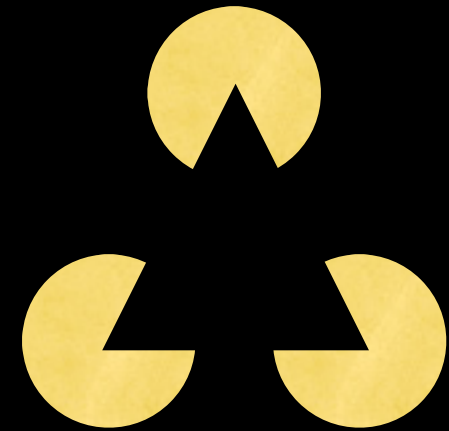


COMPSCI 715 Part 2

Lecture 11 - Non Photo Realistic Rendering

Non Photo Realistic Rendering

- Goal is to communicate a concept as clearly as possible
- Reduces information and removes redundant information
- Makes use of human perception and pattern recognition

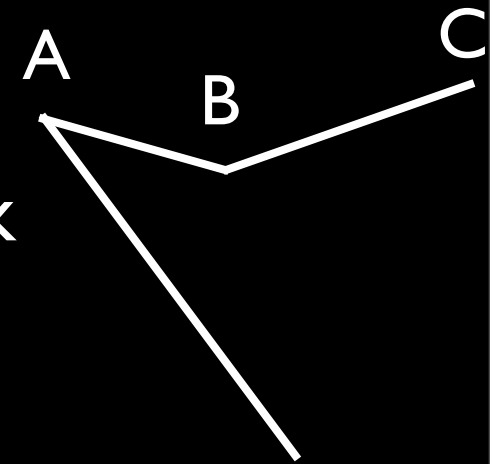


Non Photo Realistic Rendering

- Edge Rendering
- Gooch Shading
- Cell shading
- Painting Effects
- TF2 Lighting Model

Edge Rendering

- Uses edges to define the object
- 3 types of edge
 - Contour: edge shared by front and back facing triangles
 - Crease: edge shared between 2 front facing triangles
 - Boundary: edge not shared by another triangle



Edge Rendering

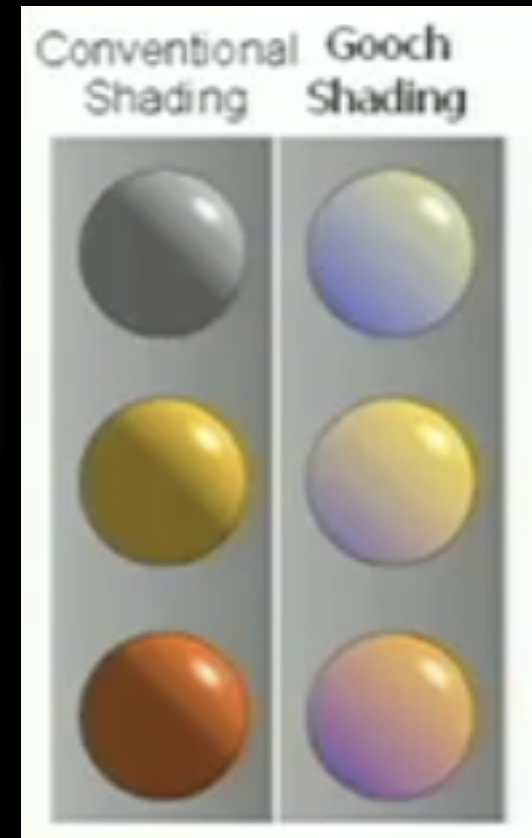
- Can be done in image or object space
- 3 methods:
 - 4 jittered viewport renders + a final
 - Use front and back facing passes to find contours
 - Geometric analysis approach

Gooch Shading

- Technique from 1998
- Hue and luminance indicate surface orientations
- Blend between warm and cool colours based on Lambertian and material properties
- Extreme light / dark reserved for edge lines

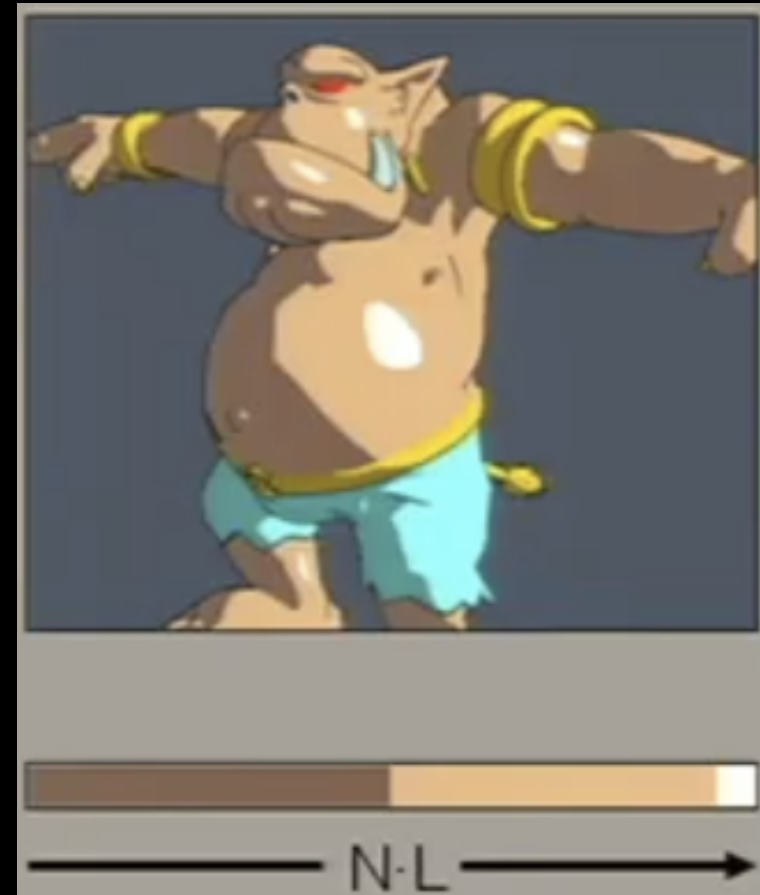
Gooch Shading

$$\left(\frac{1}{2}(n \cdot l) + \frac{1}{2}\right)(k_{blue} + \alpha k_d) + \left(1 - \left(\frac{1}{2}(n \cdot l) + \frac{1}{2}\right)\right)(k_{yellow} + \beta k_d)$$



Cell Shading

- Technique by Lake from 2001
- ID Texture lookup based on lambertian term
- Simulates limited colour palette
- Also includes pseudo specular highlights



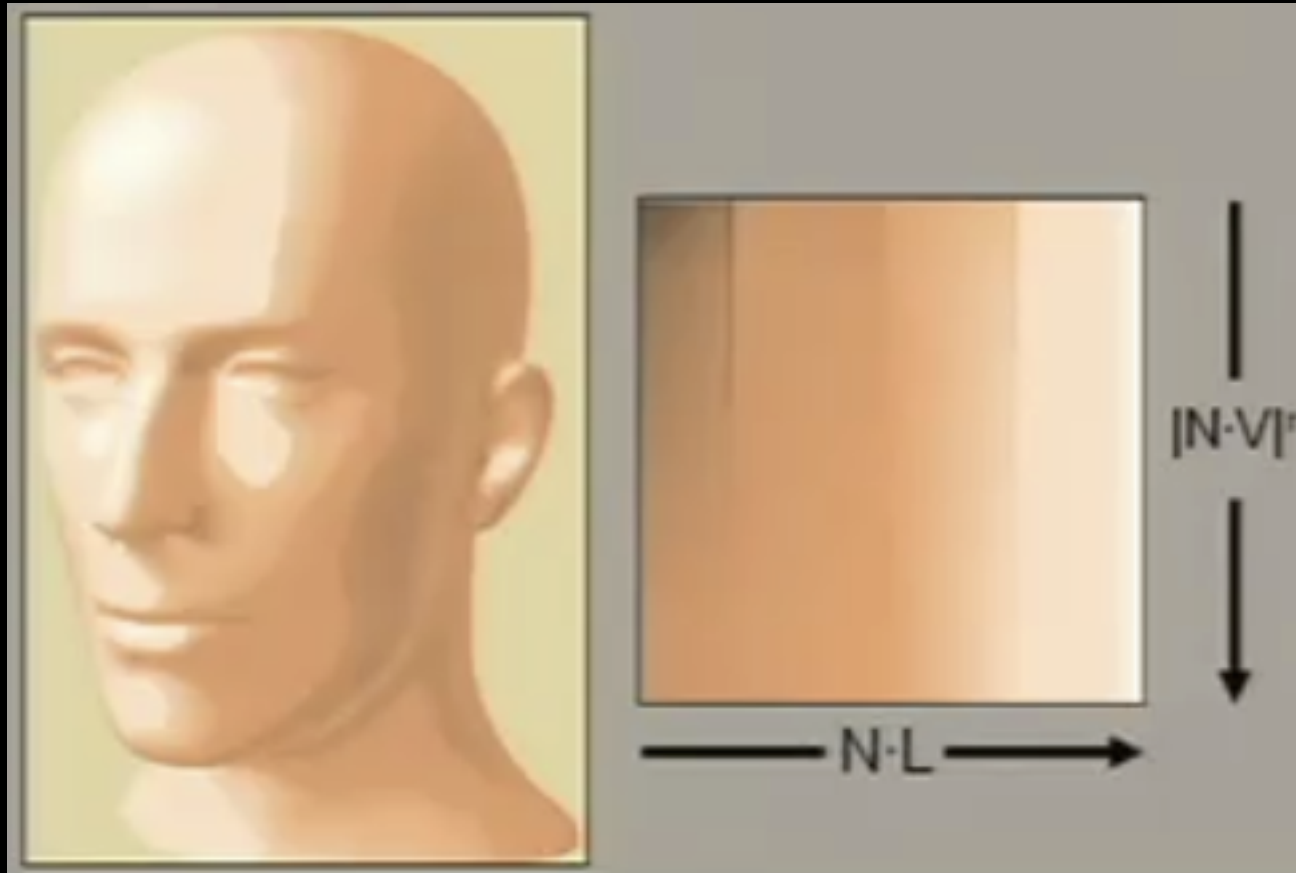
Advanced Cell Shading

- Technique by Barla from 2006
- 2D texture lookup to incorporate view-dependent / level of detail
- Fresnel creates rim highlight / virtual backlight
- Can be tweaked to show variety of feature types

Barla Example I



Barla Example 2



Rim Lighting

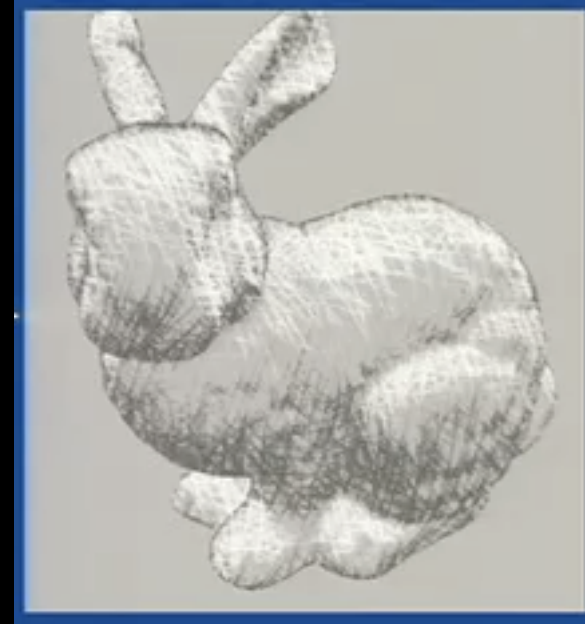


Rim Lighting



Painting Effects

- Crosshatching
 - Short strokes indicate form and tone
- Charcoal
 - Contrast enhanced noise texture
- Pencil Drawing
 - Jittered lines to appear hand-drawn



TF2 Lighting Model

- Valve's new game Team Fortress 2
- Incorporates NPR techniques for toon style gameplay

TF2 Lighting Model Movie Demo

TF2 Lighting Model

- View independent

$$k_d \left[a(n) + \sum_{i=1}^L c_i w((\alpha(n \cdot l) + \beta)^\gamma) \right]$$

- View dependent

$$\sum_{i=1}^L [c_i k_s \max(f_s (v \cdot r)^{k_{spec}}, f_r k_r (v \cdot r)^{k_{rim}})] + (n \cdot u) f_r k_r a(v)$$

TF2 Lighting Model

- View independent

$$k_d \left[a(n) + \sum_{i=1}^L c_i w_i ((\alpha(n \cdot l) + \beta)^\gamma) \right]$$

- Spatially varying directional ambient
- Modified Lambertian
 - Bias, scale and exponent part of material properties
 - Texture lookup function

TF2 Lighting Model

- Ambient Cube:
 - Pre computed irradiance samples create irradiance volume
 - This is used to create an environment map
 - Directional ambient term using normal as lookup

TF2 Lighting Model



TF2 Lighting Model

- View dependent

$$\sum_{i=1}^L [c_i k_s \max(f_s (v \cdot r)^{k_{spec}}, f_r k_r (v \cdot r)^{k_{rim}})] + (n \cdot u) f_r k_r a(v)$$

- Two phong terms per light (material and rim)
- k_{rim} & k_{spec} : Exponents
- f_s & f_r : Fresnel terms
- k_r & k_s : Masking textures

TF2 Lighting Model

- View dependent

$$\sum_{i=1}^L [c_i k_s \max(f_s (v \cdot r)^{k_{spec}}, f_r k_r (v \cdot r)^{k_{rim}})] + (n \cdot u) f_r k_r a(v)$$

- $a(v)$: Directional ambient evaluation
- Uses same mask and Fresnel
- $n \cdot u$ makes rim highlights appear to come from above

Sources

- Astle, D (ed.) (2006) More OpenGL Game Programming. Thomson Boston, MA
- Mitchell et al (2007). Illustrative Rendering in Team Fortress 2. SIGGRAPH 2007 Course Notes