More General Illumination

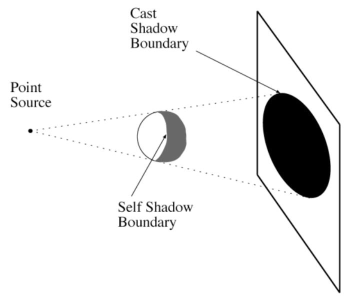
The effect of light is generally linear (know this)!

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

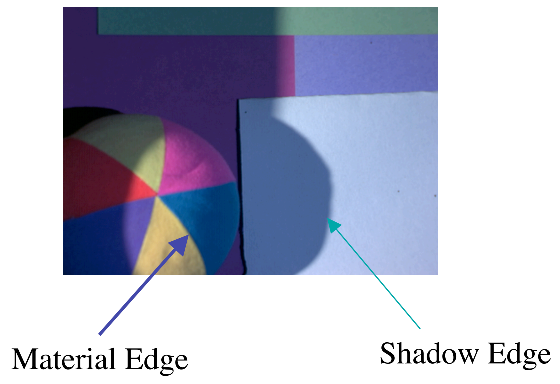
If we want to model an extended source, we add up (i.e., integrate) the contributions to small (e.g. point) bits, using the point source model.

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)

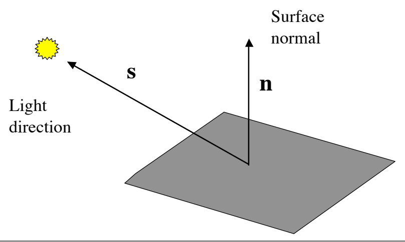


The Shadow Problem



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

- 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 $I_i(x,y) = \mathbf{V}_i \bullet \mathbf{g}(x,y)$ (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?

Simpler---how to solve for **g**(x,y)?

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

Photometric Stereo

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

Hopefully this looks like the i'th row of matrix, V, multiplied by a vector, **g**(x,y).

Photometric Stereo

Thus combining the conditions given by each light, i, we get

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

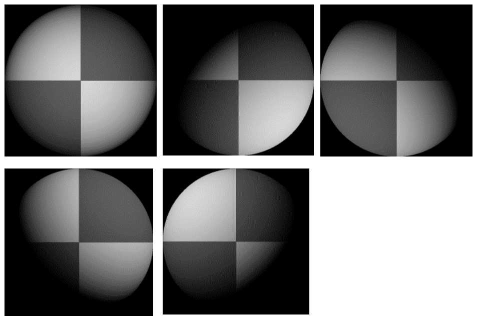
Where the ith element of **i** is $I_i(x,y)$ and the ith row of V is V_i

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!

You should understand the construction of this problem.

Example figures



Dealing with shadows

Each point is in K images (one for each light)

If $I_i(x,y)$ is in shadow, then ignore it.

Can simplify this in a program by multiplying both sides by a diagonal matrix with the image intensities on the diagonal.

This approach weights the equations according to image intensity, and so pixels in shadow are ignored (weight is zero)

Dealing with shadows

$$\begin{pmatrix} I_1^2(x,y) \\ I_2^2(x,y) \\ .. \\ I_n^2(x,y) \end{pmatrix} = \begin{pmatrix} I_1(x,y) & 0 & .. & 0 \\ 0 & I_2(x,y) & .. & .. \\ .. & .. & .. & 0 \\ 0 & .. & 0 & I_n(x,y) \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ .. \\ \mathbf{V}_n^T \end{pmatrix} \mathbf{g}(x,y)$$

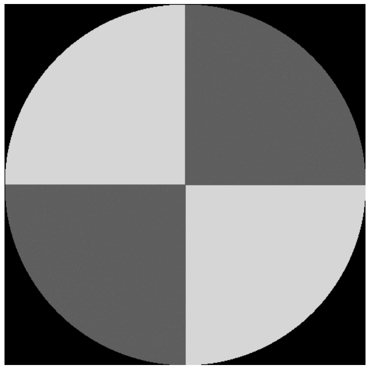
image intensity becomes squared

shadow => 0

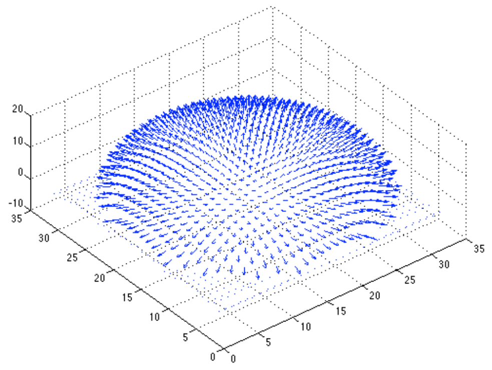
known light vectors

unknown

Recovered reflectance



Recovered normal field



From Normals to Shape

From **g** we can get the normal $\hat{\mathbf{n}} = \frac{\mathbf{g}}{|\mathbf{g}|}$

It is natural to represent surface as a depth map (x,y,f(x,y))

But what is the relationship between that and the normals?

From Normals to Shape

Given (*x, y, f(x,y)*), what is the surface normal direction?

Method one (cross product)

$$\hat{\mathbf{n}} \propto \frac{\delta}{\delta x}(x,y,f(x,y)) \times \frac{\delta}{\delta y}(x,y,f(x,y))$$

(If you are on the surface the partial with respect to x gives the direction in 3D if you try to go in the x direction. Ditto for y. The normal is perpendicular to these two directions).

From Normals to Shape

Given (*x, y, f(x,y)*), what is the surface normal direction?

Method two (level curves)

Given a surface, S, specified by $g(x,y,z)=0$

$\nabla g(x,y,z)$ is normal to S

So, find $g(x,y,z)$ such that $g(x,y,z)=0$ is our surface

$$g(x,y,z)=z-f(x,y)$$

From Normals to Shape

Given (*x, y, f(x,y)*), what is the surface normal direction?

Method two (level curves)

$$g(x,y,z)=z-f(x,y)$$

$$\nabla g(x,y,z)= (-f_x,\ -f_y,\ 1)$$

From Normals to Shape

Either way, $\hat{\mathbf{n}} \propto (-f_x,\ -f_y,\ 1)$

It should be clear that $f_x=-\frac{n_x}{n_z}$ and $f_y=-\frac{n_y}{n_z}$

(In general, **n** will embody the albedo, so we must be prepared for an arbitrary scale factor in **n**---most easily dealt with using the above ratio if we want f_x and f_y).