



GRAPHICS PIPELINE

Computación Gráfica

Rendering

Proceso de generar una
imagen 2D a partir de un
modelo 3D

Offline vs. Real-time Rendering

- Exactitud sobre rapidez.
- Películas, diseño de productos, aplicaciones médicas.
- Sacrificar detalles por desempeño.
- Motores de videojuego, aplicaciones interactivas.
- Flujo constante de imágenes para crear animaciones suaves y fluidas.

Graphics Pipeline

- **INPUT:** escena compuesta de modelos en 3D.
- Los modelos se introducen a mano o por medio de un software especializado.
- **OUTPUT:** imagen (en display, impresora, memoria ...)
- La forma de dibujar debe ser independiente al dispositivo de salida.

Un vistazo al pipeline

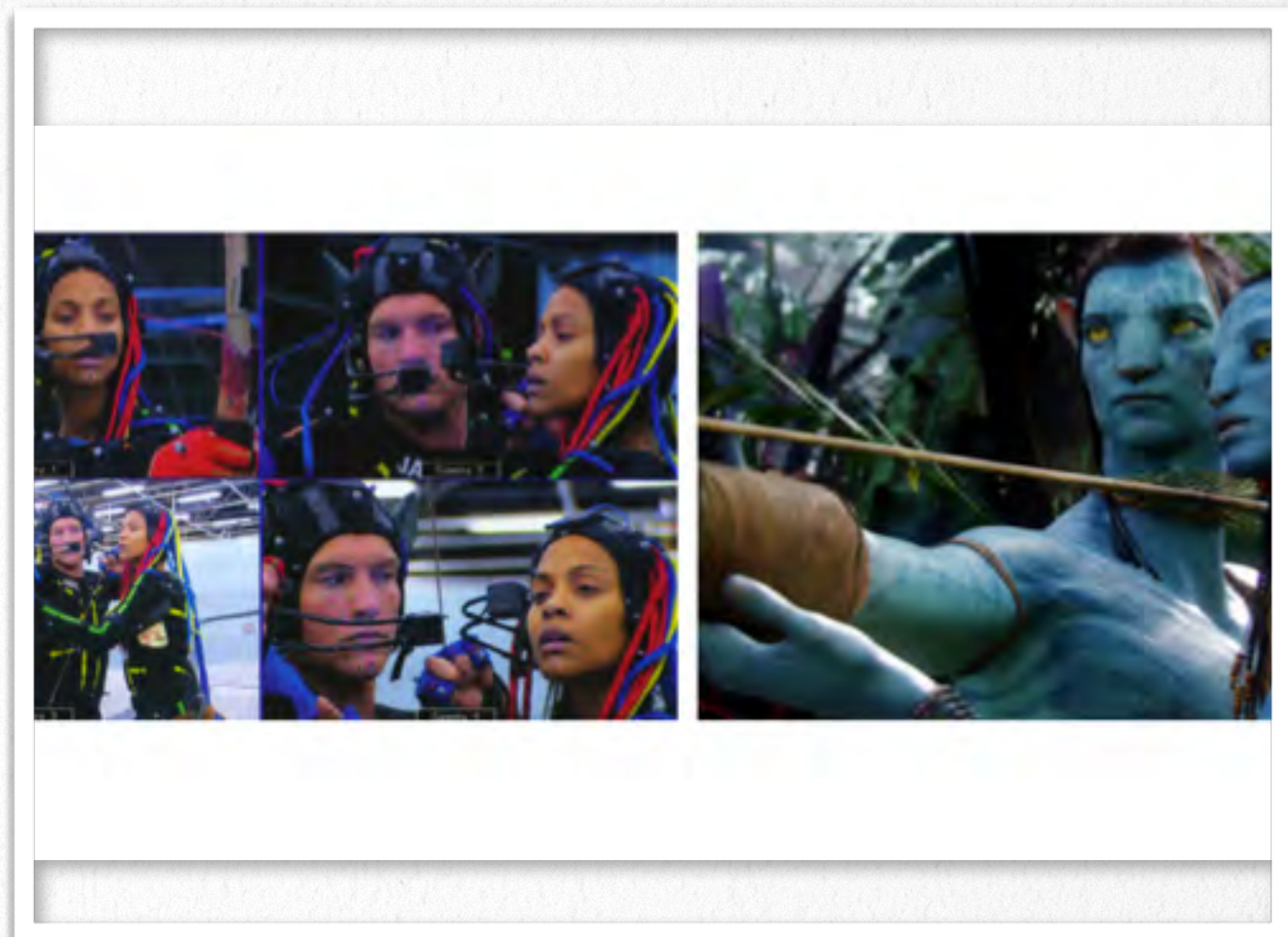


Entrada



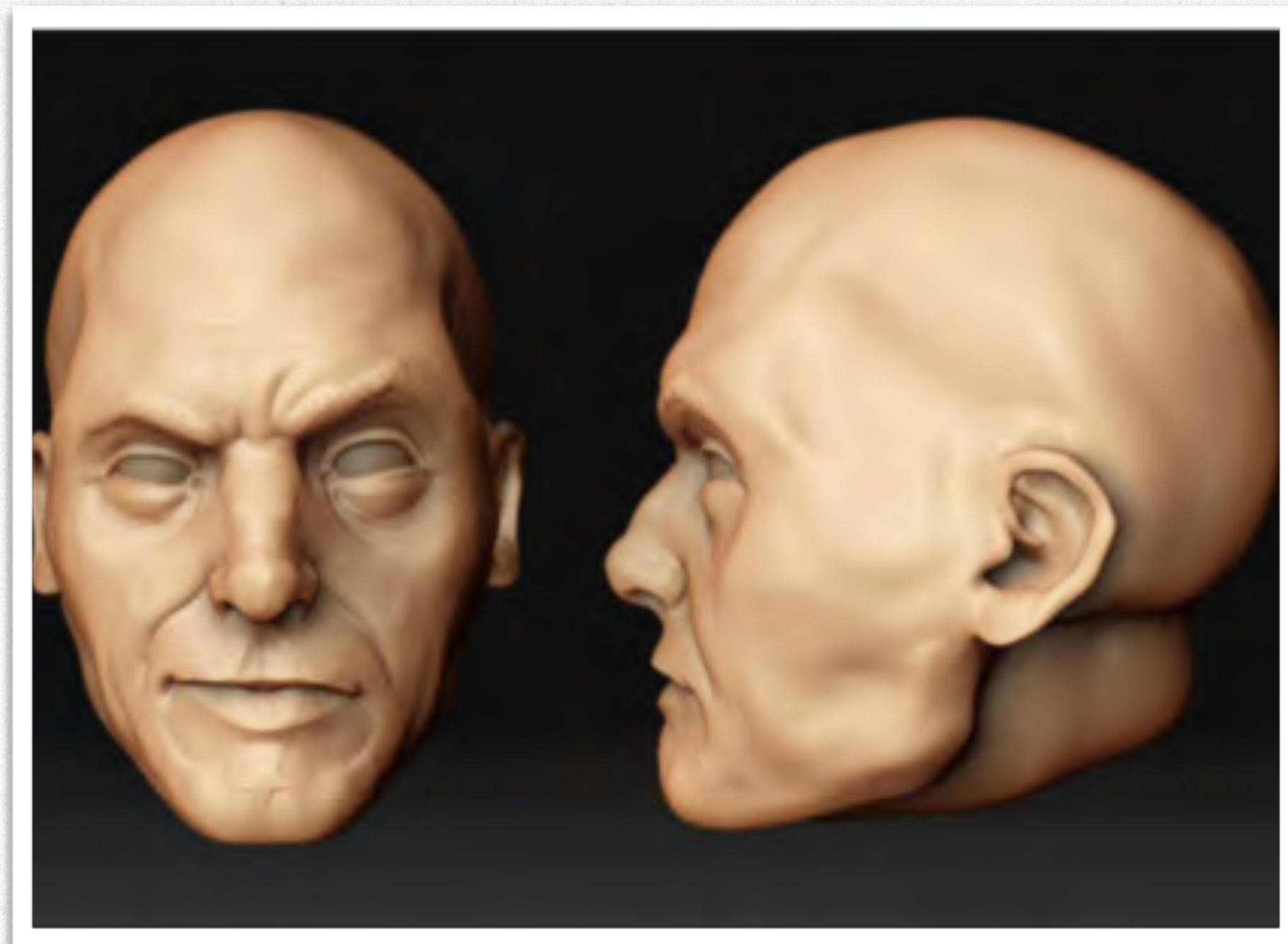
[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Entrada



[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Procesamiento



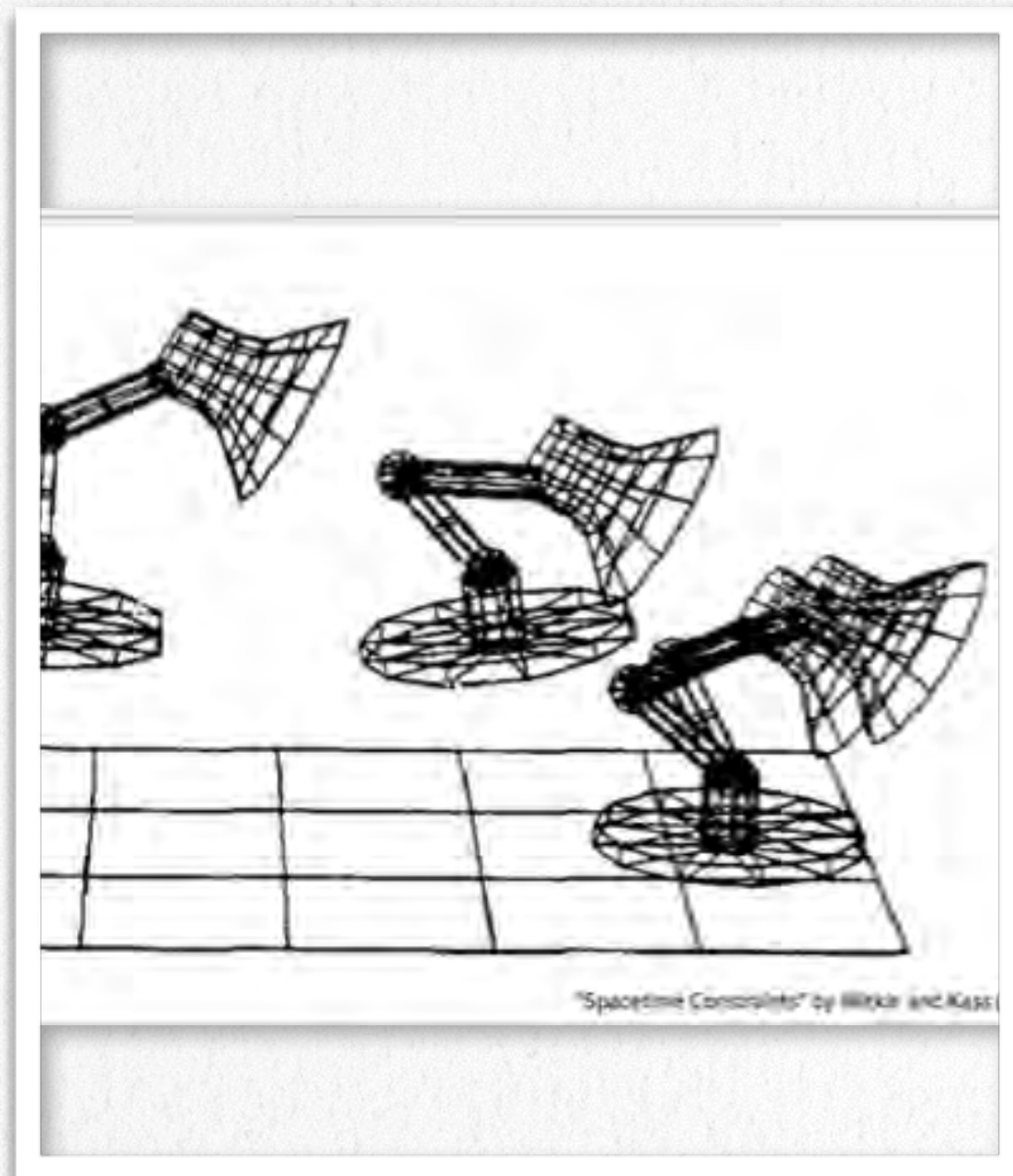
Modelado "a mano"



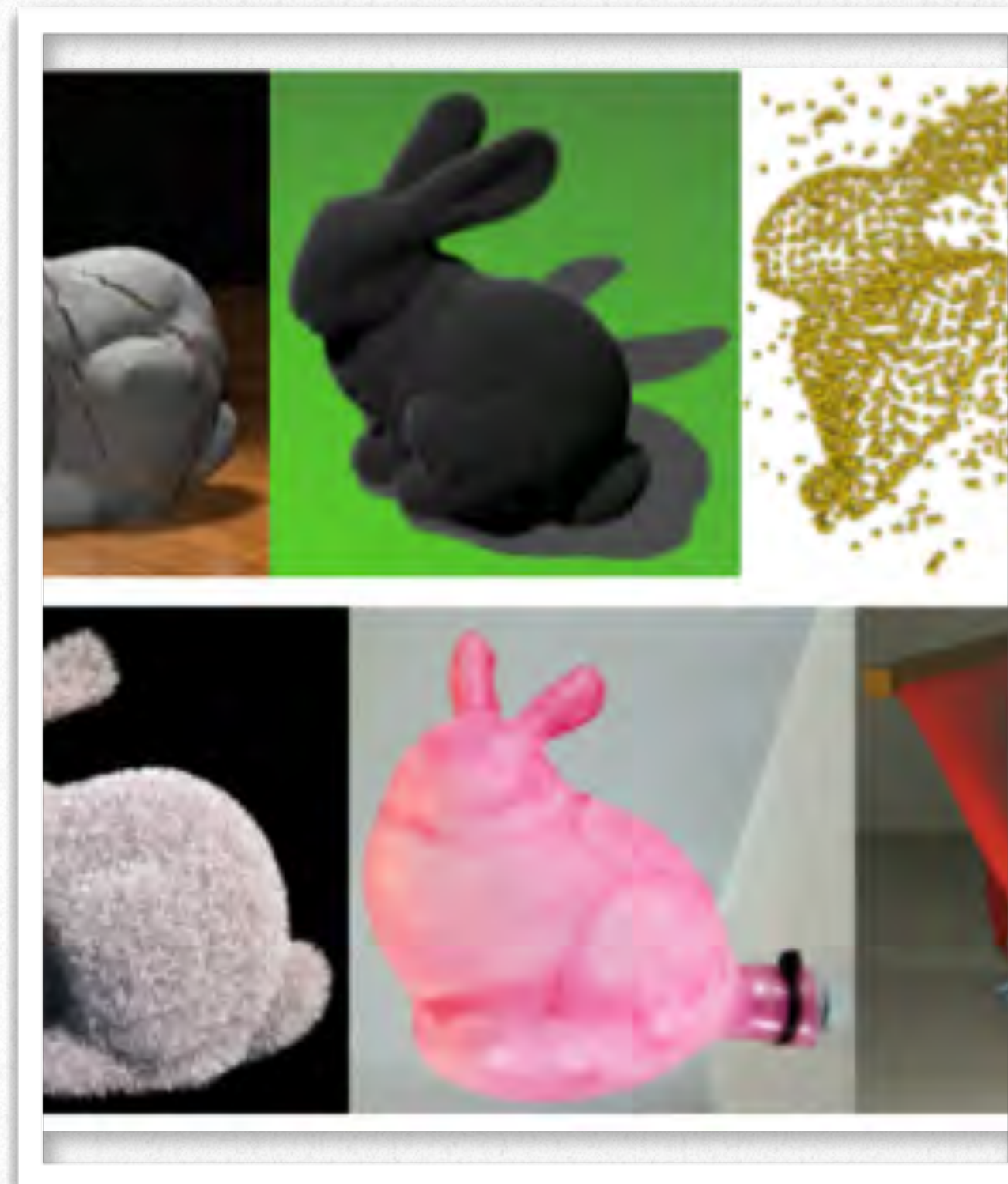
Modelado procedural

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

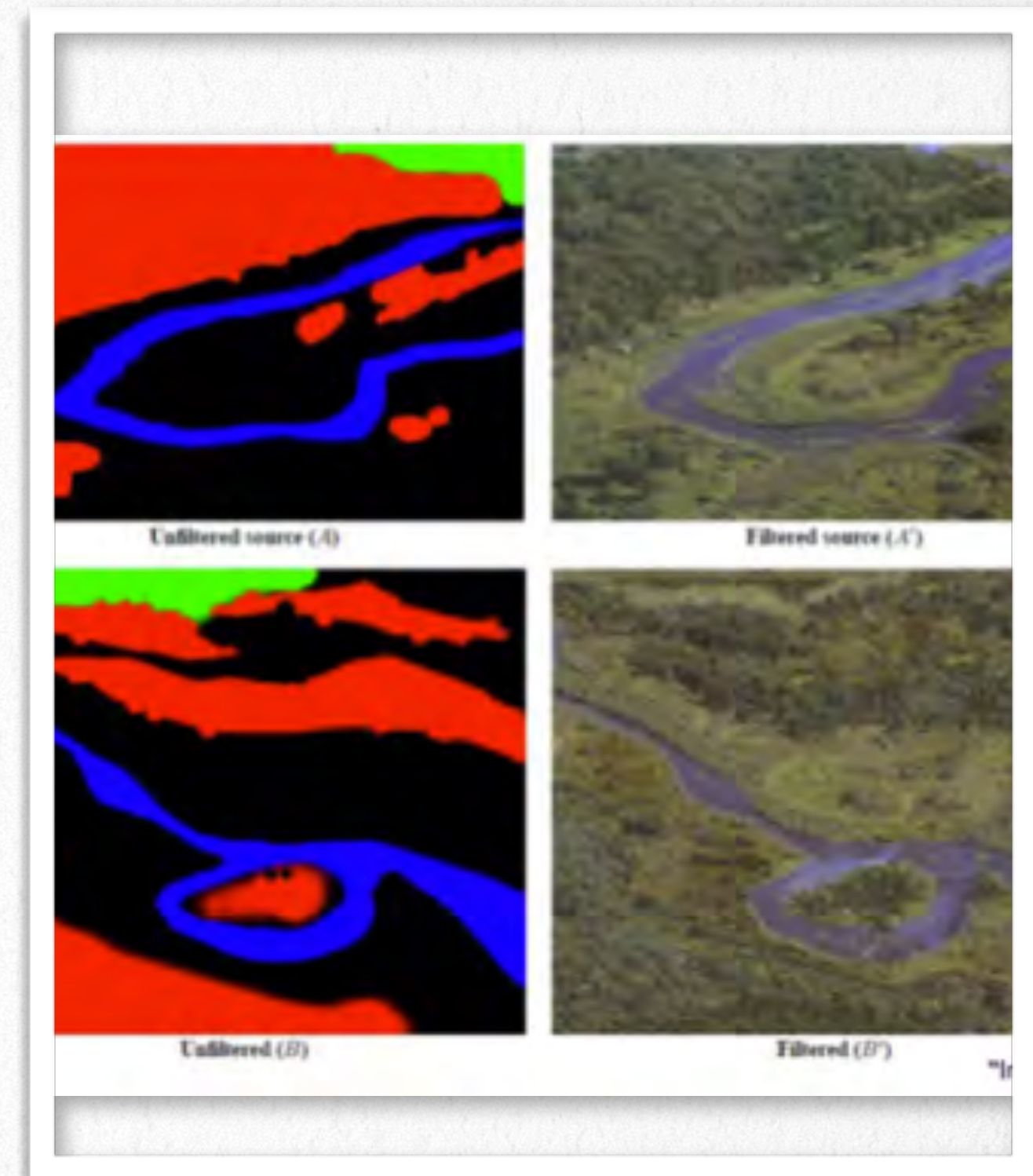
Procesamiento



Animación



Simulación



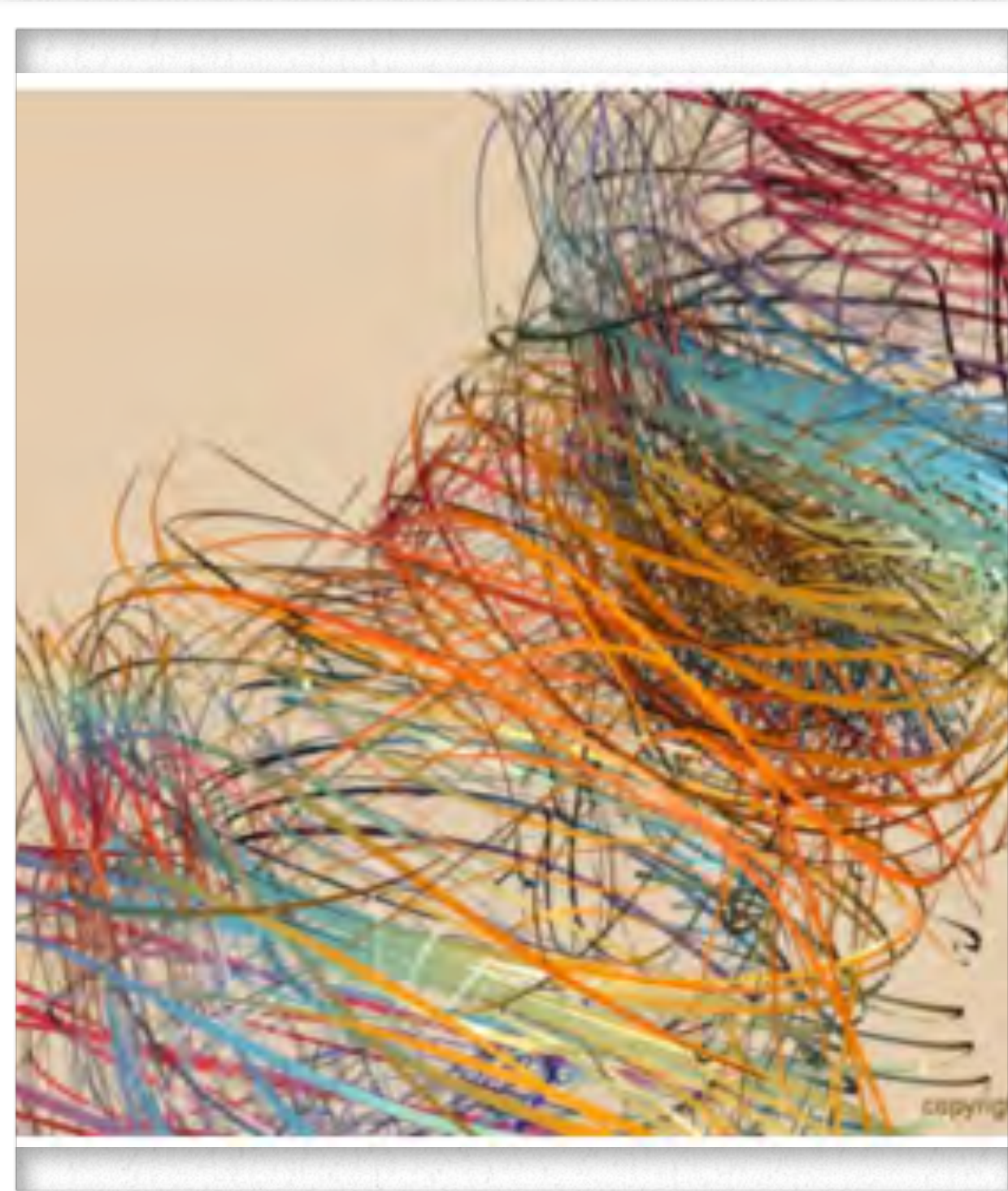
Procesamiento de imágenes

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Salida



Cuadros renderizados



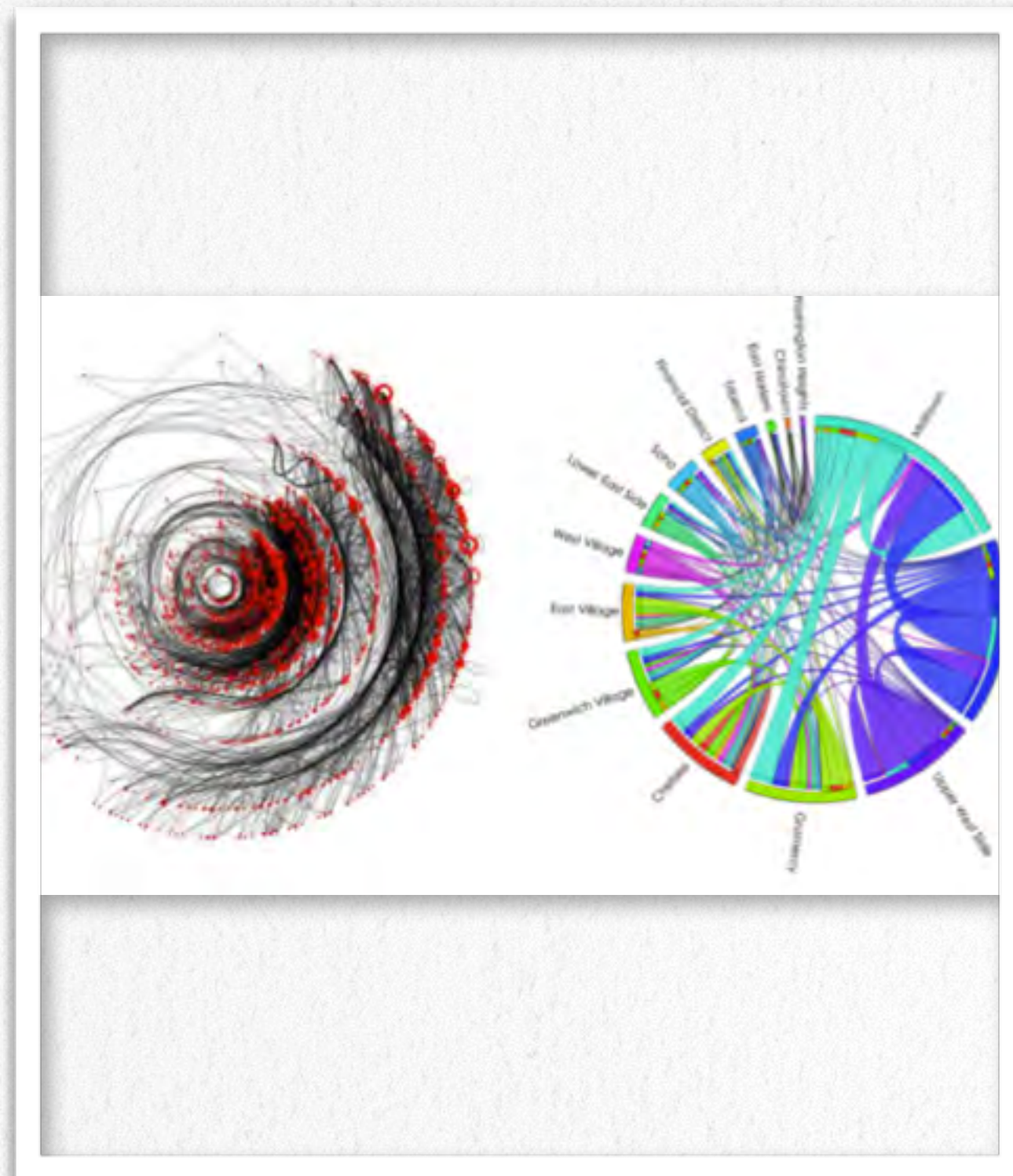
Arte



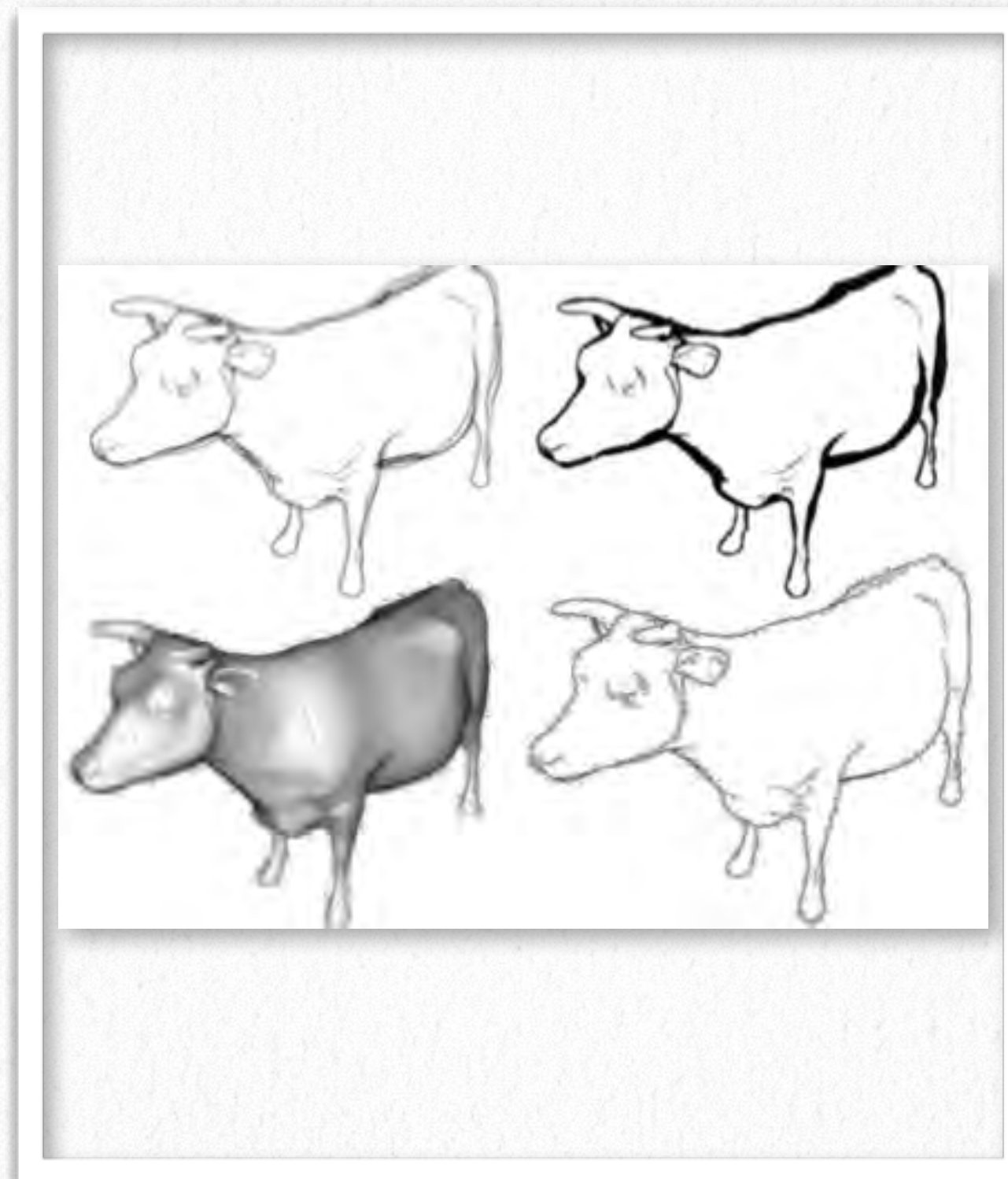
CAD

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Salida



Visualización



Estilizado



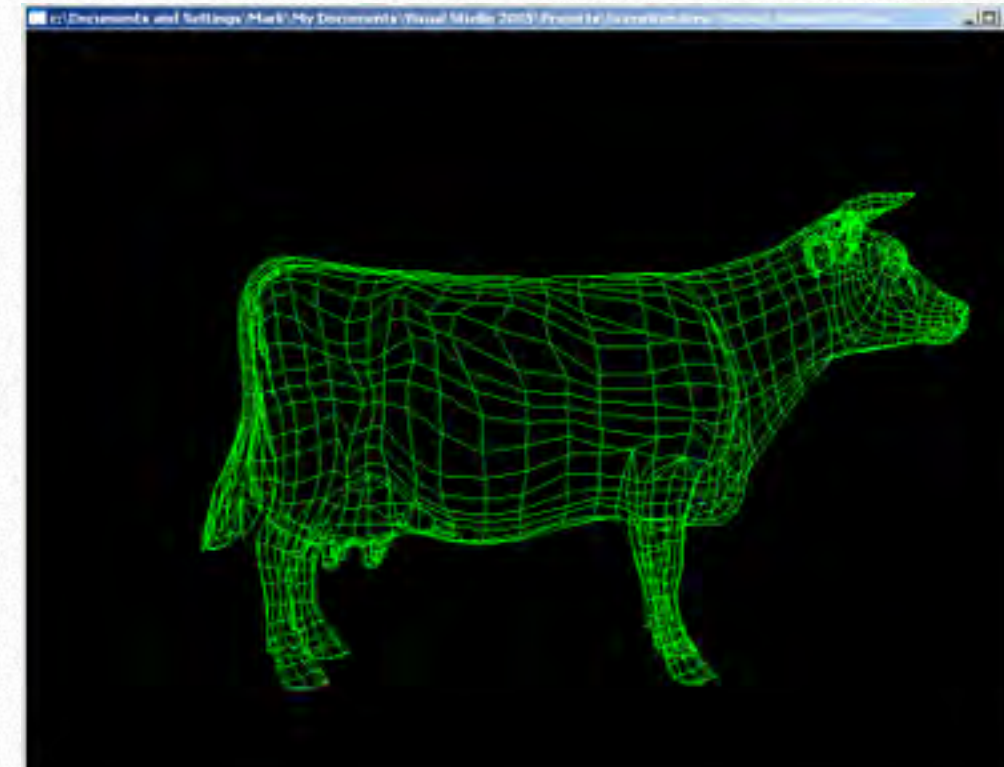
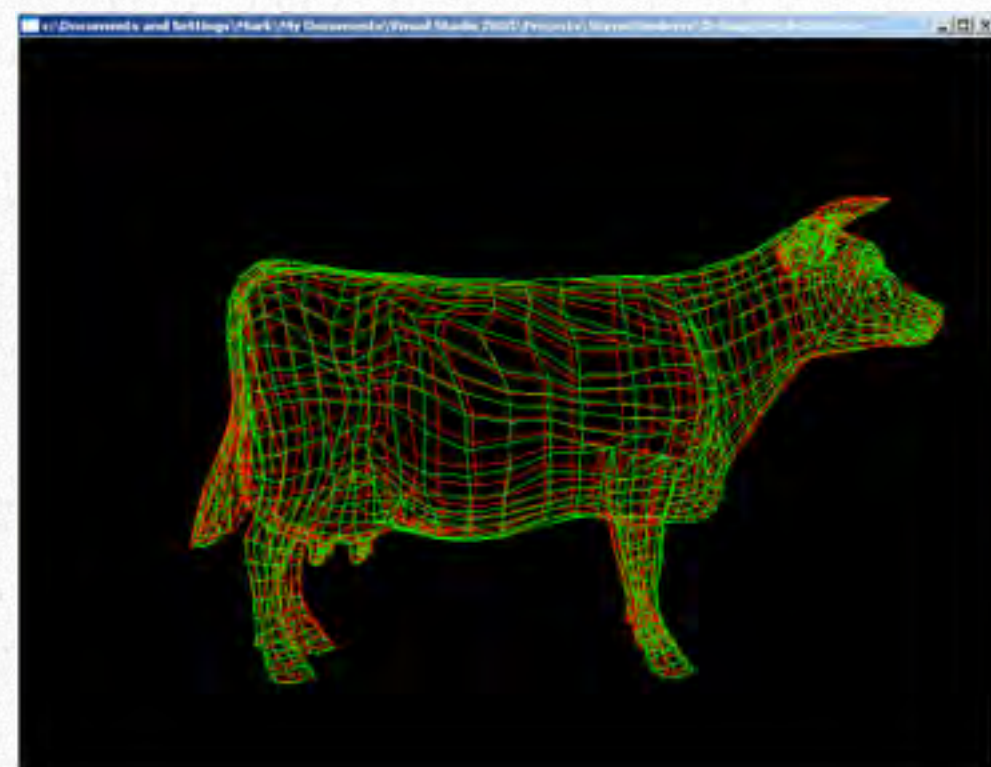
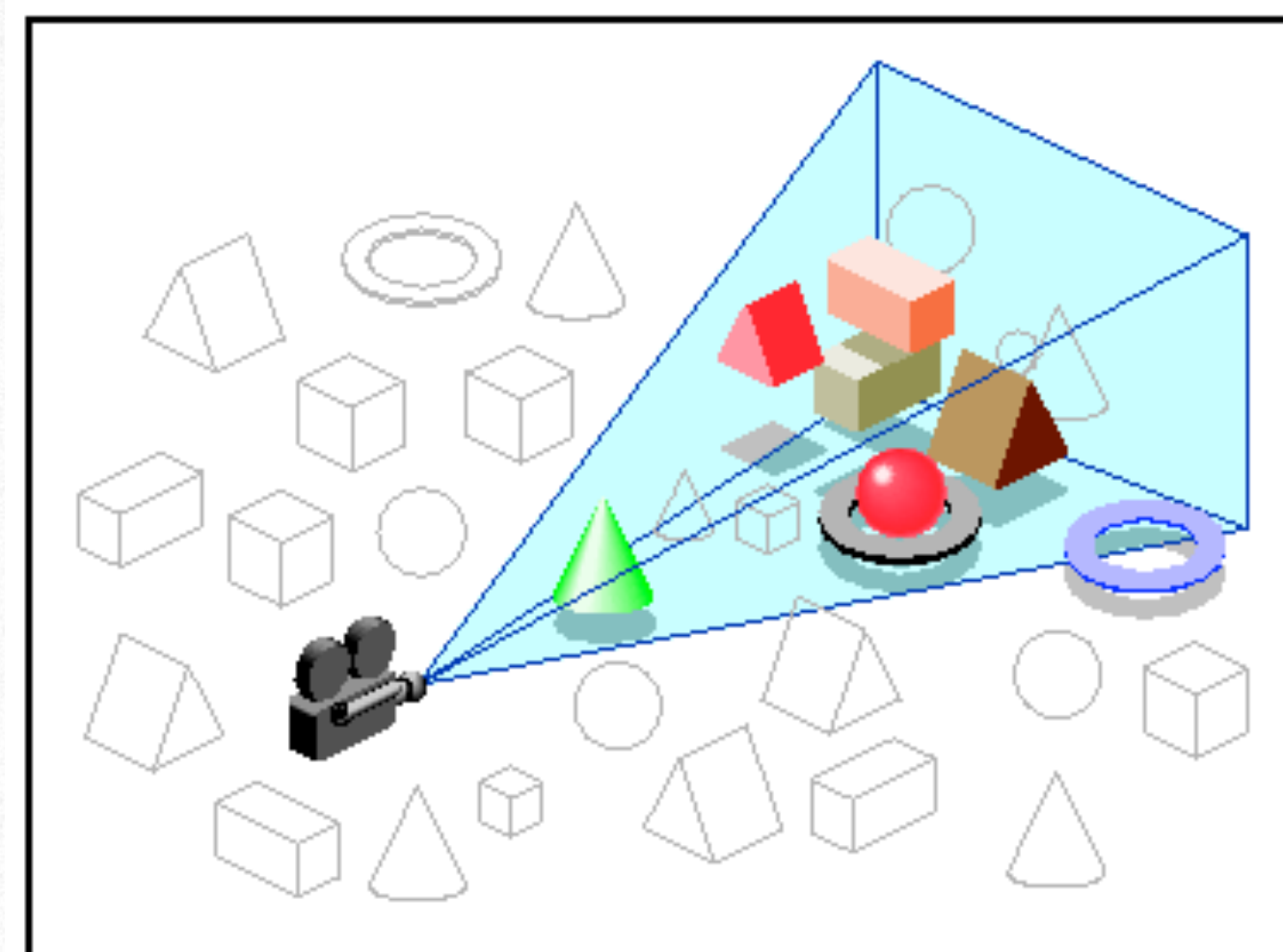
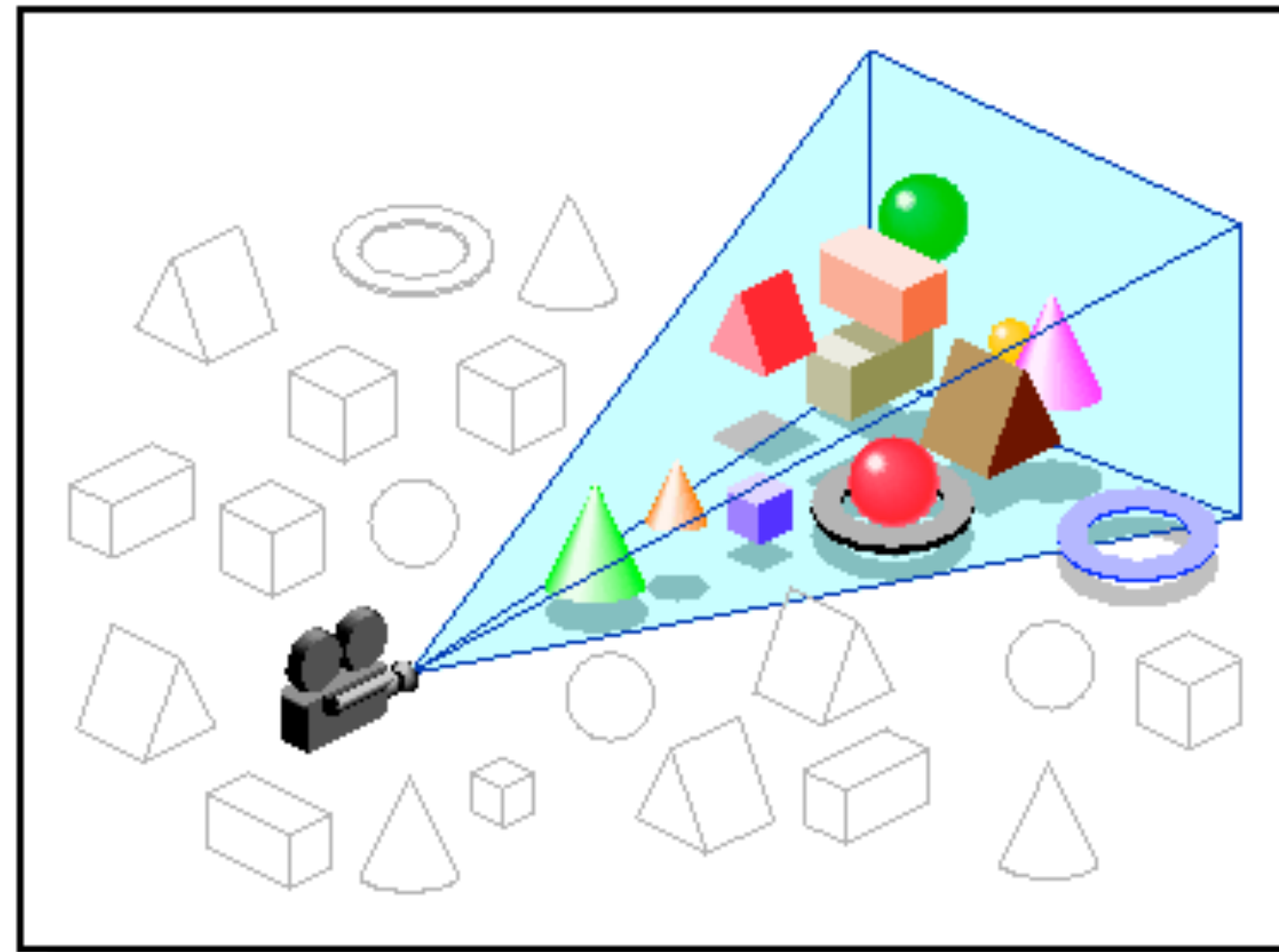
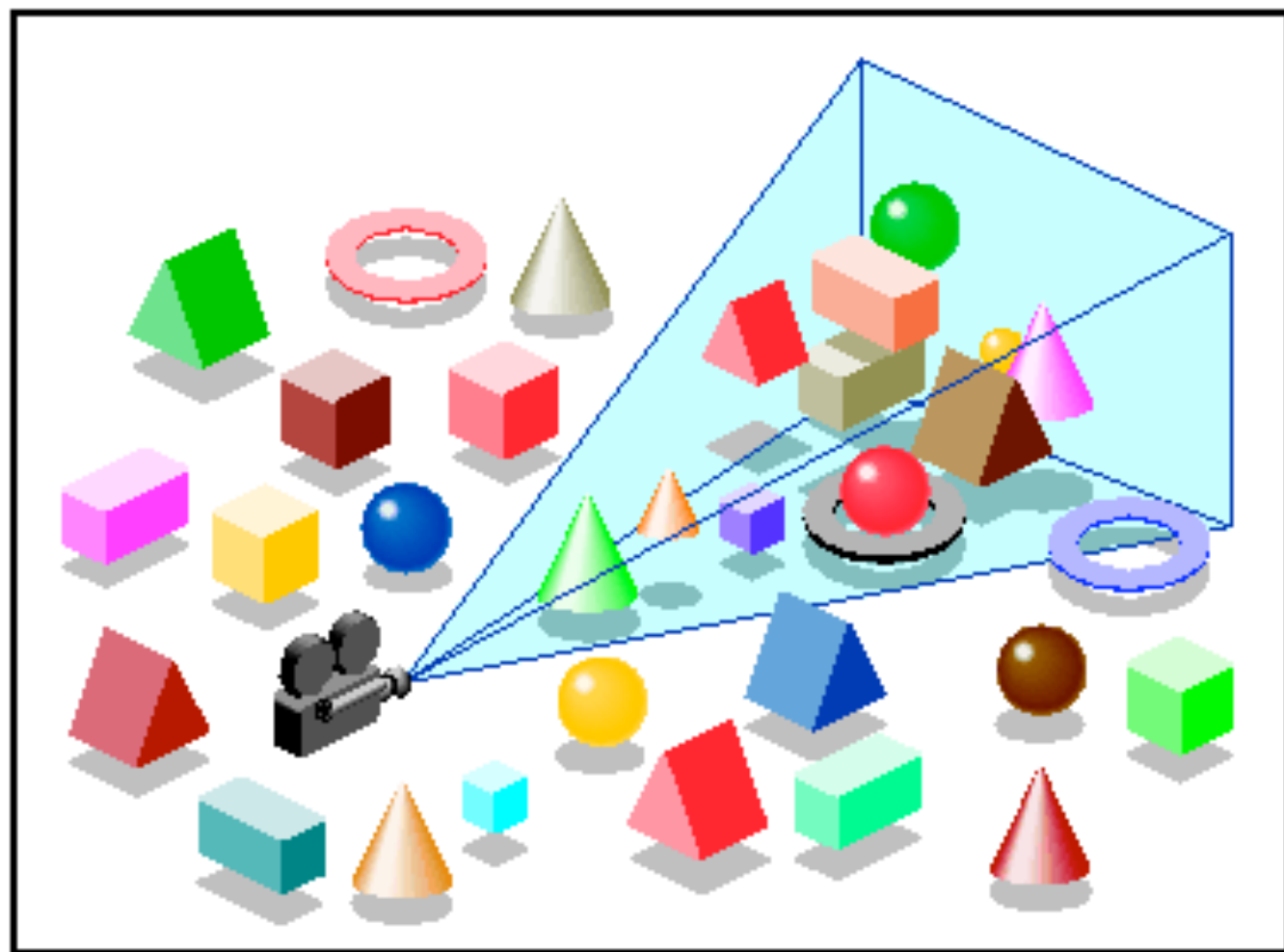
Interactividad

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Desempeño

- La gran cantidad de procesamiento necesario en una sola escena lleva a hacer “trampas” al momento de renderizar las imágenes.
- Reducir detalles visuales para mejorar el desempeño sin afectar la calidad de la imagen.
- En gráficos el procesador y la memoria son bienes preciosos y limitados.
- Ejemplos: culling, algoritmos de nivel de detalle (LOD)...

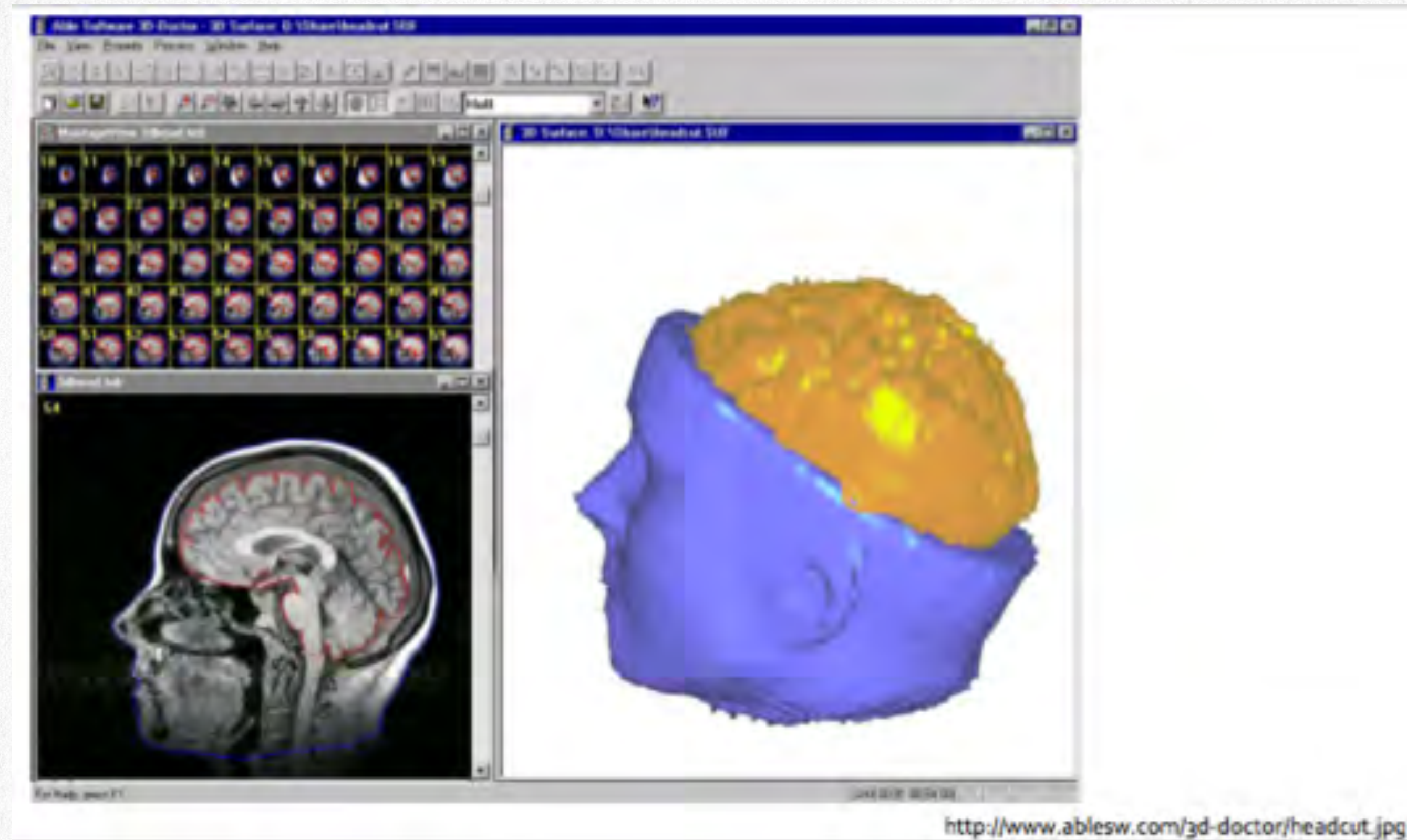
Desempeño



Desempeño

- Algunas etapas del pipeline se pueden implementar en hardware para obtener desempeño interactivo.
- GPUs actuales permiten pipelines programables.
- Métodos programables: modelos de iluminación o transformaciones geométricas propias.
- Se debe saber lo que se hace para usar el GPU.

Salida



Interactividad

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Salida



Interactividad

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

Salida

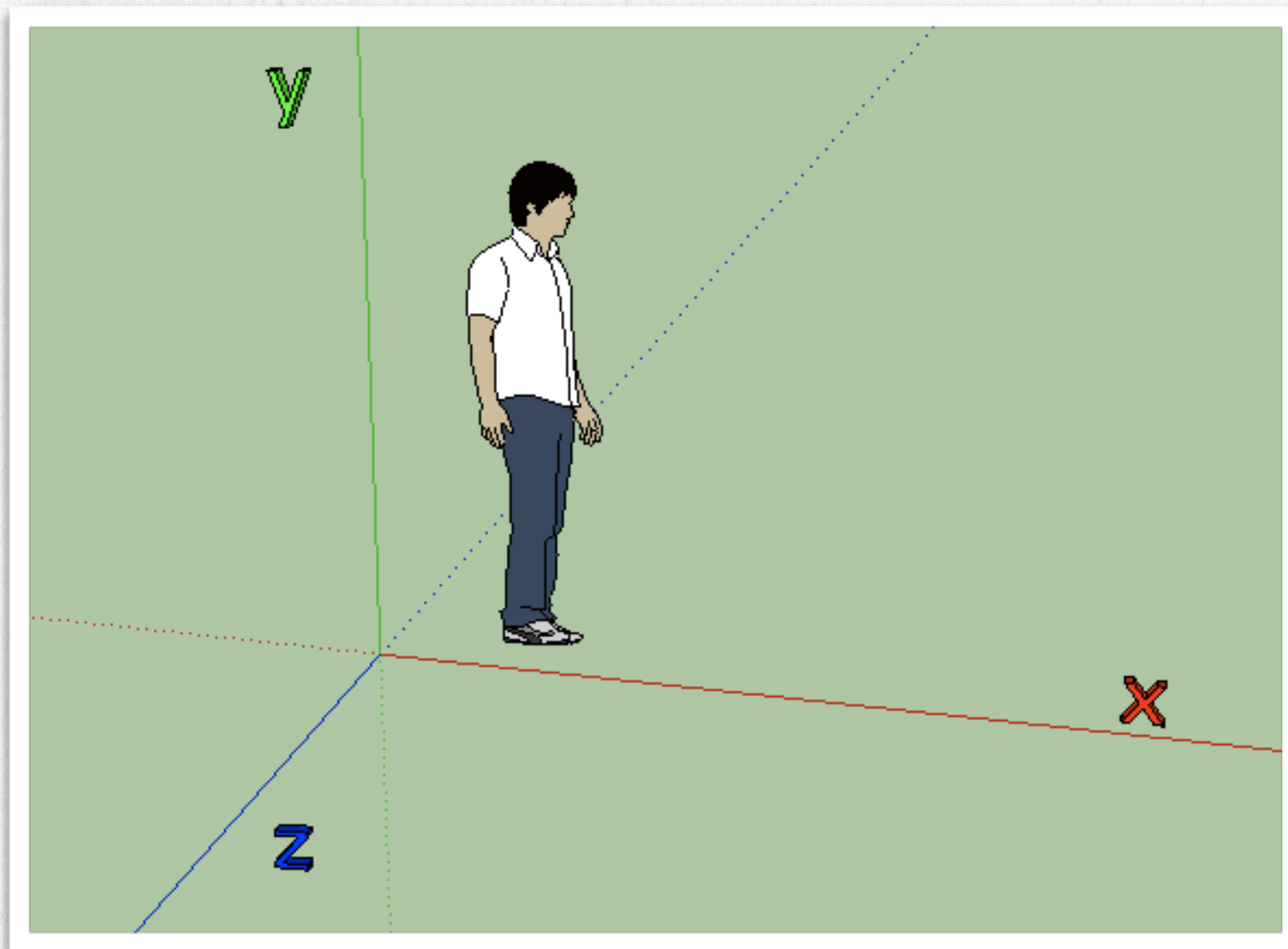


<http://o.tqn.com/id/comicbooks/1/0/a/K/bonecrusher.jpg>

Combinación

[del curso de Justin Solomon, Geometric Computing Group, Stanford University]

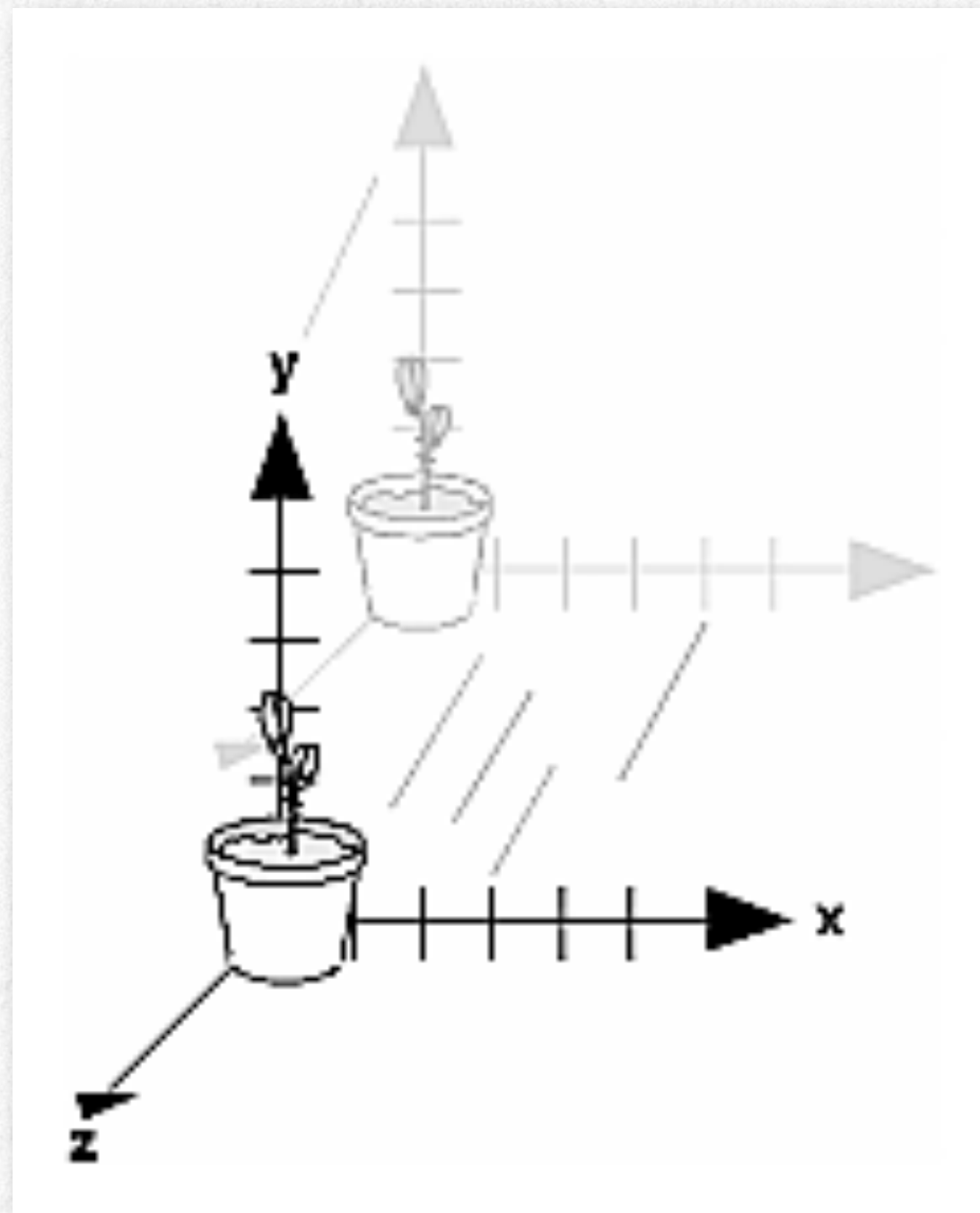
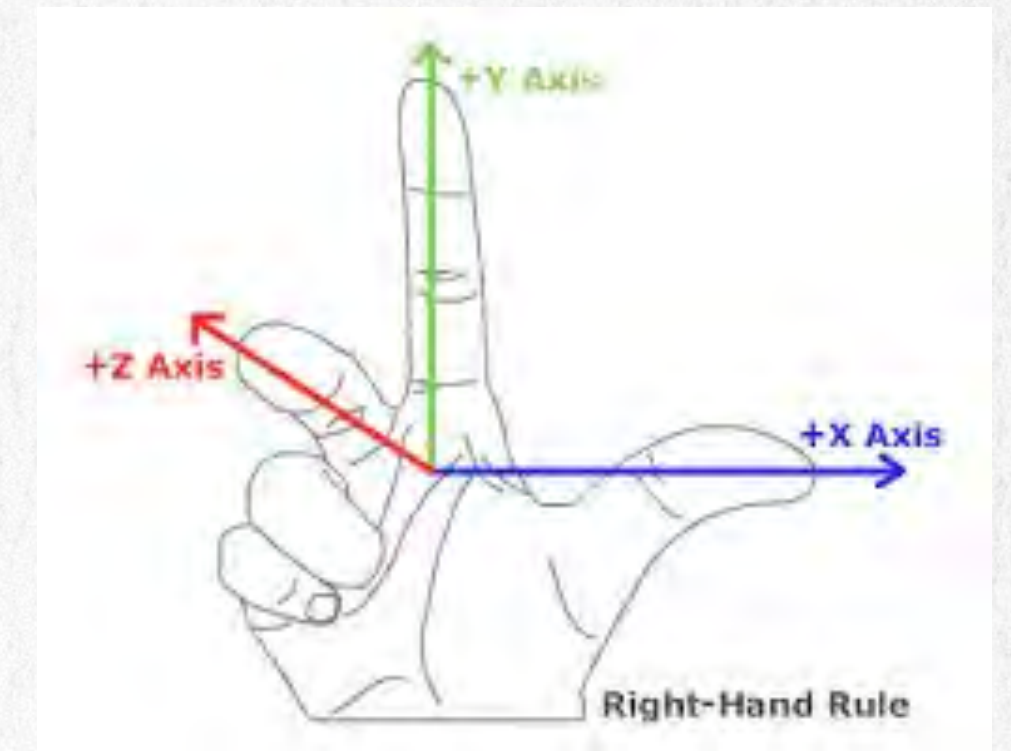
Marcos de referencia



[from Google SketchUp]

- > Sistema de coordenadas cartesiano: marcos de referencia.
- > marco de modelo (model coordinate system - MCS)
- > marco del mundo. (world coordinate system - WCS)
- > marco de vista o de cámara. (viewer coordinate system - VCS)
- > marco de despliegue o pantalla. (screen coordinate system - SCS)

Model coordinate system (MCS)



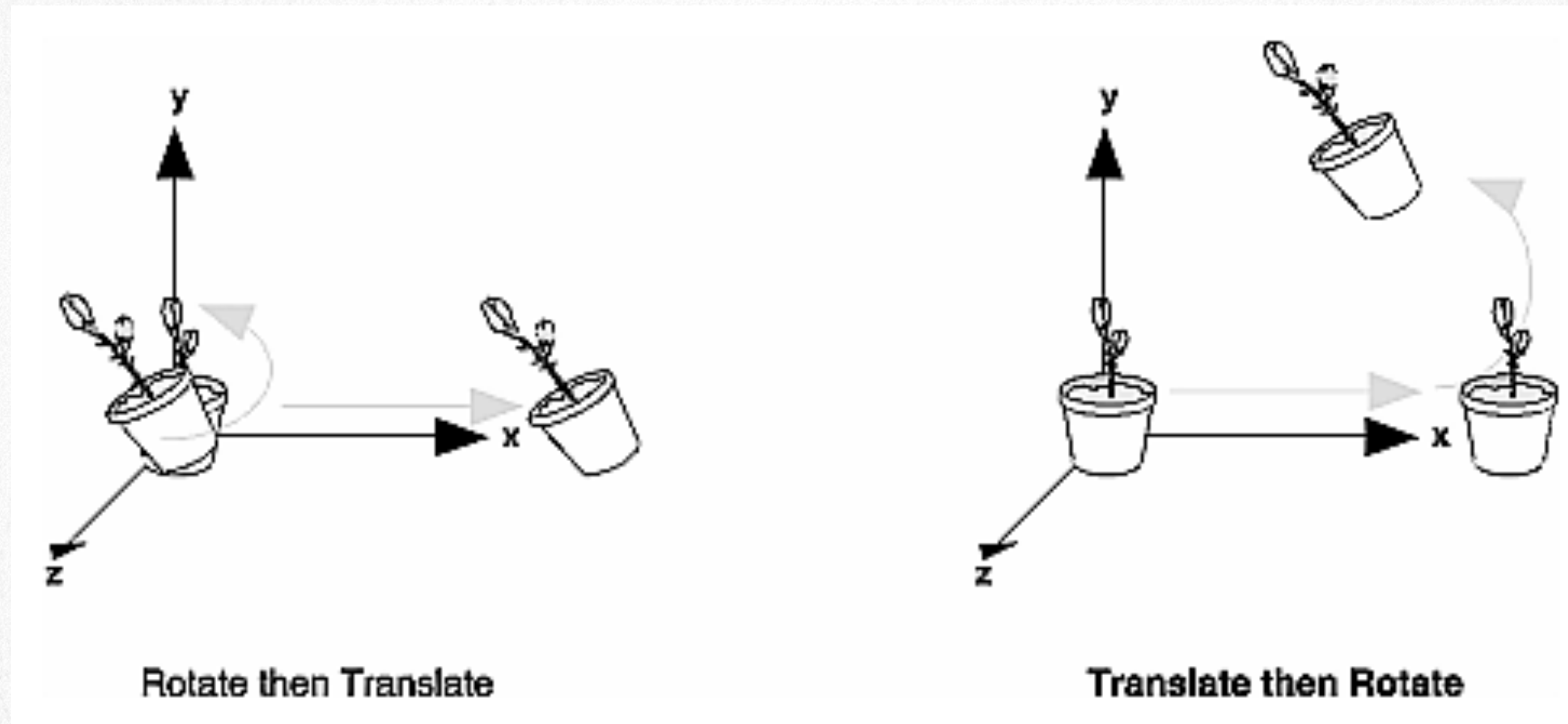
[from OpenGL Red Book. Ch 3.]

- > Cada objeto está centrado en su propio marco de referencia.
- > Su origen es algún punto en el modelo (centroide, un pie ...)
- > El modelo cuenta con un punto de control.
- > El objeto permanece constante ante este punto.
- > Si el punto se transforma (translación, rotación) el modelo se transforma también.

World coordinate system (WCS)

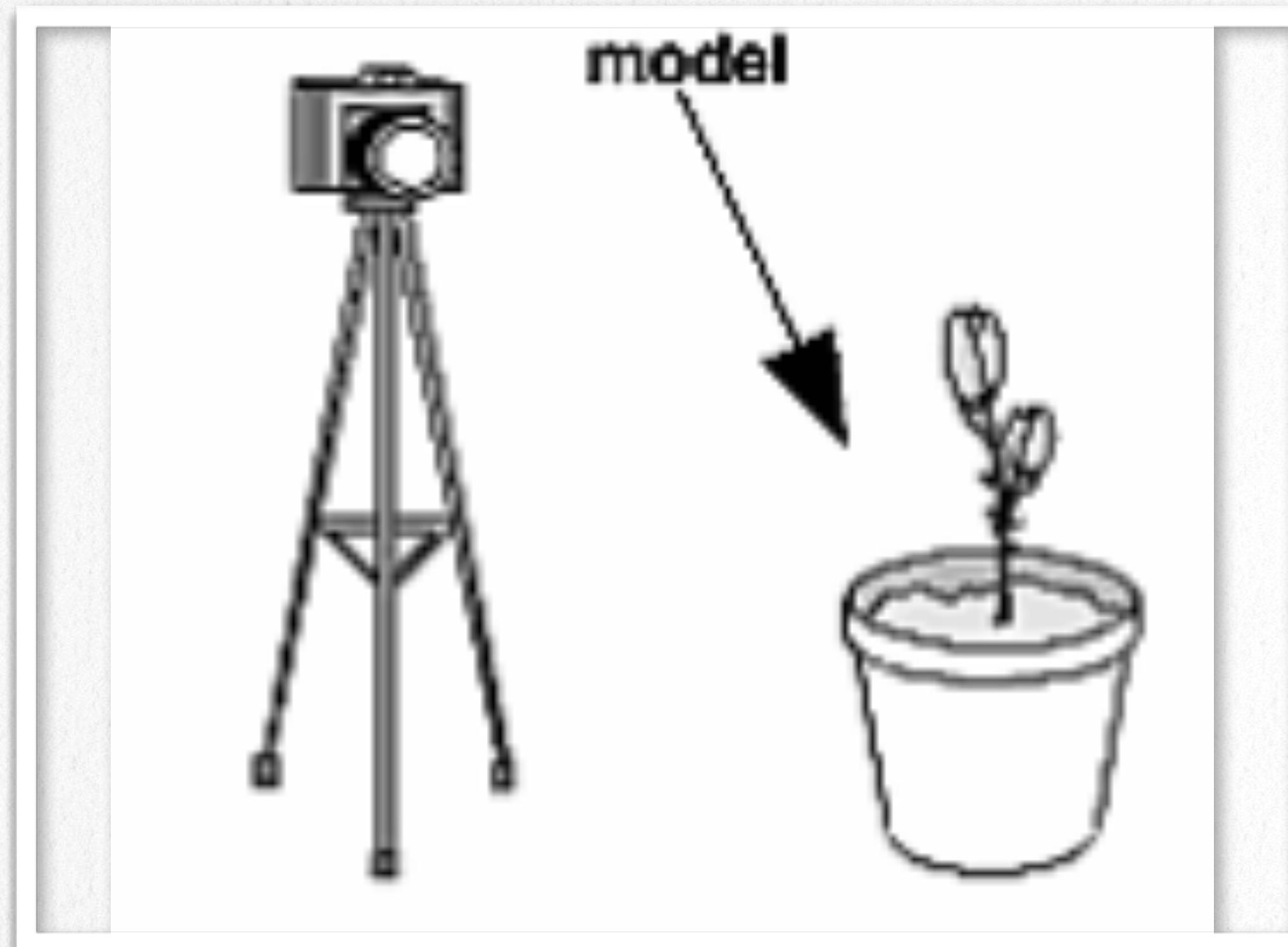
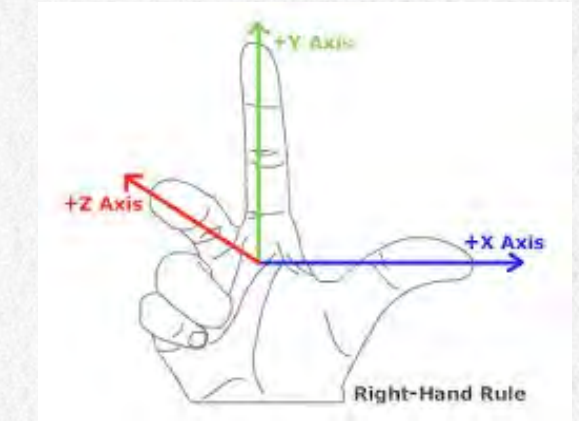
- Al colocar los modelos en el mundo 3D deben tener un origen común.
- Muchos programas 3D evitan esta transformación y van directamente al marco de la cámara (por ejemplo OpenGL)

Transformaciones



[from OpenGL Red Book. Ch 3.]

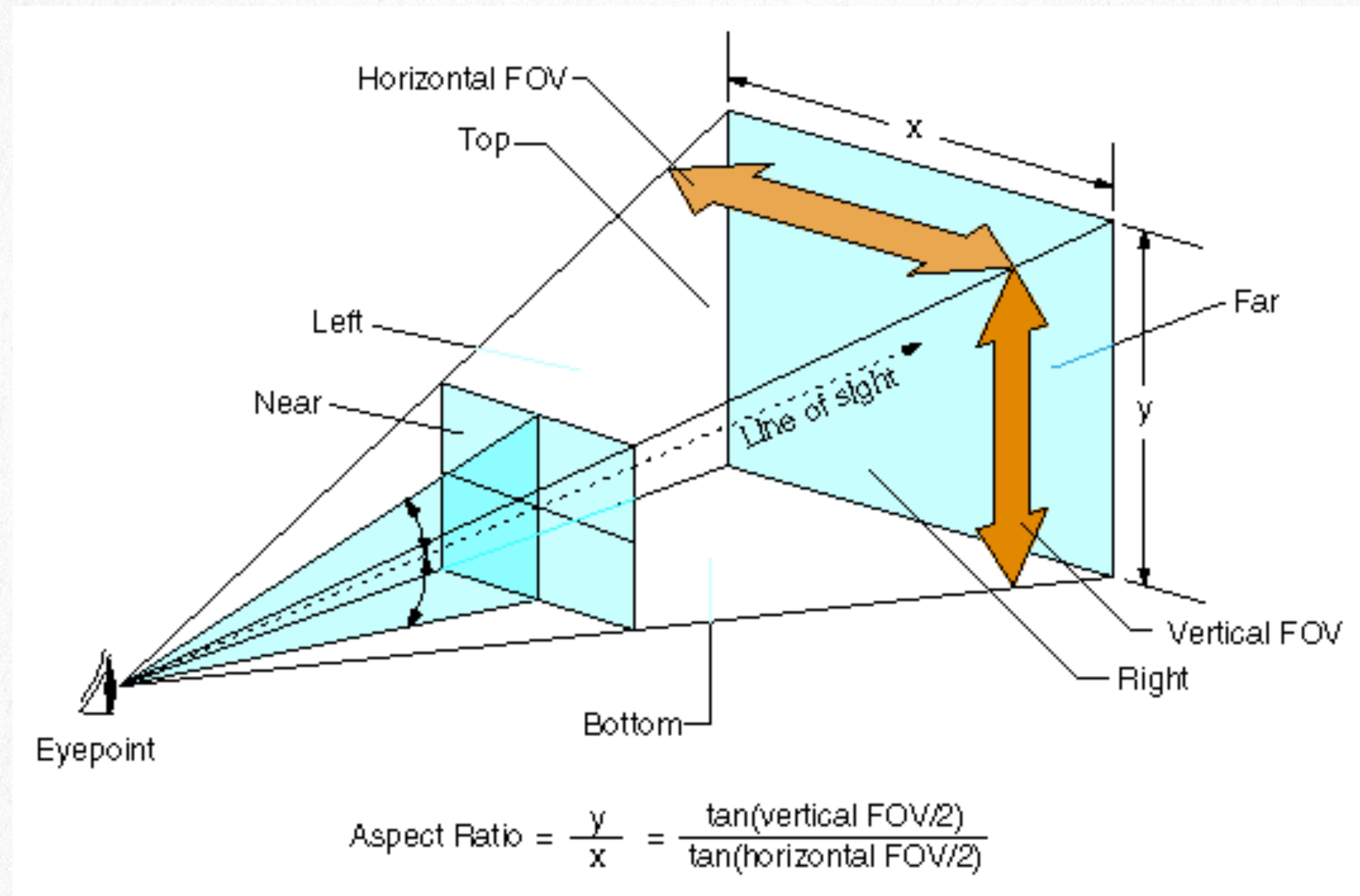
Viewer coordinate system (VCS)



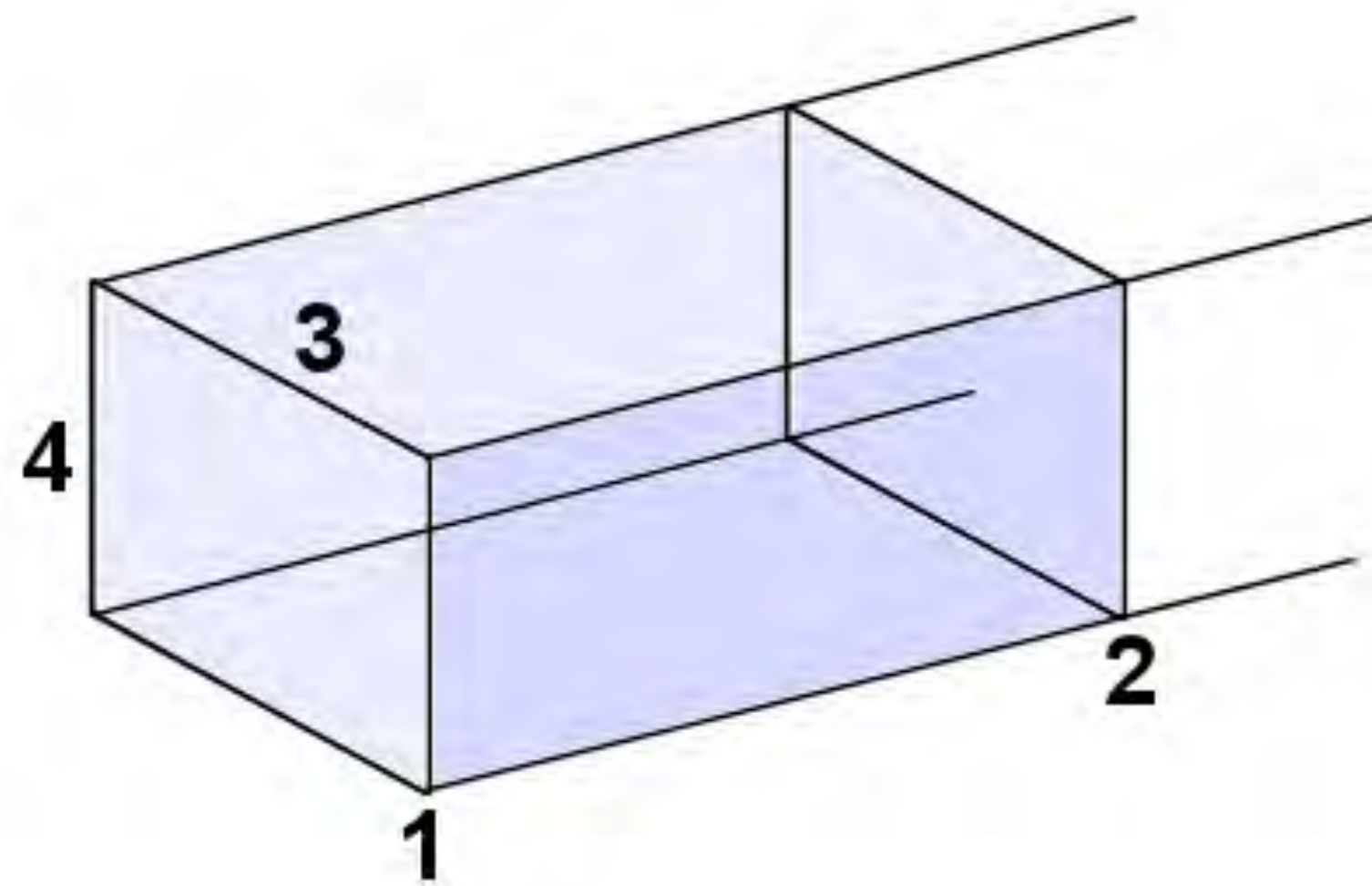
[from OpenGL Red Book. Ch 3.]

- > La cámara se coloca en algún lugar del espacio 3D unificado (mundo).
- > El ojo de la cámara se vuelve el origen del sistema coordenado, mirando hacia el fondo (eje-z) de la escena.
- > Las operaciones de culling, back-face culling e iluminación se realizan con esta referencia.
- > El volumen visible se produce con una proyección de la escena frente a la cámara: pensar en un proyector.
- > El volumen visible se define de acuerdo a los parámetros de la cámara.
- > La forma del volumen (frustum) será rectangular (proyección paralela u ortográfica) o piramidal (proyección perspectiva)

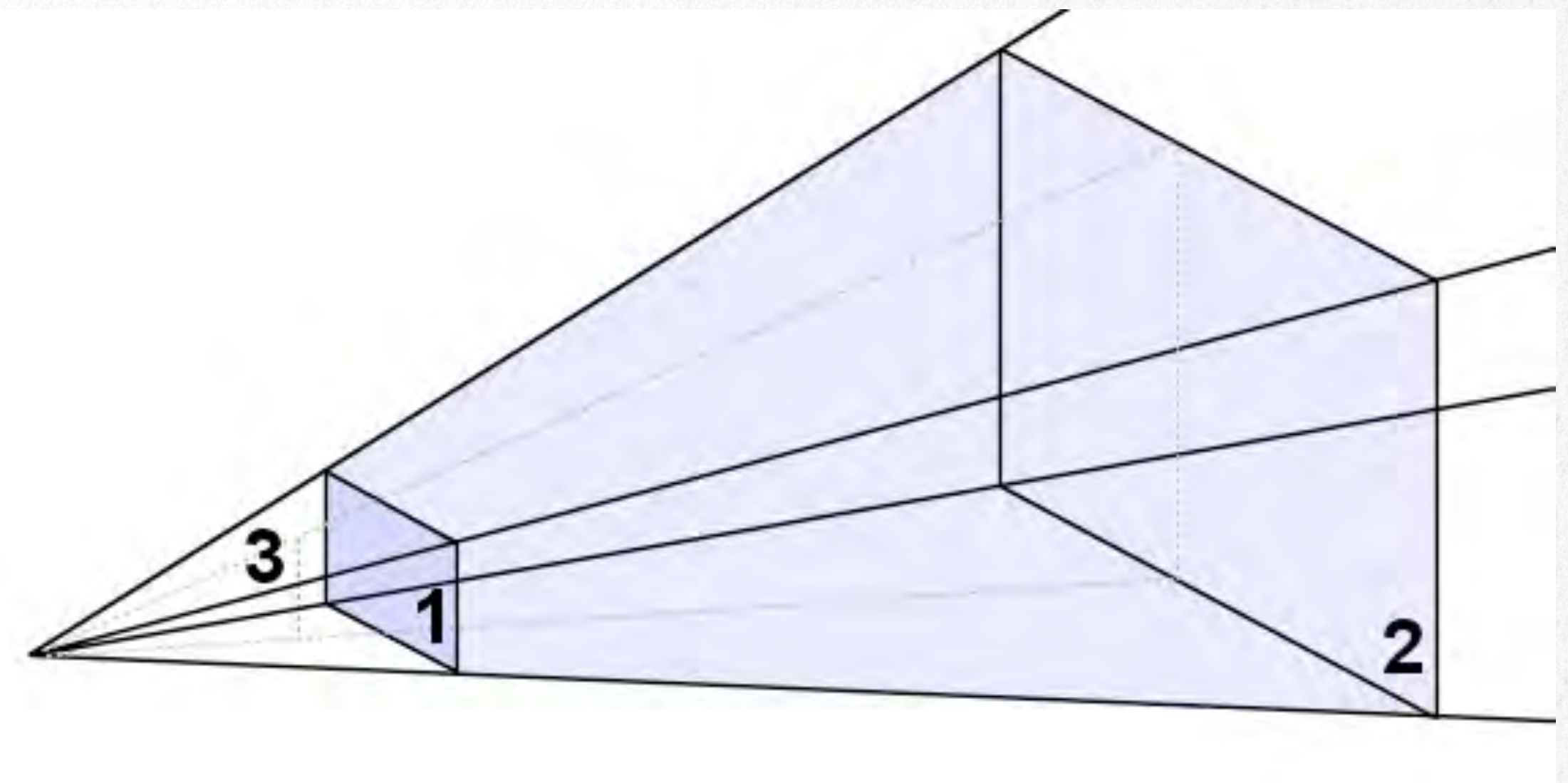
Viewer coordinate system (VCS)



Viewer coordinate system (VCS)



ortográfico o isométrico



perspectiva

Clipped coordinate system (Normalizado)

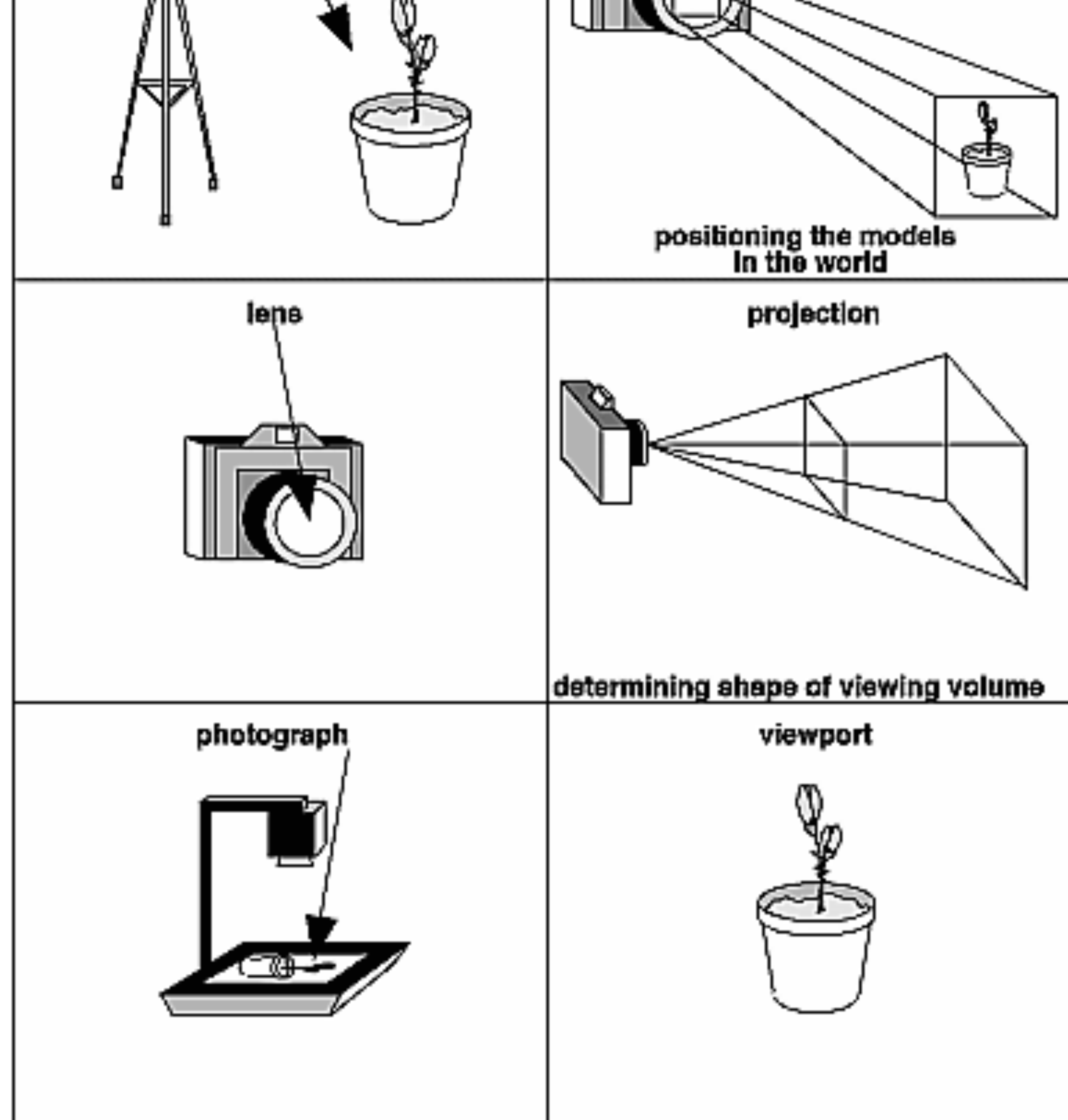
- Similar al sistema coordenado de vista o de cámara pero el frustum se adapta a un cubo unitario.
- El eje-x y el eje-y normalizados en un rango $[-1,1]$ y el eje-z en un rango $[0,1]$.

Clipped coordinate system (Normalizado)

- Similar al sistema coordenado de vista o de cámara pero el frustum se adapta a un cubo unitario.
- El eje-x y el eje-y normalizados en un rango $[-1,1]$ y el eje-z en un rango $[0,1]$.

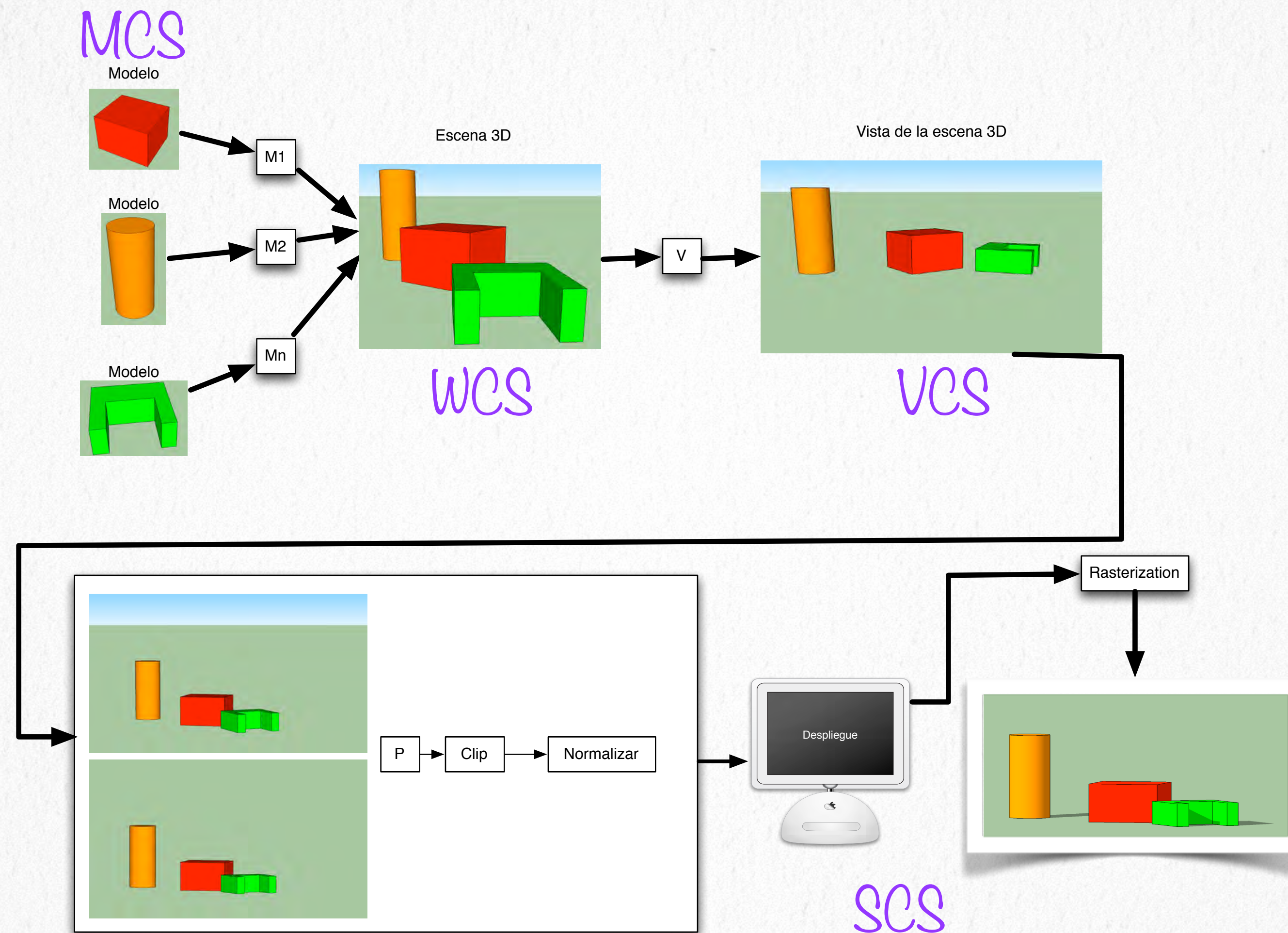
Screen coordinate system (SCS)

- Las coordenadas 3D de la escena se toman el eje-x y el eje-y como coordenadas de la imagen 2D.
- Las coordenadas del eje-z se guardan para las pruebas del z-buffer.
- En esta etapa no se han realizado todavía la transformación de la escena a pixeles (rasterización).



[from OpenGL Red Book. Ch 3.]

Graphics Pipeline



API de gráficos

- Las etapas de un pipeline se administran por una interfaz de programación de aplicaciones (Application programming interface o API)
- Ejemplos de API de gráficos: OpenGL, Microsoft's Direct3D, Pixar's Renderman ...
- El hardware gráfico se comunica con los APIs.
- Se hacen también capas de abstracción del hardware (HALs) pudiendo hacer al API independiente del dispositivo.

OpenGL

- Es un API con muchas funciones que permiten comunicarse con el hardware de gráficos.
- Multi plataforma.
- Da funciones para dibujar cosas.
- Utilizada en muchas aplicaciones de gráficos.
 - p.e. videos juegos, visualización.

OpenGL

- Diseñado para producir imágenes 3D de apariencia razonable de manera simple y rápidamente.
- Gran parte de su diseño es una aproximación burda de cómo los fenómenos visuales se comportan en el mundo real.
- Se ocupa solo de rendering no de manejo de ventanas, manejo de eventos, etc.
- Abstrae el manejo de los gráficos del hardware en una interfaz estándar y limpia.