CMSC 635

Non-Photorealistic Rendering

based on slides by Jon Cohen

NPR

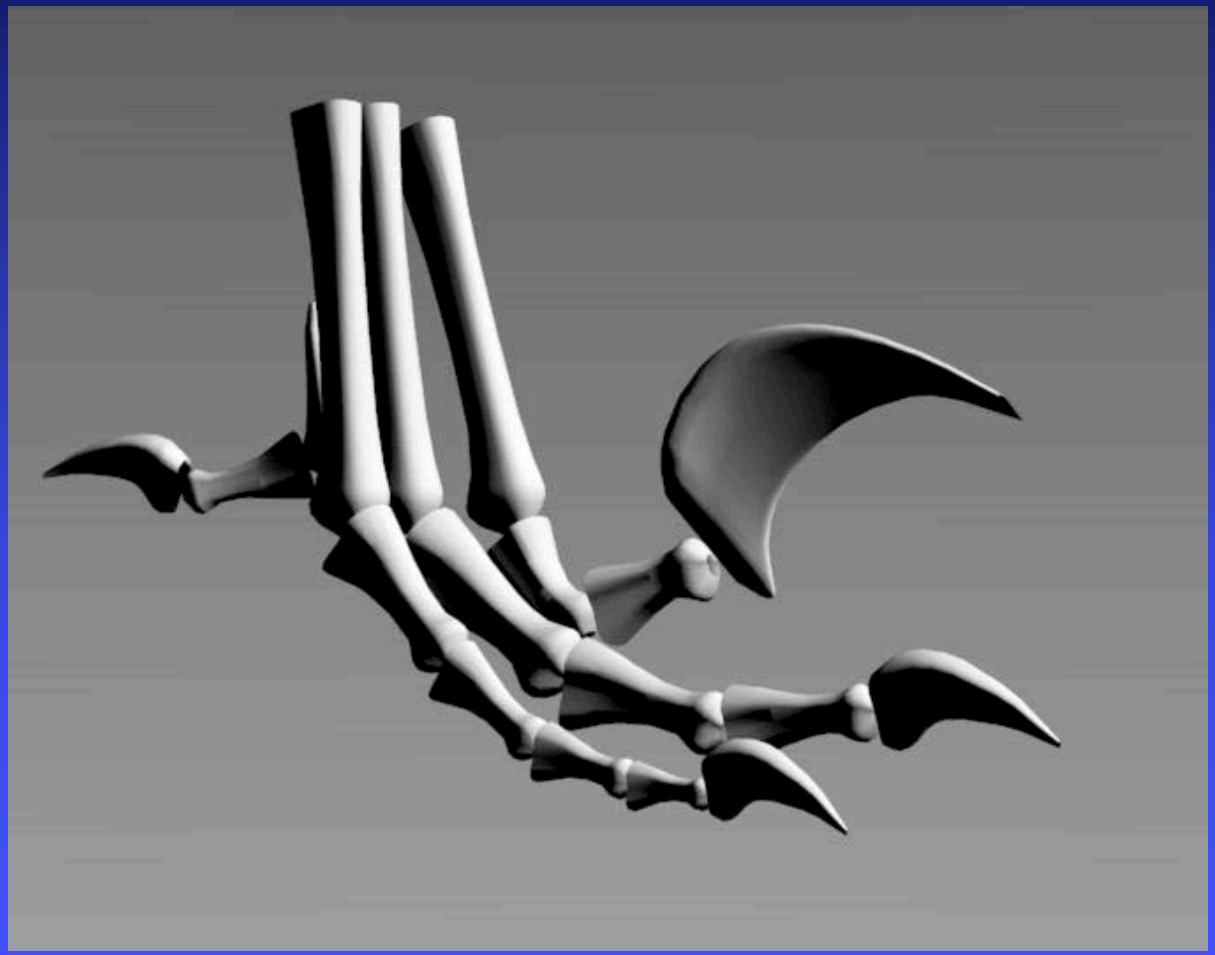
- Photorealistic
  - Look as much like a photo as possible
- Non-photorealistic
  - Not trying to look like a photo
- Technical illustration
- Artistic rendering

Technical Illustration

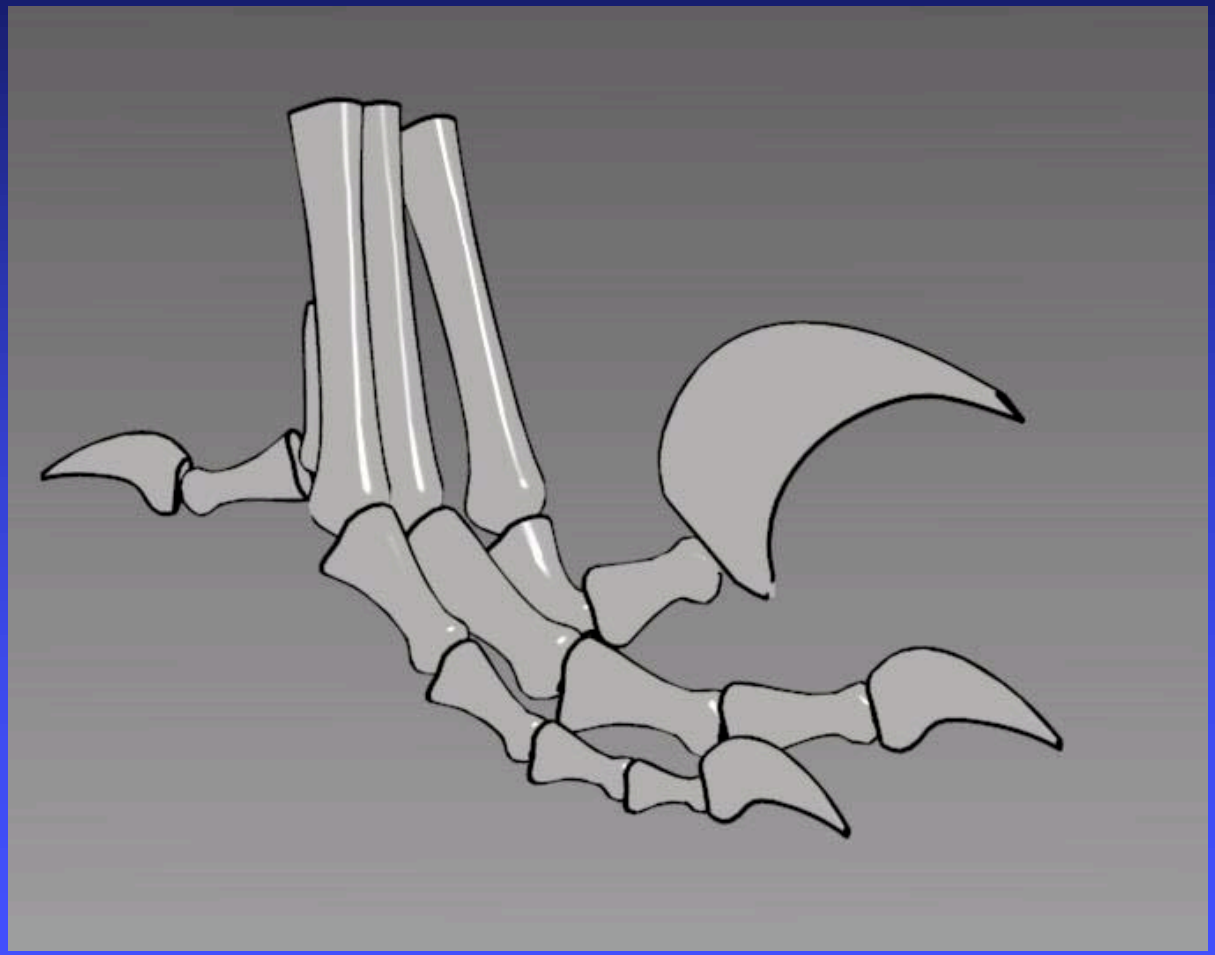
[Gooch, Gooch, Shirley & Cohen, SIGGRAPH 98]

- Edge lines drawn with black curves
  - boundaries, silhouettes, discontinuities
- White highlights from single light source
- Shading stays far from black and white
  - limited intensity range
- Hue changes (warm to cool) help to indicate surface normal

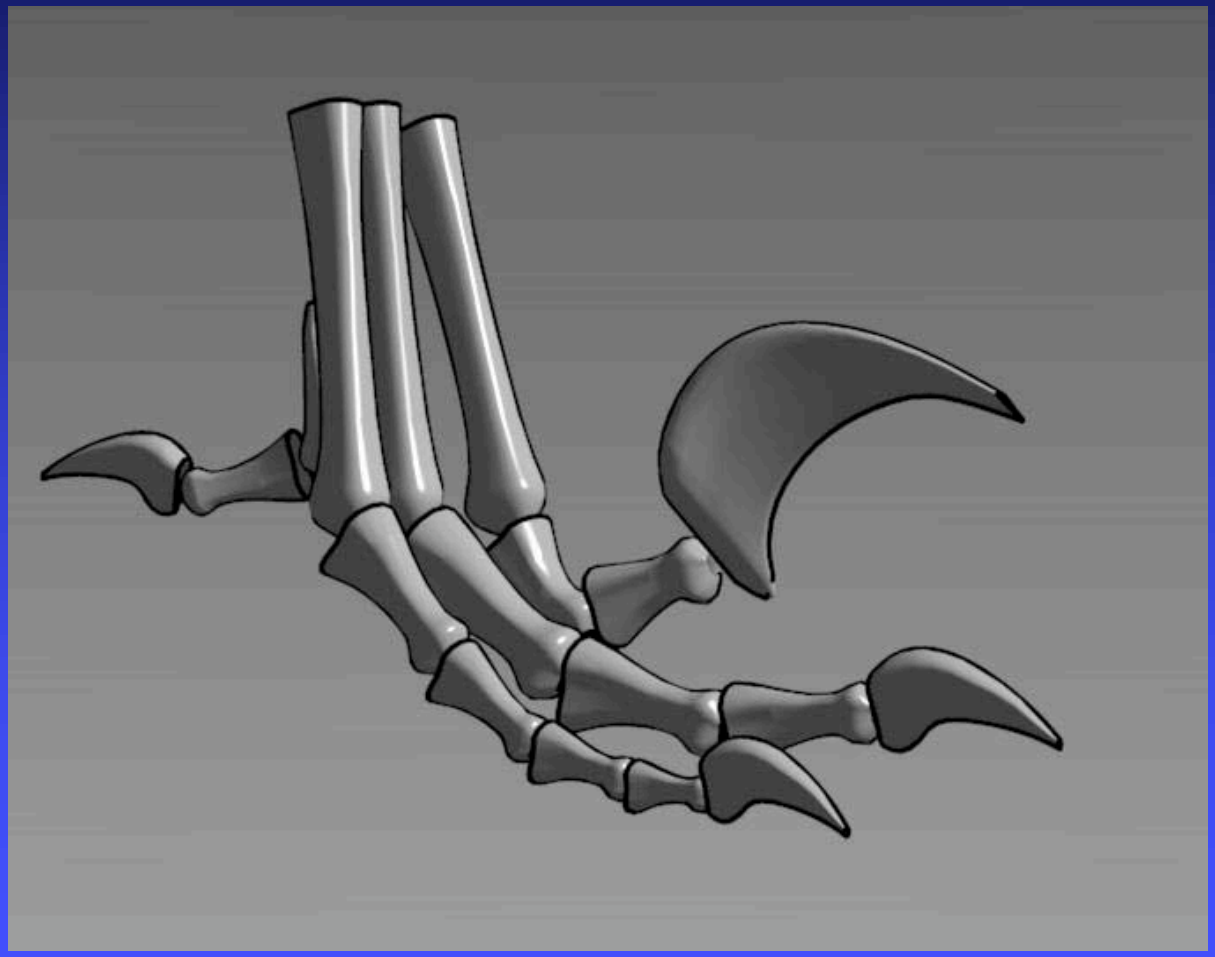
Phong Illumination



Solid + Highlights and Edges



Restricted Intensity Phong



Diffuse Illumination

- Standard Lambertian Model

  $I = C_d * k_a + C_d * \max(0 , \mathbf{N} \bullet \mathbf{L})$

  - $\mathbf{N} \bullet \mathbf{L} < 0$ = constant color

- Color Interpolation Model

  $I = (1 + \mathbf{N} \bullet \mathbf{L})/2 * C_1 + (1 - \mathbf{N} \bullet \mathbf{L})/2 * C_2$

  - Variation across entire range of normals
    - $\mathbf{N} \bullet \mathbf{L} \in [-1,1]$

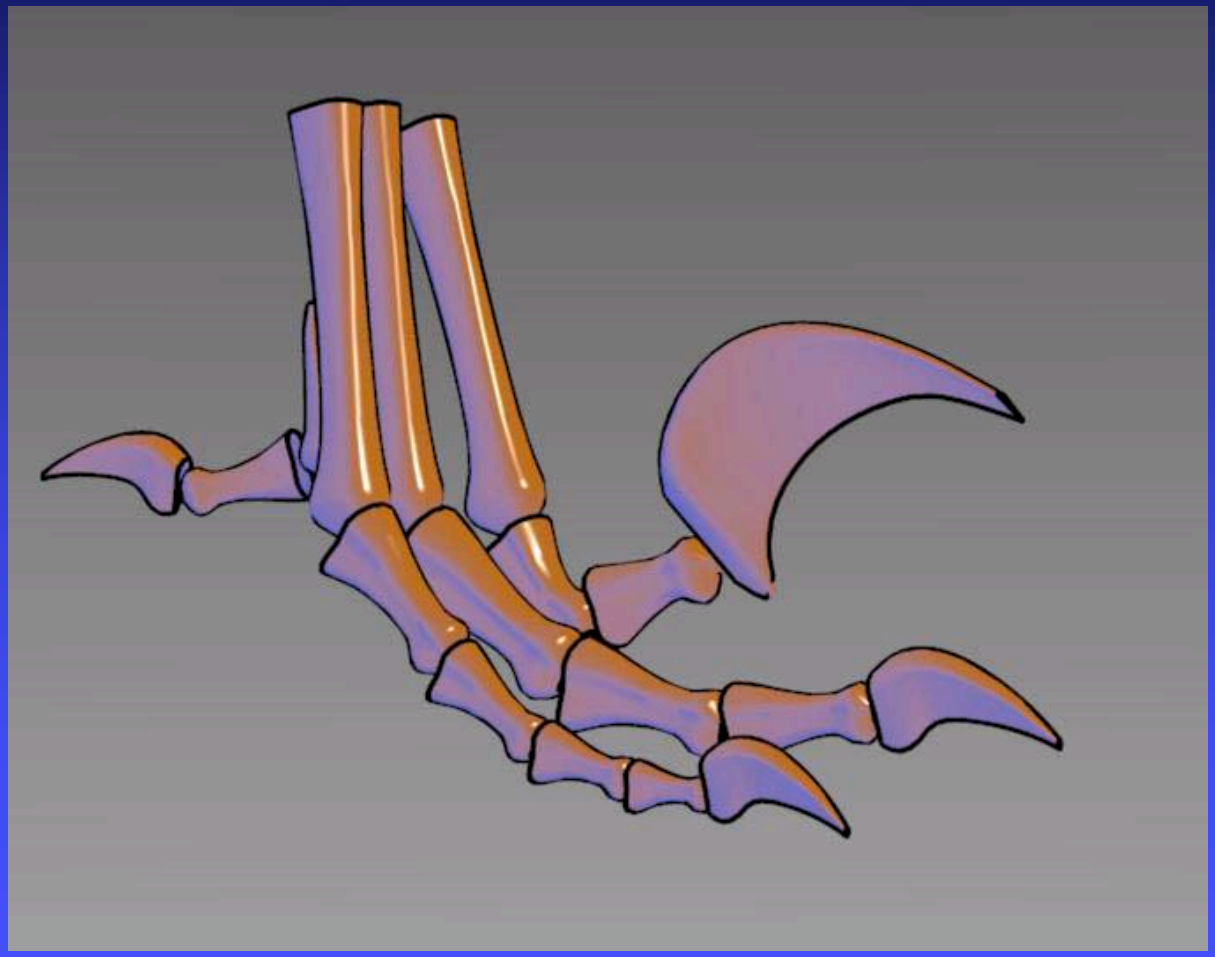
Color Temperature Principles

- Warm colors approach
  - red, yellow, orange
- Cool colors recede
  - Blue, violet, green
- Sun & incandescent lights warm
  - Shadows appear cool (complementary color)

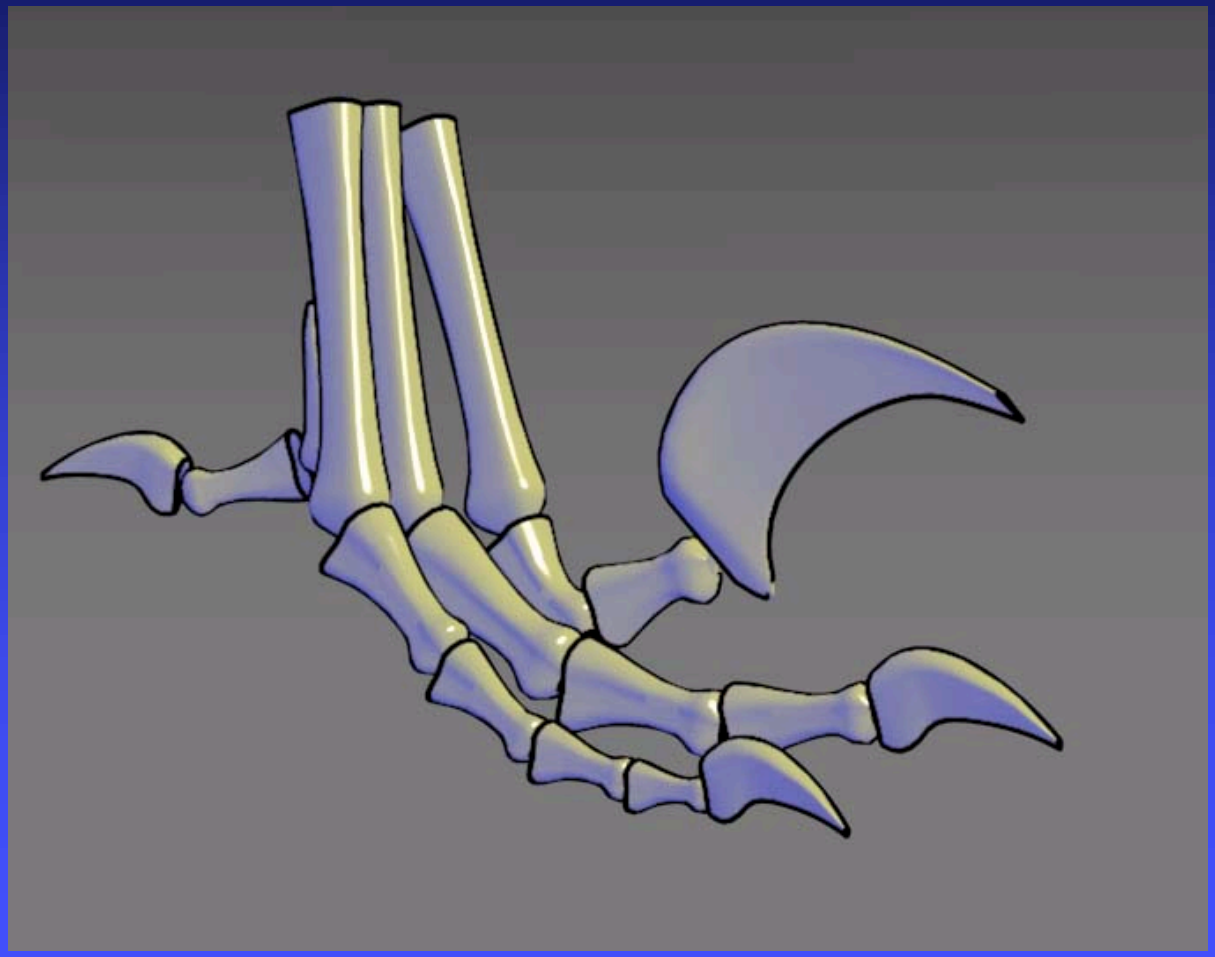
Cool-to-Warm Illumination

- Blue-to-yellow illumination
  - C_1 = blue = (0,0,b)
  - C_2 = yellow = (y,y,0)
- Scaled object-color illumination
  - C_1 = black = (0,0,0)
  - C_2 = object color = C_d
- Combined model
  - $C_1 = C_{cool} = (0,0,b) + C_d$
  - $C_2 = C_{warm} = (y,y,0) + C_d$

Constant Lum., Changing Hue



Changing Hue and Luminance



OpenGL Cool-to-Warm

- Two directional lights
  - Direction L, intensity $(C_{warm} - C_{cool})/2$
  - Direction -L, intensity $(C_{cool} - C_{warm})/2$
    - Negative intensities are legal!
- Ambient light
  - Intensity $(C_{cool} + C_{warm})/2$
- White surface color
- Add white highlights using second pass

OpenGL Cool-to-Warm

- Fixed lights
- Use cool-to-warm environment map

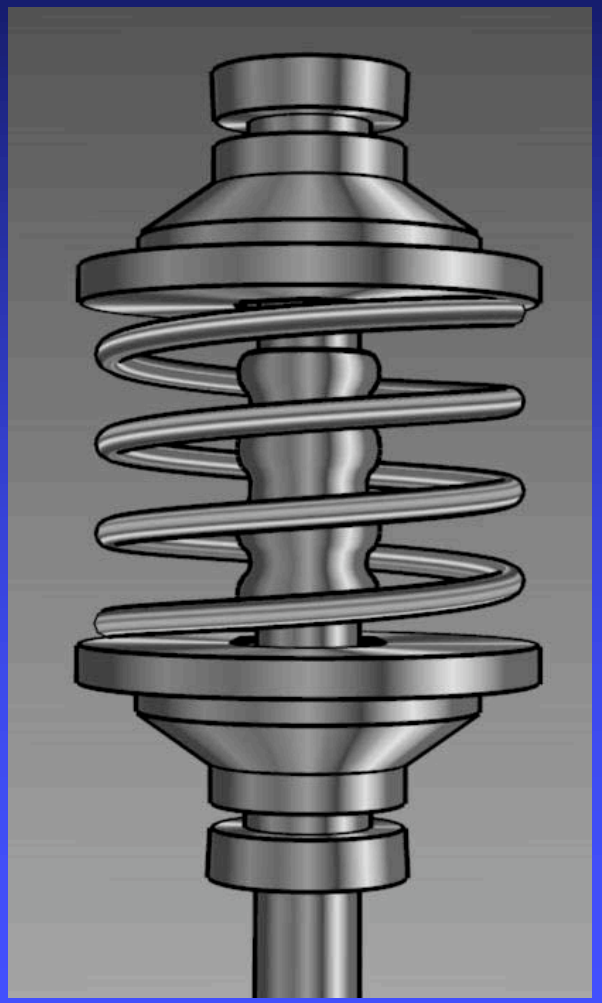
Illustrative Metal Shading

- Milled metals exhibit streaks along milling axis
- Simulate this anisotropy using stripes of various intensities along milling axis
  - Random stripe intensities from 0.0 to 0.5
  - Stripe closest to light direction is white
  - Linearly interpolate colors between stripes

Metal Shading + Edges

Phong

Metal



Pen and Ink

- Strokes
  - Curved lines of varying thickness and density of placement
- Texture
  - Character conveyed by collection of strokes, e.g. crisp and clean vs. rough and sketchy
- Tone
  - Perceived gray level across the image
- Edges
  - Lines to disambiguate structure

Algorithm Goal

- Place strokes on surfaces to achieve particular tone functions

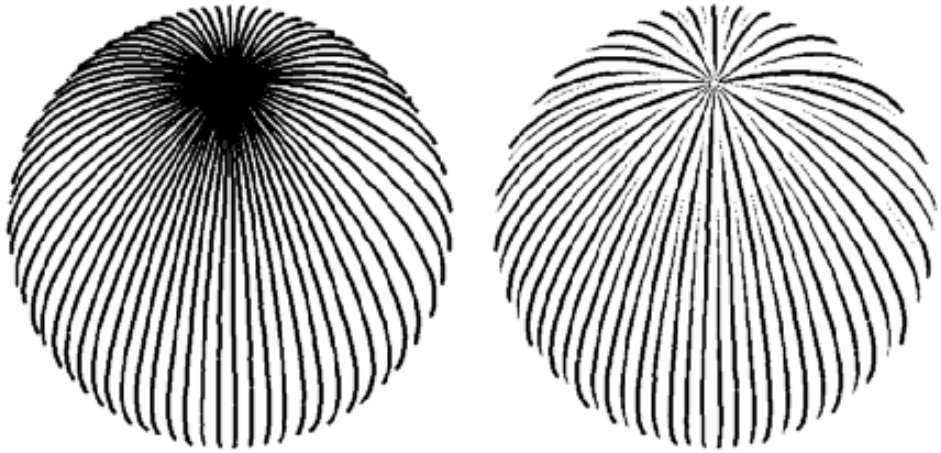


Figure 2 Controlled-density hatching for a perspective view of a sphere. Again, rendering isoparametric curves with constant thickness results in an image with varying tones (left). Using varying stroke thicknesses keeps the "apparent tone" constant (right).

from Winkenbach and Salesin. "Rendering Parametric Surfaces in Pen and Ink." *Proceedings of SIGGRAPH 96*. Page 471.

Algorithm Components

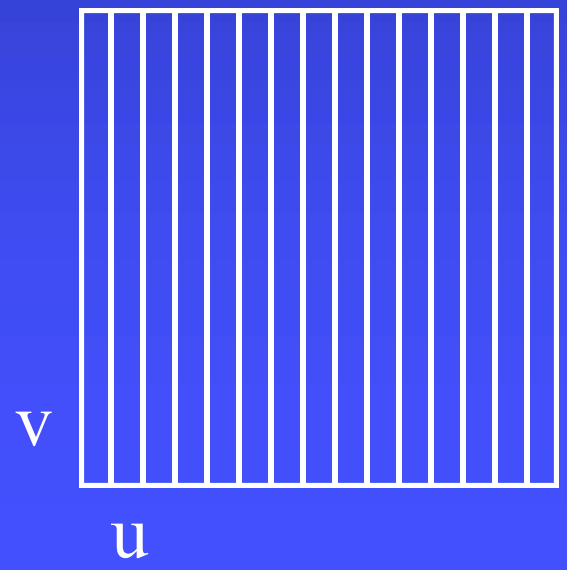
- Tone specification
- Stroke placement
- Stroke width computation

Tone Specification

- Gray levels may be assigned according to conventional rendering:
  - Local/global Illumination
  - Material color
  - Texture mapping
  - Bump mapping
  - Environment mapping
  - Shadow mapping

Stroke Placement

- Places strokes along isoparameter lines of parameterized surface
- Choose density according to maximum gray level and maximum allowable stroke width



Planar Maps

- Compute visibility and store in planar map
  - Planar map is partition of image plane
  - Each partition corresponds to a visible portion of a primitive.
  - Shadows may be explicitly represented as map partitions
- Clip strokes according to planar map
  - Reduces computation and allows rendering with hidden surfaces already removed
- Create outlines from partition boundaries

Planar Map Example

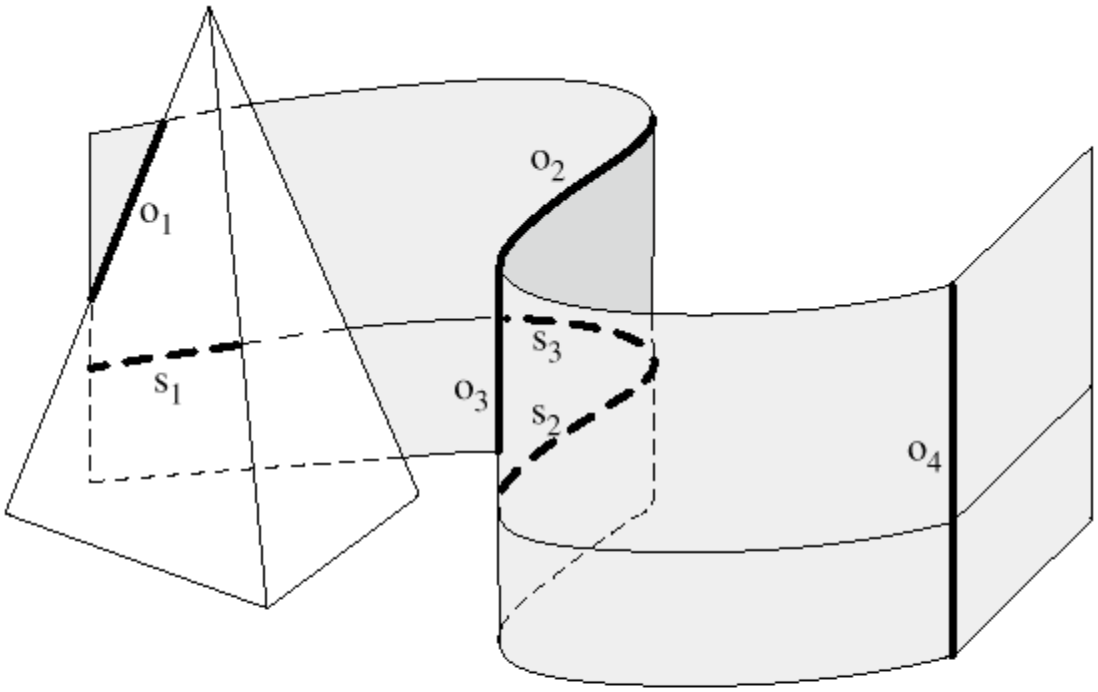


Figure 3 Several cases must be considered when tracing outlines (edges labeled o_1 to o_4), and clipping strokes (edges labeled s_1 to s_3).

from Winkenbach and Salesin. "Rendering Parametric Surfaces in Pen and Ink." *Proceedings of SIGGRAPH 96*. Page 474.

Stroke Width

- Vary width across each stroke line
- S: (u,v) □ (xw,yw,zw)
- V: (xw,yw,zw) □ (xs,ys)
- M = V S : (u,v) □ (xs,ys)
- Use Jacobian of M to estimate divergence of lines in screen space
- Adjust width to account for divergence and desired tone along each stroke

Advanced Techniques

- Recursive filler strokes
  - Allow larger gaps between strokes, then fill gaps by adding new strokes
- Stippling
  - draw stipple pattern along strokes
- Cross hatching
  - use more than one hatching direction
- Prioritized strokes
  - stroke thicknesses determined in prioritized order

Pen and Ink Example

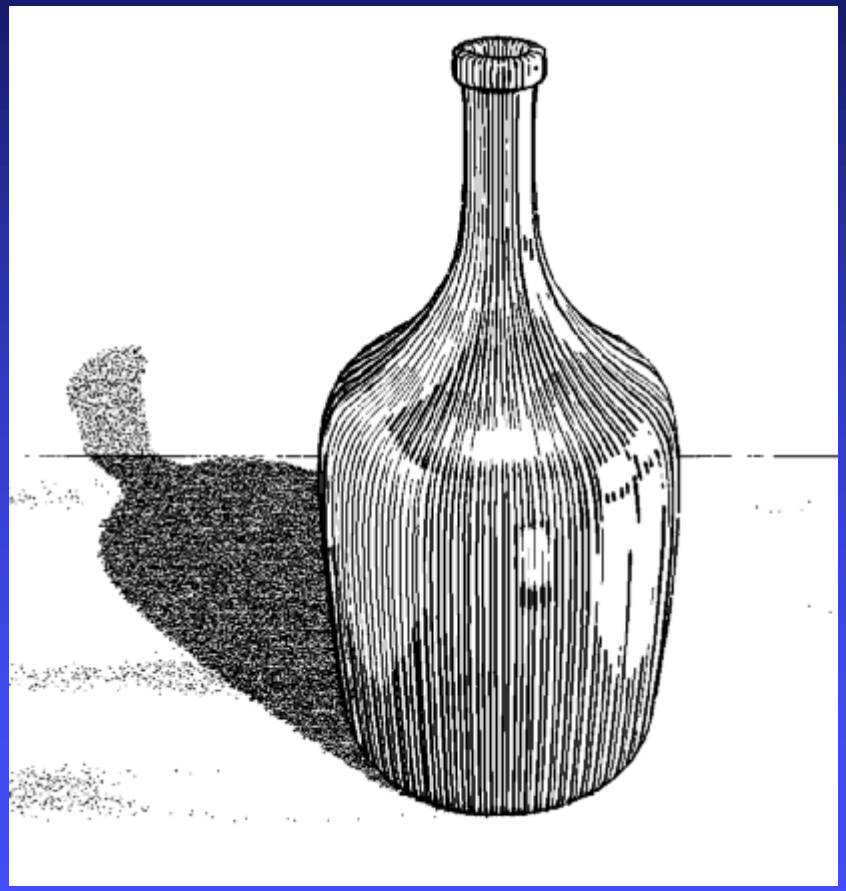


Figure 5 Glass bottle. An environment map is used to give the illusion of a reflected surrounding.

from Winkenbach and Salesin. "Rendering Parametric Surfaces in Pen and Ink." *Proceedings of SIGGRAPH 96*. Page 474.

Pen and Ink Example

from Winkenbach and Salesin. “Rendering Parametric Surfaces in Pen and Ink.” *Proceedings of SIGGRAPH 96*. Page 475.

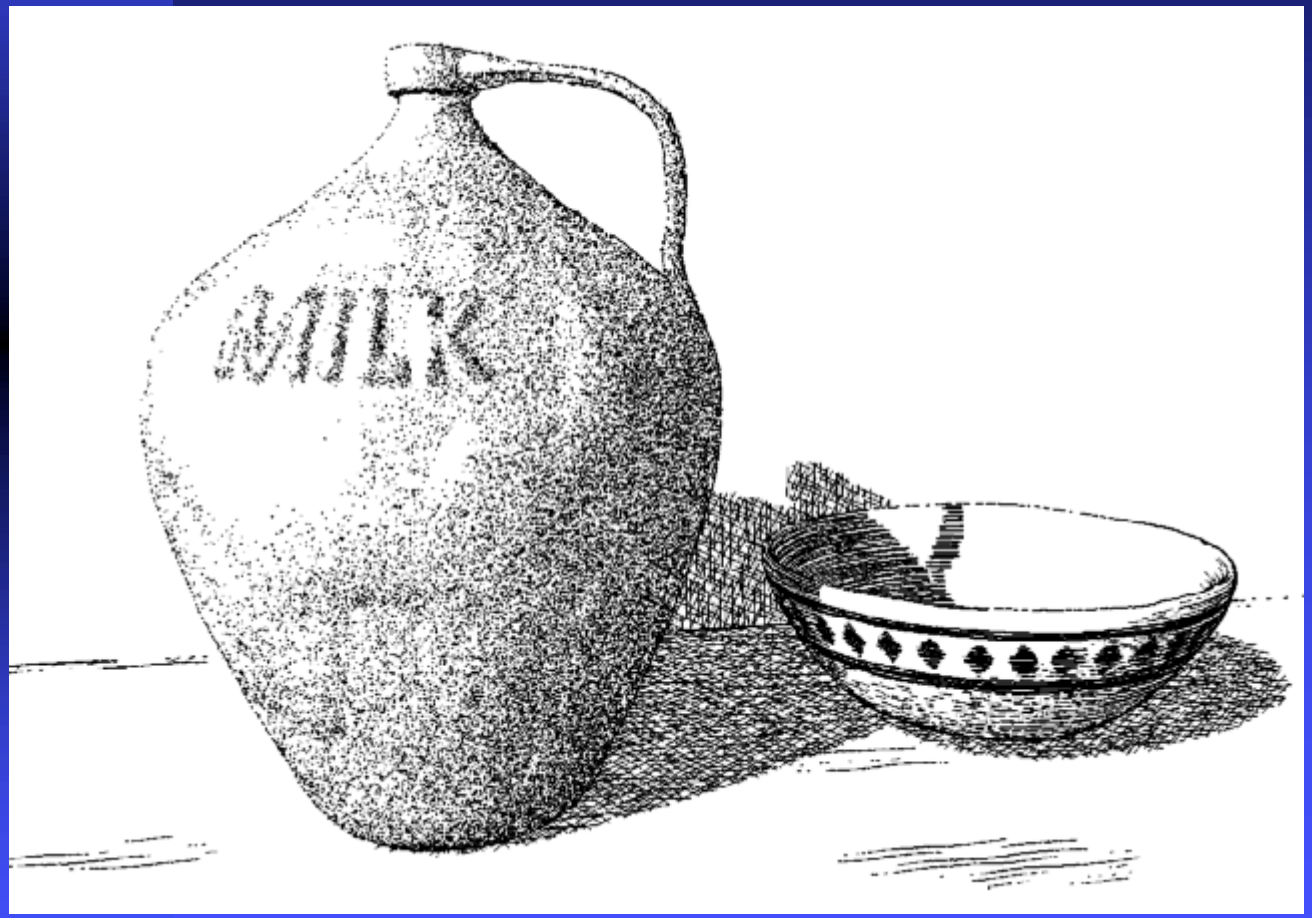


Figure 7 Ceramic jug and bowl. A traditional (image-based) texture map is used to model the details on the bowl as well as the stains on the table. A bump map is used to emboss the word “MILK” on the jug, and to give some irregular variation to its surface.

Pen and Ink Example

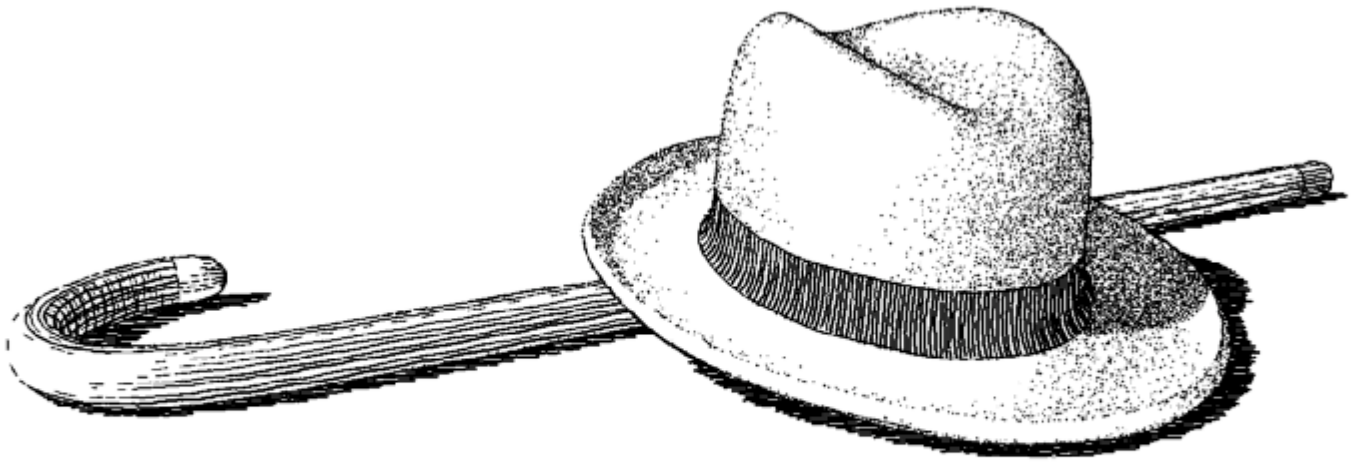


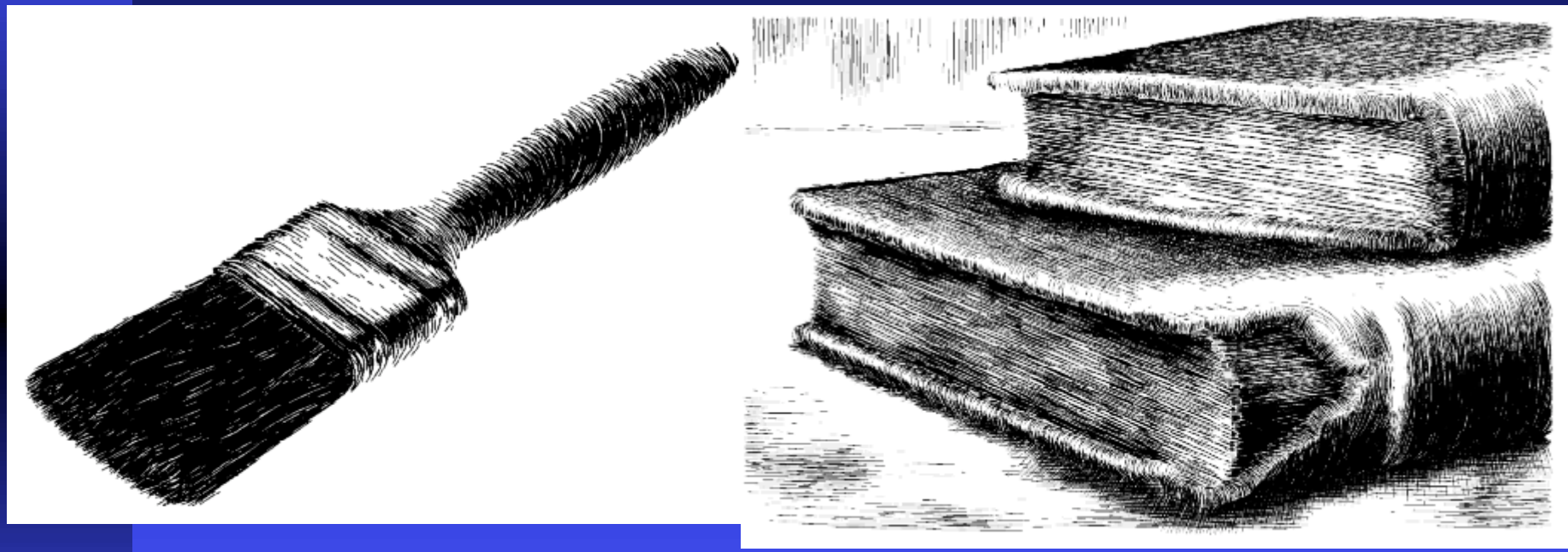
Figure 8 Hat and cane. Both the hat and the cane are modeled with B-spline surfaces. The ribbon is modeled as a separate B-spline surface. Note the curved shadow that the hat projects on its rim, and the use of crosshatching on the curved portion of the cane.

from Winkenbach and Salesin. "Rendering Parametric Surfaces in Pen and Ink." *Proceedings of SIGGRAPH 96*. Page 476.

Other Variants of Pen and Ink

- Orientable Textures
  - Greyscale image as input (describes tone)
  - User specifies direction field and stroke character
  - Stroke shaded image output
- Real-time NPR
  - Fast visibility computation of silhouette and other feature edges
  - Render visible edges in modified styles

Orientable Textures Examples



from Salissbury et al. “Orientable Textures for Image-Based Pen-and-Ink Illustration.” *Proceedings of SIGGRAPH 97*. Pages 402, 403.

Real-Time NPR Examples

from Markosian et al. “Real-Time Nonphotorealistic Rendering.” *Proceedings of SIGGRAPH 97*. Page 420.

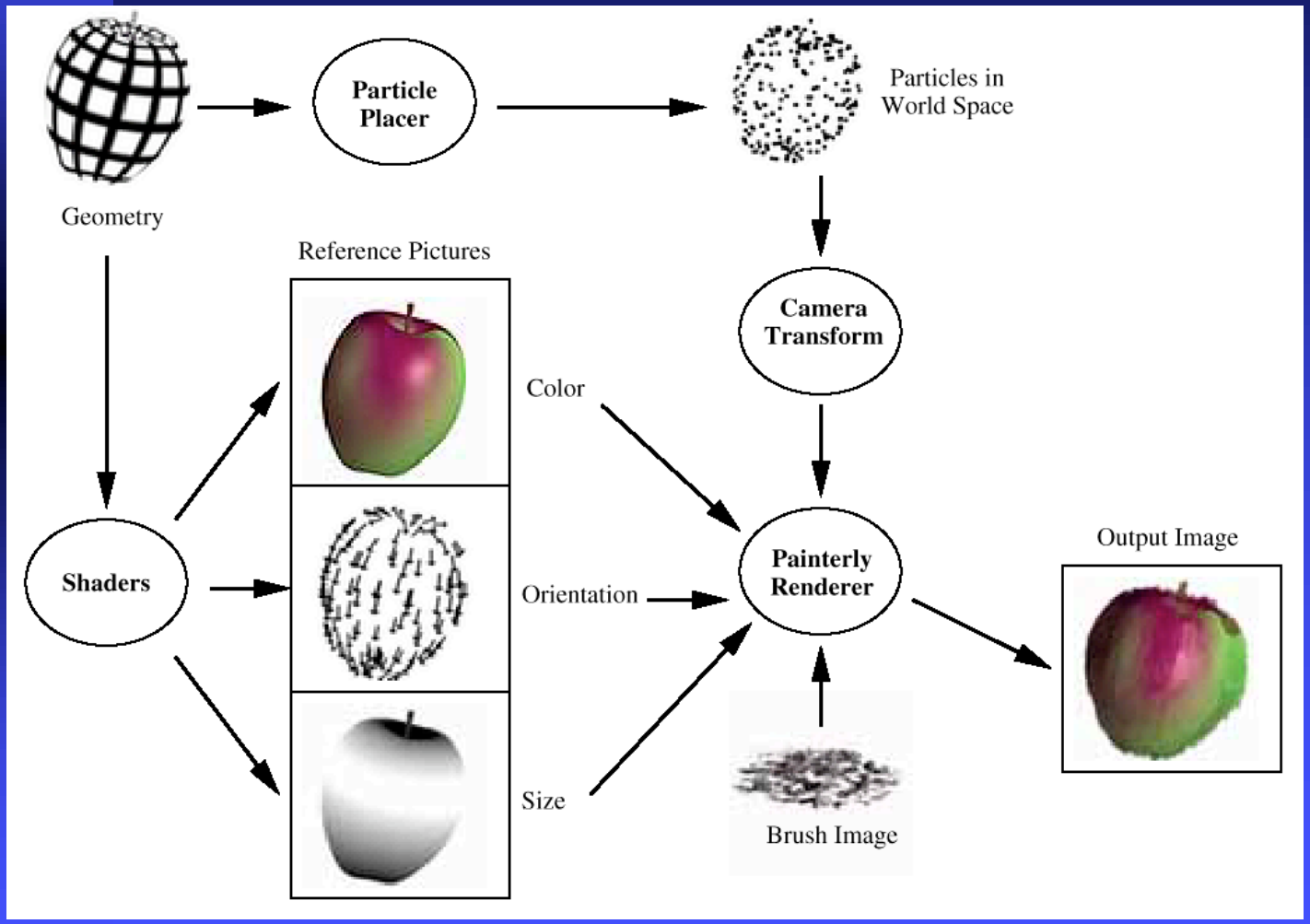
Painterly Rendering

- Physical simulation
  - User applies strokes
  - Computer simulates media (e.g. watercolor on paper)
- Automatic painting
  - User provides input image or 3D model and painting parameters
  - Computer generates all strokes

Painterly Rendering Systems

- "Painterly Rendering for Animation"
  - Meier, SIGGRAPH 96
- "Painterly Rendering with Curved Brush Strokes of Multiple Sizes"
  - Hertzmann, SIGGRAPH 98

Painterly Rendering Pipeline



from Meier, "Painterly Rendering for Animation, *Proceedings of SIGGRAPH 96*, page 480.

Basic Approach

- Algorithm
  - Surface particles placed in world space
  - Reference images rendered
  - Each particle becomes a screen-space stroke
- Features
  - Greater temporal coherence than purely screen-space approaches
  - More natural style than purely geometry (texture-mapped) approaches

Particle Generation

- Compute area of surface primitives
- Randomly place particles on primitives
  - number proportional to area

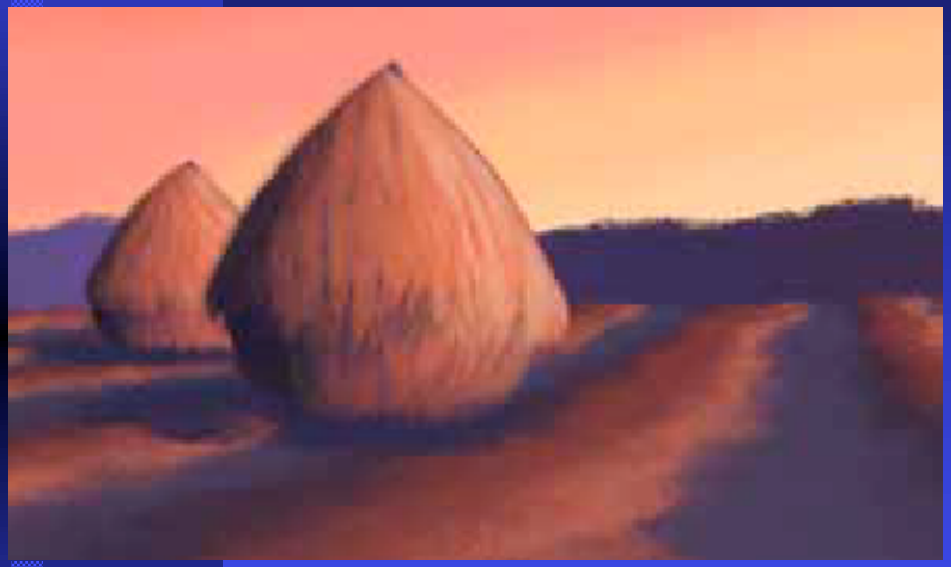
Reference Images

- Used to determine stroke attributes
  - color
  - orientation
  - size
  - many others possible
- Rendered with programmable shaders

Stroke Rendering

- Particle transformed to screen-space
- Stroke parameters from reference images
  - perturbed according to user-specified variation
- Brush image rendered according to stroke parameters
  - oblong brush shapes work best
  - grayscale brushes typically sufficient
    - color brush textures may be used to modify particle colors

Example - Haystacks



Haystacks *without* random parameter perturbation

Similar view *with* random parameter perturbation

from Meier, “Painterly Rendering for Animation, *Proceedings of SIGGRAPH 96*, pages 481 and 478.

Example - fruit

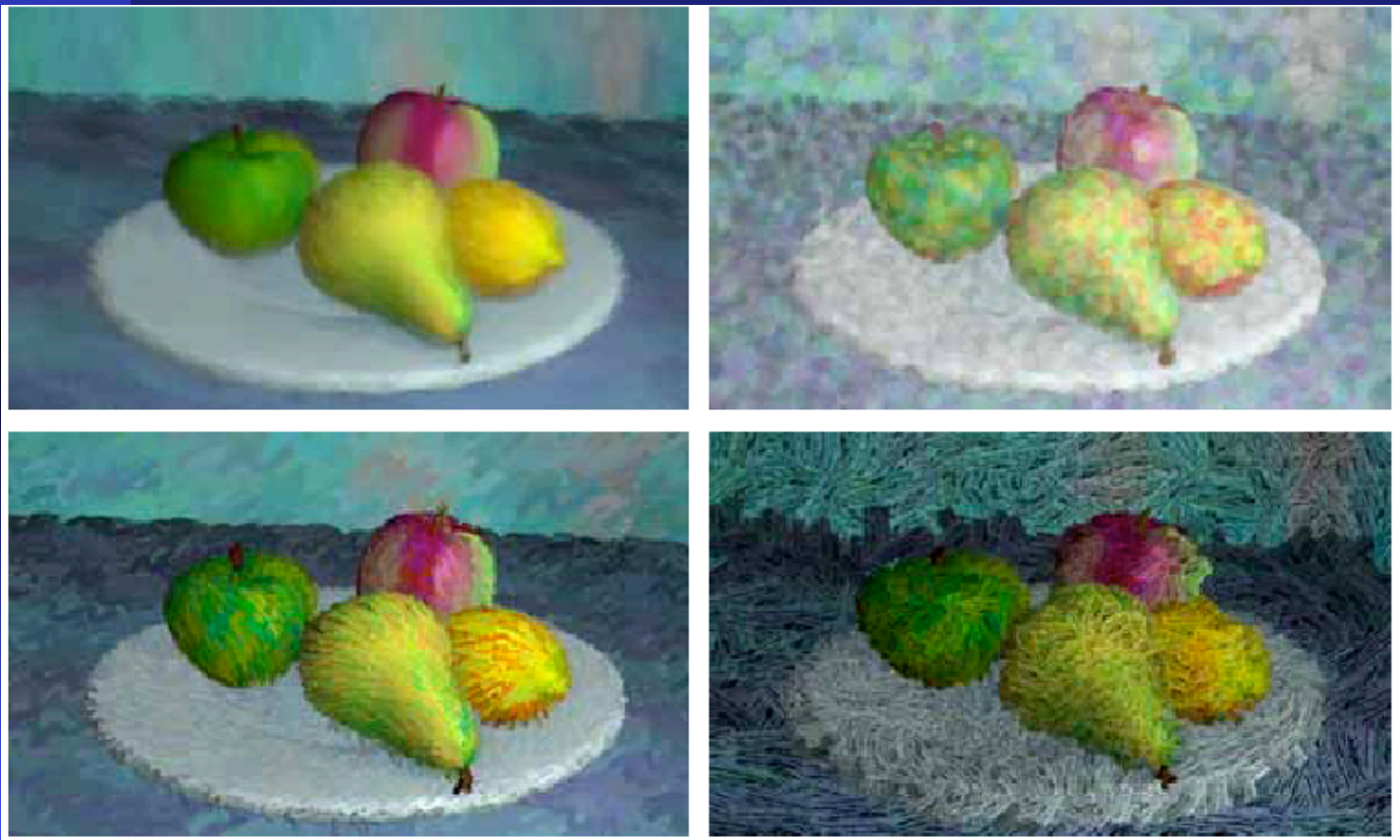


Figure 5: **Four styles of painterly rendered fruit.** By choosing different brush images and painting parameters, we have created four different looks from the same set of reference pictures. The upper left image has the soft, blended quality of a pastel painting. The pointillistic version, in the upper right, remaps the original saturations and values from the color reference picture to a new range. A squiggle brush image and increased hue variation were used to create marker-style strokes in the lower left image. The brush used to create the lower right contained some opaque black that helps to create a woodcut print style.

from Meier, "Painterly Rendering for Animation, *Proceedings of SIGGRAPH 96*, page 481.

Layered Approach

- Similar objects rendered together
- Dissimilar objects often rendered as separate layers and composited later
  - Large strokes intrude less onto nearby objects

Hertzmann's Approach

- Apply to color images with no 3D model information
- Allow longer, curved brush strokes
  - makes different styles possible
- Multiple rendering passes
  - larger strokes first
  - add detail with smaller strokes

Stroke Description

- Constant color per stroke
- B-spline path
- Constant radius circle (or other shape) swept along path
- Applied in layers, with opacity control

Building Up Layers

- Start with large strokes
- Each pass reduces stroke size
- New strokes placed according to error metric of current painting

Painting a Layer

- Select stroke size for layer
- Blur input image
- Start strokes within uniform grid cells
- Start each stroke at point of maximum error within grid cell
- Walk perpendicular to image gradient to place control points
- Render strokes in random order as circles along cubic B-spline path

Style Parameters

- Approximation threshold
- Brush sizes
- Curvature filter
- Blur Factor
- Min/Max stroke lengths
- Opacity
- Grid size
- Color jitter

Example Styles

- “Impressionist”
- “Expressionist”
  - long strokes, color value jitter
- “Colorist Wash”
  - transparency, RGB color jitter
- “Pointillist”
  - densely placed circles, random hue and saturation

Example - adding passes

Figure 2: Painting with three brushes. *(a) A source image. (b) The first layer of a painting, after painting with a circular brush of radius 8. (c) The image after painting with a brush of radius 4. (d) The final image, after painting with a brush of size 2. Note that brush strokes from earlier layers are still visible in the painting.*

from Herzmann, "Painterly Rendering with Curved Brush Strokes of Multiple Sizes, *Proceedings of SIGGRAPH 98*, page 456.

Example - styles

Three styles: impressionist, expressionist, colorist wash

from Herzmann, “Painterly Rendering with Curved Brush Strokes of Multiple Sizes, *Proceedings of SIGGRAPH 98*, page 460.