes usando luz lateral o trasera, jugando con las sombras.
- Una luz situada en la dirección de la cámara tiende a reducir la sensación de volumen.
- Conviene no iluminar de manera uniforme las grandes superficies planas.
- Jugando con charcos de luz acentuamos la ilusión de espacio (calle de noche iluminada por farolas). Sin embargo, hay que dar al ojo la posibilidad de conectar esos charcos de luz.

Luz Virtual

Creando profundidad

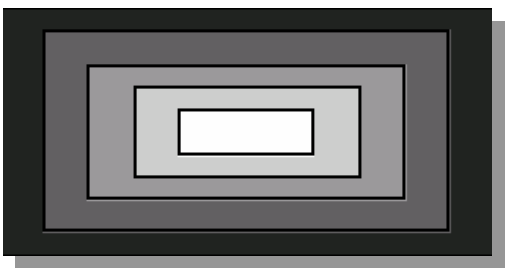
Luz Virtual

Creando profundidad

Luz Virtual

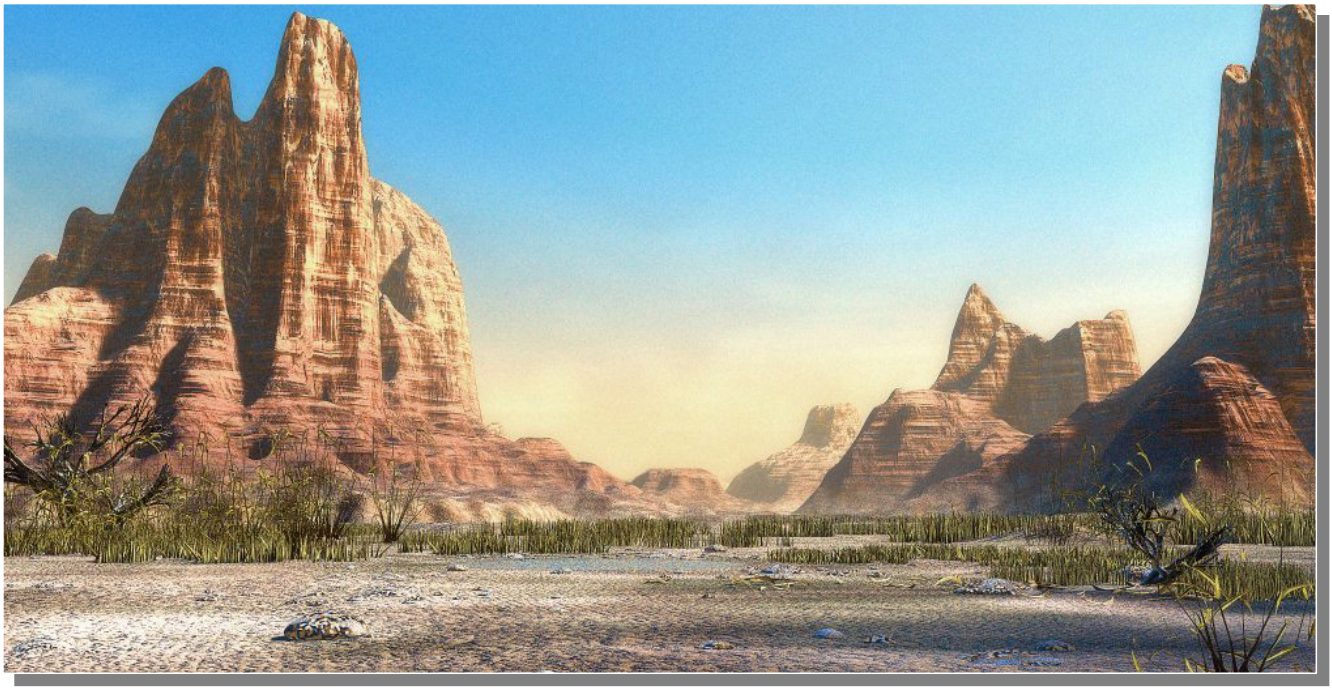
Creando profundidad

- Atmósfera: Menos en días de mucho viento o después de una tormenta, la atmósfera está llena de partículas de polvo y agua en suspensión, que reflejan y refractan luz.
- Este efecto se nota sobre todo con la distancia, donde se reduce el contraste y la saturación de los colores.
- Profundidad con el color: Un objeto cálido sobre un fondo frío añade ilusión de profundidad, igual que un objeto brillante sobre un fondo oscuro.



Luz Virtual

Atmósfera



Luz Virtual

Atmósfera

Atmósfera

Atmósfera

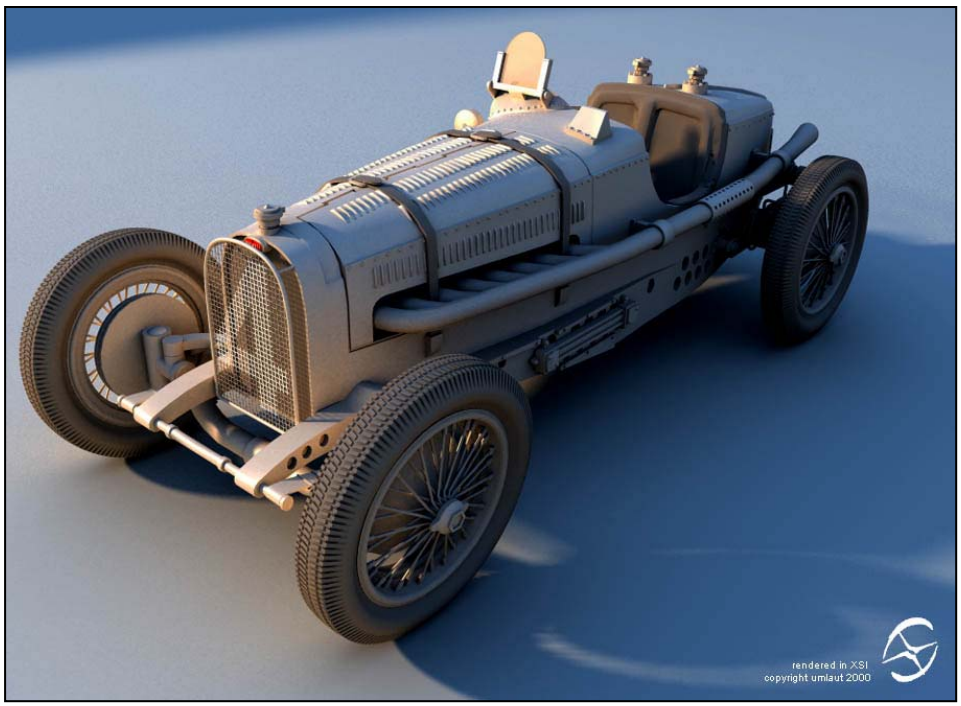
Luz Virtual

¿Qué hora es?

- No hay que ser muy estricto a la hora de simular de manera realista la posición del sol en escenas diurnas. Basta con dar una idea de cuándo se está desarrollando la historia.
- Al amanecer podemos usar tonos azulados, que van saturando a blanco mientras nos acercamos al mediodía. Durante la puesta del sol la luz se vuelve rojiza.
- De manera similar, la primavera conlleva una iluminación más clara y fresca, el verano cálida, para irse enrojeciendo con el otoño y enfriándose en invierno.
- Hay que aprovechar la subjetividad de los colores.

Luz Virtual

¿Qué hora es?



Luz y composición

- La iluminación es parte intrínseca de la composición final. Para comprobar esto, basta con iluminar una escena cualquiera con un sólo foco tipo “spot”, e ir moviéndolo por la escena para ver cómo su lectura cambia sustancialmente.
- Cada escena es única y requiere un análisis individual.
- La mejor forma de comenzar a iluminar una escena nueva es (discutiblemente) resaltar el punto focal, para minimizar luego elementos que interfieran con este punto.

Luz y composición

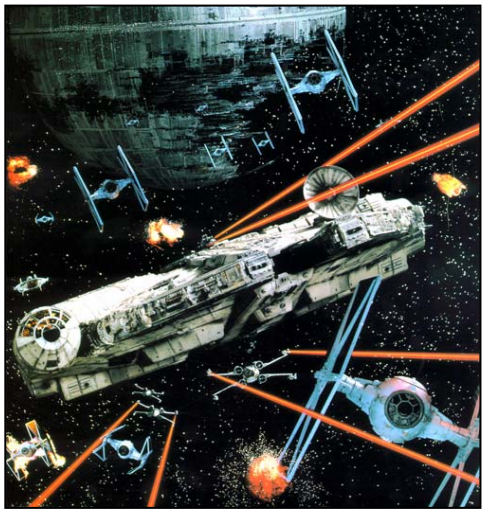
Continuidad

- A todos nos gustaría hacer de cada escena una obra maestra de la iluminación. Pero en una escena de un segundo debemos decirle al ojo dónde mirar de manera inmediata, sin dejarle vagar por la imagen.
- Por otra parte, debemos ser consistentes a lo largo de la animación, y esto suele obligarnos a hacer trampas (Cameron en Titanic...).
- Trucos de edición (cortes, fundidos...) pueden ayudarnos a alterar ligeramente la iluminación de una escena a otra (o de un plano largo a uno medio), sin perder continuidad.
- Hay que previsualizar las escenas en su formato final (VHS, monitor...) para evitar sorpresas al final.

Luz Virtual

El sistema visual humano

- Invariancia: propiedades *intrínsecas* de las escenas u objetos (reflectividad…). Las propiedades *extrínsecas* incluyen las condiciones de iluminación, el punto de vista…
- Constancia: capacidad de extraer condiciones accidentales y extraer los invariantes
- Ejemplos: tamaño y perspectiva o constancia del color (interior de un tren...)



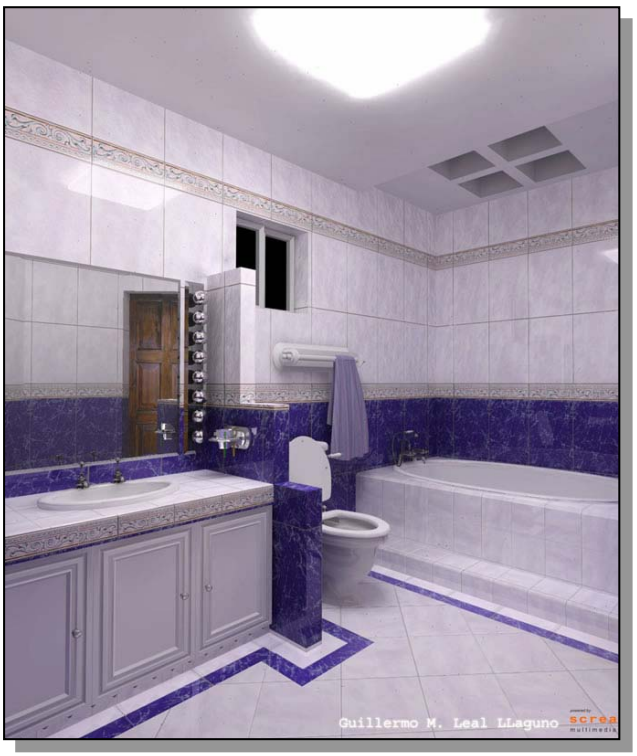
Luz Virtual

Constancia del color

- Fijémonos en la tapa del inodoro

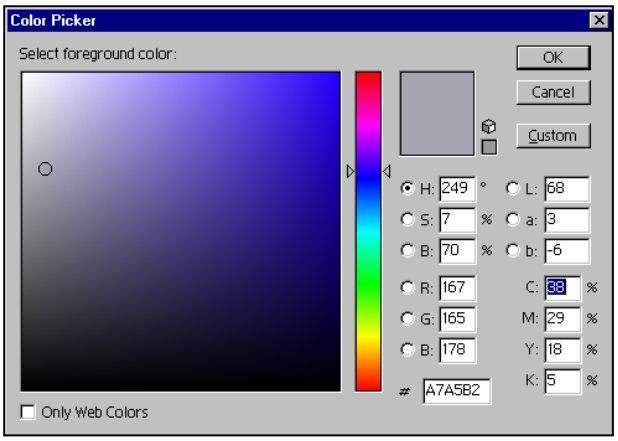
Luz Virtual

Constancia del color



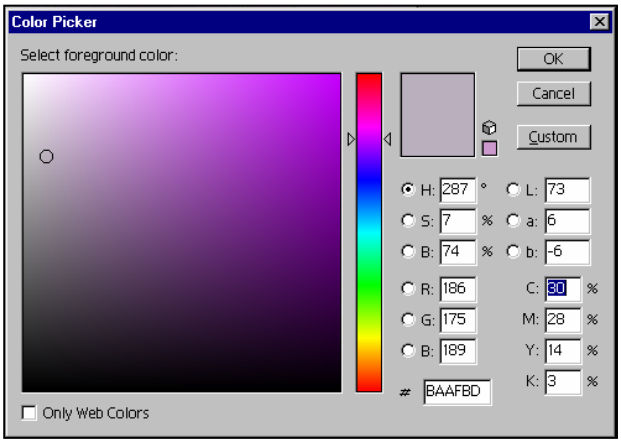
Luz Virtual

Constancia del color



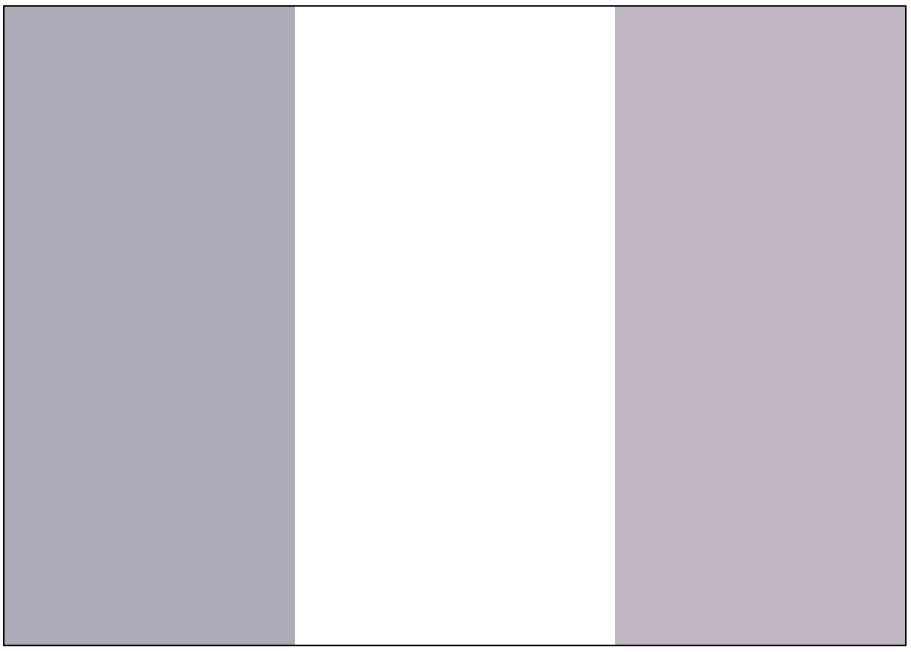
Luz Virtual

Constancia del color



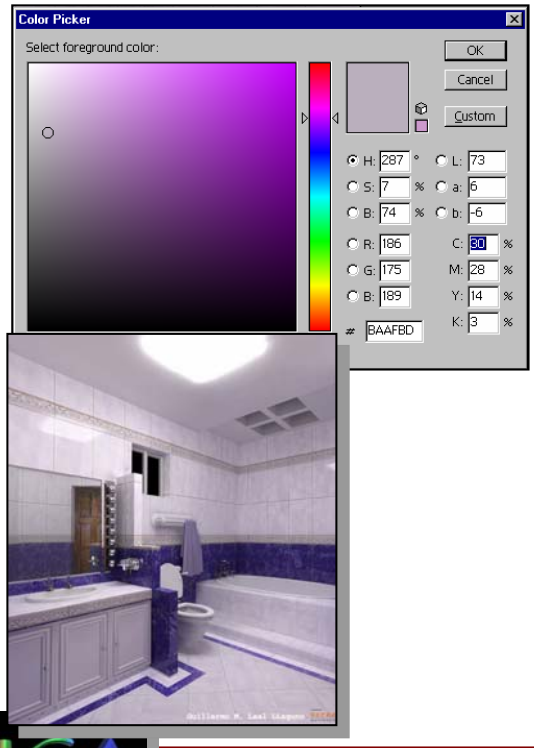
Luz Virtual

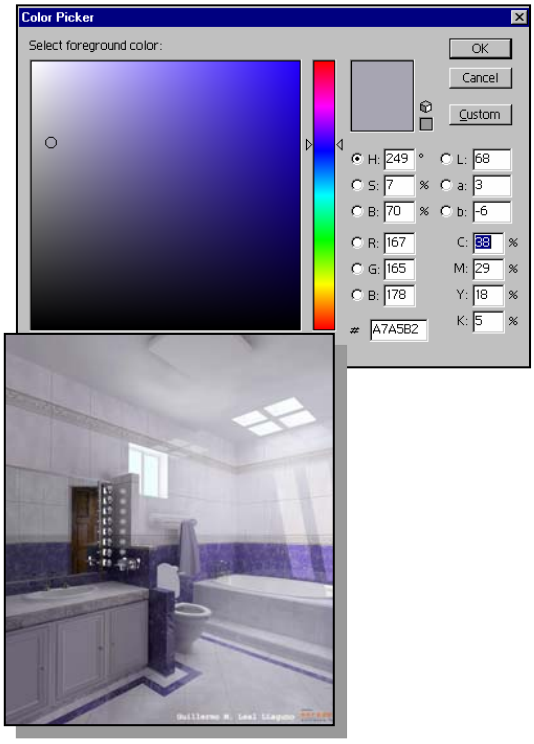
Constancia del color



Luz Virtual

Constancia del color





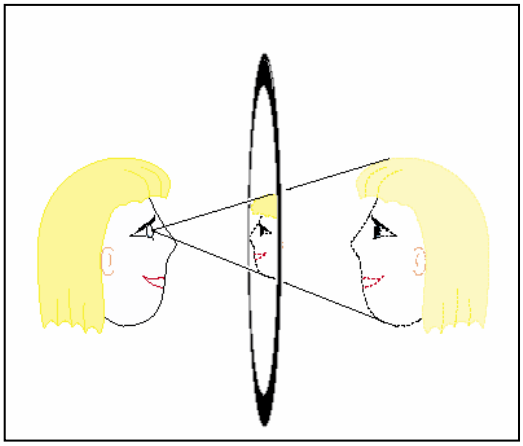
Luz Virtual

Constancia del color

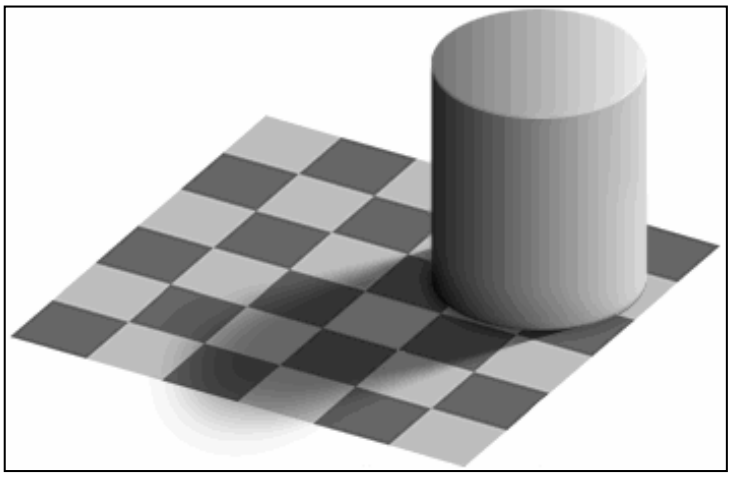
- Es fácil estimar el color intrínseco de un objeto, pero muy difícil el color de la luz que refleja!

Constancia

- Otro ejemplo: nuestra imagen en el espejo es sólo la mitad de grande que nosotros!

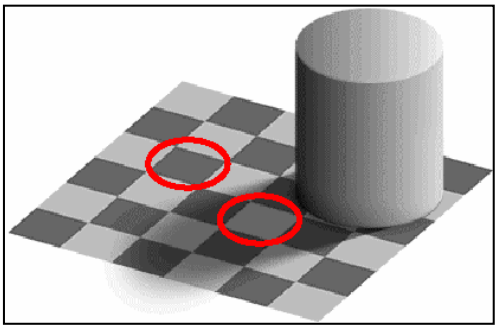


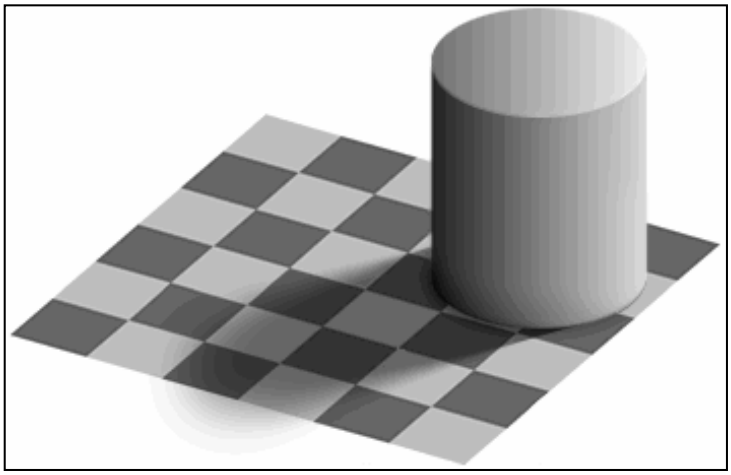
El sistema visual humano



Luz Virtual

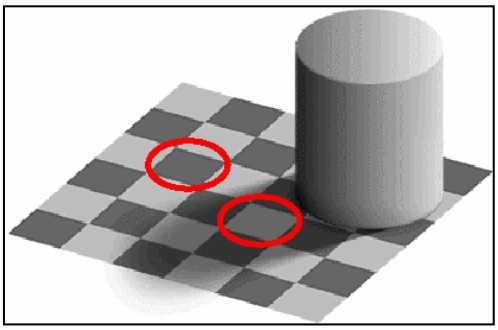
El sistema visual humano

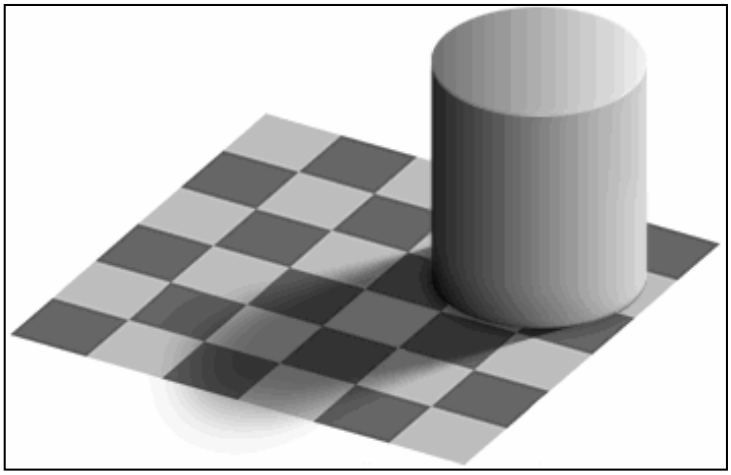


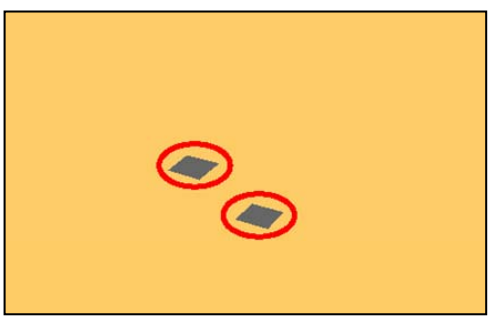


Luz Virtual

El sistema visual humano

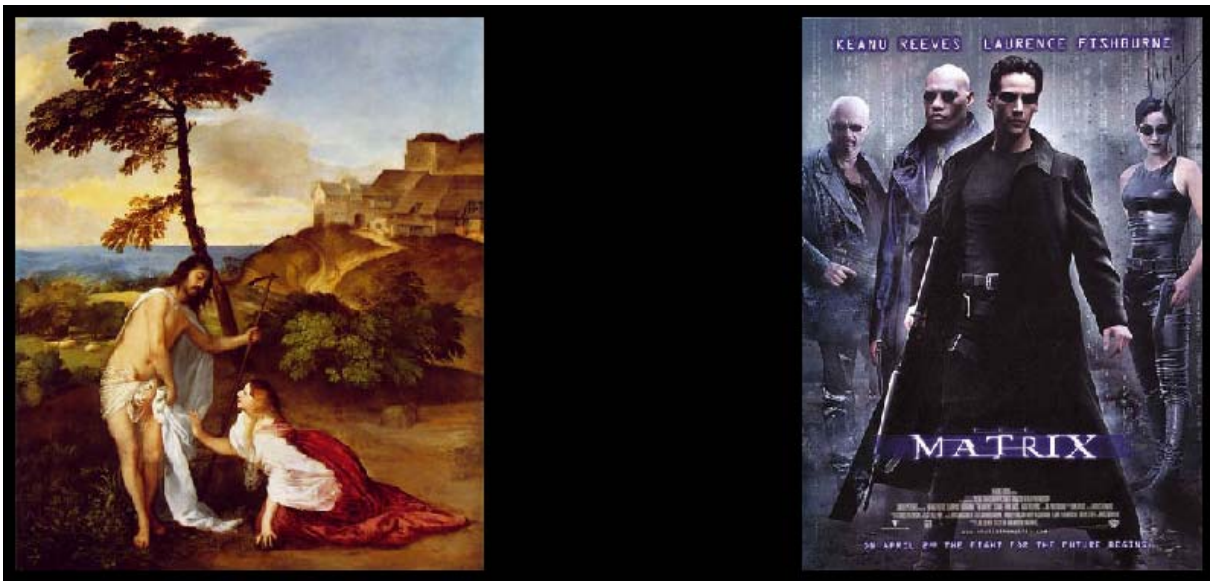






Luz Virtual

El sistema visual humano



- Esta capacidad de detectar contrastes es utilizada constantemente (aumenta el rango dinámico aparente)

Luz Virtual

El sistema visual humano

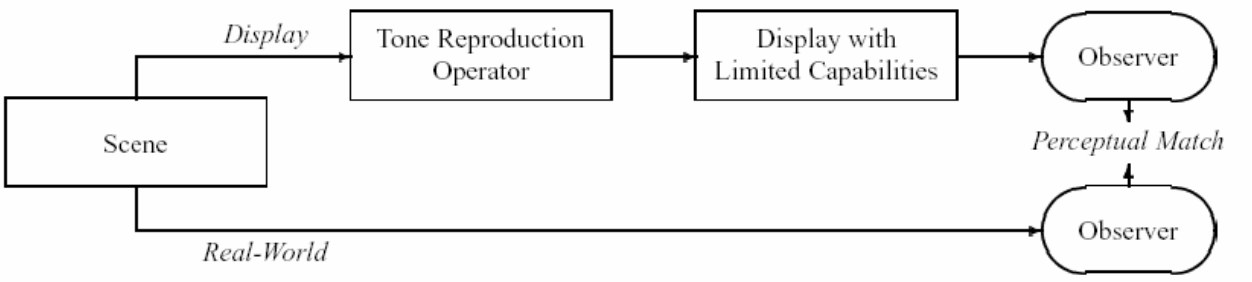
Luz Virtual

El sistema visual humano

- Nuestra percepción de contrastes da lugar a muchos otros fenómenos curiosos…

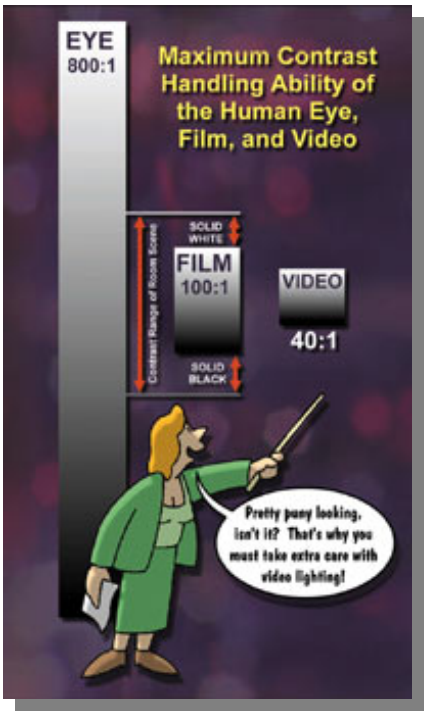
Luz Virtual

Rango dinámico



Luz Virtual

Rango dinámico



Luz Virtual

El aspecto subjetivo

Luz Virtual

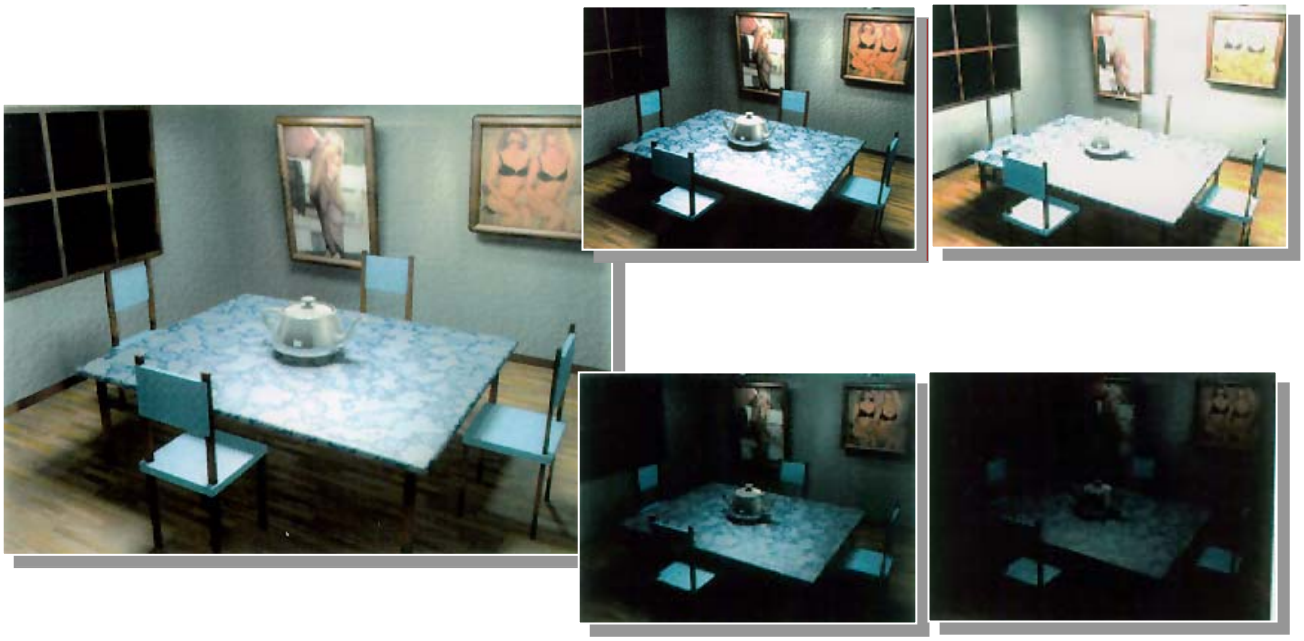
El aspecto subjetivo

Luz Virtual

El aspecto subjetivo

Luz Virtual

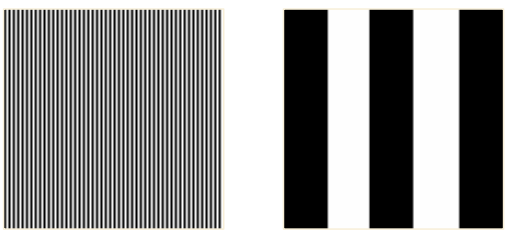
El aspecto subjetivo

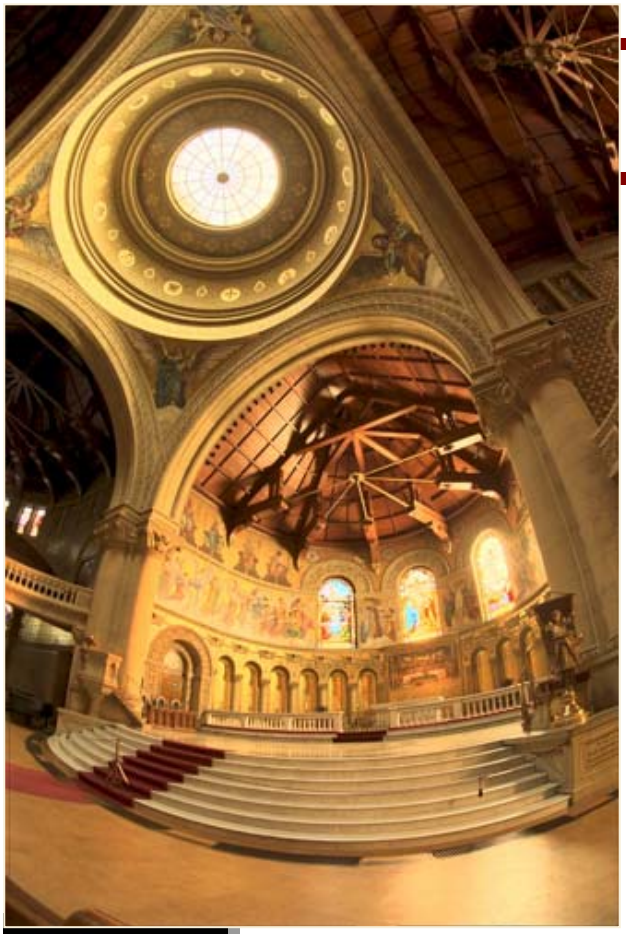


Luz Virtual

El sistema visual humano

Imágenes basadas en la percepción: Sin limitarnos a los cálculos físicos de la interacción de la luz con la materia





Luz Virtual

El sistema visual humano

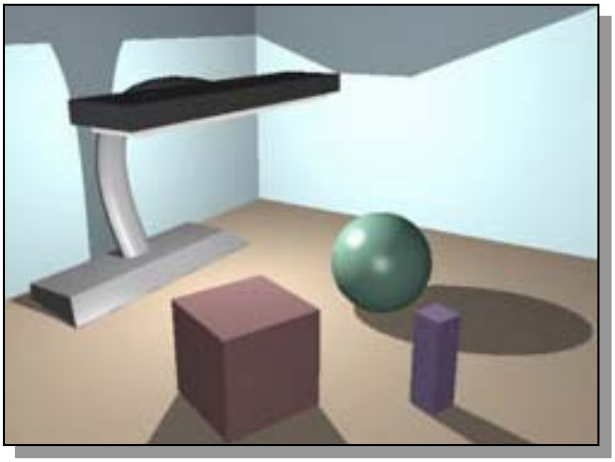
Luz Virtual

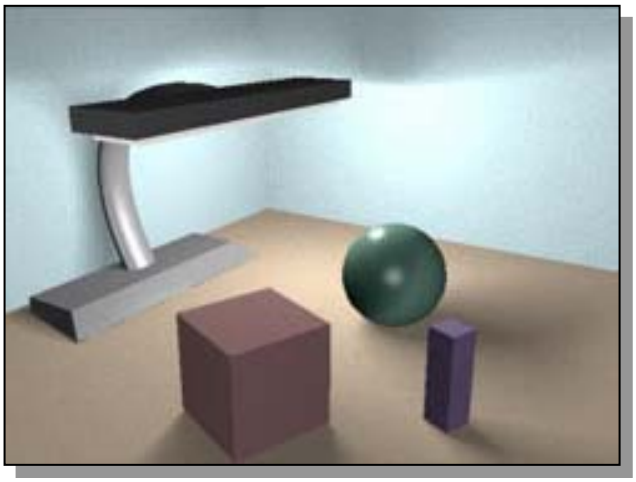
Luces motivadas o no motivadas

- Una luz motivada es aquella cuya fuente aparece en la escena o es fácilmente deducible de ella (una lámpara en una habitación, el sol...). Suele arrojar sombras.
- Una luz no motivada es aquella sin fuente aparente. Suelen aplicarse para resaltar detalles o ensalzar el ambiente de la escena. Debido a su no motivación, no arrojan sombras ya que éstas delatarían su origen.
- "Empiezan donde el renderizador acaba".
- El truco es integrar ambos tipos en la escena, logrando transmitir el efecto deseado sin que el espectador detecte la presencia de luces no motivadas, lo cual rompería toda la ilusión generada.

Luz Virtual

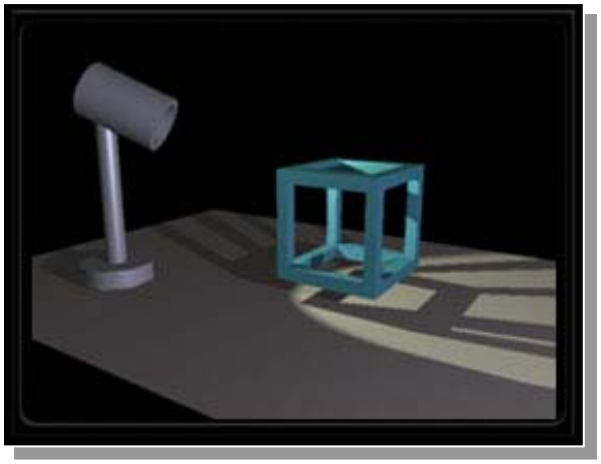
Luces puntuales o de área

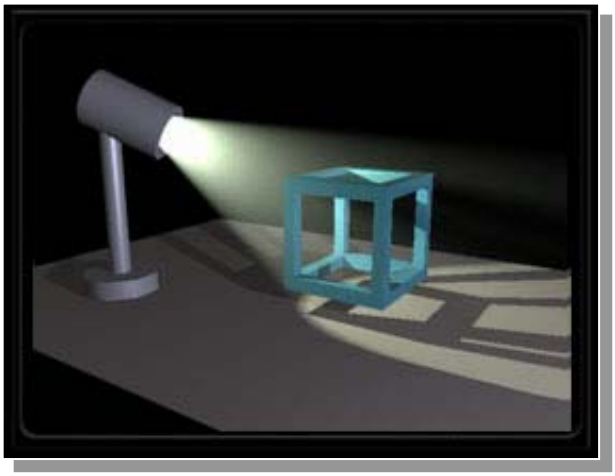




Luz Virtual

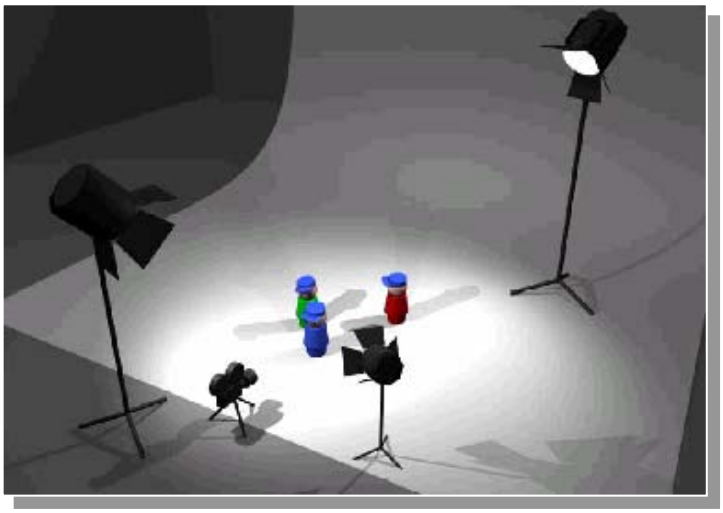
Luces volumétricas





Configuración básica

- Un buen comienzo es empezar con la configuración de tres luces (3-light setup): luz principal (key light), luz de relleno (fill light) y luz posterior (rim light).



Luz principal

- Luz principal: Suele ser motivada, y por tanto arroja sombras. De hecho, las sombras de la luz principal ayudan a esculpir la escena tanto o más que la propia luz.
- Suele situarse frontalmente, a un lado de la cámara, más o menos elevada según el tipo de sombras buscado. También pueden ponerse en un lateral o incluso detrás (side key, rear key).
- La posición básica es una “3/4”: aproximadamente tres cuartas partes del personaje están iluminadas por ella. Esta es la posición que arroja de entrada sombras más agradables.

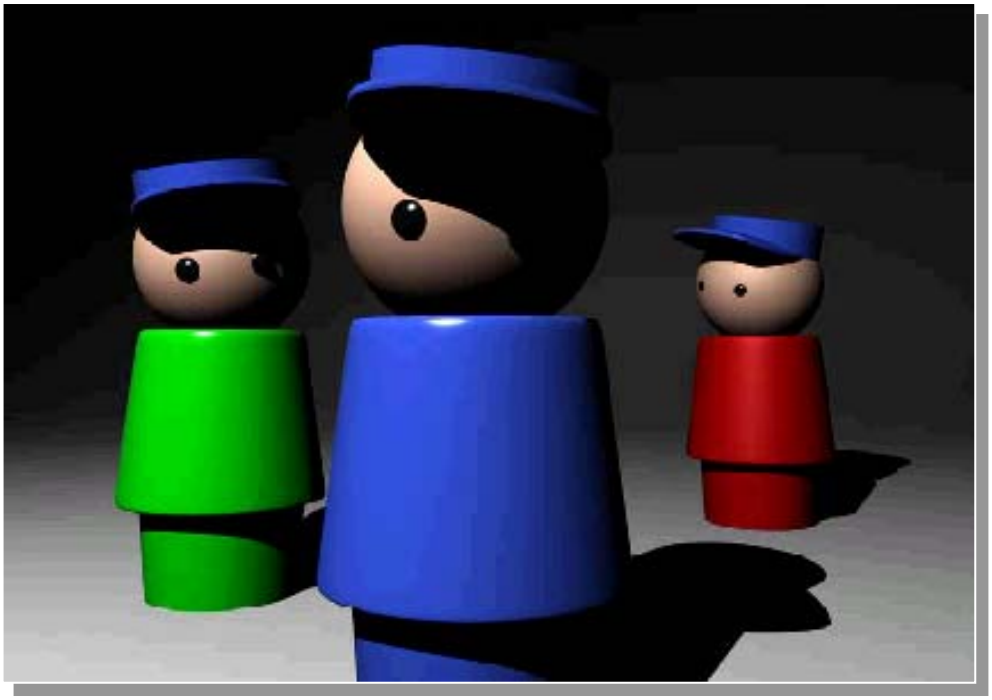
Luz Virtual

Luz principal

- Suele ser la primera luz en el proceso de iluminación. Las otras dependen de ella.
- Un error muy cometido por principiantes es el de situar la luz principal justo donde está la cámara, aplanando lamentablemente la imagen. Hay que dejar que las sombras añadan detalle al modelo.
- Las sombras que arroja pueden ser definidas o difusas, según el origen de la luz.

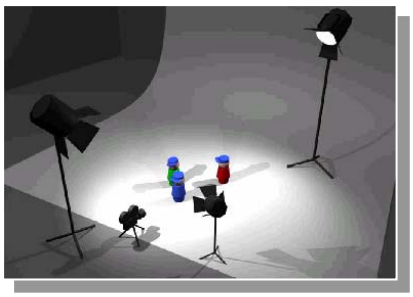
Luz Virtual

Luz principal



Luz de relleno

- La luz de relleno suaviza y rellena las sombras creadas por la luz principal. Tal y como se trata en el lenguaje cinematográfico, es diferente de la luz ambiental.
- Suele ser no motivada, sin sombras.
- Se sitúa de manera que cubra todas las sombras que deja la luz principal, al menos las visibles en la escena (entre 45º y 180º al otro lado de la cámara).
- Altura antagonista con respecto a la luz principal.
- Su intensidad parte de 1/2 de la principal

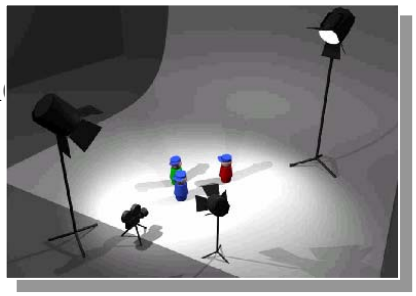


Luz Virtual

Luz de relleno

Luz posterior

- La luz posterior resalta el contorno del personaje, separándolo del fondo (también separa la vida real de Hollywood...).
- Es una de las “aberraciones” más aceptadas en el lenguaje cinematográfico (Men In Black...), dotando a la imagen de un cierto aspecto estilizado.
- Suele situarse detrás y por encima del personaje, diametralmente opuesta a una luz principal de 3/4.
- Su origen estaba motivado por las películas antiguas en blanco y negro, pero sobrevivi a la llegada del color.



Luz Virtual

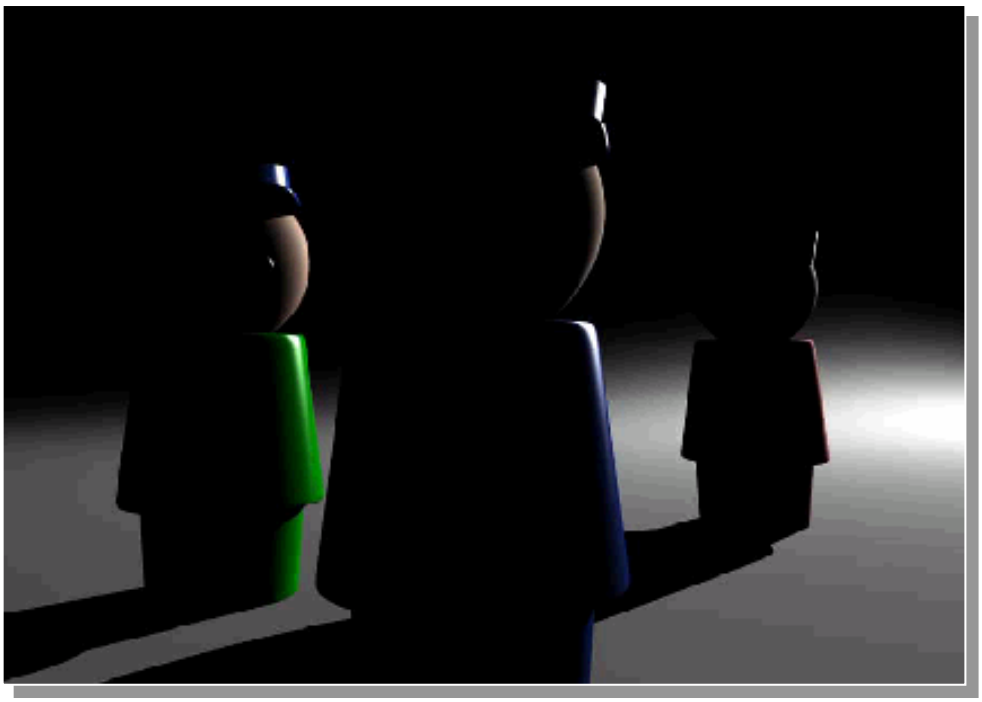
Luz posterior

Luz posterior

- Aquí la luz posterior ayuda claramente a perfilar el contorno del conejo en aquellas zonas demasiado oscuras, permitiendo leer la imagen rápidamente al identificar al conejo de manera instantánea.

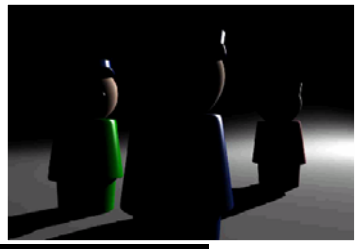
Luz Virtual

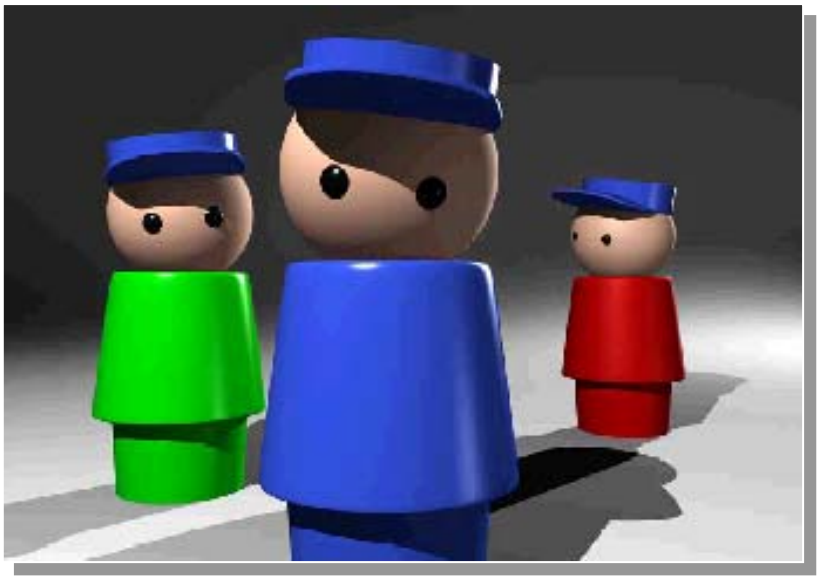
Luz posterior



Luz Virtual

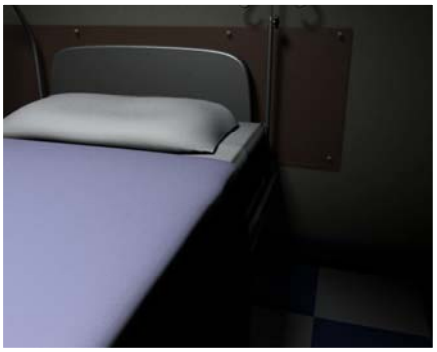
Principal + relleno + posterior





Luz Virtual

Un ejemplo



Luz Virtual

Profundidad a través del color

- Una buena composición generalmente juega con los colores de cada fuente de luz

Luz Virtual

Radiosidad a través del color

- Así integramos el modelo 3D en el ambiente (real o sintético) a través de una simulación de la interreflexión difusa de la luz.

Luz Virtual

Radiosidad a través del color

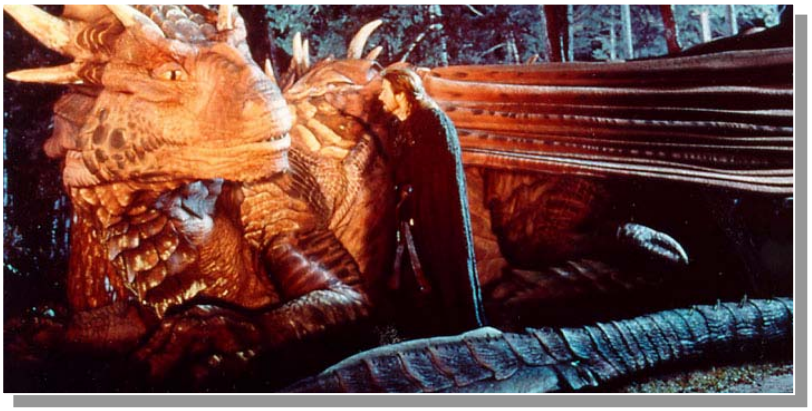
Luz Virtual

Hablando de integración...

- Unas bolas difusas y reflectantes capturan la iluminación del escenario

Hablando de integración...

- La integración de un modelo CG con una escena real SIEMPRE conlleva una corrección del color en post.



El papel de las luces

- Las tres luces estudiadas pueden intercambiarse los papeles.
- Por ejemplo, en una escena nocturna es fácil tener la luz posterior como fuente principal de luz. Esto ayuda a mantener un tono oscuro, sin afectar ni la legibilidad ni la ilusión de nocturnidad, ni la sensación tridimensional de la escena. La luz principal, drásticamente reducida, pasaría a ser la luz de relleno.
- La motivación de las luces es más fuerte que las reglas de la configuración básica.

Luz Virtual

El papel de las luces

Luz Virtual

El papel de las luces

Luz Virtual

Luz ocular y kicker

- La luz ocular se utiliza mucho en cinematografía tradicional. Los ojos de los actores cobran una mayor expresividad si pueden verse reflejos especulares en ellos.
- El kicker es una luz, justificada o no, que sutilmente destaca al personaje u objeto principal del resto (Adam´s Family, Antz, el anuncio de Carlsberg...).
- El kicker, cuando es necesario, suele ir reforzado por otros elementos (una chaqueta o vestido rojos en medio de una multitud permiten localizar enseguida al personaje principal...)

Luz Virtual

Luz ocular y kicker

- Luz ocular

Luz ocular y kicker

- Kicker

Luces prácticas

- Las luces prácticas están en la escena porque la historia lo requiere. La iluminación puede basarse en éstas, o permitir que la sobrepasen, si ayuda a transmitir la idea (en Encuentros en la Tercera Fase, los destellos de la nave nodriza deslumbran completamente al espectador...)
- En cinematografía digital, muchas veces es necesario simularlas con otra luz más potente fuera de encuadre (foco simulando la luz de una vela en una película de terror...). En cinematografía digital esto no es necesario, podemos actuar directamente sobre la propia luz práctica.

Luz ambiental

- Viene a suplantar el fenómeno de interreflexión difusa de la luz con la atmósfera en una escena real. La mayoría de los renderizadores comerciales están basados en algoritmos de trazado de rayos, e ignoran este fenómeno. Los renderizadores basados en radiosidad se acercan algo más al proceso real, por lo que necesitan menos luz de relleno.
- Si utilizamos la luz ambiental por defecto de los paquetes de animación 3D, el resultado es bastante pobre. La imagen se aplana. No conviene abusar de ella, o será imposible quitar a nuestra escena el look característico de las imágenes generadas por ordenador.

Luz ambiental

- Interreflexión difusa:

Luz Virtual

Luz ambiental

- ¿Qué sucede cuando, en una habitación del mundo real, encendemos la luz? ¿Qué sucede en un mundo análogo digital?
- En cinematografía tradicional, se utilizan grandes lonas blancas para aumentar la luz ambiental, o negras para absorberla.
- El truco es saber equilibrar este tipo de iluminación, difícilmente controlable.
- La regla de oro que nunca hay que olvidar es que cuanta más luz ambiental hay en una escena, menos contraste obtenemos.

Luz Virtual

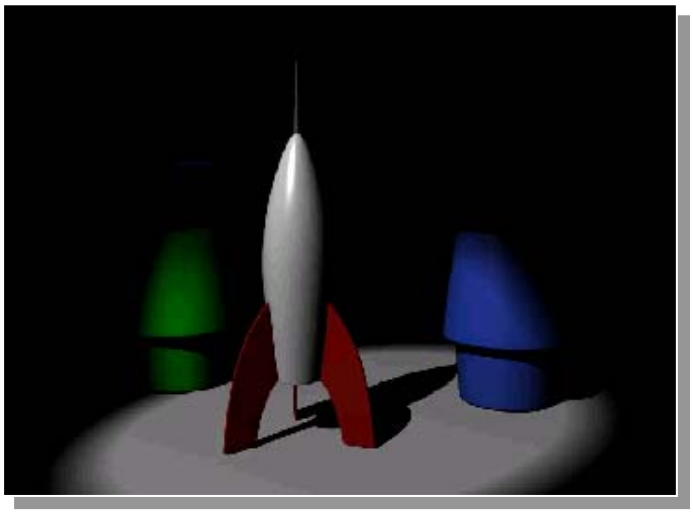
Luces locales y otros trucos

- Pero donde de verdad nos desmarcamos de los cinematógrafos tradicionales es en herramientas tales como:
  - La posibilidad de aislar el efecto de cada luz (incluyendo o excluyendo).
  - La creación de focos de luz negativa.
  - La posibilidad de definir luces que no arrojen sombras.
  - El hecho de que nuestros focos de luz no existen como objetos en el mundo creado, y no se renderizan
- A cambio, debemos apañarnos con modelos de iluminación limitados.

Luz Virtual

Simulando luz ambiental (1)

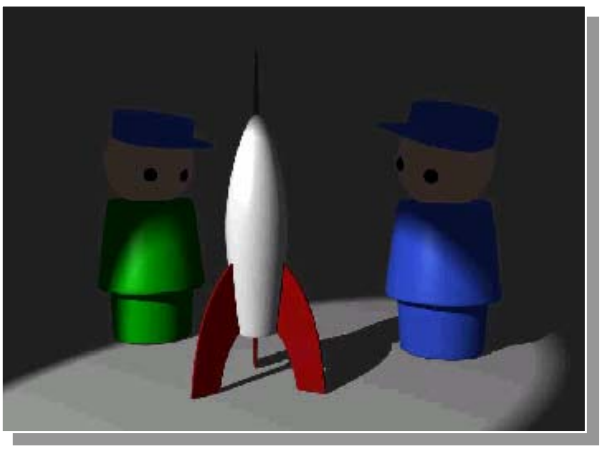
- Supongamos un caso de iluminación extrema: un sólo foco.



Luz Virtual

Simulando luz ambiental (1)

- Para simular la luz rebotada del suelo, aumentamos la luz ambiente...

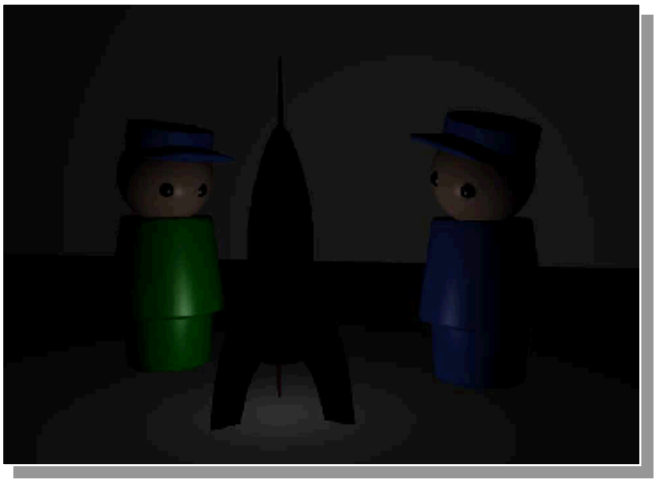


- El resultado: hemos aplanado la imagen. Esto sucede SIEMPRE que usemos la luz ambiental de cualquier software

Luz Virtual

Simulando luz ambiental (1)

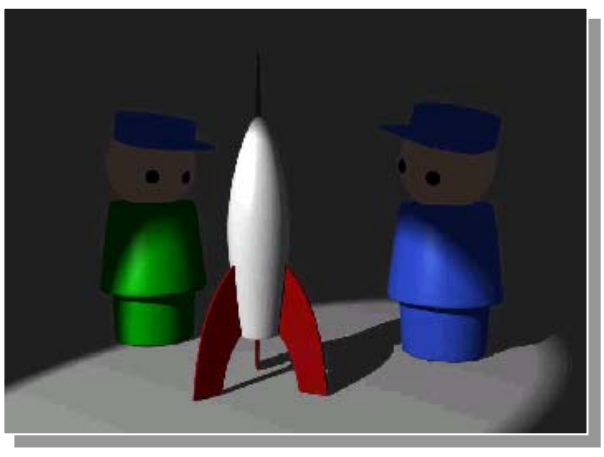
- Si en lugar de usar la luz ambiental (más mala que el tabaco!!), nos complicamos la vida un poco más, podemos colocar una suave luz puntual en el suelo que simule la luz rebotada en él...

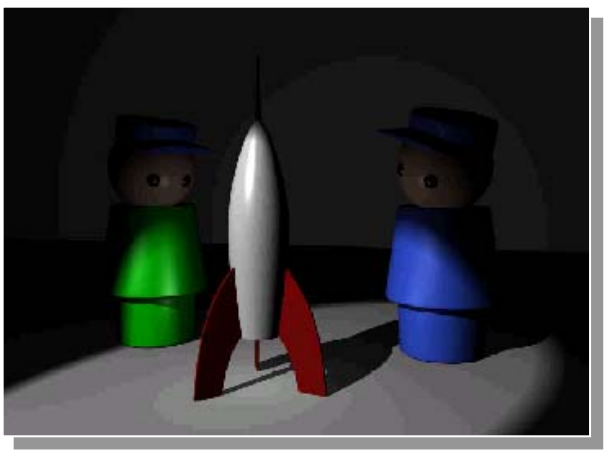


Luz Virtual

Simulando luz ambiental (1)

- El resultado es una imagen con mucho más volumen que la anterior:

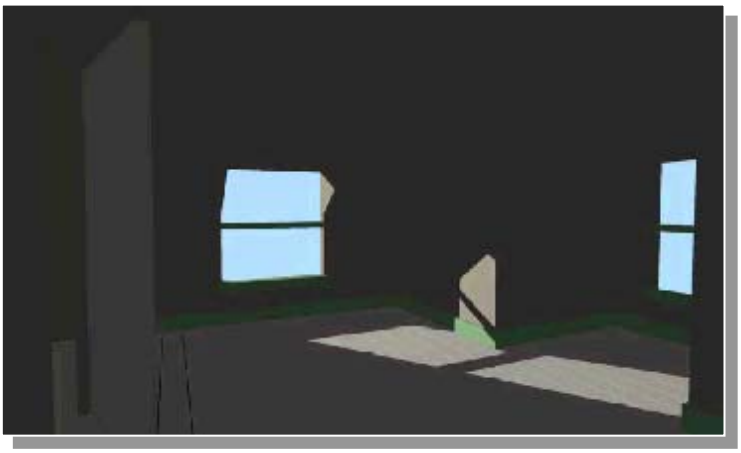




Luz Virtual

Simulando luz ambiental (2)

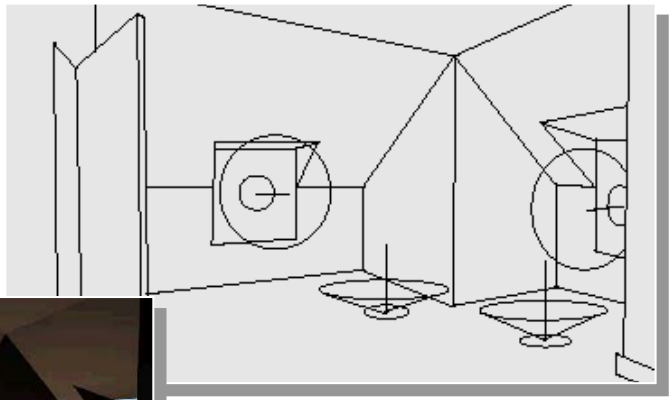
- Otro ejemplo: una habitación soleada. Pensamos “hummm, la luz rebota por todos los lados y satura la habitación de luz difusa. ¡Ya está!. Le meto un montón de luz ambiental y arreglados”.
- El resultado no es demasiado espectacular que digamos...
- Y eso que usamos uno de los mejores programas que existen (Softimage | 3D)

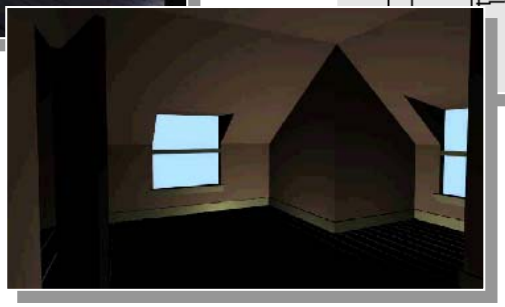


Luz Virtual

Simulando luz ambiental (2)

- Para simularla, situamos un tenue foco azulado en cada ventana apuntando al suelo, y dos más tenues y cálidos desde el suelo a las paredes y el techo:





Luz Virtual

Simulando luz ambiental (2)

- De nuevo, el resultado es una imagen mucho más atractiva que la anterior:

Luz Virtual

Amortiguación

- Cualquier renderizador que se precie nos permitirá definir una velocidad de amortiguación de las luces, que simule su comportamiento real.
- Puede simularse, a la vez que ahorramos tiempo de rendering, excluyendo objetos lejanos de la influencia de una luz dada.
- ¿Utilizaríamos amortiguación para las luces de la escena de la habitación soleada?

El aspecto subjetivo

- La forma en que el ojo capta la información de la luminancia de una misma escena varía según la situación.
- La misma medida física de luminancia en una habitación es percibida de forma diferente si el ojo que la percibe ha estado sumido mucho tiempo en la oscuridad o si, por el contrario, acaba de quedarse medio ciego mirando al sol.
- Los renderizadores comerciales no tienen en cuenta este efecto subjetivo, por lo que, si la narración de la historia lo requiere, habrá que simularlo sin más que exagerar o disminuir la intensidad de las luces.

Bibliografía

(Blinn, 1978) J. Blinn, Simulations of Wrinkled Surfaces. Computer Graphics 12(3) August 1978, 286-292.

(Blinn and Newell, 1976) J. Blinn and M. Newell, Texture and Reflection in Computer Generated Images. Communications of the ACM 19(10) October 1976, 542-546.

(Catmull, 1974) E. Catmull, A Subdivision Algorithm for Computer Display of Curved Surfaces. PhD thesis, Department of Computer Science, University of Utah, December 1974.

(Catmull, 1978) E. Catmull, A Hidden-Surface Algorithm with Anti-Aliasing. Computer Graphics 12 (3) August 1978, 6-10.

(Cook, 1984) R. Cook, Shade Trees. Computer Graphics 18 (3) July 1984 223-231.

(Crow, 1981) F. Crow, A Comparison of Antialiasing Techniques. IEEE Computer Graphics and Applications. 1 (1) January 1981, 40-49.

(Peachey, 1985) D. Peachey, Solid Texturing of Complex Surfaces. Computer Graphics 19 (3) July 1985, 279-286.

(Perlin, 1985) K. Perlin, An Image Synthesizer. Computer Graphics 19 (3) July 1985, 287-296.

(Williams, 1983) L. Williams, Pyradmial Parametrics. Computer Graphics 17 (3) July 1983, 1-11.

D. Ebert, F. K. Musgrave, D. Peachey, K. Perlin and S. Worley, Texturing and Modeling: A Procedural Approach. Academic Press, 1994.

J. Foley, A. vanDam, S. Feiner and J. Hughes, Computer Graphics: Principles and Practice. 2nd. ed. Addison-Wesley, 1990.

A. Watt and M. Watt, Advanced Animation and Rendering Techniques: Theory and Practice. Addison-Wesley, 1992.

Diego Gutiérrez
Francisco José Serón

Informática Gráfica

Texturado, iluminación y render de imágenes sintéticas

Grupo de Informática Gráfica Avanzada
Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior
Universidad de Zaragoza