# Announcements

- Project 3b due Friday

# Last time

- Shape from shading
- Photometric Stereo

# Lambertian reflection

$$I_e = k_d \mathbf{N} \cdot \mathbf{L} I_i$$

is light source intensity

Lets assume that $I_i = 1$

can achieve this by dividing each pixel in the image by $I_i$

$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

image intensity at a single point

**Lighting** direction (same for all points)

**Albedo** at a point

Surface **normal** at a point

# Shape from shading

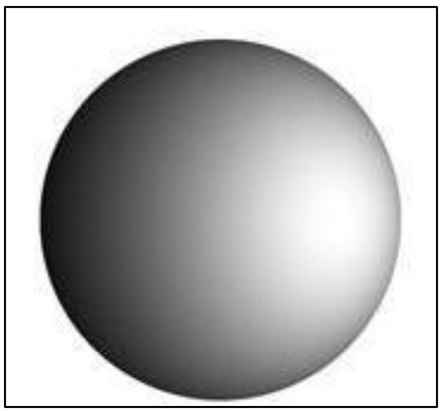
<u>Input:</u>

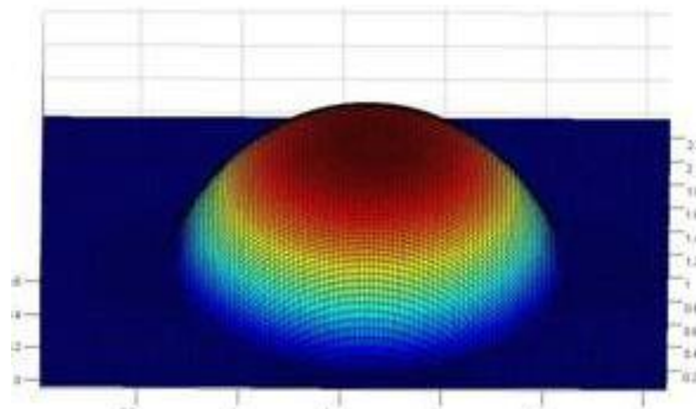
- Single Image

<u>Output:</u>

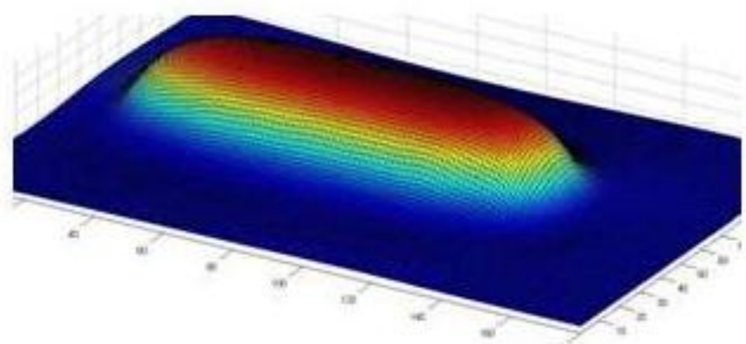
- 3D shape of the object in the image

I(x,y)

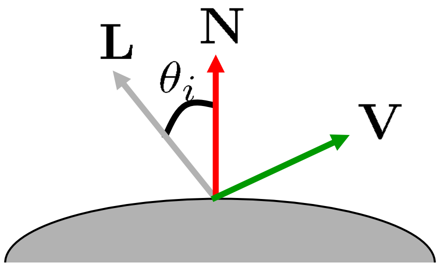




Z(x,y)



# Shape from shading




Suppose $k_d = 1$

$$\begin{aligned} I &= k_d \mathbf{N} \cdot \mathbf{L} \\ &= \mathbf{N} \cdot \mathbf{L} \\ &= cos\ \theta_i \end{aligned}$$

You can directly measure angle between normal and light source

- Not quite enough information to compute surface shape

$$I(x, y) = \cos(0.5)$$

$$\theta = 60^{\circ}, \phi = ?$$

- But can be if you add some additional info, for example
  - Assume normals along the sihouette are known
  - Constraints on neighboring normals—“integrability”
  - Smoothness

# Surface Normal

$$N = (n_x, n_y, n_z)^T$$

A surface $z(x, y)$

A point on the surface: $(x, y, z(x, y))^T$

Tangent directions

$$t_x = (1, 0, z_x)^T \qquad t_y = (0, 1, z_y)^T$$

$$N = \frac{t_x \times t_y}{\| t_x \times t_y \|} = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}} \left( -z_x, -z_y, 1 \right)^T$$

# Shape from shading

$$I(x,y) = N \bullet L = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}} \Rightarrow \sqrt{z_x^2 + z_y^2} = \sqrt{\frac{1}{I(x,y)^2} - 1}$$

$$L = (0,0,1)^T$$

But both unknowns come from an integrable surface:

Z(x,y) thus we can use the **integrability** constraint:

$$z_{xy} = z_{yx}$$

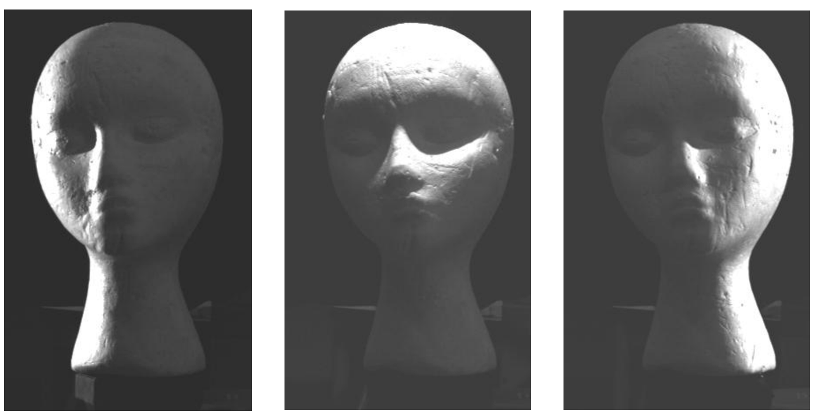
# Photometric stereo

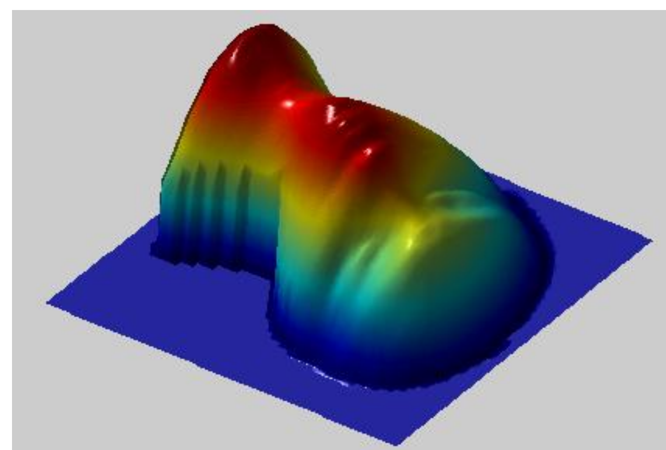
Input:

- Several Images
  * same object
  * different lightings
  * same pose

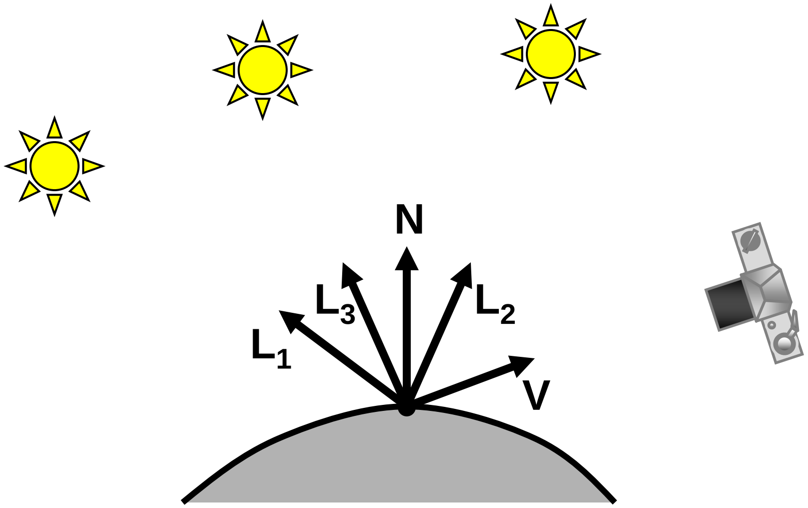
Output:

- 3D shape of the object in the images
- Albedo
- Lighting





# Photometric stereo




$$I_1 = k_d \mathbf{N} \cdot \mathbf{L_1}$$
$$I_2 = k_d \mathbf{N} \cdot \mathbf{L_2}$$
$$I_3 = k_d \mathbf{N} \cdot \mathbf{L_3}$$

Can write this as a matrix equation:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = k_d \begin{bmatrix} \mathbf{L_1}^T \\ \mathbf{L_2}^T \\ \mathbf{L_3}^T \end{bmatrix} \mathbf{N}$$

# Solving the equations

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}}_{\substack{\mathbf{I} \\ 3\times 1}} = \underbrace{\begin{bmatrix} \mathbf{L_1}^T \\ \mathbf{L_2}^T \\ \mathbf{L_3}^T \end{bmatrix}}_{\substack{\mathbf{L} \\ 3\times 3}} \underbrace{k_d \mathbf{N}}_{\substack{\mathbf{G} \\ 3\times 1}}$$

$$\mathbf{G} = \mathbf{L}^{-1}\mathbf{I}$$

$$k_d = \|\mathbf{G}\|$$

$$\mathbf{N} = \frac{1}{k_d}\mathbf{G}$$

# More than three lights

Get better results by using more lights

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \mathbf{L_1} \\ \vdots \\ \mathbf{L_n} \end{bmatrix} k_d \mathbf{N}$$

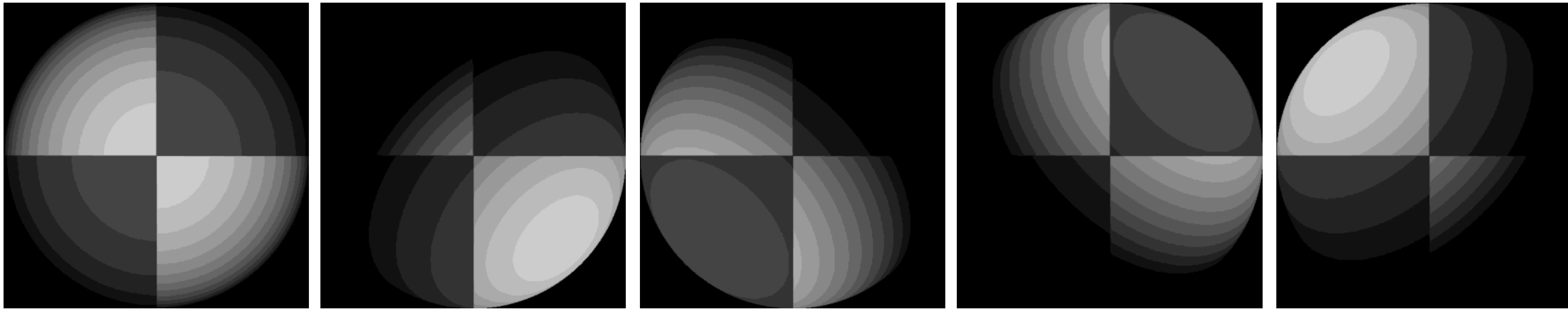
Least squares solution:

$$\begin{aligned} \mathbf{I} &= \mathbf{L}\mathbf{G} \\ \mathbf{L}^{\mathbf{T}}\mathbf{I} &= \mathbf{L}^{\mathbf{T}}\mathbf{L}\mathbf{G} \\ \mathbf{G} &= (\mathbf{L}^{\mathbf{T}}\mathbf{L})^{-1}(\mathbf{L}^{\mathbf{T}}\mathbf{I}) \end{aligned}$$

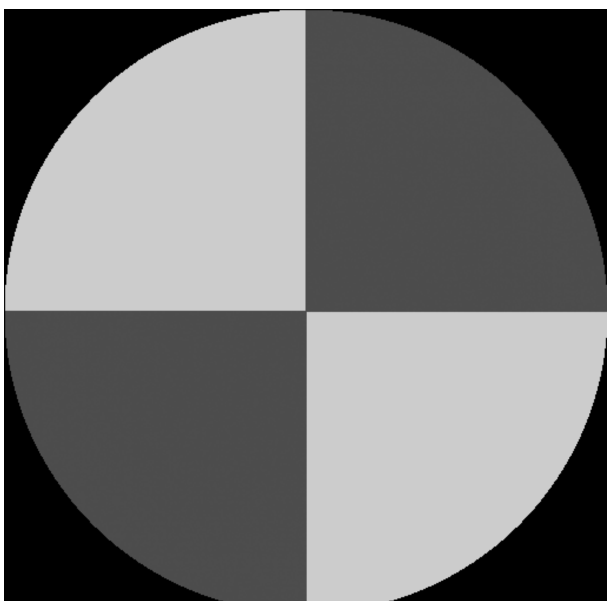
Solve for N, $k_d$ as before

What's the size of $\mathbf{L}^{\mathbf{T}}\mathbf{L}$?

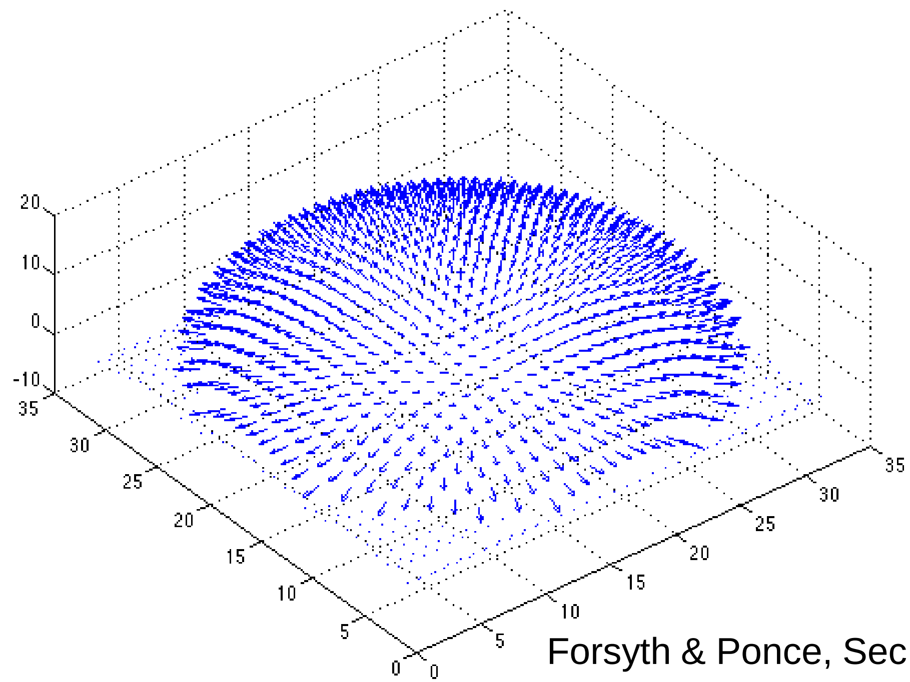
# Example



Recovered albedo



Recovered normal field




Forsyth & Ponce, Sec. 5.4

# Color images

## The case of RGB images

- get three sets of equations, one per color channel:

$$\begin{aligned}\mathbf{I}_R &= k_{dR}\,\mathbf{LN}\\ \mathbf{I}_G &= k_{dG}\,\mathbf{LN}\\ \mathbf{I}_B &= k_{dB}\,\mathbf{LN}\end{aligned}$$

- Simple solution: first solve for **N** using one channel or grayscale
- Then substitute known **N** into above equations to get $k_d$ s

$$k_d = \frac{\sum_i I_i L_i N^T}{\sum_i (L_i N^T)^2}$$

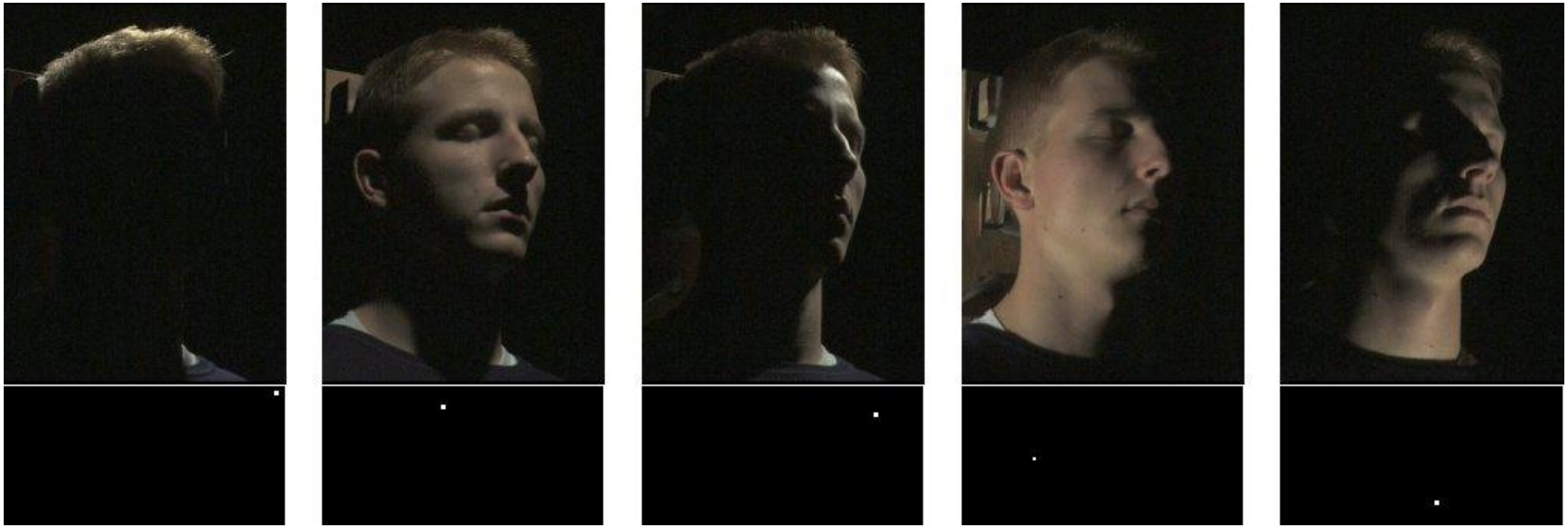
# Where do we get the lighting directions?

# Capture lighting variation

Illuminate subject from many incident directions

From Ravi Ramamoorthi

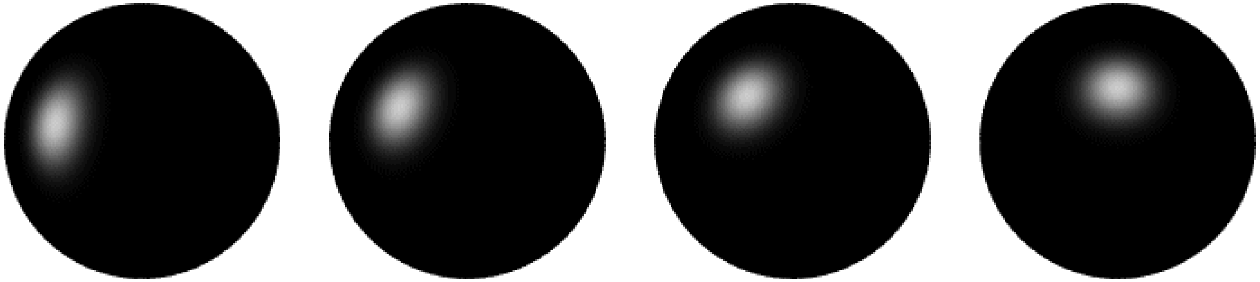
# Example images:



From Ravi Ramamoorthi

# Computing light source directions

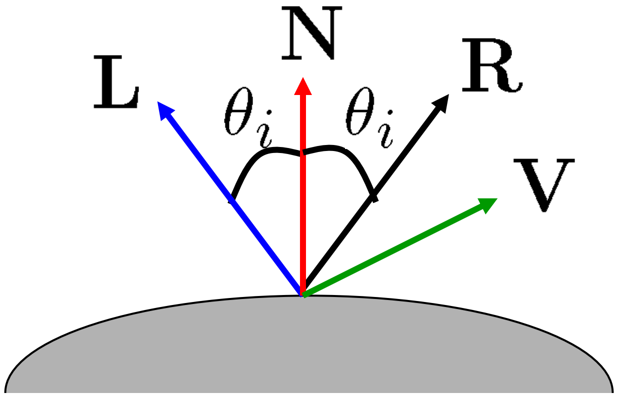
Trick: place a chrome sphere in the scene



- the location of the highlight tells you where the light source is

# Recall the rule for specular reflection

For a perfect mirror, light is reflected about **N**




$$I_e = \begin{cases} I_i & \text{if } \mathbf{V} = \mathbf{R} \\ 0 & \text{otherwise} \end{cases}$$

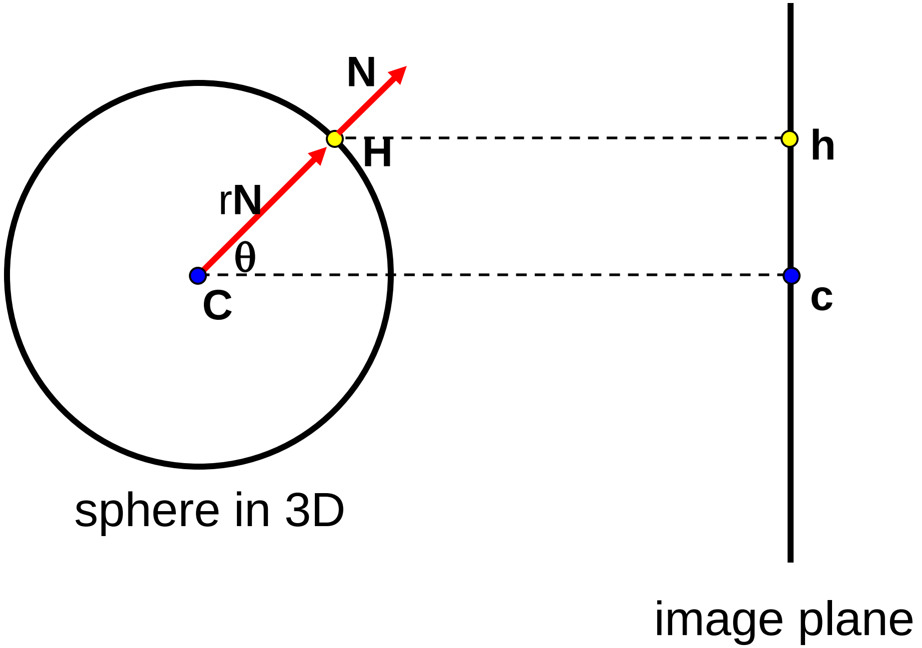
We see a highlight when **V** = **R**

- then L is given as follows:

$$\mathbf{L} = 2(\mathbf{N} \cdot \mathbf{R})\mathbf{N} - \mathbf{R}$$

# Computing the light source direction

Chrome sphere that has a highlight at position **h** in the image

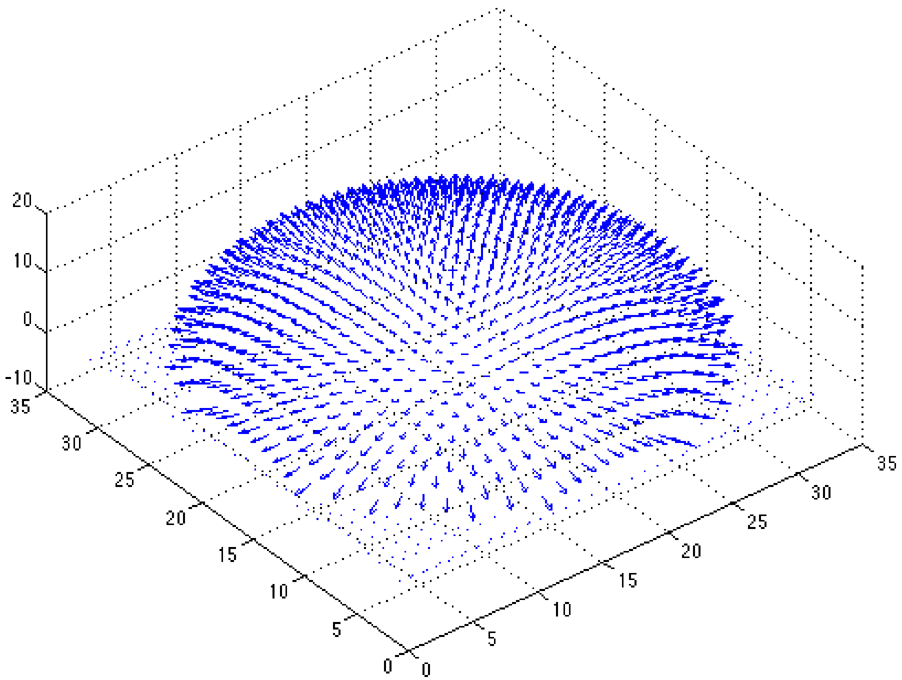



$$N = \begin{pmatrix} \frac{x_h - x_c}{r} \\ \frac{y_h - y_c}{r} \\ \frac{\sqrt{r^2 - x^2 - y^2}}{r} \end{pmatrix}$$

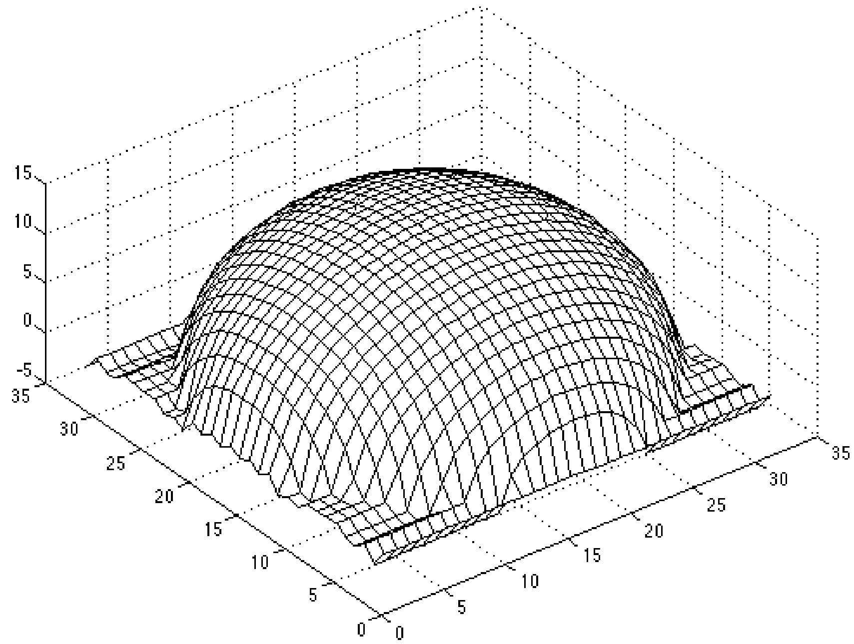
Can compute θ (and hence **N**) from this figure

Now just reflect **V** about **N** to obtain **L**

# Depth from normals




What we have




What we want

Forsyth & Ponce, Sec. 5.4