

Standard nearby point source model (Lambertian reflection)

$$\varrho_d(x) \frac{N(x) \cdot S(x)}{r(x)^2}$$

- N is the illuminated surface normal
- ρ is diffuse albedo
- S is source vector - a vector from x to the source, whose length is the intensity term
 - works because a dot-product is basically a cosine
- $r(x)$ is distance from surface point to source --- term occurs because source “looks smaller” as we move away--or, alternatively, its energy is spread out over a larger surface.

Standard distant point source model

- Nearby point source gets bigger if one gets closer, but the effect for far away points is negligible (e.g. the sun).
- Assume that all points in the model are close to each other with respect to the distance to the source. Then the source vector doesn't vary much (and can be assumed to not vary with x ---ignoring the possibility that it is occluded), and the distance doesn't vary much either, and we can roll the constants together to get:

$$\square_d(x) \left(N(x) \bullet S_d \right)$$



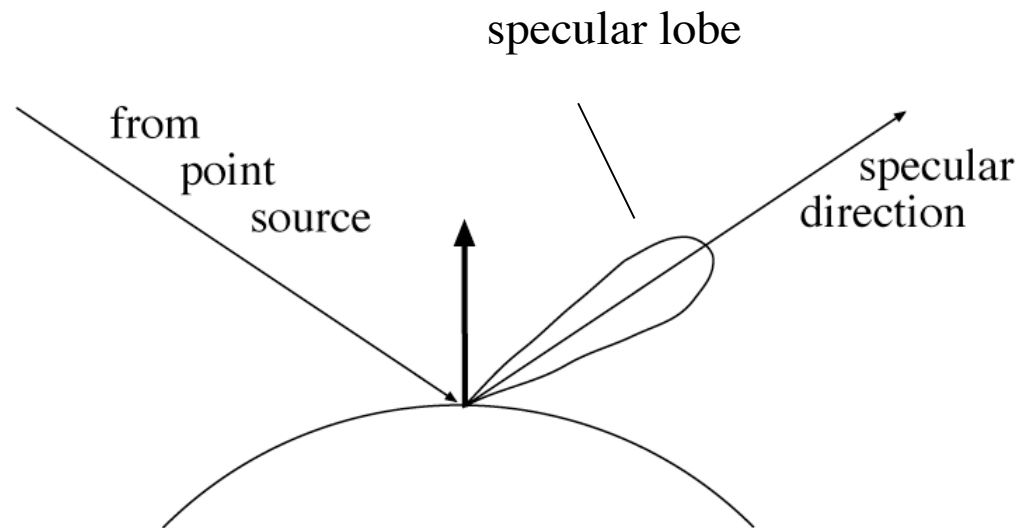
Diffuse Lighting

from

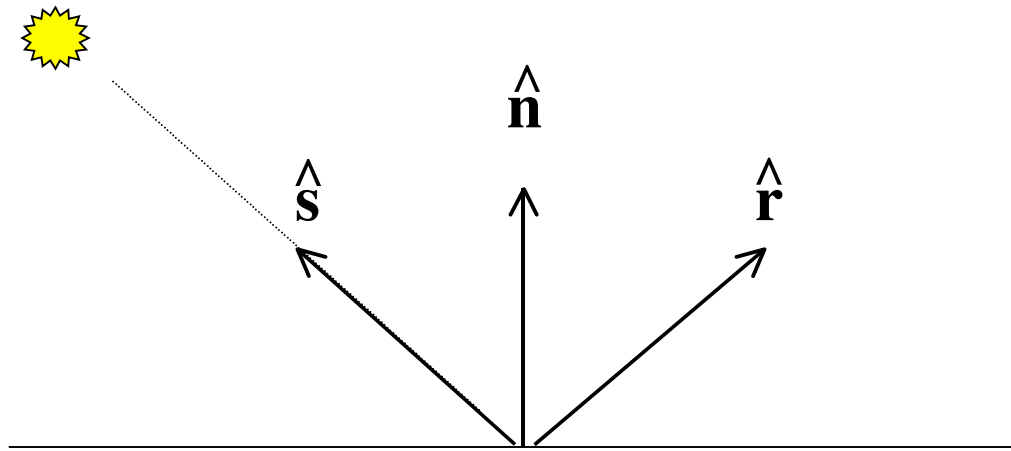
<http://www.geocities.com/SiliconValley/Horizon/6933/shading.html>

Specular surfaces

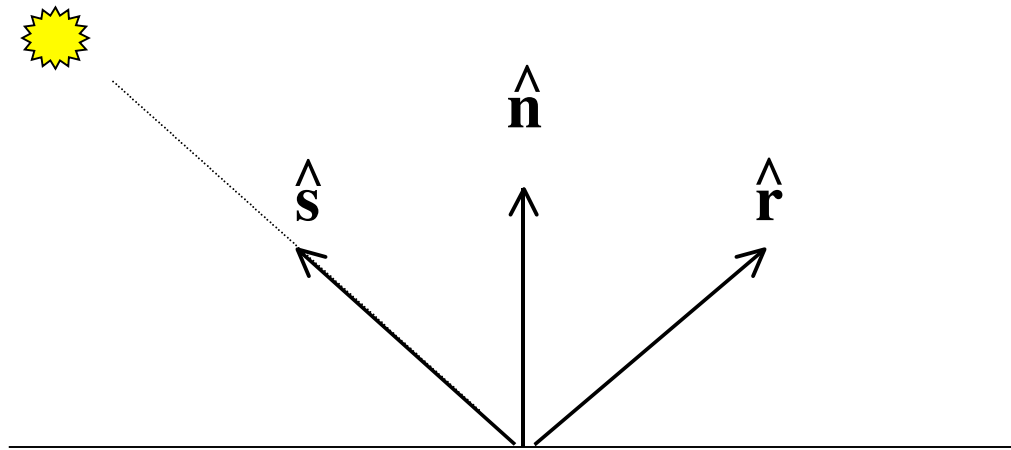
- Another important class of surfaces is specular (somewhat mirror-like).
 - specular surfaces reflect a significant amount of energy in the specular (mirror) direction
 - a significant amount may also be reflected in a direction roughly in the mirror direction (specular lobe)
 - typically there is a diffuse component as well
 - writing a BRDF approximation is possible, but beyond the scope of this course



Computing reflection (specular) direction



Computing reflection (specular) direction



$$\hat{\mathbf{s}} + \hat{\mathbf{r}} = k\hat{\mathbf{n}}$$

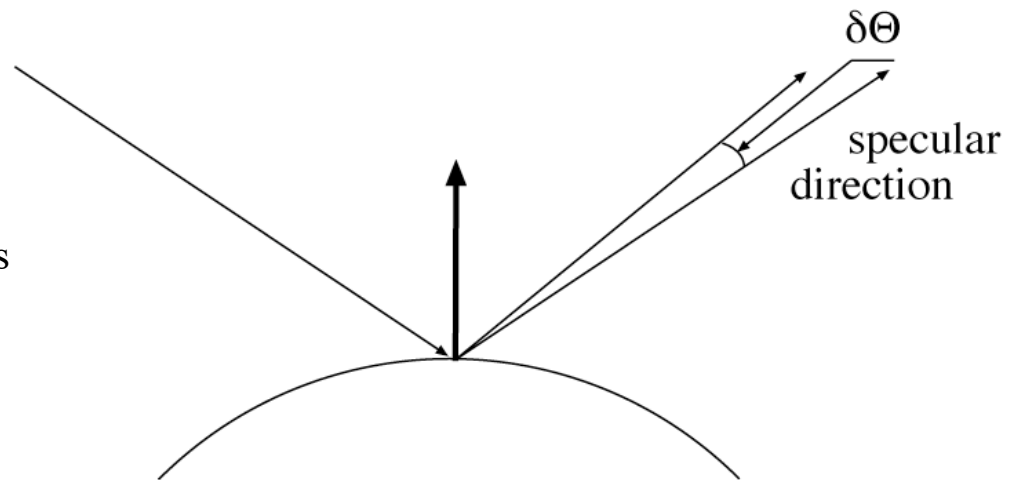
$$\hat{\mathbf{n}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{n}} \cdot \hat{\mathbf{r}} = k \quad k = 2\hat{\mathbf{n}} \cdot \hat{\mathbf{s}}$$

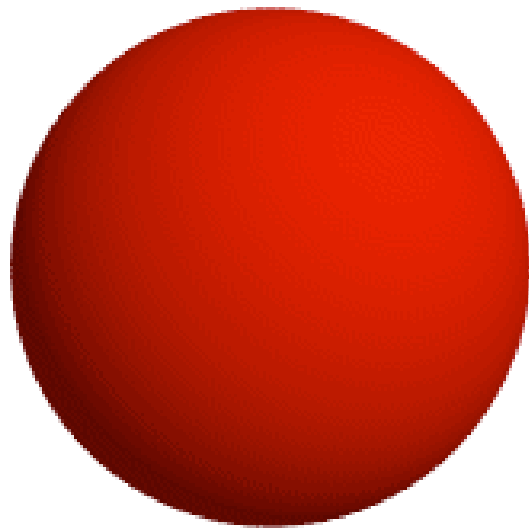
$$\text{So } \hat{\mathbf{r}} = 2(\hat{\mathbf{n}} \cdot \hat{\mathbf{s}})\hat{\mathbf{n}} - \hat{\mathbf{s}}$$

Phong's model of specularities

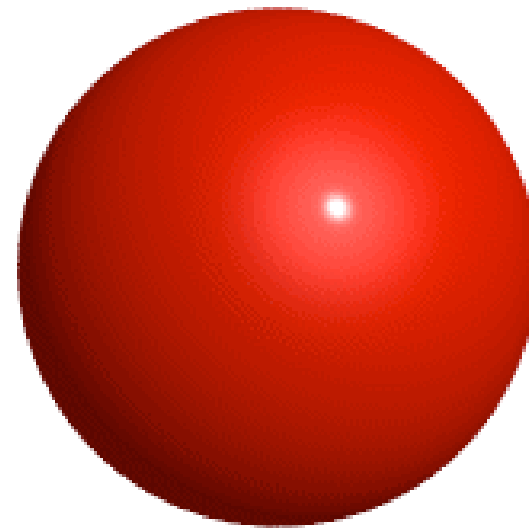
- There are very few cases where the exact shape of the specular lobe matters.
- Typically:
 - very, very small --- mirror
 - small -- blurry mirror
 - bigger -- see only light sources as “specularities”
 - very big -- faint specularities
- Phong's model
 - reflected energy falls off with

$$\cos^n(\theta)$$





Diffuse Lighting



Plus Specular Highlight

from

<http://www.geocities.com/SiliconValley/Horizon/6933/shading.html>

Simple shading model

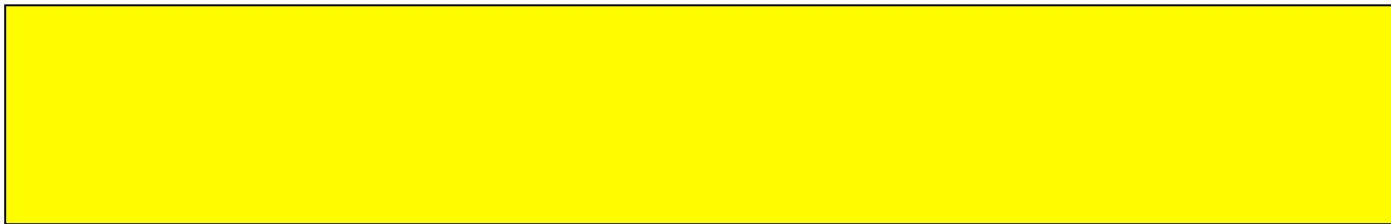
- Assume that all surface radiance is due to sources alone
 - i.e. both diffuse and specular, but no exitance.
- Can use standard point source model for diffuse term
 - either nearby or at infinity
- Common simplification:
 - drop $1/r^2$ term from nearby point source (still have direction variation)
- Intensity = Diffuse intensity due to sources + specular term due to sources
- Multiple sources are added up (or integrated over)
- For some special cases, the integral can be estimated in advance

Line sources



Radiosity due to line source varies with inverse distance, if the source is long enough (derivation is through integration of the contributions along the line)

General extended sources



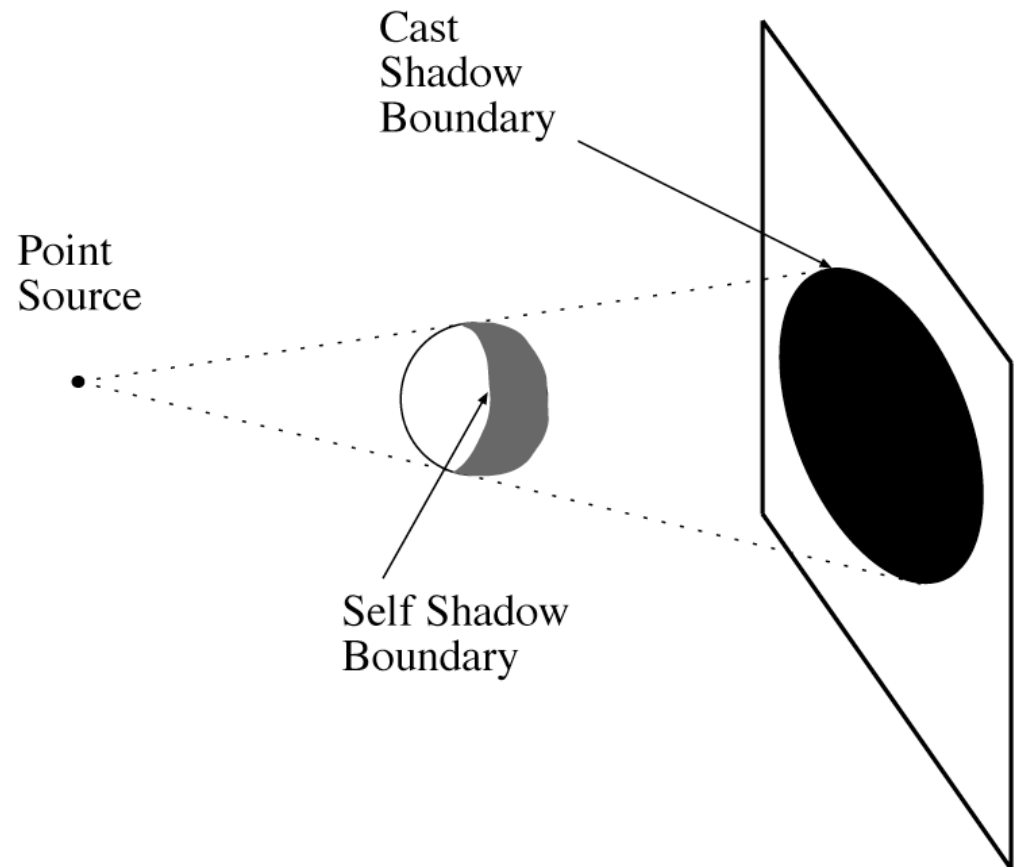
Can be handled by doing the integration (we won't)

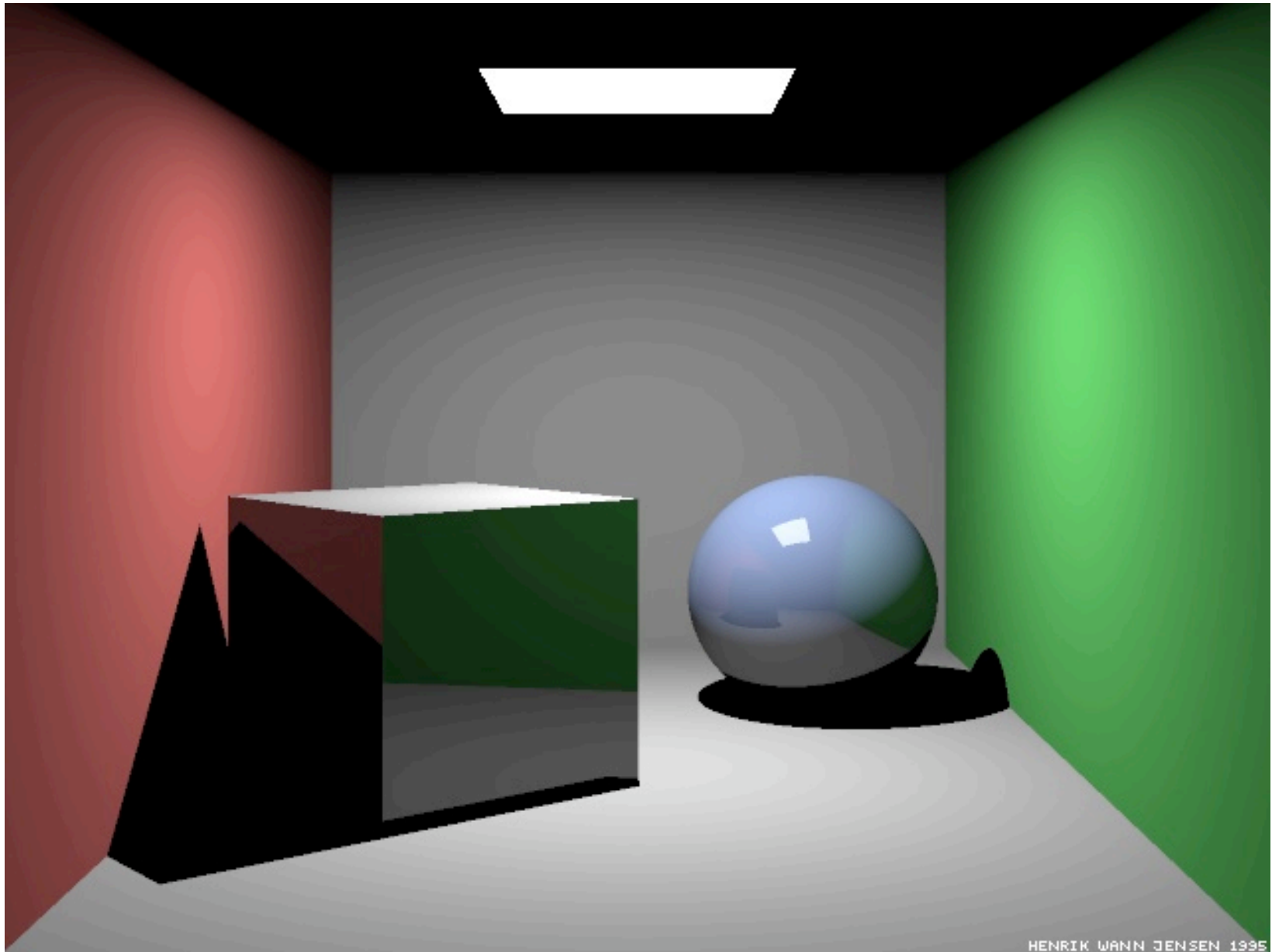
What if the source is large relative to the distance to it?

How about the hemisphere of the sky?

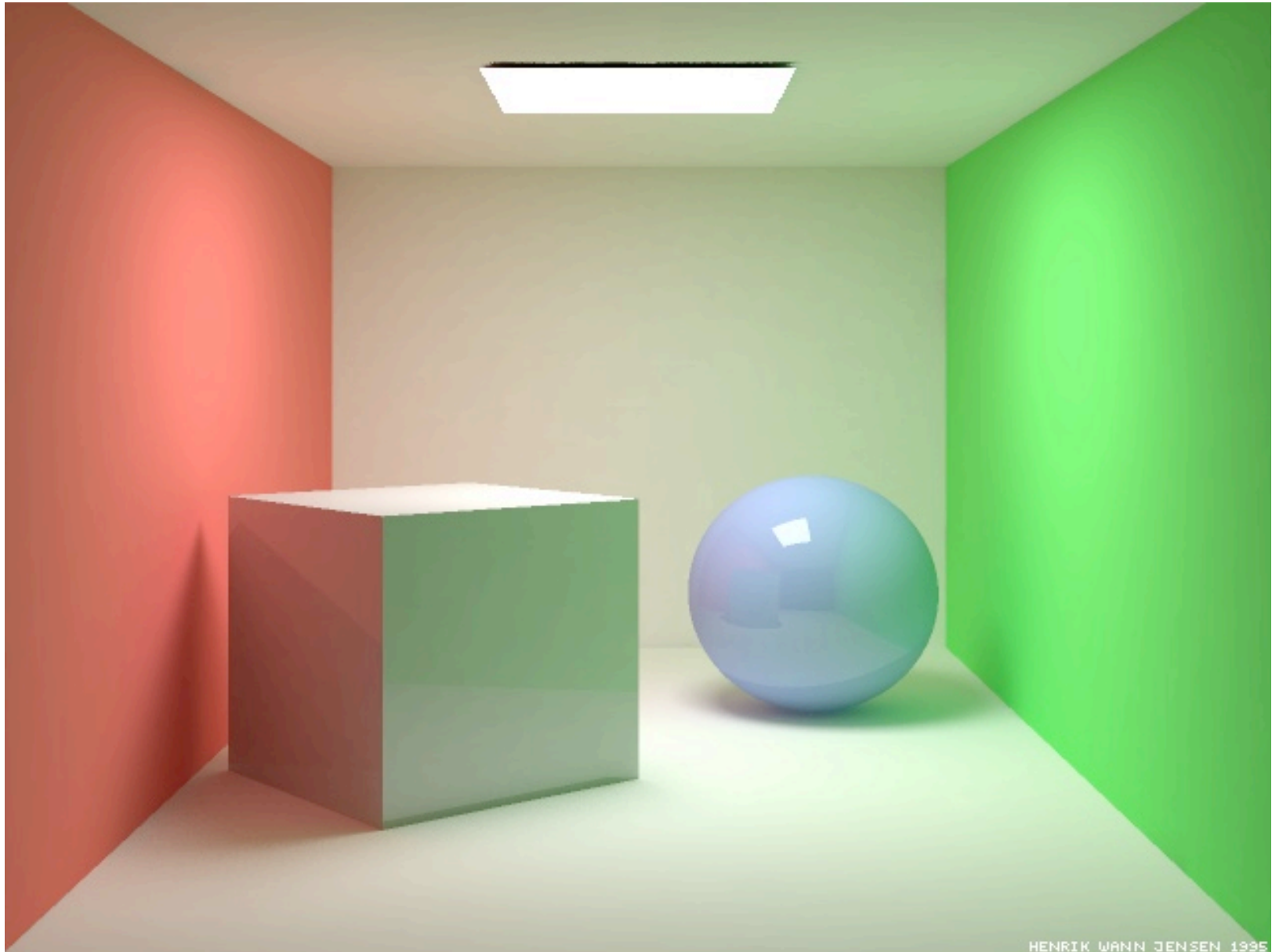
Shadows cast by a point source

- A point that can't see the source is in shadow
- For point sources, the geometry is simple





Ray-traced Cornell box, due to Henrik Jensen,
<http://www.gk.dtu.dk/~hwj>



Radiosity Cornell box, due to Henrik Jensen,
<http://www.gk.dtu.dk/~hwj>, rendered with ray tracer

Flat shading

- Compute shading value inside polygon using interpolate
- Flat shading
 - use polygon normal to shade
 - Adv:
 - fast -- one shading value per polygon
 - Disadv:
 - inaccurate -- looks blocky

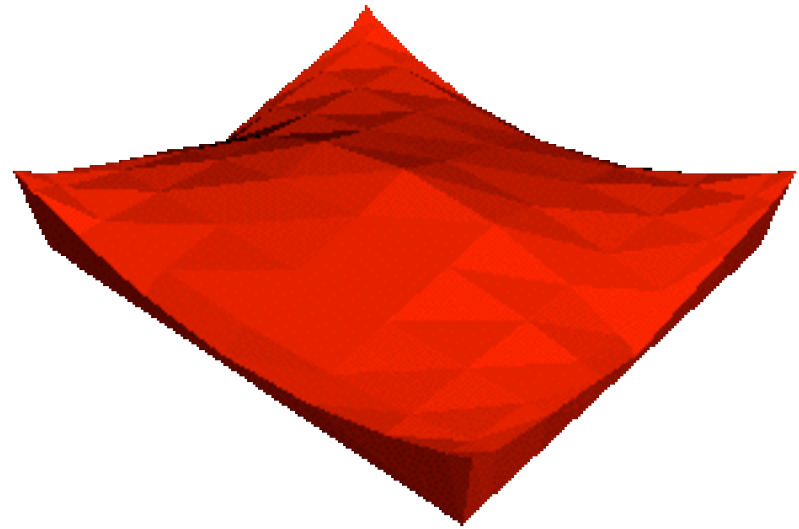
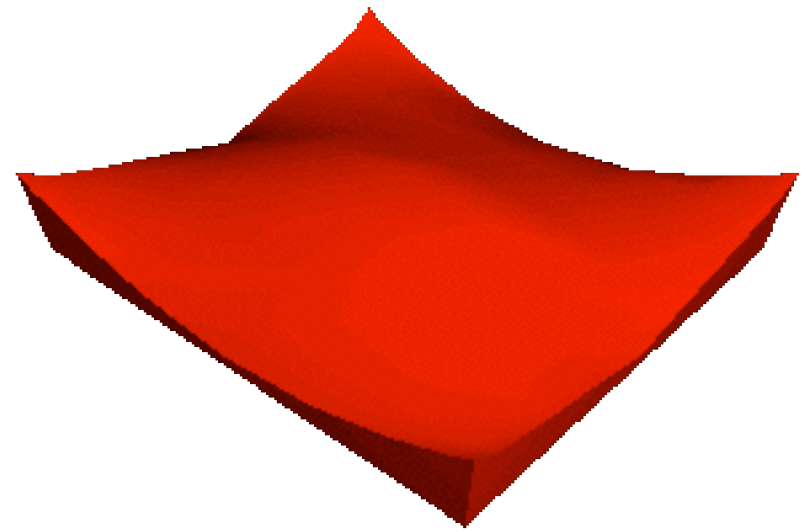


Figure from

http://freespace.virgin.net/hugo.elias/graphics/x_polygo.htm

Gouraud (Interpolated) Shading

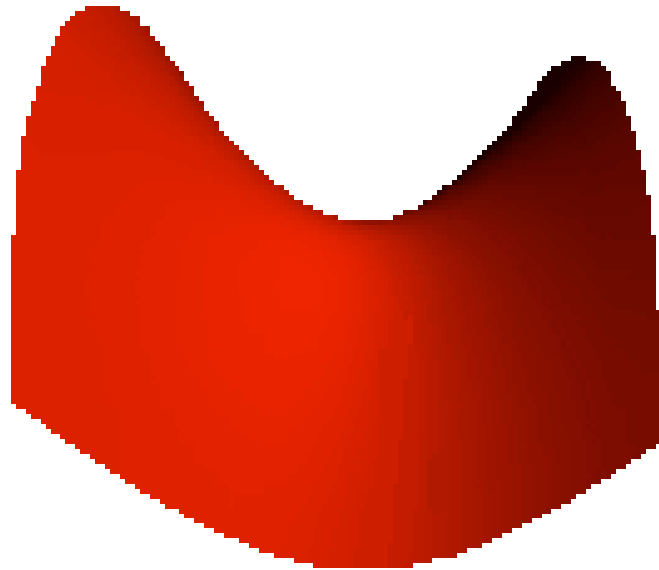
- Use normal at each vertex of polygon
 - known either from the “mesh” construction, OR, average normals where polygons meet.
- Shade these, and linearly interpolate
- Advantages
 - fast (interpolation can even be put into scan line algorithm)
 - much smoother
- Disadvantages:
 - specularities get lost



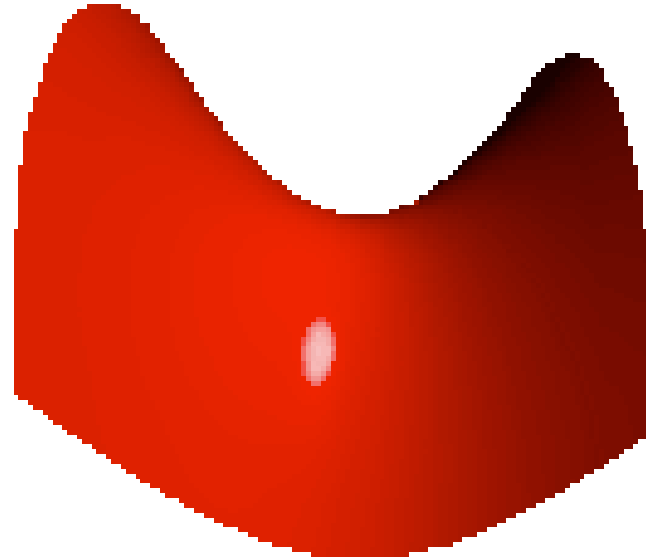
Phong Shading

- Interpolate normals instead of pixel values
 - Shade using normal estimate for each point
 - Advantage
 - high quality, narrow specularities
 - Disadvantage
 - more expensive than Gouraud

Gouraud



Phong



from

<http://www.geocities.com/SiliconValley/Horizon/6933/shading.html>