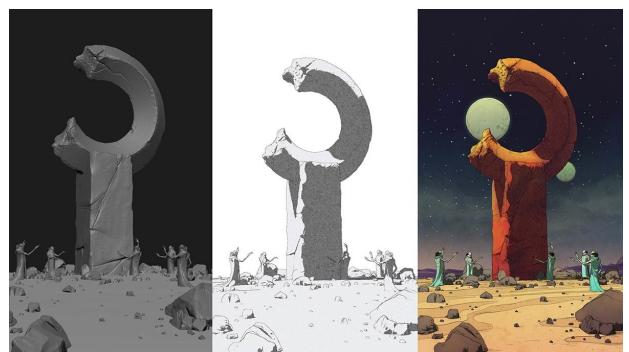
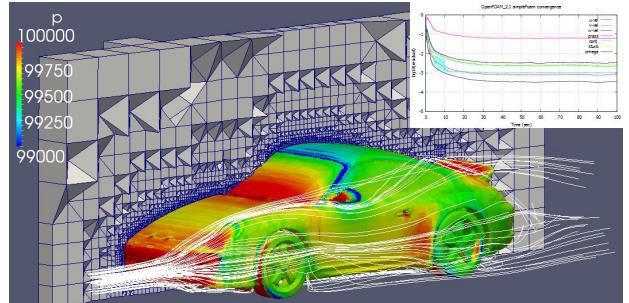
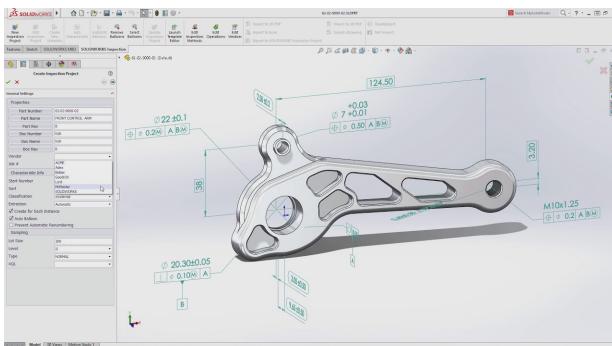


Introducción a la Computación Gráfica

La computación gráfica es el conjunto de técnicas utilizadas para comunicar información de forma gráfica, usualmente tridimensional. Requiere organizar adecuadamente la información y generar automáticamente, a partir de la misma, una imagen (o una secuencia animada)¹.

El contenido transmitido puede ser de cualquier índole, por ejemplo:

- una animación o un juego para la industria del entretenimiento
- la descripción gráfica de objetos reales y sus detalles, para comprenderlos o fabricarlos
- la visualización de datos científicos o técnicos, y/o grandes volúmenes de información (*big data*)
- otros renderizados no fotorealistas (NPR).



¹Excluiremos a las herramientas de retoque fotográfico o de análisis de imágenes (interpretación automática del contenido de una imagen), ya que se podría decir que son la inversa de la computación gráfica.

Las problemas generales que se abordan en computación gráfica son:

- Cómo modelar la escena:
 - Definir analíticamente y organizar los objetos virtuales que se quieren mostrar.
 - Asignar detalles superficiales mediante colores, imágenes (texturas) y/u otras propiedades.
 - Ubicar todo el modelo en un ambiente, junto con la cámara y las luces virtuales.
- Cómo procesar la escena para generar una imagen.
- Cómo alterar la escena e iterar para construir cada nueva imagen:
 - Definir analíticamente los movimientos autónomos del modelo, cámara o luces.
 - Interactuar con el usuario para guiar los movimientos interactivos.
 - Interactuar con el usuario para reconstruir alguna parte del proceso o métodos.

Las historias de la Computación Gráfica que suelen encontrarse en web, casi siempre se refieren al hardware. Comienzan en los años 60 con los primeros intentos de dibujar utilizando computadoras ² y con la invención de los primeros dispositivos apropiados para esa tarea: el monitor gráfico, el lápiz óptico, el mouse³ y el plotter. Pero de la misma época son los intentos por remplazar los planos del dibujo técnico por medio de representaciones analíticas que puedan trasmisitirse mediante ecuaciones.

Para definir las formas industriales y comunicarlas se utilizan los planos acotados del dibujo técnico clásico; pero en los años 60 se pusieron de moda las formas suaves y orgánicas, con curvas arbitrarias, que no se pueden acotar. Esas formas suaves ya se utilizaban en la arquitectura moderna y en las industrias naviera y aeronáutica, pero en todos estos casos, se fabricaban una o pocas piezas en un mismo lugar; con patrones recortados de madera o de papel, como en la industria textil y la costura. Muy diferente es el caso de la industria automotriz, que requiere comunicar un diseño suave hacia múltiples puntos de fabricación y control de medidas. Así es que De Casteljau en Citroën y Bézier en Renault llegaron a un formato, casualmente equivalente, que permite representar curvas y superficies paramétricas de forma simple y eficiente, y que aún se utiliza.

Conocer la historia es importante pues gran parte de la historia *antigua* (TV, imprenta, máquinas de escribir, etc.) aún condiciona los métodos y el lenguaje que se utilizan. La historia actual es la de una guerra económica entre corporaciones que fabrican el hardware especializado (Nvidia, ATI, Intel) y que tratan de imponer sus criterios en la estandarización de los métodos que mejor aprovechen el hardware que venden. Las grandes empresas de software (Apple, Microsoft, Google) tampoco están fuera de esas batallas pues proveen algoritmos e interfaces que quieren popularizar y estandarizar para seguir compitiendo en el mercado.

Estas guerras económicas generan un avance tan acelerado que resulta imposible mantenerse al día a quien no está trabajando en ese tema o quien solo utilice una fracción de los nuevos desarrollos mientras

²un hito notable en esta historia es el desarrollo de [Sketchpad](#) en 1968 por parte de Ivan Sutherland. Este software es considerado por muchos el primer CAD, y le valió a su autor, entre otras cosas, un premio *Turing* (el *Nobel* de la informática).

³Douglas Engelbart (otro ganador del premio *Turing*) es conocido por ser el inventor del mouse, pero en realidad sus invenciones llegan muchísimo más lejos. En 1968 presentó un sistema de software y hardware en lo que con el tiempo llegaría a ser considerado [La Madre de Todas las Demos](#). El video de aquella demostración está disponible en muchos sitios de Internet, y en él podrán reconocer, además del 1er mouse, las semillas de ideas como la videoconferencia, el procesamiento de texto, los hipervínculos, la edición colaborativa en tiempo real, el control de versiones, entre otras.

trata de mantener funcionando lo que vende. Es decir que tenemos las ventajas de una gigantesca velocidad de avance y la desventaja de la eterna antigüedad de los conocimientos adquiridos. Para evitar la obsolescencia precipitada, dedicaremos nuestro limitado tiempo al desarrollo de contenidos fundamentales, aquellos que están en la base de cualquier método o desarrollo pasado, presente o futuro, y nos contentaremos con una praxis exemplificadora, pero no necesariamente con las herramientas que se utilizan en producción, sino algunas más básicas, fáciles de aprender y que sean parte integrante de los sistemas más complejos.

En la actualidad son las GPUs o *graphic processing unit*, las encargadas de la mayor parte del cómputo masivo involucrado en la generación de la imagen. En determinado momento, los datos preparados en la CPU pasan a la GPU para seguir un protocolo de cálculos masivos y en paralelo. La generación de gráficos implica el modelado para simular la realidad física que daría lugar a esa imagen, la conformación de los objetos mostrados, su compleja superficie, la interacción entre ellos y con el ambiente, y en particular con la luz. Pero también se modela una realidad virtual totalmente irreal, que requiere complejos personajes ficticios en un ambiente también complejo y que mezcla elementos de la realidad con la fantasía libre del autor.

Esos procesos involucran una miríada de cálculos, pero permitieron pasar de los *dibujos animados*, hechos cuadro a cuadro por el artista, a la *animación 3D* en la que el artista utiliza complejas herramientas de software y hardware para definir los personajes y acciones, que se preparan de antemano y luego se procesan en clústeres de cálculo masivo y en paralelo. Aun así, tanto las placas como los modernos algoritmos ayudan también en el dibujo animado y en la presentación interactiva.

La presentación interactiva es aquella en la que el usuario interactúa constantemente con la escena que se muestra. Esto requiere que los cómputos se realicen "en tiempo real". Esto es, que el resultado sea prácticamente instantáneo. Una medida habitual del desempeño suele ser la cantidad cuadros (frames) por segundo, FPS. Esto es, cuantas veces por segundo el sistema es capaz de actualizar la información que muestra. Con unos pocos frames por segundo alcanza para que exista cierta interacción en tiempo real, pero usualmente se requieren algunas decenas (digamos que más o menos a partir de 20FPS, idealmente al menos 60FPS) para que el efecto de movimiento se perciba como fluido. El ejemplo más crítico son los videojuegos, donde se pretende cada vez mayor velocidad y potencia de cálculo para generar escenas vívidas, movimientos más suaves y ambientes más complejos y realistas.

1. Un poco de Hardware

Para comunicar gráficos se utiliza hardware específico en la entrada de datos, en el cómputo y en la presentación. Necesitamos conocer unos pocos detalles técnicos, sobre todo los que condicionan o son condicionados por el software, la historia y las técnicas de transmisión de los datos entre el hardware y el software.

Un ingeniero que pretenda hacer software duradero, debería ser capaz de comprender cualquier desarrollo de hardware futuro y, al menos en principio, preverlo. Además, es imprescindible que no se deje engañar por publicidades engañosas de hardware o desarrollos improbables.

Analizaremos algunos dispositivos de salida para la presentación gráfica; dispositivos de entrada de los datos necesarios para crearla, y algunos dispositivos de hardware intermedio que realizan los cálculos.

1.1. Dispositivos de Salida

Para la presentación gráfica inmediata, mostrada en tiempo real, se utilizan básicamente el monitor o *display* y el cañón proyector. La imagen impresa en forma más duradera se realiza mediante una impresora o un plotter.

El término *monitor* proviene de los estudios de TV, era un pequeño televisor para monitorear el resultado que salía *al aire*, esos mismos aparatos se utilizaron en las computadoras y por ello persiste el nombre. También se utiliza mucho el término *pantalla*, que proviene del cine, pero una pantalla no emite luz, refleja la luz proyectada. Quizás el término más apropiado sea *display* porque es un dispositivo que *muestra* una imagen, incluyendo los actuales displays de los pequeños dispositivos móviles, como los pads y los teléfonos.

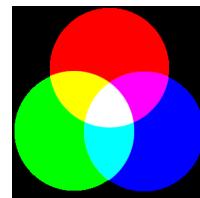
1.1.1. Imagen Digital

La imagen digital es en la mayoría de los casos una disposición densa de pequeños puntos de color llamados píxeles. Tanto la imagen como el color son discretos; es el cerebro humano el que interpreta el resultado como un continuo suave y tanto más suave cuanto mayor la resolución (cantidad de píxeles) de la imagen y la profundidad (cantidad de bits disponibles para representar un color).

El término *resolución* proviene de la microscopía, en ese ámbito es la capacidad para diferenciar dos puntos o líneas muy próximas. Por extensión, la jerga informática lo utiliza de un modo distinto e inverso, aquí es la cantidad de píxeles (ej.: 1024x768) o, a veces, la densidad de puntos impresos (ej.: 600dpi); en todos los casos mide la capacidad de mostrar detalles finos de la imagen.



Zoom de una foto de un monitor led



Combinaciones de rojo, verde y azul puros

En casi cualquier display, el color de un pixel es la combinación de tres luces: roja, verde y azul (RGB); cada una con una intensidad variable, tomando típicamente uno de 256 posibles valores (8bits). Esas *lamparitas* están muy próximas (subpixel) y la vista las interpreta como un solo color de una gama muy amplia de posibles colores.

Cuando un material absorbe energía, una fracción se convierte en calor y el resto se devuelve como luz. Así se dan varios fenómenos como la reflexión, transmisión y emisión de luz. La transmisión (vidrio rojo) y

la reflexión (pintura roja) son fenómenos en que la luz emitida es una fracción que ya existía en la energía lumínica recibida. Si la luz emitida no es parte de la luz recibida, tenemos la fluorescencia inmediata (lámpara fluorescente, jabón para la ropa *más blanco*) y la fosforescencia lenta (brilla en la oscuridad). Cuando la fuente de energía no es luz, decimos sencillamente que el material emite luz (lámpara incandescente, fuego, luciérnaga, LED).

1.1.2. Cañones y Monitores

Los primeros monitores eran tubos de rayos catódicos o CRT. El CRT es un recipiente de vidrio al vacío, que en la parte trasera tiene un cañón que emite un haz de electrones (rayos catódicos) que impacta sobre la pared frontal o *pantalla* donde una sustancia (fósforo) emite luz en el punto donde chocan los electrones. Para que el haz se desvíe de la trayectoria recta se utilizan dos campos magnéticos deflectores, uno horizontal y uno vertical. El haz es desviado por las bobinas deflectoras para *barrer (scan)* o recorrer la pantalla en forma secuencial. Comienza en el ángulo superior izquierdo de la pantalla y barre una línea horizontal, al terminar el *barrido horizontal* de la línea (*scan line*), el haz se desvía rápidamente al principio de la siguiente y al terminar el *barrido vertical* se desvía rápidamente al punto inicial. Es la persistencia de la sensación de luz en el ojo humano la que logra que el CRT parezca mostrar una imagen continua, aun cuando hay solo un punto iluminado en cada momento y la pantalla se barre entera unas veinticinco veces por segundo.

La *señal* de video que modula la intensidad del haz es una intensidad (voltaje) variable en el tiempo, mientras el haz barre la pantalla. Esa es exactamente la forma que se ha adoptado para almacenar una imagen en un archivo o en un buffer de memoria de color o color buffer. Este contiene una intensidad (una terna RGB) para cada pixel en una línea, comenzando arriba a la izquierda y barriendo todo el ancho, luego la línea siguiente y así hasta la cantidad de líneas de altura de la imagen. El peso y el volumen del tubo necesario para establecer y curvar el haz, el gran consumo de energía y un falso miedo a radiaciones nocivas, terminaron con el CRT.

Luego vinieron los displays de cristal líquido LCD (liquid cristal display). Un cristal líquido es una dispersión muy densa de grasa en agua u otro líquido lipofóbico; donde las moléculas largas "cristalizan" en cilindritos agrupados. La estructura se alinea mediante campos eléctricos y polariza la luz que atraviesa la sustancia. Mediante un sándwich de dispositivos, desde una luz trasera o un espejo, polarizadores, circuitos transparentes, cristal líquido, y protector frontal, se logra atenuar en mayor o menor medida la luz de cada pixel. Este tipo de monitores utiliza capas de circuitos transparentes para llevar la energía que altera cada pixel.

Luego surgió brevemente el plasma, que se utilizaba solo para los grandes televisores, porque cada pixel es una lámpara fluorescente. El plasma se produce por una descarga eléctrica (un rayo) en un gas que se ioniza y emite luz ultravioleta, tres tipos de fósforo que cubren cada lámpara se encargan de transformar el UV en luz R, G o B.

El LED en cambio es un diodo emisor de luz. Un diodo es un dispositivo semiconductor que deja pasar corriente en un solo sentido. Cualquier diodo emite luz, normalmente la luz es de muy baja energía, es decir: calor o infrarrojos. Los LEDs infrarrojos se usan en los controles remotos. Con materiales que

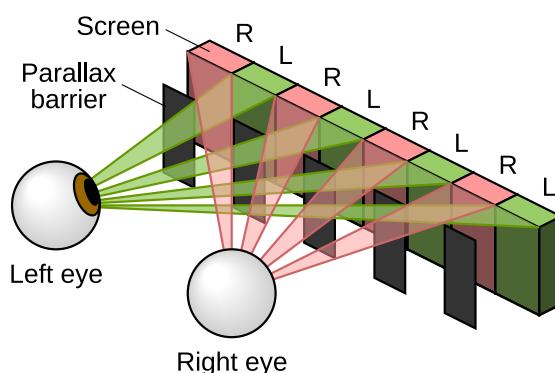
propician saltos de mayor energía, la luz emitida pasa a ser visible, desde el rojo al violeta. Prácticamente toda la energía del LED se transforma en luz monocromática y esta es la tecnología más promisoria en la actualidad, por el gran ahorro de energía. Los monitores y televisores que conocemos como LED son en realidad LCD con luz trasera (backlight) producida mediante LEDs con gran ahorro de energía. Pero existe una tecnología de LEDs orgánicos (OLED) en un sustrato plástico y transparente que ya se utiliza para displays y que pueden hacerse flexibles.

Los otros dispositivos para mostrar gráficos en modo inmediato son el cañón proyector y los lentes estereoscópicos de realidad virtual. Los lentes son un pequeñísimo monitor para cada ojo y el cañón también funciona con un pequeño monitor LCD transparente, para cada color. El gran problema del cañón son las altas temperaturas, sobre todo mientras muestra negro, pues algún dispositivo interno tiene que recibir y evacuar la energía, sin emitir luz, los pequeños LCD se deterioran rápidamente y están siendo reemplazados por nuevas técnicas micromecánicas (microespejos DMDs) y las lámparas delicadas y de alto consumo por LEDs.

1.1.3. Dispositivos "3D"

La sensación visual de tridimensionalidad la obtenemos por tener dos ojos separados que ven la misma escena. El cerebro reconoce las diferencias entre ambas imágenes y determina con ella la profundidad. El método más antiguo y sencillo para reproducir esa sensación consiste en presentar una imagen diferente a cada ojo. Existen, hace más de un siglo, los estereoscopios de dos fotos separadas, que se dirigen a cada ojo; eso se ha replicado en los lentes de realidad virtual. En el cine 3D se mezclan las dos imágenes en la pantalla, pero se separan mediante lentes. Puede ser una imagen de dos colores y lentes que filtran cada uno, dos polarizaciones cruzadas (el caso más habitual) y lentes que las filtran, o imágenes en secuencia, una para cada ojo con lentes electrónicos shutters (IMAX) que son LCDs transparente o negro en sincronía con la secuencia.

La tecnología para estereoscopía sin lentes es similar a las tarjetas que muestran dos imágenes dependiendo del ángulo. Un método es la barrera de paralaje, se dispone de una serie de líneas opacas al frente de dos imágenes formadas por líneas entrelazadas, las líneas opacas tapan las líneas de una imagen a un ojo y las otras al otro ojo. Se pueden usar también líneas de lentes cilíndricas (lenticulares) que amplifican el paralaje.

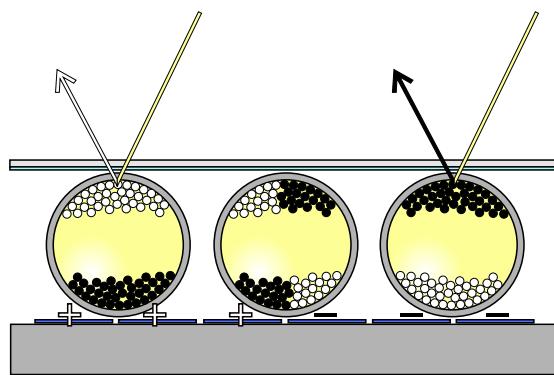


En cuanto a monitores 3D, pero no estereoscópicos (sino que puedan verse desde cualquier ángulo como si generasen una escultura virtual), están aún en el terreno de la ciencia ficción y la publicidad engañosa. Si bien hay tecnología para iluminar un punto (laser-plasma) del volumen virtual que se muestra; no puede opacar la luz proveniente de otros puntos (occlusión y paralaje) de modo que se ven “puntos en el aire”. Hay técnicas que hacen girar rápidamente un monitor, mostrando una imagen en cada dirección, sirven para un recorrido cilíndrico, pero no esférico. También se puede hacer un rápido barrido plano con una superficie semi-reflectiva como un Mylar, que sube y baja muy rápidamente y al que se le proyecta un corte (tomo) en cada posición.

1.1.4. Tinta electrónica

Un desarrollo intermedio entre la imagen inmediata de un display y la imagen impresa permanente es el de los libros electrónicos o ebooks, como kindle, nook y papyre, con tinta electrónica E-Ink. Esta tinta electroforética (EPD) permite un gasto mucho menor de energía y la posibilidad de ser leídos al aire libre.

La mayoría funcionan en escalas de grises, pero ya se consiguen (muy caros) en colores. La energía se utiliza para cambiar el color de cada pixel, que queda sin consumo subsecuente; por lo tanto, un ebook puede quedar encendido y mostrando una página por semanas. Una desventaja es la lentitud del proceso de migración de la tinta (electroforesis) que impide aun aplicaciones como la reproducción de video. Sin embargo, por su bajo consumo y la no emisión de luz, se utilizan en e-readers (lectores de libros electrónicos como los dispositivo Kindle de Amazon).



1.1.5. Impresoras y Plotters

Las impresoras son dispositivos que fijan tinta en un papel (u otro substrato) proporcionando un grabado duradero de las imágenes y el texto. La imagen impresa refleja selectivamente la luz ambiente; el papel blanco refleja toda la luz y la tinta negra absorbe toda la luz; las de color absorben todo, excepto el color de la tinta.

La impresión electrónica comienza con las máquinas de escribir eléctricas, con caracteres grabados en metal que impactaban sobre una cinta entintada delante del papel. La impresión en computación evolucionó a partir de eso y por mucho tiempo se dedicó solo a imprimir caracteres y aun los dibujos se hacían

así (ascii art).

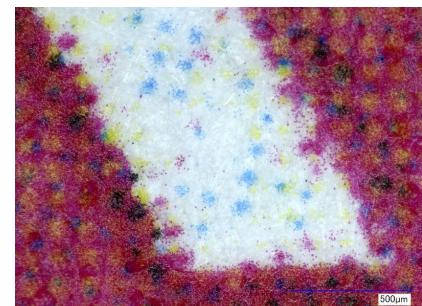
Una máquina de escribir eléctrica podía imprimir un carácter a la vez mediante un cabezal giratorio e intercambiable (font) con las letras grabadas; pero la impresora de línea (LP) imprimía una línea de texto a la vez, con mucha mayor velocidad, pero también complejidad y costo. Cuando se popularizó la PC, la impresora volvía a ser de caracteres, pero ya de matriz de puntos; era una matriz de 3x3 o 3x8 agujas que se disparaban sobre una cinta entintada para imprimir un carácter o ya un gráfico aceptable. Con una cinta tricolor se lograba imprimir en colores. Estas impresoras suelen usarse aún para la impresión de formularios continuos.

Luego aparecieron las impresoras térmicas, el cabezal era también una matriz de puntos, pero que calentaban un papel cubierto de sustancias termosensibles, que reaccionan al calor y cambian de color o bien micro cápsulas de cera que se derriten y liberan la tinta que contienen. Aún se utilizan en tickets de cajeros automáticos.

Una tecnología remanente del pasado son las fotocopiadoras xerográficas. Tienen un rodillo que se carga al recibir la luz y con esa electricidad estática recoge toner, un polvo especial con carbonilla, para luego fundirlo presionado contra el papel. Con muy poco esfuerzo, ya en 1969, se transformó ese aparato para funcionar como impresora láser, el láser es el encargado de generar la luz que requiere cada punto del cilindro.

Para el trazado de líneas continuas como en el dibujo técnico se utilizan los plotters. Son robots que recogen una pluma de tinta, seleccionada por su grosor o su color y trazan curvas analíticas definidas en forma paramétrica. La forma paramétrica es un par de funciones que definen el movimiento de un motor horizontal y uno vertical en función del tiempo. Cada carácter y cada línea deben establecerse de ese modo, aun los movimientos con la pluma levantada. Hewlett-Packard desarrolló el lenguaje HPGL para manejar estos dispositivos mediante secuencias de comandos (PenUp, PenDn; GoTo(x,y), etc.). Hoy se utilizan muy poco para dibujo, pero subsisten en otras tareas, como ser el corte de chapas mediante un soplete, un láser o un chorro de agua; o bien con cuchilla, para cortar cueros, telas, cartones o autoadhesivos.

La tecnología que remplazó a las impresoras de matriz de puntos y a los pen plotters, es la de chorro de tinta; donde el cabezal eyecta pequeños puntos de color sobre el papel. Se utilizan tintas cian (celeste), magenta (fucsia), amarilla y negra. El cian absorbe el rojo, el magenta absorbe el verde, el amarillo absorbe el azul y el negro absorbe todo. Combinando las proporciones de tintas mediante un puntillado de puntos solapados y manejando además el tamaño y el grado de dispersión de los puntos se logran los distintos colores que puede proveer una impresora.



zoom de una foto de una impresión

También existen impresoras tridimensionales que aportan o solidifican material en sucesivas capas, para formar una pieza tridimensional. Pero en esta misma dirección podría considerarse que toda máquina de control numérico computarizado o CNC es una impresora tridimensional. Las máquinas-herramientas parten de un trozo de material, que se elimina por partes, hasta dejar la pieza terminada (ej.: el torno de maderas). La única diferencia conceptual es que las impresoras aportan material en un proceso aditivo y las máquinas clásicas lo desbastan en un procedimiento sustractivo. No hay ninguna razón obvia para distinguir los métodos, pero en esta materia nos concentraremos en la comunicación gráfica y no en la producción de piezas, prototipos o modelos. El término "impresora 3D" proviene del primer intento, con una impresora modificada para inyectar, en lugar de tinta sobre papel, un adhesivo sobre una cama de polvo, una " impresión" por cada capa.

1.2. Dispositivos de entrada

Los dispositivos clásicos de entrada de datos son el teclado y el mouse; pero todo dispositivo capaz de comunicar datos desde el ambiente a la computadora es un mecanismo potencial de introducción de datos. También se pueden ingresar imágenes y sonidos, como datos para condicionar los gráficos.

El teclado es prácticamente el de una vieja máquina de escribir y ha resistido todas las amenazas incumplidas de entrada alternativa por medio de la voz, la mente o cualquier otro medio. Sólo se amplió la cantidad de teclas y se redujo muchísimo el esfuerzo de presión, hasta virtualizarlo totalmente en las pantallas táctiles.

El mouse está siendo remplazado por los dedos en los touch-pads o touch-screens, pero en esencia son todos mecanismos y dispositivos señalar un punto en el display; aunque el uso de los dedos permite ahora la utilización de "gestos" (patrones de movimientos relativos de dos o más dedos en simultáneo).

La pantalla de cualquier dispositivo moderno está repleta de "metáforas gráficas" (widgets); por ejemplo, tenemos los sliders o deslizadores, los menús (gráficos o no), cuadros de diálogo, cursores, etc. y todos se manejan mediante operaciones que nos resultan estándar y obvias, pero son convenciones con raíces históricas: comenzando por la letra A que es un gráfico convencional para representar un sonido, un diskette para grabar un archivo o una casa para volver al inicio. El cursor muestra el punto que estamos señalando en la pantalla, pero sin hacer nada hasta hacer clic con algún botón del ratón. El cursor es una metáfora gráfica, el clic una metáfora auditiva (onomatopeya) y el ratón también es metafórico.

Para el ingreso de imágenes digitales, el primer sistema fue el scanner que lee una línea por vez; ahora se está remplazado por las cámaras digitales. El micrófono es otro dispositivo de entrada, pero no suele usarse para definir los gráficos. La interpretación de imágenes y sonido pertenecen al área del Análisis de Señales; aunque se entrecruza en muchos aspectos con la Computación Gráfica, podemos decir que son técnicas inversas en el sentido de que una genera y la otra interpreta las imágenes.

Antes del mouse, ya existía el lápiz óptico, que sobrevivió algunos años en las pistolas de los videojuegos. Dependía del monitor CRT; pues una vez sincronizado con el barrido, al recibir un pulso de luz en el fondo de un tubo, se sabía desde qué punto provenía. Ya no puede utilizarse en los modernos monitores sin barrido.

Un sobreviviente del pasado es el joystick que, como su nombre lo indica, fue creado para los juegos. Tanto el de palanca (stick) como el mando con flechas, están siendo remplazados por acelerómetros, dispositivos de telemetría y de balance (Wii) o la interpretación de los movimientos del cuerpo humano (Kinect). Todo parece indicar que el camino es la interpretación del movimiento humano, dedos, manos, cuerpo, cabeza y aun los ojos; en esa misma dirección, quizás llegue la lectura de los labios. Las técnicas más nuevas de introducción visual de datos se deben a la existencia de dispositivos para la entrada de imágenes y a la velocidad de procesamiento e interpretación de las mismas. No resultaría extraño ver en breve que las pantallas de televisión y computación (juegos) tengan dos o más cámaras separadas e interpreten gestos en tres dimensiones.

1.3. Unidad de procesamiento gráfico o GPU

Antiguamente se conocían como *placas gráficas* o *placas aceleradoras de video*, pero desde el año 2000 aproximadamente se utiliza el término *GPU* (Graphic Processing Unit) para resaltar la programabilidad de los procesadores gráficos. Hoy, hasta los teléfonos celulares, contienen una GPU.

La GPU es una parte esencial del hardware de visualización en modo inmediato, pero la sepáramos del monitor y dejamos para lo último por su importancia capital en la CG. Como procesador, la GPU es una colección de muchísimos procesadores en paralelo para operar mayormente con matrices y vectores en cuatro dimensiones. También suele incorpora hardware especializado en la compresión, descompresión y transferencia de imágenes y video.

En la CPU, se define la escena: el conjunto de objetos a mostrar (su geometría y propiedades), las luces (posición, color, etc.) y la "pose" de la cámara (dónde está y hacia dónde apunta). Esos datos pasan a la GPU, que es la que los procesa generando a partir de ellos una imagen.

Las viejas placas gráficas tenían una funcionalidad fija. Es decir, que para hacer esta conversión de los datos de entrada (la descripción de la escena) a una imagen ejecutaban un programa fijo, grabado en hardware. Con el tiempo, se empezaron a incluir en el proceso algunas etapas programarse mientras las otras permanecían fijas. Actualmente, la mayoría de las etapas fijas han pasado a formar parte de las programables. Una GPU es una unidad de cómputo paralelo, con varias etapas programables por el desarrollador y muy poca funcionalidad fija.

La principal ventaja en el diseño de la arquitectura de una GPU, es que no se requiere tanta compatibilidad hacia atrás (backward compatibility) como en una CPU. Por eso ha evolucionado en forma asombrosa, tanto que hay procesadores en paralelo basados en GPU que no se usan para gráficos e imágenes, sino para el cálculo de propósito general (general purpose → GPGPU) y lenguajes para programarlas como CUDA (de Nvidia), u OpenCL (libre)). Algunas de las supercomputadoras más potentes utilizan GPGPUs para el procesamiento de problemas que requieren cómputo masivo (HPC por High Performance Computing).

Por ejemplo, para la minería de BitCoins se suelen utilizar clústeres con un gran número de GPUs en lugar de CPUs.

2. Rendering

Se denomina rendering⁴ al conjunto de cálculos que realiza la GPU para transformar una escena virtual, normalmente tridimensional, en una imagen o una película. La escena se construye en la CPU y es básicamente el conjunto de objetos o personajes a mostrar (e incluye posiblemente alguna forma de ambiente, cielo, imagen de fondo, etc.), y otros elementos necesarios como las luces, o la cámara. Los objetos incluyen su "forma" (descripción geométrica) y detalles sobre su "apariencia" (generalmente color y/o texturas).

Más allá de los procedimientos individuales, el resultado se suele clasificar como fotorealista o no-fotorealista (NPR: non photorealistic rendering). El primero trata de simular un modelo físico de la iluminación real de objetos y caracteres virtuales, el segundo (NPR) trata de simular el producto de una mano artística, o puede utilizarse para representaciones visuales abstractas, como en la visualización científica o matemática.

Fotorealista



No-Fotorealista



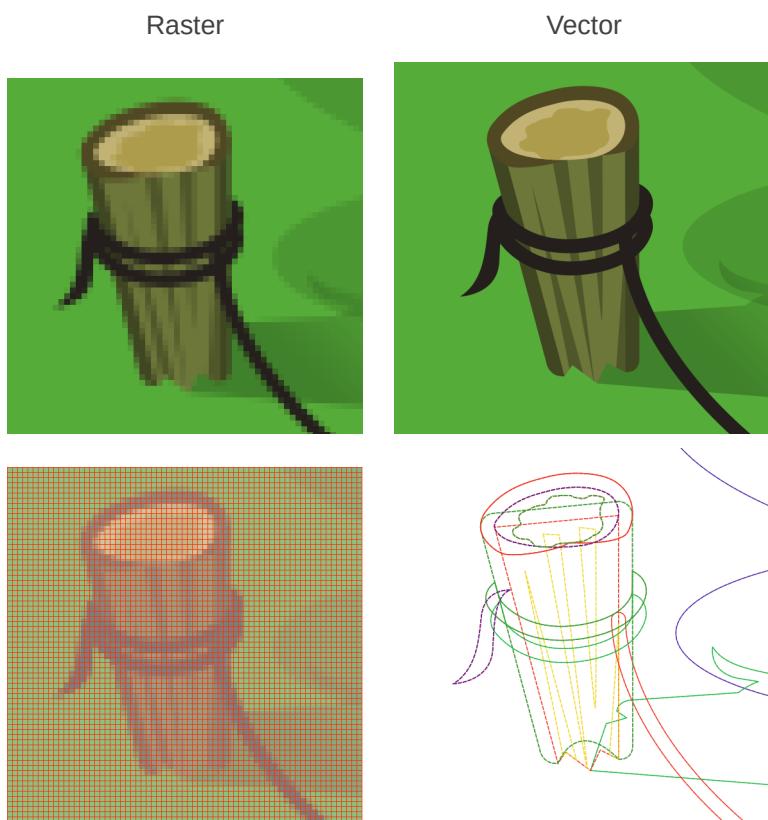
⁴"to render" se traduce "representar" pero en computación gráfica es más común encontrarlo castellanizarse como renderizado.

El renderizado en Computación Gráfica es, entonces, la serie de procesos en la GPU que logran una imagen a partir de una escena tridimensional modelada en la CPU; de modo tal que esa imagen, puesta delante del ojo, muestre la misma escena (realista o no) que obtendríamos con una cámara.

Casi todas las imágenes que se generan mediante la Computación Gráfica son de tipo raster; es decir que contienen información en los pixeles o puntos discretos con color. Estas imágenes se representan usualmente como una matriz cuyos elementos son colores. Pero existe también el renderizado vectorial, donde las formas se representan con funciones matemáticas analíticas, lo que permite un trazado continuo de los contornos y rellena cada interior con un color.

El trazado vectorial clásico en el mundo "físico" es el de los plotters, pero en software se utiliza para almacenar imágenes llenas, como por ejemplo, las definiciones de CorelDRAW o Inkscape de logotipos y otros dibujos sencillos.

Una imagen raster consta entonces de información discreta; al ampliarla se ve "pixelada": los pixeles pasan a ser cuadraditos visibles. La imagen vectorial guarda las ecuaciones de cada borde y el color (constante o variable) de cada forma y puede ampliarse sin perder definición, solo varía el tamaño del dominio del cálculo de la función.



El renderizado es entonces una gigantesca secuencia de procedimientos de cálculo, que requiere software y hardware especial para adecuarse a las exigencias modernas. Tanto es así que en este momento se están desarrollando ambos tipos de herramientas y cualquier aseveración que hagamos seguramente quedará obsoleta en un pocos años. La GPU permite optimizar y paralelizar gran parte del cómputo, pero

aun así hay algoritmos especiales que se incluyen en una amplia variedad de programas para obtener distintos tipos de resultados. No es lo mismo reproducir la iluminación de una superficie suave que la de una rodaja de pan o una polvareda.

Otra clasificación del renderizado se basa en el modo de proceder:

- "offline" o modo "batch": esto es como seguir una receta y cocinar en el horno, una "bacha" o bandeja por vez. Se utiliza cuando prima la calidad del resultado por sobre el tiempo que requiera generarlo, como en las películas de animación, y las complejas imágenes fotorealistas.
- interactivo o "en tiempo real": donde el usuario interactúa constantemente, modificando la escena o las variables. Ejemplos del interactivo son los videojuegos, el diseño (CAD) y la exploración de imágenes médicas o datos técnicos.

Los pasos del renderizado definen el así llamado pipeline gráfico de la GPU, que incluye algunos procesos específicos que iremos viendo a lo largo del curso, pero cuyo significado podemos adelantar. Antes debemos describir como se crean las escenas en CPU y algunas técnicas utilizadas.

2.1. Modelado de los objetos y caracteres

El objetivo primordial en CG es representar escenas virtuales, compuestas principalmente por objetos y personajes tridimensionales. Es importante notar que usualmente solo vemos la superficie y no el volumen de los objetos. Al cortar o romper un objeto físico podemos ver el interior, pero aun así lo que vemos es en realidad una superficie nueva, la del corte. Por lo tanto, en la gran mayoría de los casos, lo que se representa en CG son superficies tridimensionales, la "piel" de los objetos sin nada dentro. La superficie visible se subdivide casi siempre en pequeños triángulos (a veces cuadriláteros) que la aproximan. Los triángulos son las formas primitivas que componen casi cualquier superficie.

Las líneas se utilizan en los dibujos técnicos y diagramas, pero también sirven para simular cabellos, esqueletos manipulables de caracteres animados, trayectorias de movimientos o bien para resaltar los bordes en técnicas de NPR como el cartoonistic rendering (que simula el estilo de los dibujos animados). Las primitivas que conforman las líneas son pequeños segmentos rectos que las aproximan; como los triángulos a las superficies.

También se utilizan puntos aislados como primitivas; estos sirven para modelar sistemas físicos de partículas, normalmente para la representación de humo, llamas u otros elementos complejos representados por puntos.

Independientemente del método que se utilice para modelar los objetos, generalmente terminan pasando por la tubería o pipeline de cálculo gráfico como una sucesión de primitivas: puntos, líneas y/o triángulos. Los vértices definen coordenadas espaciales y llevan algunas propiedades asignadas (por ejemplo: color) que luego se interpolan en el interior cuando se desarma (muestrea) la primitiva en fragmentos, para calcular el color de cada píxel que se ve. Una esfera, por ejemplo, es una colección de triangulitos, y lo mismo sucede con cualquier objeto, por más complejo que sea.

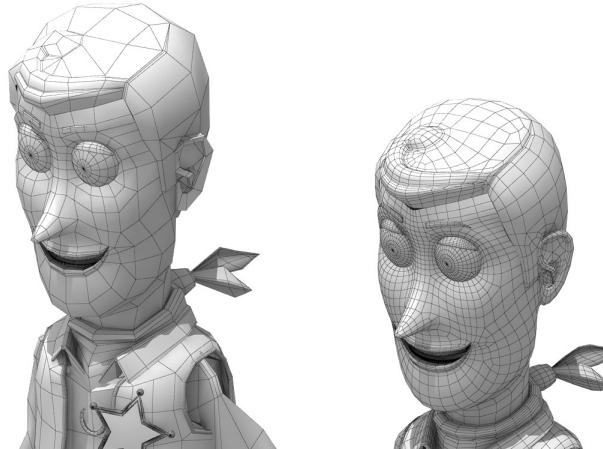
Describiremos a continuación las más habituales para definir un objeto tridimensional.

2.1.1. B-Rep

La representación de la superficie (Boundary Representation o B-Rep) es el método más común y que ya se describió. Utiliza una malla de polígonos sencillos, en general triángulos, que representa la superficie aproximada. Cuando se muestran las aristas sin relleno, se denomina "jaula de alambres" o *wireframe*; si se muestran sólo los vértices suele denominarse "nube de puntos" (*point cloud*).



Render



Wireframes con diferente nivel de detalle

El cálculo de los modelos matemáticos de la física (ecuaciones diferenciales ordinarias o en derivadas parciales) se realiza mediante técnicas de cálculo numérico aproximado, como la de elementos finitos (FEM); allí el dominio de cálculo 3D se divide en una malla de elementos sencillos y de tamaño pequeño pero finito, normalmente prismas, ladrillos o tetraedros, las superficies quedan aproximadas mediante triángulos o cuadriláteros. La computación gráfica ha adoptado y generalizado esta técnica de subdivisión finita, utilizando una malla para el cálculo de la iluminación: la interacción de la luz con la superficie del objeto. Las modernas GPUs pueden procesar miles de millones de triángulos por segundo. Existen otros métodos que veremos a continuación, pero casi todos terminarán en una descripción B-Rep, para poder pasar triángulos por el poderoso pipeline gráfico de la GPU.

La densidad de la malla viene dada por los detalles de la geometría del objeto y la mayor o menor precisión requerida en el cálculo de una buena iluminación. Las mallas se pueden definir mediante primitivas sueltas o pueden provenir de la subdivisión de una porción (patch) de superficie analítica, de acuerdo al nivel de detalle (LOD o level of detail) requerido; o bien puede provenir de un proceso recursivo (subdivision surfaces) que comienza simple y se va complicando a medida que el desarrollador o diseñador introduce detalles o requiere más precisión para la iluminación variable.

Las GPUs permiten optimizar el tratamiento de las mallas. Hay algunas variables globales, otras por objeto, otras por primitiva y también por vértice, estas últimas se pueden interpolar en el interior de cada primitiva. Todas ellas se definen en la CPU y se envían para procesar a la GPU. En las etapas programables de la GPU, el resultado parcial puede "retocarse" por vértice o por pixel mediante software especial para la GPU (*shaders*).



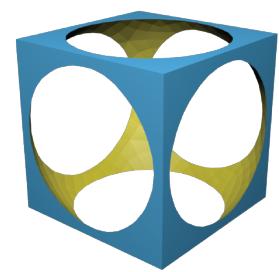
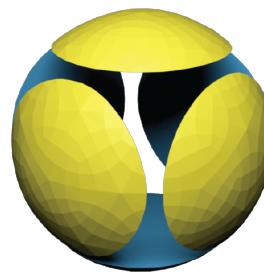
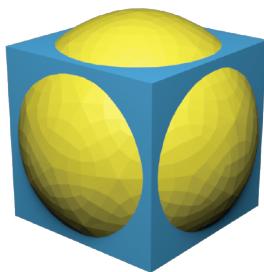
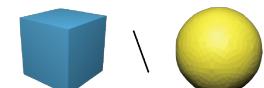
Render



Wireframes

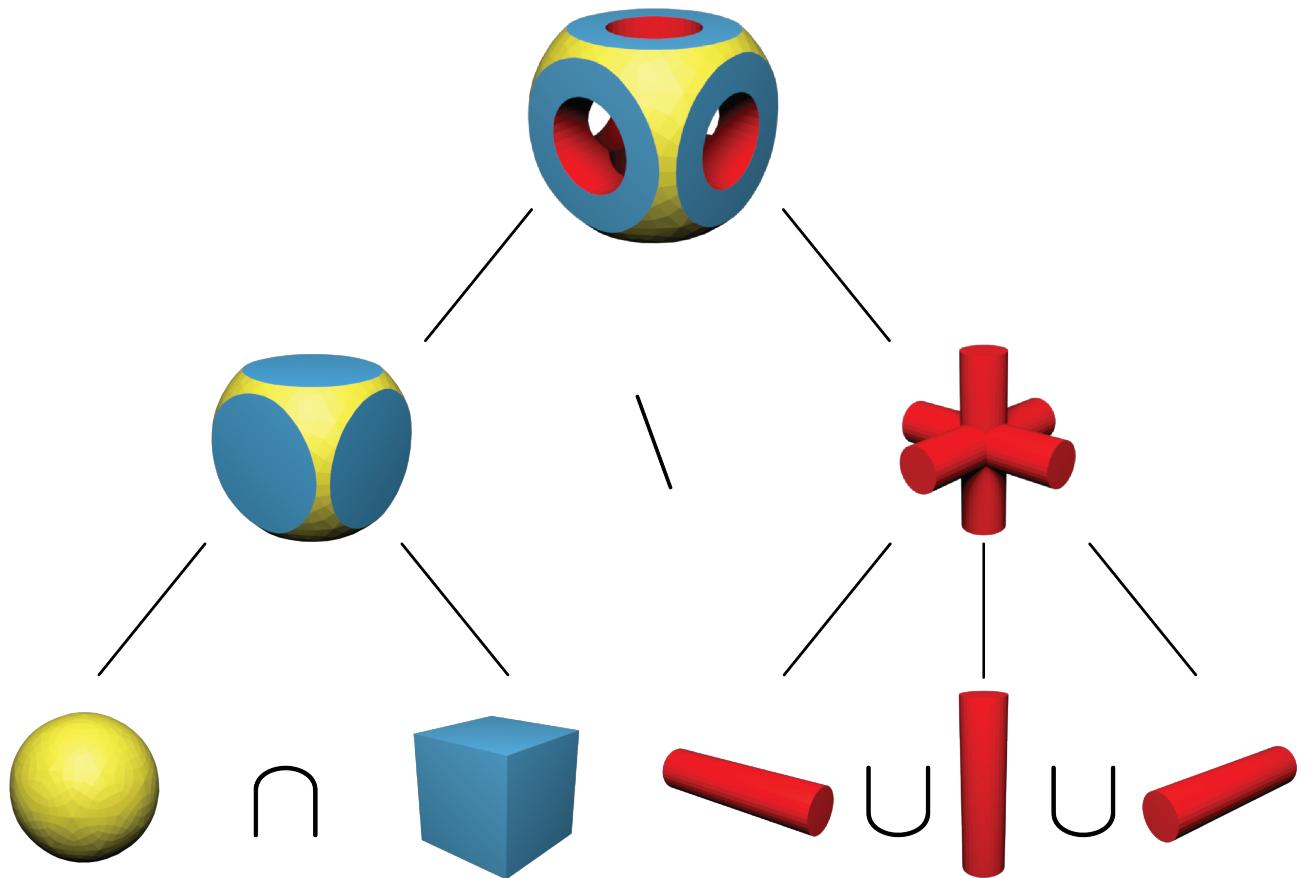
2.1.2. CSG-Tree

El método conocido como Constructive Solid Geometry consiste en utilizar un conjunto reducido de cuerpos simples y analíticos y realizar operaciones booleanas entre ellos. Se parte de un conjunto predefinido que puede constar de esferas, cilindros, conos, prismas, etc. para luego realizar operaciones booleanas o de conjuntos entre ellos: unión, resta, intersección.



Operaciones booleanas entre una esfera y un cubo

Partiendo de los objetos primitivos y las operaciones booleanas, se construye un árbol binario, desde abajo hacia arriba, que representará un objeto complejo. Para CG, el resultado se convierte a B-Rep antes de renderizar.



2.1.3. Level Sets

Hay algunas situaciones típicas que requieren almacenar el contenido tridimensional: las imágenes médicas 3D (tomografía) y los objetos de materiales complejos y anisotrópicos, como el mármol o la madera, que se modelan en forma analítica o algorítmica (procedural) mediante funciones con algunos parámetros aleatorios. En ambos casos se cuenta con uno o varios campos escalares (funciones de $R^3 \rightarrow R$) definidos en una región del espacio (normalmente un cubo o un prisma rectangular) discretizada en sólidos o elementos de volumen, cada uno representado por un punto con un valor numérico asociado que puede suponerse como una escala de grises.

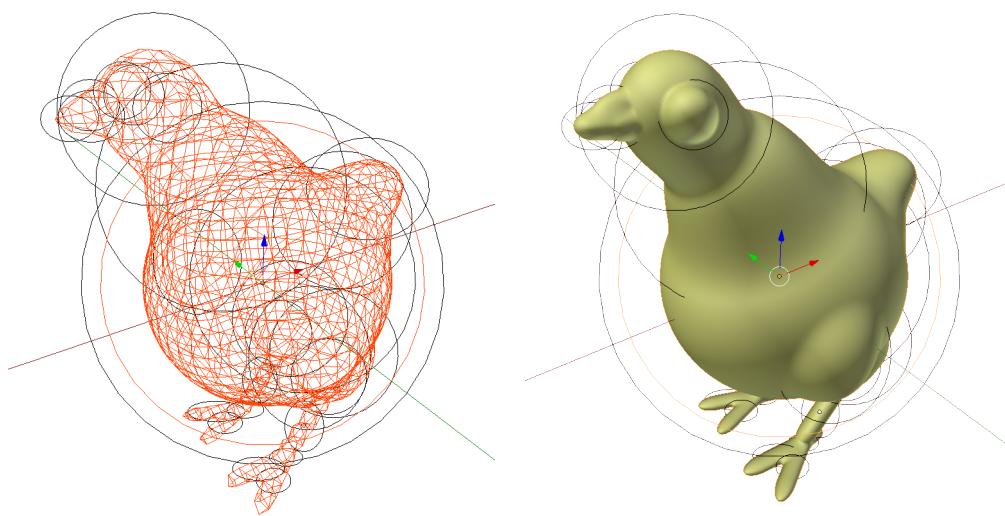
En estos casos la técnica de level-sets (conjuntos de nivel) permite definir isosuperficies o superficies de nivel, es decir superficies de igual valor de la variable. En un dominio espacial, se cuenta con un campo escalar; esto es: en cada punto del espacio se define un valor de alguna variable (temperatura); una isosuperficie (isoterma) es el lugar geométrico de los puntos donde la variable (temperatura) toma un dado valor ($25^\circ C$).

La aplicación más notable es con imágenes médicas 3D. Un equipo de Resonancia Magnética Nuclear (MNR) toma datos tridimensionales de una región del espacio en la que mide de forma indirecta el tiempo de recuperación de los núcleos atómicos al alinear y liberar sus momentos magnéticos por medio de un gran imán externo (mide las frecuencias de resonancia). Por otro lado, la Tomografía Computarizada

(CT) es una pila de tomos o cortes, cada uno de los cuales es similar a una radiografía y se obtiene por radiación de rayos X desde múltiples fuentes que se integran mediante métodos de la geometría digital (transformada inversa de Radón) para obtener los datos espaciales. Cada tipo de tejido produce un valor dado, típicamente desde el hueso blanco hasta el aire negro. Buscando un determinado valor (con ciertos tratamientos para evitar el ruido) se puede encontrar el límite de un tejido u órgano.

En términos muy simplificados, un determinado valor o nivel de gris puede servir para definir los límites de una zona de interés (segmentación), como ser un determinado órgano o un tumor. Para que el resultado no se vea como una serie de cubitos apilados (Lego), la superficie se interpola a partir de los valores en los puntos discretos, cuando hay dos puntos vecinos, uno con valor por encima y otro por debajo del nivel buscado (Marching Cubes).

Otra aplicación, más en el campo del modelado virtual, consiste en definir objetos simples, normalmente puntos, esferas o elipsoides, y una función que depende de la distancia al objeto. Los objetos simples se agrupan como músculos para crear un objeto complejo y luego una isosuperficie de distancia, suavizada, hace las veces de piel. Esta técnica se conoce como blobby objects o metaballs y también requiere marching cubes para transformar el resultado y mostrarlo como una B-rep.



Metaballs y su conversión a B-Rep

Superficie generada con iluminación

2.1.4. Sistemas Físicos

Se trata del modelado dinámico de sistemas de objetos simples (puntos, líneas, esferitas) en su interacción compleja entre sí o con el medio. Se utiliza en efectos especiales como simulación de fuego y humo o pelos o en la visualización de fenómenos físicos simulados mediante métodos numéricos sobre partículas o líneas sueltas. Actualmente se simulan también interacciones más complejas entre objetos 3D, ya sean sólidos, líquidos u objetos deformables.

Los modelos más simples de sistemas de partículas se componen de puntos o sprites (pequeñas imágenes

variables) que se mueven y cambian mediante alguna ley razonable para el efecto buscado (humo, fuego, fuegos artificiales). El realismo cada vez mayor se obtiene con un mejor modelado de la física y la interacción. El modelado con muchísimas partículas y ecuaciones diferenciales en derivadas parciales, requiere de complejas técnicas de Cálculo Numérico en la Mecánica Computacional y del análisis optimizado de la interacción entre objetos que proporciona la Geometría Computacional. Hay software y hardware (GPGPU o general purpose GPU) especializados para esas tareas, que utilizan muchos recursos de la Ingeniería Informática.

Los efectos especiales comenzaron siendo muy simples y burdos, pero actualmente no se puede distinguir fácilmente una aproximación realista del cine versus la visualización de un resultado para ingeniería clásica (fluidos, sólidos deformables, fractura, interacción). El cine requiere un realismo que se asemeja al cómputo en ingeniería y la ingeniería se presenta con las más avanzadas técnicas de gráficos. De todos modos, para la ingeniería clásica es fundamental conocer el error y los límites de aplicación de la aproximación; en el cine no.

2.2. Material y Terminación superficial

Los objetos modelados terminan siendo un conjunto de puntos, líneas y triángulos; desde allí hasta que se vean como fuego, pelos, madera o acero inoxidable hay que realizar varias operaciones, que podemos englobar como métodos de definición de materiales y de terminación superficial.

2.2.1. Color

Los cosméticos, telas, papeles y pinturas requieren de una ingeniería muy complicada para definir y medir un color. Color "salmón" no tiene un significado indiscutible, pero ni siquiera color "rojo" es algo exacto. Se necesita una denominación analítica, un mecanismo de trasmisión y control inequívoco entre el cliente, el vendedor y el fabricante. En la industria multimedia y gran parte de la gráfica la situación es mucho más sencilla. En su estado actual y condicionado por la historia del hardware, el color se representa, en la informática, mediante una mezcla (aditiva) de tres valores estandarizados: un rojo, un verde y un azul normalizados.

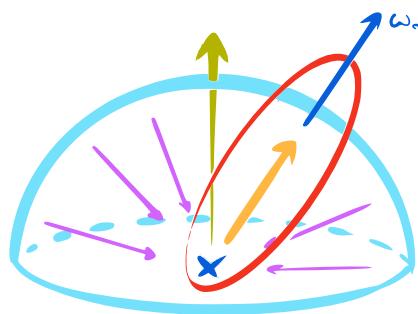
En CG, además se utiliza un cuarto valor, componente o canal alfa; éste "viaja" junto con el color y se utiliza para diversos efectos, particularmente para la mezcla con otros colores (transparencias). El color será aquí una lista de cuatro valores o vector de cuatro dimensiones RGBA. Más adelante veremos una teoría más completa, pero por ahora digamos que el color de algo es una mezcla de RGB y un valor A de alfa. El color que vemos en un punto de la superficie de un objeto opaco, es el reflejo parcial de la luz que recibe. Simplificando demasiado, podemos decir que un objeto rojo es aquel que recibe luz blanca ($R=G=B=1$) y refleja sólo la parte roja ($R=1, G=B=0$), absorbiendo el resto. Además, si el objeto es translúcido como un vidrio rojo, trasmite y refracta solo la parte roja de la luz que recibe y absorbe el resto. Del mismo modo un tubo de luz de Neón, emite luz roja. Es decir que la superficie de un objeto puede reflejar, refractar o emitir luz.

2.2.2. Iluminación y sombreado

¿Cuánta luz recibimos de cada punto del objeto? Una esfera roja opaca no se ve como un disco rojo; el efecto de degradé o cambio de intensidad del color se denomina sombreado o shading y se debe a que la intensidad que llega desde cada punto de la superficie de un objeto varía con las posiciones relativas de la superficie, la luz y el ojo del observador.

Esto implica considerar el material de la primitiva y la interacción con el resto de la escena. Respecto al material, hay propiedades del mismo que definen cómo interacciona con la luz que recibe. Respecto a la escena, es la que determina cual es esa "luz que recibe", y eventualmente cómo la luz que refleja llega a la cámara. La luz que recibe puede provenir de luces o de reflexiones/emisiones de otros objetos, considerando además las posibles occlusiones (es decir, las sombras que proyecten otros objetos). Además, luego de rebotar en el objeto, en su camino a la cámara, la luz puede verse afectada nuevamente, simulando fenómenos como por ejemplo la "niebla".

Determinar la luz reflejada a partir de la recibida en un punto es la parte central y más compleja del proceso de iluminación, y una forma de formular este problema de manera generalizada es mediante la ecuación conocida como la *ecuación del renderizado*:



$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \underbrace{\int_{\Omega} L_i(\mathbf{x}, \omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos\theta_i d\omega_i}_{L_r(\mathbf{x}, \omega_o)}$$

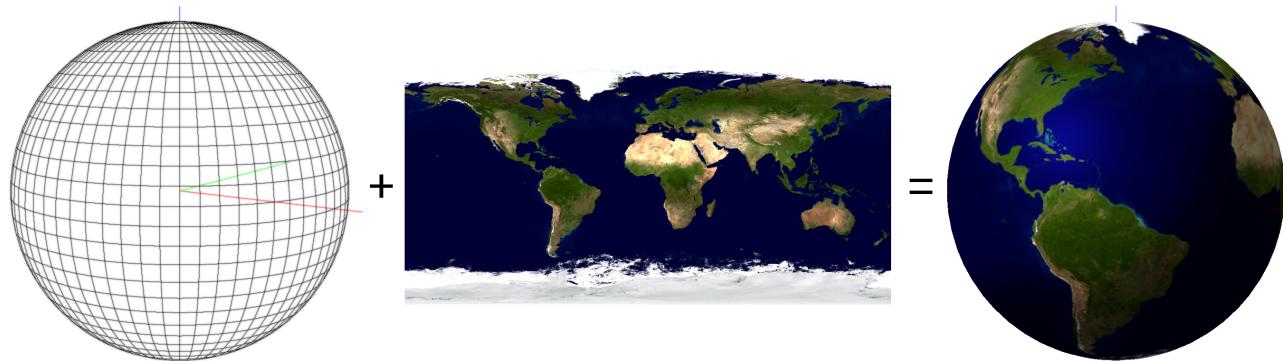
- L_o es la luz total desde un punto (el punto de la primitiva que vamos a "colorear" hacia una dirección dada (hacia el ojo/la cámara))
- L_e es la luz que el material emite por sí mismo (si es luminoso/fluorescente)
- L_r es la luz que refleja a partir de la que recibe
 - L_i es la que recibe desde una dirección dada
 - f_r representa la proporción entre cuanta se recibe y cuanta se refleja en una cierta dirección (la probabilidad de que la recibida rebote justo en esa dirección)
 - $\cos\theta_i$ modela cuánto se ve del "punto" (imaginarlo como un pequeño disco... si se mira de frente abarca más superficie que cuando está inclinado)
 - Se integra sobre una semi-esfera (Ω) para "sumar" la luz que recibe desde todos los ángulos.

Cada parte de esta ecuación requiere de un modelado que puede llegar a ser arbitrariamente complejo. En los modelos interactivos se suele partir de una función muy muy simplificada, conocida como *modelo de Phong*. Más adelante estudiaremos en detalle dicho modelo, y también veremos modelos simplificados para las luces, y para el decaimiento que esta sufre desde que refleja en el objeto hasta que llega a la cámara.

2.2.3. Texturas

Si pretendemos modelar analíticamente algunos materiales variables, como la madera; o superficies complejas, como un suelo de césped o una pared de ladrillos; no habrá potencia de cálculo, ni horas de modelado que resulten suficientes. En CG se recurre a un truco simple: se "pega" una imagen sobre la superficie, de modo que se asemeje a lo que queremos representar. El caso más sencillo es una calcomanía, pero también se puede pegar la textura de una piel sobre el modelo curvo de una cara.

Sin embargo, el truco ya dejó de ser simple hace años dada la potencia y eficiencia que se ha adquirido para hacerlo en la GPU. Hoy se utilizan los mecanismos de aplicación de texturas para "pegar" mucha más información sobre una superficie.



Más adelante estudiaremos en los mecanismos más habituales involucrados en la aplicación de texturas.

2.3. Montaje de la Escena

Los objetos habitualmente se generan en programas de modelado y se importan (sus triángulos, con material y textura) al programa gráfico para renderizar. En nuestra práctica, para aprender estos temas, usaremos o construiremos objetos simples desde el mismo programa, pero la principal diferencia estará solo en la cantidad de triángulos.

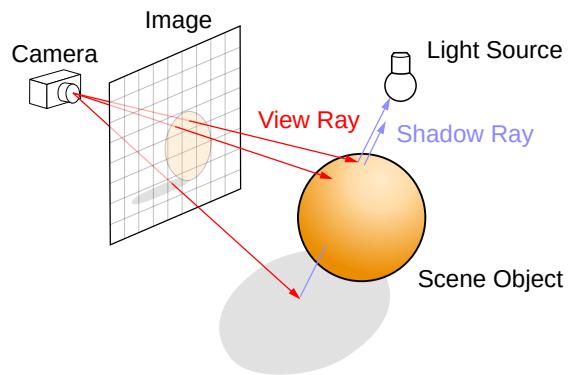
Todos los objetos se ubican en un sistema de coordenadas que se denomina espacio del modelo (o *model space*) definido arbitrariamente a conveniencia del usuario/programador. Allí se colocan también las luces y la cámara. Cada pieza se define o modela en un sistema de coordenadas propio y luego se debe ubicar en la escena. A su vez toda la escena se debe analizar desde el punto de vista de la cámara. Finalmente,

ese espacio se "aplasta" ("proyecta" sería la palabra correcta) para pasar del volumen 3D modelado a una imagen 2D. Las propiedades de la cámara definirán cuál es la porción visible del espacio y también el tipo de proyección. Para implementar todo esto se utilizan transformaciones. Dedicaremos posteriormente una unidad de la materia al estudio de espacios y transformaciones, por lo que no entraremos en mayores detalles en esta introducción.

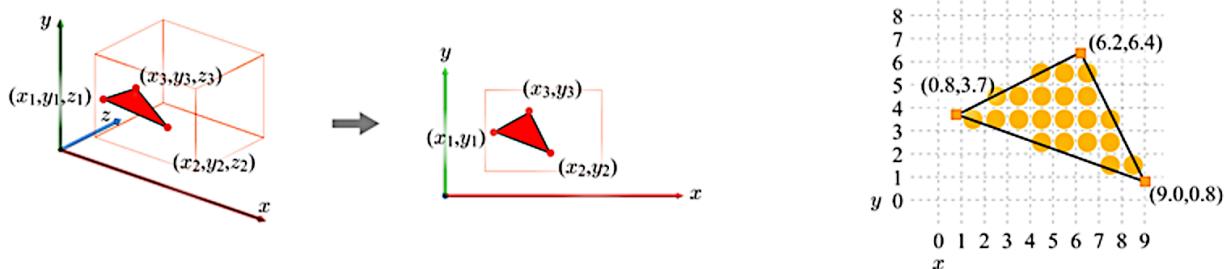
2.4. Raytracing vs Rasterización

A nivel algoritmos e implementación, las técnicas para *renderizar* (es decir, generar una imagen a partir de la descripción) se pueden dividir en dos grupos:

- Basadas en **Raytracing** (o *trazado de rayos*): estas técnicas se basan en "lanzar" desde la cámara un rayo por cada pixel de la imagen final, y determinar contra cuál primitiva "choca" primero. Esa será la primitiva que se "verá" por ese pixel, y se debe calcular para el mismo su "color".



- Basadas en **Rasterización**: estas técnicas se basan en tomar cada primitiva, determinar qué pixeles "potencialmente" cubre, determinar para cada uno de ellos (que se denominan *fragmentos*) si efectivamente es visible (si no está "detrás de" otra primitiva), y en caso afirmativo calcular el su "color".



En líneas generales ambas técnicas pueden verse como similares. Ambas tienen un loop por primitivas y uno por pixeles/fragmentos, solo que el anidamiento de estos loops está invertido:

Raytracing

para cada pixel (px) de la ventana
 para cada primitiva (pr) de la escena
 intersectar el rayo por px con pr
 quedarse con la más cercana
 calcular "color" y pintar

Rasterización

para cada primitiva (pr) de la escena
 para cada fragmento (px) que cubre la primitiva
 determinar si es visible
 si lo es calcular "color" y pintar

Sin embargo, aunque la estructura de partida pueda parecer similar, las implicaciones en las implementaciones entre una y otra técnica suelen ser muy diferentes entre sí. Las técnicas basadas en raytracing suelen permitir un modelado más realista de ciertos fenómenos como sombras, reflejos, refracciones, etc. pero no suelen ⁵ ser suficientemente rápidas como para utilizarlas en tiempo real. Por ello, son las más usuales en renderizado offline (por ejemplo, cine). Las técnicas basadas en rasterización suelen ser mucho más apropiadas para renderizado en tiempo real, aunque pueden requerir de muchos "trucos" adicionales específicas para admitir aproximar efectos como sombras sombras, reflejos, etc.

En el transcurso de esta materia se dará mayor atención a las técnicas basadas en rasterización, ya que son las más habituales en aplicaciones interactivas. Sin embargo, muchos conceptos fundamentales serán aplicables a ambos grupos.

Ambas técnicas finalizan con el cálculo del "color" del pixel. Se ha puesto "color" entre comillas para manifestar que si bien la salida del proceso es un color, el cálculo involucra mucho más que el color de la primitiva (como se analizó antes, iluminación y sombreado, texturas, mezcla, etc).

2.5. Rendering Pipeline

Una vez definidos los modelos y la escena, los datos y el programa (secuencia) se envían a la GPU. La serie más o menos rígida de cálculos y pasos que transforman la escena en una imagen se denomina rendering pipeline. La imagen final será mostrada en un viewport que es una porción del display, de tamaño y posición que también fue definido en CPU y enviado a GPU.

En primer lugar, se aplican las transformaciones modelview para convertir todas las coordenadas (vértices, luces, normales, etc.) al espacio del ojo o espacio visual.

En el vertex processor se aplican los materiales y se calcula la iluminación en los vértices (per-vertex shading), se asignan las coordenadas de textura (como se va a pegar la imagen) y todo dato de los vértices que luego deba ser interpolado en los fragmentos interiores de las primitivas básicas (segmentos o triángulos).

Luego se realiza el ensamblado de primitivas, que consiste en identificar los vértices adecuados (de la lista general) para construir cada línea o triángulo definido.

⁵Hasta hace pocos años era impensable renderizar escenas de cierta complejidad en tiempo real, pero dado el avance en la capacidad de cómputo de las GPUs y alguna técnicas nuevas para reducir la cantidad de rayos necesarios, ya existen algunos demos que indican que ya podría ser considerado una alternativa viable.



A la primitiva que llega hasta aquí se aplica la matriz de proyección, que convierte el volumen visual en un cubo normalizado que en seguida se adapta al tamaño de la imagen final (el viewport).

Para los objetos opacos y cerrados no tiene sentido procesar las caras traseras, que son invisibles y pueden descartarse (backface culling). También se verifica si la primitiva debe descartarse (cull) por completo o recortarse (clip) mediante los seis clipping planes que definen el volumen visual y otros definidos por el usuario.

El proceso de rasterización divide las primitivas en fragmentos que ocuparán los píxeles de la imagen final. Para cada fragmento se interpolan el color y todas las variables que tenían los vértices de las primitivas.

A partir de aquí se cuenta con información raster en los fragmentos de primitivas; cada uno corresponde a un determinado pixel del viewport. (notar el cambio de color de las flechas en el diagrama de arriba)

Así se llega al fragment processor que debe definir el color final del fragmento (pero no puede alterarle la posición (x,y) que ya está asociada a un determinado pixel). Aquí normalmente se aplica iluminación individual (per-fragment shading, en lugar de interpolada) y se mezcla (blending) con el color que asigna cada textura a ese fragmento. También aquí se puede mezclar el color con un color definido de neblina o fog.

El resto de las operaciones que sufren los fragmentos corresponden a los tests y escritura de los buffers auxiliares (memoria) que mantienen información por pixel, en paralelo con el buffer de color que es el resultado que se muestra a la salida. Algunos tests descartan definitivamente el fragmento, el ejemplo más importante es el test de z o depth test que verifica si ese fragmento de primitiva está tapado visualmente por otro, en cuyo caso lo descarta, el fragmento más cercano al ojo (hasta ahora) se mantiene en un buffer de profundidad o z-buffer que, básicamente es una imagen gris, con un valor para cada pixel representando la distancia al ojo.

Luego se realizan algunas operaciones de alteración del color de los fragmentos que se mezclan (blend) por interpolación u operaciones de bits (logical operations) con el color preexistente en el buffer de color.

A partir de aquí se procesan píxeles; el principal es almacenado en el color buffer, pero hay otros, como el de profundidad. El framebuffer es el conjunto de buffers (porciones de memoria) que tienen la información por píxel. Hay varios y tienen múltiples aplicaciones. Al final solo importa el buffer de color (RGB) que se guarda como imagen o se muestra en el viewport, una porción de la ventana del programa.

El pipeline muy simplificado que se mostró aquí sirve para visualizar las operaciones básicas; en realidad

es más complejo y contiene etapas en las que el programador puede introducir alteraciones a los cálculos estándar.

Esa serie de cálculos está más o menos estandarizada por razones históricas. Bien podría pensarse en placas gráficas dedicadas a otro mecanismo de rendering, por ejemplo, ray casting: desde el “ojo” pasamos un rayo visual por cada pixel del viewport y analizamos como rebota o atraviesa los objetos hasta definir su color final. El camino inverso del rayo visual es el real, que acumula luz hasta llegar al ojo pasando por un pixel del viewport y define, por lo tanto, el color de ese pixel. Pero dichas placas gráficas no existen.

3. Notas para el estudio del tema

Este tema es esencialmente informativo del contenido de la Computación Gráfica, pero hay algunos tópicos que no volveremos a tratar en profundidad y cuyos conceptos deben aprenderse en esta única instancia. Estos son:

- Las etapas básicas de la CG, poder explicar la secuencia y los lineamientos básicos de cada etapa.
- El hardware y sus mecanismos de funcionamiento básico (en la medida en que estos generen requisitos algorítmicos especiales, sin entrar en la física, química o electrónica básicas).
- Los conceptos de primitiva, escena, rendering, vector vs. raster, batch vs. interactivo o en tiempo real (concepto de “tiempo real”), fotorrealista o NPR.
- Métodos de modelado (csg, b-rep, level-sets, etc.).
- Estructura general del *rendering pipeline*.

Los temas de color, iluminación y terminación superficial se verán con más profundidad en otras unidades, al igual otros detalles del viaje de los fragmentos por la GPU, con sus tests y buffers de almacenamiento. Los métodos específicos para definir curvas y superficies analíticamente y luego generar a partir de ellas las *b-reps* también serán estudiados en detalle más adelante.

Las preguntas de examen sobre este tema son referidas a los ítems mencionados y generalmente del tipo “describa...”. Pero es necesario aclarar que el hecho de que no se evalúe el resto de los temas no reduce la necesidad de que un ingeniero informático los domine en la profesión. En la práctica, es el ingeniero quien decide en qué tecnologías invertir tiempo o dinero y, para ello, debe ser capaz reconocer la diferencia entre tecnologías posibles o de ciencia ficción y meros engaños, las posibilidades se distinguen mediante el conocimiento de la física, química y electrónica básicas que aquí no exigimos.

De igual modo, las cuestiones históricas tampoco serán exigidas en las evaluaciones, pero complementan la formación del ingeniero y en muchos casos sirven para entender por qué las cosas son como son en la actualidad y ayudan a formar una idea de qué aspectos pueden “evolucionar libremente”, y cuales están más condicionados por cosas como la “compatibilidad hacia atrás” o caprichos del mercado que por la evolución de la técnica.