# Overview

# *3D Production Pipeline*

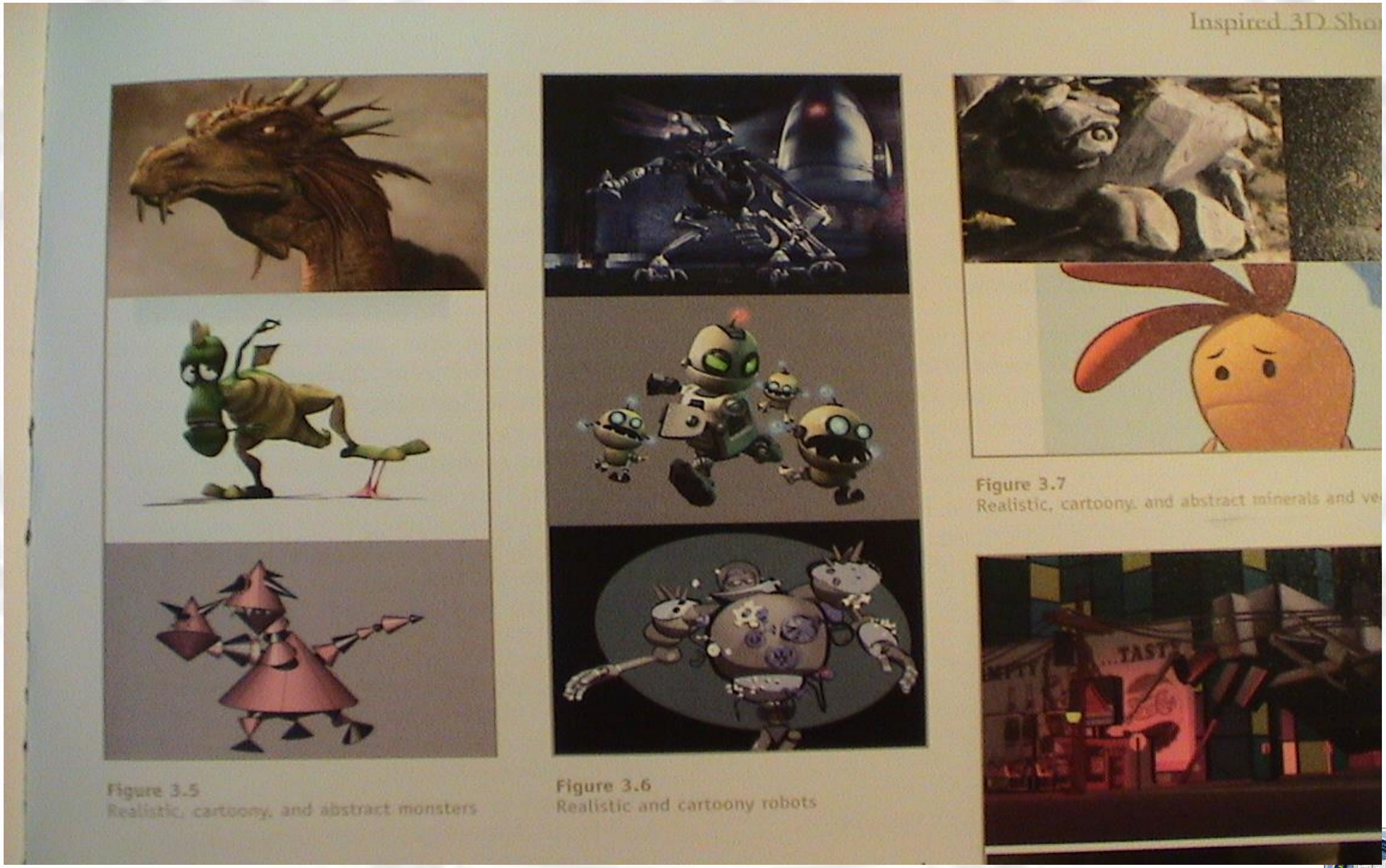
- Story → Character Design → Art Direction → Storyboarding → Vocal Tracks → 3D Animatics → **Modeling** → **Animation** → **Rendering** → **Effects** → Compositing
- **Basics** : OpenGL, transformation
- **Modeling** : curves and surfaces
- **Animation** : kinematics (FK/IK), shape interpolation
- **Rendering** : shader, file texture, raytracing
- **Effects** : pariticle systems, soft boy, rigid body, hair

# Story

- CG vs. non-CG
- 2D vs. 3D
- Mixtures (Lord of the Ring, Harry Potter, Who Frames Roger Rabbit, Avatar) vs. complete (Mr. Incredibles)

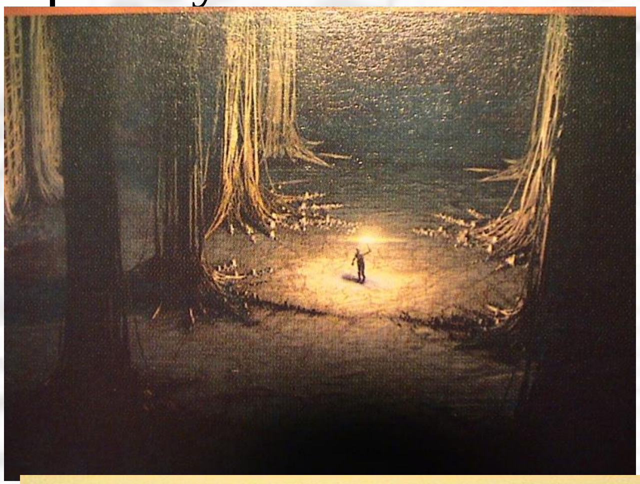
# *Character Development*

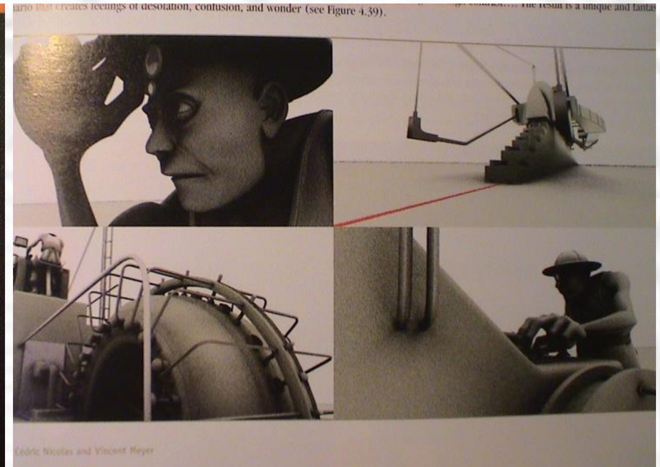
- Style, proportion, different poses and emotions, clay models, anatomical study, behavior, etc. (Add life)



# *Art Direction*

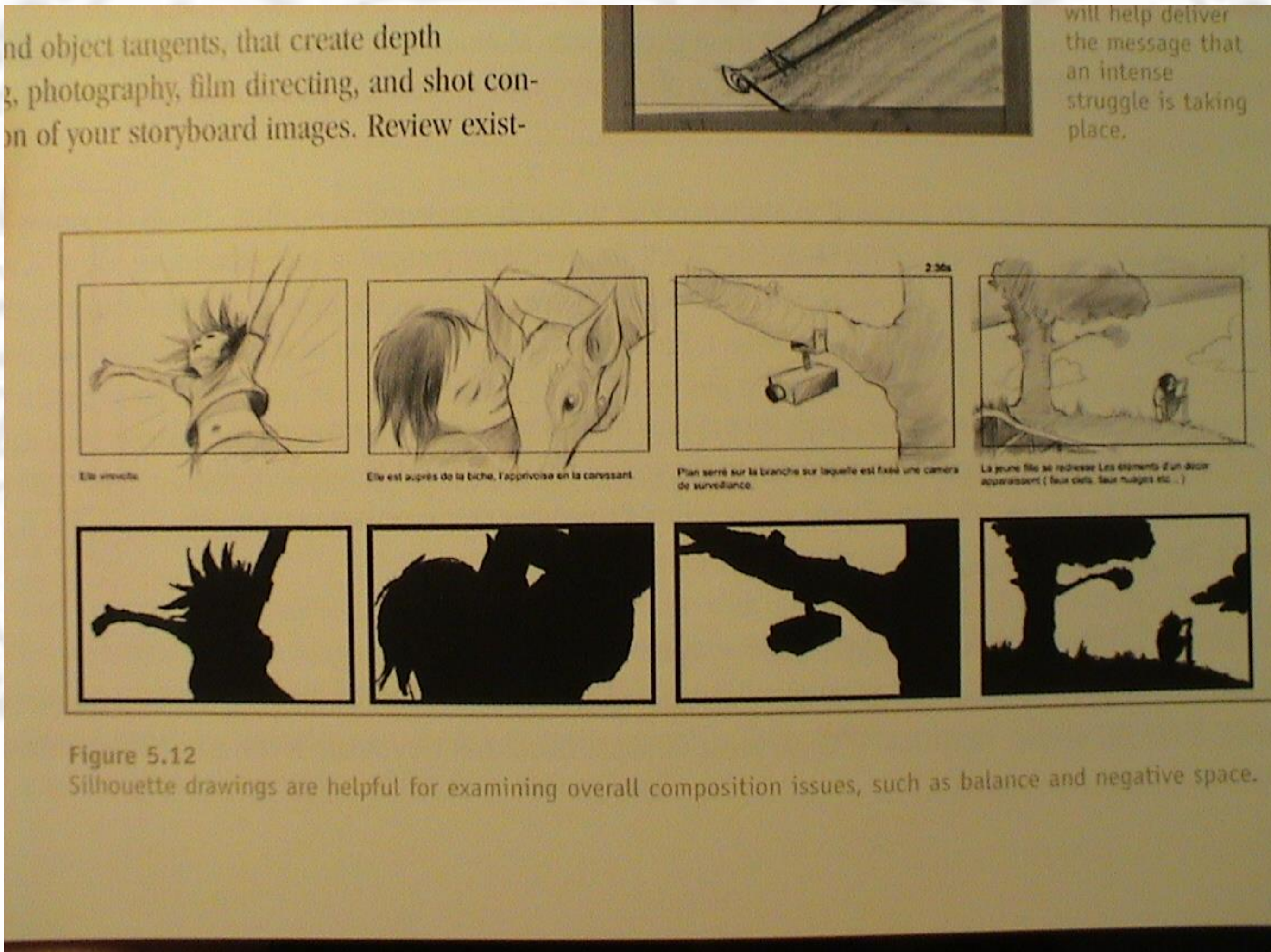
- Visual style (realism, cartoon, abstract), color palettes, overall complexity





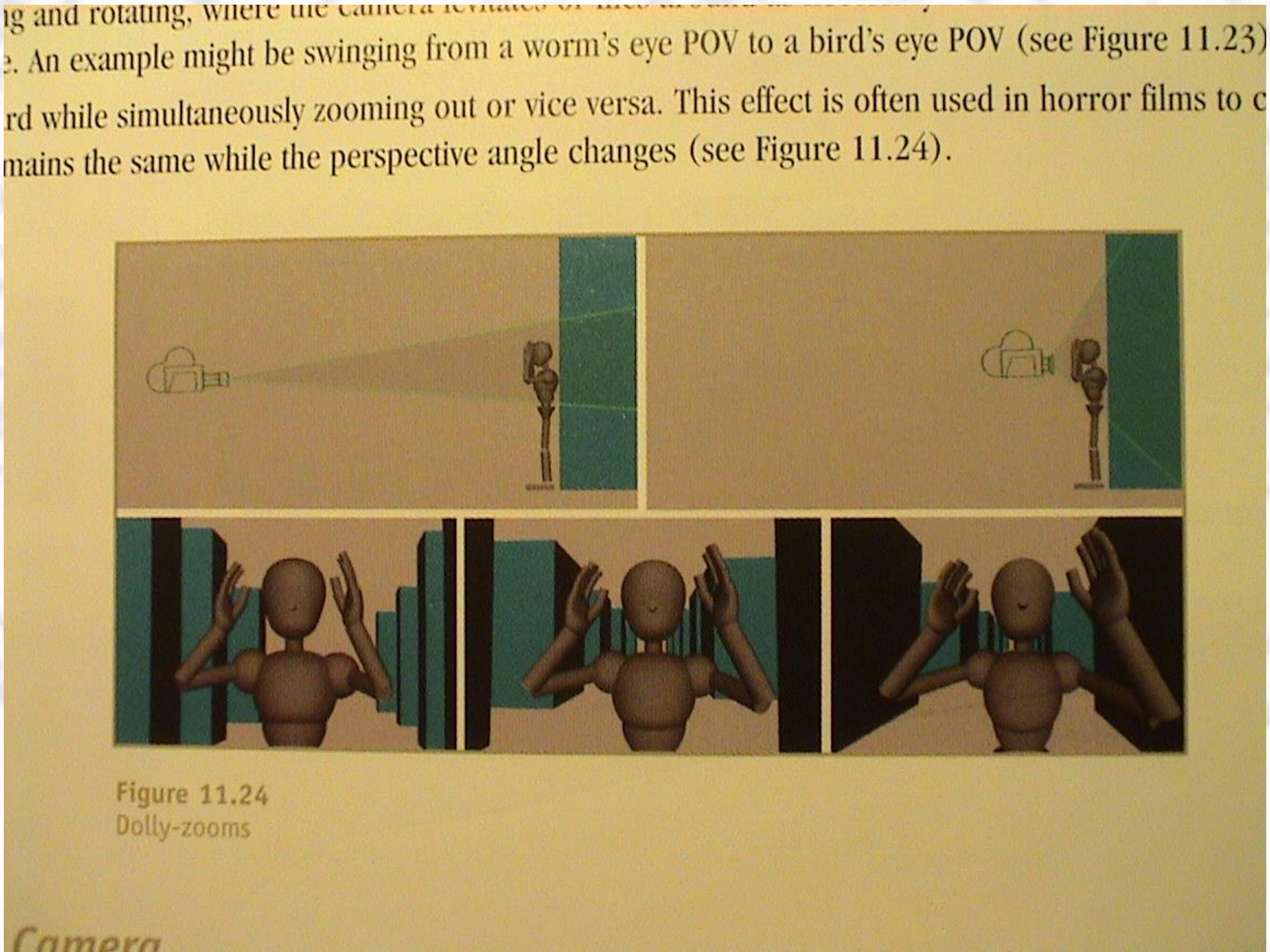

# *Storyboarding & Vocal Tracks*

- Tell the story visually (beats, flow, tightness), planning shots, camera, layout, etc.



# 3D Animatics

- Planning shots, camera, layout, layering with billboards and simple geometry, only animate the camera



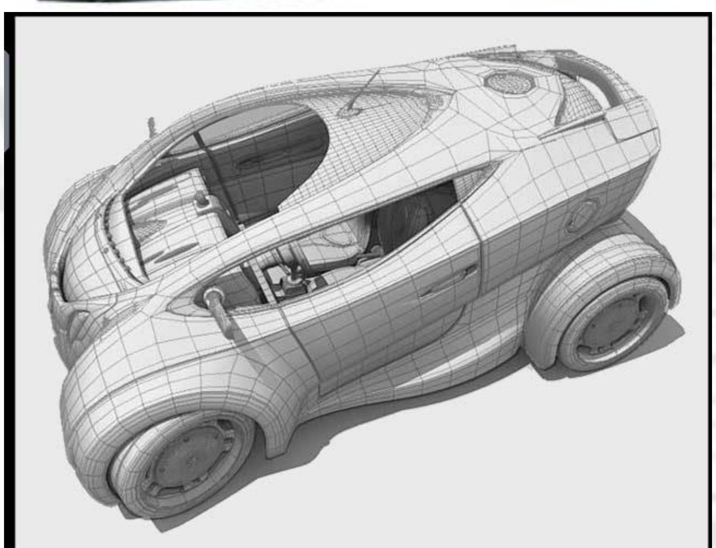
ng and rotating, where the camera ...
. An example might be swinging from a worm's eye POV to a bird's eye POV (see Figure 11.23)
rd while simultaneously zooming out or vice versa. This effect is often used in horror films to c
mains the same while the perspective angle changes (see Figure 11.24).

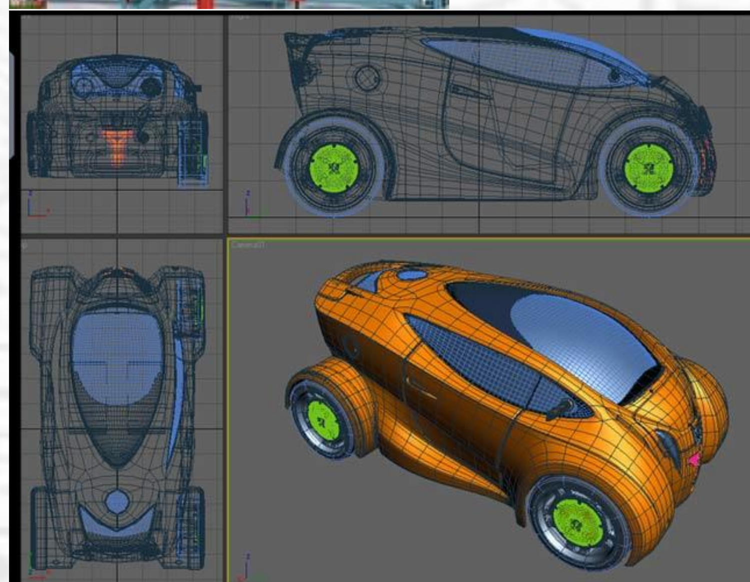
Figure 11.24
Dolly-zooms

Camera

# *Modeling*

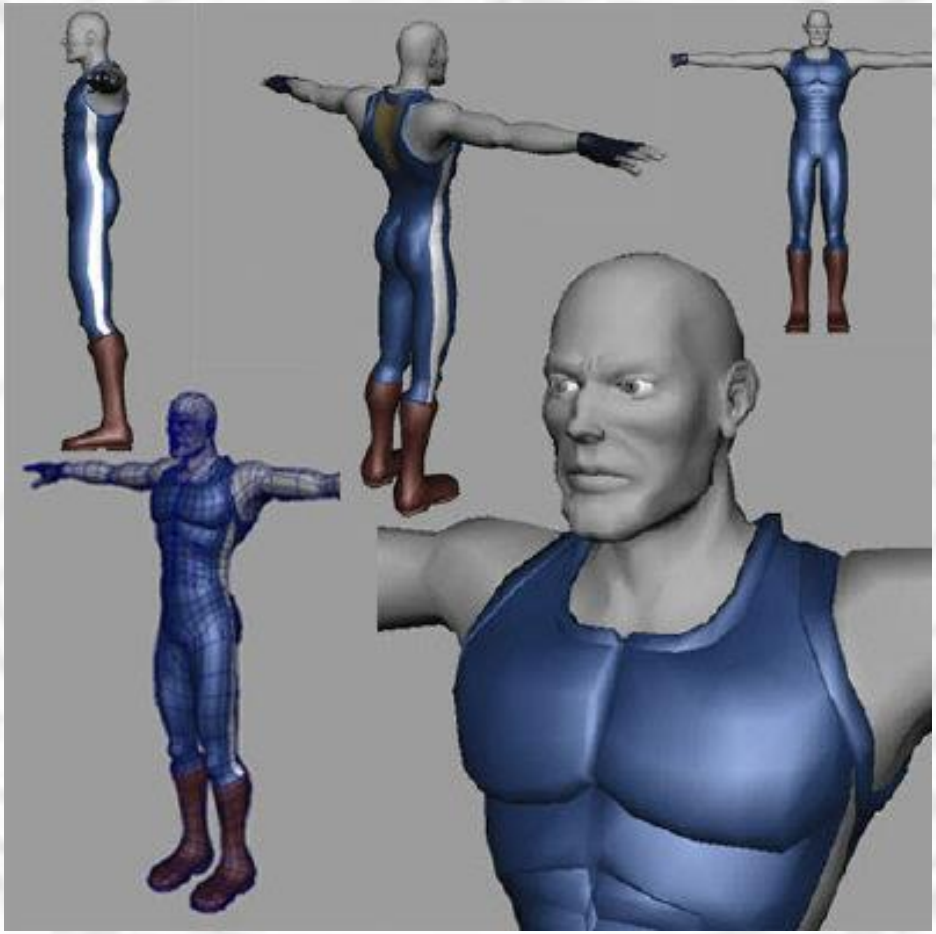
- Characters, props, and background elements, low & high resolution, models, 3D scanners



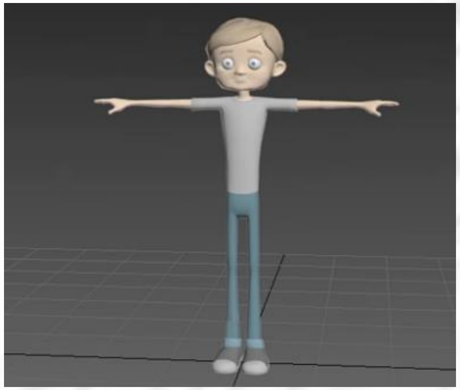


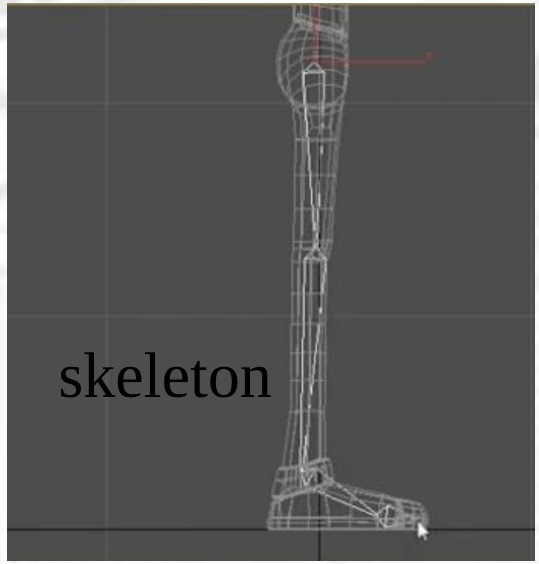
# Character Setup

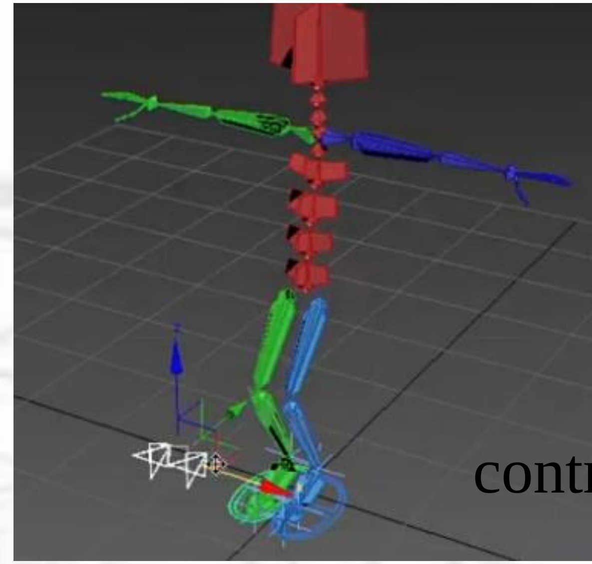
- (Character rigging) : to setup IK or FK joints, skinning, blend, shape, deformer, skinning

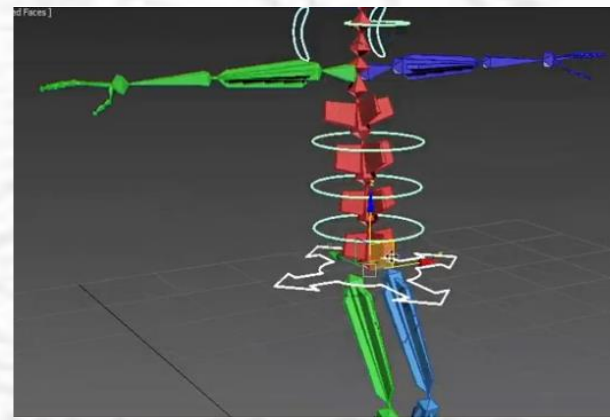


# *Rigging*

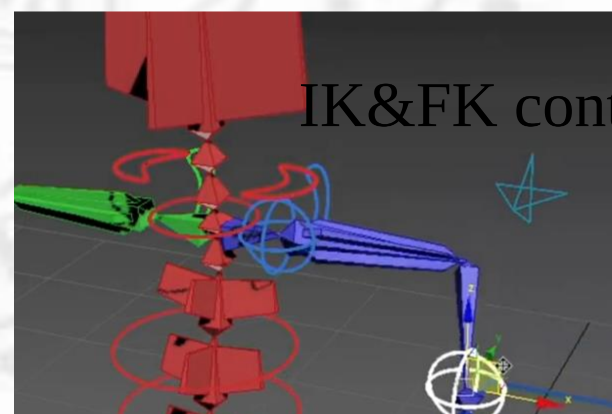




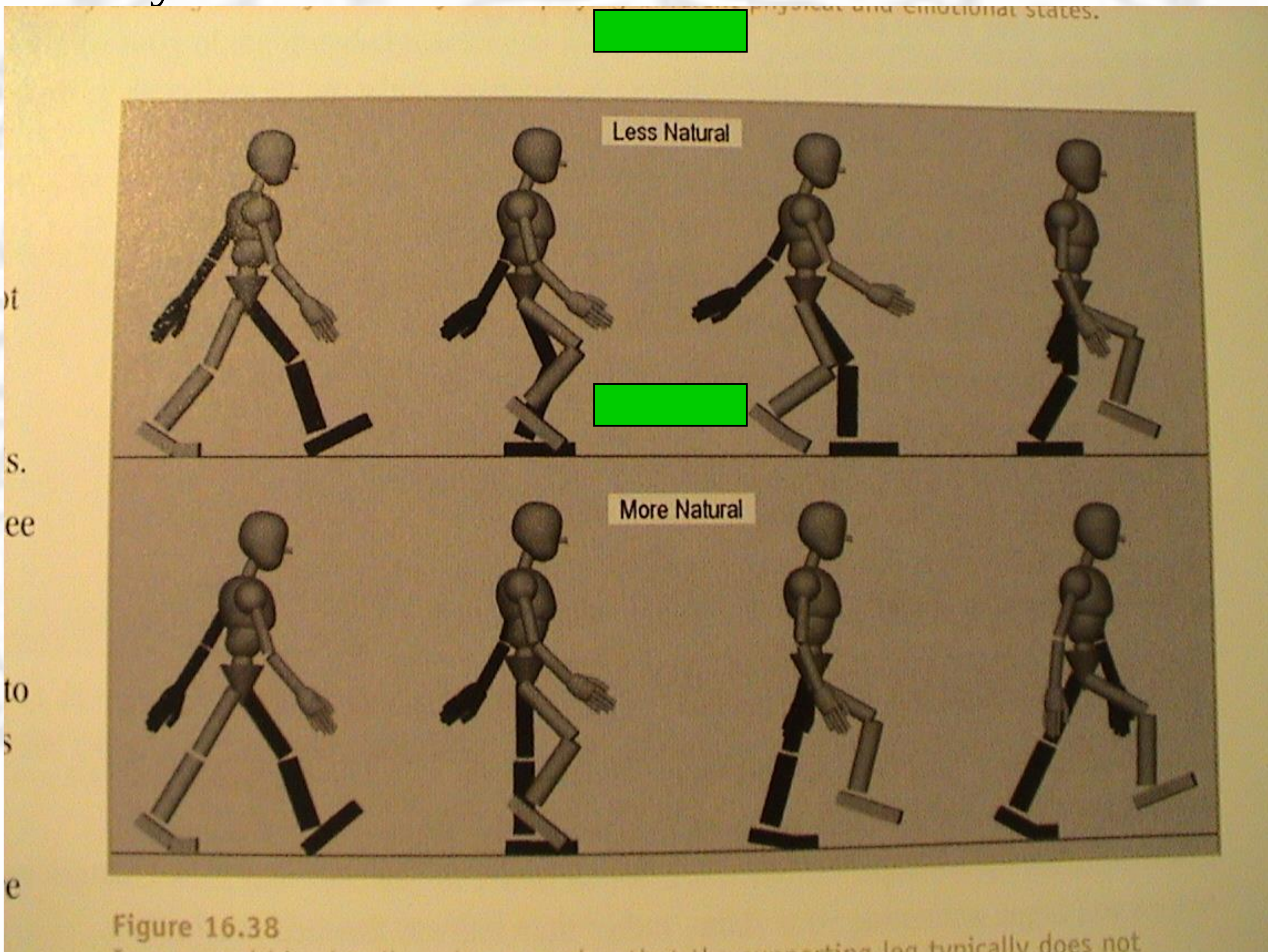



Facial control



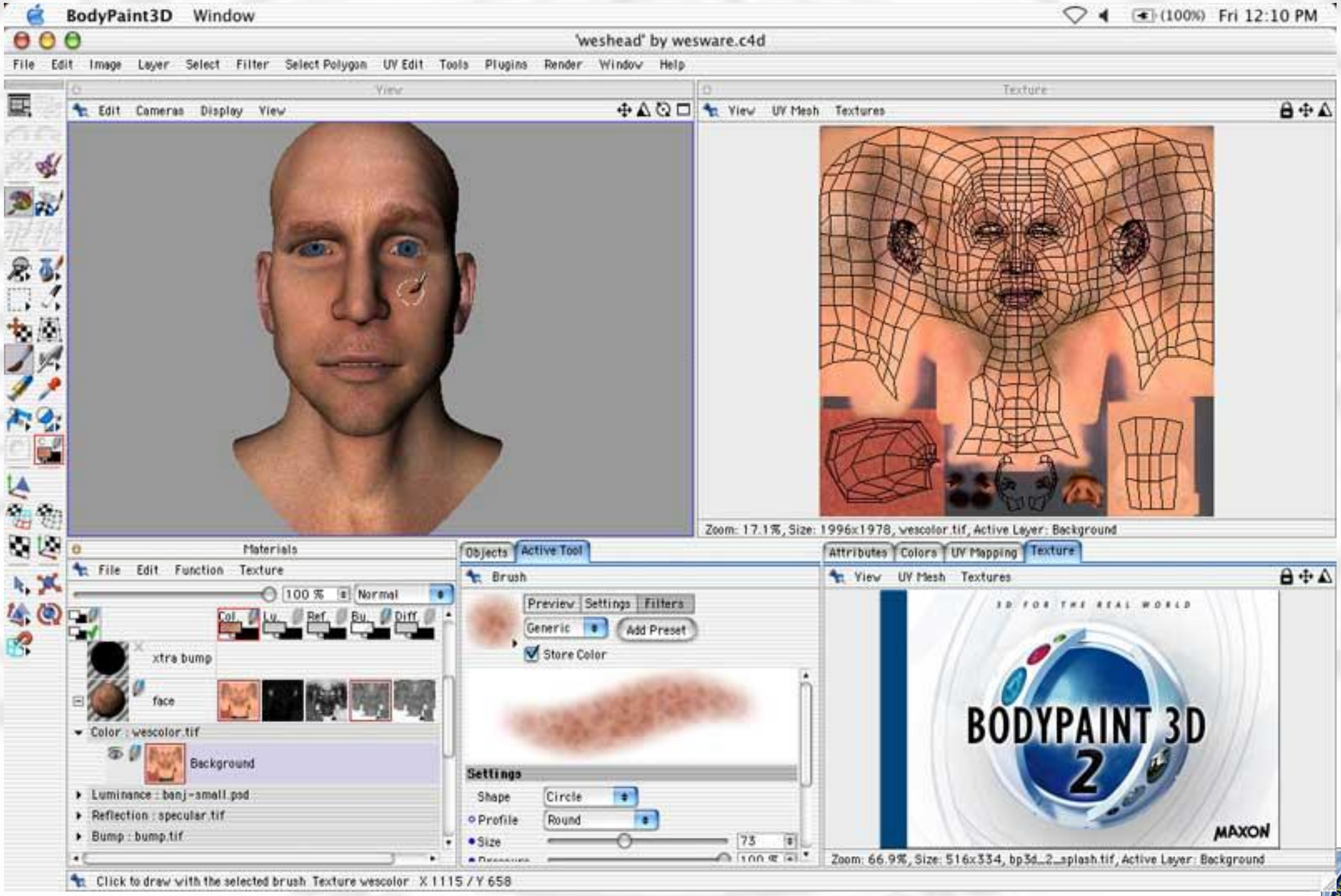
# *Animation*

- Low resolution model, blocking, timing, details such as secondary motion

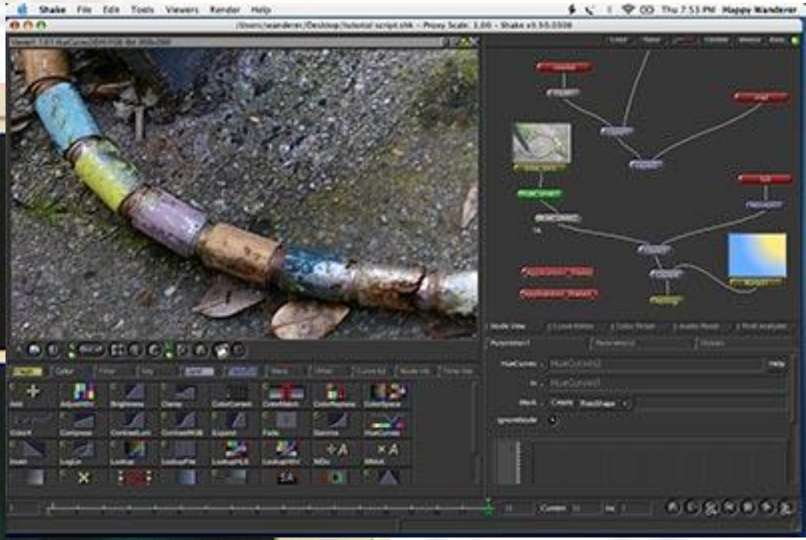


# Visual Effects (FX)

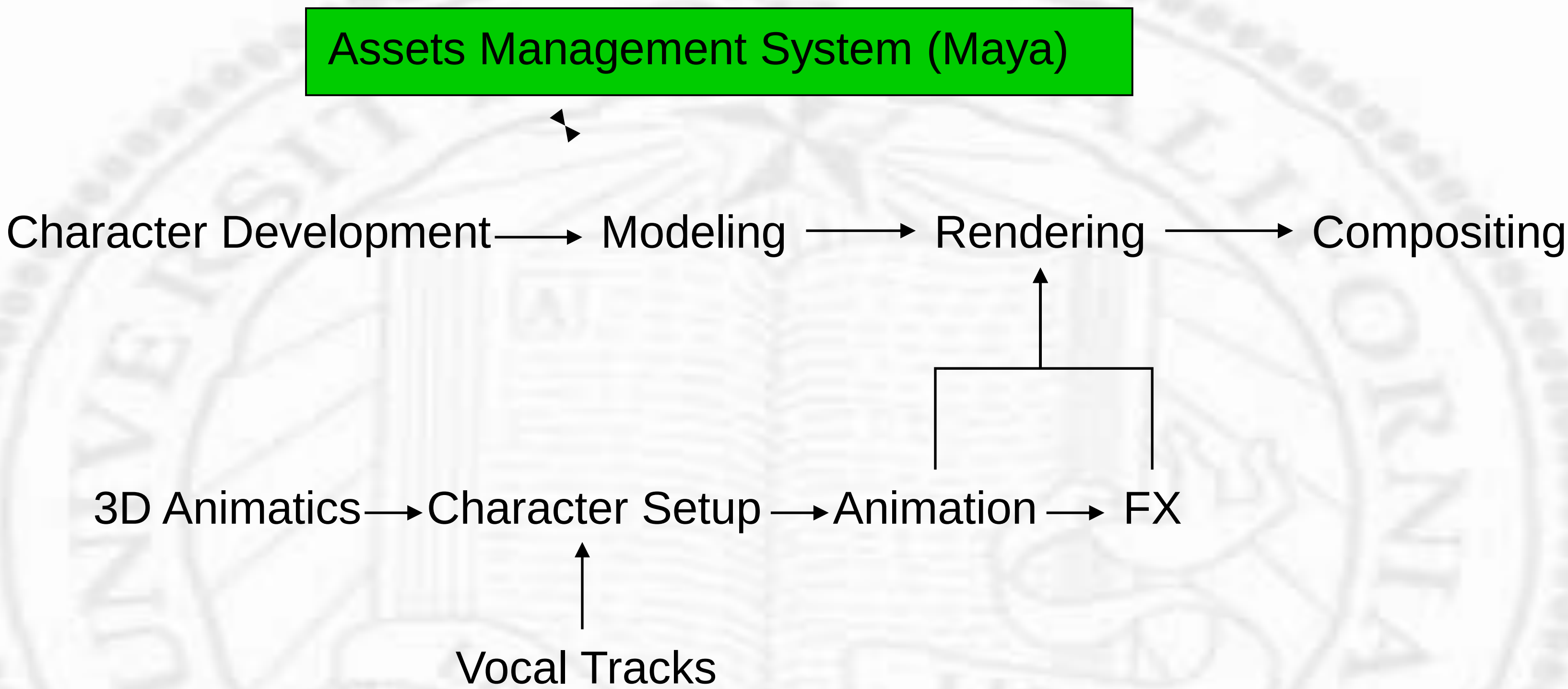
- Shading, Texturing, Lighting, Rendering : writing shaders, assigning materials, testing global illumination approaches, baby sit rendering farm



# Compositing : multi-pass



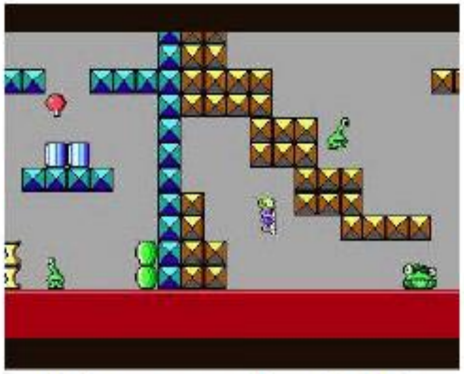
# *Movie Pipeline (Non-realtime)*



# Game Pipeline

- Art vs Game Engine
- Art pipeline is very similar to film production pipeline
- Game Engine
  - Put together the animation and stages into the virtual world
  - Mimic tools in 3D package (memory and performance constraints)
  - Taking advantage of the state-of-art graphics hardware
  - Using 3rd party game engine components
  - Physics are important (Angry Bird, Bowling, Flight simulator, etc.)

# *Image Quality*

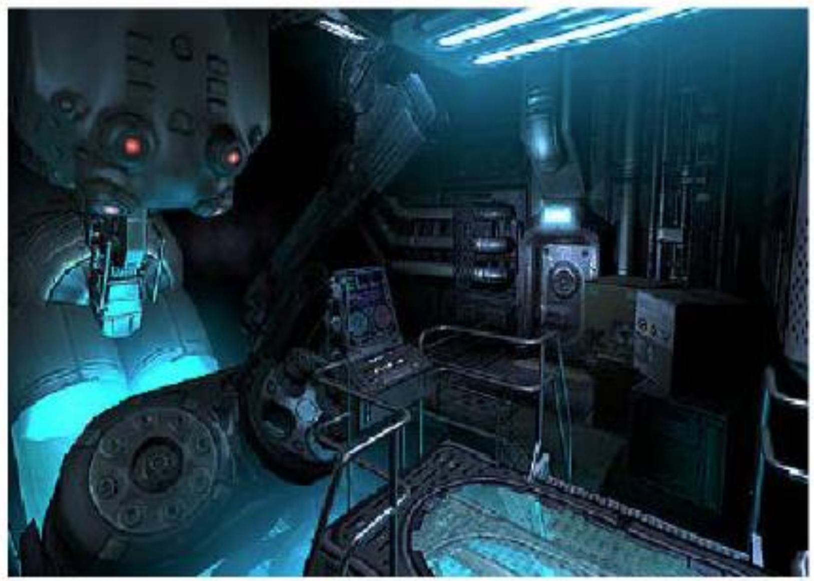


Marooned - 1990

Hovertank 3D - 1991

Doom - 1993

Quake - 1996



Doom3
2004

# *Pipeline Requirements*

- Minimize the amount of data being passed around (separating model & animation)
- Use multiple 3rd party tools and internal tools (File format)
- Performance (File referencing : save only the changes, work at different resolutions)
- Controlled access
- Handling blind data
- Data protection (Versioning system)

# 3D Software

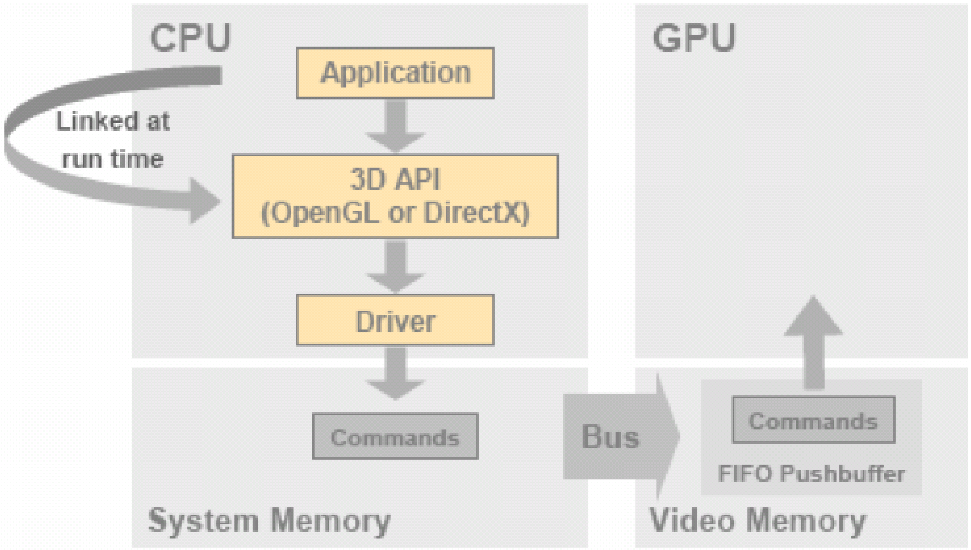
- Complete tool sets
  - Modeling, Animation, Rendering, Cloth, Dynamics, Fluids, Hair, etc.
- Graph architecture with node as black box
- Pull model and dirty propagation
- Refresh and getting an attribute to trigger graph evaluation
- Undoable commands
- Scripting language
- Run in interactive, prompt, and batch modes
- API (application programming interface)
- Powerful UI paradigm
- Interpreted via scripting, marking menu, hot key
- Alias/Wavefront (Maya), Autodesk (3D Studio Max)

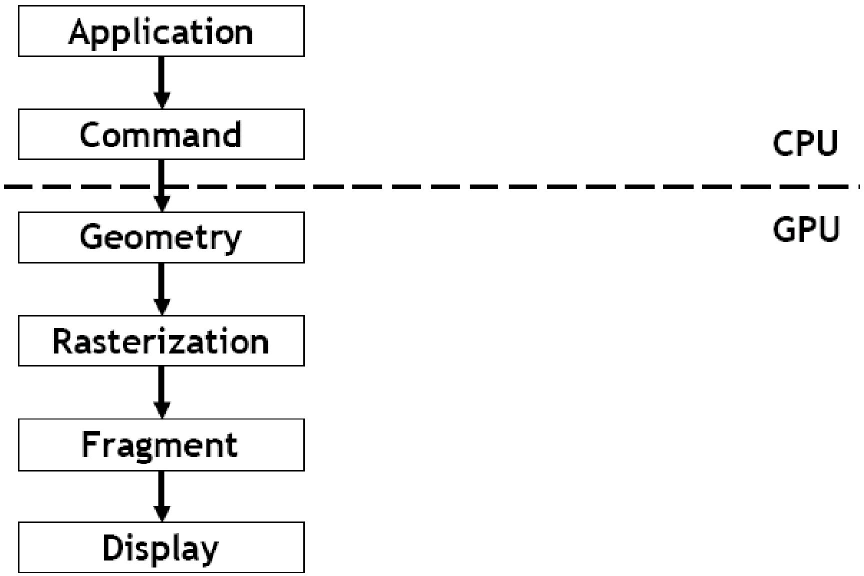
# *Who is Who*

- CG Film Industry
  - Pixar – Toy Story I & II, Monster Inc., Bug's Life, Incredible, Cars
  - Pacific Data Image – Ants, Shrek, Madagascar
  - DreamWorks – Shark Tale, Over the Hedge
  - Disney – Chicken Little, Toy Story III
  - Blue Sky – Ice Age, Robots
- FX House
  - ILM (leader in FX) – Star Wars
  - Weta – Lord of the Rings, King Kong
  - SONY Image Works (animal, fur, motion capture) – Stuart Little, Polar Express
- Game Industry $$
  - Learning from film production, shorter time frame
  - Electronic Arts (EA) sports games
  - Activision, Microsoft, Nintendo, Sony, Lucas Arts

# *Future*

- Performance and memory issues with fluids (GPU?)
- Still way too much effort to make 3D animation
- Unified solver
- Build in intelligence so that the secondary animation is handled automatically



# *Graphics Hardware Pipeline*

(input) triangles → vertex transformation → (output) transformed vertices

(input) transformed vertices → rasterization → (output) pixel location stream

(input) pixel location stream → fragment process → (output) frame buffer

Vertex shading with constant vertex color

Programmable vertex shader (Nvidia's Cg)

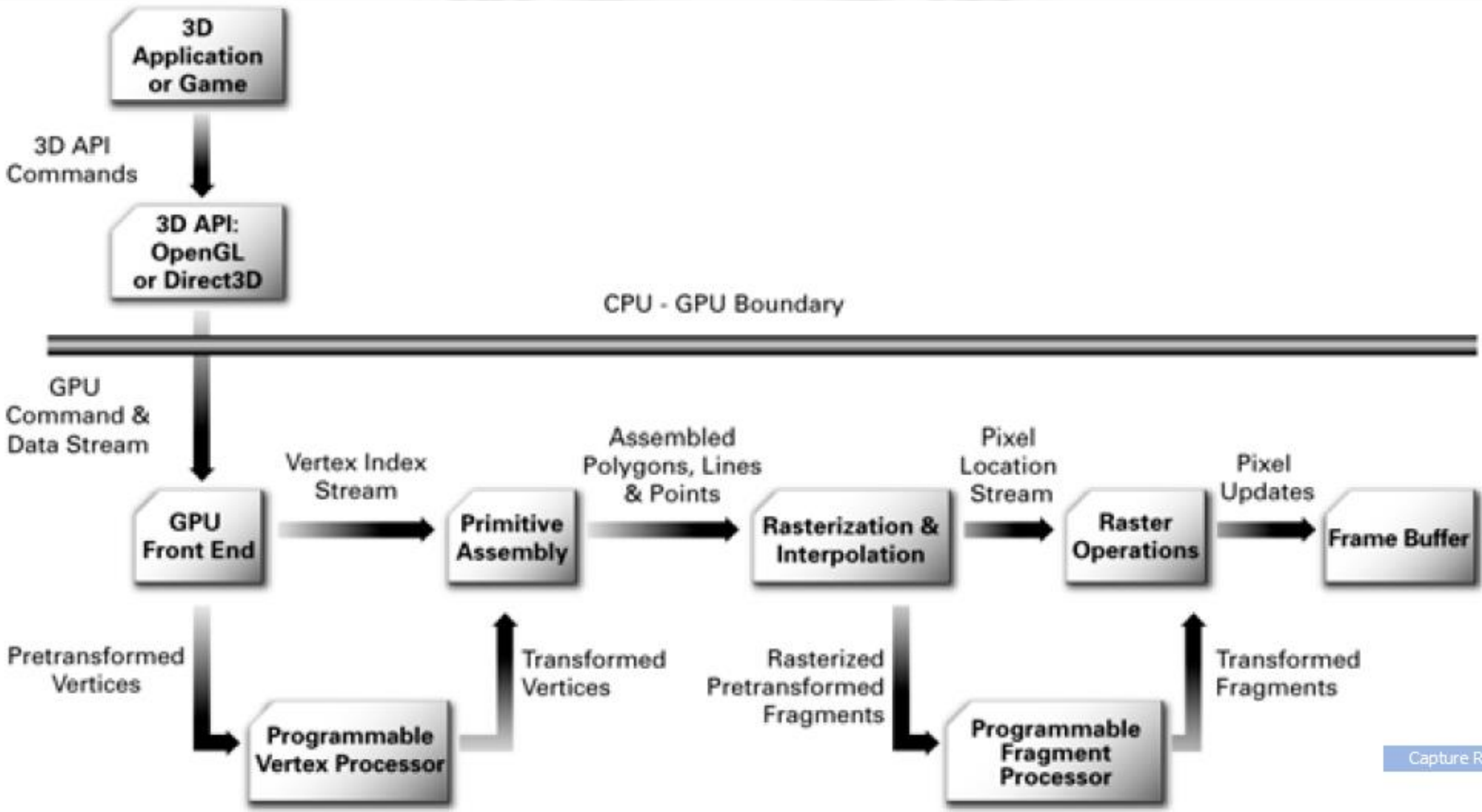
Pixel shader (NVIDIA's Cg)

Read Cg Tutorial Chapter 1, p.13 – p.20, available at

http://download.nvidia.com/developer/cg/Cg_Tutorial/Chapter_1.pdf

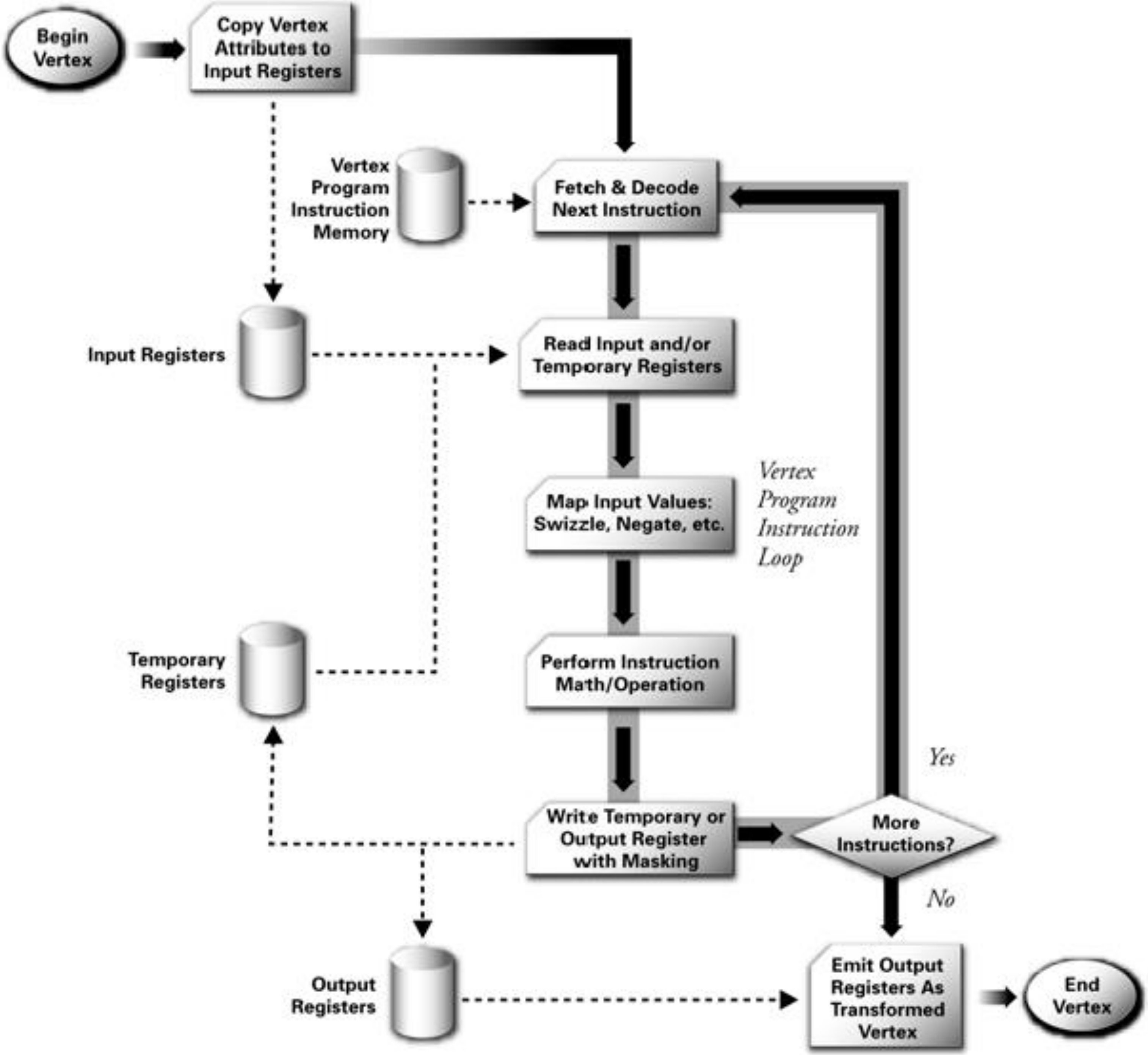
- Hardware graphics is super fast. However, its single precision is not sufficient to handle large scene or give enough depth precision for compositing.
- The trend is hardware accelerated software rendering and GPU programming.

# Programmable GPU

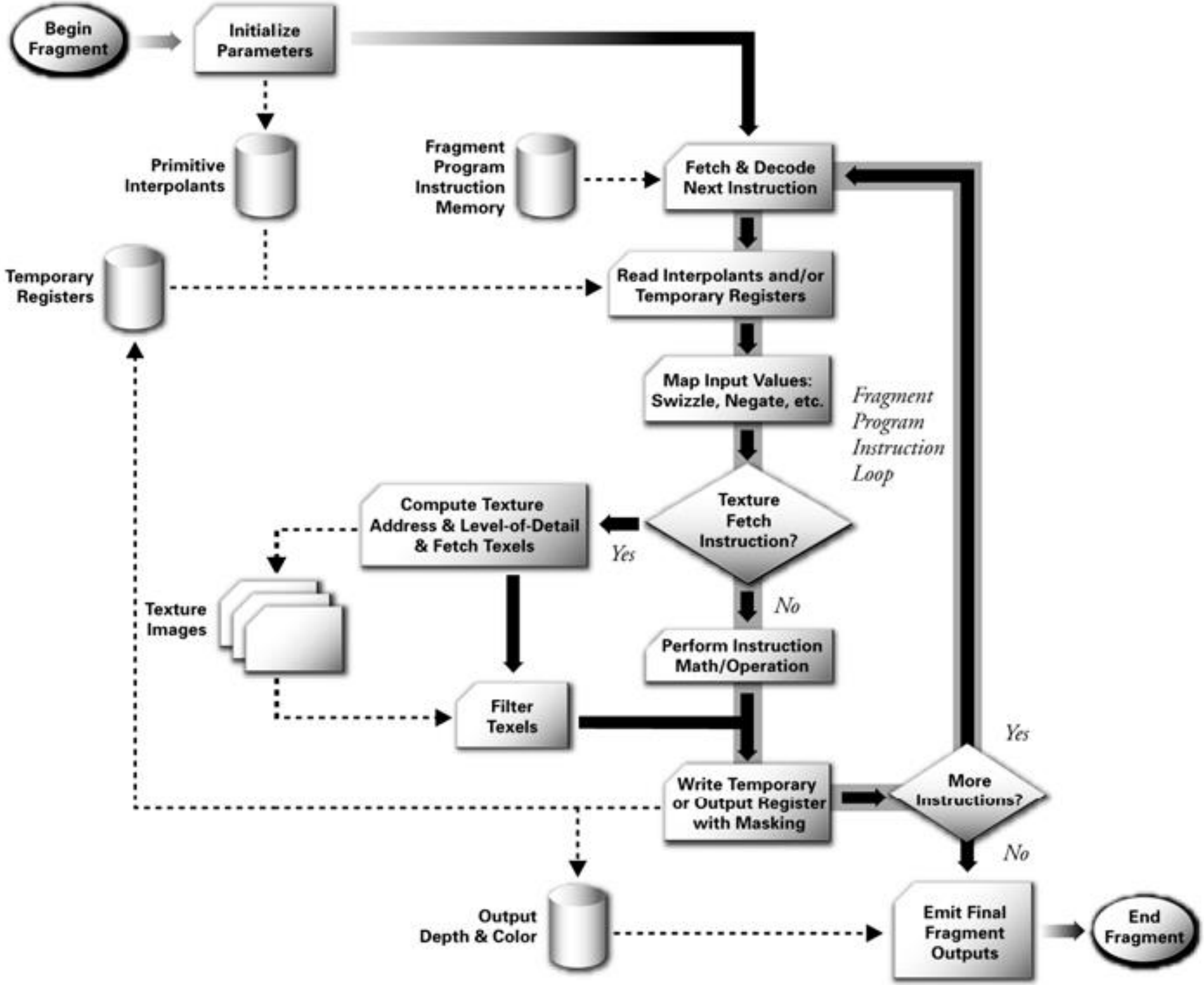



Capture Rec

# Vertex Processing



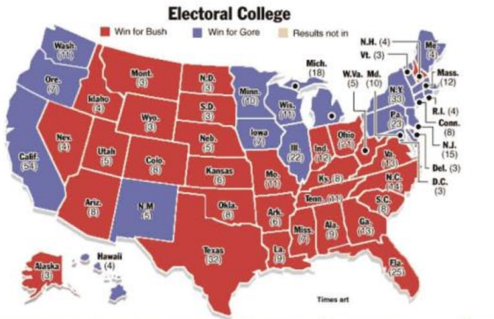
# Fragment Processing

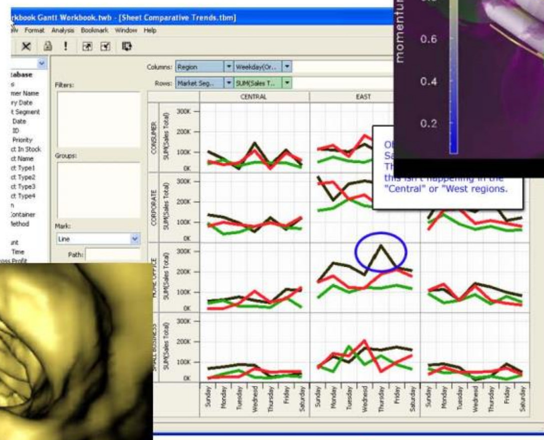


# Movies

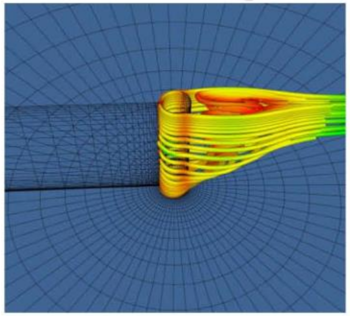
# Games

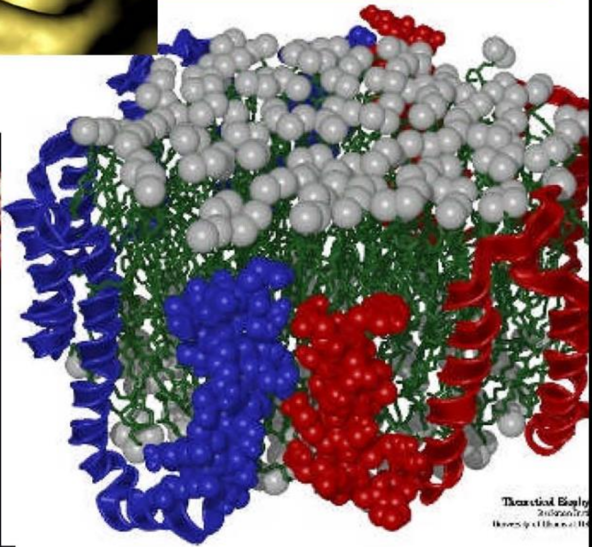
# Visualization

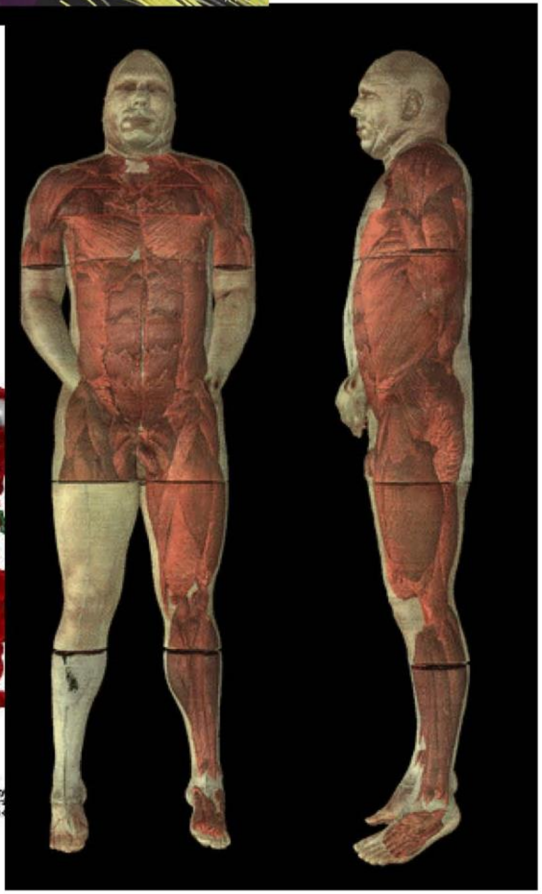


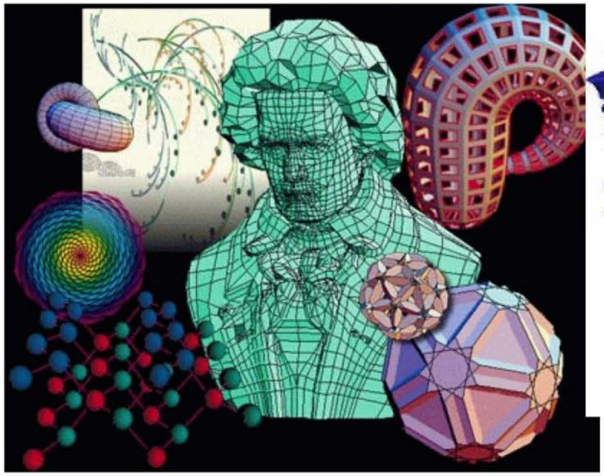





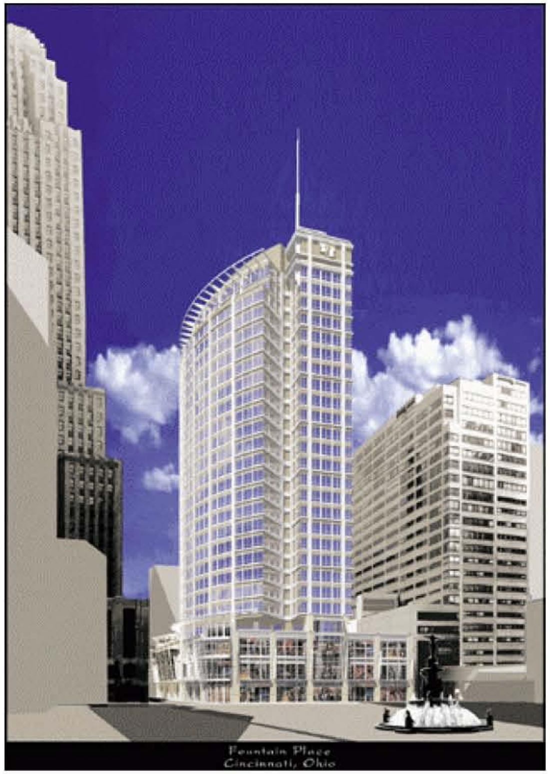




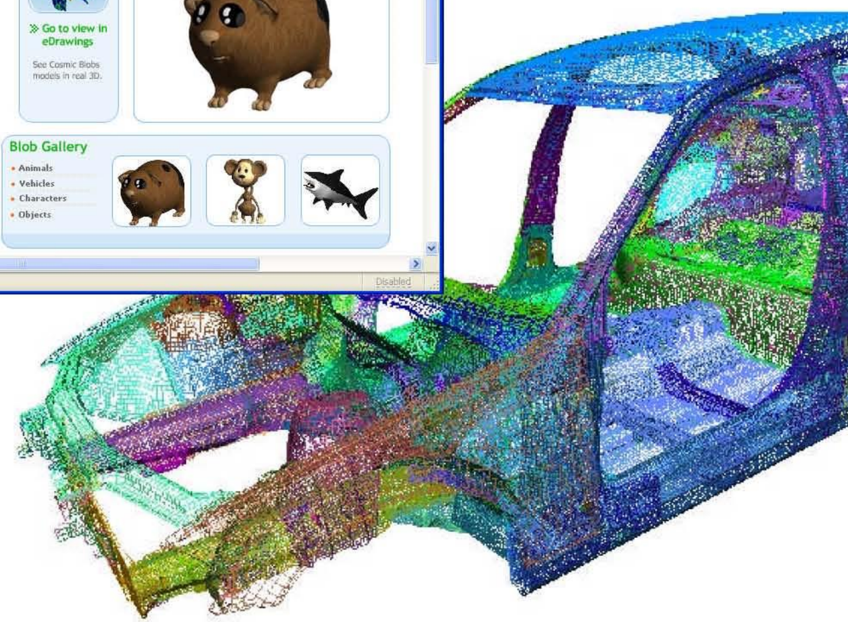




# Computer Aided Design (CAD)







# Virtual and Augmented Reality

# Mobile Media



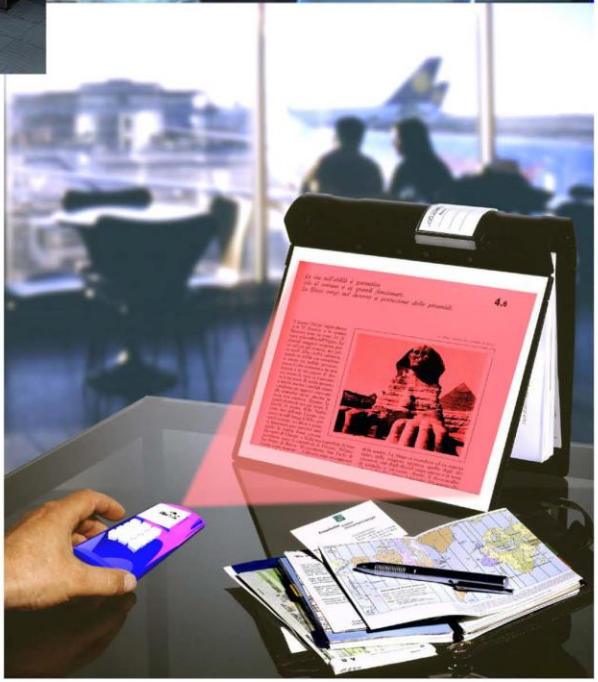

Sharp auto stereoscopic mobile phone

GPS phone



# New Information Spaces





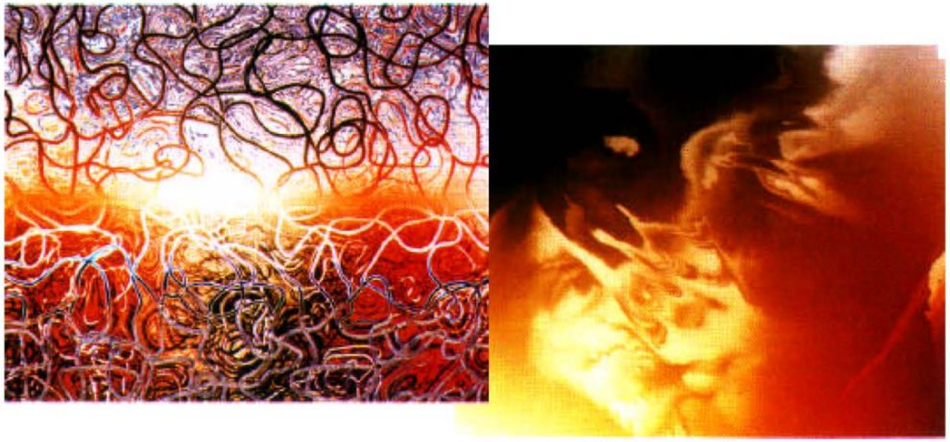
# Art

*"Contact Water"*

Taisuke Murakami, 2001

Steven Parente

http://www.alohablooms.com/atomica1.html



*Artificial Evolution for Computer Graphics*

Karl Sims, SIGGRAPH '91

# *For This Course*

- Programming heavy
- ***Not a course that teaches you artistic skills!***
- Expect nitty-gritty details instead
- CS130A is essential
- CS130B is helpful
- Math: review your matrix theory

# *Vectors and Matrices*

- Matrix and vector
- (L2-)norm of a vector
- Orthogonal vectors
- Norm of a matrix
- Matrix-vector multiplication
- Matrix-matrix multiplication
- Transpose of a matrix
- Inverse of a matrix

# *Orthogonal Matrices*

- Square matrix
- $AA^T=I$, $A^TA=I$
- $A^{-1}=A^T$
- Has orthogonal rows and columns
- Does not change the norm of a vector
- Represent a rotation (determinant=1), a reflection (determinant = -1), or a combination