

The Rendering Problem

Part II: Architectures

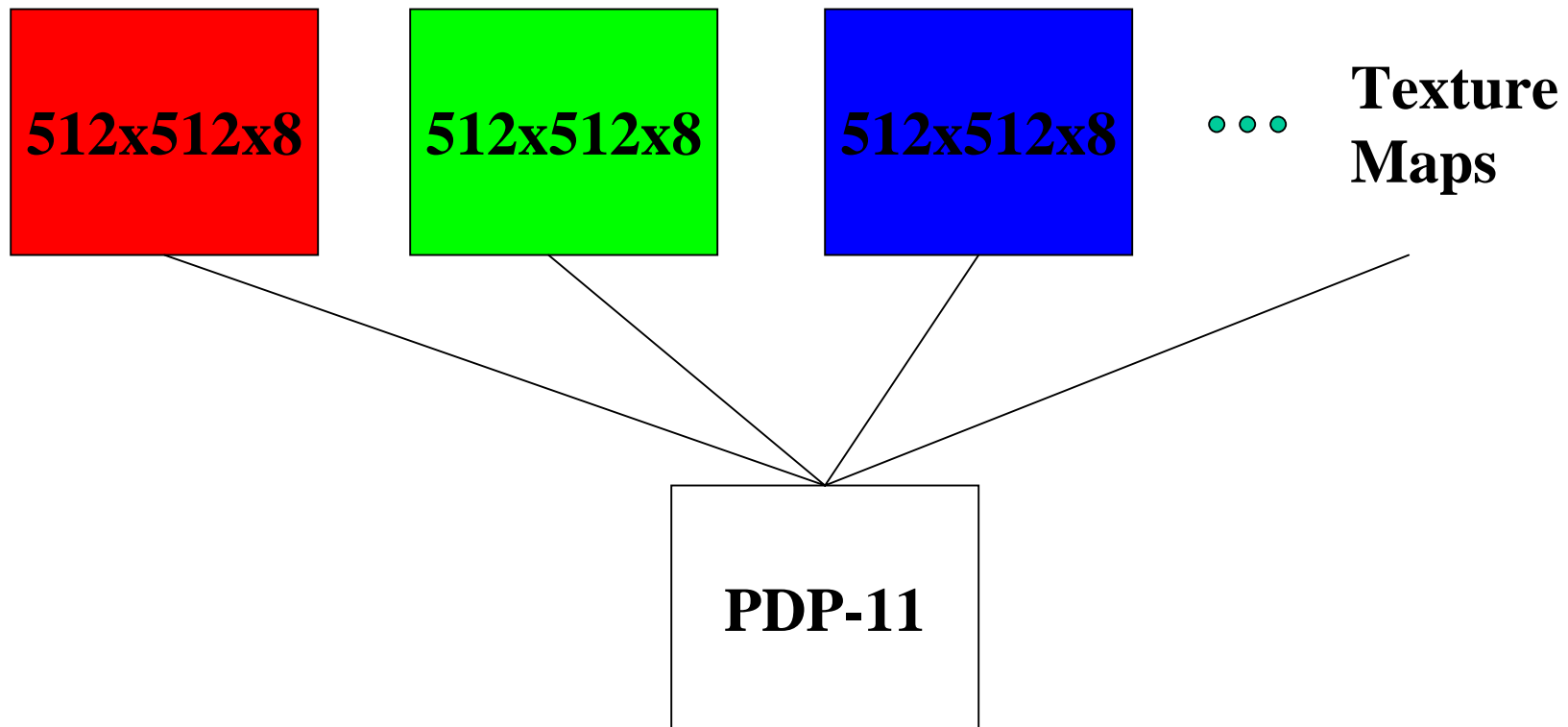
Turner Whitted
Hardware Devices Research Group



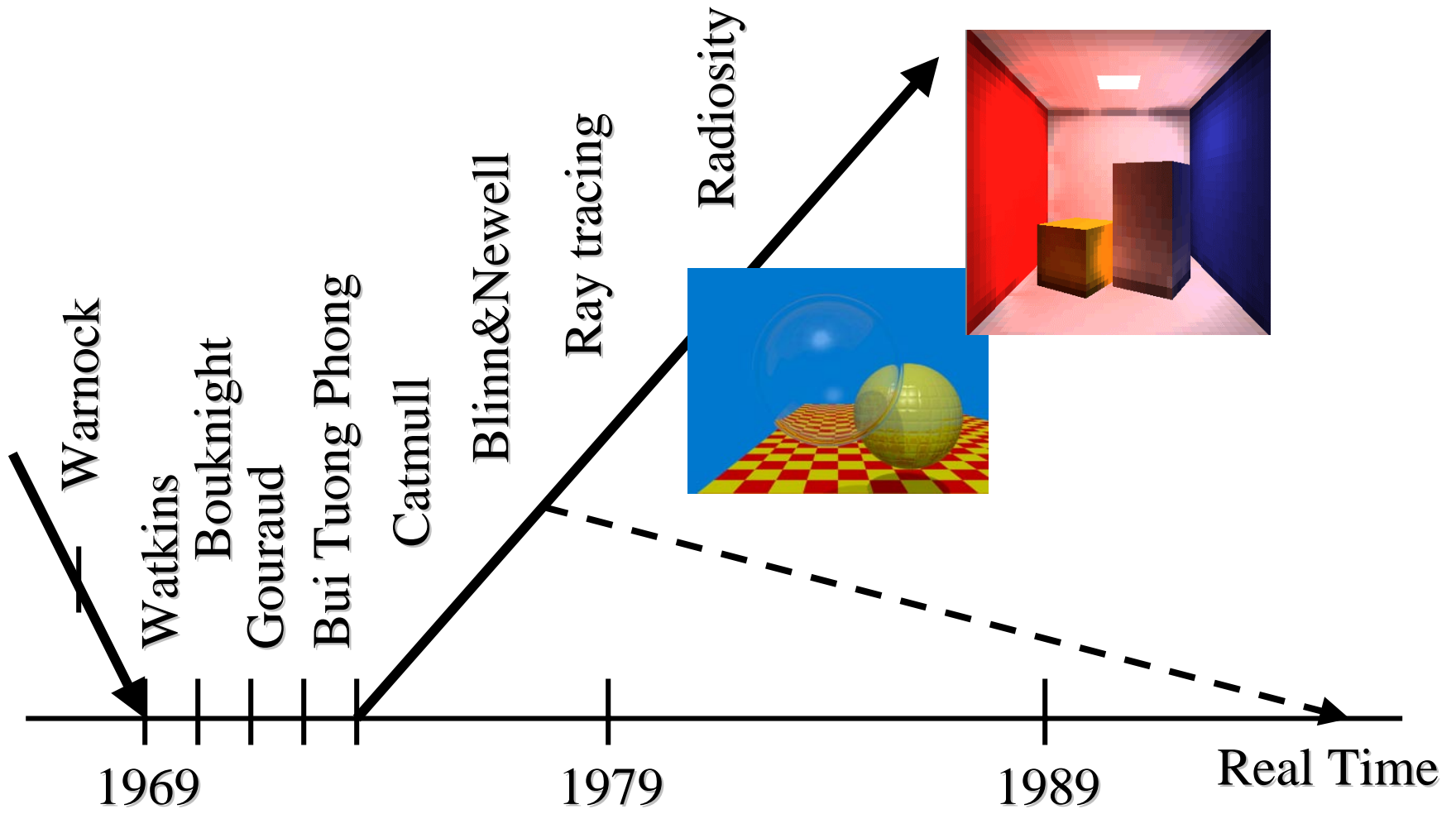
Well-worn Strategies

- Memory intensive algorithms
 - texture mapping
 - IBR
- Processor intensive algorithms
 - geometry-based methods
 - scan conversion
 - ray/object intersection
 - matrix inversion

NYIT Rendering Machine memory intensive! (1978)



Extrapolation of Trends



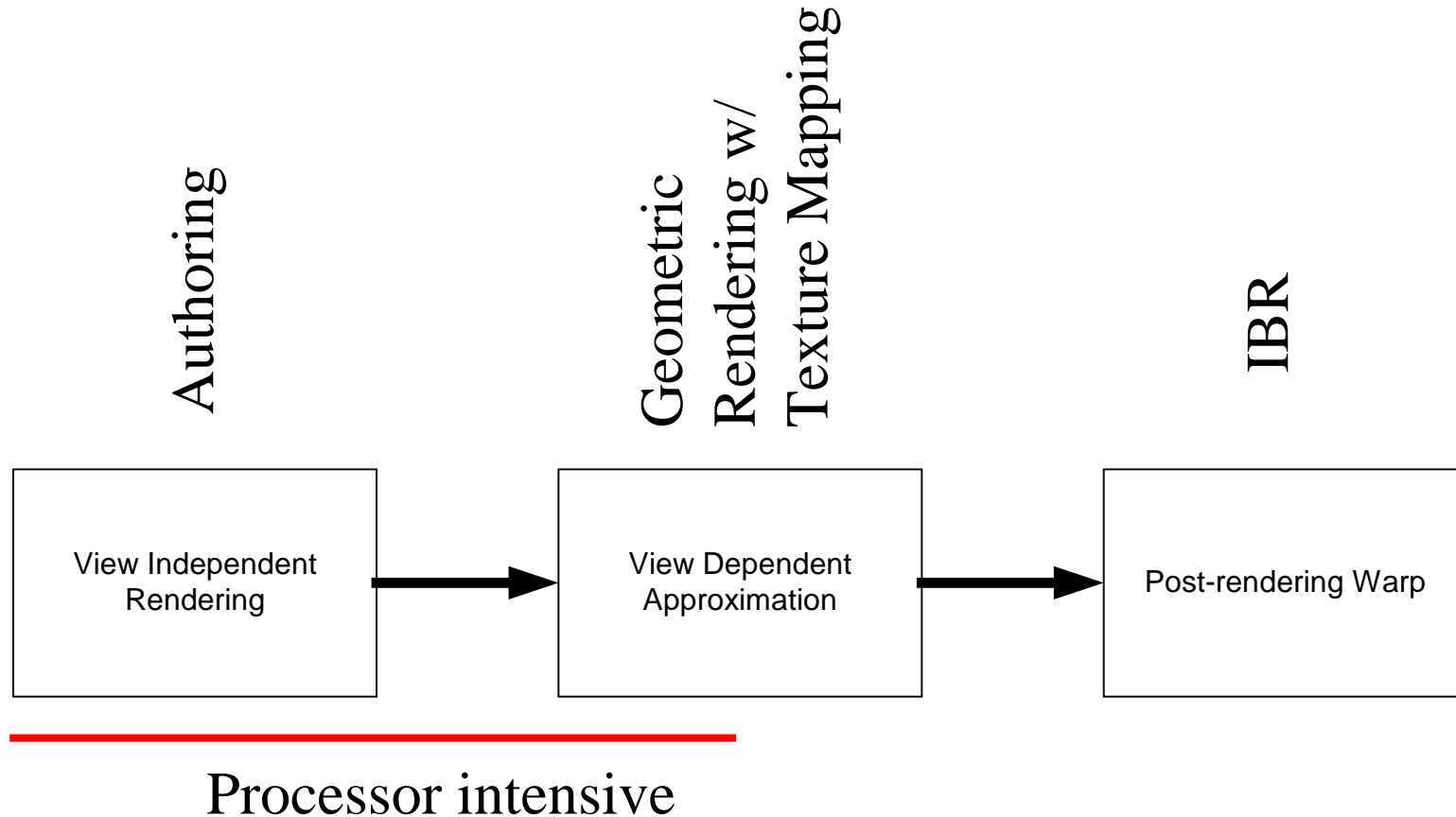
Real Time Realism

- Is there a workable architecture?
- Does it lie within the spectrum of the “well-worn strategies”?
- If not, what’s a new candidate?

Ideas That Never Got a Fair Chance

- Special function register sets
 - FPGA implementation
- (Useful) instruction set extensions
- Wave model rendering
- add your own.

The Default Pipeline



Processor Intensive Guidelines

Processor Intensive Guidelines

- The Kajiya Principle
 - “Ray tracing isn’t too slow; computers are too slow.” (1986)

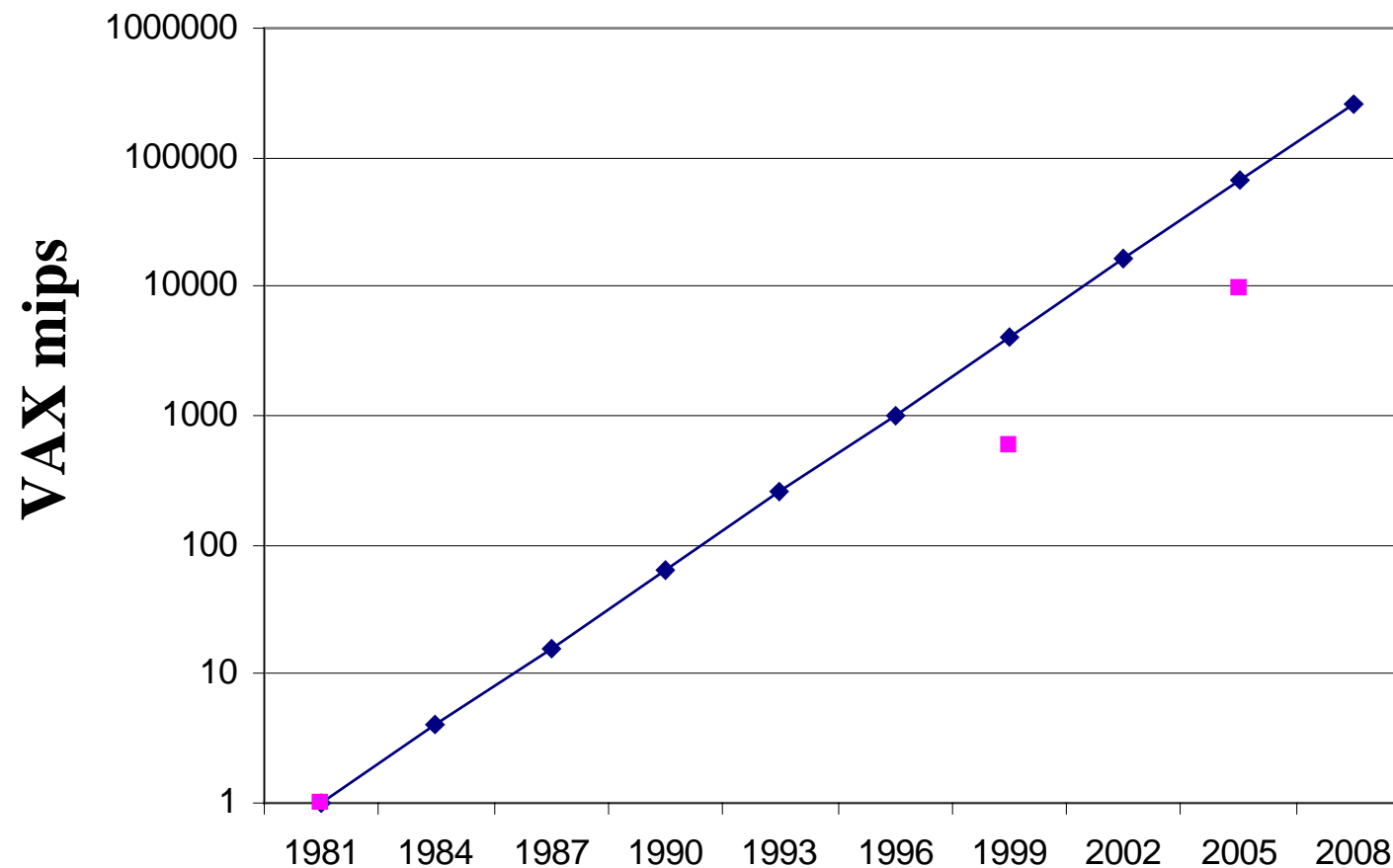
Processor Intensive Guidelines

- The Kajiya Principle
 - “Ray tracing isn’t too slow; computers are too slow.” (1986)
- Whitted’s Method
 - If someone else is willing to solve the problem, then use their result.

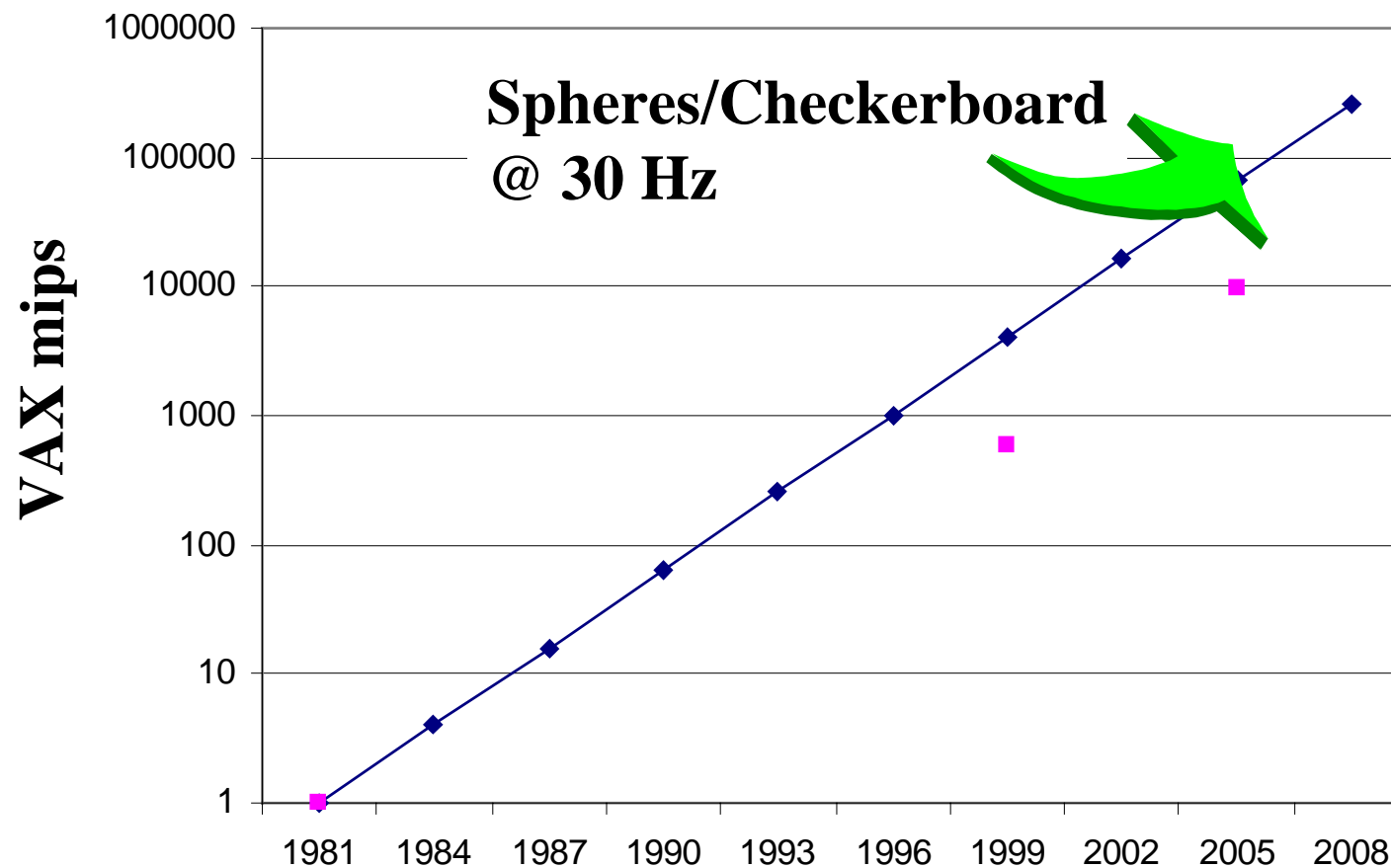
Processor Intensive Guidelines

- The Kajiya Principle
 - “Ray tracing isn’t too slow; computers are too slow.” (1986)
- Whitted’s Method
 - If someone else is willing to solve the problem, then use their result.
- Moore’s Law
 - Processors get faster fast.

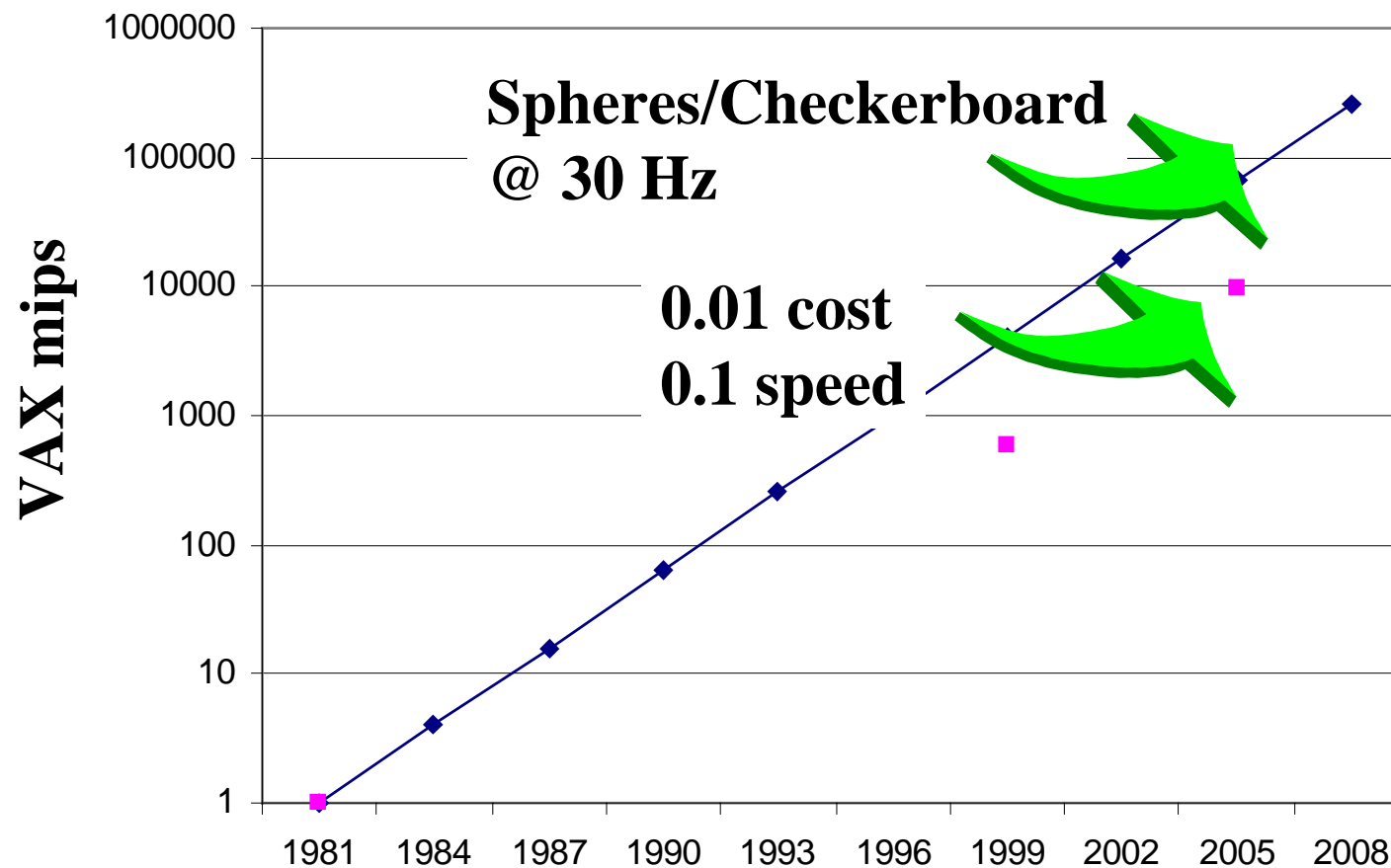
Processor Intensive Methods: *the Moore's Law numbers*



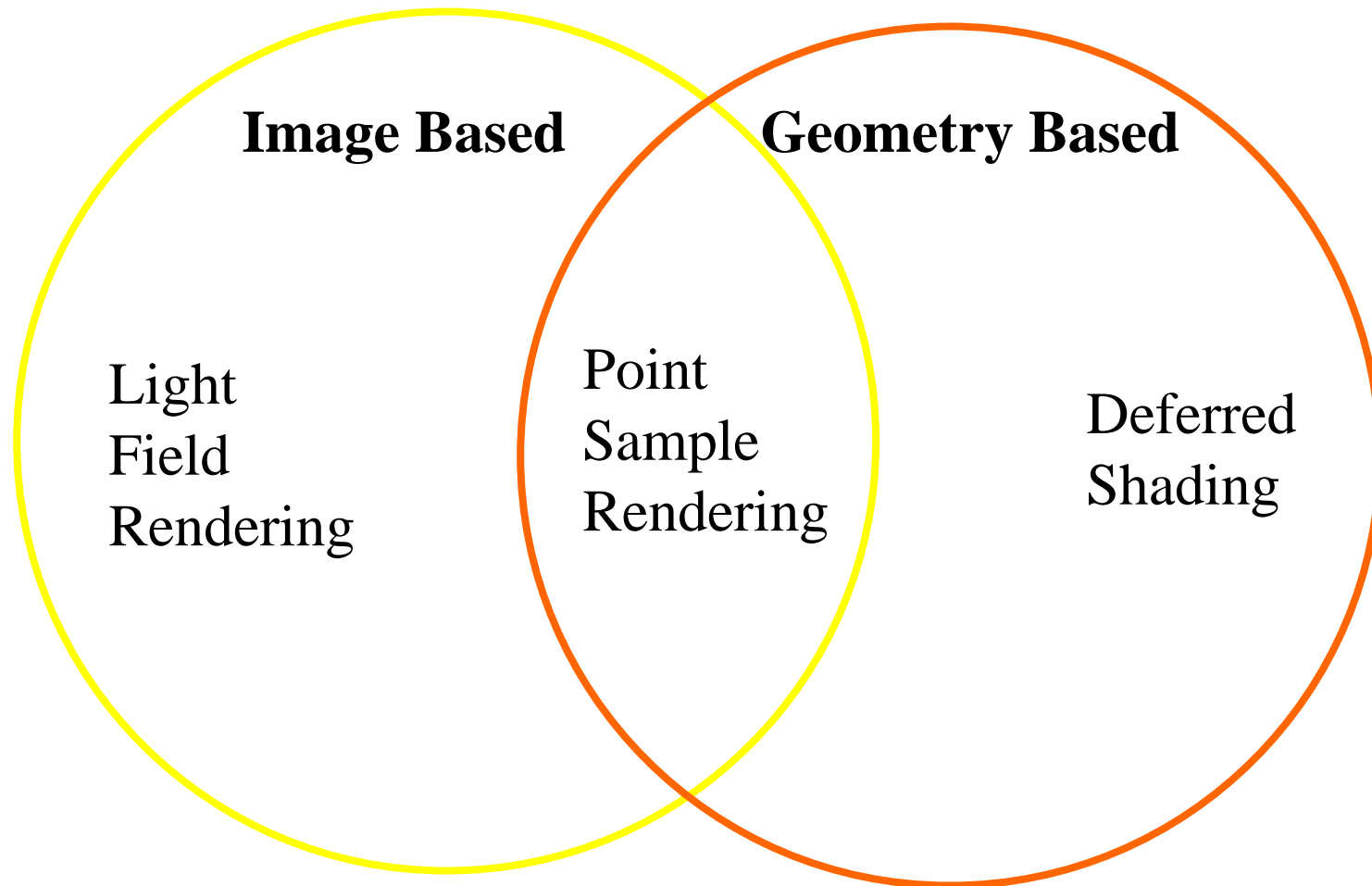
Processor Intensive Methods: *the Moore's Law numbers*



Processor Intensive Methods: *the Moore's Law numbers*



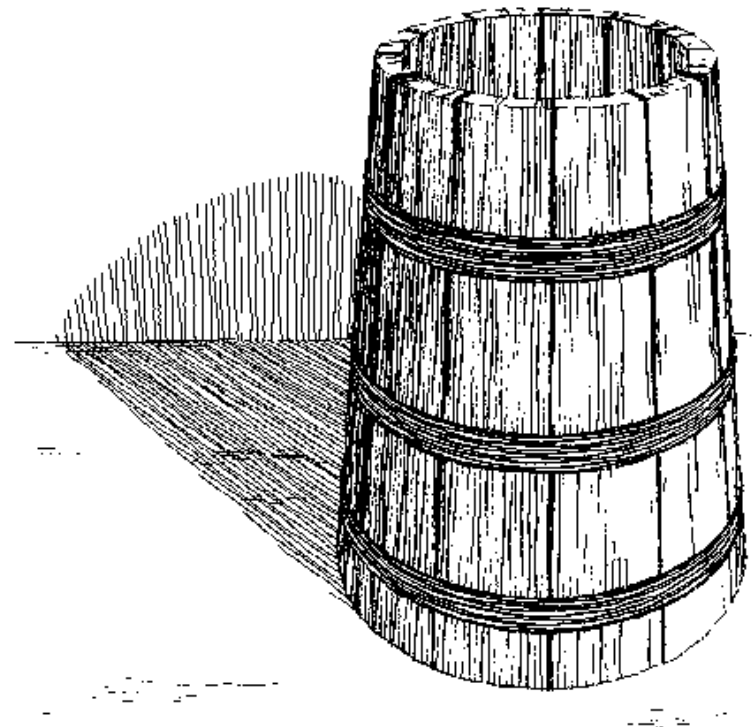
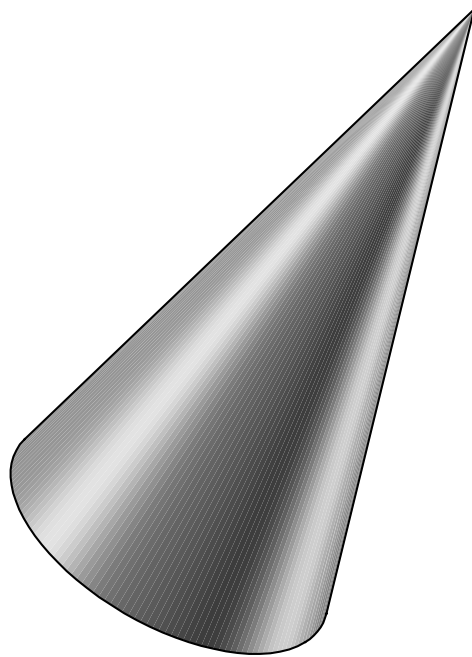
Alternative architectures



Abstractions/Representations

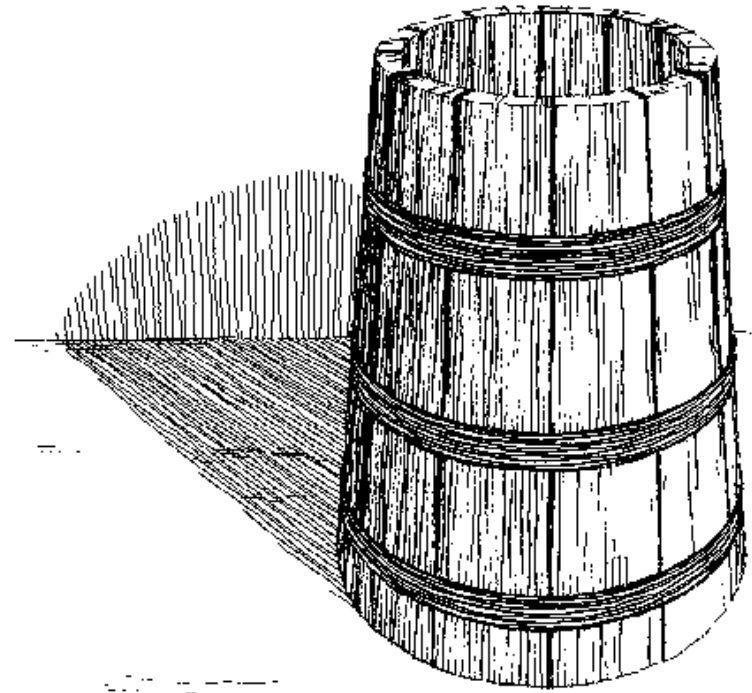
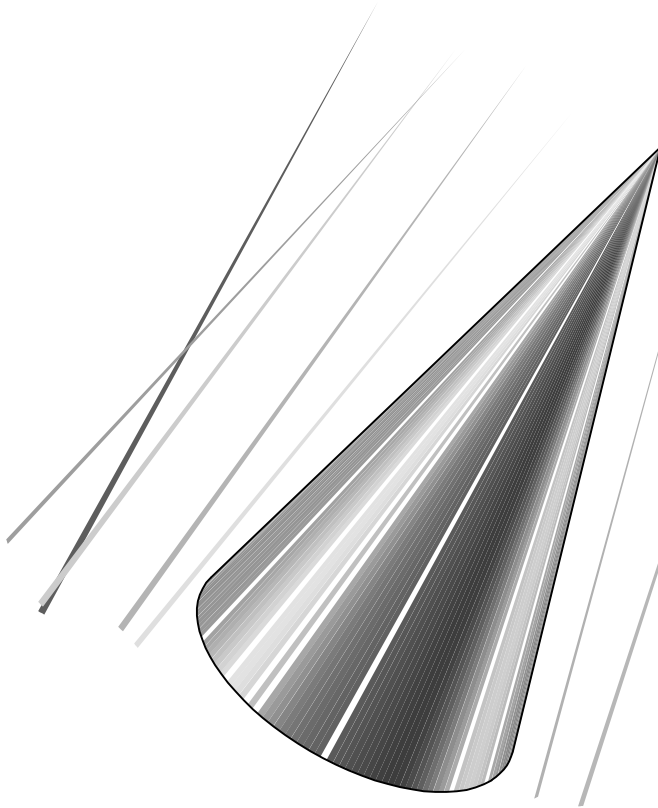
- Pixels are a very thin layer of abstraction over physical bits in display memory.
- Polygons are a convenient abstraction for geometric operations within a rendering process.
- Strokes are an effective abstraction for conveying information to human viewers.

Polygons and Pixels vs. Strokes

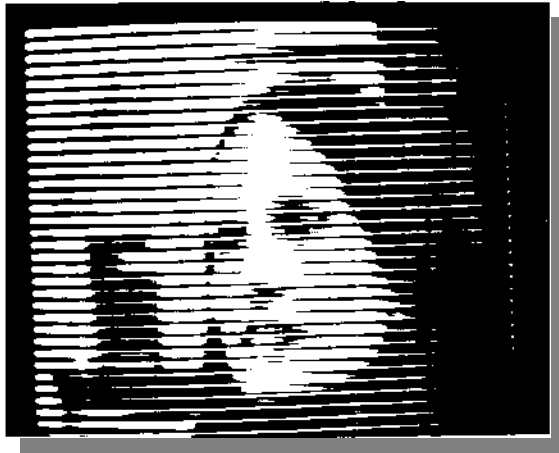


Polygons and Pixels

There is nothing sacred about either.

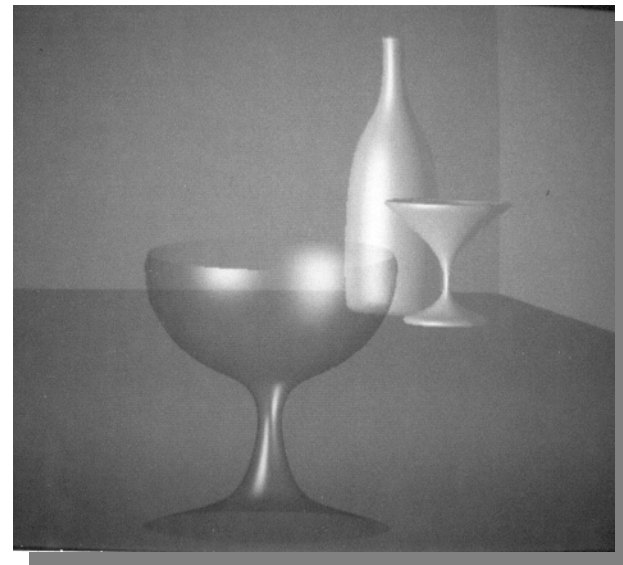


Why Not Strokes?



+

inexpensive memory =



Architecture/Representation

- Current representations are an artifact of displays.
- Architecture follows from representation.
- There is no reason for representations to be related to the displays
 - except for a backend mapping.
- Rates should be adaptive
 - shaders ought not to be mapped to pixels

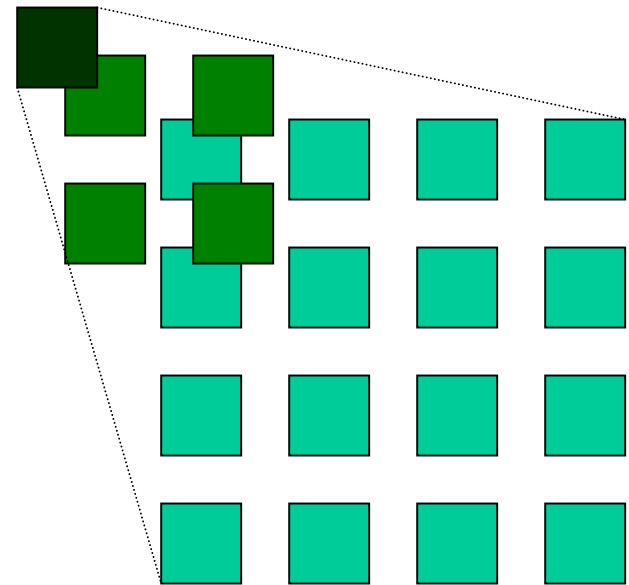
Representations¹

- Surfaces
- Volumes
- Points
- Projections
 - IBR
- *Abstractions?*
- Visibility paths
 - ray tracing
- Illumination paths
 - radiosity
- Illumination regions
 - various shadow methods

¹And of course, add the adjective *hierarchical* in front of each.

So, What Does an Hierarchical Renderer Look Like?

- It's parallel
- It's recursive
- It needs indeterminate amounts of memory

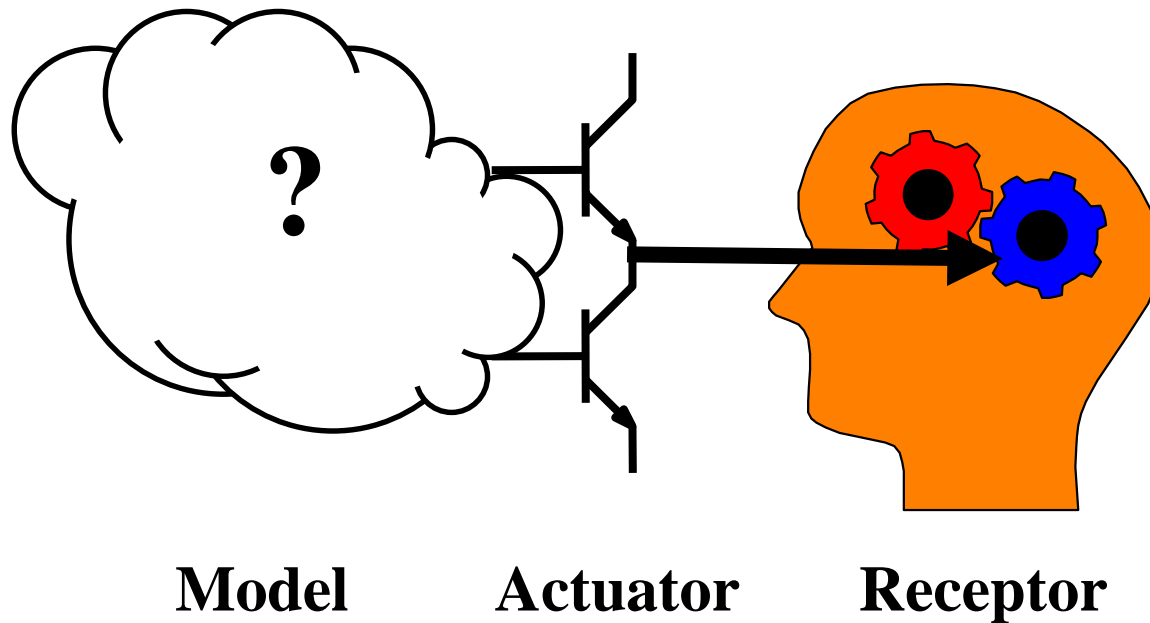


Other People's Models and Abstractions

- Music
 - Physical models
- 3D audio rendering
 - HRTF, artistry
- Speech synthesis
 - Component physical model
- Speech analysis
 - Complex!!!
- Low-level behaviour
 - HMMs
- Vision
 - ad hoc
- Particle physics
 - Whatever seems to work
- Linear systems
 - Frequency domain

Out-of-the-box Abstractions

- Frequency domain
- Perceptual models



Summary Observations

- The prevailing architectures are ad-hoc
 - what *can* be implemented rather than what *should* be implemented
- Moore's law falls short in the near term
- Representation is still a stumbling block
- We have yet to discover the right model and actuator for visual perception.

The Rendering Problem

- Yes, there *is* a rendering problem.
- Visual perception is demanding in ways that we don't adequately understand.
- Part of the problem is us - we are not tinkering with new models and architectures.