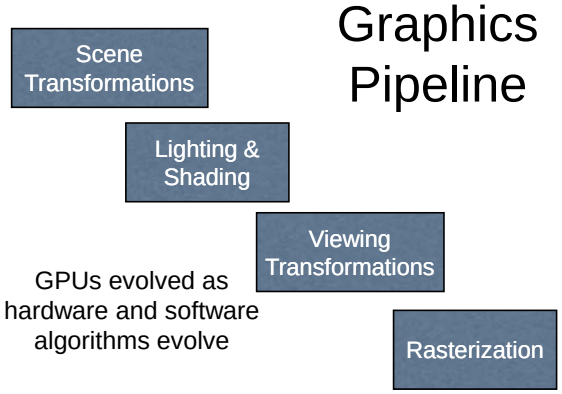
GPU

Graphics Processing Unit

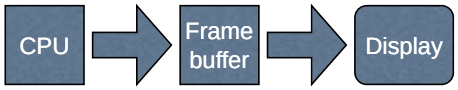
Graphics Pipeline



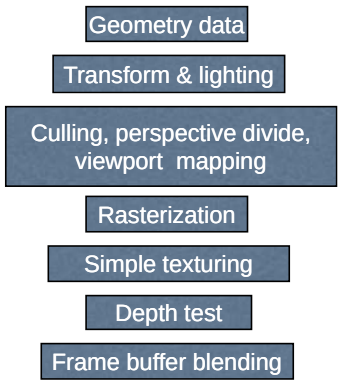
GPUs evolved as hardware and software algorithms evolve

Early Graphics

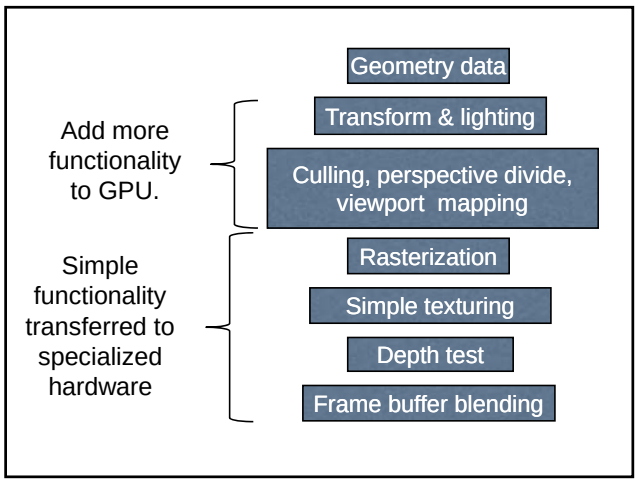
- Originally, no specialized graphics hardware
- All processing in software on CPU,
- Results transmitted to frame buffer
  - first, external frame buffers
  - later, internal frame buffers.



More detailed pipeline

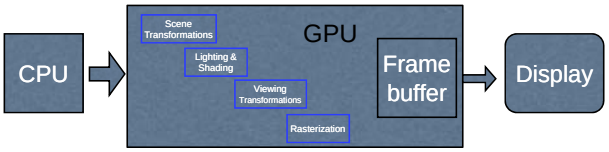


Simple functionality transferred to specialized hardware.



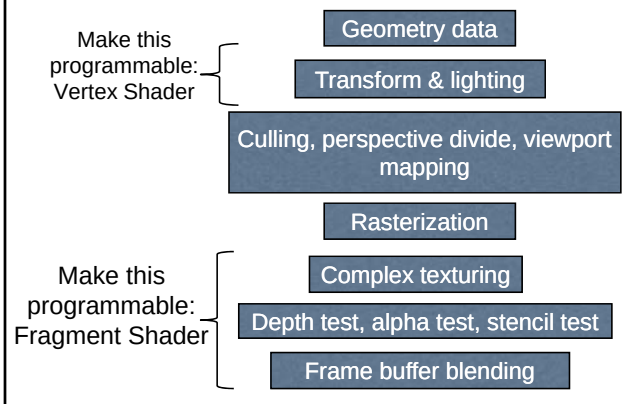
Fixed function GPU pipeline

- Pipeline implemented in hardware
- Each stage does fixed task
- Tasks are parameterized
- Inflexible – fixed, parameterized functions
- Vector-matrix operations (some parallelism).



Technology advances

- Hardware gets cheaper, smaller, and more powerful
- Parallel architectures develop
- Graphics processing get more sophisticated (environmental mapping, displacement mapping, sub-surface scattering)
- Need more flexibility in GPUs.

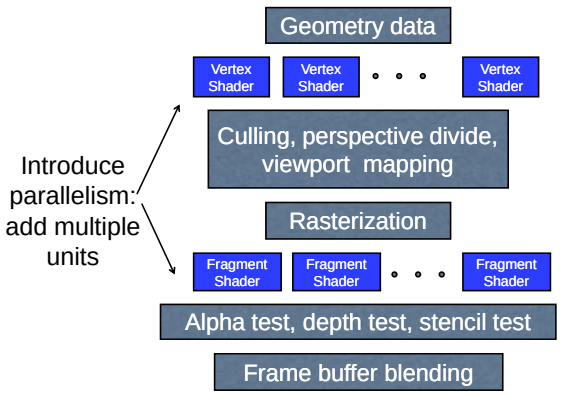


Vertex shader

- Graphics systems: convert everything to triangles
- Pass vertices, normals, colors, texture coordinates to GPU processor
- GPU: vertex-based computations,
  - Independent of other vertices
- Later, assemble into triangles.

Fragment shader

- Fragment is triangle clipped to pixel
  - Interpolate values
- Multiple textures, Alpha, stencil, depth
  - Independent of other fragments
- Blend with contents of frame buffer.



Shading language

- Shade trees -> Pixar's Renderman shader

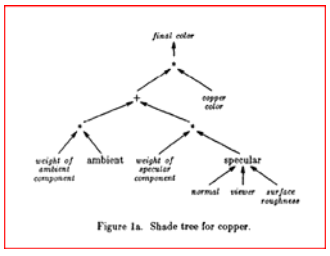


Figure 1a. Shade tree for copper.

Shader Language

- Low level (like assembler) but high-level language compilers: nVidia's Cg
- 4 component floating point data type
- SIMD

Cg: C-based graphics program

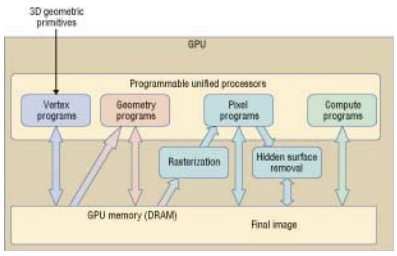
- Array & structures
- Flow control
- Vectors & matrices
- No memory allocation, file I/O

Power

- GPUs have moved away from the traditional fixed-function 3D graphics pipeline toward a flexible general-purpose computational engine.
- The raw computational power of a GPU dwarfs that of the most powerful CPU, and the gap is steadily widening.
- GPUs have moved away from the traditional fixed-function 3D graphics pipeline toward a flexible general-purpose computational engine

Next: unify shaders

- One set of shaders
- Allocate to either vertices or fragments



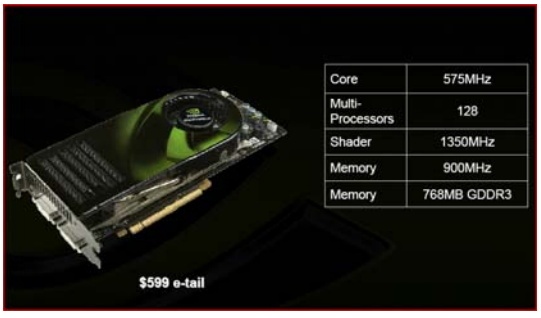
Pipeline evolved

Evolved pipeline

GPGPU

- Make GPU more general – adapt certain types of programs to it's pipelined, parallel architecture
- Single GeForce 8800 chip achieves a sustained 330 billion floating-point operations per second (Gflops) on simple benchmarks
- Cost-effective: graphics driving demand up, supply up, price down for GPUs
- Finding uses in non-graphics applications.

GeForce 8800 GTX



More general: NVIDIA's CUDA

- More general data parallel model
- Decompose across threads
- Sharing and communication between threads..