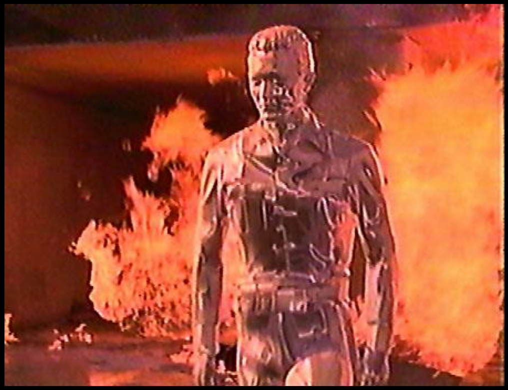
Non-Photorealistic Rendering

Computer Graphics:

What impresses people most is the photorealistic quality of the images



15

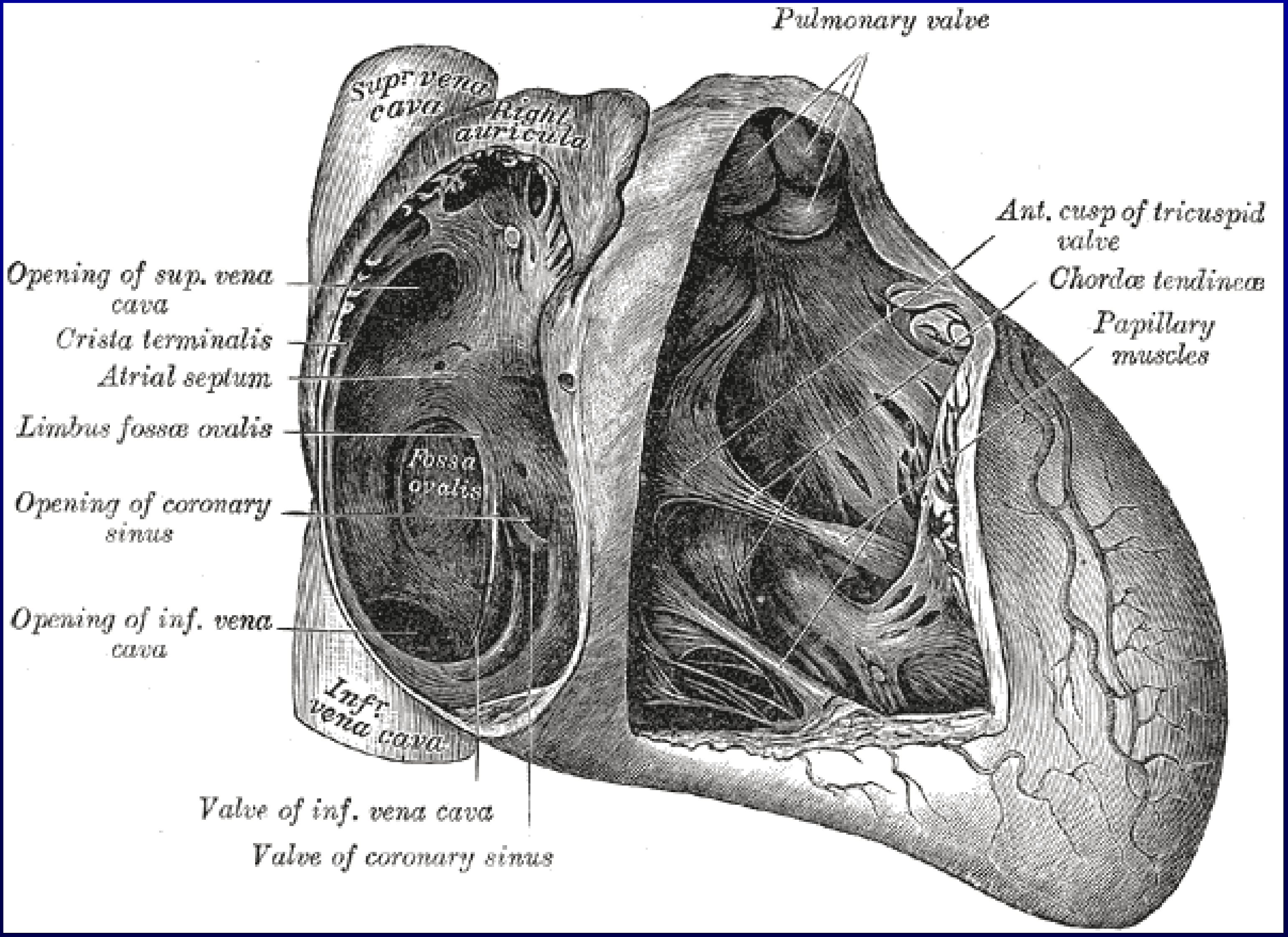
Photorealism

- Driving force behind computer graphics for the past 50 years
- Quality of image judged by how closely they resemble a photograph
- Images are rendered by running a physics-simulation which emulates the behavior of light inside the modeled scene

Effects needed for realism

- Shadows
- Reflections (Mirrors)
- Transparency
- Interreflections
- Detail (Textures etc.)
- Complex Illumination
- Realistic Materials
- And many more

What's Wrong with Photorealism?



Computers: Process, Transform, *Communicate*

Information Theory

Origin (where does information come from?)
Throughput (how frequent?)
Latency (how long do I have to wait)?
Presentation (what does it look like?)

Computers Graphics:
technology for presenting information

Information transported by an image can take many different forms

Photograph of a sailboat

Vast amount of information:

Time of day
The weather
Wind direction
Speed
Relationships between boat and environment
etc…

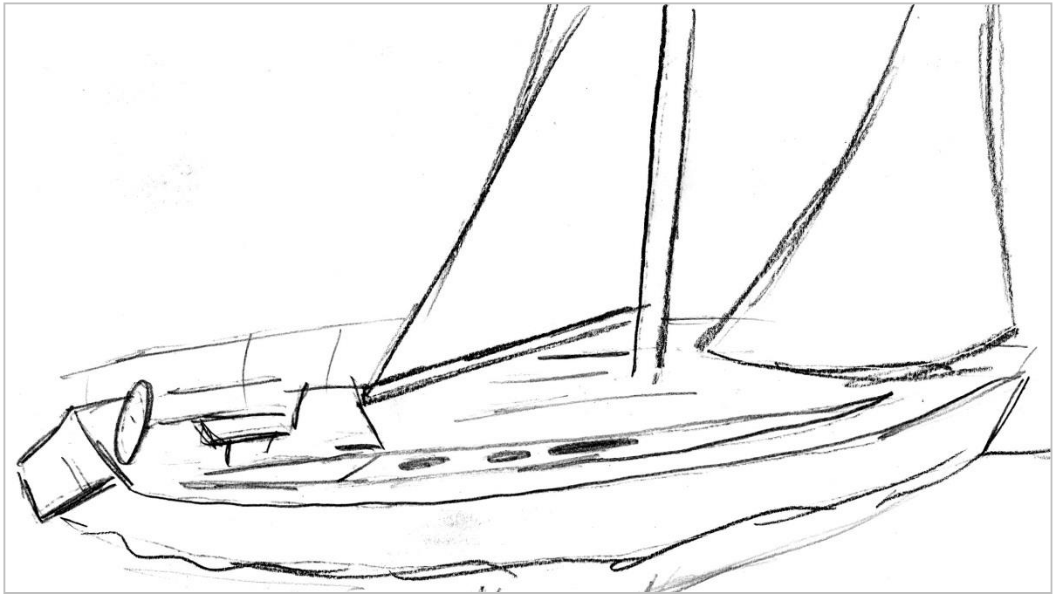
HOWEVER…

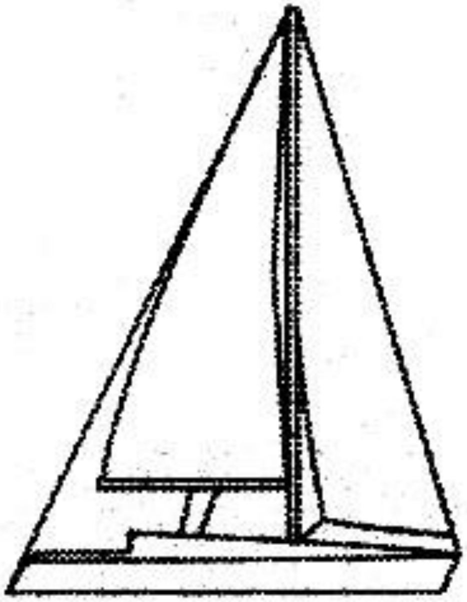
⇩

Little use to someone attempting to build a sailboat!

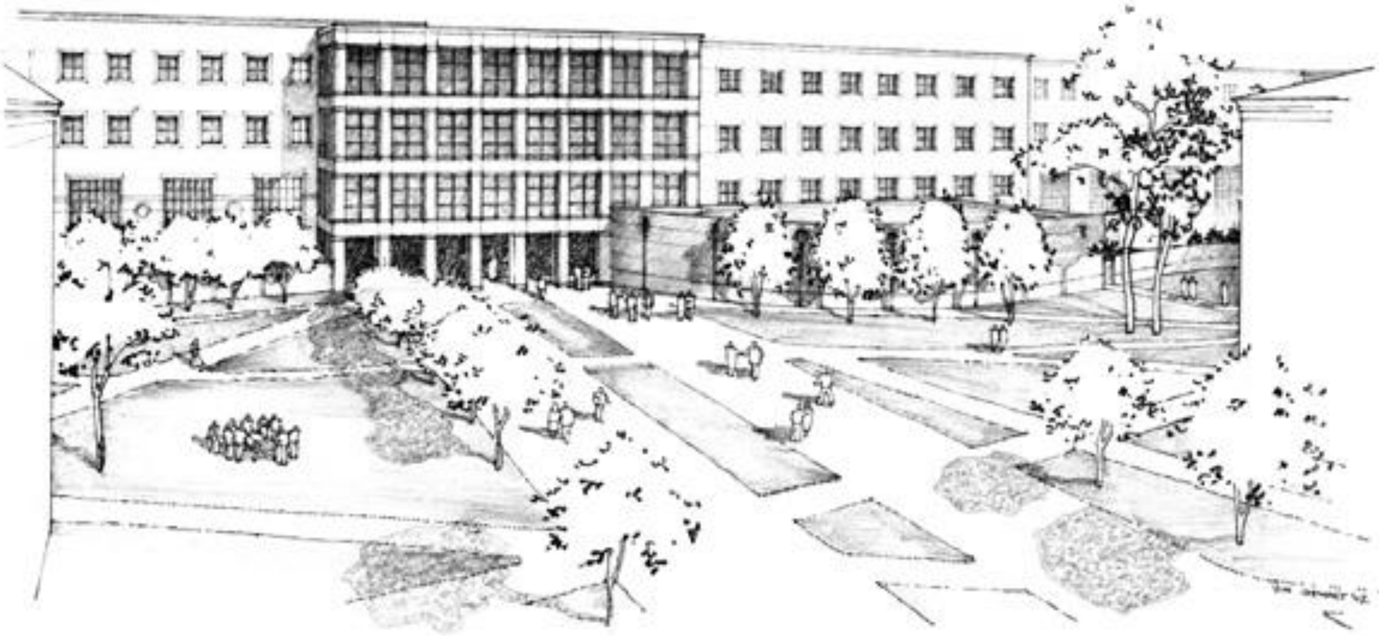
⇩

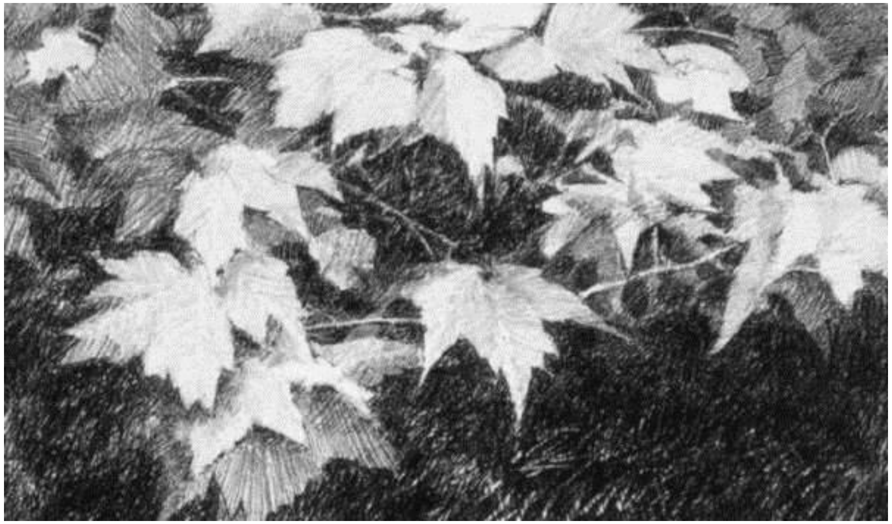
A sailboat builder would certainly prefer:
Technical drawings, blueprints
Line drawing representations of boat parts
Sketches of design ideas

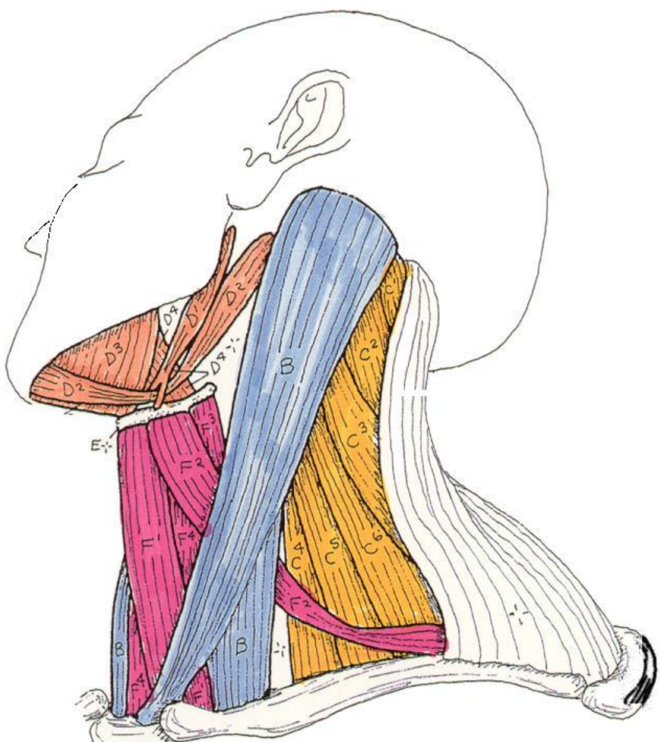


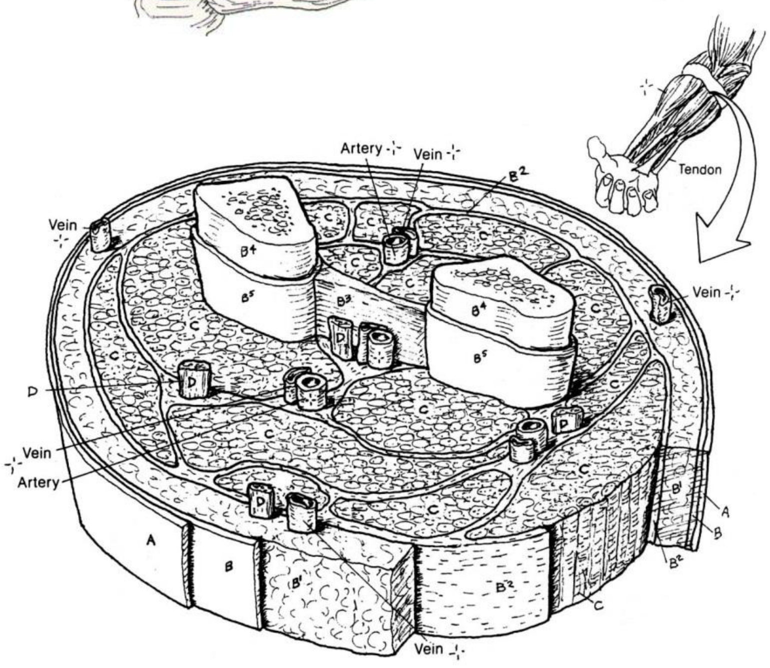



Megayacht Features
226 sq. ft. Double Salon
Fantastic panoramic views from the main salon
Automatic safe companion way door
Flush fitted Bimini

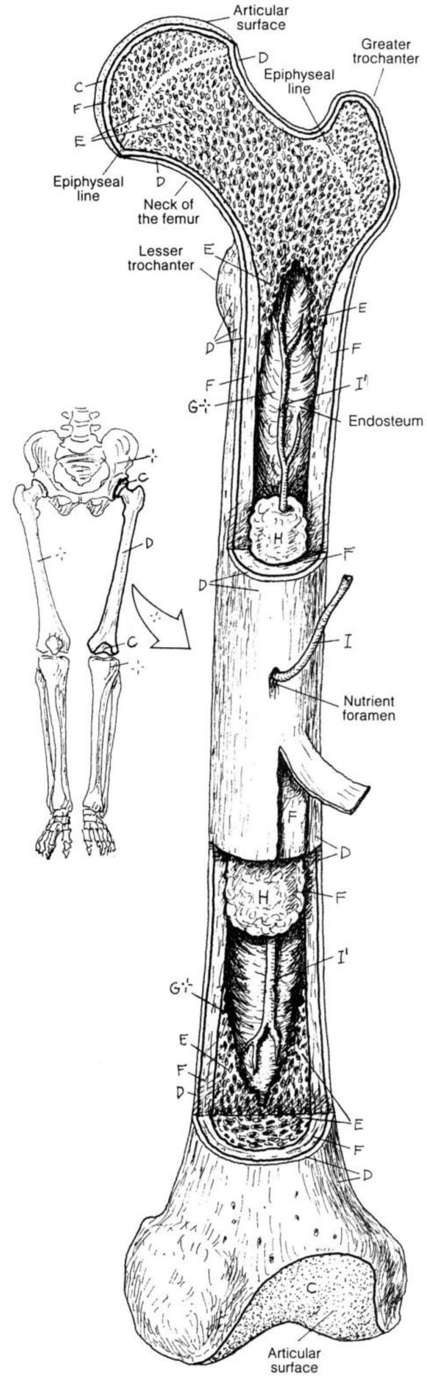




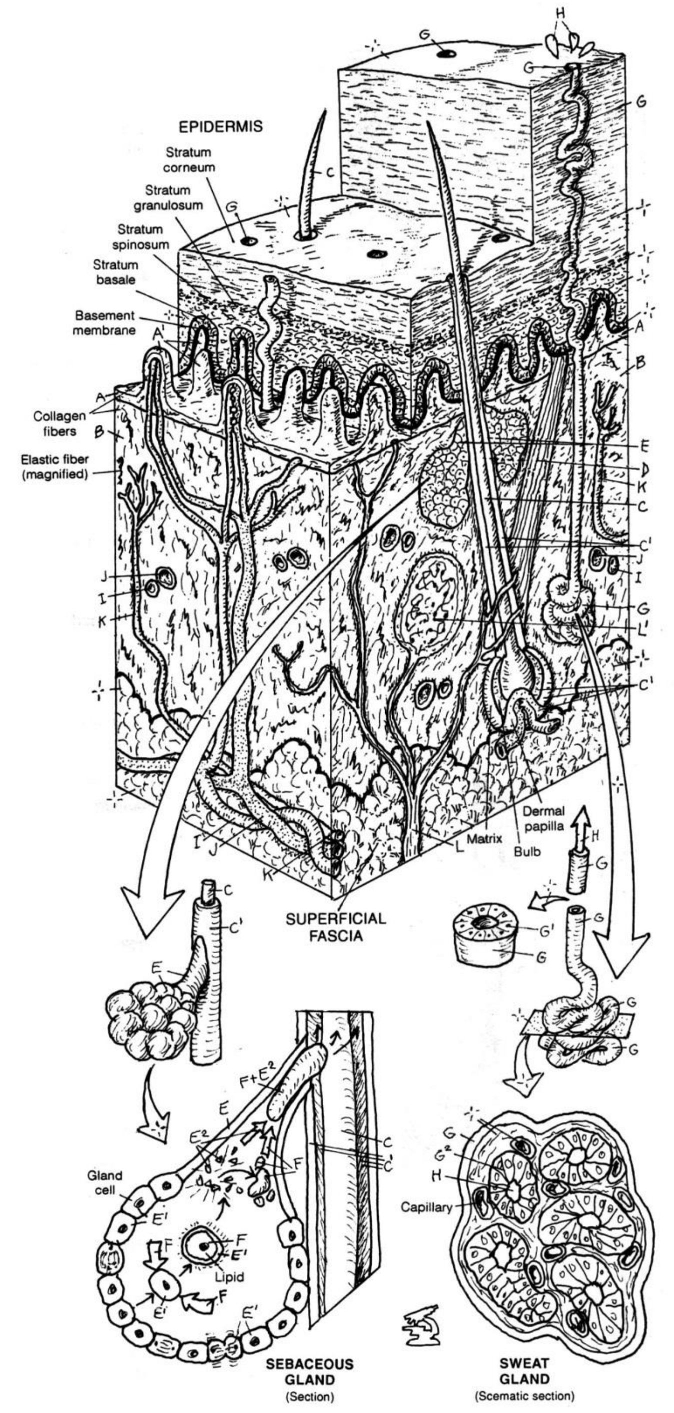





Tendon
Artery
Vein
Vein
Vein
Vein
Artery
Vein



Articular surface
Greater trochanter
Epiphyseal line
Epiphyseal line
Neck of the femur
Lesser trochanter
Endosteum
Nutrient foramen
Articular surface



EPIDERMIS
Stratum corneum
Stratum granulosum
Stratum spinosum
Stratum basale
Basement membrane
Collagen fibers
Elastic fiber (magnified)
Matrix
Bulb
Dermal papilla
SUPERFICIAL FASCIA
Gland cell
Lipid
SEBACEOUS GLAND
(Section)
Capillary
SWEAT GLAND
(Scematic section)

Non-Photorealistic Rendering

- **Departs from the limits of photorealism to better communicate visual information**
- **Uses concepts from art instead of physics**
- **Two fundamental visual cues**
  - *Silhouette – the visible edges of a surface*
  - *Hatching – the use of texture to indicate the local orientation (shading) of a surface*

Traditionally…

Imagery generated by illustrators (artistic, technical, scientific) has been used to provide information that may not be readily apparent in photographs or real life.

Traditionally…

Imagery generated by illustrators (artistic, technical, scientific) has been used to provide information that may not be readily apparent in photographs or real life.

Non-Photorealistic Rendering (NPR):

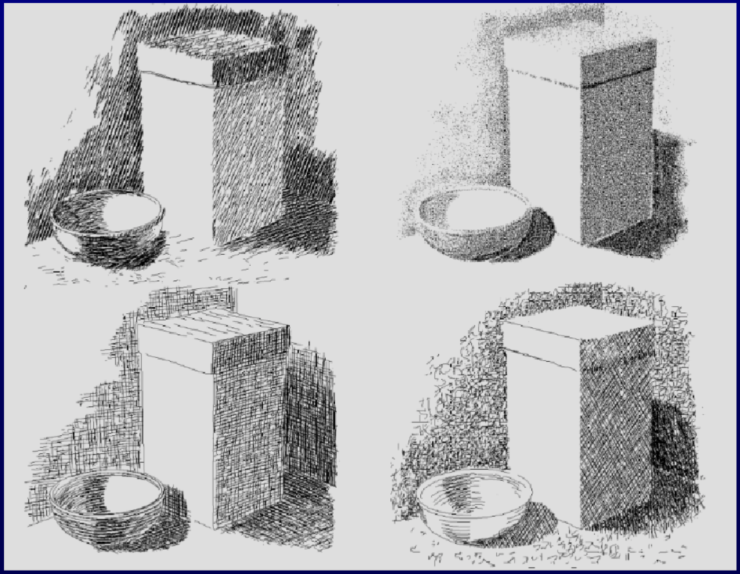
- Apply a similar goal to computer generated-images
- New field of research in computer graphics

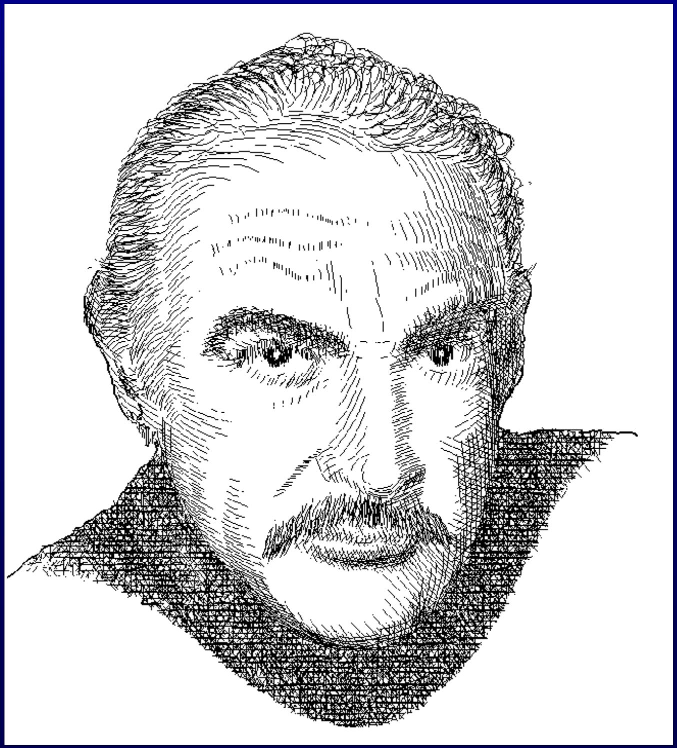
Non-Photorealistic Rendering (NPR)

- Images are judged by how effectively they communicate
- Involves stylization and communication, usually driven by human perception
- Knowledge and techniques long used by artists and illustrators
- Emphasis on specific features of a scene, expose subtle attributes, omit extraneous information
- Brings together art and science

Non-Photorealistic Rendering (NPR)

- Alternate display models
- Increase expressive power of Computer Graphics
- Many fields
- Computer Graphics revisited





Recipe for Image Generation

Computer Science

Math

Linear Algebra *(Matrices, Vectors)*

Discrete representations *(Images)*

Physics

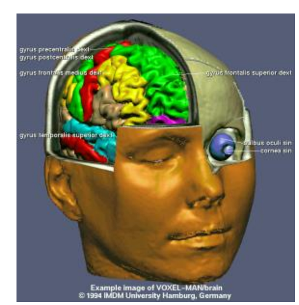
Lighting/Energy transfer

Color reflections/refraction

Mechanical Engineering

CAD/CAM

Solid Models



Recipe for Image Generation

Art and Illustration

Non-Photorealistic Rendering

Computer Science

Math

Linear Algebra *(Matrices, Vectors)*

Discrete representations *(Images)*

Physics

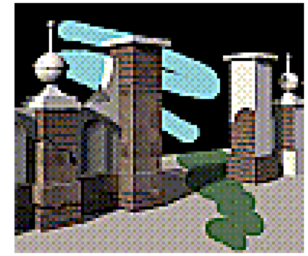
Lighting/Energy transfer

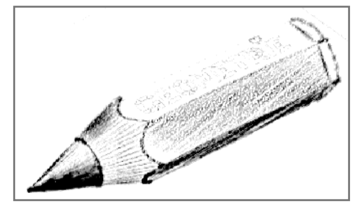
Color reflections/refraction

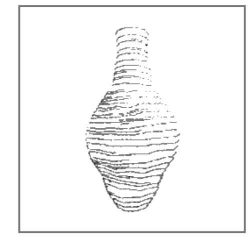
Mechanical Engineering

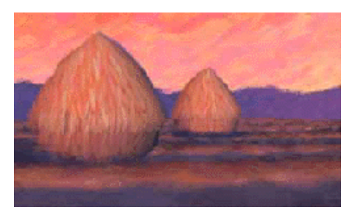
CAD/CAM

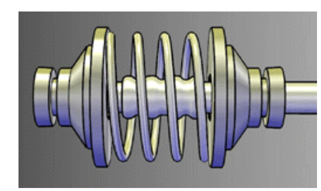
Solid Models











Applications → NPR == CG&Viz!

- **Art, Entertainment, Publishing**
  - *Movie Production, Animation, Special Effects*
  - *Computer Games*
  - *World Wide Web*
  - *Slide, Book, Magazine, Multimedia Design*

- **Science and Engineering**
  - *Computer-Aided Design (CAD)*
  - *Simulations*
  - *Scientific Analysis and Visualization*
  - *Process Control and Monitoring*

Traditional Art & Illustration

- **Art, Entertainment, Publishing**
  - *Artists, Designers, illustrators*
- **Science and Engineering**
  - *Technical Illustrators*
  - *Scientific Illustrators or Natural Science Illustrators (medicine, botany, archeology, etc)*

Art, Entertainment, Publishing :

Movie Production, Animation, Special Effects

Tools for artists, painting systems, pre-production

What Dreams May Come
PB14 Final

"What Dreams May Come"
First motion picture with full NPR effects

"Tarzan"
"Star Wars II"
"Matrix II"
…

Art, Entertainment, Publishing :

Computer Games, Interactive Theater

- Tools for artists
- Painting systems
- Pre-production
- Stylized output
- Real-time NPR
- …

Art, Entertainment, Publishing :

WWW, Slide, Book, Magazine, Interactive Multimedia Design

Science and Engineering:

Computer-Aided Design (CAD) ⇨

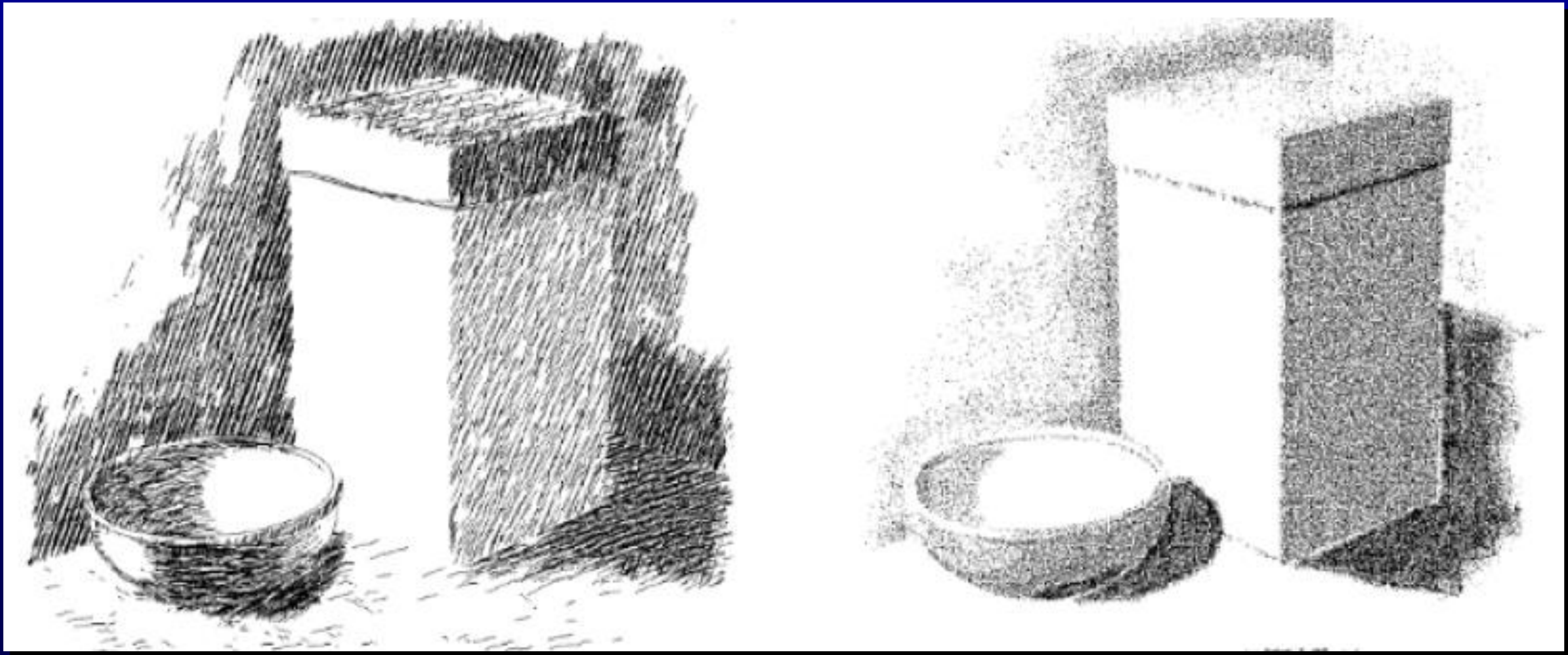
- Mechanical Engineers
- Land Developers
- Civil Engineers
- Landscape Architects
- Architects
- Civil Planners

Scientific Analysis and Visualization

Simulations ⇨

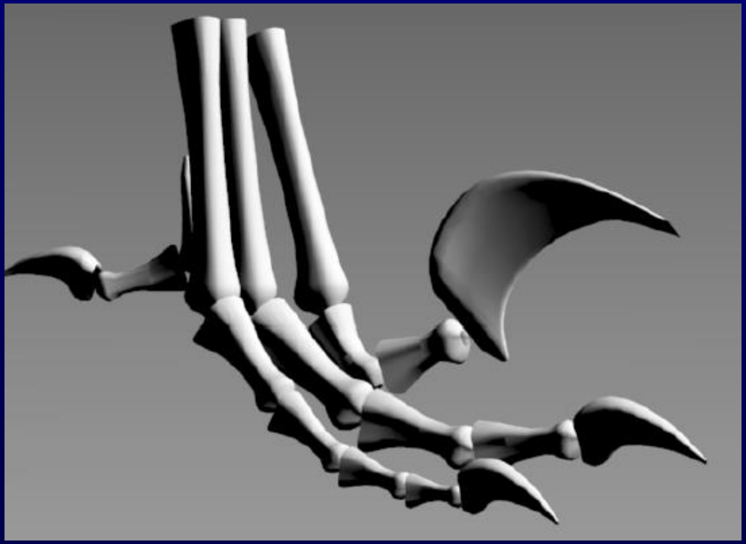
- Medicine
- Botany
- Archeology
- Paintings

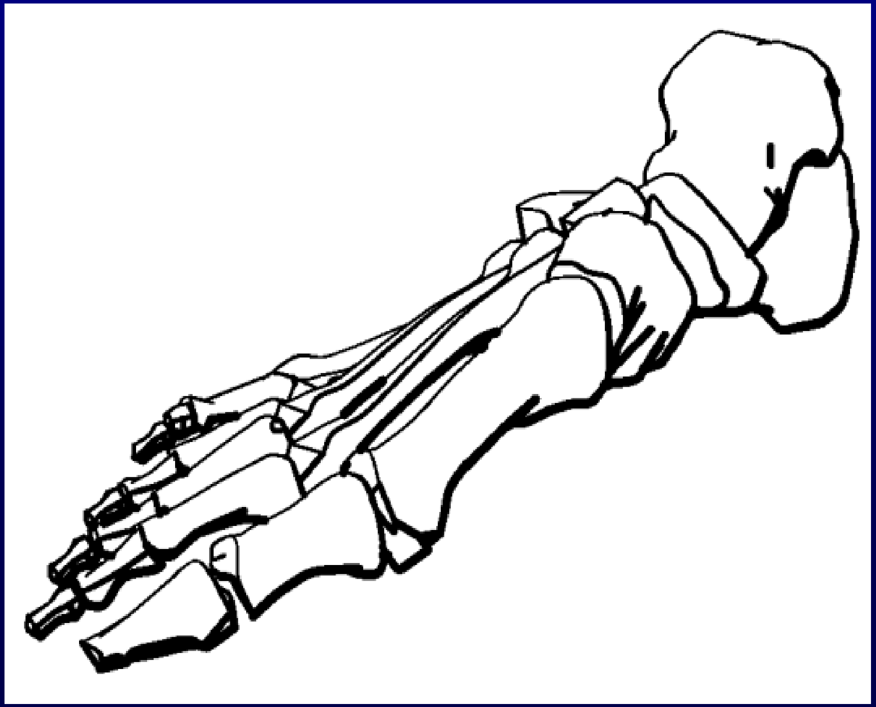
Non-Photorealistic Rendering (NPR)



Non-photorealistic Rendering

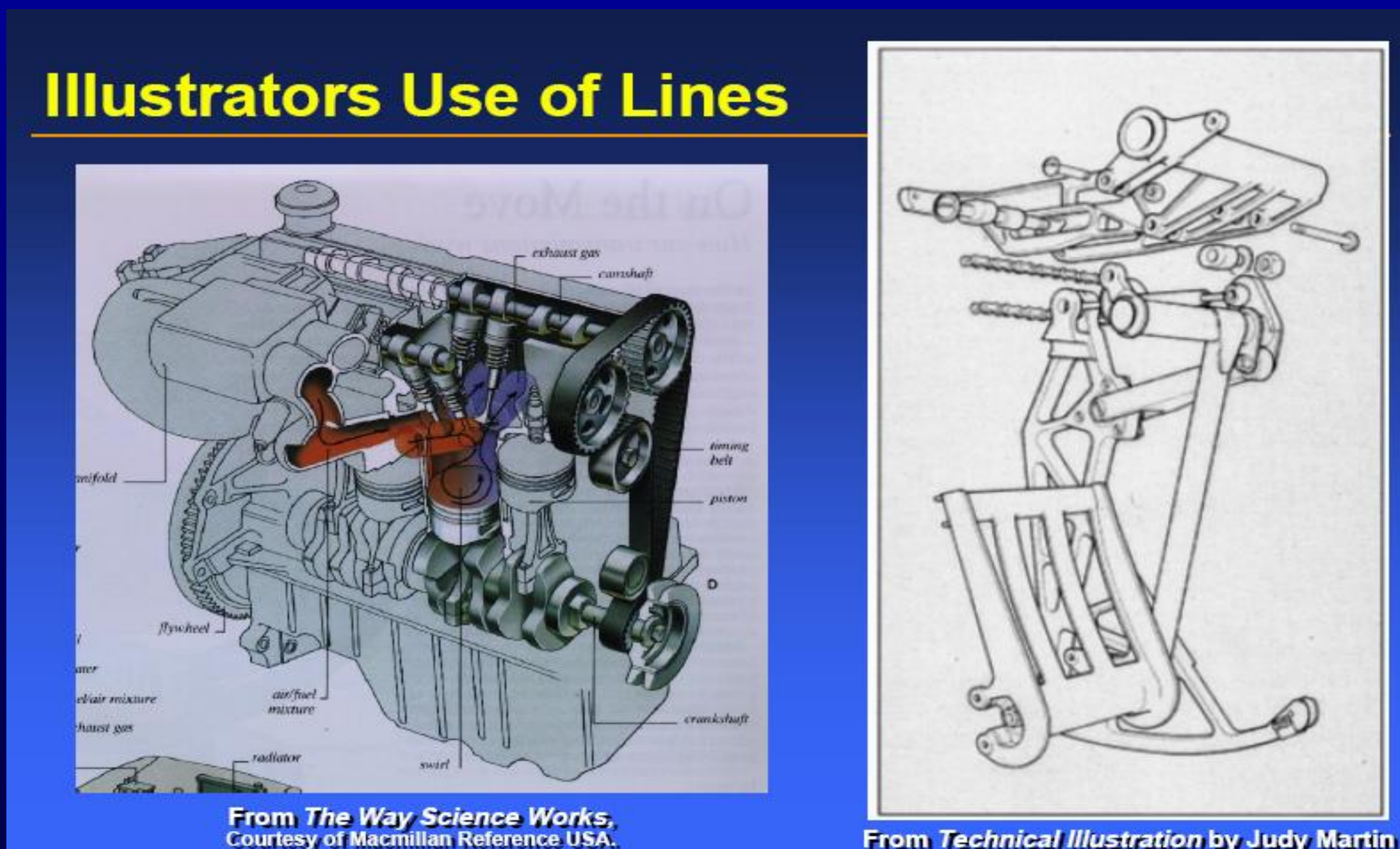
- Most computer graphics work strives for photorealism
- Other types of depiction can be more expressive or artistic





NPR – Technical Illustration

- **Illustrate important features**

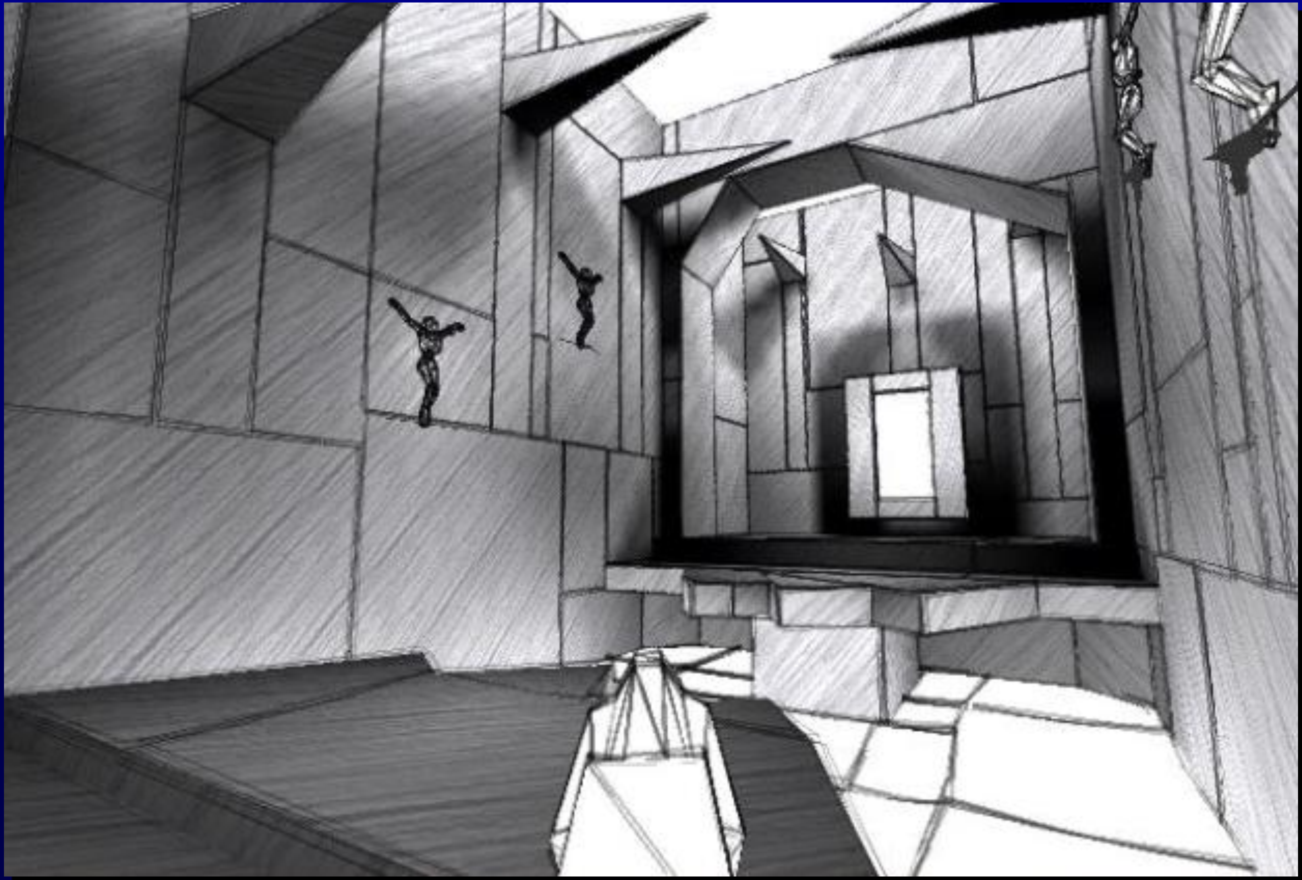


NPR – Painterly Rendering

- Make it look like being created using brush strokes and paint

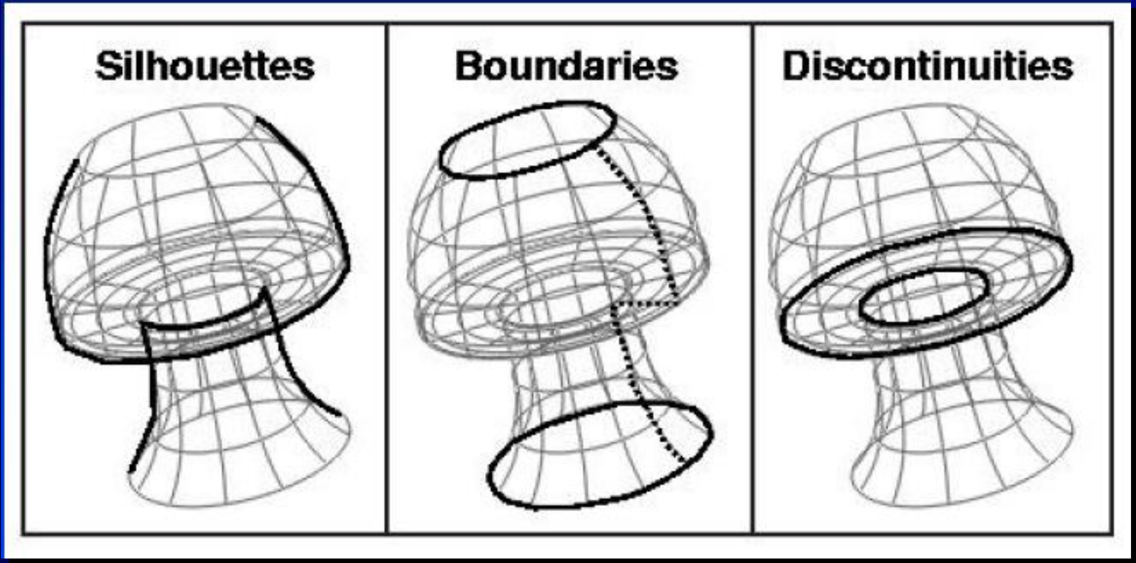
NPR – Sketchy Rendering

- Make it look like being created with pencil sketch



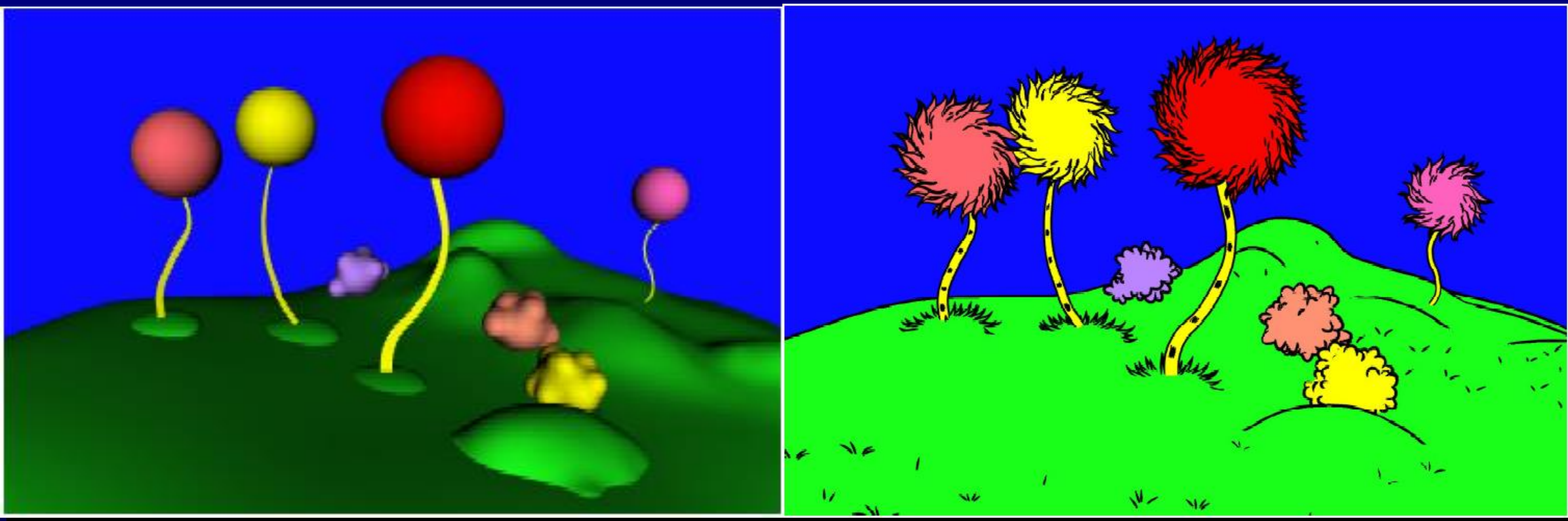
Shape Abstraction by Lines

- Boundary lines
- Silhouette lines
- Creases
- Material edges

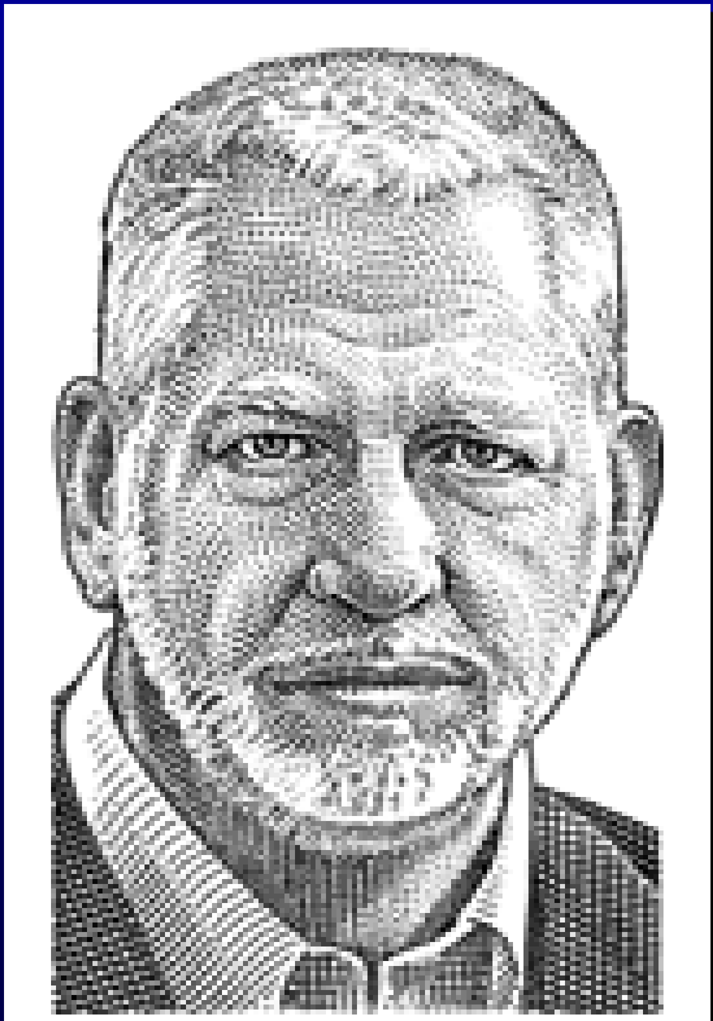


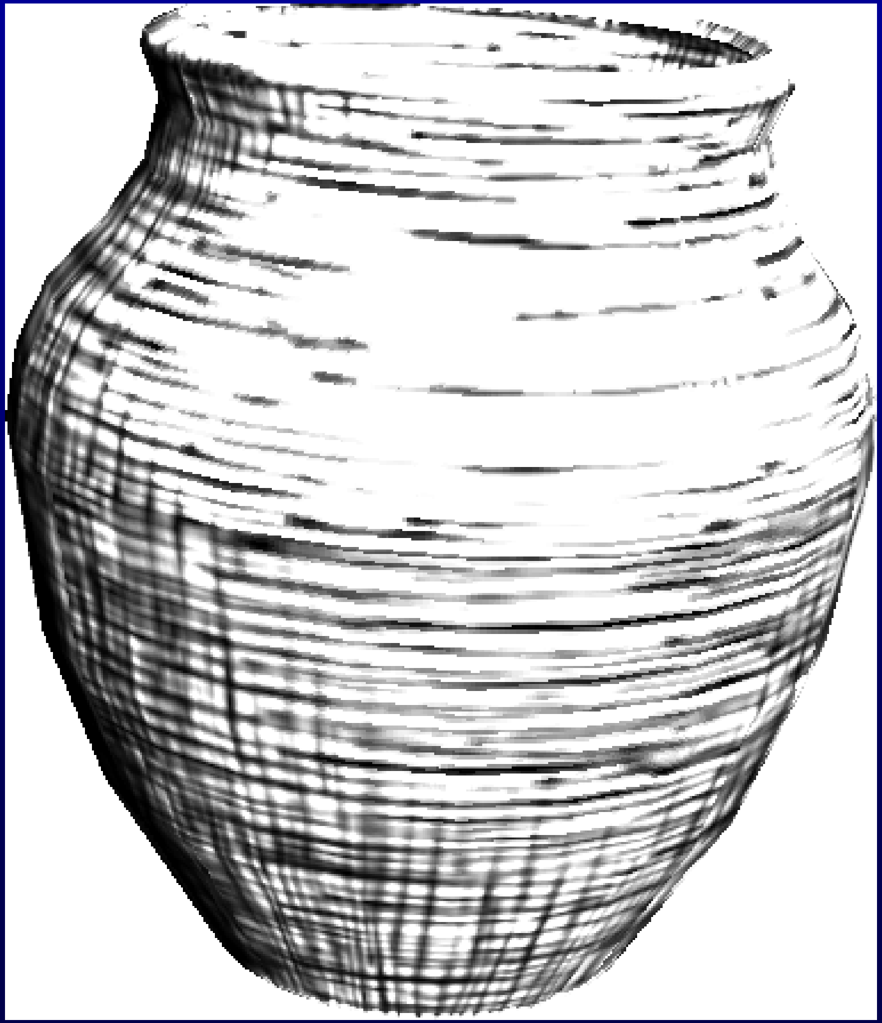
Shape Abstractions by Lines

- Various line styles can be used

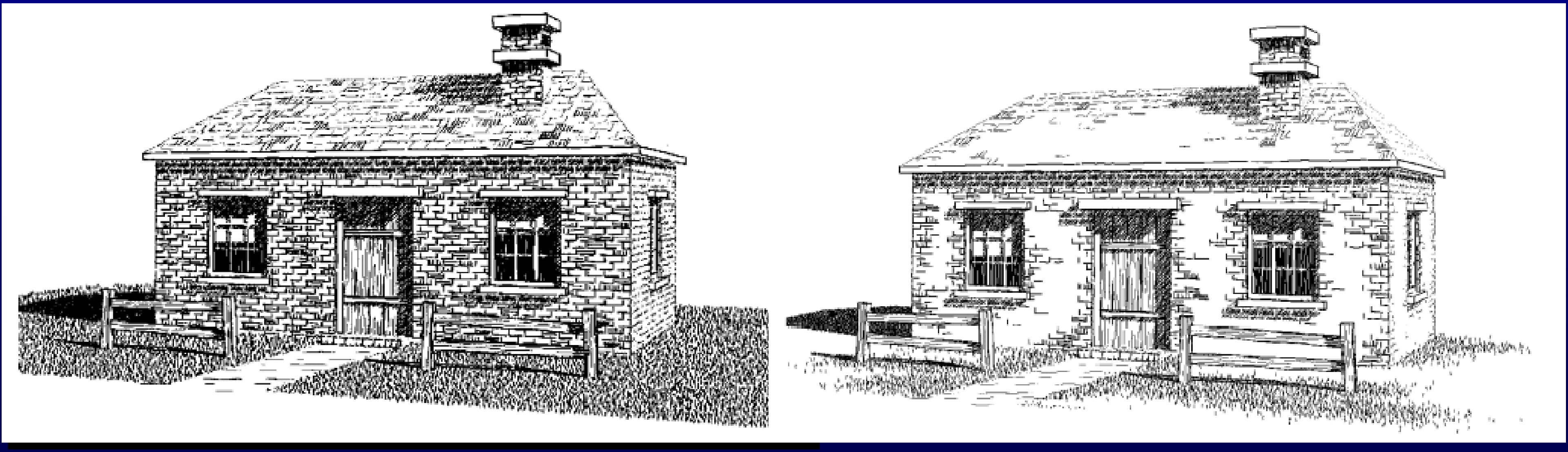


Shape Abstraction by Textures

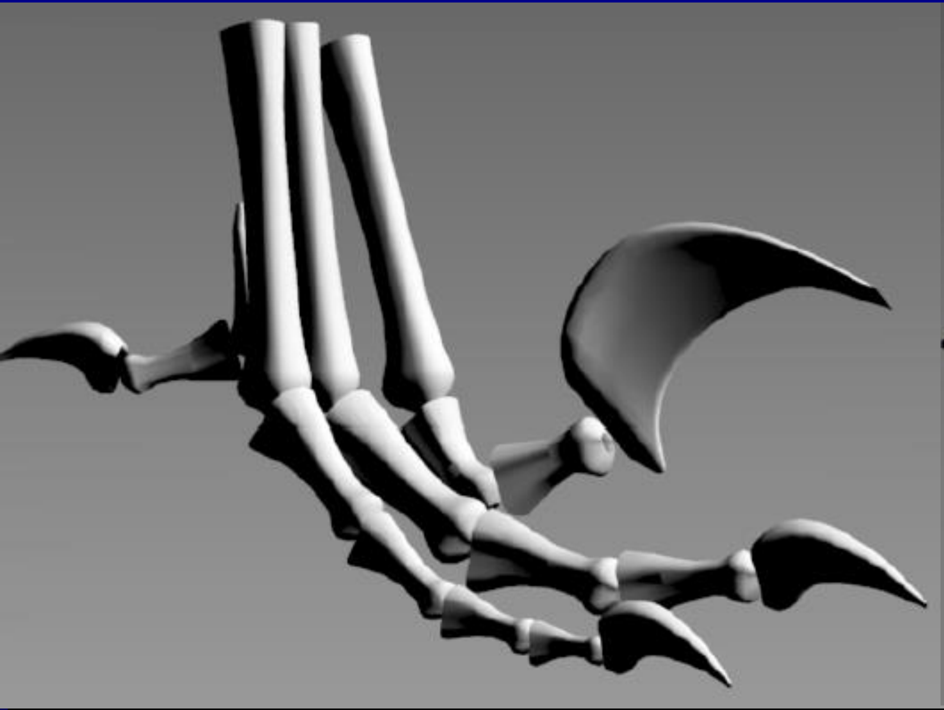




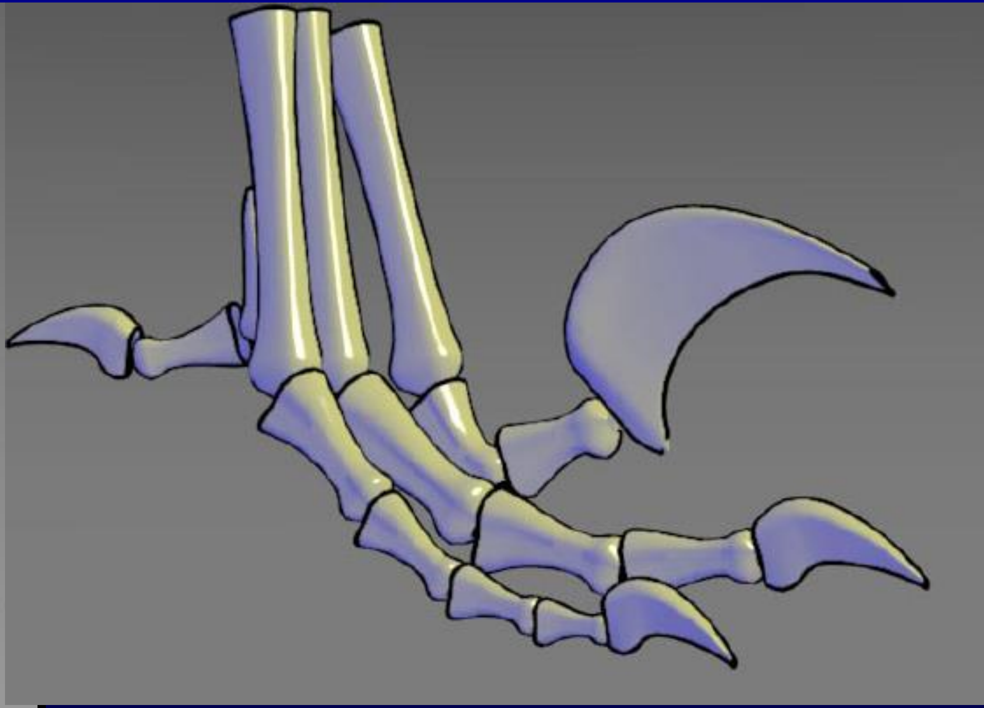
Shape Abstraction by Textures



Shape Abstraction by Shading



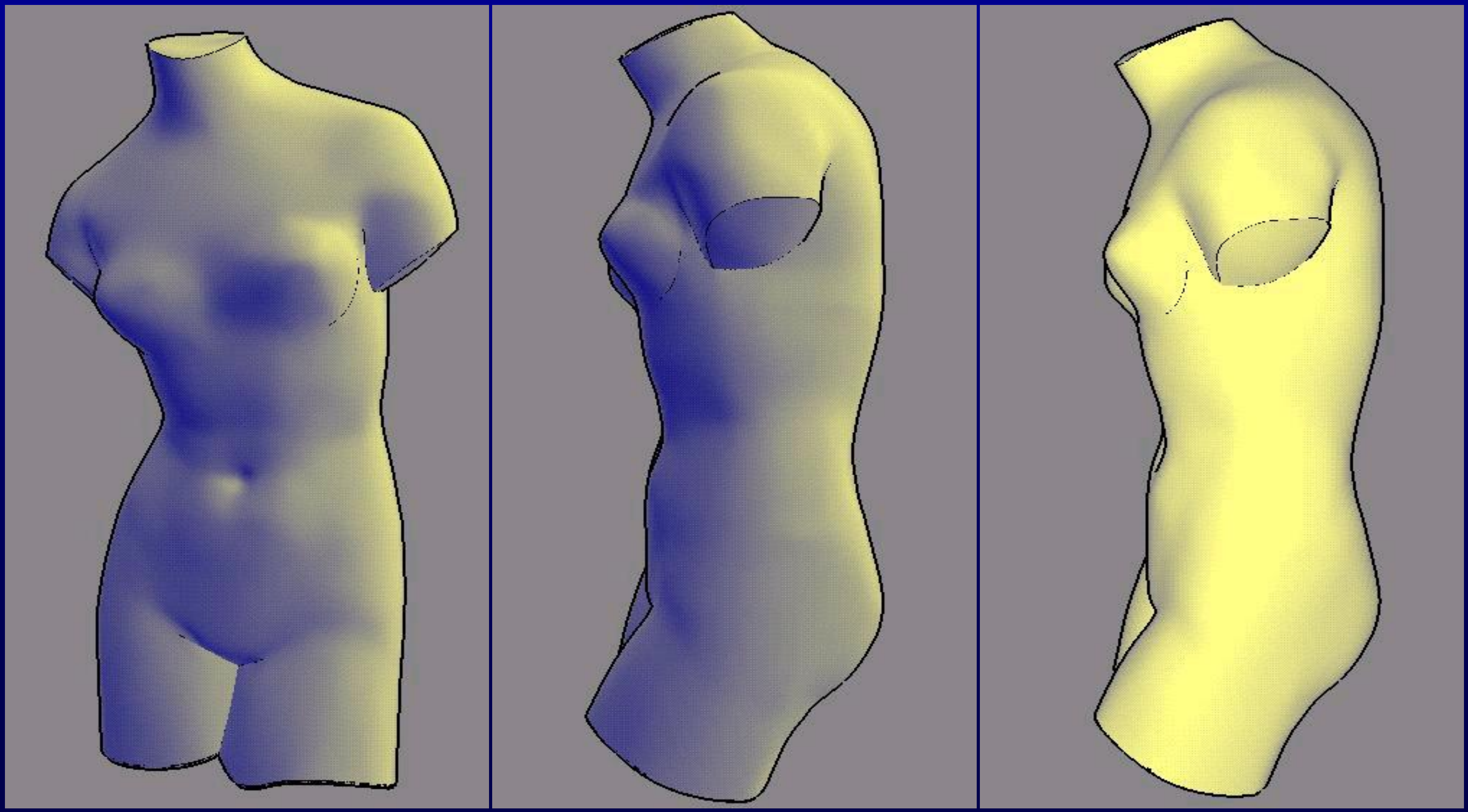
Regular OpenGL Gouraud Shading



Tone Shading

Shape Abstraction by Shading

- More effective when combined with lines



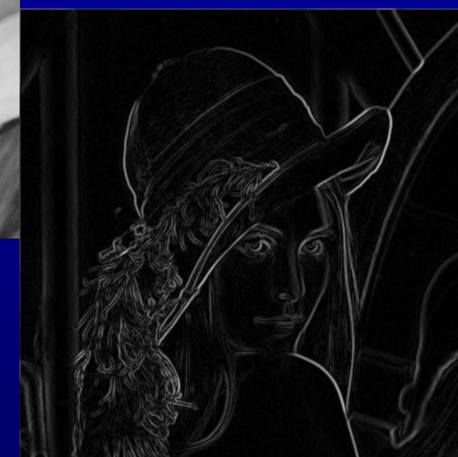
Feature Line Detection

- Image space method – analyze the rendered images
- Object space method – analyze the mesh

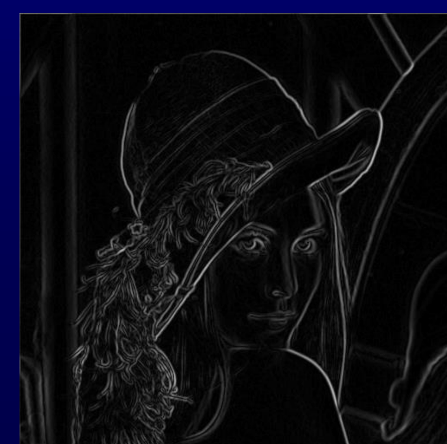
Image Processing Silhouettes

Canny edge detector

- Gaussian smoothing
- First derivative operator
- Non-maximal suppression



Produces image of silhouette edges

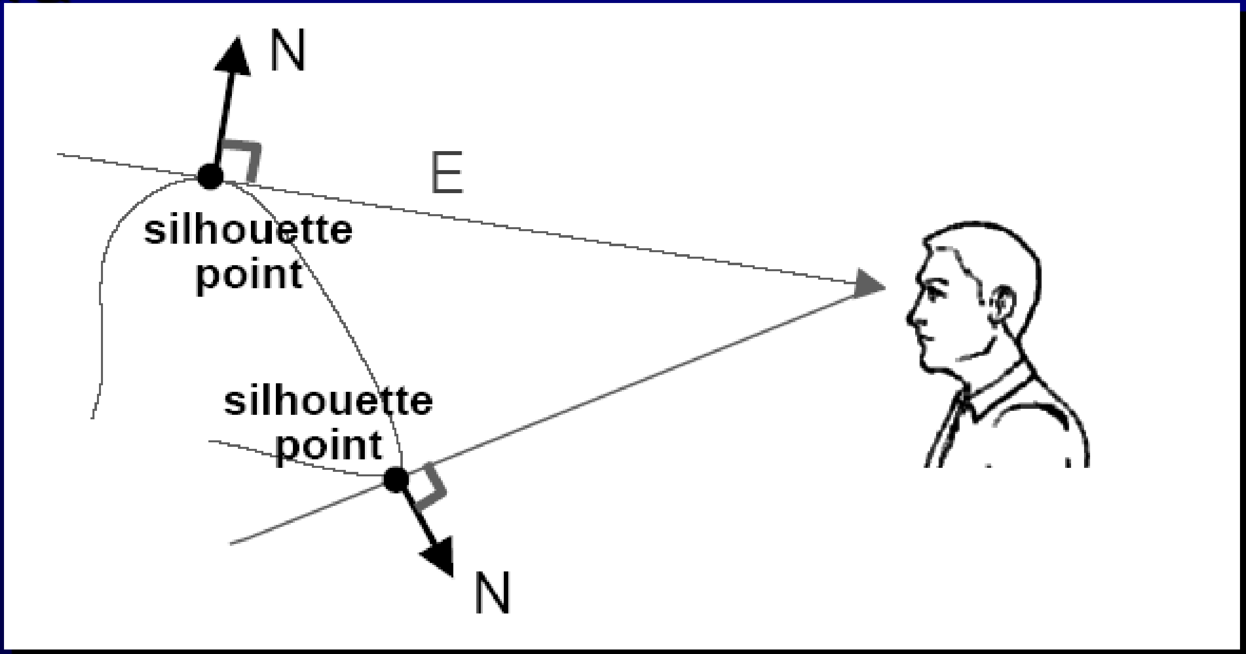


Also yields other non-silhouette edges from illumination, texture that are also useful for illustration

Silhouette

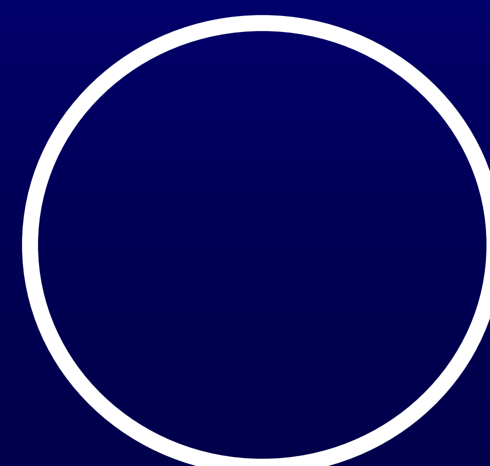
For a smooth surface, a silhouette can be defined as:

- N. (X-E) = 0 ; N: normal, X: silhouette point; E: camera



Environment Map Silhouettes

- Environment map captures visible light from a given point
- Sphere map captures visible light
- NPR environment map: white with black border



Edge Detector

- **Discontinuity in depth map or normal map can be detected using edge detector**

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Ix(x,y) = I(x,y) x Sx; Iy(x,y) = I(x,y) x Sy

IM = sqrt (Ix(x,y)^2 + Iy(x,y)^2)

Edge detection by thresholding IM

Image Space Method

- Analyze the depth buffer – look for depth discontinuity using edge detector

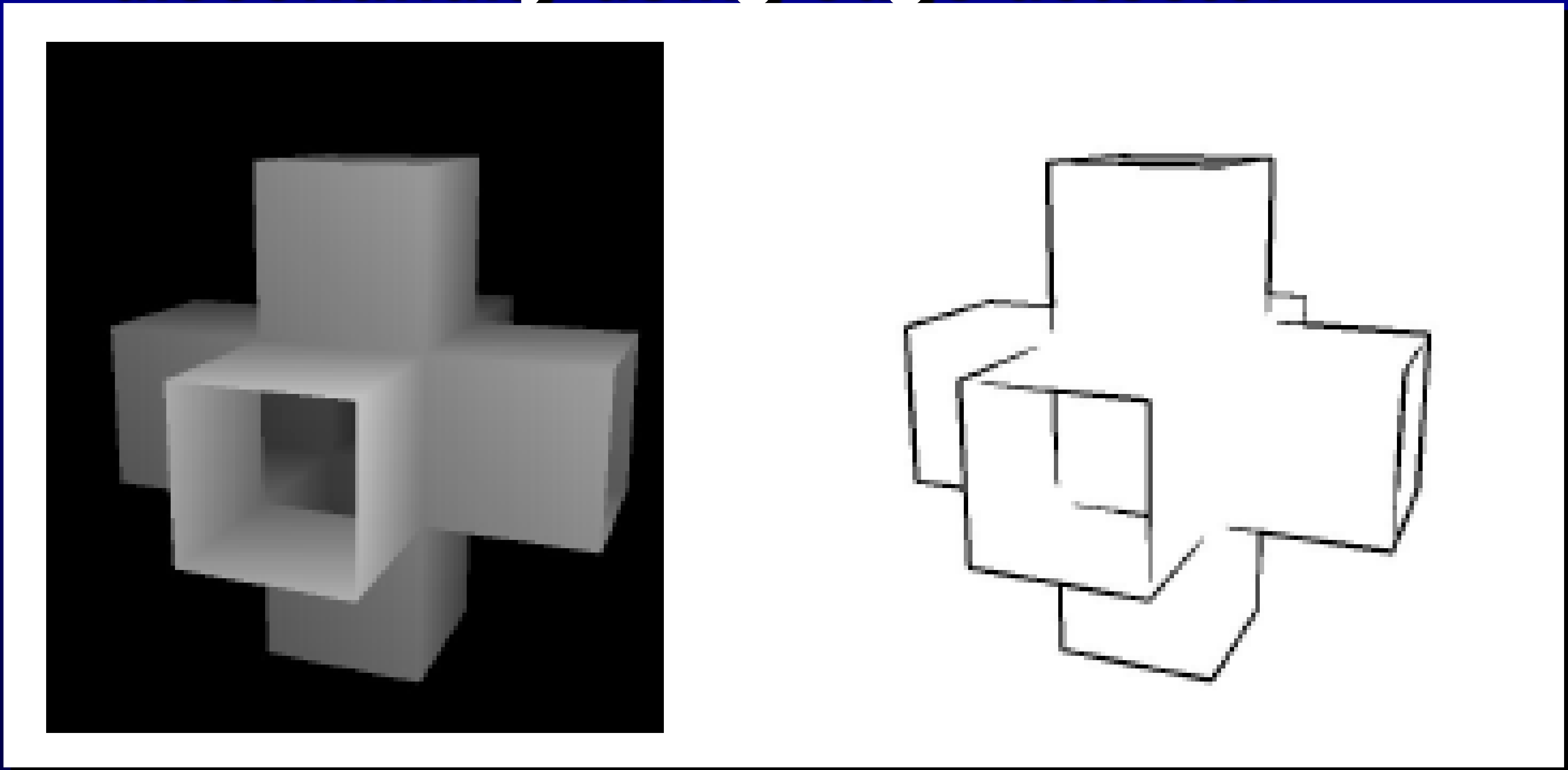


Image Space Method

- Analyze the normal map – convert surface normal (x,y,z) to (R,G,B) and then detect the color discontinuity

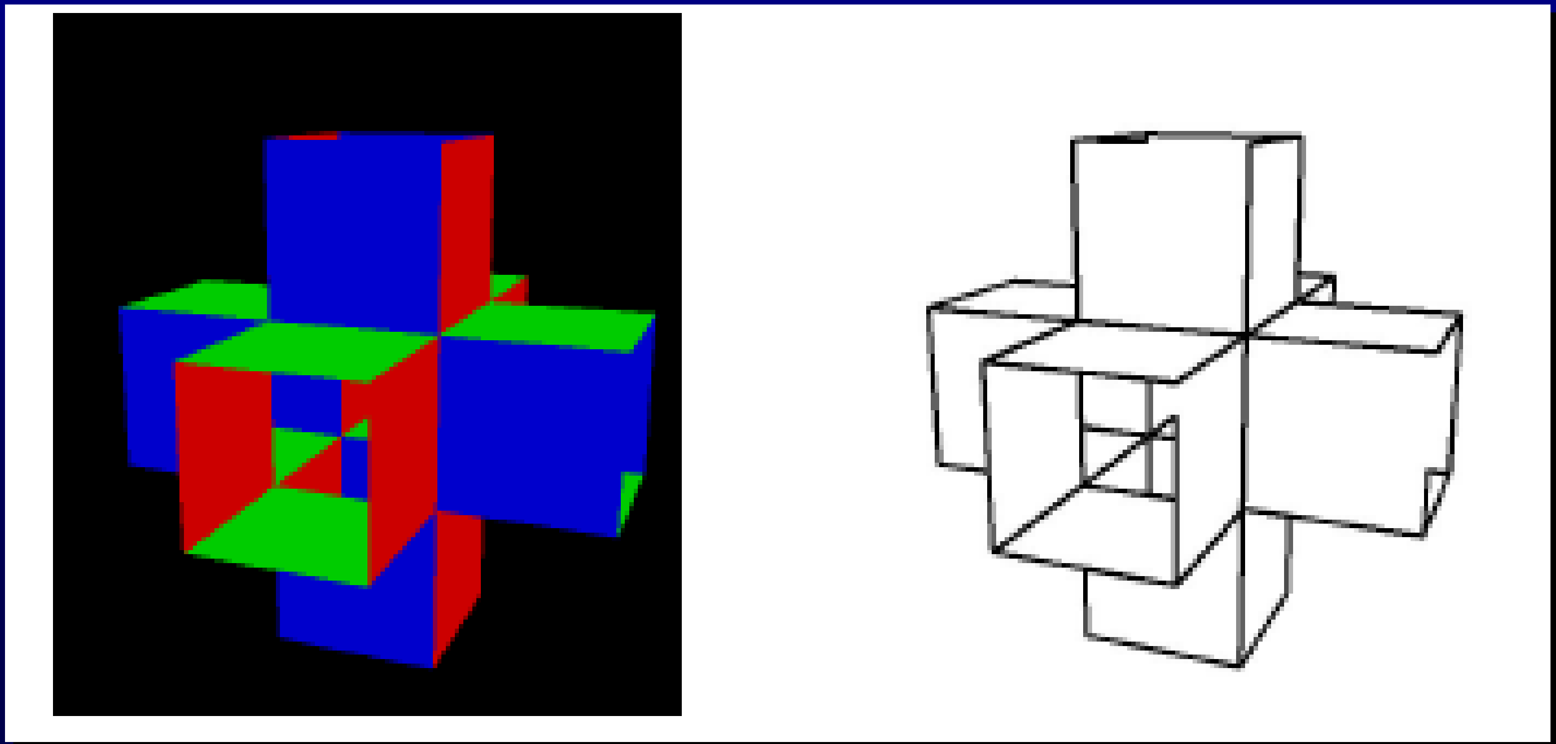


Image Space Method

- Better result can be obtained if both edges are combined

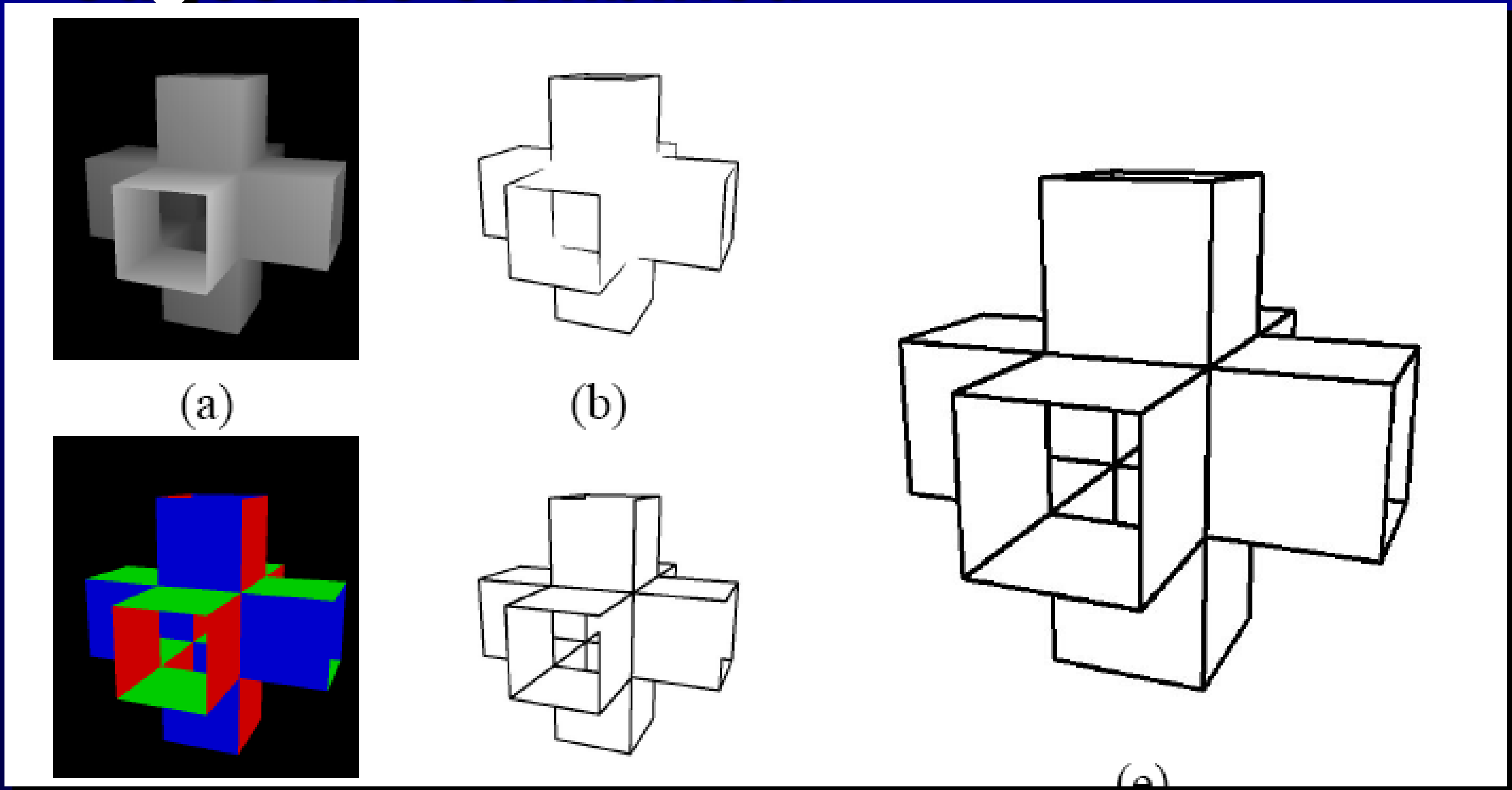
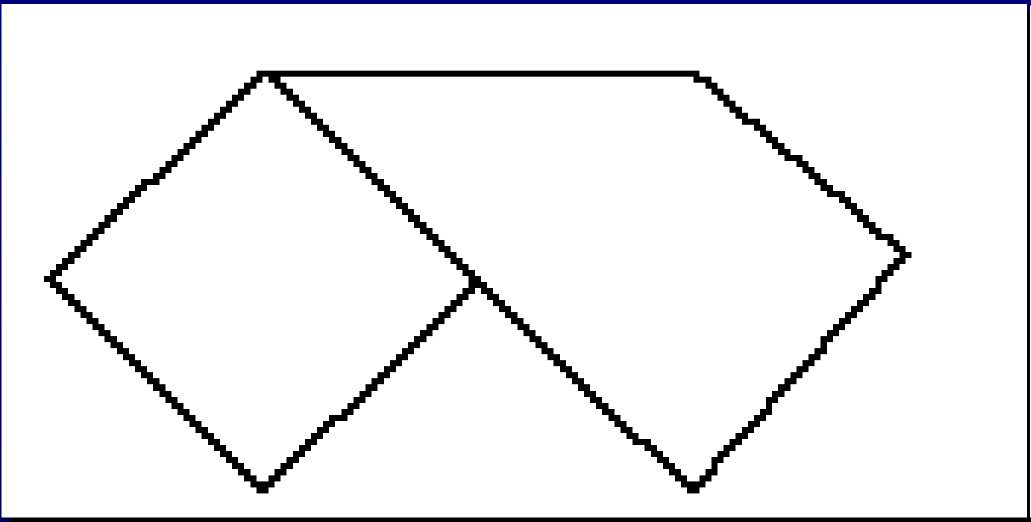
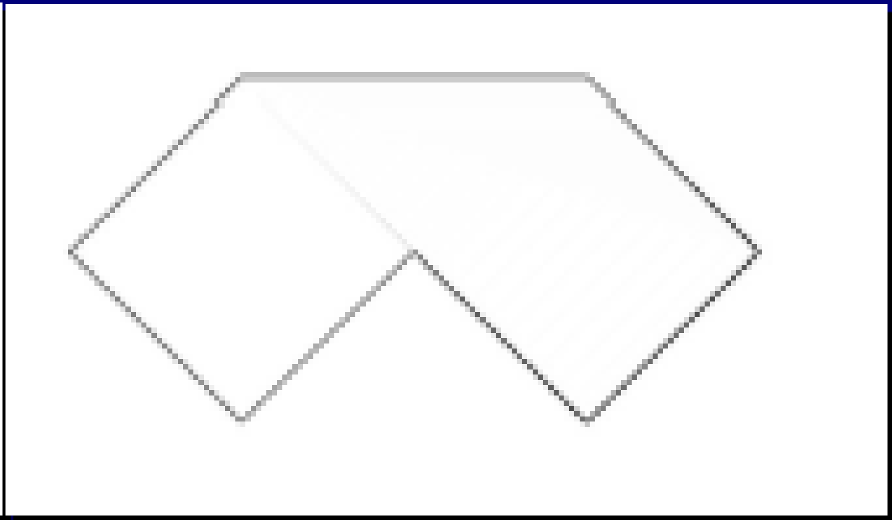


Image Space Method Problem

- For a folded piece of a paper, the edge cannot be detected



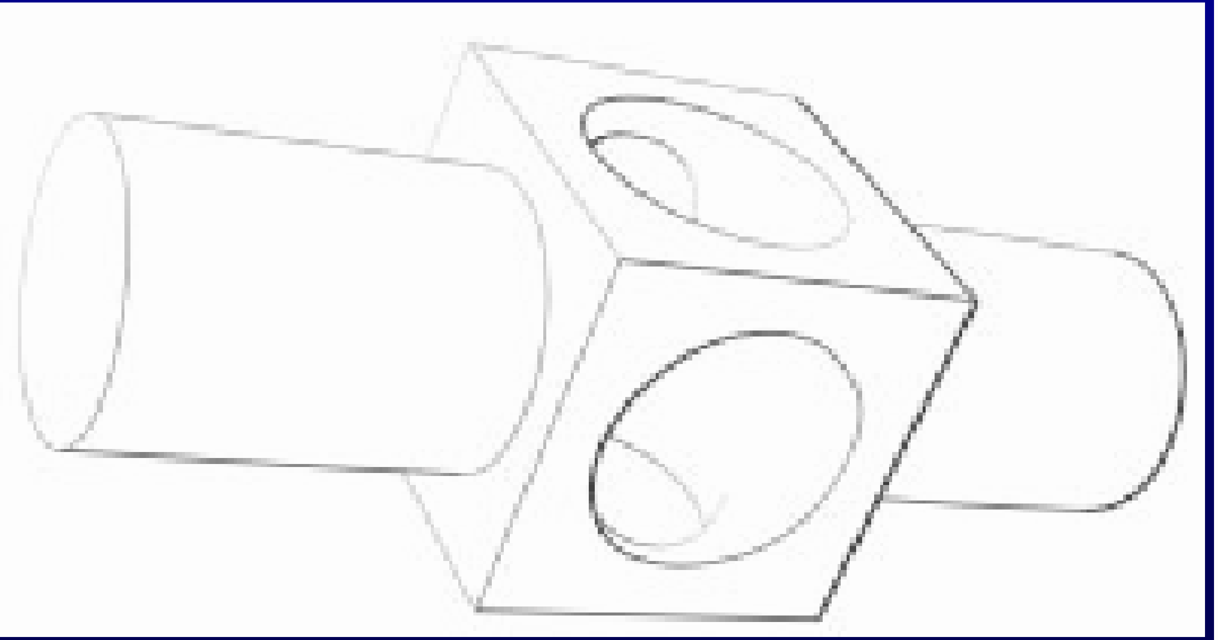


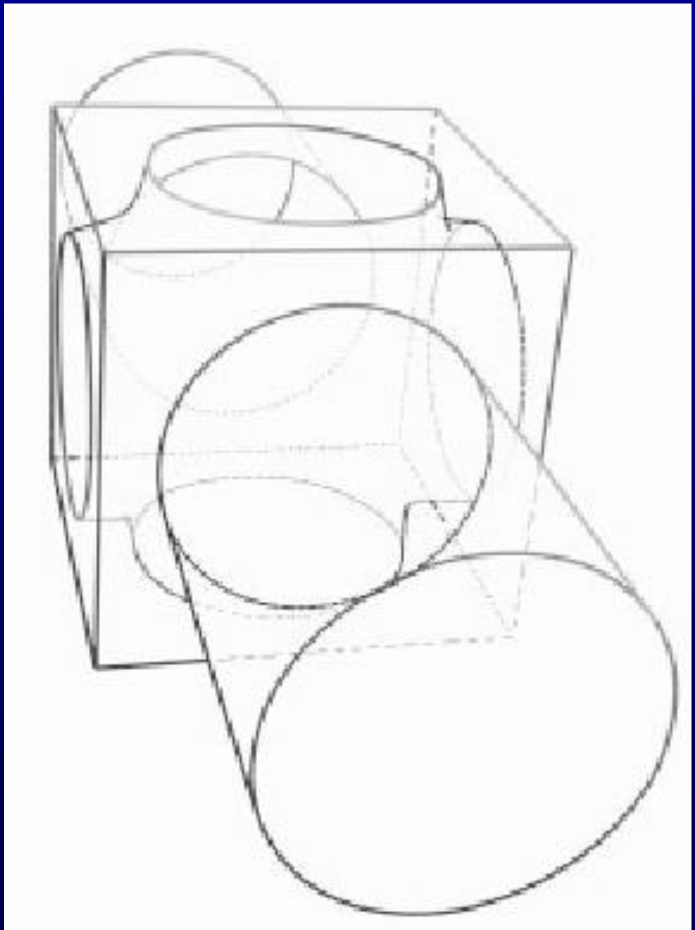
Object Space Method

- **Mainly used to detect silhouettes and creases**
    - *Silhouettes: edges that connect front and back faces*
    - *Creases: A discontinuity on an otherwise smooth edges*

Silhouette Curves

- Constructed from edges shared by both front-facing and back-facing mesh polygons
- Also include boundary edges
- Can be traced incrementally as a string of silhouette edges
- May not be visible, or not entirely visible
- Probability that an edge is a silhouette is proportional to $\pi - \theta$, where θ is the edge's dihedral angle





Software Method

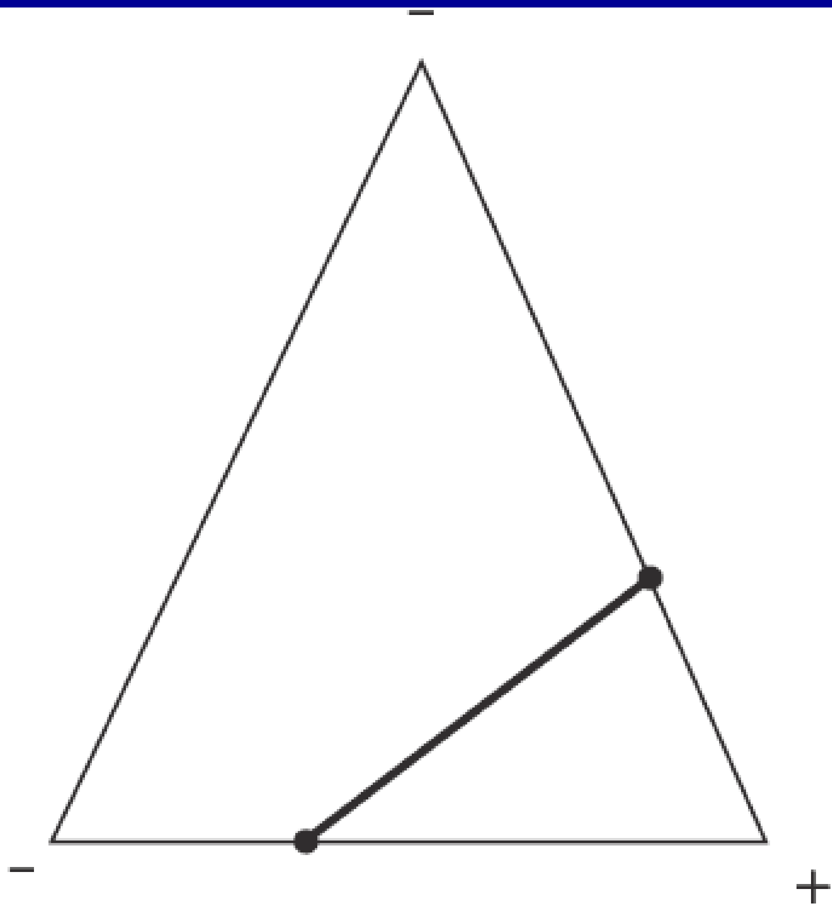
Detect Silhouettes from all triangle edges

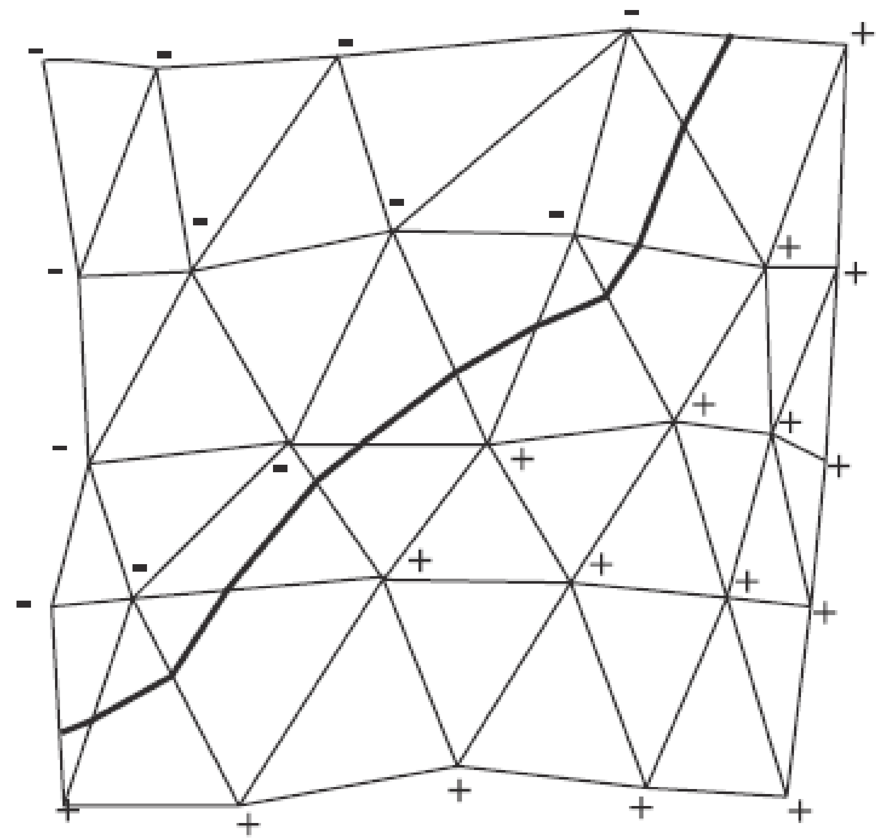
For each vertex, evaluate:

- d = n. (x-e) / |n|*|(x-e)|
- s = + if d >0; else –

Find s = 0 along face edges

Software Method



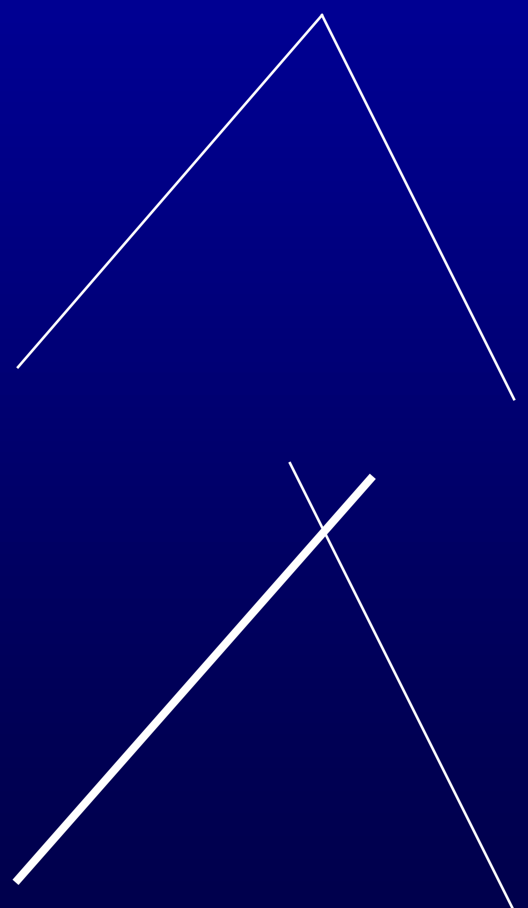


Hardware Method

- Use OpenGL to draw silhouette edges (no explicit search)
- Pseudo code (a three pass method)
  - draw shaded front faces
  - draw front faces in line mode, set stencil
  - draw back faces in line mode where stencil was set; decrementing stencil

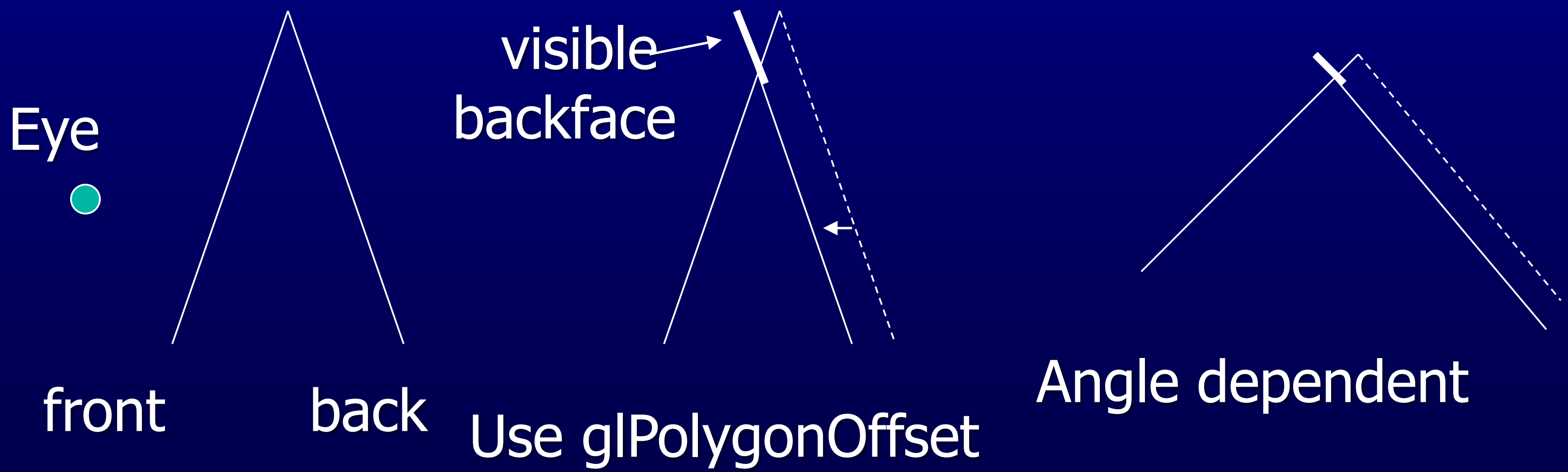
Object-based Silhouettes

- **Thicken all polygons**
- **Turn two-sided shading on**
  - Front face white
  - Back face black
- **Ramesh Raskar. Hardware Support for Non-photorealistic Rendering, Proc. Graphics Hardware 2001.**



Hardware Method

- **Reduce to 2 pass by push the backface forward (z bias)**



Issues of the Previous Method

Non-uniform z resolution needs to be taken care of – translate by k*z;

- K: a scaling factor, z: the polygon distance

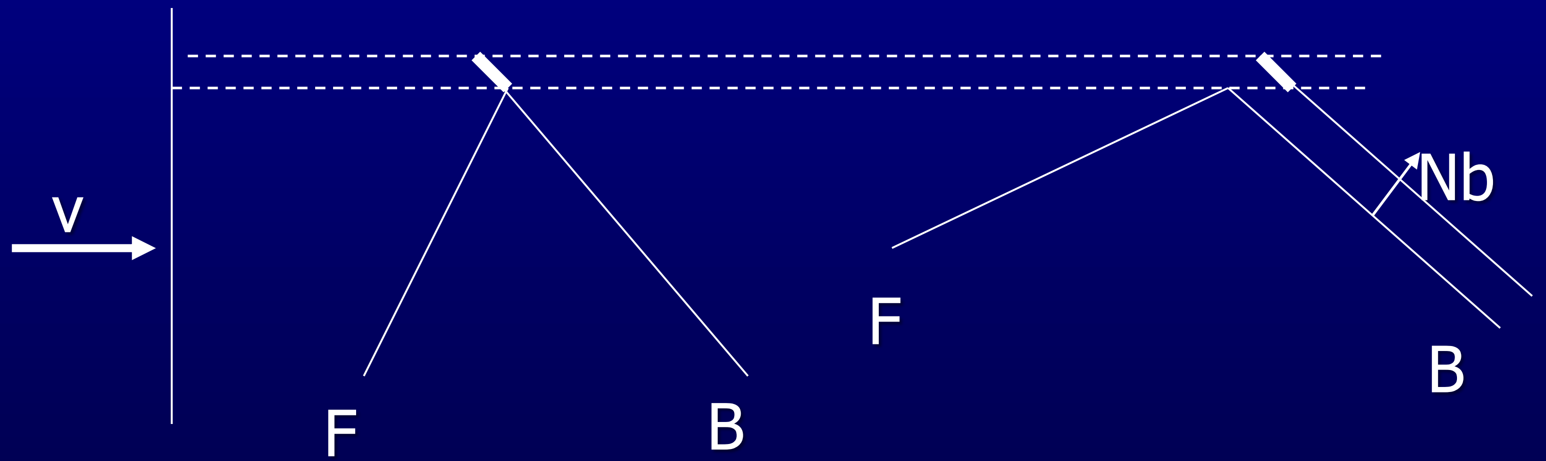
The width of the line will depend on the orientation of the back-facing polygon and front-facing polygon

Raskar and Cohen – fatten the back-facing polygons

Raskar and Cohen's Fix

- **The back-facing polygon edge is pushed outwards**

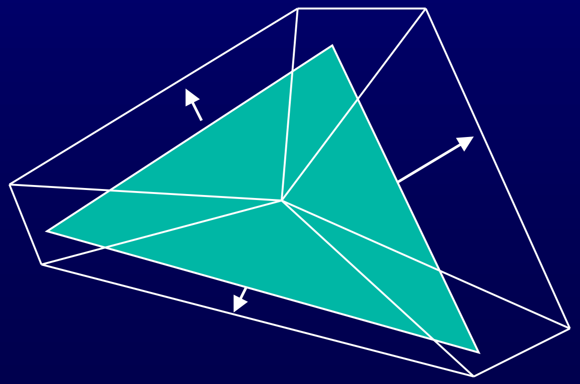
By Offset = K * z/V.Nb



The distance to push only depends on the orientation of back-facing polygon

Raskar and Cohen's Fix

- In fact, each of the polygon edges needs to be pushed by a different amount: z*sin(α)/V.Nb; where cos(α) = v.e, e is the polygon edge vector



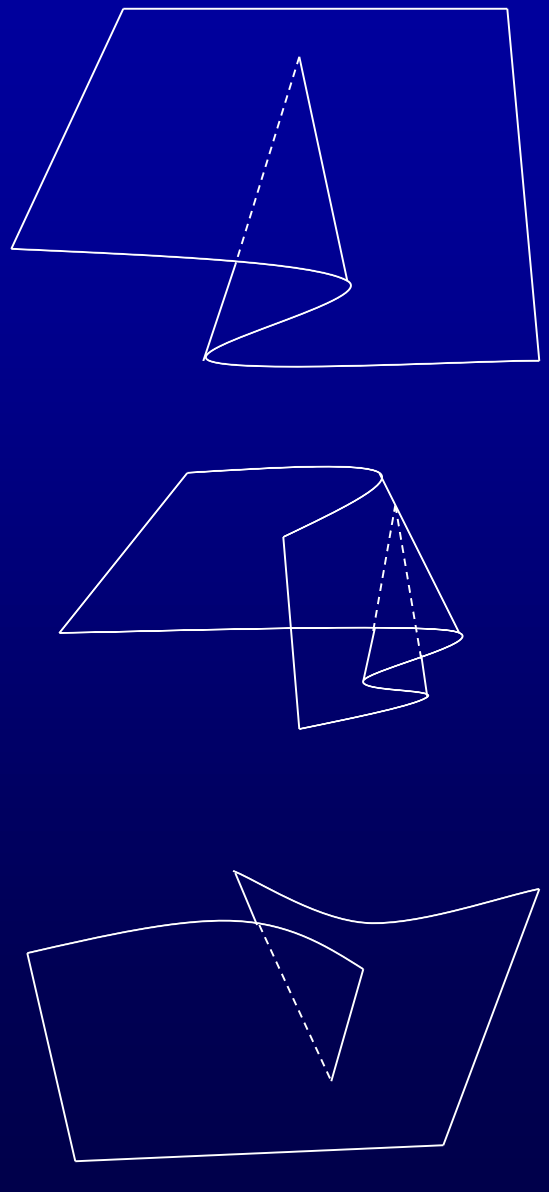
Catastrophes

Cusp vertices

- Connect concave and convex silhouette edges
- Connect more than two silhouette edges
- Connect at least one border edge

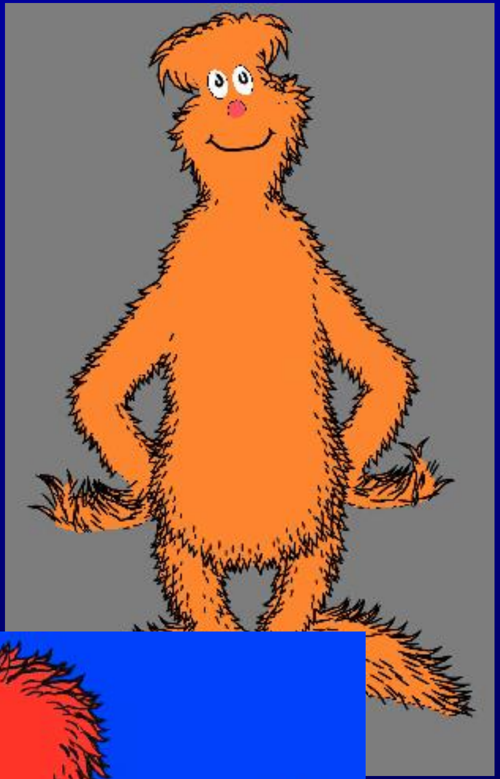
Silhouette edges are

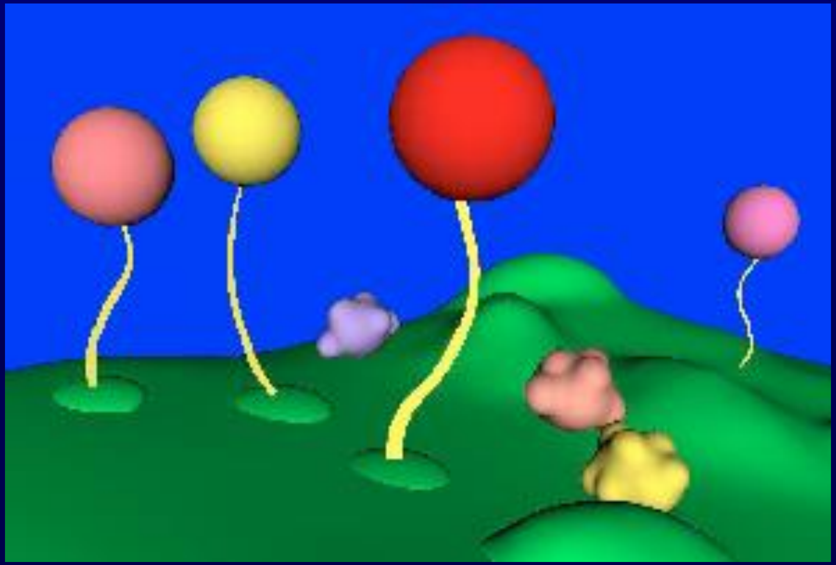
- Convex if closer face is front facing
- Concave is closer face is back facing

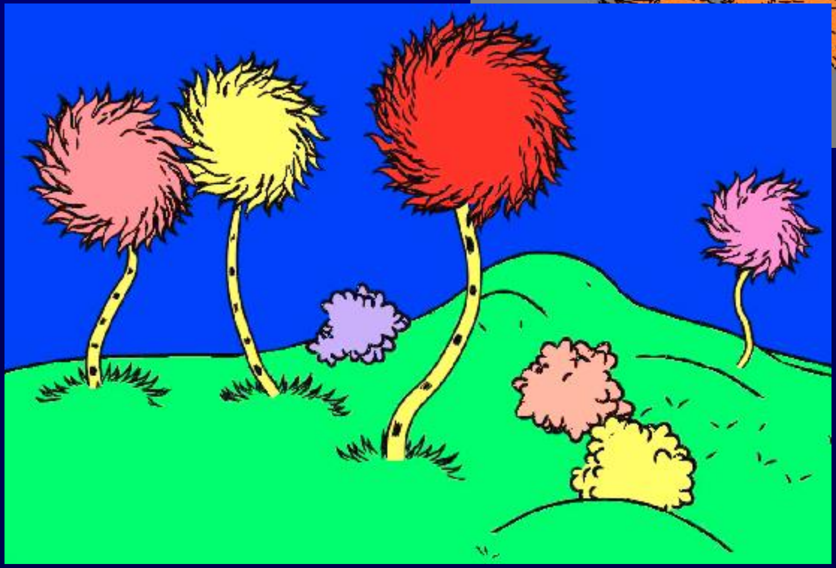


Displacement Silhouettes

- Kowalski *et al.*, S99
- Add displacement texture to silhouette
- Texture controlled by $N \cdot V$







Charcoal Effect

- Tessellate the polygon to smaller pieces
- Also fatten front-facing polygons with $0<N.V<0.1$
- Assign color $I = (1+V.N)/3$

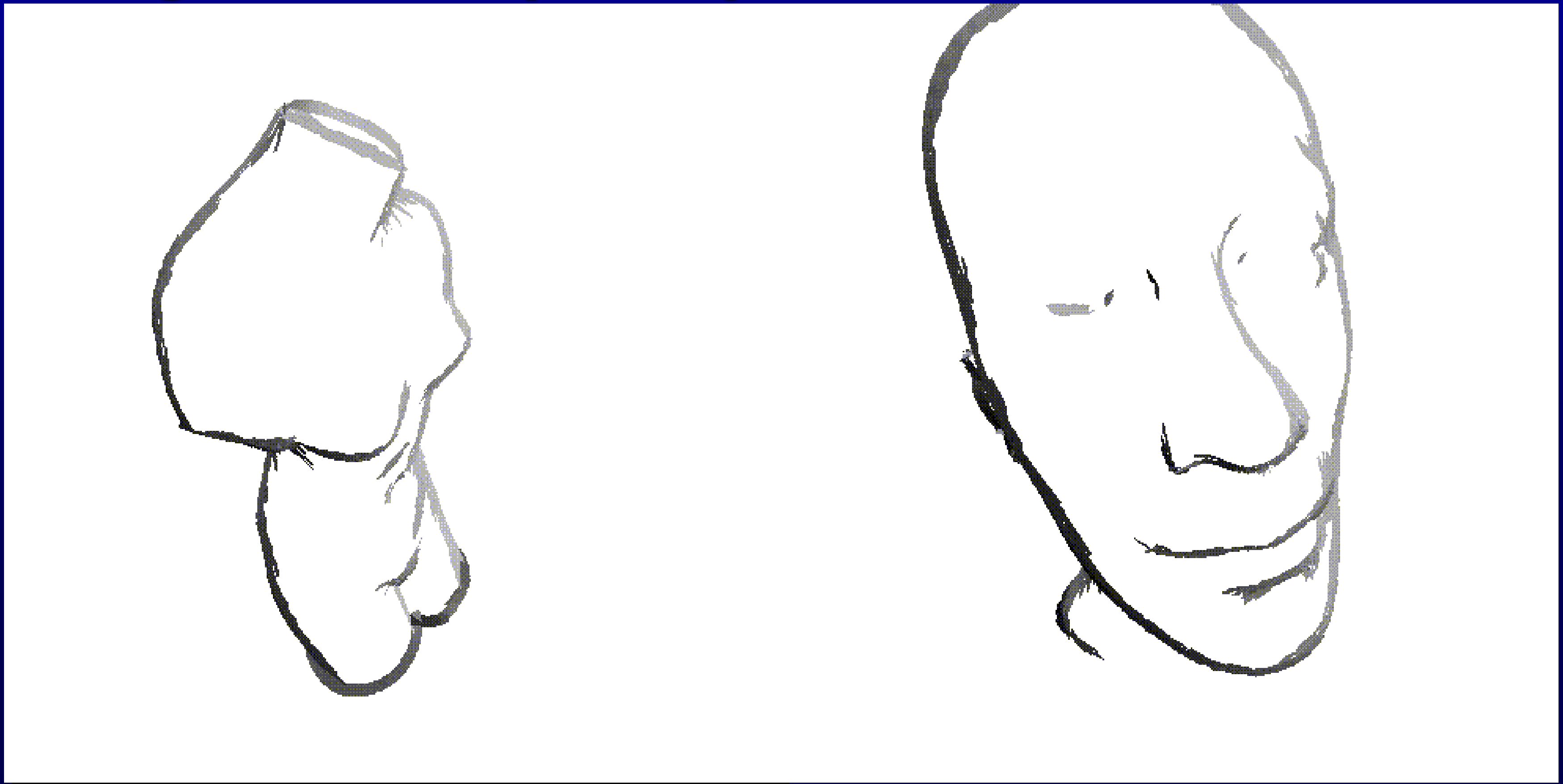
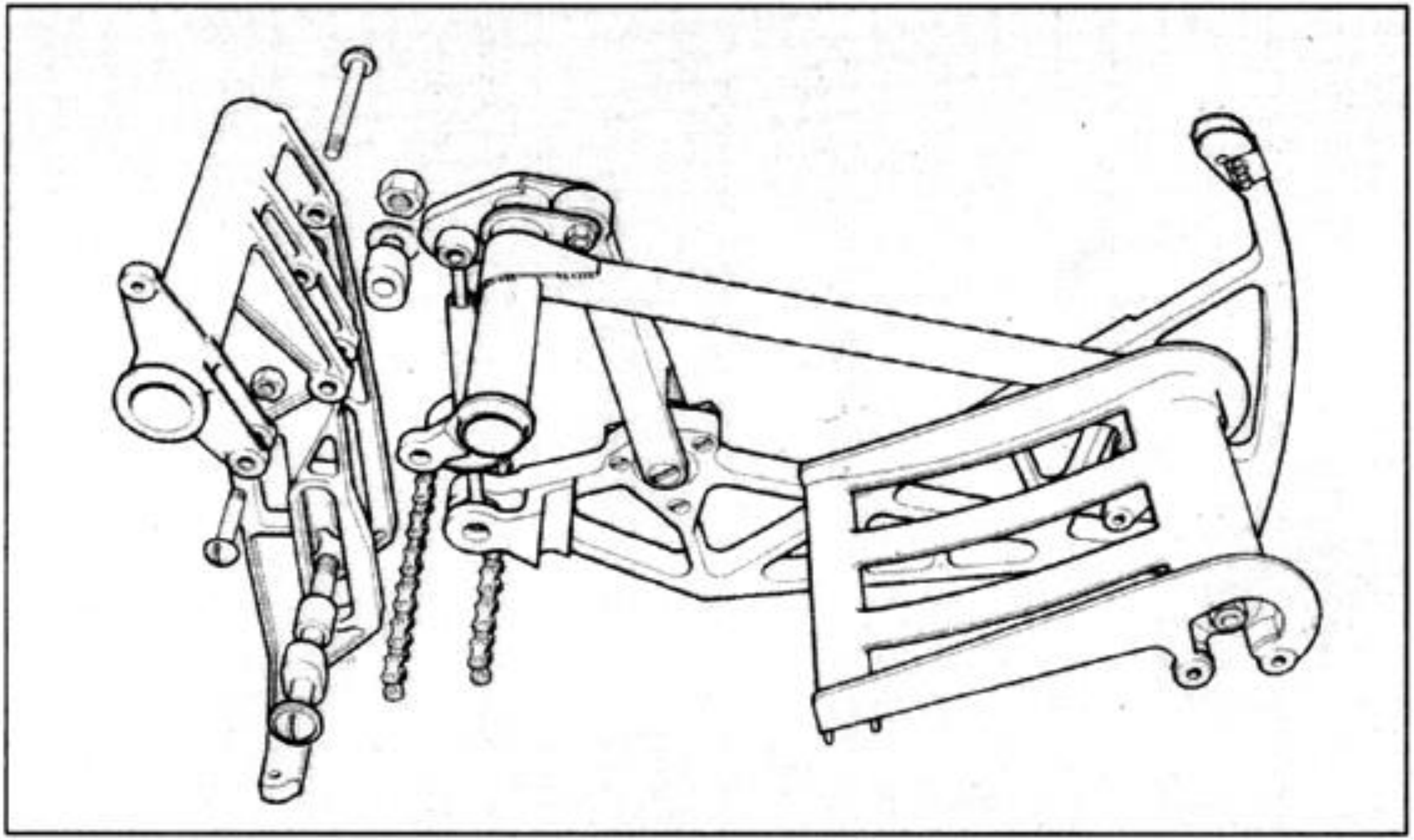


Illustration Example

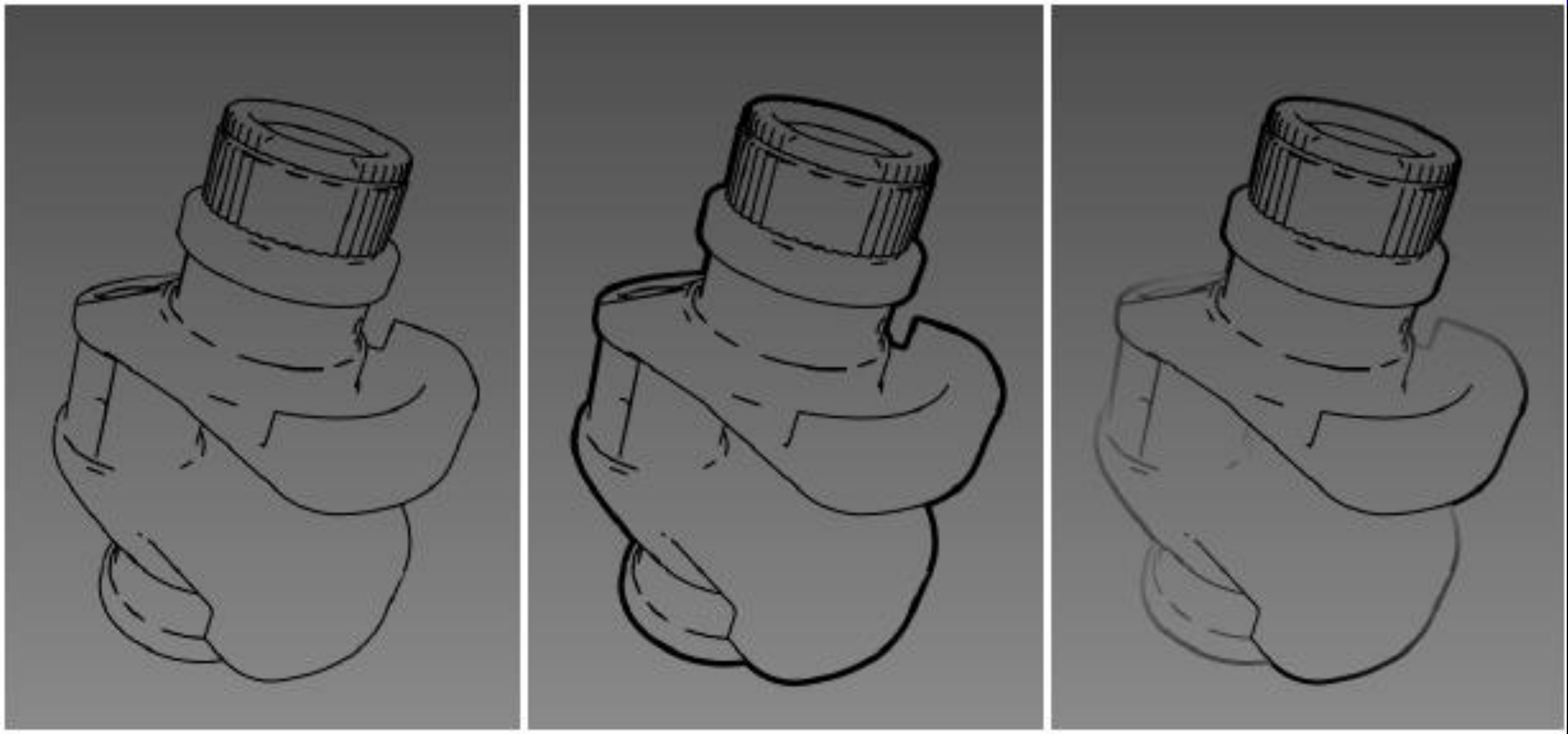


Line Weight

Some possible choices:

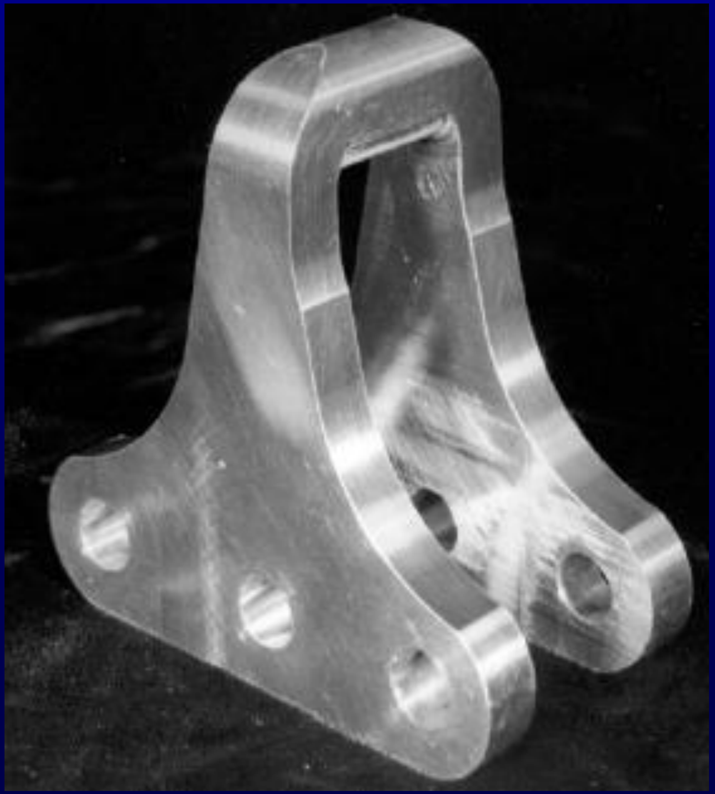
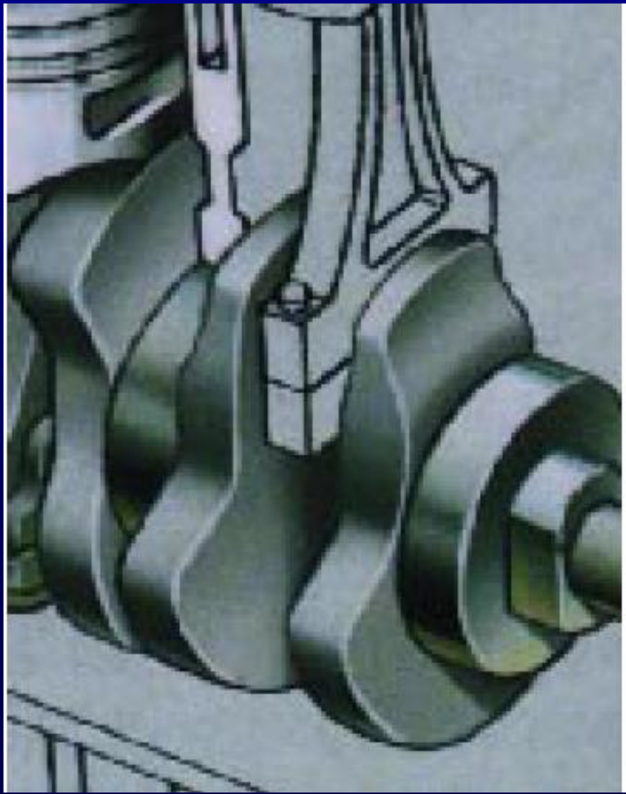
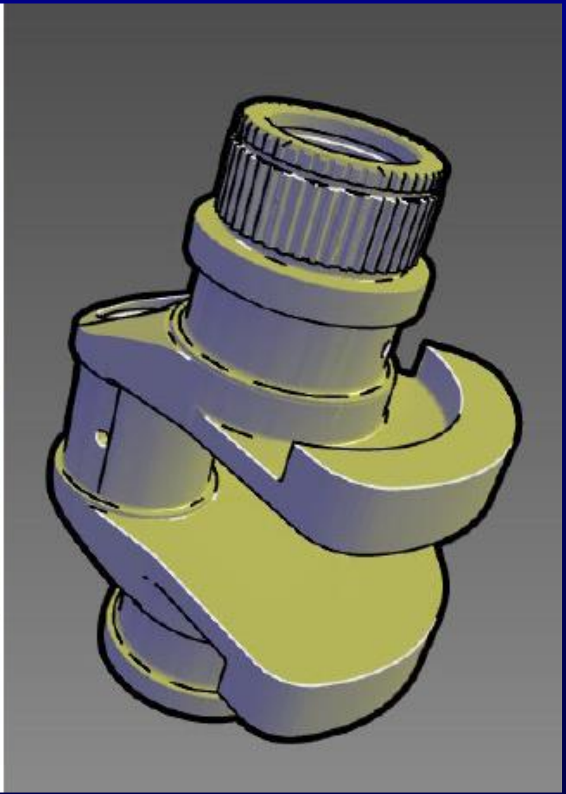
- Single line weight used throughout the image
- Two line weights, with heavier describing the outer edges (boundary and silhouette)
- Various light weight along a single line, emphasizing perspective effect (heavy lines in the foreground, tapering toward the farther part of the object)

Line Weight



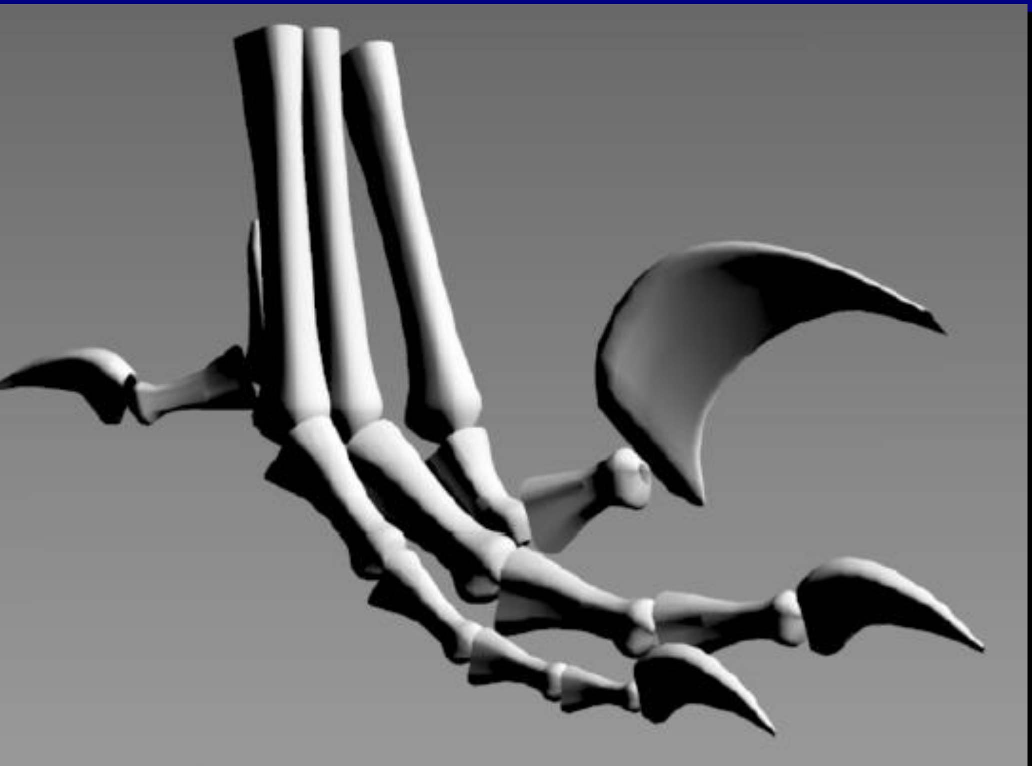
Line Color

- Attempt to incorporate shading
- Interior lines can be drawn in white, simulating highlight

Tone Shading

- **The standard Phong Shading model is not always satisfactory**



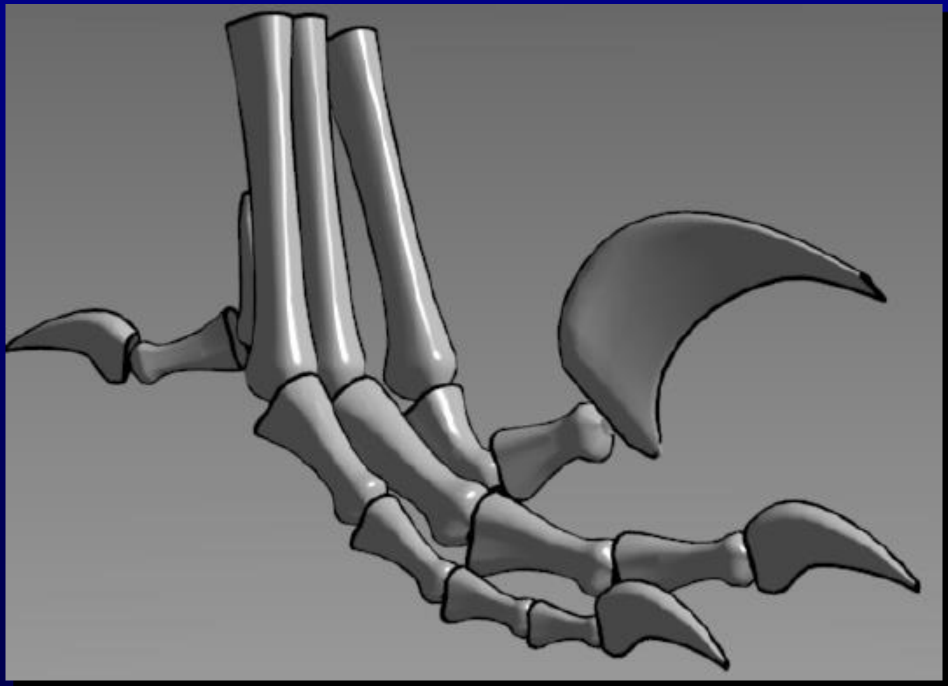
Problems in regions where N.L < 0

-Only ambient colors are seen
-Difficult to deduce shapes
-Object outlines can not be seen

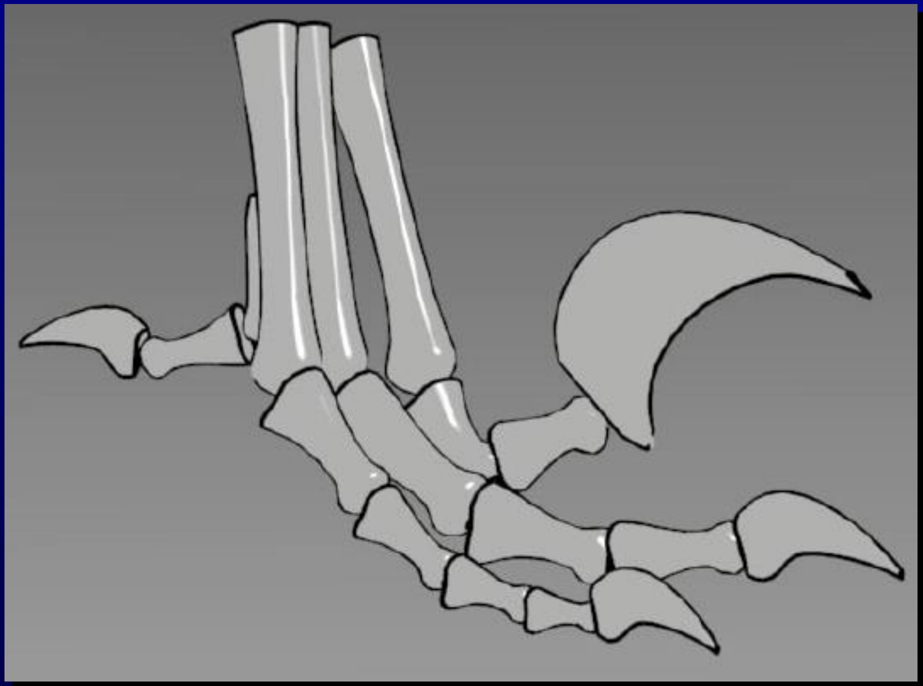
Two ad hoc Solutions

Hand-tuned ambient color

Just highlights and edge lines



Ambient is only a constant ☹



Not enough surface detail ☹

Effective Shading Model Needed

Shading model is insufficient

Lost shape information

- Especially in the areas of subtle curvature (small claws above)

Not automatic, lots of hand-tuning

Tone Shading Goals

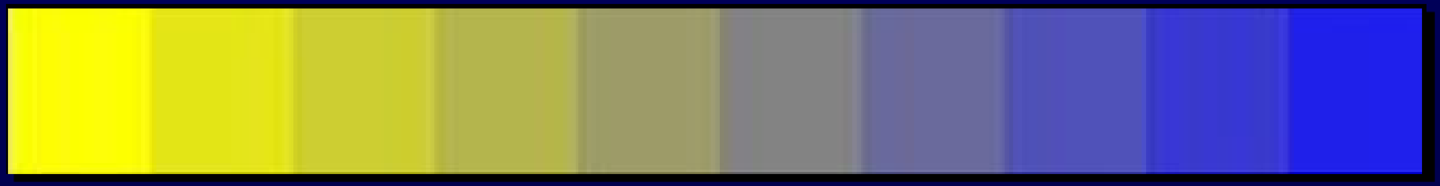
To include shading in an image with back edge lines and white highlights visible

- Use a compressed dynamic range for shading
- Use color visually distinct from black and white

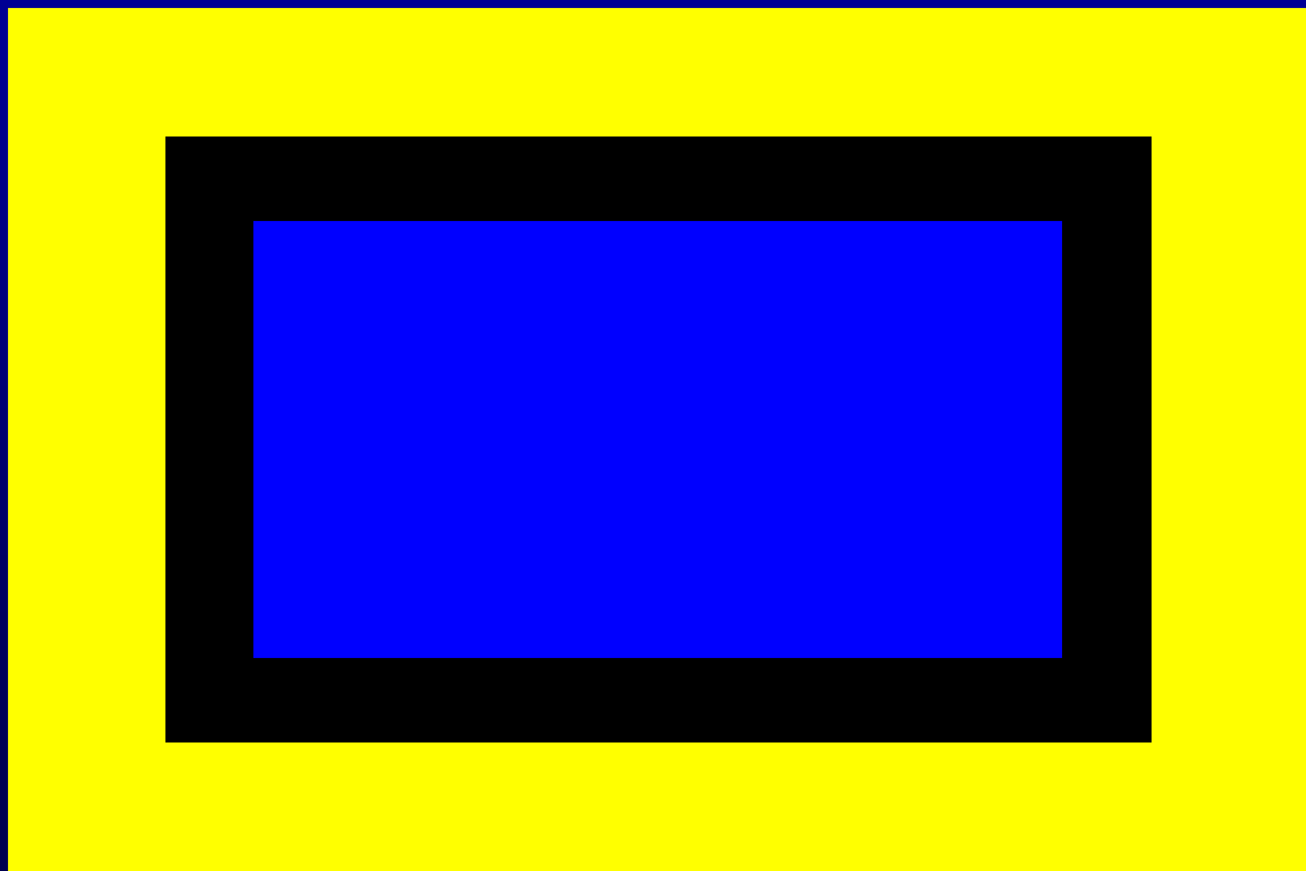
Create Undertone

- To further differentiate different surface orientations, we can use cool-to-warm color undertones

- Cool colors – blue, violet, green
- Warm colors – red, orange, yellow

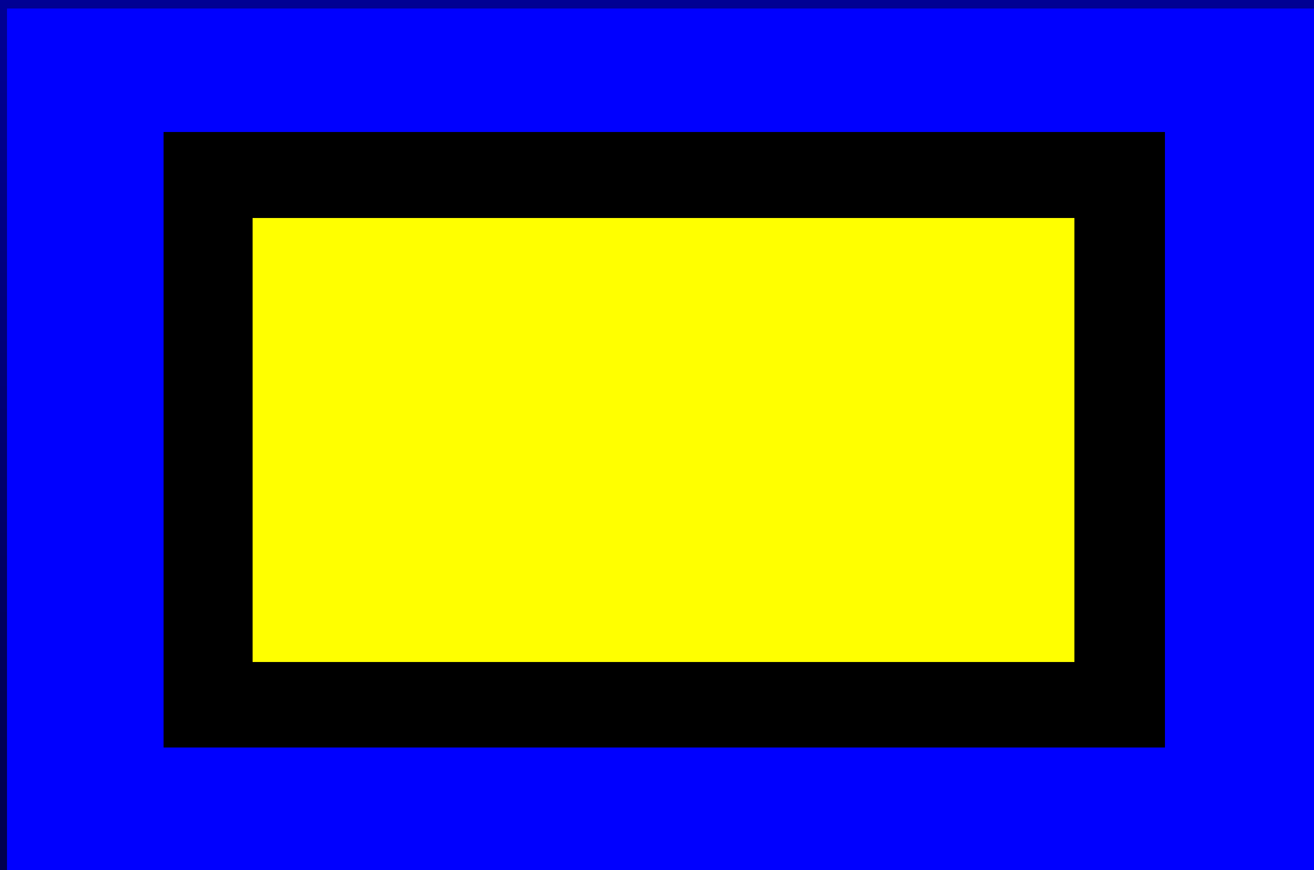


Test Your Perception



Which color (yellow or blue) seems closer?

Test Your Perception



What about now?

Use Warm-to-cool Undertone

We can modify the diffuse Phong Lighting Model

(Blend cool and warm color)

$$I = (1 + L.N)/2 * K_{cool} + (1- (1+L.N/2)) * K_{warm}$$

The Light vector should be place in perpendicular to the gaze direction (usually place at up and to the right)

Tone Shading Equation

$K_{cool} = K_{blue} + \alpha Kd$ (undertone and tone)

$K_{warm} = K_{yellow} + \beta Kd$ (undertone and tone)

$K_{blue} = (0,0,b)$ b in [0,1]

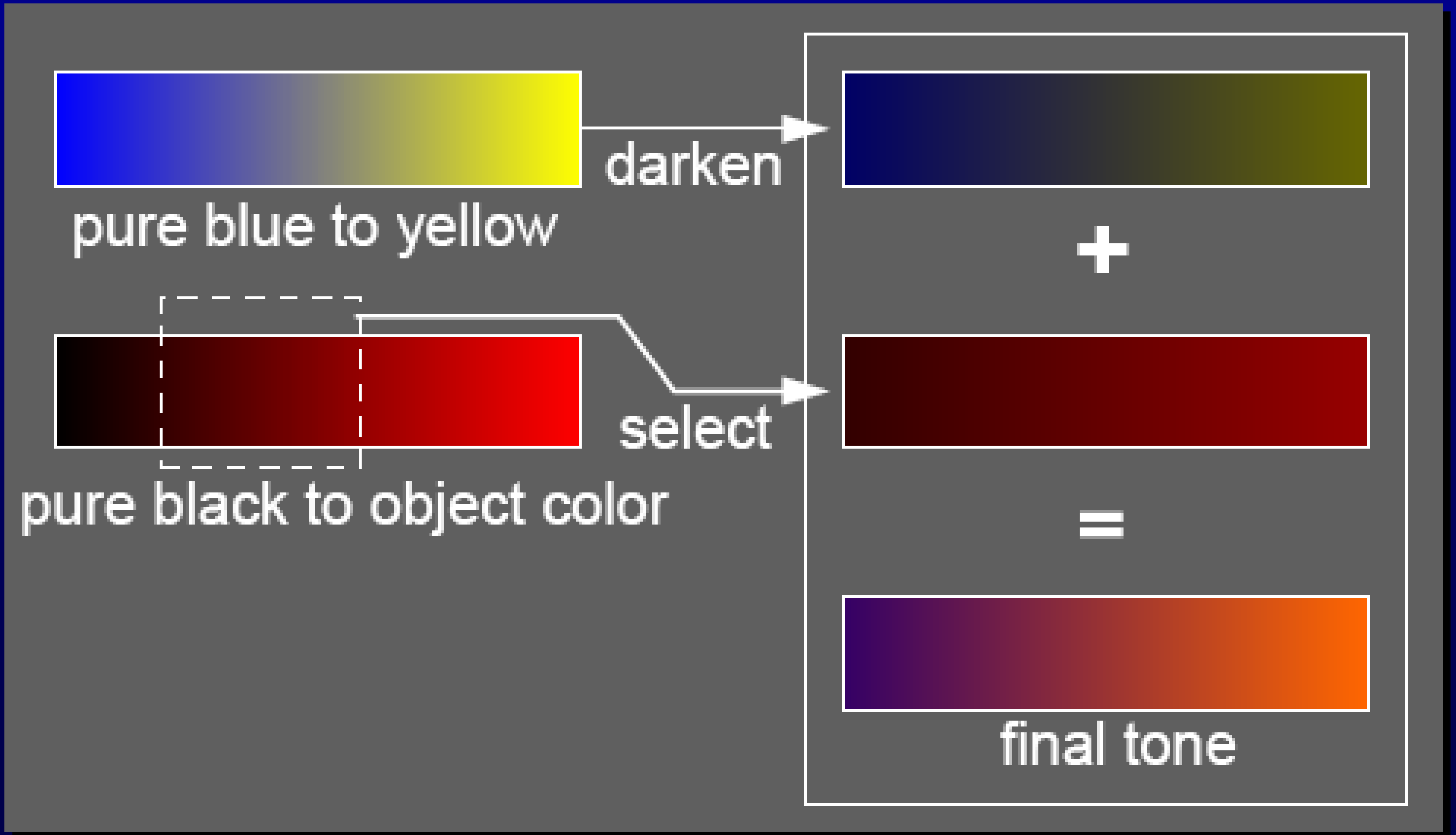
$K_{yellow} = (\gamma,\gamma,0)$ γ in [0,1]

α and β are user-specified parameters

Kd is the object diffuse color

Blend Tone and Undertone

- Add warm-to-cool undertone to a red object

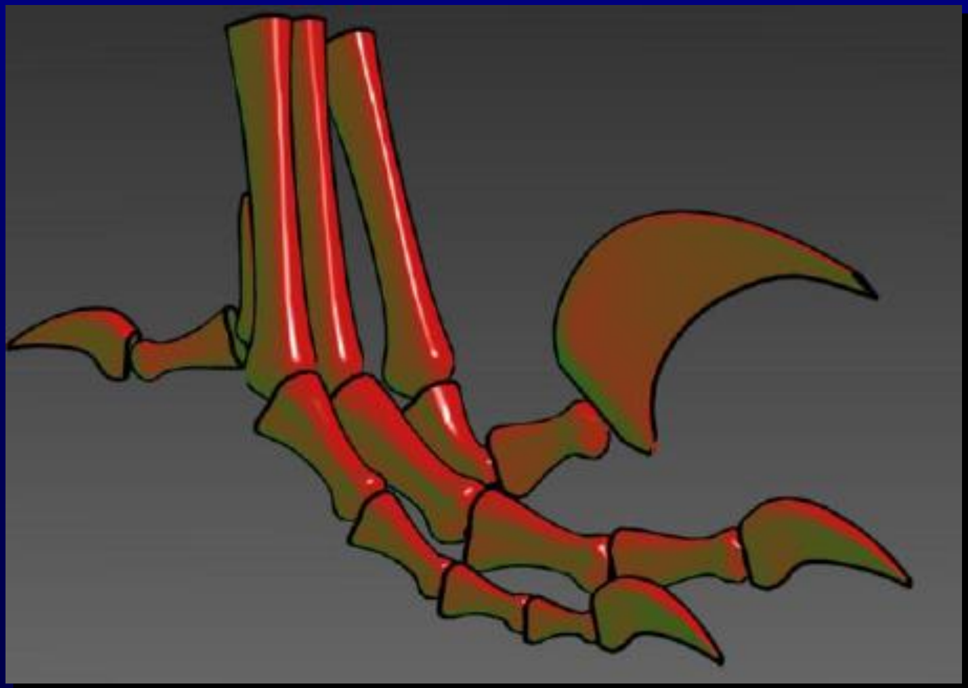


Reduce Dynamic Range

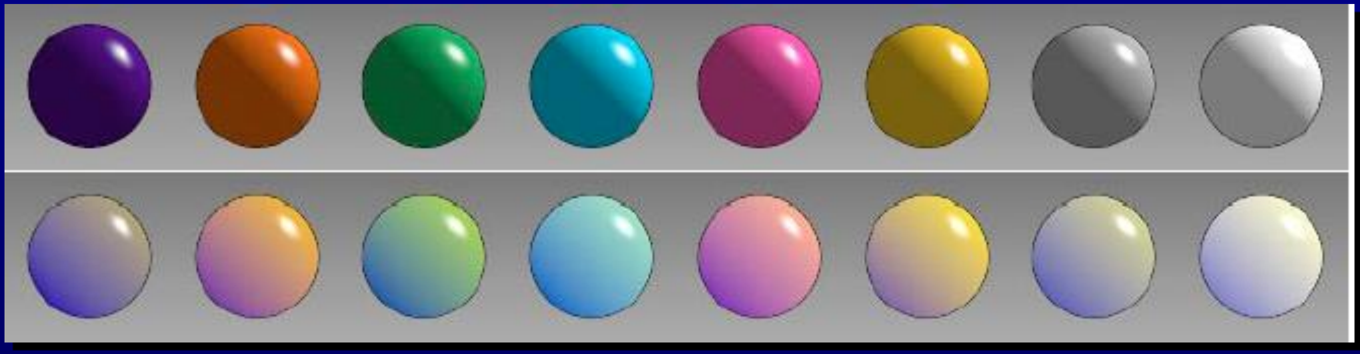
One way to compress dynamic color range is to use colors of different tones

- Add gray to a color to generate different tones

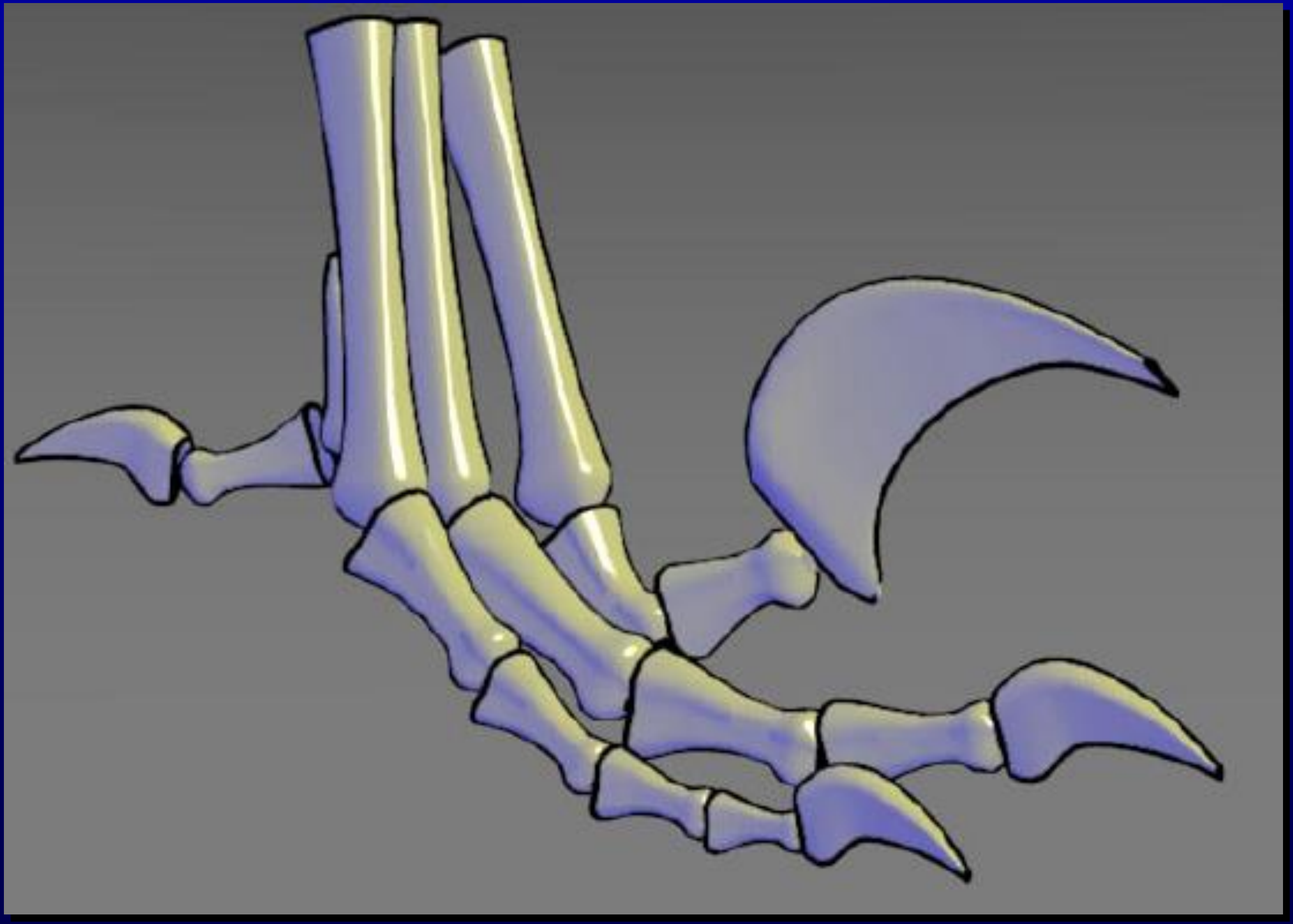
Unnatural color
Lack of luminance difference



Phong and Tone Shadings

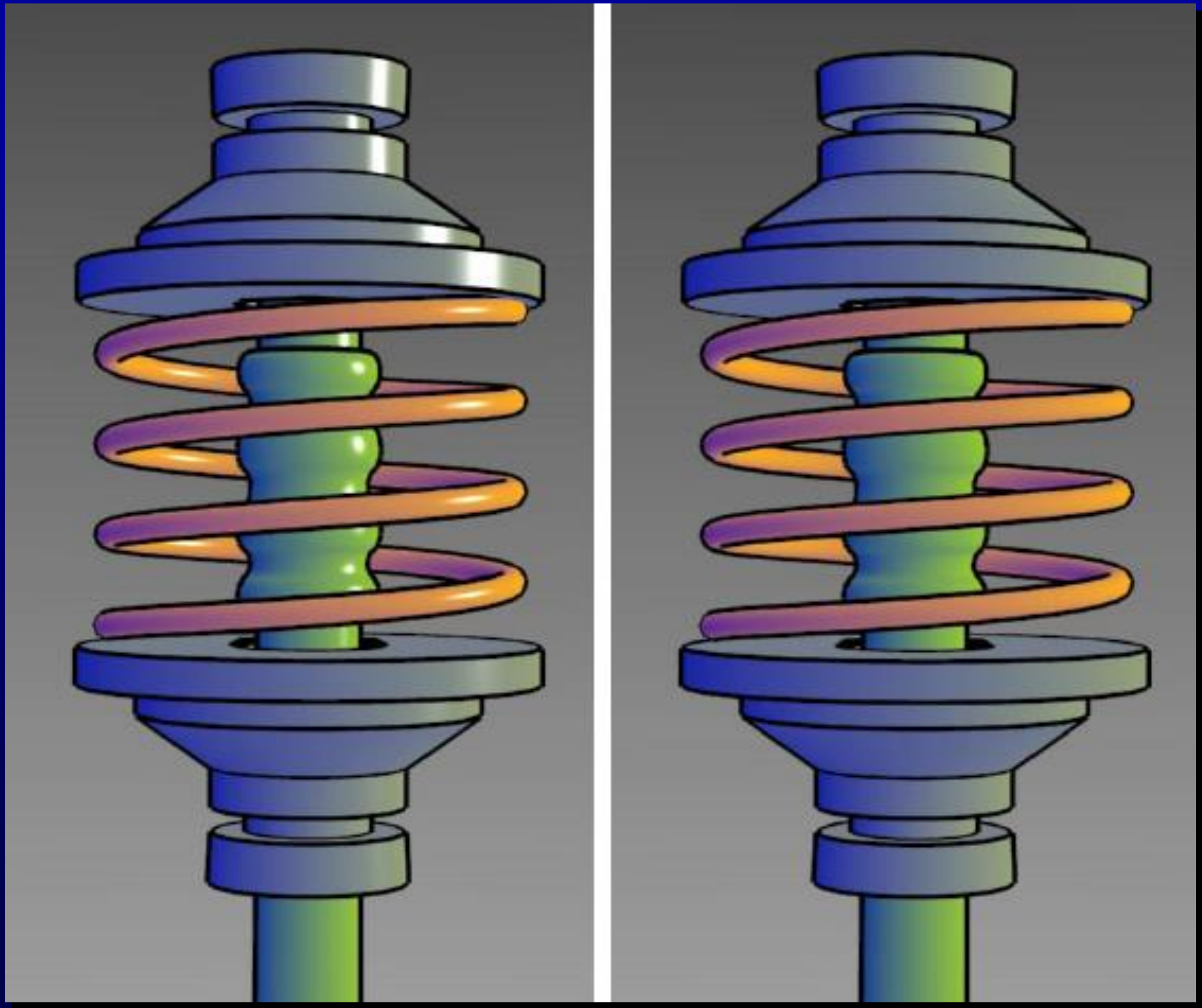


Tone Shading Results

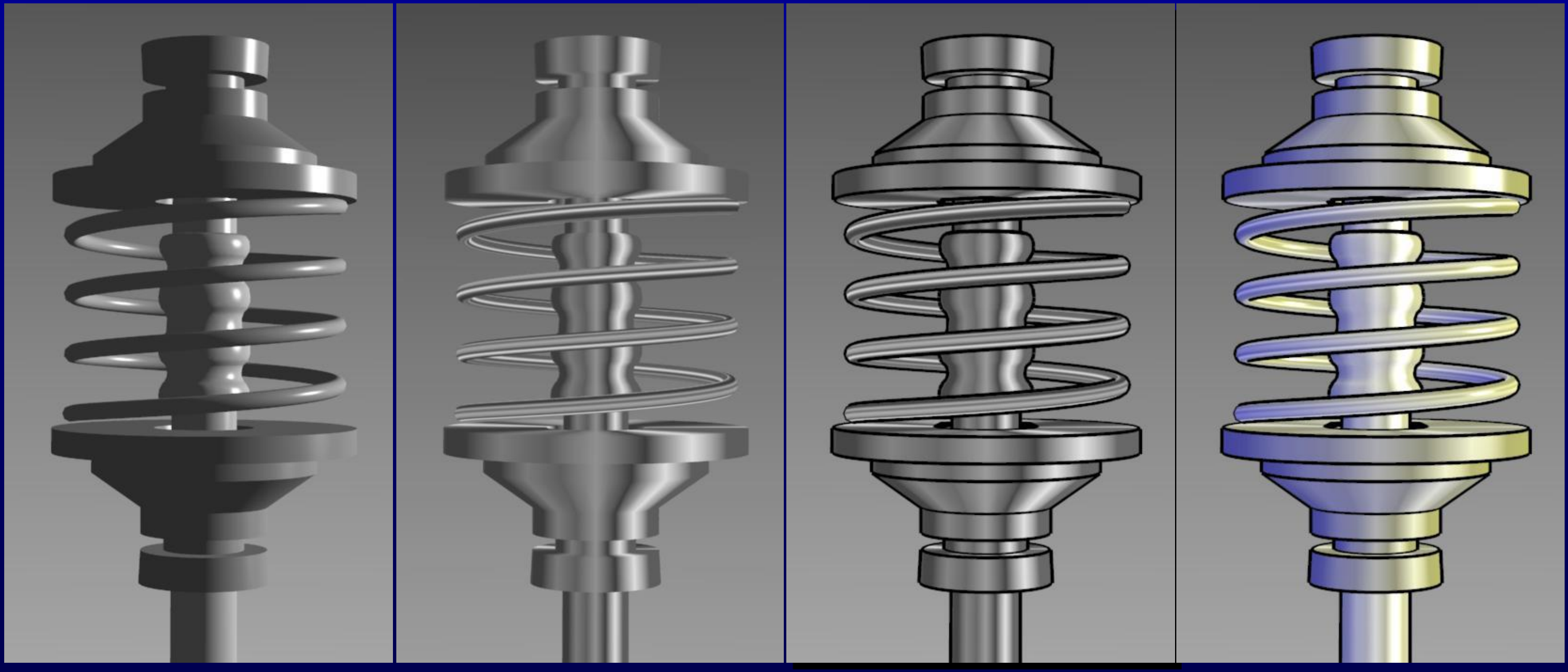


$b = 0{:}55$, $\gamma = 0{:}3$, $\alpha = 0{:}25$, $\beta = 0{:}5$

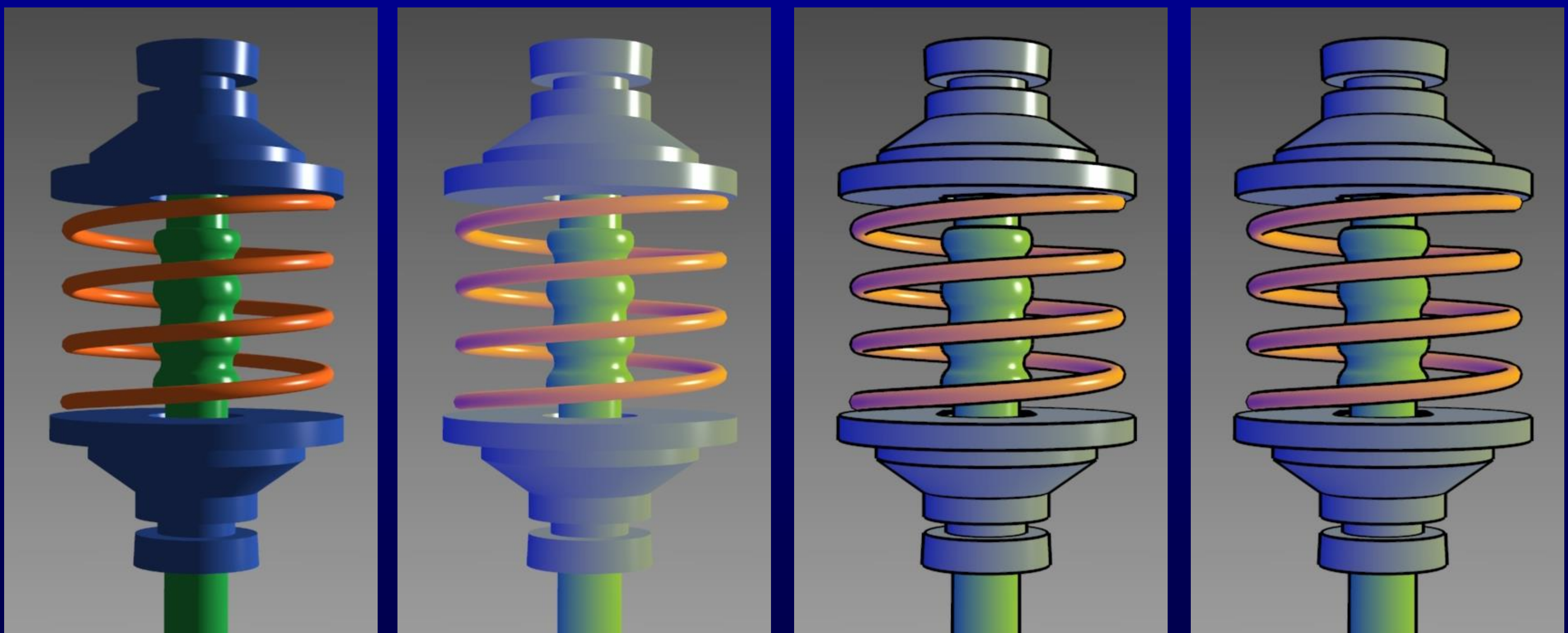
Tone Shading Results



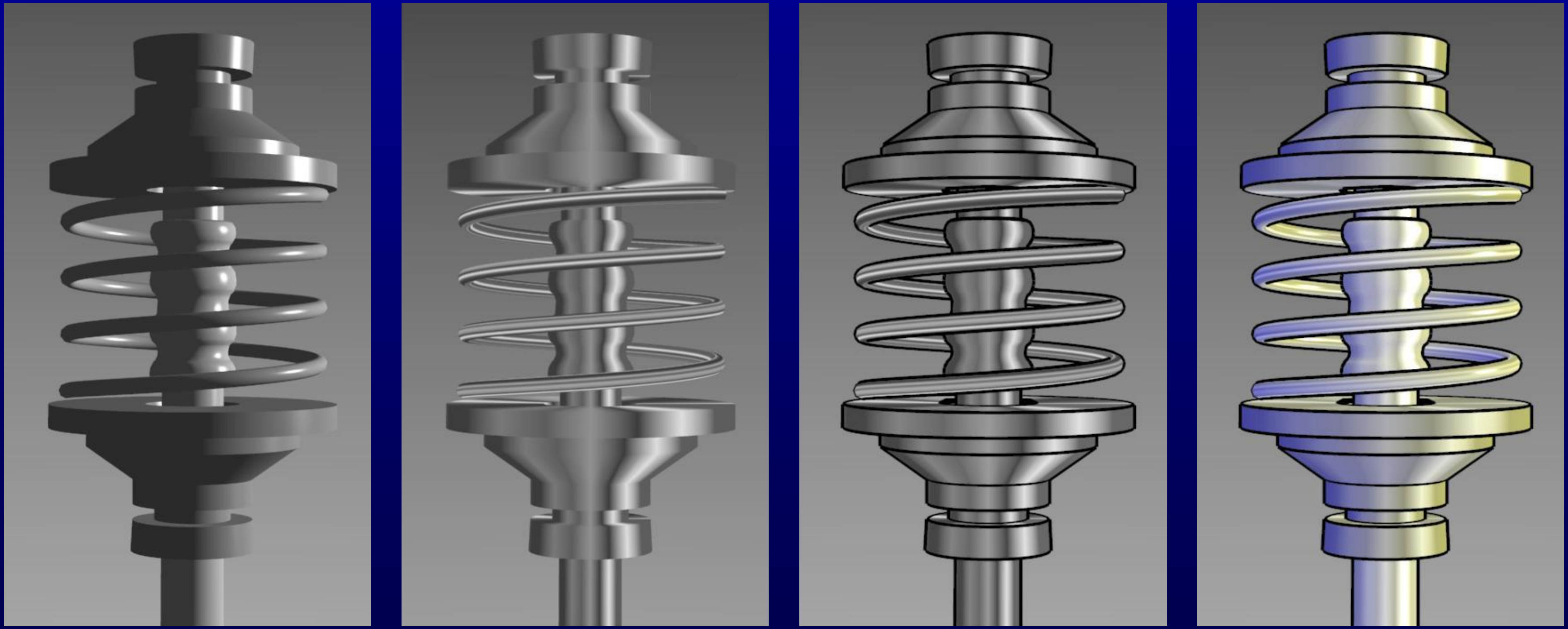
Metal Objects



Plastic Renderings

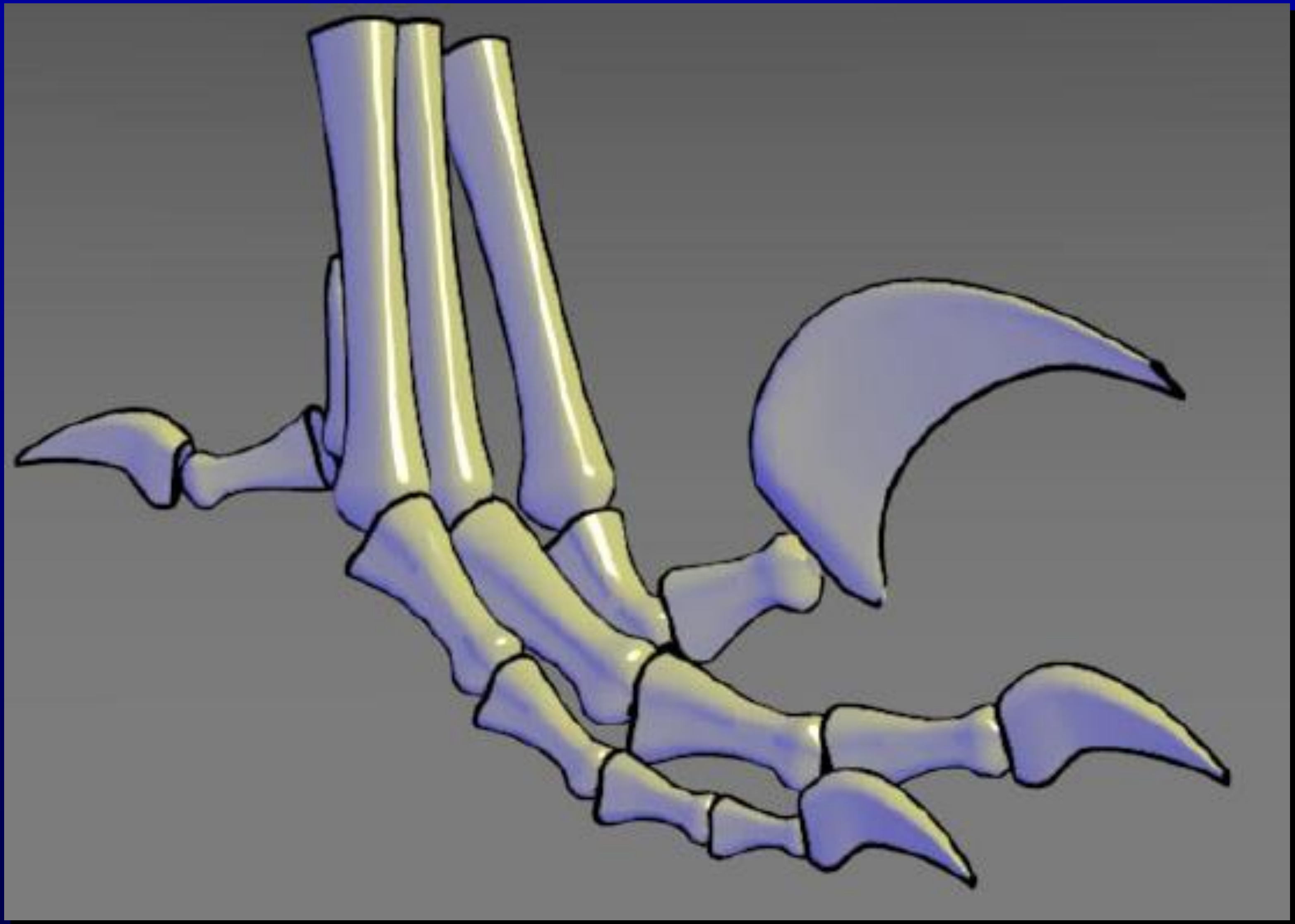


Metallic Renderings



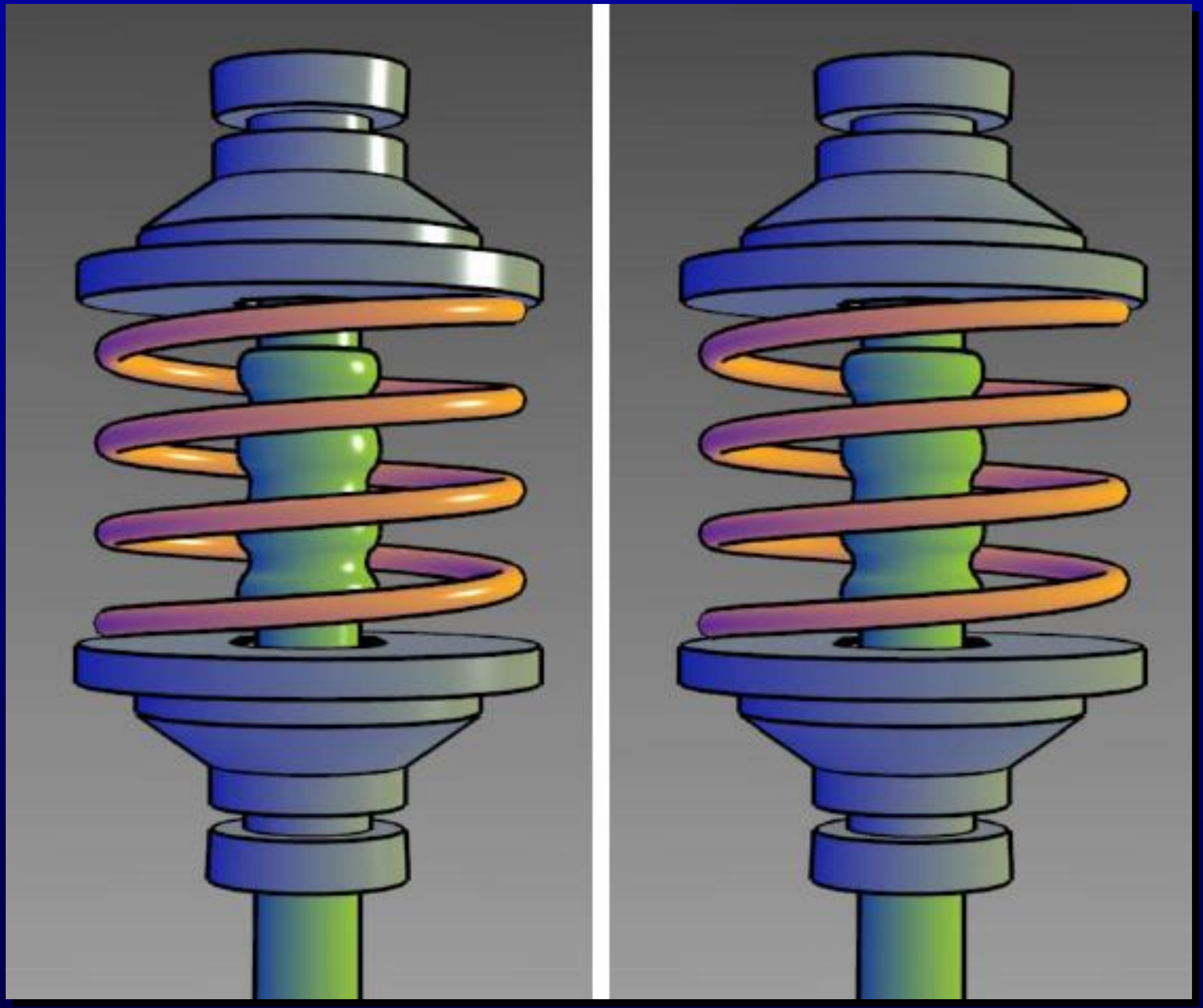
Phong and Tone Shadings

Tone Shading Results

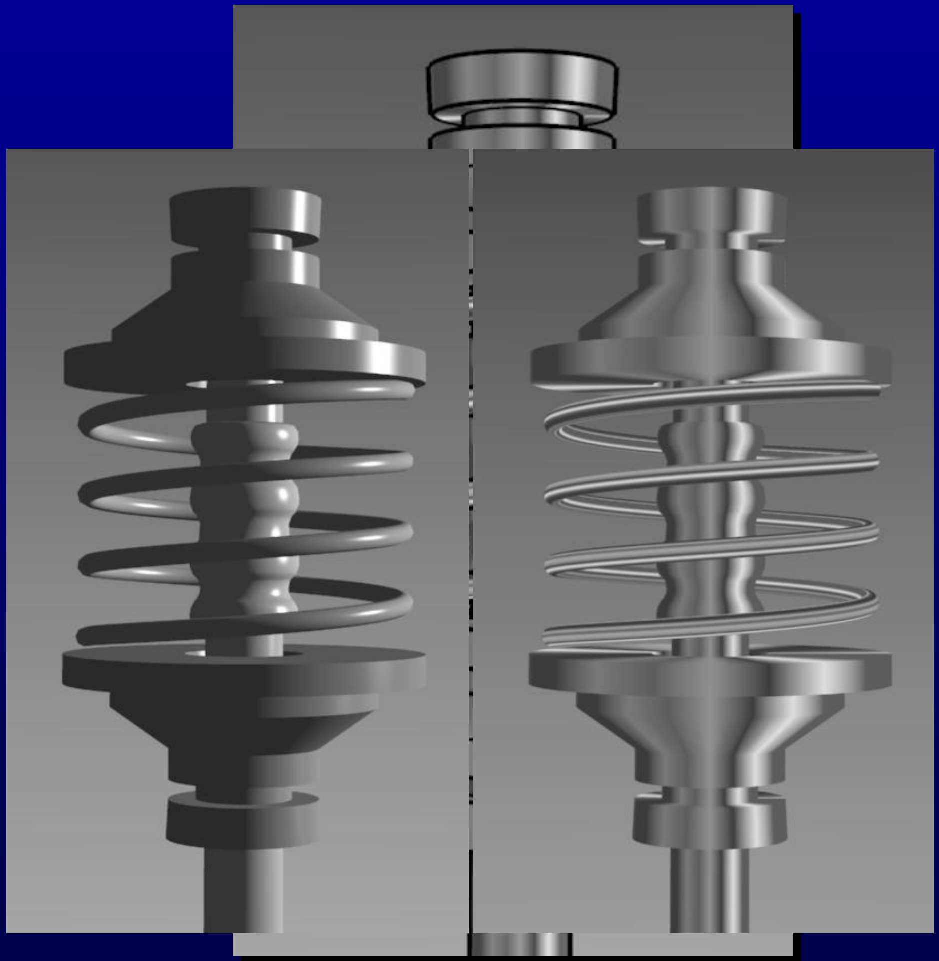


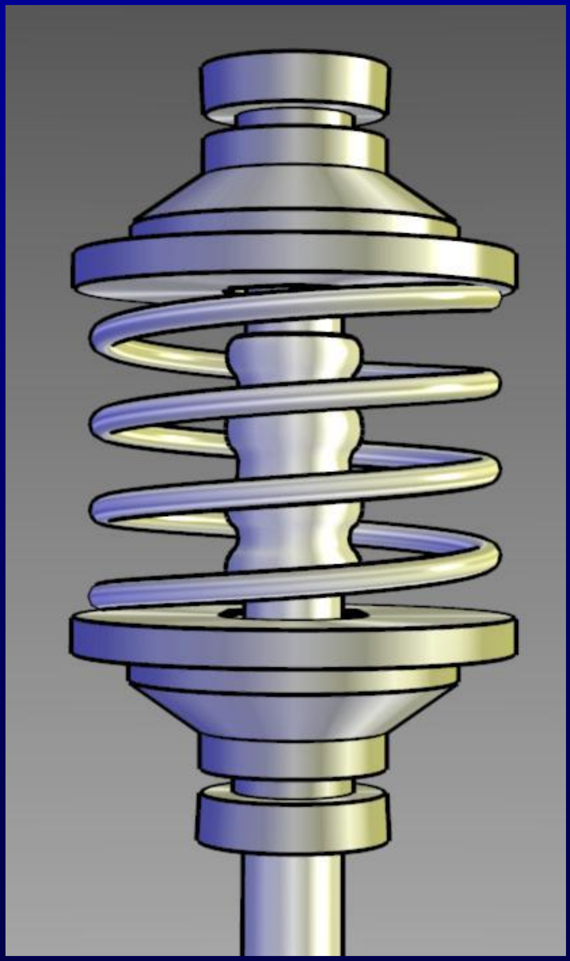
b = 0:55, γ = 0:3, α = 0:25, β = 0:5

Tone Shading Results



Metal Objects

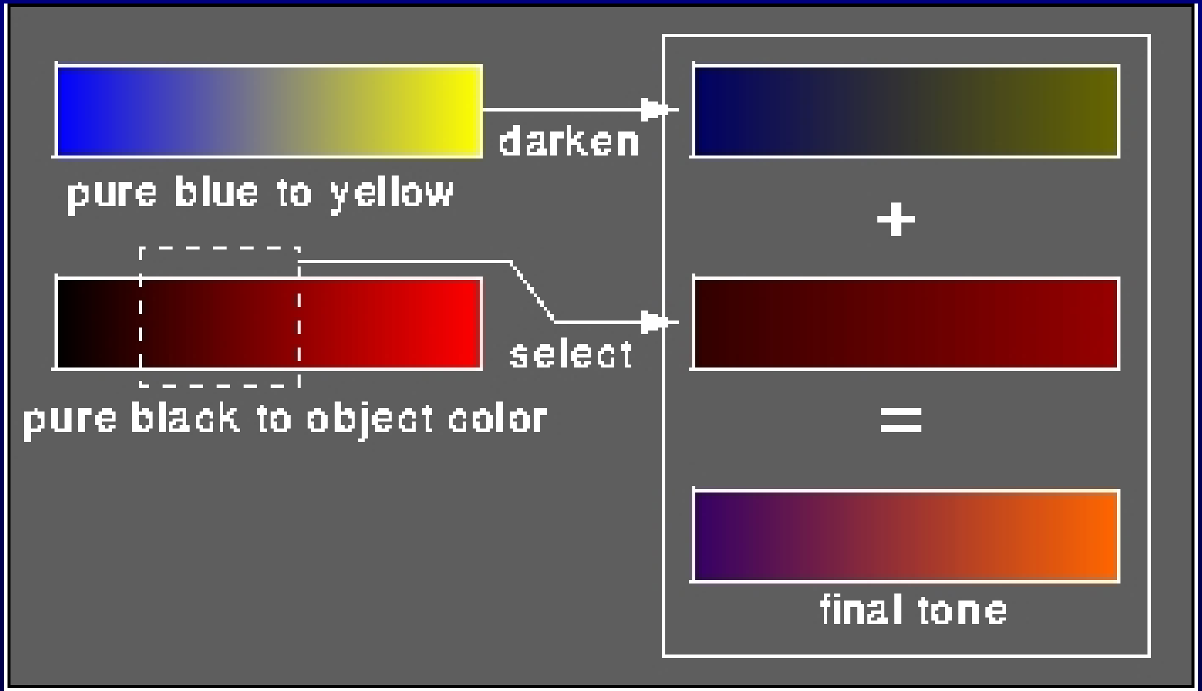




Hue and Luminance Shifts

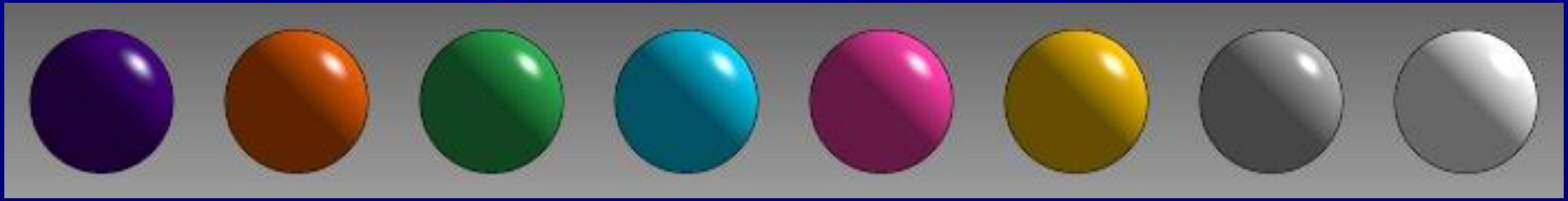
Amy Gooch, Bruce Gooch, Peter Shirley, Elaine Cohen. A Non-Photorealistic Lighting Model For Automatic Technical Illustration. Proc. SIGGRAPH 98.

http://www.cs.utah.edu/~gooch/SIG98/abstract.html

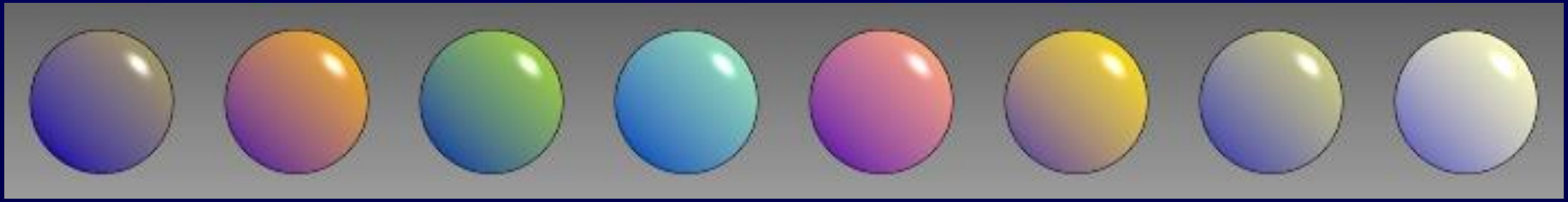


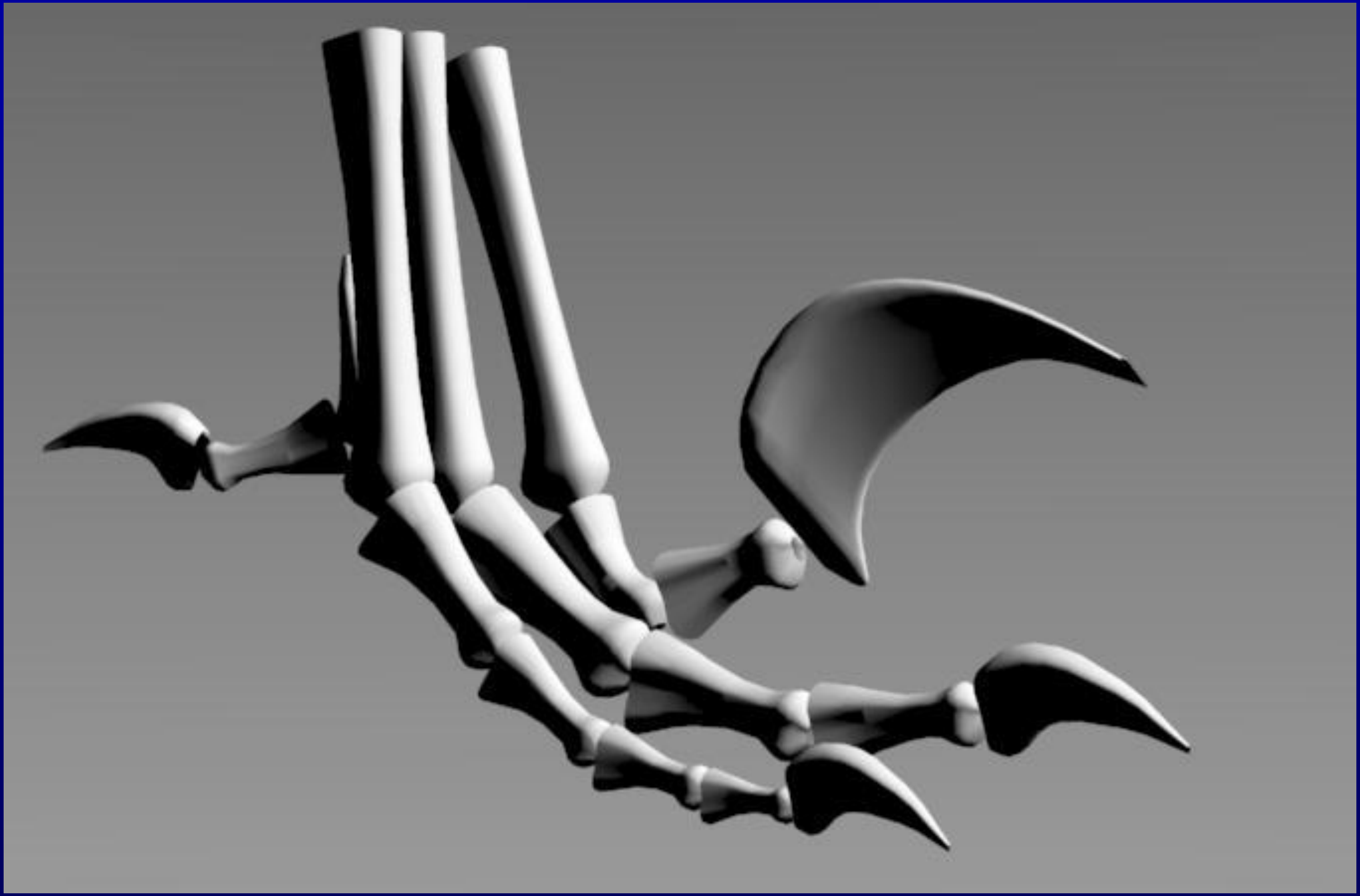
Hue and Luminance Shifts

- Phong shaded spheres with edges and highlights



- Spheres shaded with hue and luminance shift, and highlights and edges.





Diffuse shaded image. Black shaded regions hide details, especially in the small claws; edge lines could not be seen if added. Highlights and fine details are lost in the white shaded regions.

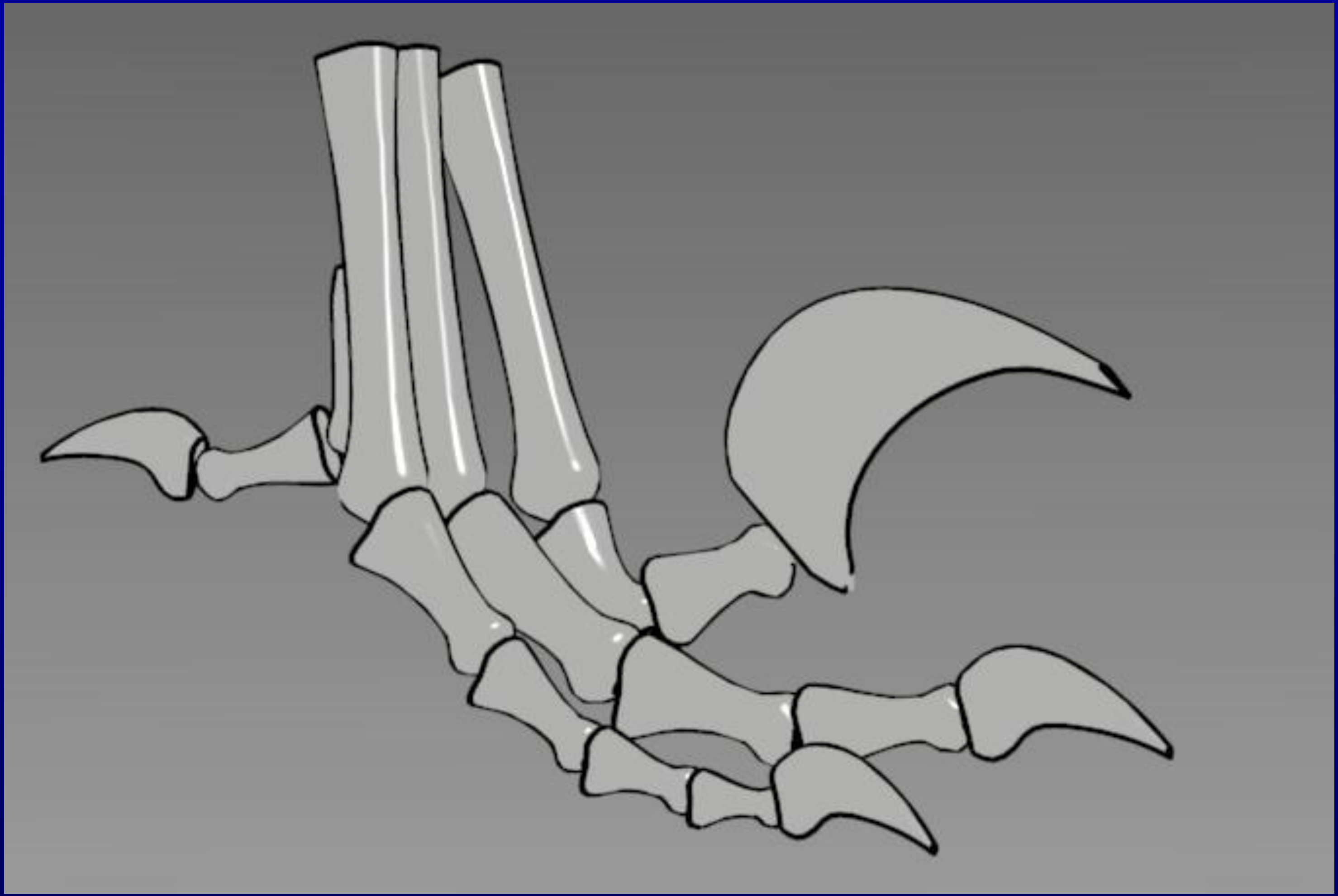
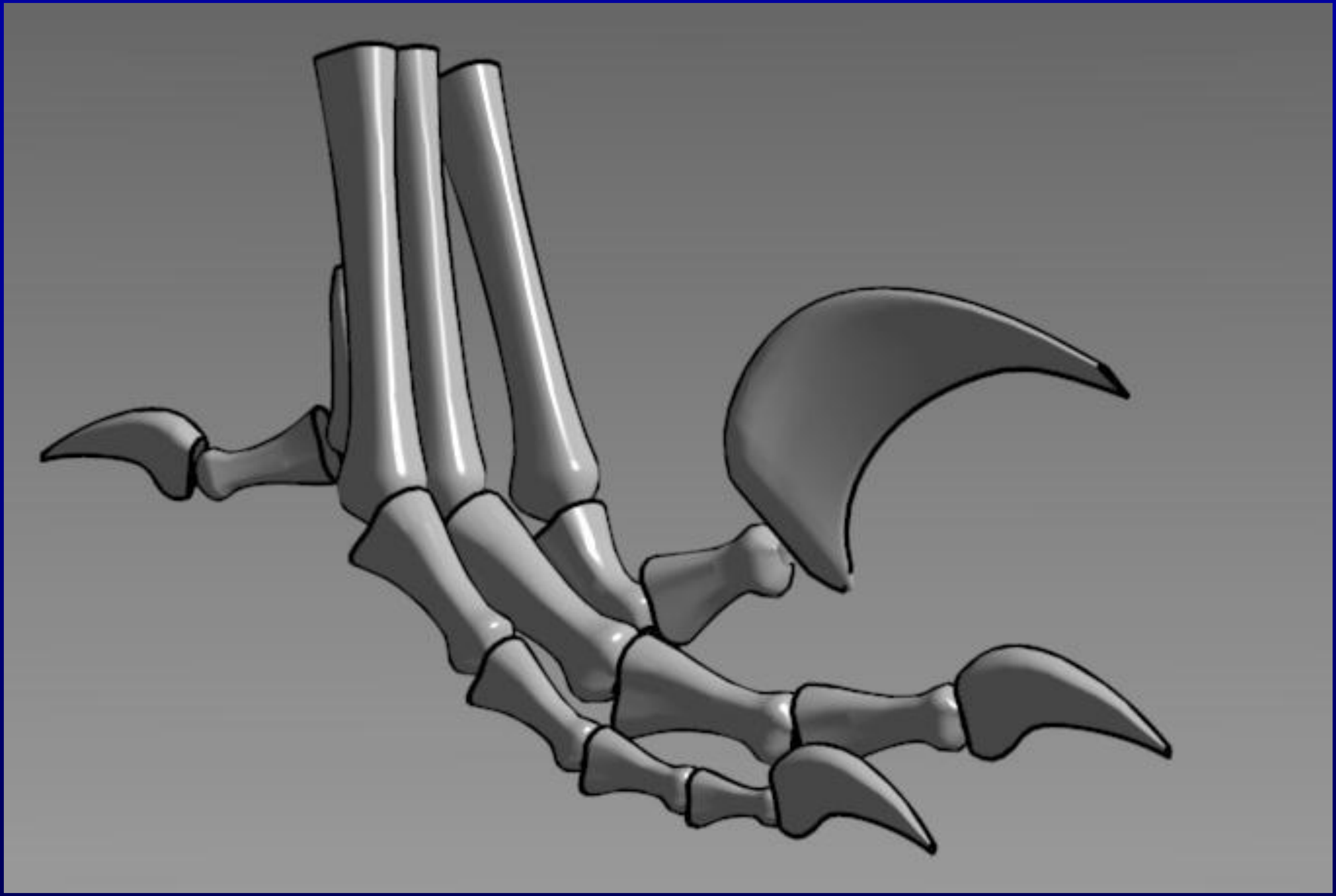
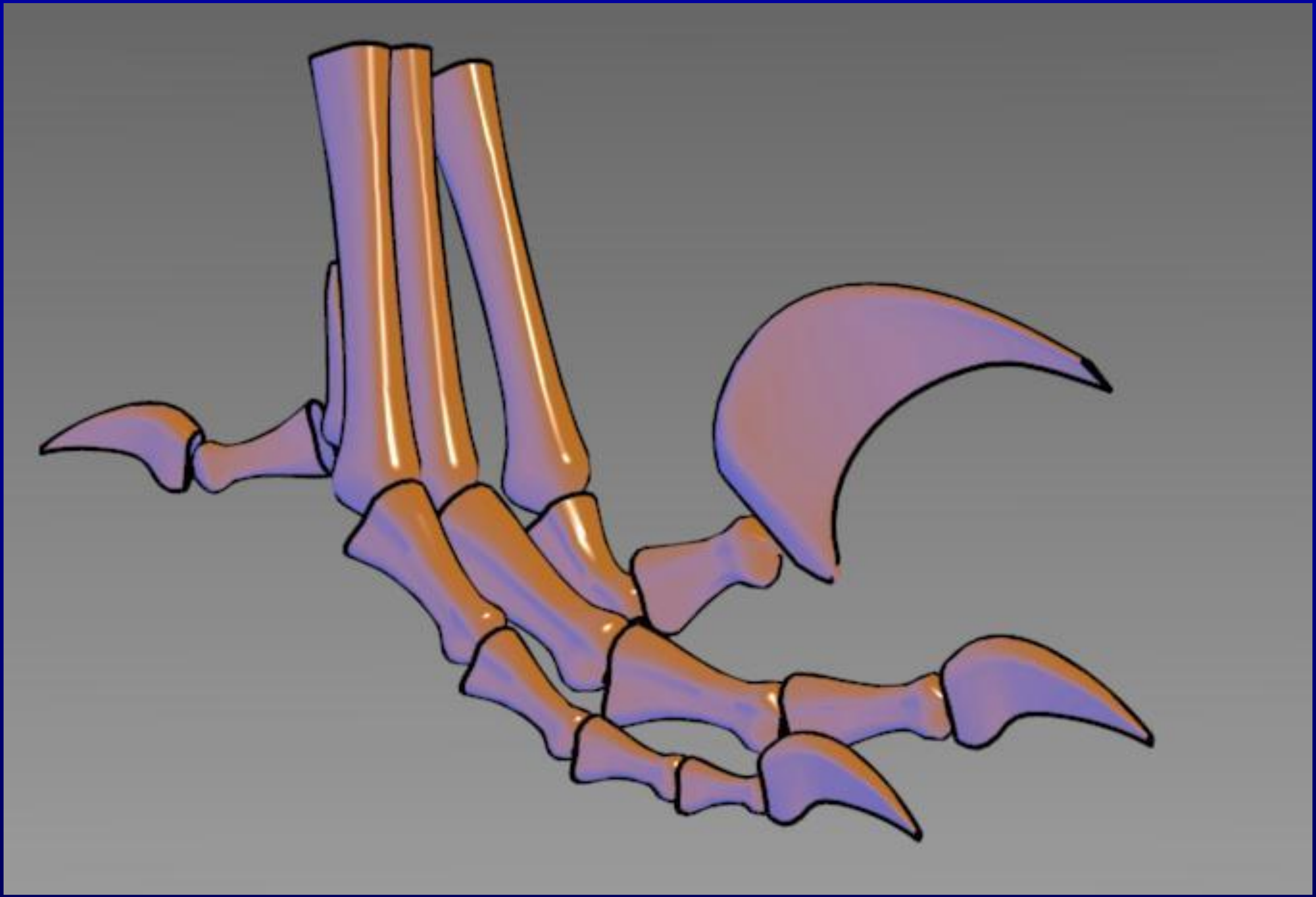


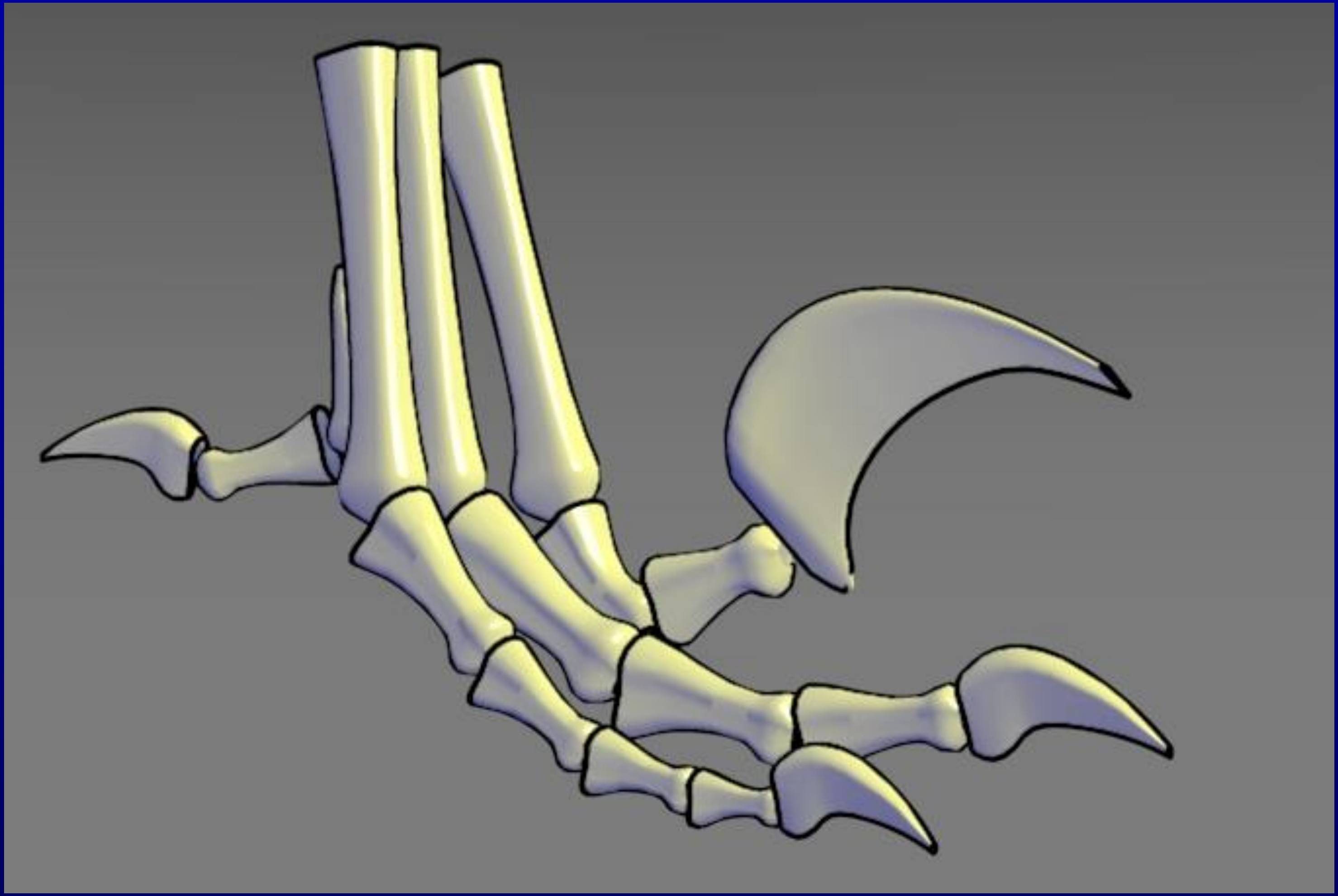
Image with only highlights and edges. The edge lines provide divisions between object pieces and the highlights convey the direction of the light. Some shape information is lost, especially in the regions of high curvature of the object pieces.



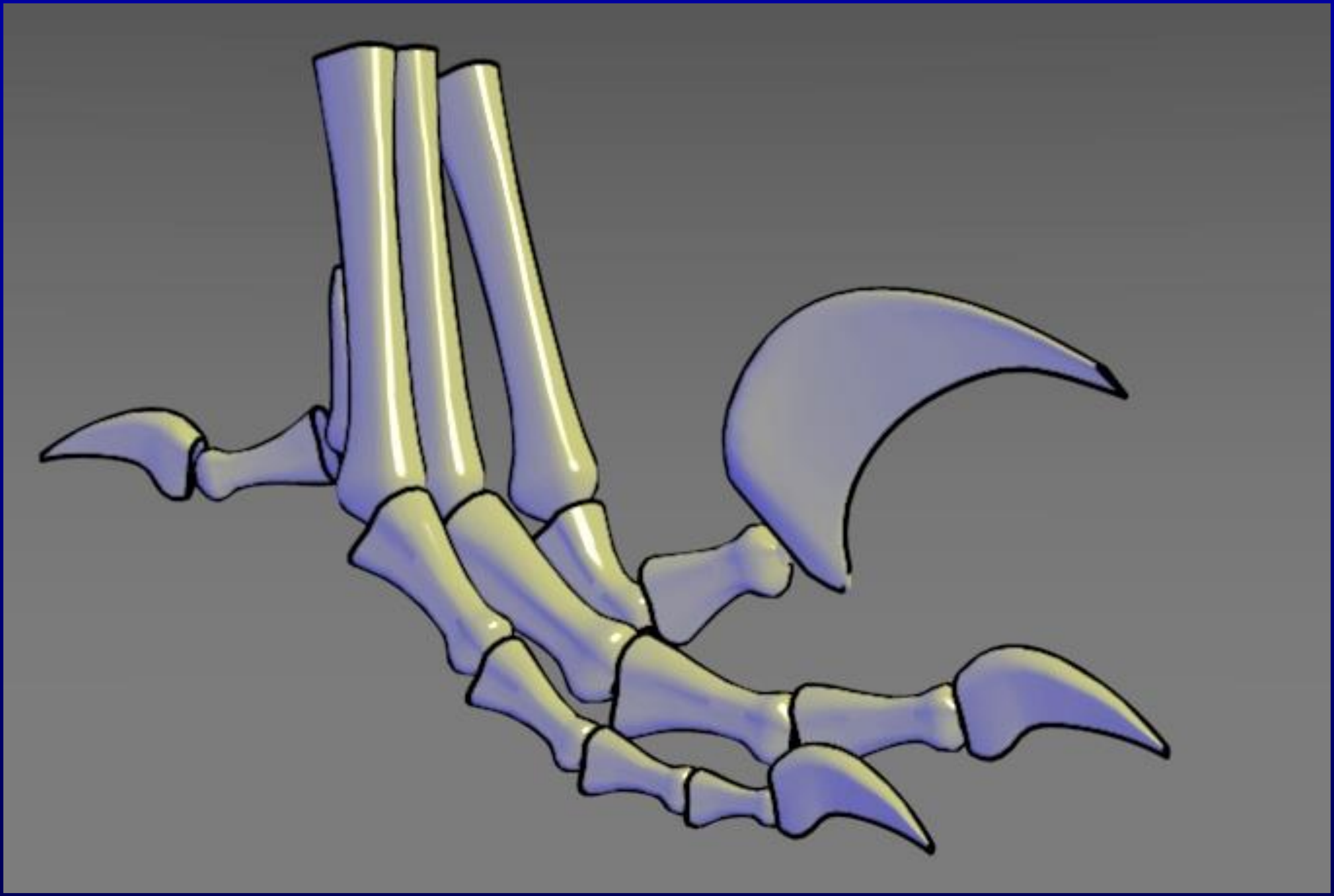
Phong shaded image with edge lines. Details are lost in the dark grey regions, especially in the small claws, where they are entirely ambient.



Approximately constant luminance tone rendering. Edge lines and highlights are clearly noticeable. Some details in shaded regions, like the small claws, are visible. The lack of luminance shift makes these changes subtle.



Luminance/hue tone rendering combining a luminance shift with a hue shift. Edge lines, highlights, fine details in the dark shaded regions such as the small claws, as well as details in the high luminance regions are all visible.



Alternative luminance/hue tone rendering with a different overall temperature shift, prominence of the object color, and strength of the luminance shift.