10 – surface shading

Scene Graphs (A2) (DAG)

Node (Object3D)

↳ children

↳ matrix

Group

Mesh

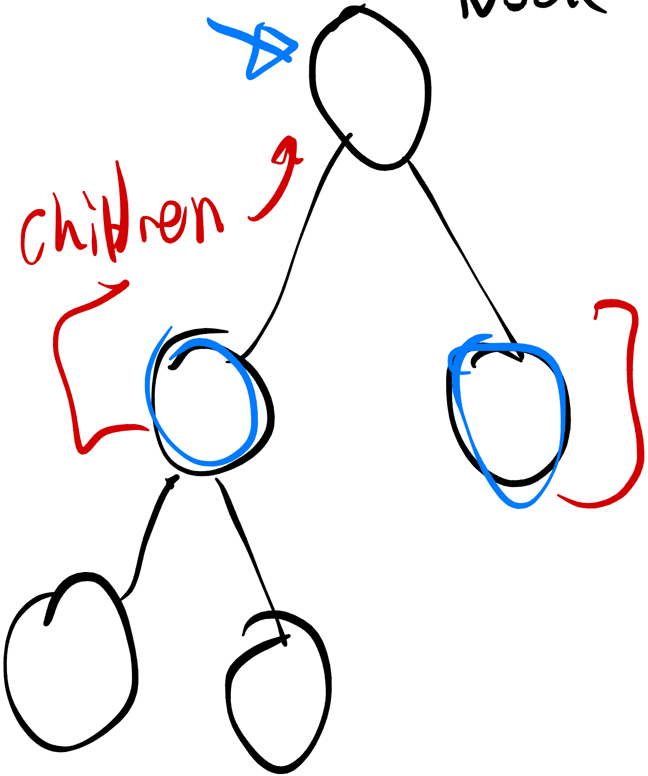
⋮

Unity

GameObject

children

transformation



Group rotate ...

Sphere scale(4, 1, 1)
translate(2, 0, 0)

Local origin
defined
by matrix

Illumination and Shading

determine color of point

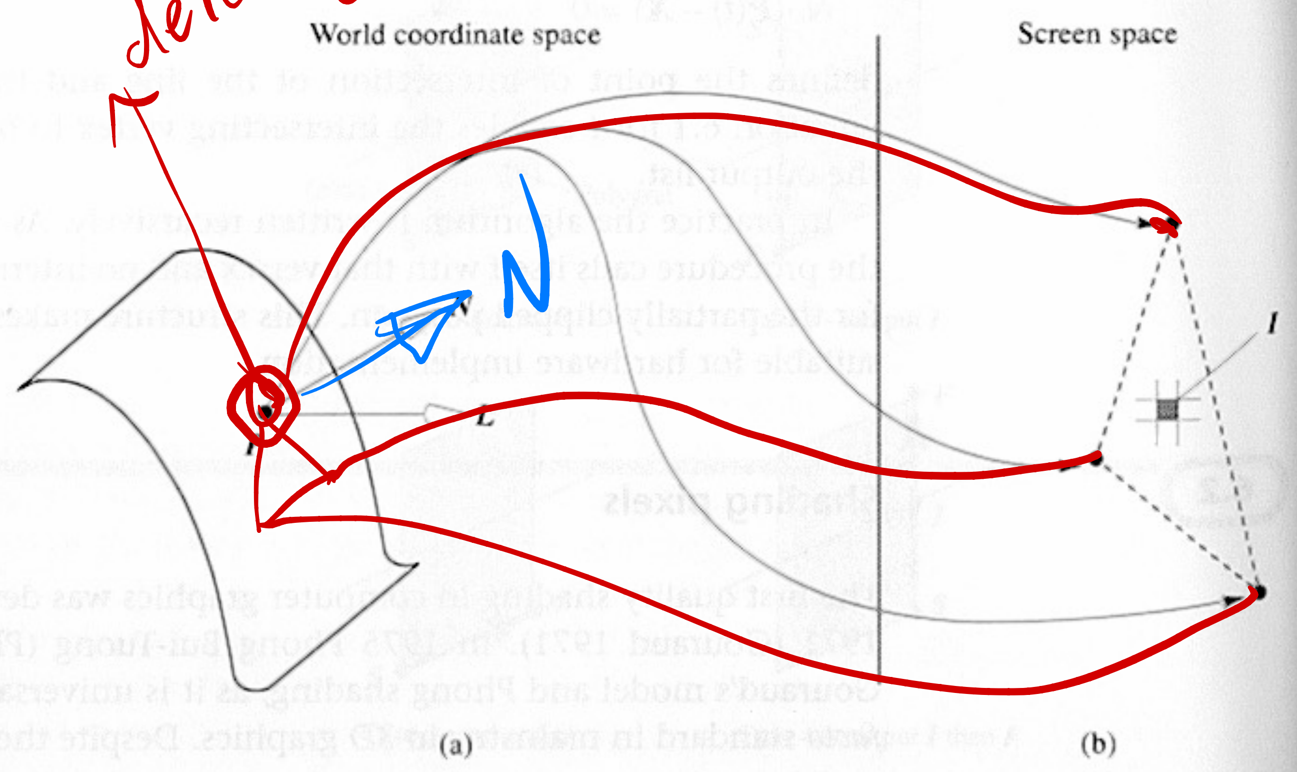


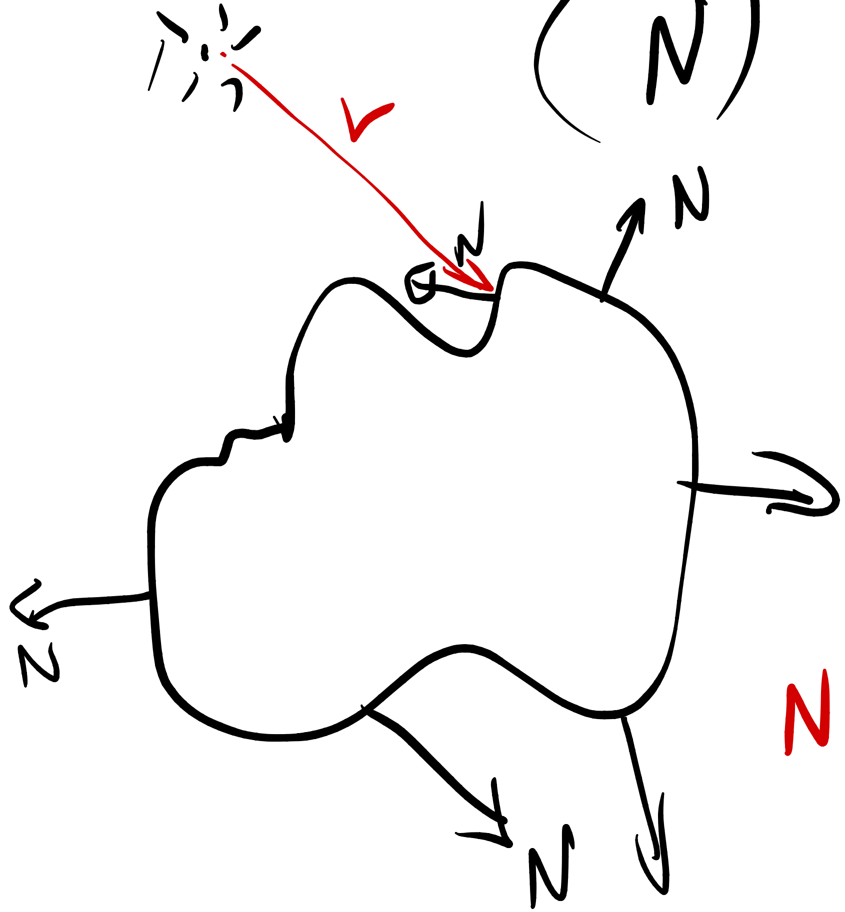
Figure 6.5
Illustrating the difference between local reflection models and shading algorithms. (a) Local reflection models calculate light intensity at any point ***P*** on the surface of an object. (b) Shading algorithms interpolate pixel values from calculated light intensities at the polygon vertices.

Surface Normals

(N)

→ is a vector perpendicular to the surface at a point

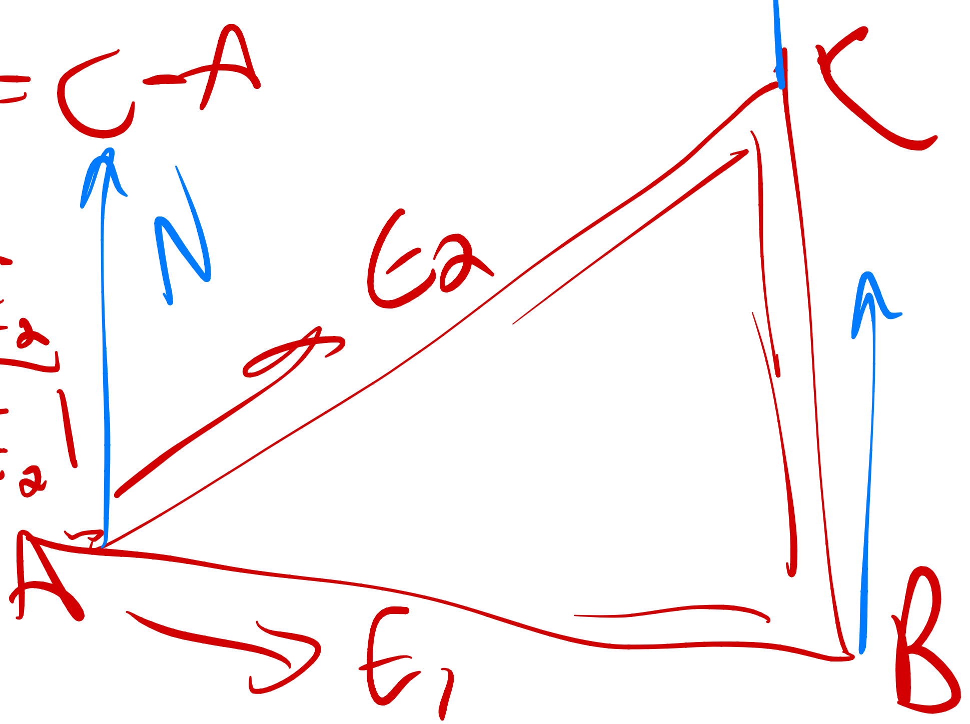
→ unit length



$E_1 = B - A$

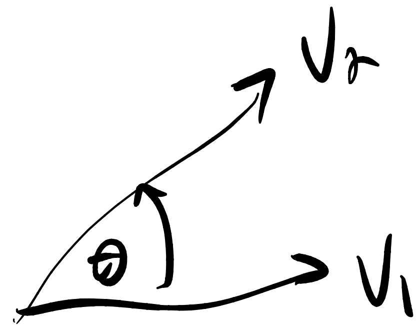
$E_2 = C - A$

$$N = \frac{E_1 \times E_2}{|E_1 \times E_2|}$$



Vector dot product: $V_1 \cdot V_2$ (unit length)

$$V_1 \cdot V_2 = \cos\theta$$



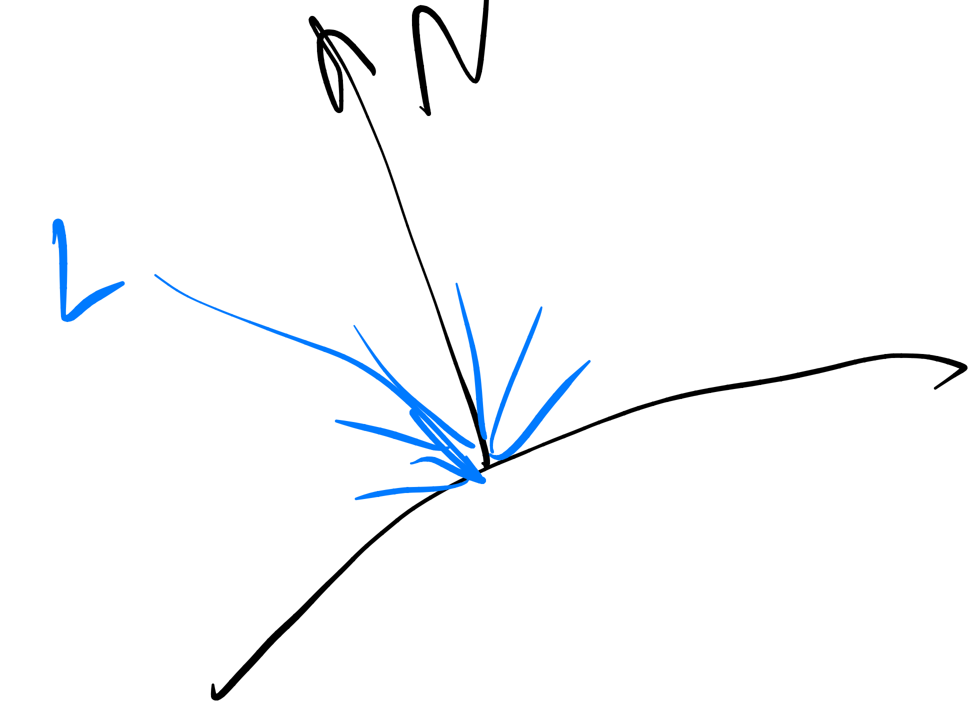
Illumination and Shading

- Illumination Models
  - Ambient
  - Diffuse
  - Attenuation
  - Specular Reflection
- Interpolated Shading Models
  - Flat, Gouraud, Phong
  - Problems

Surface Shading

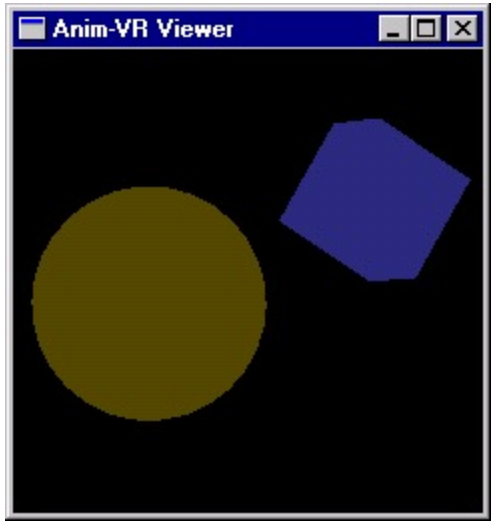
Two factors: light sources, material properties

Lambertian Surfaces



Illumination Models: Ambient Light

I = intensity



- Simple illumination model

  $$\boxed{I = k_i}$$

- Use nondirectional lights

  $$I = I_a k_a$$

  contribution to object color

- I_a = ambient light intensity
  (I_{aR}, I_{aG}, I_{aB})
- k_a = ambient-reflection coefficient
  (k_{aR}, k_{aG}, k_{aB})
- Uniform across surface

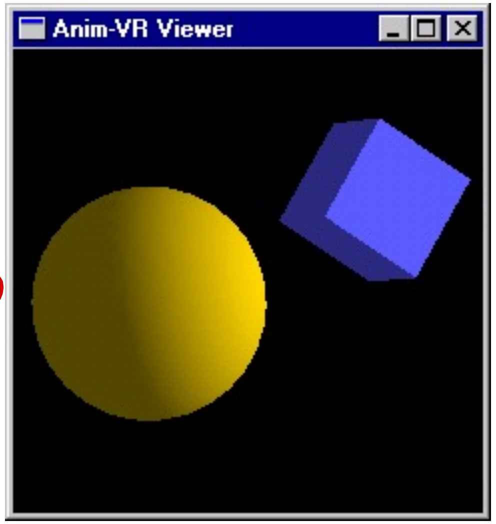
p = point light

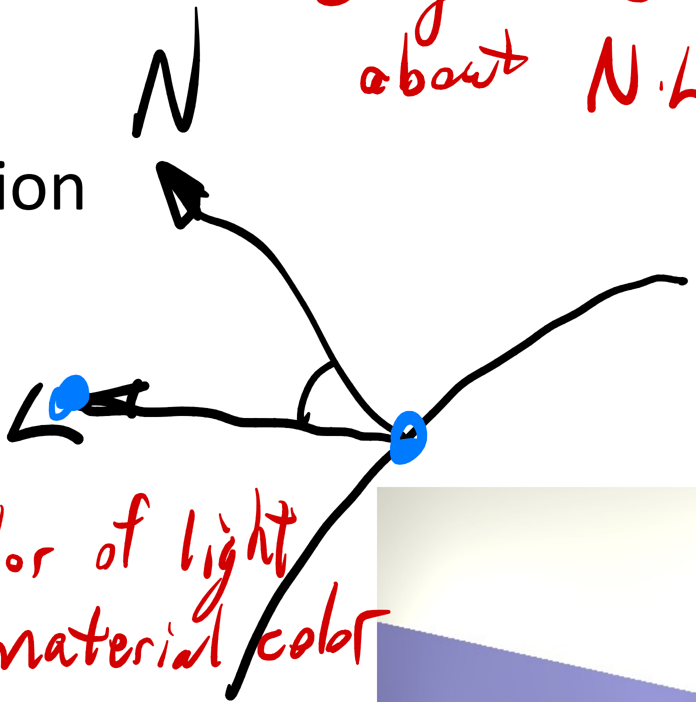
Small # of lights

Diffuse Light

$\max(0, N \cdot L)$

only care about $N \cdot L \geq 0$





- Account for light position
  - Ignore viewer position
- Proportional to cosΘ between *N* and *L*

$$I = I_p k_d \cos\Theta$$
$$= I_p k_d (N \cdot L)$$

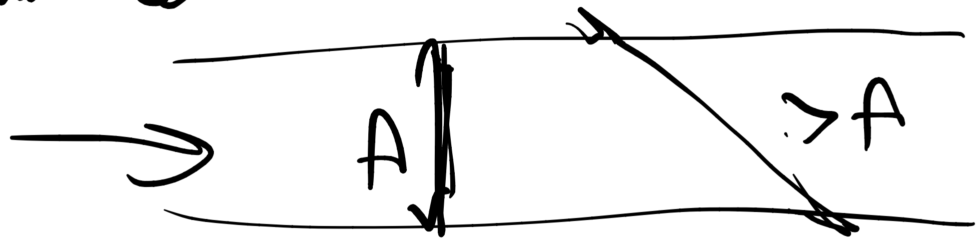
I_p = color of light

k_d = material color

- Model:

$$I = I_a k_a + I_p k_d (N \cdot L)$$

Radience



Point of this slide: We ~~usually~~ sometimes separate material into color + intensity

1 value or 3 values

(depending of library)

Again, Colored Lights

(slightly different, but equivalent, to book)

k_d ← Intensity of color

O_d ← just color

- O_d: diffuse color

  $$\boxed{O_d = (O_{dR}, O_{dG}, O_{dB})}$$

- Compute for each component
- i.e. red component is

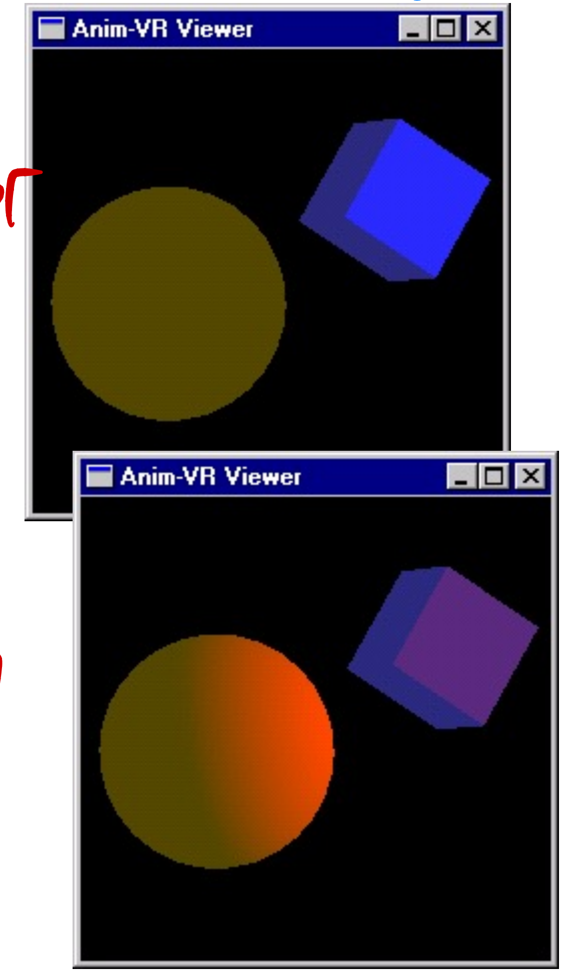
  $$I_R = I_{aR} k_{aR} O_{dR} + f_{att} I_{pR} k_d O_{dR} (N \cdot L)$$

- Note: use O_d for ambient and diffuse

$I_a k_a O_d$

Red

(1, .1, .1)



Light Intensity Values

- I_a, I_d
  - Represent intensity
  - Have R,G,B components
  - Do not need to fall in the 0..1 range!
    - Often need I_d>1
    - Final computed $I \leq 1$

~~color = (1, .1, .1)~~

color = (25, 2, 15)

Attenuation: Distance

k_d could be k_d (or could be $k_d O_d$)

- f_{att} models distance from light

  $I = I_a k_a + \boxed{f_{att}} I_p k_d \, (N \cdot L)$

- Realistic

  $f_{att} = 1/(d_L{}^2)$

- Hard to control, so often use

  $f_{att} = 1/(c_1 + c_2 d_L + c_3 d_L{}^2)$

  (c_1, c_2, c_3)

typically for scene

or light

Recall Reflectance Equation

$$L(\mathbf{x}, k_o) = \int_{k_i \in \Omega} L(\mathbf{x}, k_i)\, \rho(k_i, k_o) \cos\theta_i \, dk_i$$

Attenuation: Atmospheric (fog, haze)

far start

- z_f and z_b: near/far depth-cue plane → cur. $z_b \rightarrow z_f$
- s_f and s_b: scale factors — light intensity
- I_{dc}: depth cue color — z_0 0-1
- Given $z_f > z_0 > z_b$
  interpolate $s_f > s_0 > s_b$ → s_0 $s_b \ldots s_f$ → same range

$z_b \Rightarrow I$

$z_f = I_{dc}$

- Adjust intensity

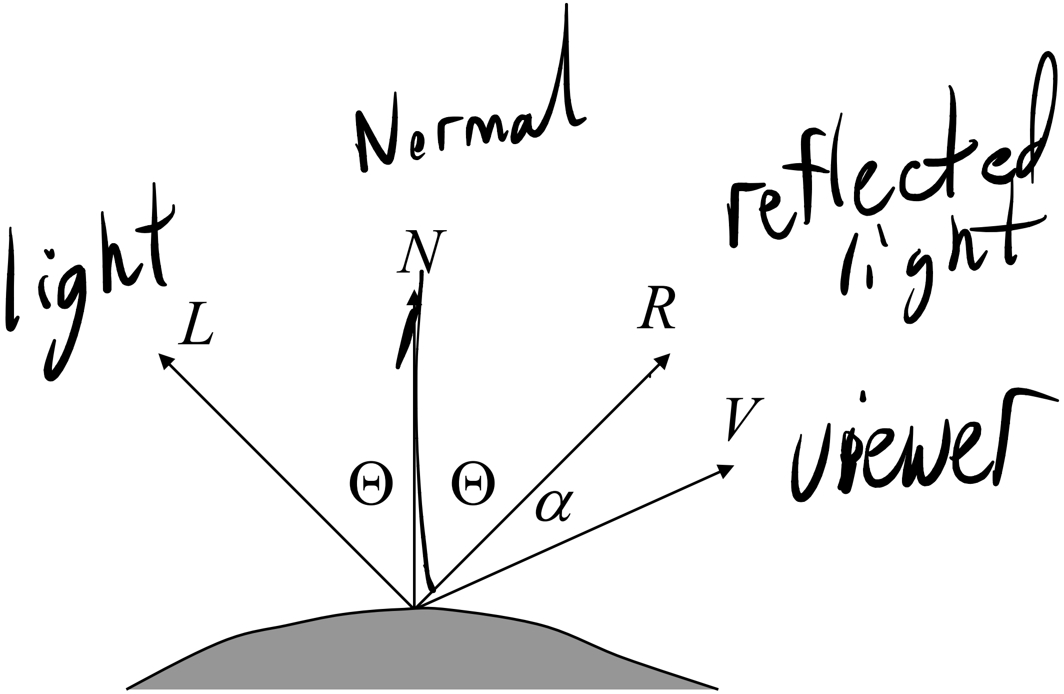
$$I' = \underbrace{s_0 I}_{s_b I} + (1 - s_0)I_{dc}$$

$I + s_f I_{dc}$

Specular Reflection: Phong Model

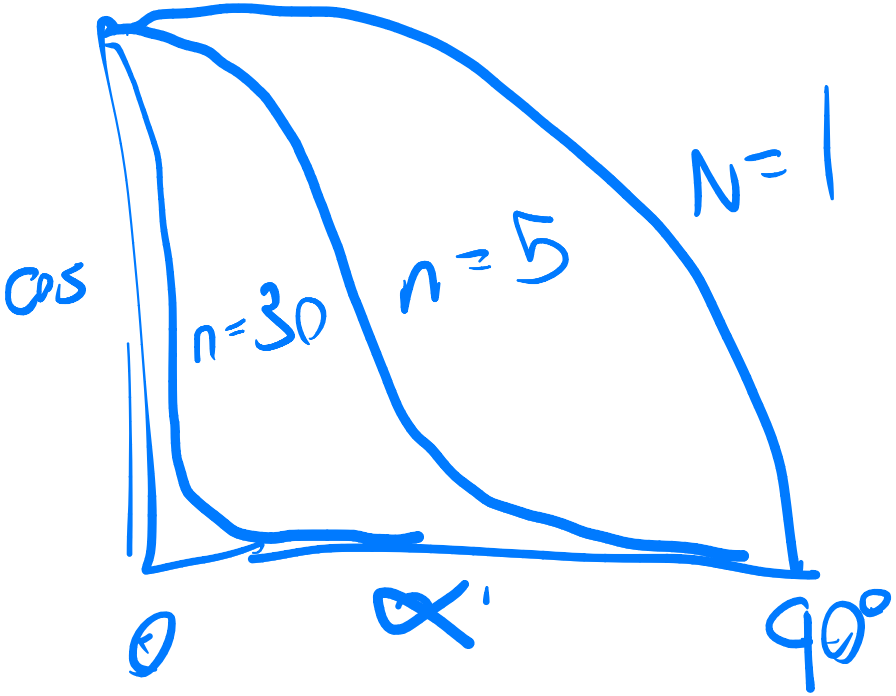
- Account for viewer position
  - Create highlights
- Based on $\cos^n \alpha = (R \cdot V)^n$
  - Larger *n*, smaller highlight
- k_s: specular reflection coef.

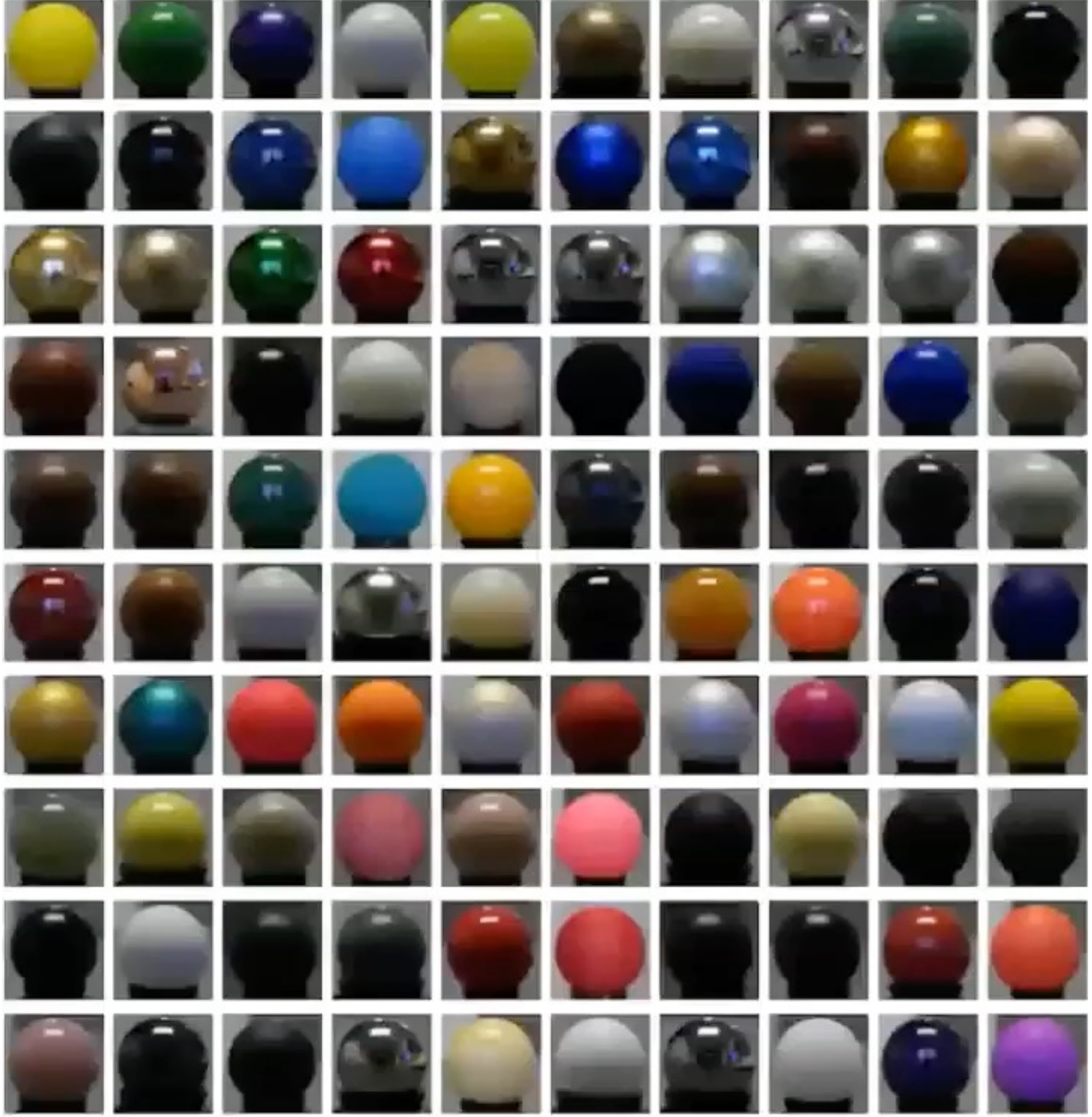
$$I = \underbrace{I_a k_a O_d + f_{att} I_p [\, k_d O_d (N \cdot L)}_{\text{previous slides}} + \boxed{k_s (R \cdot V)^n}\,]$$



Specular Power

$k_s (R \cdot V)^n$





Materials, Highlight Color

plastic

$k_s = (1, 1, 1)$ unintuitive

metal

$k_d\ (0.1, 0.1, 1)$

$k_s\ (0.5, 0.5, 1)$

Multiple Light Sources

I_a = ambient

color we've computed for a point

colors (r, g, b)

$0 \le r \le 1$

$0 \le g \le 1$

$0 \le b \le 1$

Obvious summation over *m* lights:

$$I = I_a k_a O_d + \sum_{1 \le i \le m} f_{att\,i} I_{pi} [k_d O_d (N \cdot L_i) + k_s (R_i \cdot V)^n]$$

material properties

$k_a O_d$ = ambient

$k_d O_d$ = diffuse

k_s = specular

"color"

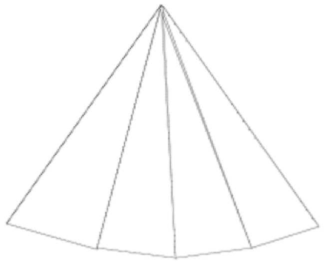
Shading Models

Surface color in this model = ambient + diffuse + specular

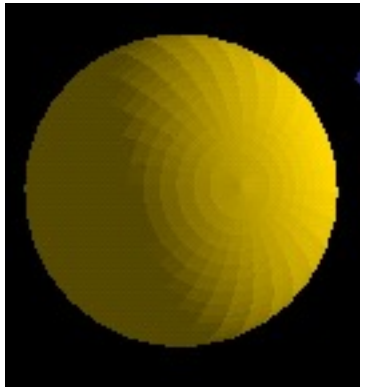
To shade triangles:

1) Per Triangle
2) Per Vertex
3) Per Pixel

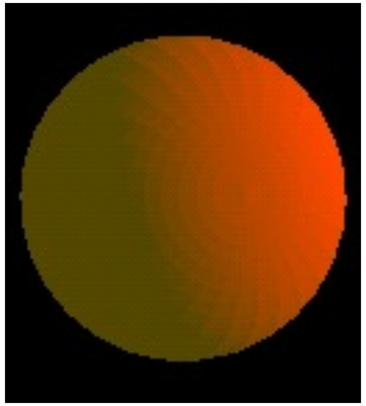
Shading Models: Per Triangle (Flat Shading)



- Compute one color for polygon
  - Use polygon normal in lighting eqs.
- Every pixel is assigned same color

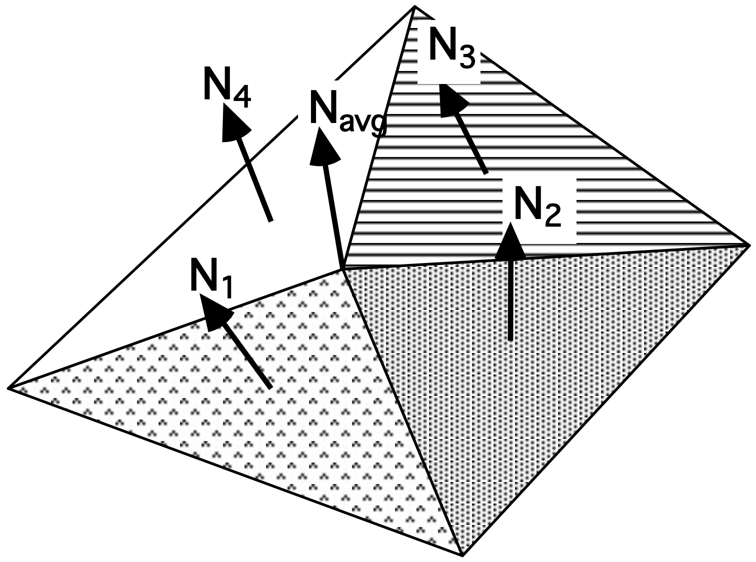


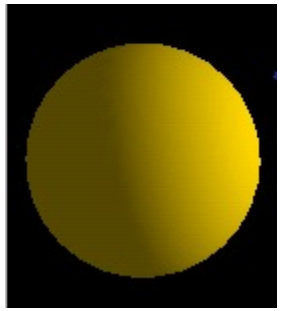
- Fast and simple
- Shade of polygons independent

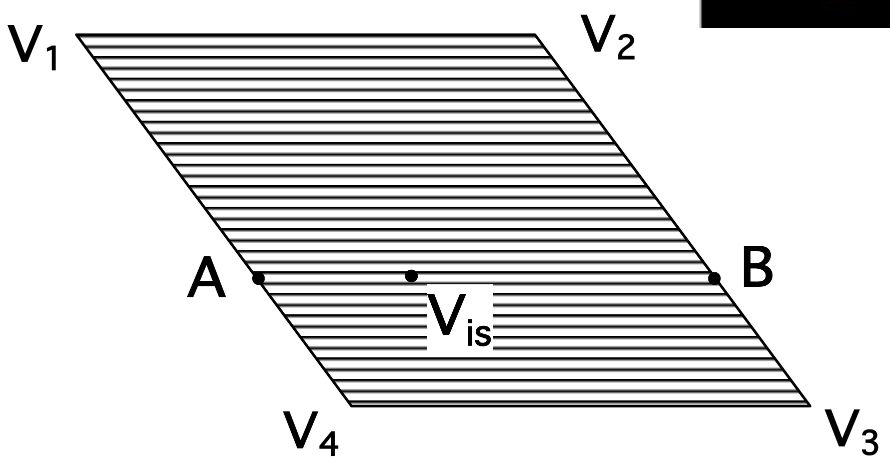


Shading Models: Per Vertex (Gouraud Shading)

- Compute vertex normals
  - Average normals of abutting polygons
- Use vertex normal in lighting eqs.
- Linearly interpolate vertex intensities
  - Along edges
  - Along scan lines







Gouraud Shading

Often appears dull, chalky

- Lacks accurate specular component
  - If included, will be averaged over entire polygon

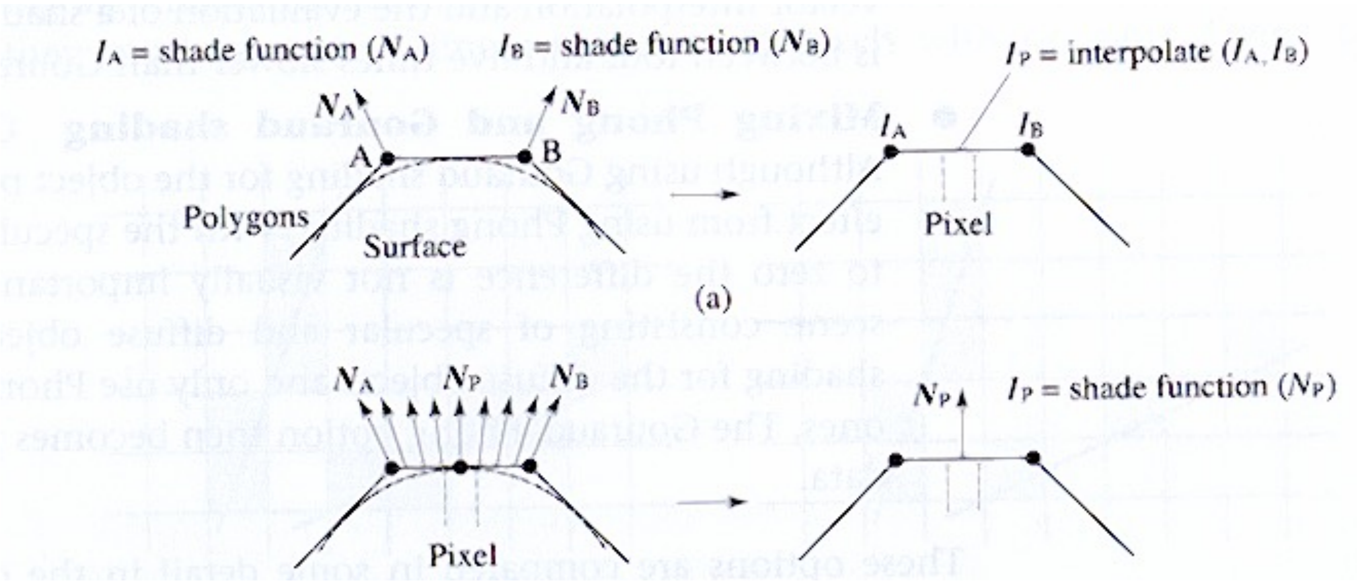
Flat Shading

Mach banding

- Artifact at discontinuities in intensity or intensity slope

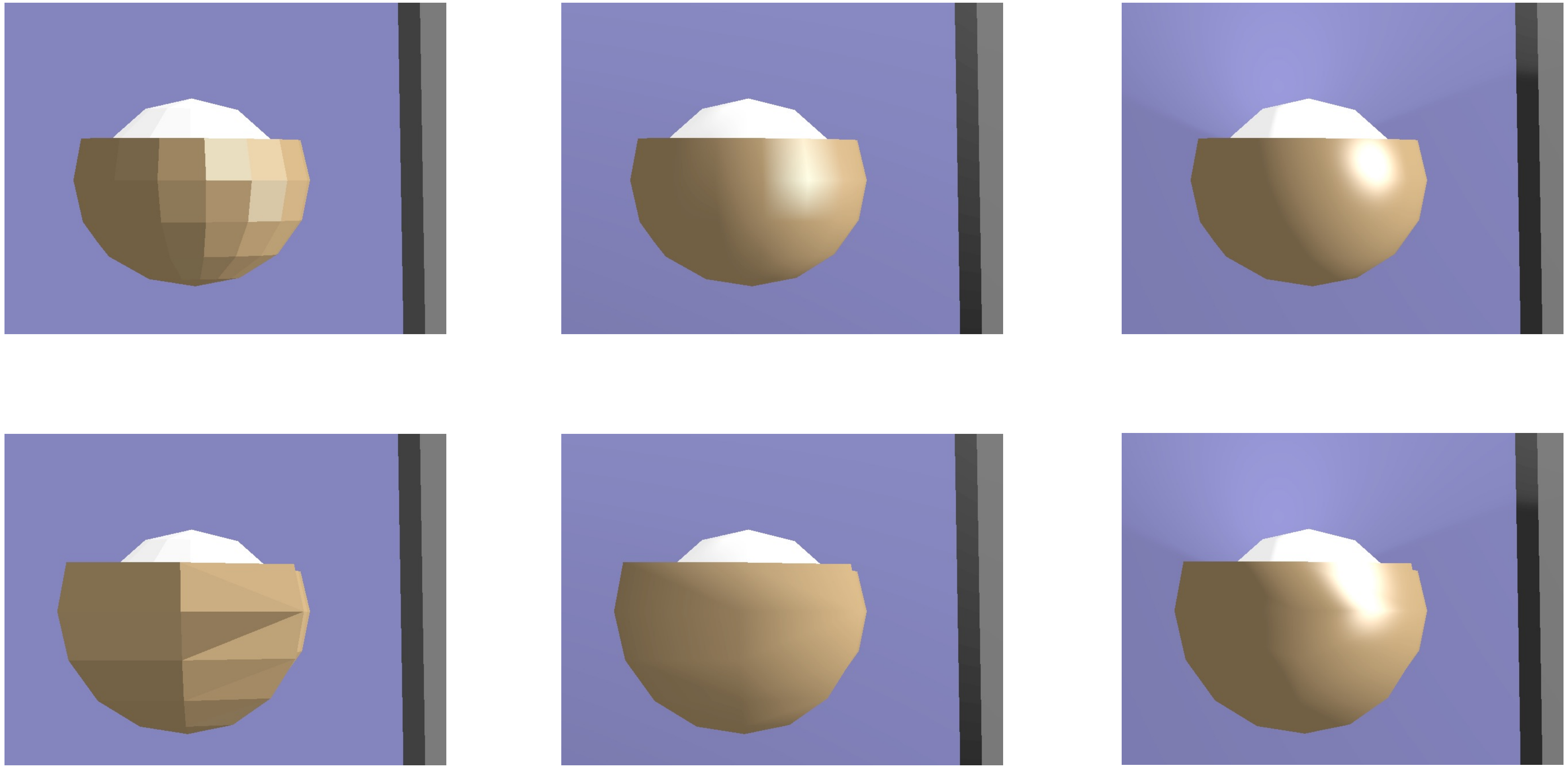
Shading Models: Per Pixel (Phong Shading)

- Linearly interpolate vertex normals
  - Compute lighting eqs. at each pixel
    - Normals must be backmapped to WC



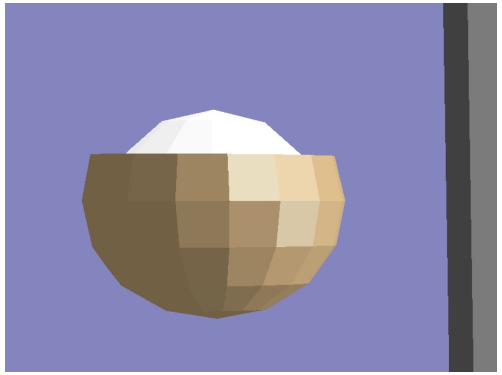
- Can use specular component

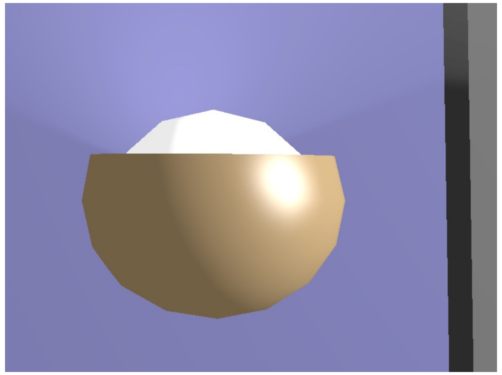
Closeup: Flat, Gouraud, Phong



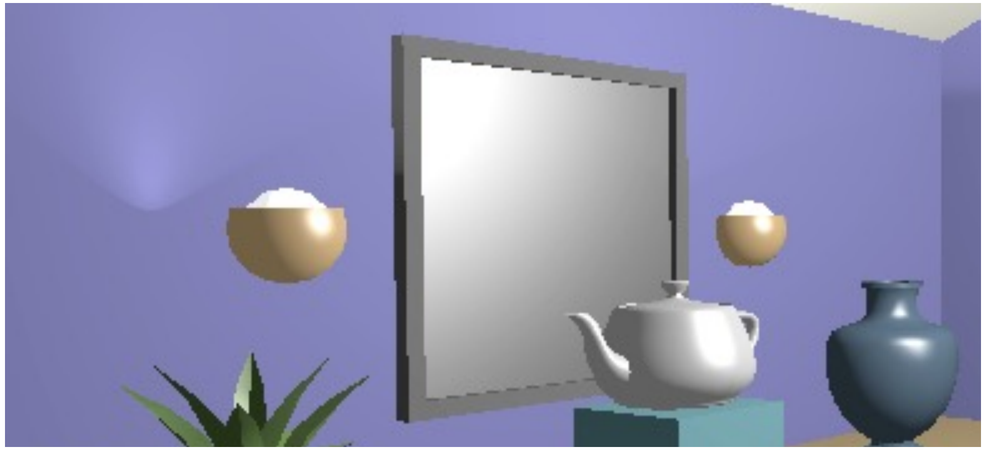
Problems with Interpolated Shading

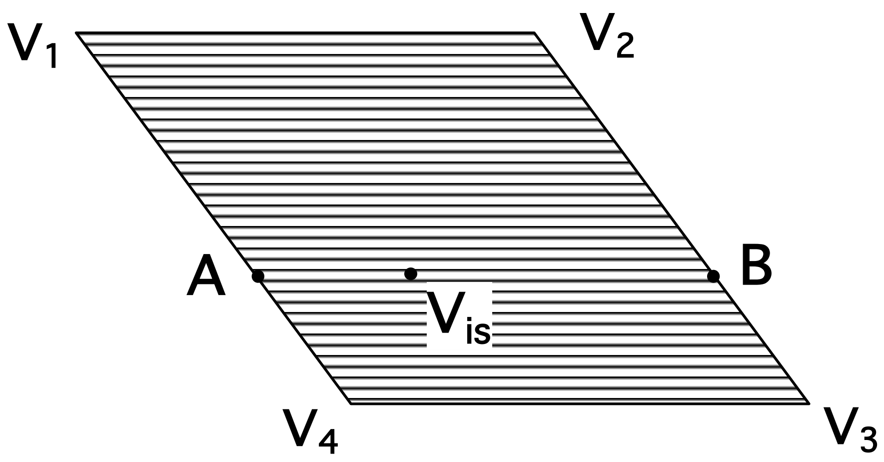
- Polygonal silhouette





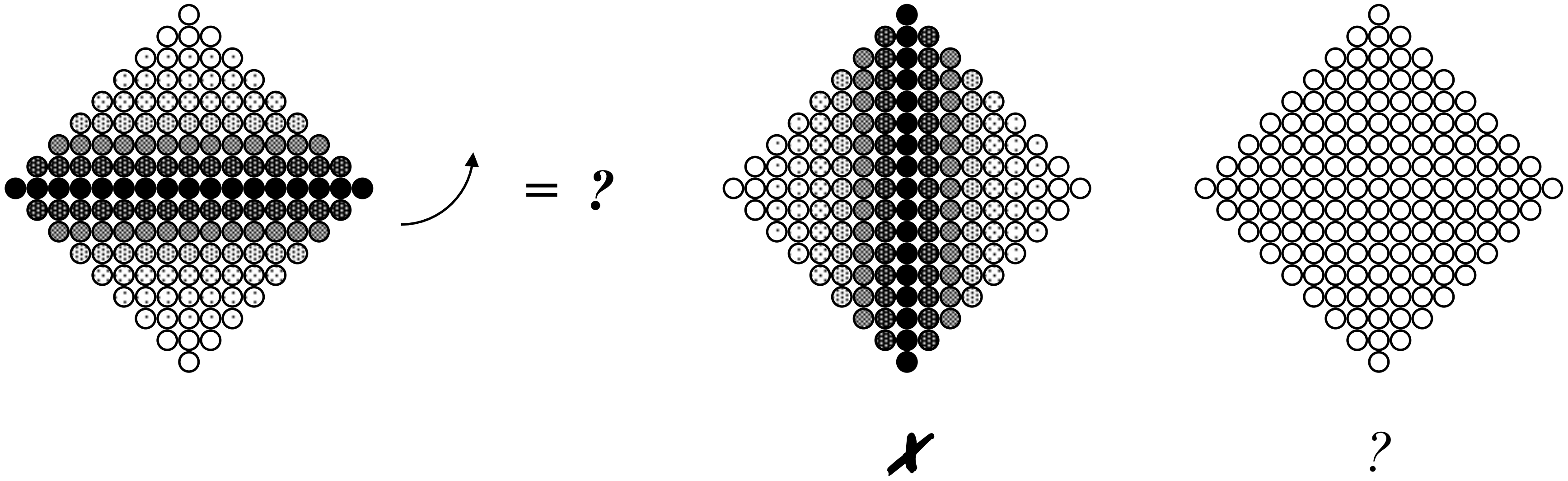
- Perspective distortion





Problems with Interpolated Shading

- Scanline/orientation dependent
  - Creates temporal aliasing when used to render animation frames:



Problems with Interpolated Shading

- Shared vertices
- Unrepresentative vertex normals
  - Missed specular highlights
  - Missed geometry