Standard nearby point source model

$$\rho_d(x)\left(\frac{N(x)\bullet S(x)}{r(x)^2}\right)$$

- N is the surface normal
- rho 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 giving the foreshortening.
- 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 is negligible for sources far enough away (e.g. the sun).
- Note that the sun is a point source because the direction to it does not change much as you move about.
- 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 the distance doesn't vary much either, and we can roll the constants together to get:

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

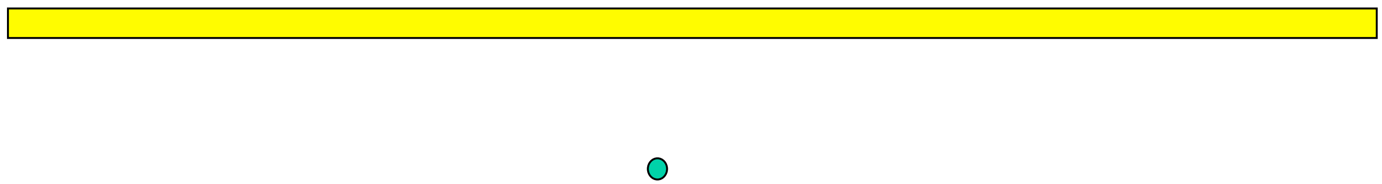
More General Illumination

The effect of light is generally linear (know this)!

We can break the effect of multiple sources into a sum, with one component for each source of light.

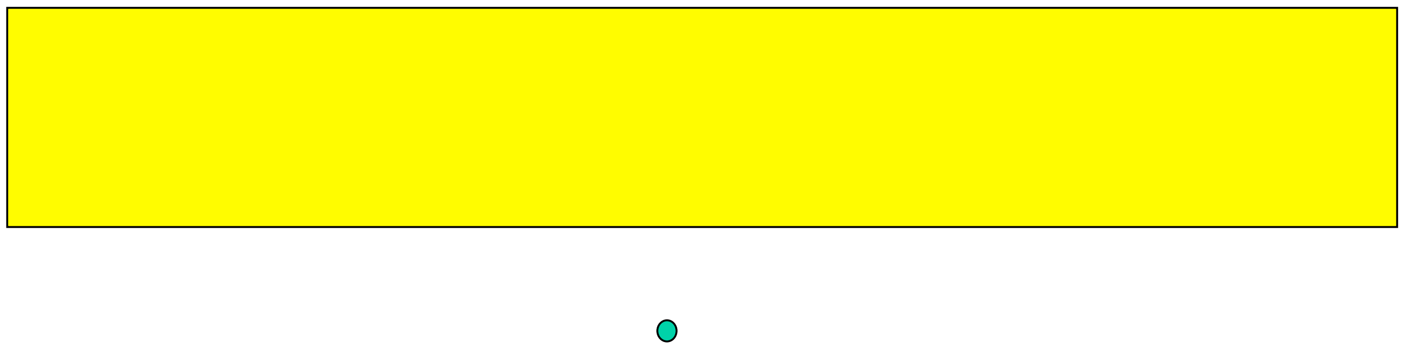
If we want to model an extended source, we add up (i.e., integrate) the contributions to small (e.g. point) bits, using the point source model.

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



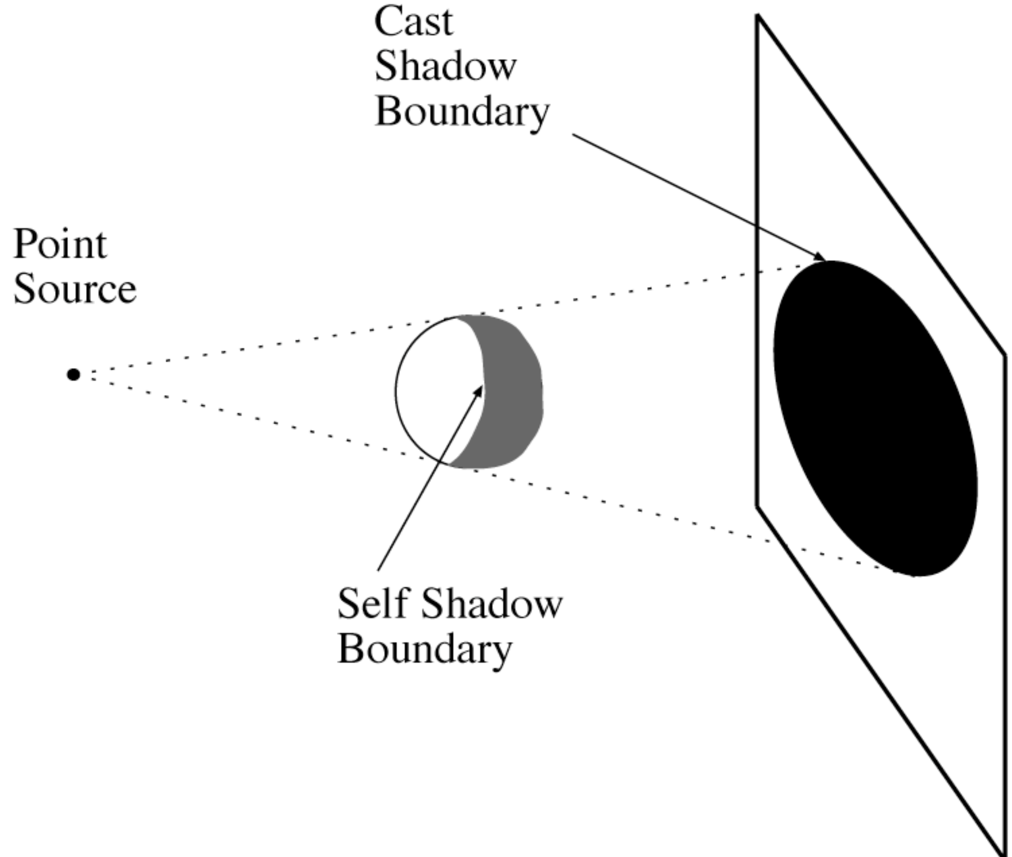
Can be handled by doing the integration (we won't)

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

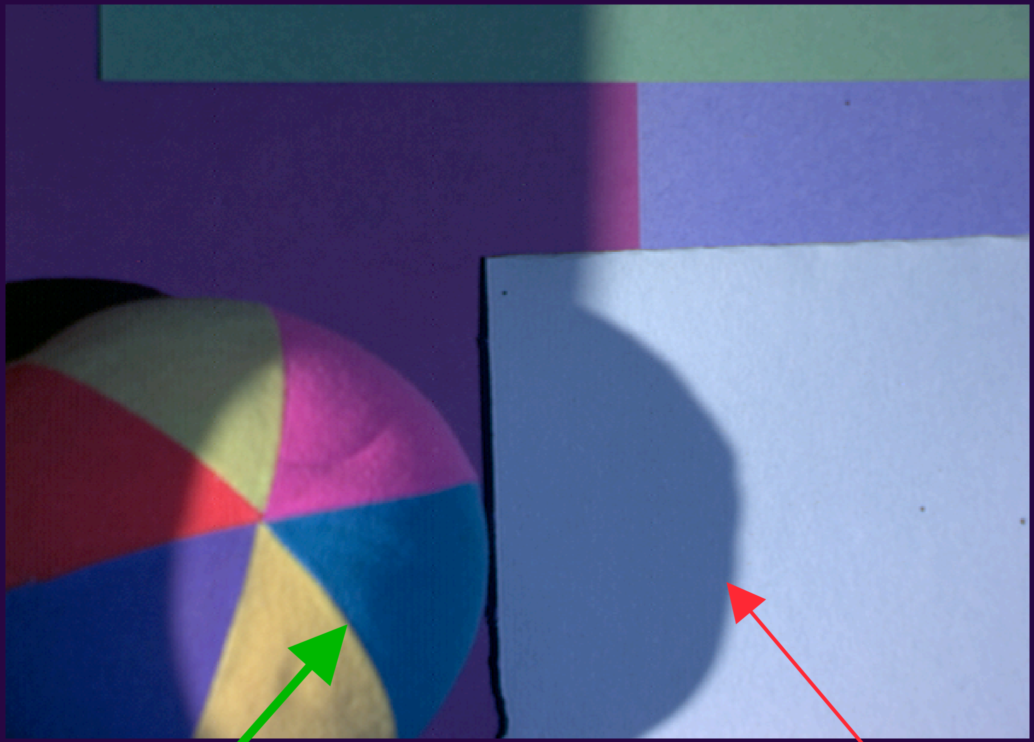
How about the hemisphere of the sky?

Shadows cast by a point source

- A point that can't see the source is in its shadow
- For point sources, the geometry is simple
- For extended sources, we have an umbra (points seeing no light), and a penumbra (seeing some parts of the light but not all)



The Shadow Problem

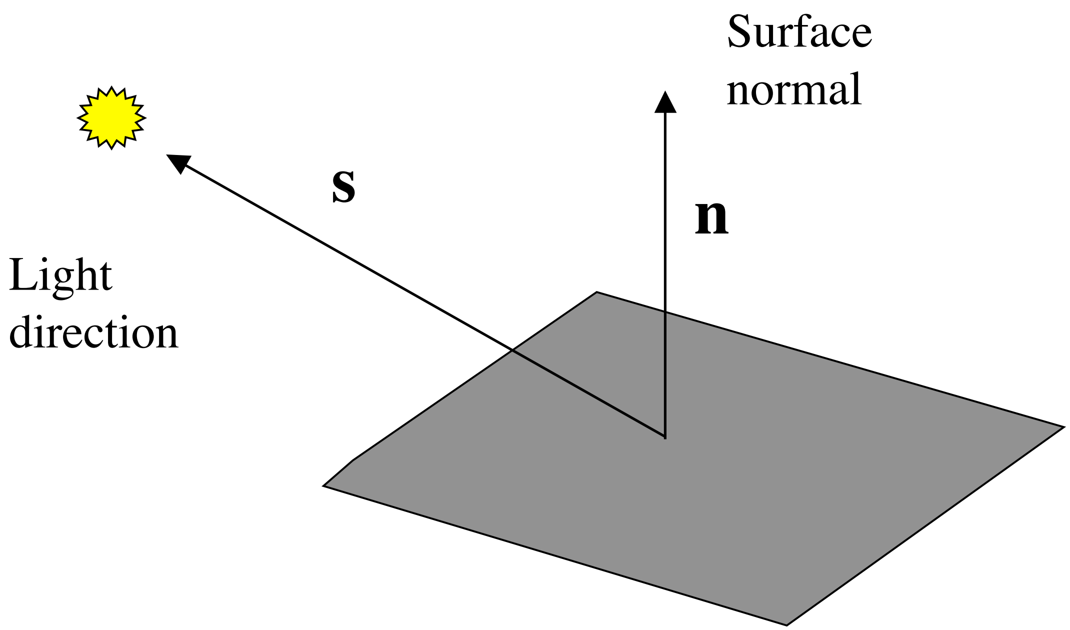


Material Edge

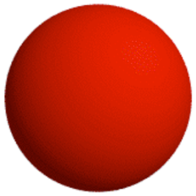
Shadow Edge

Shape from shading

- Suppose that we know the direction of a point source (there are ways to guess)
- Suppose Lambertian reflectance
- We know that image pixel brightness is proportional to **n•s**
- Can we figure out shape from brightness?



Shape from shading



- Problems
  - Can we find the normals at every point.
  - Do the normals give us shape? How?

Shape from shading

- Can we find the normals at every point?
  - Under-constrained! (Only have one piece of data per pixel, but we need 2 or 3 (if we need to estimate albedo as well).
  - Can impose regularization (smoothness) and consider boundary conditions
- Do normals give us shape?
  - Normals are not shape, but they can be related to the partial derivatives of the shape as a function--the surface is given by (x,y,f(x,y))
  - The partial derivatives must satisfy integrability constraints--random normals do not come from a continuous surface!

Photometric Stereo

- Shape from shading is hard! Consider an easier problem.
- Suppose that we have a number of known point sources, and we have successive pictures taken with each one used in turn.
- Let **g**(x,y) be the surface normal times the albedo (for the point in the world corresponding to image point (x,y).
- Let $\mathbf{V}_i$ be the light source direction, i, times a scalar embodying the light source magnitude and camera sensitivity
- Let I_i(x,y) be image intensity

Then $$I_i(x,y) = \mathbf{V}_i \bullet \mathbf{g}(x,y) \qquad \text{(Lambert's law)}$$

Photometric Stereo

$$I_i(x, y) = \mathbf{V}_i \bullet \mathbf{g}(x, y) \qquad \text{(Lambert's law)}$$

So how to solve for the surface?

How many lights do we need
(assuming that albedo is not known) ?

Photometric Stereo

$$I_i(x, y) = \mathbf{V}_i \bullet \mathbf{g}(x, y) \qquad \text{(Lambert's law)}$$

Combining the conditions given by each light, i, we get

$$\mathbf{i} = V\mathbf{g}$$

Where the i^{th} element of **i** is $I_i(x,y)$ and the i^{th} row of V is V_i

Since g has three elements, we need at least 3 lights.

If the number of lights is more than than 3, then use least squares!

$\boxed{\text{You should understand the construction of this problem.}}$