

# A Real-time Procedural Shading System for Programmable Graphics Hardware

Peter Stuart (presented by)

April 21, 2002

# Overview

- Analysis
- Global Variables
- Operations
- Light Shaders
- Example

## Analysis

- Function inlining
- Combine surfaces and lights
- No branches
- No memory random access

## Global Variables

- Set before shader is called
- Some examples:
  - $N$  surface normal
  - $B$  binormal vector ( $\frac{dP}{du}$ )
  - $E$  eye vector
  - $P_{obj}$  object space surface point

## Operations

- Built-in to shading language, not necessarily atomic to hardware
- Some examples:
  - *dot* dot product
  - *max* maximum of two values
  - *texture* Get the texel from a texture at the given coordinates.
- Noise is notably missing

## Light Shaders

- “perlight” data type
- Global “perlight” variables:
  - $C$  color of light
  - $L$  light vector
  - $H$  halfangle vector
- “integrate” operation

# Shading language example

---

```
surface shader float4
anisotropic_ball (texref anisotex, texref star)
{
    // generate texture coordinates
    perlite float4 uv = { center(dot(B, E)),
                        center(dot(B, L)),
                        0, 1 };

    // compute reflection coefficient
    perlite float4 fd = max(dot(N, L), 0);
    perlite float4 fr = fd * texture(anisotex, uv);

    // compute amount of reflected light
    float4 lightcolor = 0.2 * Ca + integrate(Cl * fr);

    // modulate reflected light color
    float4 uv_base = { center(Pobj[2]), center(Pobj[0]),
                    0, 1 };
    return lightcolor * texture(star, uv_base);
}
```



**Example contd.**



## References

- Kekoa Proudfoot, William R. Mark, Svetsolav Tzvetkov, Pat Hanrahan: *A Real-Time Procedural Shading System for Programmable Graphics Hardware*, Computer Graphics (SIGGRAPH 2001 Conference Proceedings)
- Kekoa Proudfoot: *Stanford Real-Time Shading Language*, <http://graphics.stanford.edu/projects/shading/docs/lang5.txt>