# *Computer Graphics*

***by Ruen-Rone Lee***

***ICL/ITRI***

# Introduction to Computer Graphics

***Basic concept of computer graphics***

# What is Computer Graphics

***Differences between IP, CV/PR, and CG***

***CG Applications***

***Graphics System***

# Categorization

- What are the difference between Computer Graphics, Computer Vision (Pattern Recognition) and Image Processing

| *Input* | *Output* | *Category* |
|---|---|---|
| Image | Image | Image Processing |
| Description<br>Images | Description | Computer Vision<br>Pattern Recognition |
| Description | Image | Computer Graphics |

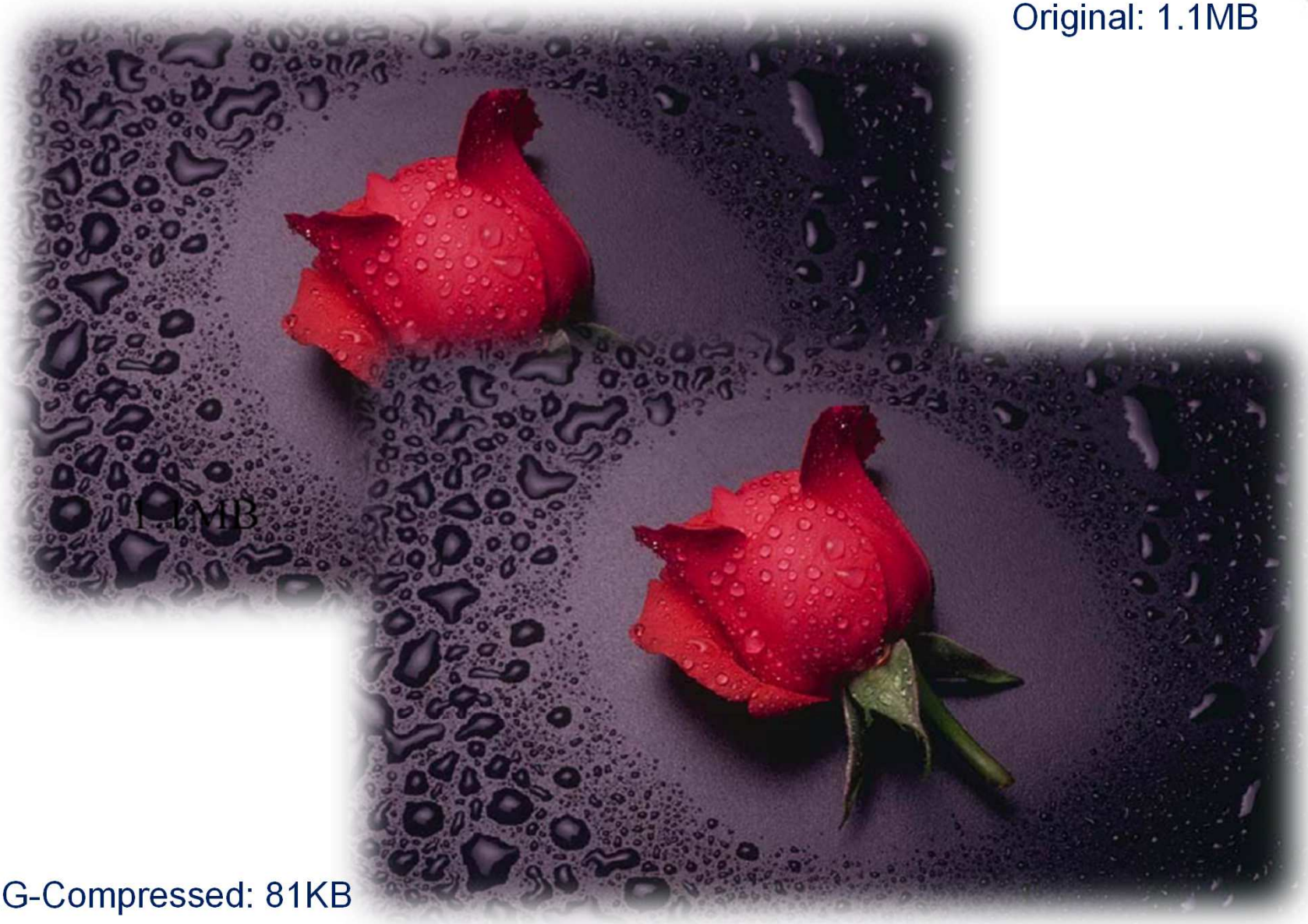
# *Image Processing*

- **The *analysis* of scenes, or the *reconstruction* of models of 2D or 3D objects from their pictures**

# Example: Image Compression

# Example: Image Compression

Original: 1.1MB




JPEG-Compressed: 81KB

# Example: Image Compression

Original: 1.1MB

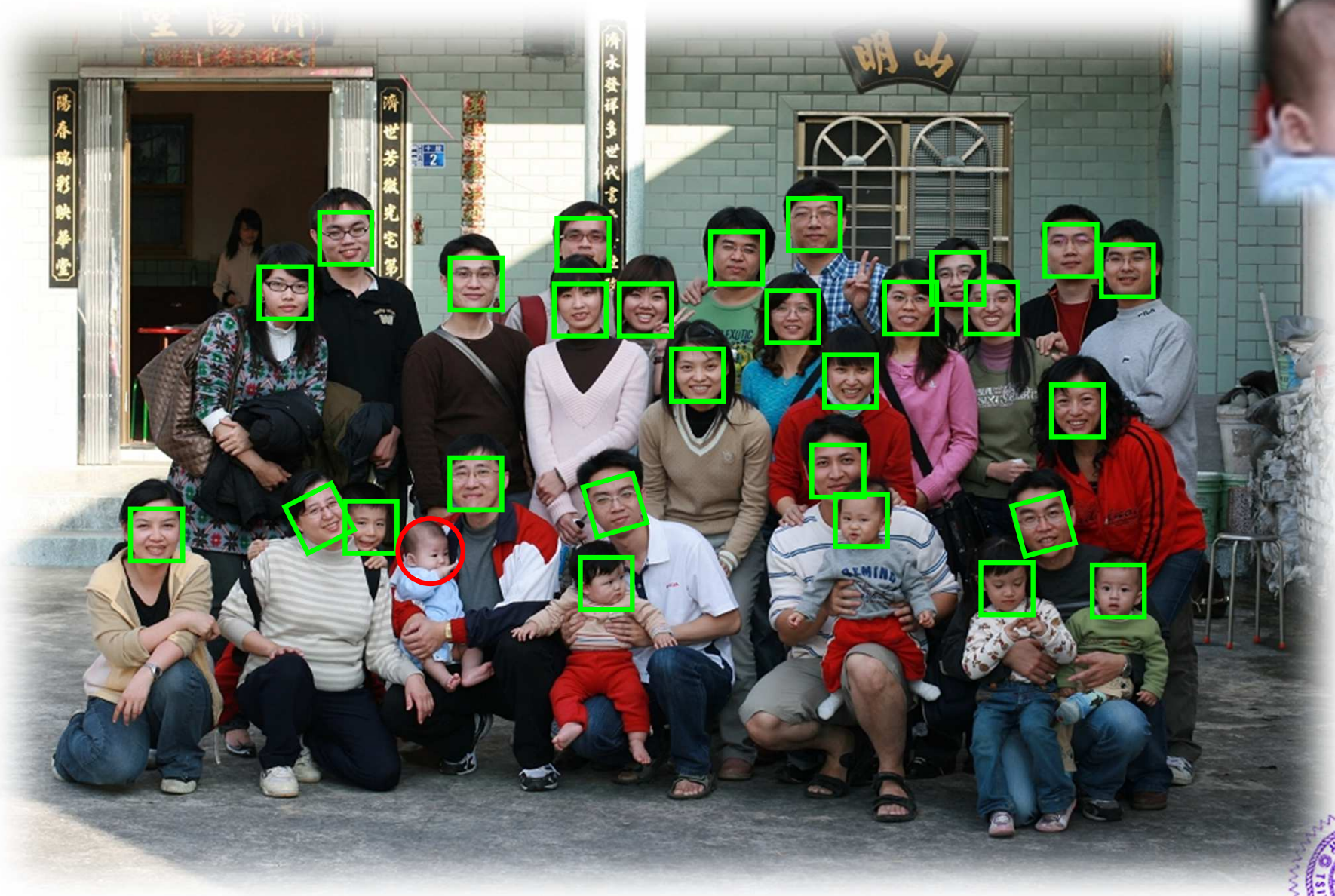
JPEG-Compressed: 81KB

# Example: Edge Detection

# *Computer Vision / Pattern Recognition*

- **Computer Vision is concerned with the theory for building artificial system that obtain information from images**
- **Pattern Recognition aims to classify data (patterns) based on either prior knowledge or on statistical information extracted from the patterns**

# Application of Computer Vision



# *Computer Graphics*

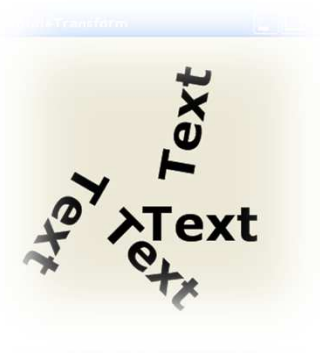
- **Computer graphics concerns the pictorial *synthesis* of real or imaginary objects from their computer-based models.**

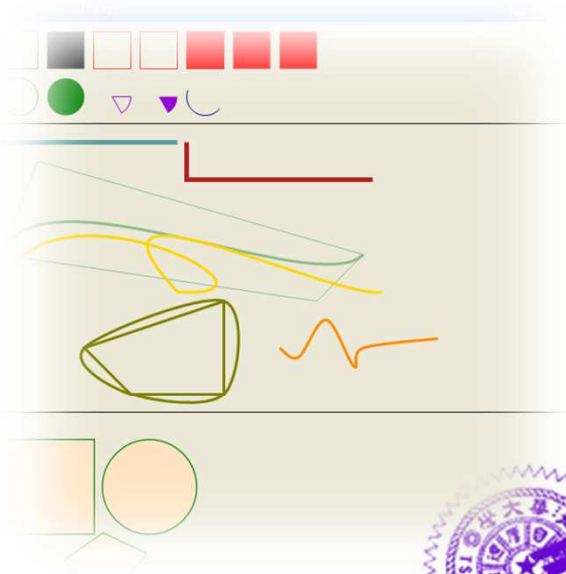
Columbia Pictures
Stuart Little

DreamWorks Pictures
Transformers

# Types of Computer Graphics

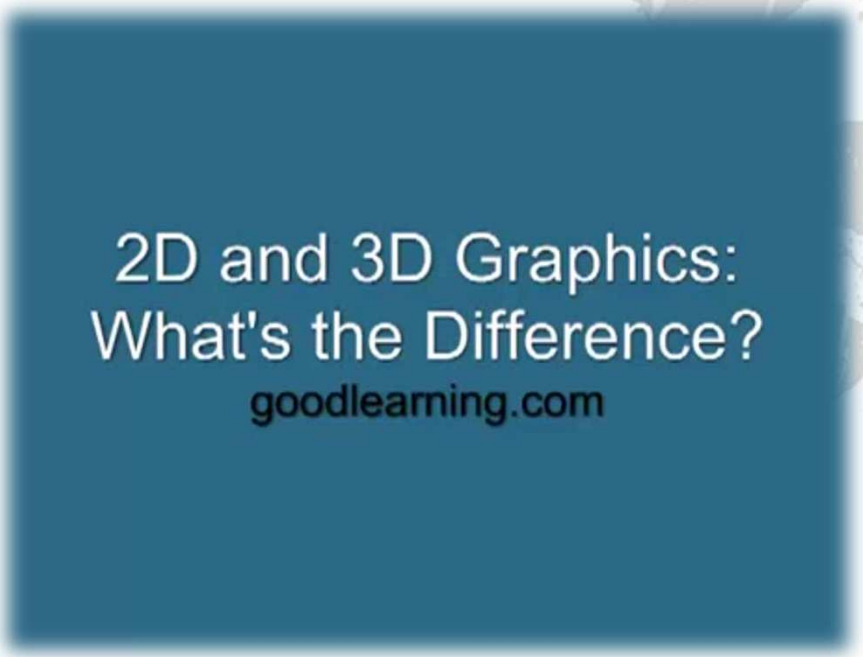
- 2D Graphics
  - The object being drawn is defined in 2D (*x*, *y*) coordinate system
  - Eg. Texts, 2D lines / arcs / polygons, images, …
  - Vector graphics, 2D drawing tools, Adobe Flash, …



# Types of Computer Graphics

- **3D Graphics**
  - The object being drawn is defined in 3D (*x*, *y*, *z*) coordinate system
  - Project the objects onto 2D projection plane and render
  - Model the human visual system with one eye only
  - Similar to the process in using camera to take a photo on the 3D world



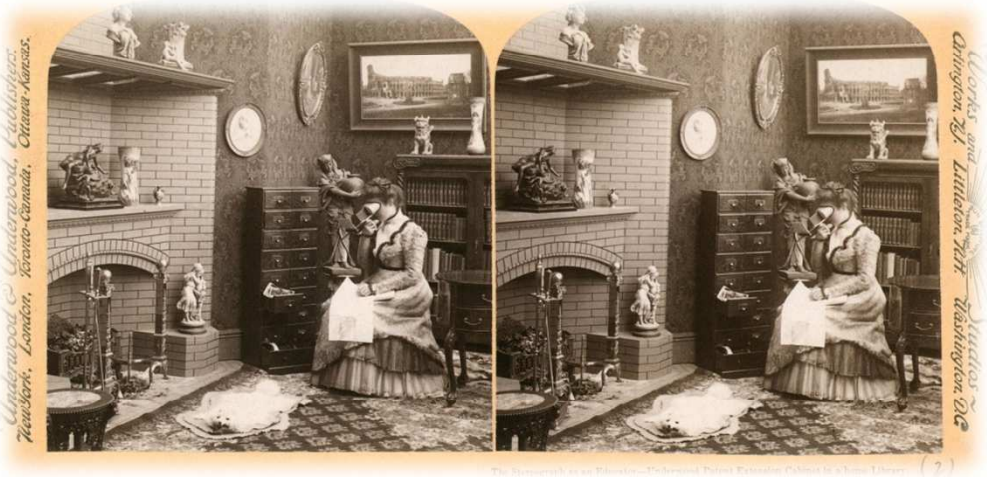
# Types of Computer Graphics

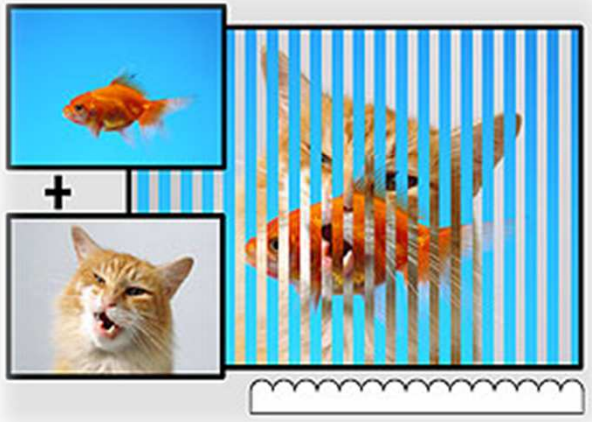
- **Stereo 3D Graphics**
  - **Modeling human visual system with both eyes**
  - **Render left and right images with respect to the images seen by the left and right eyes, respectively**
  - **Need special devices, such as glasses or display, to present the stereo 3D effect**




# Types of Computer Graphics

- Example Stereo 3D Image



# *Computer Graphics Applications*

***Computer Games / Mobile Games***

***Modeling and Animations***

***Photo-Realistic Rendering***

***Visualization and Simulation***

***Visual Effects and Composition***

***Artwork Creation***

***Graphics User Interface***

***Virtual Reality / Augmented Reality***

***E-Commerce / E-Book***

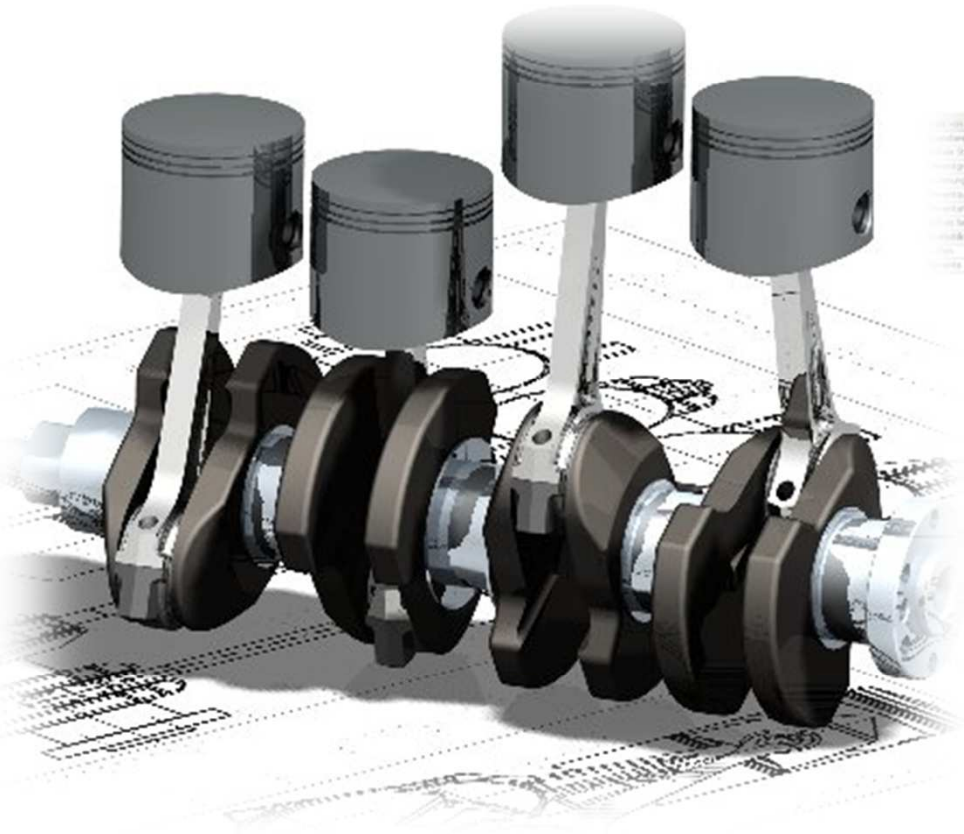
***…***

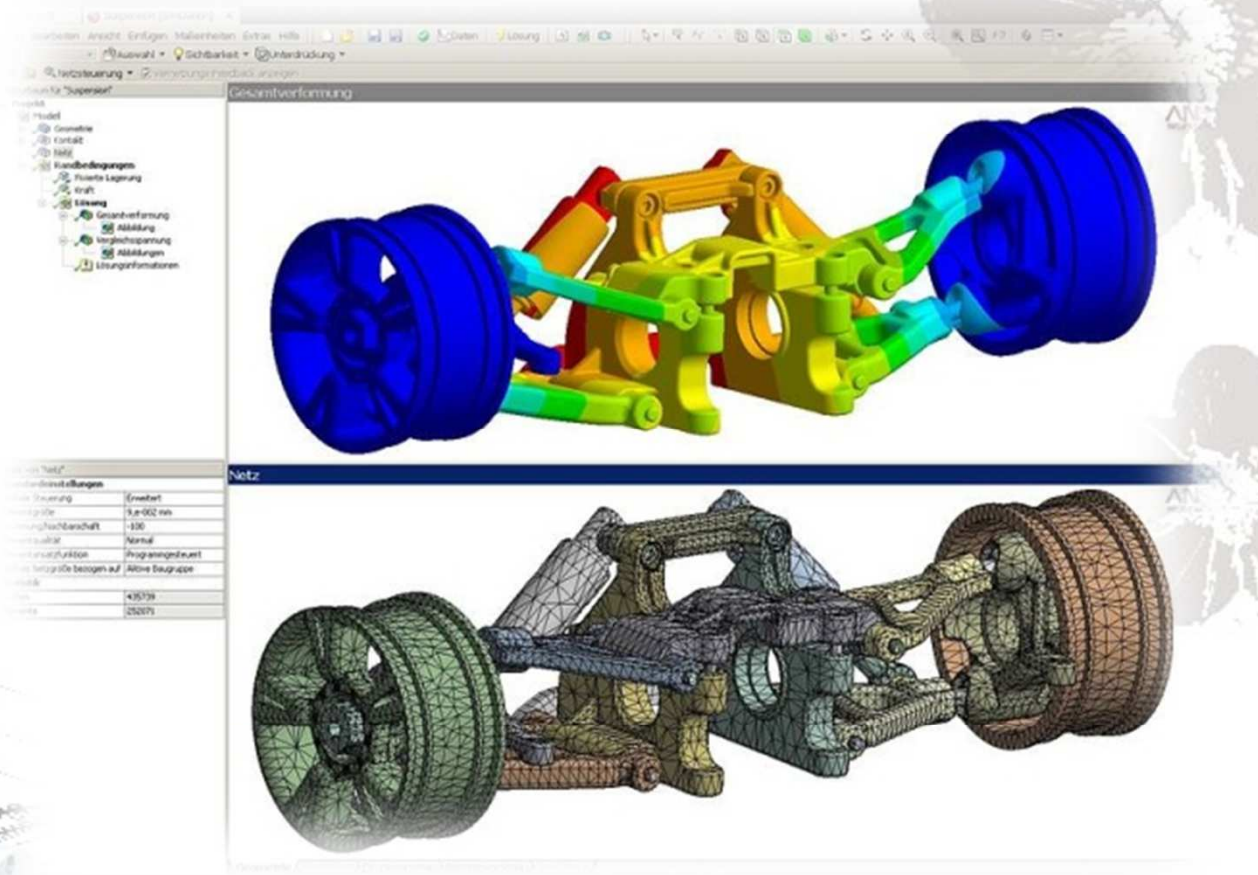
# Computer Games / Mobile Games


**2D Game**
**Angry Birds, Rovio**

**3D Game**
**Dead or Alive 4, TECMO**

# Modeling





# Modeling

# Animations

Big Hero 6 (Disney)

Hotel Transylvania 2 (Sony Pictures Animation)

Minions (Illumination)

Inside Out (Pixar)

The Good Dinosaur (Pixar)

# Animation

- **Facial Expression Transfer**

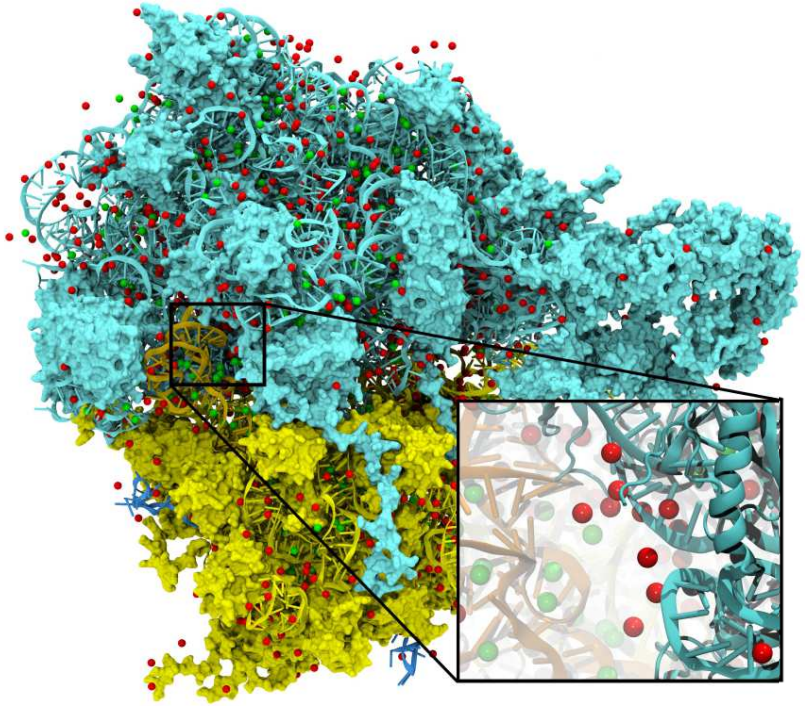
# Photo-Realistic Rendering

# Photo-Realistic Rendering

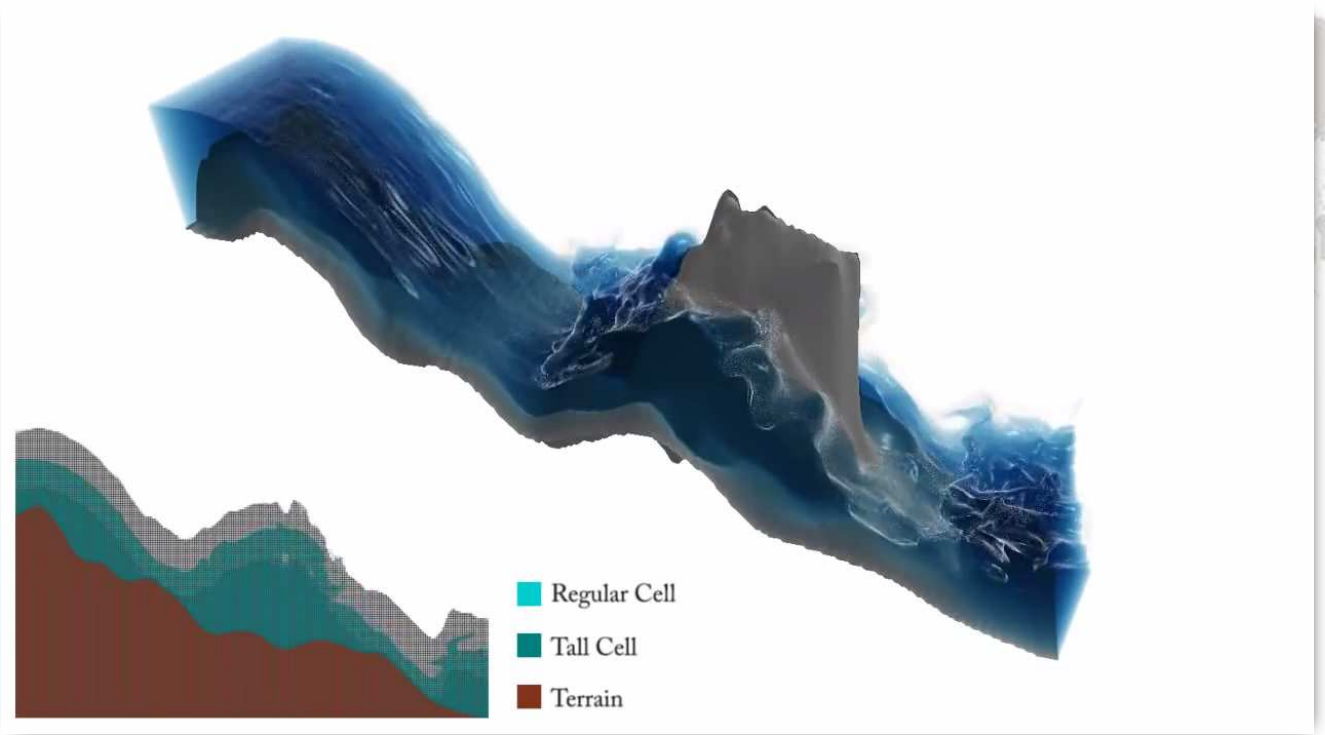
# Photo-Realistic Rendering

# Visualization and Simulation

- **Scientific Visualization and Physic Simulation**



Molecular Modeling




Real-Time Fluid Simulation

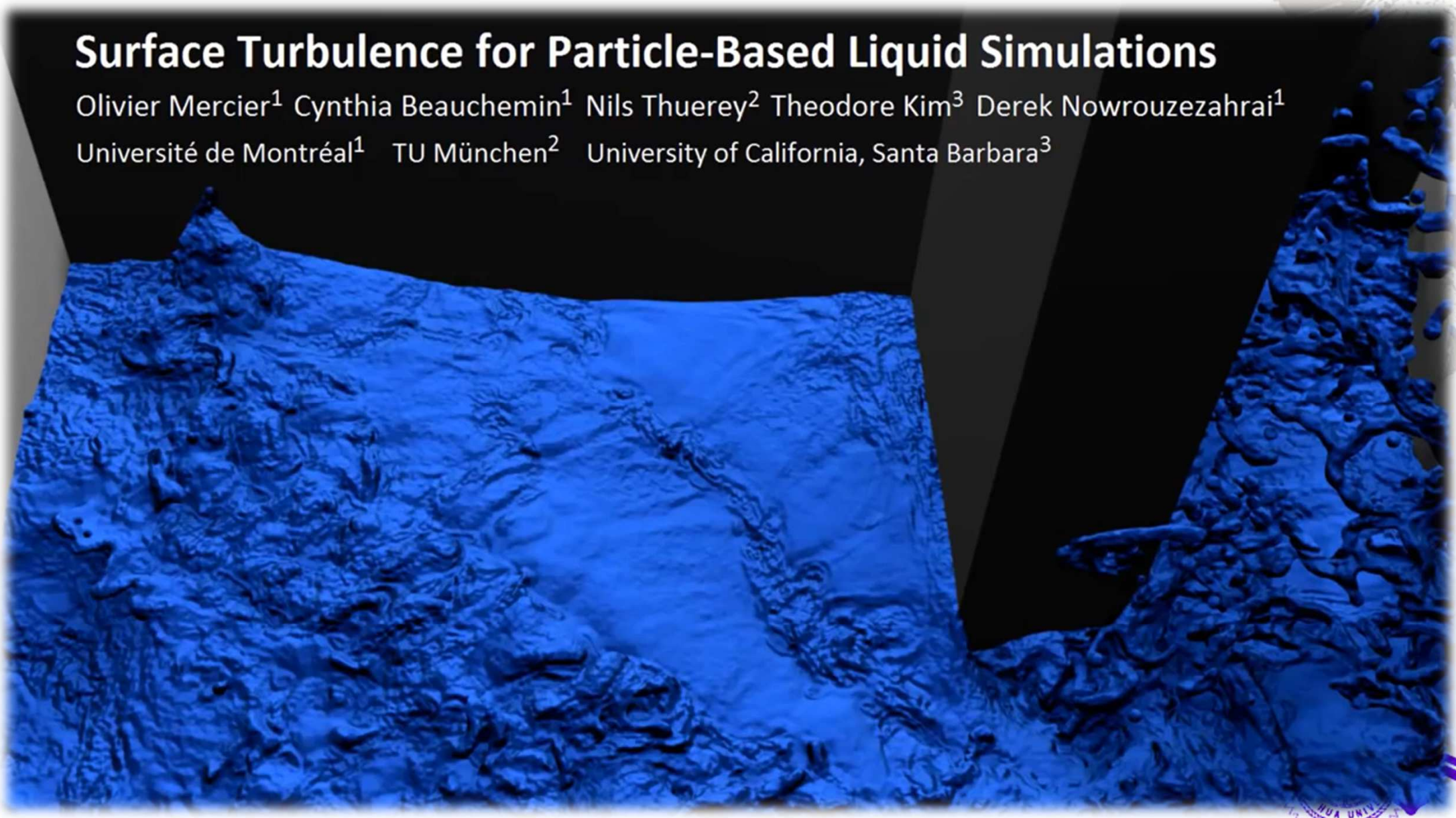
# *Visualization and Simulation*

- **Water Simulation**

# *Visualization and Simulation*

- **Fire Simulation**

# Visualization and Simulation



# Artwork Creation

**Wetbrush: GPU-based 3D Painting Simulation at the Bristle Level**

Zhili Chen[1,2] Byungmoon Kim[2] Daichi Ito[2] Huamin Wang[1]

[1] The Ohio State University [2] Adobe Research

# *Artwork Creation*

## Tone- and Feature-Aware Circular Scribble Art

Chun-Chia Chiu, Yi-Hsiang Lo, Ruen-Rone Lee, Hung-Kuo Chu

National Tsing Hua University, R.O.C

# Graphics User Interface

- Windows UI

Microsoft Windows 7

Microsoft Windows 8

Mac OSX

Ubuntu Linux

# Graphics User Interface

- Not just Windows

# Graphics User Interface

- GUI and Information Display

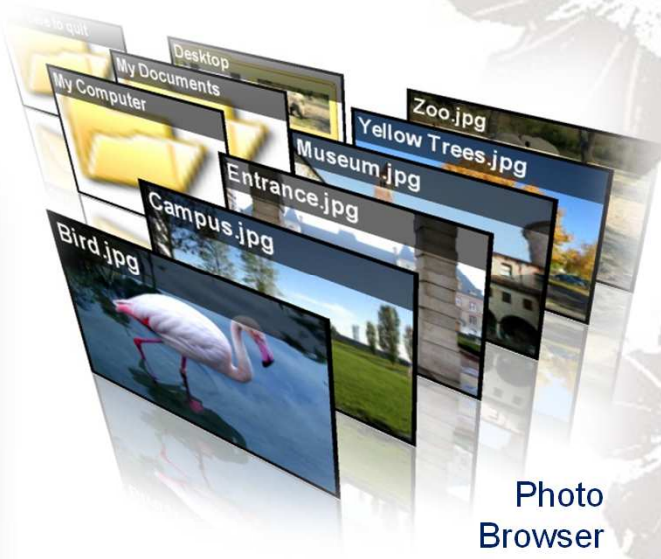



Photo Browser


Photo Browser

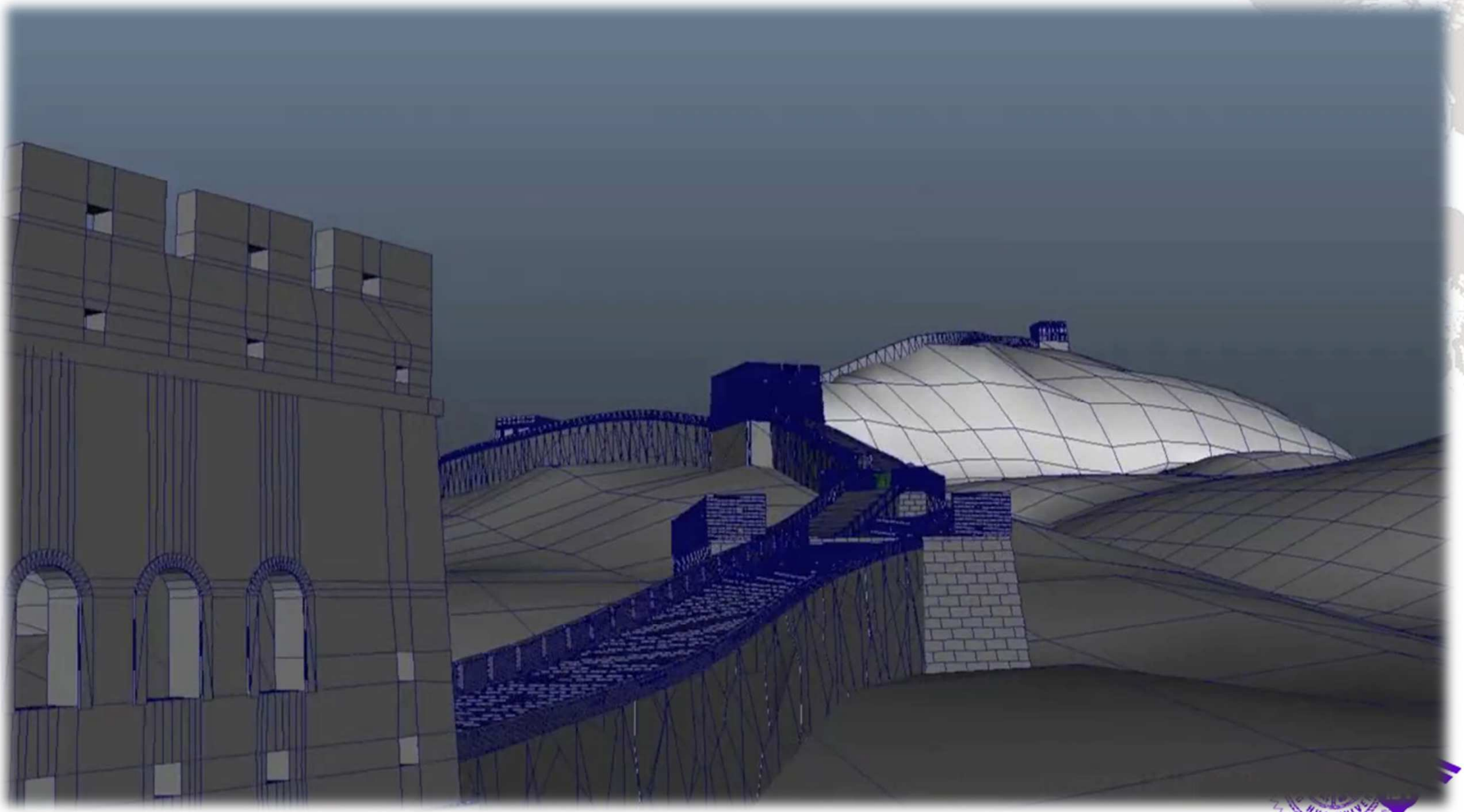

Car Dashboard

Cockpit Display


GPS Navigation

# Visual Effect and Composition


# Visual Effect and Composition



# Visual Effect and Composition

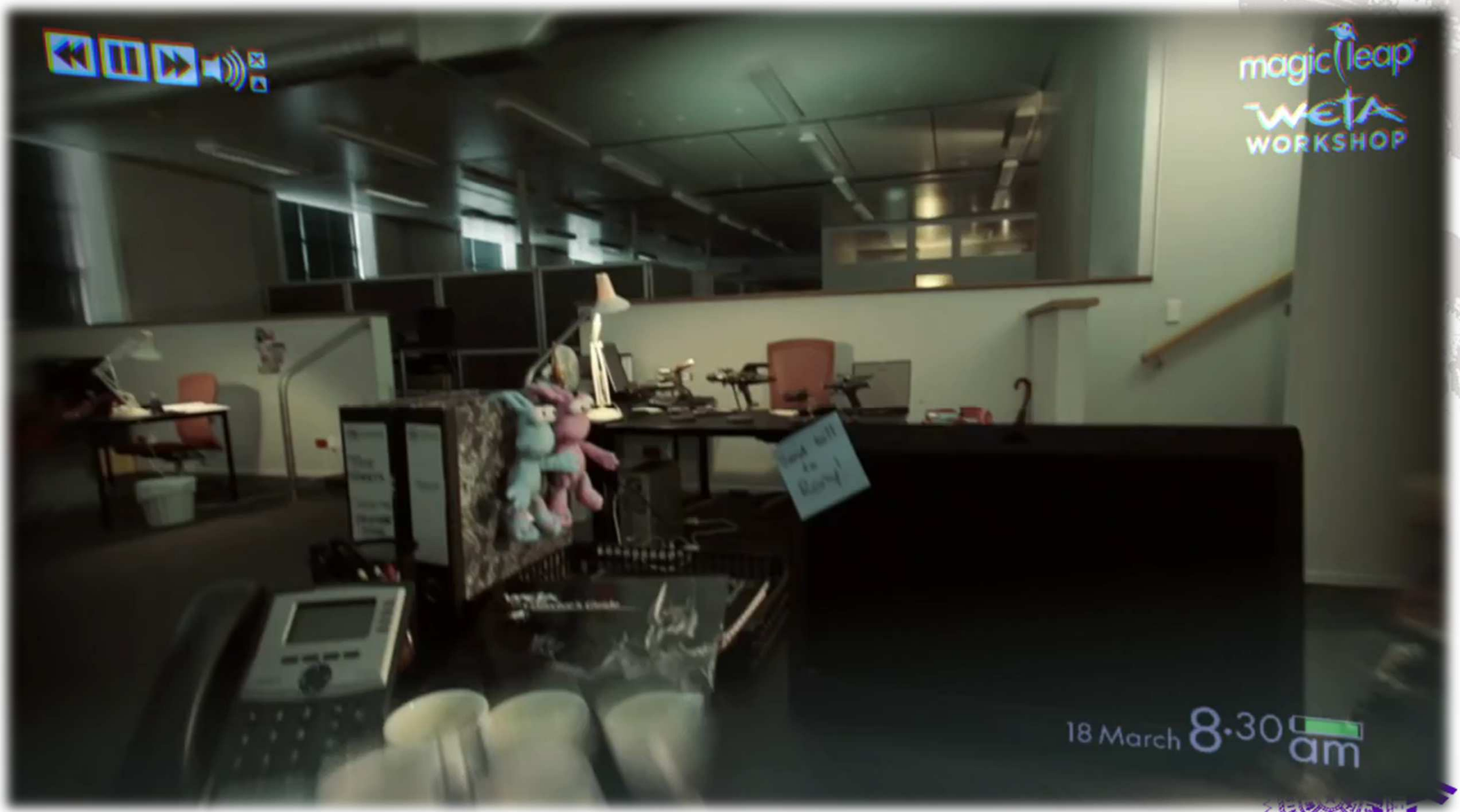
# Visual Effect and Composition

# Virtual Reality

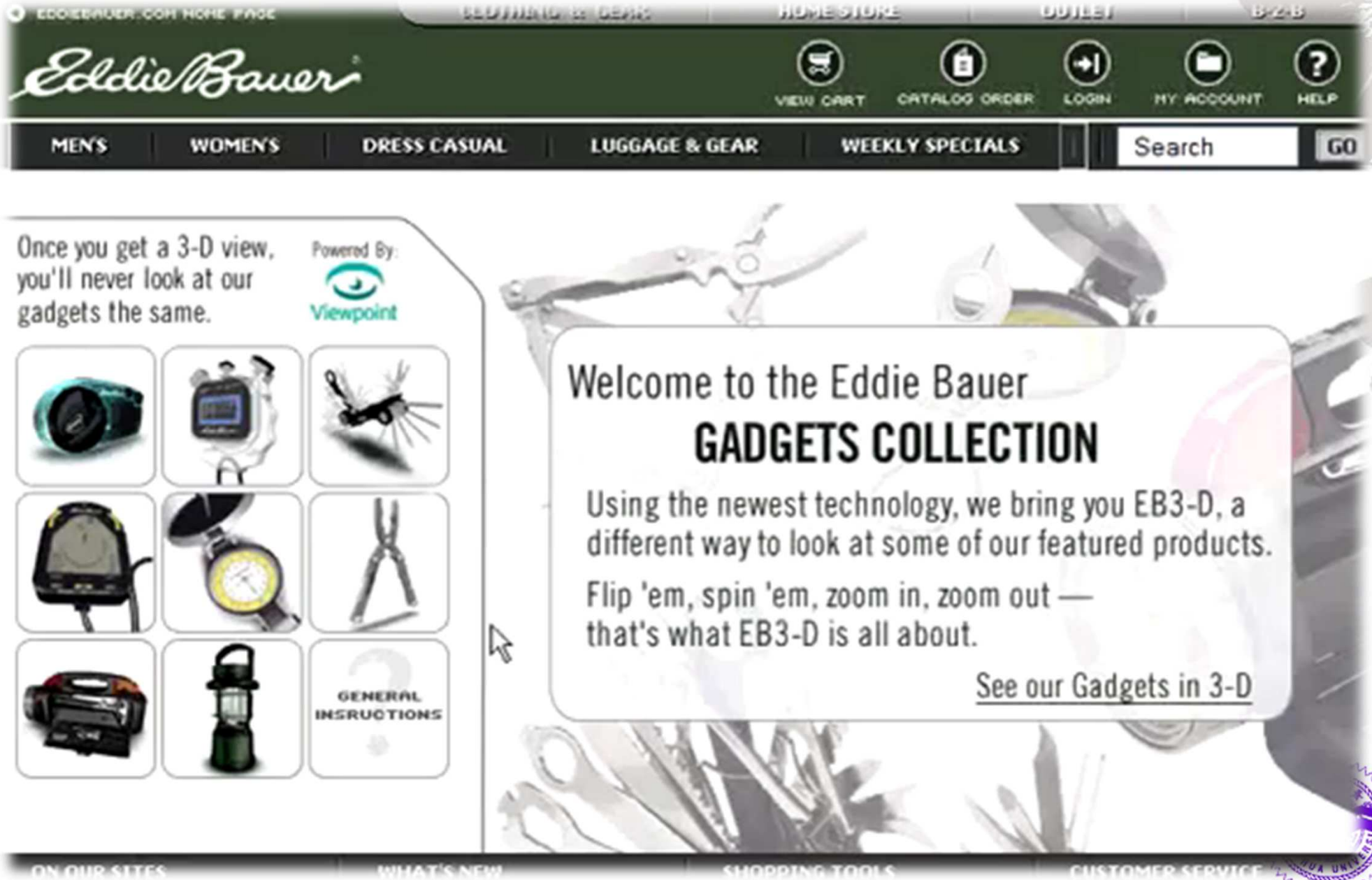
# Virtual Reality

# Augmented Reality

# Augmented Reality

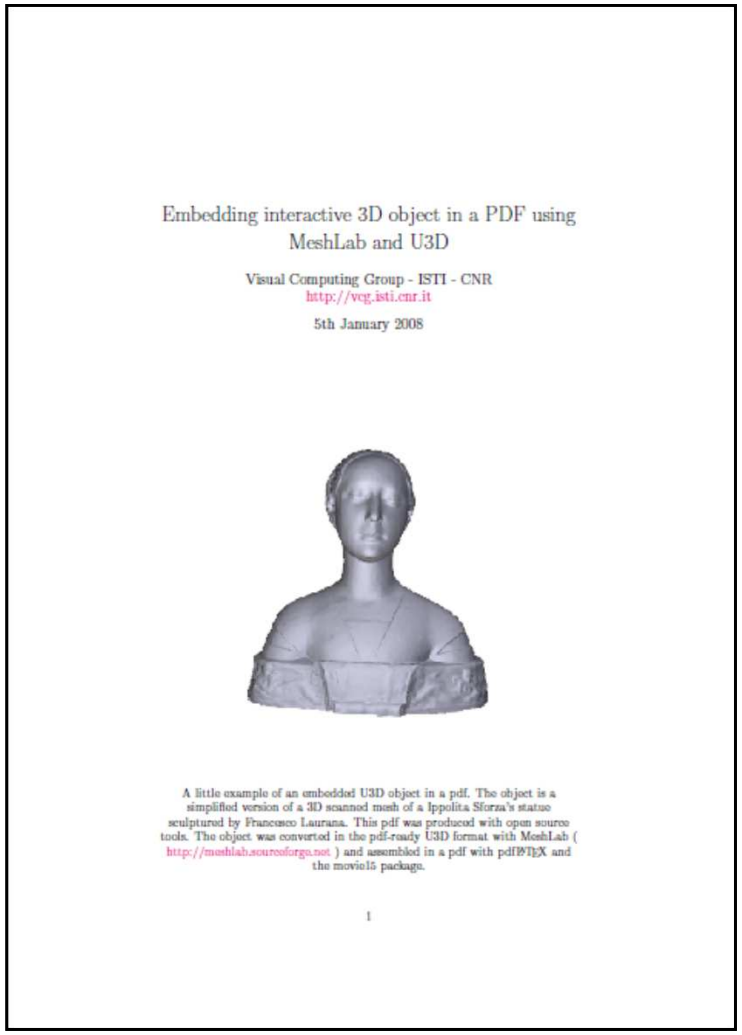


# E-Commerce


Eddie Bauer
VIEW CART
CATALOG ORDER
LOGIN
MY ACCOUNT
HELP
MEN'S
WOMEN'S
DRESS CASUAL
LUGGAGE & GEAR
WEEKLY SPECIALS
Search
GO
Once you get a 3-D view, you'll never look at our gadgets the same.
Powered By:
Viewpoint
GENERAL INSRUCTIONS
Welcome to the Eddie Bauer
GADGETS COLLECTION
Using the newest technology, we bring you EB3-D, a different way to look at some of our featured products.
Flip 'em, spin 'em, zoom in, zoom out — that's what EB3-D is all about.
See our Gadgets in 3-D
ON OUR SITES
WHAT'S NEW
SHOPPING TOOLS
CUSTOMER SERVICE

# E-Book Document

- **U3D format: pdf with 3D for better illustration**



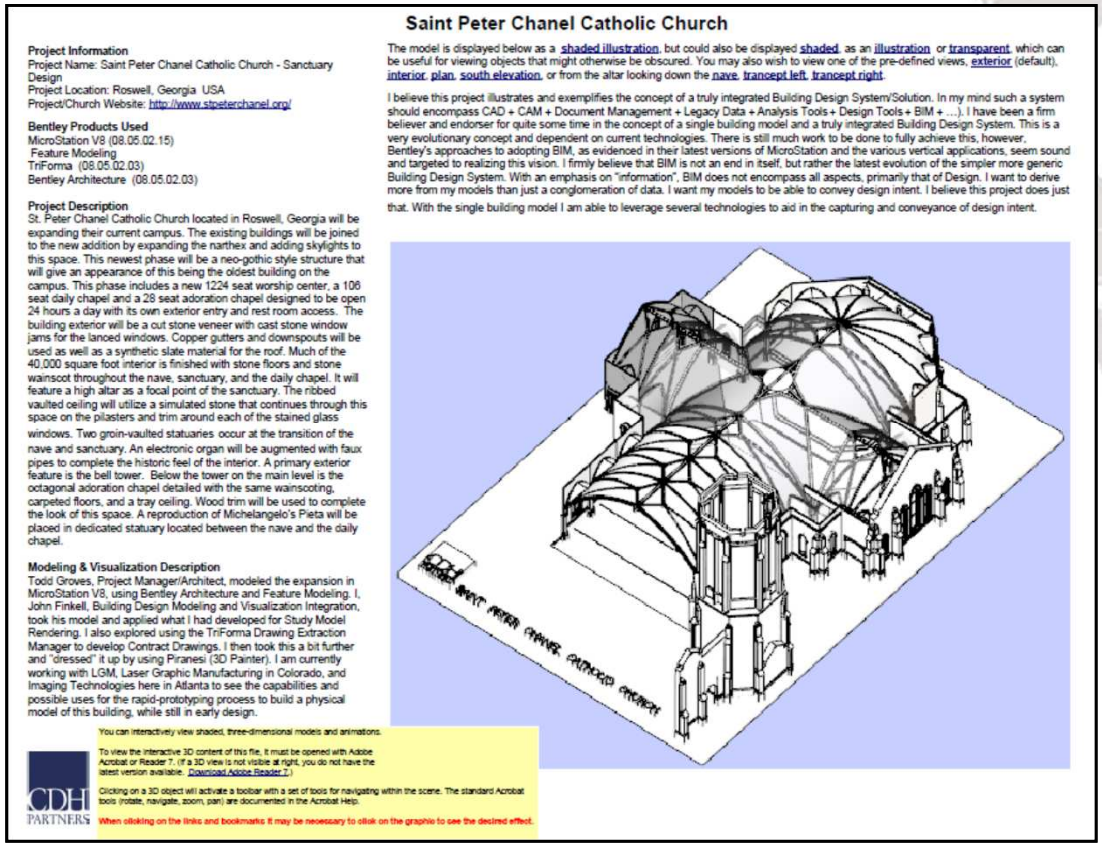
Embedding interactive 3D object in a PDF using MeshLab and U3D

Visual Computing Group - ISTI - CNR
http://vcg.isti.cnr.it

5th January 2008

A little example of an embedded U3D object in a pdf. The object is a simplified version of a 3D scanned mesh of a Ippolita Sforza's statue sculptured by Francesco Laurana. This pdf was produced with open source tools. The object was converted in the pdf-ready U3D format with MeshLab ( http://meshlab.sourceforge.net ) and assembled in a pdf with pdfLaTeX and the movie15 package.

1



**Saint Peter Chanel Catholic Church**

**Project Information**
Project Name: Saint Peter Chanel Catholic Church - Sanctuary Design
Project Location: Roswell, Georgia USA
Project/Church Website: http://www.stpeterchanel.org/

**Bentley Products Used**
MicroStation V8 (08.05.02.15)
Feature Modeling
TriForma (08.05.02.03)
Bentley Architecture (08.05.02.03)

**Project Description**
St. Peter Chanel Catholic Church located in Roswell, Georgia will be expanding their current campus. The existing buildings will be joined to the new addition by expanding the narthex and adding skylights to this space. This newest phase will be a neo-gothic style structure that will give an appearance of this being the oldest building on the campus. This phase includes a new 1224 seat worship center, a 106 seat daily chapel and a 28 seat adoration chapel designed to be open 24 hours a day with its own exterior entry and rest room access. The building exterior will be a cut stone veneer with cast stone window jams for the lanced windows. Copper gutters and downspouts will be used as well as a synthetic slate material for the roof. Much of the 40,000 square foot interior is finished with stone floors and stone wainscot throughout the nave, sanctuary, and the daily chapel. It will feature a high altar as a focal point of the sanctuary. The ribbed vaulted ceiling will utilize a simulated stone that continues through this space on the pilasters and trim around each of the stained glass windows. Two groin-vaulted statuaries occur at the transition of the nave and sanctuary. An electronic organ will be augmented with faux pipes to complete the historic feel of the interior. A primary exterior feature is the bell tower. Below the tower on the main level is the octagonal adoration chapel detailed with the same wainscoting, carpeted floors, and a tray ceiling. Wood trim will be used to complete the look of this space. A reproduction of Michelangelo's Pieta will be placed in dedicated statuary located between the nave and the daily chapel.

**Modeling & Visualization Description**
Todd Groves, Project Manager/Architect, modeled the expansion in MicroStation V8, using Bentley Architecture and Feature Modeling. I, John Finkell, Building Design Modeling and Visualization Integration, took his model and applied what I had developed for Study Model Rendering. I also explored using the TriForma Drawing Extraction Manager to develop Contract Drawings. I then took this a bit further and "dressed" it up by using Piranesi (3D Painter). I am currently working with LGM, Laser Graphic Manufacturing in Colorado, and Imaging Technologies here in Atlanta to see the capabilities and possible uses for the rapid-prototyping process to build a physical model of this building, while still in early design.

The model is displayed below as a shaded illustration, but could also be displayed shaded, as an illustration or transparent, which can be useful for viewing objects that might otherwise be obscured. You may also wish to view one of the pre-defined views, exterior (default), interior, plan, south elevation, or from the altar looking down the nave, transept left, transept right.

I believe this project illustrates and exemplifies the concept of a truly integrated Building Design System/Solution. In my mind such a system should encompass CAD + CAM + Document Management + Legacy Data + Analysis Tools + Design Tools + BIM + ...). I have been a firm believer and endorser for quite some time in the concept of a single building model and a truly integrated Building Design System. This is a very evolutionary concept and dependent on current technologies. There is still much work to be done to fully achieve this, however, Bentley's approaches to adopting BIM, as evidenced in their latest versions of MicroStation and the various vertical applications, seem sound and targeted to realizing this vision. I firmly believe that BIM is not an end in itself, but rather the latest evolution of the simpler more generic Building Design System. With an emphasis on "information", BIM does not encompass all aspects, primarily that of Design. I want to derive more from my models than just a conglomeration of data. I want my models to be able to convey design intent. I believe this project does just that. With the single building model I am able to leverage several technologies to aid in the capturing and conveyance of design intent.

CDH PARTNERS

You can interactively view shaded, three-dimensional models and animations.

To view the interactive 3D content of this file, it must be opened with Adobe Acrobat or Reader 7. (If a 3D view is not visible at right, you do not have the latest version available. Download Adobe Reader 7.)

Clicking on a 3D object will activate a toolbar with a set of tools for navigating within the scene. The standard Acrobat tools (rotate, navigate, zoom, pan) are documented in the Acrobat Help.

When clicking on the links and bookmarks it may be necessary to click on the graphic to see the desired effect.

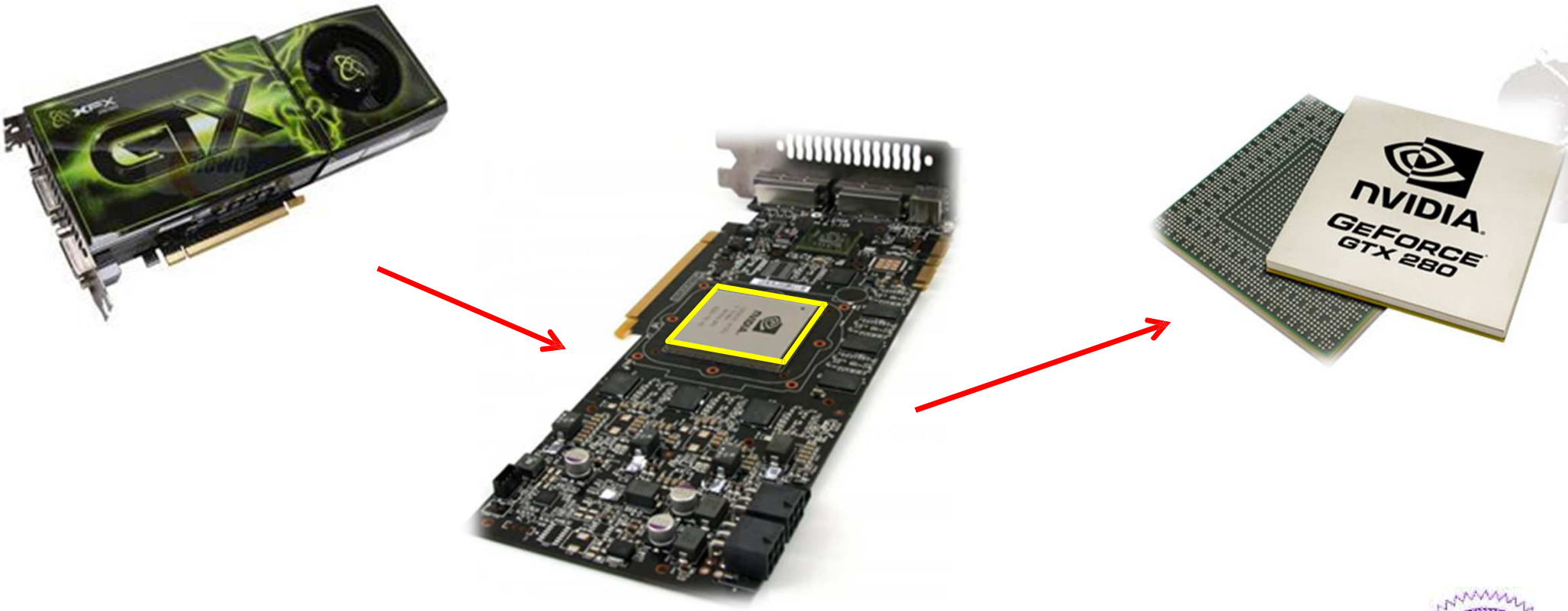
# Graphics Hardware System

***Graphics Processing Unit***

***Graphics System***

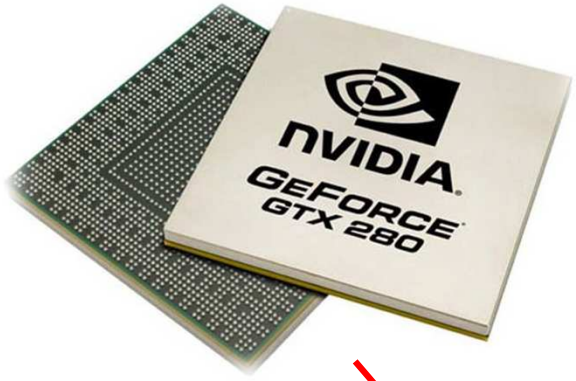
# GPU – Graphics Processing Unit

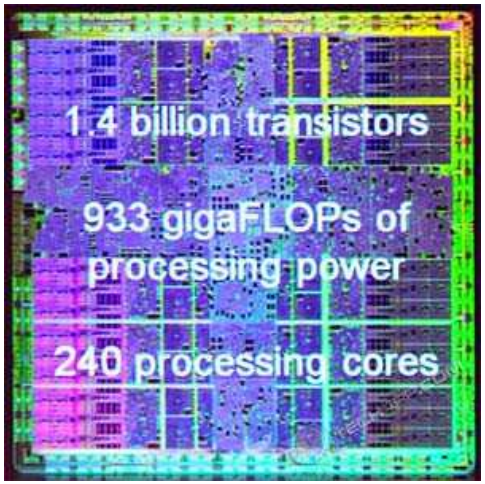
- **Acceleration of Graphics Processing**

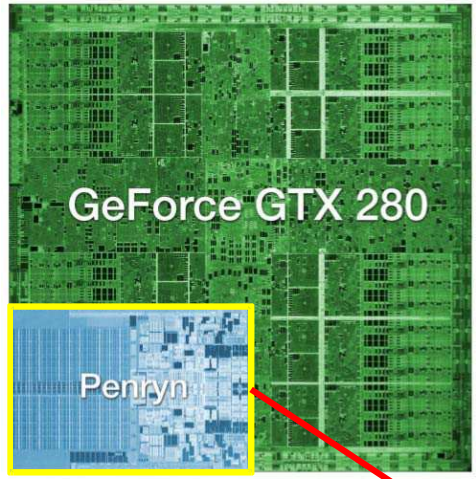


# GPU – Graphics Processing Unit

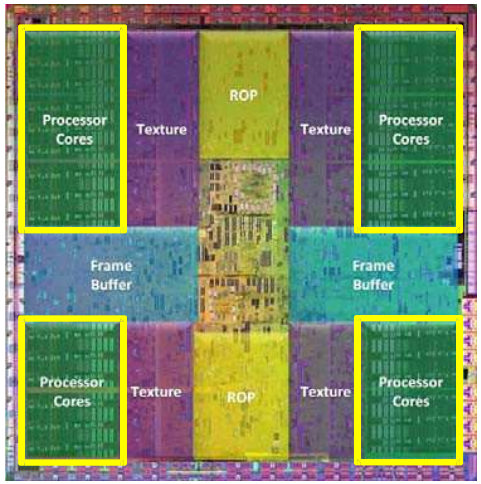
- Very high complexity with massively parallelism

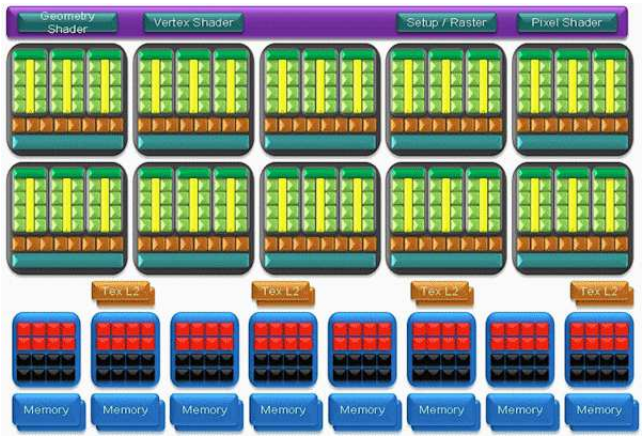


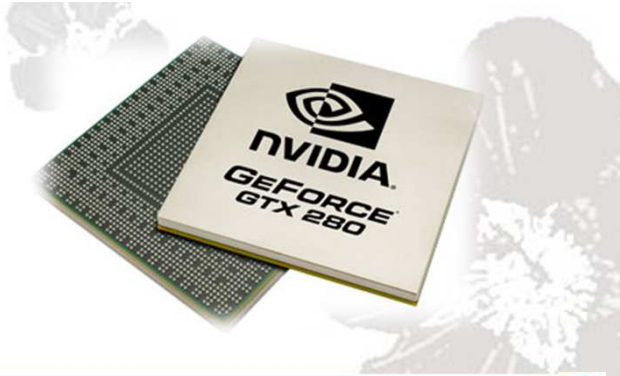

Intel Core 2 Duo



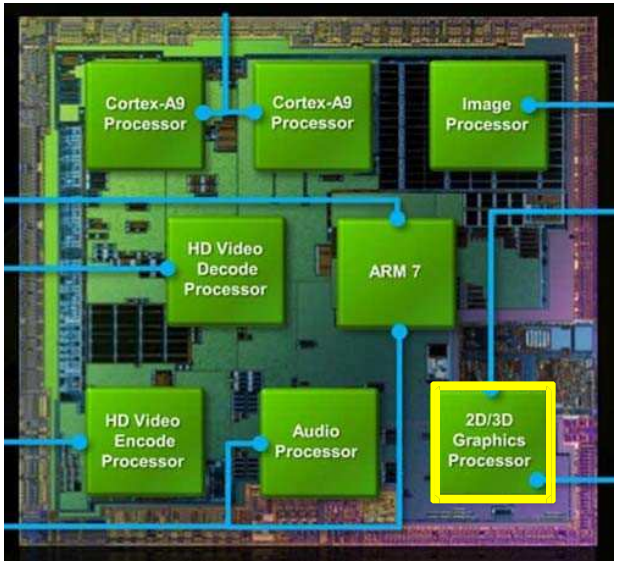




Processor Cores (Shaders)

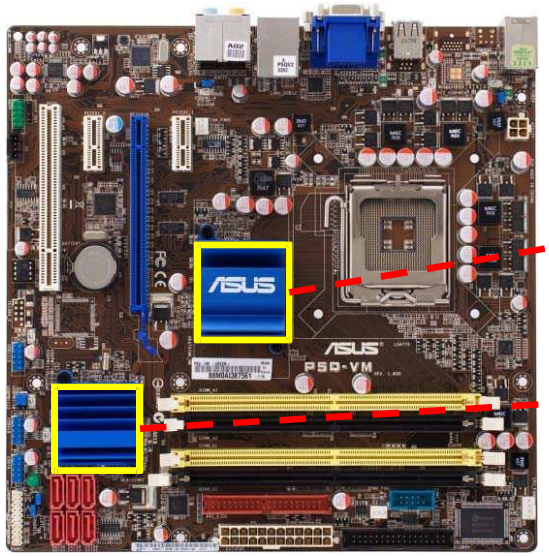
# Where is GPU

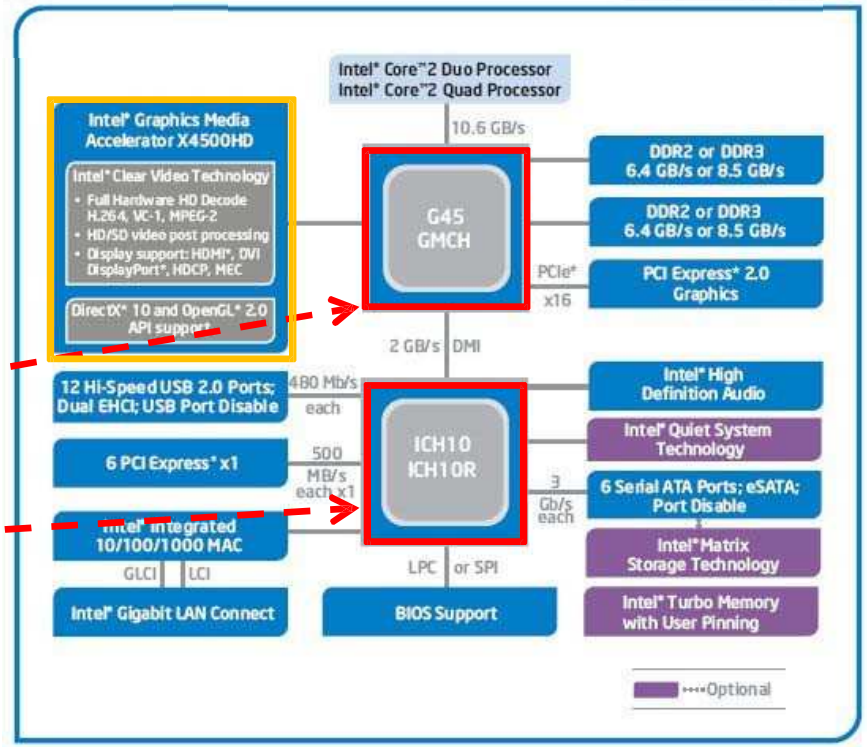



- Graphics Cards
- Chipset
- CPU
- SoC

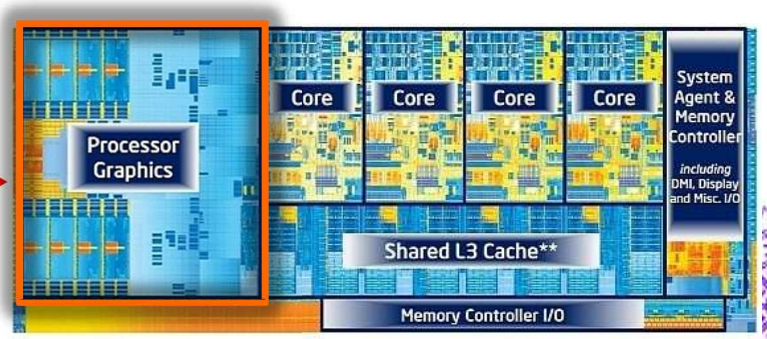



NVIDIA Tegra 2







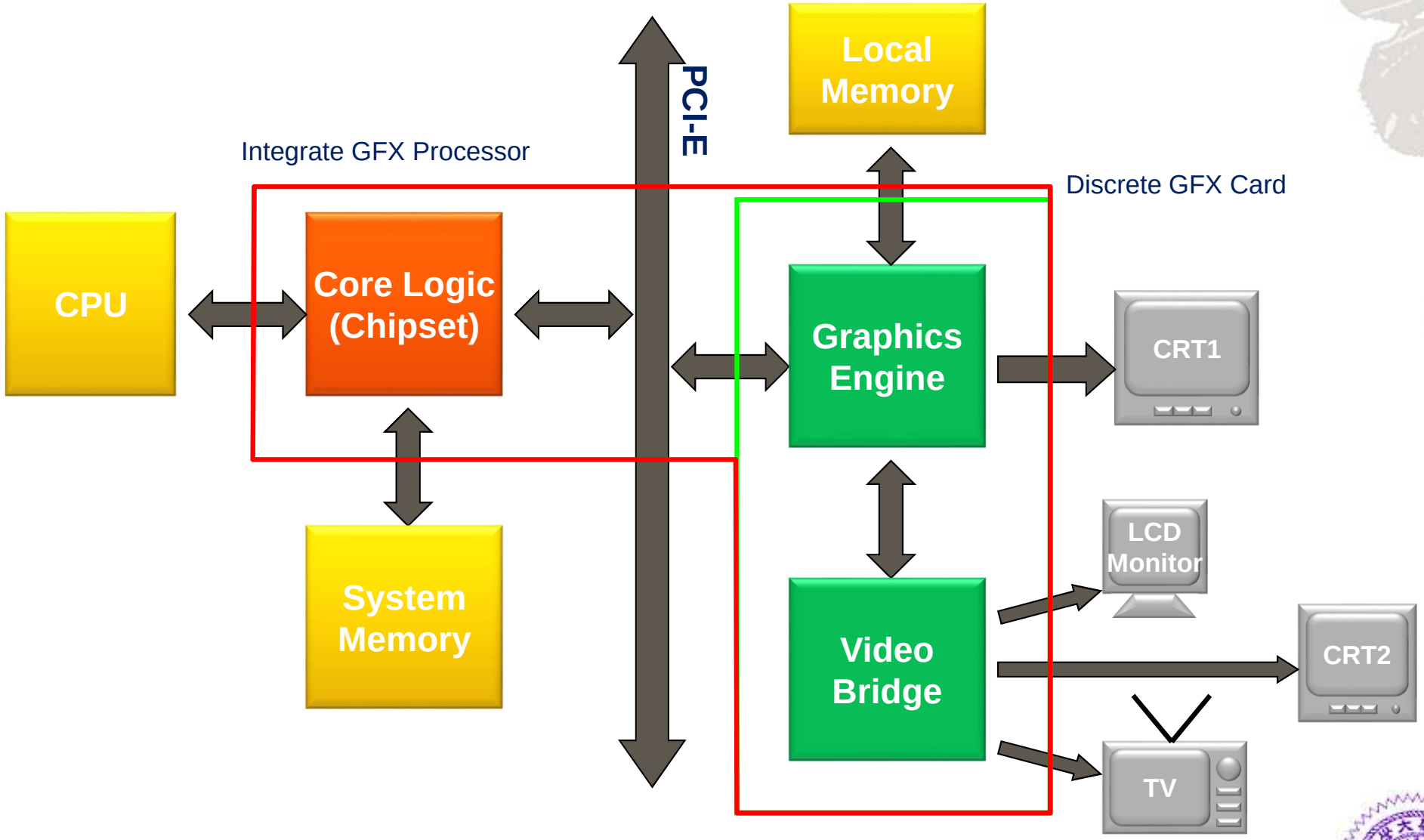
# GPU Everywhere


- Nowadays, almost every device, that supports output to a display, equips with a GPU
  - Desktop PCs
  - Notebooks
  - Mobile phones
  - Tablets
  - Smart TVs
  - Set-top boxes
  - Game consoles
  - Wearable devices
  - …

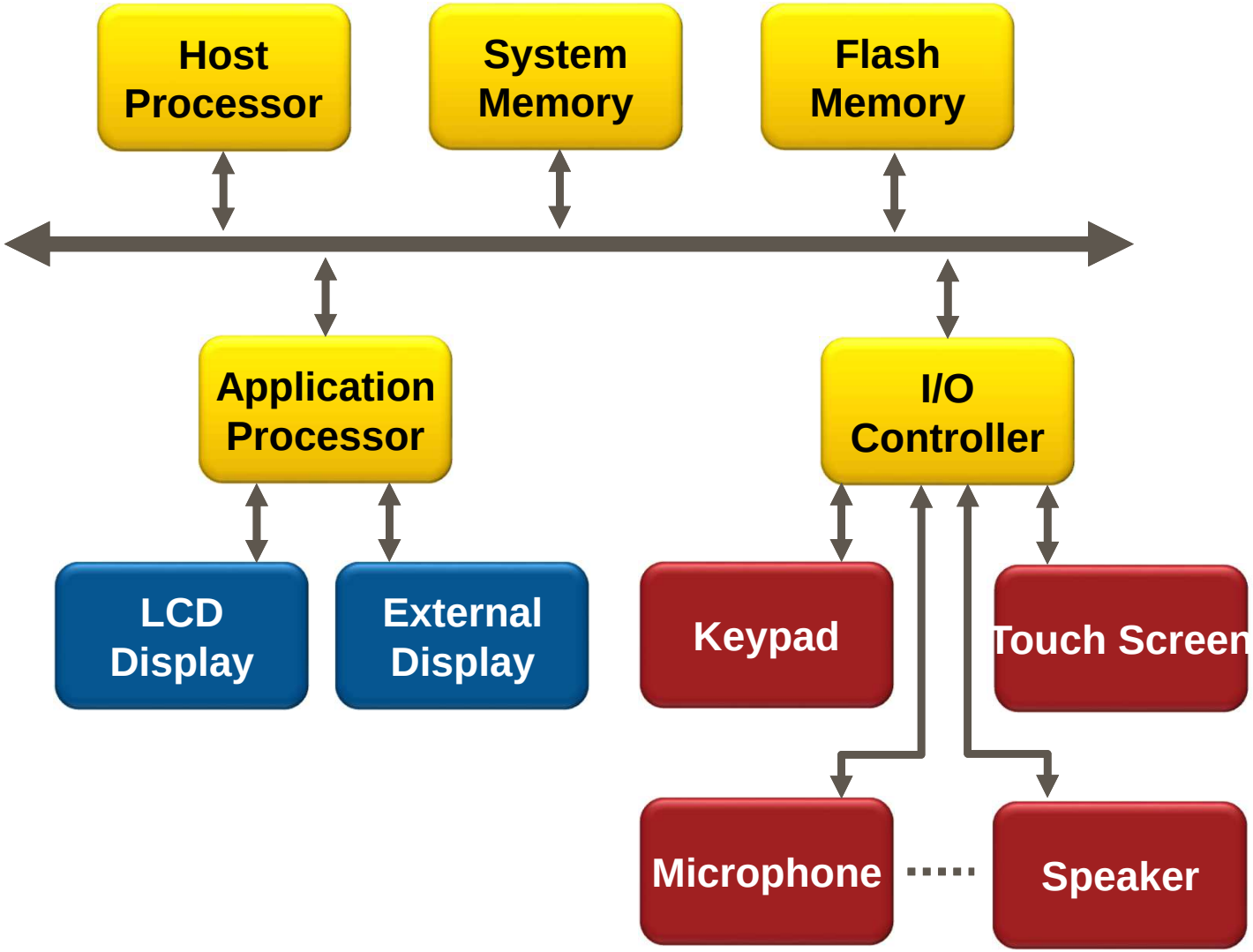




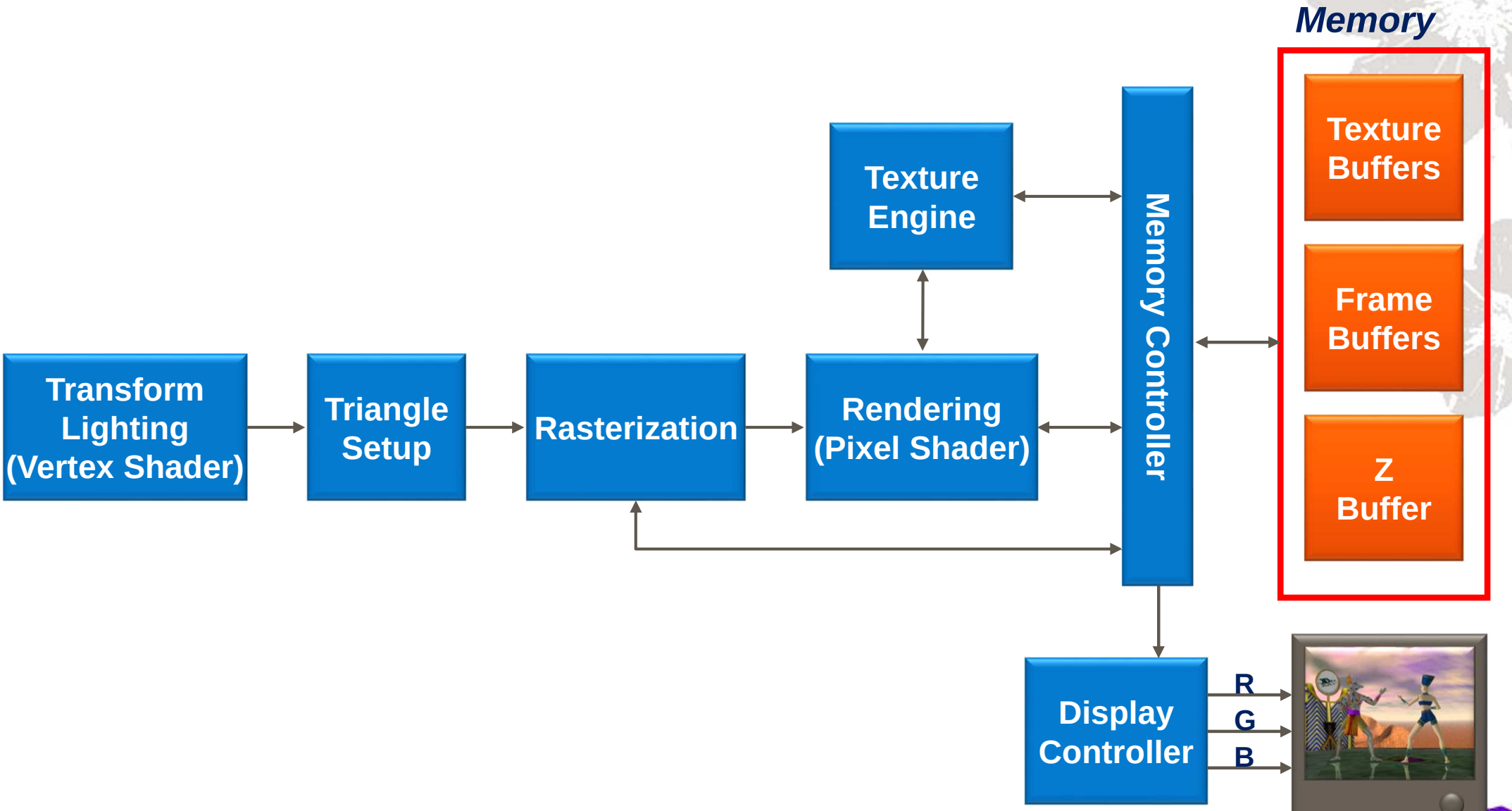
# PC Graphics System



# Mobile Graphics System

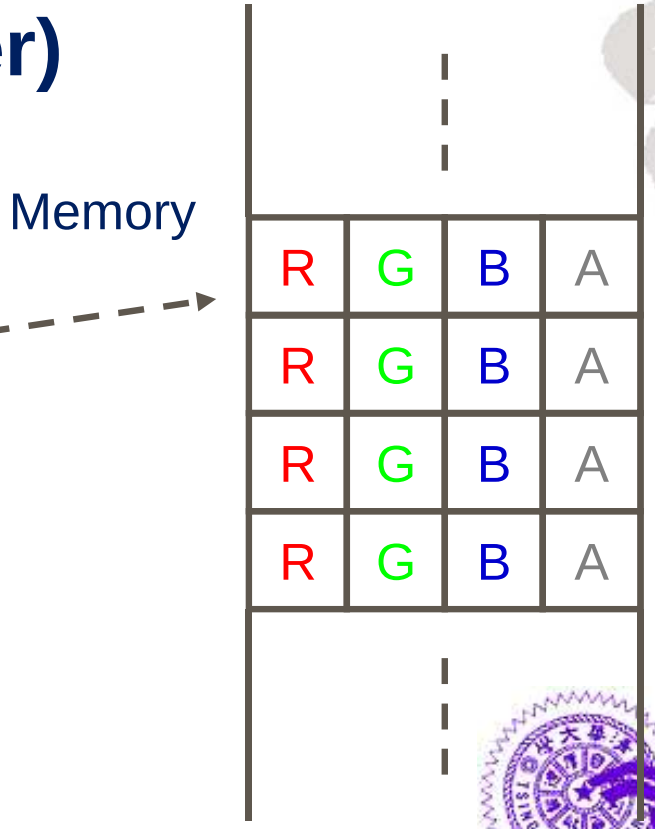


# A Generalized Graphics Sub-System



# Frame Buffer

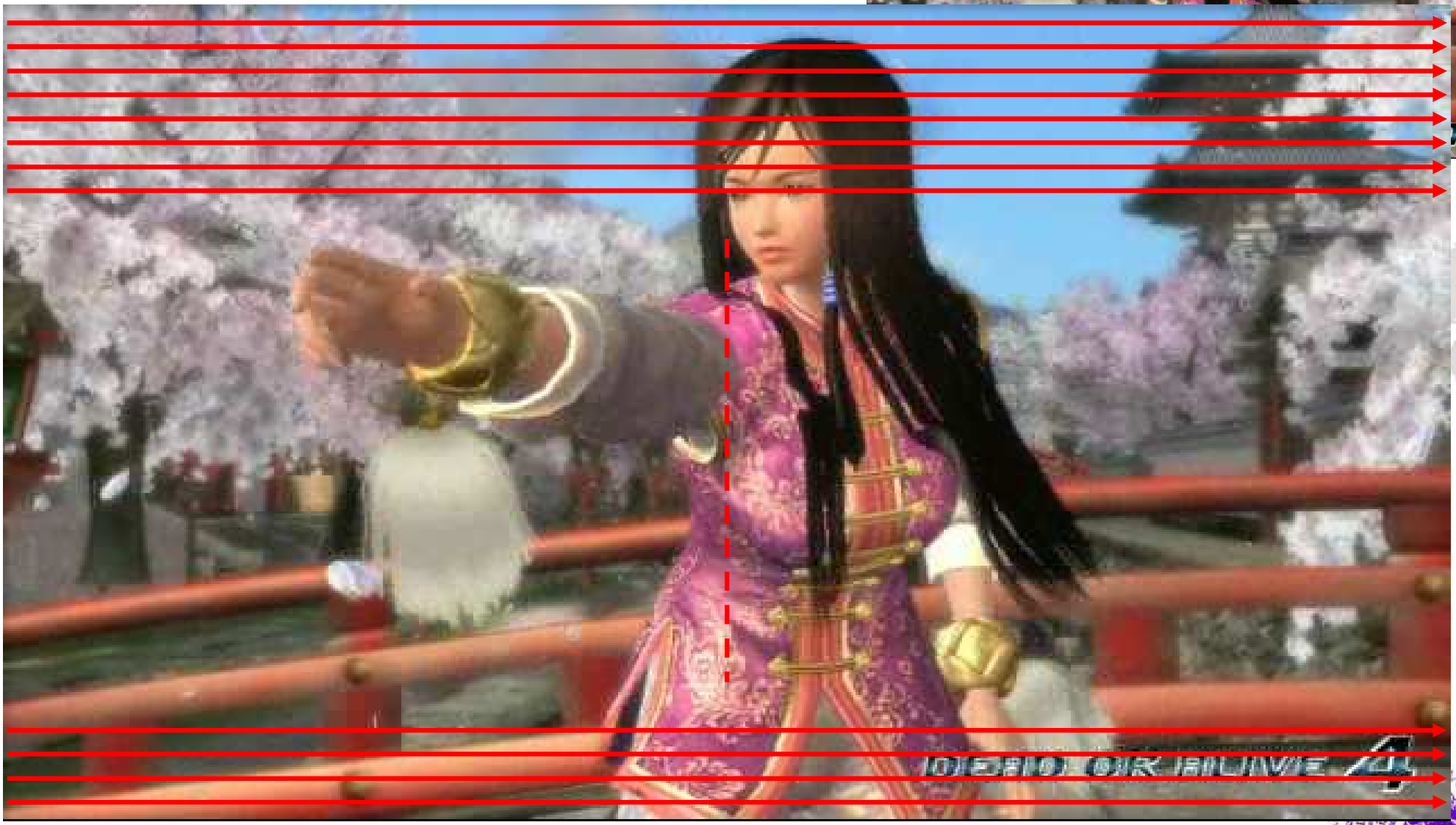
- Memory location for storing display data during processing
  - Front buffer (Current display buffer)
  - Back buffer (Next display buffer)



# Pixel

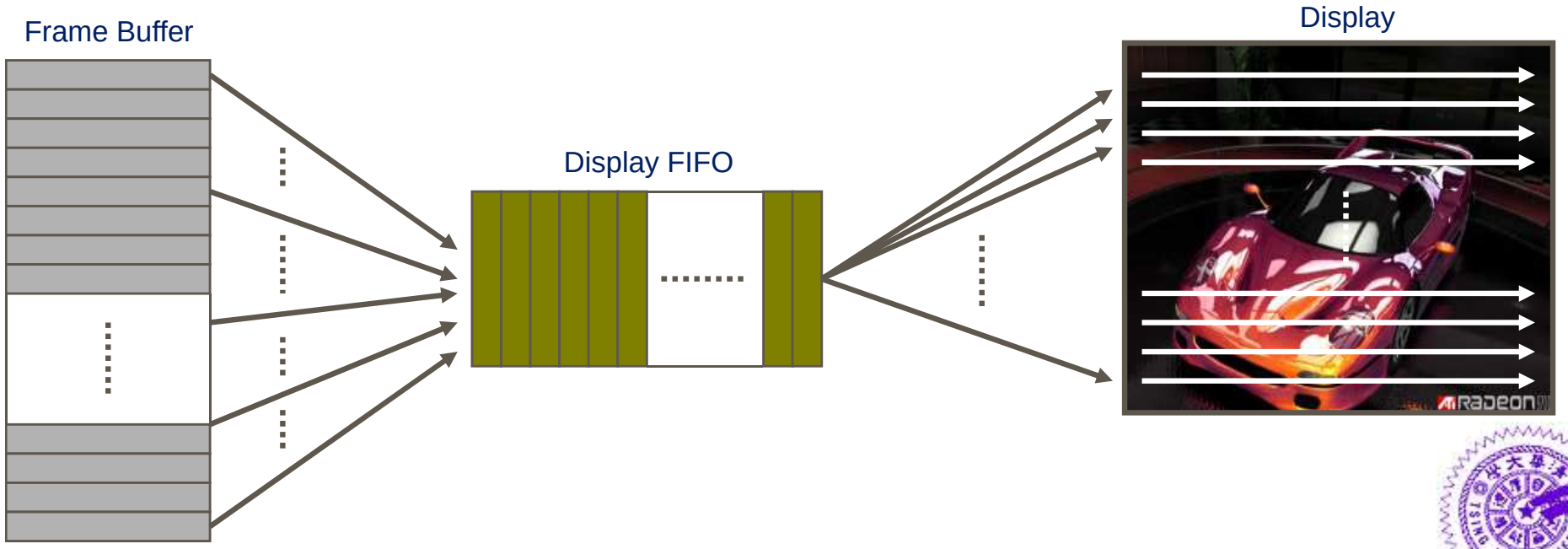
- Each pixel is consisting of red, green, and blue color components

# Display Buffer



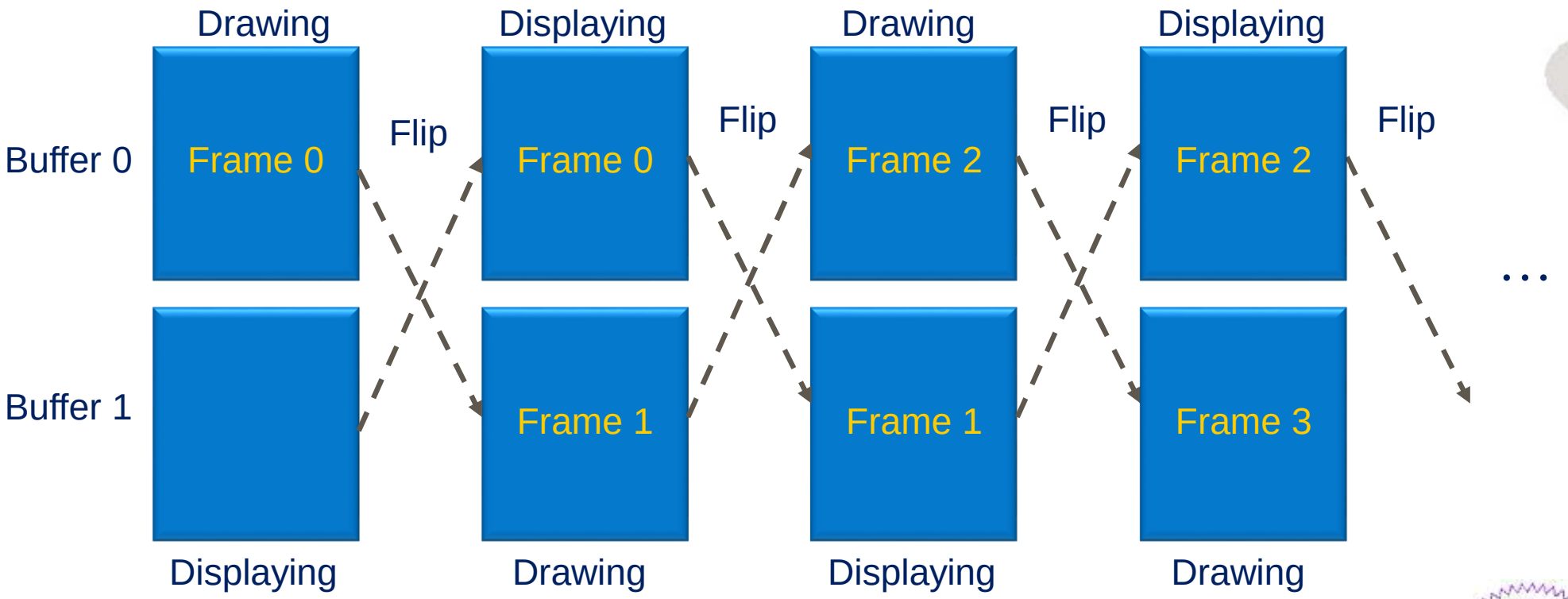
# Screen Refresh

- Retrieve front buffer pixels for displaying pixel colors on screen
- Constant refresh rate
- Scan-line based display



# Double Buffering

- A technique to prevent from displaying incomplete frame



# Full Screen Display

- **Full Screen Display**
  - **Uses double buffering or triple buffering**
  - **Flip display by changing display base address to corresponding buffer**

Display →



# Full Screen Display

- **Full Screen Display**
  - **Uses double buffering or triple buffering**
  - **Flip display by changing display base address to corresponding buffer**


Display →

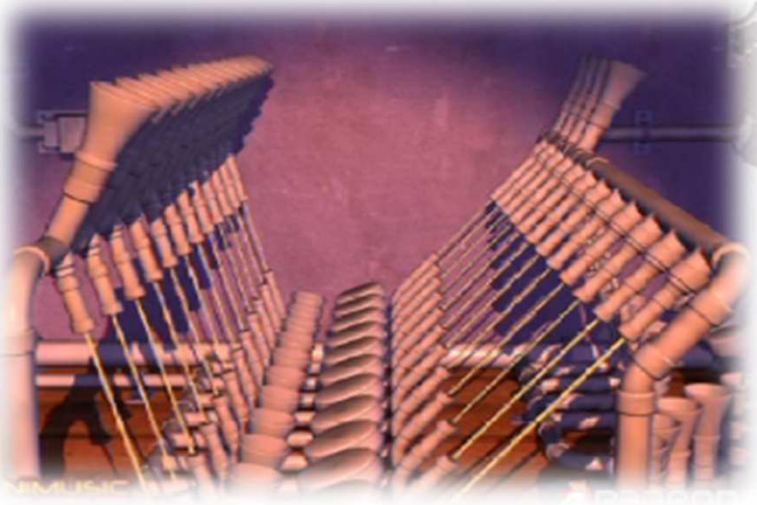

# Window Mode Display

- **Window Mode Display**
  - **Uses double buffering or triple buffering**
  - **Flip display by using 2D Bitblt to move back buffer frame on to front buffer display**

# Window Mode Display

- **Window Mode Display**
  - **Uses double buffering or triple buffering**
  - **Flip display by using 2D Bitblt to move back buffer frame on to front buffer display**

# Hardware Capabilities



# Benchmarks

- Set the Level of Quality and Performance
  - PC Graphics Benchmarks

# Benchmarks

- Set the Level of Quality and Performance
  - Mobile Graphics Benchmarks

# *Heart of the GPU – A 3D Graphics Pipeline*

***What is the 3D Graphics Pipeline***
***Inside a 3D Graphics Pipeline***
***Programmable Shaders***

# What is the 3D Graphics Pipeline

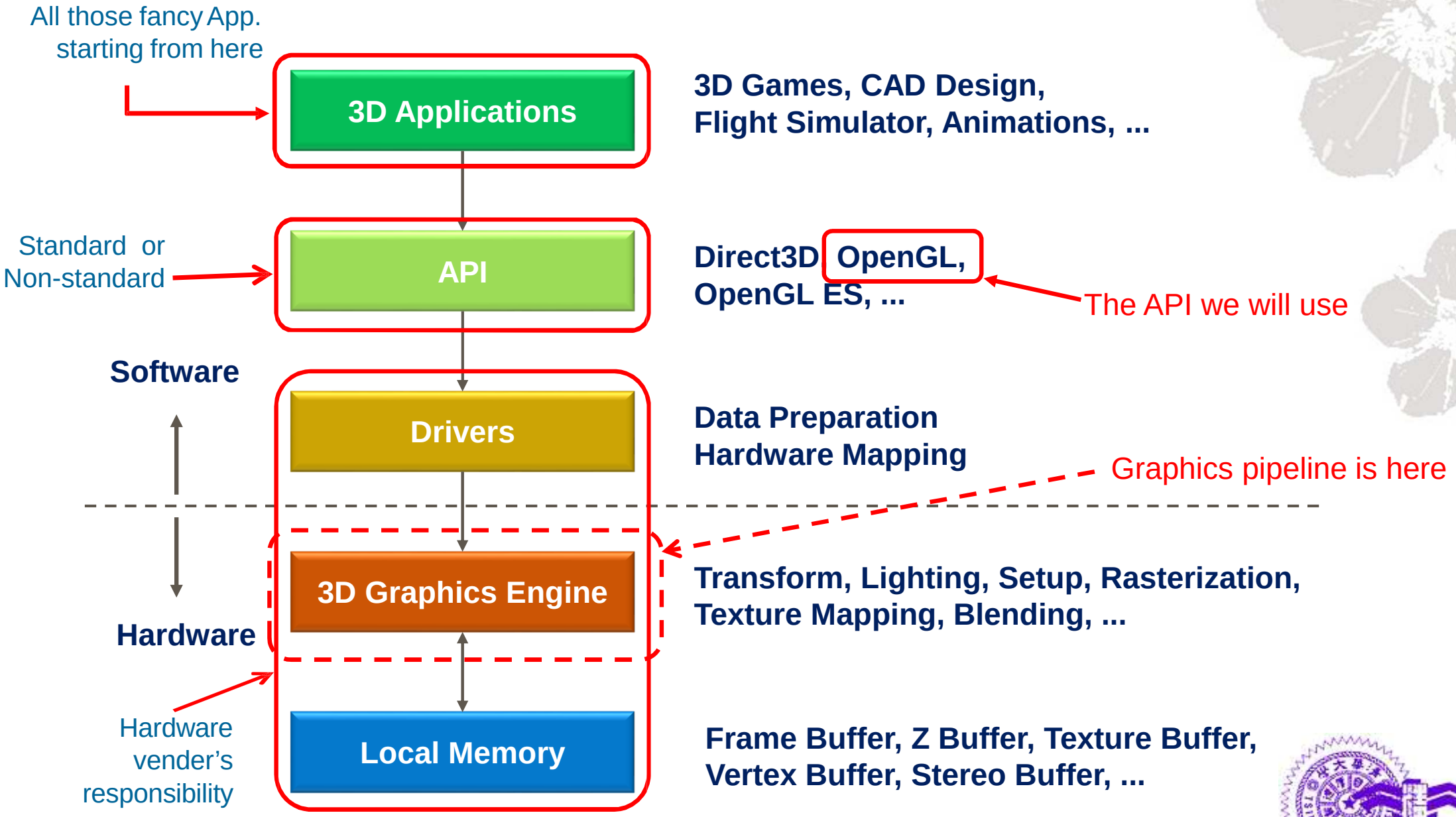
- The one who renders the frames (images) you see on the display
  - Almost everything you see on the display is processed by the graphics pipeline
  - It is realized by graphics hardware (GPU) or by software (CPU)

Windows UI

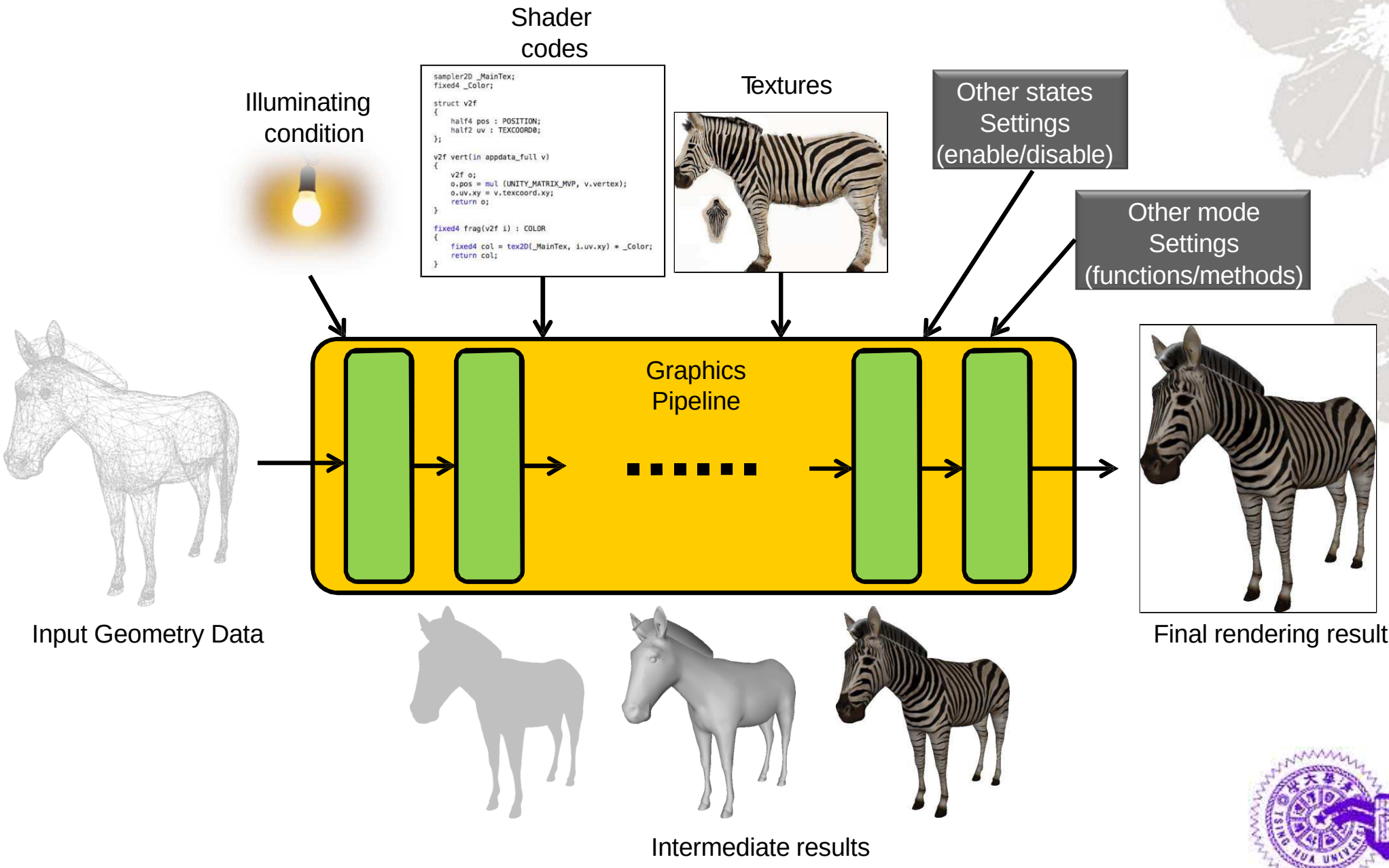

Video Games


Photo Browser

# Graphics Rendering Process

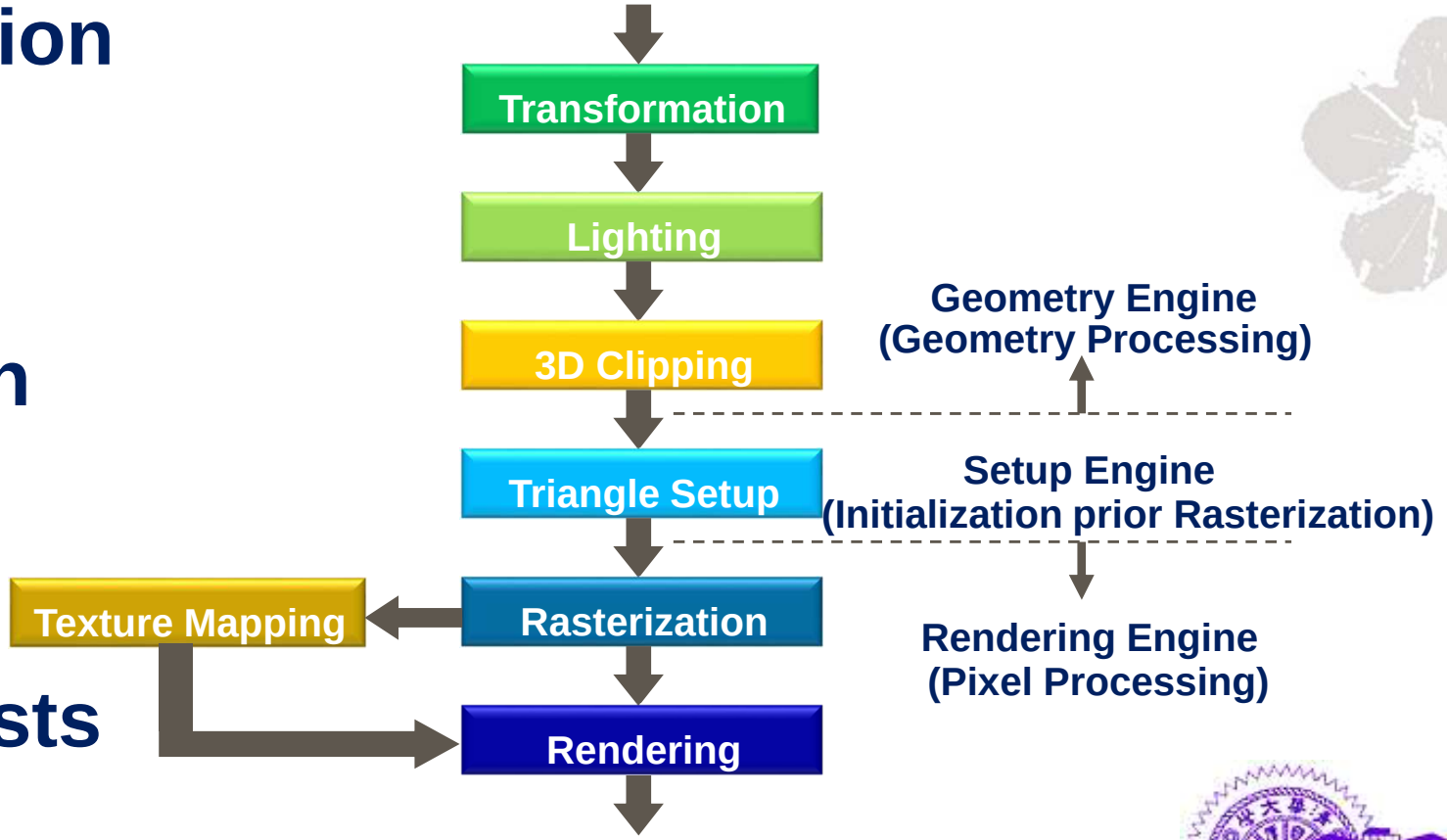


# Basic 3D Graphics Programming



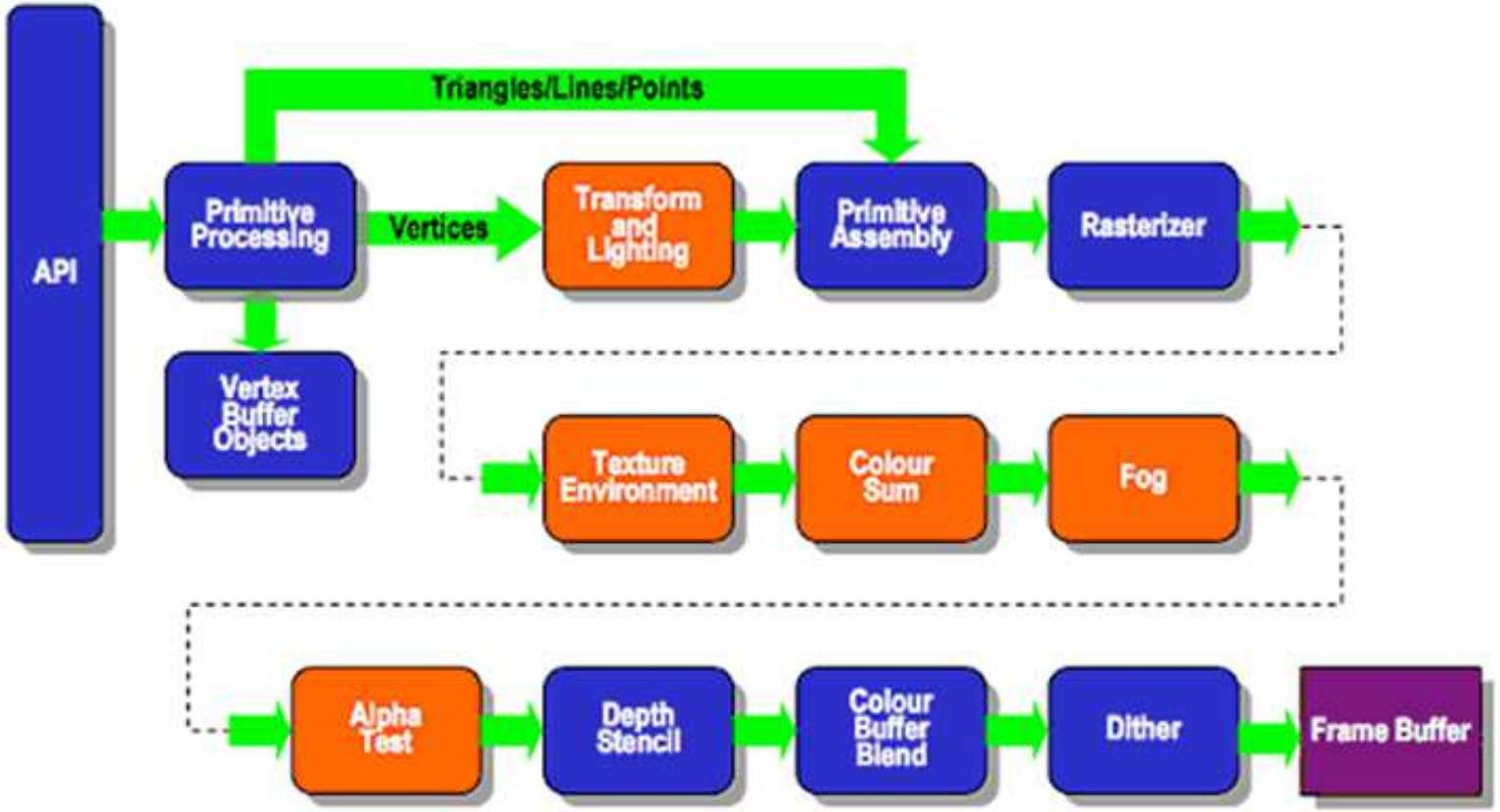
# Inside a 3D Graphics Pipeline

- A pipeline consisting of several modules to accomplish the 3D rendering tasks
  - Transformation
  - Lighting
  - Setup
  - Rasterization
  - Texturing
  - Blending
  - Fragment tests
  - Shaders



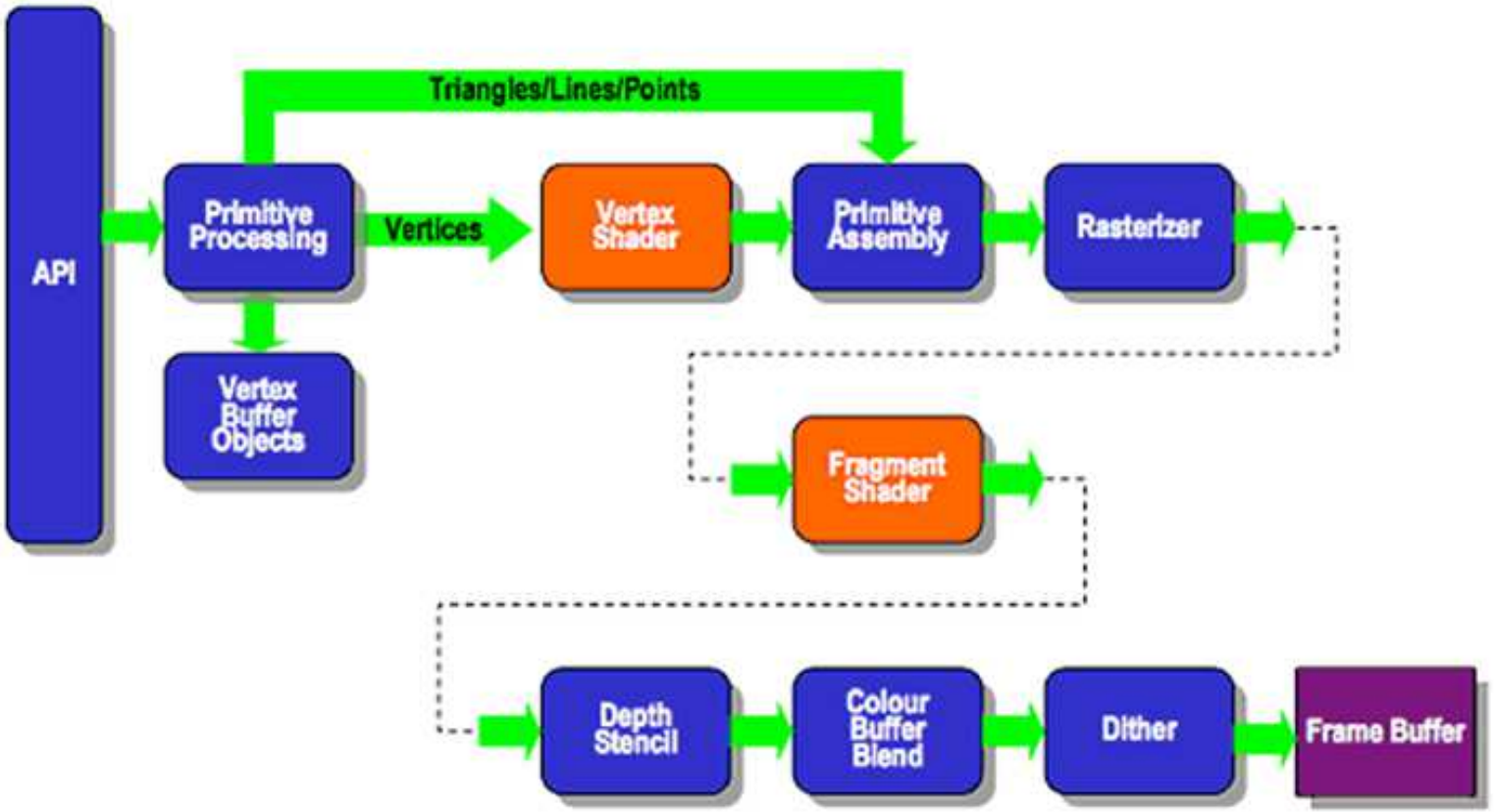
# Example of a 3D Graphics Pipeline

- OpenGL ES v1.1 (fixed function pipeline)

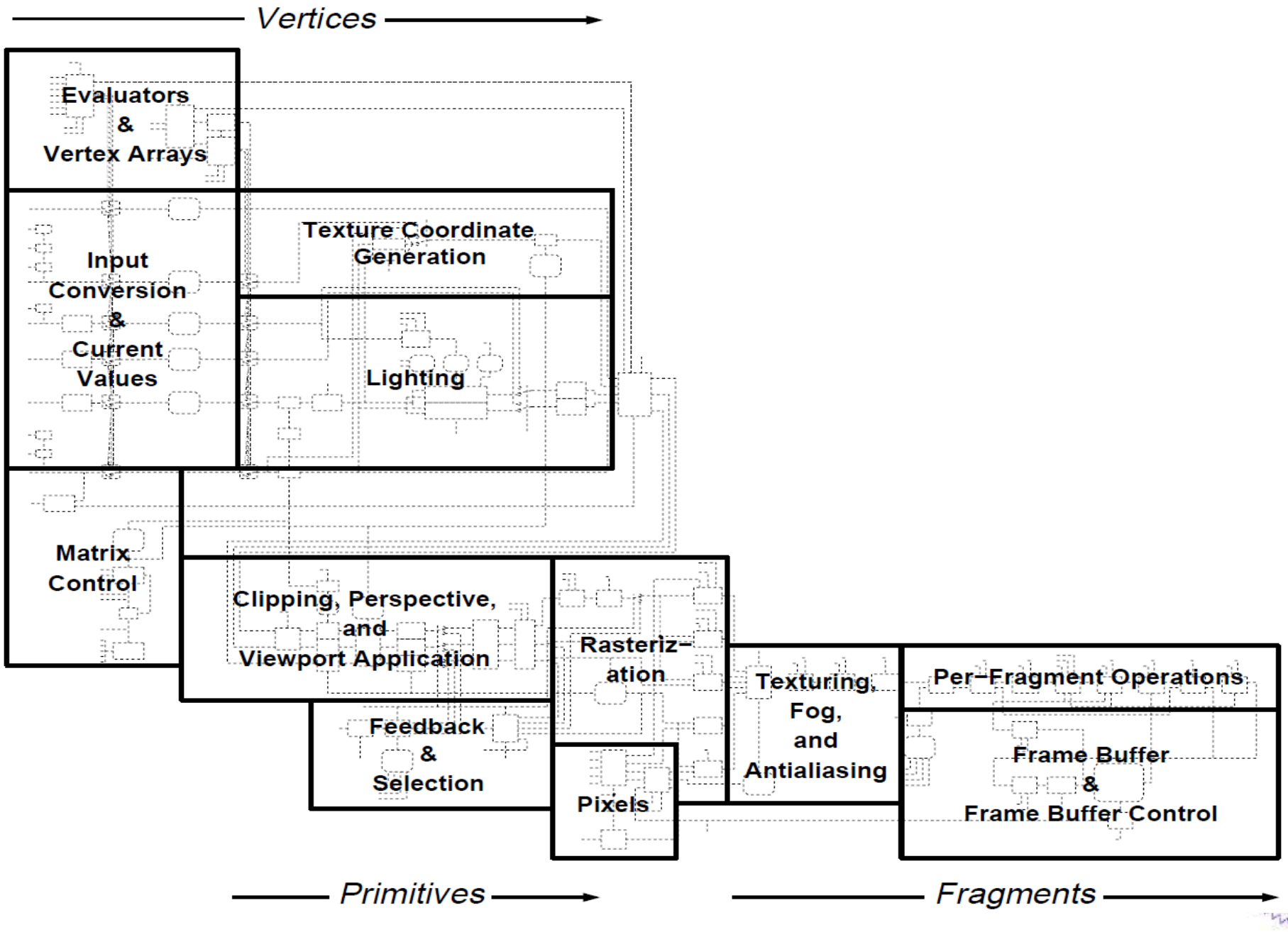


# Example of a 3D Graphics Pipeline

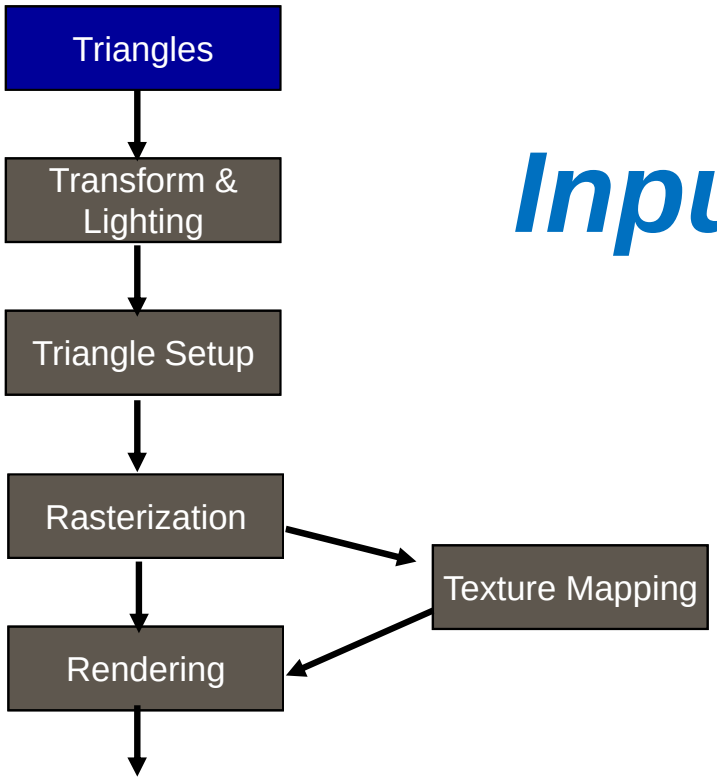
- OpenGL ES v2.0 (programmable pipeline)



# OpenGL 3D Graphics Pipeline (v1.1)



# 3D Graphics Pipeline



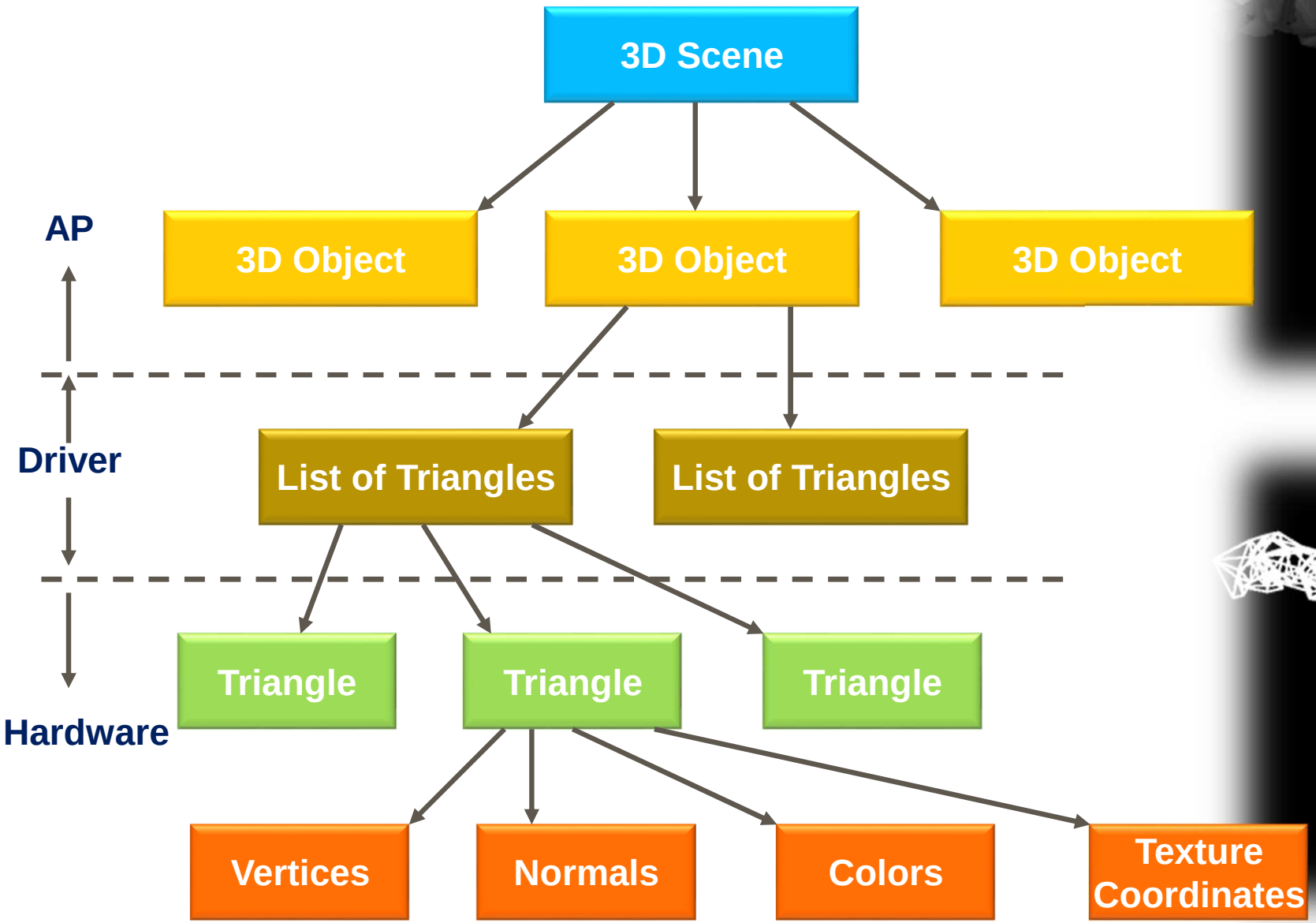

Conventional 3D Graphics Pipeline

## Input Triangles

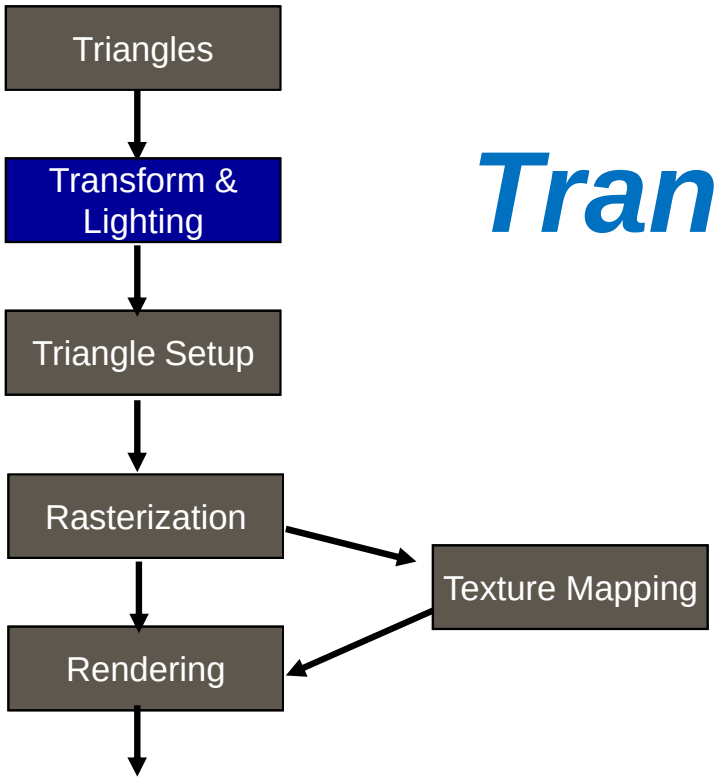
***Vertices***

***Attributes***

# Modeling


3D Scene
AP
3D Object
3D Object
3D Object
Driver
List of Triangles
List of Triangles
Triangle
Triangle
Triangle
Hardware
Vertices
Normals
Colors
Texture Coordinates

# 3D Graphics Pipeline




Conventional 3D Graphics Pipeline

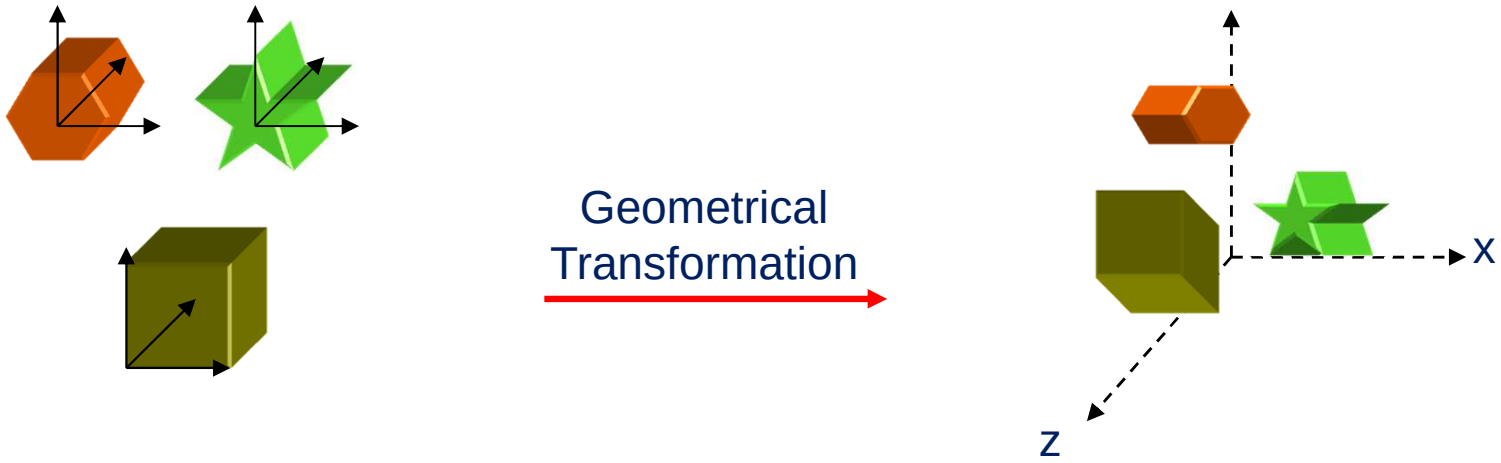
## Transformations

***Geometrical transformations***

***Viewing transformations***

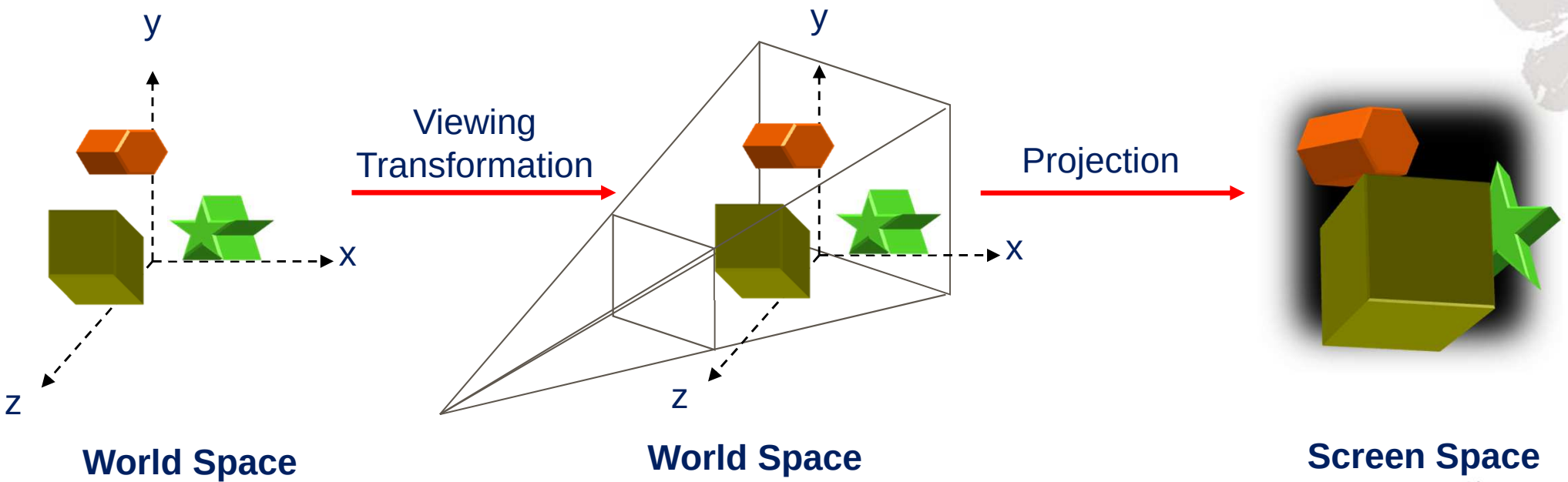
# Transformation

- **Geometrical Transformation**
  - **From Object Space to World Space**

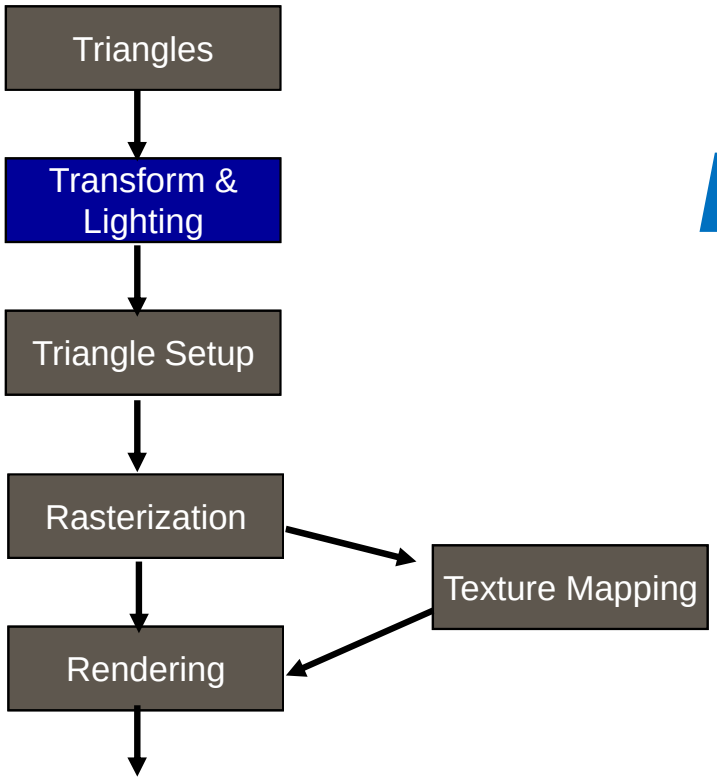


# Transformation

- **Viewing Transformation**
  - **From World Space to Screen Space**



# 3D Graphics Pipeline




Conventional 3D Graphics Pipeline

## Lighting

***Illumination model***

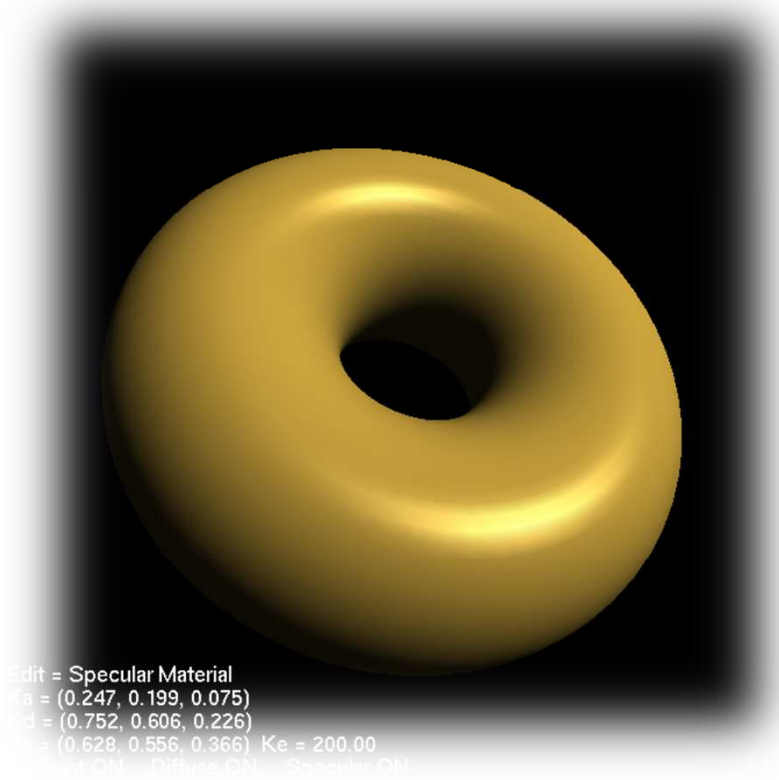
# *Lighting*

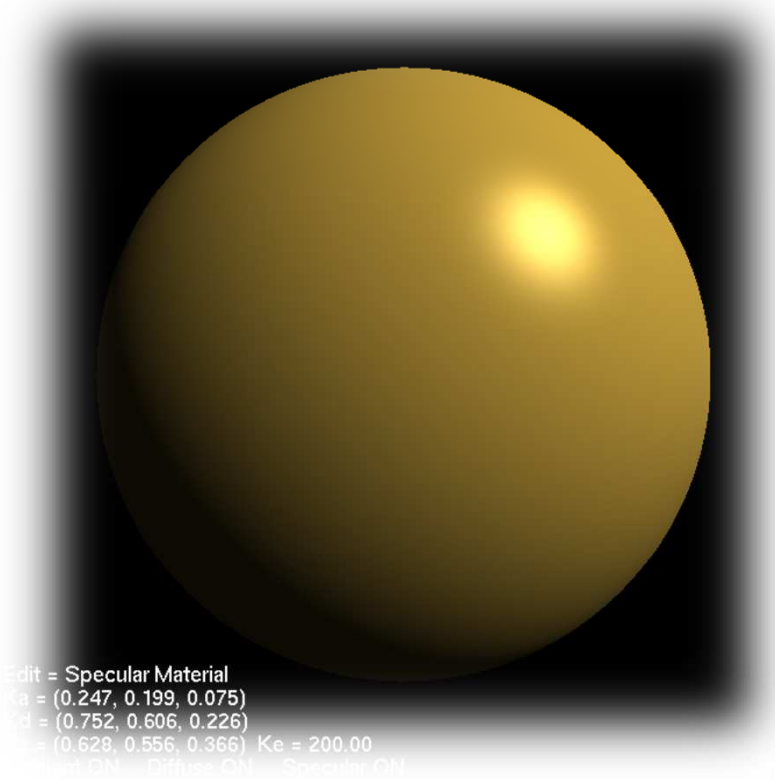
- **Simulate the Effect of Light-Object Interaction**

by Gilles Tran

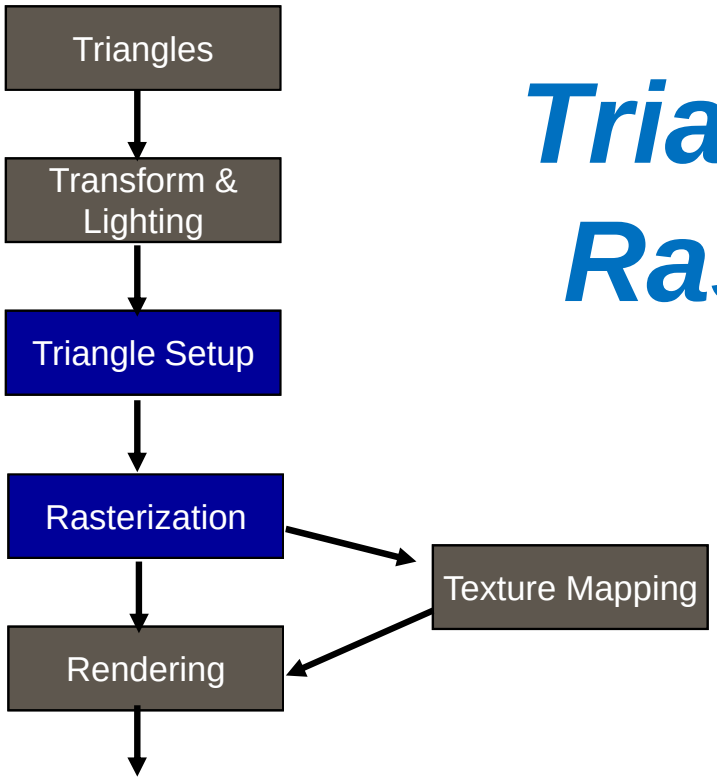
# Lighting

- Ambient + Diffuse + Specular Reflection



# 3D Graphics Pipeline




Conventional 3D Graphics Pipeline

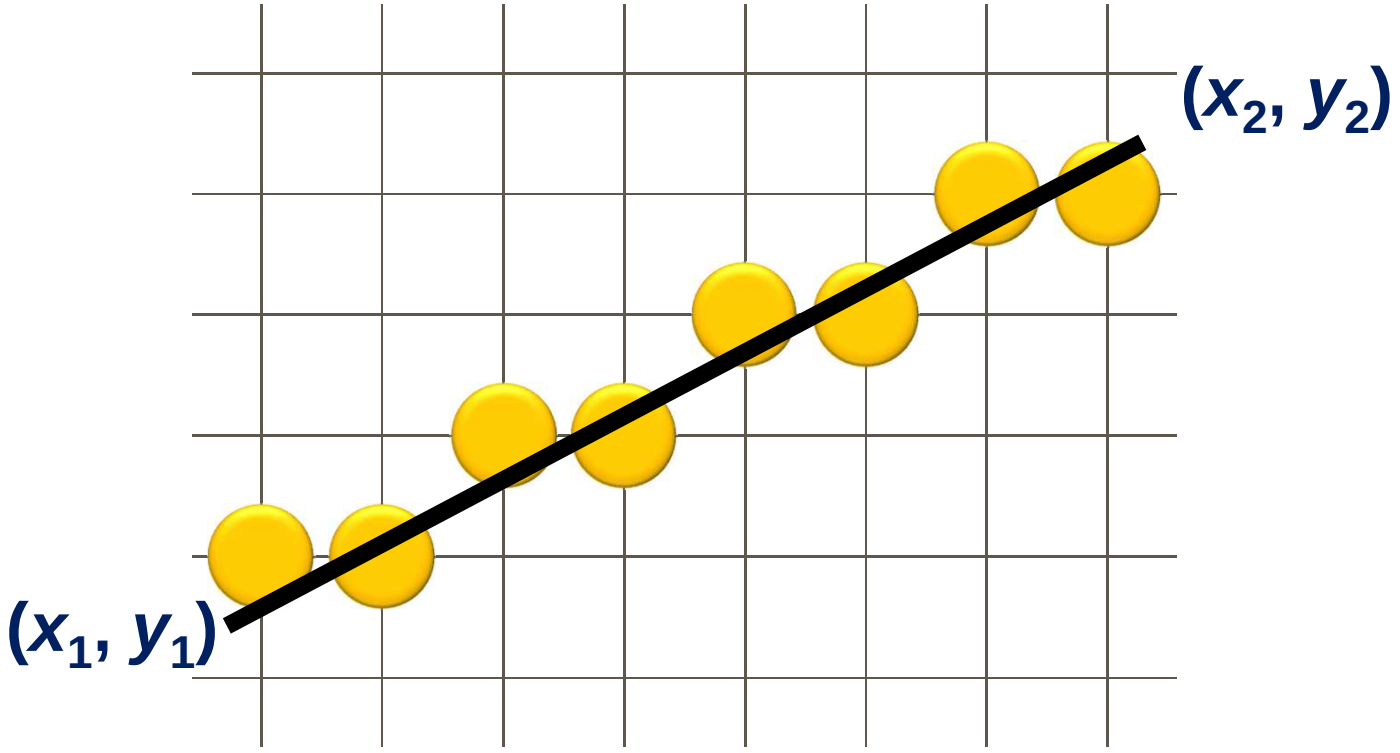
## Triangle Setup and Rasterization

***Initial setup***

***Rasterization***

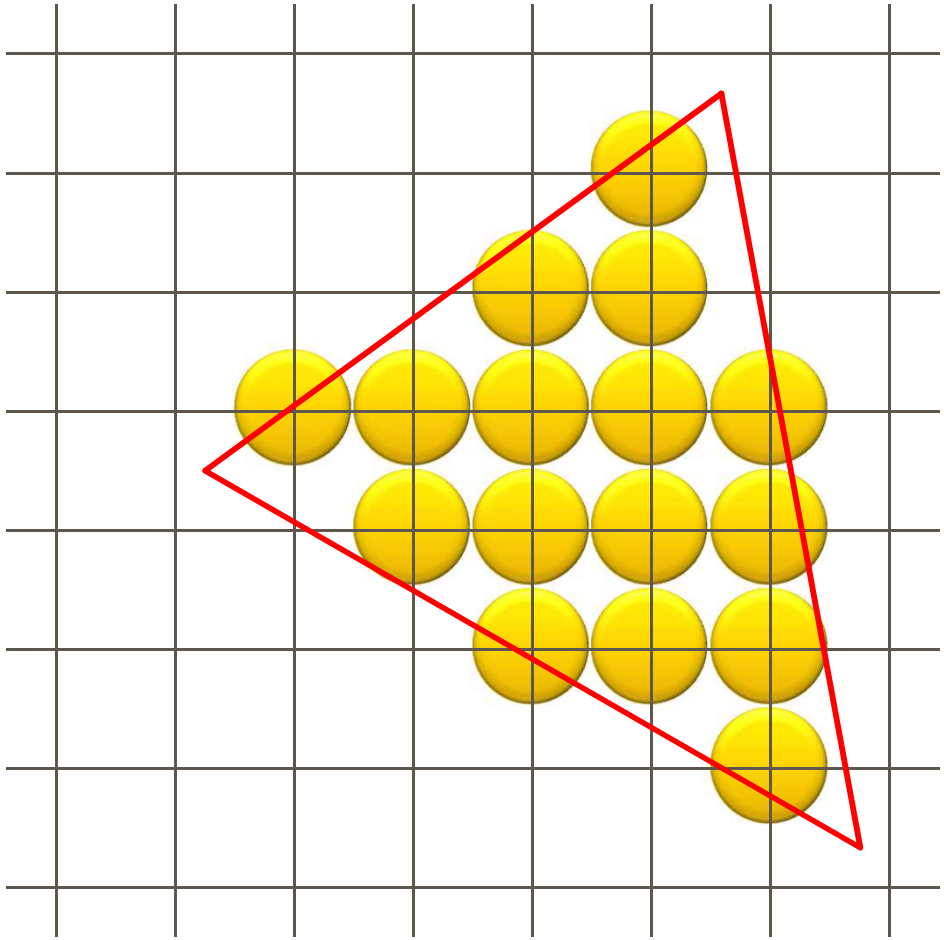
# Rasterization

- Line Rasterization: draw from $(x_1, y_1)$ to $(x_2, y_2)$

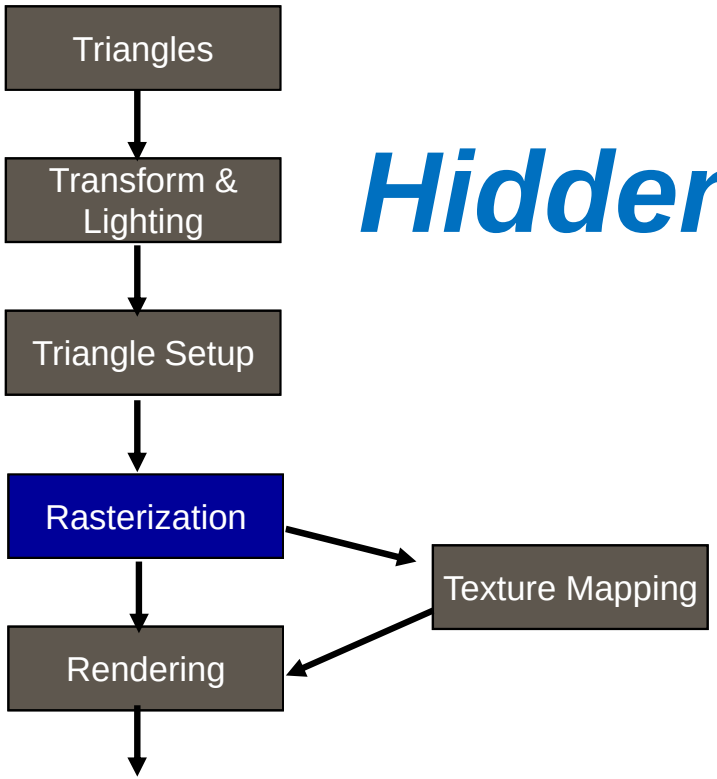


# Rasterization

- Generate Interior Pixels of a Triangle



# 3D Graphics Pipeline

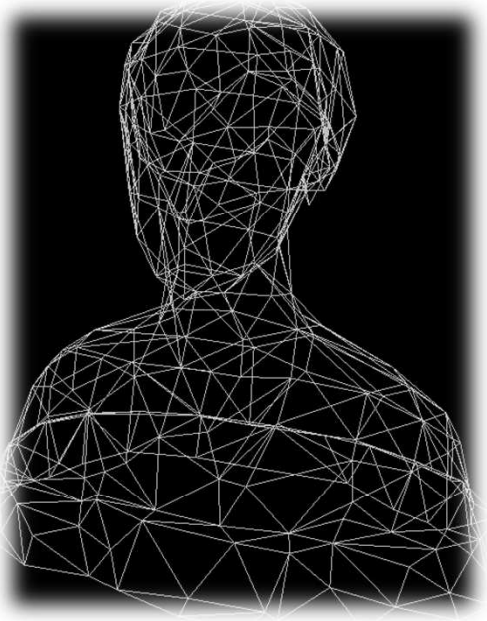


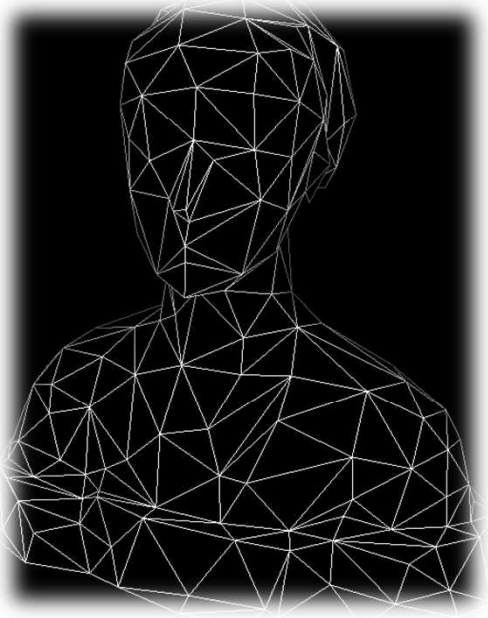

Conventional 3D Graphics Pipeline

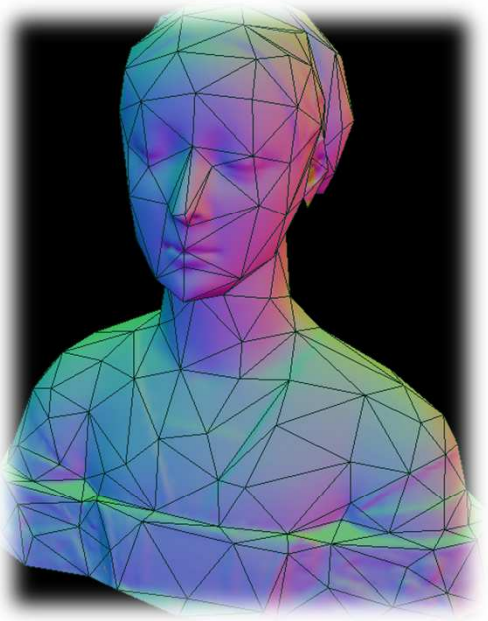
## Hidden Surface Removal

***Remove hidden pixels / primitives***

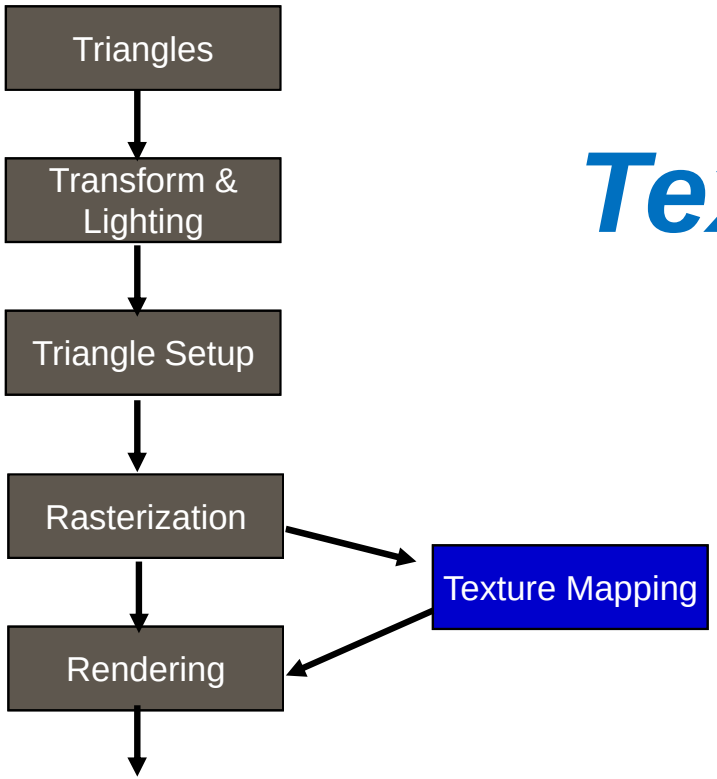
# Hidden Surface Removal







# 3D Graphics Pipeline



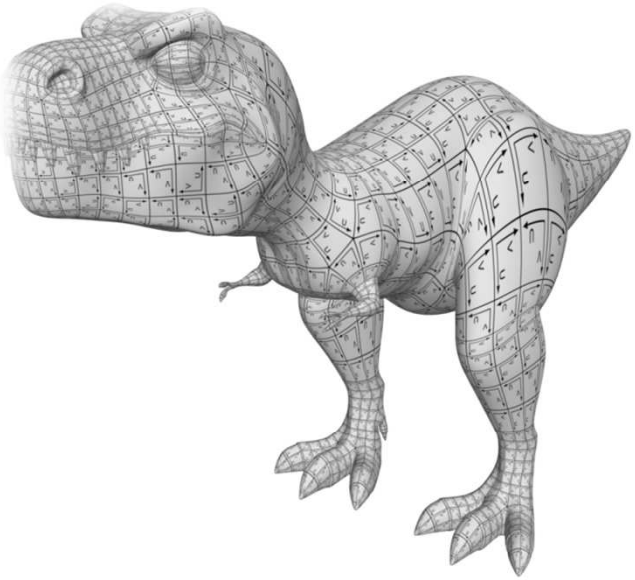

Conventional 3D Graphics Pipeline

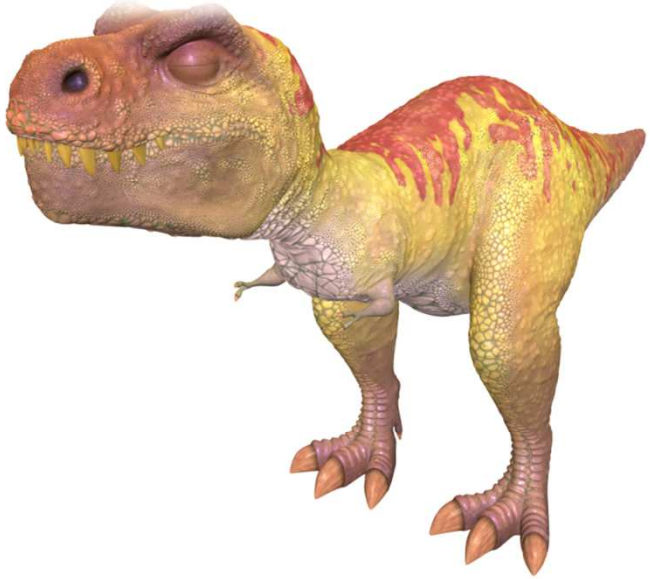
# Texture Mapping

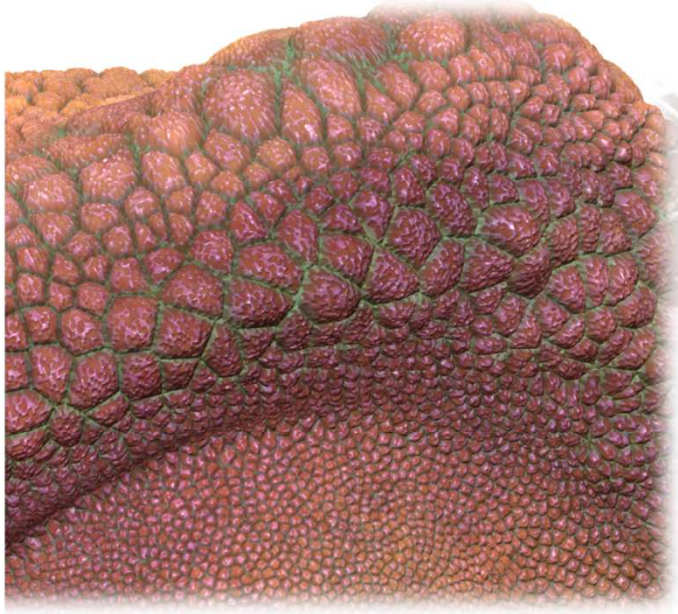
***Add some detail textures***

# Texture Mapping

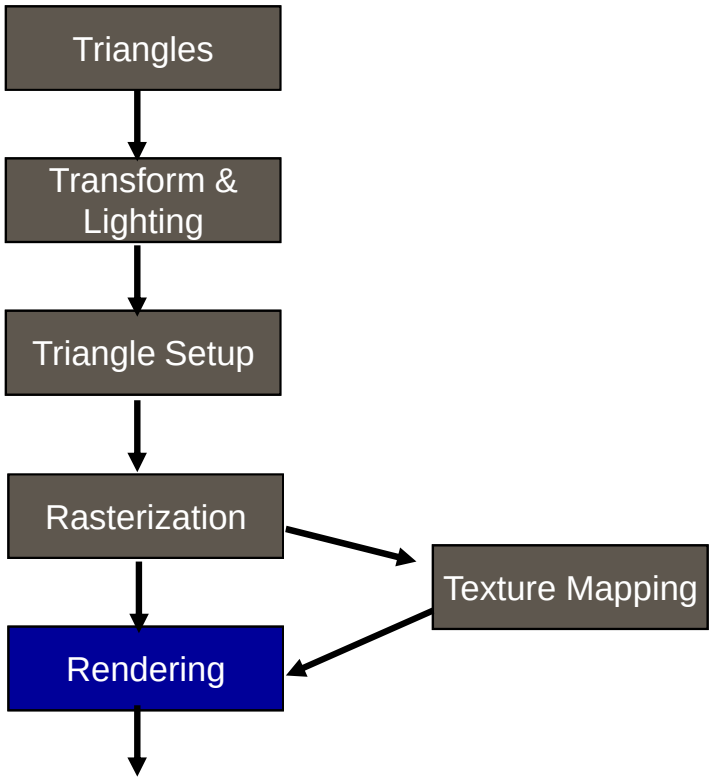
- **Adding Realism**







# Conventional 3D Graphics Pipeline




Conventional 3D Graphics Pipeline

## Rendering

***Blending***

# Blending

- Alpha Blending

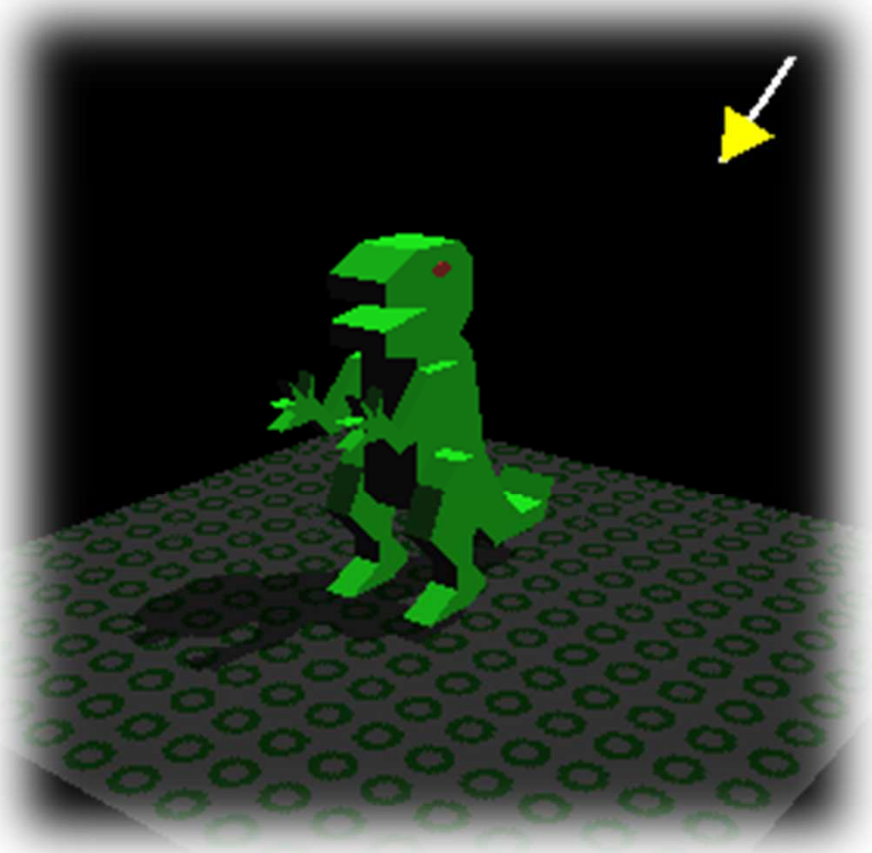
# *Advanced Features*

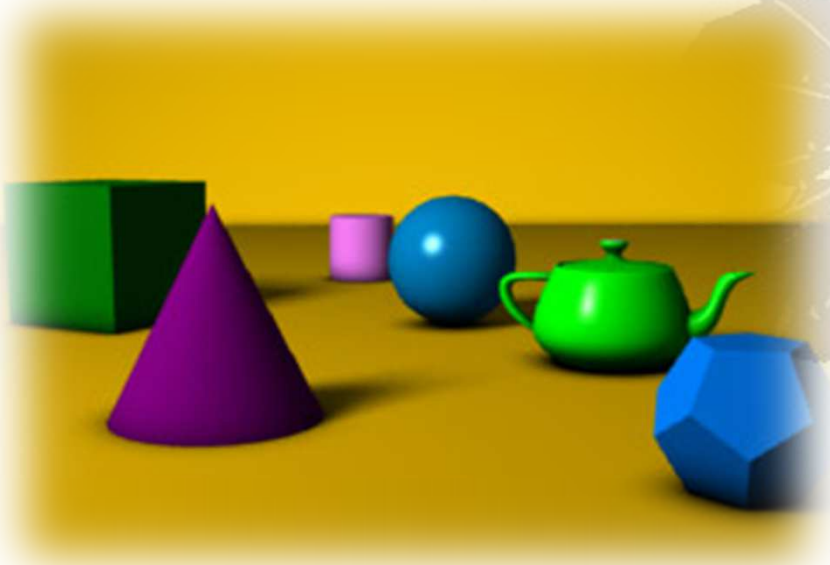
***Shadow***

***Anti-aliasing***

***Shaders***

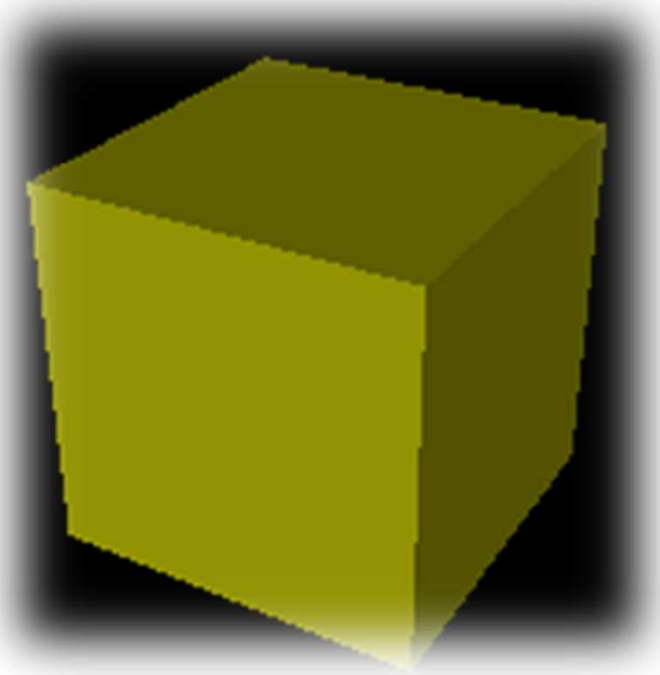
# Shadow

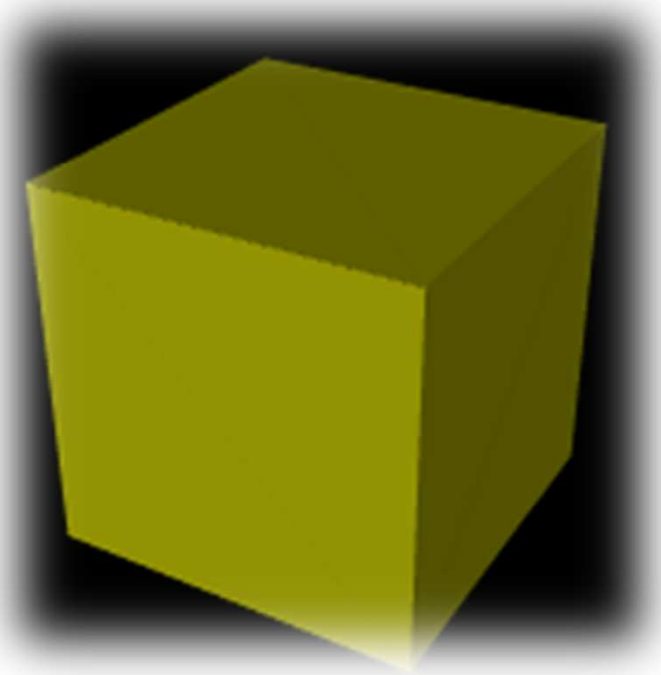




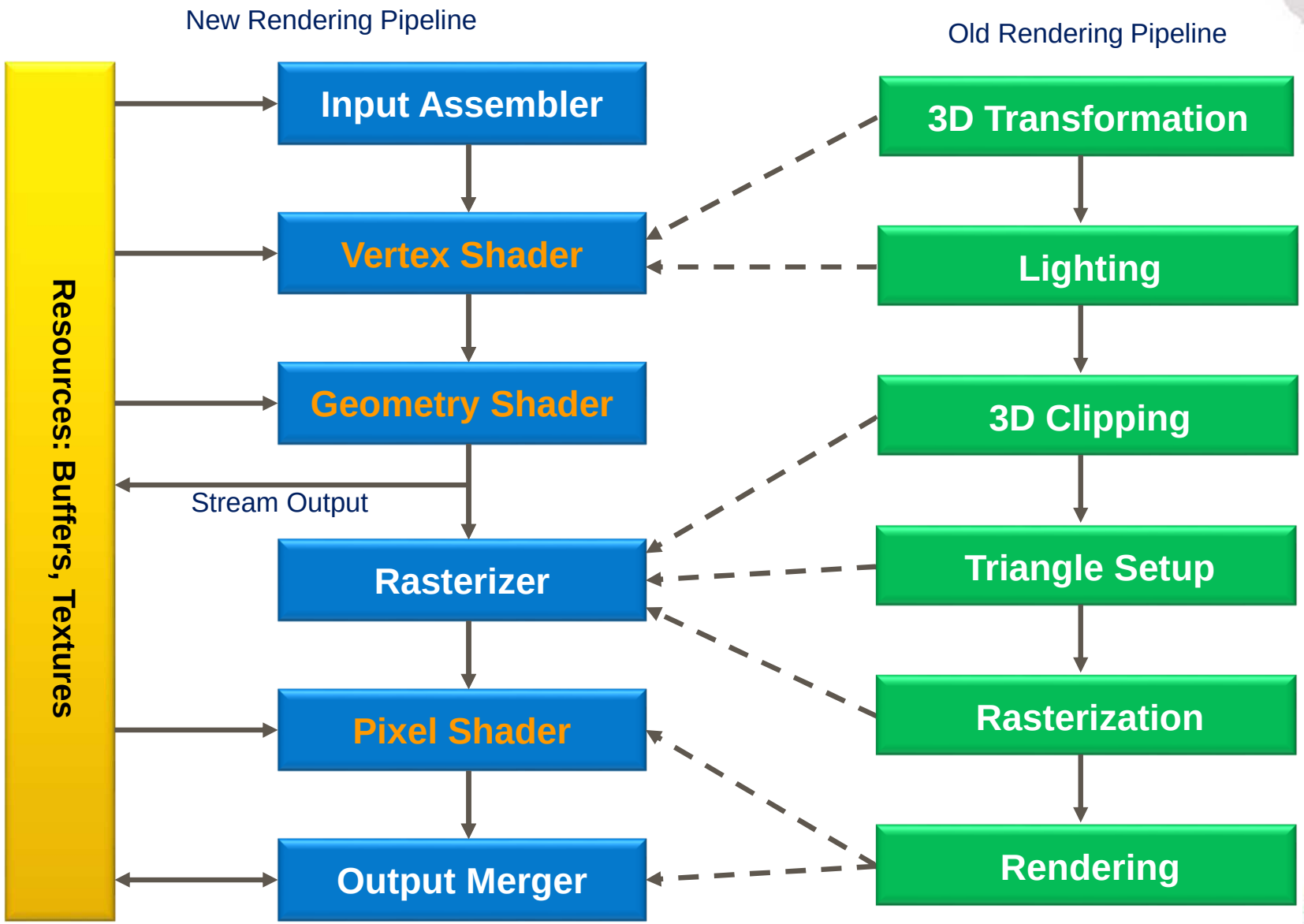
# *Anti-Aliasing*

- **Smooth out the Jaggy Edges**

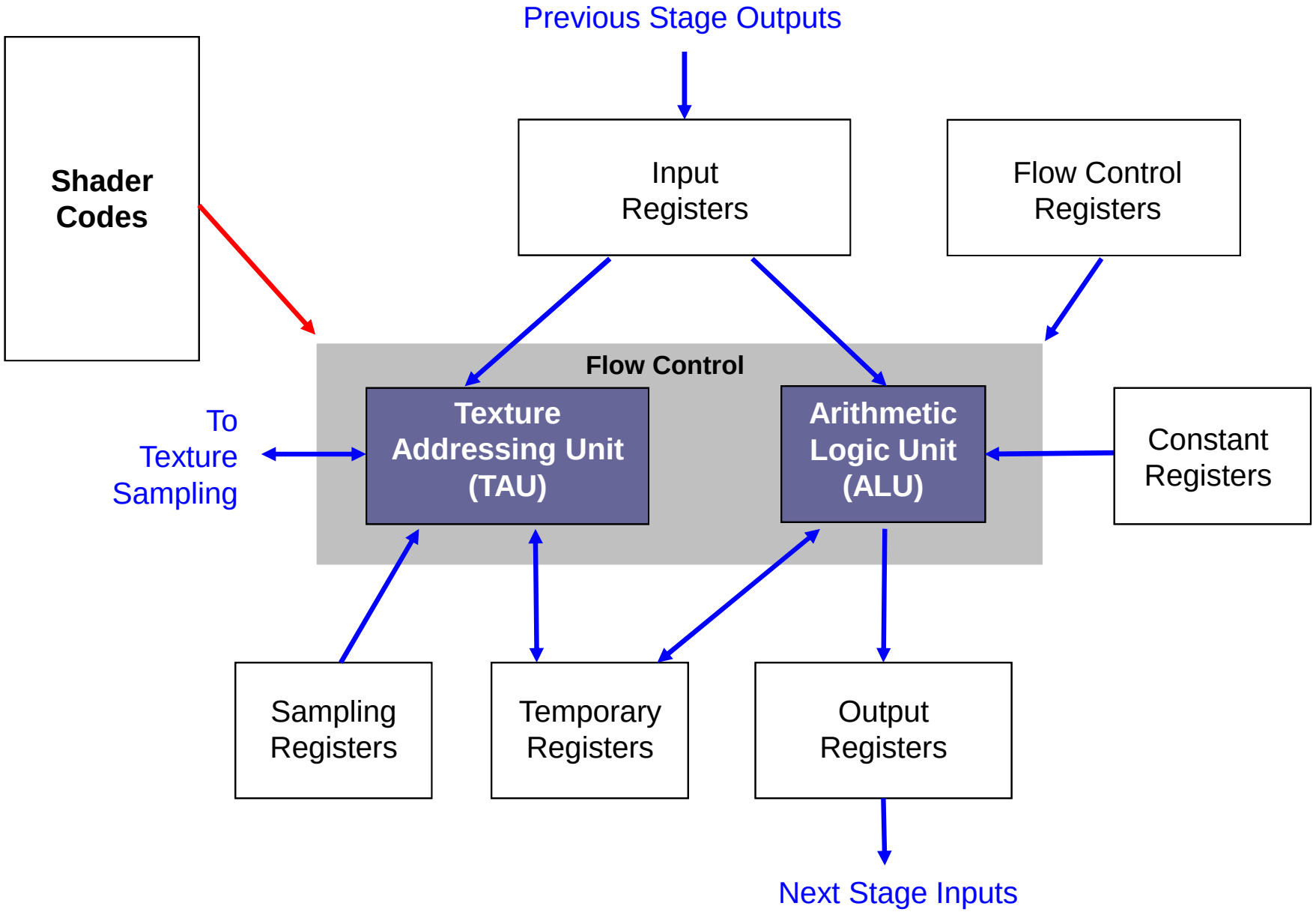




# New Rendering Pipeline with Shaders

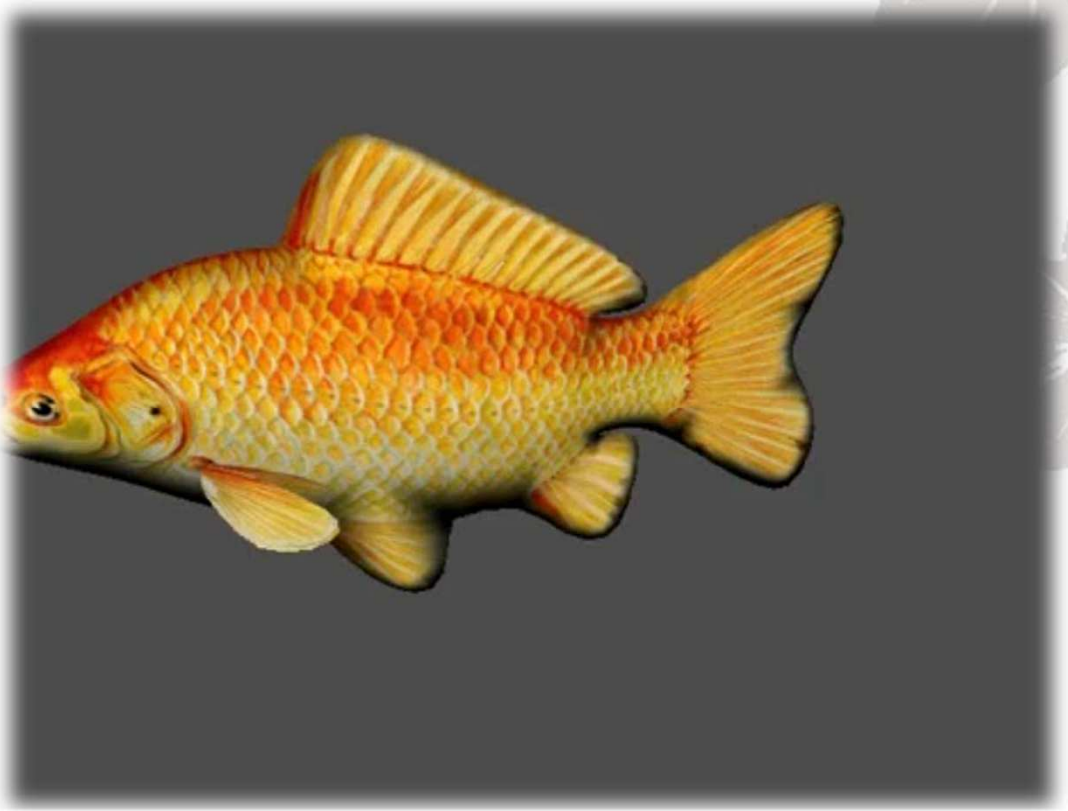


# Shader Architecture



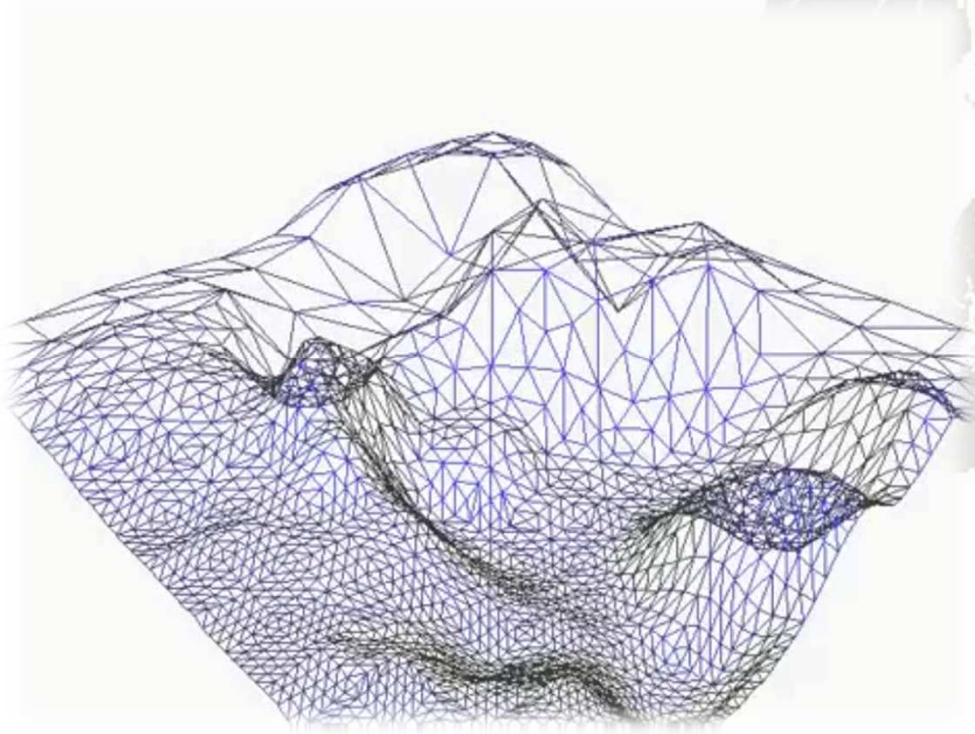
# *Vertex Shader*

- **Processes vertices**
  - **Transformation**
  - **Lighting**
  - **Displacement**
- **Operate on a single input vertex and produce a single output vertex**



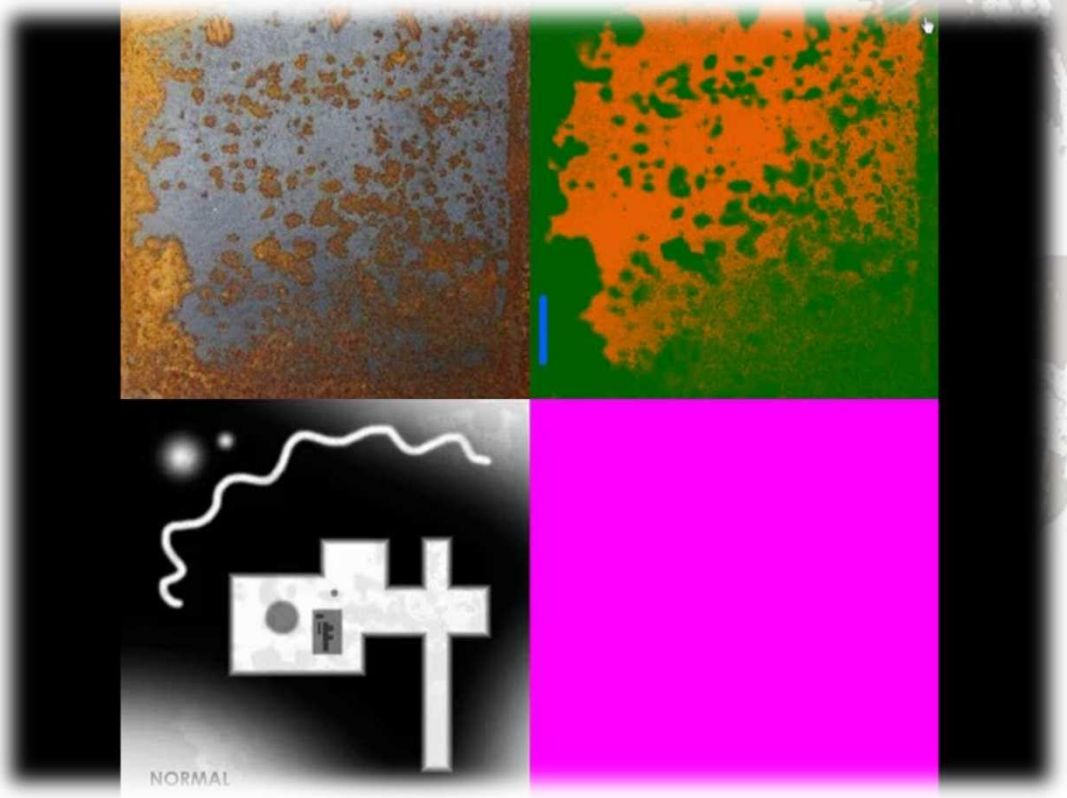
# Geometry Shader

- Process primitives
  - Point sprite tessellation
  - Wide line tessellation
  - Shadow volume generation
  - Surface subdivision
- Inputs one primitive. Outputs can be more than one primitives



# *Pixel Shader*

- **Process pixels**
  - **Texture mapping**
  - **Color combine**
  - **Per-pixel lighting**
  - **…**
- **Inputs one pixel. Outputs one pixels at same position, or no pixel.**



# Q&A