

Creación de gráficos 3D

Katia Leal Algara

Web: <http://gsyc.urjc.es/~katia/>

Email: katia.leal@urjc.es

Dept. Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación (GSyC)
Escuela Superior De Ingeniería De Telecomunicación (ETSIT)
Universidad Rey Juan Carlos (URJC)



Creación de gráficos 3D

- El proceso de creación de gráficos 3D por computadora puede ser dividido en las siguientes etapas básicas:
 - **Modelado.**
 - **Composición de la escena.**
 - **Animación.**
 - **Renderizado.**

Creación de gráficos 3D: Modelado

- La etapa de modelado consiste en dar forma a objetos individuales que luego serán usados en la escena creada.
- Existen diversos tipos de **geometría** para modelar, aunque también se puede usar el **modelado basado en imágenes** (IBM, pasar una fotografía a 3D).

Creación de gráficos 3D: Modelado

Las técnicas más representativas:

- **Modelos representados por polígonos.**
 - Un cubo tiene 6 caras, cada una de ellas es un polígono.
 - Una pirámide se compone de 4 triángulos y una base cuadrada.
 - Una forma redondeada también se representa mediante polígonos, por ejemplo, un balón de fútbol se compone de 12 pentágonos y 20 hexágonos.
- **Modelos definidos por curvas matemáticas.**
 - Una circunferencia se puede representar como una función matemática entre dos variables: X e Y (el conjunto de los puntos de un plano que equidistan de otro).
 - El usuario trabaja con un *programa vectorial* para trazar curvas perfectas en un modelador no poligonal, y también dispone de diferentes tipos de herramientas (NURBS, Spline, Patch, Bezier, etc.) para crear superficies curvas complejas.

Creación de gráficos 3D: Composición

- Distribuir los diferentes elementos (objetos, luces, cámaras...) en una escena que será utilizada para producir una imagen estática o una animación.
- Aspectos que forman parte de la composición de una escena:
 - **Sombra.**
 - **Iluminación.**

Creación de gráficos 3D: Composición

- **Sombra:** definición de la forma de las sombras de los objetos.
 - Para ello se utilizan materiales denominados shaders, algoritmos que controlan la incidencia de la luz combinando texturas con materiales.
- **Iluminación:**

Creación de gráficos 3D: Composición

- **Iluminación:** Creación de luces puntuales, direccionales en área o volumen, con distinto color o propiedades.
 - Las luces tipo **omni** generan rayos de luz en todas las direcciones.
 - Las **luces direccionales** los rayos de luz se dirigen a una sola dirección.
 - Las luces tipo **domo** iluminan a toda la escena.
 - En relación con el color, se puede configurar un ambiente con colores cálidos o fríos, los cuales se consiguen modificando los valores del RGB de cada una de las luces.
 - La **Iluminación Global** (en inglés Global Illumination), es un conjunto de algoritmos que tratan de simular o aproximar cómo una luz, emitida por alguna fuente, rebota en cada superficie de la escena iluminando espacios.

Creación de gráficos 3D: Animación

- La Animación en 3D es un proceso complejo, porque conlleva la realización previa de otros procesos como el diseño y modelado de la imagen a animar.
- Consiste en **la deformación o movimiento de los objetos** de un modelo 3D **a lo largo del tiempo**.
- Para que haya animación, esta deformación o movimiento debe variar en algún aspecto respecto al tiempo: cambio de luces y formas, movimiento de objetos y cámaras, etc.

Creación de gráficos 3D: Animación

Los objetos se pueden animar a partir de:

- Transformaciones básicas en los tres ejes (XYZ), rotación, escala y traslación.
- Modificaciones en formas:
 - Mediante esqueletos: a los objetos se les puede asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. En el proceso de animación el movimiento del esqueleto automáticamente afectará a las partes correspondientes del modelo.
 - Mediante deformadores.
 - Mediante dinámicas para simulaciones de ropa, pelo, etc.

Creación de gráficos 3D: Renderizado

- Rénder se denomina al proceso final de generar la imagen 2D o animación a partir de la escena creada.
- En esta parte se procesan polígonos, sombras, reflejos, iluminación, etc. para generar imágenes realistas.
- El software de rénder puede simular efectos cinematográficos como profundidad de campo o *motion blur* (desenfoque de movimiento).
 - Estas imperfecciones mecánicas de la fotografía física aportan realismo a la escena.
 - Técnicas para simular efectos atmosféricos o naturales como lluvia, humo, fuego, niebla, etc.

Creación de gráficos 3D: Renderizado

- El proceso de rénder requiere *simular una gran cantidad de procesos físicos complejos*.
- A través de los años, la capacidad de procesamiento se ha incrementado permitiendo un nivel superior de realismo en los rénders.
- Uno de los motores de rénder más conocidos y utilizados es V-Ray por su facilidad de configuración y por sus resultados de buena calidad.

APIs para generación de gráficos 3D

- Debido a la popularidad de los gráficos 3D se han creado *APIs* especializadas para facilitar los procesos en todas las etapas de la generación de gráficos por ordenador.
- Estas interfaces proveen un *camino directo al programador para acceder al hardware* de manera abstracta, aprovechando las ventajas de la tarjeta de video.
- Interfaces más populares:
 - **OpenGL**
 - **Direct3D**

Taxonomía de Flynn

- Clasificación de las diferentes arquitecturas de computador:
 - **SISD**: single instruction, single data
 - **SIMD**: single instruction, multiple data
 - **MISD**: multiple instructions, single data
 - **MIMD**: multiple instructions, multiple data

Taxonomía de Flynn - SISD

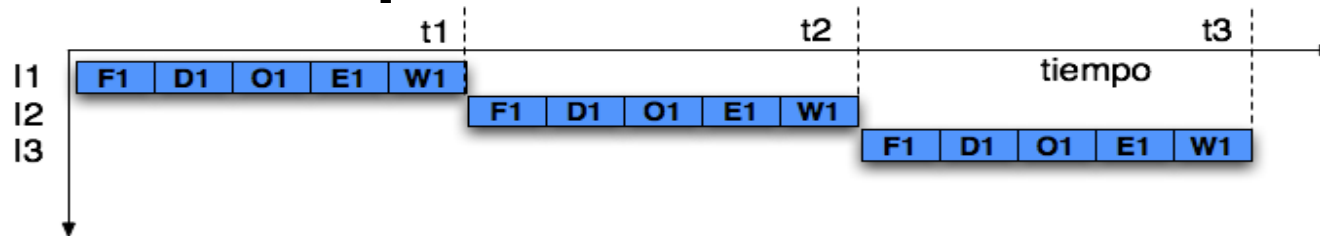
- Computador **secuencial** que no explota el paralelismo en las instrucciones ni en flujos de datos.
- Es la **Arquitectura Von-Neumann**:
 - Un único procesador ejecuta un sólo flujo de instrucciones para operar datos en una única memoria.
 - Se ejecuta una única instrucción y un dato en cada ciclo de reloj.
 - Puede utilizar técnicas de **segmentación** o de pipelining (***paralelismos interno***).
 - Ejemplos: máquinas con monoprocesador como el PC o los antiguos mainframe.

Taxonomía de Flynn - MIMD

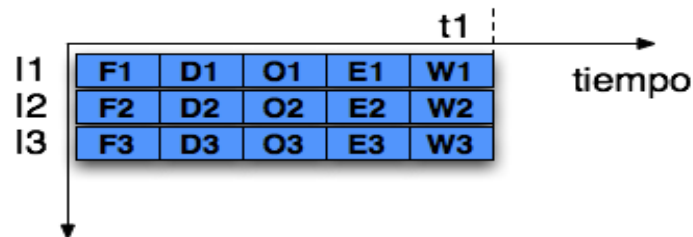
- Las máquinas MIMD tienen un número de procesadores que funcionan de manera asíncrona e independiente para lograr **paralelismo puro**.
- En cualquier momento, cualquier procesador puede ejecutar diferentes instrucciones sobre distintos datos.
- Las computadoras MIMD pueden categorizarse por tener memoria compartida o distribuida, clasificación que se basa en cómo el procesador MIMD accede a la memoria.

Pipelining Vs Paralelismo

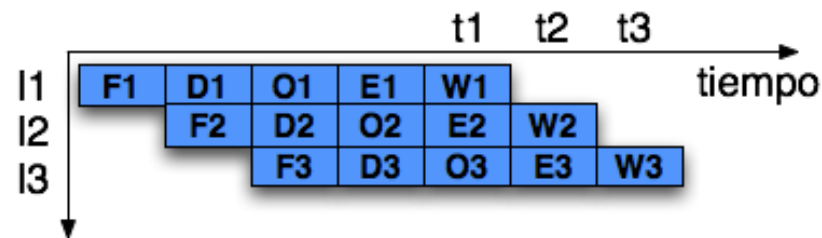
- **Secuencial pura:** un resultado cada 5 ciclos.



- **Paralela pura:** N resultados cada 5 ciclos.



- **Pipeline:** un resultado por ciclo.



Taxonomía de Flynn - SIMD

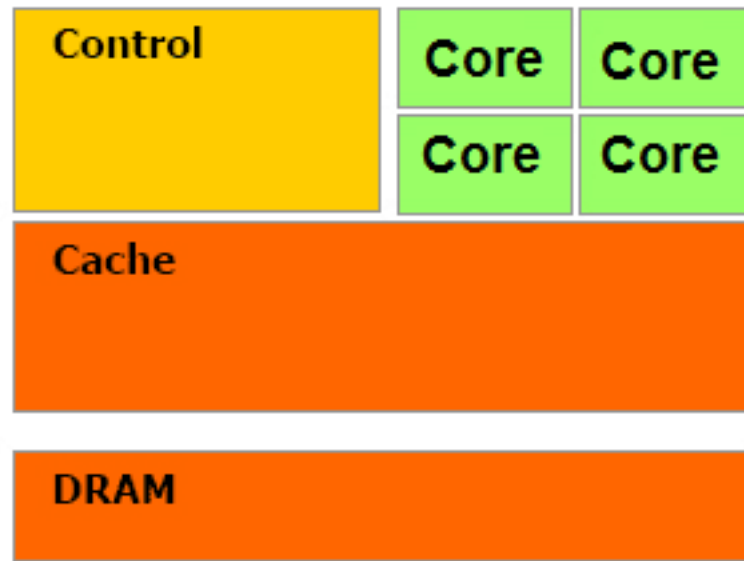
- En el mismo instante de tiempo, una instrucción opera sobre un conjunto de datos.
 - **CPU + juego extendido de instrucciones**, como el repertorio AVX (Advanced Vector Extensions, 2008) de Intel, que se utiliza para álgebra lineal: suma de vectores, multiplicación escalar, multiplicación de matrices, etc.
 - **Basic Linear Algebra Subprograms (BLAS)**: es una especificación (interfaz) que define un conjunto de rutinas de bajo nivel para realizar operaciones comunes de álgebra lineal. Este conjunto de rutinas suponen el *estándar de facto* para las librerías de álgebra lineal.
 - Ejemplos de implementación de la librería BLAS: AMD Core Math Library (**ACML**), **ATLAS**, Intel Math Kernel Library (**MKL**) y **OpenBLAS**.
 - **GPU**

GPU Computing

- Las GPUs se han convertido en el principal complemento de las CPUs - GPU computing or General-purpose computing on GPUs (GPGPU).
- Evolución de la arquitectura de la GPU hacia una plataforma de computación programable masivamente paralela.
- Commodore Amiga – 1985.
- Nvidia GeForce 256 – 1999.
- La industria de los videojuegos ha conducido a una rápida evolución de las GPUs.
- Primera versión de Nvidia CUDA en 2007.
- Deep learning – “big bang” en 2009.

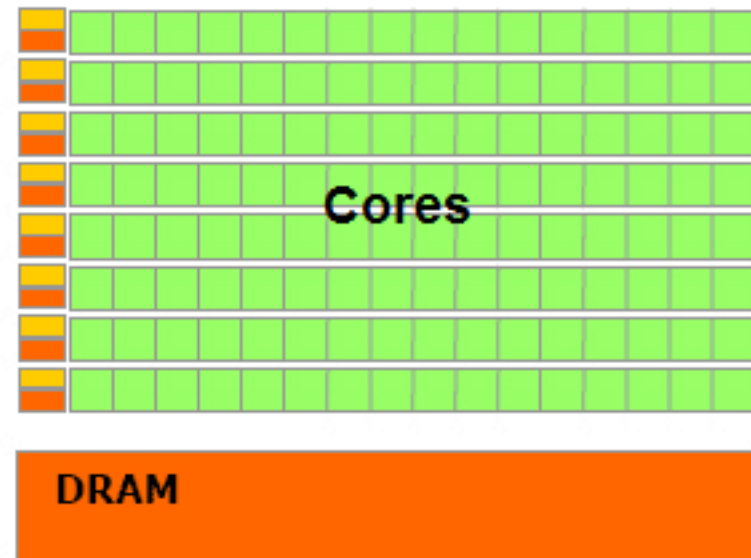


CPU Vs GPU



CPU

Von Neumann, multicore.



GPU

SIMD, manycore.

CPUs: Diseño orientado a la latencia

- **Arquitectura Von Neumann:** frecuencia de procesador alta.
- **Cachés grandes:** reducen el tiempo medio de acceso a la jerarquía de memoria.
- **Control sofisticado:**
 - *Predicción de salto* para reducir la latencia por riesgos de control.
 - *Adelantamiento de datos* para reducir la latencia por riesgos de datos.
- **ALU potente:**
 - Operaciones con poca latencia

GPUs: Diseño orientado al *throughput*

- **Arquitectura SIMD:** frecuencia de procesador moderada.
- **Cachés pequeñas:** para impulsar el throughput de memoria.
- **Control sencillo:**
 - No *predicción de salto*.
 - No *adelantamiento de datos*.
- **ALUs energéticamente eficientes:**
 - Muchas unidades con latencia alta, cauce largo para incrementar el throughput.
 - Requiere una cantidad masiva de hilos para “ocultar” la latencia.

CPUs y GPUs son complementarias

- **CPUs:** para la partes secuenciales en las que prima la latencia.
 - Las CPUs pueden ser órdenes de magnitud más rápidas que las GPUs para código secuencial.
- **GPUs:** para las partes paralelas en las que gana el throughput.
 - Las GPUs pueden ser órdenes de magnitud más rápidas que las CPUs para código paralelo.

Problemas adecuados para GPUs

1. Requerir de cálculos complejos sobre grandes volúmenes de datos.
2. Presentar alta intensidad aritmética (ratio operaciones ALU Vs memoria).
3. Permitir un paralelismo significativo.
4. Ser significativamente más dependiente del throughput general que de la latencia de cada operación.