CMSC 491A/691A
Artistic Rendering

Penny Rheingans
UMBC

Announcements

• Lab meeting: Tues 2pm, ITE 352, starting next week
• Proposal due Thurs
Shape Cues: Outlines

- Outline flat parts
- Outline important boundaries
- Omit outlines if contour is most important

Phyllis Wood, in Wood94, pg 40

Silhouettes and Outlines

- Draw expressive silhouettes and outlines of objects
- Key issues:
  - Identifying silhouettes
  - Drawing stylized silhouettes
Silhouettes and Outlines

- Papers
  - Raskar99
  - Hertzmann00
  - DeCarlo03
  - Kalnins03

Image Precision Silhouette Edges

Ramesh Raskar and Michael Cohen
I3D 99
**Basic Approach**

- Use hardware to draw silhouette edges at image precision
- General method:
  - Identify all front facing visible polygons
  - Identify back facing polygons
  - The intersection of these two is the silhouette

**Fattening Lines**

- Render back in wireframe using ≤
- Translate back faces forward
- Use view-dependent extension of back faces
Results

• Fattening to produce a charcoal-like style
Basic Approach

- Fattening using wirefame, translation, lengthening methods

Illustrating Smooth Surfaces

Aaron Hertzmann and Denis Zorin
Overview

- Surface representation
  - polygonal mesh
  - construct piecewise-smooth subdivision
- Strictly polygonal methods create artifacts
Method

• Rendering algorithm
  – determine hatch direction field (view independent)
  – compute silhouette curves (view dependent)
  – generate hatches (view dependent)

Silhouettes

• Include boundaries, creases, silhouette lines, self-intersection lines
• Silhouette set: points $p$ such that
  $g(p) = (n(p) \cdot (p-c)) = 0$
• Curvature
  – principal curvatures: $\kappa_1$, $\kappa_2$
  – determine coordinate system $(r,s,t)$
• Smooth silhouette differs from pgon mesh
Silhouette Method

- Approximate silhouette set (zero set of g(p))
  - calculate normal and g(p) at vertices
  - approximate g(p) across pgon by linear interpolation
  - zero set is line segments across pgons

Fast Silhouettes

- Optimize using dual surfaces
  - each point mapped to \[ N = [n_1, n_2, n_3, -(p \cdot n)] \]
  - map viewpoint to C
  - silhouette is all points from which C is in the tangent plane at that point: \( (C \cdot N) = (c-p) \cdot n = 0 \)
Fast Silhouettes (2)

- In dual space, intersect plane with surface

Fast Silhouette Algorithm

- For each vertex \( p \) with normal \( n \), compute dual position \( N \)
- Normalize each \( N \) using \( l_\infty \) norm (at least one component becomes 1 or -1; on cube)
- Each tri assigned to list of each face it’s on
- Octree constructed for each face
- Each frame, octree used to find intersection of dual plane with dual surface
Direction Fields

- Observations from art
  - principal curvature shows geometry on cylinder
  - isometric lines work when parameterization exists
  - artists tend to use straight hatches

Cross Fields

- Defined on nonorientable surfaces
- Some natural cross-hatching patterns cannot be decomposed into two smooth fields
Hatch Field Construction

- Create smooth copy of mesh
- Identify areas where curvature ratio is high and at least one curvature can be computed reliably
- Initialize field over surface from principle curvature directions
- Fix field in reliable regions; optimize rest of field

Direction Fields

(a) Just silhouette  (b) raw principal curvature  (c) smooth cross field  (d) hatching smooth field  (e) very smooth field  (f) hatching very smooth field
Hatching Levels

- Four levels of hatching
  - no hatching: highlights and Mach bands
  - single hatching: midtones
  - cross-hatching: shadowed regions
  - dense cross-hatching: undercuts

- Use surface shape to determine level
  - opposite undercut is unhatched Mach band
  - hatch straight
  - hatch thickness proportional to lighting (opt)

Hatch Placement

- Hatching process
  - Identify Mach bands and undercuts
  - Cover single and double regions with cross-hatches; add extra hatches to undercut regions
  - Remove cross-hatches from single regions
  - Hatches clipped to hatch region
Suggestive Contours for Conveying Shape

Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella
SIGGRAPH03
Concept

- Adding contours for nearby viewpoints improves expressiveness

Contour
Suggestive Contour

- Suggestive contours are inflection points where contours will eventually appear
- Generator where radial curvature is 0 and directional derivative is positive

Suggestive Contour Region
**Derivative Test**

Contours add suggestive contours

---

**Object Space Algorithm**

Contours add suggestive contours
Object vs Image Space Algorithm

DeCarlo 03

Object vs Image

DeCarlo 03