Non-Photorealistic Rendering

Wojciech Jarosz

June 5, 2008

CSE 168

Computer Graphics

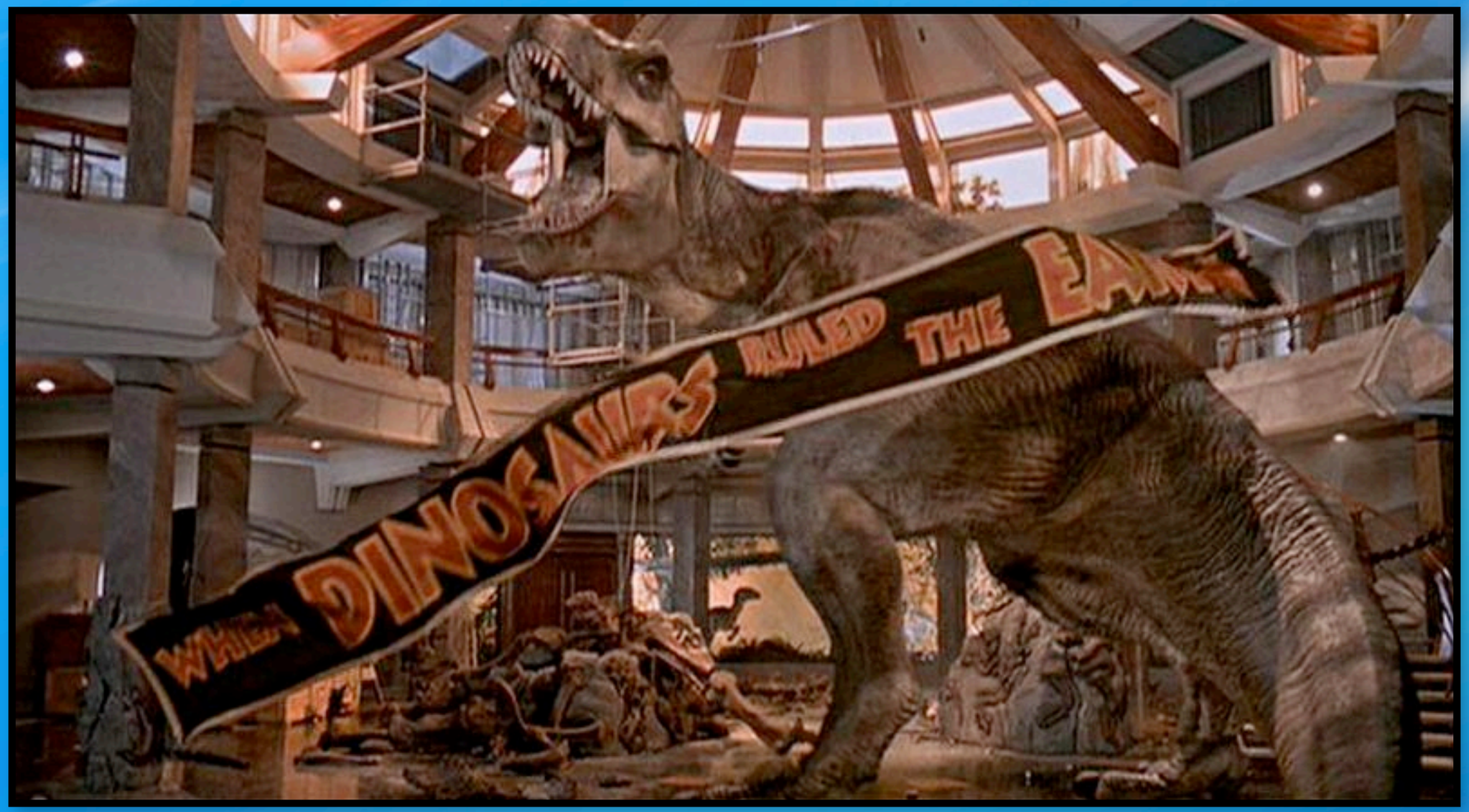
Computer Graphics

- What impresses people about computer generated imagery?

Computer Graphics

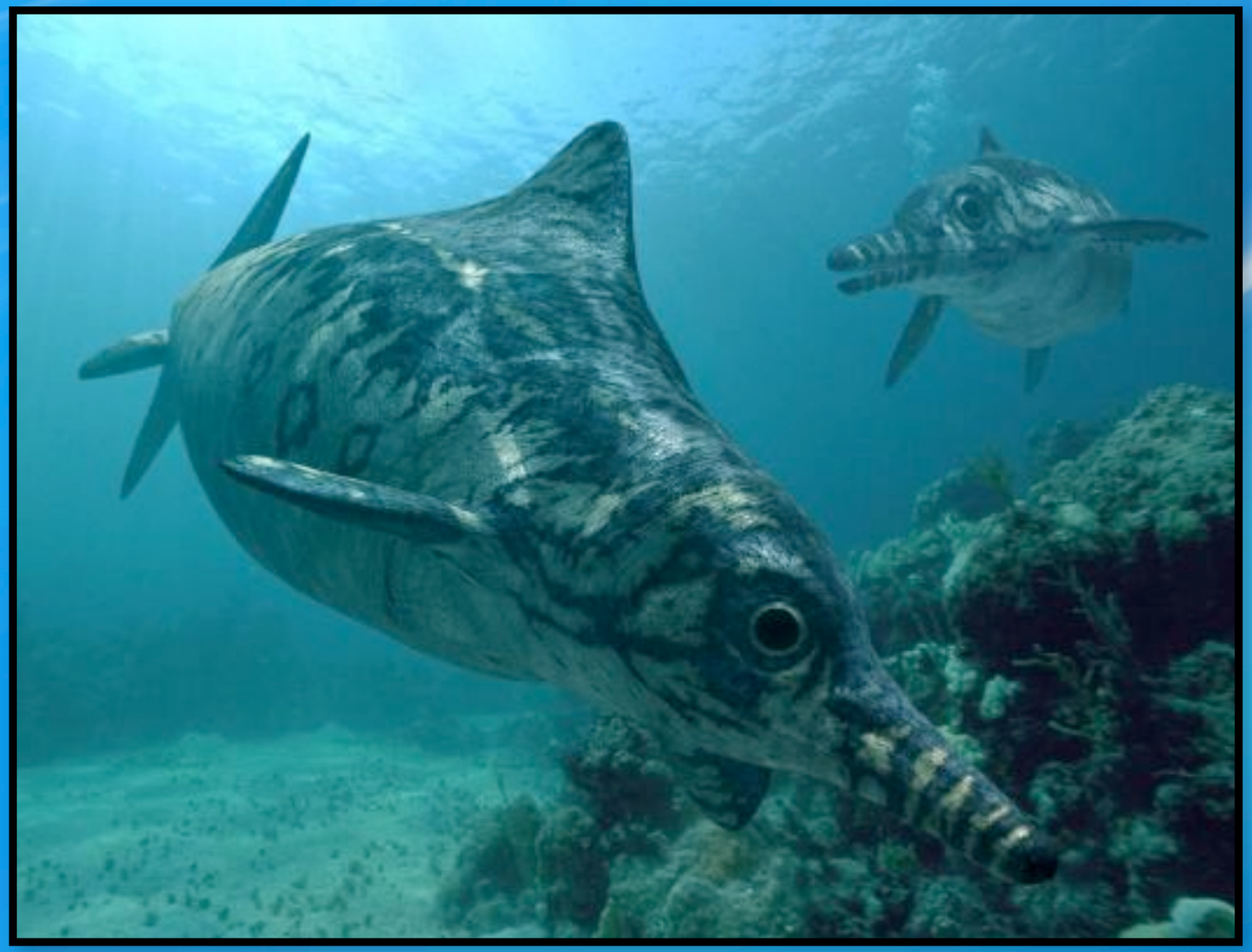
- What impresses people about computer generated imagery?
  - One possible answer: photorealistic images - indistinguishable from photographs.

Jurassic Park, 1993



Jurassic Park, 1993

Final Fantasy: The Spirits Within, 2001



Walking with Dinosaurs, 2001

Walking with Dinosaurs, 2001


Ulf Lundgren 2001

Interior Apt - Chang Kyung Seok.

City of Lost Children, 1995

City of Lost Children, 1995

City of Lost Children, 1995

Computer Graphics

- What impresses people about computer generated imagery?
  - Photorealistic images - indistinguishable from photographs.
  - For the last 35 years, photorealistic and physically-based rendering has been the driving forces behind computer graphics.

Computer Graphics

Computer Graphics

- What are the ultimate goals in computer graphics?

Computer Graphics

- What are the ultimate goals in computer graphics?
  - Photorealism:

Computer Graphics

- What are the ultimate goals in computer graphics?
  - Photorealism:
    - Synthetic images made to mimic photographs of real objects.

Computer Graphics

- What are the ultimate goals in computer graphics?
  - Photorealism:
    - Synthetic images made to mimic photographs of real objects.
    - Image synthesis intentionally includes distracting artifacts of the photographic process (e.g. depth of field, lens flares, etc.

Computer Graphics

- What are the ultimate goals in computer graphics?
  - Photorealism:
    - Synthetic images made to mimic photographs of real objects.
    - Image synthesis intentionally includes distracting artifacts of the photographic process (e.g. depth of field, lens flares, etc.
    - Image quality judged by how well a rendering resembles a photograph.

Computer Graphics

Computer Graphics

- What is the ultimate goal in computer graphics?

Computer Graphics

- What is the ultimate goal in computer graphics?
  - Communication:

Computer Graphics

- What is the ultimate goal in computer graphics?
  - Communication:
    - "An image is worth a thousand words."

Computer Graphics

- What is the ultimate goal in computer graphics?
  - Communication:
    - "An image is worth a thousand words."
    - Graphics provide a high-bandwidth medium for transmitting information.

Computer Graphics

- What is the ultimate goal in computer graphics?
  - Communication:
    - "An image is worth a thousand words."
    - Graphics provide a high-bandwidth medium for transmitting information.
    - Image quality judged by how well a rendering communicates information.

Graphics as Communication

Graphics as Communication

- Why not just use photographs, or photorealistic images for everything, since that would communicate reality?
- Not always interested in all aspects of reality.
- Depending on context, different aspects of the scene may be more or less important.

Graphics as Communication

Graphics as Communication

- We can choose to present the same information in many different forms.
- This decision influences what information is emphasized and what information is obscured.
- Effective communication involves including essential information, but also omitting irrelevant information.

Sailboat Photograph

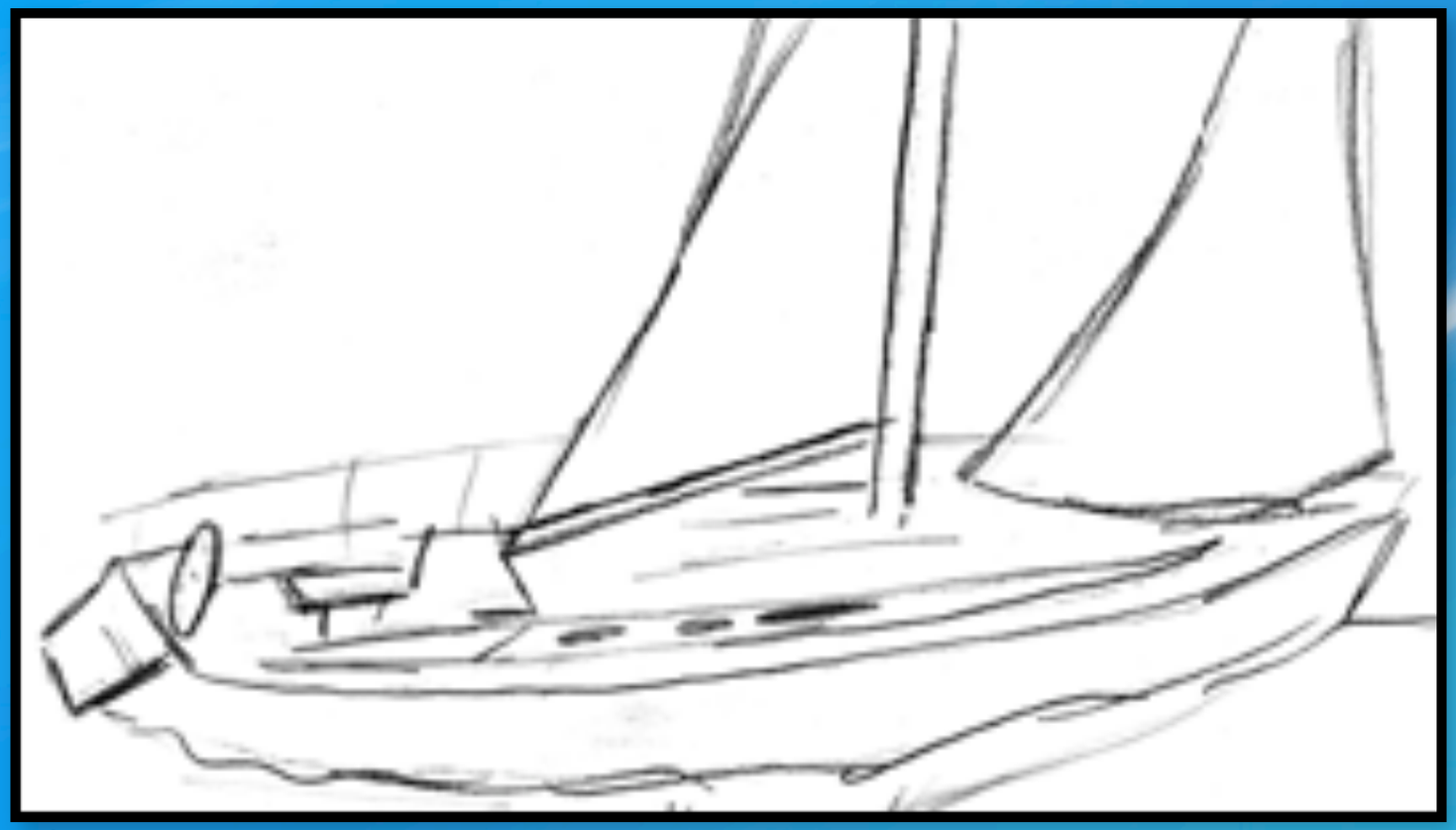
Sailboat Photograph

~ What information is being presented to the viewer?

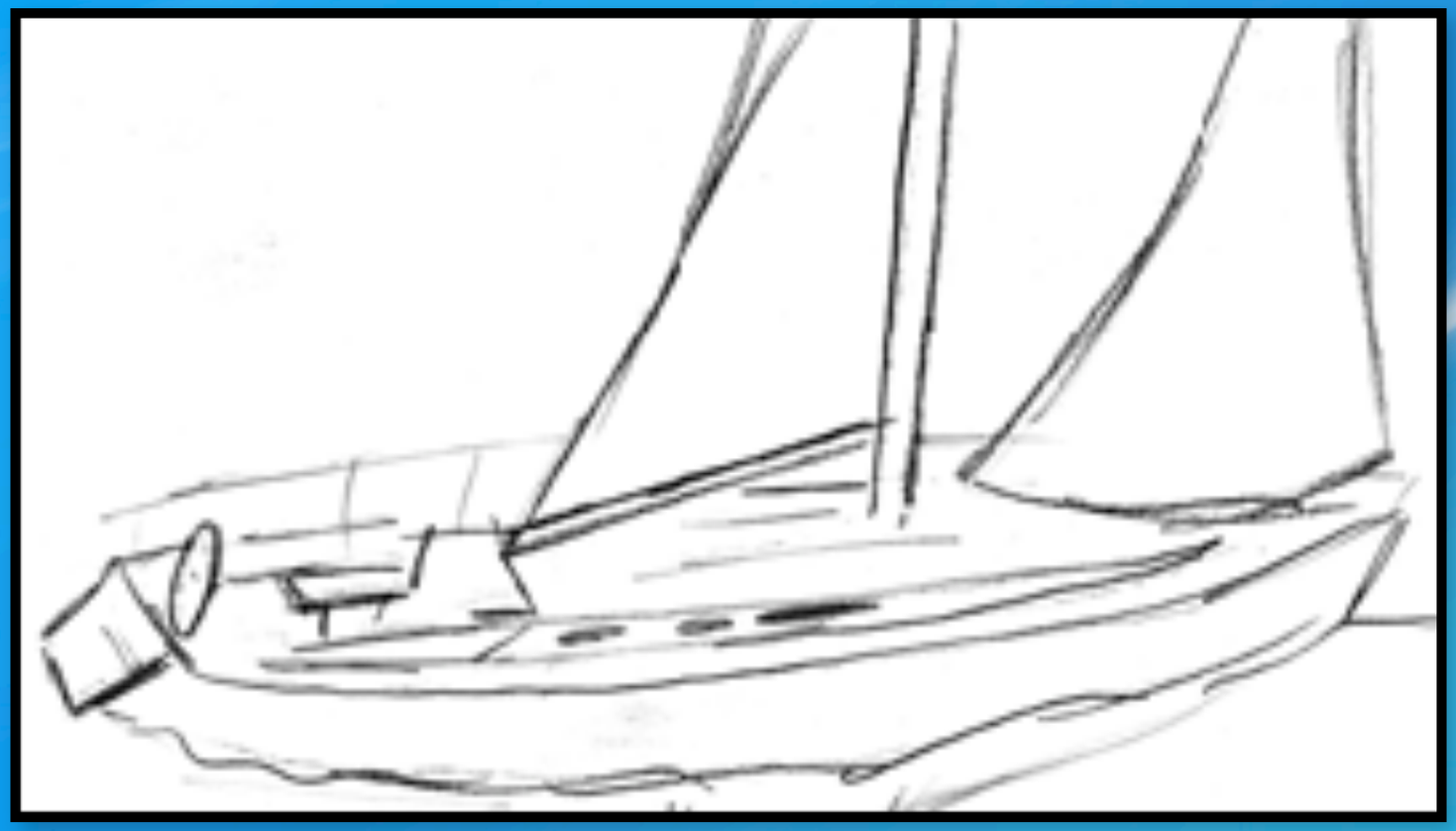
Sailboat Photograph

- What information is being presented to the viewer?
- sailboat in water, color, texture, time of day, wind, strong waves

Sailboat Abstraction

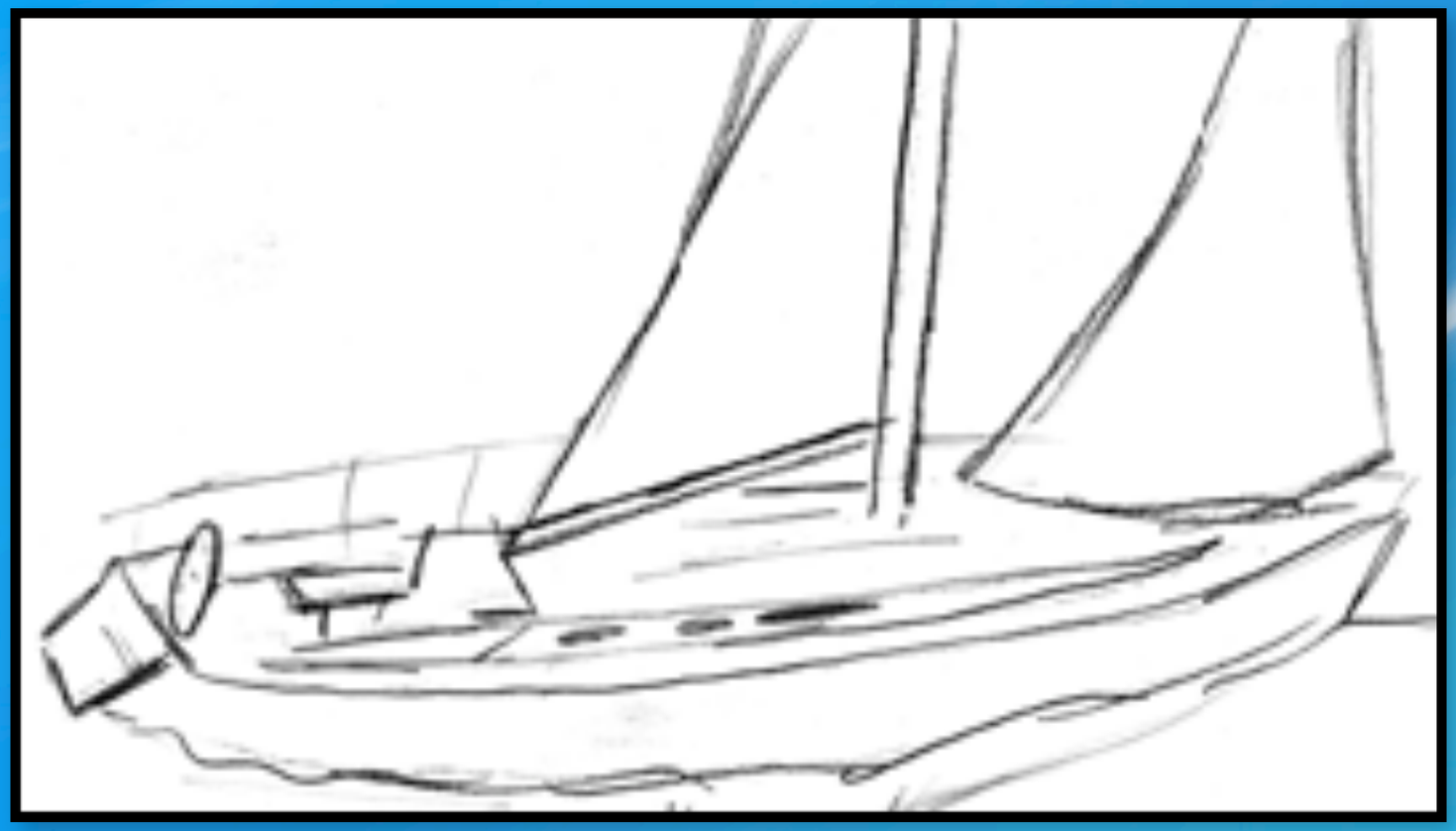


Sailboat Abstraction



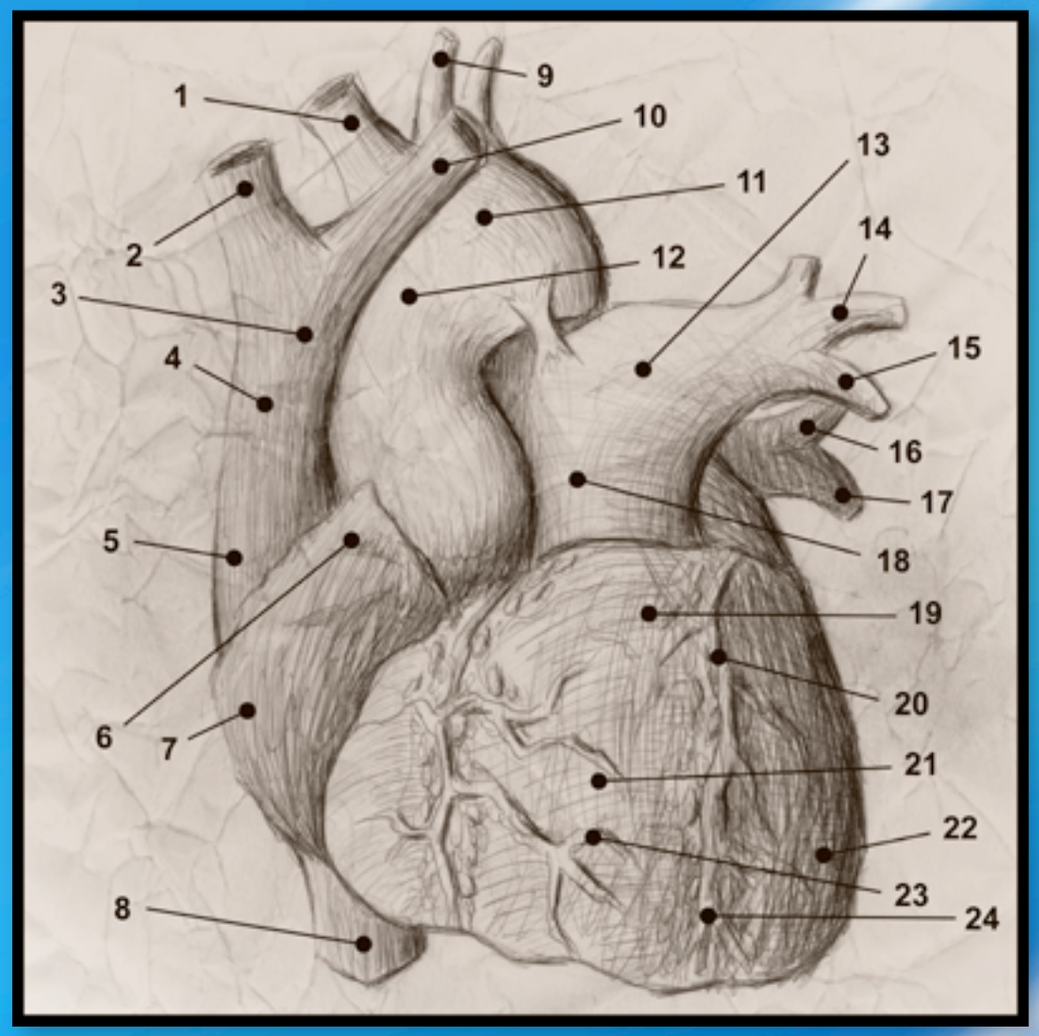
~ What information is being presented to the viewer?

Sailboat Abstraction

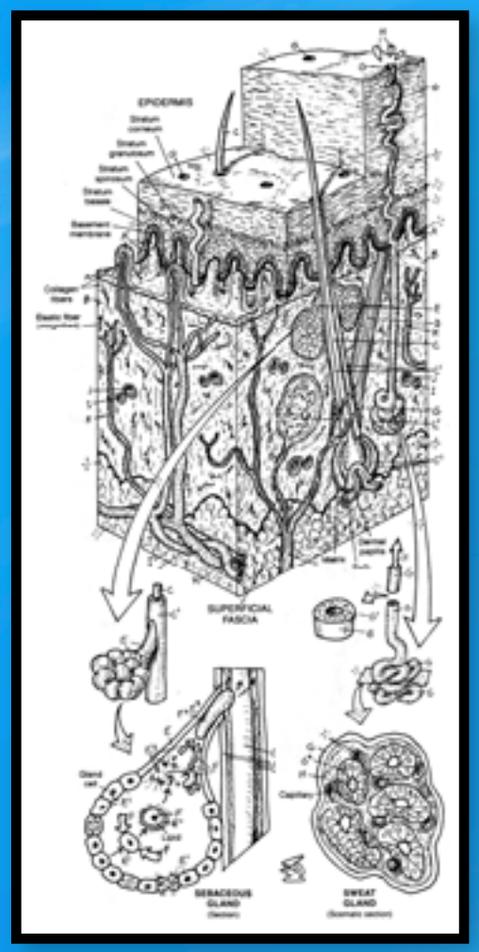


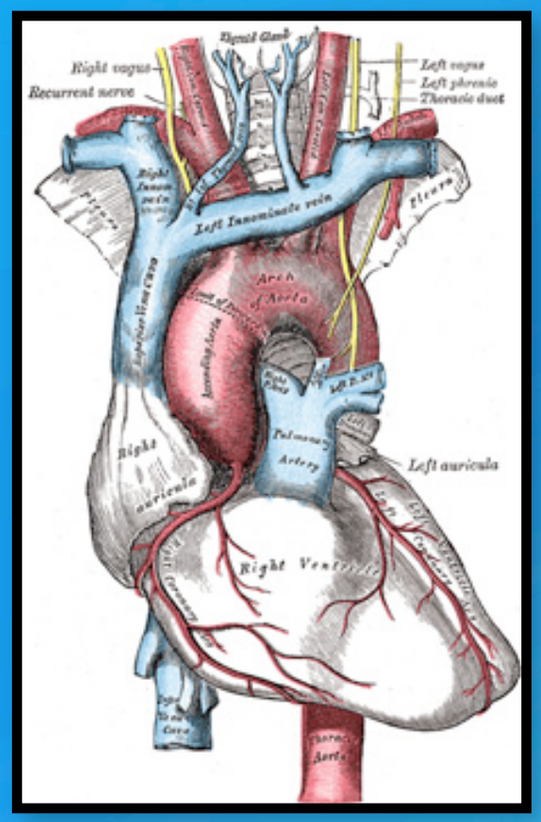
- What information is being presented to the viewer?
  - sailboat in water

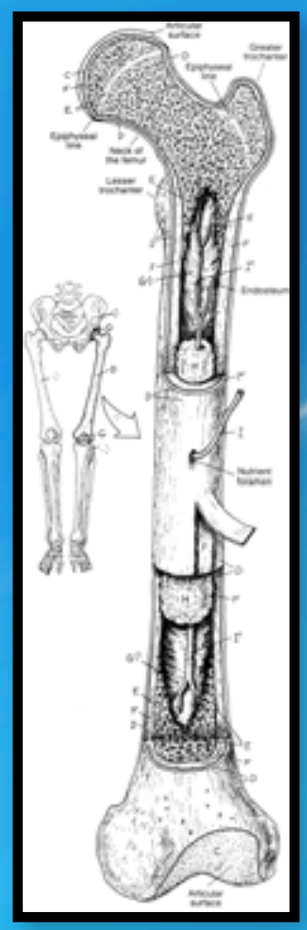
Photo vs. Illustration

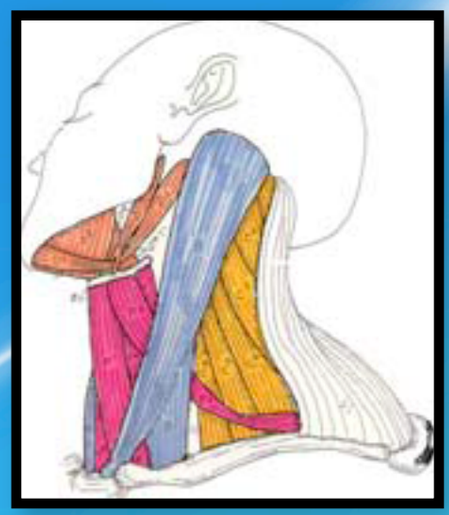


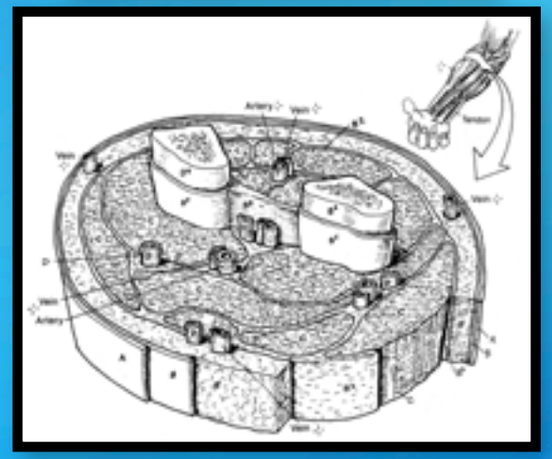
More Examples











Non-Photorealistic Rendering

- Imagery generated by illustrators (artistic, technical, scientific) has long been used to provide information that may not be readily apparent in photographs or real life.
- NPR tries to apply this concept to computer graphics.
- By departing from the limits of photorealism NPR strives to better communicate visual information.

NPR

- Apply known techniques from art instead of science.
- Focus on effective, stylized communication as opposed to “correctness.”

NPR

- We will focus on two fundamental visual cues:
  - Feature edges
  - Hatching/Shading

Feature Edges

- Early NPR algorithms used standard computer line-drawing algorithms but modified to produce hand-drawn effects.
- However, some lines are more informative than others.
- Feature edges: Silhouettes, Boundaries, and Creases.

Feature Edges

Image Space Algorithms

- Depth Map edge detection
- Normal Map edge detection

Feature Edges

Image Space Algorithms

- First- and Second-Order Image Differentials [Saito, Takahashi 90]

- Profile edges: C^0 discontinuities.

- Internal edges C^1 discontinuities.

- Use depth image (in OpenGL, glReadPixels with GL_DEPTH_COMPONENT).

- Apply Sobel filter.

Feature Edges

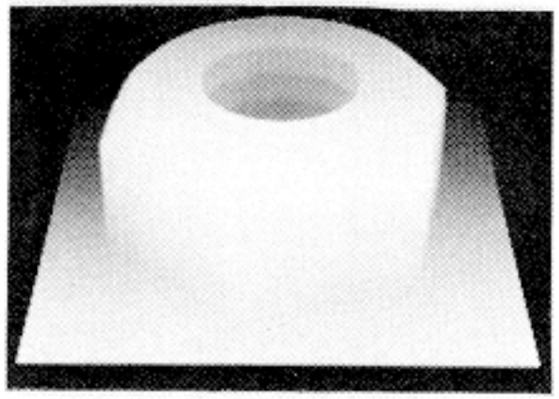
Image Space Algorithms

- First- and Second-Order Image Differentials [Saito, Takahashi 90]
  - C^0 discontinuity:
    - $g(X) = (|A+2B+C-F-2G-H| + |C+2E+H-A-2D-R|)/8$
  - C^1 discontinuity:
    - $l(X) = (8X-A-B-C-D-E-F-G-H)/3$

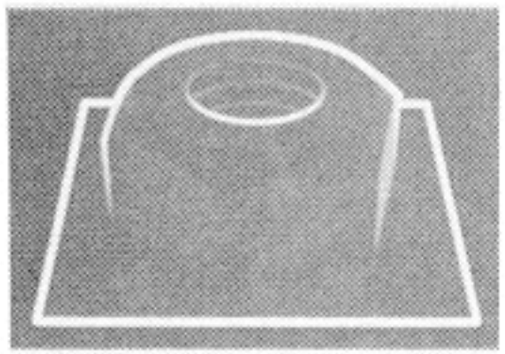
Feature Edges

Image Space Algorithms

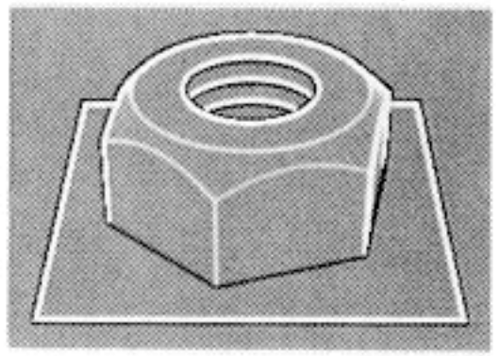
- First- and Second-Order Image Differentials [Saito, Takahashi 90]
  - Avoid “fat” edges by performing non-maximal suppression.



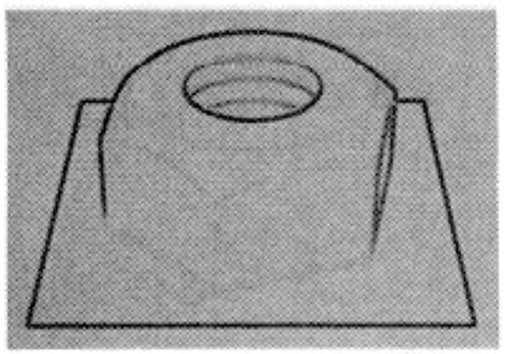
depth image



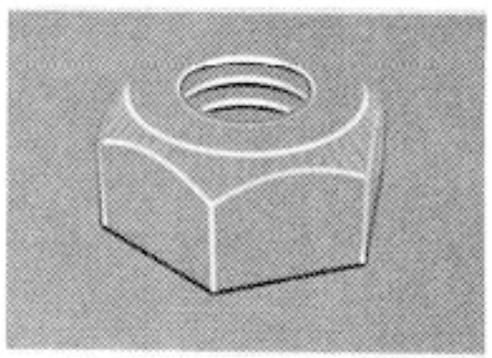
1st order differential



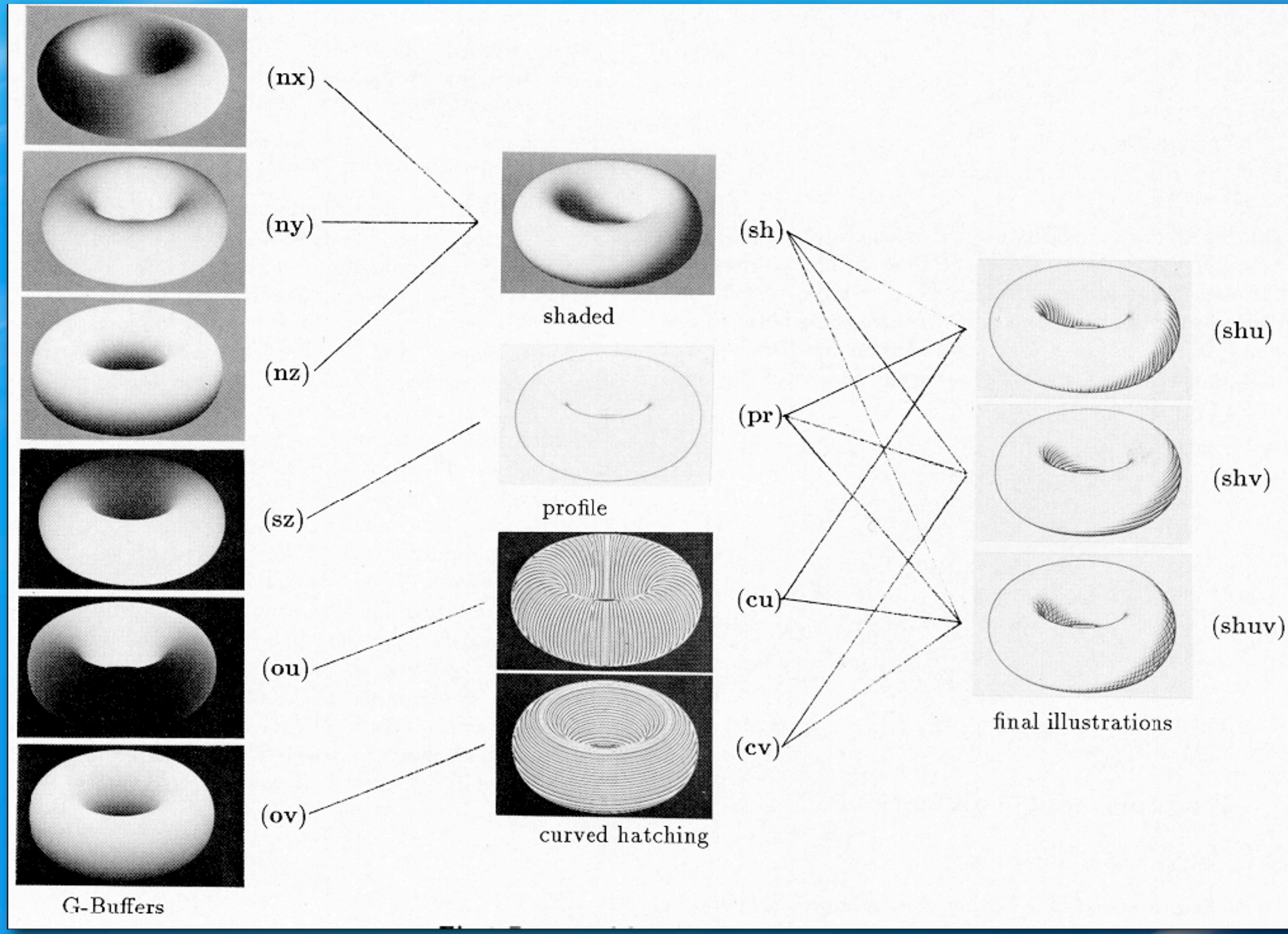
2st order differential

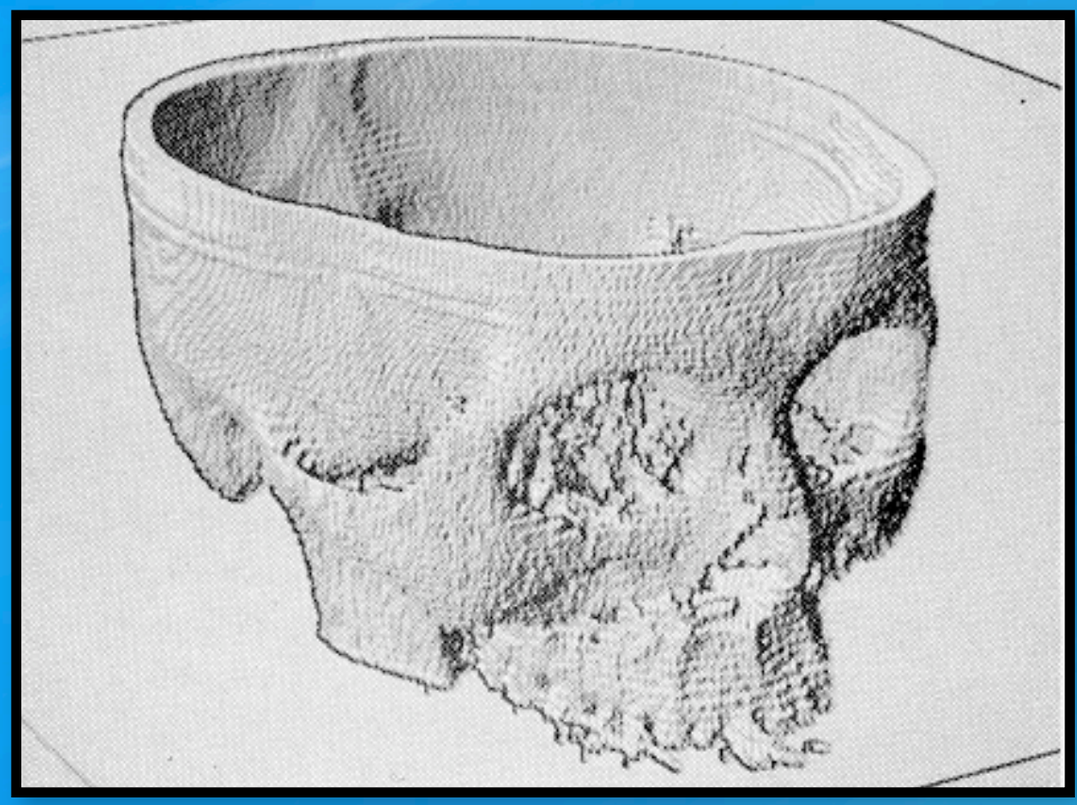


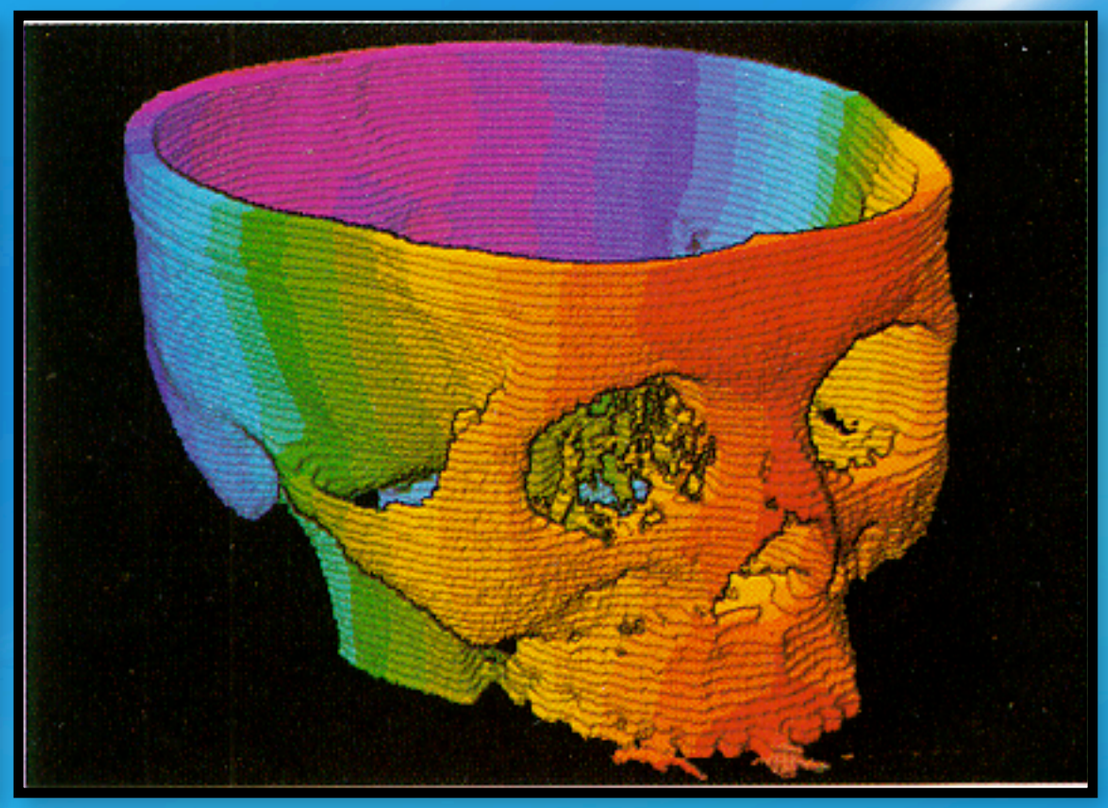
profile image



internal edge image


(nx)
(ny)
(nz)
(sz)
(ou)
(ov)
G-Buffers
shaded
profile
curved hatching
(sh)
(pr)
(cu)
(cv)
(shu)
(shv)
(shuv)
final illustrations

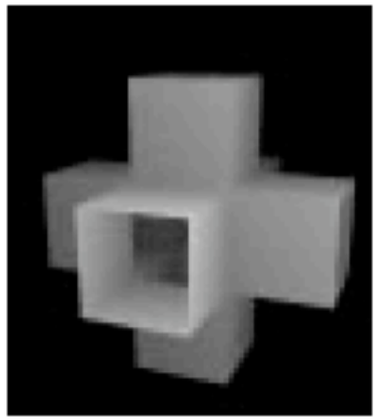




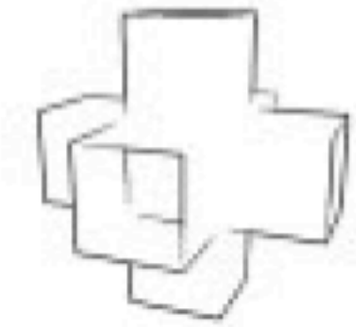
Feature Edges

Image Space Algorithms

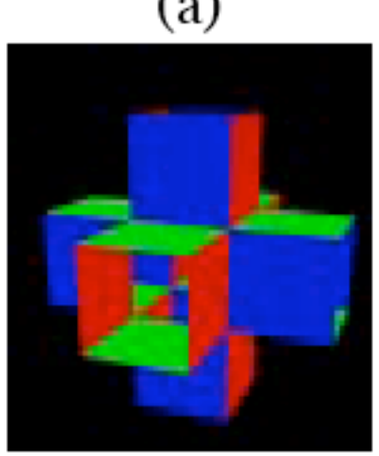
- Using Normal Maps to Find Creases and Boundaries [Decaudin]
  - Using just depth can miss certain discontinuities and object boundaries.
  - Perform edge detection on an image of the surface normals.



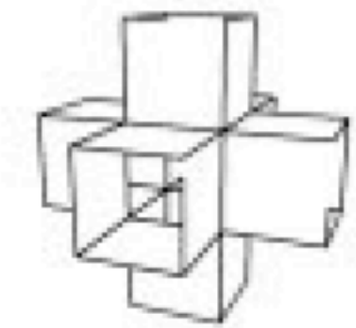
(a)



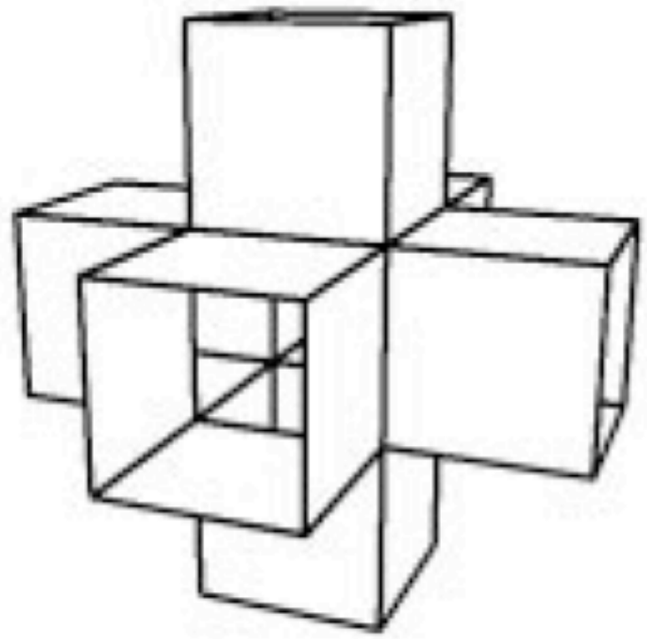
(b)



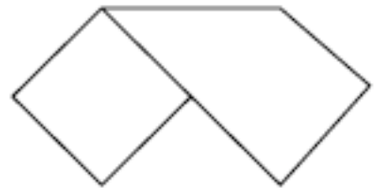
(c)



(d)

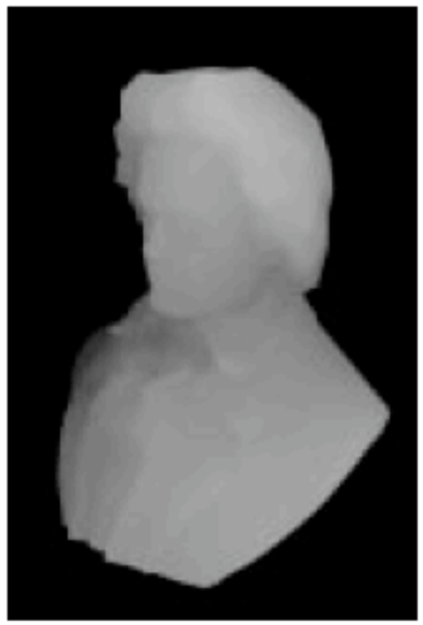


(e)



(f)

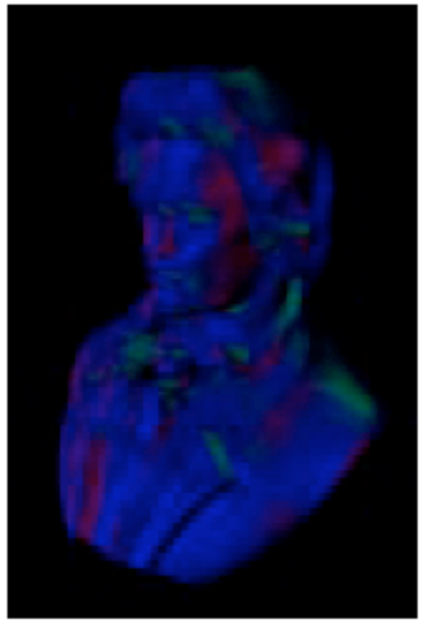
(g)



(a)

(b)

(e)



(c)

(d)

Feature Edges

Object Space Algorithms

- More involved than the image space methods.
- Can produce curves with much higher precision.
- Resulting curves are also suitable for additional processing.
- e.g. render curves with natural brush strokes.

Feature Edges

Object Space Algorithms

- Interested in:
  - Silhouettes
  - Surface boundaries
  - Creases: discontinuities on an otherwise smooth surface.
  - Self-intersections
  - Other surface curves, such as isoparametric curves.

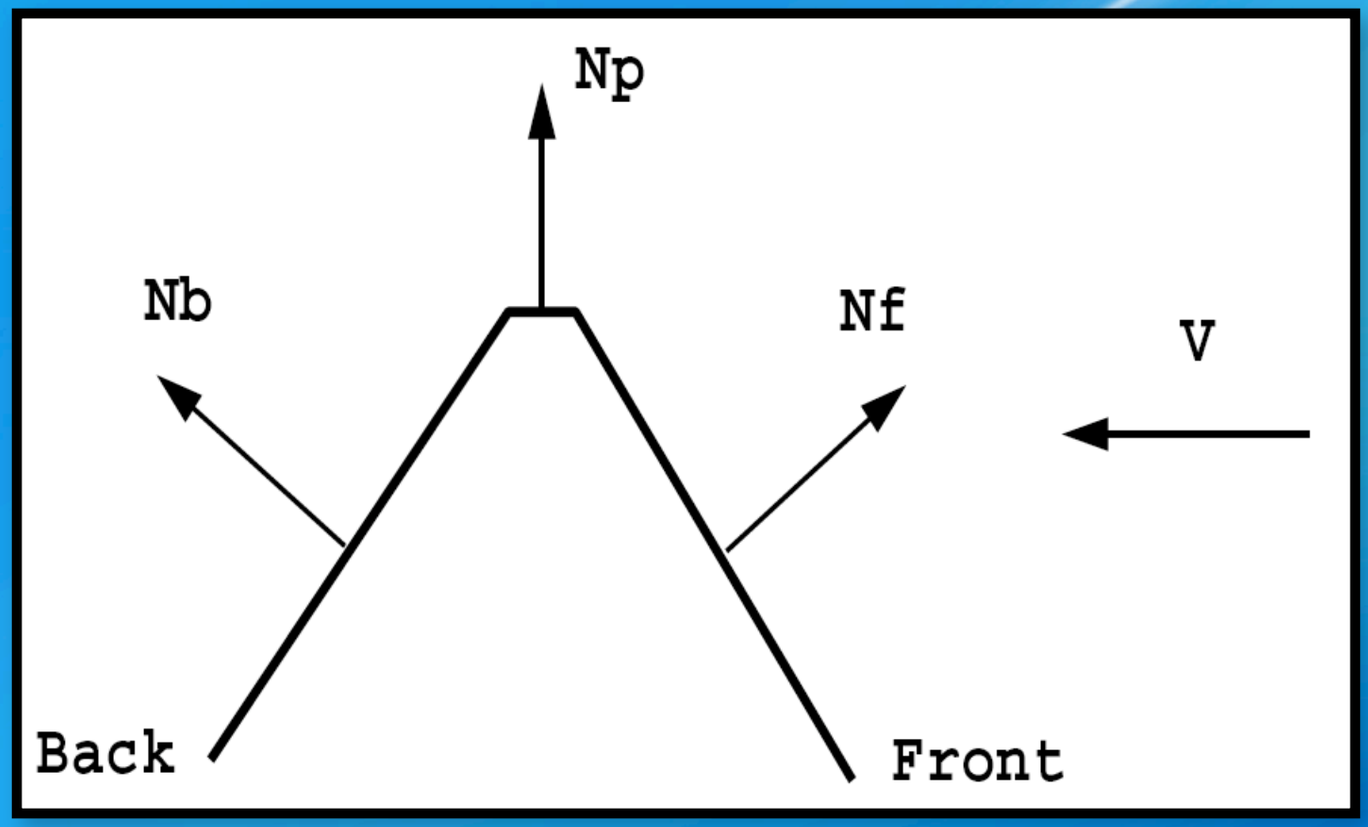
Feature Edges

Object Space Algorithms

- Surface boundaries and Creases can be detected as a pre-process.
  - e.g. brute force: for each edge in mesh, if adjacent triangle normals differ by more than some α, mark as crease.
- We will focus on silhouettes.
  - Cannot naively pre-process because it is view-dependent.

Silhouette Edges
Object Space Algorithms

~ For a polygonal mesh, the silhouettes are exactly the set of mesh edges that connect a front-facing polygon and a back-facing polygon.



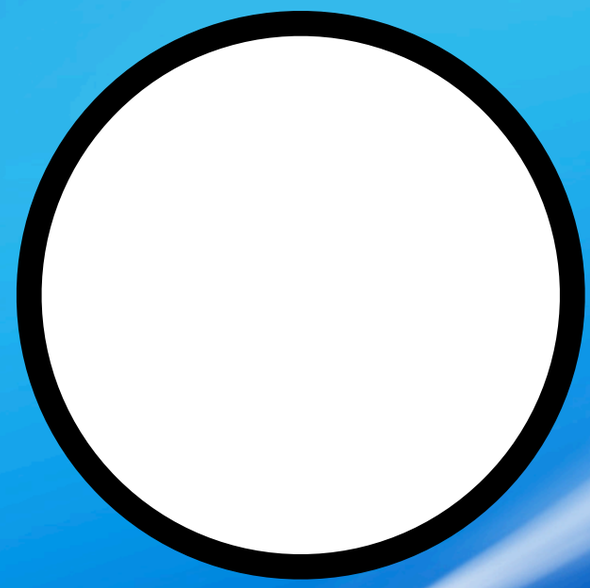
Silhouette Edges

Object Space Algorithms

- Brute force: Iterate over every mesh edge and check the normals of the adjacent faces - expensive.
- Speed up by testing only a few edges randomly [Markosian et al]. This produces an approximation.
- The edge buffer: use a clever data structure to speed this up [John W. Buchanan, Mario C. Sousay].

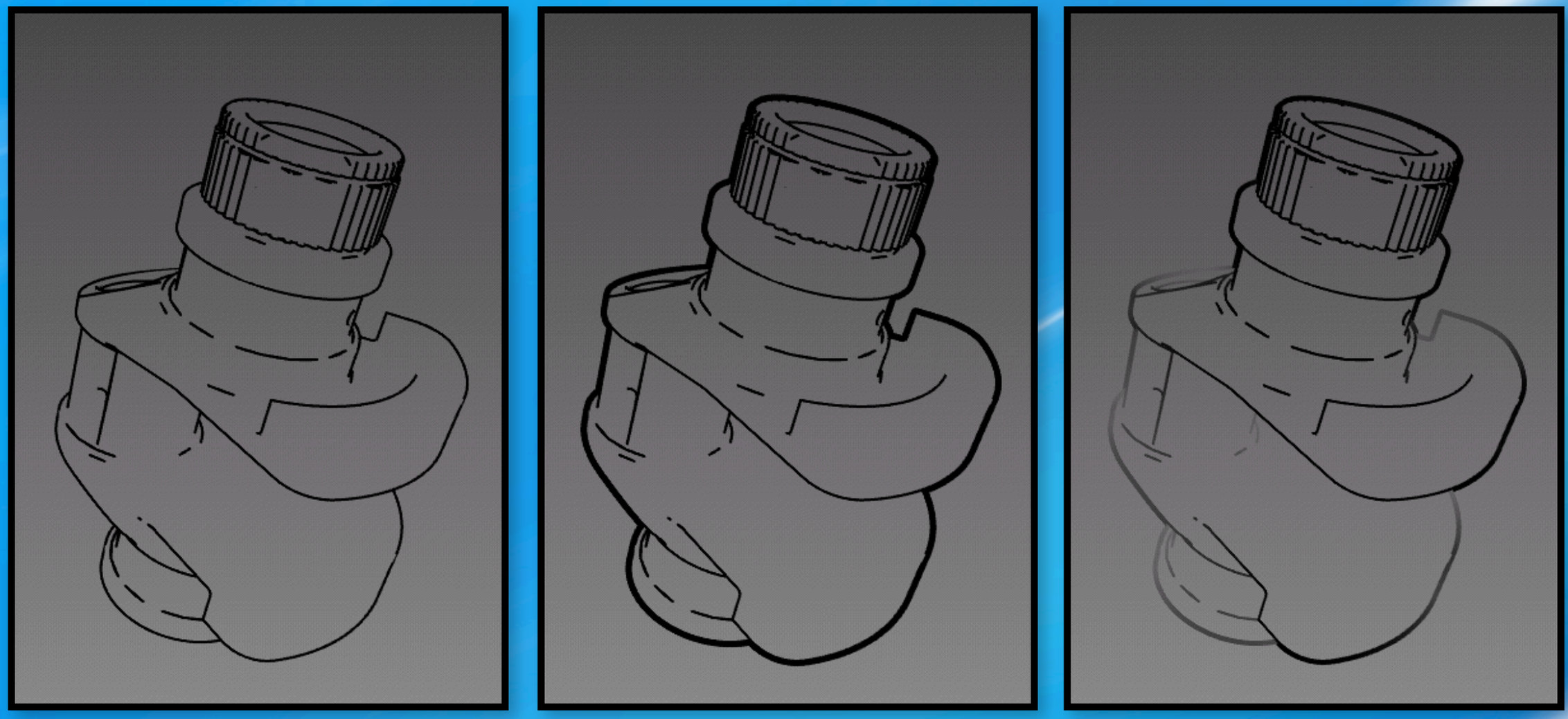
Silhouette Edges
Other Algorithms

- Silhouette edges by environment mapping
  - Use a circular environment map which is white, with a black border. This image is accessed based on the reflection direction. Pixels were $N \cdot V = 0$ will map to periphery of the circle.



Feature Edges

Extras

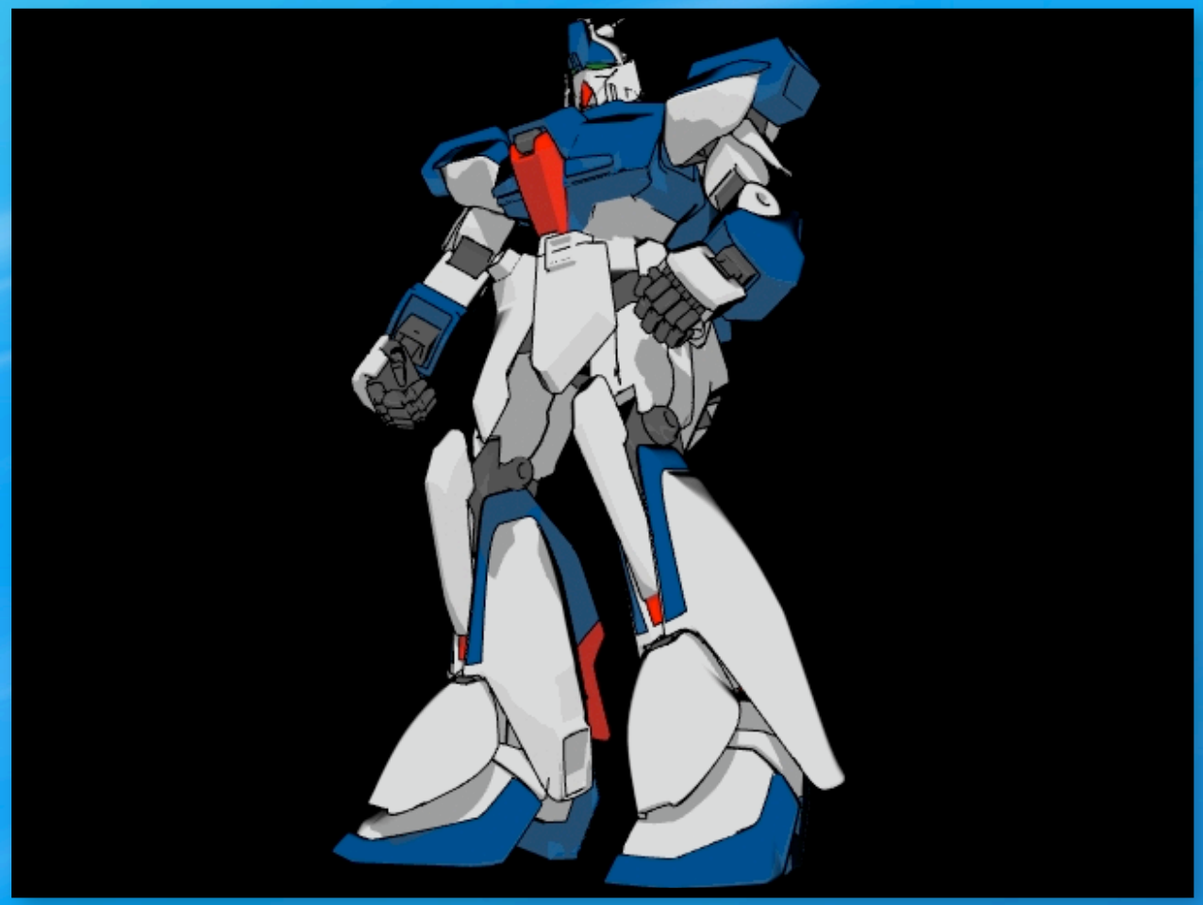


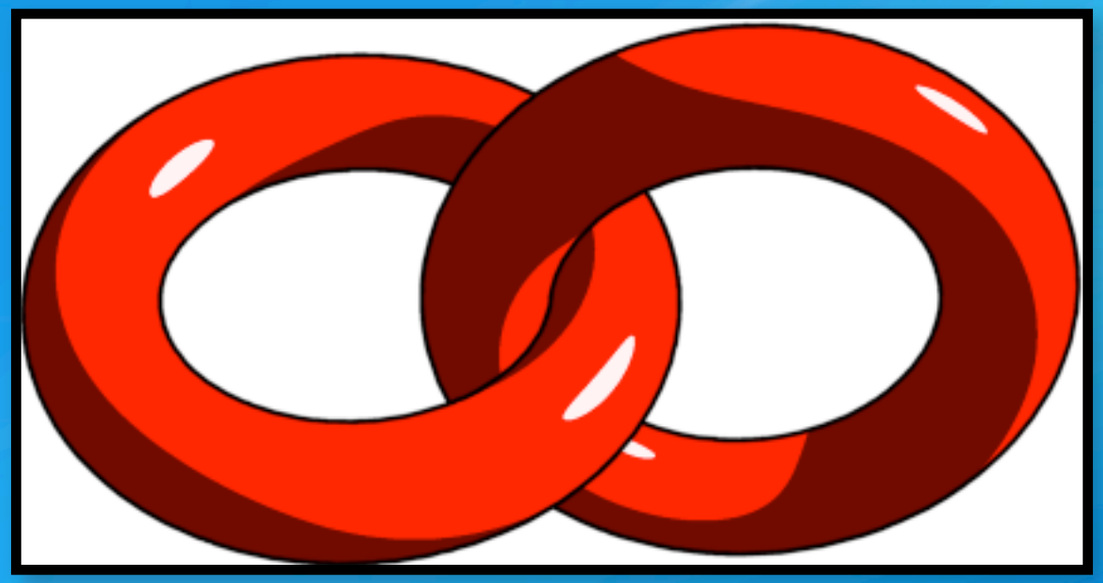
~ Use line weight to suggest structure.

Filling in the Lines

Cel Shading

- We are now able to extract various types of feature edges.
- How do we fill in the empty regions?
- One common technique, called Cel-shading, mimics the appearance of “cartoon”-style drawings.





Filling in the Lines

Cel Shading

~ How does cartoon-style shading differ visually from the normal diffuse shading we have used in ray tracing?

Filling in the Lines

Cel Shading

- Cartoon style images do not have smoothly shaded lighting. The shaded lighting is split up into a few distinct intensity levels.
- Instead of applying regular N·L lighting equation. Use the value of N·L to interpolate a 1D texture.

Filling in the Lines

Pencil Shading

Filling in the Lines

Pencil Shading

~ What are some important visual cues present in pencil drawn images that distinguishes them from other media?

Filling in the Lines

Pencil Shading

- What are some important visual cues present in pencil drawn images that distinguishes them from other media?
  - Paper texture: influences where graphite gets deposited.
  - Artists use pencil strokes to suggest both tone and texture.
  - Stroke direction can be used to suggest structure.

Filling in the Lines

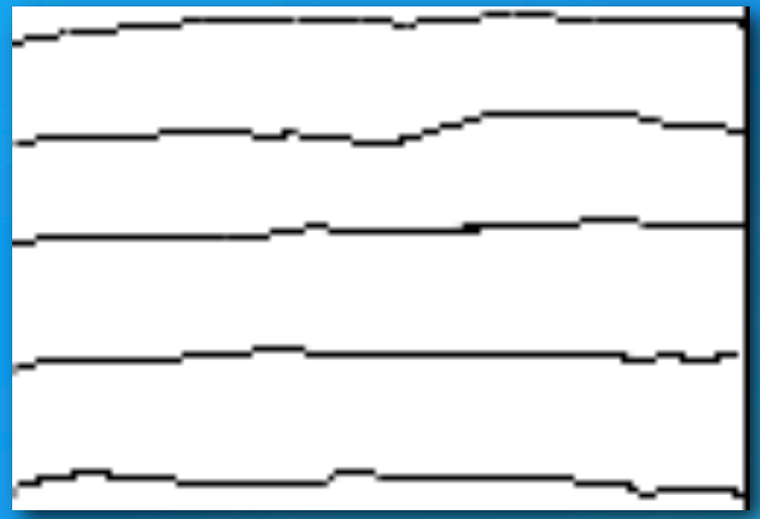
Pencil Shading

- "Stylized Rendering Techniques for Scalable Real-time 3D Animatition." Lake, Marshall, Harris, and Blackstein 00.
- Extended the cell-shading idea by using N·L to index into a specific 2D texture with the appropriate density.
- This allowed them to produce images resembling pencil shaded drawing.
- Completely empirical technique.

Pencil Shading

Algorithm Overview

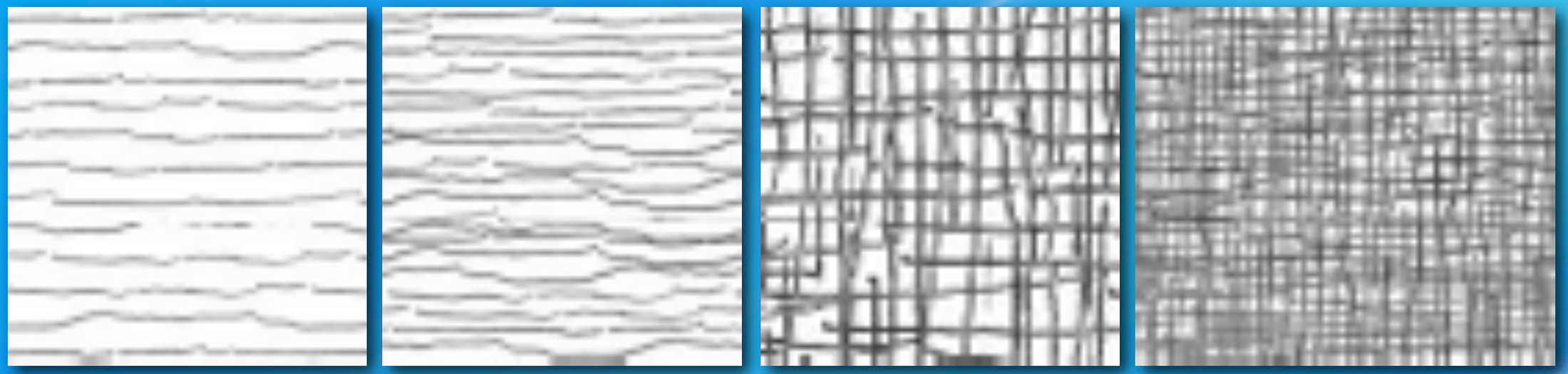
- As a preprocess, read in a pencil stroke texture and a paper texture.



Pencil Shading

Algorithm Overview

- Next, generate M two-dimensional textures by randomly selecting pencil strokes and placing them with a specific density over the texture.



Pencil Shading

Algorithm Overview

- Draw the background.
- For each vertex, compute N·L. Use this value to index into the M precomputed textures.
- Create a list of polygons for each of the M textures.
  - For each polygon, if vertices have same index value, put the polygon in that texture's polygon list. Otherwise, subdivide polygon and repeat.

Pencil Shading

Algorithm Overview

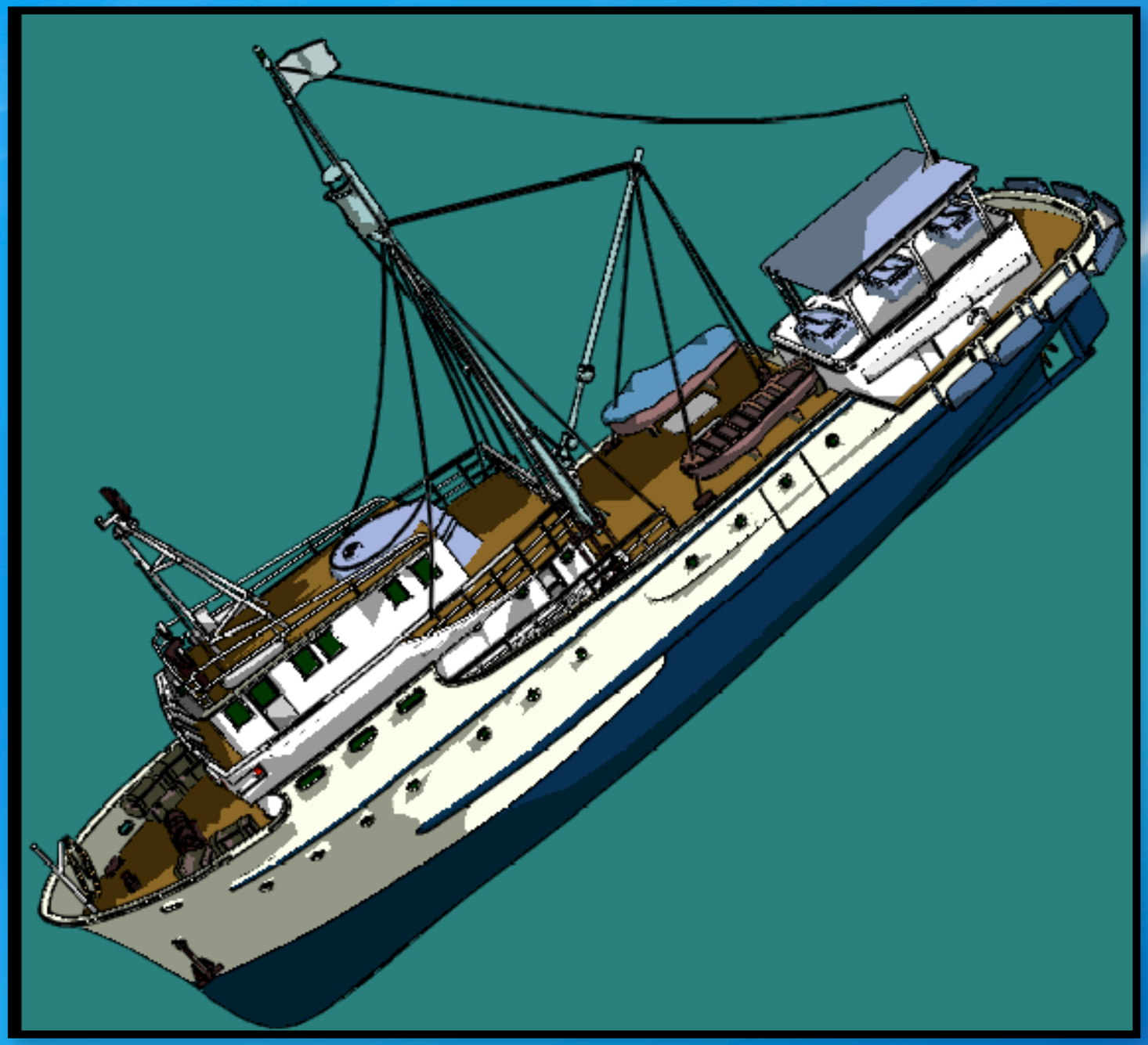
- Since we are drawing textures, and not texels, we need to compute UV coordinate for each vertex.
- Draw textures polygons using the appropriate texture.
- Two approaches for UV coordinates:
  - associate normalized screen-space coordinates for UV at every frame.
  - fix UV texture coordinates for each polygon.

Pencil Shading

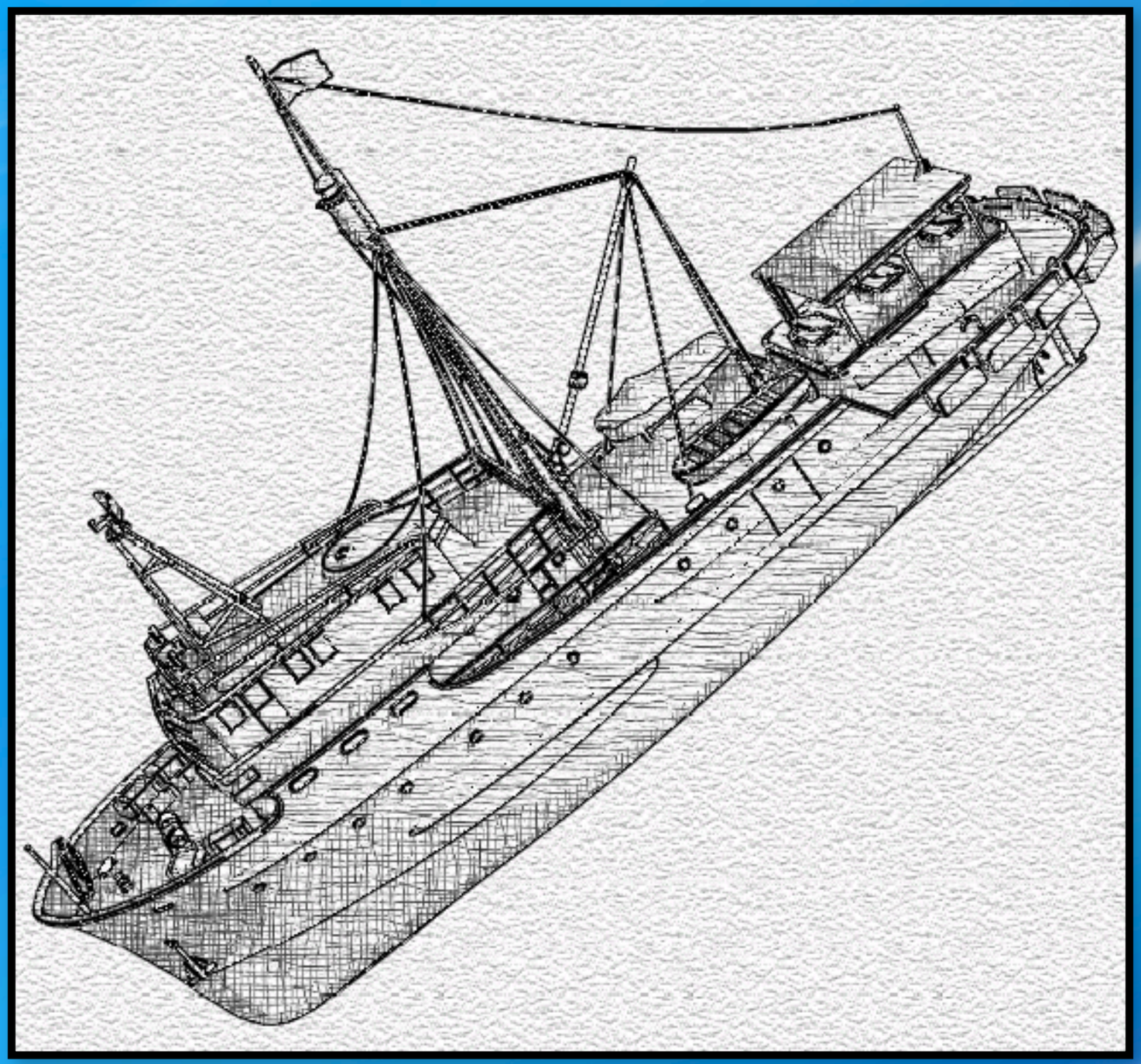
Algorithm Overview

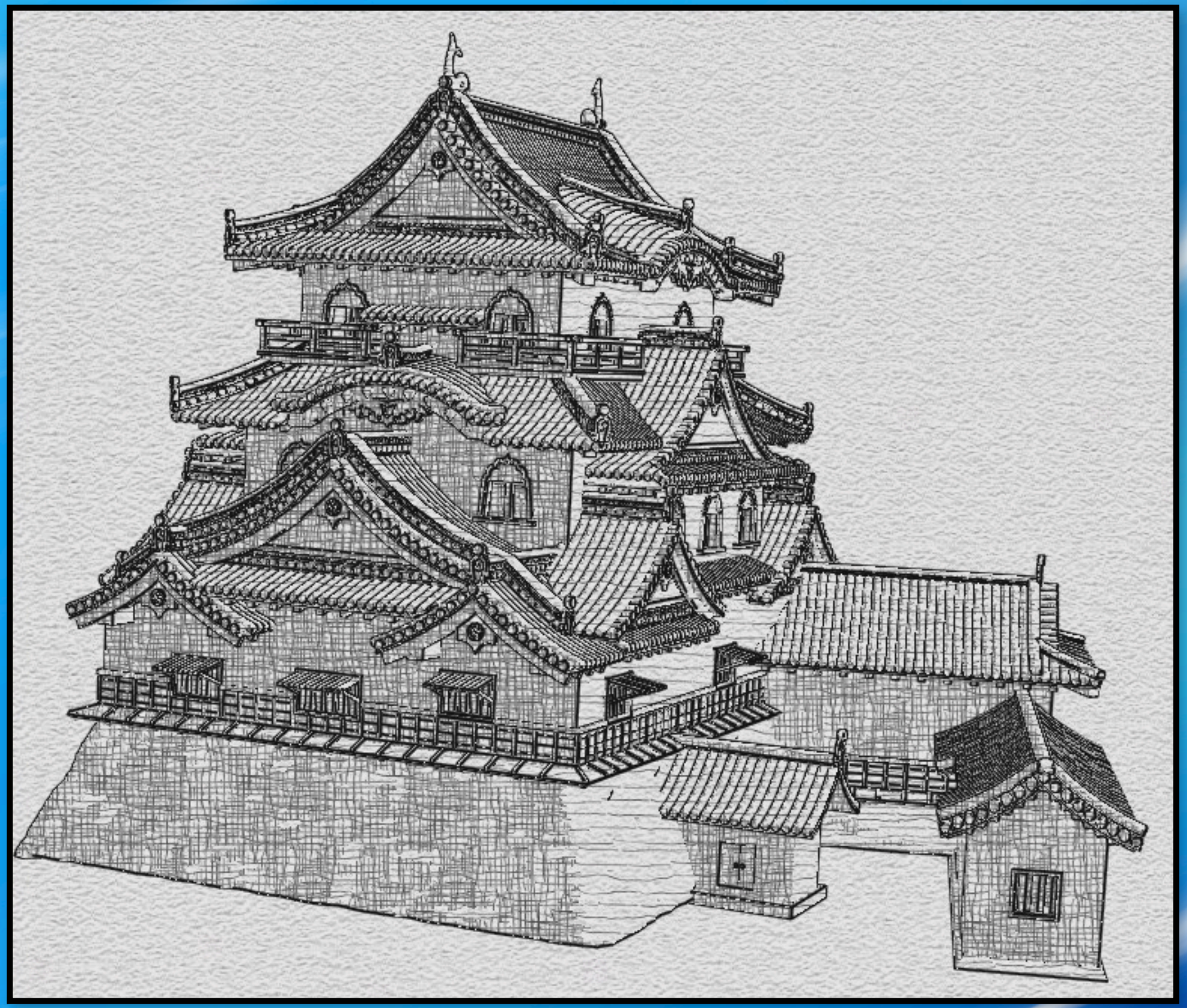
- Two approaches for UV coordinates:
  - associate normalized screen-space coordinates for UV at every frame.
    - Preserves hand-drawn feel, works well for static images. However, animated images “crawl” or “swim” through texture.
  - fix UV texture coordinates for each polygon.
    - frame-to-frame coherence, flat appearance.

Cartoon Shaded



Pencil Shaded

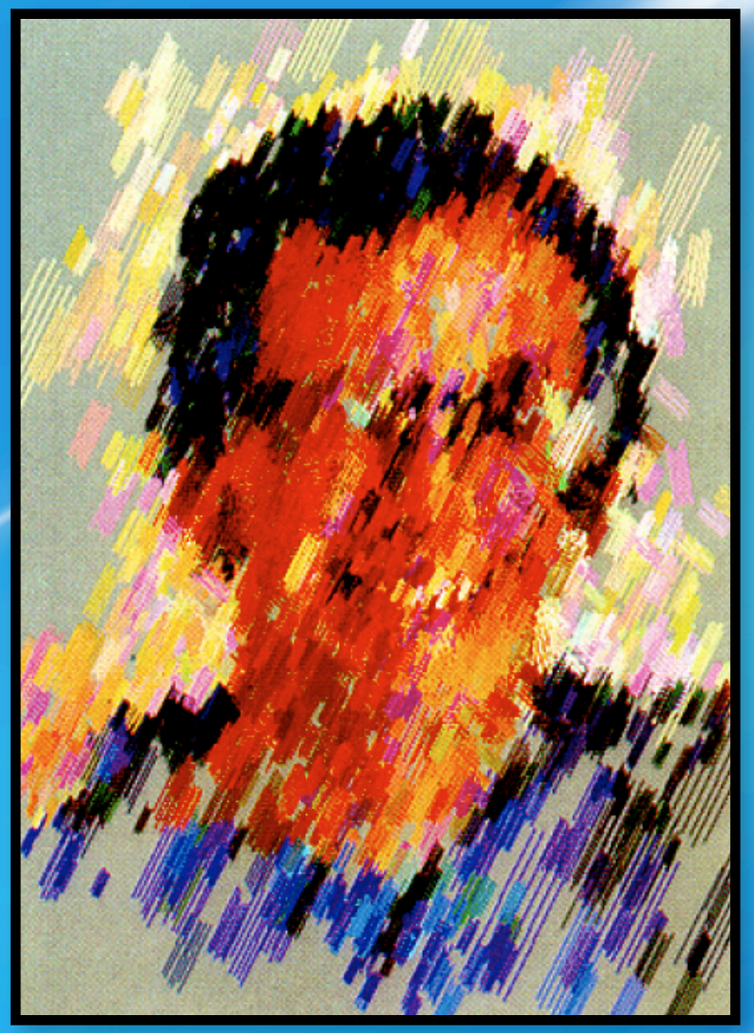


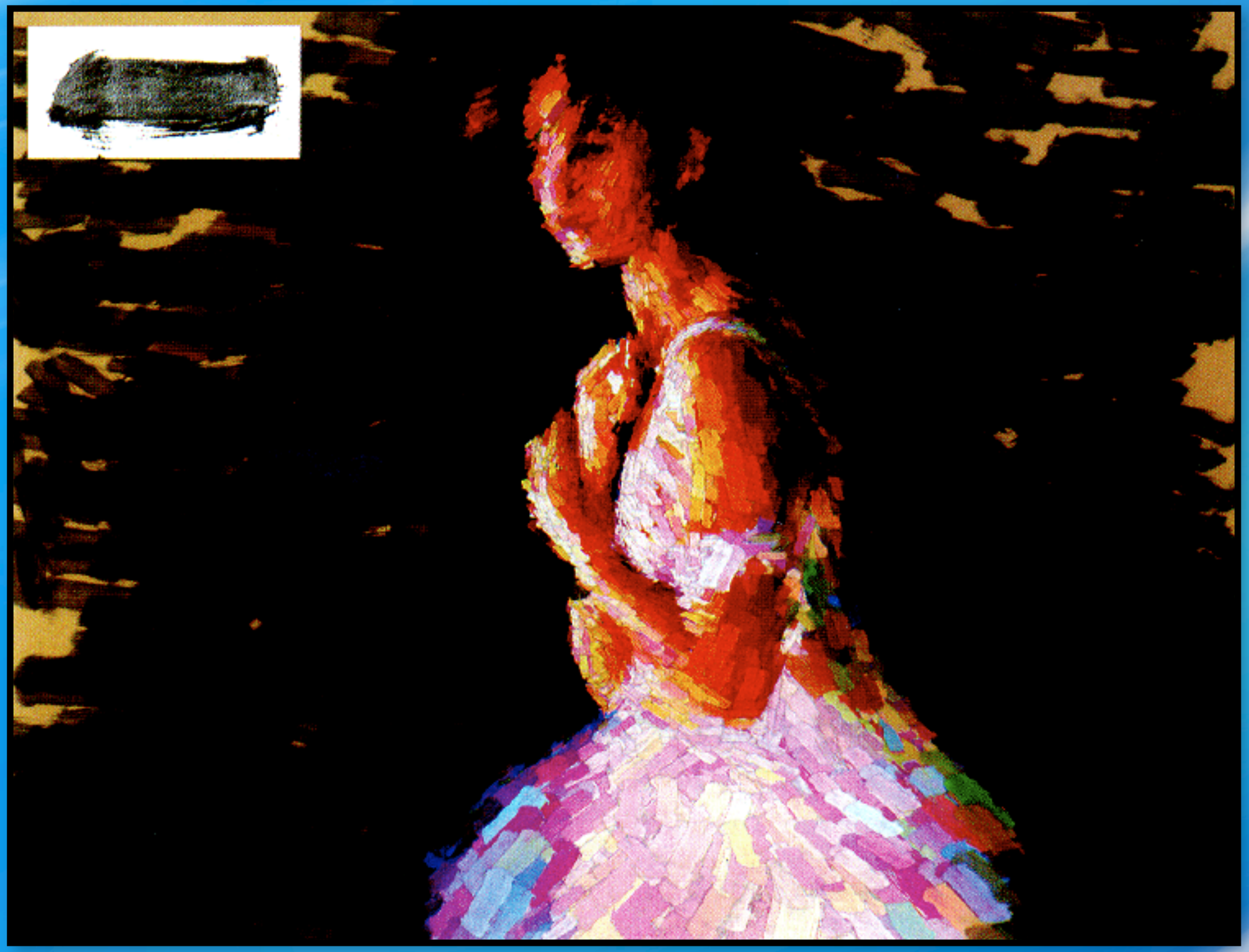


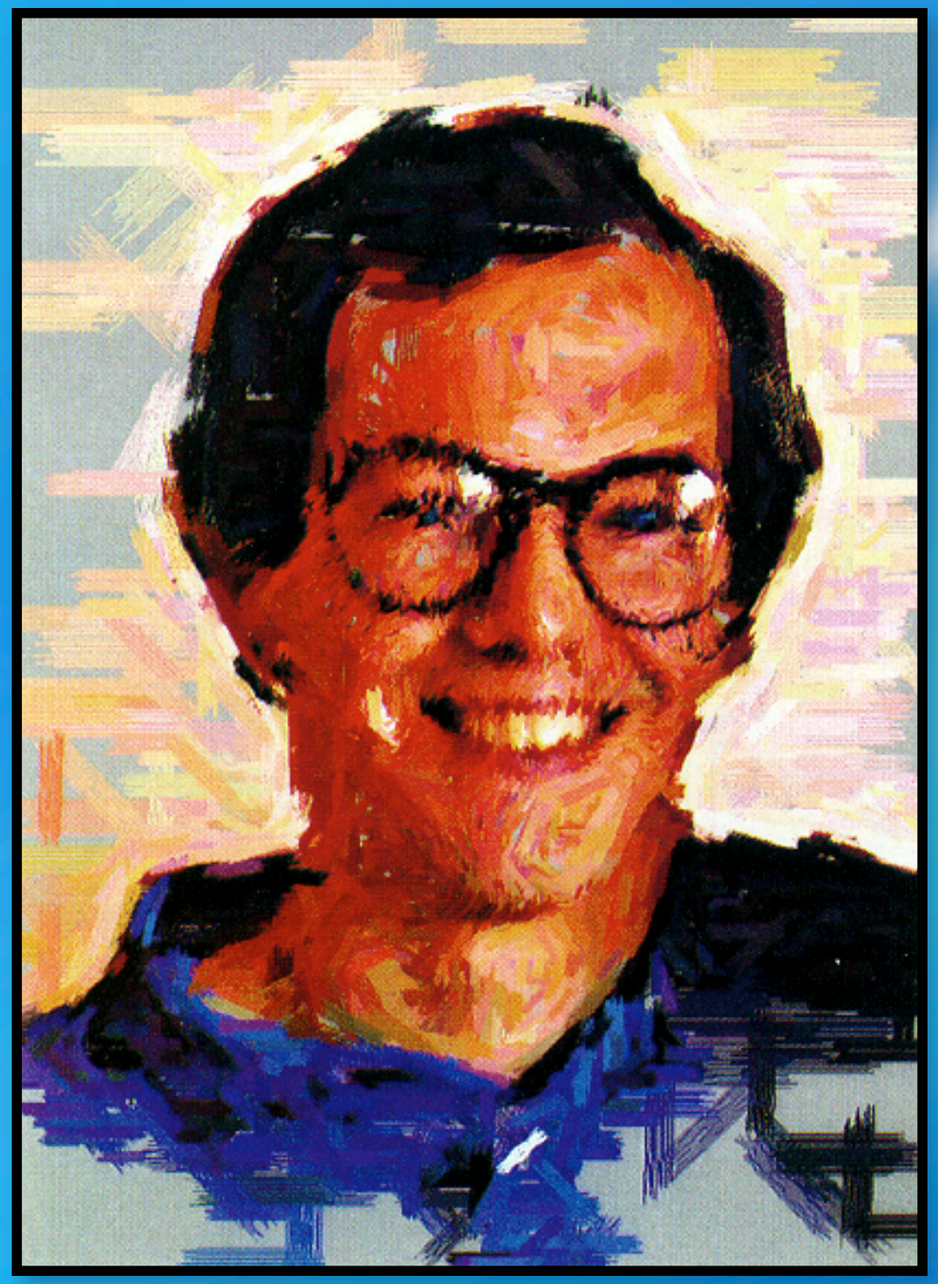
Visual Break

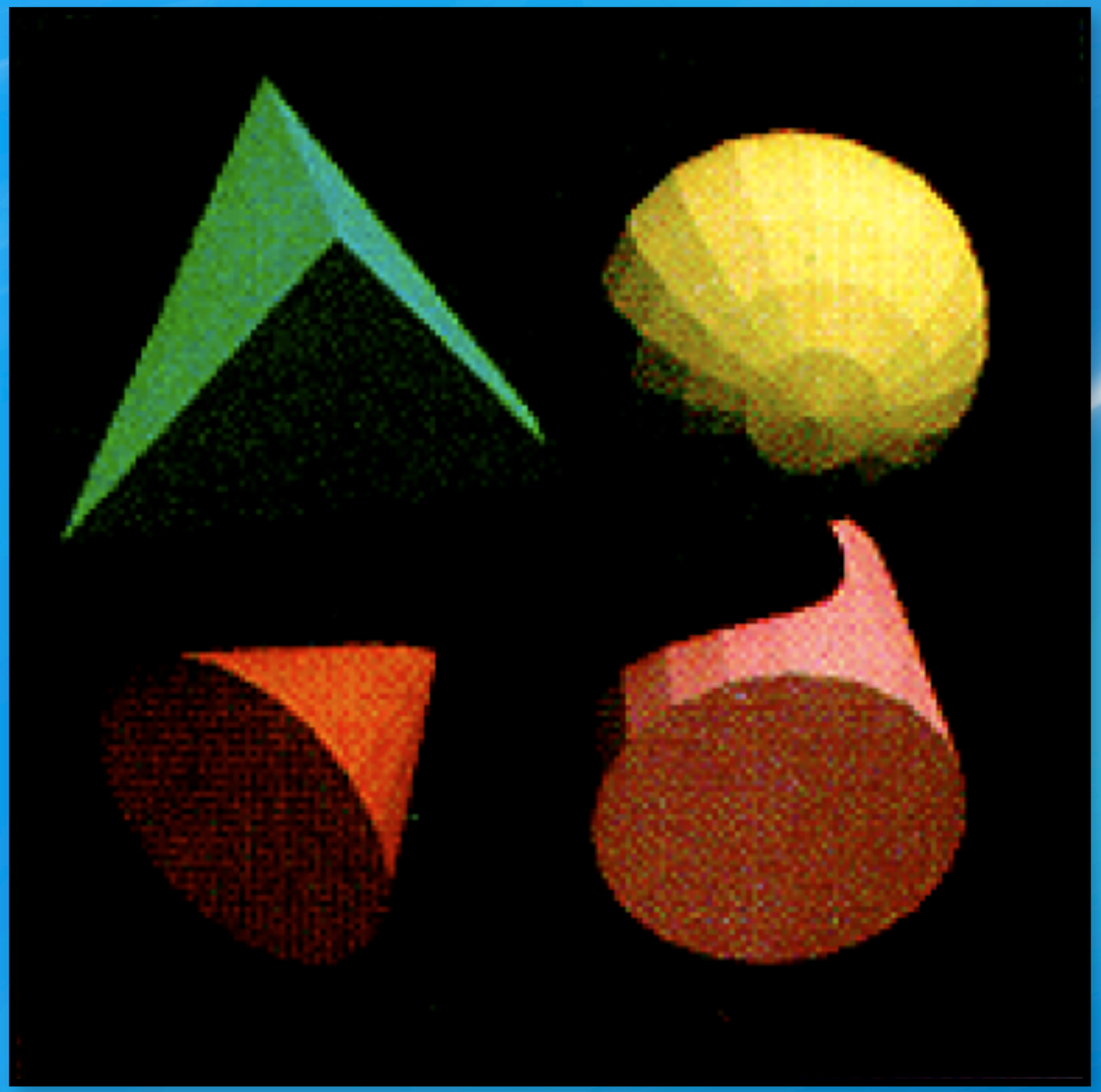
Paint By Numbers

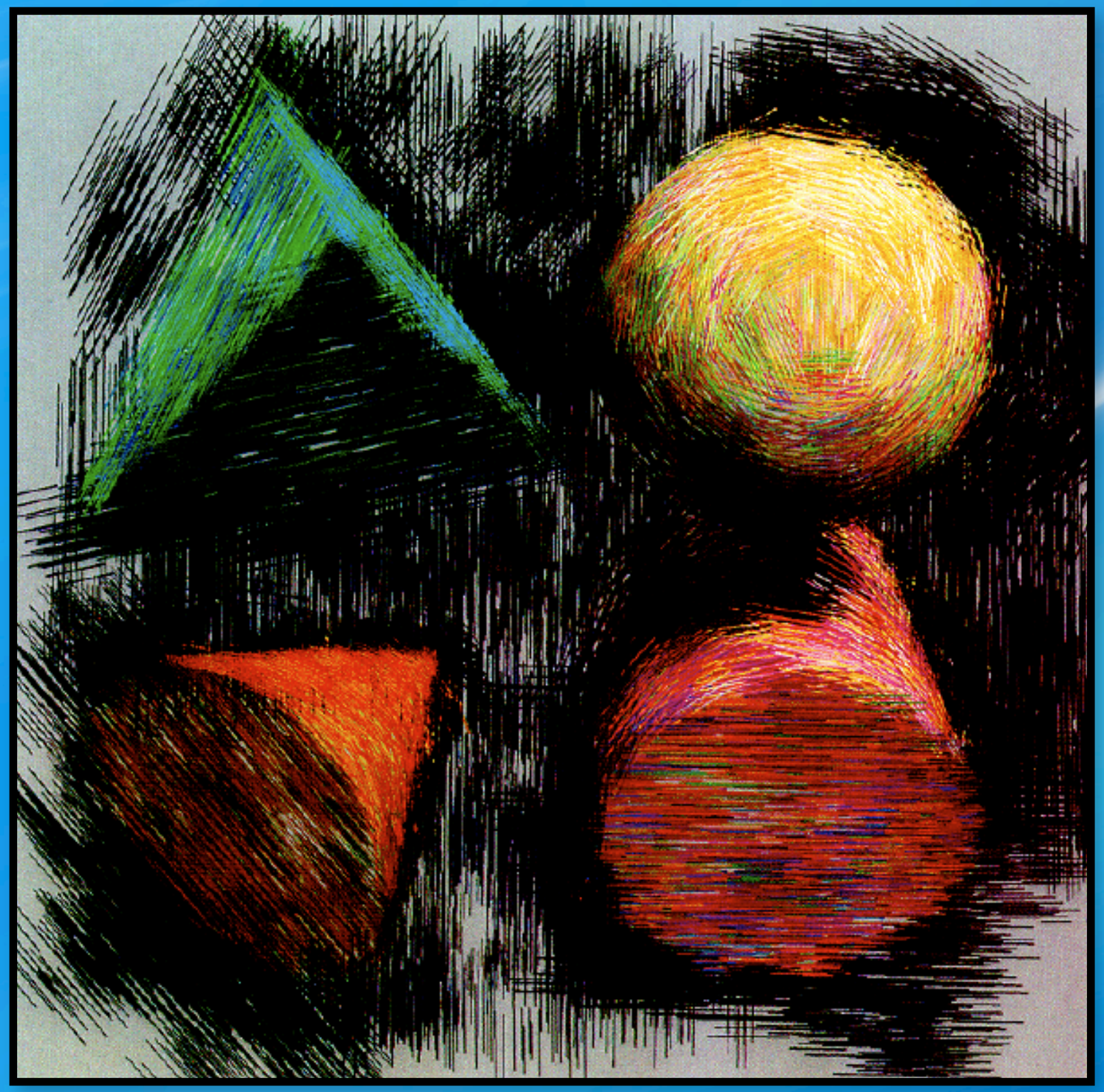
- "Paint By Numbers: Abstract Image Representations." Paul Haeberli 90.
  - Paint brush strokes on images to create stylization.











"Processing Images and Video for an Impressionistic Effect."
Peter Litwinowicz. 1997

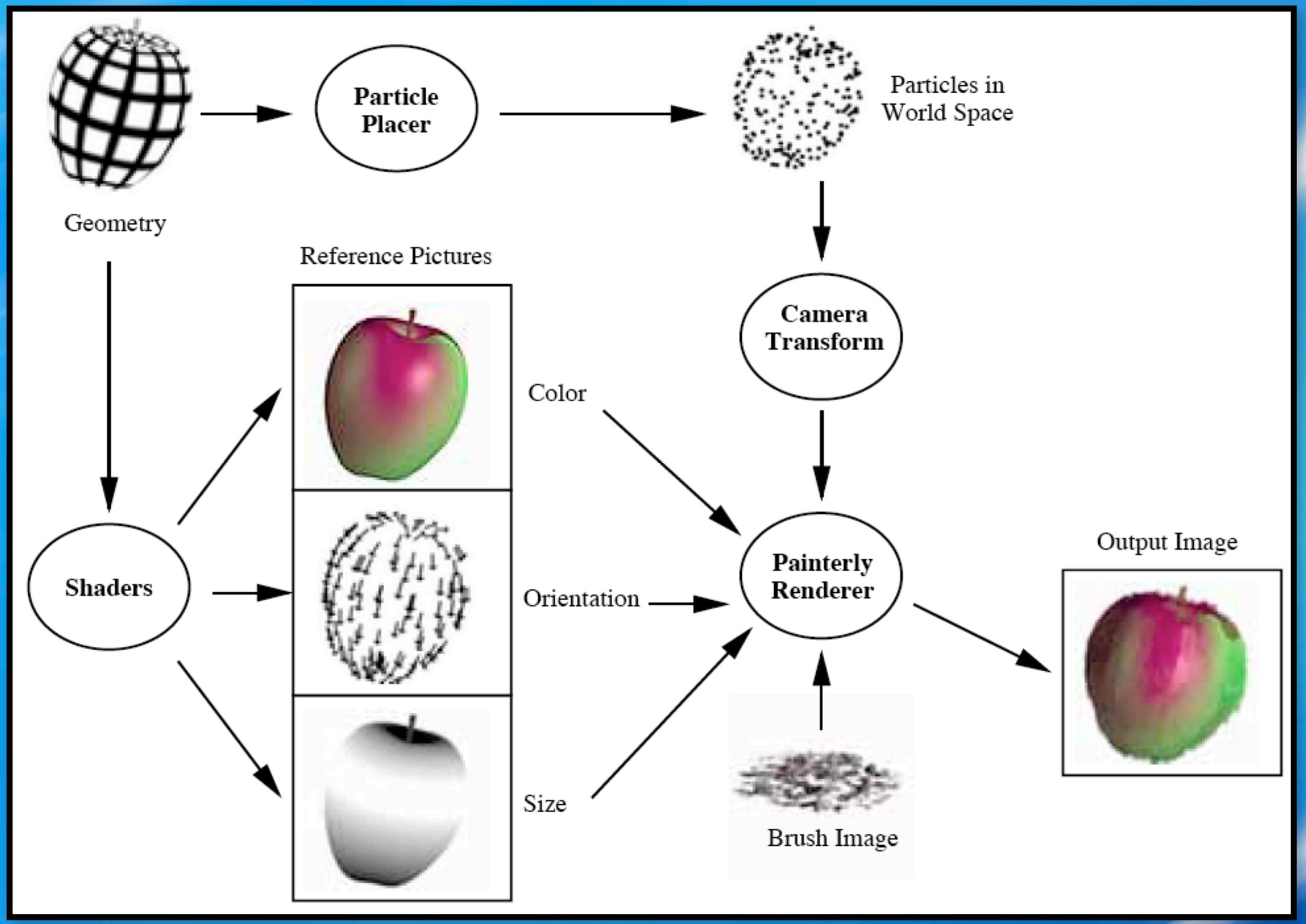
Painterly Rendering

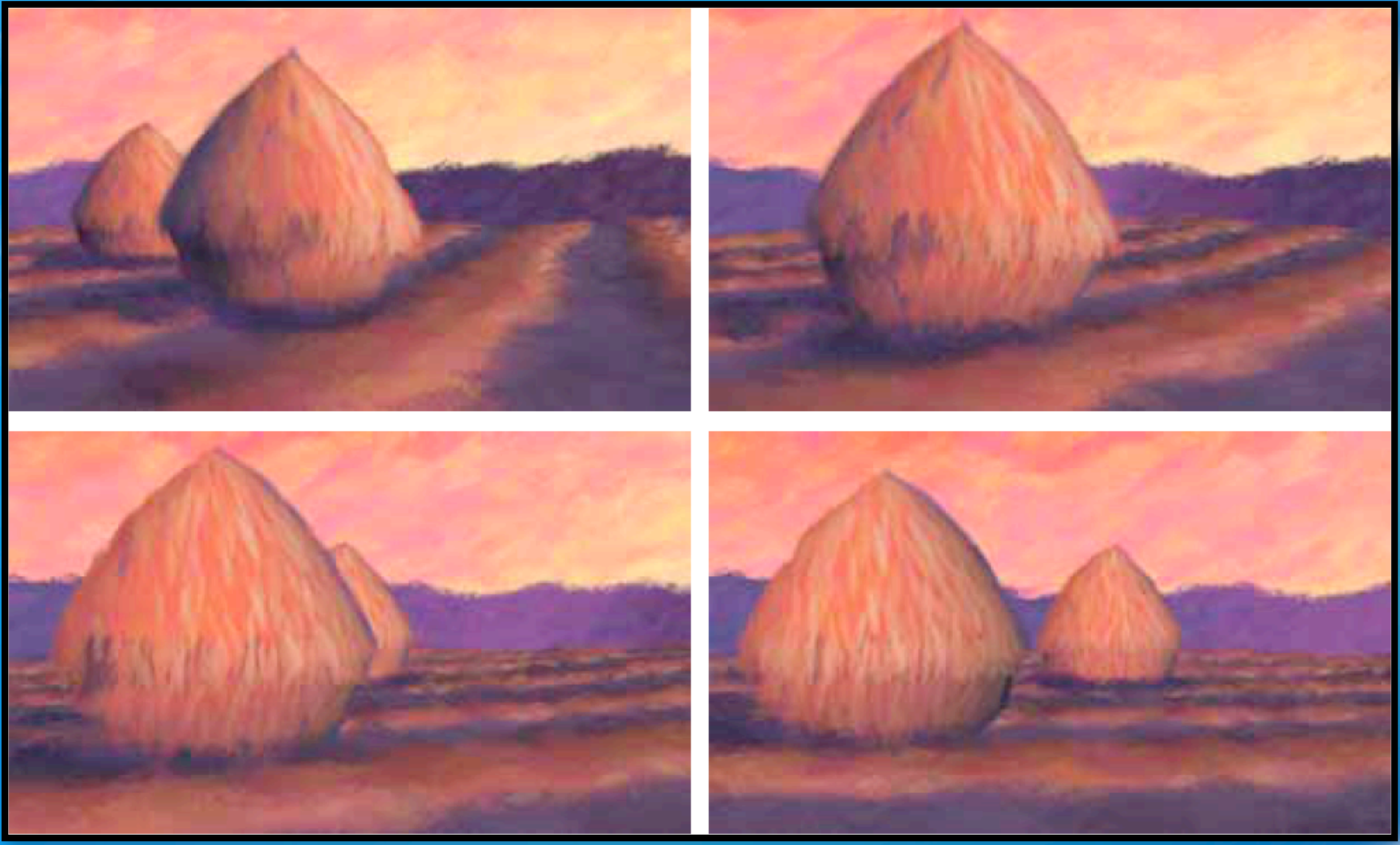
- "Painterly Rendering for Animation." Barbara Meier 96.
  - Difficulty in existing static painting NPR methods is getting the brush strokes to "stick" to surface ("shower door" effect).

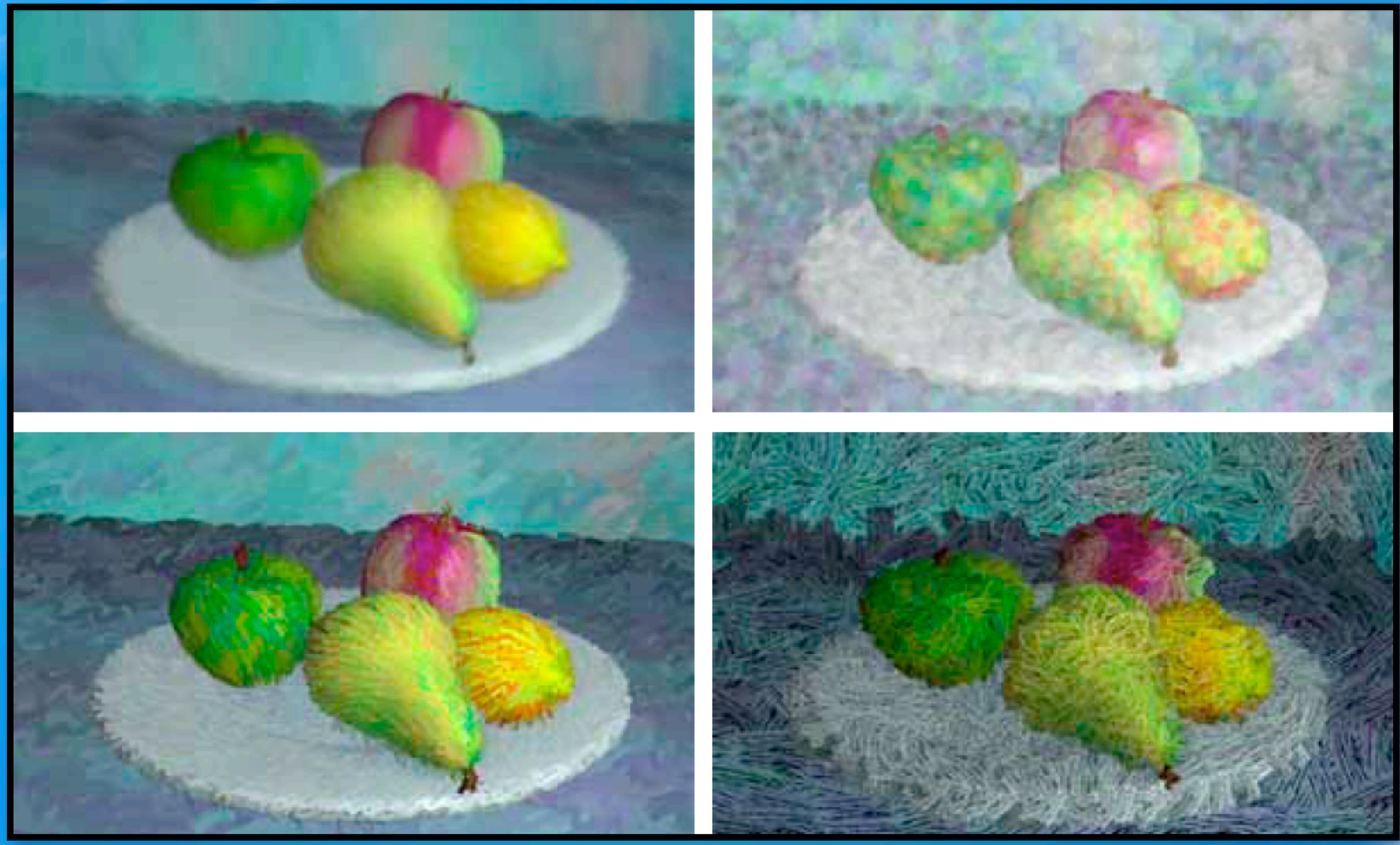
Painterly Rendering

Overview

- Model brush strokes using a particle system.
- Preprocess: distribute particles onto 3D surfaces.
  - Tessellate parametric surfaces into triangles.
    Generate random points on each triangle proportional to its area.
- Rendering: render particles as oriented textures.
  - Attributes: image, color, orientation, size, and position.


Particle Placer
Particles in World Space
Geometry
Reference Pictures
Camera Transform
Color
Shaders
Orientation
Painterly Renderer
Output Image
Size
Brush Image

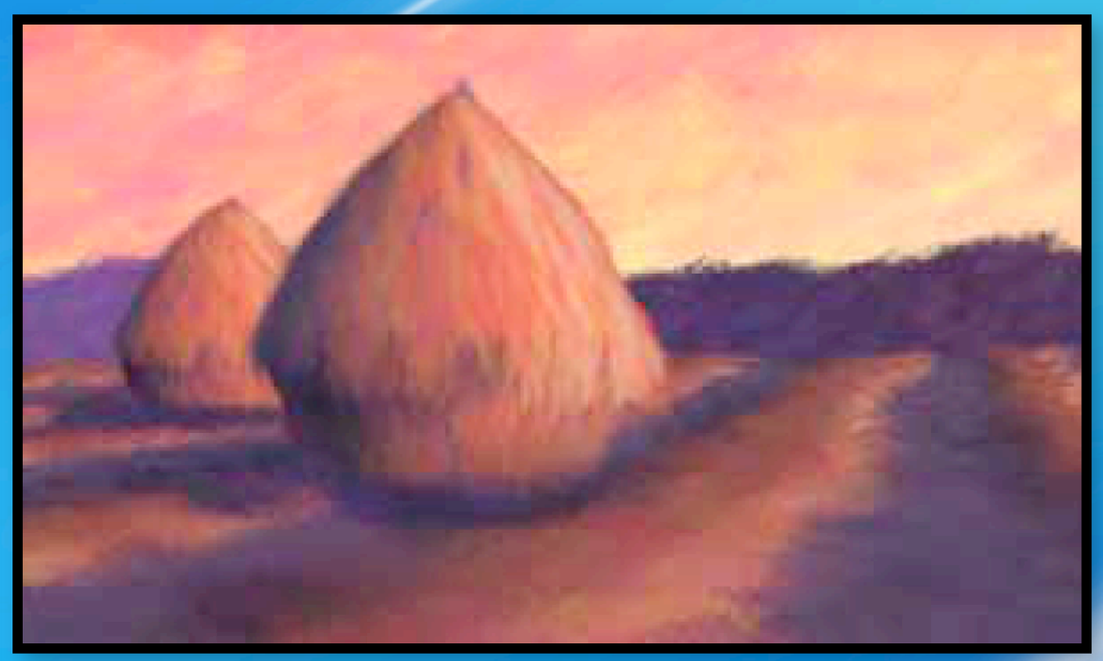




Painterly Rendering

Extras

- Animated parameters
- Adding randomness



Painterly Rendering

Problems

- Brush textures have limited resolution.
- Zooming in towards objects makes brush strokes larger and pixelated.

Graftals

Simulating Dr. Seuss

- "Art-Based Rendering of Fur, Grass, and Trees." Kowalski et al. 99.
- "Interactive Artistic Rendering." Kaplan et a. 00
- Humans can draw a teddy bear or a grassy field quickly with a few well-placed strokes.
- These exact same objects are among the hardest to model with traditional CG techniques.
- Solution: use simple geometry, and procedurally generate brush stroke textures/geometry.

Kowalski et al.

Overview

- Graftals: introduce by Alvy Ray Smith [Smith 84], as recursively defined L-systems.
- Kowalski et al. place graftal textures (actually geometry) on the image to simulate brush strokes.
- Uses a hybrid screen-space/world-space technique (DIA) to achieve proper density and frame-to-frame coherence.

Kowalski et al.

Overview

- Placing graftals:
  - Desired qualities:
    - Graftals should be placed with controlled screen-space density.
    - Should match aesthetic requirements of particular texture, but also must “stick” to surfaces in the scene, providing interframe coherence and a sense of depth through parallax.

Kowalski et al.

Overview

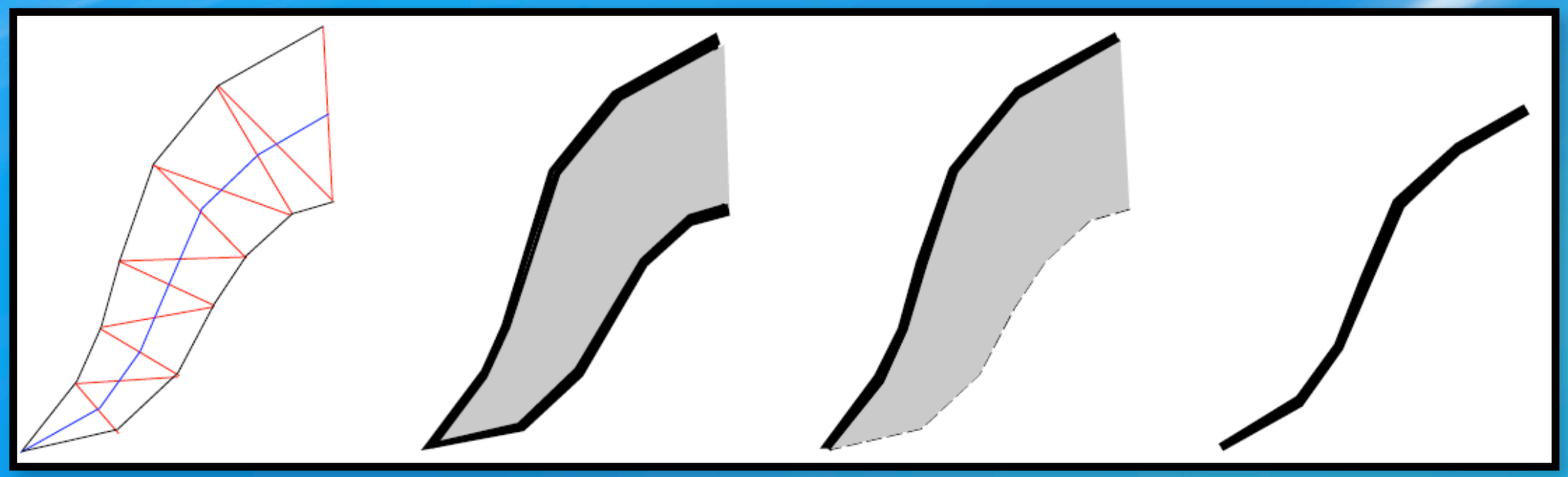
- DIA - Difference Image Algorithm
  - Originally developed by Salsbury et al. 97.
  - For each stroke drawn:
    - Subtract blurred stroke from “difference image” (initially the input image)
    - Next output stroke placed by searching in the resulting image for pixel most in need of darkening, and initiating a stroke there.

Kowalski et al.

Overview

- Handling animation with interframe coherence.
  - First frame: graftals are placed according to the DIA.
  - Successive frames: first attempt to place the graftals from the preceding frame and accept or reject base on density.
  - After all “old” graftals added, add in more graftals where needed using DIA.

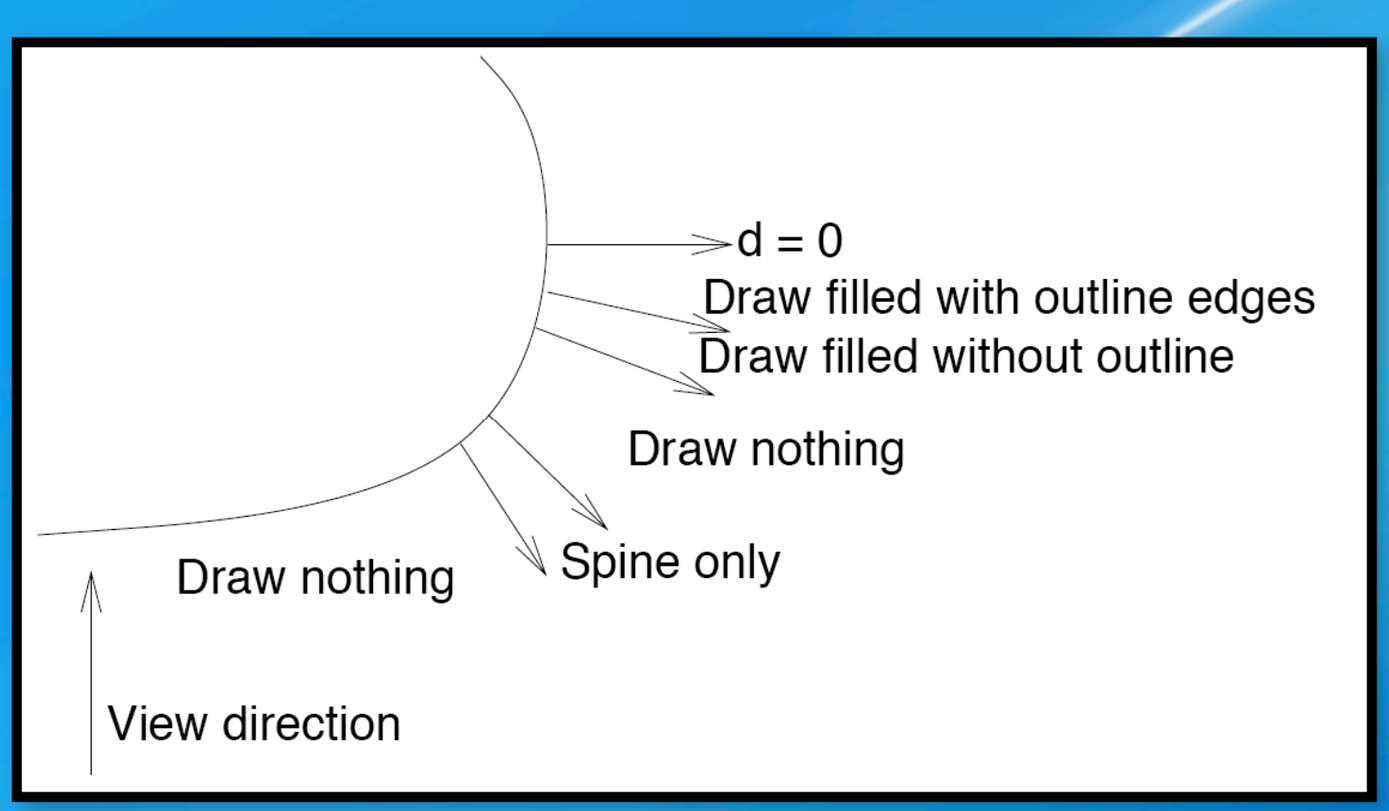
A Fur Graftal

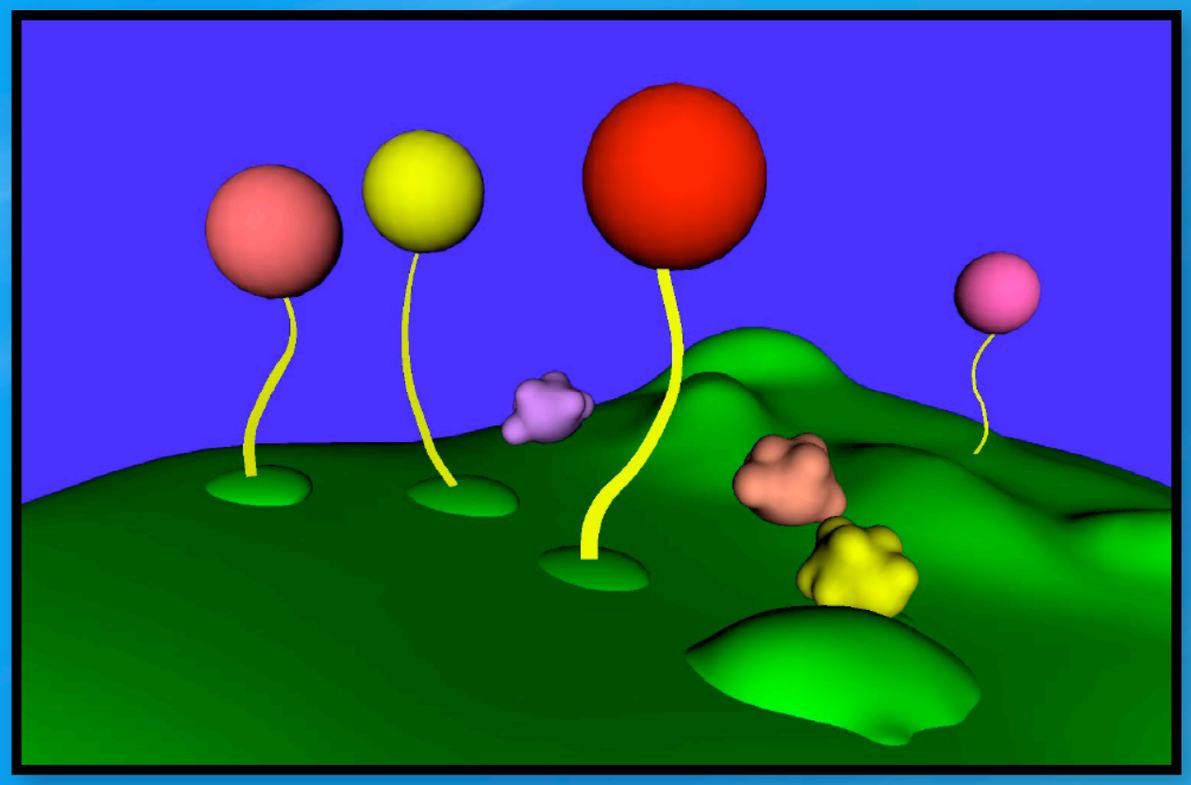


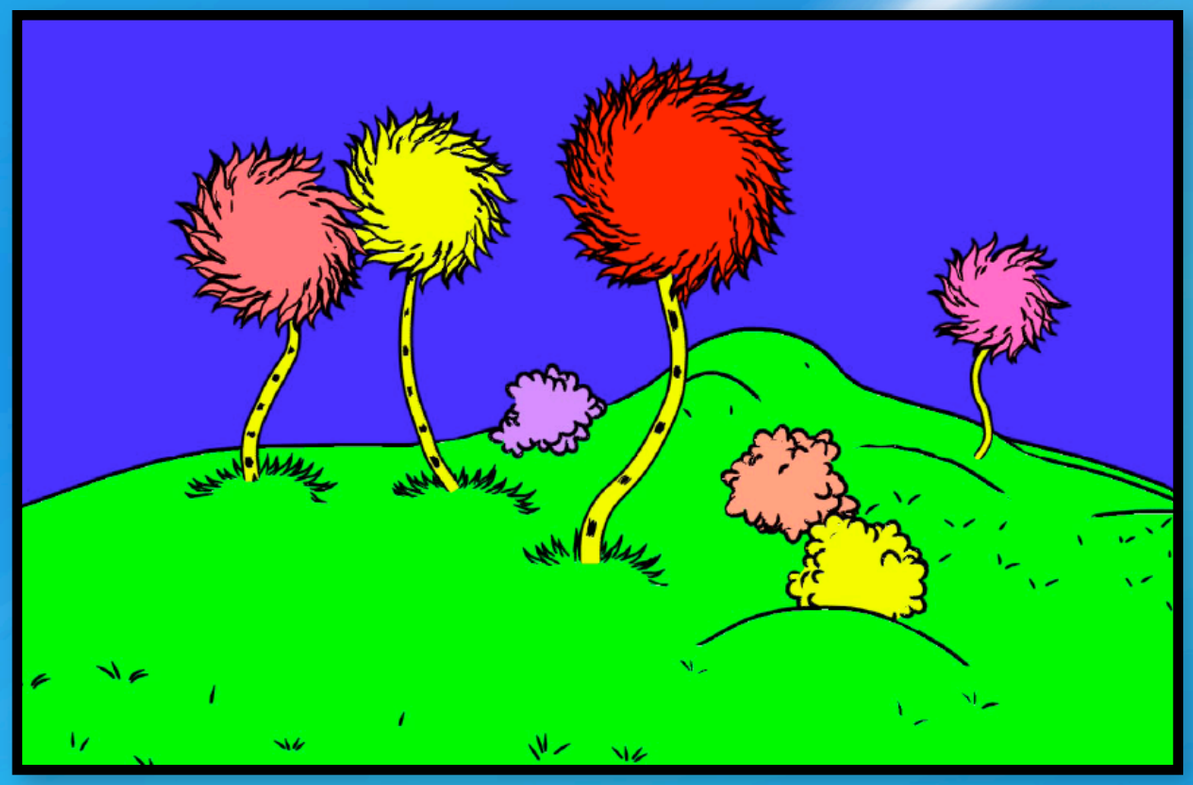
Kowalski et al.

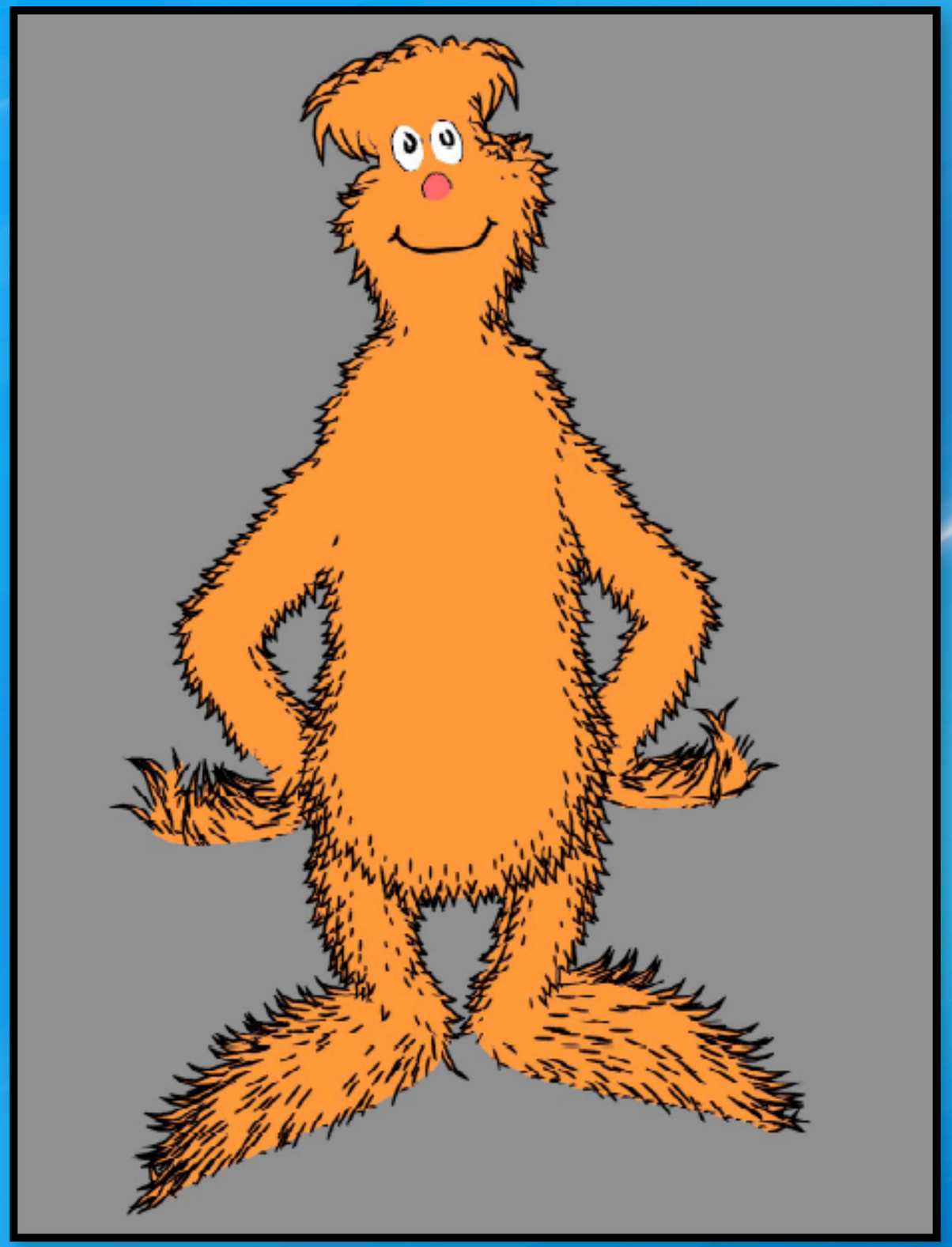
Overview

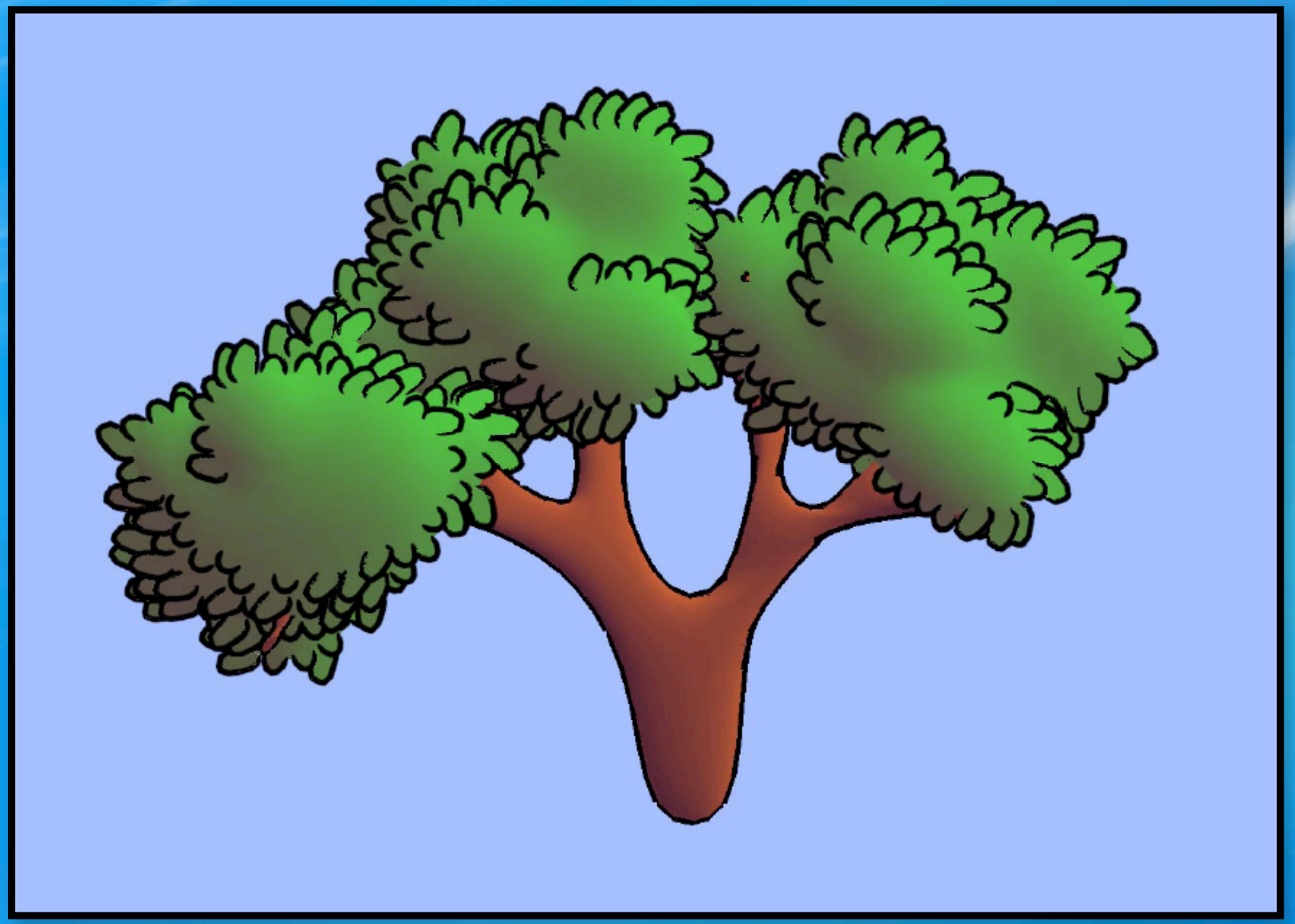
- After being placed with the DIA, each fur graftal determines how to orient and draw itself by computing the dot product $d = \mathbf{N} \cdot \mathbf{V}$.

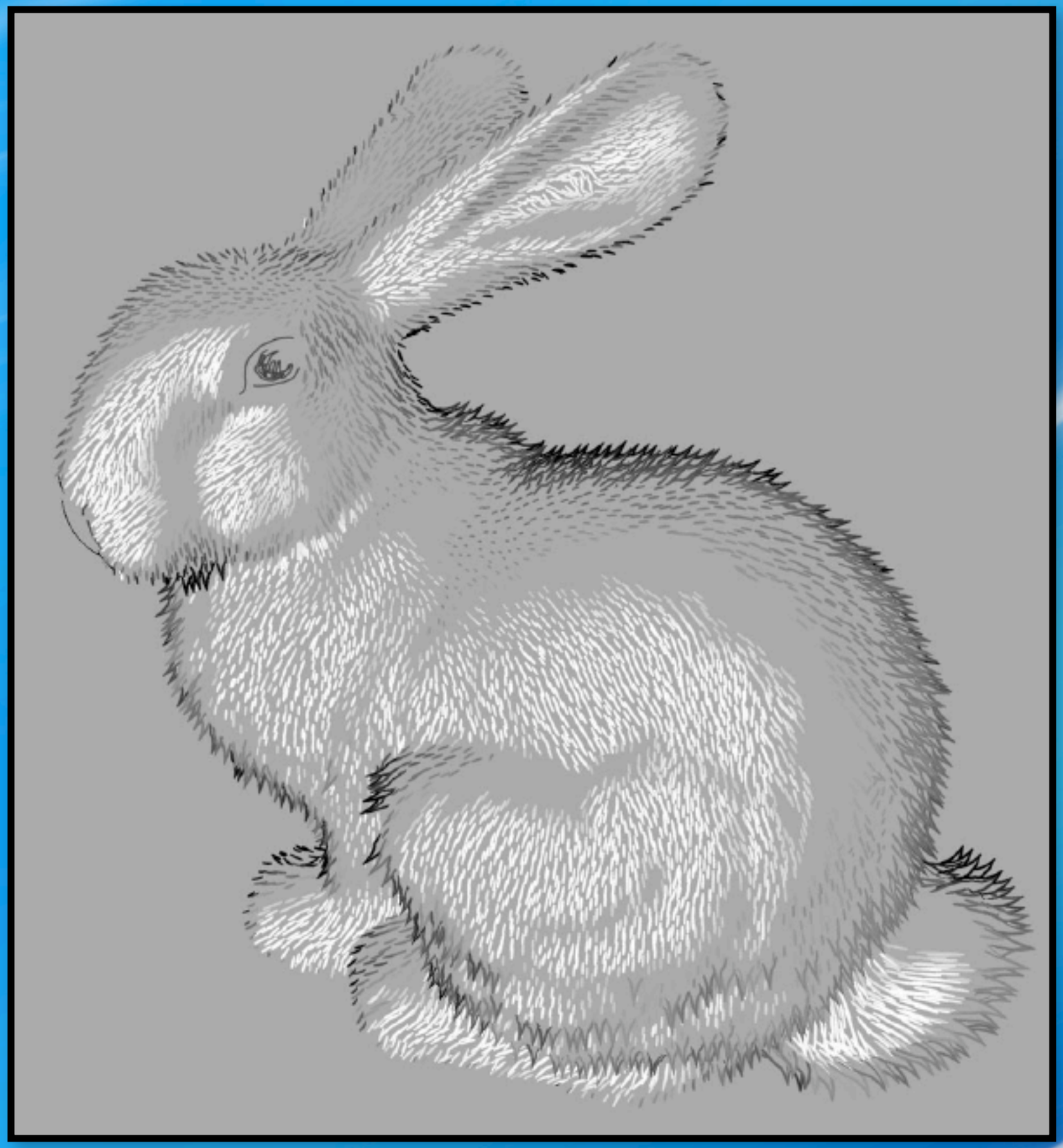


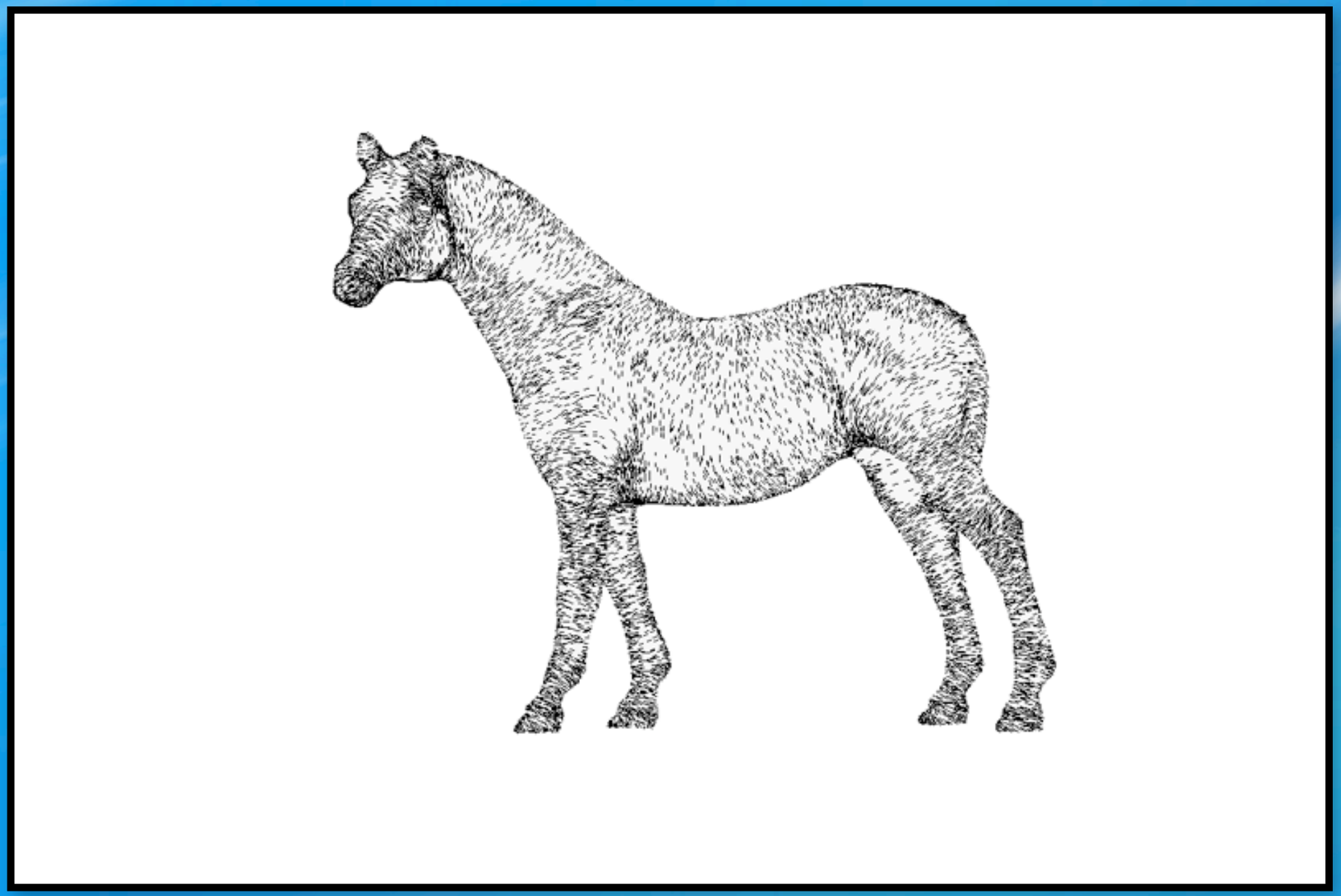








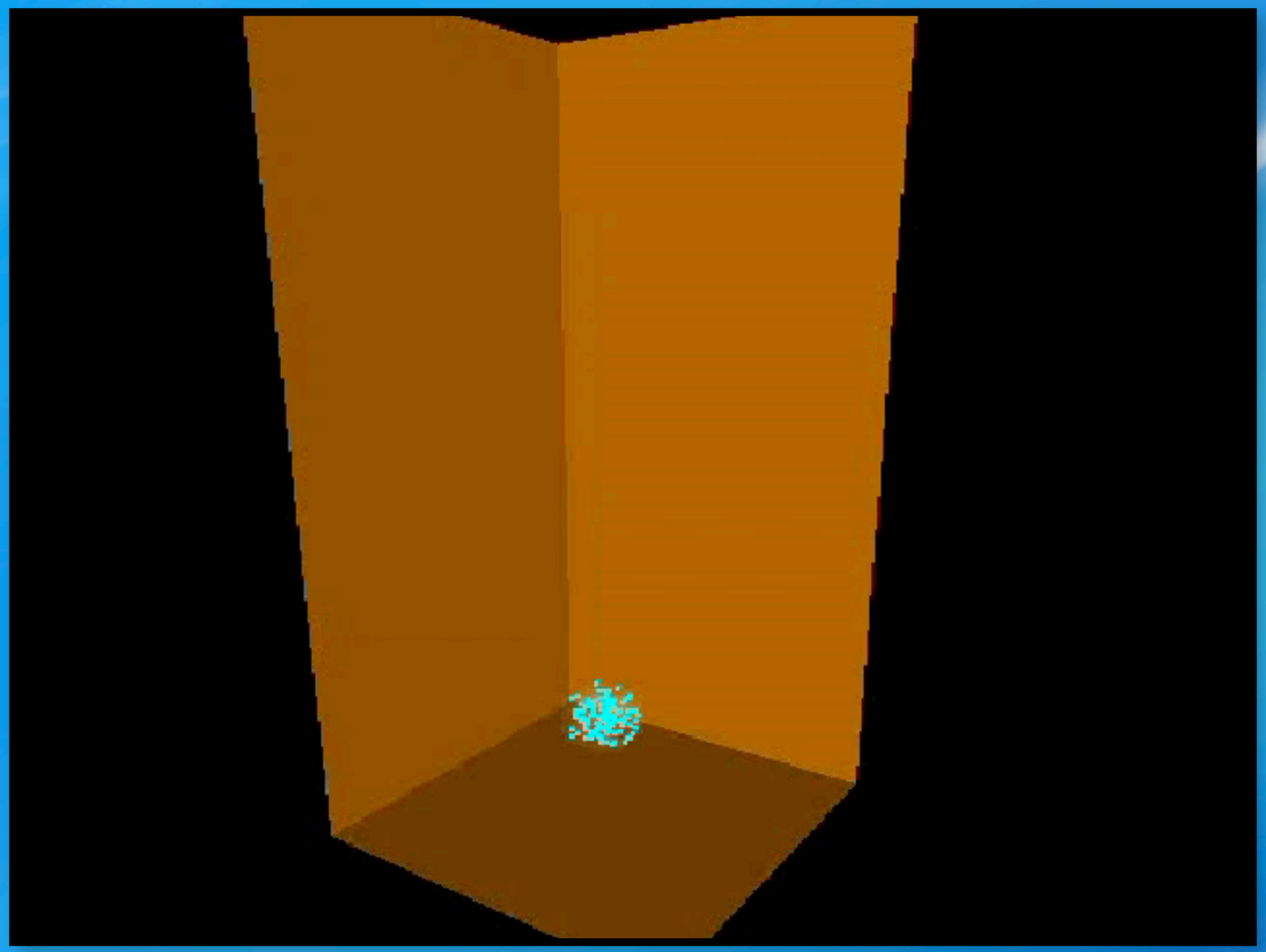




~ “Line Direction Matters: An Argument for the Use of Principal Directions in 3D Line Drawings.” Girshick, Interrante, Haker, Lemoine

NPR Smoke

Alex Mohr and Andrew Selle



Next time

- Rendering Competition!