

Caltech
Department of
Computer
Science

Monday,
October 21st
12:30 - 2pm
Lauritsen 123

Bill Mark
Nvidia Corporation
& UT Austin



Special Lecture: Hacking the GPU Course CS101-3



```
L2weight = timeval - floor(timeval);  
L1weight = 1.0 - L2weight;  
ocoord1 = floor(timeval)/64.0 + 1.0/128.0;  
ocoord2 = ocoord1 + 1.0/64.0;  
L1offset = f2tex2D(tex2, float2(ocoord1, 1.0/128.0));  
L2offset = f2tex2D(tex2, float2(ocoord2, 1.0/128.0));
```

Programmable Graphics Hardware: Beyond Real- Time Movie Rendering

The latest generation of 3D PC graphics hardware (GPUs) includes highly-programmable floating-point vertex and pixel-fragment processors. These processors are flexible enough to support high-level C-like programming languages.

GPU designers have added programmability to these GPUs mostly to support procedural shading capabilities similar to those used in off-line movie rendering. But, much of the impact of these GPUs may come from the fact that they are the first highly parallel processors that are deployed on every desktop and are user programmable. The stream-processing programming model used by these GPUs can be used to efficiently support a wide variety of algorithms, including ray tracing and various types of physical simulation.

The speaker lead the design effort at NVIDIA for Cg, a C-like language for GPU programming. This talk will describe the design goals of Cg, explain some of the key design decisions in the language, and summarize Cg's programming model and capabilities.

