

# AUMENTO DE REALISMO EN LA VISUALIZACIÓN

José García Pareja  
Introducción a la Informática Gráfica

# Índice de contenidos

- Introducción
- Determinación de superficies visibles
- Iluminación local
- Iluminación global
- Más realismo
- GPU

# Introducción

- La síntesis de imágenes por ordenador tiene como objetivo alcanzable actualmente la producción de una imagen por ordenador que sea distinguible de una fotografía de dicho entorno real.
- Los ámbitos en los que se precisan imágenes realistas son muy variados: simulación, entretenimiento, publicidad, investigación, educación, arquitectura, diseño en general, etc.

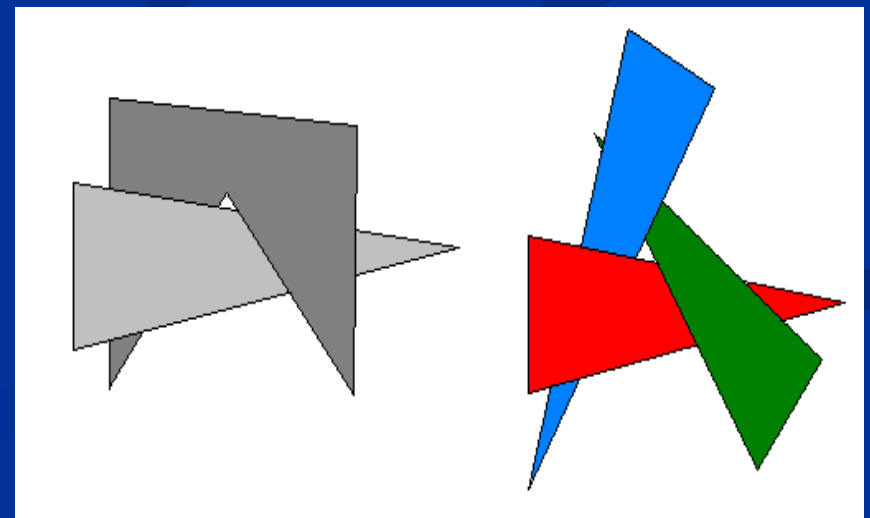
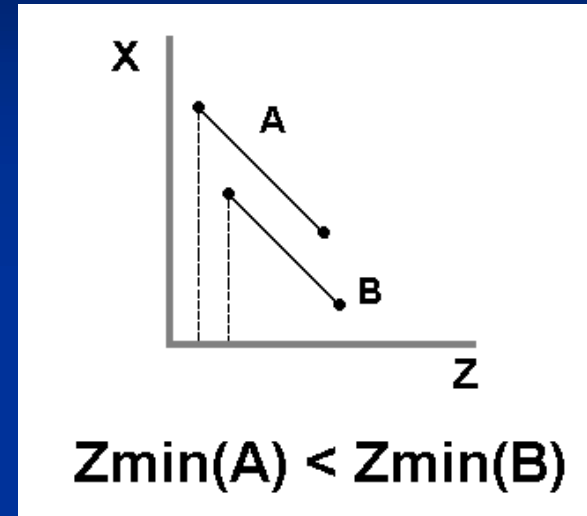
# Determinación de superficies visibles

- Determina que partes de un objeto son visibles desde el punto de observación.
- Su aplicación se centra en:
  - Espacio de la imagen
  - Espacio del objeto
- Optimización de las operaciones mediante: coherencia, transformación en perspectiva, extensiones y volúmenes acotantes, eliminación de caras posteriores, partición espacial, jerarquía.

# Métodos de lista de prioridad

## Algoritmo del pintor

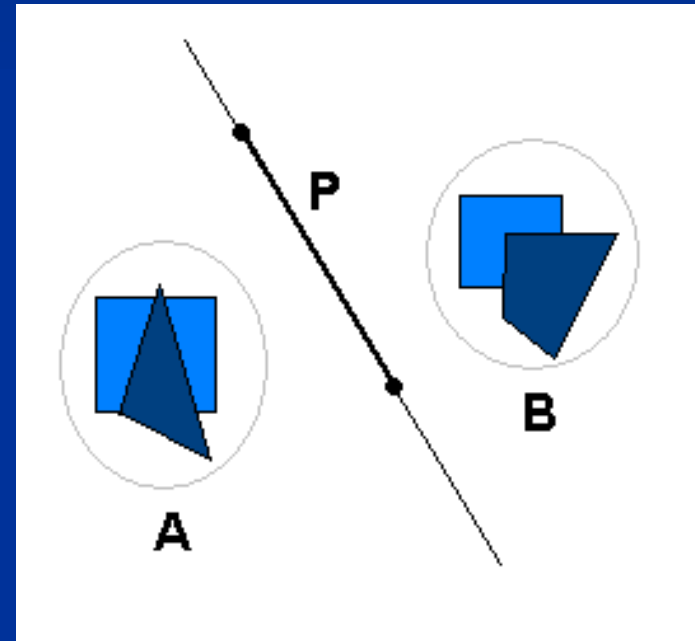
- La idea es que al pintar los objetos en el orden establecido los más alejados queden ocultos por los más cercanos al punto de observación.
- La ordenación se puede hacer según la coordenada  $z$  más lejana al observador de cada polígono.



# Métodos de prioridad de lista

## Algoritmo BSP

- Su nombre deriva de su funcionamiento: escoge un polígono y toma como hijos los polígonos que estén a un lado y otro, respecto de este.
- Su ventaja es que con este preordenamiento es posible generar la imagen desde distintos puntos de vista.



# Z-Buffer

## Algoritmo del Z-Buffer

- Fácil de implementar tanto en hardware como en software.
- Requiere un array bidimensional con una entrada para cada pixel.
- En el array se almacena la profundidad en z del objeto.
- Los polígonos aparecen discretizados en el orden en el que se procesan.

# Z-Buffer

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

+

5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	
5	5	5	5	5	5		
5	5	5	5	5			
5	5	5	5				
5	5	5					
5	5						
5							
5							

=

5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	0
5	5	5	5	5	5	0	0
5	5	5	5	5	0	0	0
5	5	5	5	0	0	0	0
5	5	5	0	0	0	0	0
5	5	0	0	0	0	0	0
5	0	0	0	0	0	0	0

5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	
5	5	5	5	5	5		
5	5	5	5	5			
5	5	5	5				
5	5	5					
5	5						
5							
5							

+

3							
4	3						
5	4	3					
6	5	4	3				
7	6	5	4	3			
8	7	6	5	4	3		

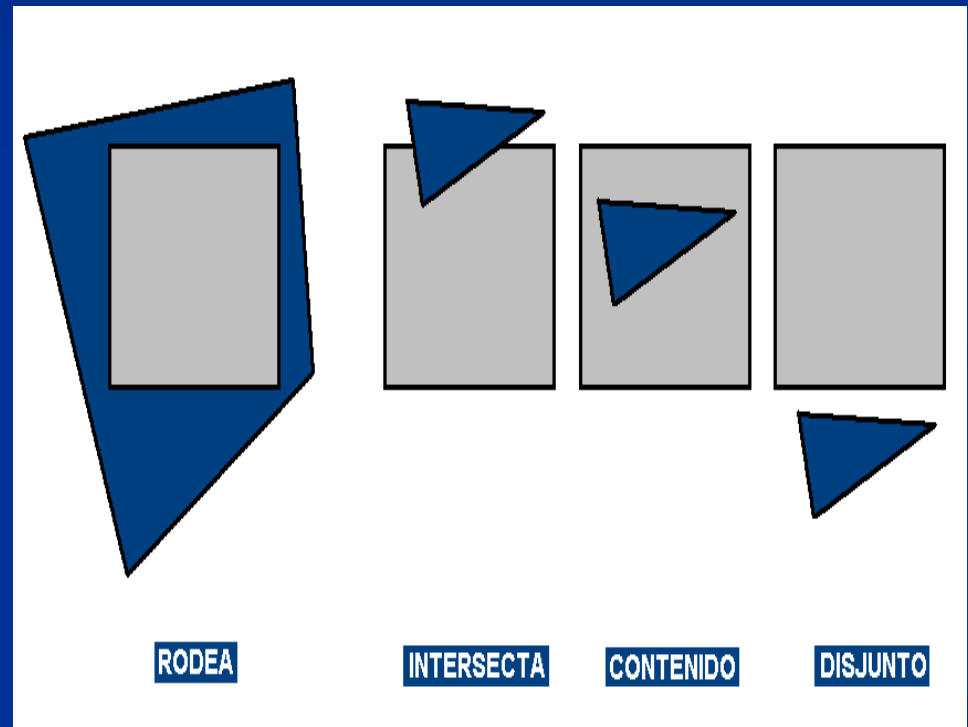
=

5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	0
5	5	5	5	5	5	0	0
5	5	5	5	5	0	0	0
6	5	5	3	0	0	0	0
7	6	5	4	3	0	0	0
8	7	6	5	4	3	0	0
5	0	0	0	0	0	0	0

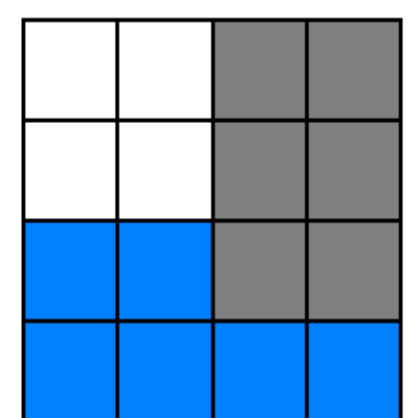
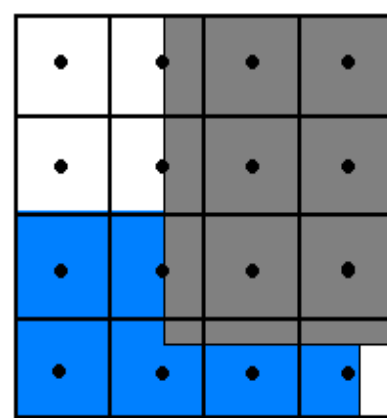
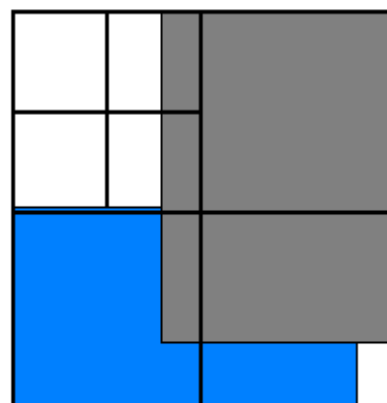
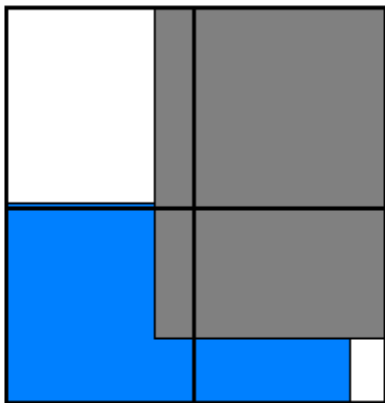
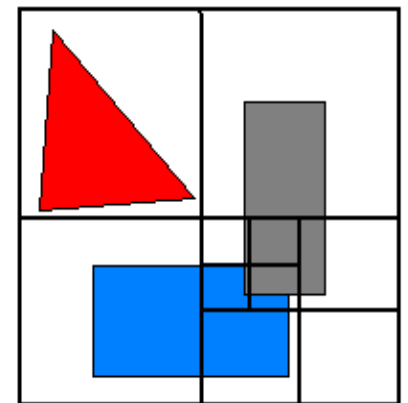
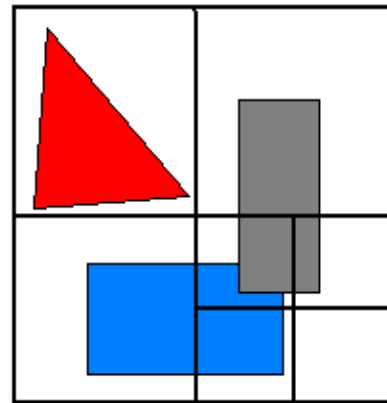
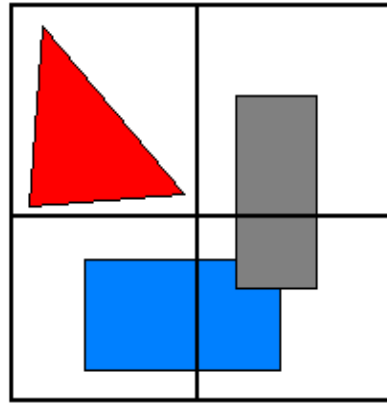
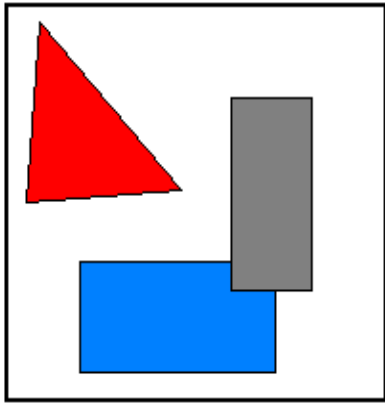
# Métodos subdivisión de la imagen

## Algoritmo de Warnok

- Subdivide el espacio de la imagen para hallar el color de un pixel determinado.
- En cada momento el algoritmo comprueba en que estado está el objeto respecto al área de estudio.



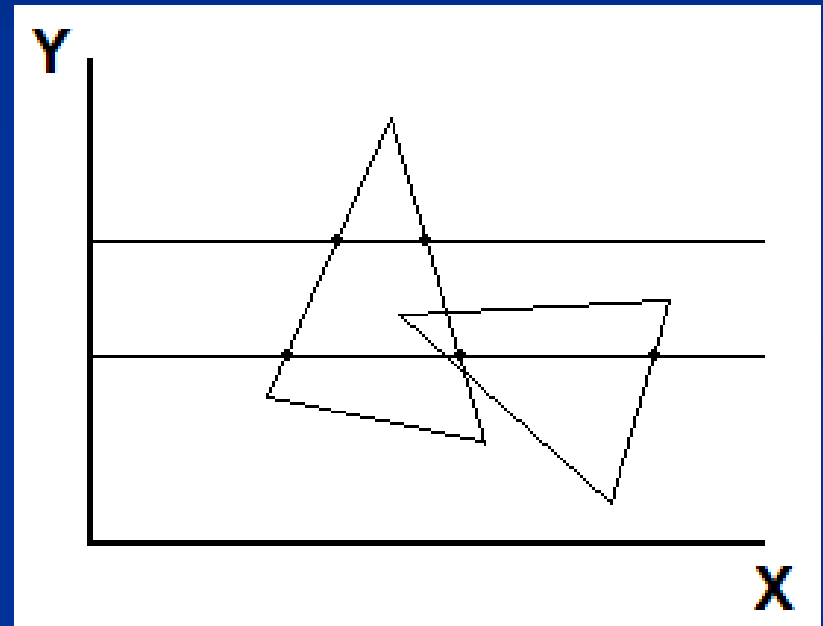
# Warnok



# Métodos de líneas de barrido

## Scan-line

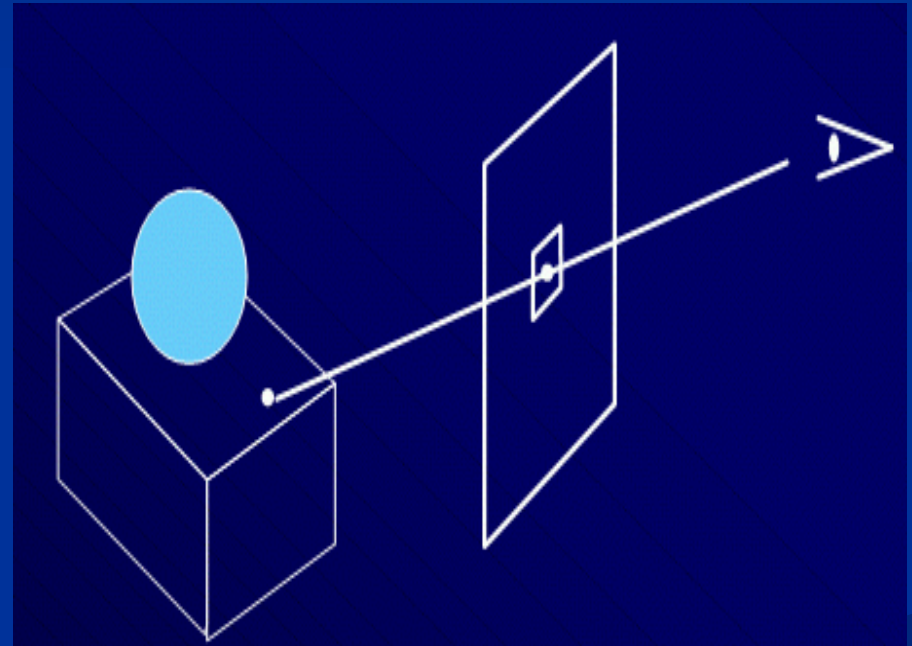
- Deriva el problema de elección en un problema 2D.
- Para cada línea de barrido se halla el color del pixel dependiendo del objeto que lo ocupe.
- Para eliminar las partes ocultas se recorren los spans.



# Métodos de línea de barrido

## Raycasting

- Determina cuál es el color a aplicar en el centro de un determinado pixel.
- Se lanzan rayos desde el punto de observación a través de cada pixel y se determina que objeto intercepta el rayo.



# Comparación de métodos

Algoritmo	10 p.	2500 p.	60000 p.
Listas prioridad	1	10	507
Z-buffer	54	54	54
<u>Scan-Line</u>	5	21	100
<u>Warnok</u>	11	64	307

- Se considera que la complejidad viene dada por el número de polígonos

# Iluminación local

- Un modelo de iluminación completo contempla dos procedencias como causas de la intensidad luminosa observada en un punto:

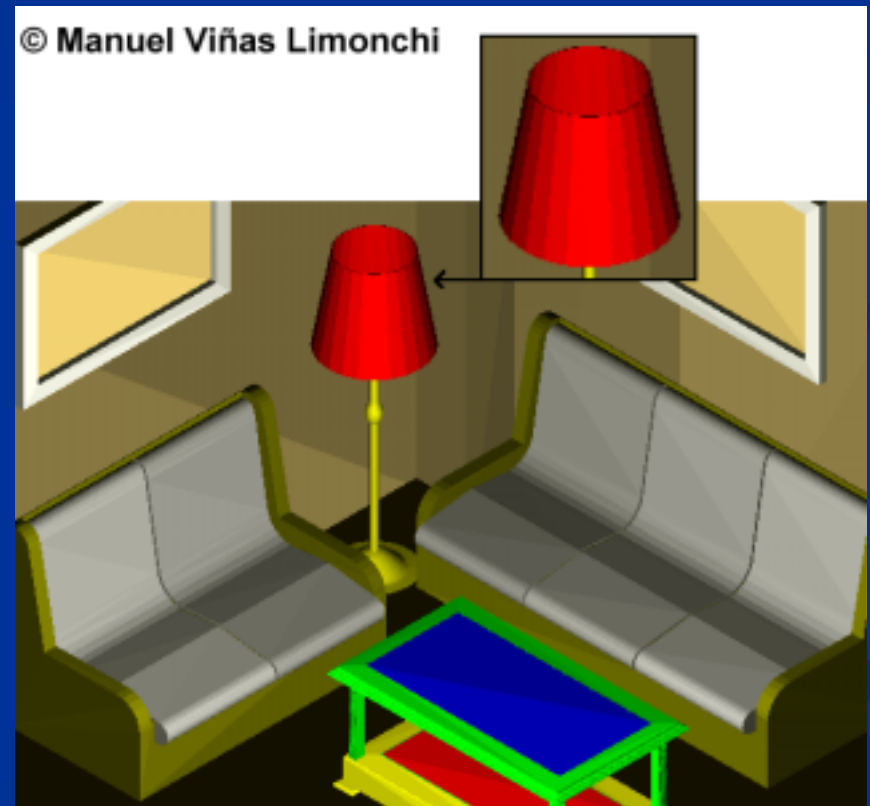
Fuentes de luz - Objetos

- La iluminación local utiliza modelos empíricos de iluminación.
- Modelo de iluminación  $I = K_i$
- $$I_R = I_{aR} k_a O_{dR} + f_{att} I_{pR} [ k_d O_{dR} (N \cdot L) + k_s O_{sR} (V \cdot R)^n ]$$
- Combinación de los modelos de iluminación ambiental, difusa y especular

# Sombreado

## Sombreado plano

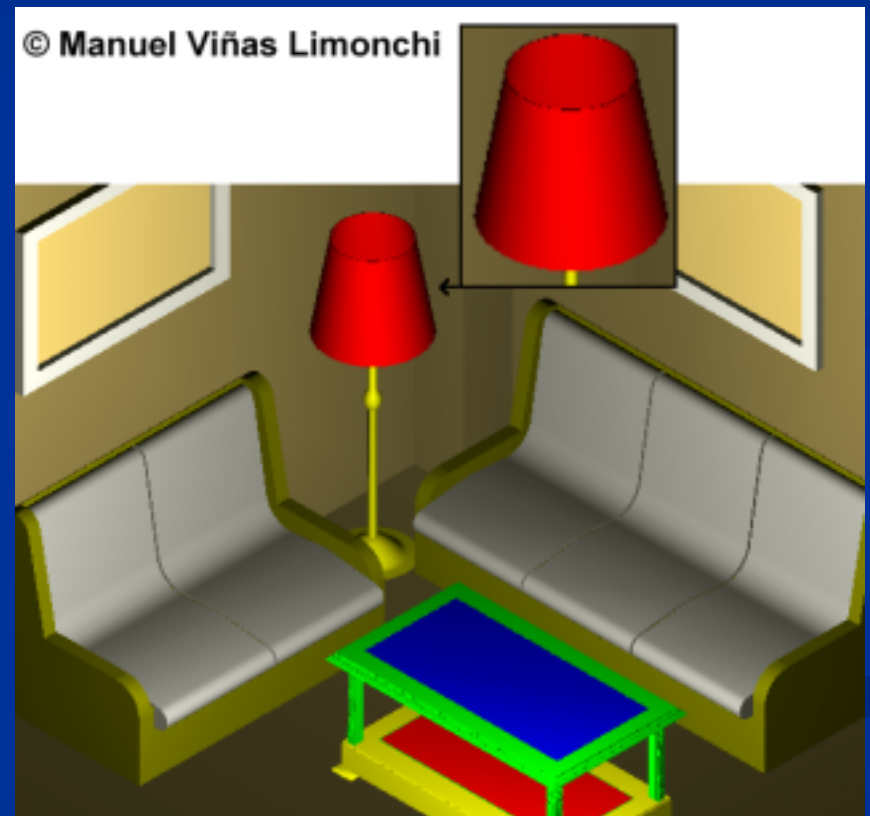
- Se muestrea el valor de la ecuación de iluminación una vez para cada polígono y se mantiene el valor en todo el polígono para reconstruir su sombra.
- Fuente de luz y observador están en el infinito.



# Sombreado

## Sombreado Gouraud

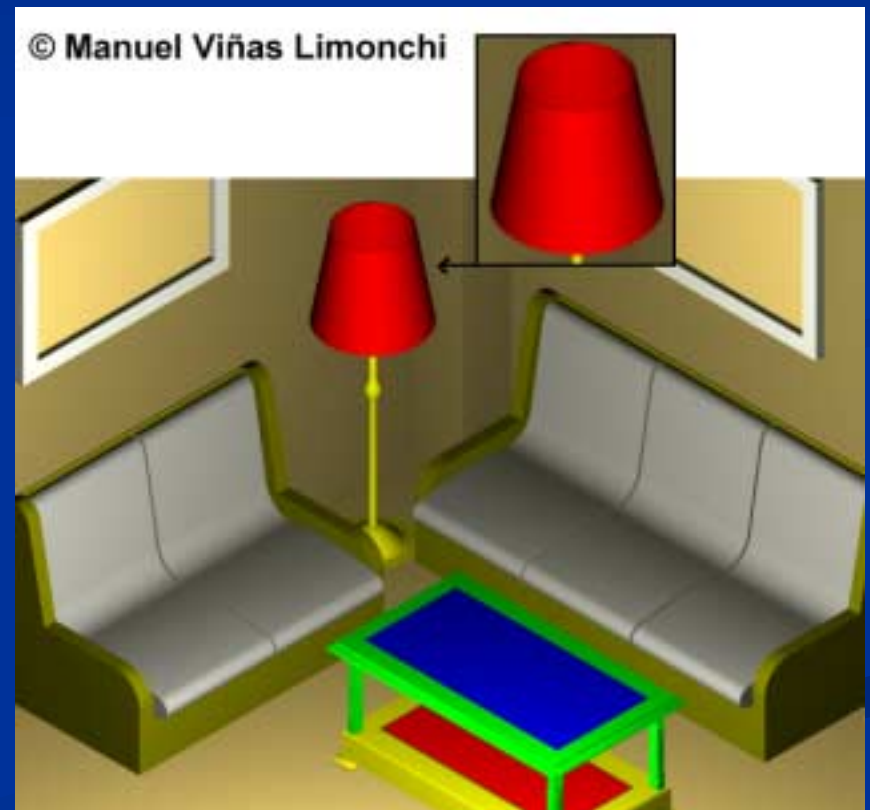
- Interpola los valores de iluminación de los vértices de los polígonos de la superficie que se aproxima.
- Precisa conocer la normal a cada vértice de la malla poligonal.



# Sombreado

## Sombreado de Phong

- Realiza una interpolación lineal de las normales a la superficie del polígono.
- Aumenta la calidad de sombreado de la imagen suavizando las aristas de los polígonos.



# Iluminación global

## Ray-Tracing

- El método de trazado de rayos emite un rayo desde una posición determinada de píxel de manera que éste, además de localizar las intersecciones de la superficie para modelar los objetos y determinar las superficies visibles de la escena, recopila información sobre la contribución de la intensidad luminosa de los objetos.



# Iluminación global

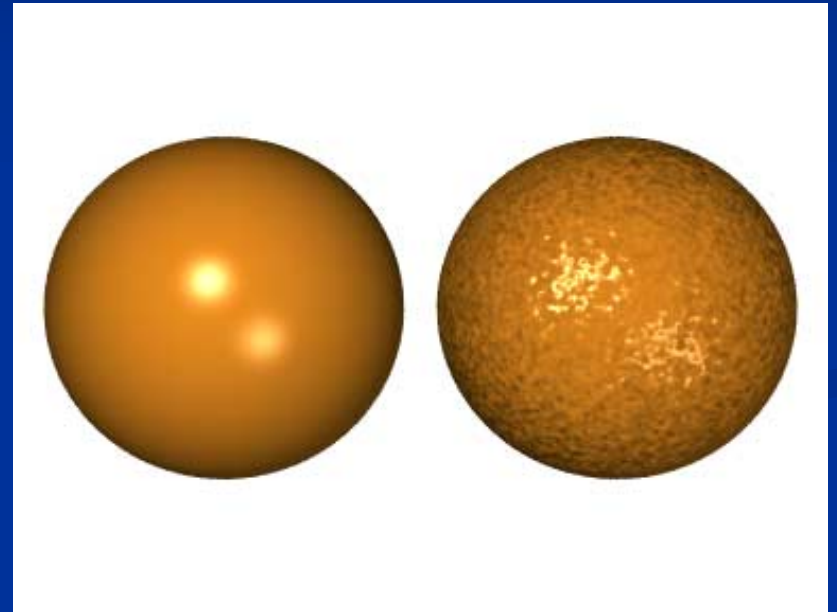
## Radiosidad

- Considera todos los objetos como fuentes potenciales de luz
- Primero se determinan todas las interacciones luminosas en un ambiente, independientemente de la vista.
- Un cambio en la geometría, posición de los objetos o de textura obliga a recalcular todo.



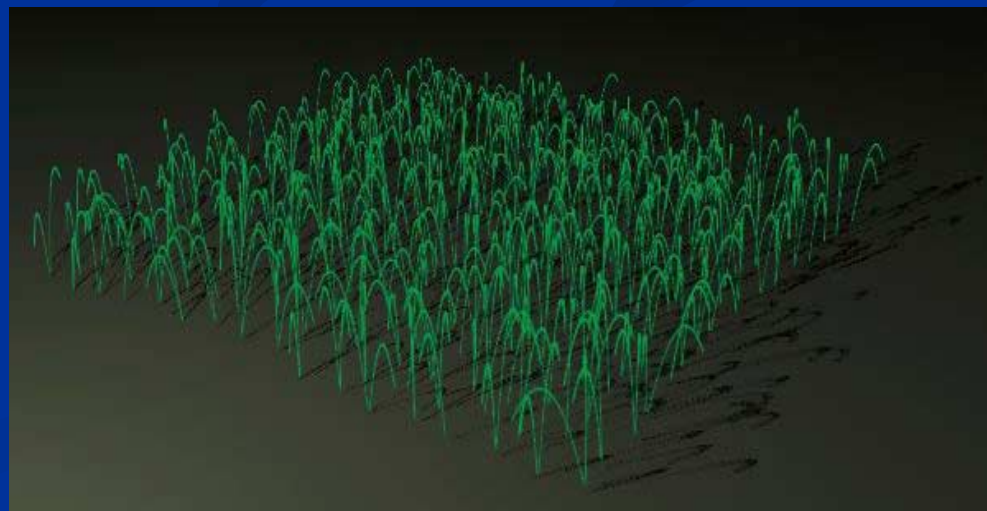
# Más realismo

- Propiedad de los materiales
- Detalle de superficies
  - Polígonos de detalle
  - Mapas de textura
  - Bump-mapping
- Modelado de superficies curvas
- Sombras proyectadas
- Transparencia y reflexión



# Más realismo

- Modelos de cámara mejorados: depth of field, lente de pez o motion blur.
- Modelos de objetos avanzados: fractales, métodos gramaticales o sistemas de partículas.



# GPU

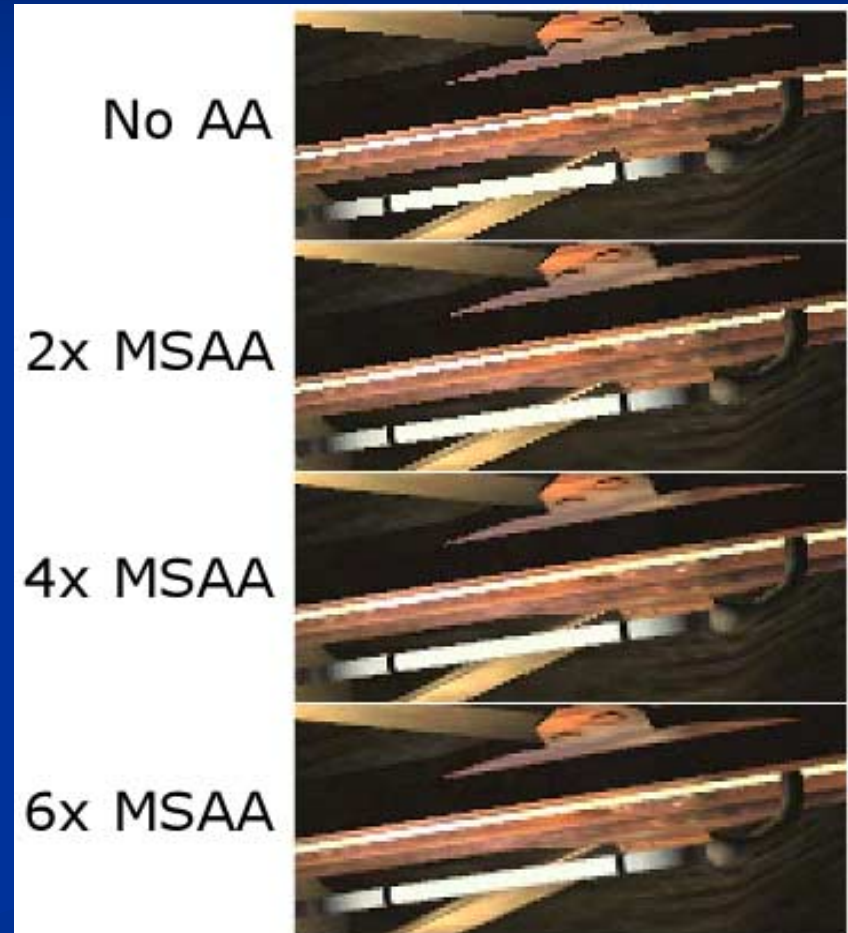
- El aumento de la capacidad de procesamiento de las unidades de proceso gráfico (gpu) ha permitido aumentar la complejidad de las imágenes a representar.
- El elemento principal ahora es el shader.
  - Vertex shader
  - Pixel shader

```
void VertexShader(  
    float4 position, // In model space  
    float3x3 modelToTangent, // Local tangent, normal and binormal  
    float4 eye, // In model space  
    float4 light, // In model space  
    uniform float4x4 modelToProjection,  
    out float4 positionProj,  
    inout float2 texCoord,  
    out float3 viewVector,  
    out float3 lightVector  
)
```

```
float4 PixelShader(  
    float2 texCoord, float3 viewVector, float3 lightVector,  
    uniform sampler2D colorMap,  
    uniform sampler2D normalMap,  
    uniform sampler3D distMap  
)
```

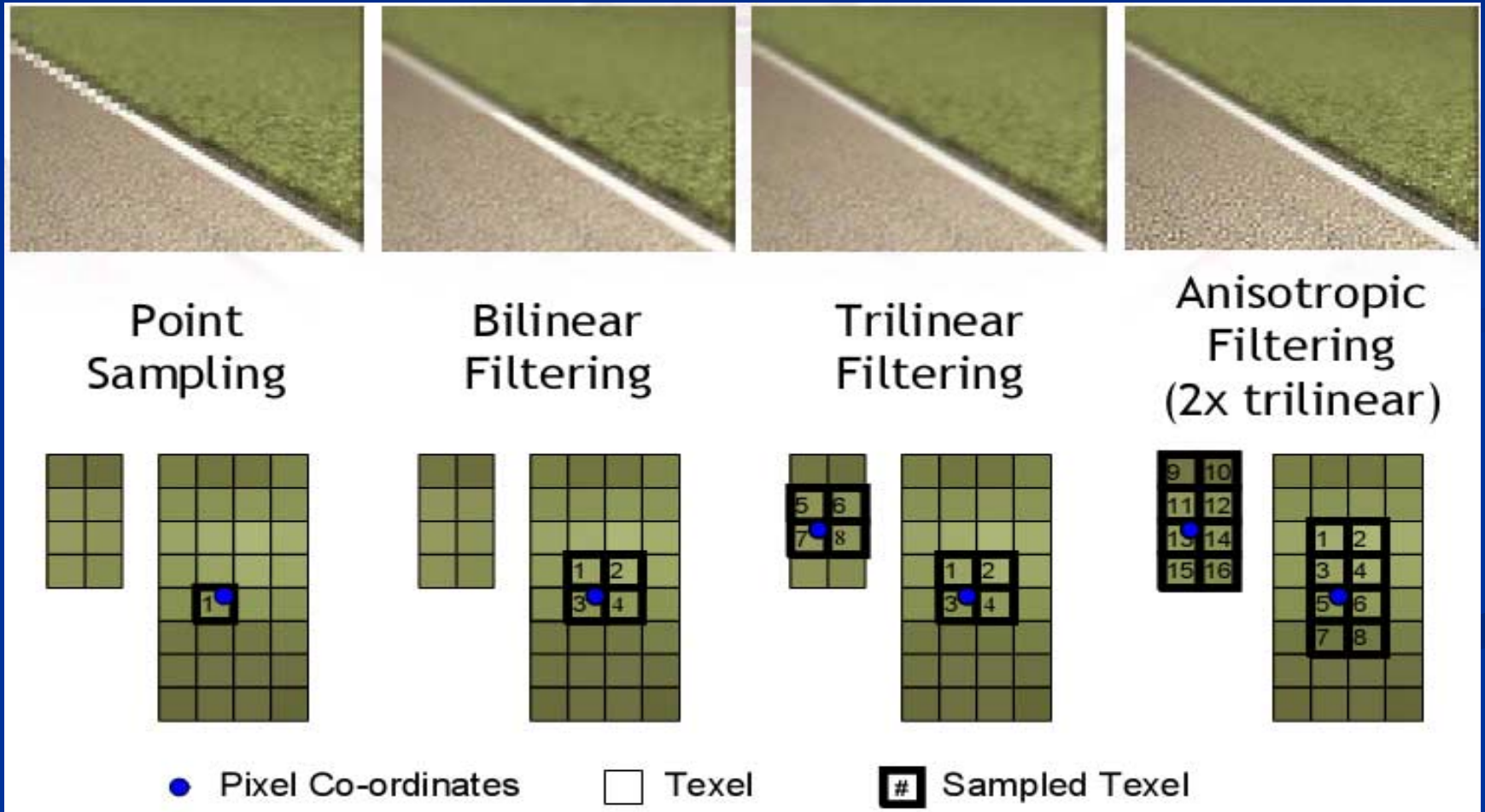
# GPU

- HDR
- Anti-aliasing
  - Multi-sampling



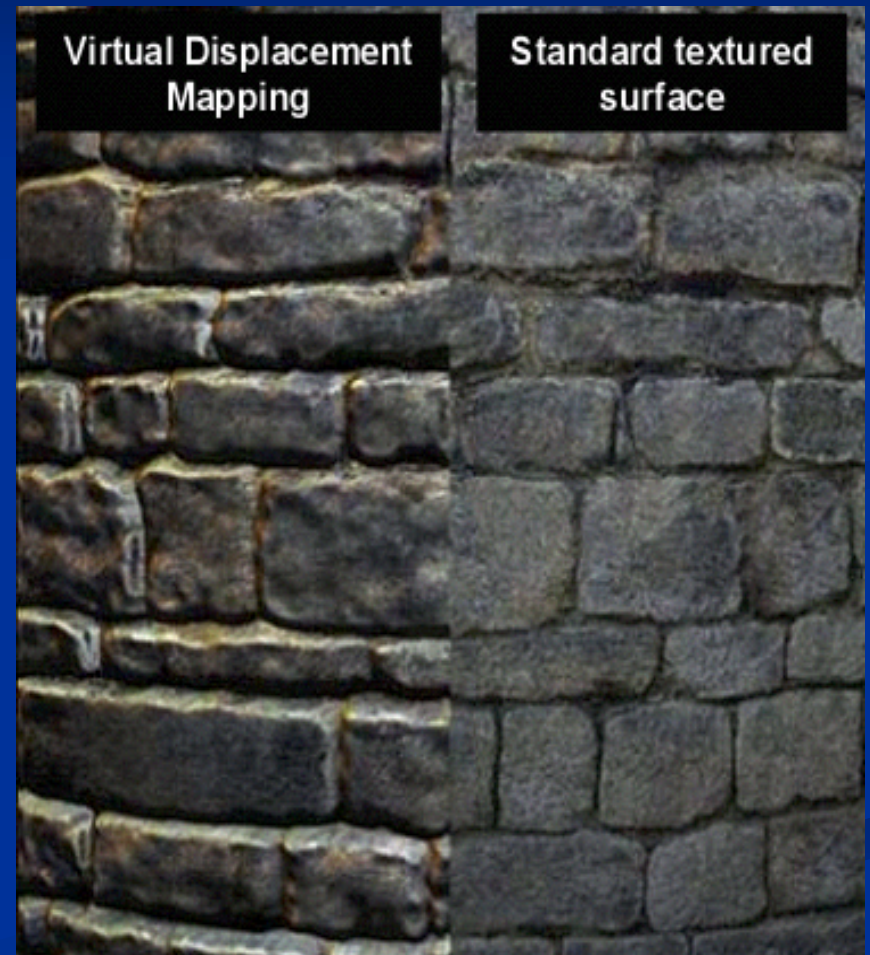
# GPU

- Filtro de texturas



# GPU

- Mapa de desplazamiento
- Mejora del procesado del Z-buffer



# Bibliografía

- Foley J., *Computer Graphics Principles and practice (2ª edición)*, Addison-Wesley, 1990.
- Foley J., *Introducción a la graficación por computador*, Addison-Wesley, 1996.
- Viñas Limonchi, M., *Técnicas de Infografía*, McGraw-Hill, 2000.
- Watt, *3D computer graphics (3ª edic.)*, Addison Wesley, 2000.

## Direcciones web

- <http://www.ii.uam.es/~pedro/graficos/teoria/index.html>
- <http://www.ncst.ernet.in/apgdst/cgfac/schedule.shtml>
- <http://www.nvidia.com>
- <http://www.ati.com>