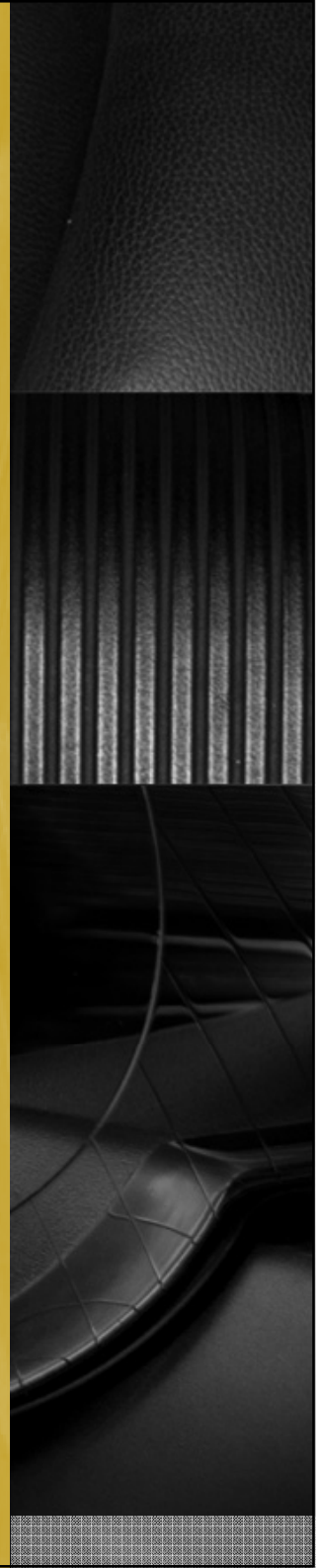
Introduction

Computer Graphics



¿What is an image?

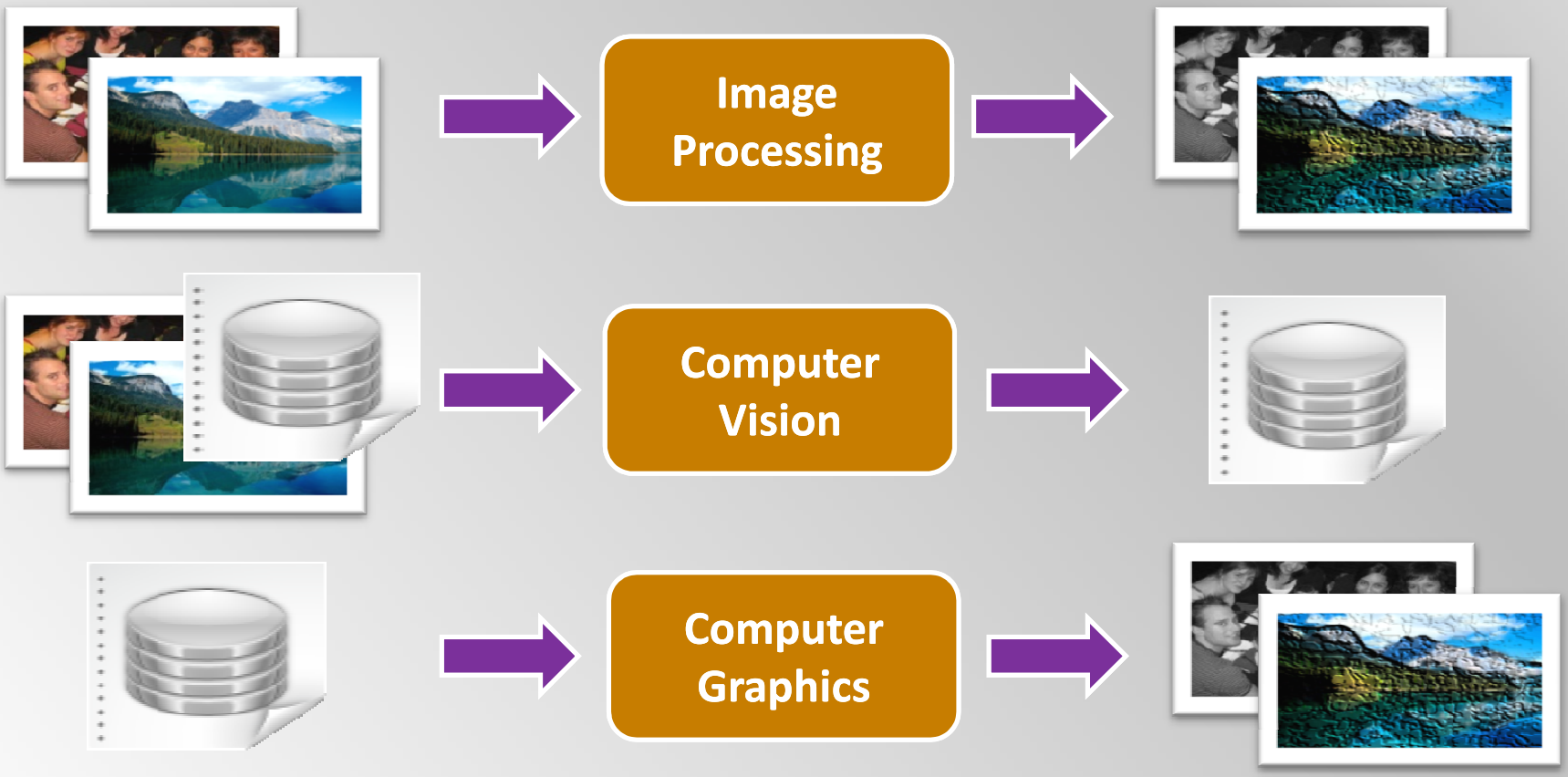
- A view on a monitor
- A file in a camera
- Numbers in a RAM (bit sequence)

- **Definition**
  - A 2D intensity or color distribution
  - A function defined on a 2-dimensional space
  - (no mention of pixels)

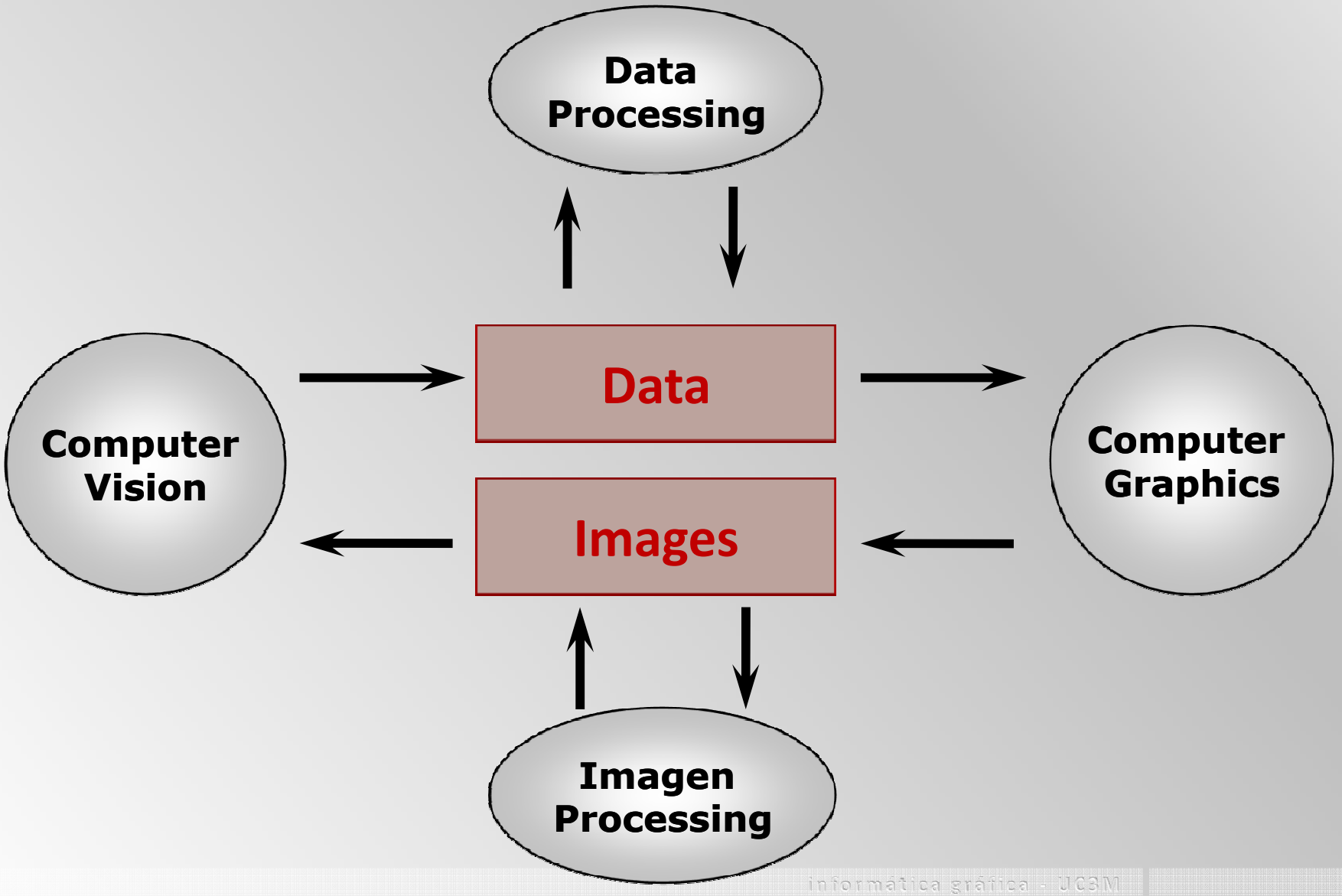
- **We need to:**
  - Represent an image (encode into numbers)
  - Show the image (apply transformations, electric currents, ink levels...)

Categorization

- Differences between **image processing**, **computer vision** and **computer graphics**

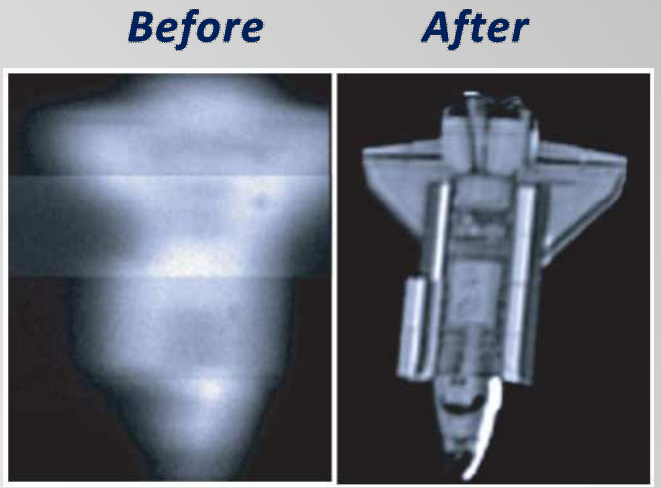


Categorization



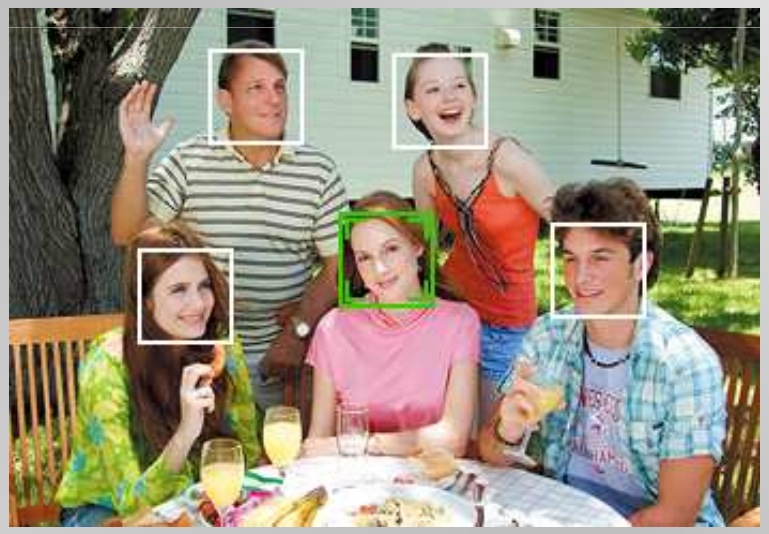
Categorization – Image Processing

- Techniques applied to a scene looking to **improve its quality** (private sector, scientific), **save resources** (image compression) or with **creative/artistic** purposes (posterization)
- We **reconstruct the image** using 2D or 3D models



Categorization – Computer Vision

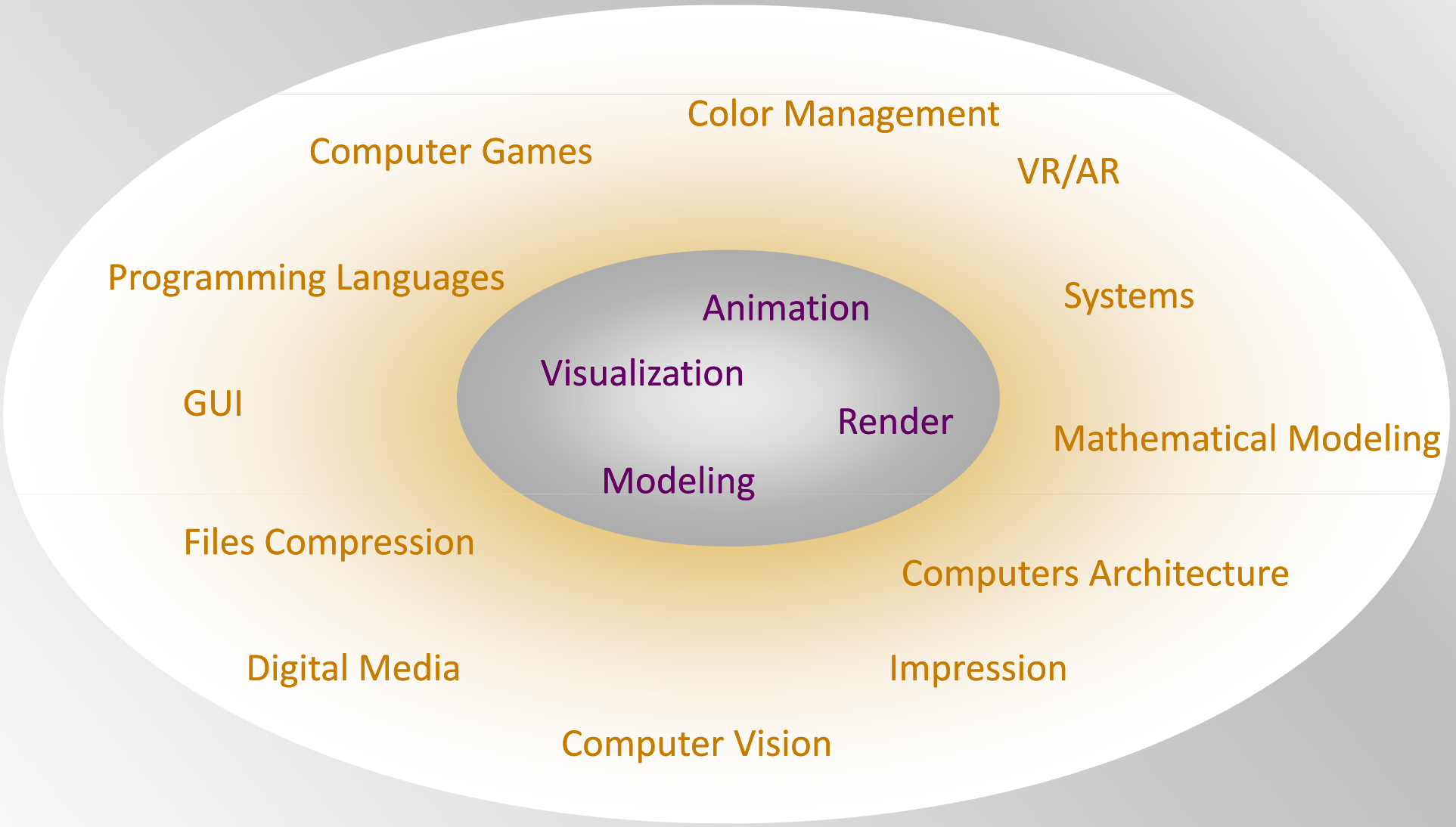
- Technology focused on the creation of **artificial systems** (machines) , capable extract information from an image and understand a scene or solve certain task, based on the image data.
- **Data classification** based on statistical information or priori knowledge
- **Examples**
  - Robots moving within a scene
  - Face recognition
  - Contents search



Categorization – Computer Graphics

- Representation and manipulation of real or imaginary objects by means of computer based models

Interdisciplinary subject



Interdisciplinary subject

- **Physics, Mathematics, Chemistry, Biology**
  - Simulation
  - Modeling
  - Numerical Analysis
- **Engineering**
  - Hardware and software systems
  - Input / Output device
  - Integration of existent environments
- **Arte, Physiology, Medicine**
  - Perception
  - Design and Composition

Applications – CAD

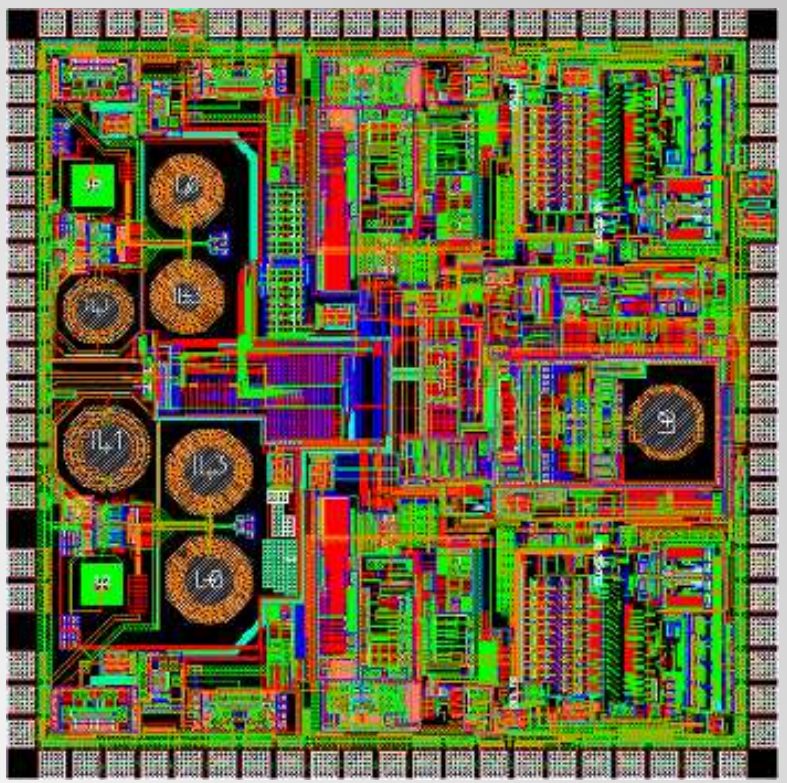
- Computer Aided Design
  - Graphic tools used for prototypes design and evaluation before building them
  - **Main Areas**: Industrial Design, Architecture, Electric Circuits, Infrastructures, Impressed and Integrated Circuits
  - **Usual Techniques**: primitives based design, curved surfaces, polygonal surfaces, etc.
  - **Other possibilities**: virtual reality, realistic presentations, design analysis, Computer-Aided-Manufacturing (**CAM**)
  - Computer Architectonic Aided Design (**CAAD**)

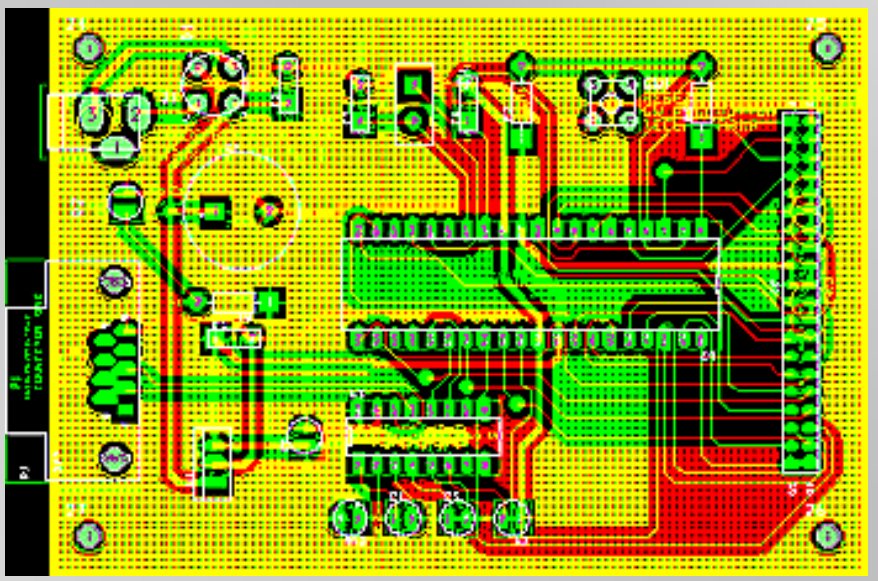
Applications – CAD

- Building

Applications – CAD

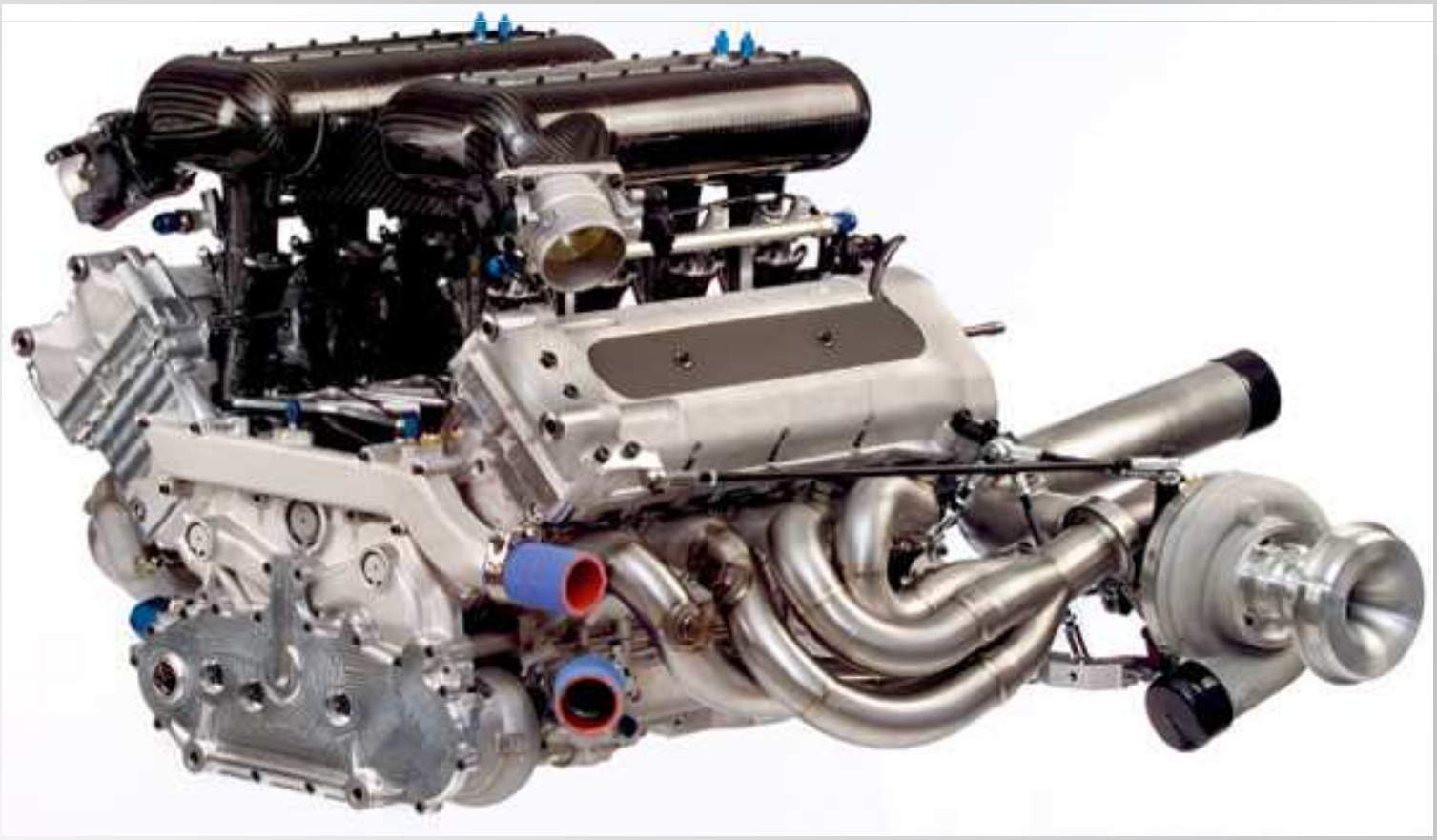
- Integrated and impressed circuits





Applications – CAD

- F1 engine



Applications – CAD

- Hong Kong bridge

Applications – CAD

- **Reference programs CAD**
  - **AutoCAD** (Autodesk, California 1982)
    - Industry leader, provides engineering, manufacturing, civil construction and architecture
    - OS: Microsoft, Windows, Mac
  - **SolidWorks** (SolidWorks, Massachusetts 1993)
    - Focused on modeling mechanical pieces through materials stress simulation and fluid dynamics
    - OS: Microsoft
  - **SketchUp** (Google, acquisition of @Last Software 2006)
    - 3D objects modeling
    - OS: Microsoft and Mac

Applications – Presentation Graphs

- **Information representation** – objectives:
  - Better understanding of a data set
  - Support to reports, projects, work presentations…
- **Main areas of application**
  - Economy, finances, Project Management, Statistics, Science
- **Main Techniques**
  - Line graphs, bars graphs, cake graphs, candlesticks (2D y 3D)
  - Surface graphics (3D)
  - Maps representations

Applications – Presentation Graphs

- Vast amount of tools nowadays
  - **Final Programs**

    Miner3D

    VisuMap

    OpenViz

    ScienceGL

    …
  - **SDKs/APIs**

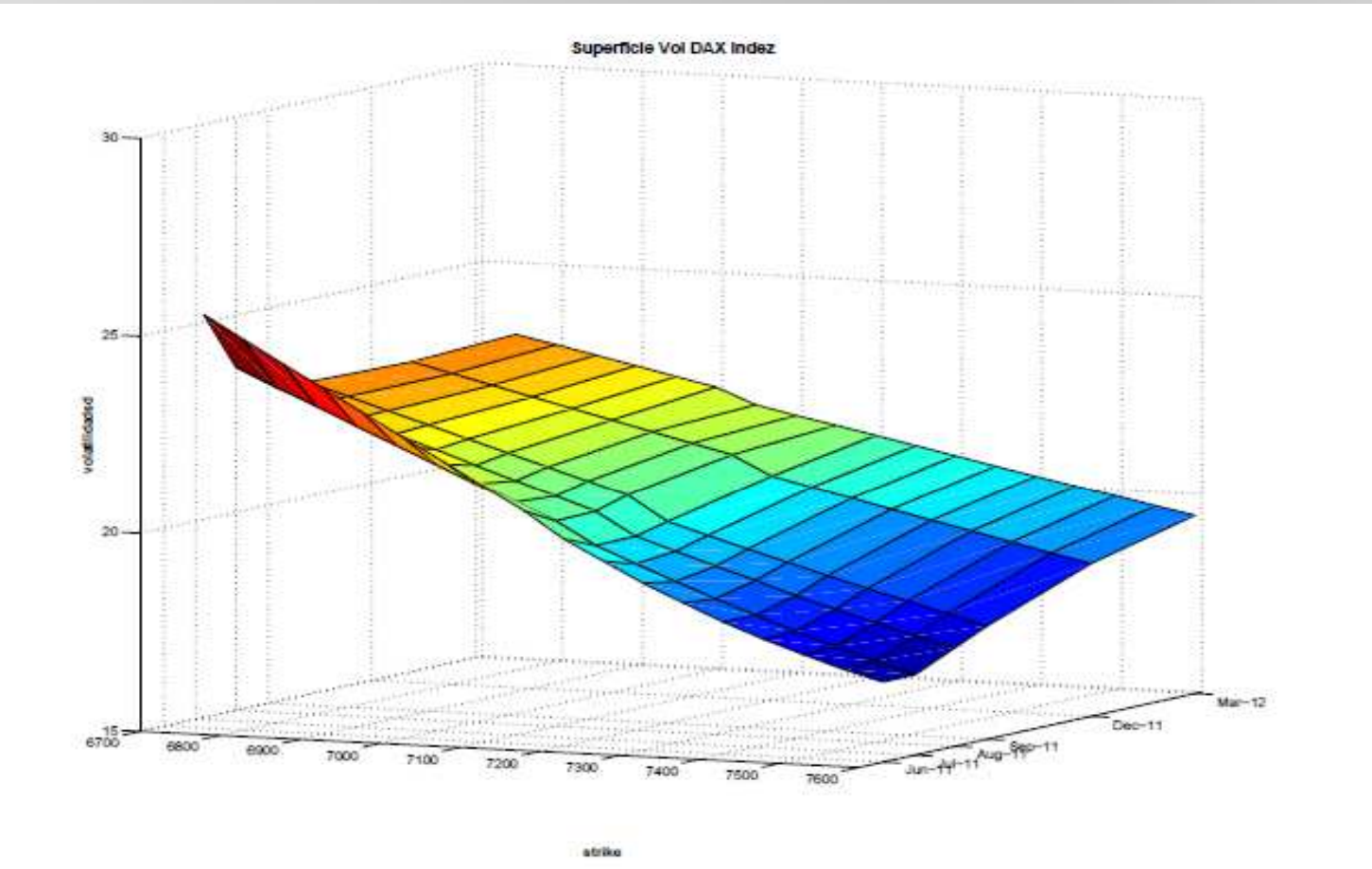
    Google Chart

    JuiceKit

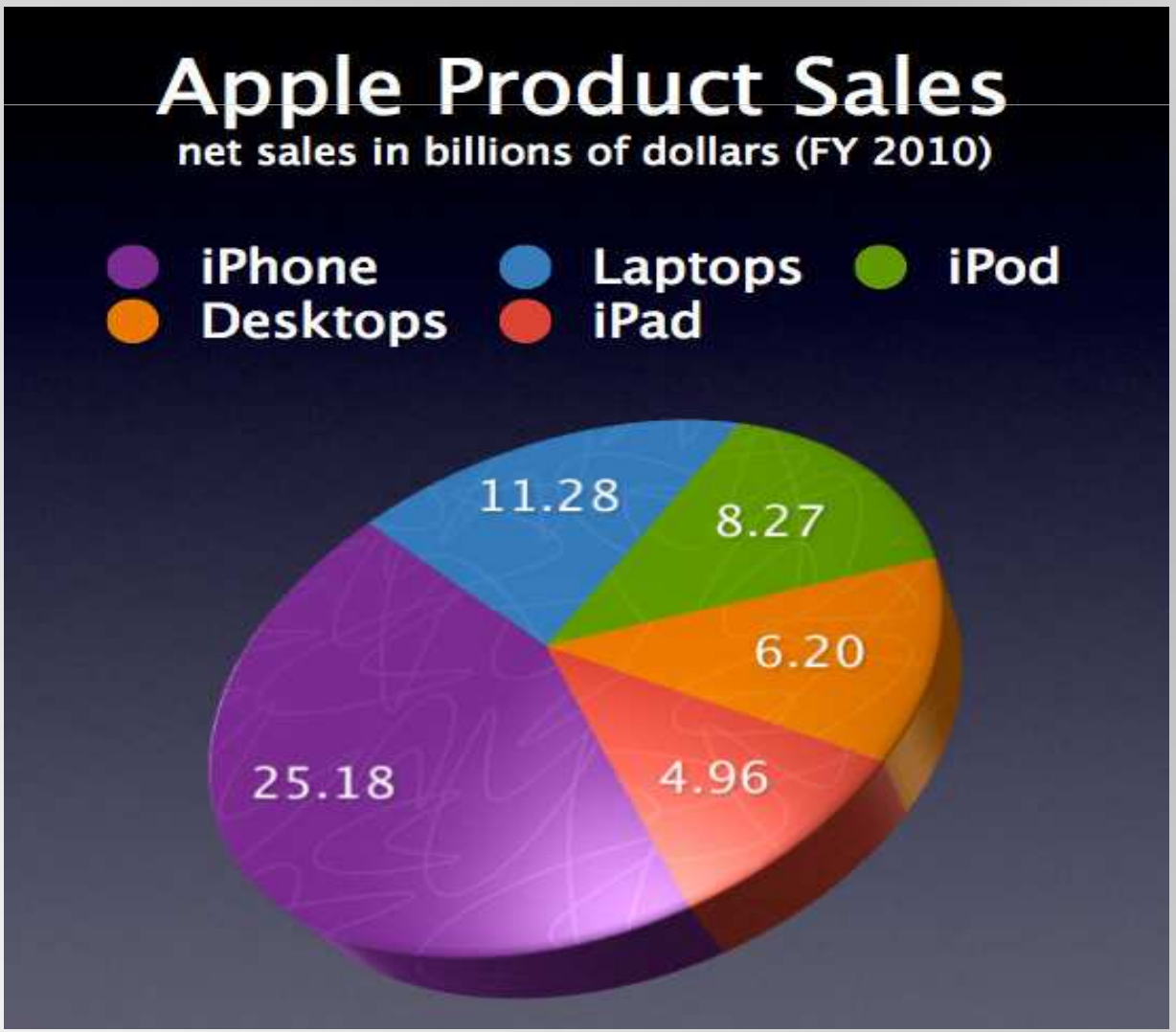
    …

Applications – Presentation Graphs

- Volatility Surface (DAX 30)

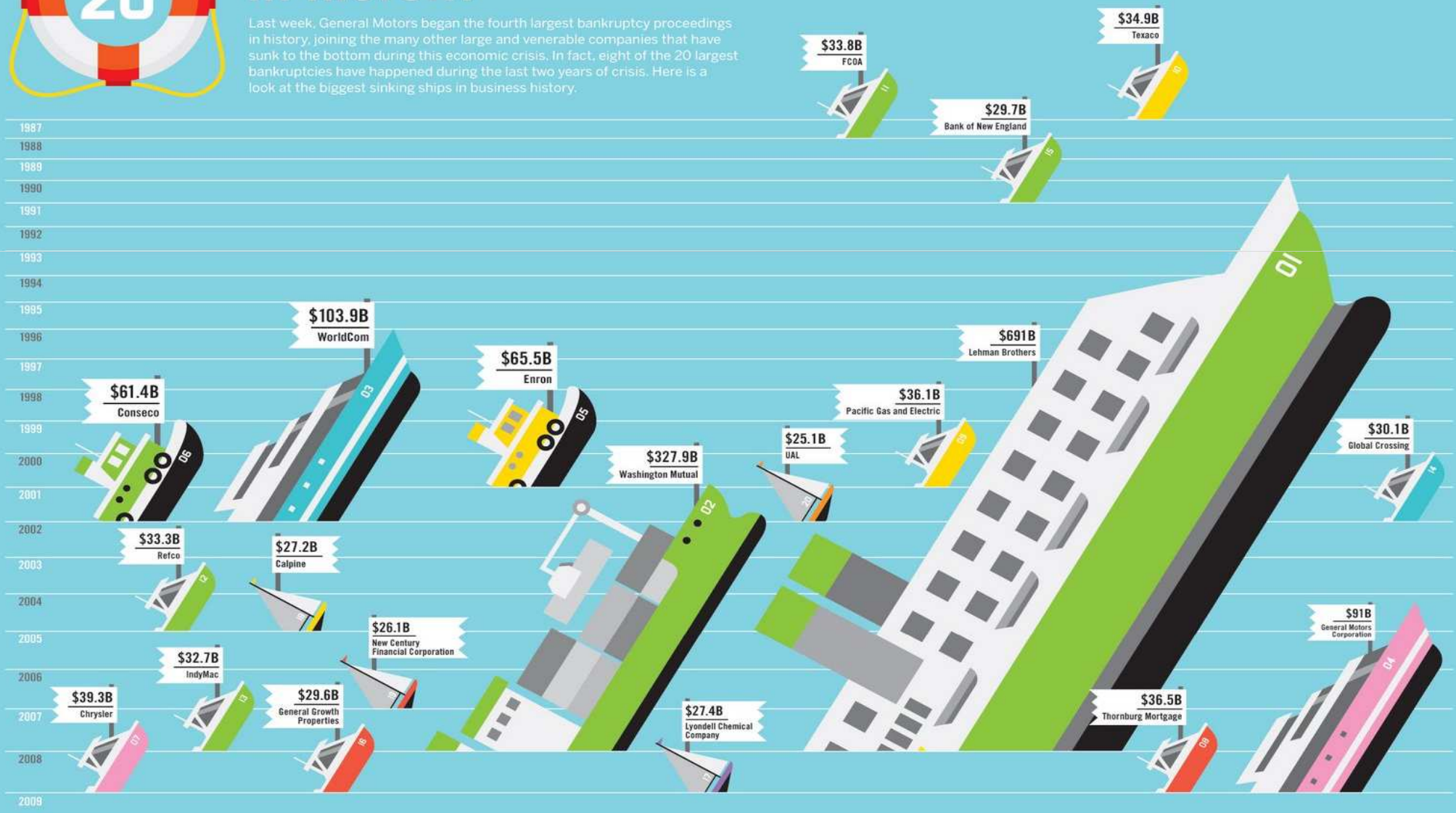


Applications – Presentation Graphs

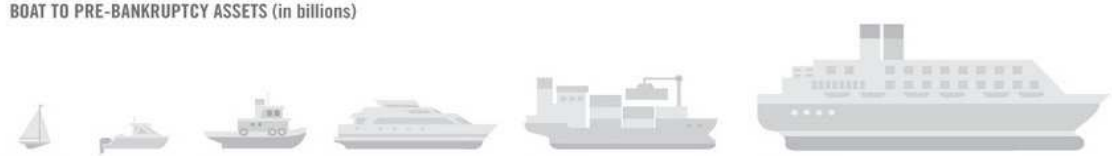


20 LARGEST BANKRUPTCIES IN HISTORY

Last week, General Motors began the fourth largest bankruptcy proceedings in history, joining the many other large and venerable companies that have sunk to the bottom during this economic crisis. In fact, eight of the 20 largest bankruptcies have happened during the last two years of crisis. Here is a look at the biggest sinking ships in business history.



BOAT TO PRE-BANKRUPTCY ASSETS (in billions)



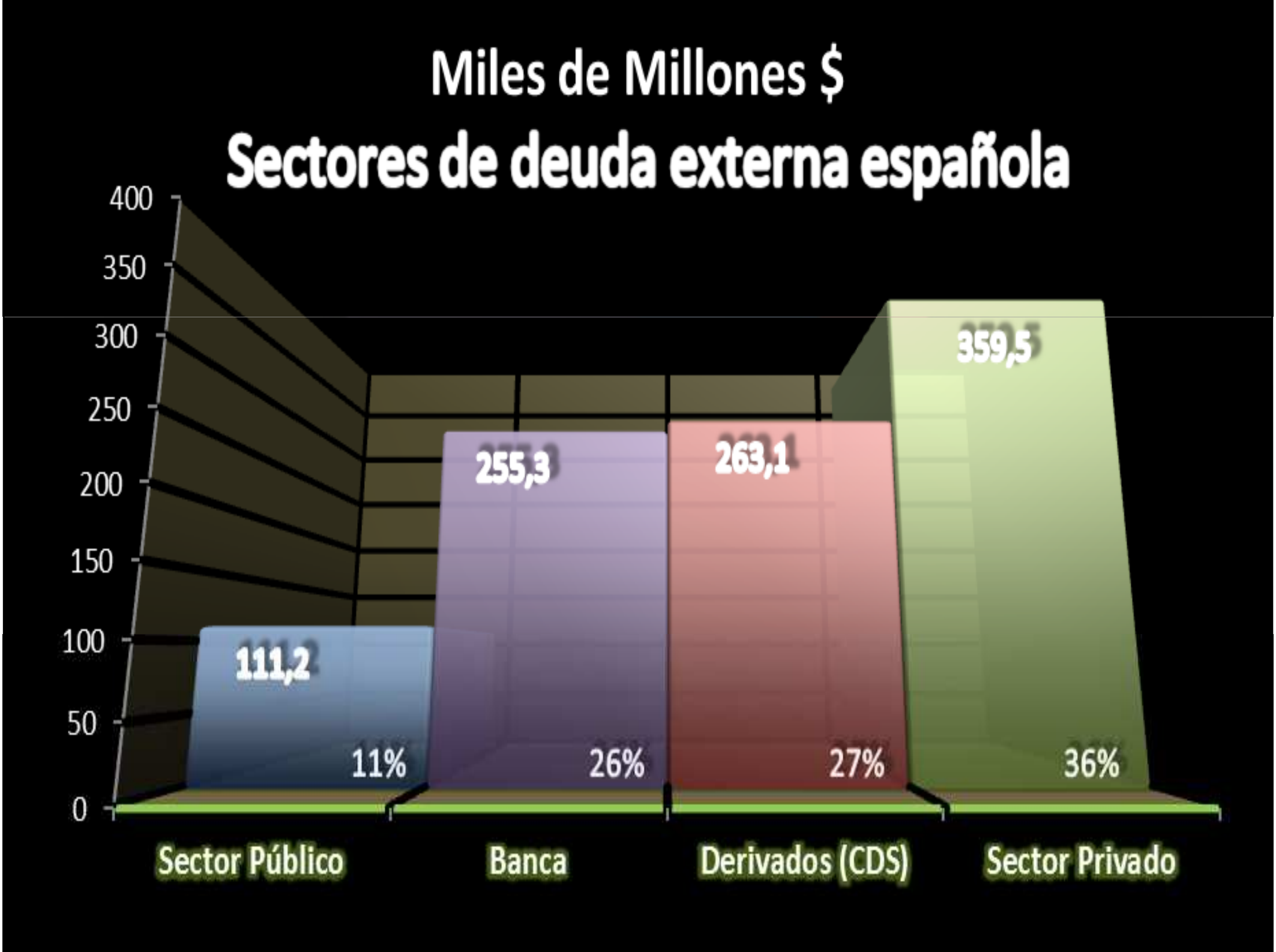
SECTOR

- Finance
- Energy
- Air Travel
- Real Estate
- Automotive
- Chemicals
- Telecommunications

PRE-BANKRUPTCY ASSETS (in billions) / Company

$XXX
Xxxxxxxxx

SOURCE BankruptcyData.com


Miles de Millones $
Sectores de deuda externa española
400
350
300
250
200
150
100
50
0
111,2
11%
255,3
26%
263,1
27%
359,5
36%
Sector Público
Banca
Derivados (CDS)
Sector Privado

Applications – Artistic Creations

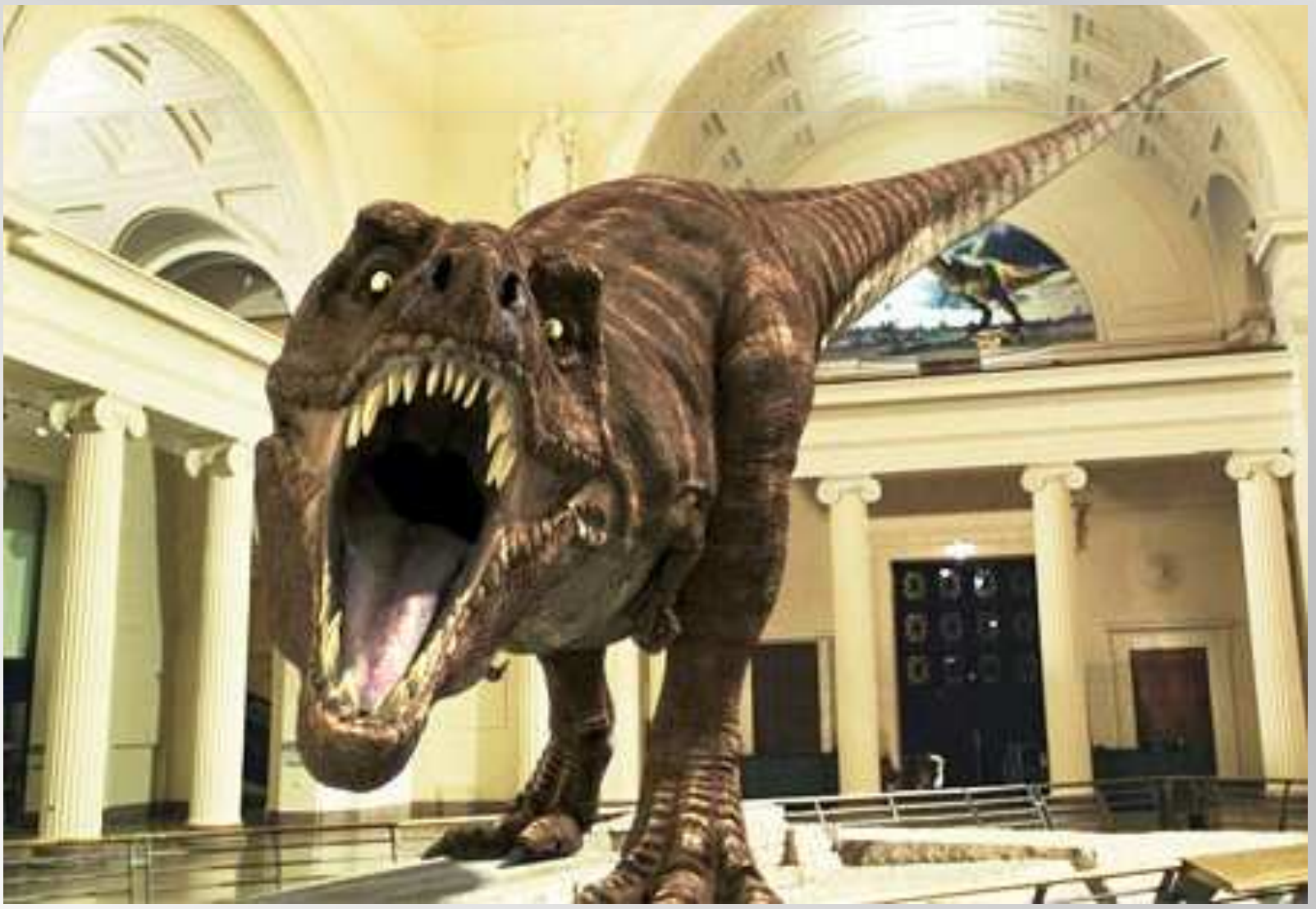
- **Artistic Creations**
  - Images created for either an artistic or commercial purposes**:** Logos design, Fine Arts, Commercial Animation
  - Techniques and software: programs like "**paintbrush**", Animation support programs, image treatment techniques, "**rendering**" techniques.

- **Software**
  - 3ds Max (Autodesk)
  - LightWave 3D
  - Maya
  - Rhinoceros
  - Shade 12

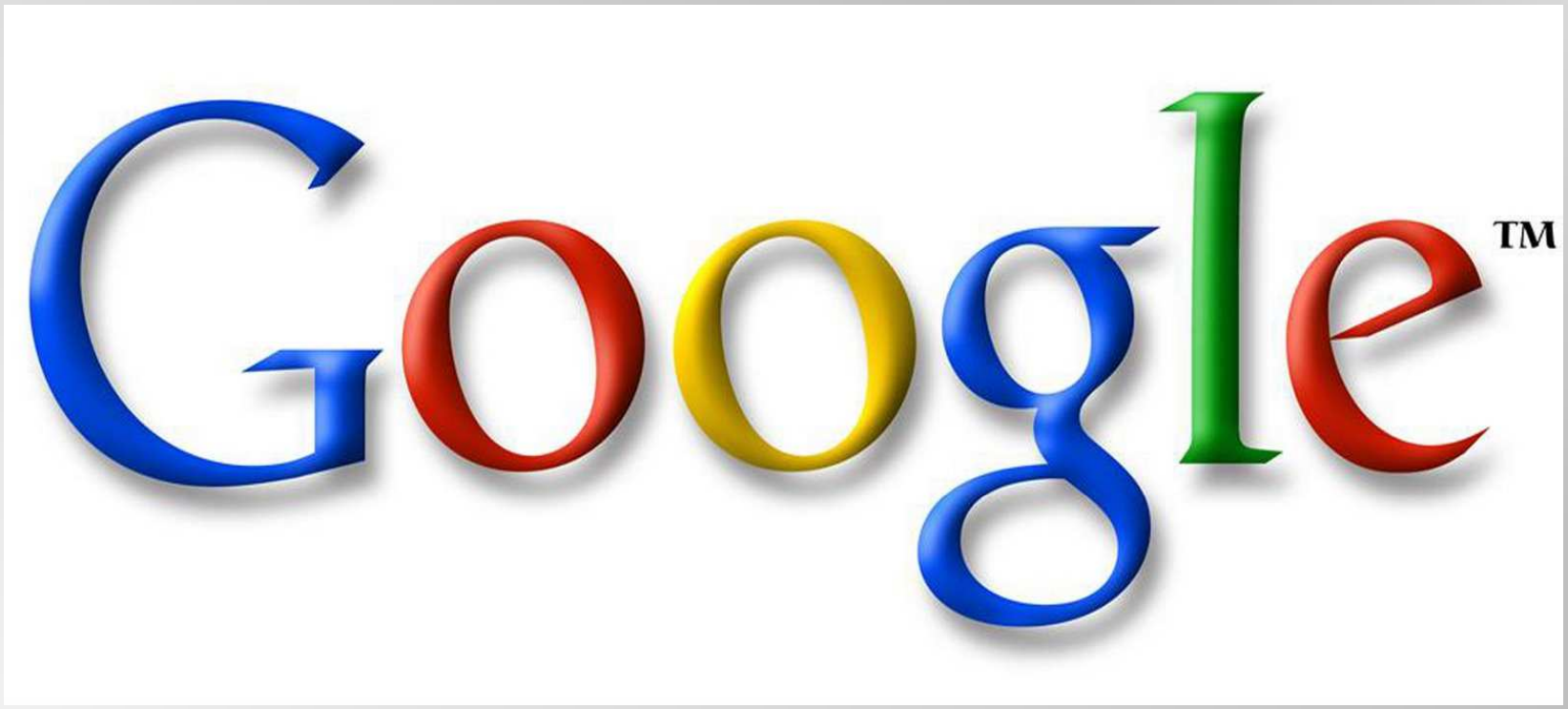
Applications – Artistic Creations

Applications – Artistic Creations

Applications – Artistic Creations



Applications – Artistic Creations



Applications – Entertainment

- **Entertainment**
  - **Areas –** Movies (Toy Story, Avatar…)Television (headers), Computer Games (Duke Nuken)
  - **Techniques** – Animation, Realistic Visualization, Special effects, Interactivity
- **Software**
  - API & SDK are used
    - Truevision 3D
    - XNA
    - OGRE (open source)
    - IrrLicht
    - Unreal
    - CryEngine 3

Applications – Entertainment

Applications – Entertainment

Applications – Entertainment

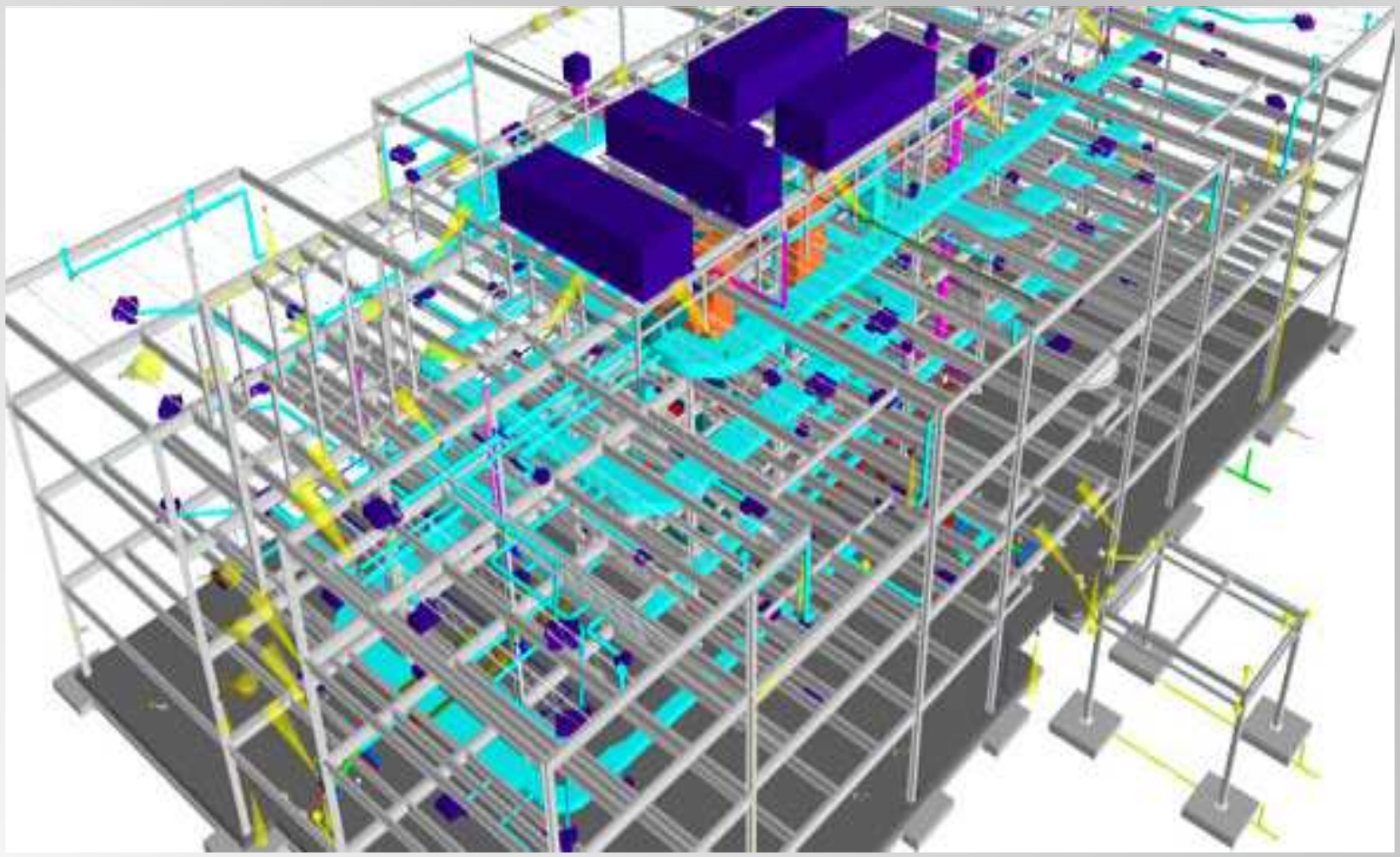
Applications – Entertainment

Applications– Simulation & Training

- **Simulation and Training**
  - **Areas**: driving simulation, flying simulation, industrial processes simulation, training (team operations, production line, medicine).
  - **Techniques**: real time, interactivity
  - **Specific Equipment:** F1 simulator
  - **New Techniques**: virtual reality
- **Software**
  - **Ad-hoc** performed systems
  - **Devices are specific** to the field of simulation (including hardware)

Applications– Simulation & Training

- Production Line



Applications– Simulation & Training

- Flying Simulator

Applications– Simulation & Training

- F1 Simulator (F1 2010)

Applications – Scientific Visualization

- **Visualización gráfica de gran cantidad de datos**
  - **Áreas**: Medicina (Ej. Resonancias), Ingeniería (Ej. Esfuerzos en mecanismos), Física (Ej. Campos), Química (Ej. Interacción molecular), Matemáticas (Ej. Solución a ecuaciones), Topografía y oceanografía (Ej. Terrenos y corrientes)
  - **Técnicas**: Codificación por color, Curvas de nivel, Visualización de volúmenes
- **Software**
  - Específico, programas asociados a elementos físicos, escáner, microscopio, telescopio, TAC….
  - **Librerías**

    MathGL

    SGT

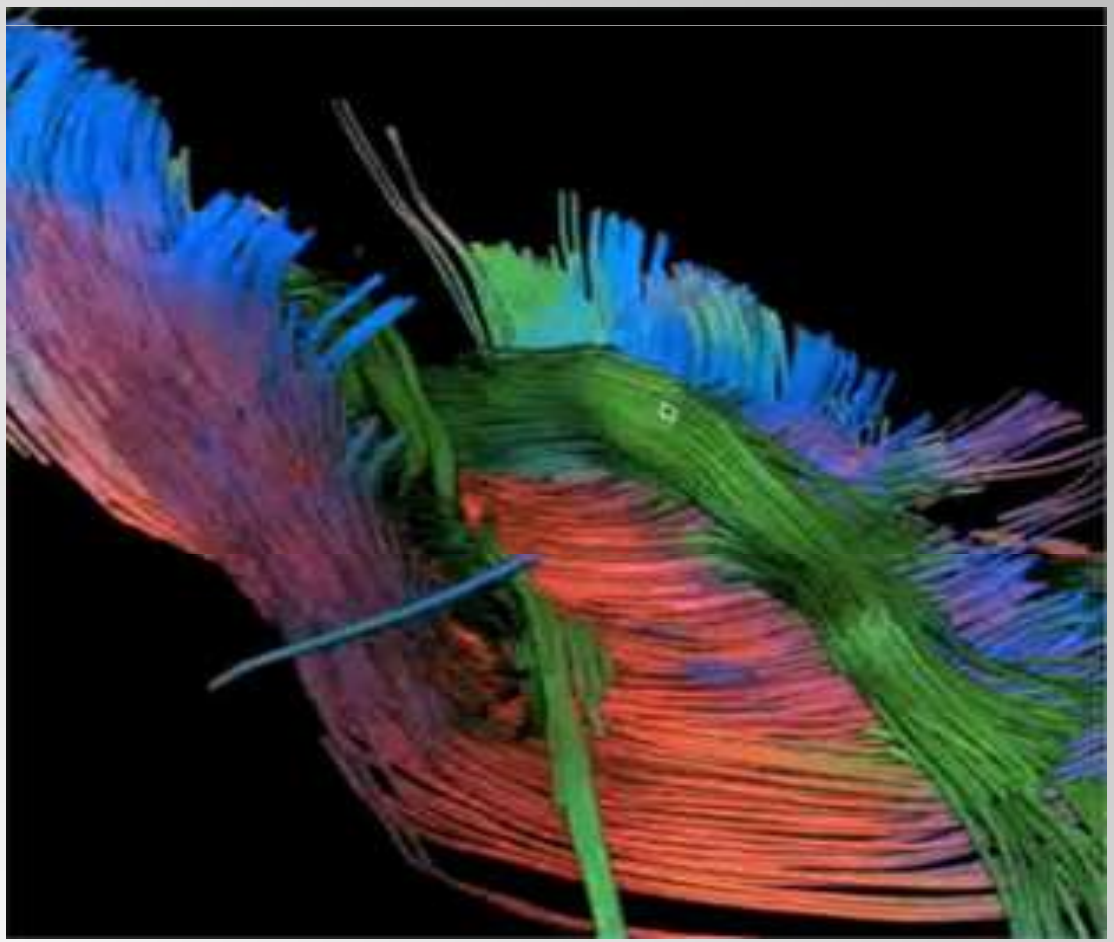
    Matlab

    SigmaPlot

    …

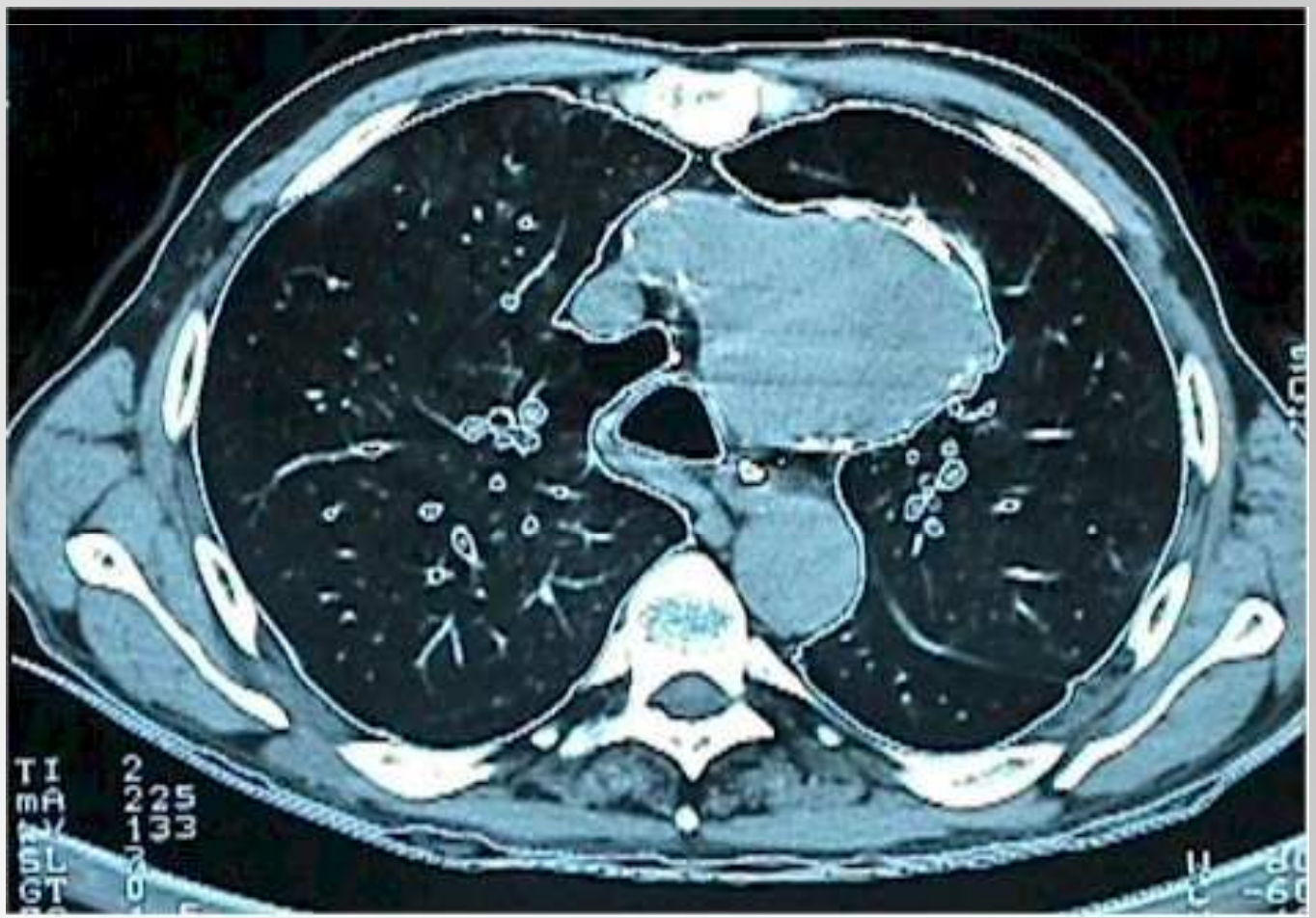
Applications – Scientific Visualization

- Brain MRI transformed into a 3D conections mapping



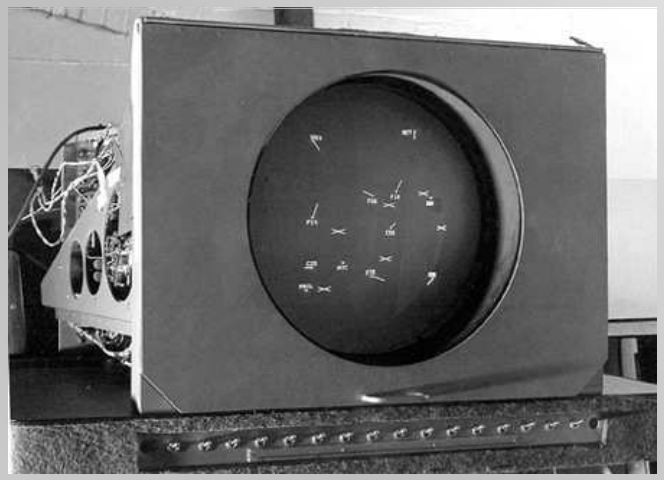
Applications – Scientific Visualization

- Thoracic TAC



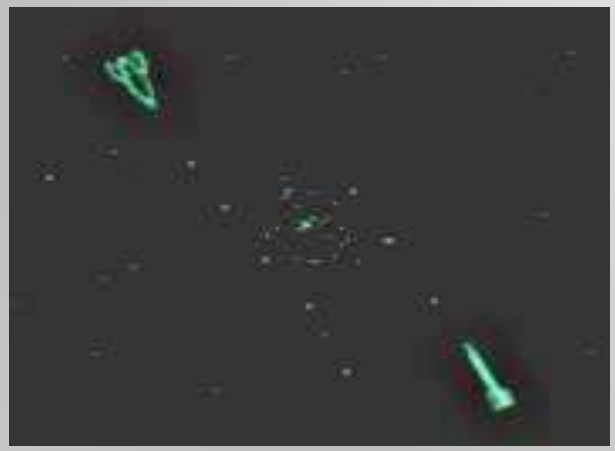
History

- 1950s
  - **SAGE** automated control system, developed for NORAD as a bomb tracker
  - Developed by MIT (1951) using **Whirlwind,** first real time computer (also developed by MIT, 1940s)
  - William Higginbotham creates the first game "**Tennis for two**"…in a Oscilloscope! (958)



History

- **1960´s**
  - Steve Russell, MIT student, develops (PDP-1 hardware) the first videogame "**Spacewar**" (1961)



  - Ivan E.Sutherland creates the first interactive graphic program **Sketchpad** (PhD, MIT1963) "*Sketchpad: A Man-machine Graphical Communications System*"

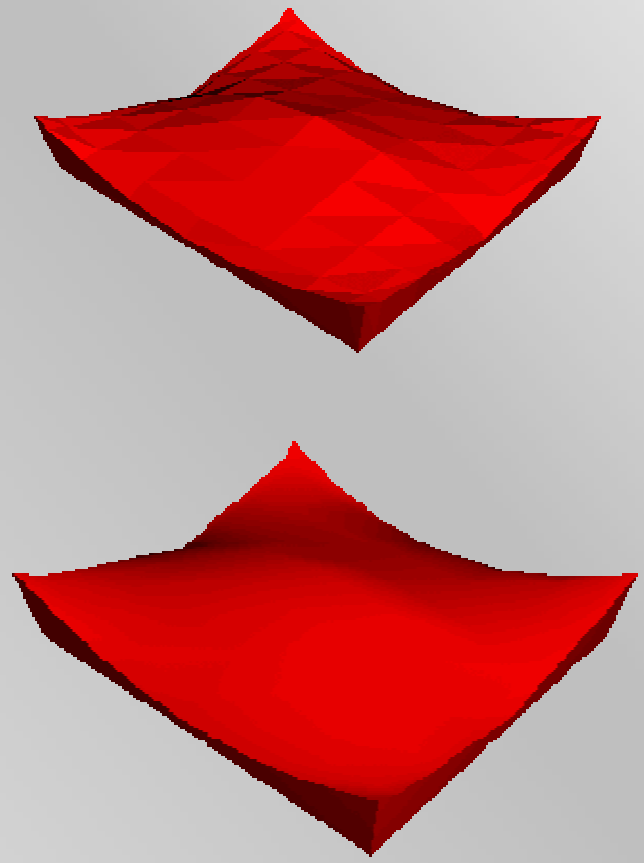
Sutherland with the TX-2 (320Kb) console

History

- **1960´s**
  - **First short animation movies** for physical effects simulation (gravity, movement, etc.) (1963)
  - Sutherland (MIT 1966) creates the first **helmet with stereoscopic visualization**
  - Catmull and others create the first **hidden surfaces algorithm** (Universidad de Utah, finales de los 60)
  - Same team starts becoming interested in **realism** through colored surfaces shading

History

- **1970´s**
  - **Gouraud algorithm** for polygonal surfaces continuous shading (1971)
  - **Microprocessor** Commercialization (1971)
  - **Atari** is founded (1972) and drive videogames rise
  - First attempts of using **computer graphics for films** editing



History

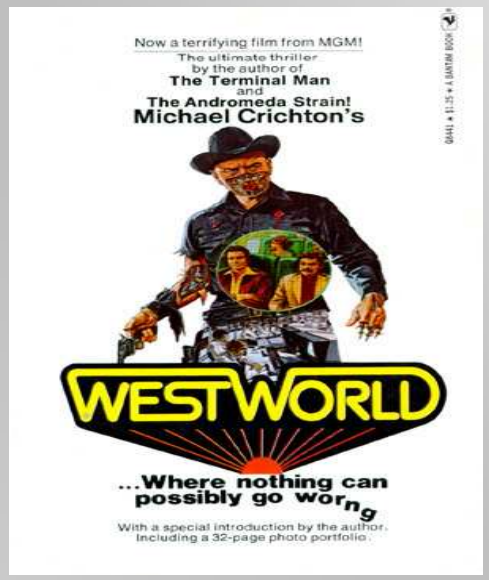
- **1970s**
  - Newell creates the famous "**Utah teapot**", which still serves as a test bank (Utah University, 1975)
  - Edwin Catmull creates the **Z-Buffer** technique (1974), used to eliminate hidden surfaces
  - Edwin Catmull introduces textures

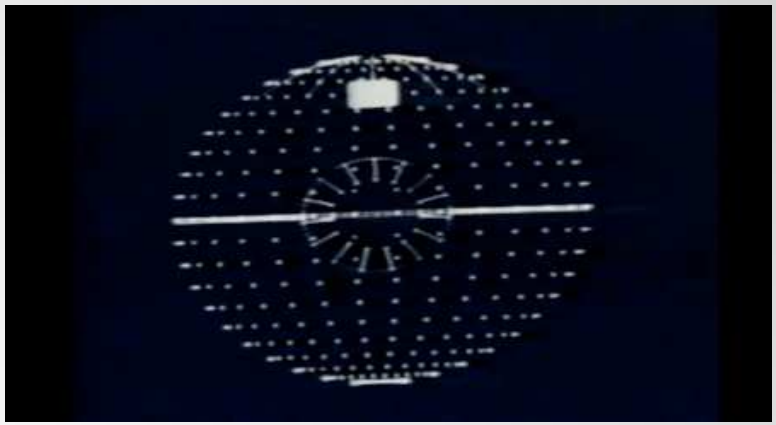
History

- **1970s**
  - Phong develops his **polygonal surfaces shading** (1974)
  - Jobs, Wozniak y R.Wayne found **Apple** (1976)
  - Gates founds **Microsoft** (1975)
  - **Lucasfilm** creates its computer graphics division, including some of the best developers wordlwide (1979)

History

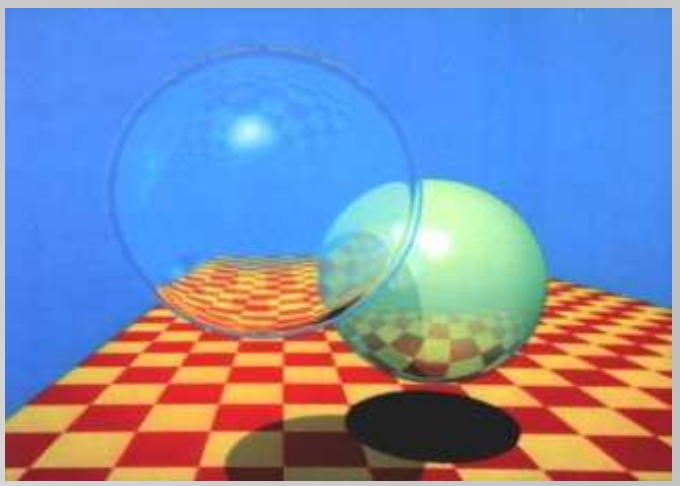
- **1970s**
  - **Westworld** (1973) was the first film using computer generated graphics
  - The "**computing virus**" concept is born
  - **Star Wars** (1977)





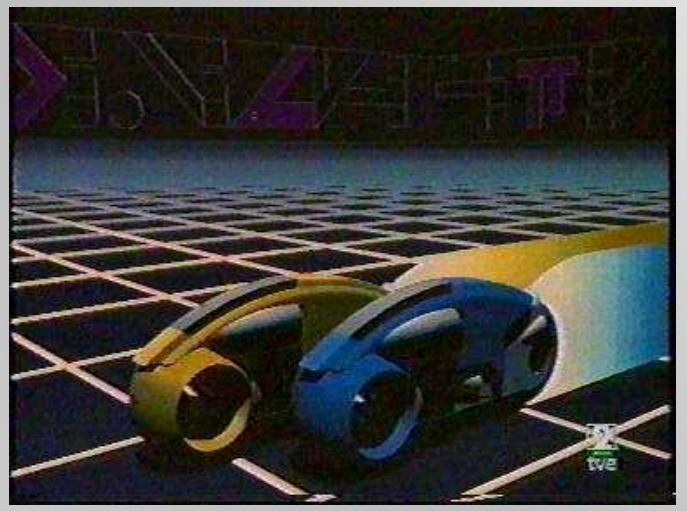
History

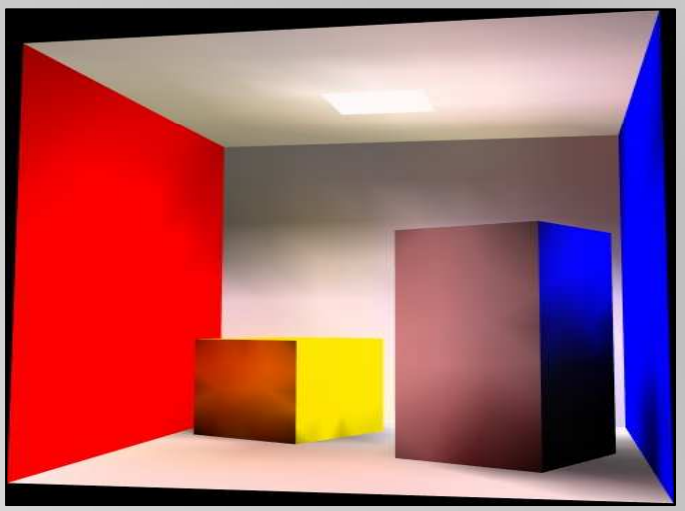
- **1980s**
  - **SIGGRAPH** becomes the annual most important event in the industry
  - Whitted publishes an article on raytracing technique '**Spheres**' Bell Laboratories, Communications ACM
  - Carpenter creates **REYES** in Lucasfilm, the first rendering software (1981).
  - "**Rendering**" equation is published (Kajiya , 1986).



History

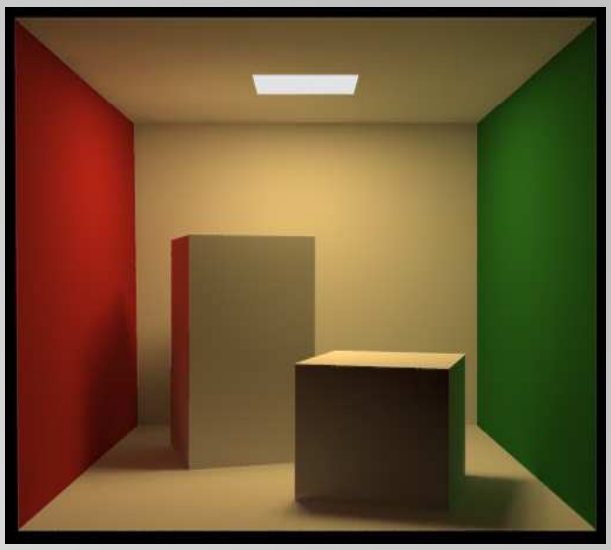
- **Años 80**
  - Disney creates the film **TRON** (Lisberger y Kushner 1982)
  - Massive sale of graphics terminals: IBM, Tektronix
  - **First ISO and ANSI** standard as a graphic library construction rule: GKS
  - IBM built the **first PC**
  - Goral, Torrance et al. (1984), Cohen (1985). **Radiosity**





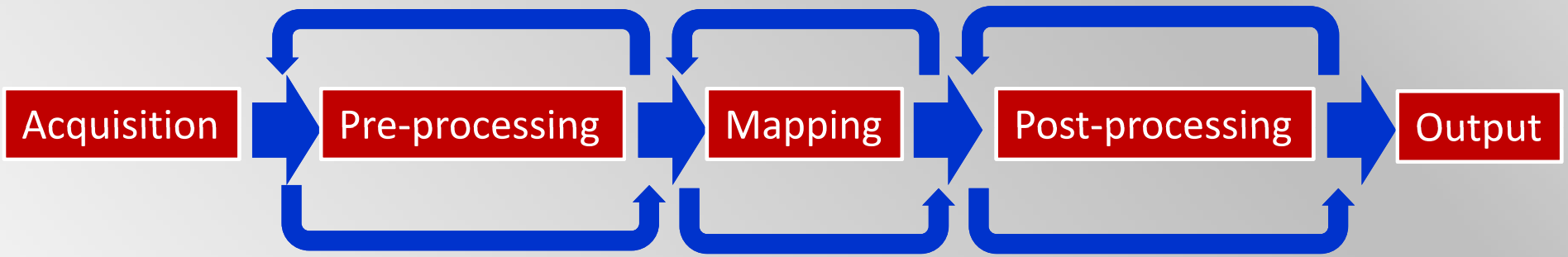
History

- **Años 80**
  - Goral, Torrance et al. present the “Cornell Box” at the 1984 SIGGRAPH
  - Created at Cornell University, it was used to test an rendering software precision



2D Image Processing

- The steps followed in 2D image processing is usually represented as a "**pipeline**"
  - Not all these steps are always performed when processing an image



2D Image Processing – Acquisition

- Many Techniques, divide into main categories:
  - **Synthesis**: computer created images
    - From object geometric descriptions (Renderman, Maya, PovRay…)
    - "Painted" images (Photoshop, CorelDraw, Fractal Painter…)
  - **Capture**: images from the real world
    - Captured o digitalized with specific hardware
      - Digital Cameras
      - Satellite taken data
      - Laser scanner
      - Kinect…
  - A hybrid category could be considered: synthetic objects with real textures (and vice versa)

2D Image Processing – Pre-processing

- Shape, size and color properties modifications
- Techniques
  - Color and levels adjustment
  - Cutting
  - Escalating
  - Blurring
  - Edges improving
  - Filtering
  - …

2D Image Processing – Mapping

- Several images combined with transformations
  - Translation
  - Rotations
  - Escalating
  - Distortions
  - Compositions
  - Transparency and Translucency effects

2D Image Processing – Post-processing

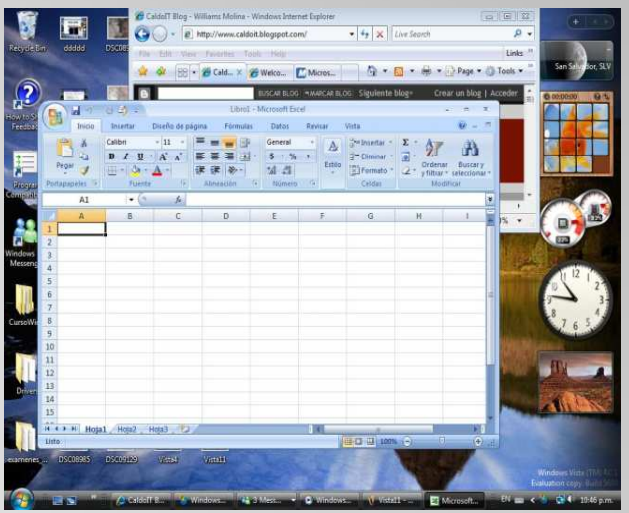
- Used to apply global effects to the image
- **Artistic Effects**
  - Posterization
  - Aging
  - Blurring
  - Texturing
- **Technical Effects**
  - Contrast improvement
  - Color variations

2D Image Processing – Output

- The type of output or storage device may affect the quality
  - A printer color map could intensify some colors over the others
  - Need a color mapping between screen and printer
- **Devices**
  - Screen
  - Printers
  - Disks
  - Textures Maps

3D Graphics Processing “3D Graphics Pipeline”

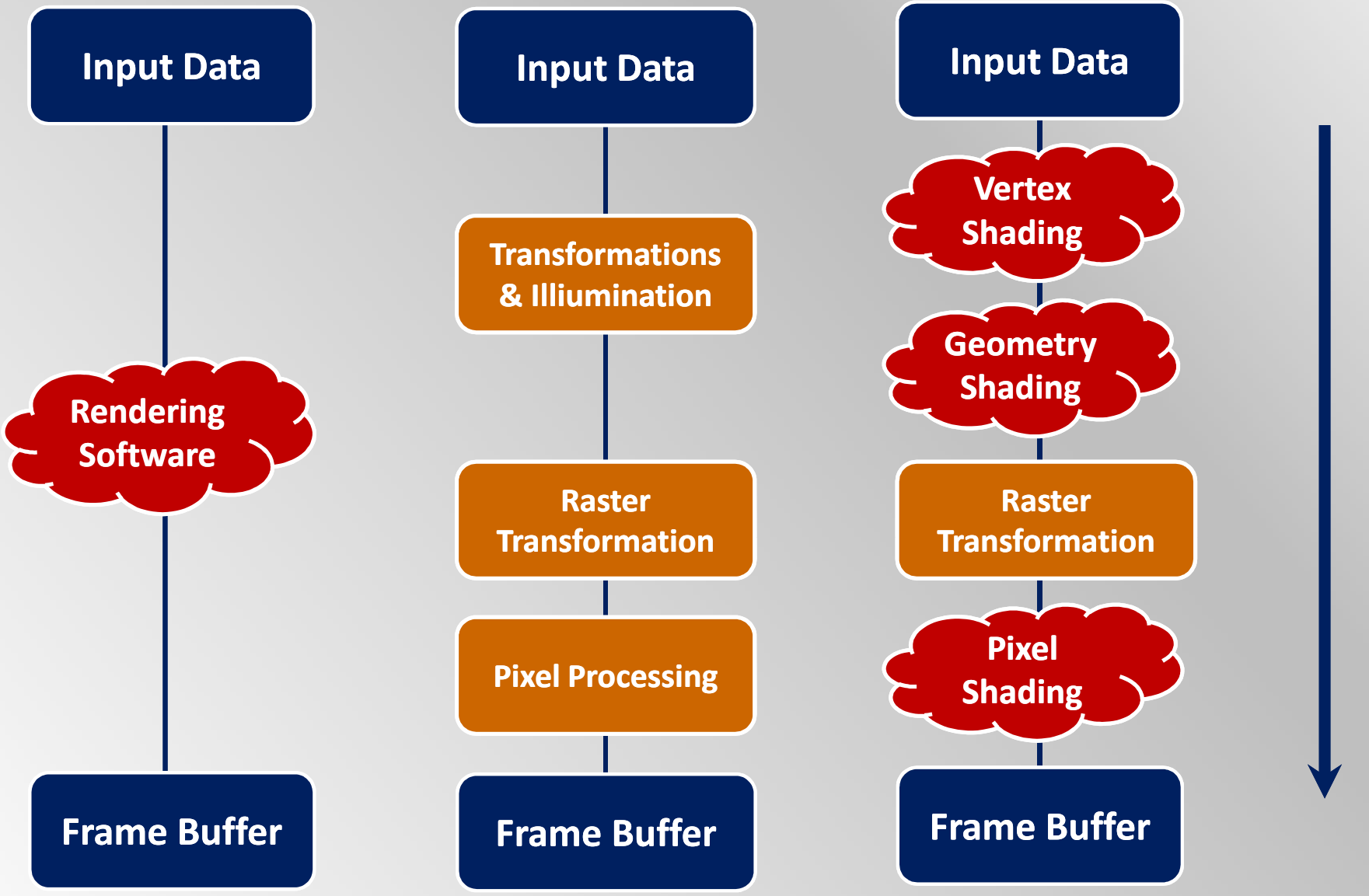
- Generate images that can be watched on a device
  - Generates the operating system view
  - Generates the console frames
  - Generates what we can see on our cell phones
  - Generates film images



3D Pipeline

- Several **modules** perform different tasks to generate the image render
  - Transformations
  - Illumination
  - Texturing
  - “Shaders”
- **Flow Diagram**
  - Input objects geometry and composition materials data
  - Illumination conditions definition
  - Images otuput

3D Pipeline – Evolution



Current Pipeline – shaders

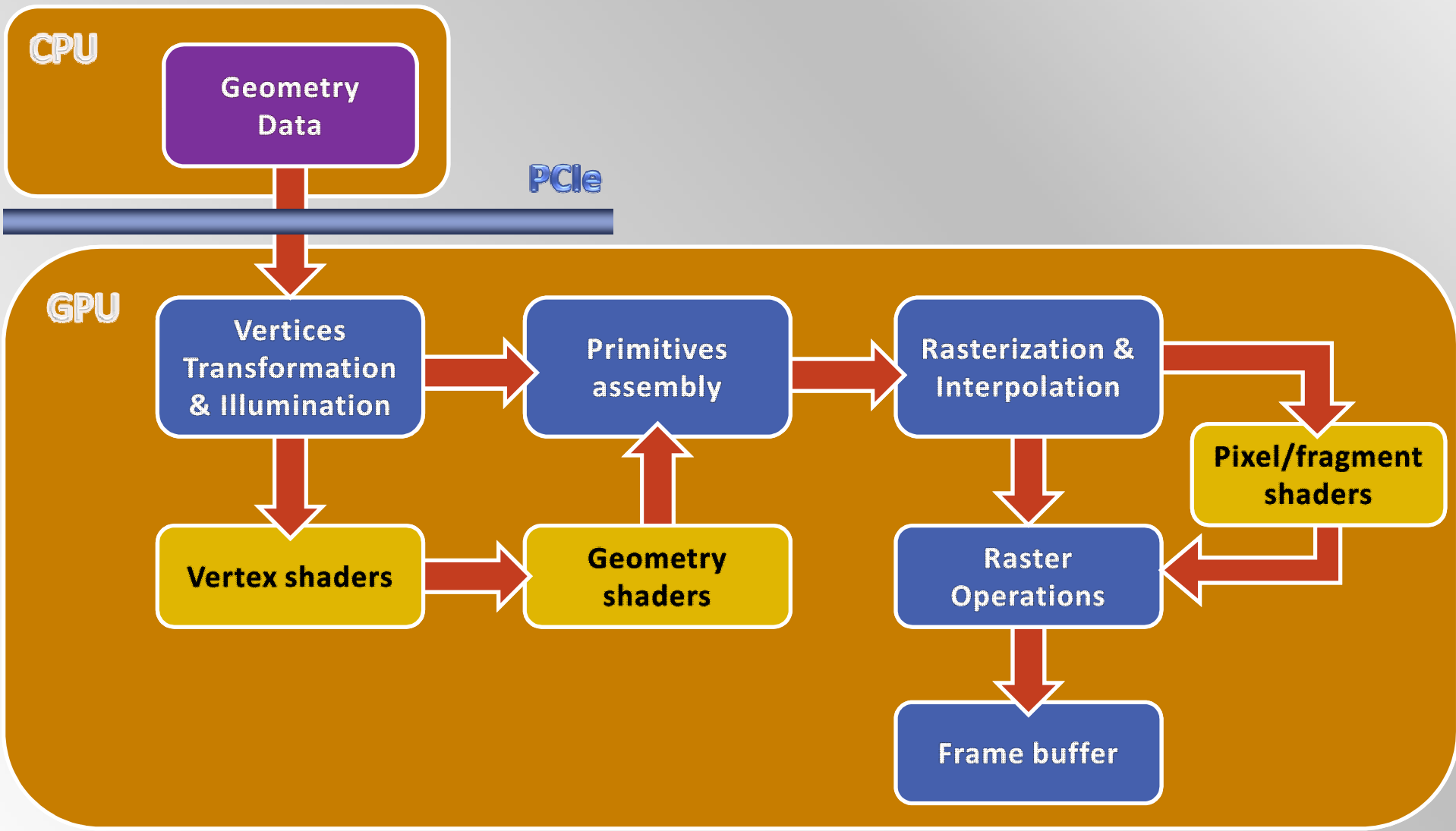
- **SHADER** process unit independently compiled, allowing local modification of objects properties within the scene
  - **Vertex**

    Transformation of coordinates, colors, textures and vertices normal vectors
  - **Geometric**

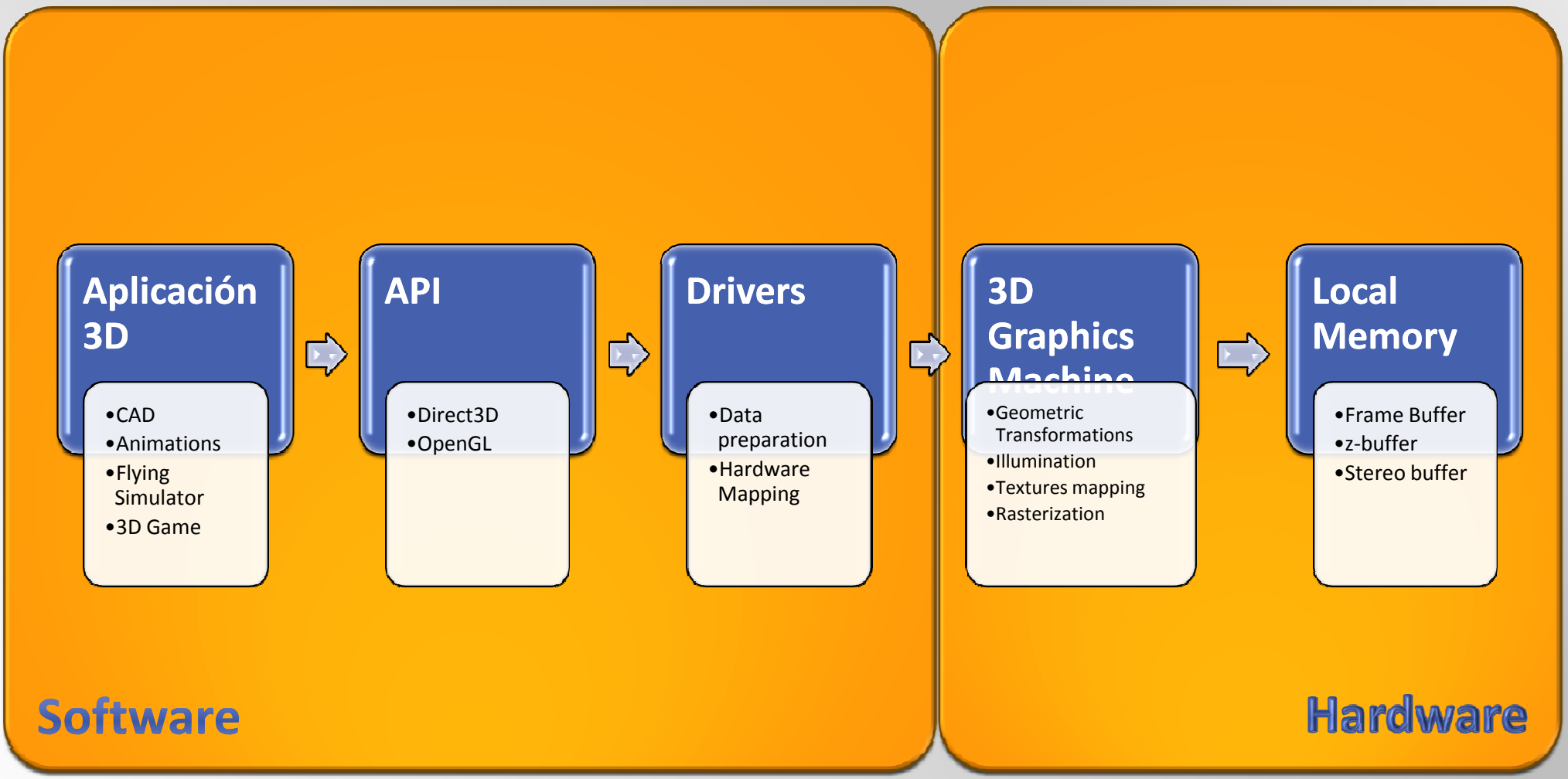
    Generate new forms (primitives) or modify existant ones (vary the number of element in a grid)
  - **Pixel / Fragment**

    Manipulate color at a “pixel” level, mainly on texturing and shading operations

Pipeline (vertices in –pixels out)

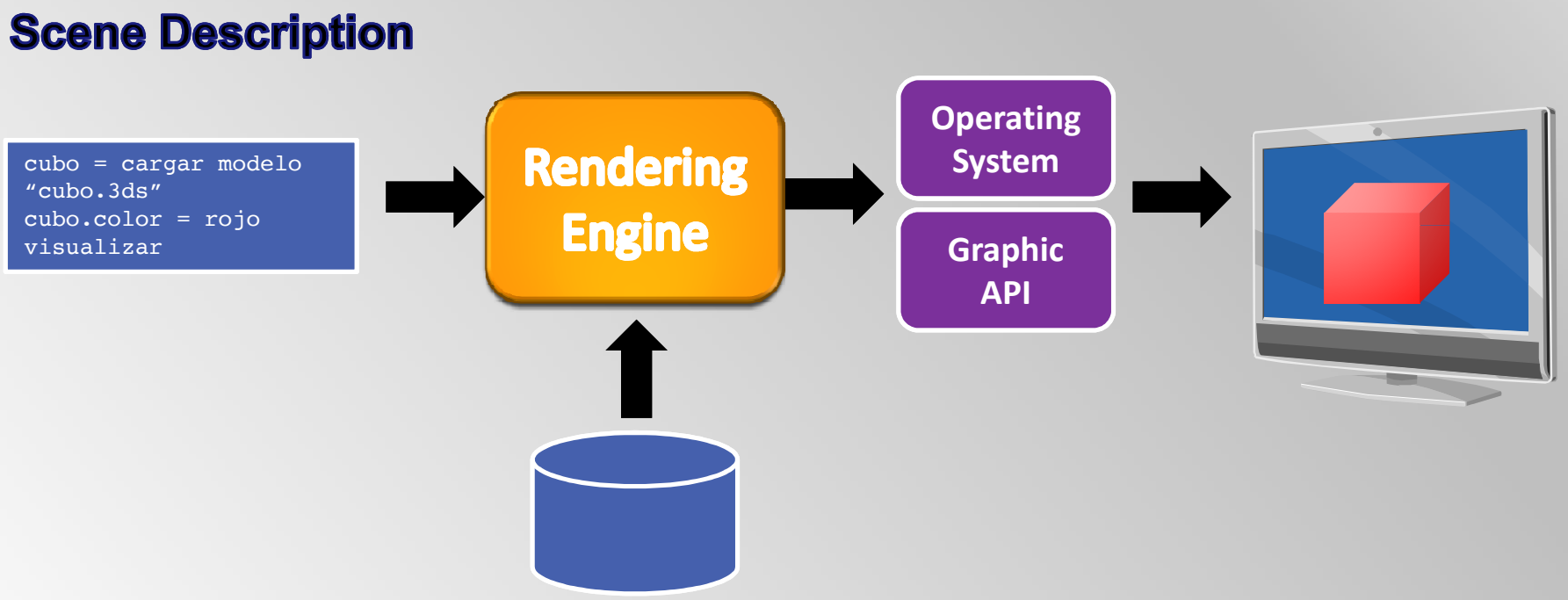


Application Flow



Rendering Engine

- Software responsible for generating and visualizing graphic information
- Abstracts the graphics creation mechanisms



Rendering Engine

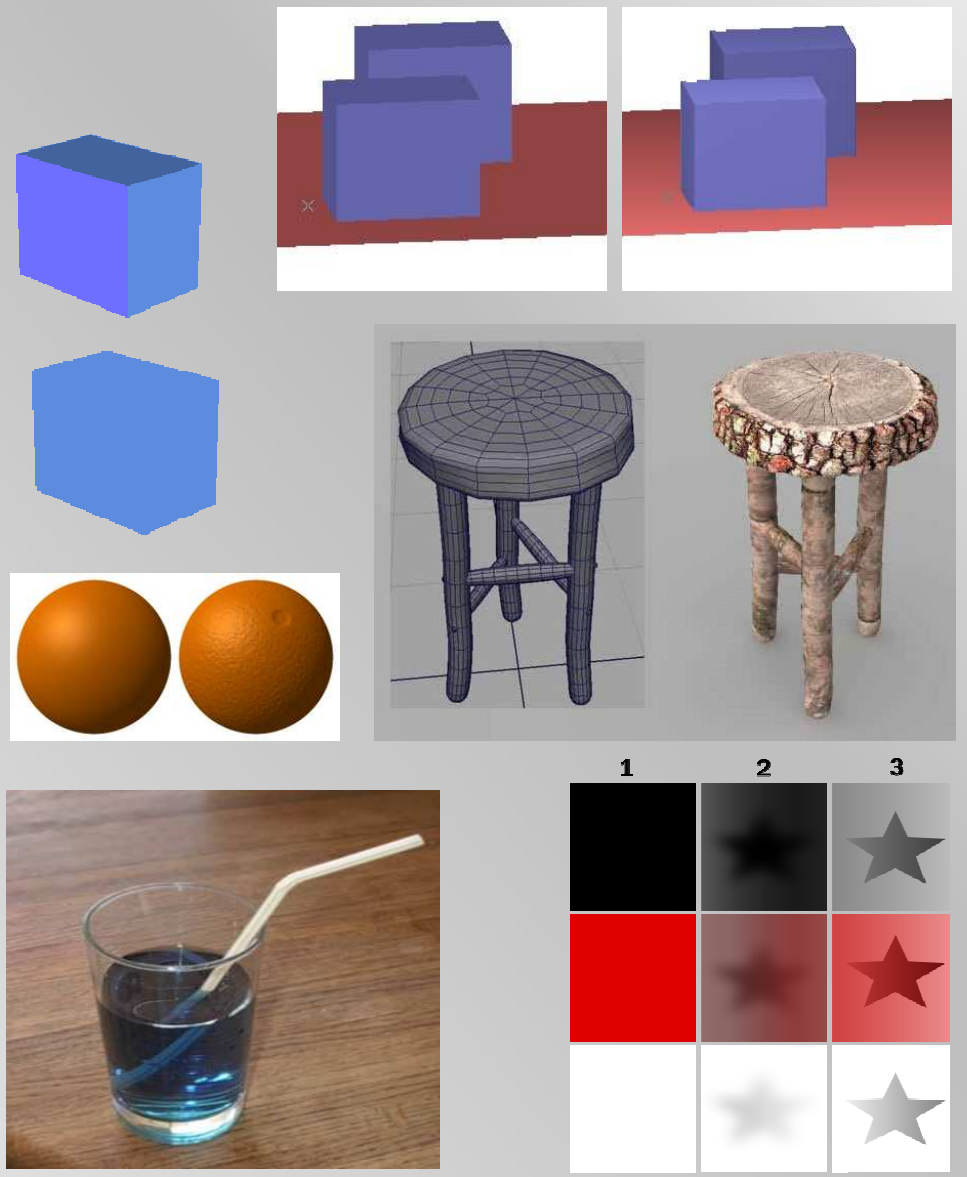
- Motivation
  - Re-usable: several application might share the same engine (**Unreal Engine** 3 is used by 79 games)



  - Abstracts communication with graphic API (makes the rest of the application independent from it)
  - Facilitates application maintenance

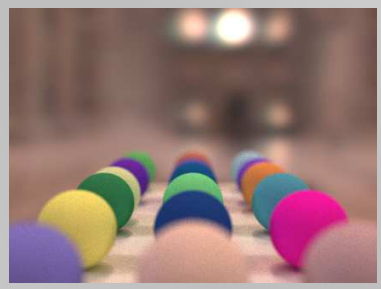
Rendering

- **Techniques I**
  - Shading
  - Textures mapping
  - Bump-mapping
  - Atmospheric effects
  - Reflections
  - Refractions
  - Transparencies
  - Translucencies



Rendering

- **Techniques II**
  - Diffraction
  - Global Illumination
  - Caustics
  - Depth of filed
  - “Motion Blur”
  - NPR (cell-shading)

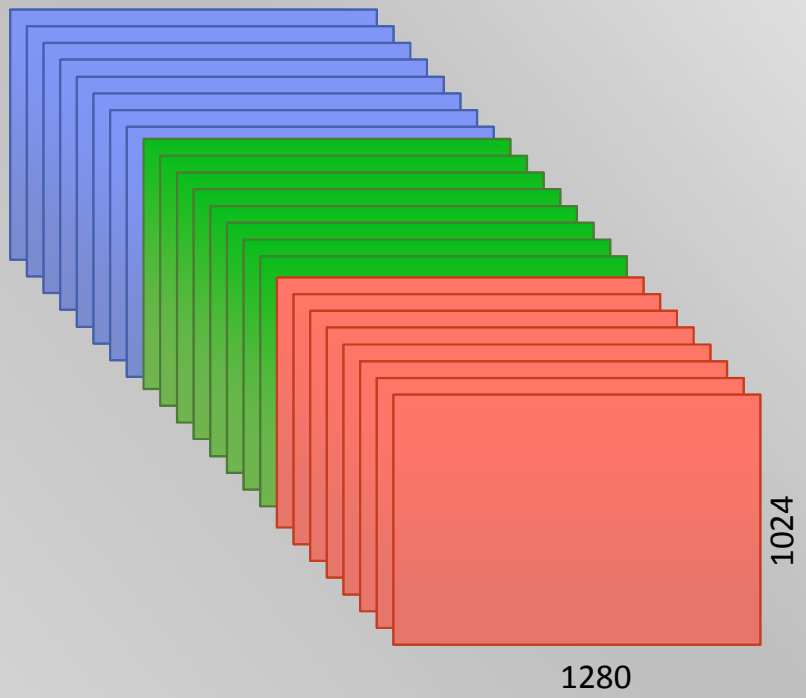


Render Engines
Graphics Engines / Games Engines

- Open-source or Free software:
  - Agar
  - Aleph One
  - Allegro library
  - Ardor3D
  - Game Blender
  - Box2D
  - ClanLib
  - Crystal Space
  - Cube
  - DarkPlaces
  - Delta3d
  - Doom engine
  - DXFramework
  - Exult
  - Genesis3D
  - id Tech 2 (Quake II)
  - id Tech 3 (Quake III)
  - ioquake3
  - Irrlicht Engine
  - jMonkeyEngine
  - Jogre
  - Nebula Device
  - OGRE Engine
  - MOGRE Engine
  - ORX
  - Panda3D
  - PLIB
  - Quake engine
  - RealmForge
  - Retribution Engine
  - Sauerbraten
  - Simple DirectMedia Layer
  - Spring
  - Stratagus
  - Xilon Engine II
  - Freeware:
  - Adventure Game Studio
  - Build engine
  - dim3
  - DX Studio
  - Game Maker Lite
  - Jogre
  - M.U.G.E.N
  - Panda3D
  - PhyreEngine
  - World Builder
  - Wintermute Engine
  - Comerciales:
  - AbyssalEngine
  - Alamo
  - Aurora Engine
  - Blade3D
  - BRender
  - C4 Engine
  - Coldstone game engine
  - CryEngine, CryEngine 2
  - Crystal Tools
  - Dagor Engine 3
  - id Tech 4 (Doom 3)
  - id Tech 5
  - DX Studio
  - Dunia Engine
  - Earth-4 Engine
  - Electron engine
  - Elflight Engine
  - Enigma Engine
  - Esenthel Engine
  - Euphoria
  - FPS Creator
  - Freescape (1986)
  - Frostbite Engine
  - Gamebryo
  - Glacier, Glacier2
  - GrimE
  - Havok
  - HeroEngine
  - IMUSE
  - INSANEInfinity Engine
  - Jade engine
  - Jedi
  - Kaneva Game Platform
  - Kinetica
  - Leadwerks Engine
  - Lithtech Jupiter Ex
  - Medusa
  - Monumental Technology Suite
  - Multimedia Fusion 2
  - Multiverse Network
  - Odyssey Engine
  - Onyx Engine
  - Quest3D
  - RAGE
  - Realm Crafter Standard Edition
  - RelentENGINE
  - RenderWare
  - Revolution3D
  - RPG Maker XP
  - SAGE engine
  - SCUMM engine
  - Serious Engine
  - Shark 3D
  - ShiVa
  - Silent Storm engine
  - Sith
  - Source engine
  - Torque Game Engine
  - Torque Game Engine Advanced
  - TOSHI
  - Truevision3D
  - Unigine
  - Unity
  - Unreal Engine
  - Vengeance engine
  - Vicious Engine
  - Virtools
  - Visual3D.NET Game Engine
  - WGAF
  - XnGine
  - Vertex3D

Frame buffer

- **Local memory** for images, mapped in a device
- Video hardware turns frame buffer content into a signal for the output device
- **Simple Frame buffer:**
  - Divided into planes
  - Each plane represents 1 bit of pixel color value
  - The planes have the output device size (1280x1024 p.e.)
  - For RGB of 8 bits, the frame buffer has a depth of 24 bits

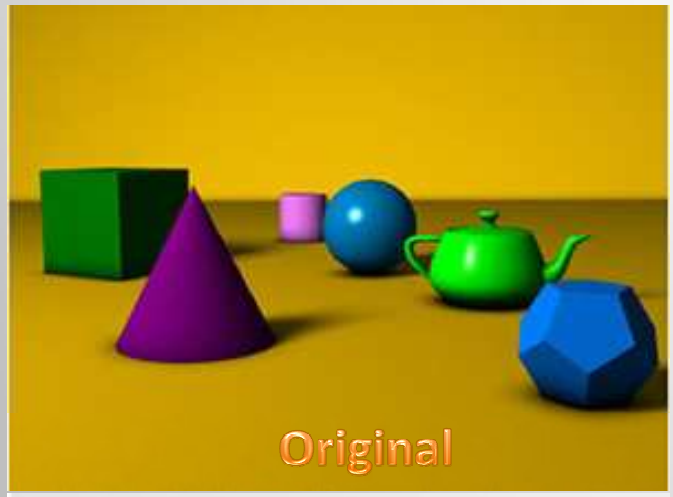


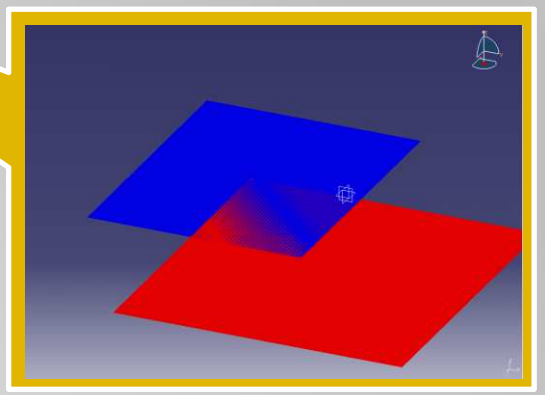
Frame buffer

- **More complex Frame buffer**
  - "Double buffering" (double the number of planes)
  - z-buffer for depth values
  - 3D vision, a set of buffers per eye
  - A-buffer, for transparencies, anti-aliased, shades
- **z-buffer**
  - Precision used is critical
  - **"z-fighting"** problems could arise:

    Several primitives have the same depth

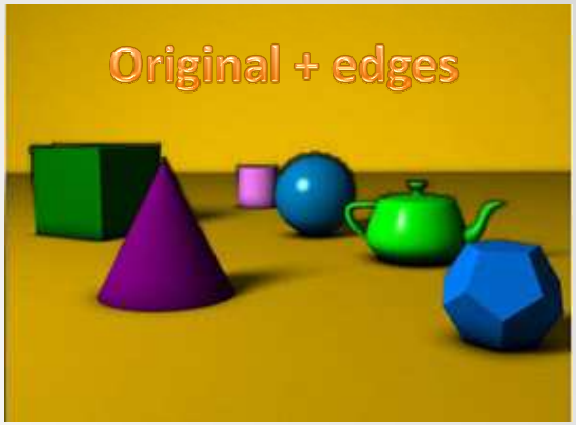
    ¿Which one becomes visible?
- **Hardware problem** for frame buffer
  - Really fast memories are needed





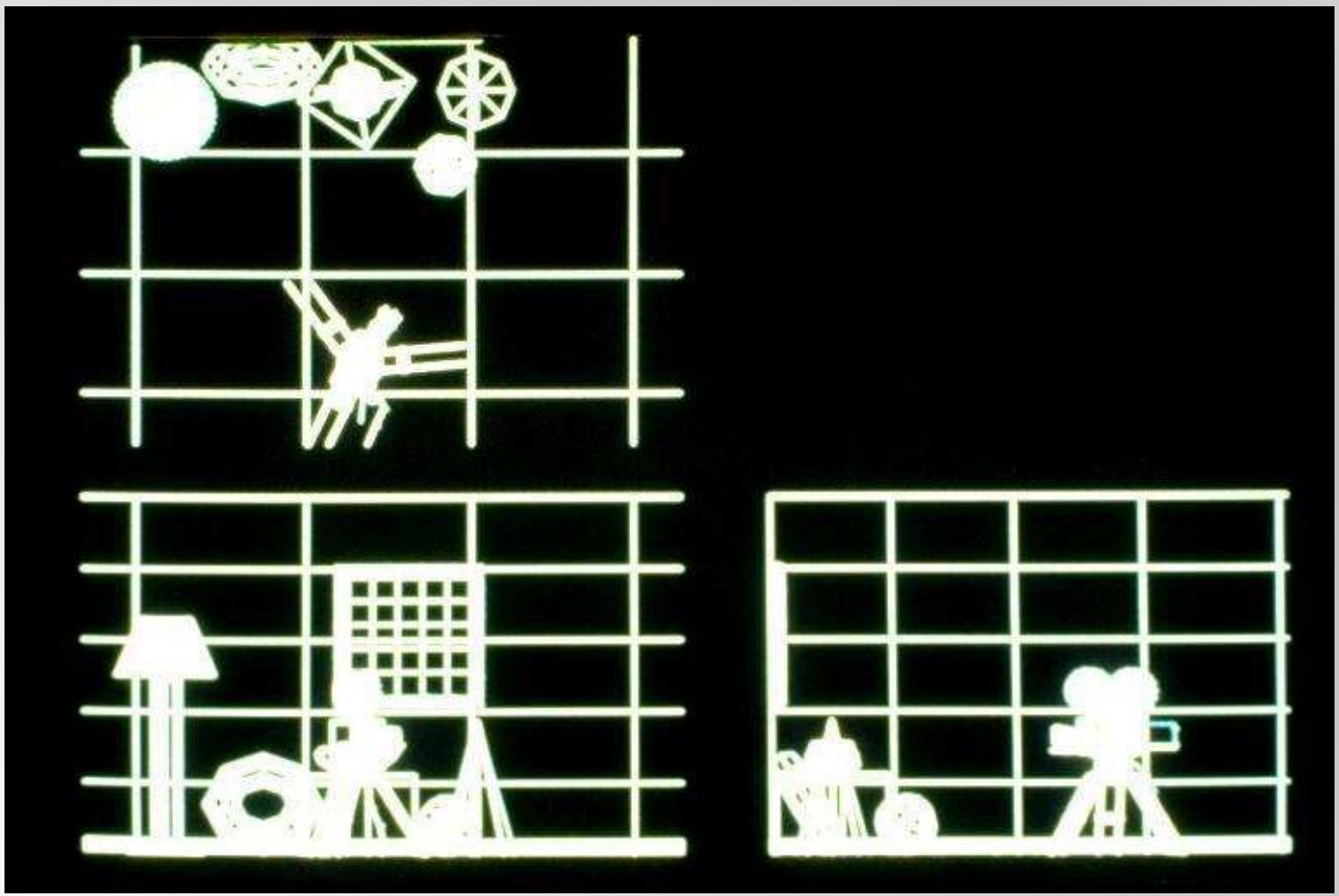
Frame buffer – some tricks

- If we paint abrupt changes in the Z-buffer



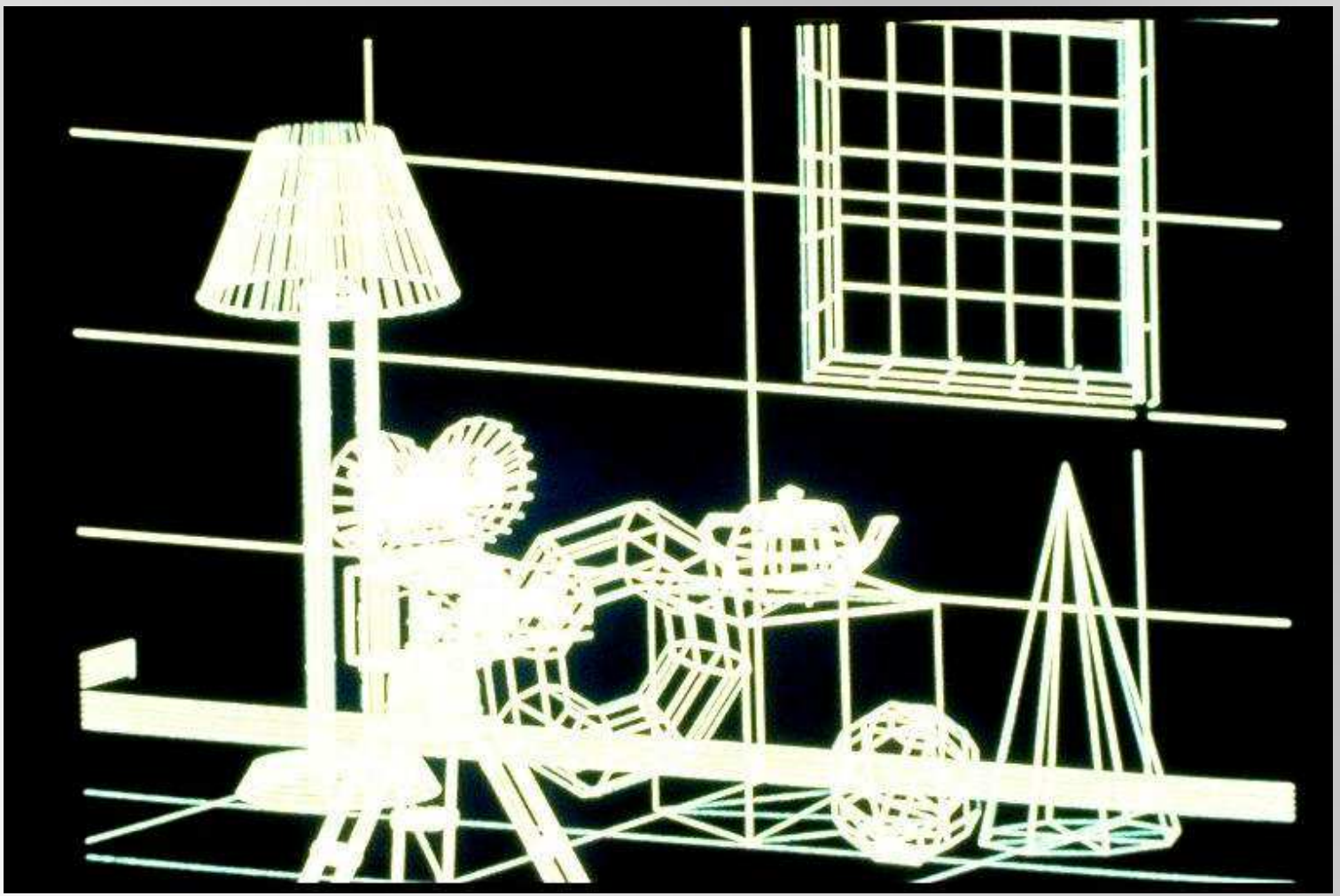
Graphic Process Example

- Wireframe with orthographic views



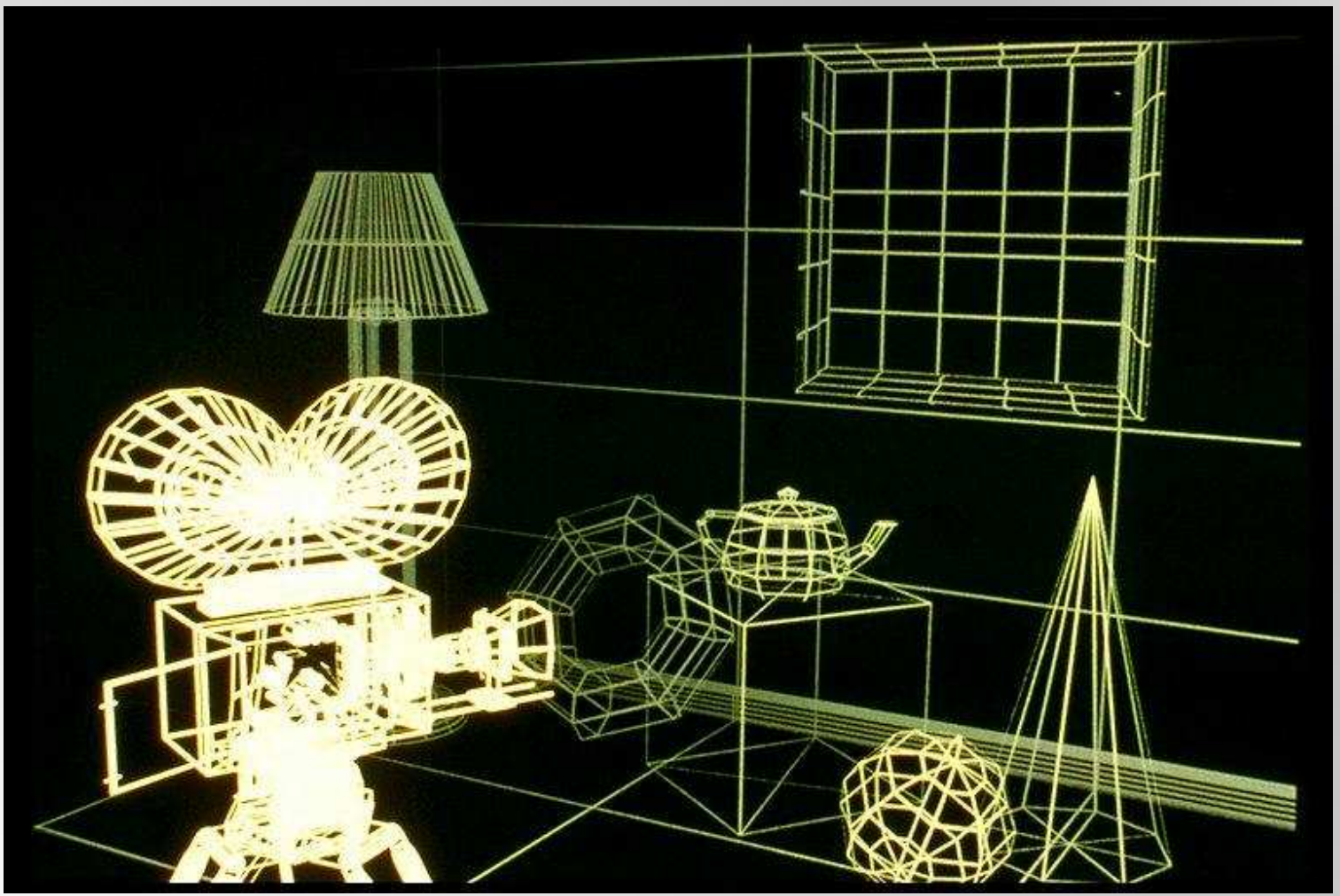
Graphic Process Example

- Wireframe with perspective views



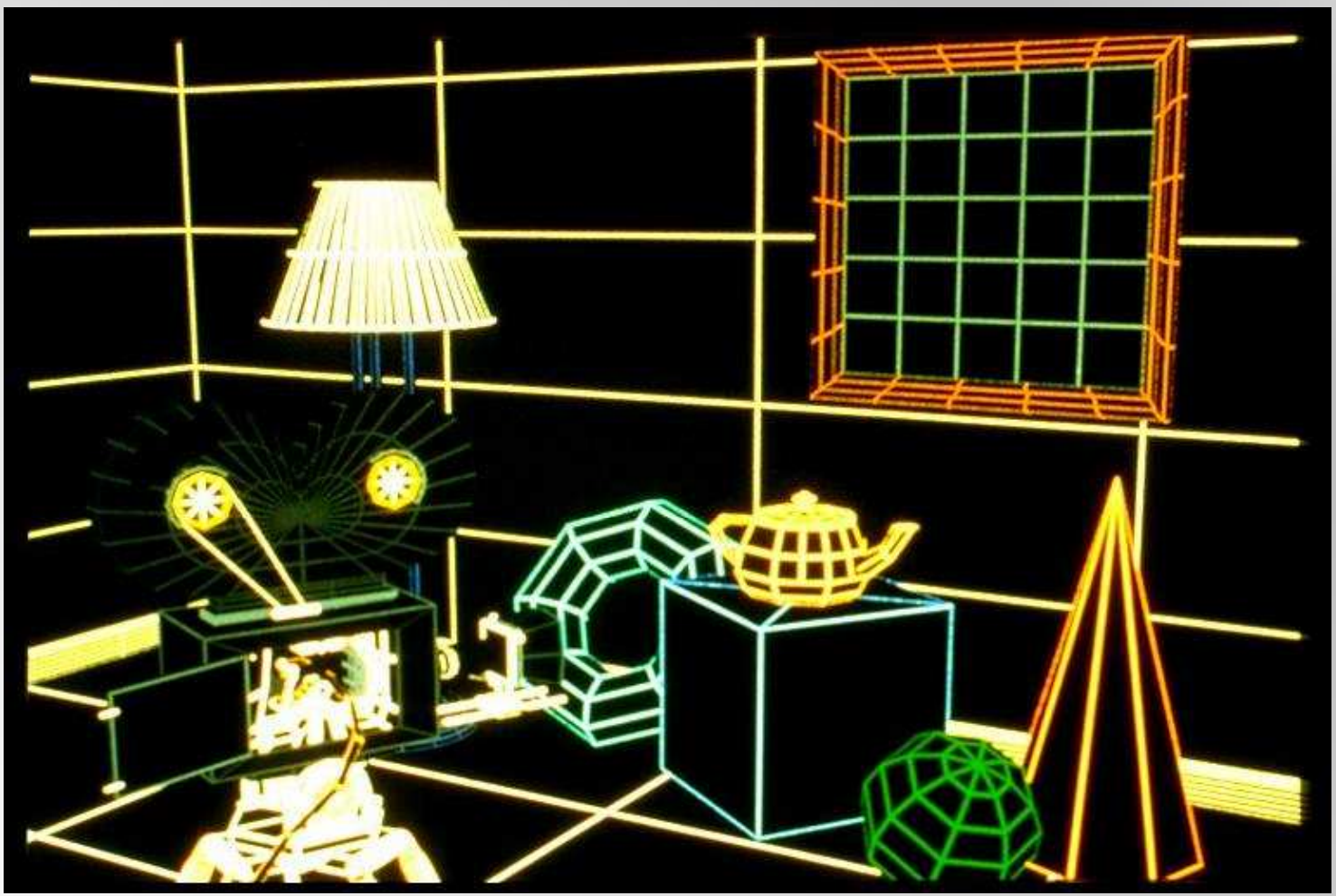
Graphic Process Example

- Enhance depth



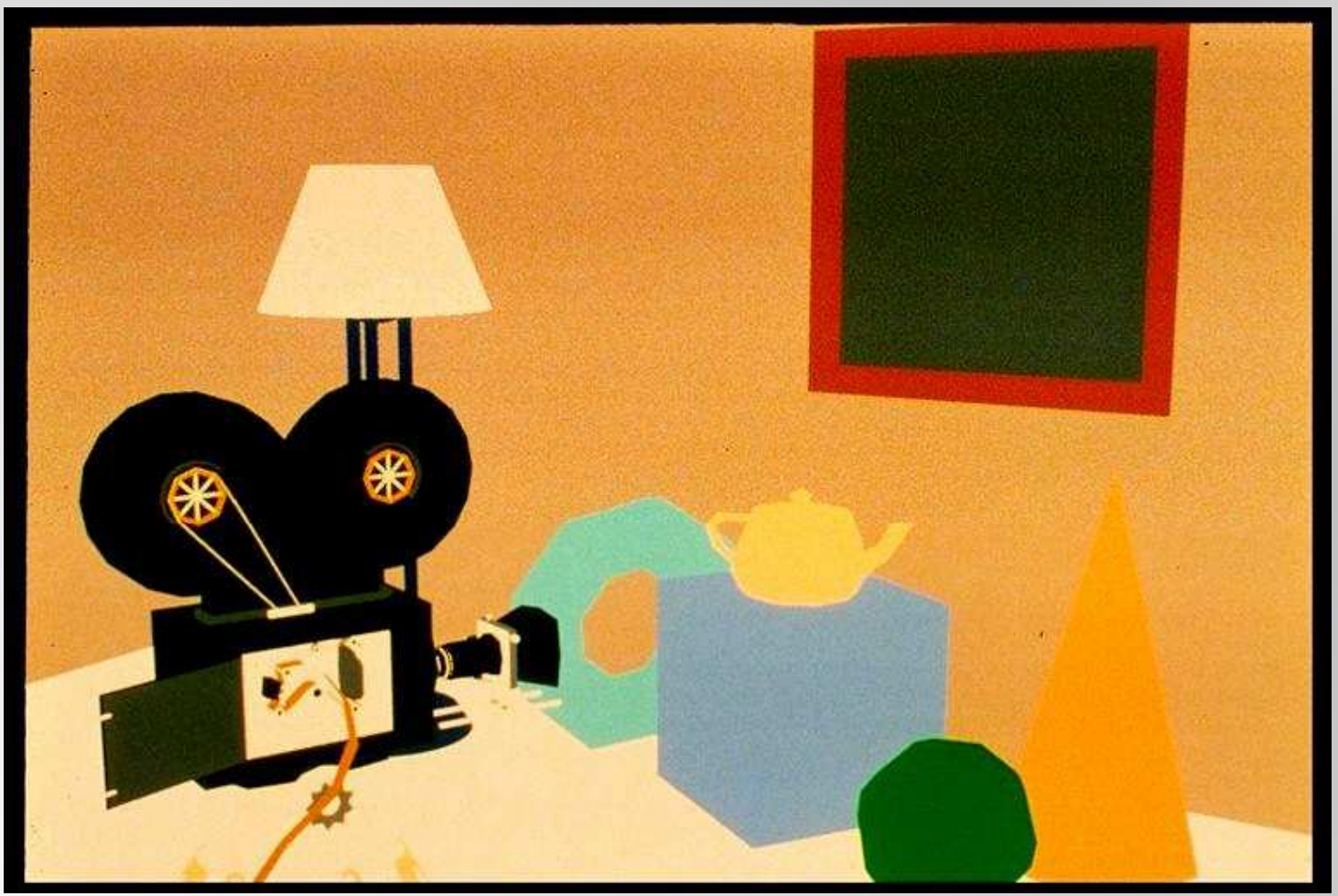
Graphic Process Example

- Eliminate hidden lines and color the scene



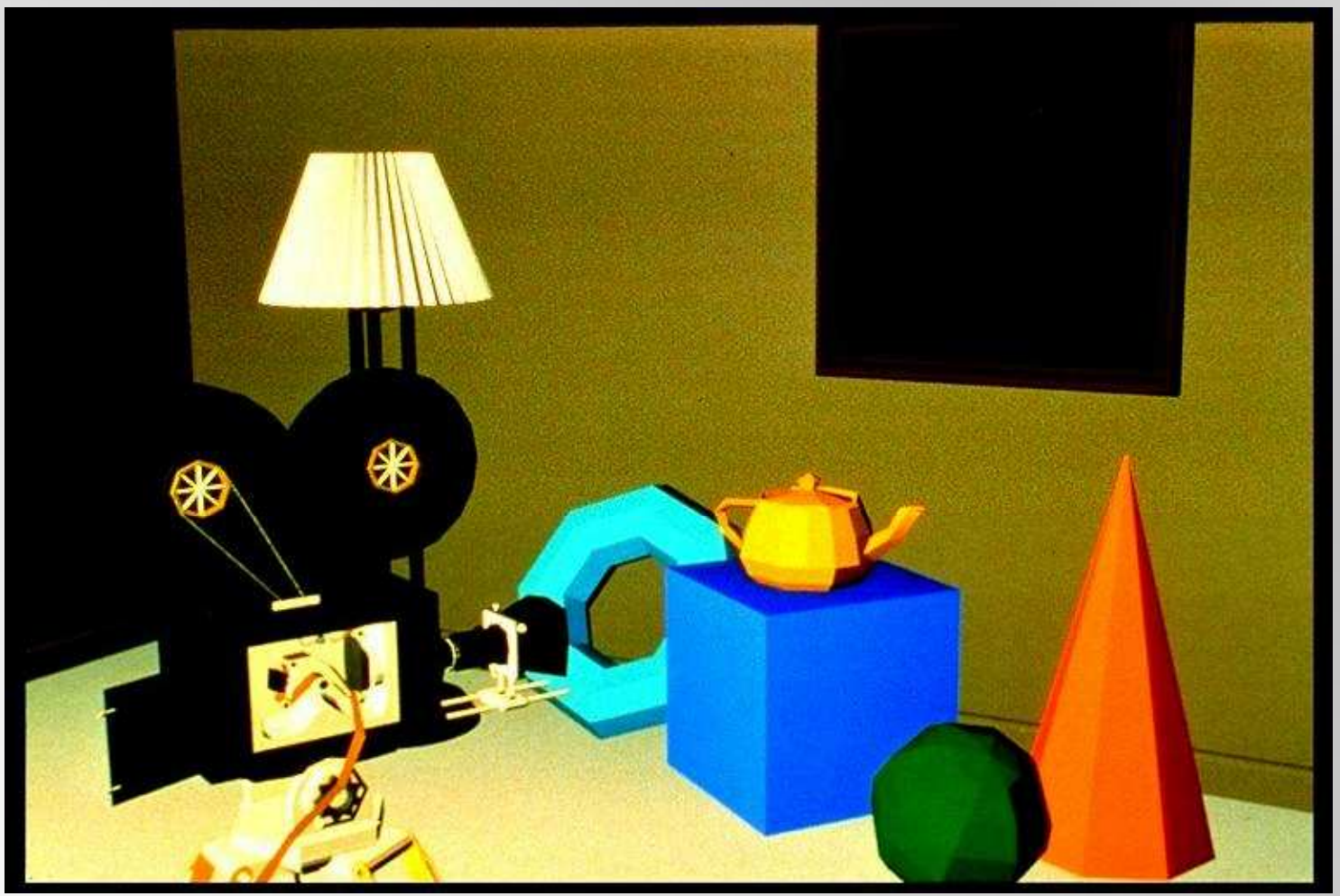
Graphic Process Example

- Constant shading



Graphic Process Example

- Lambert shading (plane shading)



Graphic Process Example

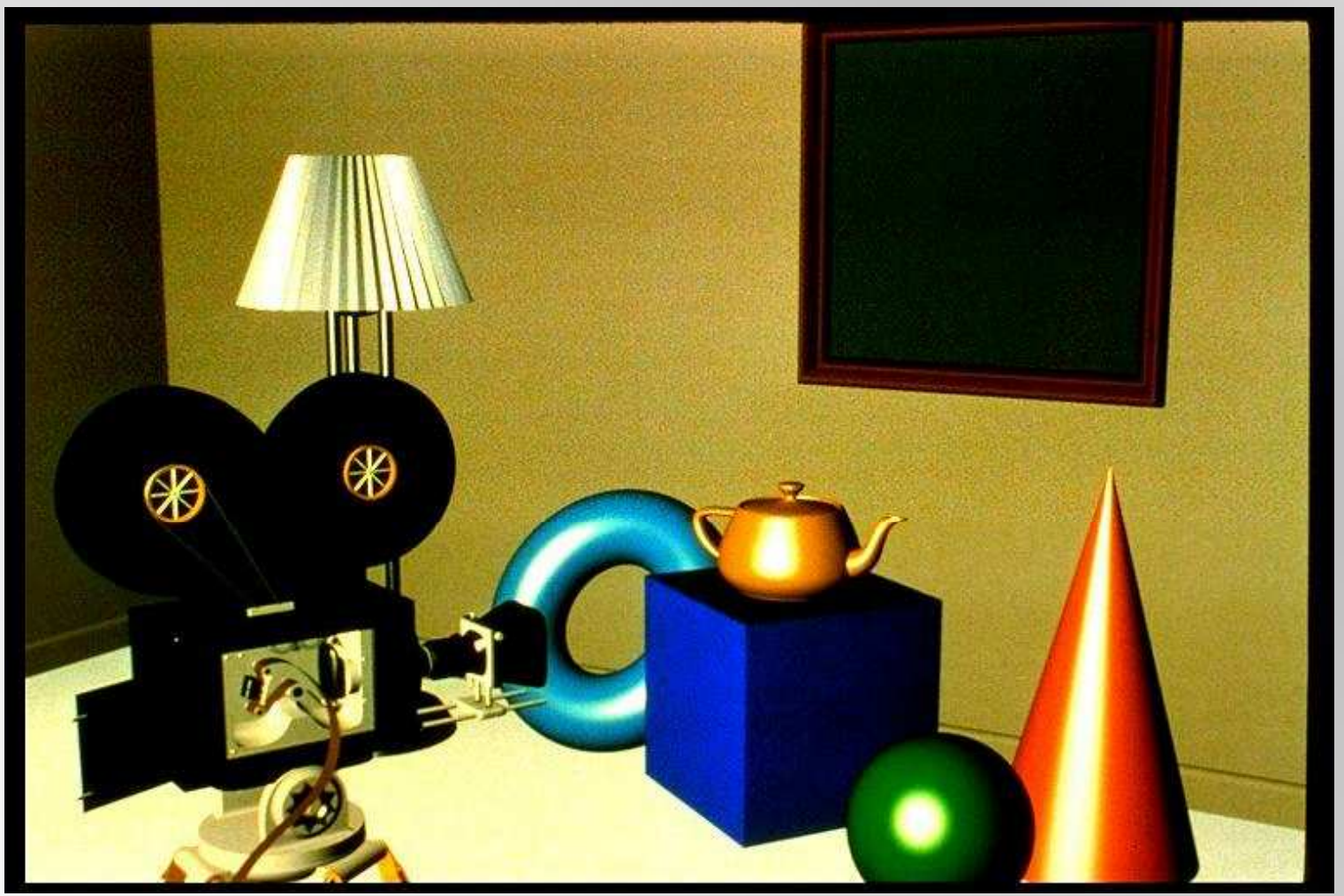
- Gouraud shading without specular reflectance

Graphic Process Example

- Specular brightness

Graphic Process Example

- Phong shading



Graphic Process Example

- Textures Mapping

Graphic Process Example

- Reflections, shades and bump-mupping

Example: Disney & TSL