Introduction

Hardware features improve with time

- Wireframe
- Gouraud shaded
- Texture mapped
- Multi-texturing, with flexible calculations
- Fully programmable shading (PixelFlow)

It will get even better

- Rate of change has accelerated:
  PC graphics companies produce a new generation of chips every year.
- Computation is cheap; not a big deal to add more of it to the graphics pipeline
- Trends in graphics hardware:
  - More flexible pixel (fragment) computations
  - Programmable vertex computations

Topics in today's course

- Introduction
- Fully programmable hardware (PixelFlow)
- Proposals for future hardware
- Shading algorithms for current hardware
- Procedural shading on current hardware
- Q & A

History of off-line shading

- Simple shading models
- Complex, parameterized shading models
- “Hack the renderer source code”
- Abstractions for programming
  - Shade trees
  - Perlin’s pixel stream editor
  - RenderMan

What’s different about hardware?

- PC graphics hardware is ~100x faster than CPU rendering, with the same die area.
- Must ask “why is graphics hardware fast”?
  - Heavily pipelined
  - No branches or data dependencies
  - Aggressive pre-fetching from memory
  - Fixed algorithm

Three options for HW shading

- Creatively use existing hardware
  - It's amazing what can be done right now.
- Develop completely different HW architectures
  - PixelFlow took this path
- Enhance existing hardware architectures, without abandoning pipeline model
  - Introduce carefully-designed flexibility, while maintaining high performance
  - PC graphics cards are taking this path
  - Must adapt and redesign algorithms for this HW
  - This approach is very promising

Why complex shading?

As visually oriented animals, people are very sensitive to subtleties of shading, and that sensitivity expresses itself as a nearly limitless demand for subtle, flexible control over shading. At the same time, the more successful a program is, the wider the range of physical reality it is expected to duplicate. Both of these facts place extraordinary demands on the shading portion of any renderer.

-- Upstill

Most real-time 3D applications can use procedural shading

- Entertainment
  - Quake III already uses a shading language
- Visual Simulation / VR
  - Realism is particularly useful for training
- Scientific and Medical Visualization
  - Flexible mapping of multi-dimensional data to 2D image colors.
- WWW
  - Displaying realistic images of merchandise, etc.

What the future holds

- Increasingly flexible PC graphics hardware
- Procedural shading languages specifically designed for real-time hardware
- Continued development of shading algorithms optimized for hardware's constraints