Lecture 1:

Introduction / Course Overview

EEC 277, Graphics Architecture

John Owens, UC Davis, Winter 2017

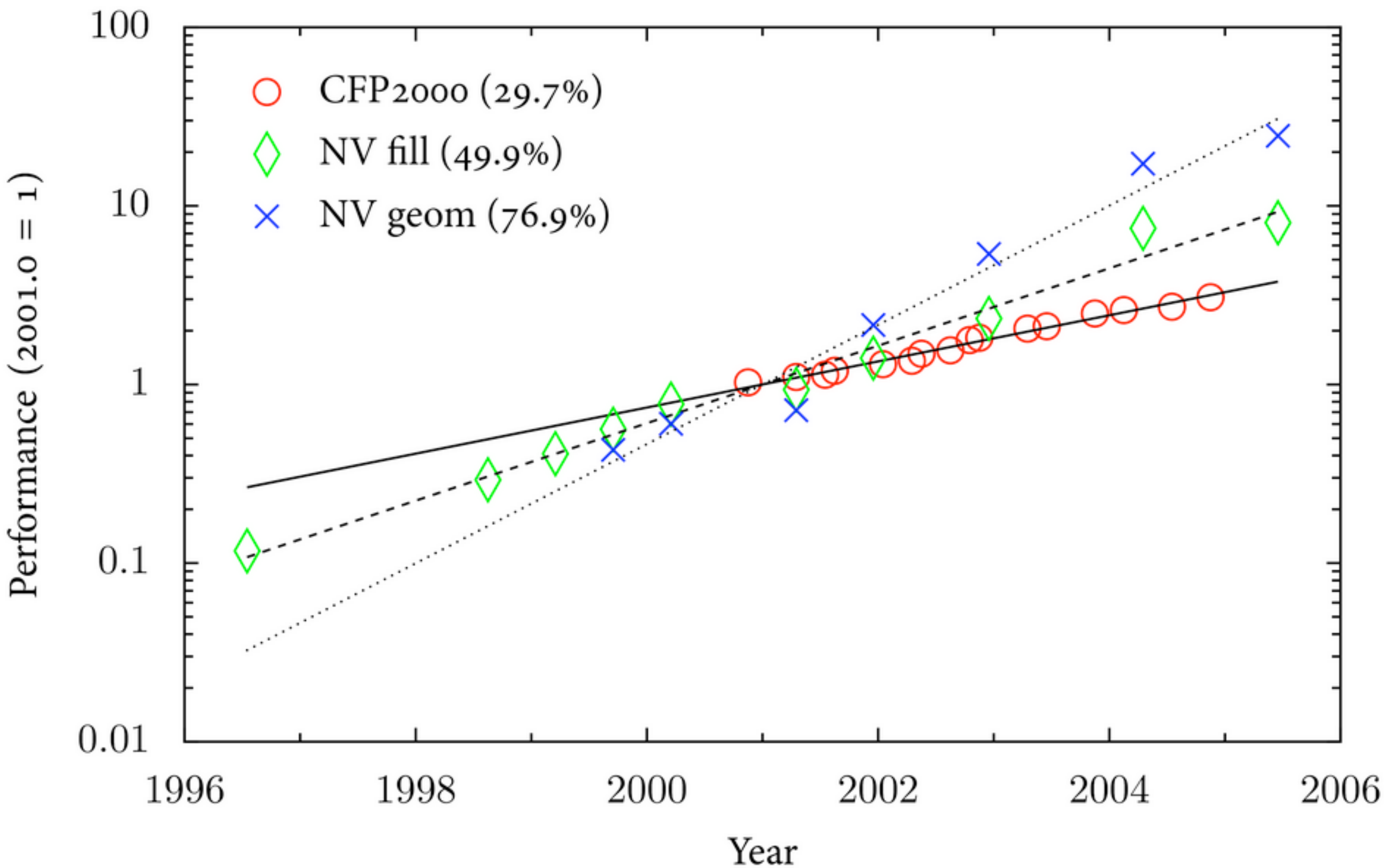
Bio

- John Owens <jowens@ece>
- Professor, ECE
  - Kemper 3175
  - Office hours: This week, T 12–1p (Kemper 3175); future Th 12–1 (Kemper 3175)
- Graduate research:
  - Imagine Stream Processor
  - Dissertation work: graphics on a stream architecture
  - Also affiliated with Flash Graphics hardware group
- Current research:
  - GPU computing
  - Programming systems for graphics hardware
    - Graphics
    - Not graphics

Lecture 1: Introduction / Overview

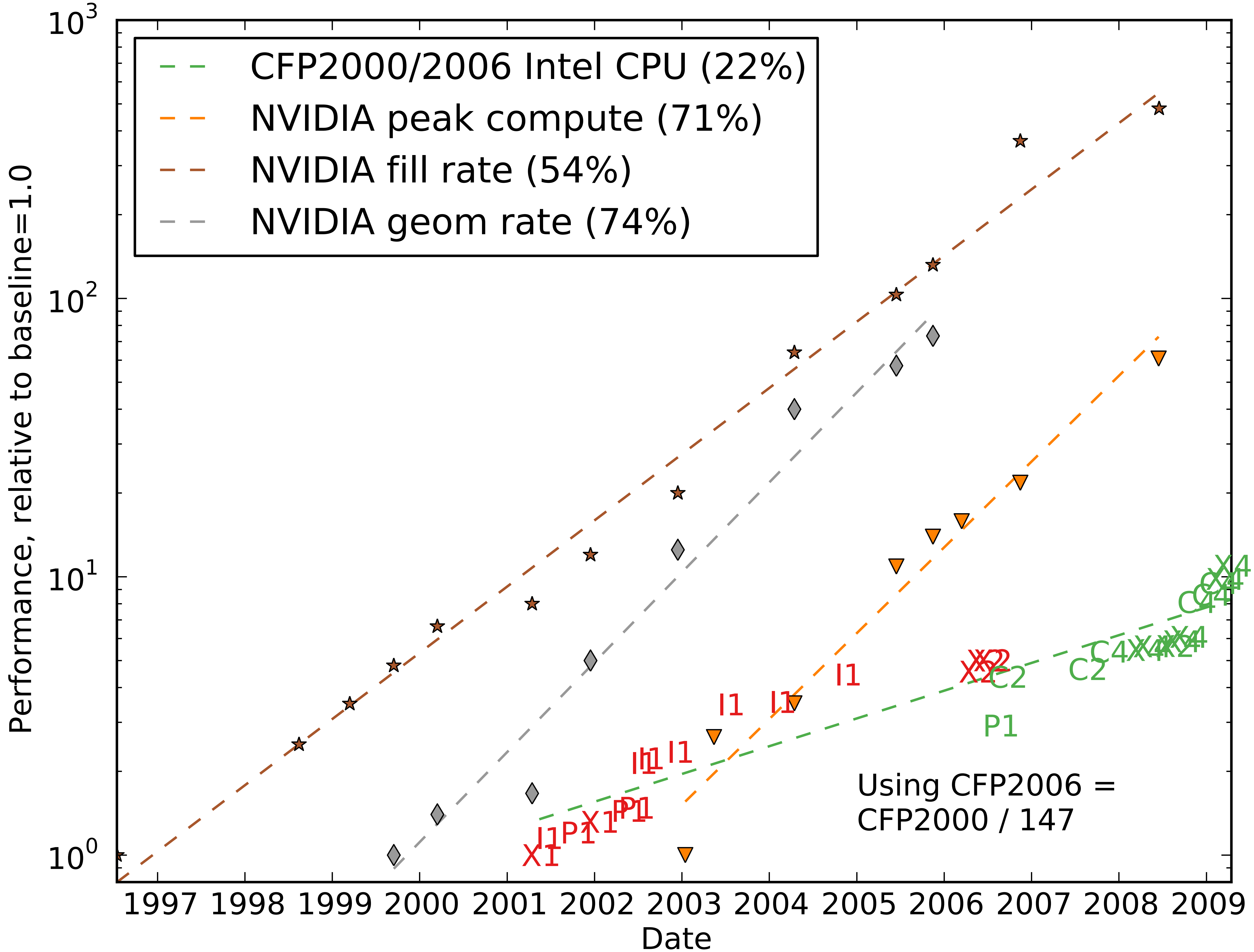
- Why graphics hardware?
- Generations of graphics hardware
- Class outline
- Administrivia

Why Graphics HW? CPU vs. GPU

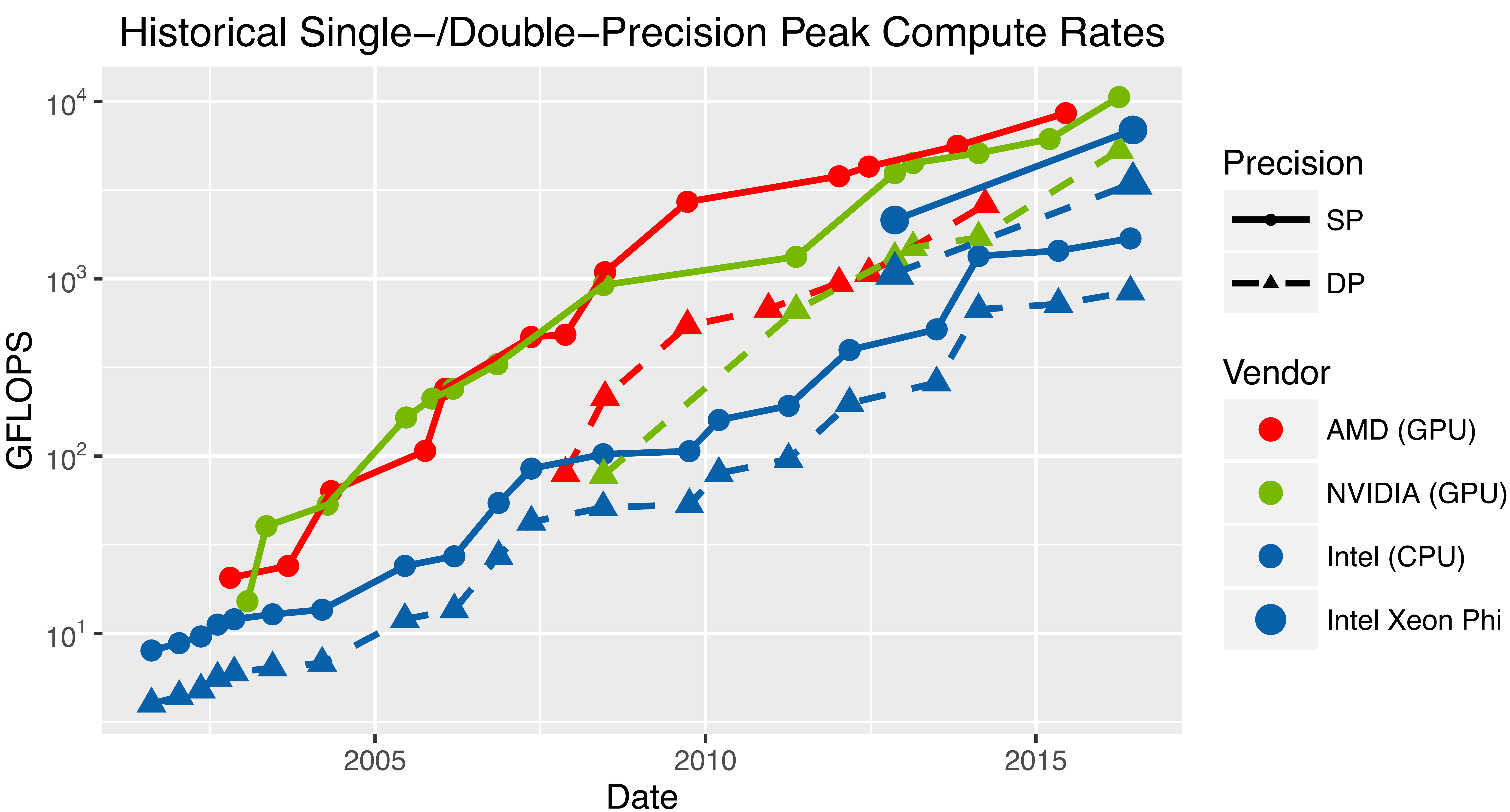


- Historically superior performance increases to CPU
- Also adding features!

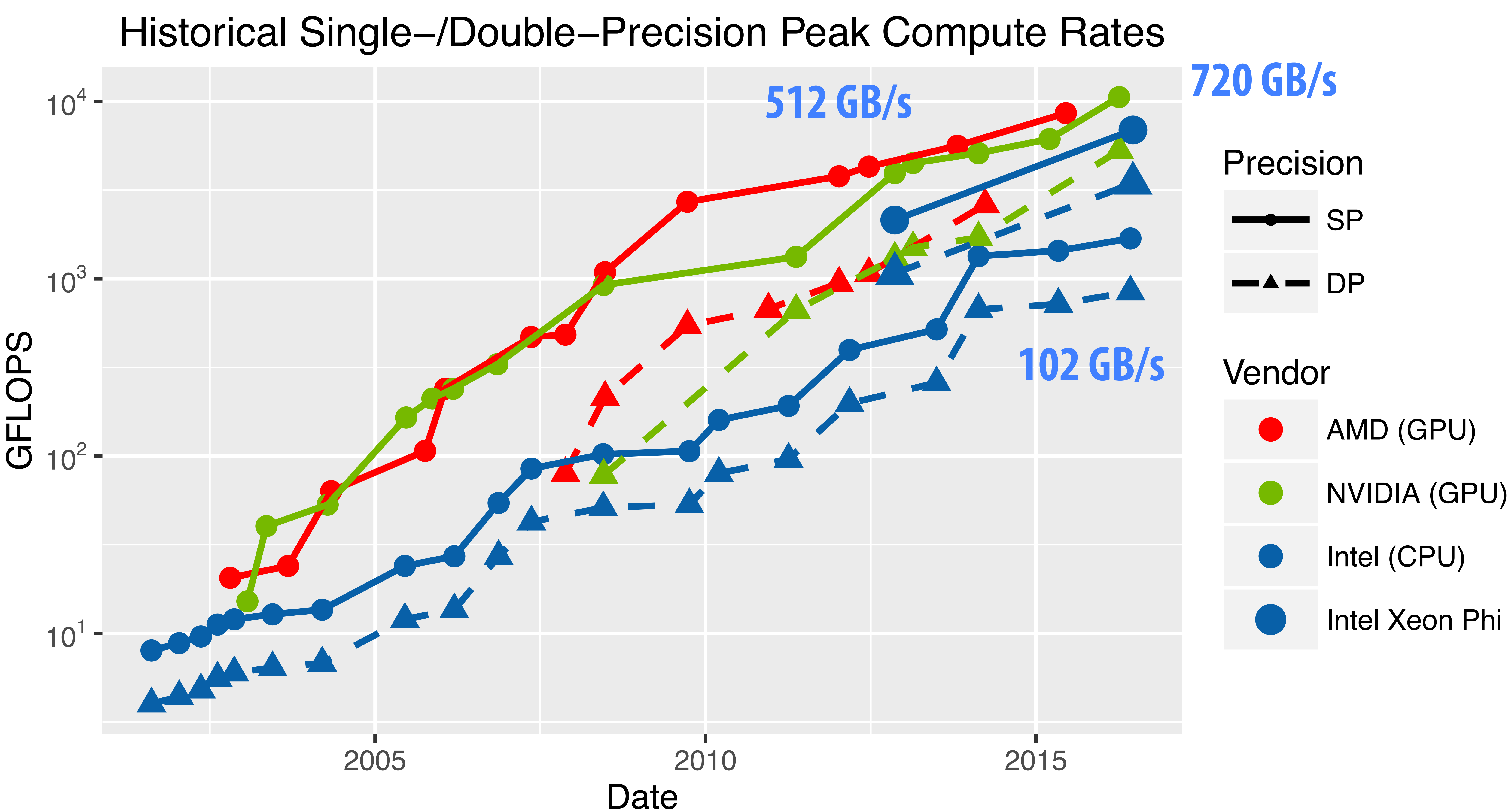
Why Graphics HW? Compute



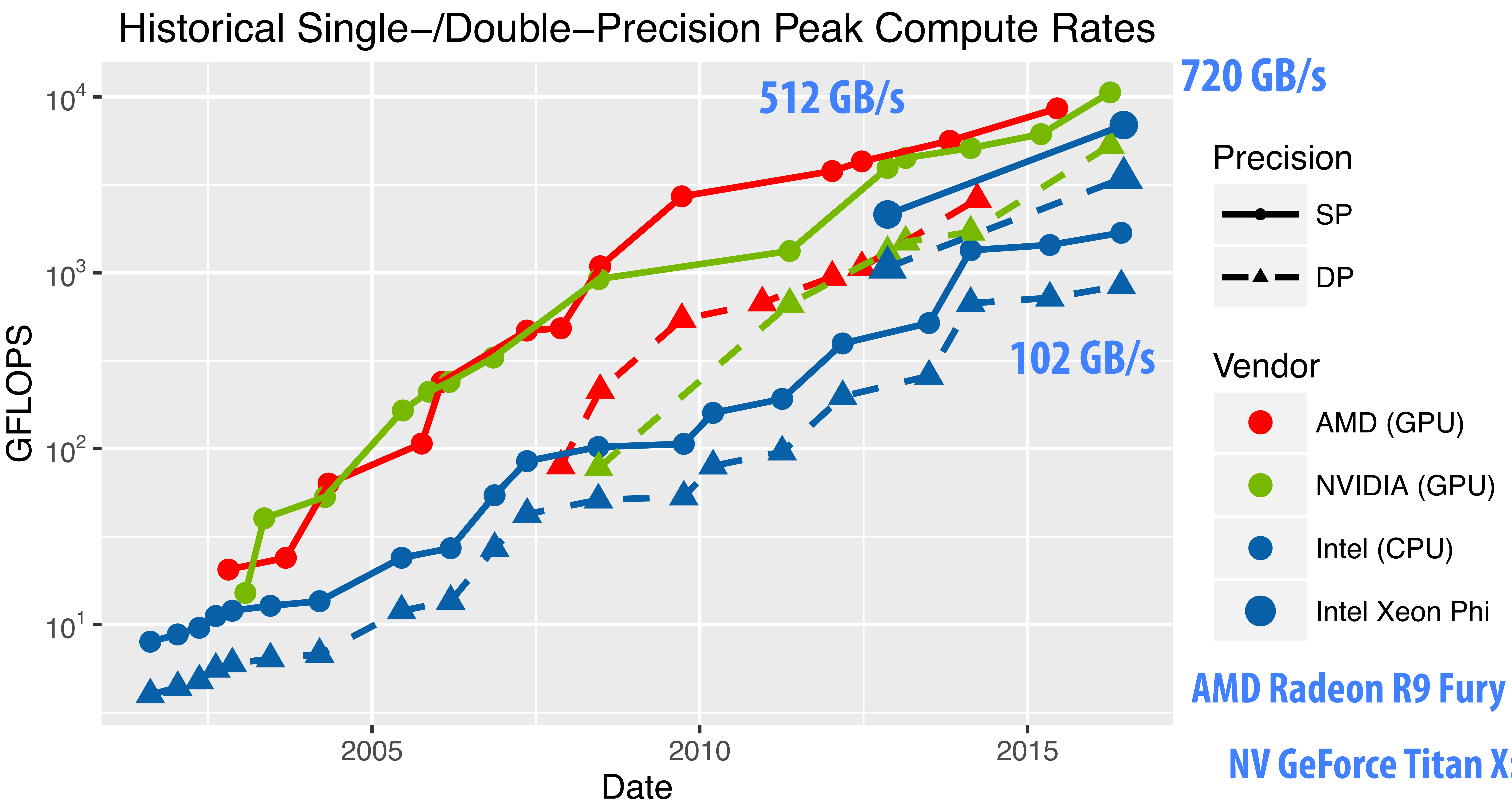
Recent GPU Performance Trends



Recent GPU Performance Trends



Recent GPU Performance Trends



AMD Radeon R9 Fury Pro: $649

NV GeForce Titan X: $1200

NV Tesla P100: $7374

Xeon E7-8890v4: $7174

Xeon Phi 7290: $6294

Why Graphics Hardware?

- … interesting problem
- … interesting algorithms
- … interesting implications for processor and system design
- … programmability
    - Hardware support?
    - Programming model?
- … possibility for general-purpose computing

Lecture 1: Introduction / Overview

- Why graphics hardware?
- Generations of graphics hardware
- Class outline
- Administrivia

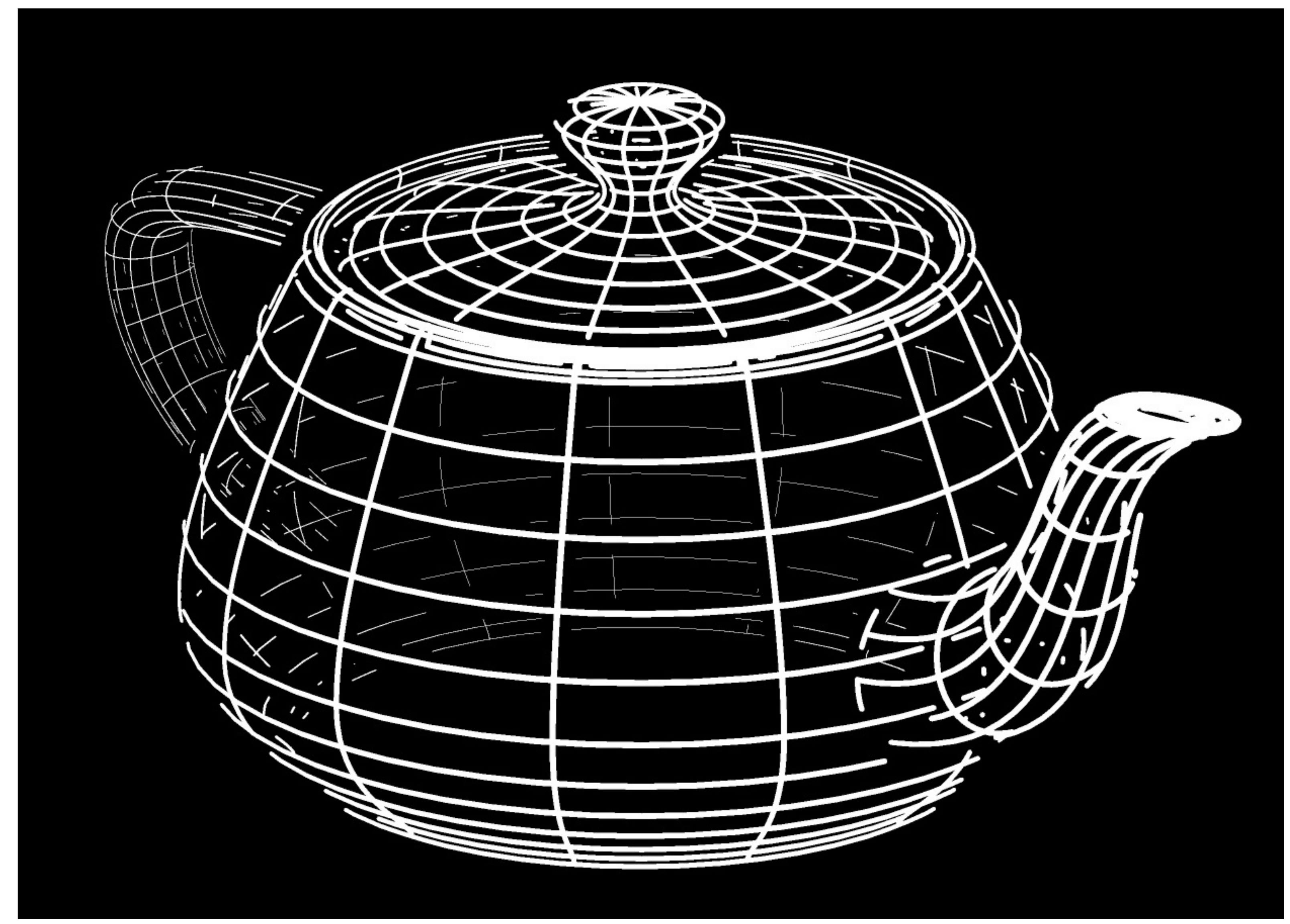
Generation 0

- Software-only systems
- Vector displays
- Framebuffers expensive ($20k)
- Military flight simulators (beam-chasing)

[Evans and Sutherland SP1, courtesy of http://www.raes.org.uk/fl-sim/]

Generation 1

- Wireframe models
- Technical achievement:
  - HW floating point computation
  - [Jim Clark, “The Geometry Engine: A VLSI Geometry System for Graphics”, SIGGRAPH ’82]
- HW: 1982–87 (workstations)



Generation 2

Shaded solids

Technical achievement:

- **Rasterization fill hardware**

Per-vertex lighting

Gouraud (smooth) shading

HW: 1987–1992 (workstations)

Generation 3

Texture mapping

Technical achievement:

- Per-fragment texture hardware

HW: 1992–2000 (workstations and consumer HW)

[courtesy of Michael Bax]

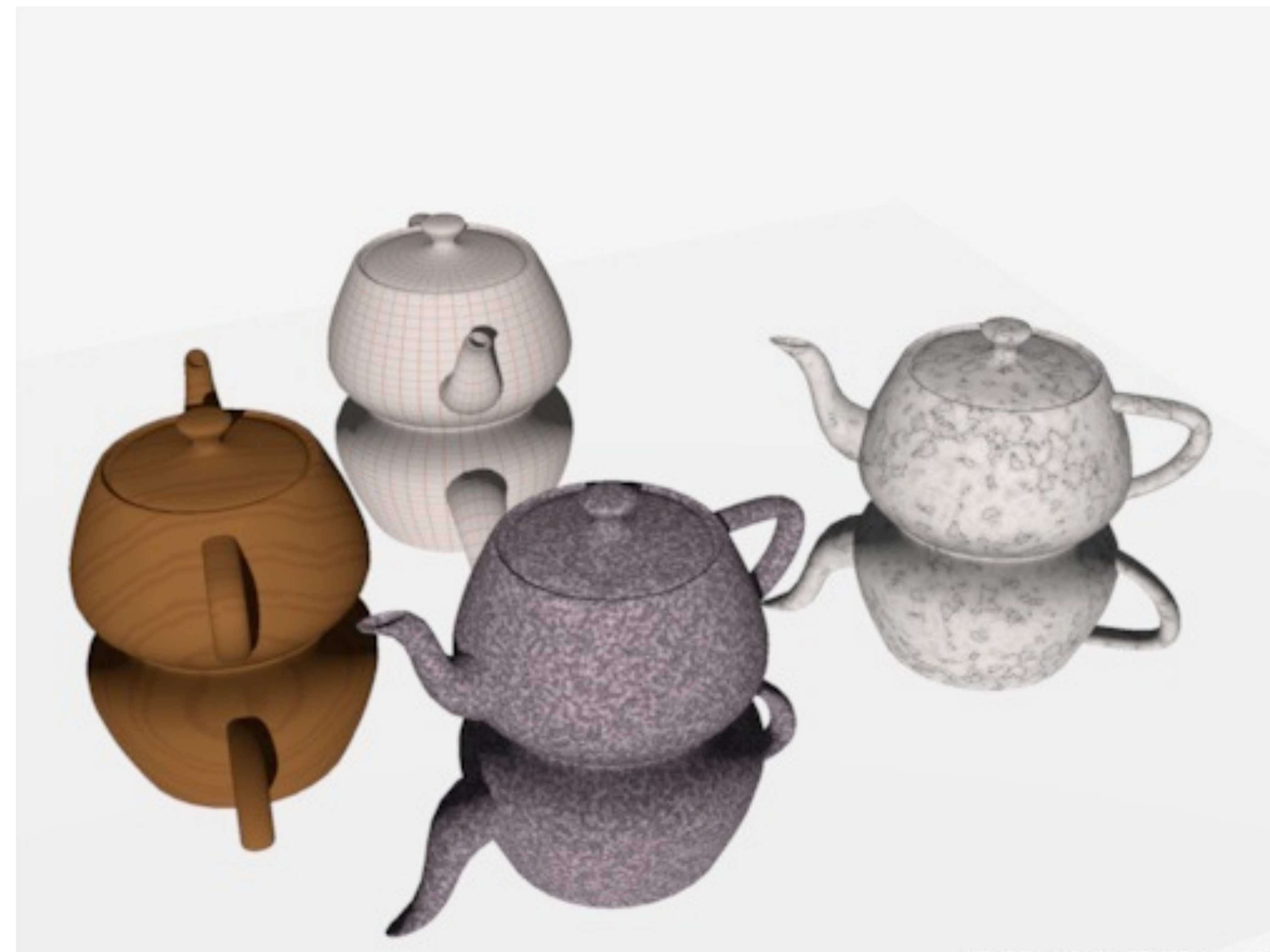
Generation 4

Programmability

Technical achievement:

- Programmable HW
  - Vertex
  - Fragment

HW: 2000– (consumer HW)



[courtesy of Henrik Jensen]

Generation 5 . . . ?

Global Evaluation

- Ray tracing
- Light transport?

Cinematic Rendering?

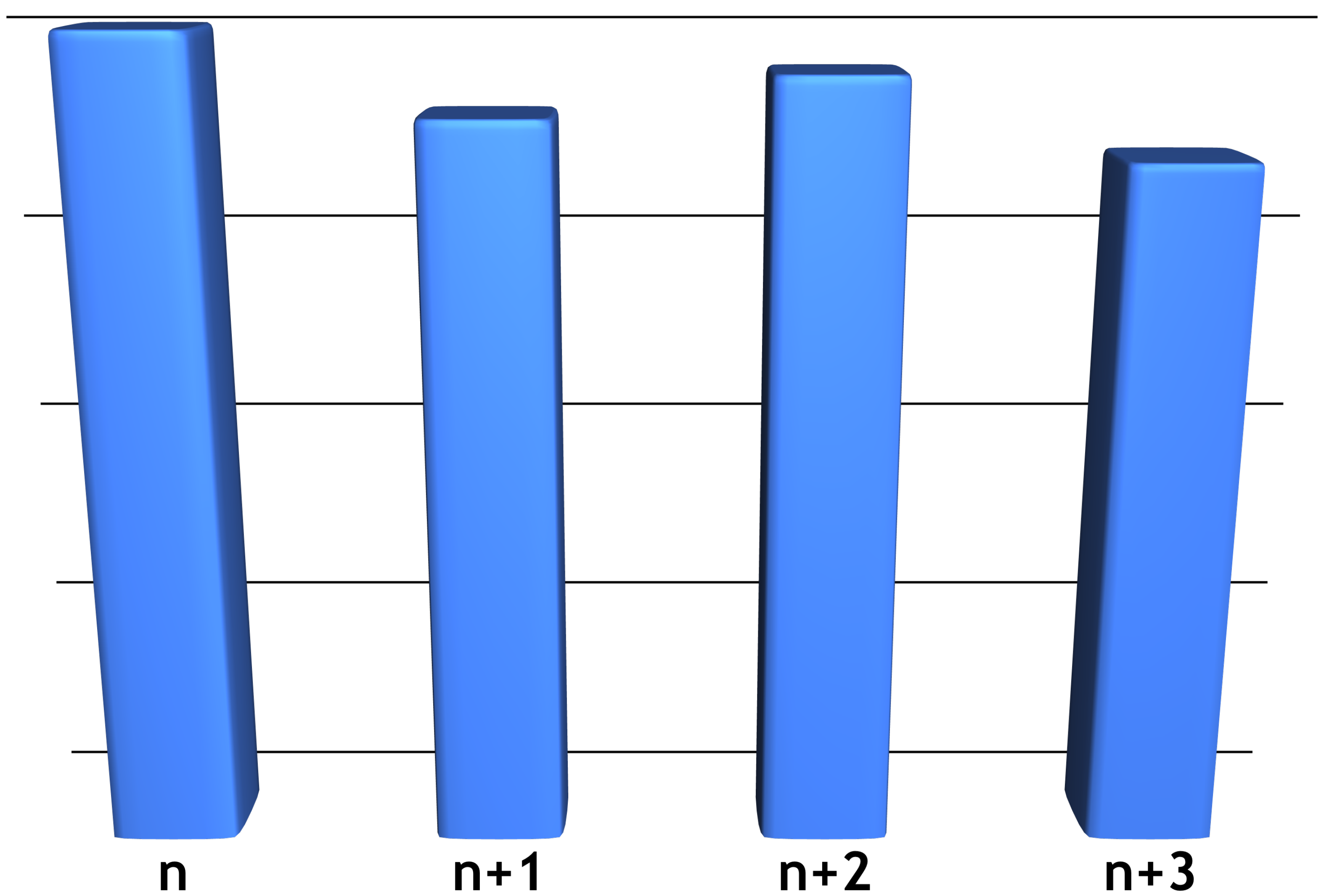
[courtesy of Henrik Jensen]

Generational Improvements

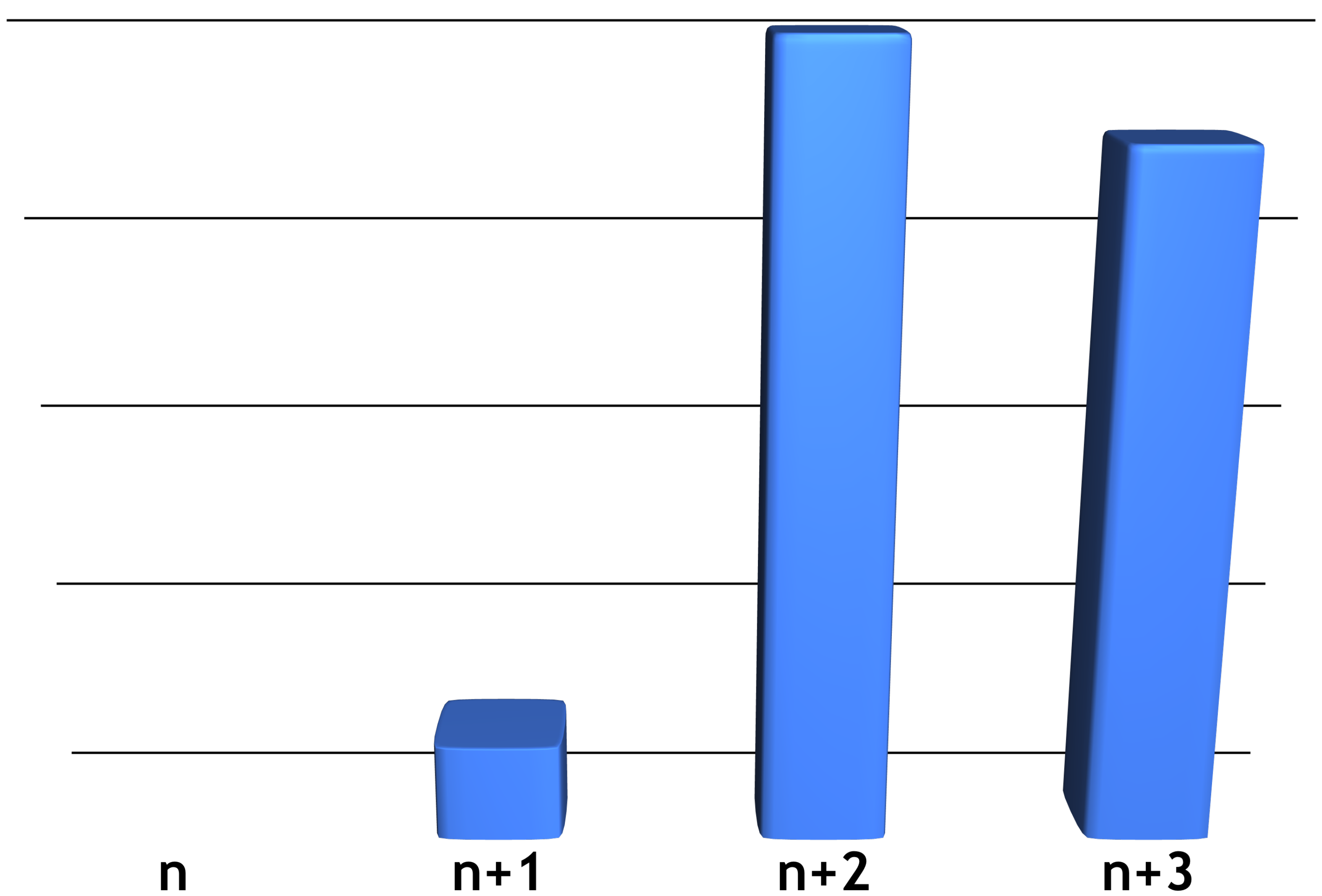
- Performance
  - Triangles/second (geometry)
  - Pixels/second (rasterization)
- Features
  - Hidden surface elimination, texture mapping, antialiasing …
- Quality
  - Bits of resolution, filtering …

Hardware additions

For existing features …

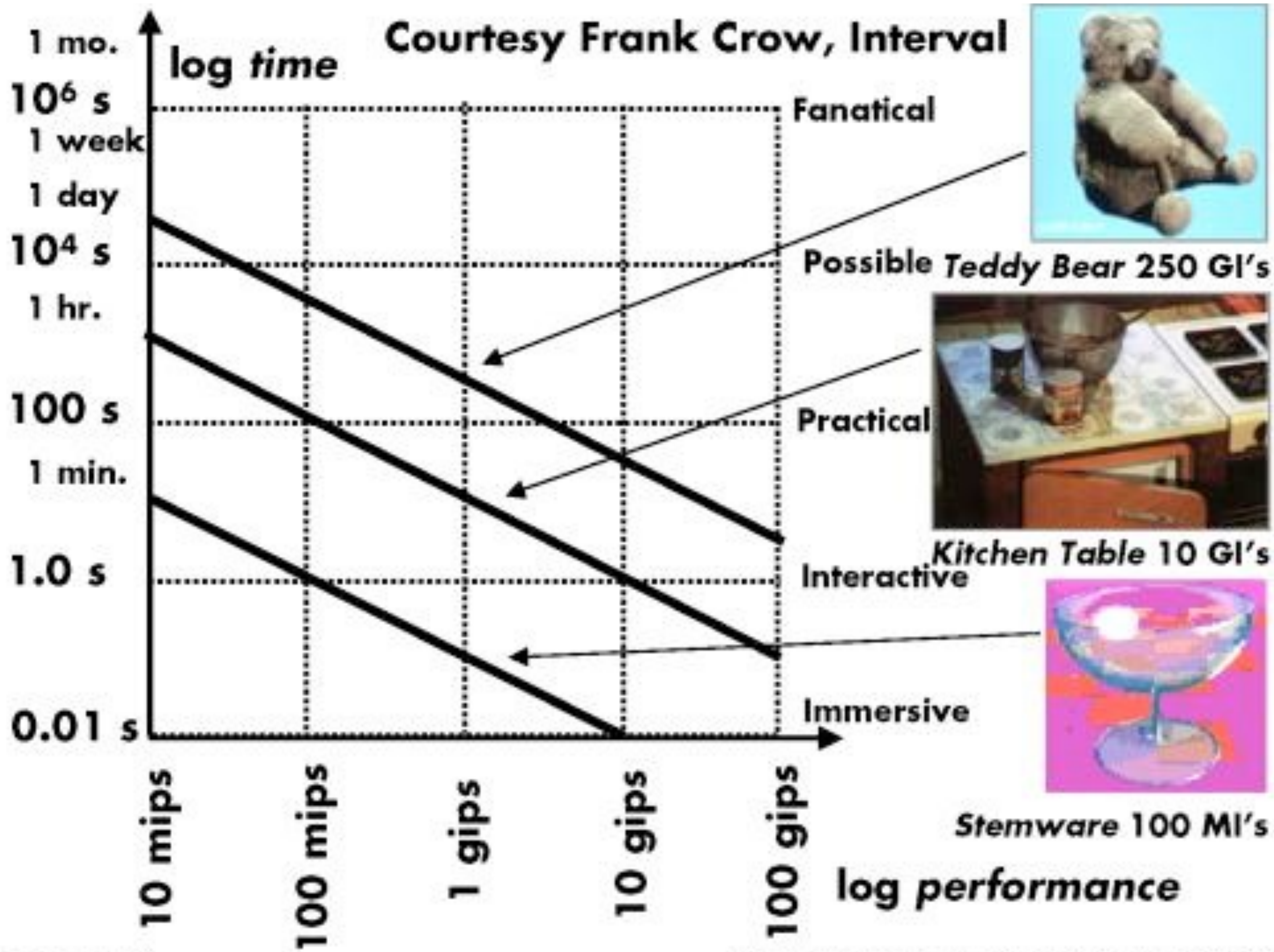


For new features …



Fast Path

Off-line to on-line to real-time . . .



[courtesy of Crow/Hanrahan/Akeley]

. . . but a long way to go.

- "Reality is 80 million polygons." – Alvy Ray Smith, Pixar

[courtesy of Matt Pharr]

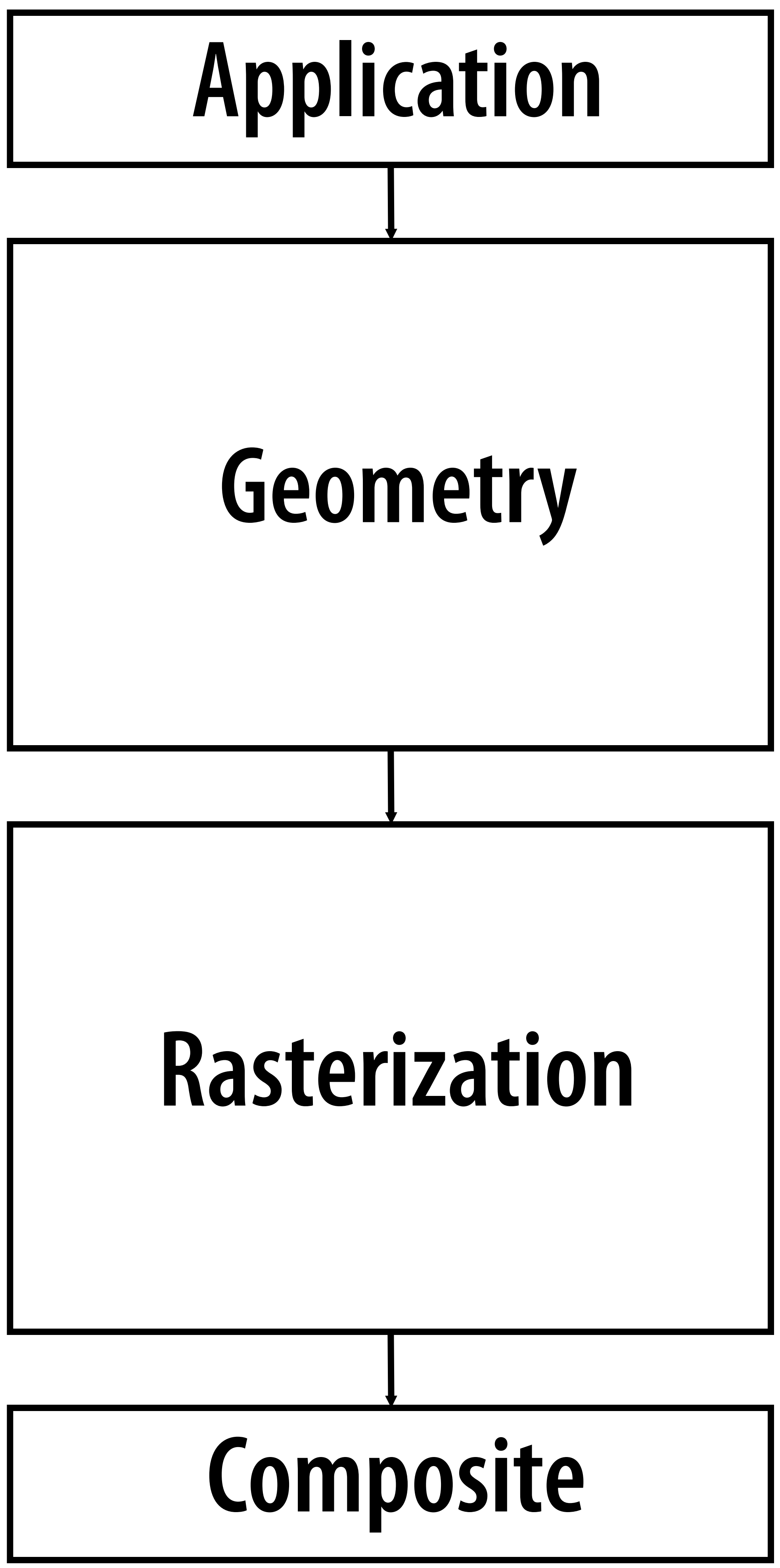
Lecture 1: Introduction / Overview

- Why graphics hardware?
- Generations of graphics hardware
- Class outline
- Administrivia

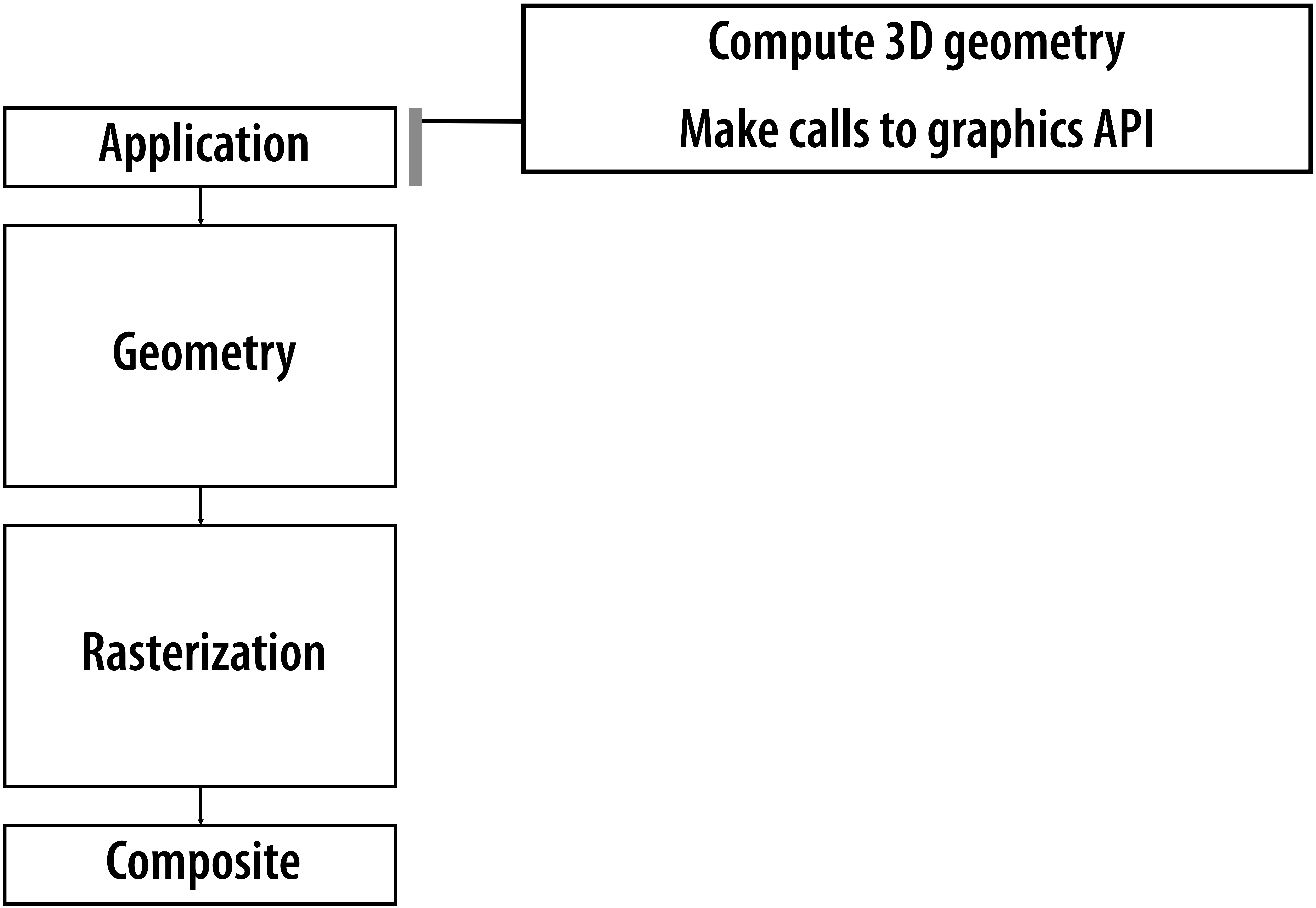
Intro to OpenGL

- For students with little experience with OpenGL:
  - Tutorial: W afternoon?
    - Time depends on room availability. I asked for 2–4 pm.
  - We'll walk through some sample code and discuss the pipeline
- Students with OpenGL experience: Skip it (but you're welcome to come)

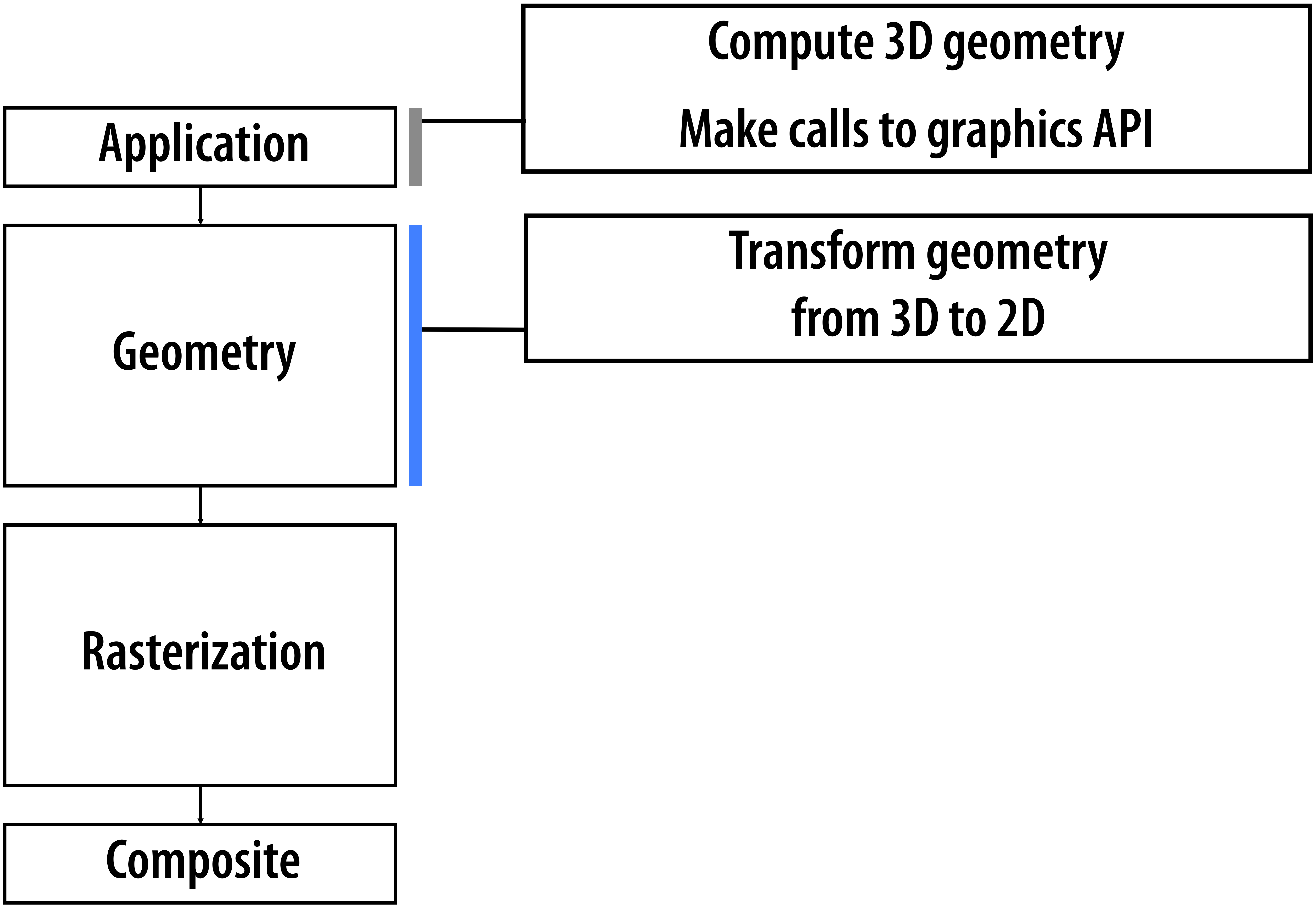
The Rendering Pipeline (Lecture 2)



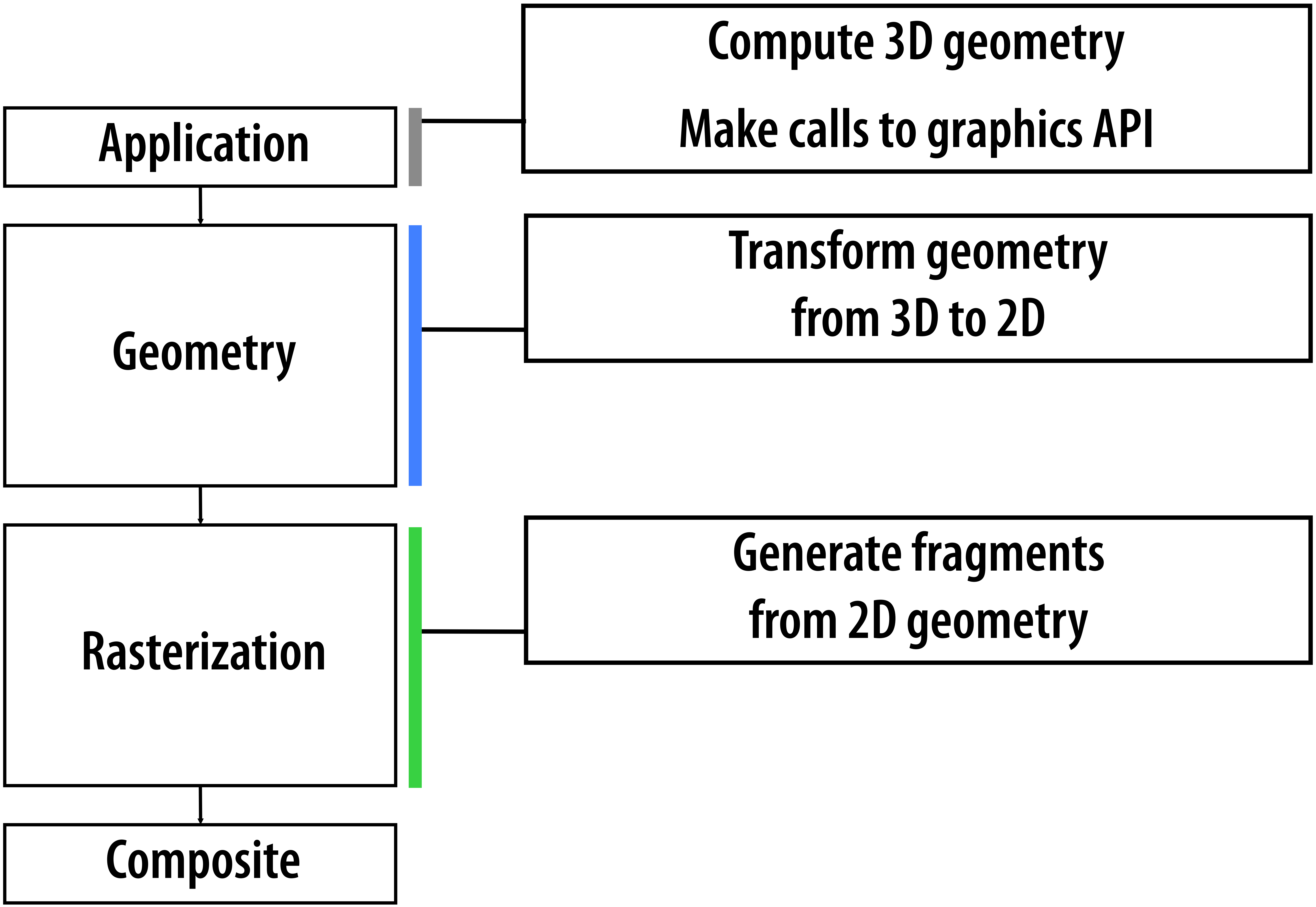
The Rendering Pipeline (Lecture 2)



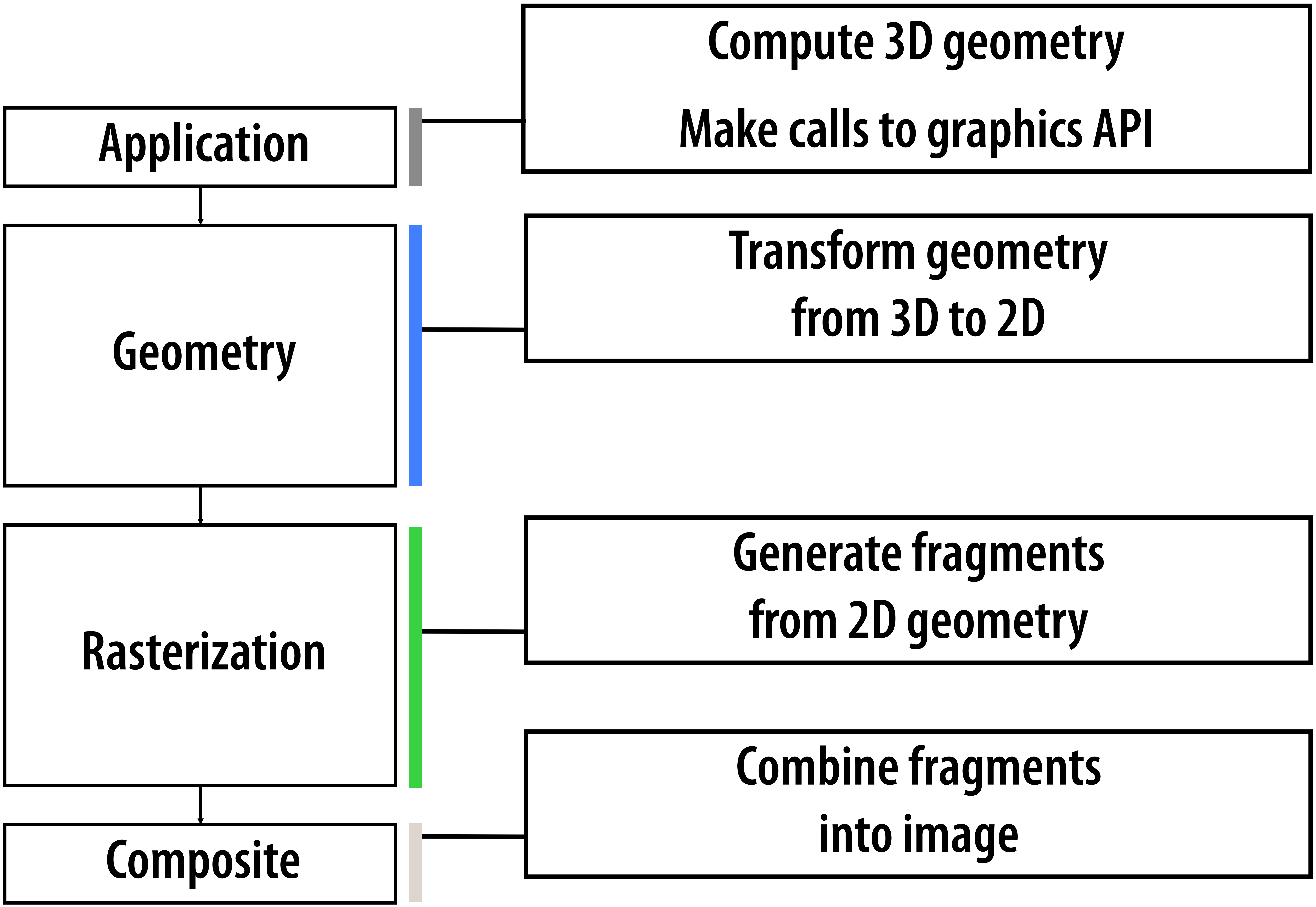
The Rendering Pipeline (Lecture 2)



The Rendering Pipeline (Lecture 2)



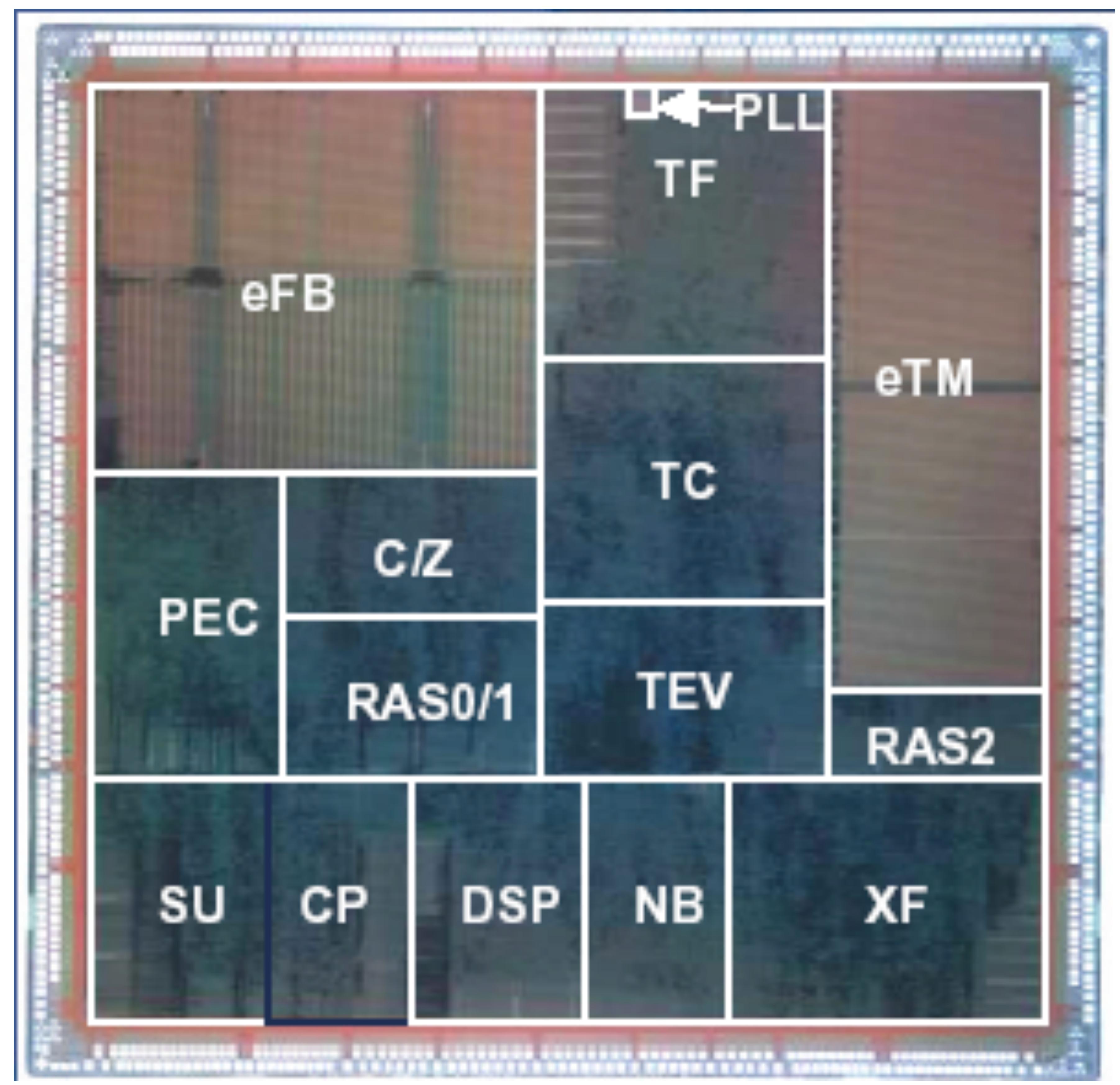
The Rendering Pipeline (Lecture 2)



Lecture 3: How a GPU Works

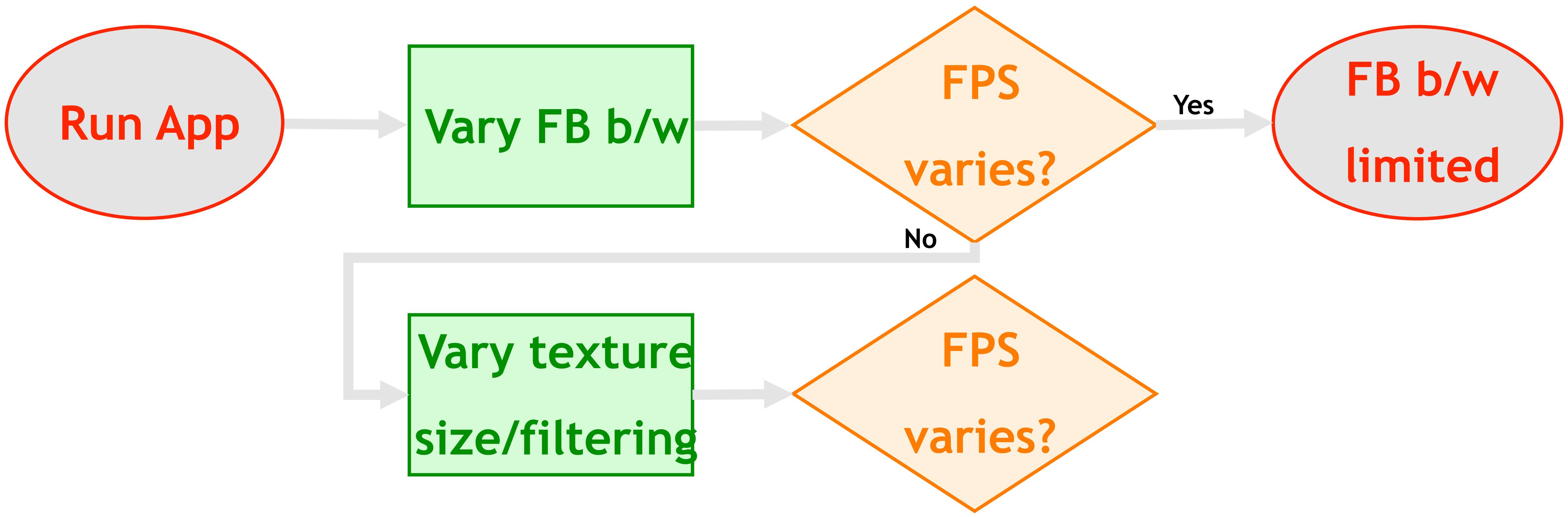
- **VLSI Trends**
- **Why is Graphics Hardware Fast?**

ATI Flipper [N64]

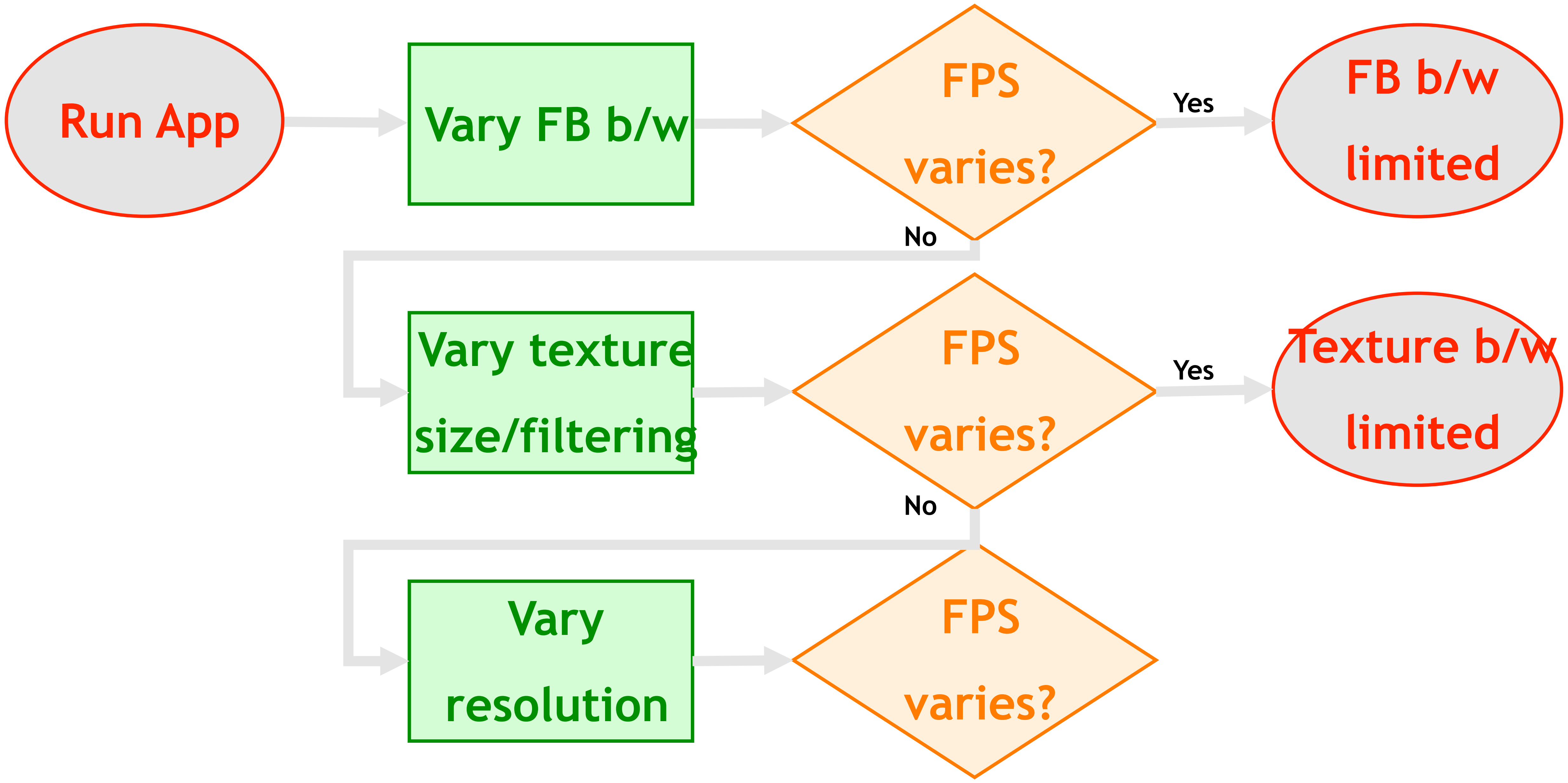


Lecture 4: Performance Characterization

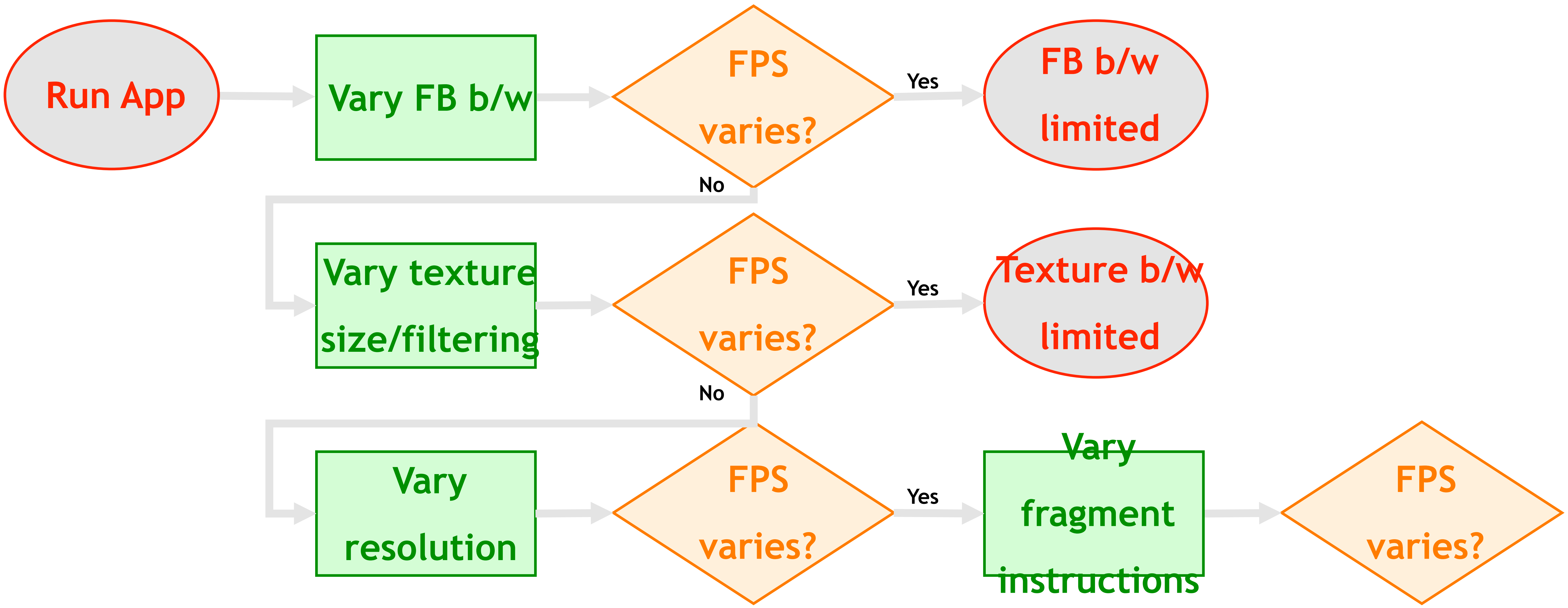
Lecture 4: Performance Characterization



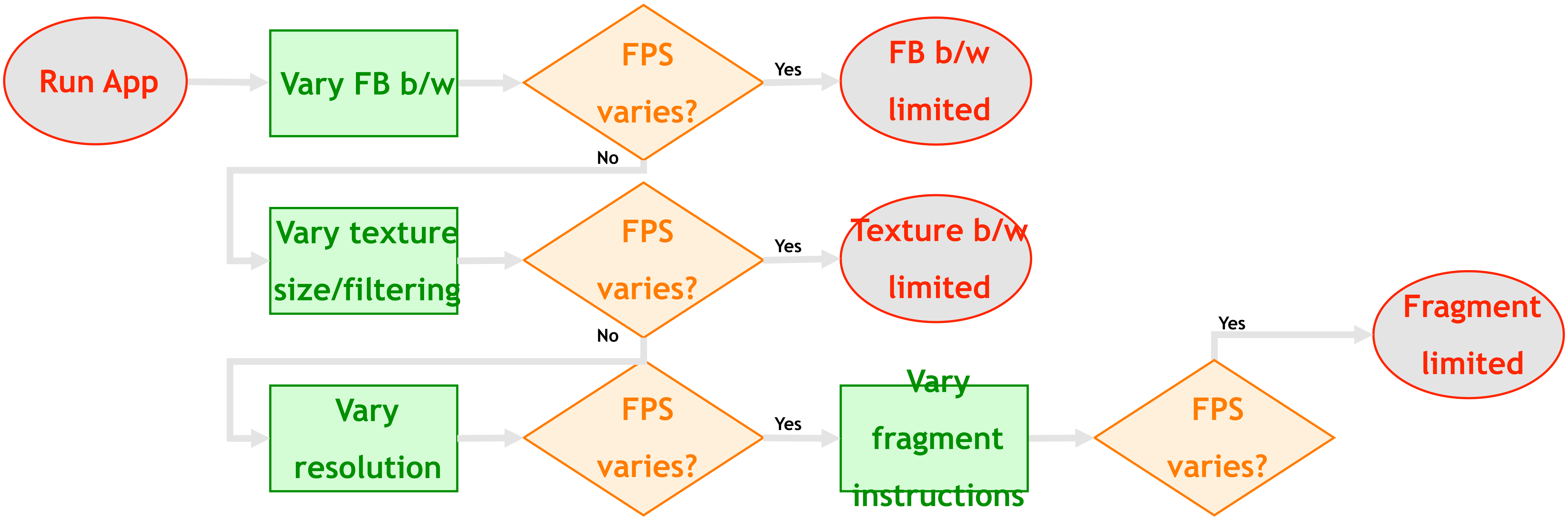
Lecture 4: Performance Characterization



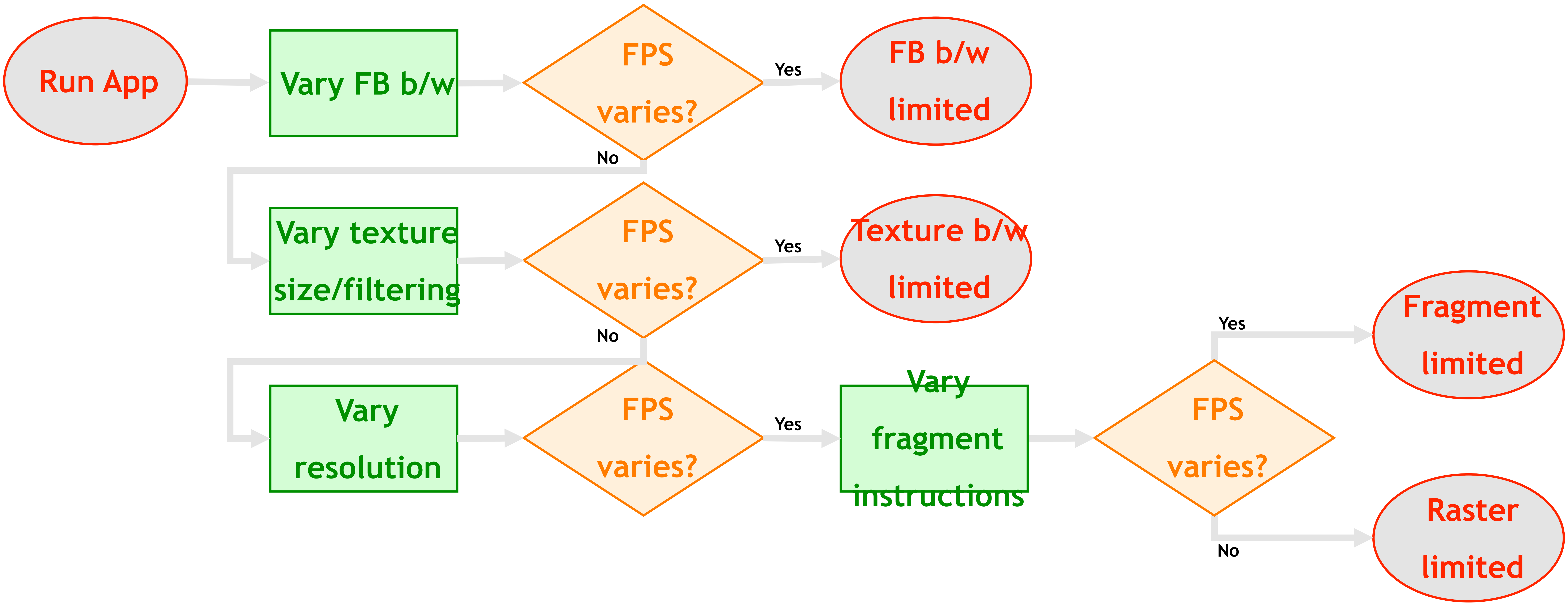
Lecture 4: Performance Characterization



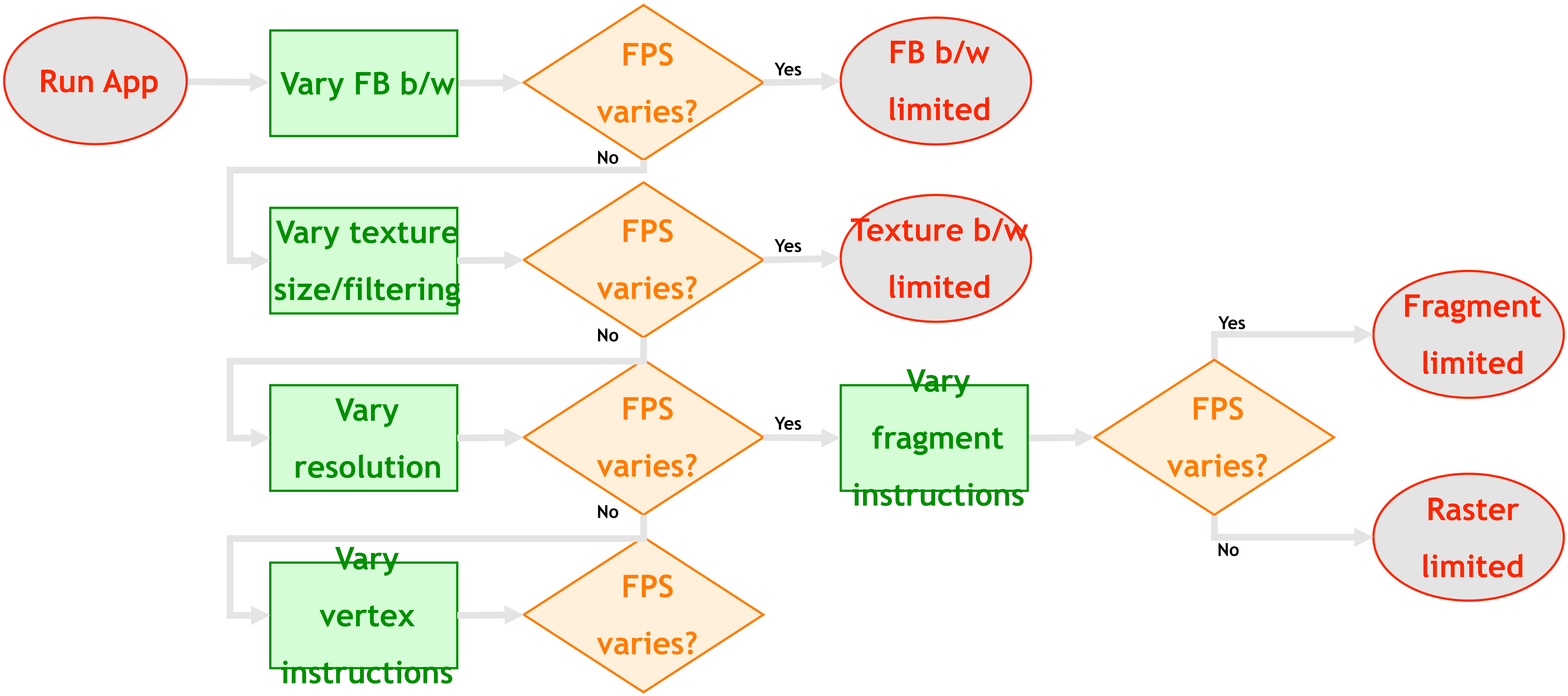
Lecture 4: Performance Characterization



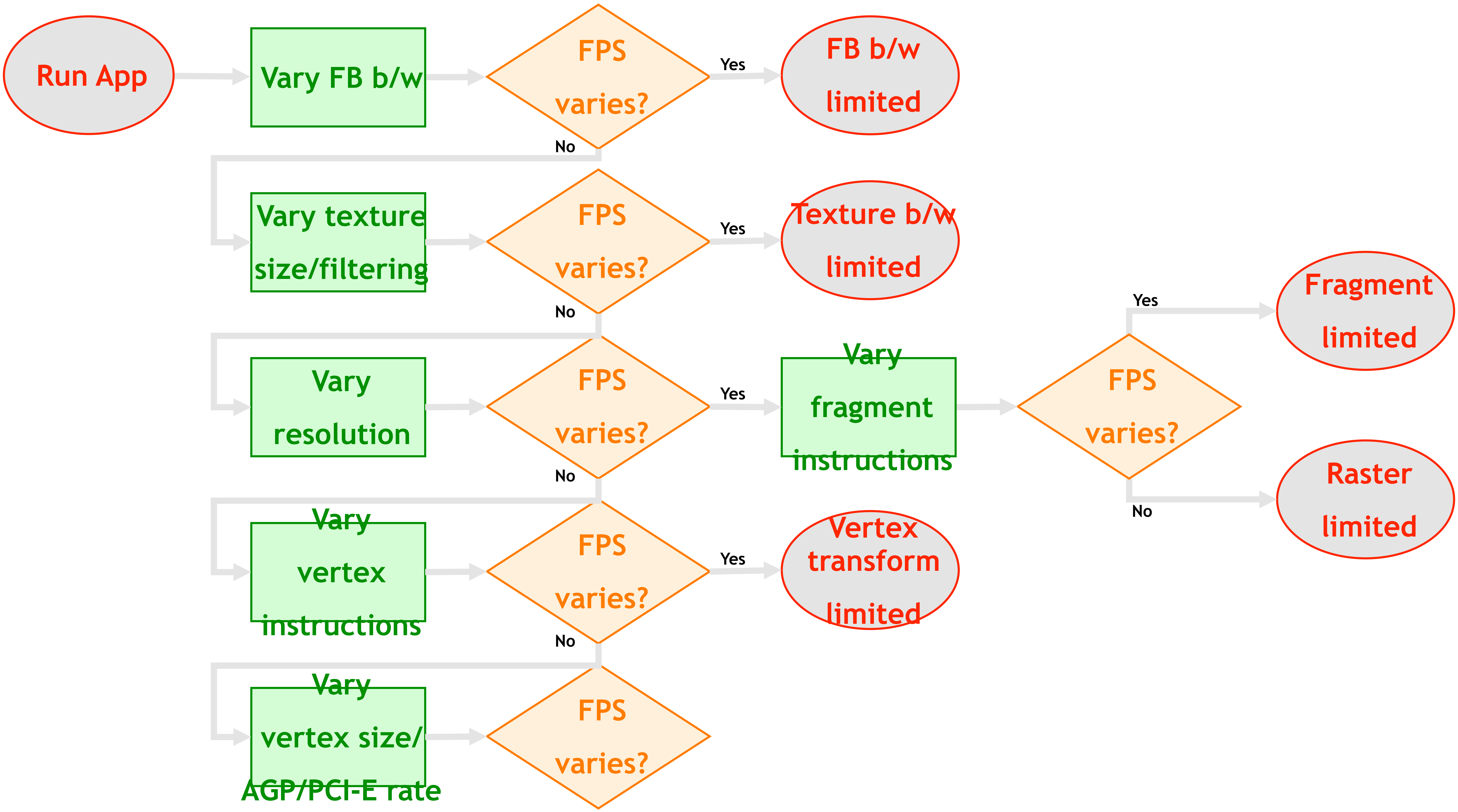
Lecture 4: Performance Characterization



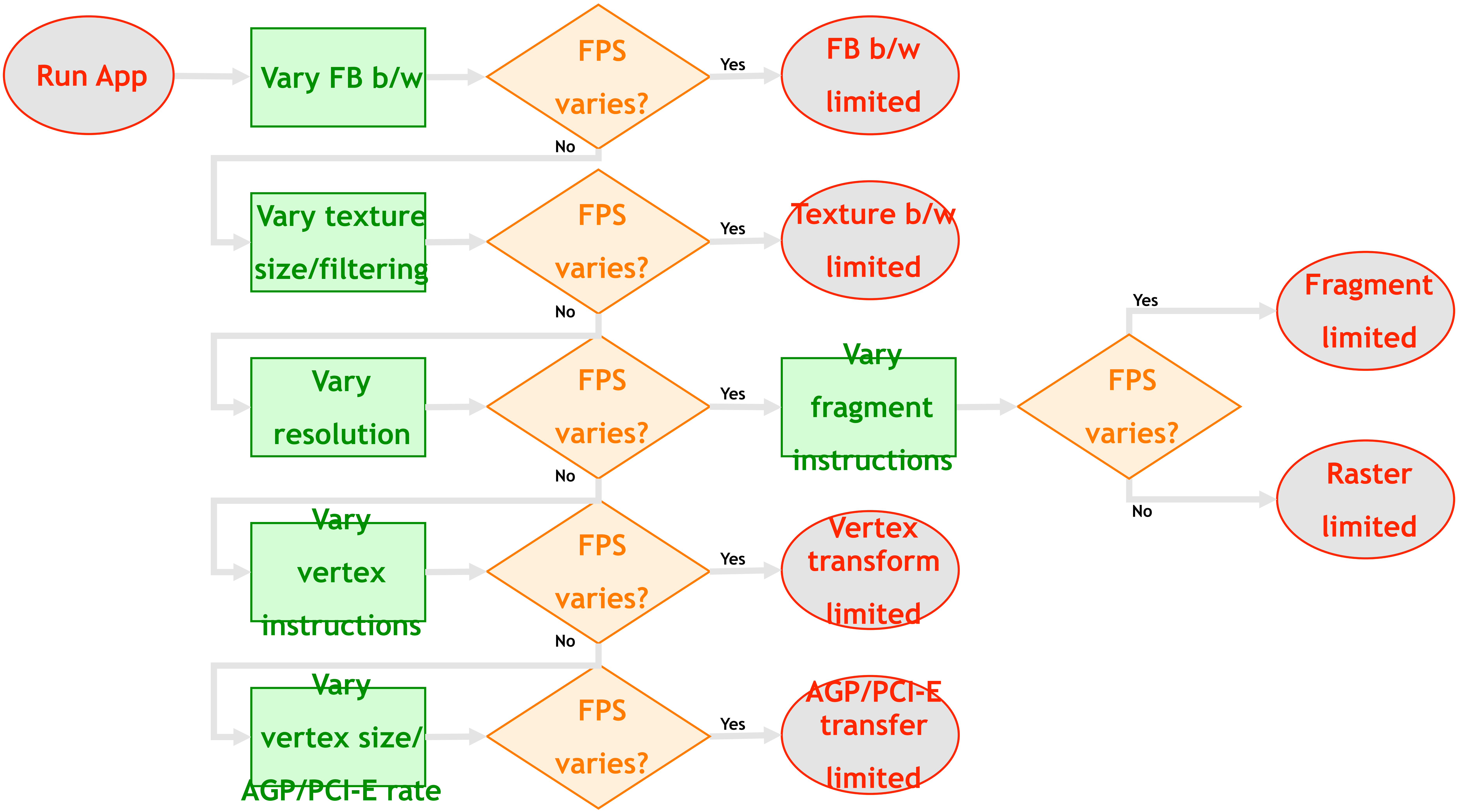
Lecture 4: Performance Characterization



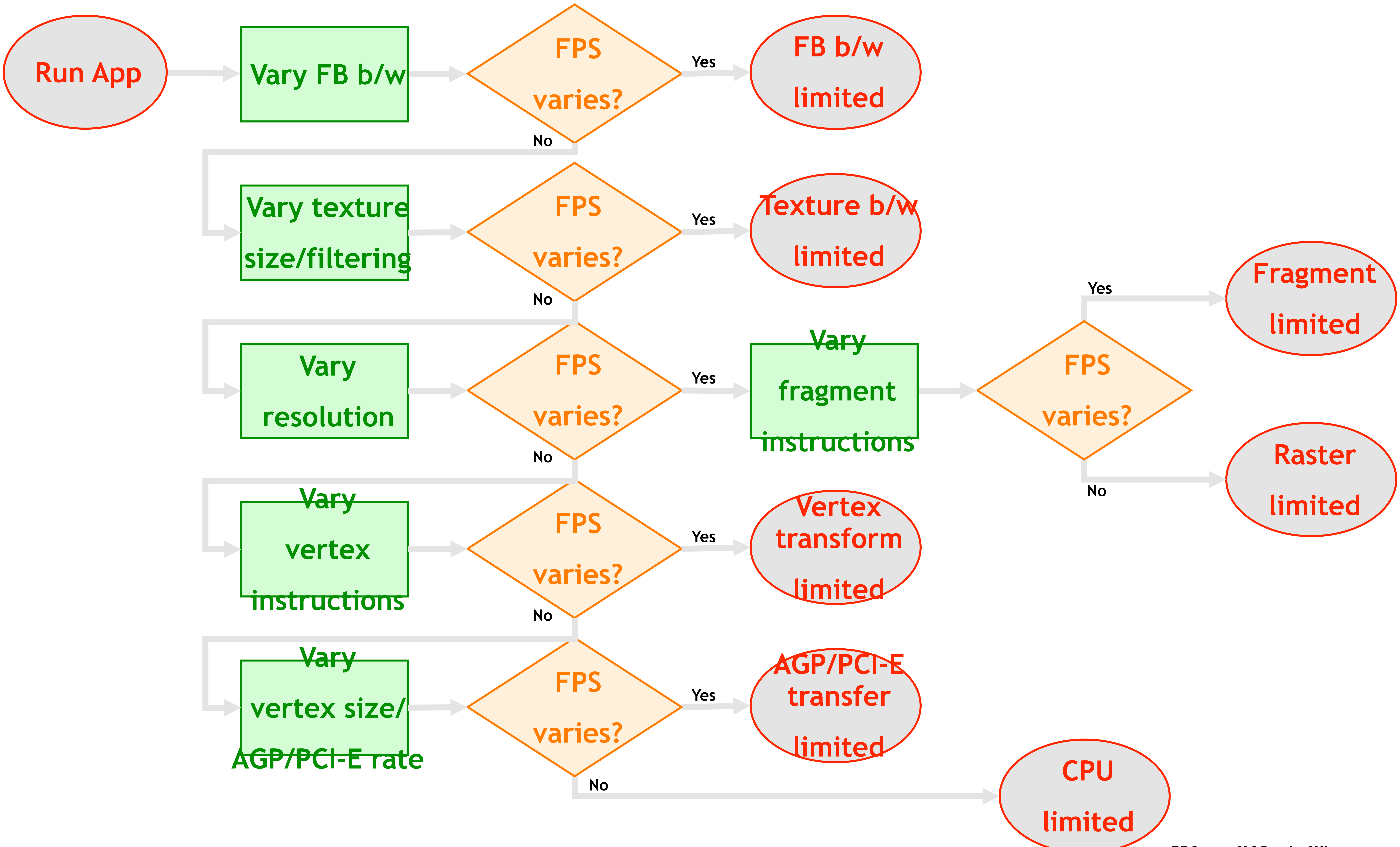
Lecture 4: Performance Characterization



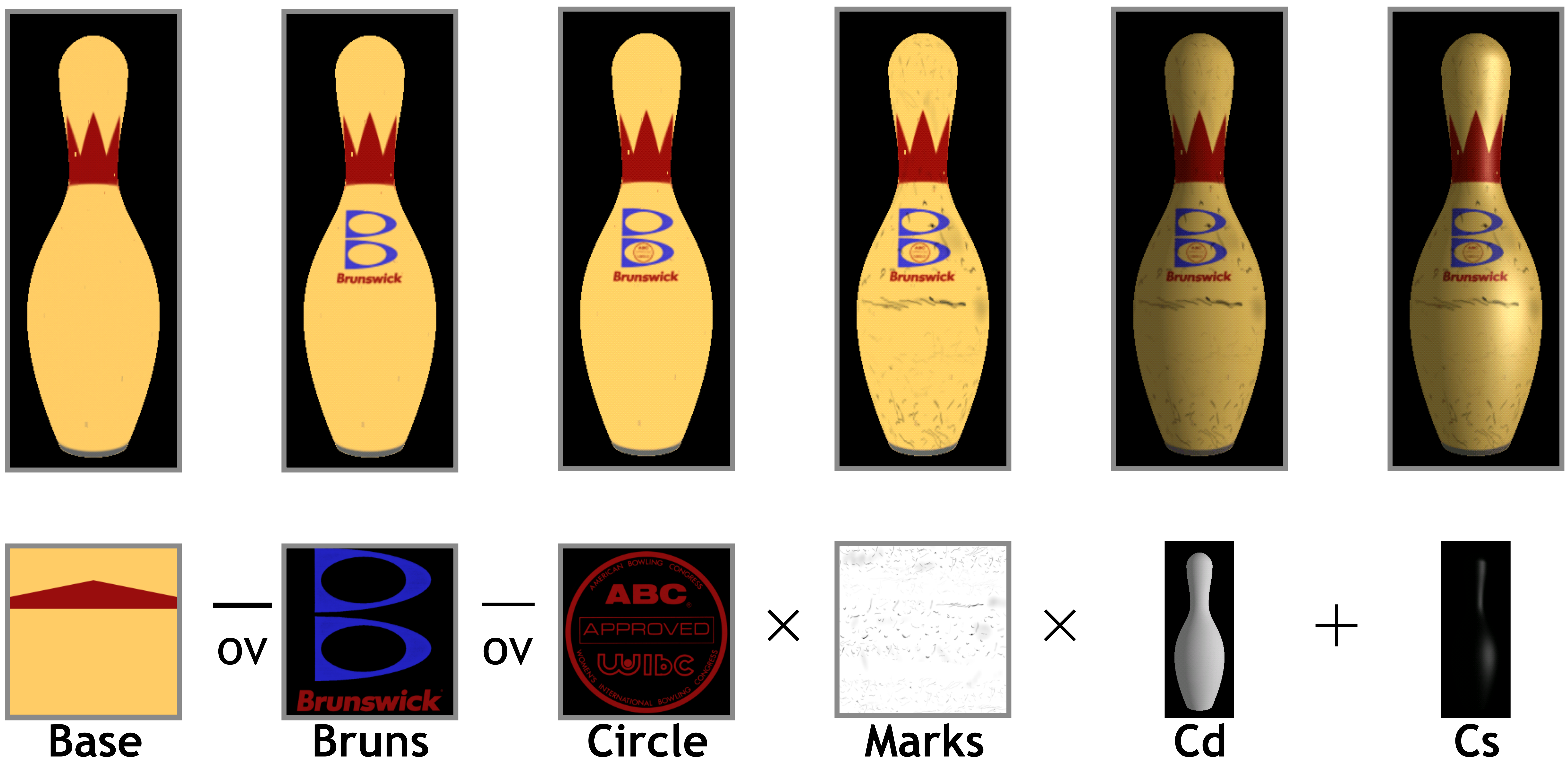
Lecture 4: Performance Characterization



Lecture 4: Performance Characterization



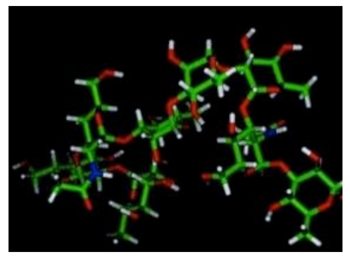
Lecture 5: The Programmable Pipeline

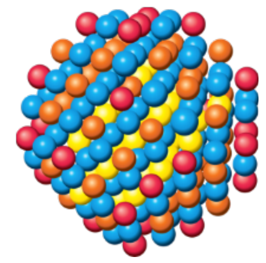


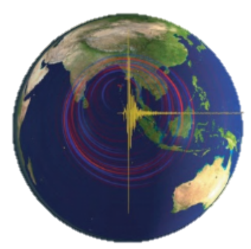
[courtesy of Kekoa Proudfoot]

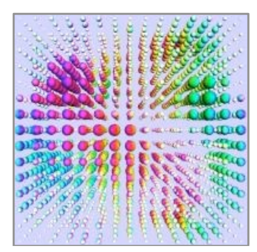
Courtesy Jerry Chen, NVIDIA

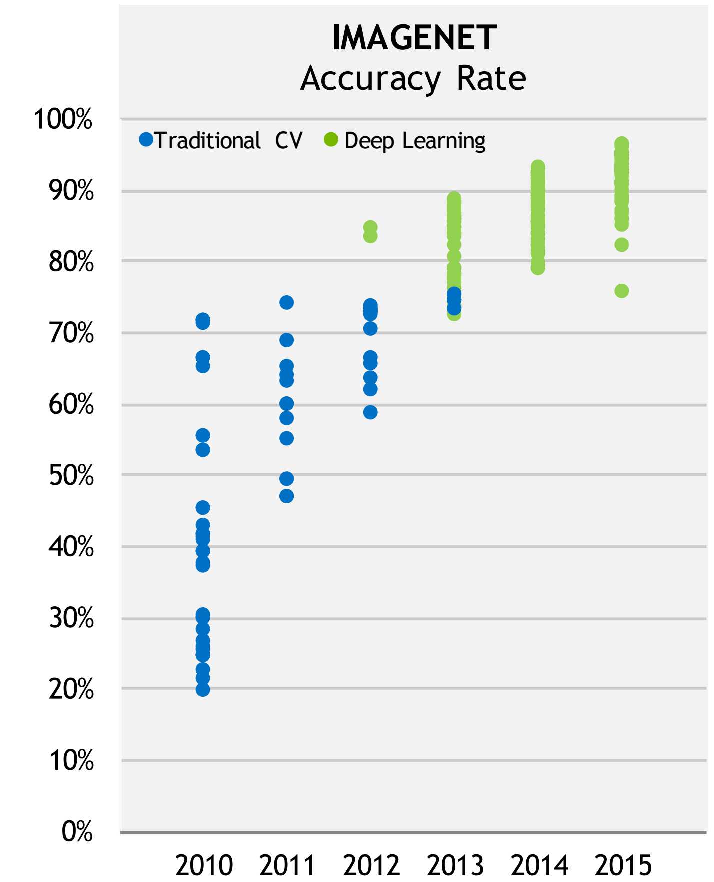
Lecture 6–7: GPGPU

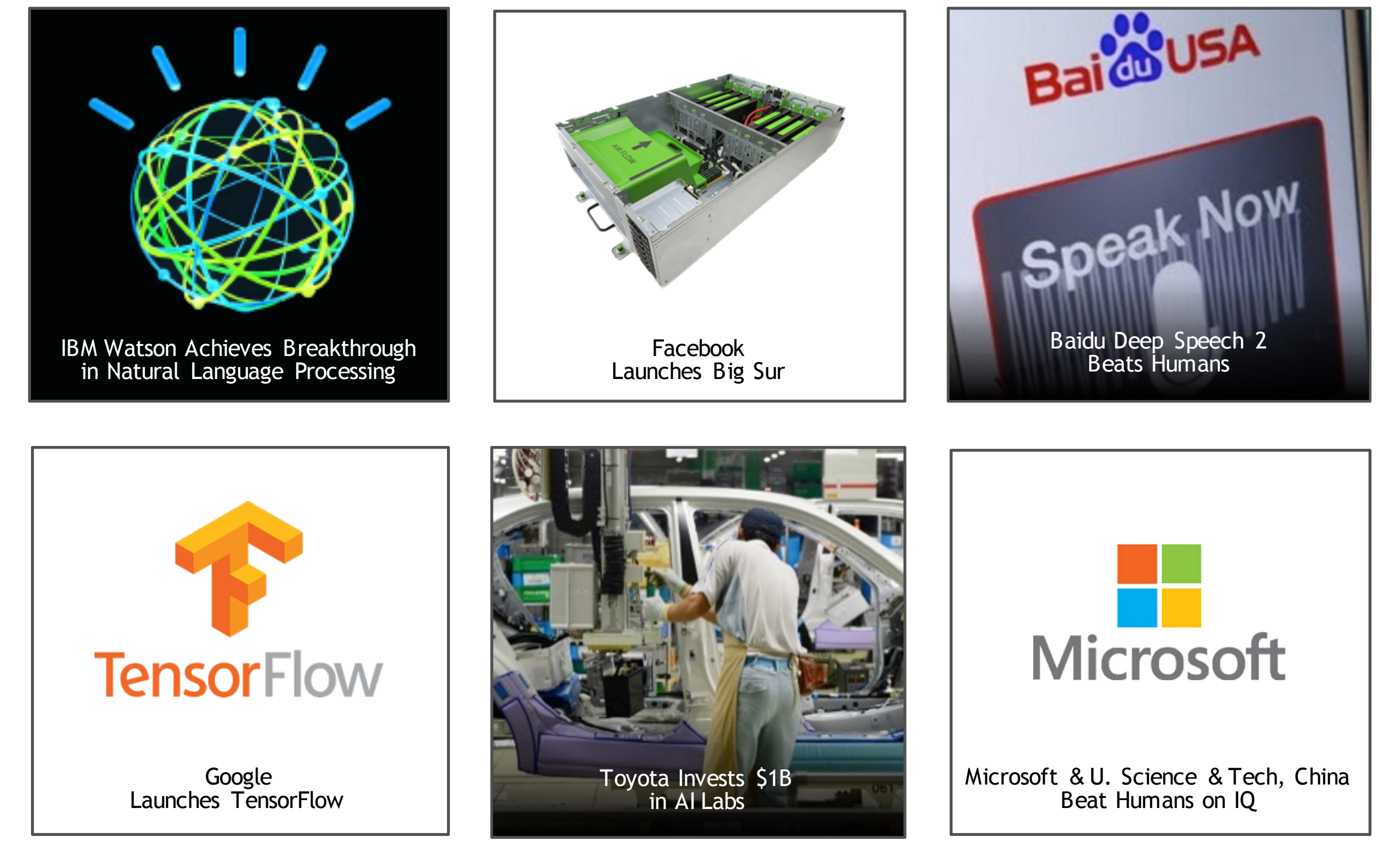

AMBER: Molecular Dynamics
7.1x

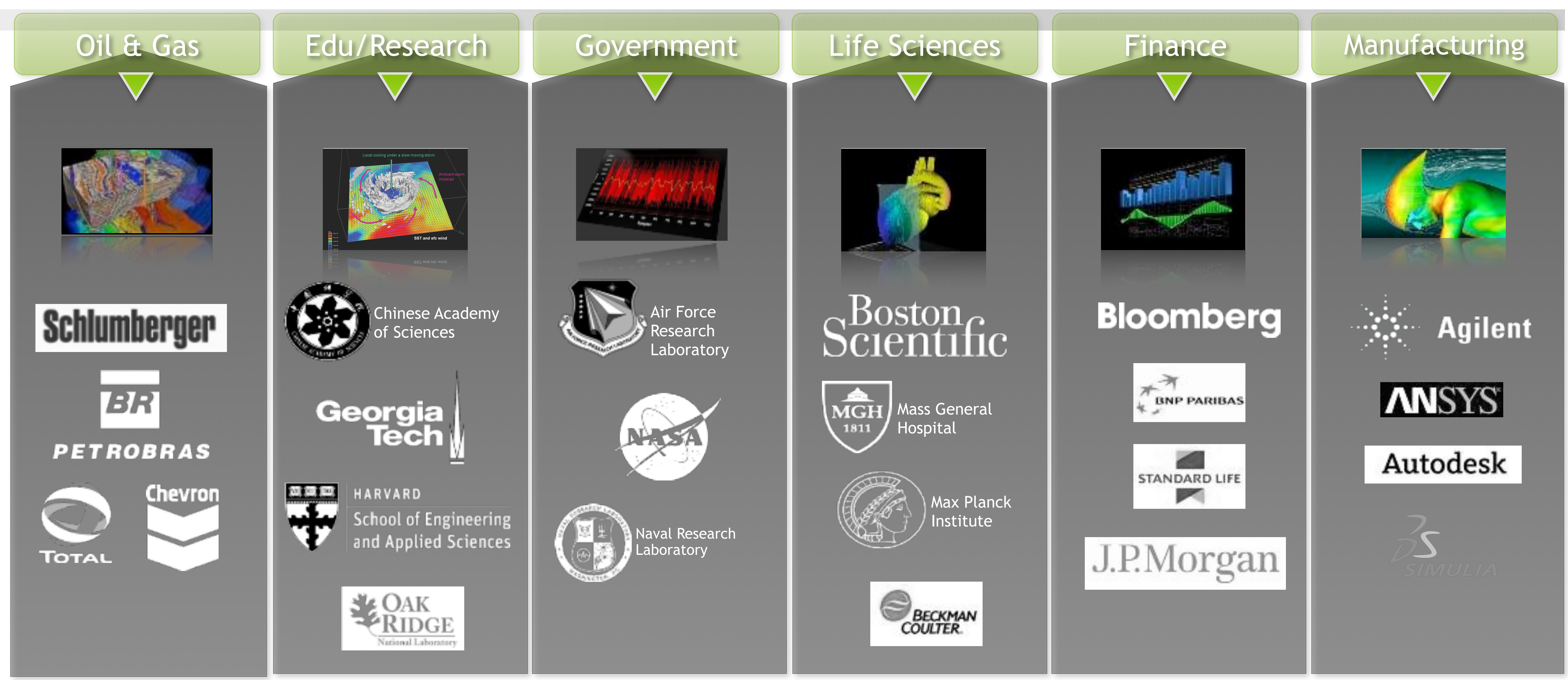

WS-LMS: Material Science
3.2x


SPECFEM3D: Wave Propagation
8.8x

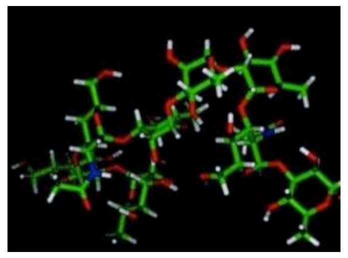

Chroma: Lattice QCD
3.7x



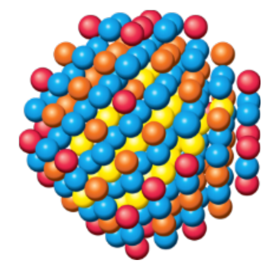




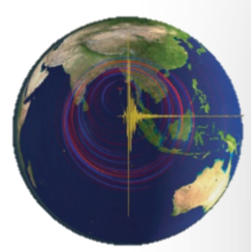
Lecture 6–7: GPGPU



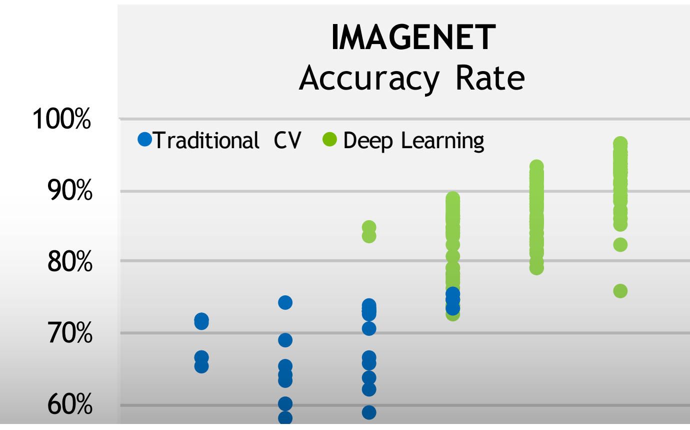
AMBER:
Molecular
Dynamics
7.1x

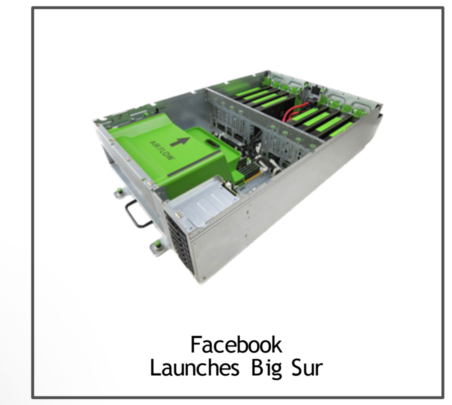


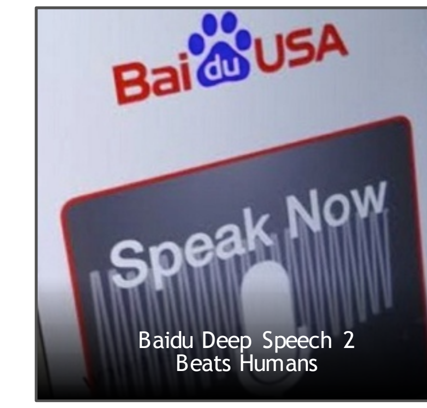
WS-LMS:
Material Science
3.2x



SPECFEM3D:
Wave Propagation
8.8x







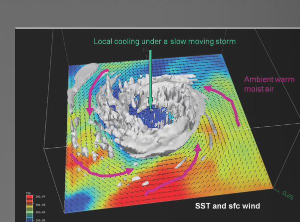
Microsoft

Microsoft & U. Science & Tech, China Beat Humans on IQ

Oil & Gas

Schlumberger
BR PETROBRAS
TOTAL
Chevron

Edu/Research



Chinese Academy of Sciences
Georgia Tech
HARVARD School of Engineering and Applied Sciences
Oak Ridge National Laboratory

Air Force Research Laboratory
NASA
Naval Research Laboratory

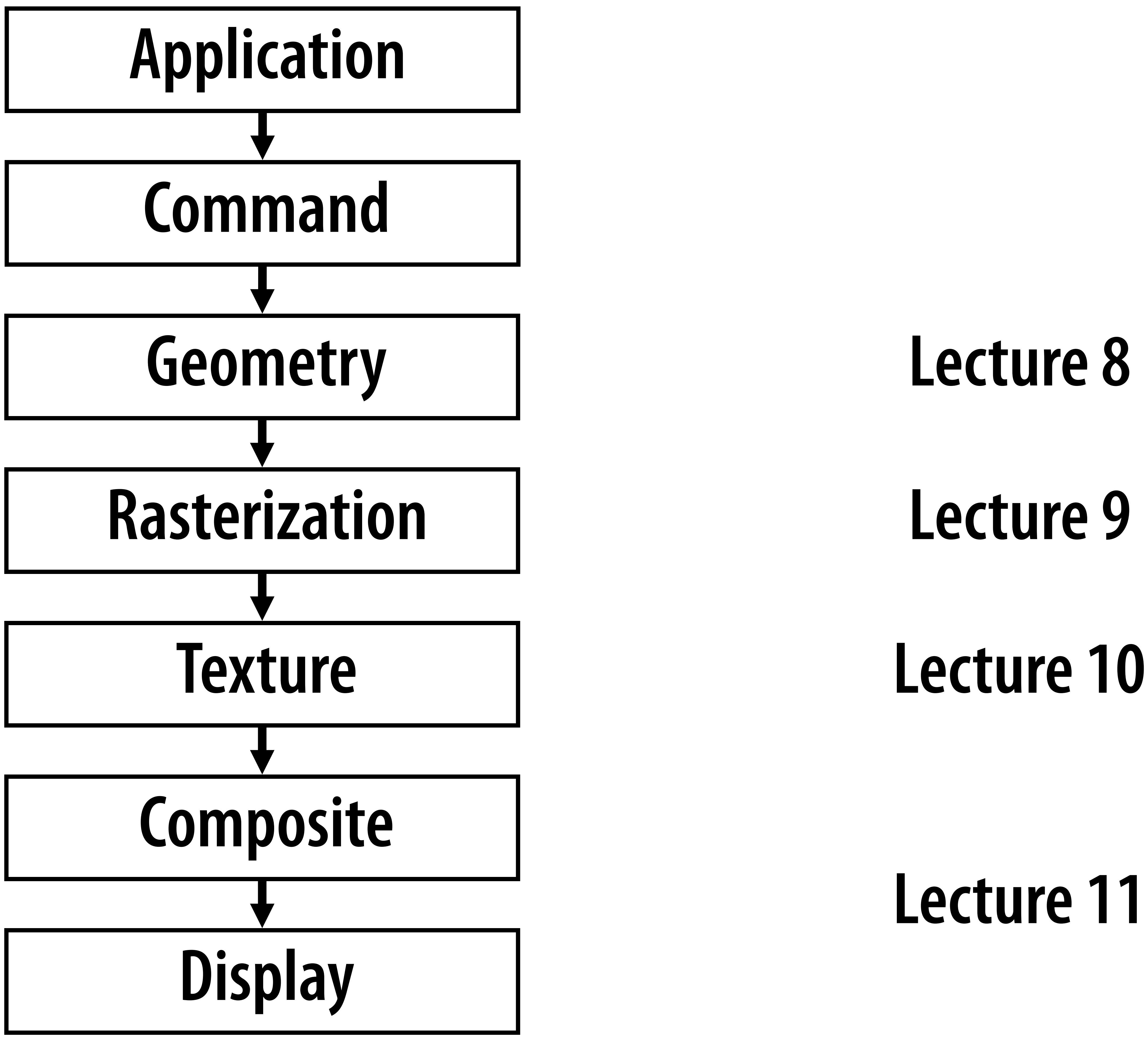
Boston Scientific
MGH 1811 Mass General Hospital
Max Planck Institute
Beckman Coulter

Bloomberg
BNP PARIBAS
STANDARD LIFE
J.P.Morgan

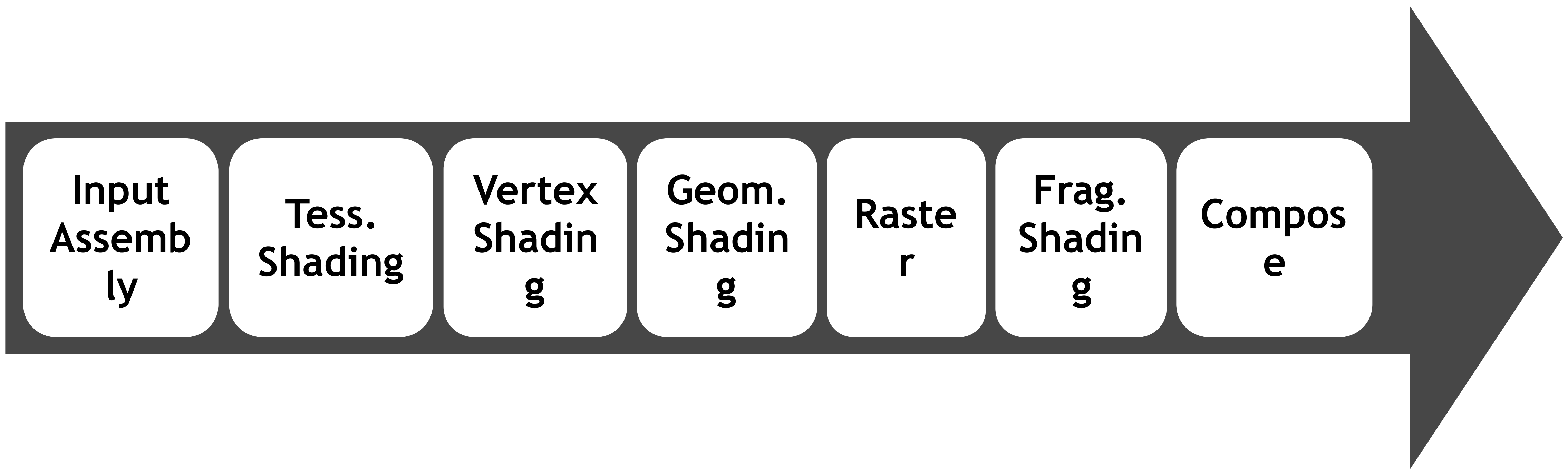
nufacturing

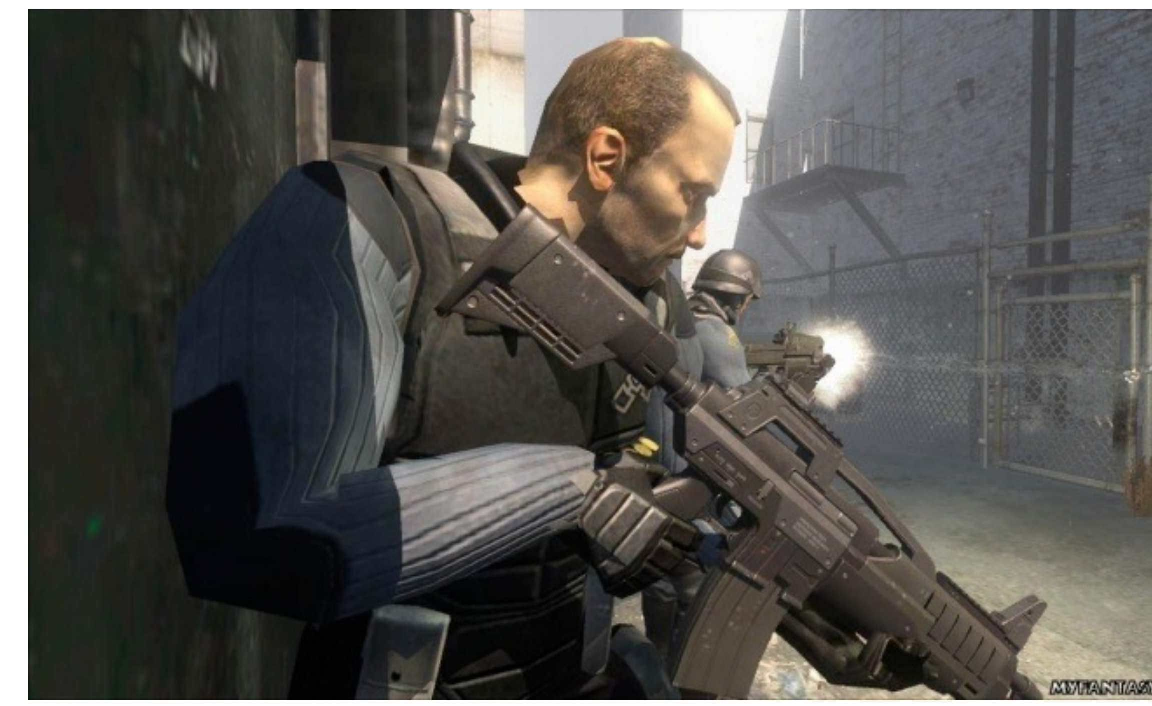
Agilent
ANSYS
Autodesk
3S SIMULIA

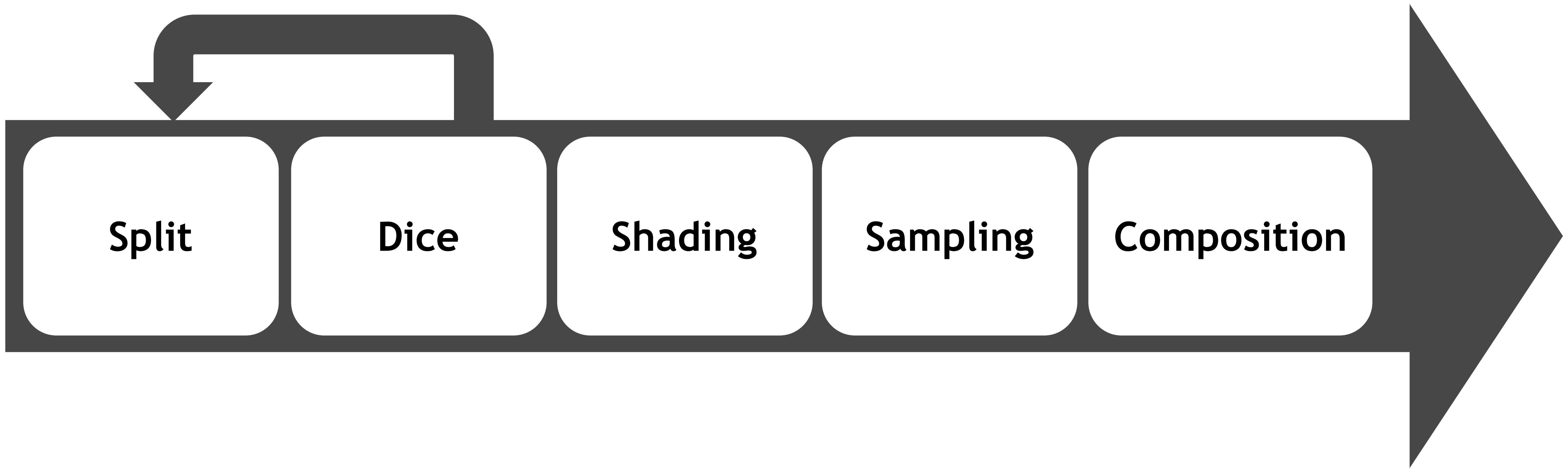
Pipeline Detail (5 Lectures)

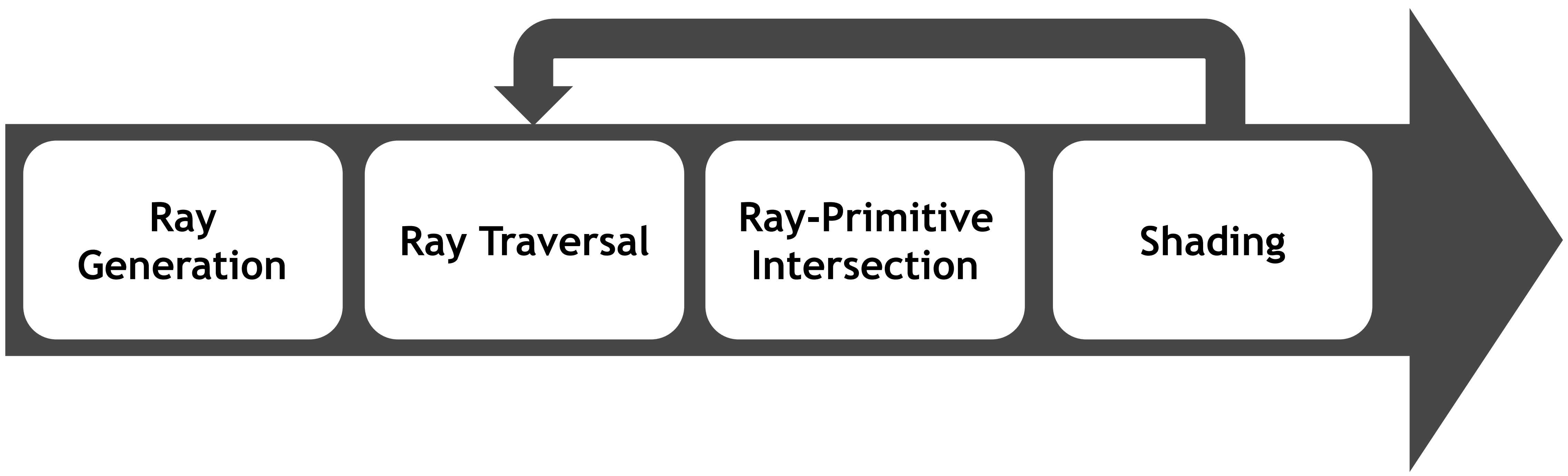


Lecture 12: Juggling the Pipeline



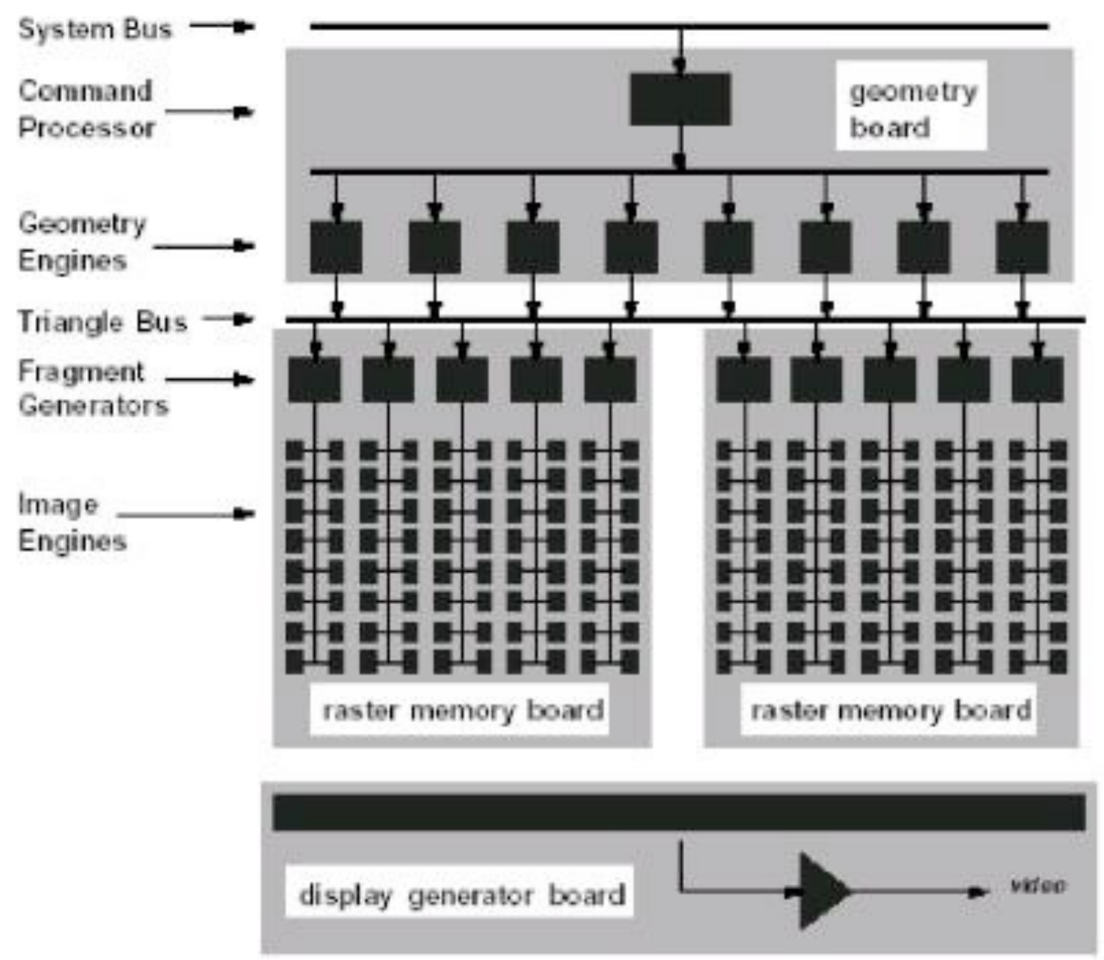






Lecture 13: Parallelizing the Pipeline

SGI RealityEngine



UNC Pixel Flow

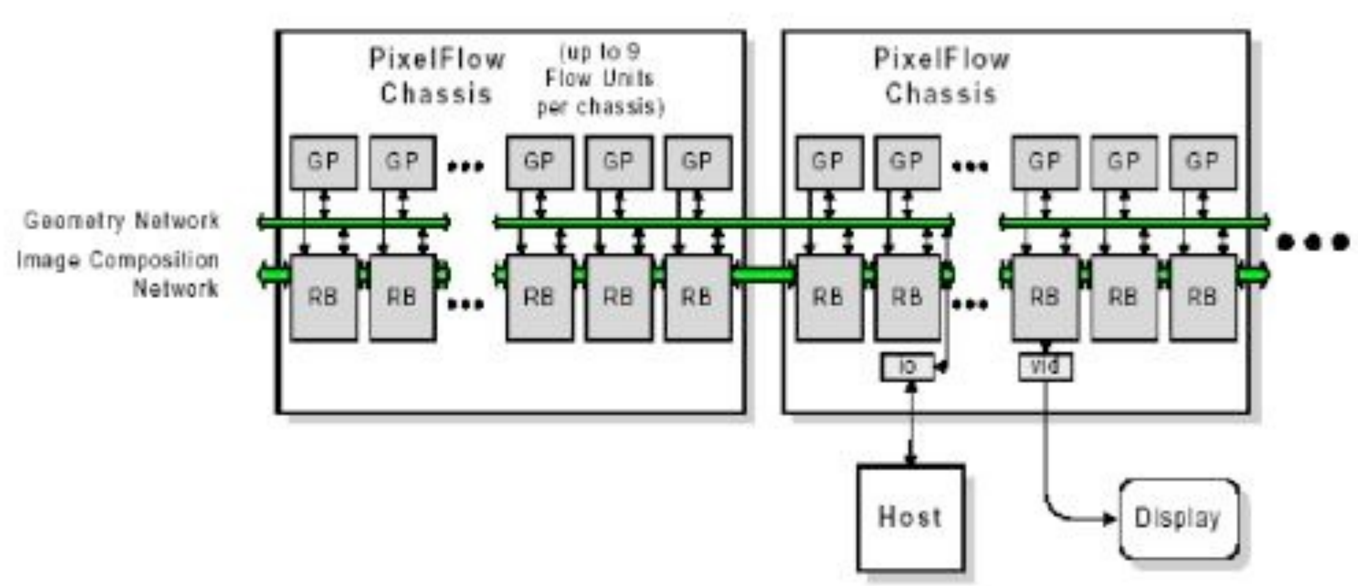
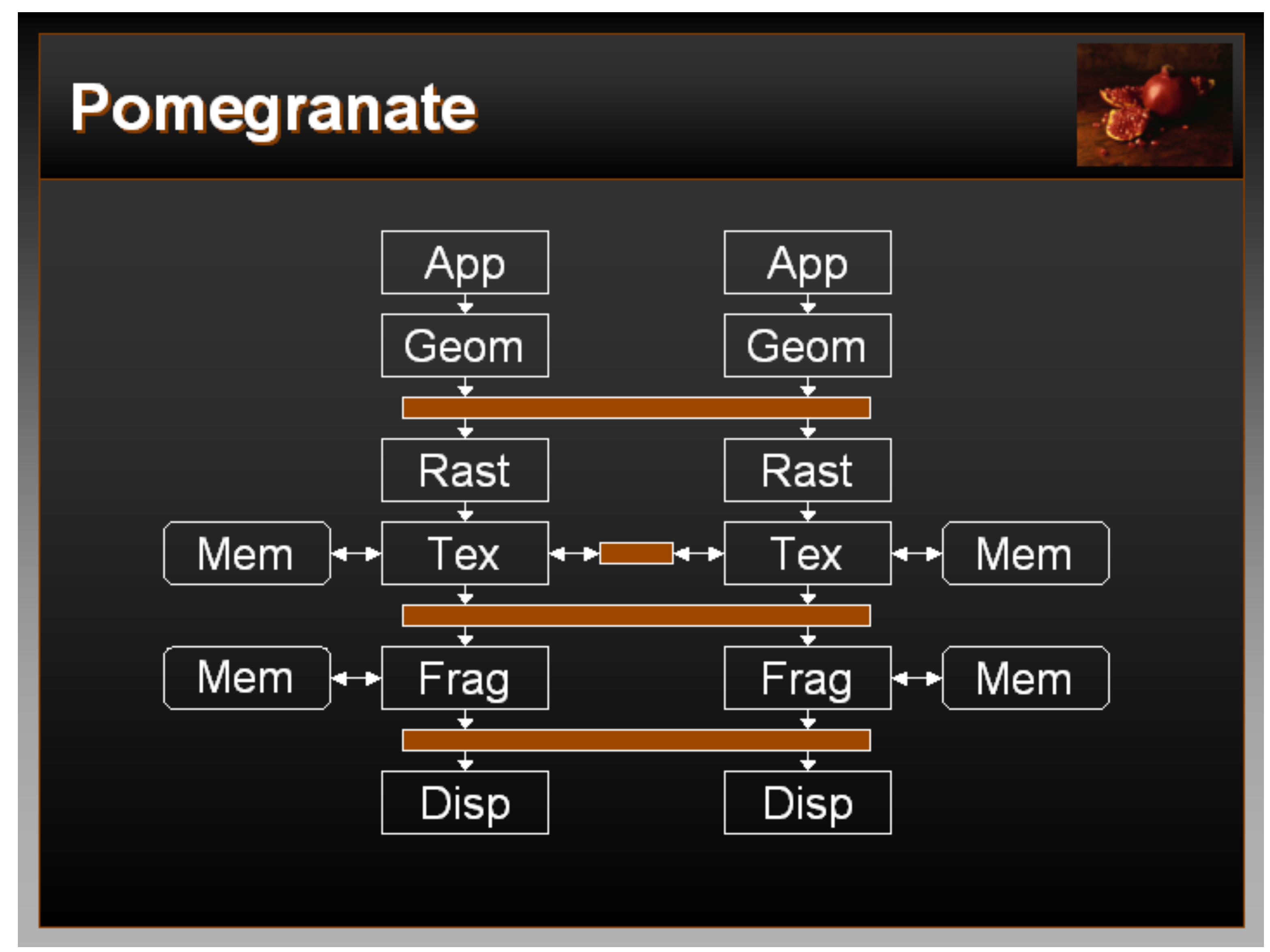


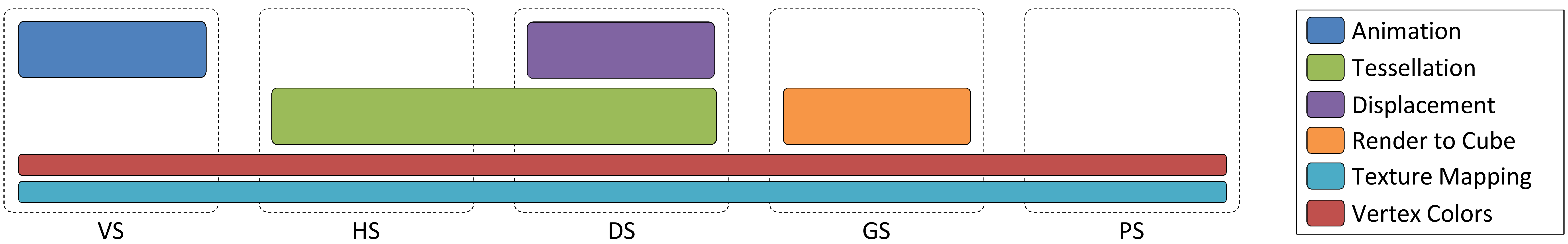
Figure 1: Typical PixelFlow System.

From J. Poulton, J. Eyles, S. Molnar, H. Fuchs, Pixel Flow: The Realization



[courtesy of Hanrahan/Akeley/Eldridge]

Guest Lecture: Tim Foley, NVIDIA Research



Guest Lecture: Dave Shreiner, Unity Labs

Guest Lecture: David Kanter, Real World Tech

real world technologies

CLOUD MOBILE GRAPHICS CHIPS SOFTWARE

CPUS GPUS SEMICONDUCTORS STRATEGY FORUMS

A Look Inside Apple's Custom GPU for the iPhone

October 25, 2016 by David Kanter

Previously, Apple's iPhones and iPads used PowerVR GPUs from Imagination Technologies for graphics. Based on our analy created a custom GPU that powers the A8, A9, and 10 processors, shipping in the iPhone 6 and later models, and some iPac documents, we demonstrate that the programmable shader cores inside Apple's GPU are different from Imagination Technolc and offer superior 16-bit floating-point performance and data conversion functions. We further believe that Apple has also dev shader compiler and graphics driver. The proprietary design enables Apple to deliver best-in-class performance for graphics, that use the GPU, such as image processing and machine learning.

Read More (2 pages) Disc

Tile-based Rasterization in Nvidia GPUs

August 1, 2016 by David Kanter

Starting with the Maxwell GM20x architecture, Nvidia high-performance GPUs have borrowed techniques from low-power mo architectures. Specifically, Maxwell and Pascal use tile-based immediate-mode rasterizers that buffer pixel output, instead of screen immediate-mode rasterizers. Using simple DirectX shaders, we demonstrate the tile-based rasterization in Nvidia's Ma GPUs and contrast this behavior to the immediate-mode rasterizer used by AMD.

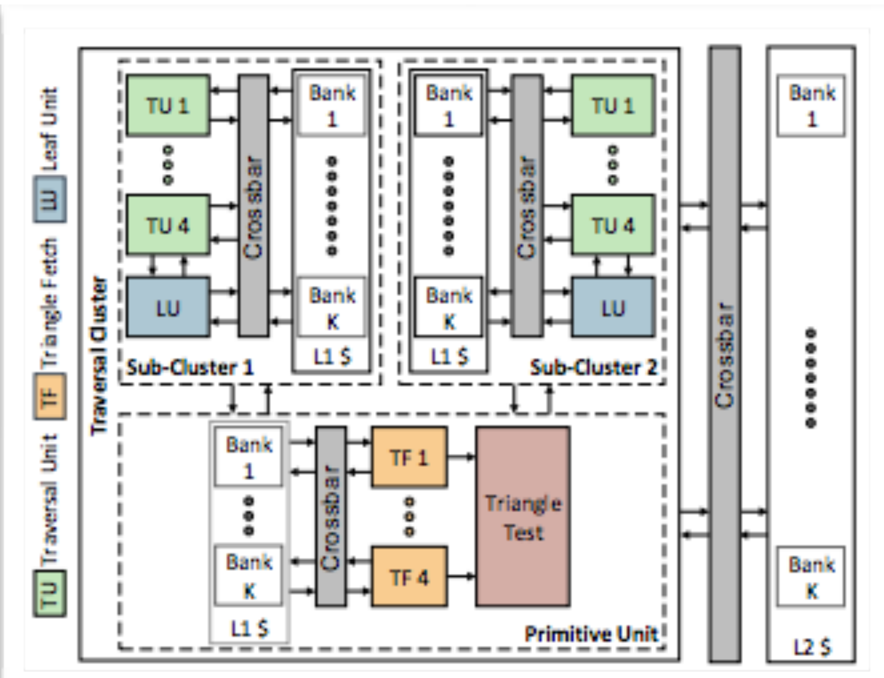
Read More Disc

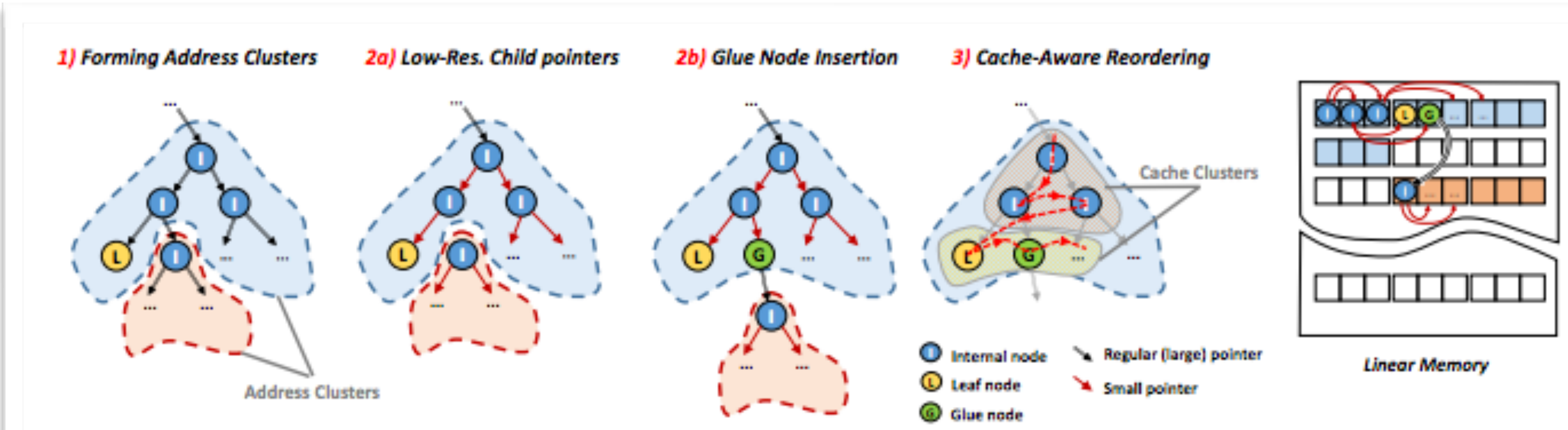
What's Next for Moore's Law? For Intel, III+V = 10nm QWFETs

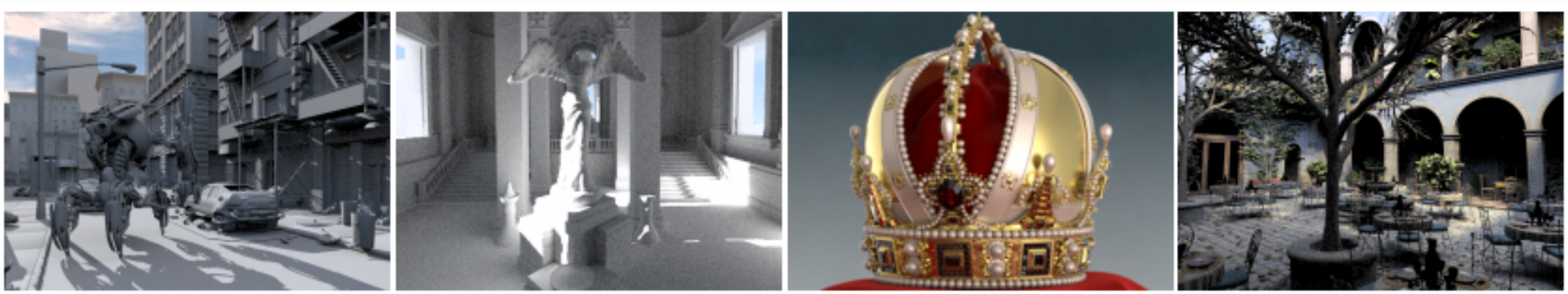
April 21, 2015 by David Kanter

On the eve of the 50th anniversary of Moore's Law, the future of silicon CMOS is an open question. With rising costs and unc

Guest Lecture: Karthik Vaidyanathan, Intel Research







Showdown | Temple | Crown | San Miguel

Last Lecture, Project Presentations

Lecture 1: Introduction / Overview

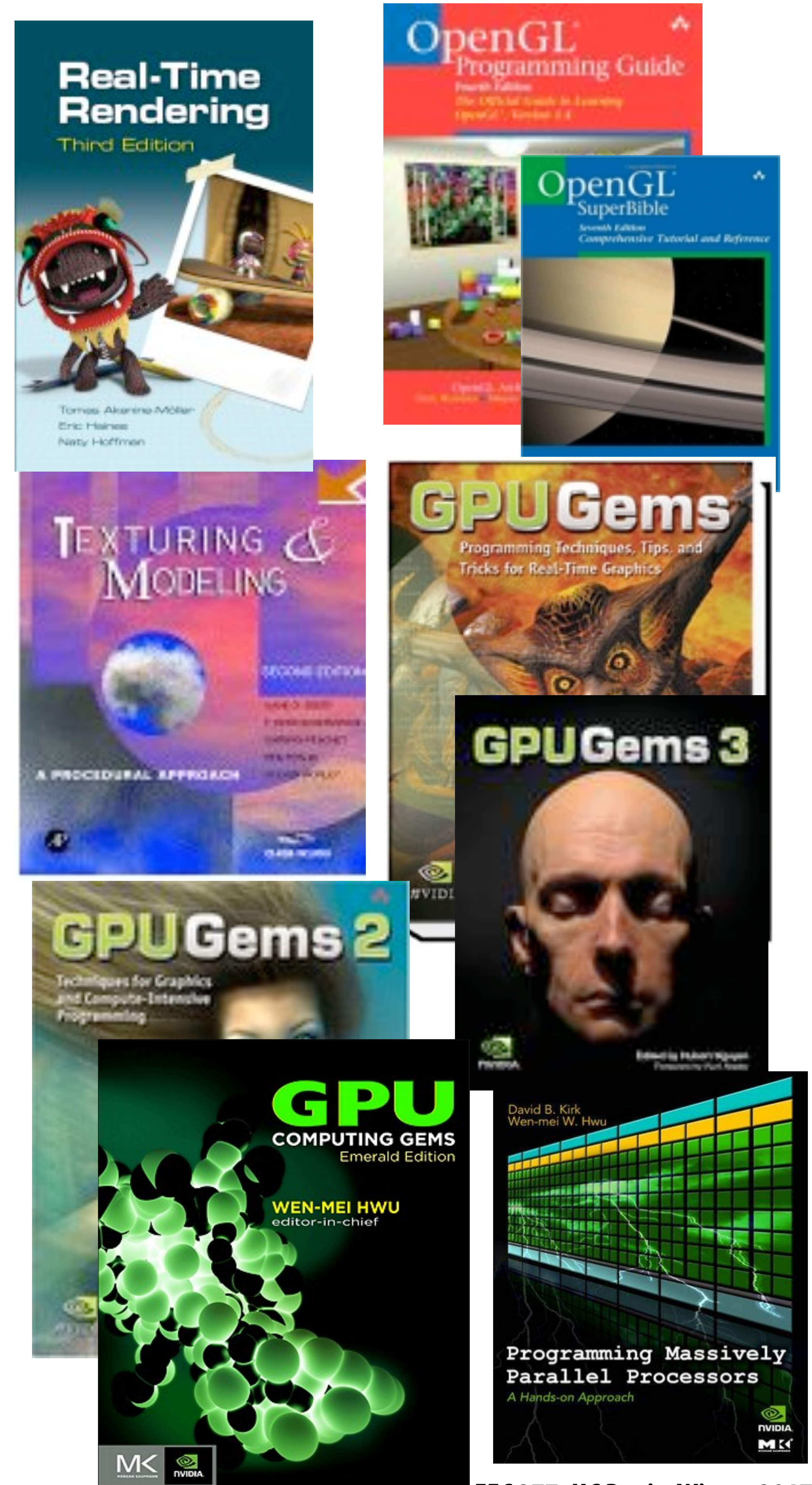
- Why graphics hardware?
- Generations of graphics hardware
- Class outline
- Administrivia

Background required

- Computer architecture (at the level of EEC 170 or ECS 154B)
- Computer graphics (at the level of ECS 175)
  - Not vital—Assignment 1
  - Need to come up to speed fast though!
- Reasonable familiarity with the C programming language, and OpenGL
- 3 units

Textbooks

- None required
- Best book: “Real-Time Rendering”, Tomas Akenine-Möller, Eric Haines, Naty Hoffman
  - http://www.realtimerendering.com/
- OpenGL tutorial and reference (online resources), plus red and blue books
- “Texturing and Modeling: A Procedural Approach”, Ebert et al.
- GPU Gems 1, 2, 3
- GPU Computing Gems 1, 2
- Hwu/Kirk: “Programming Massively Parallel Processors”



No exams, no notes

- Please come to class
- Please listen
- Please participate

- I'll post slides online (Canvas)

Assignments

- 1. 3-page paper on graphics pipeline written for the layman (due T 17 January) (10% of grade)
  - Guidelines are in Canvas assignment

Assignments

- 2. Characterization of graphics/GPGPU hardware
  - Preferably group of 2; 1 is OK
  - Triangle vs. fill rate
  - Either graphics analysis or GPU computing analysis
  - Deliverable: writeup and code
  - Due T 7 February
  - 30% of grade

Assignments

- 3. Project.
  - Groups of 1–2+
  - You choose topic
  - Complete, limited scope projects better than broad, incomplete projects
  - Be bold
  - Talk to me about it

- Requires preproposal (14 Feb), proposal (21 Feb), revision if necessary (28 Feb), report (16 Mar), presentation (16 Mar)
- 50% of grade

Software Resources

- NVIDIA CUDA, OpenCL—programming systems for GPUs
- GLSL—portable shading language
  - But feel free to use others! (e.g., HLSL, etc.)
- Embree—ray tracing (Intel, https://embree.github.io/)
- Attila—GPU simulator (http://attila.ac.upc.edu/wiki/index.php/Main_Page, but not currently maintained since Oct 2015

- (check www.realtimerendering.com)

Guidelines

- Papers must be:
- Well written and formatted
- Properly referenced
- Intellectually honest

- Prime grading standard: intellectual merit

- OK to fail, so long as you demonstrate something interesting.

Expectations

- Participate in class!
- Ask questions!
- Offer answers to my questions!
- Explore interesting topics!
- Challenge our guest speakers!
- Act honorably and honestly!
- NOT a lecture class. NOT a passive experience. ACTIVE learning.
- Most common problem: Not getting started
- Ask for help if you need it!

Feedback

- Feedback is welcome!
  - Preparing lectures (Length? Boredom?)
  - Lecturing
  - Grading
  - Writing assignments …
- Email is welcome!
- Come talk to me!
  - I will have an office hour but experience says you should just email for appointments