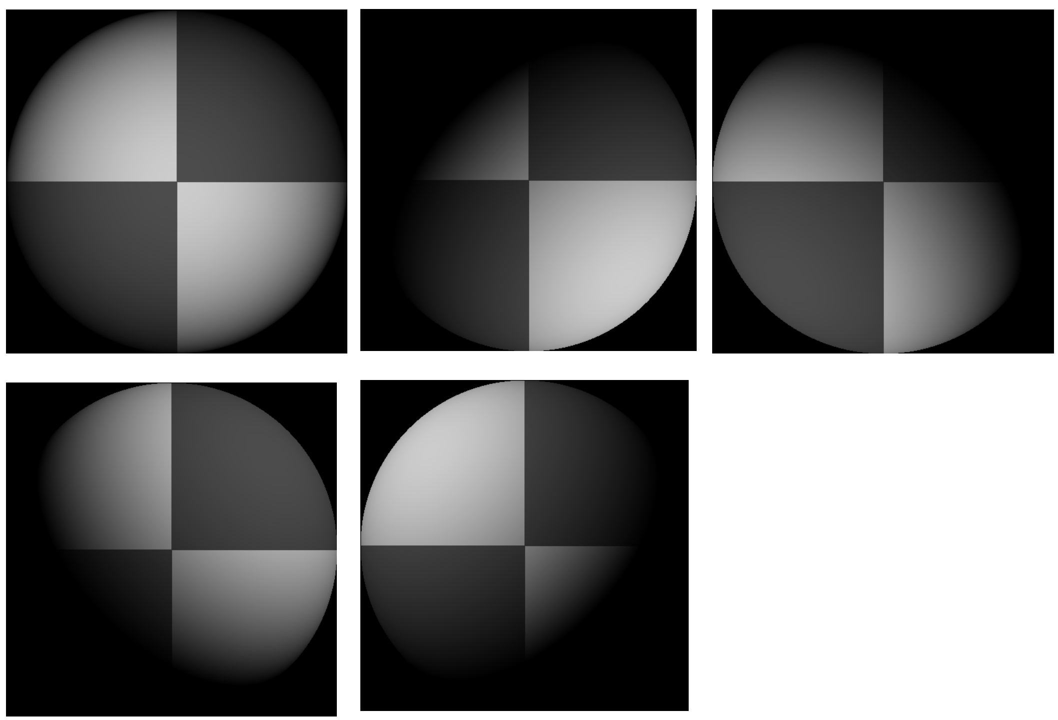
Photometric Stereo

- 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.



Photometric Stereo

Thus 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!

You should understand the construction of this problem.

One way to deal with shadows

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

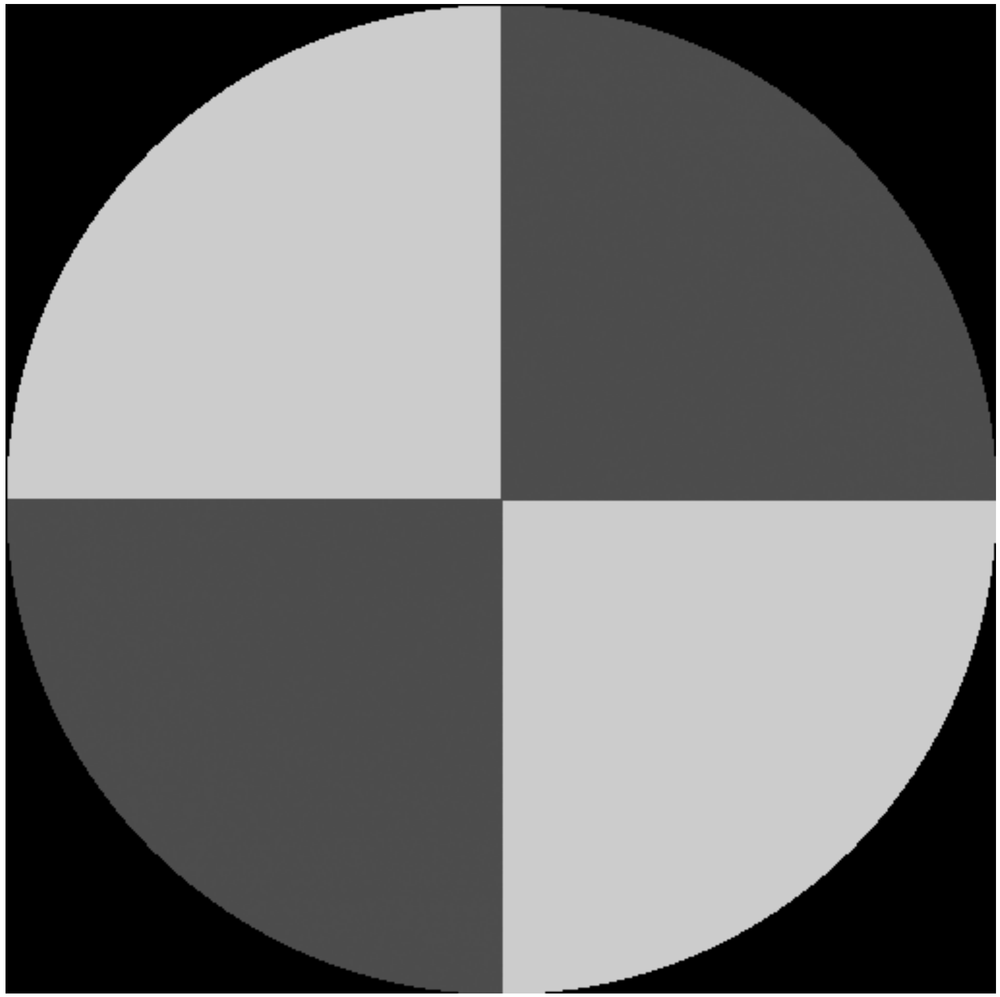
image intensity becomes squared

shadow => 0

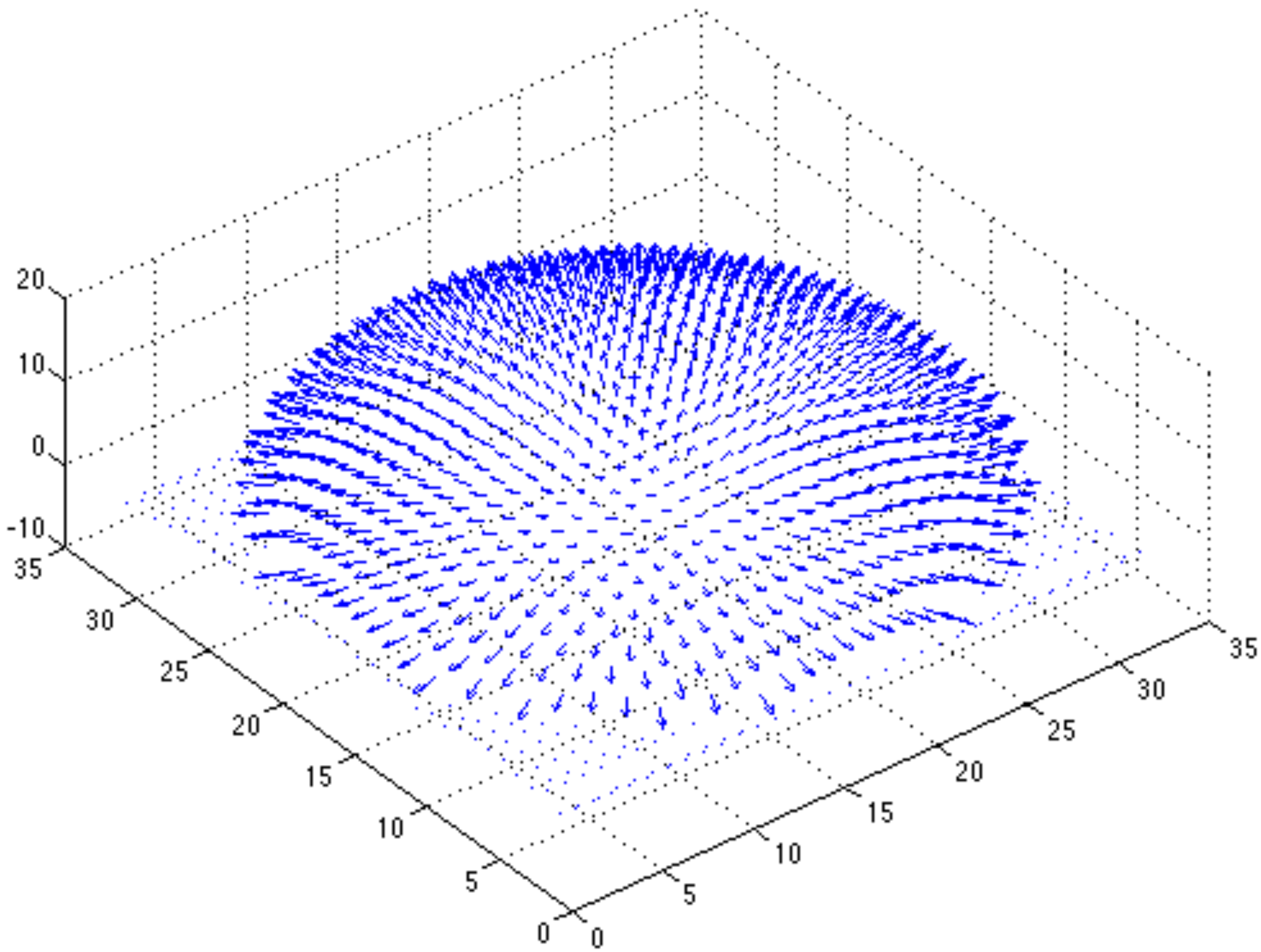
known light vectors

unknown

Recovered reflectance



Recovered normal field



From Normals to Shape

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

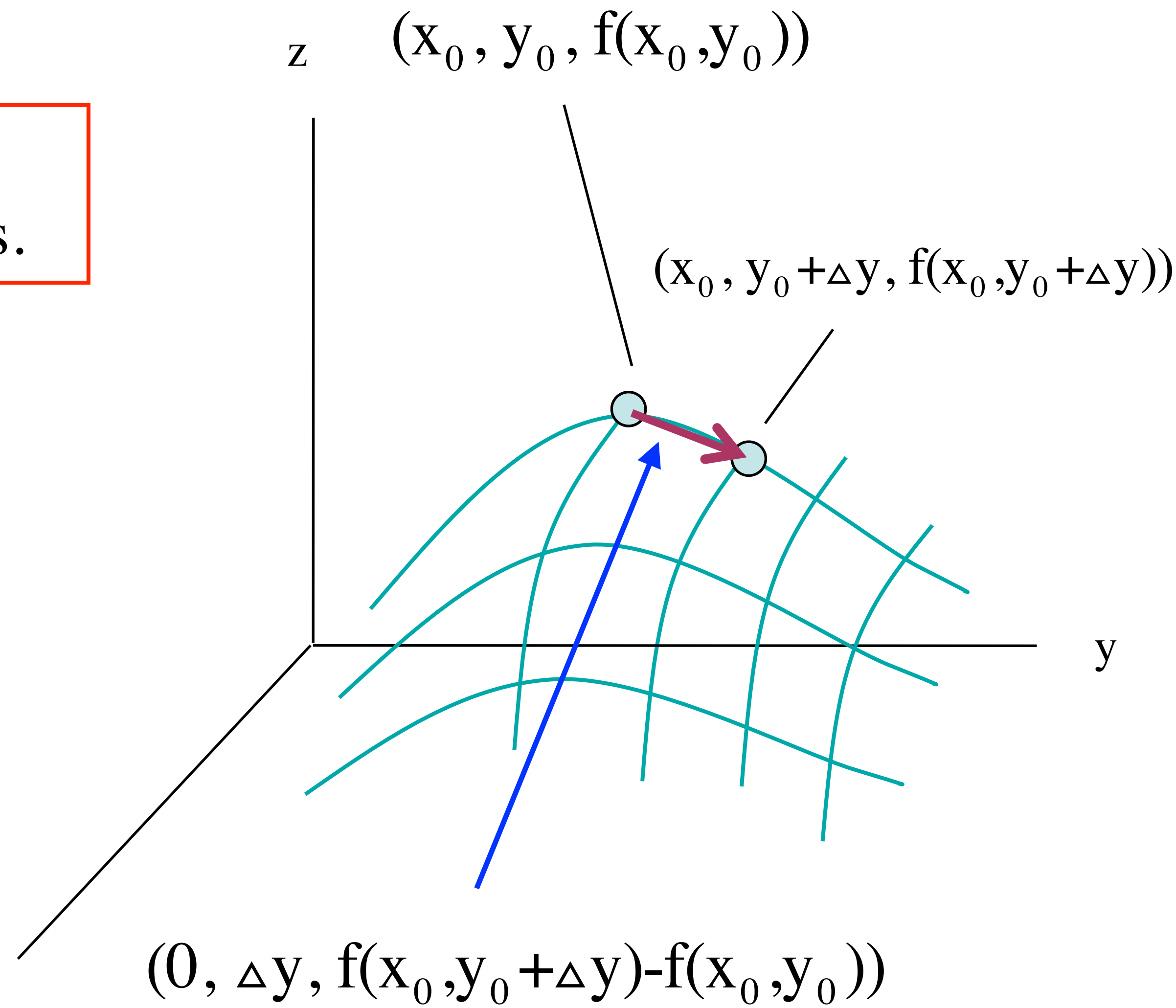
It is natural to represent surface as a depth map $(x, y, f(x, y))$

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?

Reminder why partial derivatives give tangents.

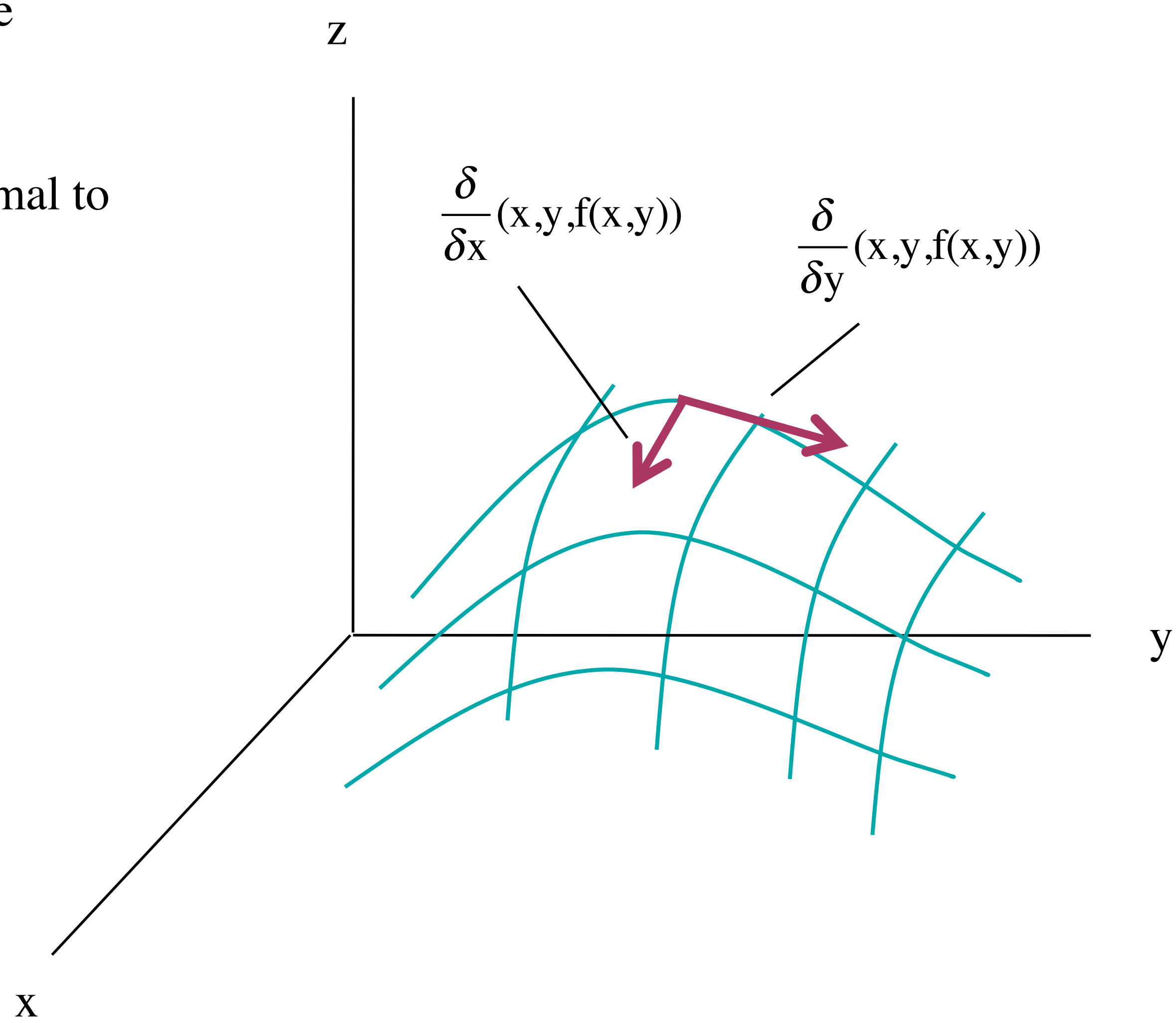


$$(0, \Delta y, f(x_0,y_0+\Delta y)-f(x_0,y_0)) = \Delta y * \left(0, 1, \frac{f(x_0,y_0+\Delta y)-f(x_0,y_0)}{\Delta y}\right)$$

approximately parallel to $\left(0, 1, \frac{\partial f}{\partial y}\right)$

The partials in x and y give us two tangents which are vectors in the plane touching the surface.

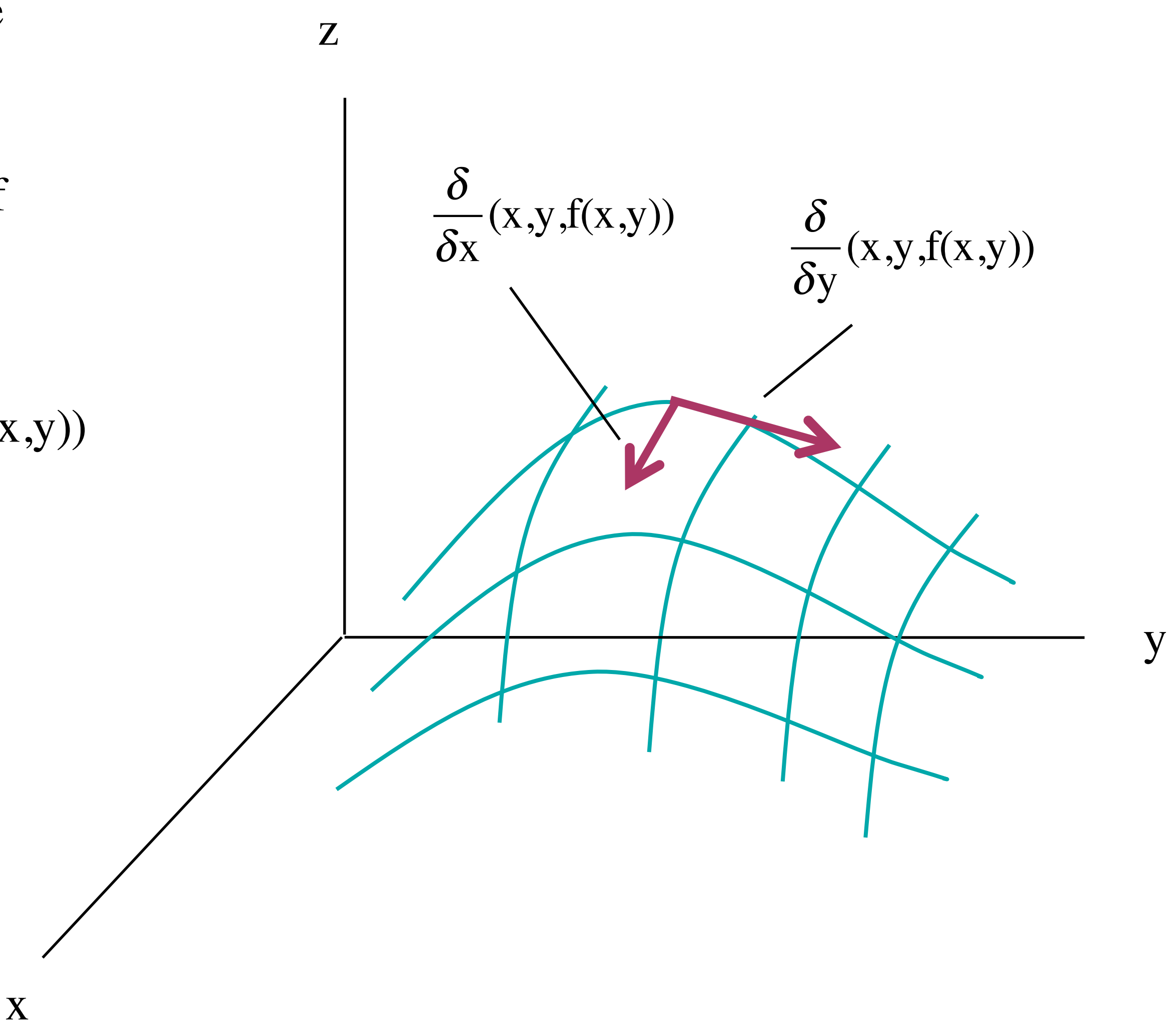
So, we want a vector that is normal to both of them.



The partials in x and y give us two tangents which are vectors in the plane touching the surface.

To get a vector normal to both of them, take their cross product.

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



Reminder about cross product

$A = (a_1, a_2, a_3) \qquad B = (b_1, b_2, b_3)$

Then $\quad A \times B = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$

$A \times B \perp A$ and $A \times B \perp B$

In right hand coordinates: A,B, and $A \times B$ form a right hand system.

Algebraicly, $A \times B \bullet A = 0$ and $A \times B \bullet B = 0$ (check!)

$|A \times B| = |A||B|\sin\theta$

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

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

$$= (1, 0, f_x) \times (0, 1, f_y)$$

$$= (-f_x \ , -f_y \ , \ 1)$$

using $A \times B = \left(a_2 b_3 - a_3 b_2 , a_3 b_1 - a_1 b_3 , a_1 b_2 - a_2 b_1\right)$

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, $\mathbf{n} = \rho\,\hat{\mathbf{n}} \propto (-f_x, -f_y, 1)$

From $\mathbf{n}$ we can get the albedo (it is $\|\mathbf{n}\|$)

Note that we have two vectors that are parallel. We do not know the relative scale, but the ratios of components cancel out the scale. So,

$$\frac{n_x}{n_z} = \frac{-f_x}{1} \quad \text{and} \quad \frac{n_y}{n_z} = \frac{-f_y}{1}$$

Rearranging a bit, $f_x = -\dfrac{n_x}{n_z}$ and $f_y = -\dfrac{n_y}{n_z}$

From Normals to Shape

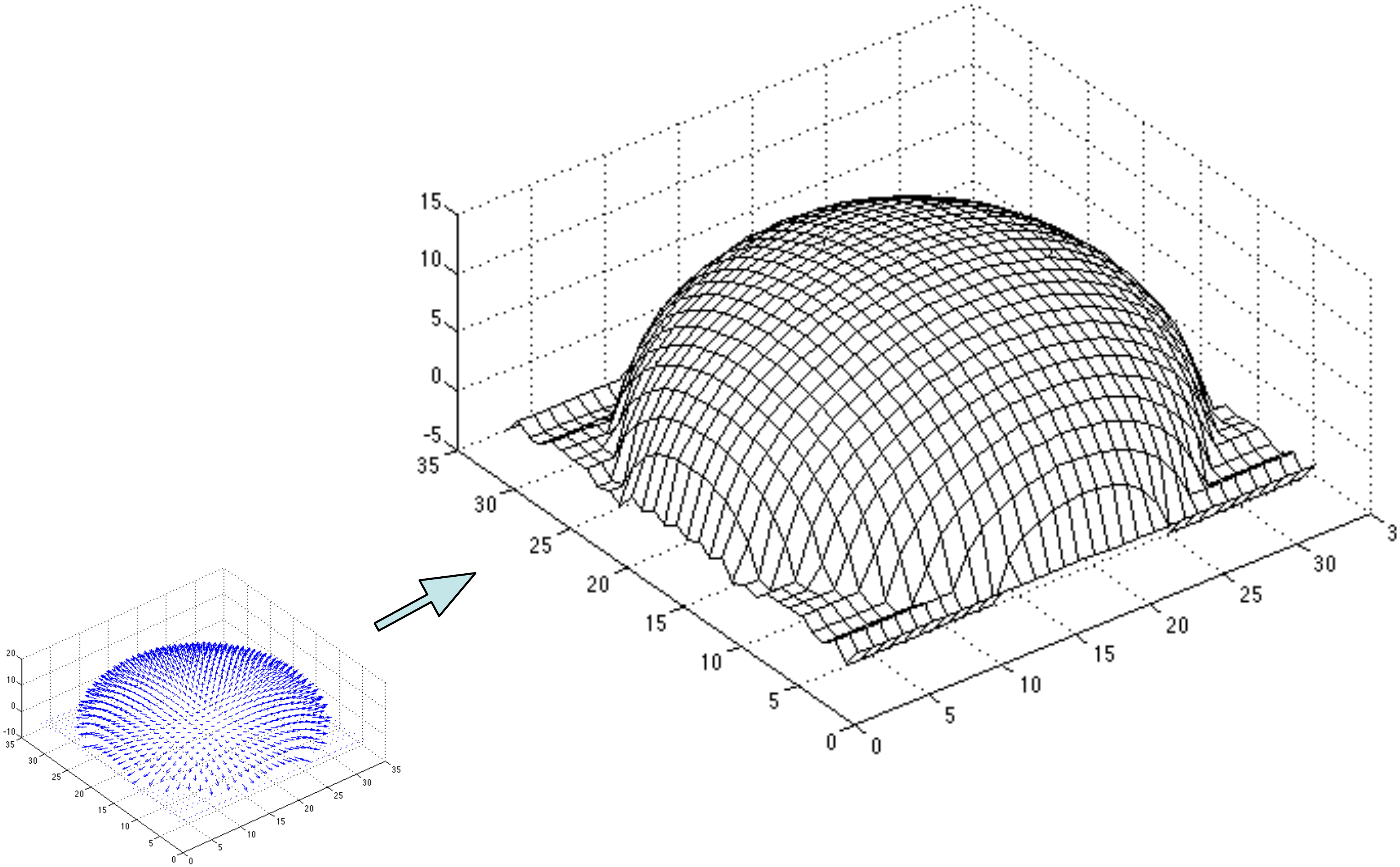
So, if have the normals, we can estimate the derivatives of f(x,y)

Minor point for those who have vector calculus: If we assume that f_x and f_y are the derivatives of a differentiable function, f (x,y) we can further check (or constrain) that $f_{xy}=f_{yx}$.

We can recover the surface height at any point by integration along some path. For example, if we declare the origin to be at height C, and go along the x axis, then parallel to the y axis:

$$f(x,y) = \int_0^x f_x(x',0)\,dx' + \int_0^y f_y(x,y')\,dy' + C$$

Surface recovered by integration



Color

Color is a sensation

Usually there is light involved, and usually there is a relationship between the world and the colors you see

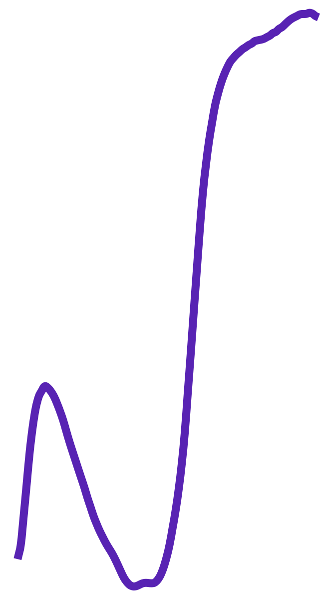
Your brain has a big effect on the colors you see

We will focus on what colors mean to a camera which is **much simpler**

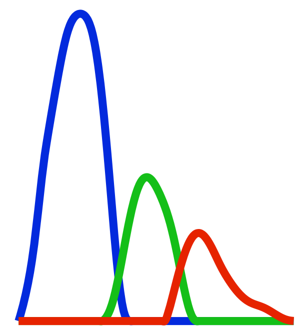
Color for a camera (R,G,B) is a very limited sampling of spectral light energy (why three values?)

Recall Image Formation (Spectral)

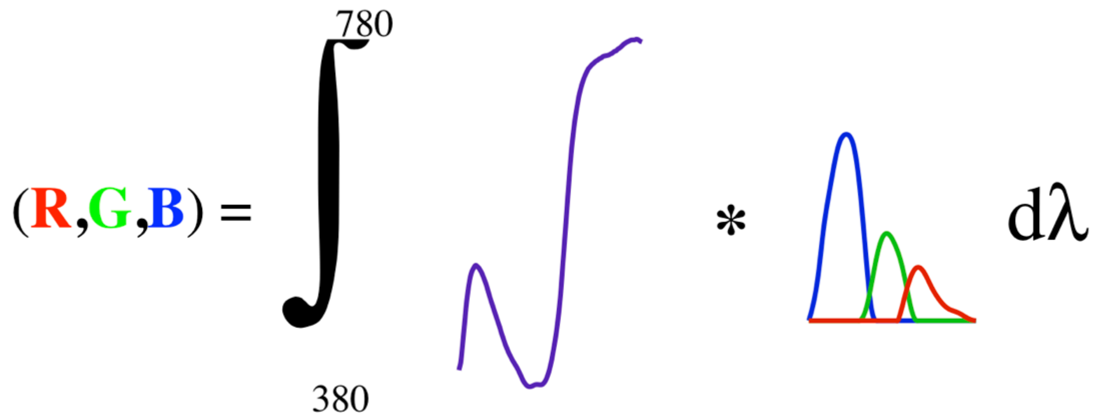
$$(\mathbf{R},\mathbf{G},\mathbf{B}) = \int_{380}^{780}$$



$$*$$



$$d\lambda$$



Alternative notation $C_k = \int L(\lambda) R^{(k)}(\lambda) d\lambda$

Discrete version: $\mathbf{C} = \mathbf{R} \cdot \mathbf{L}$

where **L** is the light spectra as a column vector

R has the sensor sensitvities as rows

C is the camera responses as a column vector

From previous slide

$$\mathbf{C}=\mathrm{R}*\mathbf{L}$$

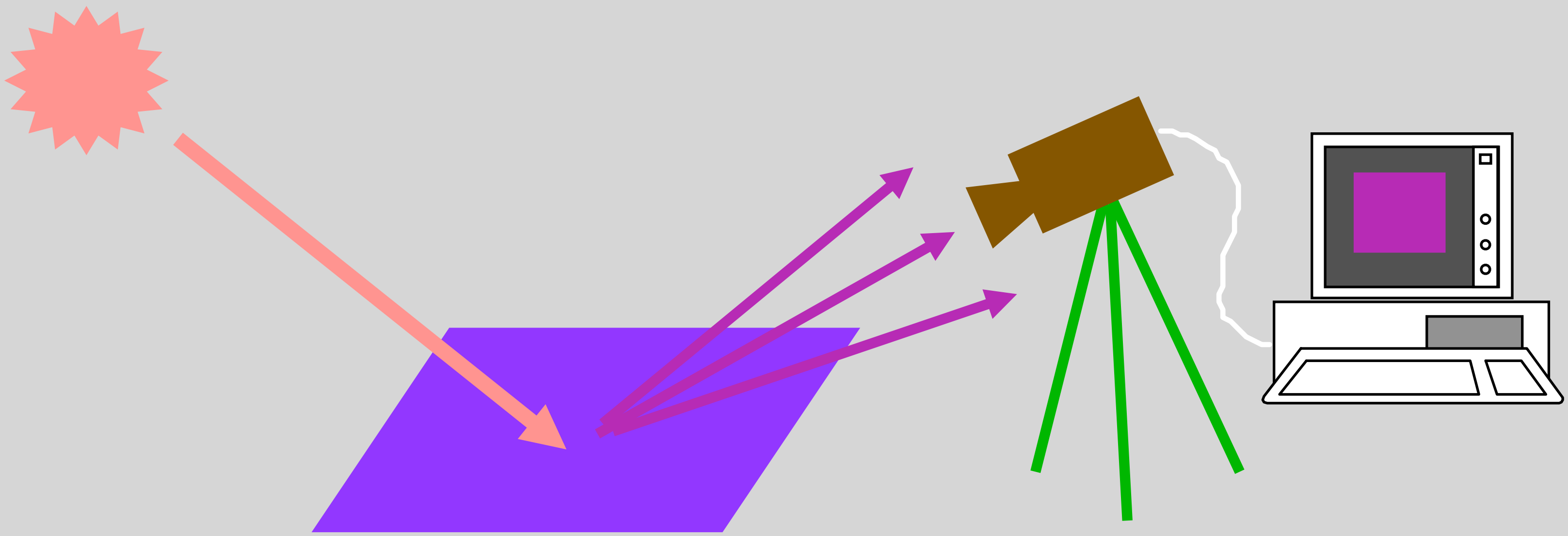
R is **not** full rank (typical values are 3 by 101 or 3 by 31)

First key observation is that you cannot recover **L** from **C**
(**L** is spectra**,** **C** is RGB)

Second observation---many spectra can have the same RGB.
These are metamers (the spectra are a metameric match).

(This is the essence of color reproduction)

(R,G,B) depends on the light, the surface, and the camera

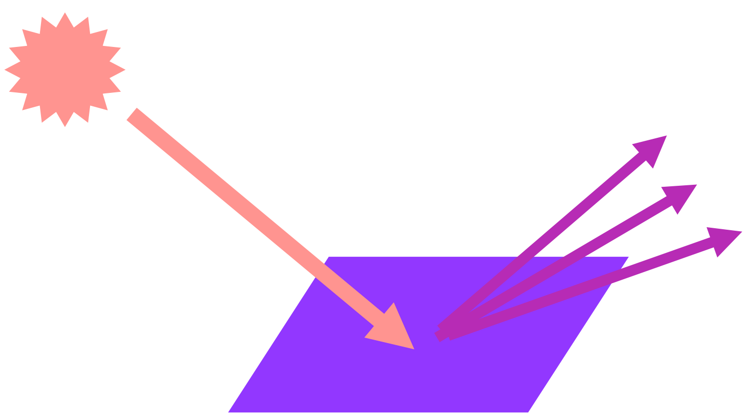


Spectral reflectance

We define spectral reflectance by

$$S(\lambda) = \frac{L(\lambda)}{E(\lambda)}$$ where is $E(\lambda)$ incoming and $L(\lambda)$ is outgoing

So $L(\lambda) = E(\lambda)S(\lambda)$



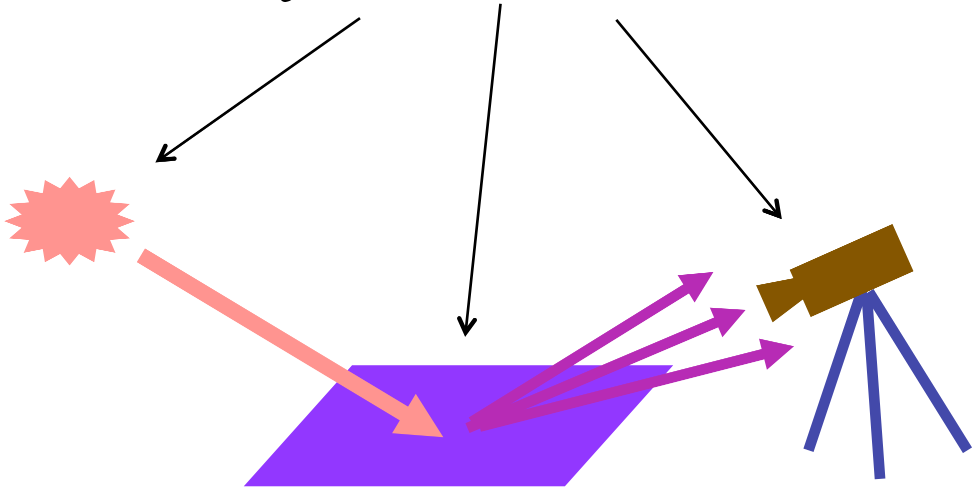
Recalling the BDRF, we know S is function of incoming and outgoing angles. Here we are assuming a particular pair of angles.

Spectral reflectance

$L(\lambda) = E(\lambda)S(\lambda)$ (previous slide)

Recall $C_k = \int L(\lambda) R^{(k)}(\lambda) d\lambda$

Now, $C_k = \int E(\lambda) S(\lambda) R^{(k)}(\lambda) d\lambda$

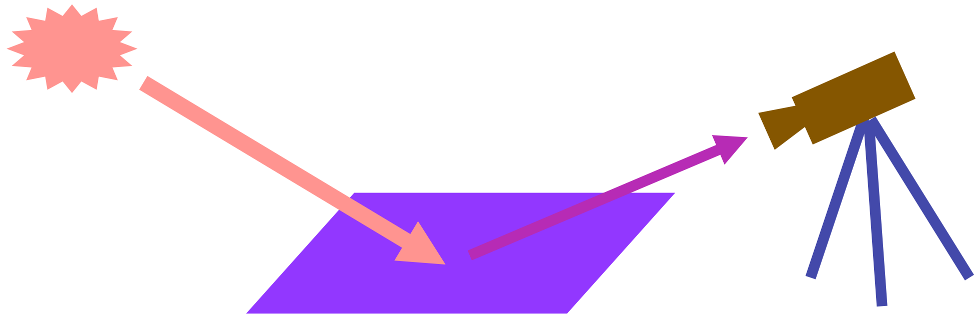


Color imaging summary

To get the color at a pixel, which is a particular outgoing direction with respect to the surface the pixel sees, we can add up the contributions of light rays reaching the surface.

For each contribution we have $E(\lambda)$ and $S(\lambda)$ and we can assume we know $R^{(k)}(\lambda)$.

We get the value of each channel by $C_k = \int E(\lambda)S(\lambda)R^{(k)}(\lambda)d\lambda$



Imaging system

In $C_k = \int E(\lambda)S(\lambda)R^{(k)}(\lambda)d\lambda$

$L(\lambda) = E(\lambda)S(\lambda)$ is about the world

The integration of $L(\lambda)$ against the senstivity functions
is a property of the **imaging** system.

This is a good model for cameras, and for human color matching experiments.

However, notice the implicit assumption that every pixel behaves the same.

Human Color Constancy Demo