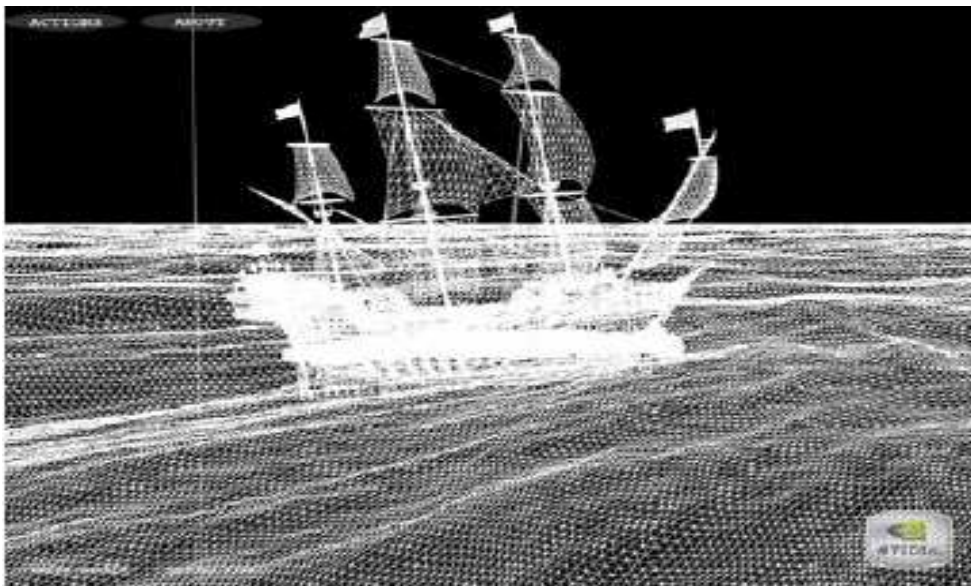


Renderizado en tiempo real

Tiempo real significa visualizar en “al menos” 10 imágenes por segundo



Escena 3d = conjunto de primitivos 3D (triángulos, líneas puntos)



Imagen = arreglo de pixeles

OpenGL vs DirectX

	OpenGL	DirectX
Ownership:	OpenGL Architectural Review Board	Microsoft Corporation
Platforms:	Windows, Linux, Unix, MacOS	Windows
Native language:	C	C++
Most popular fields:	Computer aided design, scientific visualization, movie industry, academic world...	Game industry
API specification:	OpenGL specification	DirectX documentation
API evolution:	Mostly through extensions to the core API	Mostly through complete revisions of the core API every year or so
API functionalities:	Mostly equivalent	

Aplicación



Triángulos 3D



Etapa geométrica



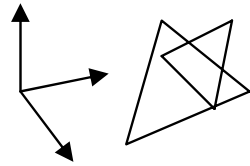
Triángulos 2D



Etapa rasterizado

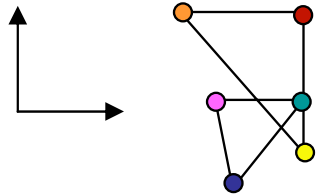


Pixels



Para cada vertex de un triángulo:

- Transforma la posición 3D en una posición de la pantalla
- Calcula los atributos

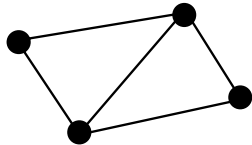


Para cada triángulo:

- Rasteriza el triángulo
- Interpola los atributos de los vertices a lo largo del triángulo
- *Shade* los pixeles, resuelve visibilidad



La etapa geométrica



Primitivos (e.g. vértices) en el marco de referencia del objeto

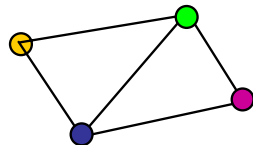
$$v = [x \ y \ z \ w]$$

Modeling
& Viewing

Matriz Modelview: $VM_{4 \times 4}$ (En OpenGL es la matriz de visualización, V , multiplicada por la matriz de transformación M)
Coordenadas oculares = posición de la cámara

$$= \text{ojo al origen mirando hacia } -z = VM \ v = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Iluminación
vertices



Posición luces en coordenadas oculares después de ser multiplicadas por VM
Uso de las normales de los vértices

Proyección

Matriz de Proyección $\times \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$

La etapa geométrica



Proyección

Matriz de Proyección \times α = Coordenadas de corte (clipping)

Espacio de recorte:

- Después de la proyección y antes del proceso de perspectiva
- Recortes con respecto al frustum son hechos aquí (vértices fuera del frustum son eliminados)
- Nuevos vértices se generan debido al recorte
- El frustum, después de la transformación es siempre un paralelepípedo independientemente de la proyección (perspectiva o ortográfica)



Homogenei-
zación

Se realiza el *perspective divide*:

- Transforma el espacio de recorte en espacio NDC (normalized device coordinates)

$[x' \ y' \ z']$: Divide $[x \ y \ z \ w]$ por w donde $w = z/-d$ ($d=1$ en OpenGL por lo que $w = -z$)

z' s sirven para cálculo de superficies cubiertas (hidden surfaces)



Viewport
transform

Escala y traslada x' , y'

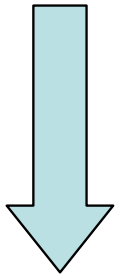
- Mapea las coordenadas NDC a la ventana
- x, y son enteros, y referenciados con la esquina izquierda inferior de la ventana

La etapa geométrica



Sistema de
Mapeo de
ventanas

Traslada x' , y' : coordenadas relativas a la esquina física de la pantalla



A etapa de rasterizado

Etapa de rasterizado

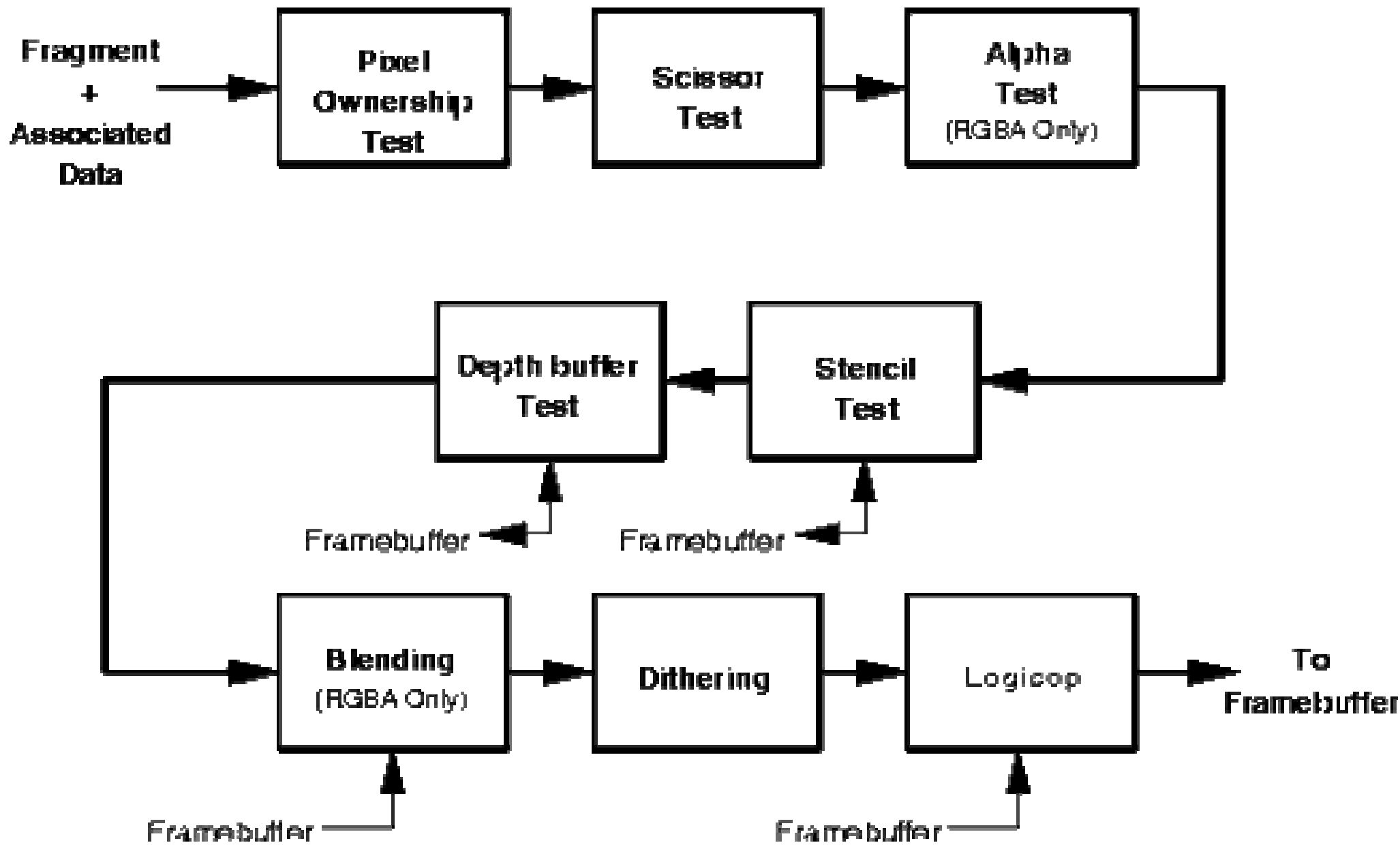
- Operaciones en cada pixel:
 - Convierte cada primitivo en fragmentos
 - Fragmento: estructura de datos transitoria
 - Profundidad
 - Colores
 - Coordenadas de texturas, etc.
- Fragmentos de los polígonos rasterizados son seleccionados (e.g. usando un proceso de comparación del z-buffer) para formar buffer de pixeles por cada imagen desplegada (*frame buffer pixels*)

Etapa de rasterizado

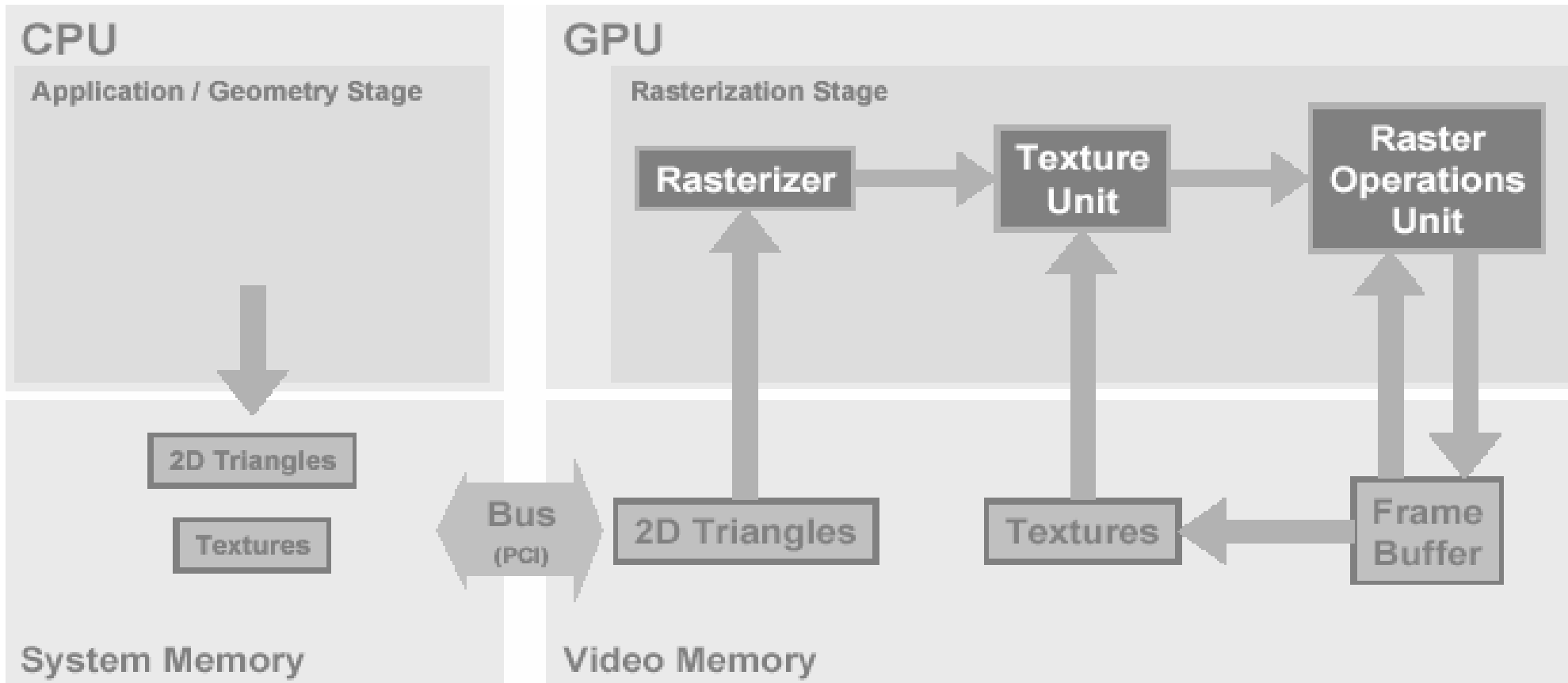
- Dos operaciones básicas
 - Selección de fragmentos: genera un fragmento por cada pixel que es intersectado (cubierto) por el primitivo
 - Asignación de fragmentos: Muestra las propiedades del primitivo (colores, profundidad, etc) para cada fragmento usando interpolación lineal

Etapa de rasterizado

- Pasos:
 - Interpolación de texturas
 - Interpolación de colores
 - Fog (niebla): combina el color de la neblina con el color del fragmento basados en los valores de la profundidad
 - Test de corte (scissor test): se hace con respecto a una región rectangular. Los fragmentos no contenidos dentro de dicha región son descartados
 - Test alpha (para transparencia, $\alpha = [0, 1]$, 0 es transparente: compara con alpha para conservar o desechar
 - Test del buffer stencil: pone una máscara al fragmento dependiendo del contenido del buffer stencil.
 - Buffer stencil: Determina que pixeles son dibujados. Permite *marcar* porciones de la imagen para que sean o no dibujados.
 - Test de profundidad (algoritmo del z-buffer)
 - Combinación de alpha (alpha blending)
 - Dithering: se acomoda la imagen a la resolución de la pantalla (imágenes pueden tener infinidad de colores. Dithering reduce el número a 256 o menos)



1995 – 1998: Mapeo de texturas y Z-buffer

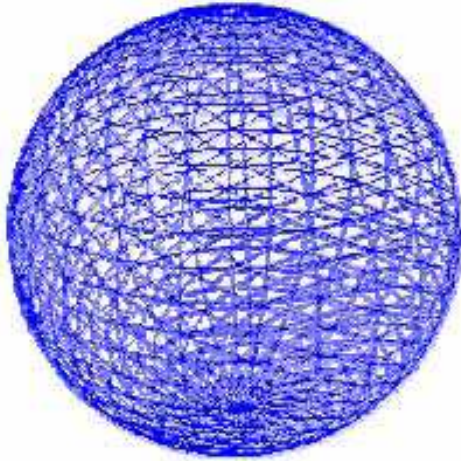


PCI : Peripheral component Interconnect

3dfx's Voodoo

Mapeo de texturas

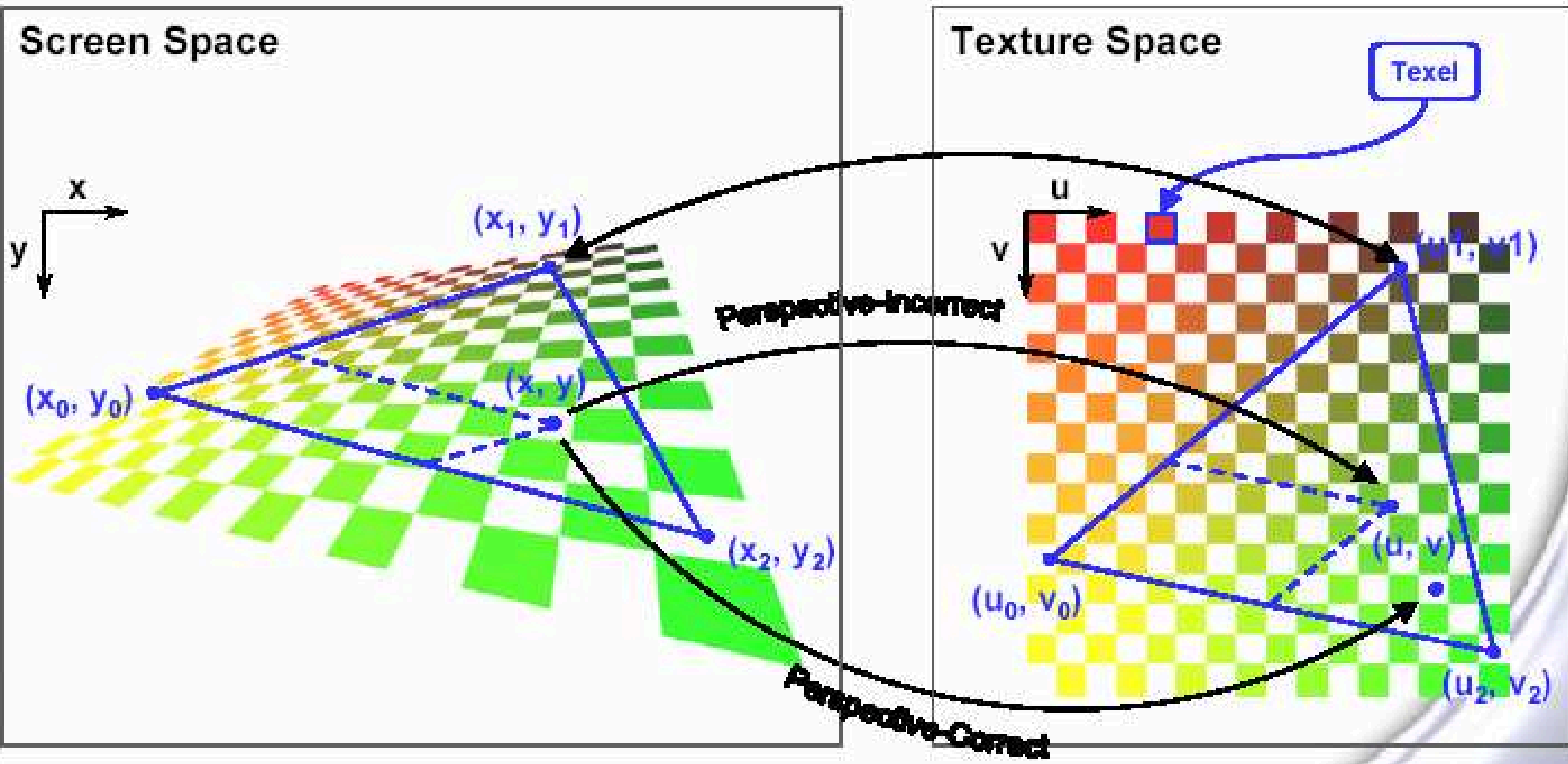
Malla de triángulos



Textura base

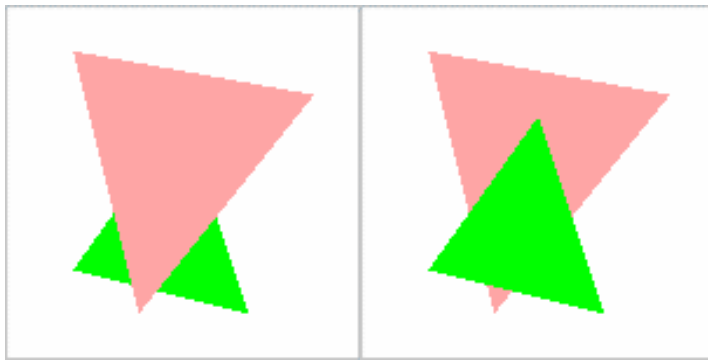


Mapeo de texturas

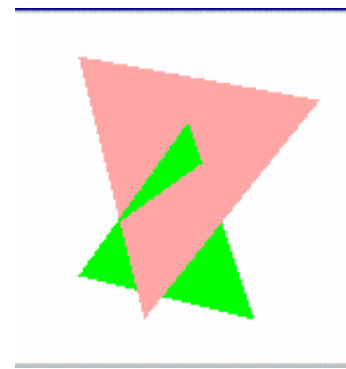


Z-buffer

- Es una parte de la memoria gráfica reservada a almacenar los valores del eje z para cada pixel.

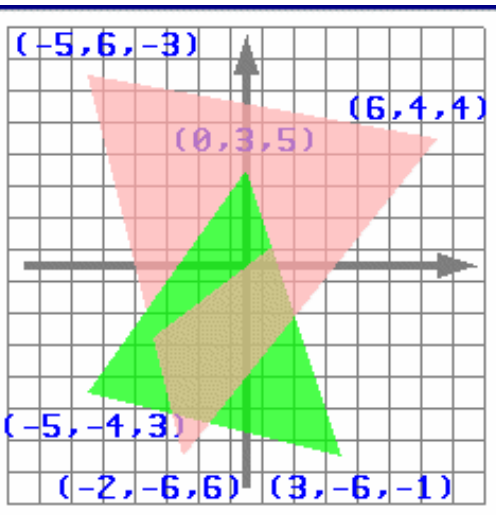


Sin z-buffer

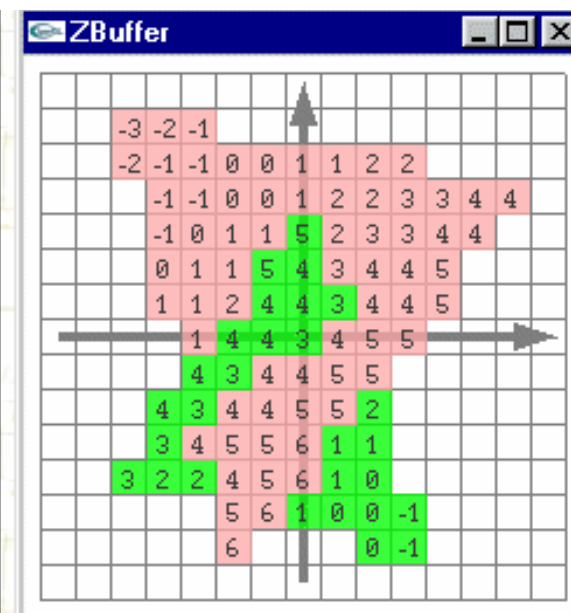
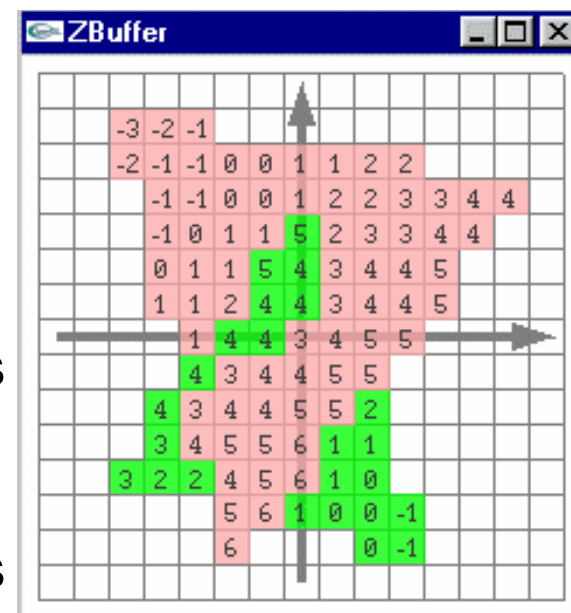
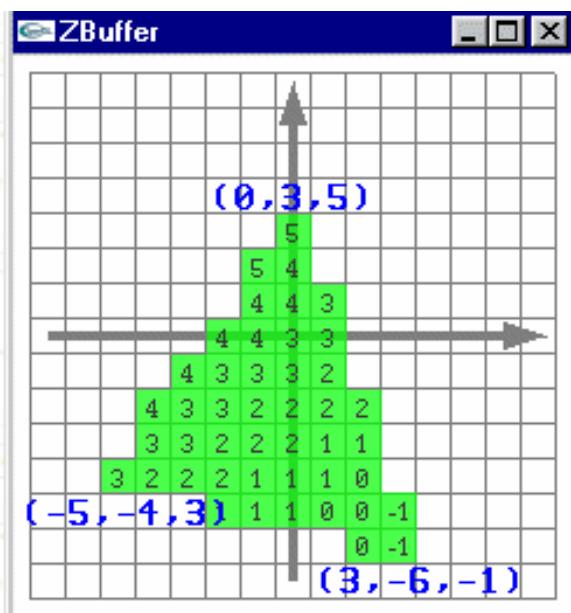
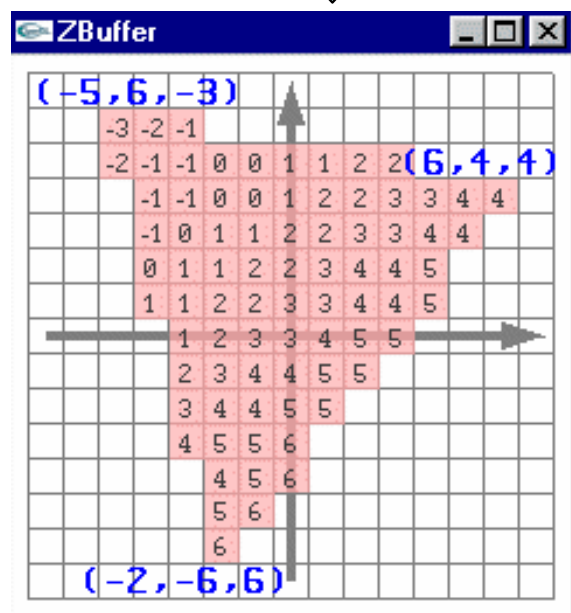


Con z-buffer

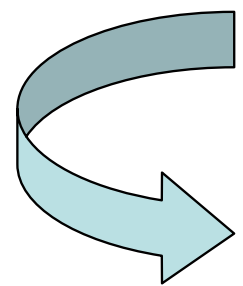
El algoritmo z buffer permite descartar partes de los objetos que se cubren entre si. Por ejemplo, imagina el triángulo verde y rosado que se están intersectando. Sin el algoritmo z-buffer esto no se percibe.



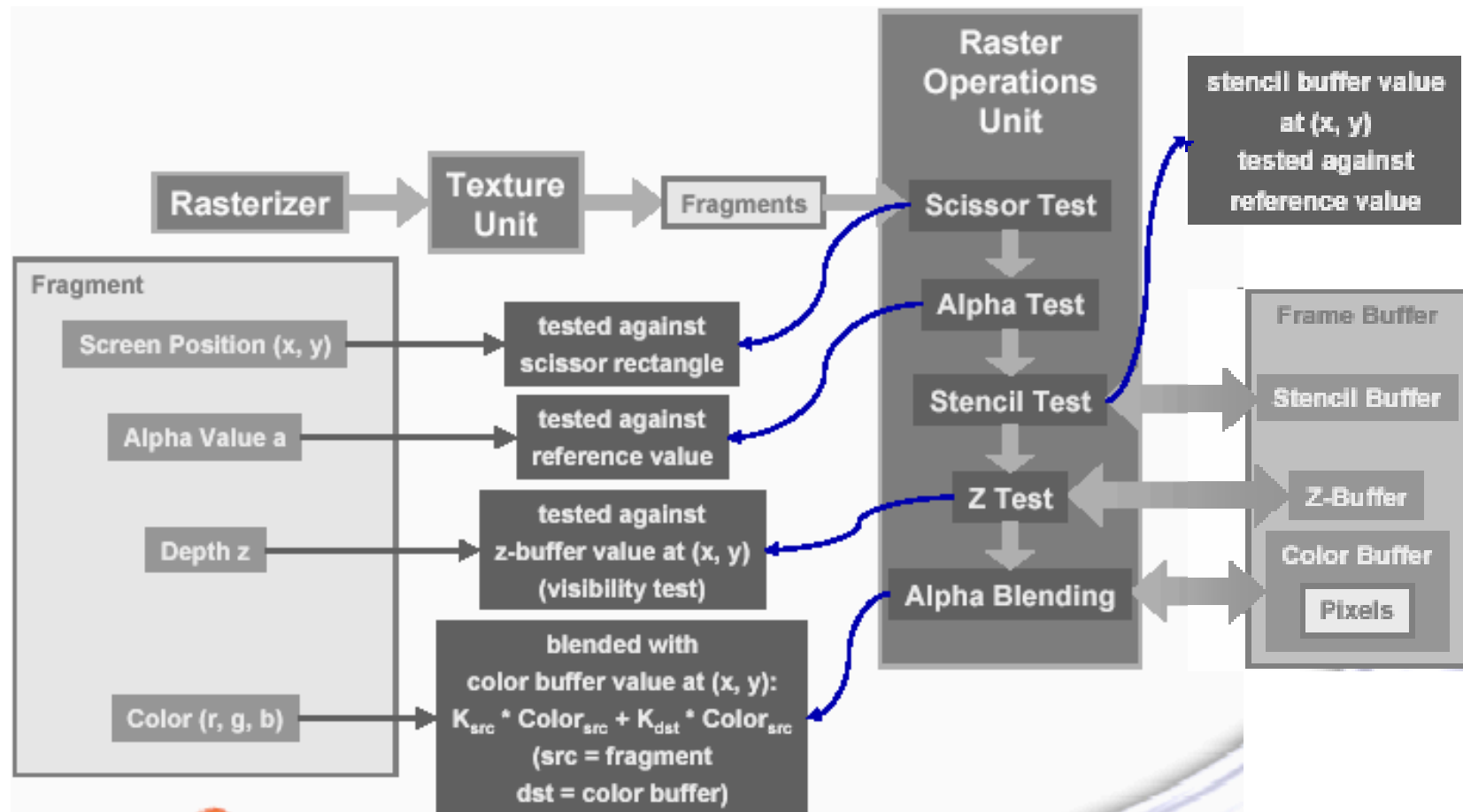
Pixelización y cálculo de la profundidad para cada una de las facetas de los triángulos



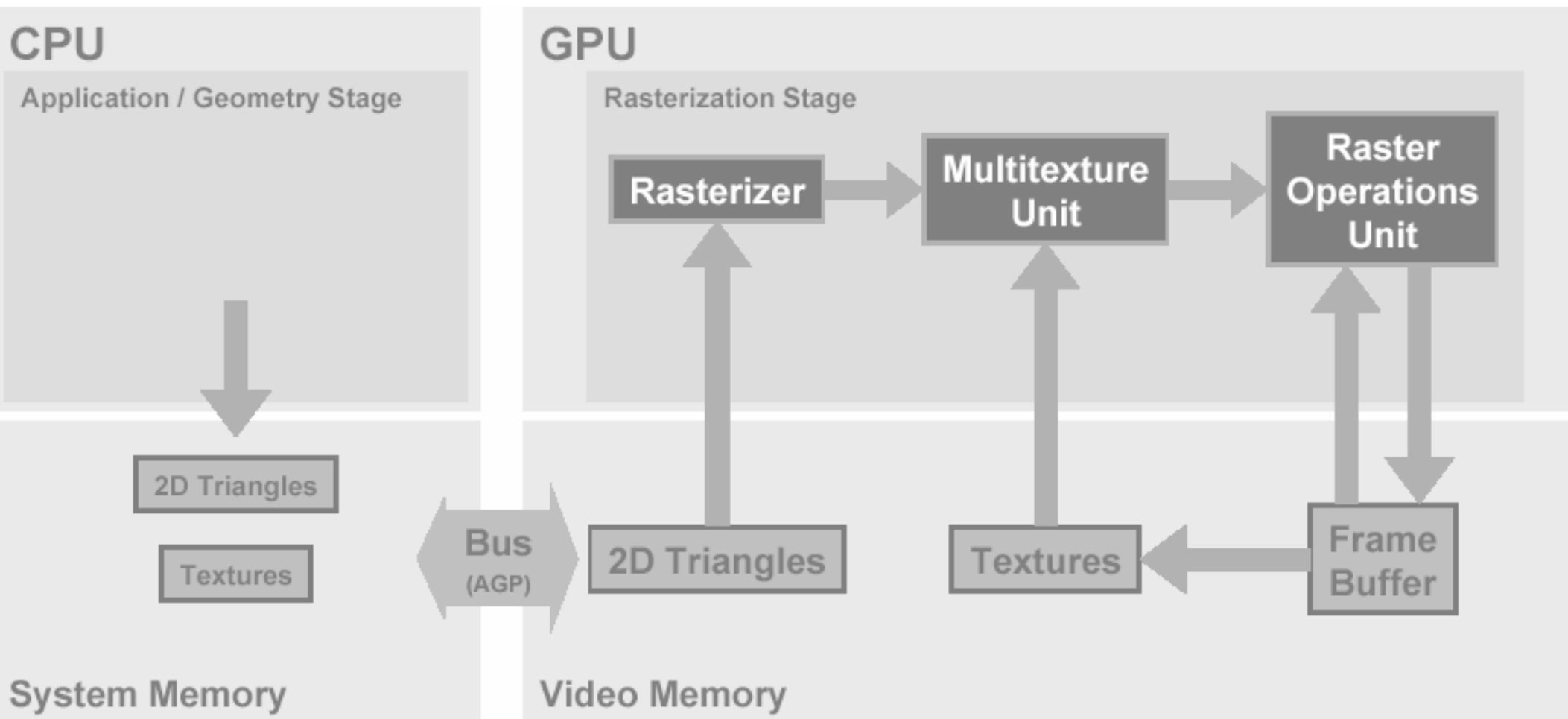
Se dibujan los pixeles dependiendo de la altitud. Dos resultados son posibles dependiendo de cómo se recorran los triángulos



Unidad de operaciones de rasterizado (Raster operations unit ROP)



1998: Multi texturado



- AGP (Accelerated Graphics Port)
- TNT's de NVIDIA, Rage de ATI

AGP

- PCI usa una conexión paralela
- AGP usa conexión serial
 - Menos pines, protocolo mas simple
 - > mas barato, mas escalable
- PCI usa un protocolo de bus compartido (shared-bus)
- AGP usa protocolo punto a punto
 - Ancho de banda no es compartido entre los dispositivos

...continuacion

- AGP usa su propia memoria (*AGP memory*):
 - La GPU puede buscar texturas directamente en la AGP memory
 - Su tamaño se conoce como AGP aperture
- Ancho de banda:
 - $AGP = 2 \times PCI$ (asi, $AGP2x = 2 \times AGP$, etc)

... multi texturado

Textura base



Modulado por

X

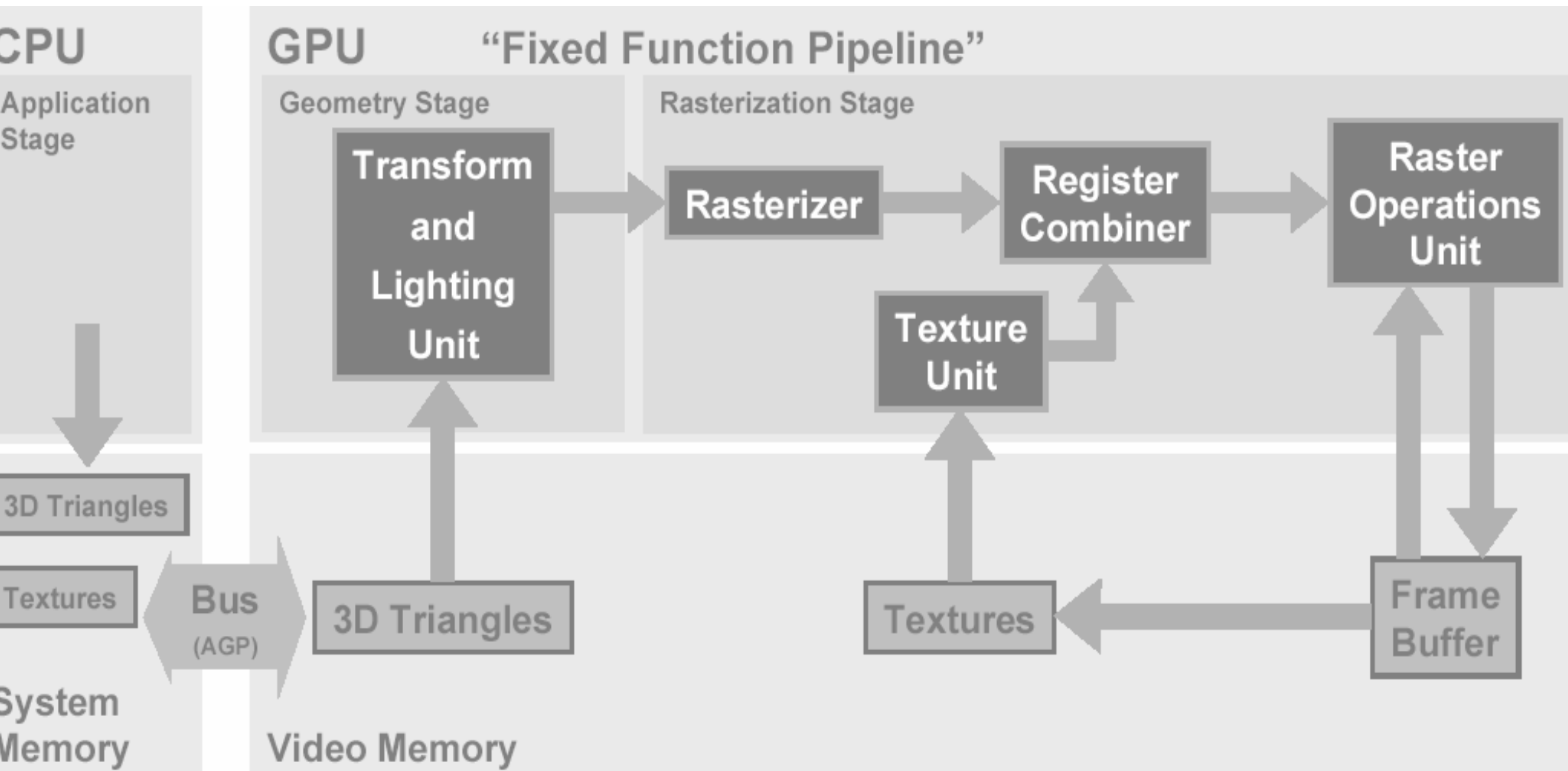
Mapa de luces



=



1999-2000: Transformaciones, luces

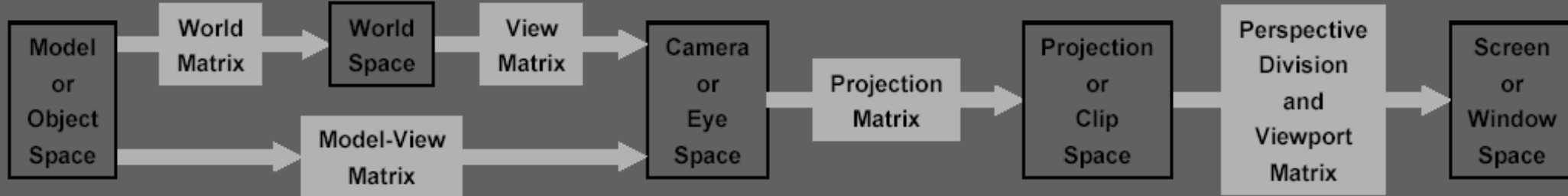


- Register combiner: Ofrece mas combinaciones de colores y texturas
- GeForce 256, GeForce 2 de NVIDIA, Radeon 7500 de ATI, serie S3 de Savage3D

Unidad de transformacion e iluminacion (TnL)

Transform and Lighting Unit

Transform

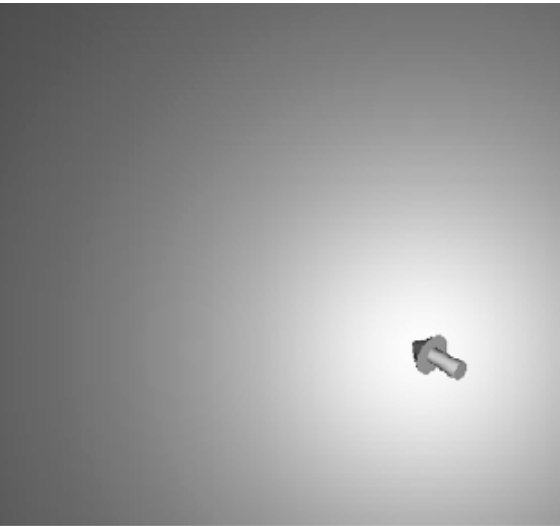


Lighting

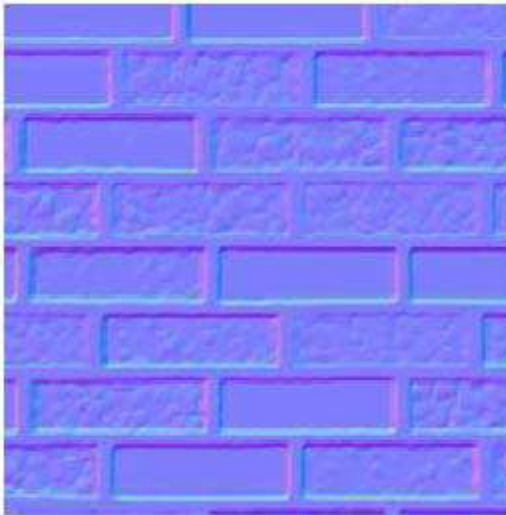


Bump mapping (mapeo a base del relieve)

Luz difusa sin bump



Mapa de normales



Luz difusa con bumps

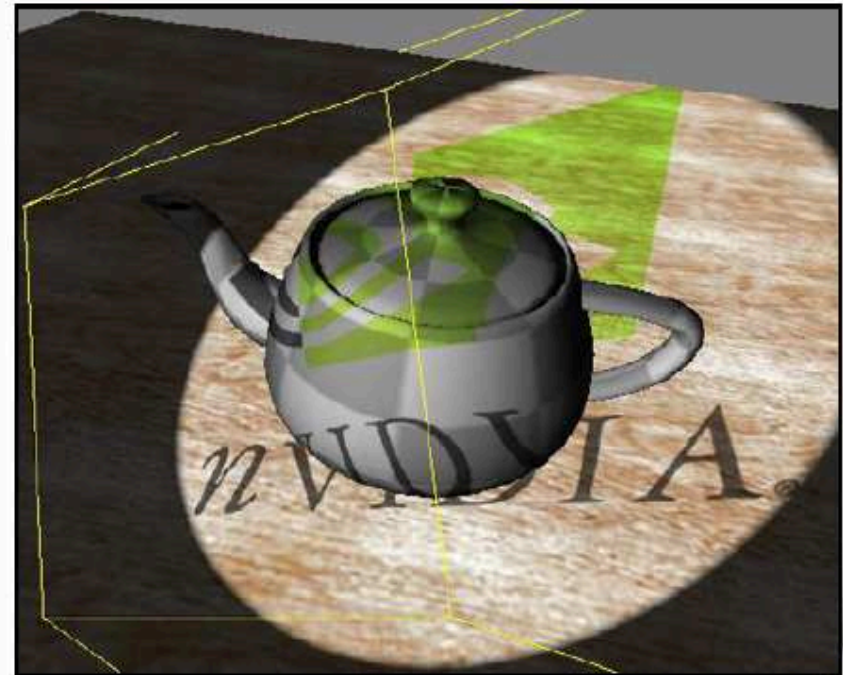
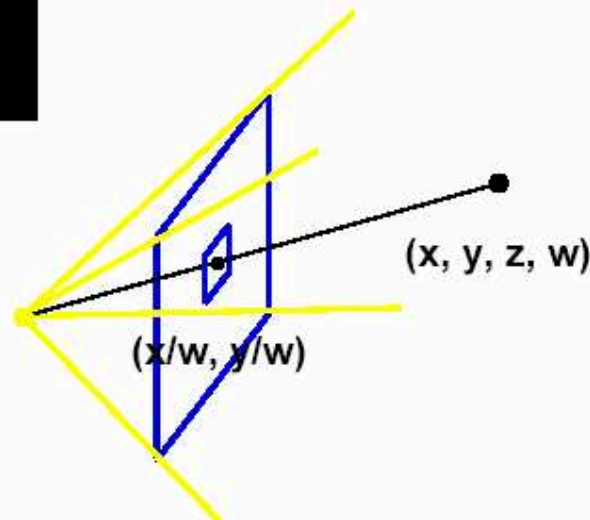


Bump mapping se refiere a el uso de las normales a partir de la textura (llamado mapa de normales) en lugar de usar interpolación de normales para calcular la luz en un pixel determinado

Mapeo de texturas en base a proyecciones

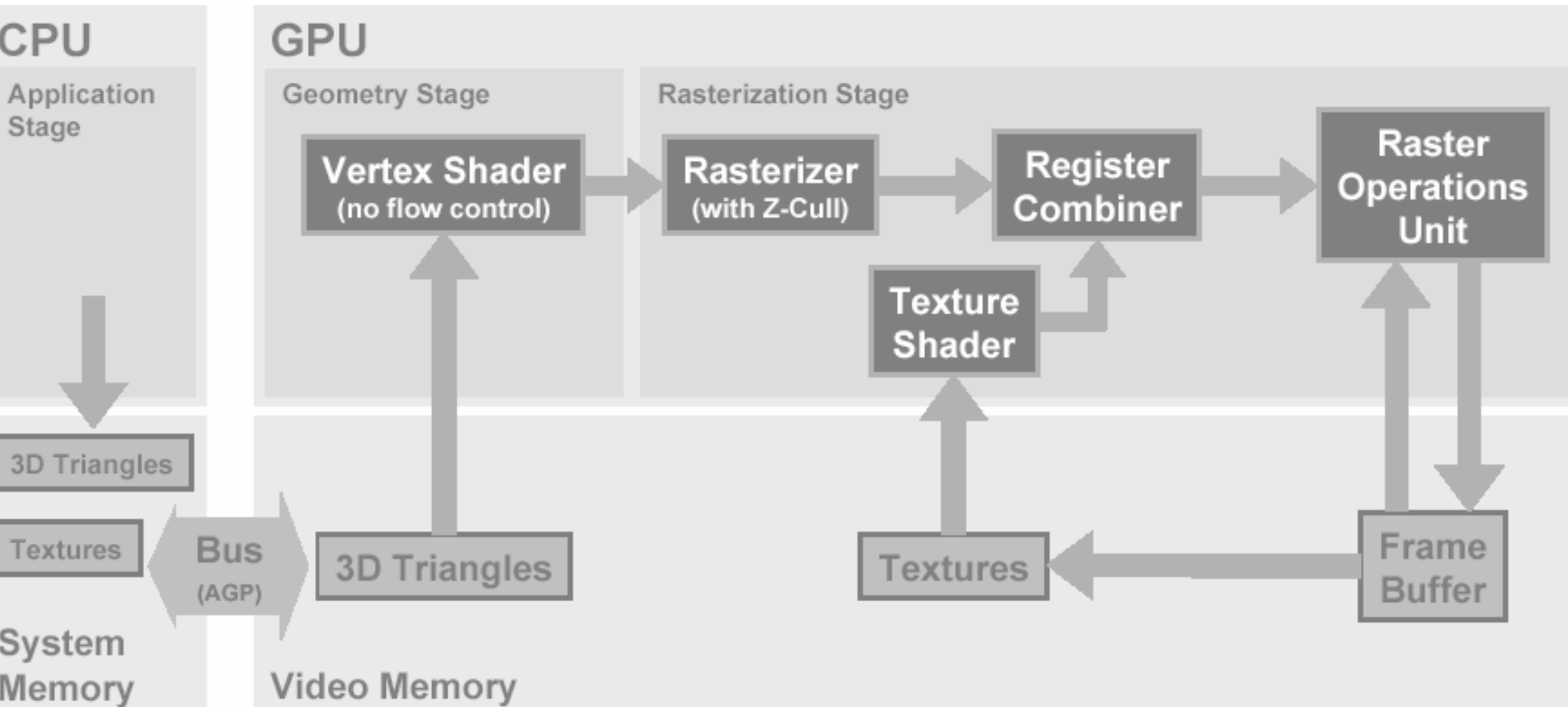


Projected Texture



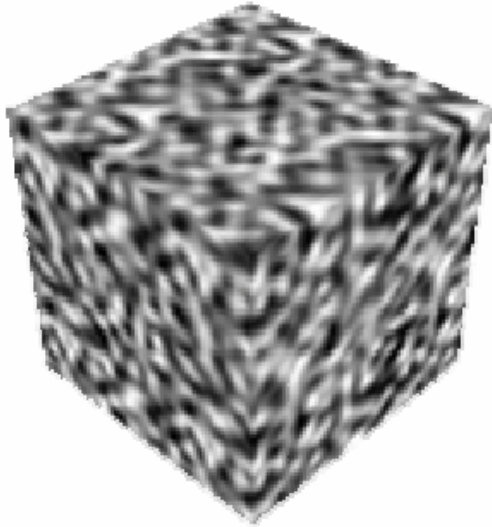
Texture Projection

2001: Vertex shader programmable

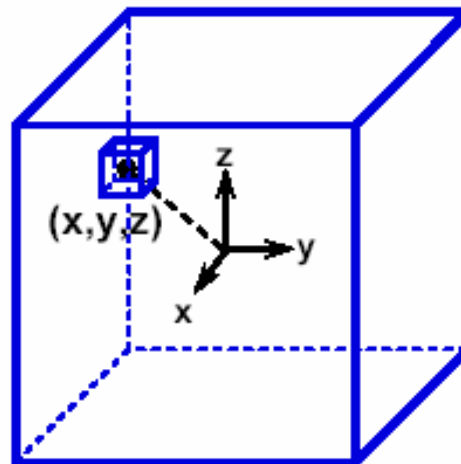
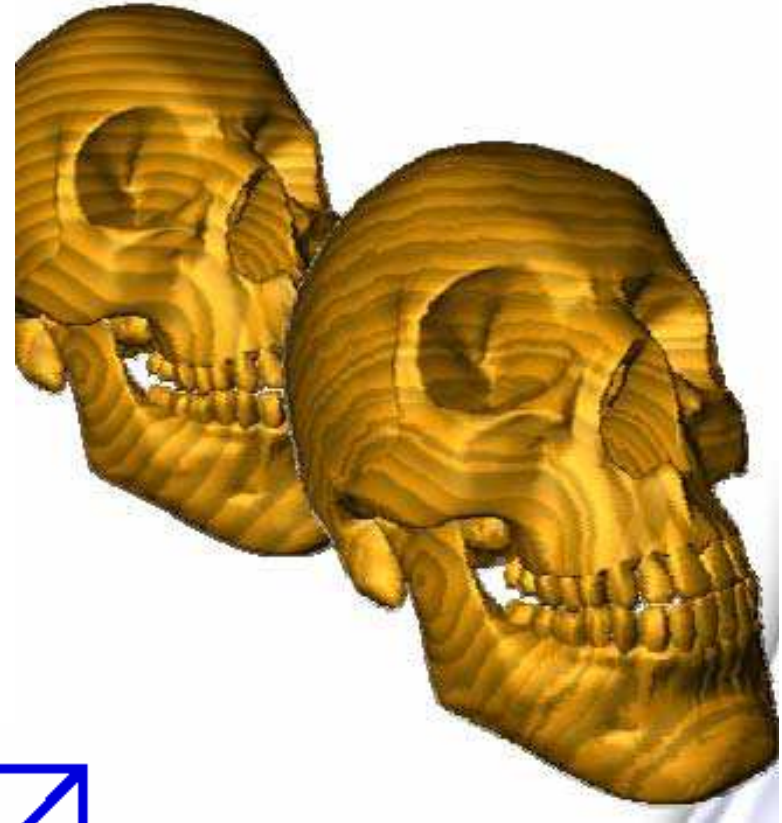


- Z-Cull : Predice que fragmentos no pasarán el z-test para descartarlos
- Texture shader: Permite más operaciones y direccionamientos para las texturas
- GeForce3, GeForce4 Ti de NVIDIA, ATI Radeon 8500

Volume texture mapping



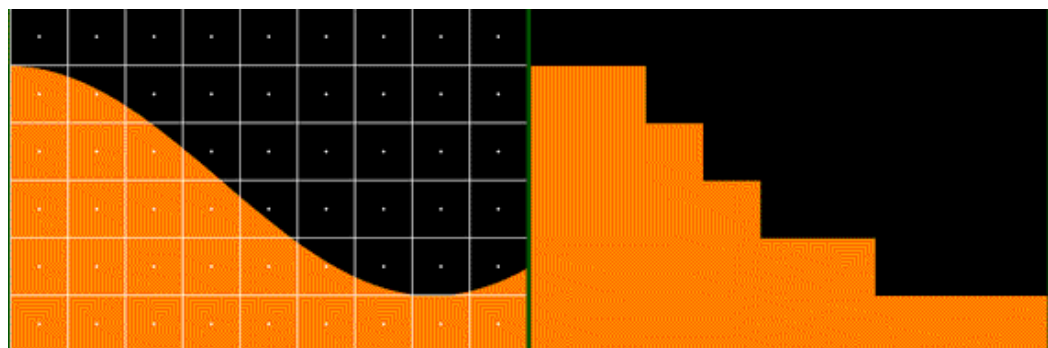
Volúmen de texturas



Mapeo de texturas en volúmen

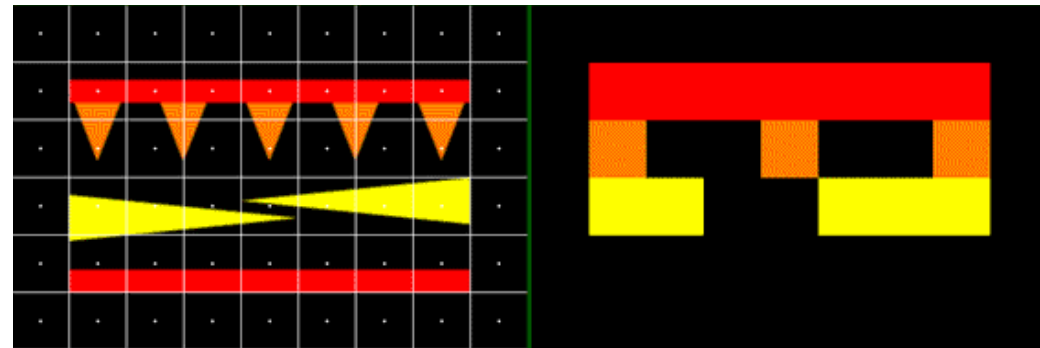
Aliasing

- Aliasing: Se refiere a problemas debido a un muestreo insuficiente de:
 - Primitivos (triángulos, líneas, etc.) → aristas irregulares (con picos)
 - Texturas o shadersEstos efectos son más notorios en imágenes animadas



Original

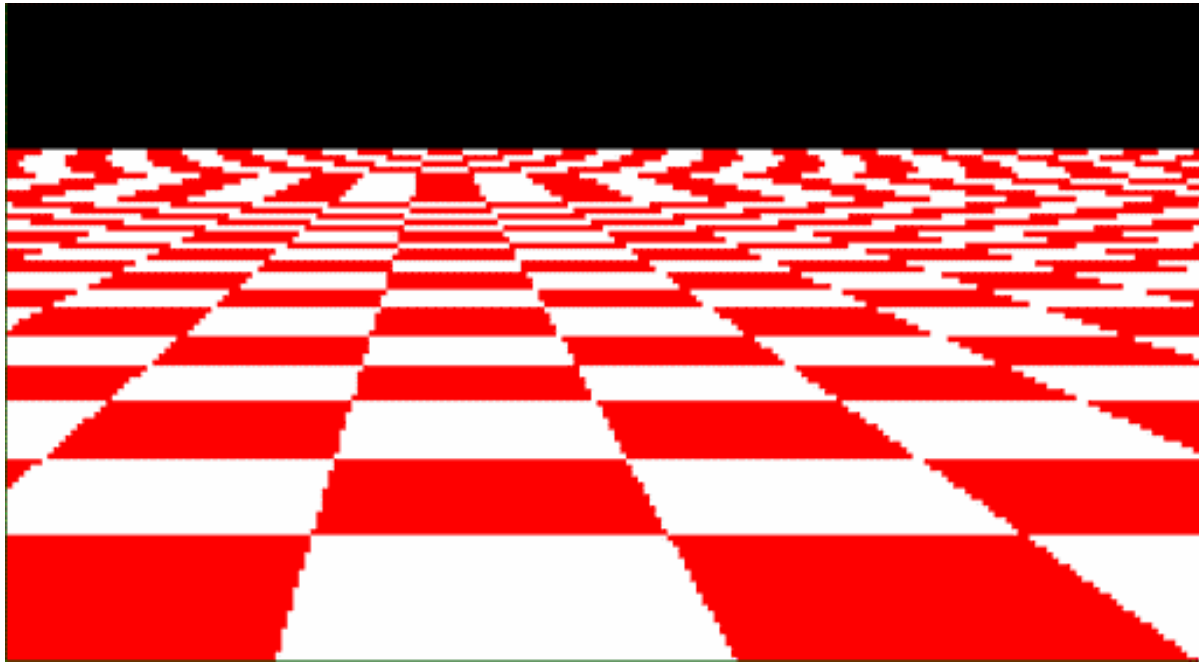
Renderizado
(aristas con picos)



Original

Renderizado
(pérdida de
detalles)

... aliasing

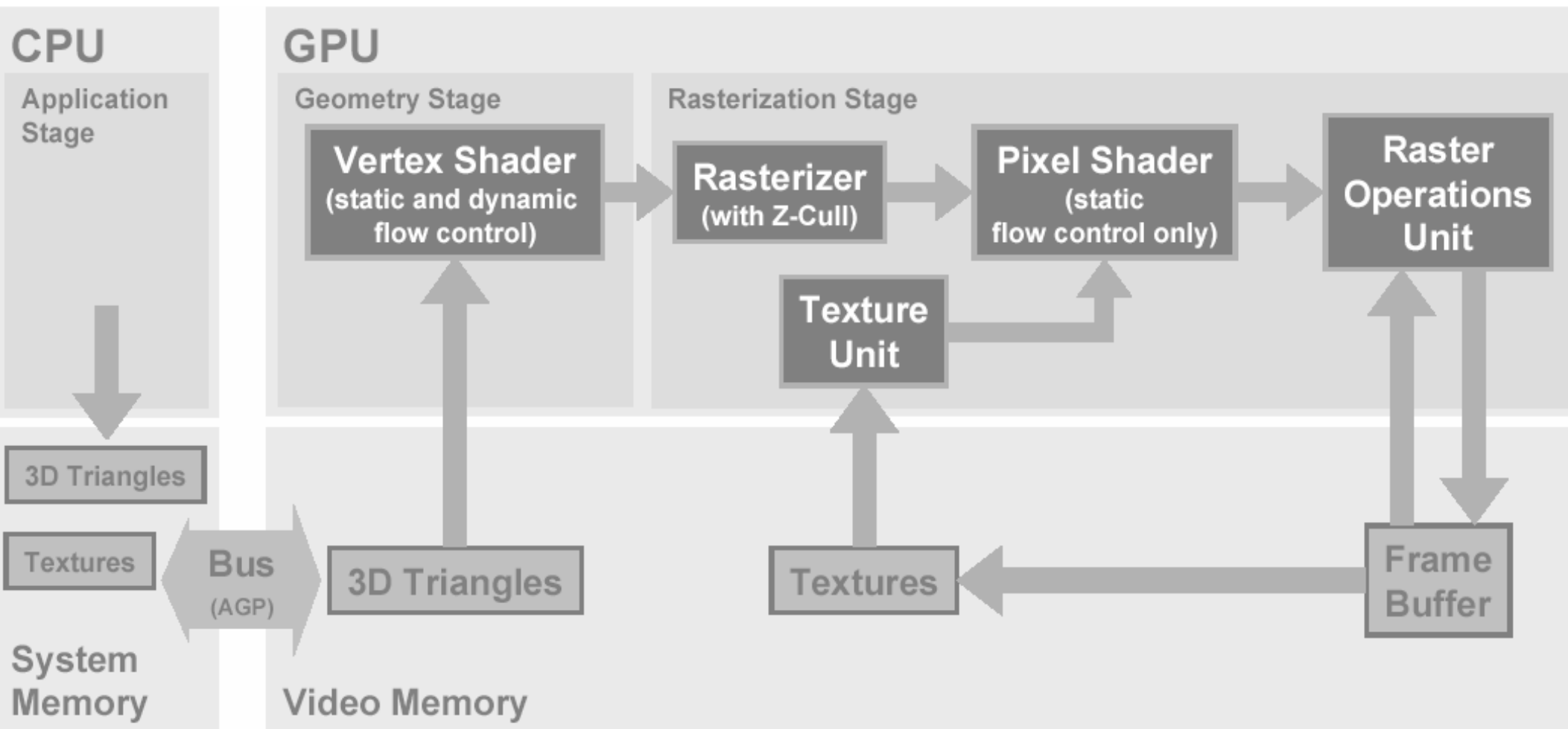


Las texturas se vuelven irregulares a medida que se alejan en el Horizonte.

Antialiasing

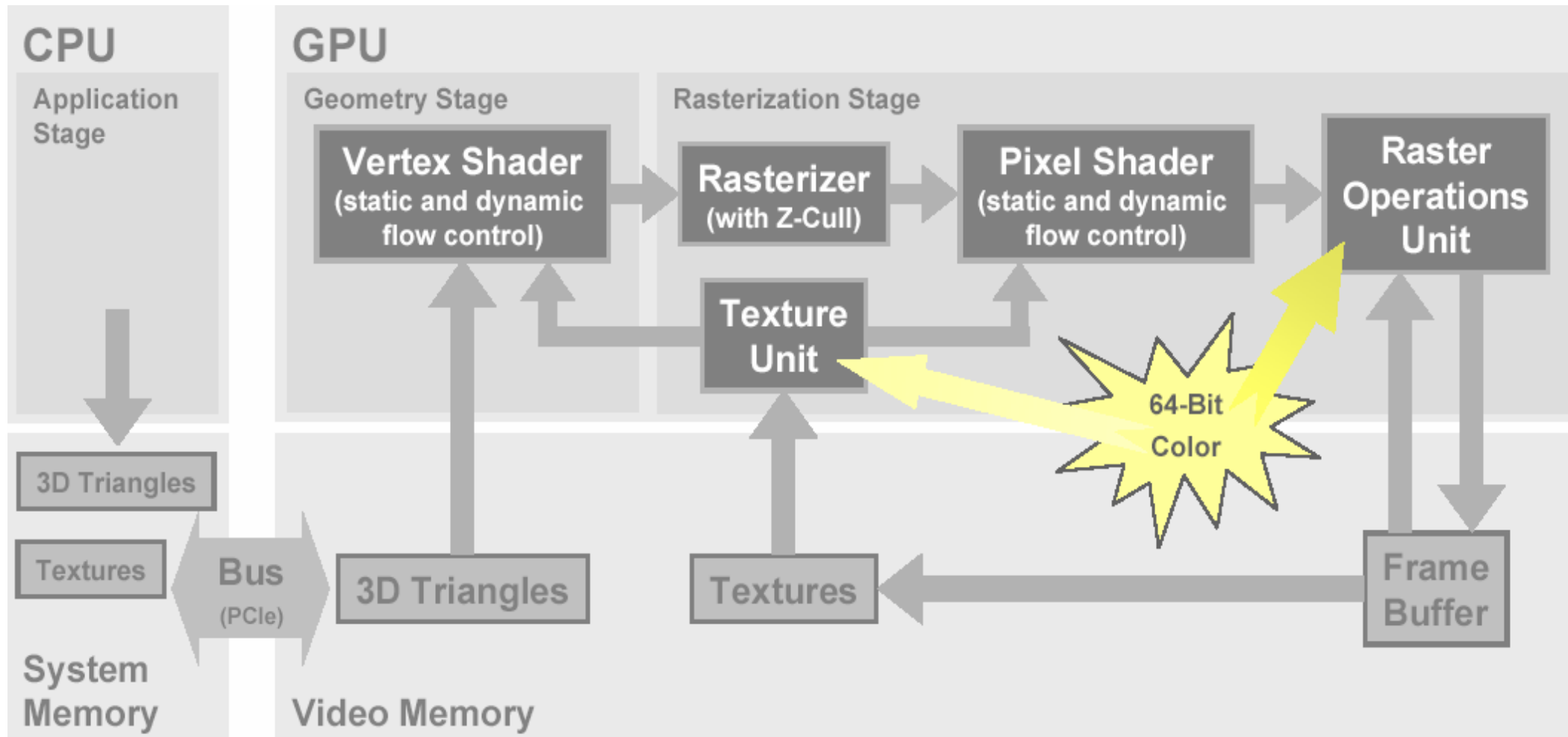
- Antialiasing: Método para reducir aliasing.
 - Texture antialiasing: Métodos de mipmapping y filtrado anisotrópico
 - Shader antialiasing

2002-2003: Pixel shader programmable



- MRT (multiple render target)
- GeForce FX de NVIDIA, ATI Radeon 9600 a 9800 y X600 a X800

2004: Shader Model 3.0 y soporte de color de 64 bits



- PCIe: Peripheral Component Interconnect Express
- Serie GeForce 6 (6800,6600) de NVIDIA

Soporte de color en 64-bits

- Apoyo en color de 64-bits: 16 bits de punto flotante por cada canal (R,G,B,A)
- Alpha blending: usa un buffer de color de 64 bits
- Filtrado de texturas usa texturas de 64 bits
- Aplicaciones:
 - Imágenes con alto rango dinámico
 - Composición de imágenes de alta precisión

Rango dinámico de una escena

- Es la tasa entre la más alta y la más baja luminiscencia de la imagen
- Escenas en la vida real pueden tener altos rangos dinámicos (varios millones)
- Pantallas e impresiones tienen bajos rangos dinámicos (alrededor de 100)
- Tone mapping: Es el proceso de visualizar altos rangos dinámicos en dispositivos de rangos dinámicos bajos.
- Rangos dinámicos muy altos usan colores de punto flotante
- e.g. OpenEXR es un formato de imagen de rango dinámico muy alto (compatible con formato de color de NVIDIA's 64-bits)

Tone mapping en tiempo real

La imagen es calculada en colores de 64 bits y con tone-mapping para el renderizado

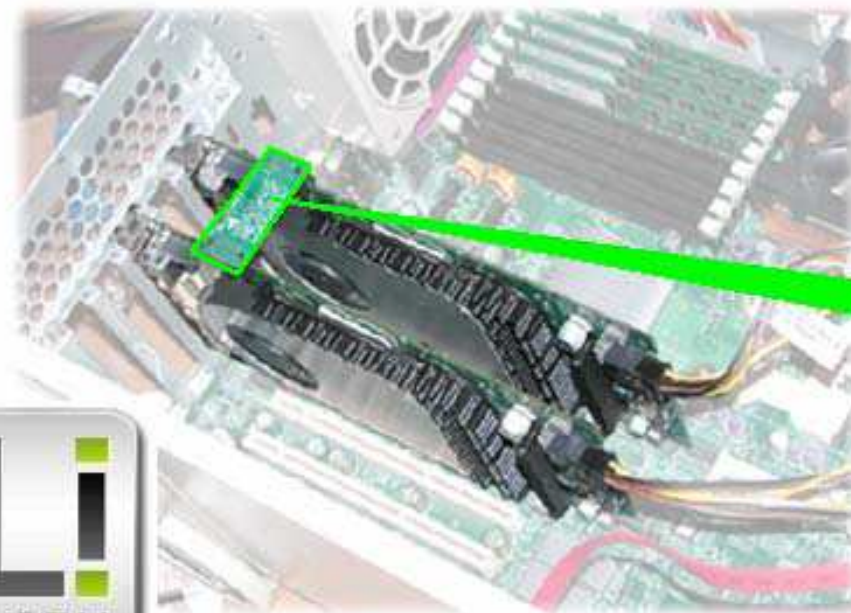


PCIe:

- Al igual que AGP:
 - Usa una conexión serial → barato, escalable
 - Usa protocolo punto a punto → No se comparte ancho de banda
- Con la diferencia de que:
 - Propósito general, no solo para gráficos
 - Dual-channel: el ancho de banda está disponible en ambas direcciones
- Ancho de banda: $PCIe = 2 \times AGP8x$

Arquitectura multi-GPU

- e.g. La tecnología de NVIDIA Scalable Link Interface (SLI) aprovecha el ancho de banda aumentado del PCIe para acelerar automáticamente las aplicaciones.



Conector SLI

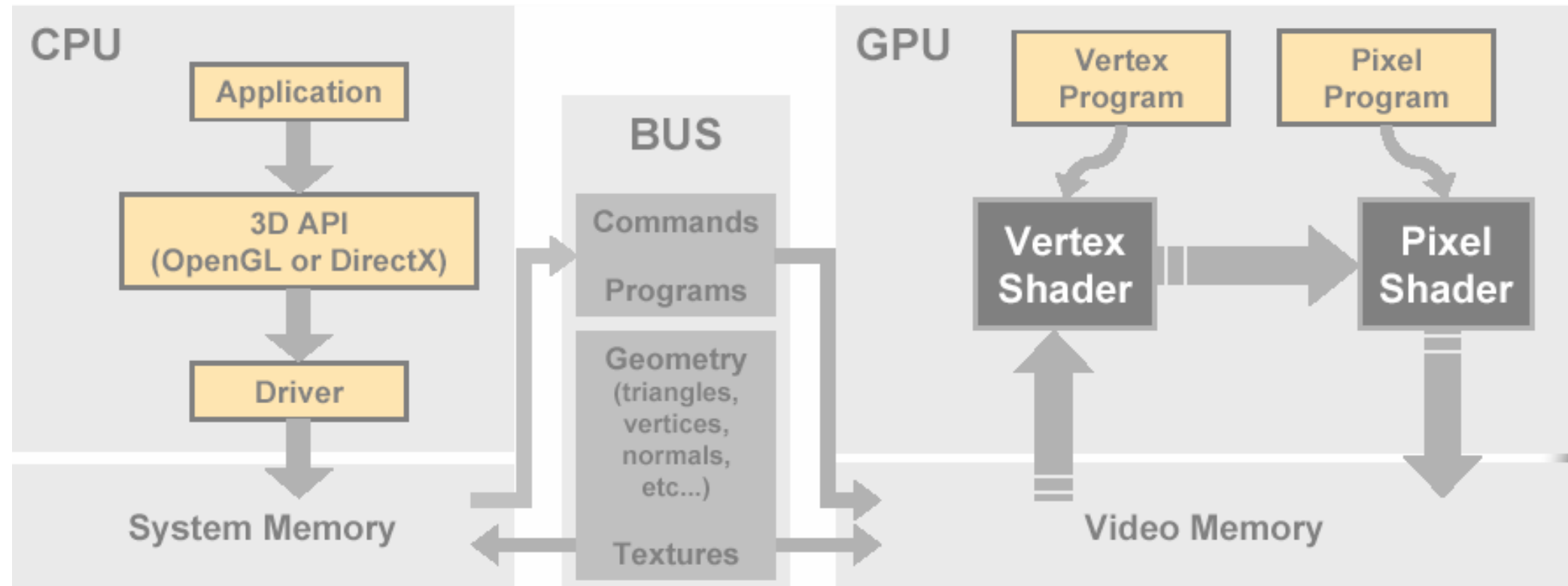


Shader model 3.0

- Shader Model 3.0 significa:
 - Shaders más largos → shading mas complejo
 - Pixel shader:
 - Control de flujo dinámico → mejor rendimiento
 - 32 bits → Menos problemas relacionados a la precisión
 - Vertex shader:
 - Acceso a las texturas → Simulación en la GPU
 - Frecuencia del vertex buffer → instanciado eficiente de la geometría

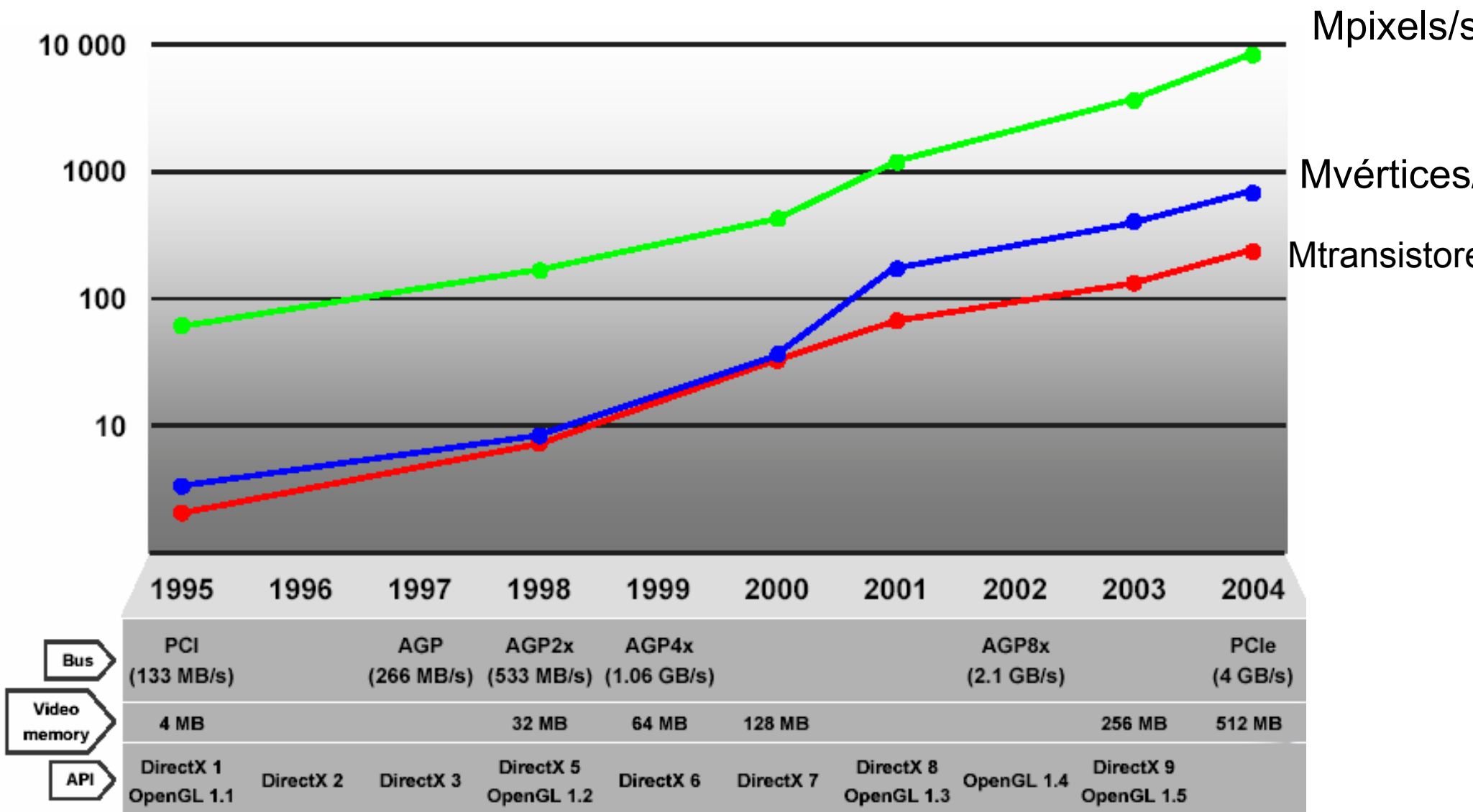


Arquitectura gráfica de una PC



- La aplicación, el API 3D y el driver son escritos en C o C++
- Pixel and pixel shaders usan programas escritos en un lenguaje de alto nivel (Cg, DirectX HLSL, OpenGL Shading Language)

Evolución del rendimiento



El futuro

- Un modelo general y unificado para la programación a niveles de las primitivas, vértices y píxeles
- Grandes, muy grandes cantidades de:
 - Computos usando precisión de punto flotante
 - Memoria de video
 - Ancho de banda entre el sistema y la memoria de video
- Chips más baratos y requerimientos de potencia más bajos