Line drawings via abstracted shading

Yunjin Lee
Lee Markosian
Seungyong Lee
John F. Hughes

presentation by Marc Winners
Overview

• Nonphotorealistic Rendering (NPR) Techniques

  • a relatively new, parallel tradition in computer graphics

  • algorithms designed to reproduce visual abstractions known to artists

    • convey shape, material and light properties with minimal cues

• Features:

  • silhouettes (contours), creases, suggestive contours (DeCarlo et al. 2003; 2004), highlight lines
Overview

- Problem: render a 3D model as a line drawing

  - technique should preserve shape and lighting cues

  - fading/tapering of lines

  - “temporally coherent” for real-time animation

  - drawn features scale according to camera

  - scale lines appropriately with some metric (e.g., depth, line length)
Abstracted shading

- Suggestive contours: do not depend on lighting or material
  - only convey properties of the shape

- Observation: important lines can be extracted from shaded images
  - “essential features” of shaded images become features of drawing

- Essential features of shaded image
  - boundaries between light and dark
  - thin areas of shading and specular highlights

- Advantage: line drawing algorithm varies smoothly with shading algorithm
Strategy

• Two rendering passes

  • Begin with grayscale, shaded, “tone image” incorporating lighting and material qualities from object space

  • Detect ridges and valleys on tone image using GLSL fragment shader

• Specular highlights are special

  • often surrounded by moderately bright diffusely lit areas

  • detect with higher threshold
Ridge Detection

• Goal: find principal curvatures of surface of tone image
  
  • measures how much the surface bends and in what directions

  • for any given point, the principal curvatures denote maximum and minimum amount a surface is bending

  • useful for finding important shading features
Ridge Detection

- Fragment shader on the tone image (image space algorithm)
  - Takes $n$ locally parameterized samples around each pixel $(0, 0)$
    - In practice, 9 samples spaced at half the line width
  - Treats tone values as “height field” of values in $(0, 1)$
  - Fit to polynomial of degree 2 to analytically compute the principal curvature:
    $$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$
  - Solve for coefficients using least squares

- Eigenvalues/vectors of symmetric coefficient matrix $M = (a_0 \ a_1, \ a_1 \ a_2)$ define principal curvatures

- Approximate relative location of ridge or valley given by
  $$c = -\frac{1}{2}M^{-1}(a_3 \ a_4)^T$$
Ridge Searching

• Where does each pixel lie?

• (a) on a ridge or valley
• (b) near a ridge or valley
• (c) on a smooth region
• (d) on an edge
Ridge Searching

- Where does each pixel lie?
  - (a) opacity = 1
  - (b) ?
  - (c) opacity = 0
  - (d) ?

- For (b) and (d), move to detected ridge

- if computed curvature is beneath a threshold, or distance to ridge is farther than half the line length, set opacity = 0, otherwise set opacity = 1
Refining the lines on ridges, valleys and edges

• So far, lines either have opacity of 0 or 1
  • can cause abrupt changes if lighting or camera changes

• Tweak opacity according to
  • an additional threshold---scale opacity with amount of curvature (c)
  • distance to ridge: fade as distance approaches half line width (d)
  • counterchange: brightness of tone image beneath line (opacity * (1 - t)) (e)
Refining the lines on ridges, valleys and edges

- Additional enhancements
  - Scale line width parameter with depth buffer values
Performance results

- NVIDIA GeForce 7900 GT GPU
- 800x600 resolution with 60k-200k triangles in scene
- 7-20 fps

Problem:

- Blurring the tone image is time consuming
Limitations and weaknesses

• Tone lines are set using a per-pixel approach
  • cannot extract lines explicitly
  • difficult to stylize

• For certain lighting conditions and parameters, essential shape cues may not be visible
  • Scene designers must adjust accordingly
Examples

(a) Hebe, David, and Athena
(b) Dama
(c) Horse
(d) Bunny
(e) Hippo
(f) Zoom-in/out of landscape