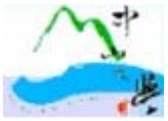
3D Computer Animation and Performance Tuning

李宗南

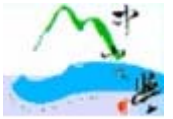


Dept. of Computer Science Engineering

National Sun Yat-Sen University

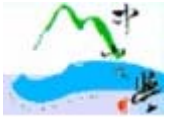
November 27, 2007

Contents



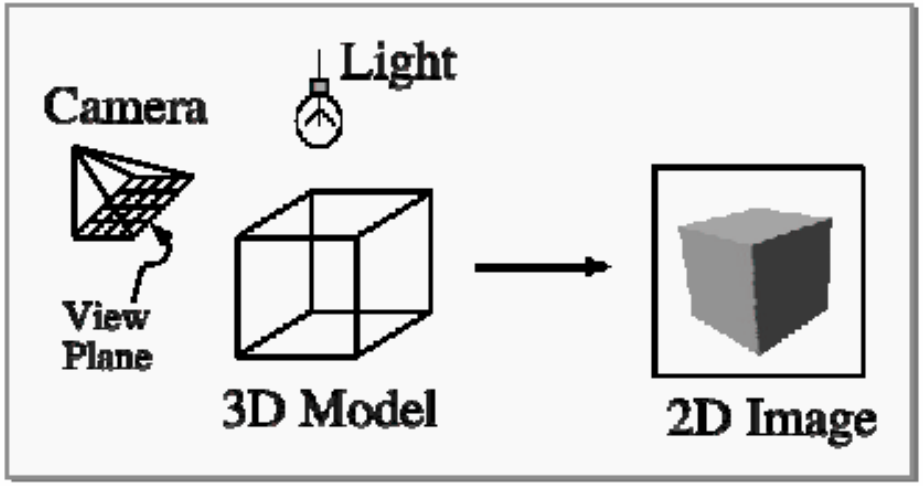
1. Introduction of Computer Graphics
2. Introduction of Computer Animation
3. Graphics Research at NSYSU
4. Performance Tuning
5. Conclusion

Introduction of Computer Graphics

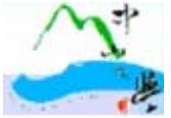


What is computer graphics?

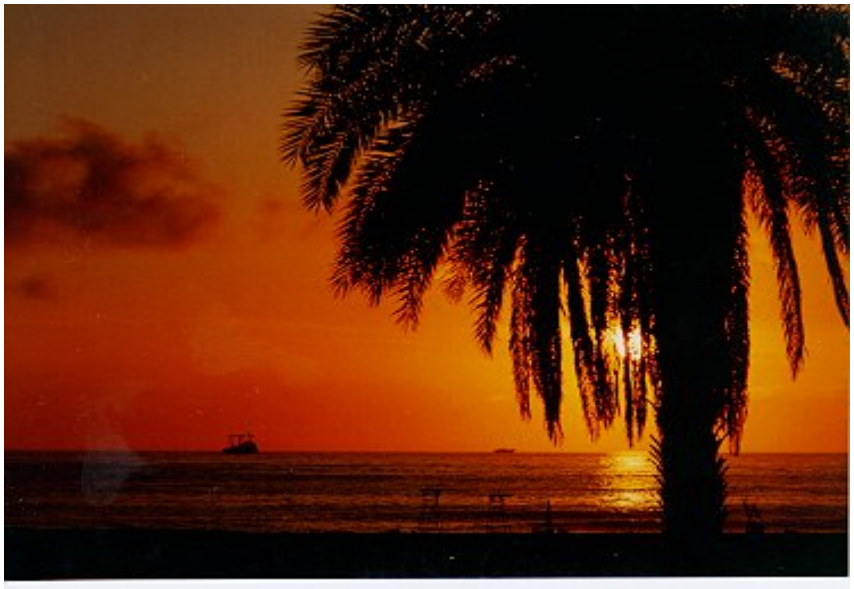
- Imaging = *representing 2D images*
- Modeling = *representing 3D objects*
- Rendering = *constructing 2D images from 3D models*
- Animation = *simulating changes over time*



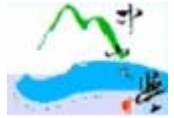
Introduction of Computer Graphics



A picture is worth of thousand words



Introduction of Computer Graphics

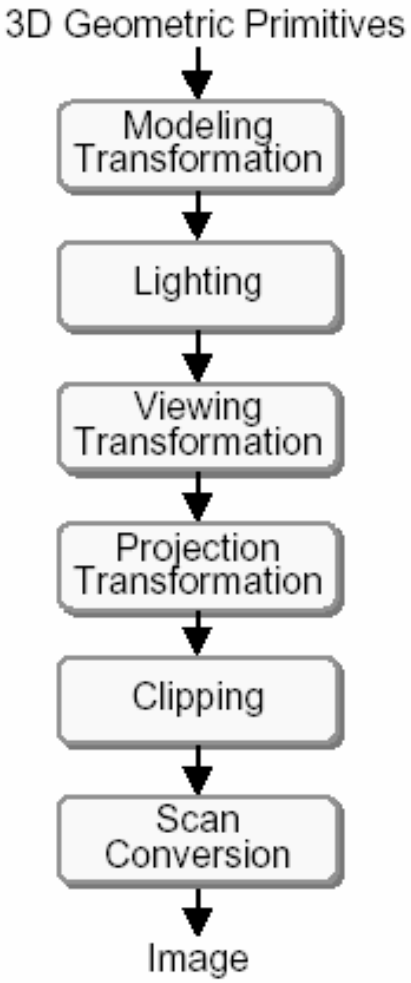


A picture is worth of thousand words

Introduction of Computer Graphics

Courtesy of nVidia

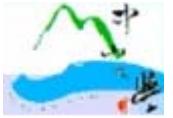
Introduction of Computer Graphics



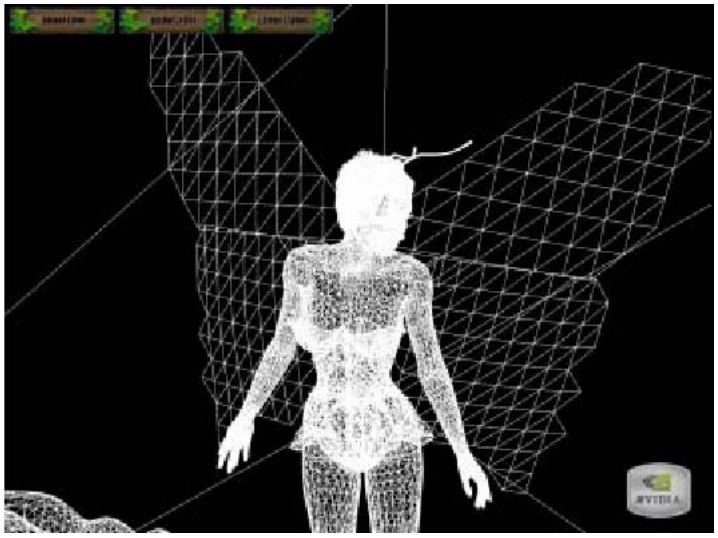
This is a pipelined sequence of operations to draw a 3D primitive into a 2D image

Rendering

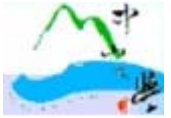
Introduction of Computer Graphics



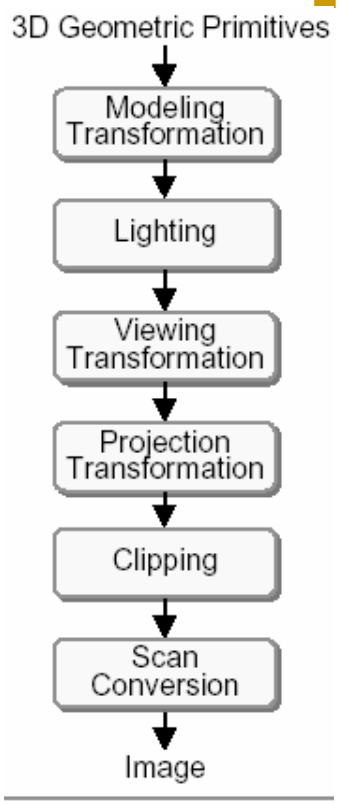
- Polygon mesh

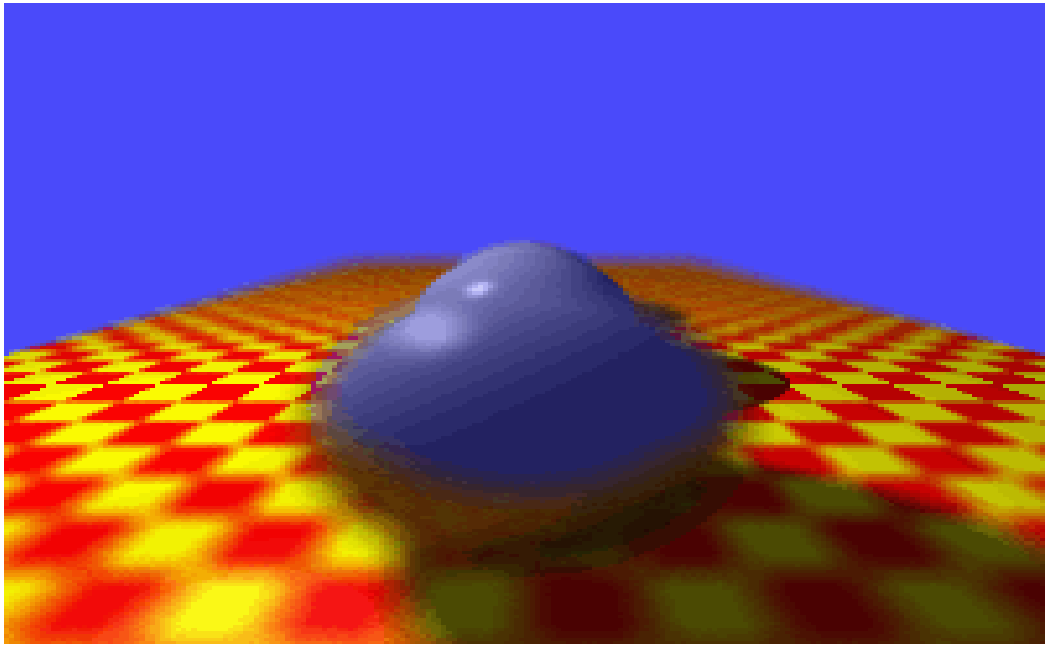


Introduction of Computer Graphics

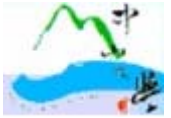


- Transformation



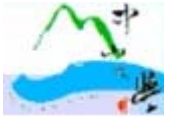


Introduction of Computer Graphics

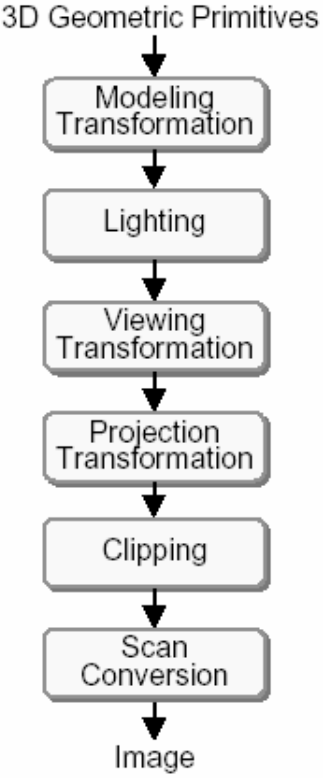


- Lighting

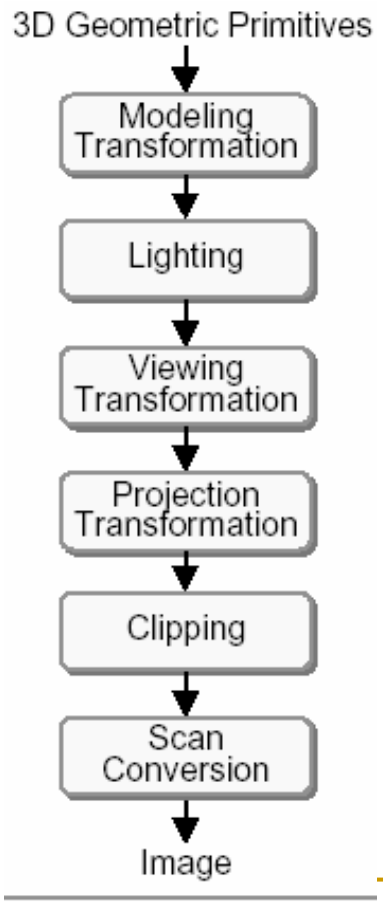
Introduction of Computer Graphics



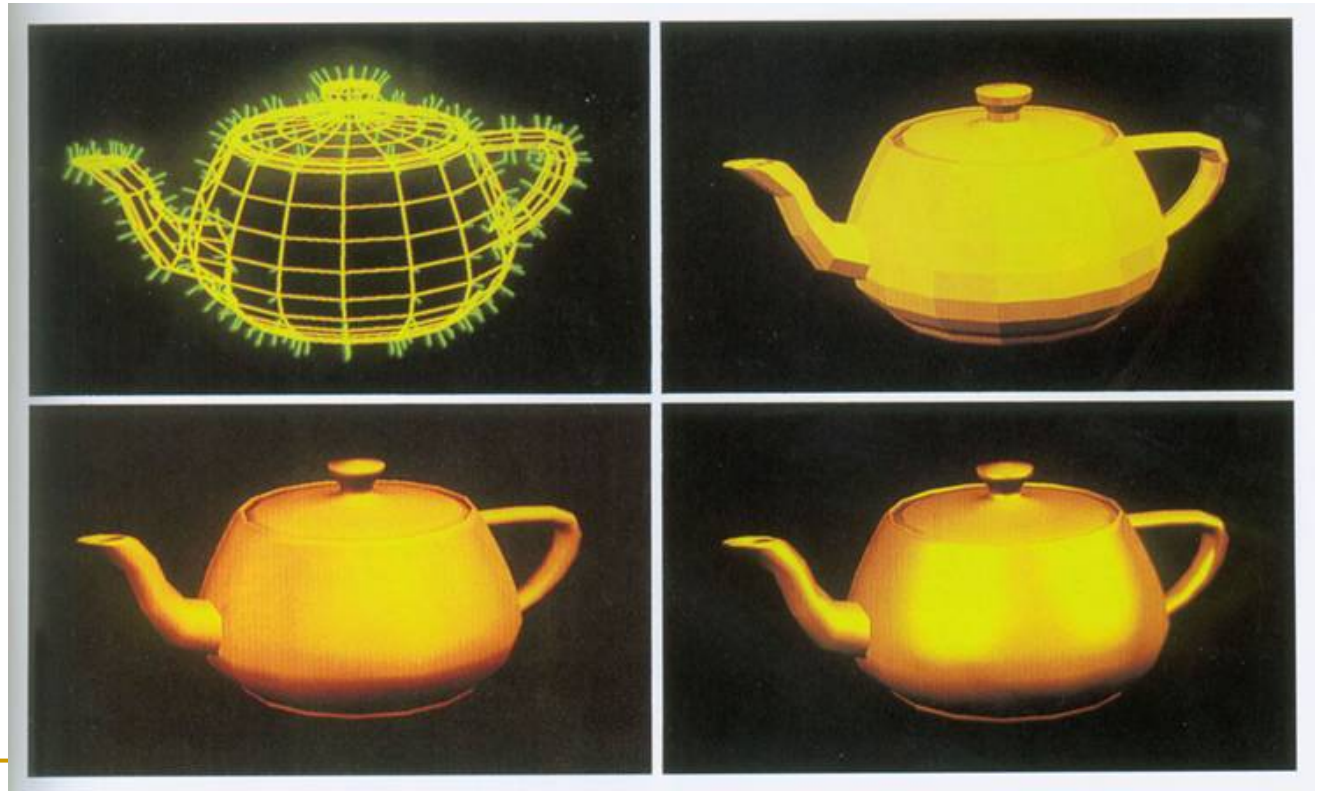
- Lighting



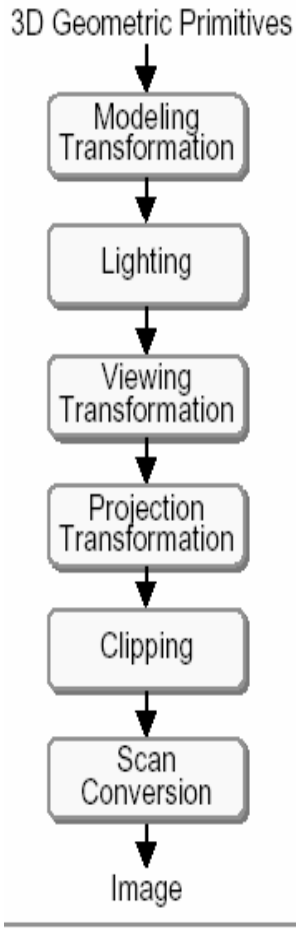
Introduction of Computer Graphics



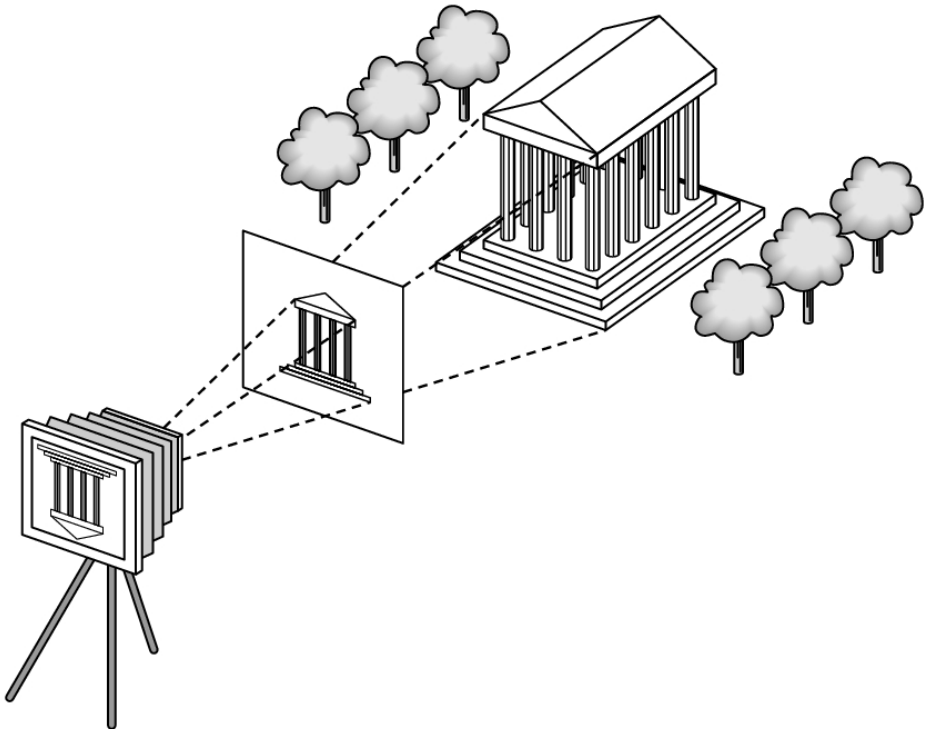
- Different lighting models



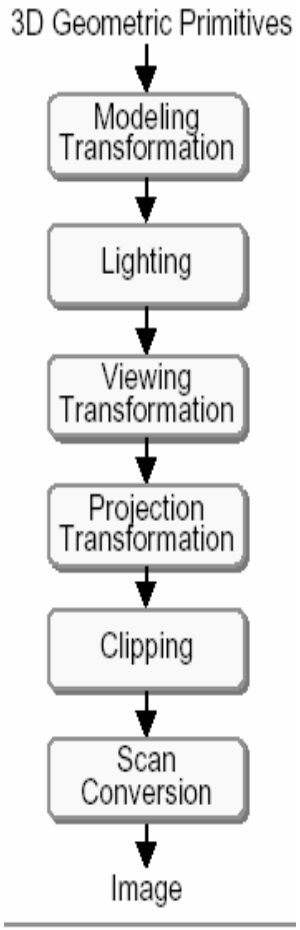
Introduction of Computer Graphics



Projection



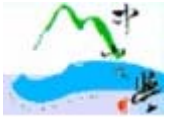
Introduction of Computer Graphics



Clipping

- Avoid drawing parts of primitives outside window
  - Window defines part of scene being viewed
  - Must draw geometric primitives only inside window

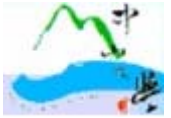
Screen Coordinates



Applications

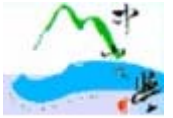
What is it good for?

Applications



- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Applications



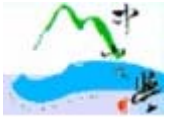
- **Entertainment**
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Geri's Game
(Ptxar Animation Studios)

Jurasic Park
(Industrial, Light, & Magic)

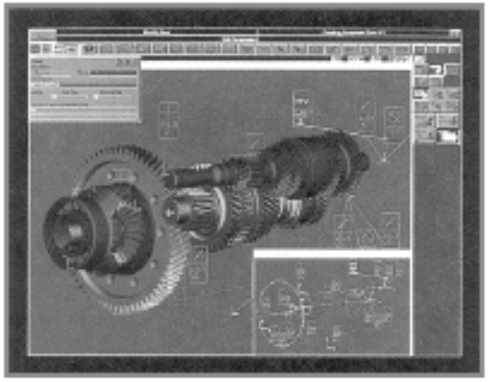
Quake
(Id Software)

Applications



- Entertainment
- **Computer-aided design**
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Los Angeles Airport
(Bill Jepson, UCLA)

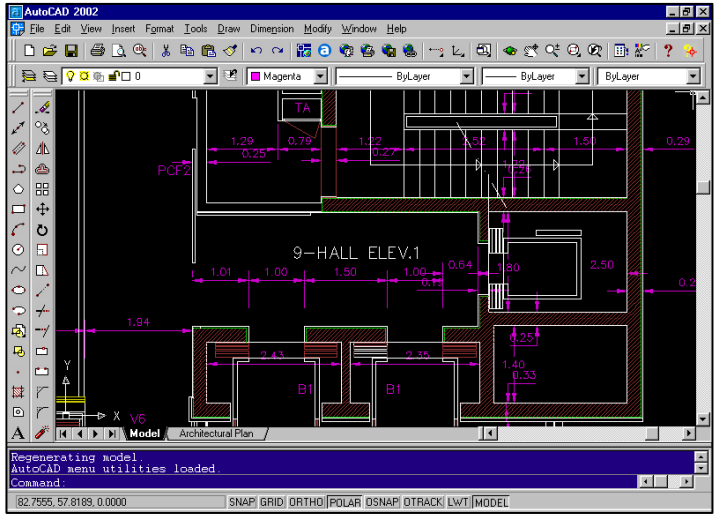


Gear Shaft Design
(Intergraph Corporation)

Boeing 777 Airplane
(Boeing Corporation)

Computer-Aided Design

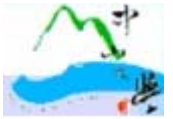
- Graphics for Engineering and Architectural System
- Design of Building, Automobile, Aircraft, Machine etc.



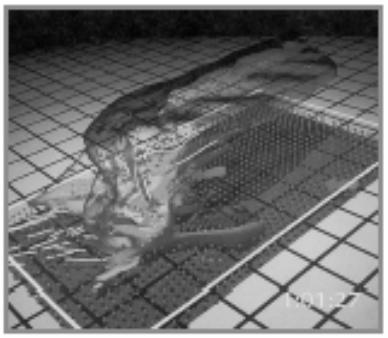
AutoCAD 2002

Interior Design

Applications

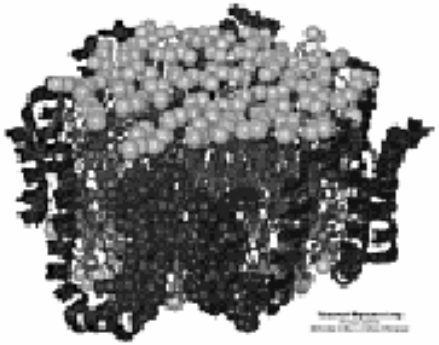


- Entertainment
- Computer-aided design
- **Scientific visualization**
- Training
- Education
- E-commerce
- Computer art



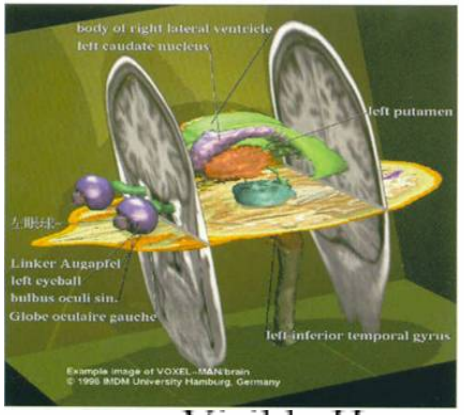
Airflow Inside a Thunderstorm

(Bob Wilhelmson, University of Illinois at Urbana-Champaign)



Apo A-1

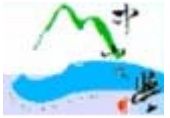
(Theoretical Biophysics Group, University of Illinois at Urbana-Champaign)



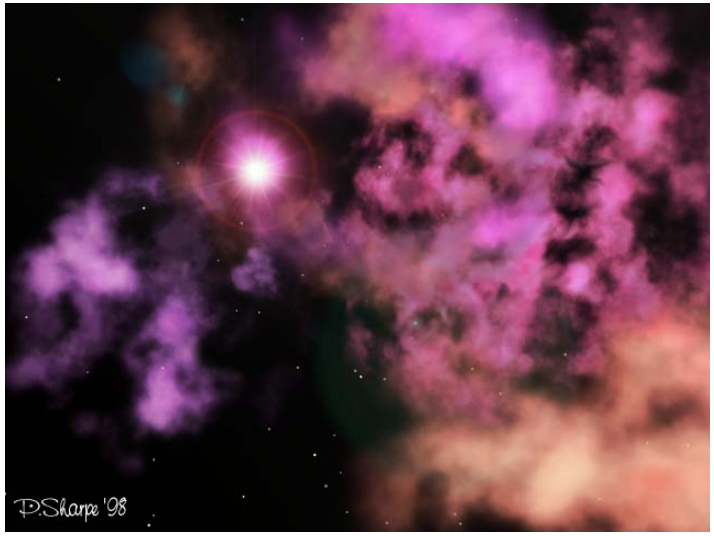
Visible Human

(National Library of Medicine)

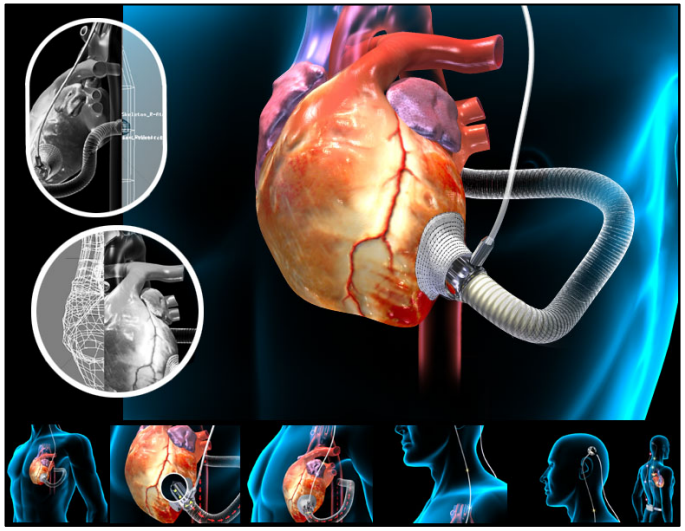
Scientific Visualization



- Graphics for Scientific, Engineering, and Medical Data

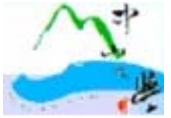


Nebula

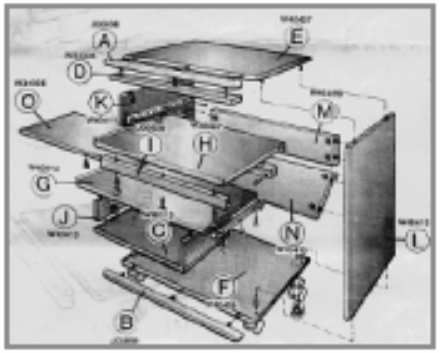


Medical Image

Applications



- Entertainment
- Computer-aided design
- Scientific visualization
- **Training**
- Education
- E-commerce
- Computer art



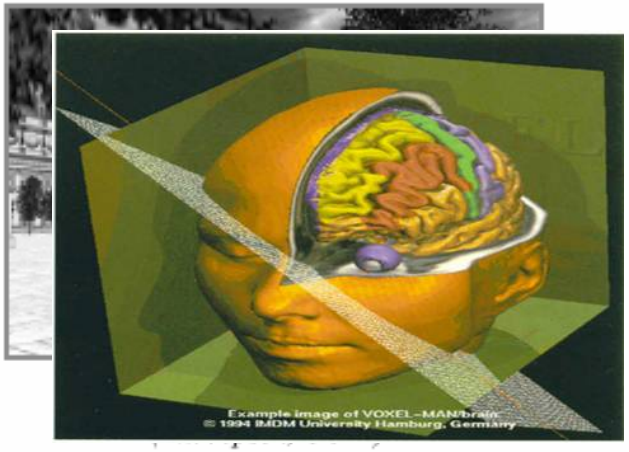
Desk Assembly
(Silicon Graphics, Inc.)

Driving Simulation
(Evans & Sutherland)

Flight Simulation
(NASA)

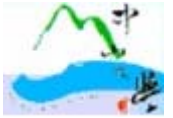
Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- **Education**
- E-commerce
- Computer art



Human Skeleton
(SGI)

Applications

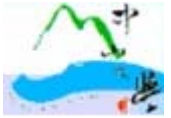


- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- **E-commerce**
- Computer art

Interactive Kitchen Planner
(Matsushtta)

Virtual Phone Store
(Lucent Technologtes)

Applications

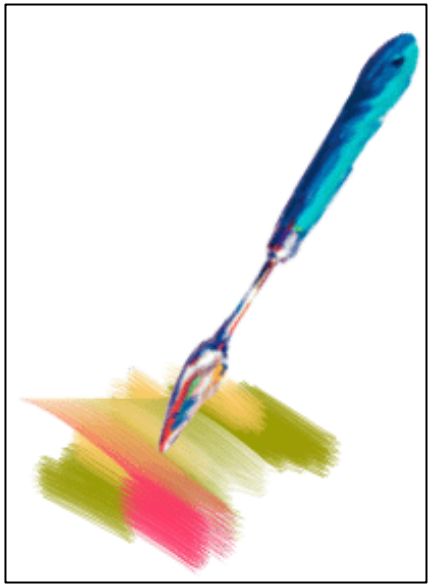


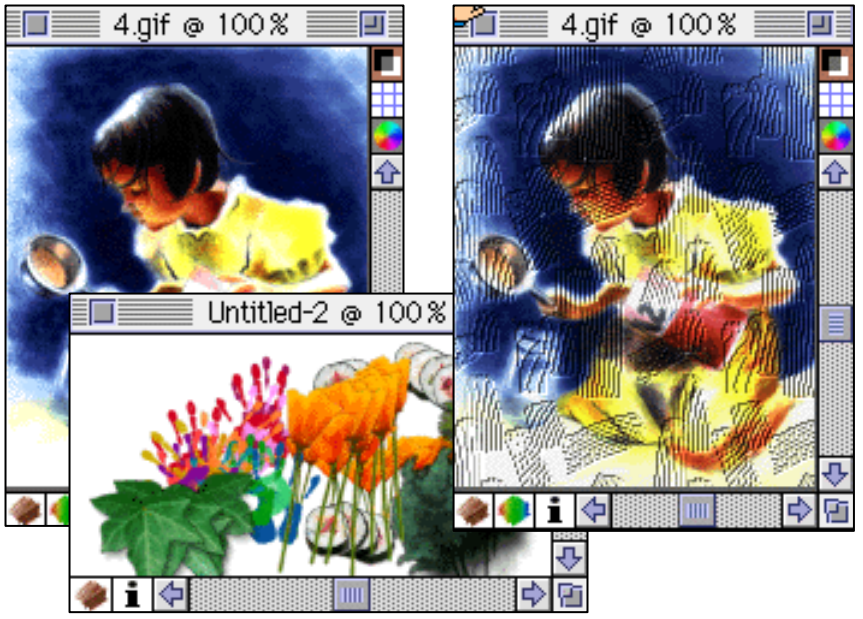
- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- **Computer art**

Blair Arch

(Marissa Range & Adam Finkelstein, Princeton University)c

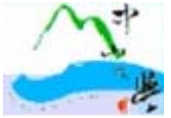
Computer art
-Graphics for artist





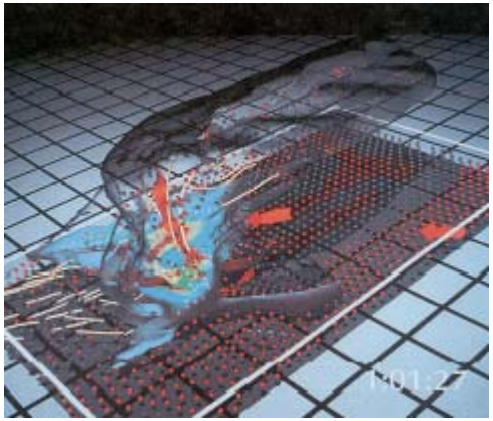
Metacreation Painter

Computer Animation



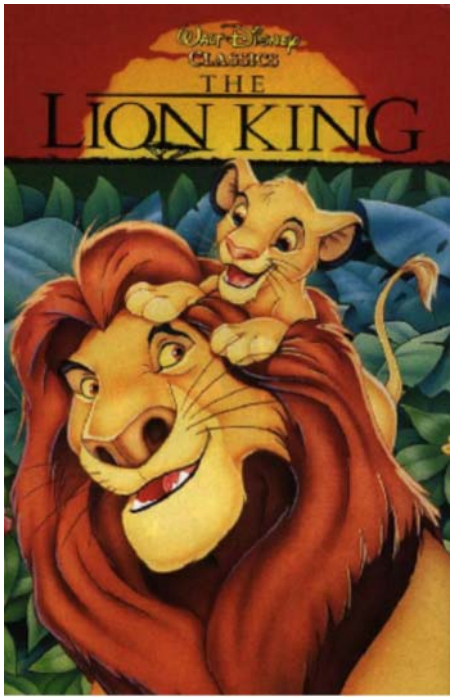
- **What is Animation?**
  - Make objects change over time according to scripted actions

- **What is Simulation?**
  - Predict how objects change over time according to physical laws



Animation types

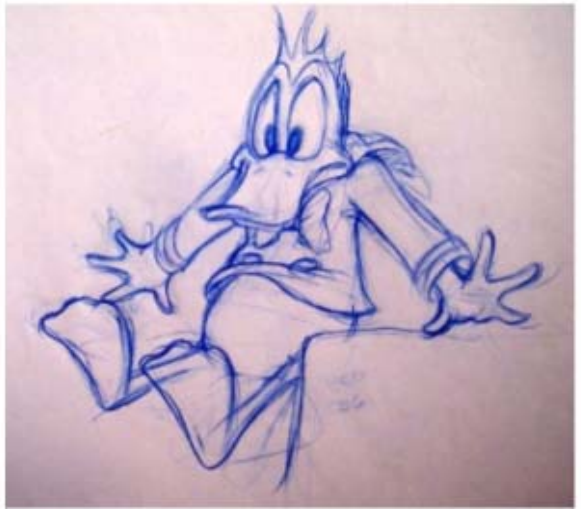
Fully animated features
Computer Animation



Computer Animation

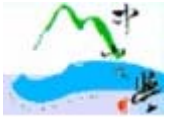
Animation types

- Cartoon shorts



Computer Animation- Visual effects

How does animation get made?

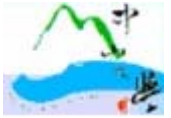


- Lot of people
- Lot of time
- Lot of money

Step by step through a pipeline…

Preproduction, Production, Post-production

Animation pipeline



Story
- •Idea, pitch
- •Script
- •Storyboards
- •Story reel (editorial) – continues through production

Visual development
- •Concept art

Preprocessing

- •Layout
- •Backgrounds or set design
- •Character design
- •Color modeling or shading
- •Animation
- •Effects
- •Ink and paint, rendering, compositing, to

Processing

Post-processing

- •Sound design, music
- •Edit
- •Marketing, distribution

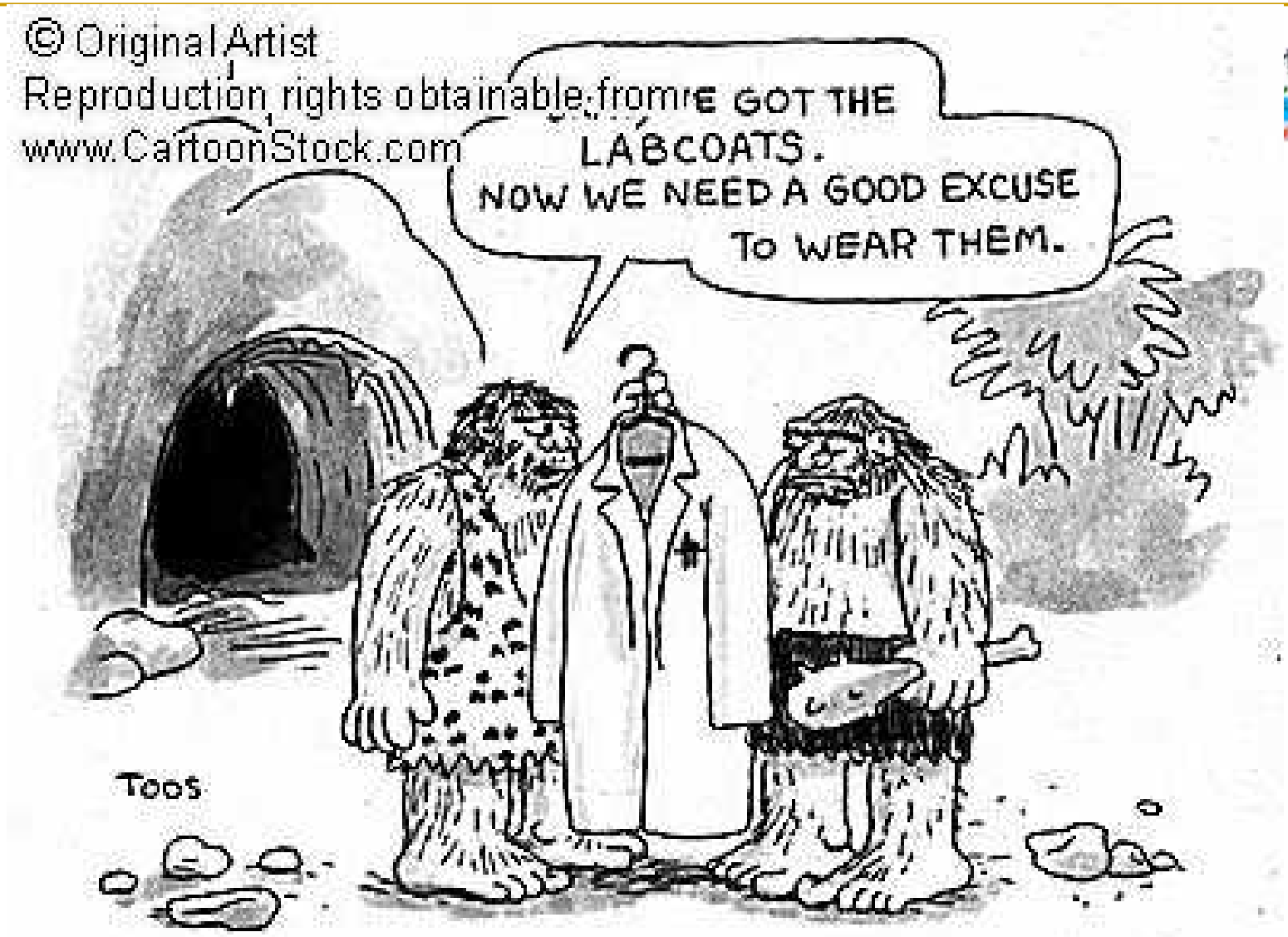
Animation pipeline: pre-production ?

Story

- Idea, pitch
- Script
- Storyboards
- Story reel (editorial) – continues through production

Visual development

- Concept art



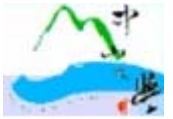
The Birth of Science

Storyboard


LOOKA THERE, EGGO — A LONESOME TYPE PICNIC
BASKET JUST ASKING TO BE PICKED UP AND PICK-NICKED!
HE'D BETTER LOOK, SERIOUS
HE'S HEADED FOR THE BASKET — HE LOOKS AROUND — NOW HE'S SNEAKIER — HE LOOKS AROUND CAREFULLY — NOW — HE'S ABOUT TO LIFT THE LID —
SPRONG-SOCK!
(WHIMPERING SOUND — SOCK!)
(PUNCH'S VOICE) WHO DA WIDE GUY?
WHO DA WIDE GUY?
WELL — THAT OUGHT TO KEEP HIM OUT OF BASKETS.
YEAH — THAT WOULD DISCOURAGE ANY BEAR!
SHALL NOT SEEK AFTER ANY MORE BASKETS ARE YOU, YOU?
OF COURSE WE ARE, BIG SIR —

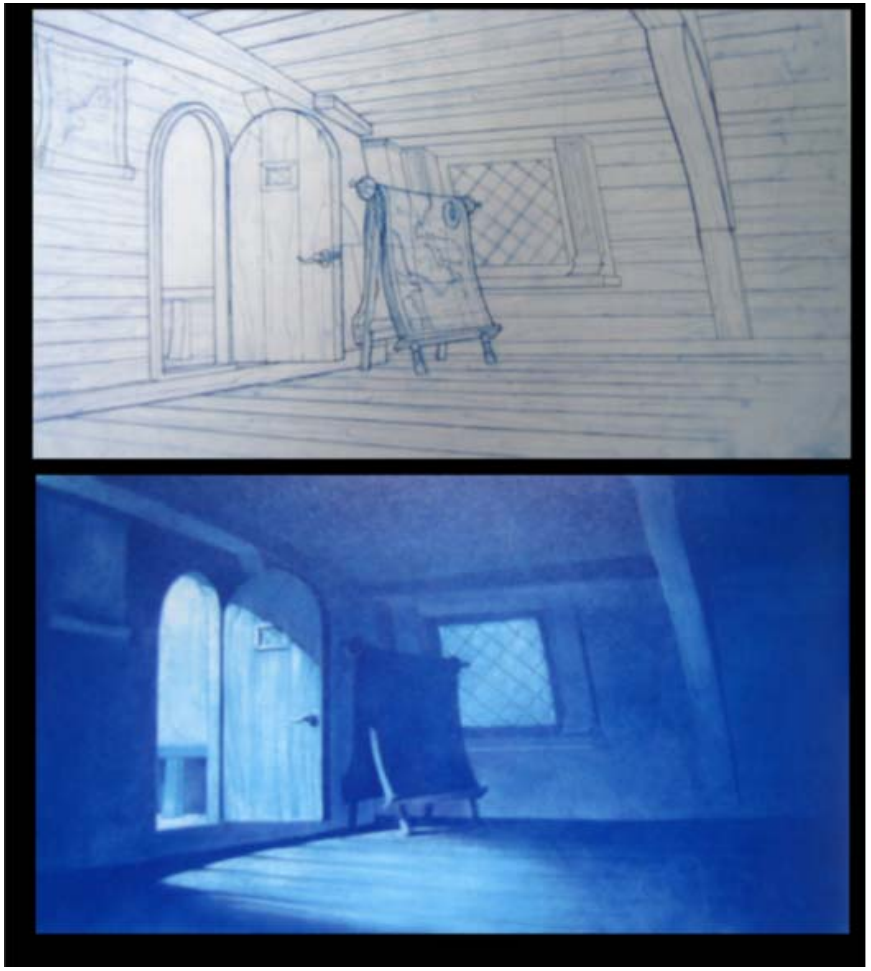
Story reel

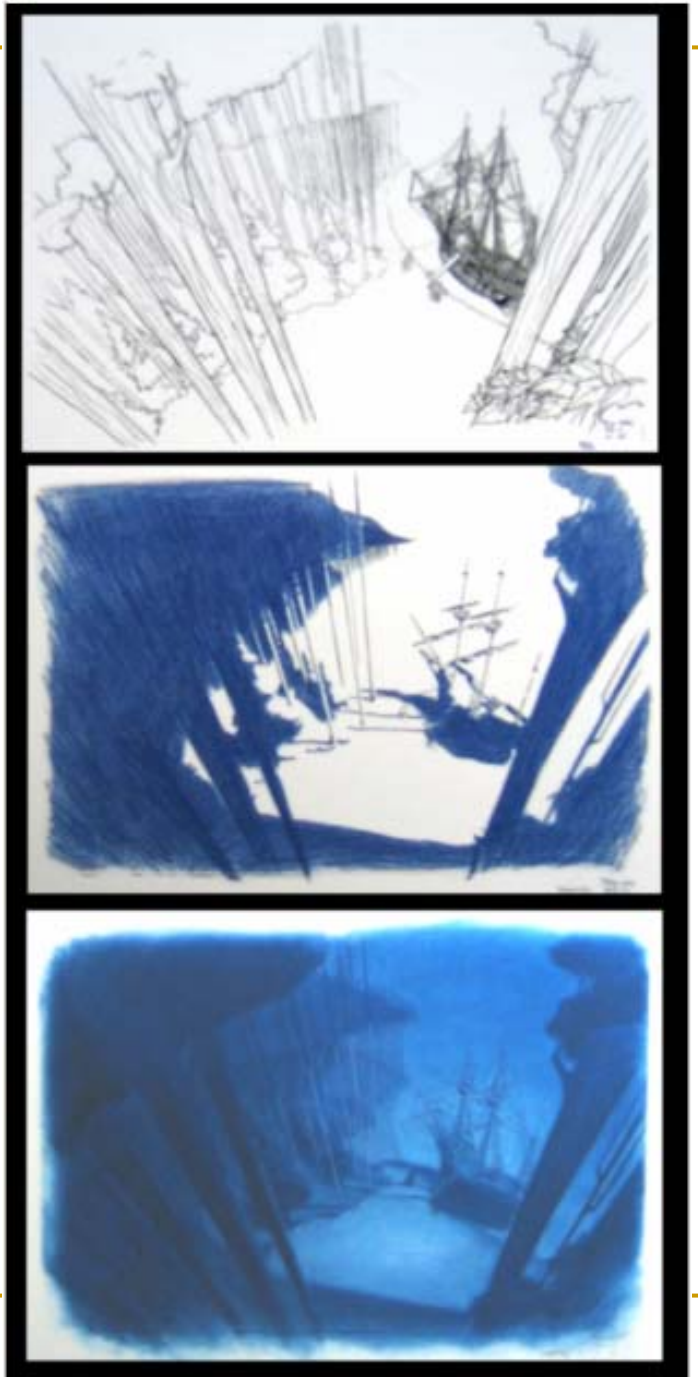
Animation pipeline: production



- Layout
- Backgrounds or set design
- Character design
- Color modeling or shading
- Animation
- Effects
- Ink and paint, rendering, compositing, to film

Layout

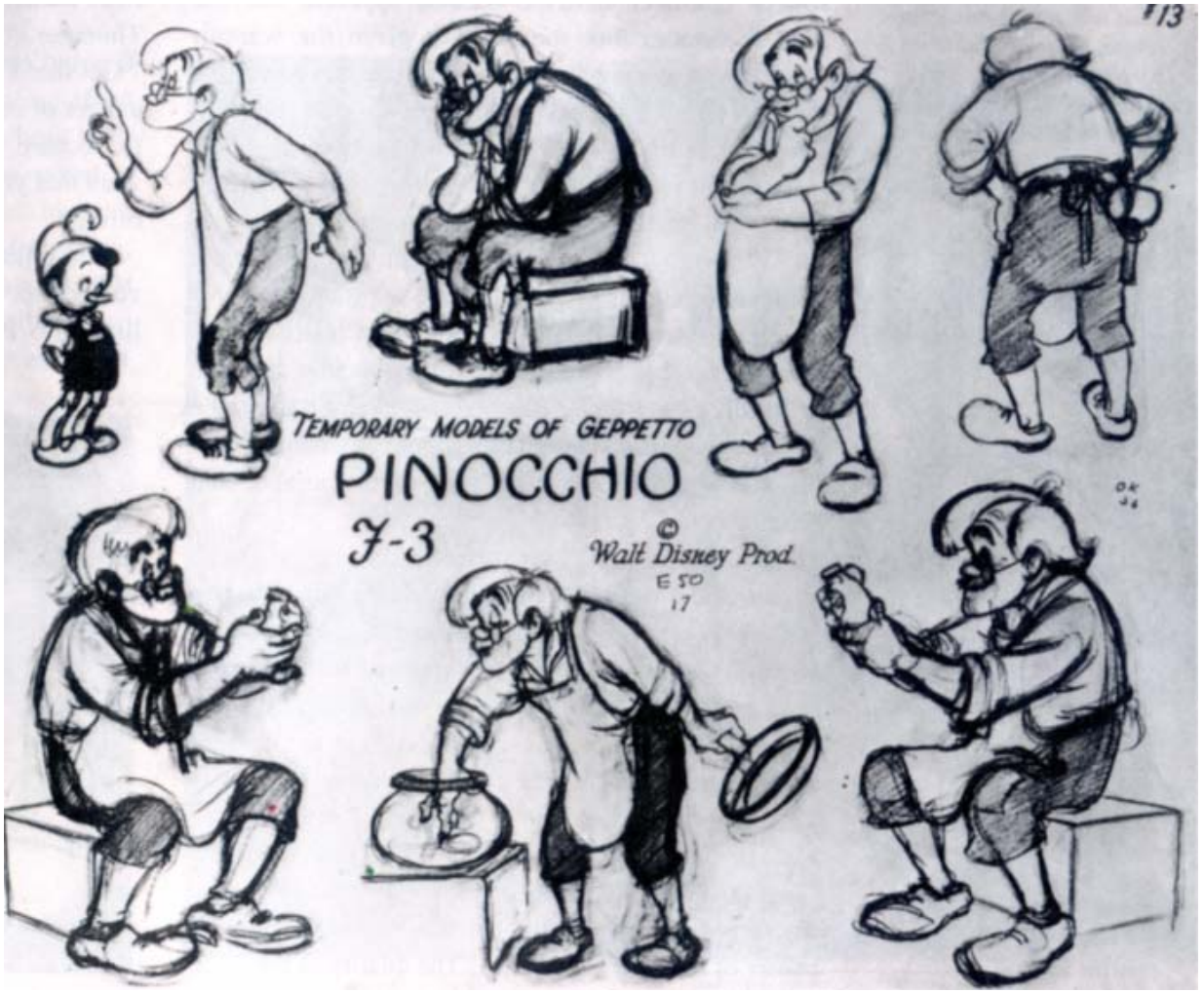




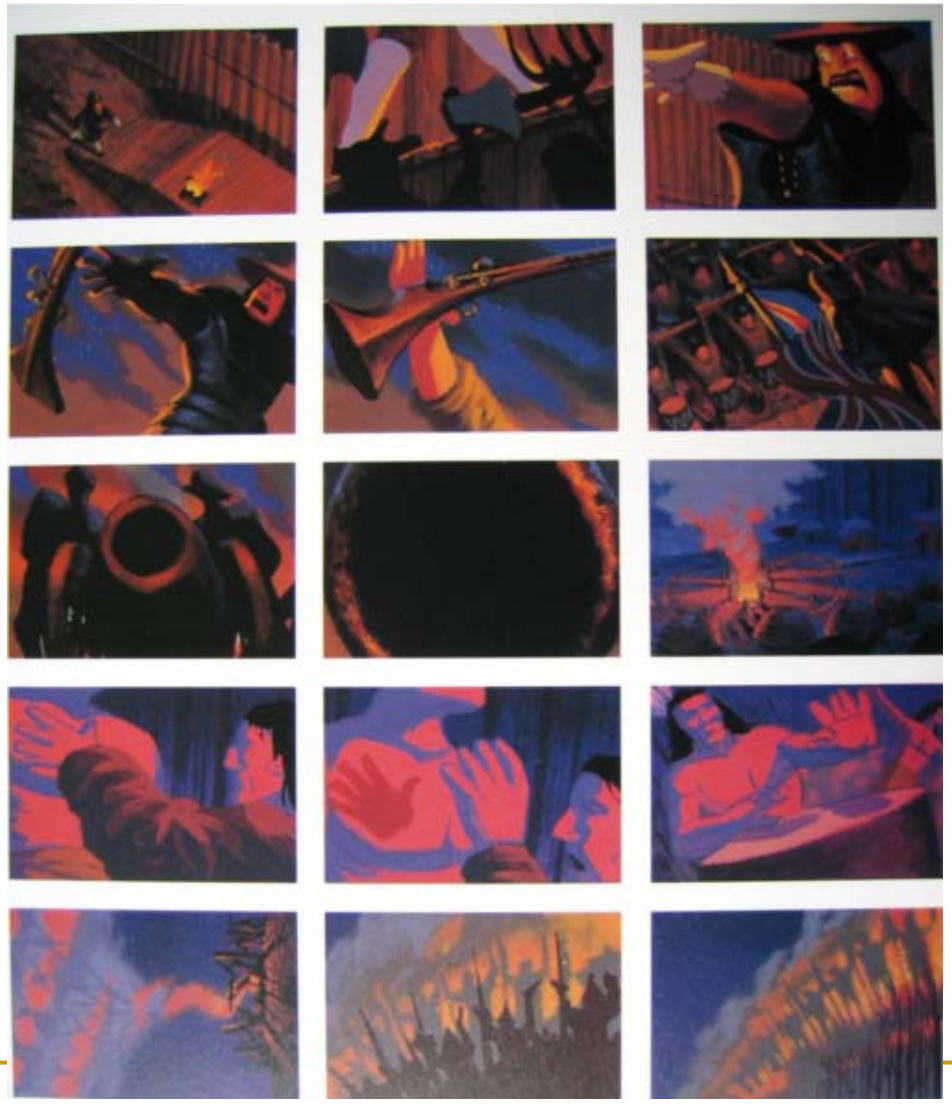
Set Design

Snow White

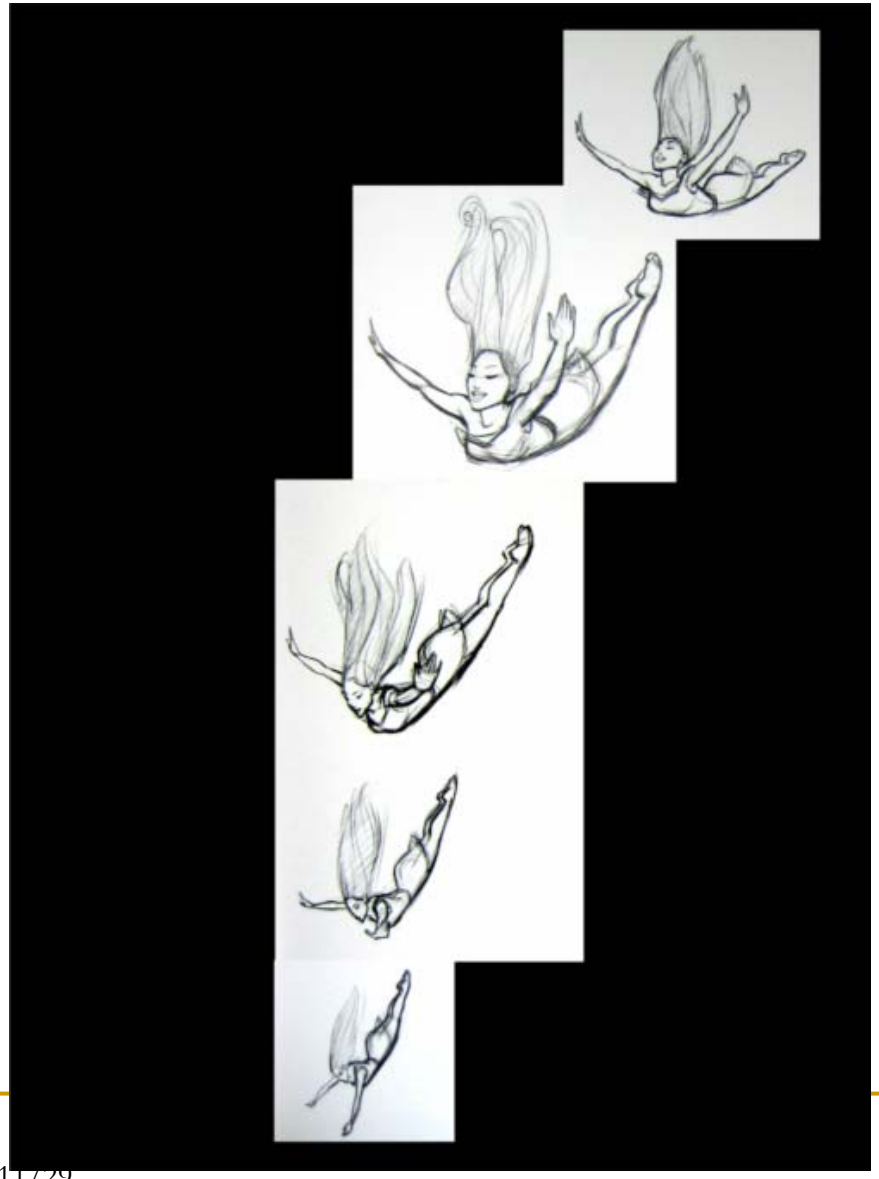
Character design



Color modeling and Shading

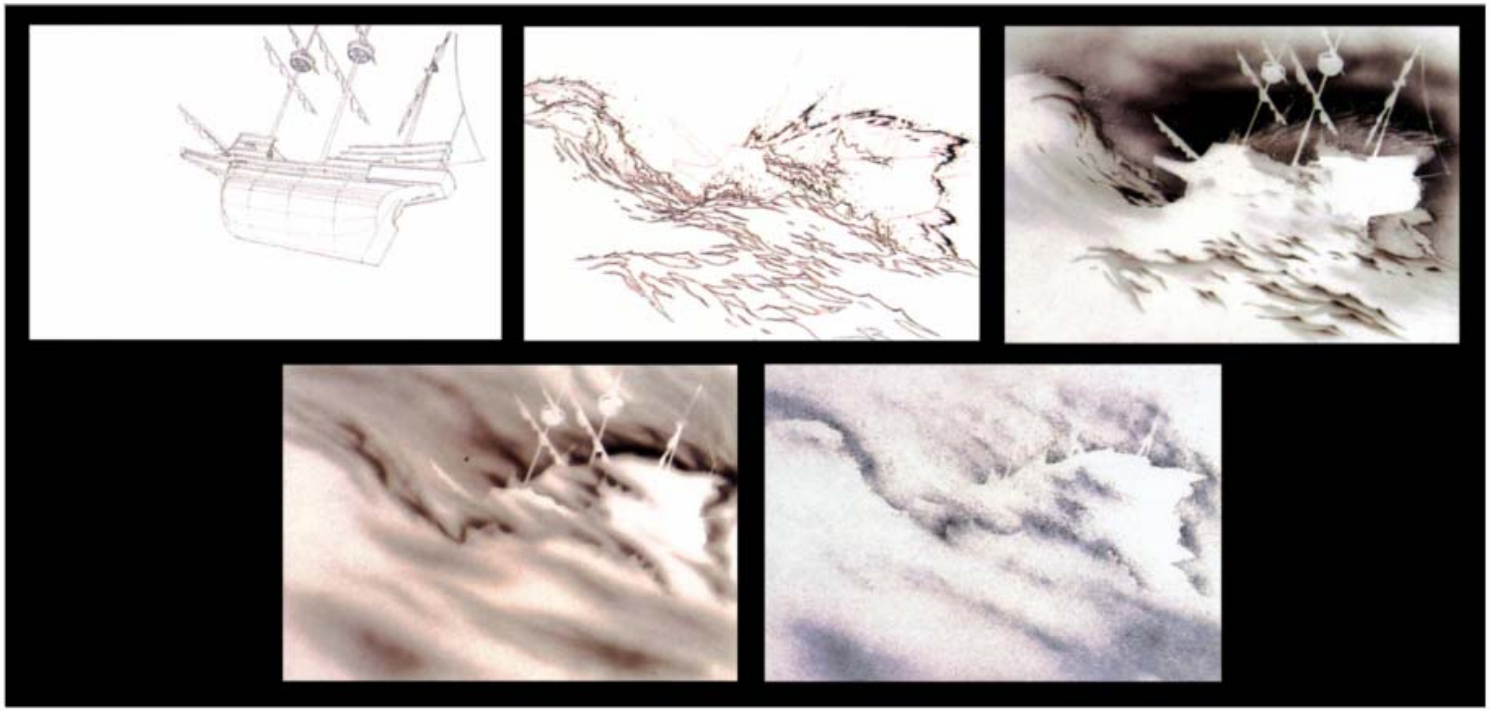


Animation



Courtesy of George Institute of Tech

Effects



Pocahontas

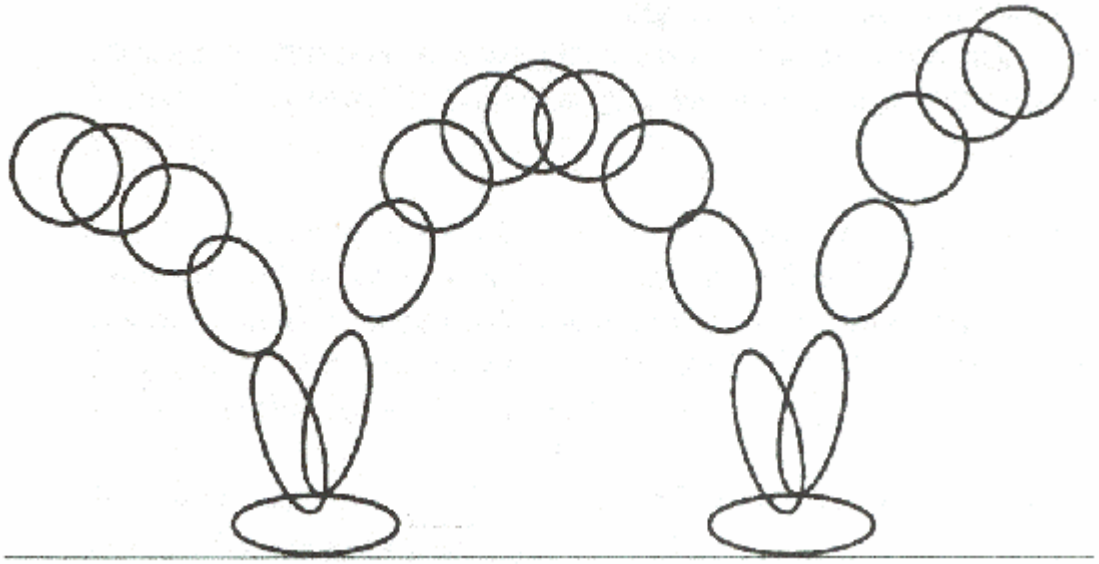
Animation pipeline: post-production

- Sound design, music
- Edit
- Marketing, distribution

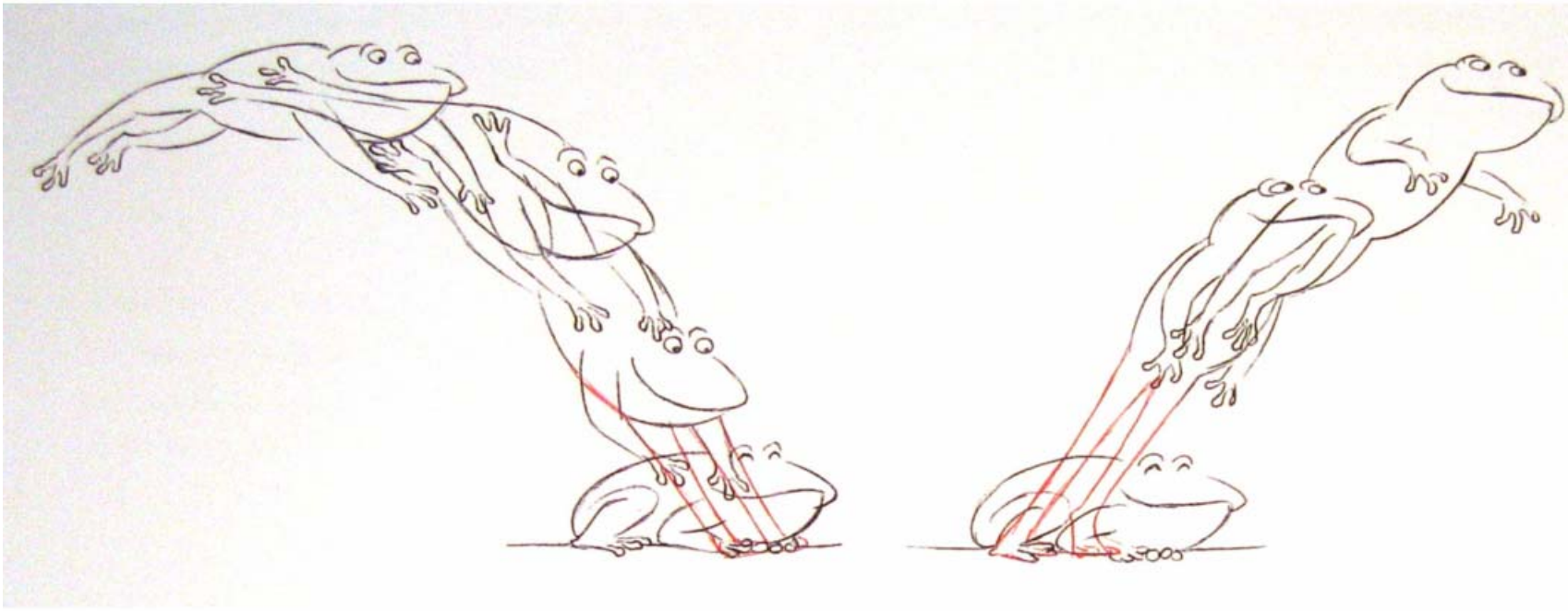
Principle of Traditional Animation – Disney –

- **Squash and Stretch**
- **Slow In and Out**
- **Anticipation**
- **Exaggeration**
- **Follow Through and Overlapping Action**
- **Timing**
- **Staging**
- **Straight Ahead Action and Pose-to-Pose Action**
- **Arcs**
- **Secondary Action**
- **Appeal**

Squash and Stretch

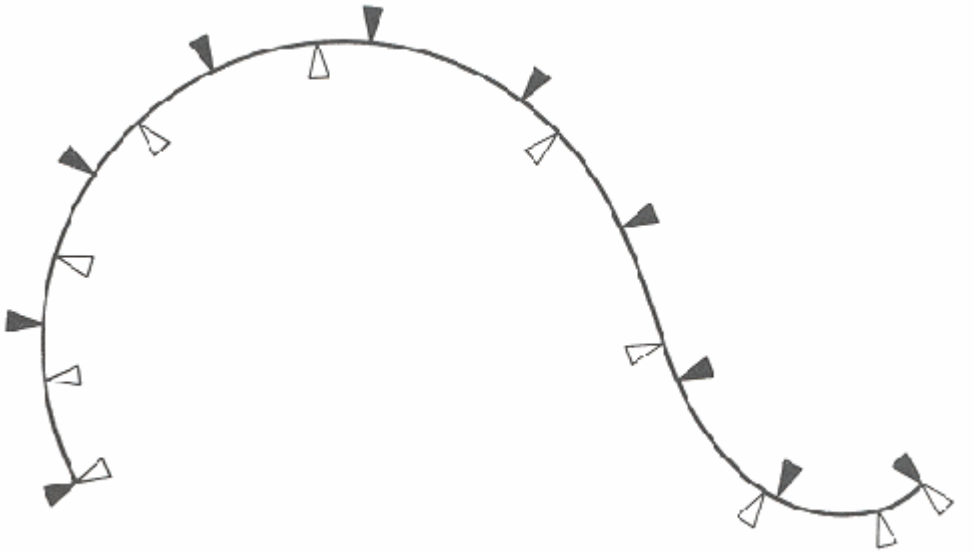


Squash and Stretch

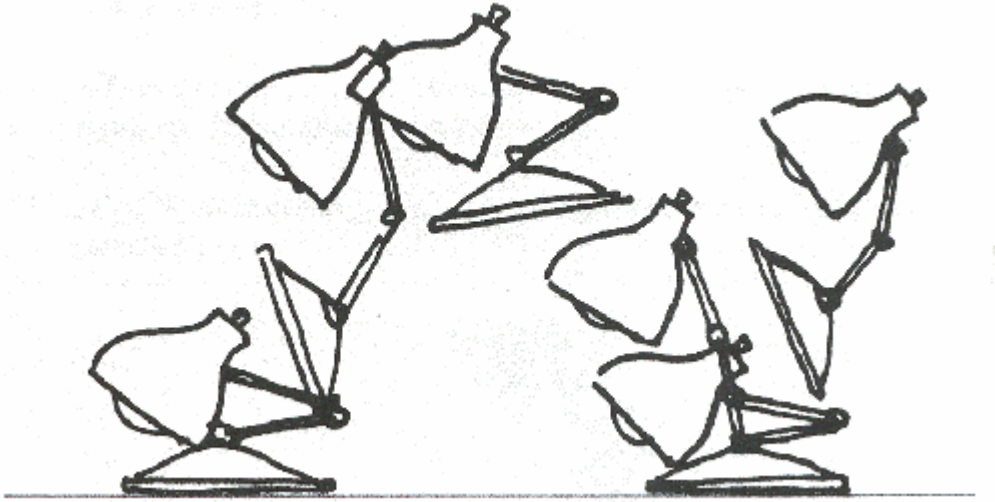


Courtesy of Barbara Meier

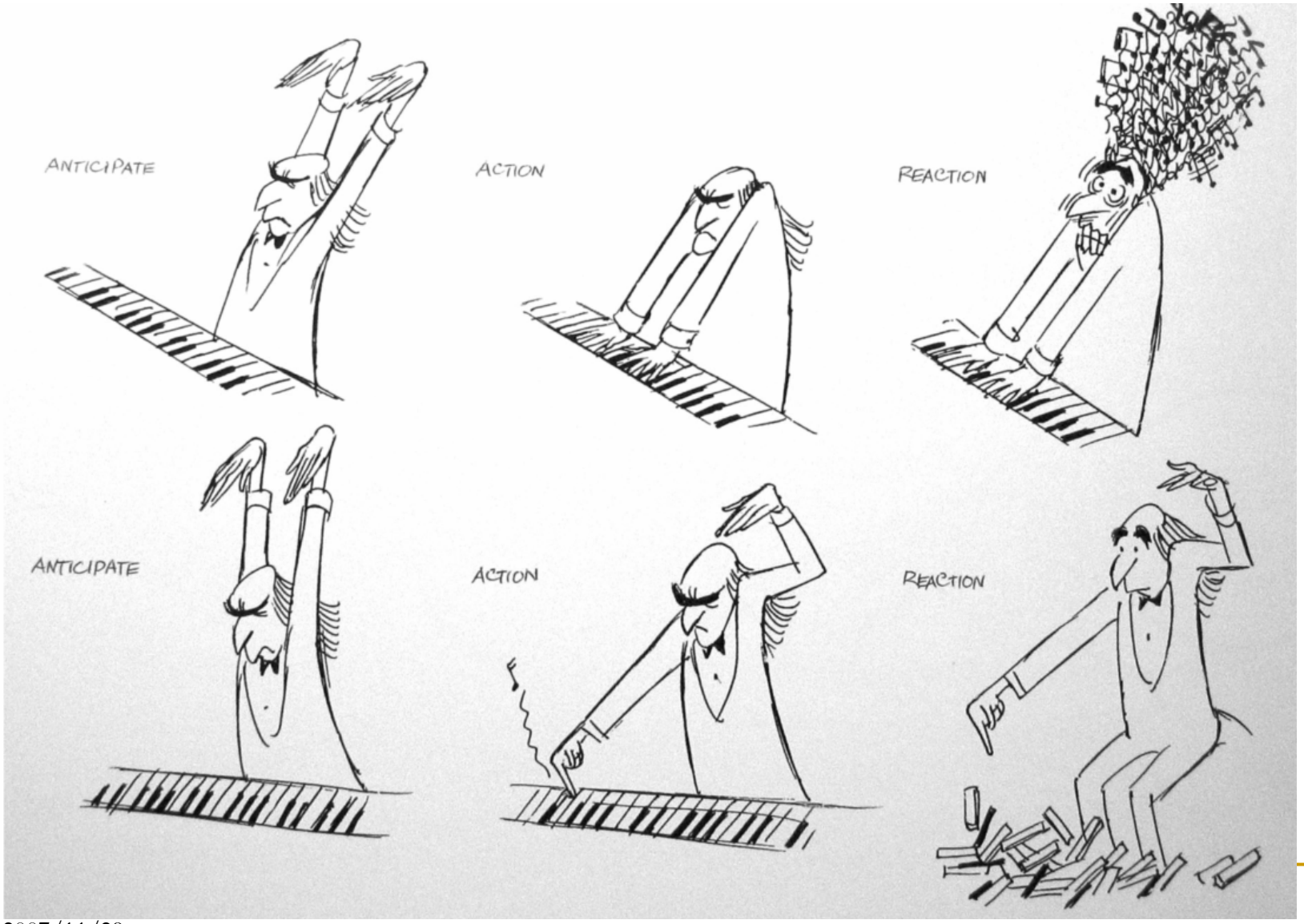
Slow In and Out



Anticipation



Anticipation



Principle of Traditional Animation – Disney –

- Squash and Stretch
- Slow In and Out
- Anticipation
- **Exaggeration**
- **Follow Through and Overlapping Action**
- **Timing**
- **Staging**
- **Straight Ahead Action and Pose-to-Pose Action**
- **Arcs**
- **Secondary Action**
- **Appeal**

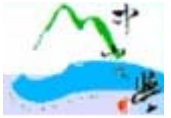
Principle of Traditional Animation – Disney –

3. Staging
   - Show viewer what to look at
   - Can do with camera angle, lights, set, motion
4. Straight ahead and pose to pose
   - SA gives spontaneity, P to P offers structure
5. Follow through and overlapping action
   - "Things don't come to a stop all at once guys."
6. Ease in and ease out
   - Can also show weight (inertia)
7. Use arcs

Principle of Traditional Animation – Disney –

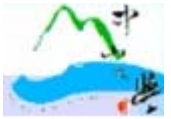
8. Secondary action
   - primary action is big gesture
   - secondary is expressions, small gestures
9. Timing
   - speed of movement conveys story
10. Exaggeration
   - extremes and most important actions
   - don't overuse or it loses effect
11. Solid Drawing
   - feeling of weight, no "twins"
12. Appeal
   - why we care
   - like charisma in an actor
   - visual interest

Computer Animation



- **Animation Pipeline**
  - 3D modeling
  - Motion specification
  - Motion simulation
  - Shading, lighting, & rendering
  - Postprocessing

國立中山大學資訊工程學系
3D電腦圖學之研究



國科會/奇景光電產學合作案－

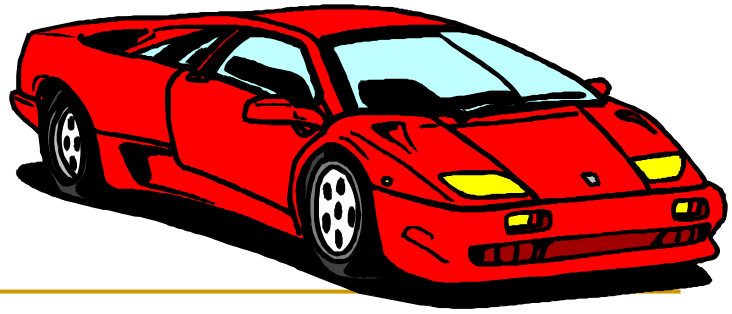
- 數位電視之三維(3D)圖形加速處理器之軟硬體技術及系統整合
  2005/08/01 ~2007/07/31

國科會

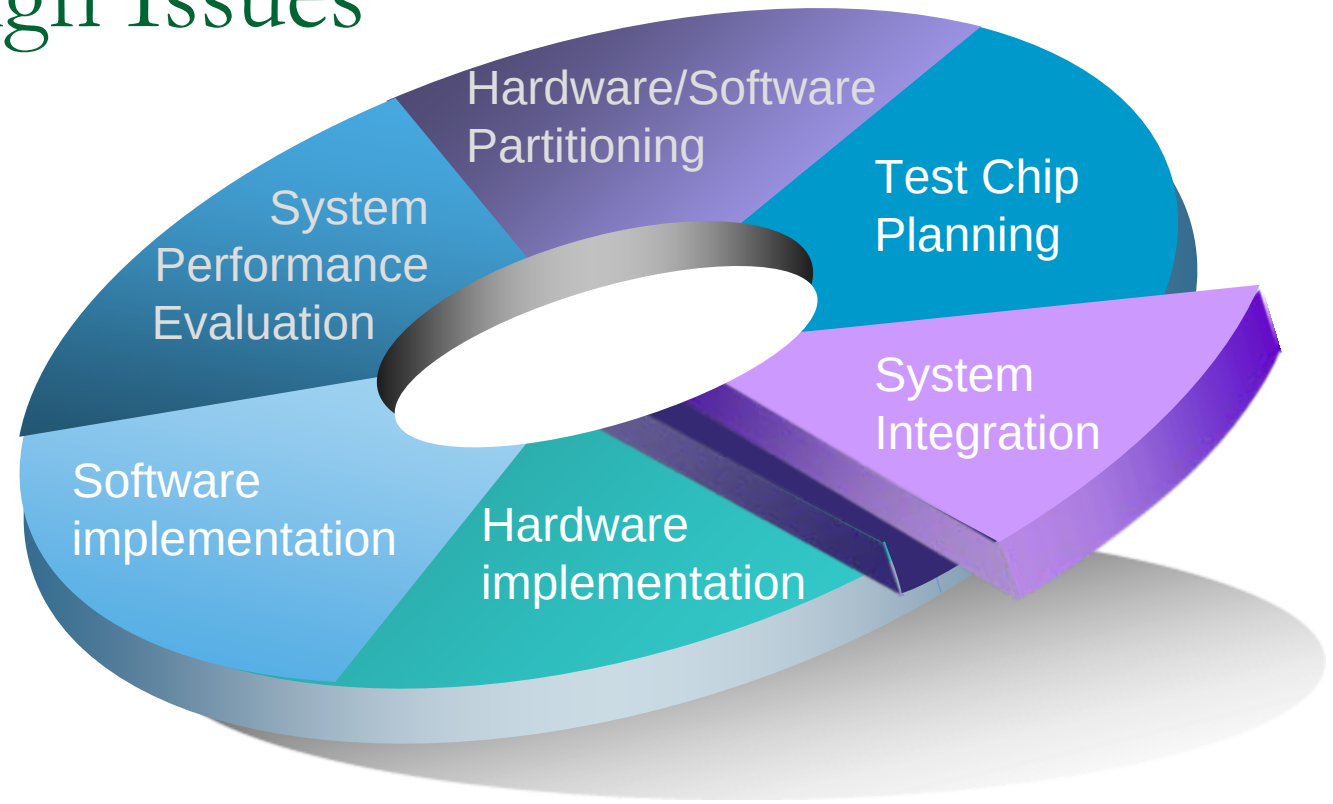
- 具有即時效能/功率監控功能的高效率可程式化
  三維電腦繪圖晶片系統：軟硬體開發及整合
  2007/08/01/2010/07/31

計　畫　簡　介

Embedded System Market

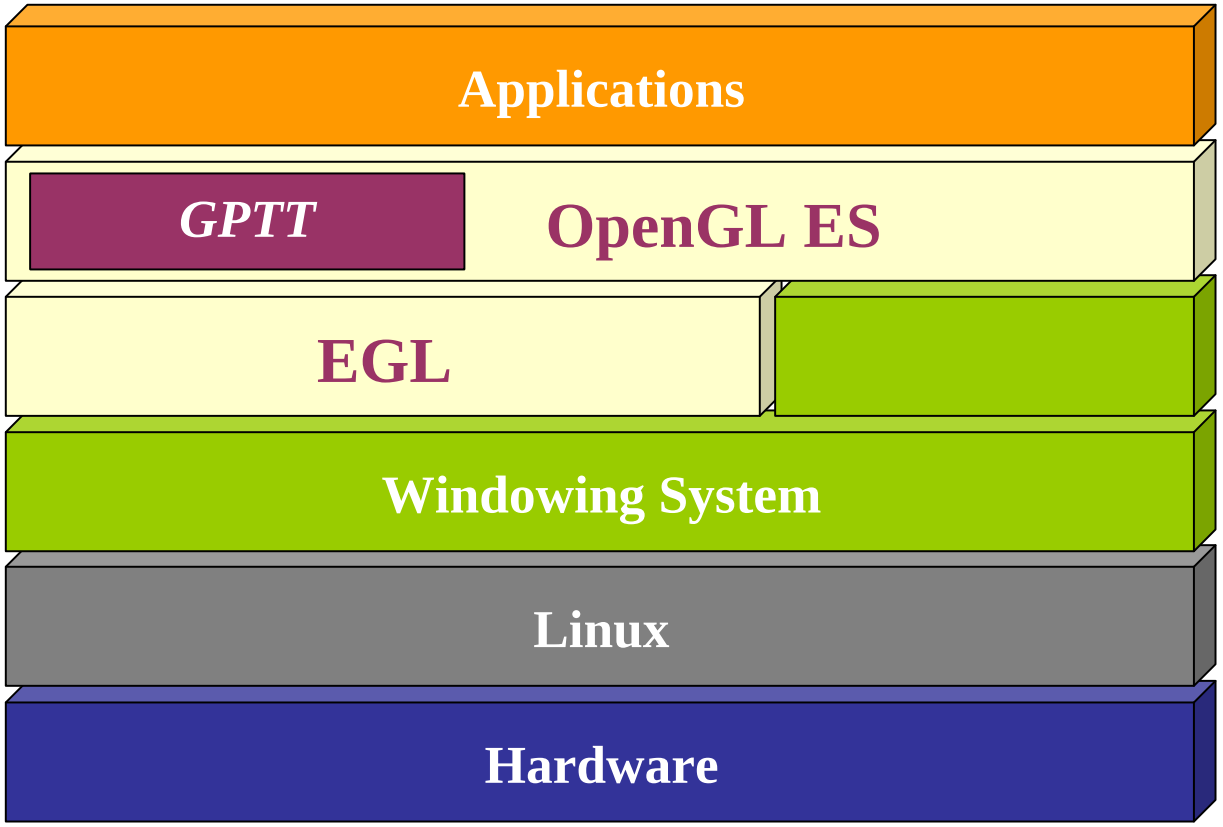


Design Issues

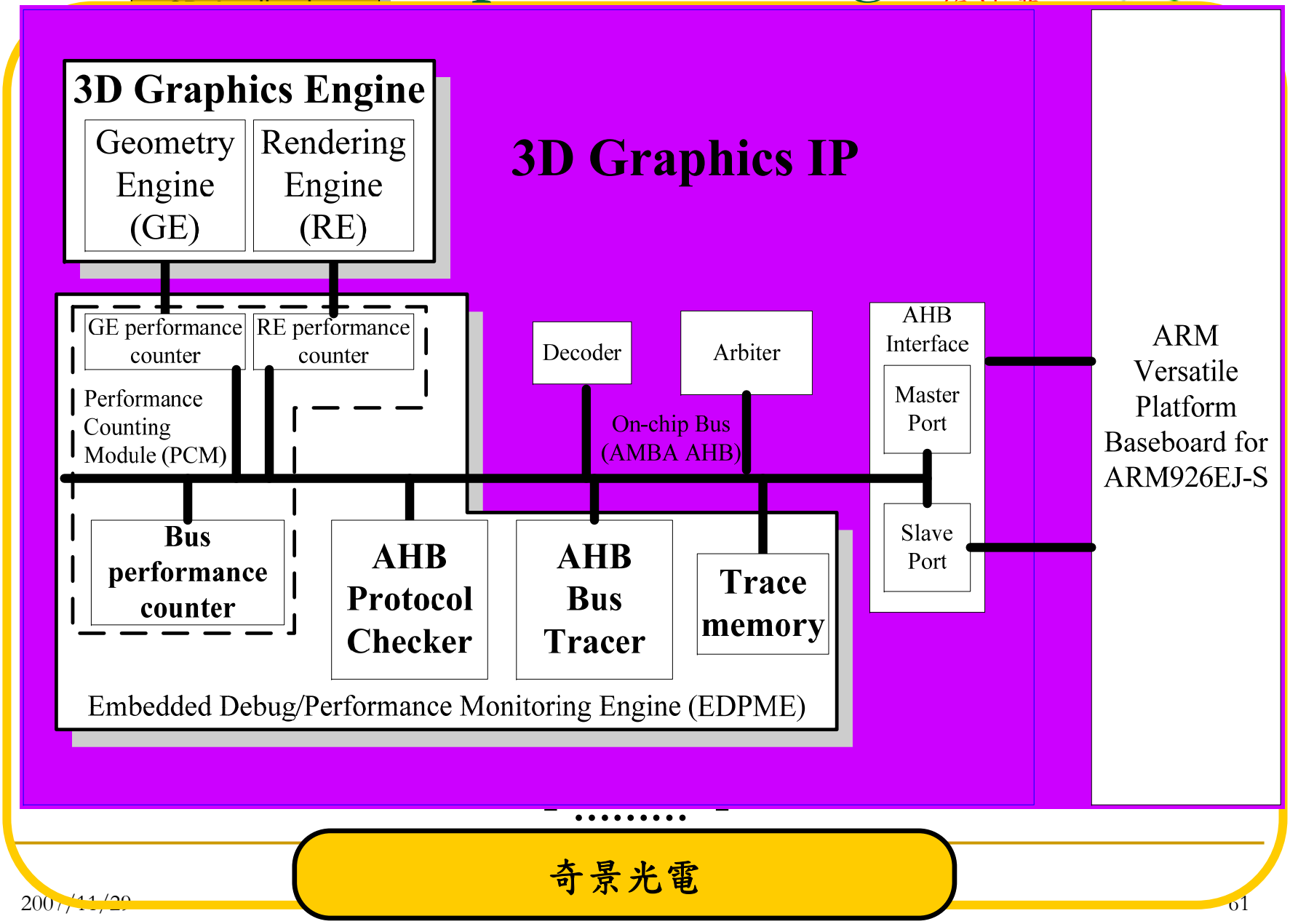


System Architecture

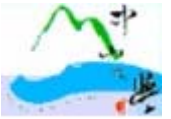
數位電視之三維(3D)圖形加速處理器之軟硬體技術及系統整合

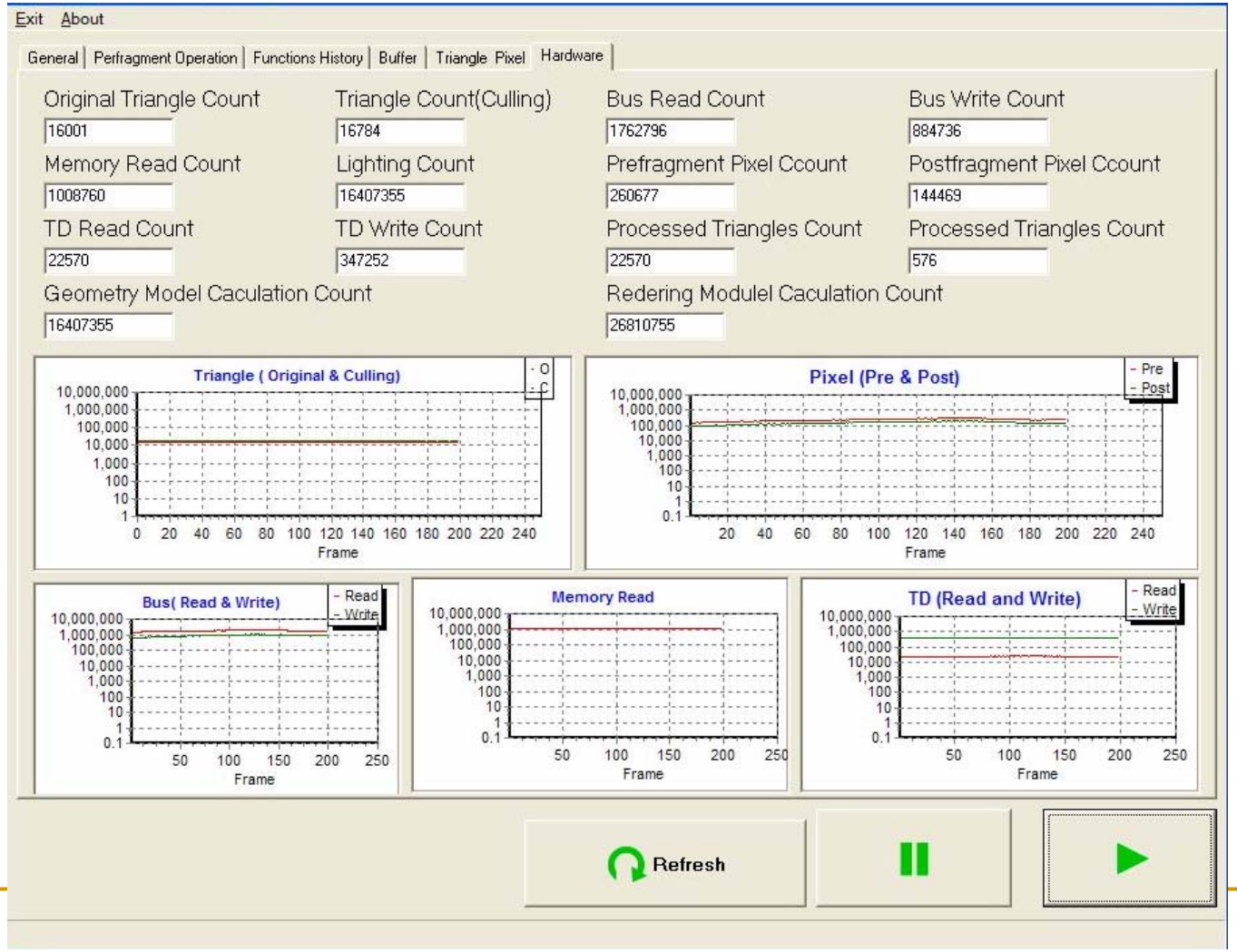


3DG SoC Chip Block Diagram

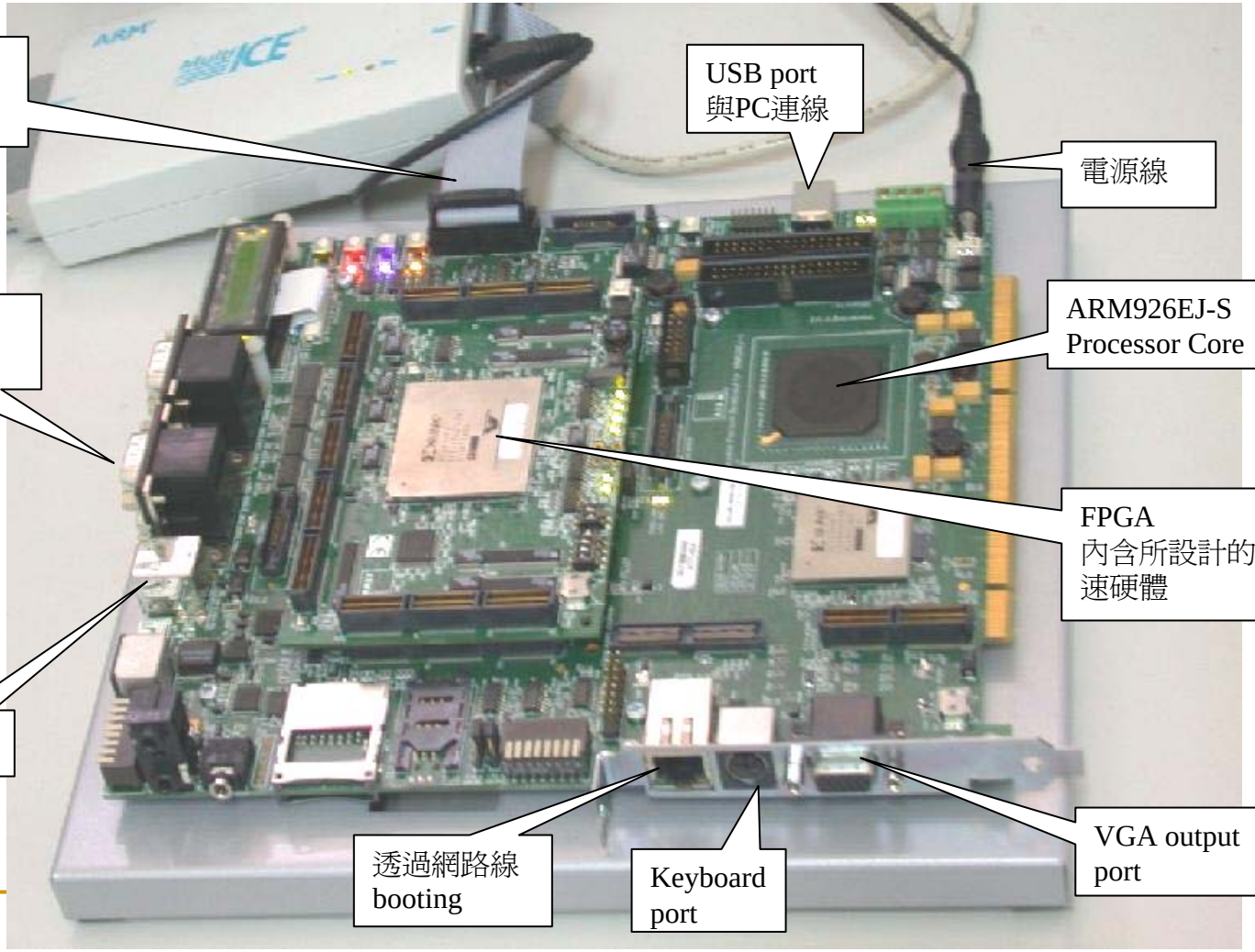


GPTT for Hardware Monitoring

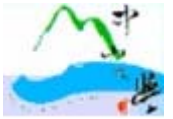




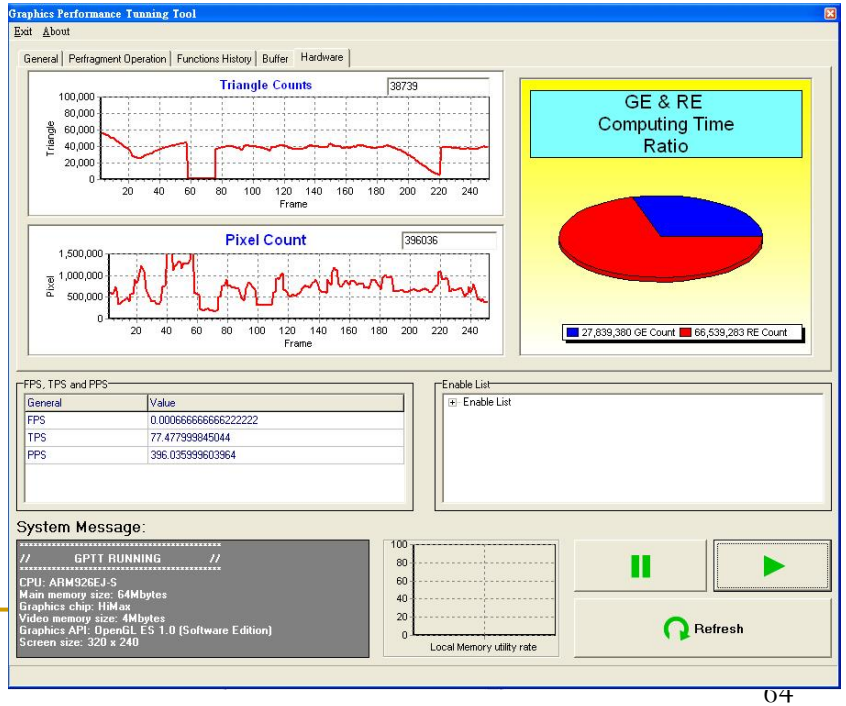
FPGA demonstration



FPGA demonstration : Angeles-Ogles + GPMT



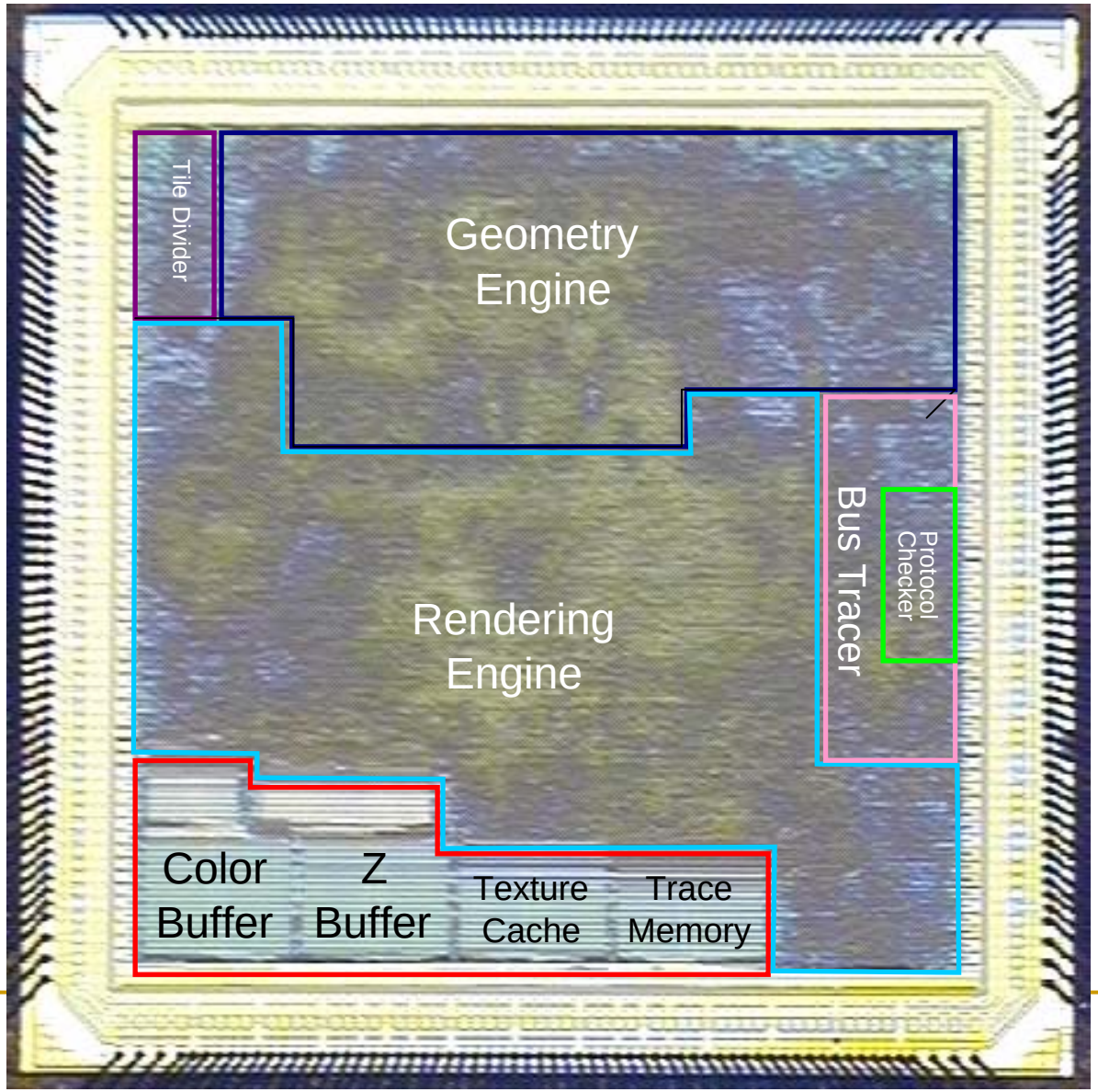
Angeles-Ogles Observatory	Graphics Performance Monitoring Tool
▪ OpenGL ES 1.1 port of the small desktop OpenGL self-running demonstration	▪ Real time display of 3DG performance information



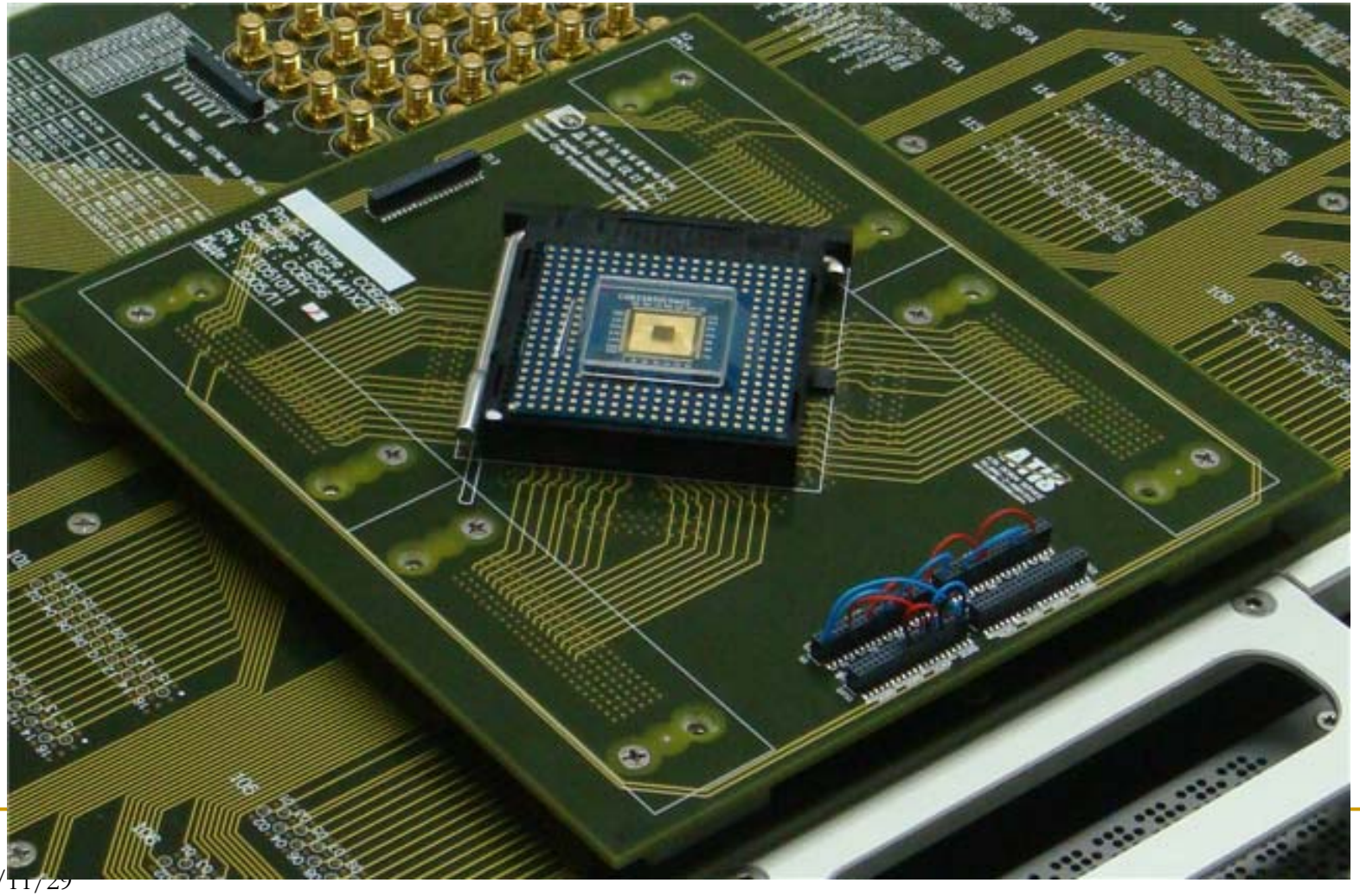
Test Chip Specification (May 2007)

Chip specification	Process Technology	TSMC 0.18 μm CMOS 1P6M
	Supply Voltage	1.8V for Core, 3.3V for I/O Cells
	Power Consumption	472.6mW (core power)
	Gate Count	986.5K gates (including SRAM and IO PAD)
	Die Size	3.96 × 3.96 mm^2
	Package	PBGA 256 pin
	Operating Frequency	139 MHz
3D graphics engine	3D Performance	8.69 Mvertices/s for GE, 278 Mpixels/s for RE, 278 Mtexels/s for Texture
	Texture Mapping	Bilinear perspective correction
	3D Operation	Fully support OpenGL-ES 1.0 functions
	Target Resolution	Up to 640×480
Embedded Debugging /Performance Monitoring Engine	Bus Tracer	Provide 4 signal/timing abstraction levels
		Real-time compression
	Performance Counting Module	Provide GE/RE/System Bus performance information, include triangle rate, pixel rate, bus idle/utilization/transfer rate and etc.
	Protocol Checker	Rule-based AHB protocol checker, total 73 rules
Software features	OpenGL-ES API for provided hardware Device driver for Linux 2.6.11 Nano-X Windowing System 3D Graphics Debugging/Performance Monitoring Tool (3DG DPM)	
Prototype	ARM Versatile Platform Baseboard for ARM926EJ-S + Logic Tile FPGA	

Chip micro photo

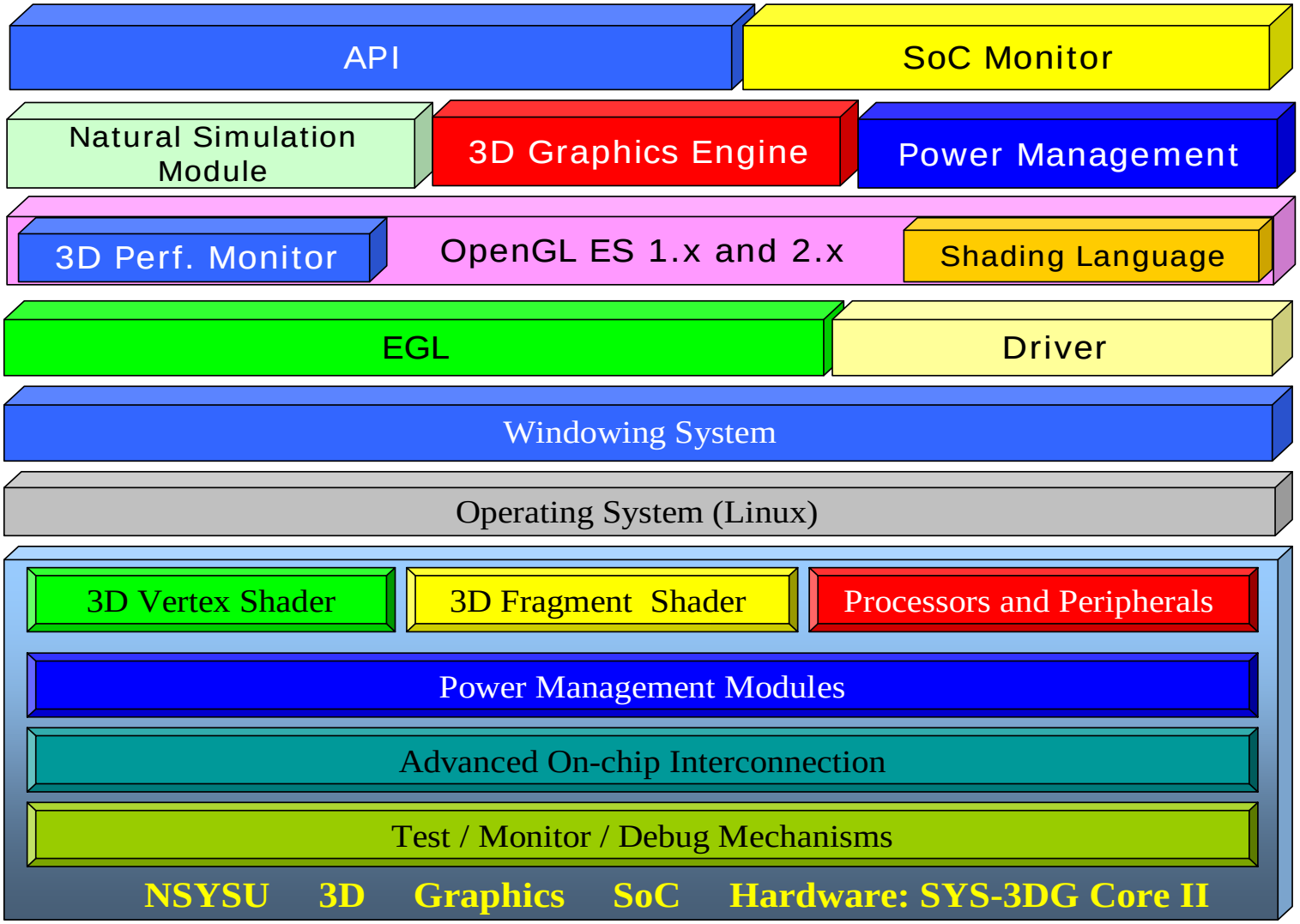


DUT Load Board (Agilent 93000)



System Architecture

具有即時效能/功率監控功能的高效率可程式化三維電腦繪圖晶片系統：軟硬體開發及整合



執行團隊


3D algorithms, Open GL ES API & OS

- 李宗南, 國立中山大學資訊工程學系
- 江明朝, 國立中山大學資訊工程學系

3D acceleration modules, vertex transformation, rendering

- 蕭勝夫, 國立中山大學資訊工程學系
- 張雲南, 國立中山大學資訊工程學系


HW/SW integration and testing

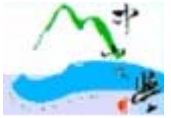
- 黃英哲, 國立中山大學資訊工程學系
- 洪敬華, 國立高雄大學電機系


支援三維電腦繪圖多媒體晶片系統多頻多電壓之電路設計

•郭可驥 國立中山大學 資訊工程學系

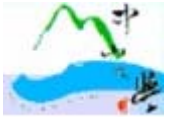
三維電腦繪圖晶片系統處理引擎之低功率系統架構設計

•鄺獻榮 國立中山大學 資訊工程學系



GPTT:Graphics Performance Tuning Tool for ARM Versatile Platform

3D Graphics Ecosystem



KHRONOS GROUP

Market demand for Handheld 3D

Great 3D Applications

Tools

GPTT

NVperfHUD

gDEBugger

Industry Cooperation

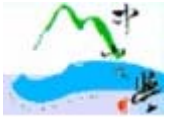
Open API Standards

OpenGL|ES

Benchmarks

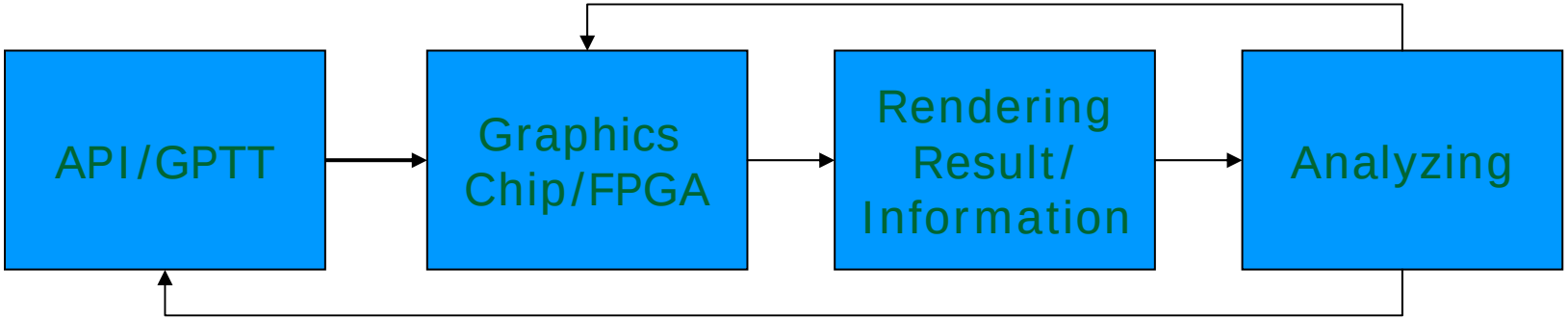
High-quality platforms

Motivation

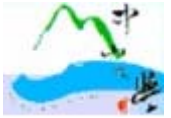


What is a graphics performance tool for?

- *Find the performance bottleneck of graphics part of your applications.*
- *Find bottleneck of your CPU*
- *Find the usage of resources*
- *Better game planning*
- *Better power management*

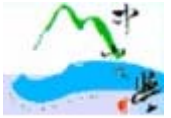


Introduction (cont.)



- Cross-platform software is difficult on embedded system
- Main problems
  - Different API
  - Different windowing system
  - Different operating system
- Results
  - More cost
  - More time
  - More work-force

GPTT Concept & Architecture

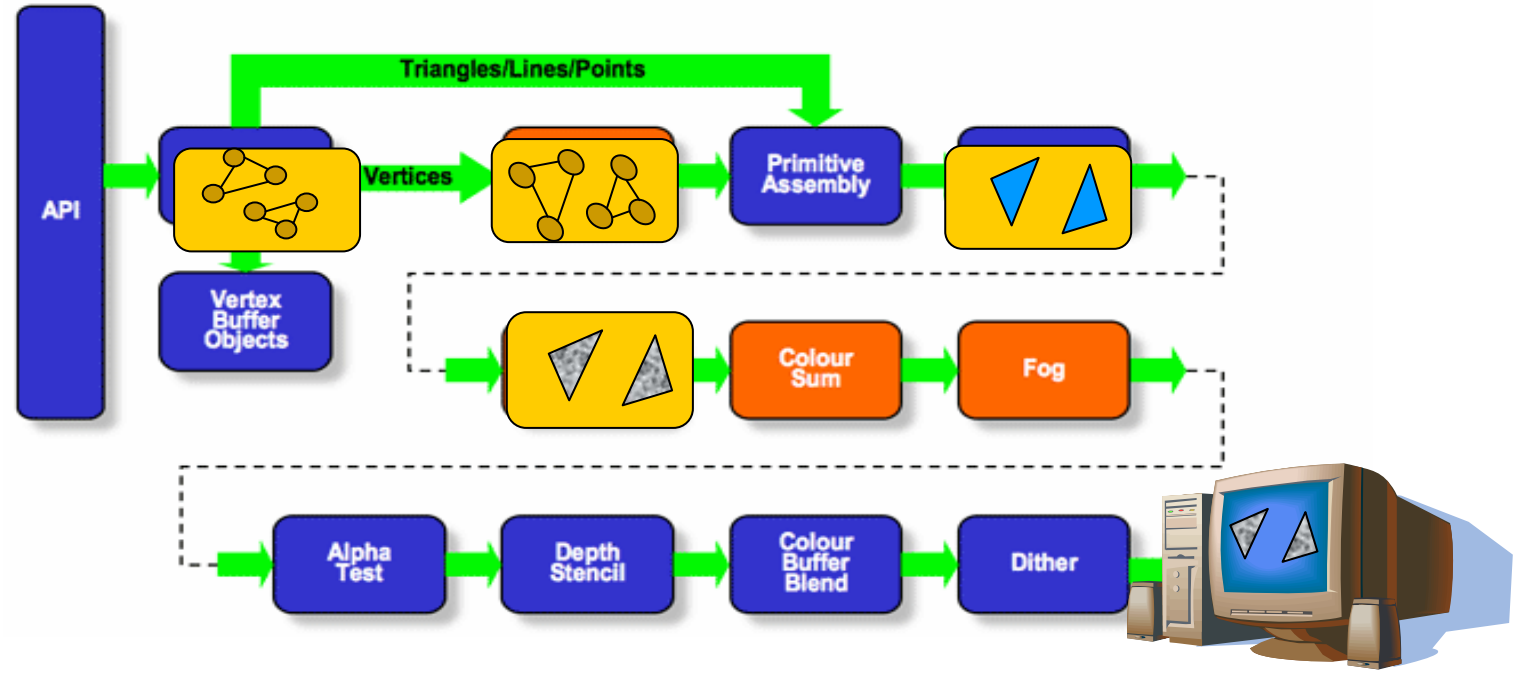


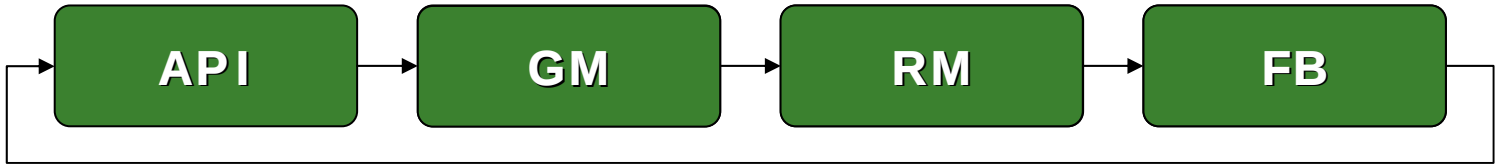
- Concept of GPTT:
  - A performance tool

    Help the game developers to find the bottleneck of applications
  - Aim at embedded systems
  - Cross-platform

    Follow the spirit of Khronos (OpenGL ES)

Rendering pipeline of OpenGL|ES





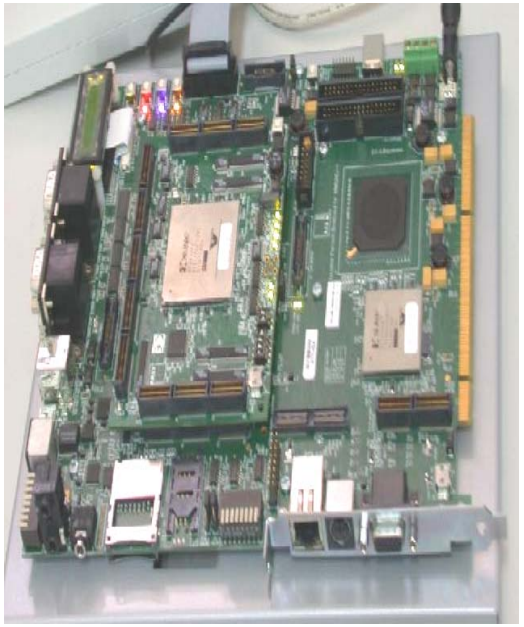
Developed environment

- OpenGL ES 1.0 (Software Edition)
- GPTT (Graphics Performance Tuning Tool)

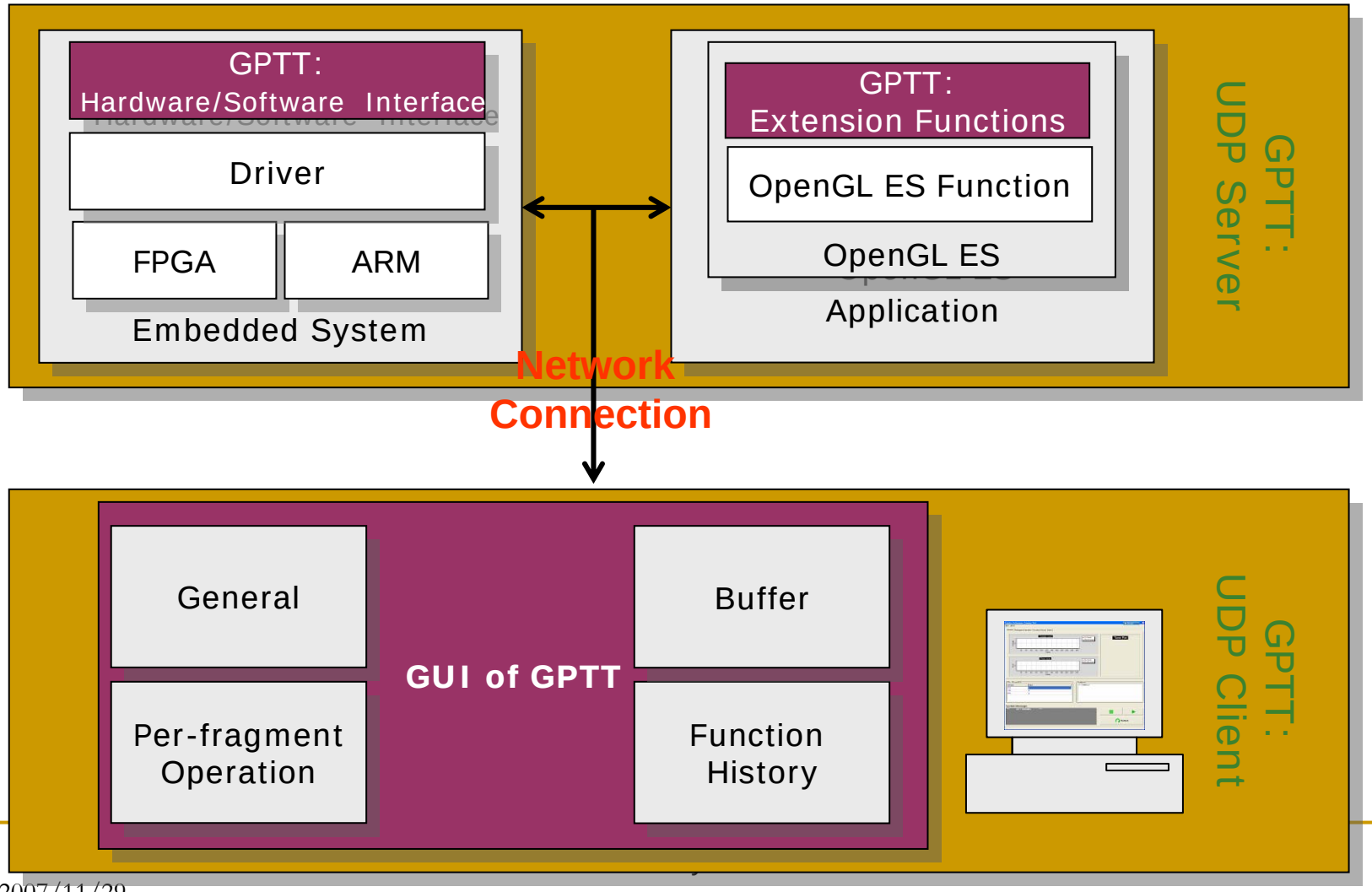
Embedded System:
- Operating System: Linux kernel 2.6
- CPU: ARM926EJ-S
- Main Memory: 64M Bytes
- Hardware : Versatile platform(FPGA)

Desktop Computer:
- Operating System: Microsoft WindowsXP
- CPU: Intel Pentium M 1.4GHz
- Main Memory: 512M Bytes

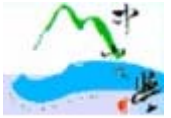
Programming Language:
- C / C++



GPTT Architecture

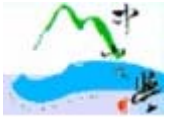


Items to be measured (software)



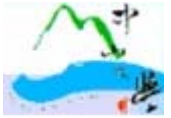
1. Primitives	12.Lighting-module Processed Time
2. Processed Primitives	13.Pixel Processed Time
3. Pixels	14.Utility Rate of Each Per-fragment
4. Processed Pixels	15.Buffer Bandwidth
5. Triangles Per Second	16.History of Functions call
6. Pixels Per Second (Full Rate)	17.Video Memory Utility Rate
7. Frames Per Second	18.GPU Idle Time (Hardware only)
8. Screen Resolution	19.Driver Idle Time (Hardware only)
9. Texture Size	20.Batch
10.Frame Processed Time	21.# of Draw Functions Call
11.Transformation Processed Time	

Items to be measured (FPGA)



1. Original triangle count	8. RM cycle count (time)
2. GM processed triangle count	9. RM bus read
3. GM memory read	10. RM bus write
4. TD read count	11.Pre processed pixel count
5. TD write count	12.Post processed pixel count
6. GM lighting count	13.RM processed triangle count
7. GM cycle count (time)	14.RM processed tile count

Extension functions definition

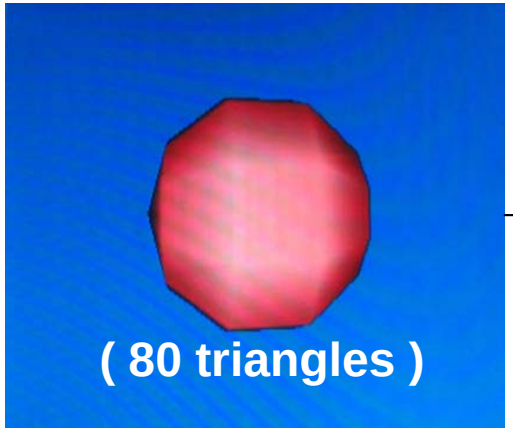


- glEnable / glDisable
  - *GL_PERFORMANCE_CHECK*
- glSetIP
- glRecordFrame
- glPerformanceReport / glPerformanceReportNetwork
- glRecordEnable / glRecordDisable
  - *GL_ALL, GL_PRIMITIVE, GL_PROCESSED_TIME*
  - *GL_GENERAL, GL_PER_FRAGMENT_UTILITY_RATE*
  - *GL_BUFFER_R_W*
  - *GL_HISTORY_OF_FUNCTION*

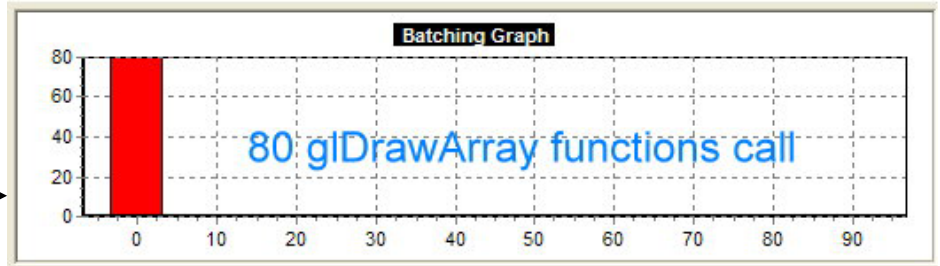
How to utilize

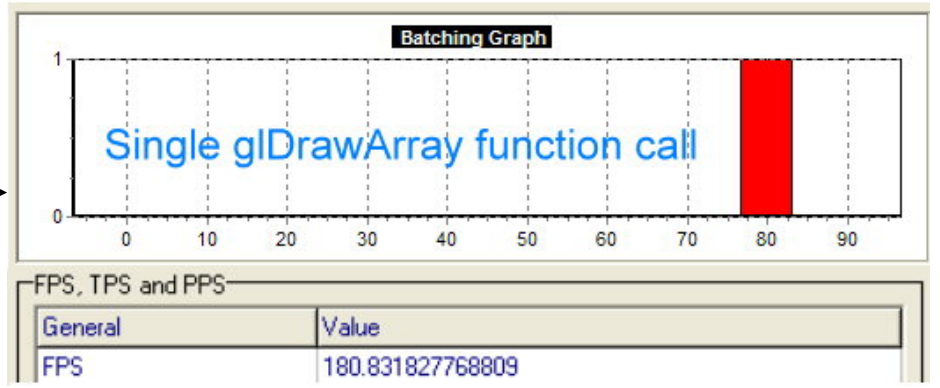
```
glInit()
{
        //Set the Ip address and Port number
        glSetIP("127.0.0.1", 8777);

        //Enable  GPTT
        glEnable(GL_PERFORMANCE_CHECK);
        //Enable All of measurements
        glRecordEnable(GL_ALL);

        … // Other OpenGL ES Code
}

glDrawScreen()
{
        … // OpenGL ES Code

        //Choose the text mode or network mode report
        glPerformanceReportNetwork();
        SwapBuffer();
}
```

How to analyze (Software)

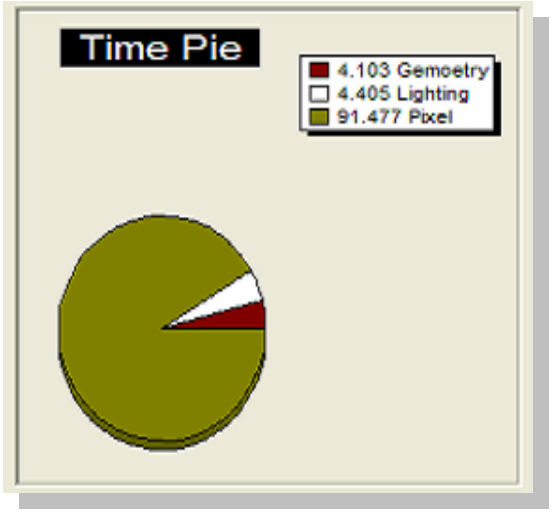


Screenshot of Ball example





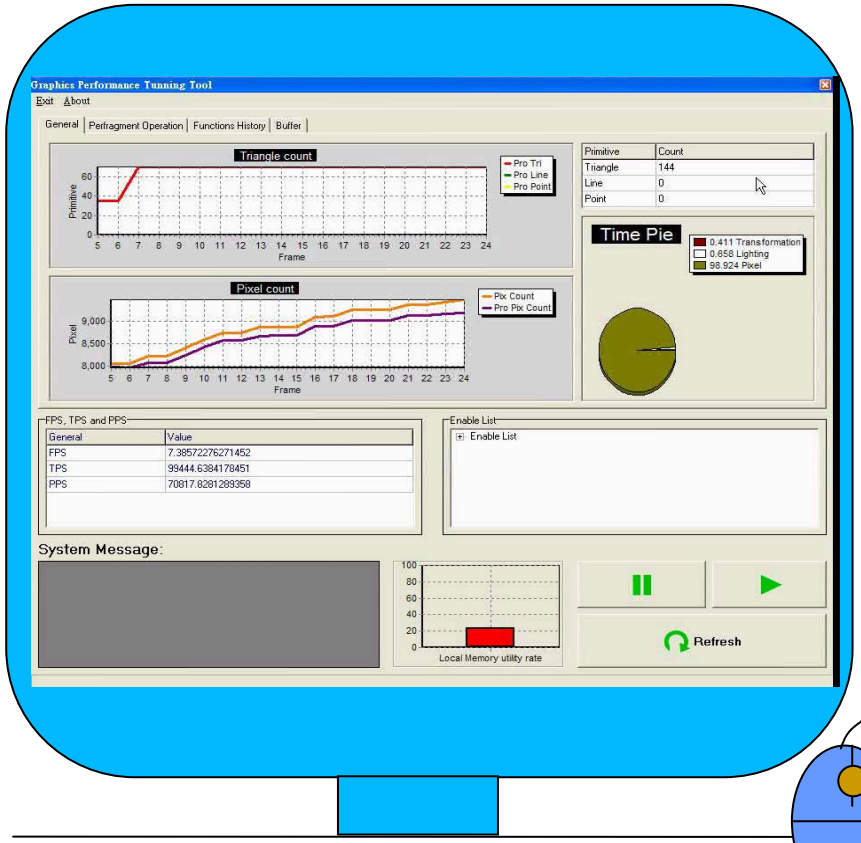
How to analyze (Software)



Solution:
Using Light-map
to improve
the performance

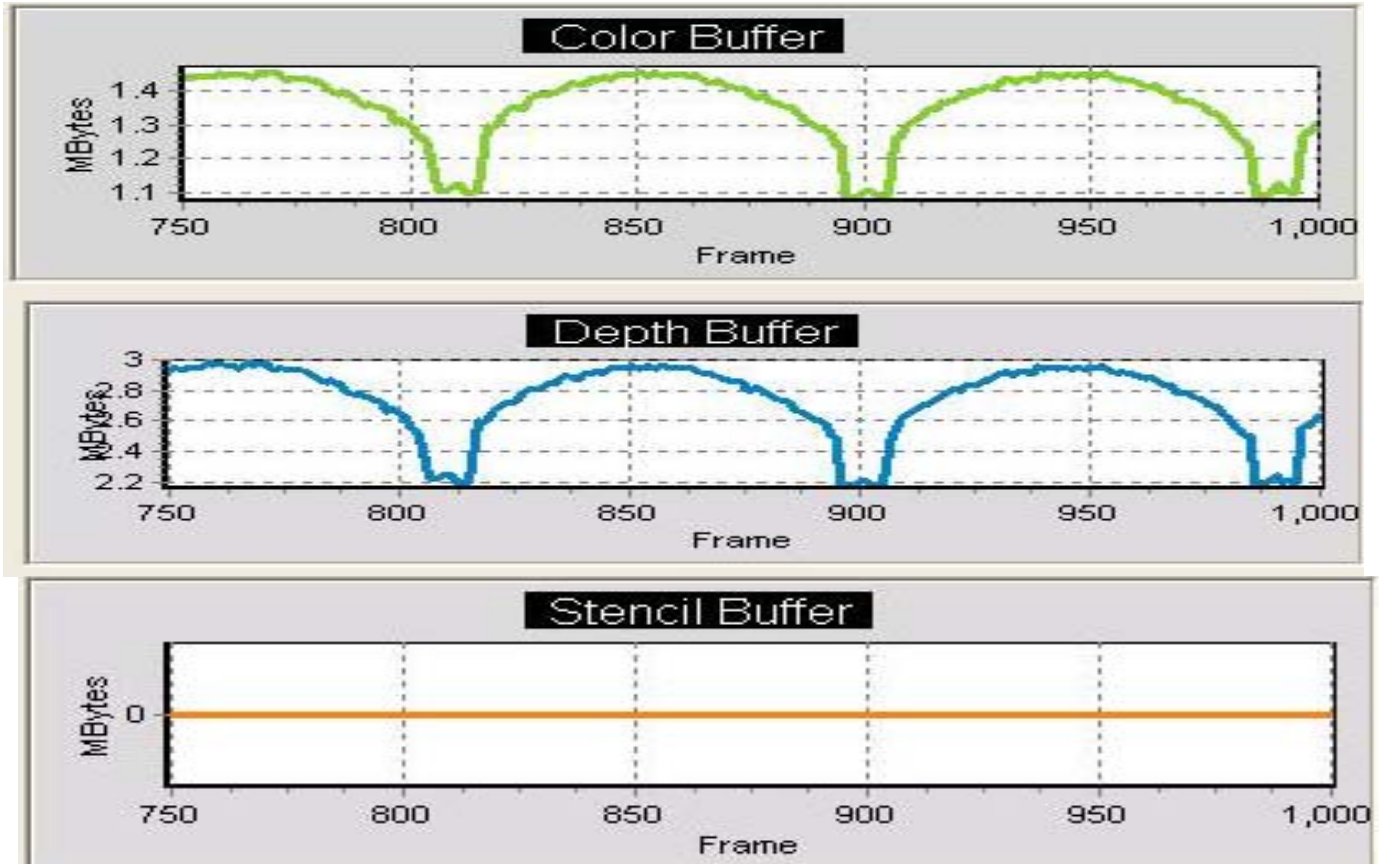
Screenshot of benchmark (Software)

Himax logo

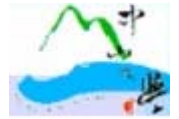


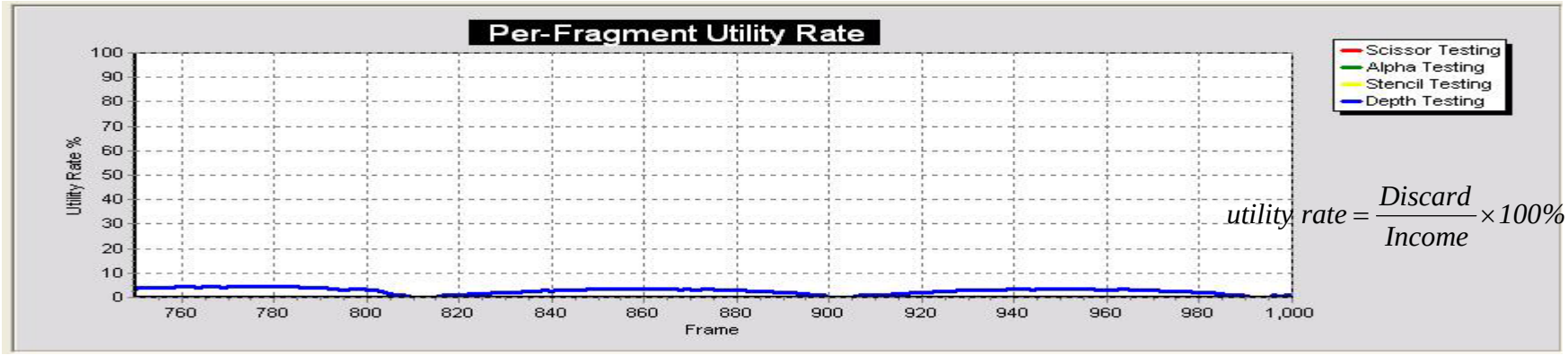
Performance results

Buffer bandwidth

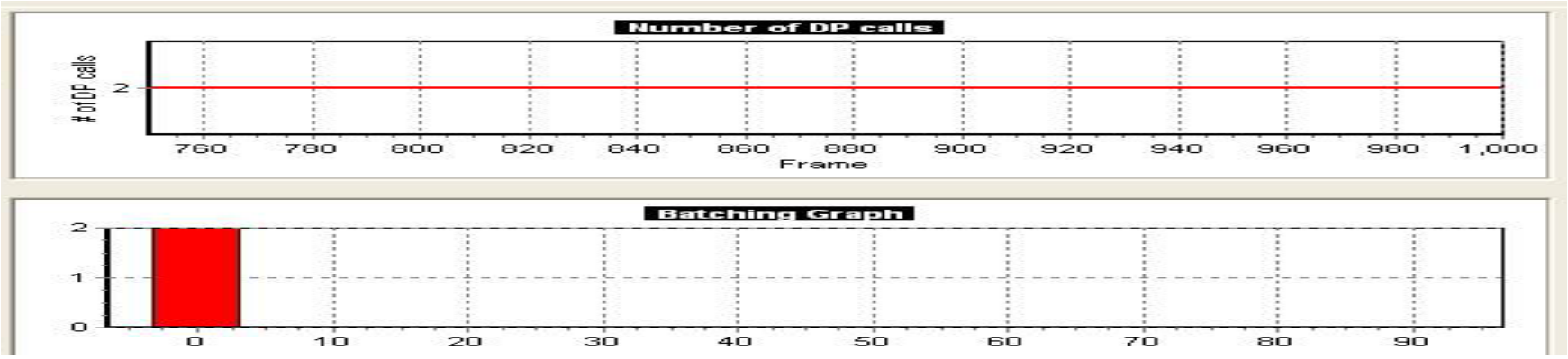


Per-fragment utility rate





$$utility\ rate = \frac{Discard}{Income} \times 100\%$$



Function history

Graphics Performance Tunning Tool

Exit About

General | Perfragment Operation | Functions History | Buffer

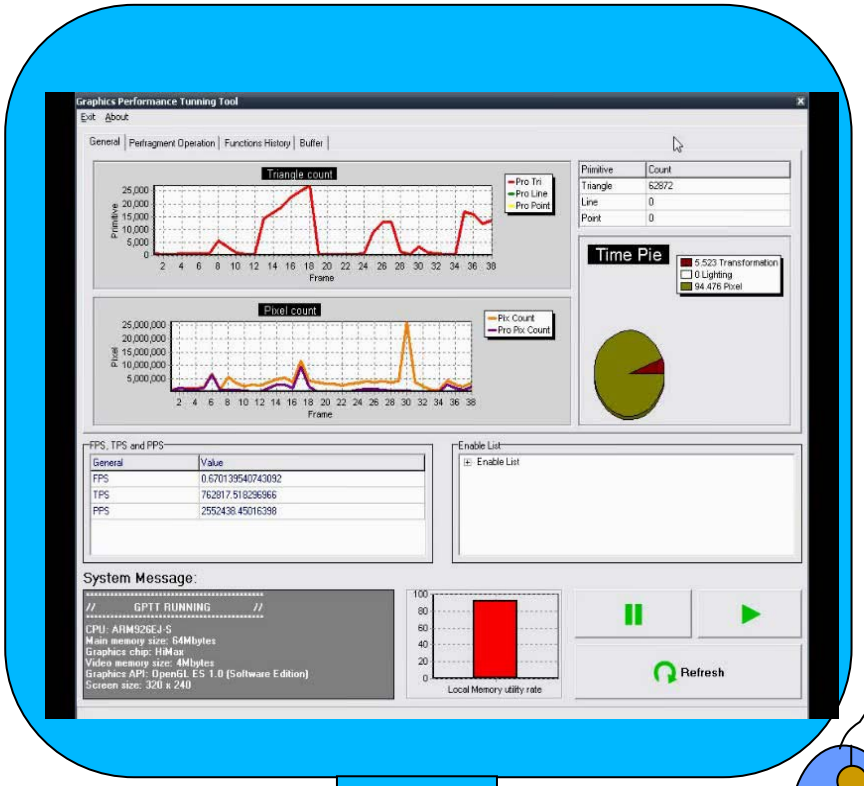
Function name	%	# of Calls
glBlendFunc	0.00172271...	1
glClear	1.72271223...	1000
glClearColor	0	0
glClearColorx	1.72271223...	1000
glClearDepthf	0	0
glClearDepthx	1.72271223...	1000
glClearStencil	1.72271223...	1000
glClientActiveTexture	0	0
glColor4f	0	0
glColor4x	3.44542447...	2000
glColorMask	0.00172271...	1
glColorPointer	0	0
glCompressedTexImage2D	0	0
glCompressedTexSubIma...	0	0
glCopyTexImage2D	0	0
glCopyTexSubImage2D	0	0
glCullFace	1.72443495...	1001
glDeleteTextures	0	0
glDepthFunc	1.72443495...	1001
glDepthMask	0.00172271...	1
glDepthRangef	0	0

FPS, TPS and PPS

General	Value
FPS	179.98559609308
TPS	116035.421470259
PPS	1993512.60646373

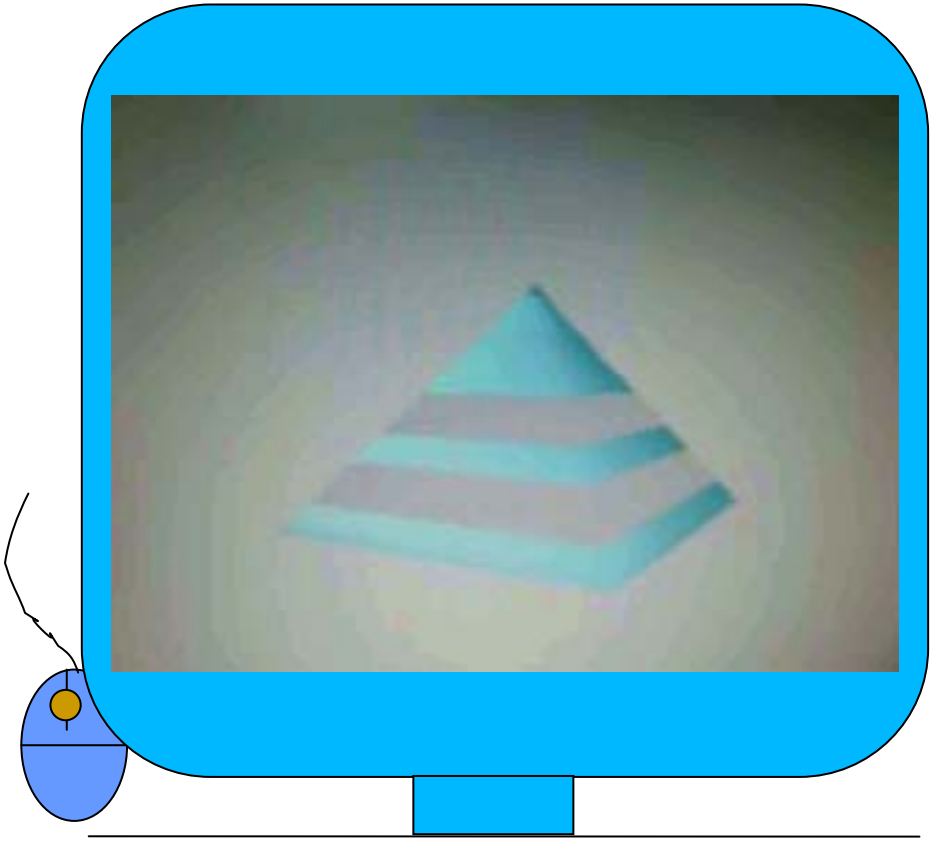
Screenshot of benchmark (Software)

Angels-ogals

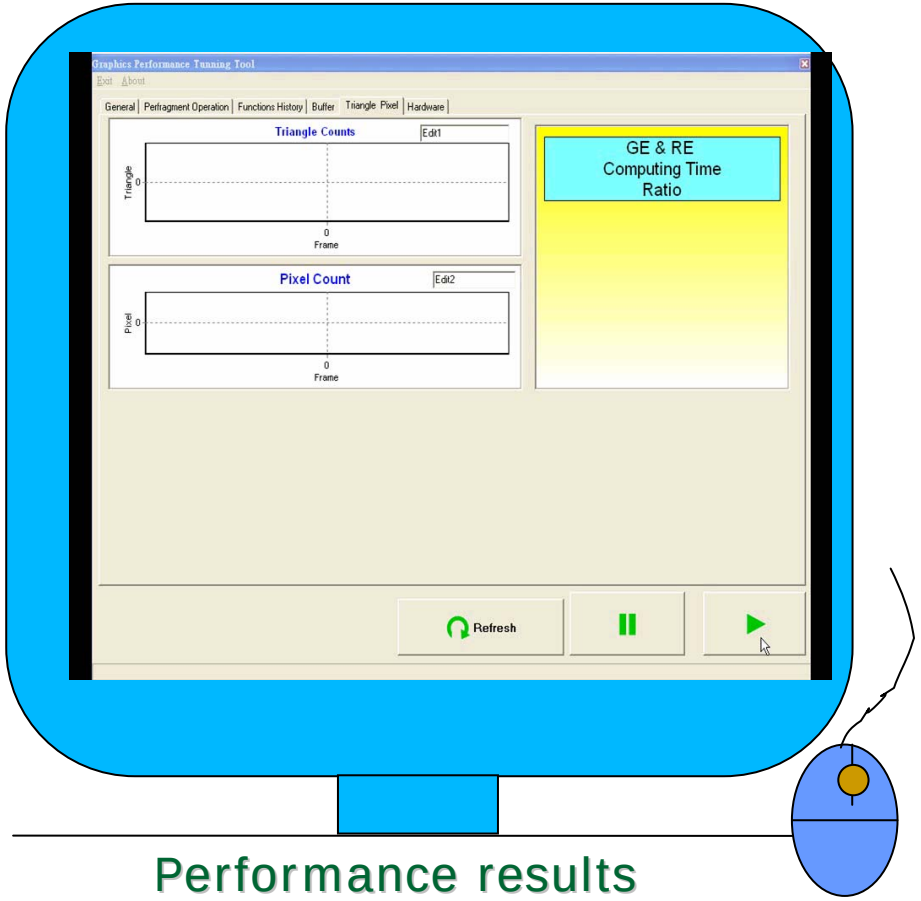


Performance results

Screenshot of benchmark (FPGA)



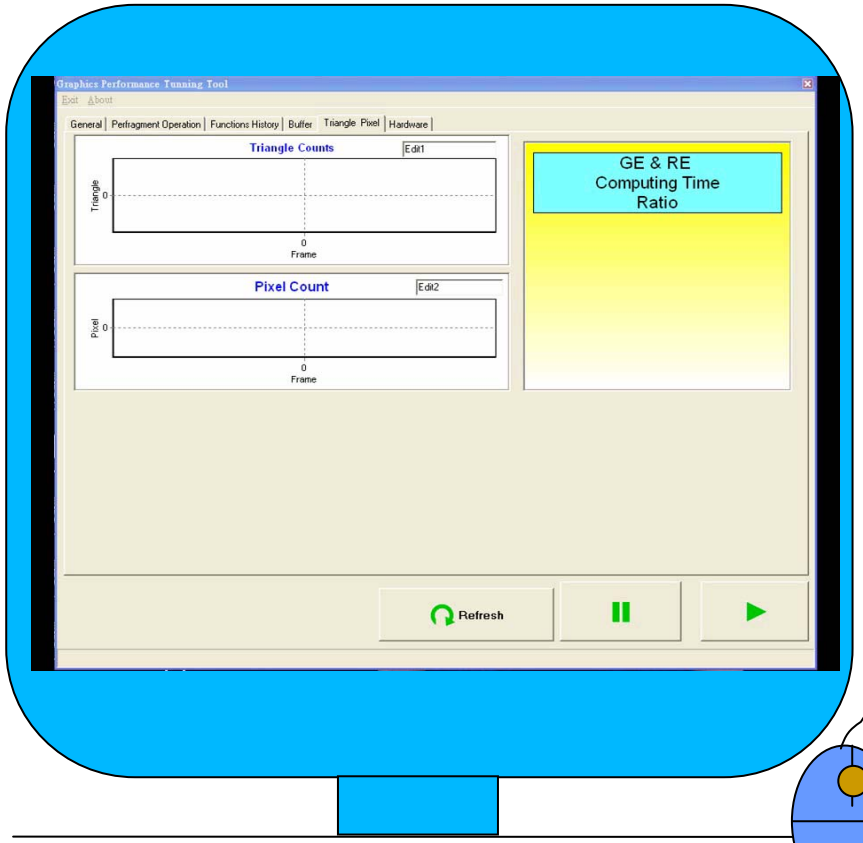
Himax logo



Performance results

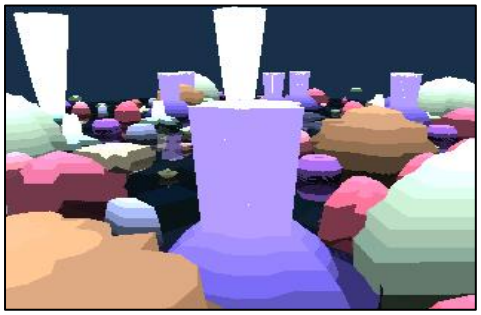
Screenshot of benchmark (FPGA)

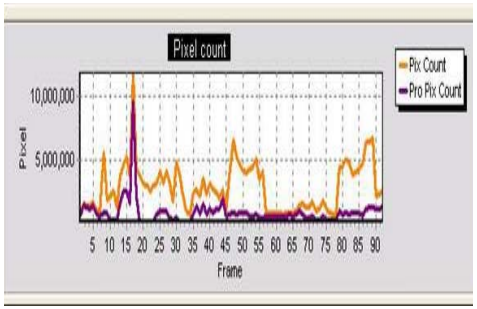
Angels-ogals

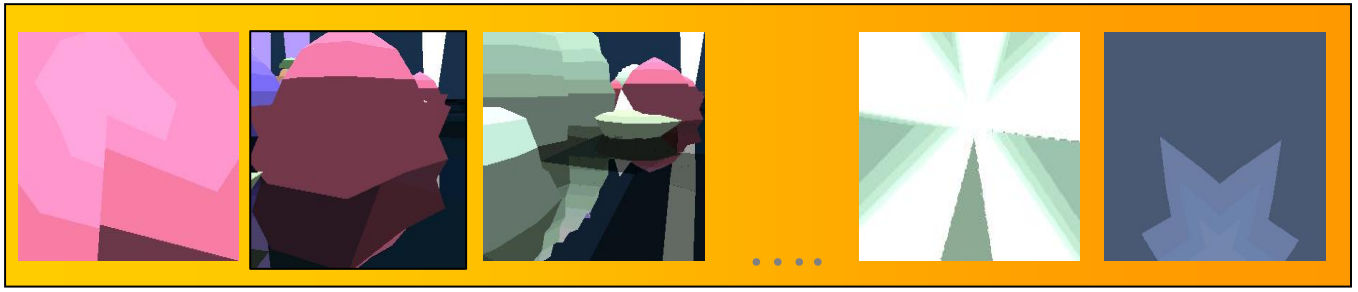


Performance results

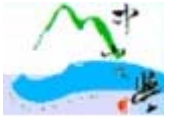
Screenshot of benchmark



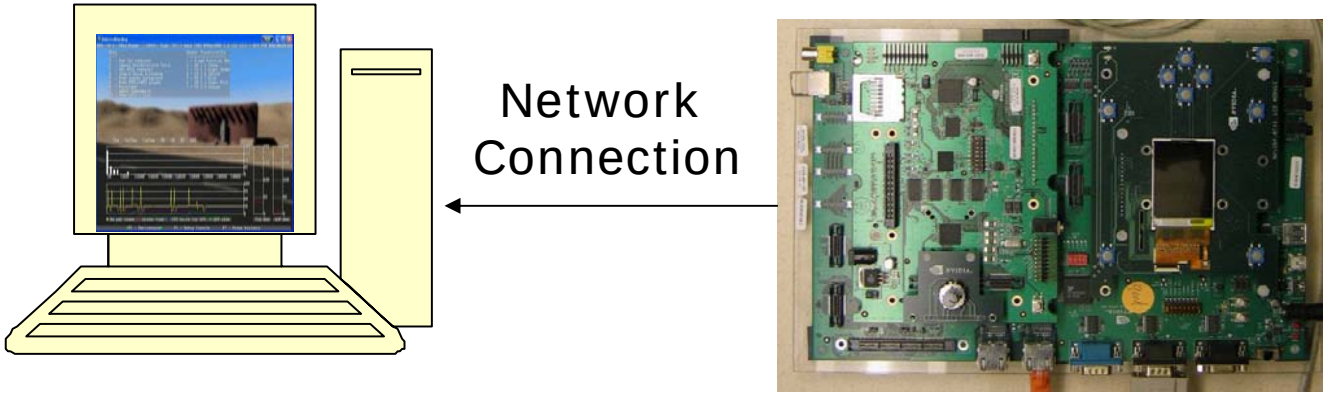




Comparison



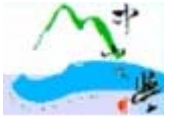
- NVperfHUD ES and gDEBugger ES



NVperfHUD ES

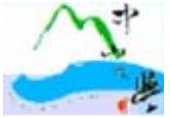
GL ES 3D Application/ GL ES Driver

GPTT vs. NVperfHUD ES



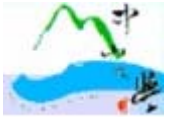
- The same characteristics
  - For embedded system
  - Base on OpenGL | ES API
  - Network connection

GPTT vs. NVperfHUD ES



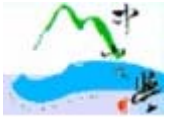
- Main Differences
  - Tool:
    - GPTT: *Define the tool to be the extension function*
    - PerfHDU ES: *Embed the tool into the driver*
  - Current developed platform:
    - GPTT: ARM 926EJ-S
    - PerfHDU ES: GoForce HW devkit
  - Cross Platform:
    - GPTT: yes
    - PerfHDU ES: no (ATI/XGI)

GPTT vs. NVperfHUD ES



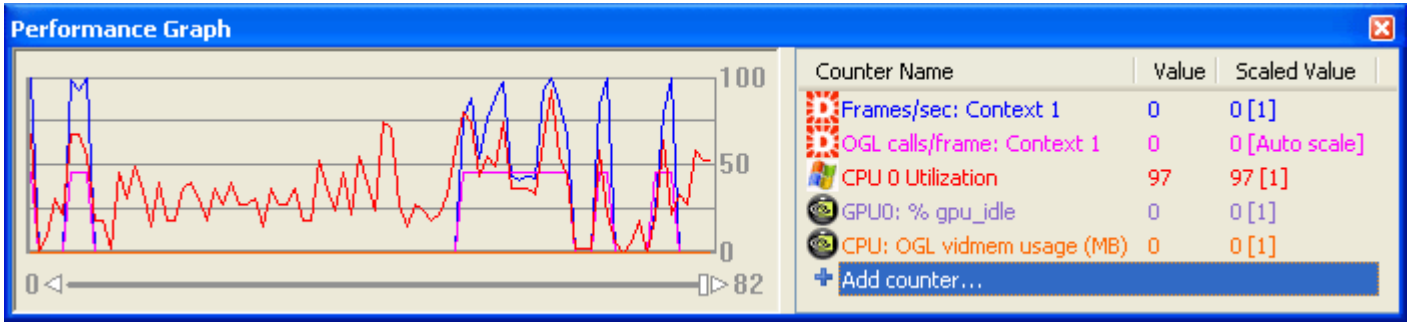
- Embed the tool into the driver(NVperfHUD ES)
  - Non-standard & Non-publication (*NVIDIA*)
  - Different company may define different measurement (gDebuger and NVperfHDU)
  - Developer needs to learn more... about the tool
- Define the tool to be the extension functions
  - Standard & Publication (*Extension function of **GLES***)
  - Developer just needs to understand how to use these standard functions

gDEBugger ES

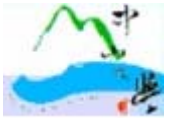


Displays performance counters graphs of:

- gDEBugger: Frames / sec, # OGL function calls or # Texture objects
- Win32: CPU utilization, Virtual memory usage,…
- ATI/NVIDIA: % GPU idle, % Driver waiting , # Pre-cull vertices, …

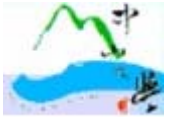


Comparison



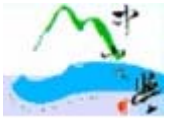
	GPTT	*NVperfHUD ES*	*gDEBugger ES*
Target	Embedded Sys.	Embedded Sys.	Embedded Sys.
API	OpenGL ES	OpenGL ES	OpenGL ES
Method	Extension functions	NVIDIA (Driver)	graphicREMEDY
Cross-Platform	✓	✗ (ATI/ XGI)	?✗
Network	✓	✓	?

Conclusions



- Introduction of computer animation
- Introduction of computer graphics
- Graphics H/S experiences at NSYSU
- Presentation of GPTT

References



- Thomas Funkhouser, Class notes, Princeton University
- Barbara Meier, Introduction to Computer Animation, Class notes
- Alan Watt, 3D computer Graphics, Addison-Wesley, 2000

Question & Suggestion