Computer Vision

2. Illumination

I-Chen Lin
College of Computer Science
National Yang Ming Chiao Tung University

Outline

- Radiometry
- Reflection model
- Photometric stereo

Textbook:

• David A. Forsyth and Jean Ponce, Computer Vision: A Modern Approach, Prentice Hall, New Jersey, (1st Ed. 2003, 2nd Ed. 2012).

Some contents are from the reference lecture notes:

• Prof. D.A. Forsyth, Computer Vision, UIUC.

• Prof. J. Rehg, Computer Vision, Georgia Inst. of Tech.

• Hearn and Baker, Computer Graphics, 3rd Ed., Prentice Hall

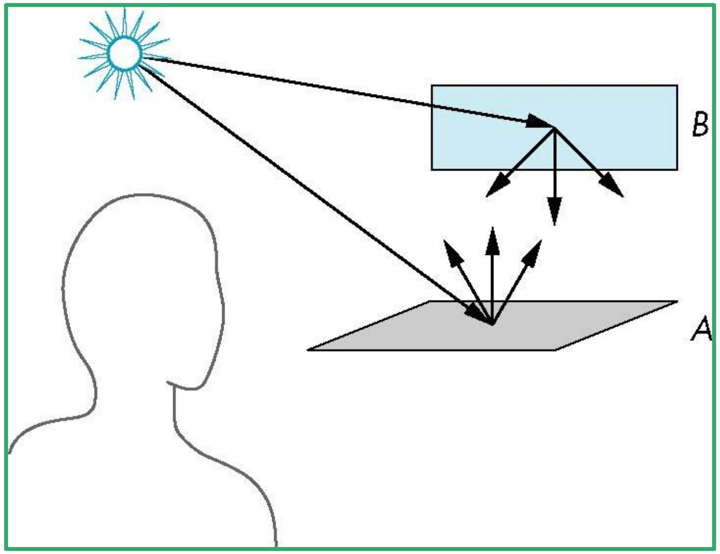
•E. Angel, Interactive Computer Graphics, 4th Ed., Addison Wesley

Illumination

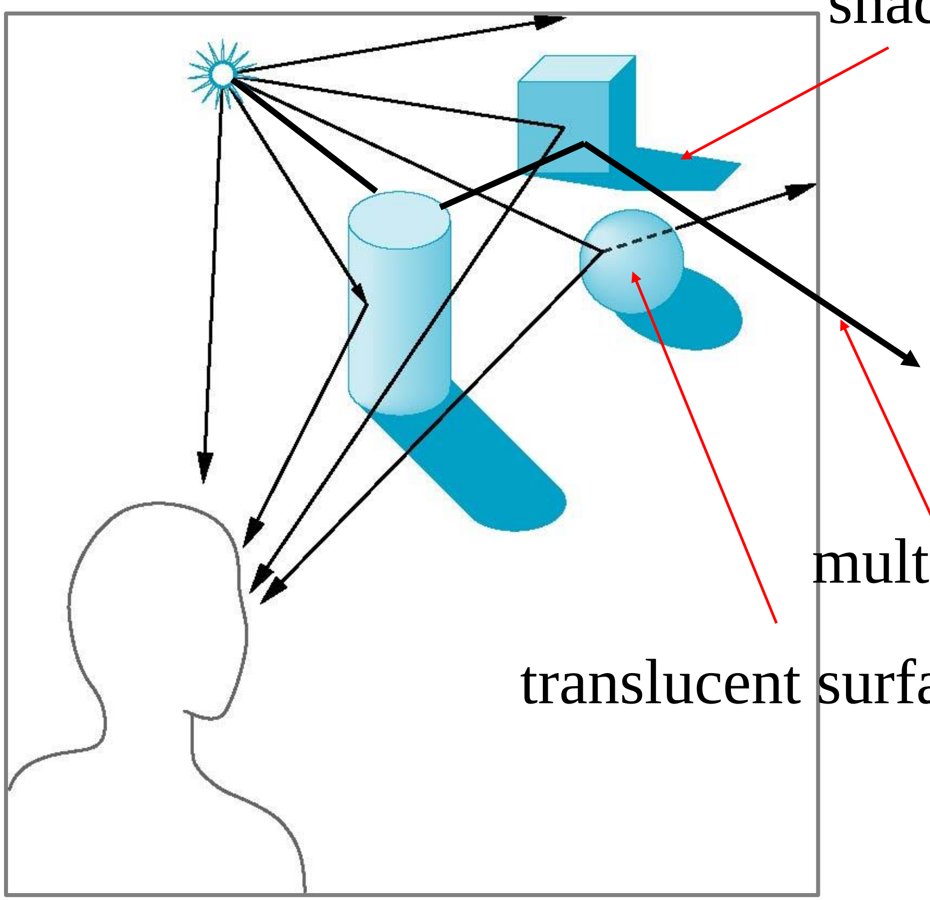
- Factors that affect the “color” of a pixel.
  - Light sources
    - Emittance spectrum (color)
    - Geometry (position and direction)
    - Directional attenuation
  - Objects’ surface properties
    - Reflectance spectrum (color)
    - Geometry (position, orientation, and micro-structure)
    - Absorption

Scattering

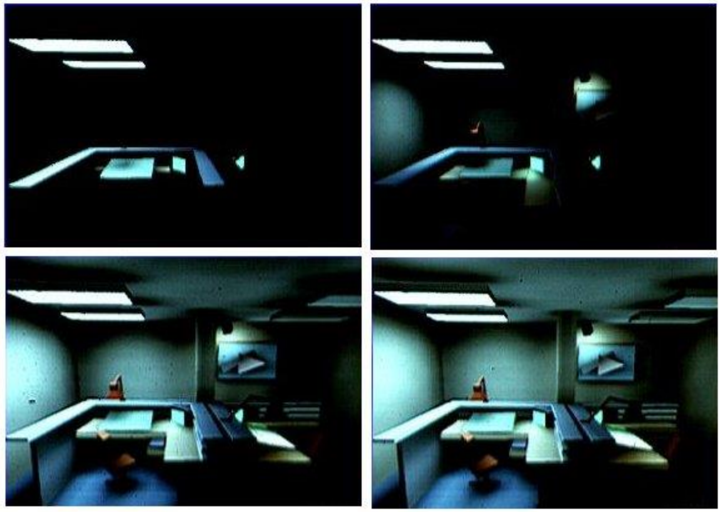
- Light strikes A
  - Some scattered
  - Some absorbed
- Some of scattered light strikes B
  - Some scattered
  - Some absorbed
- Some of this scattered light strikes A and so on

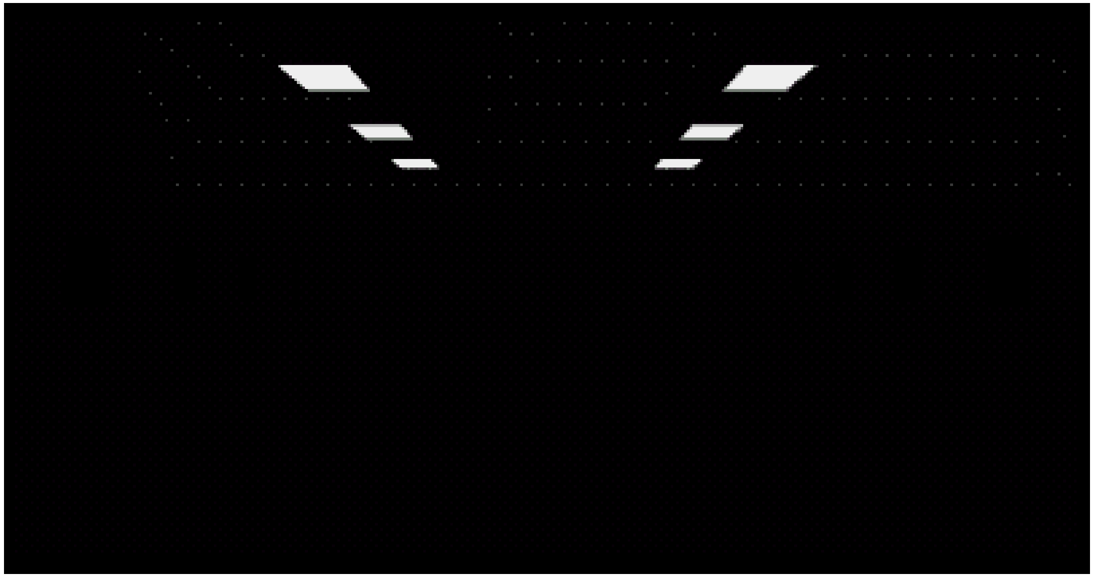


Global Effects



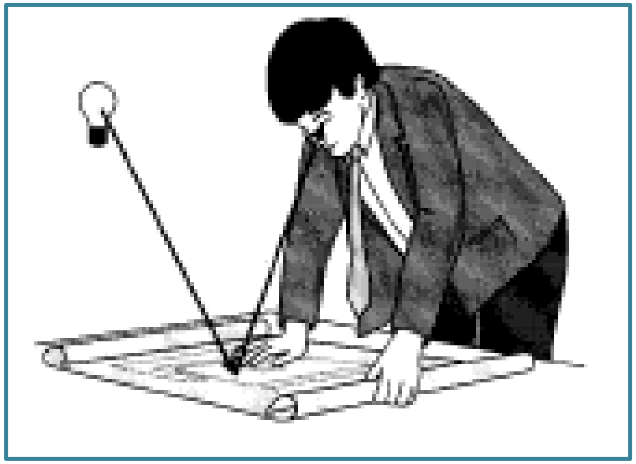
An example of the radiosity method

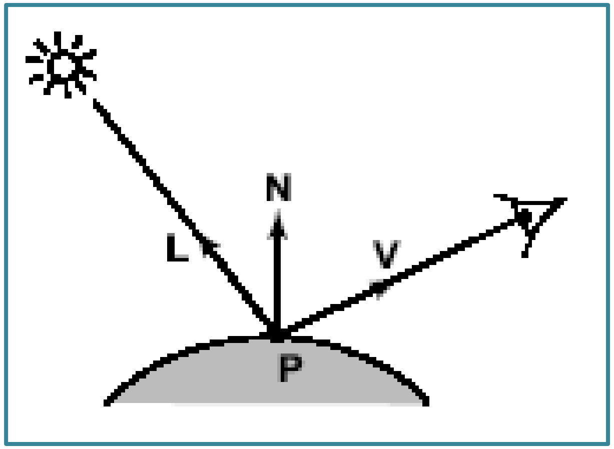




Local vs. global Illumination

- Correct illumination model requires a global calculation
- However, it is quite difficult to analyze a scene by such a complex model.
- Usually using a local illumination model instead.
    - No inter-reflection, no refraction, no precise shadow ….



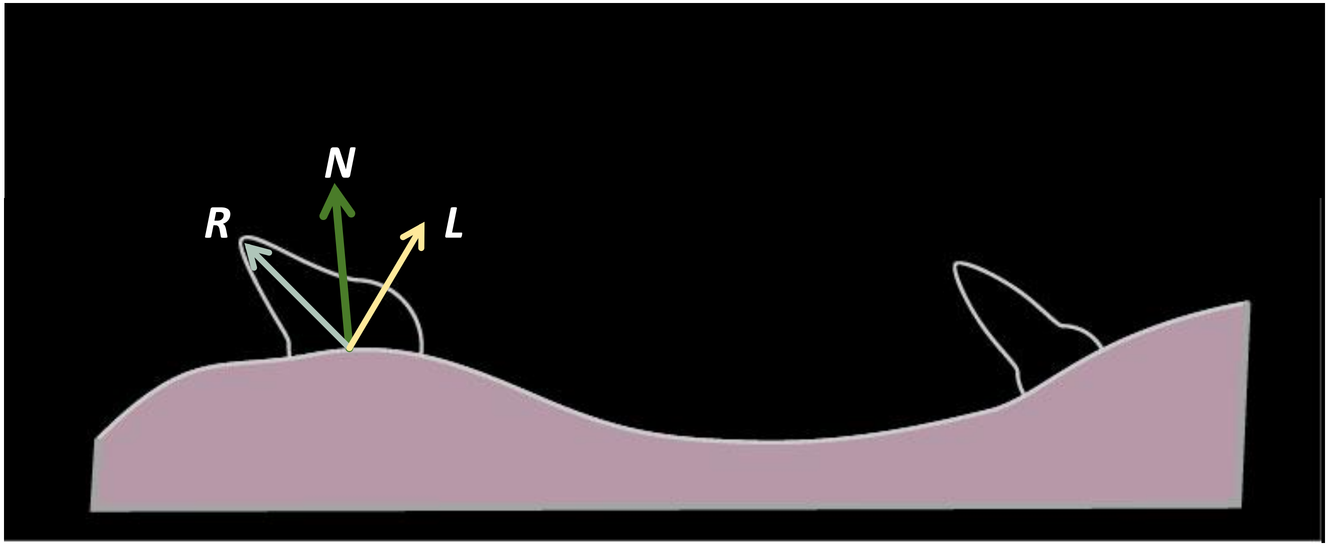


Simple light sources

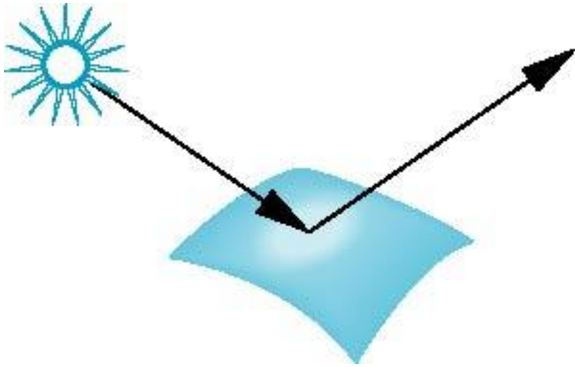
- Point source
  - Model with position and color
  - Distant source = infinite distance away (parallel)
- Spotlight
  - Restrict light from ideal point source
- Ambient light
  - Same amount of light everywhere in scene
  - Can model contribution of many sources and reflecting surfaces

Surface types

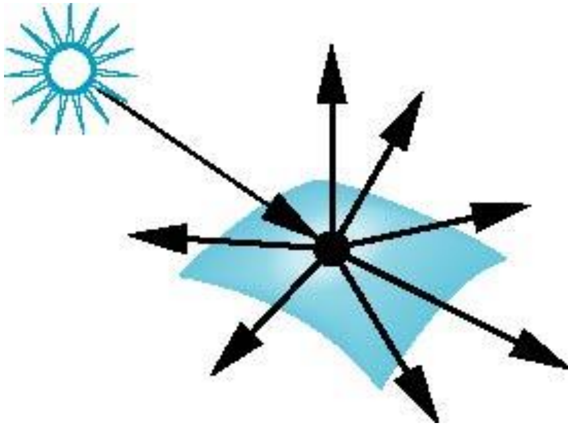
- The smoother a surface, the more reflected light is concentrated in the direction
- A very rough surface scatters light in all directions



Distribution of reflection



smooth surface



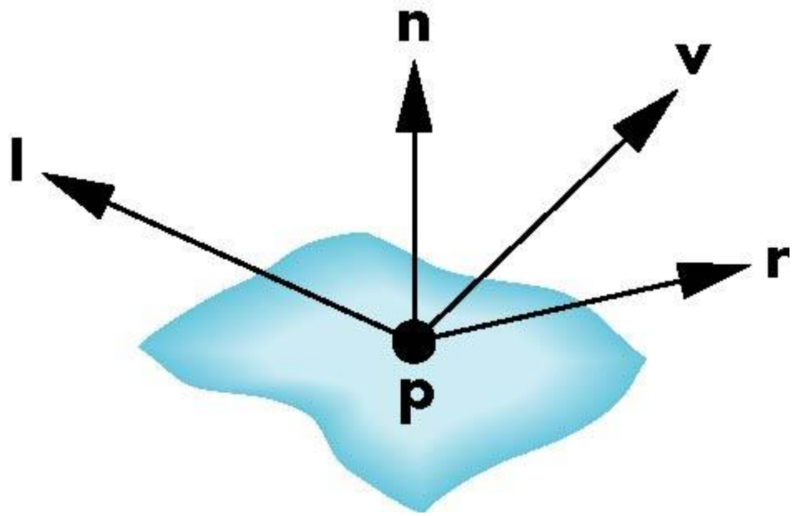
rough surface

Phong reflection model

- A simple model that can be computed or analyzed rapidly.

- Has three components
  - Ambient
  - Diffuse
  - Specular

- Uses four vectors
  - To source ***l***
  - To viewer ***v***
  - Normal ***n***
  - Perfect reflector ***r***



$$\mathbf{r} = 2\,(\mathbf{l} \cdot \mathbf{n}\,)\,\mathbf{n} - \mathbf{l}$$

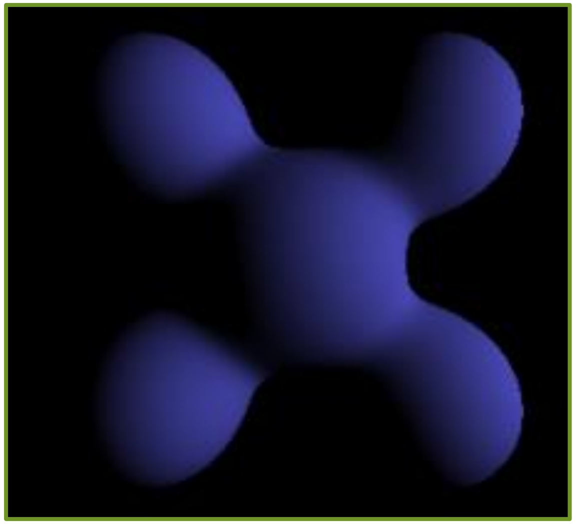
Ambient light

- The result of multiple interactions between (large) light sources and the objects in the environment.

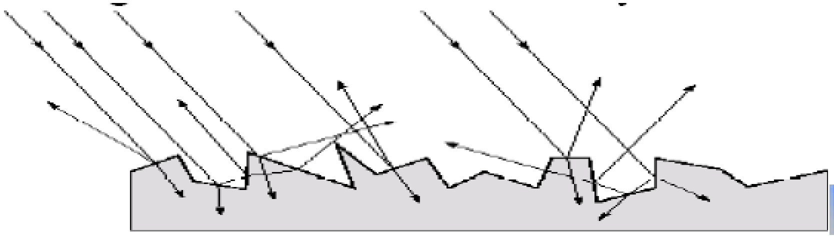
- $I_{ambient} = K_a \cdot I_a$

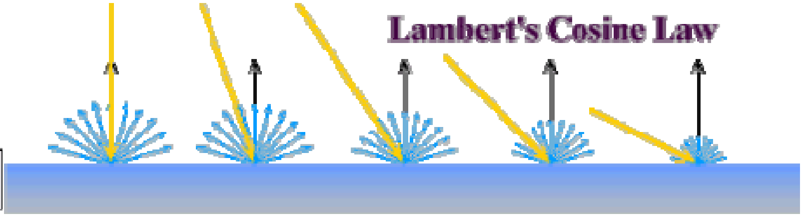
Diffuse reflection

- Light scattered equally in all directions
- Reflected intensities vary with the direction of the light.
- **Lambertian Surface**
  - Perfect diffuse reflector



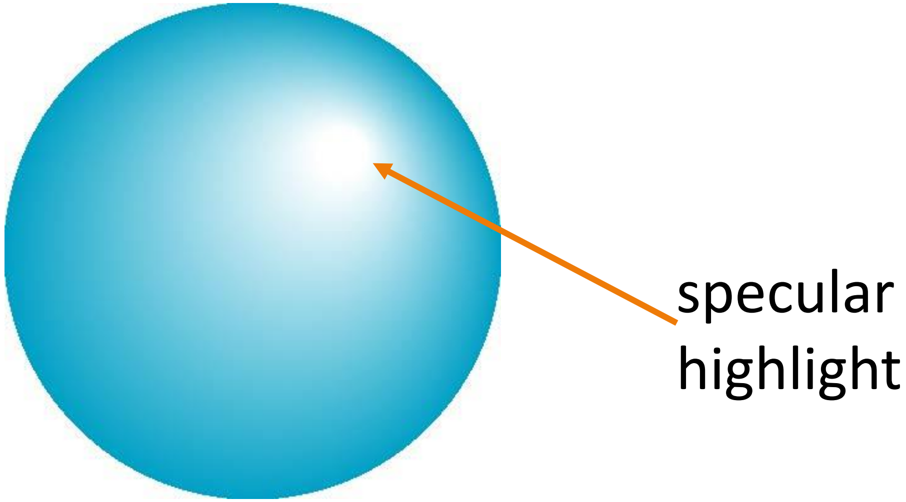
- $I_{diffuse} = K_d \cdot I_d\,(n \cdot l)$





Specular Surfaces

- Most surfaces are neither ideal diffusers nor perfectly specular (ideal reflectors)
- Incoming light being reflected in directions concentrated close to the direction of a perfect reflection



Modeling specular reflections

- Phong proposed

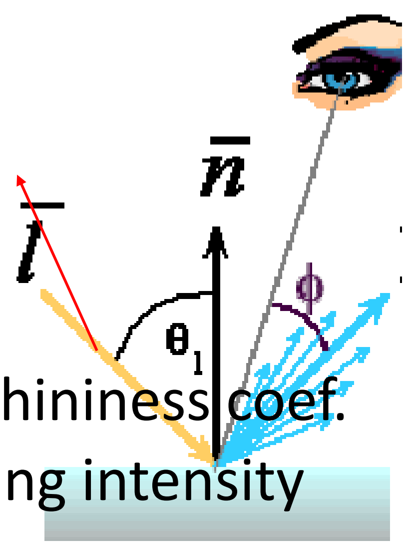
$$I_r \sim k_s\, I \cos^{a} \Phi$$

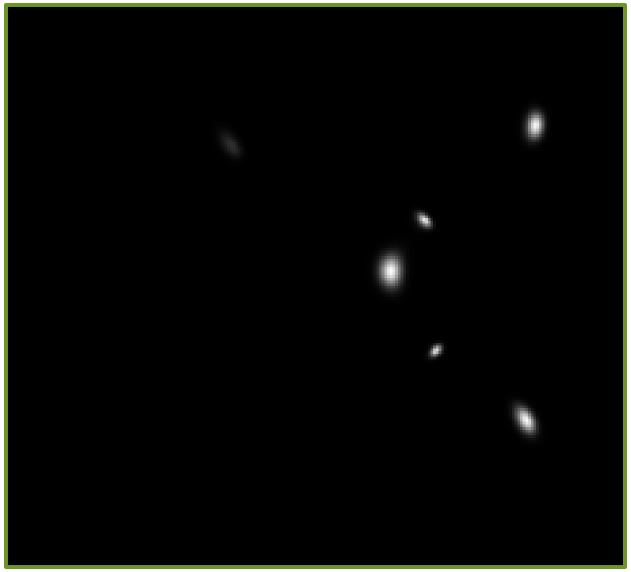
reflected intensity

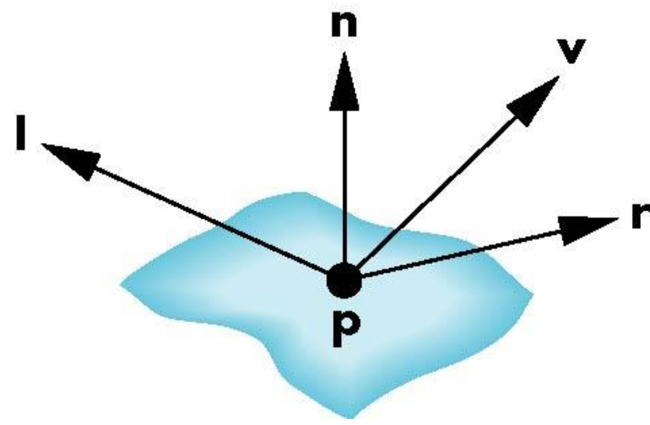
absorption coef

incoming intensity

shininess coef.

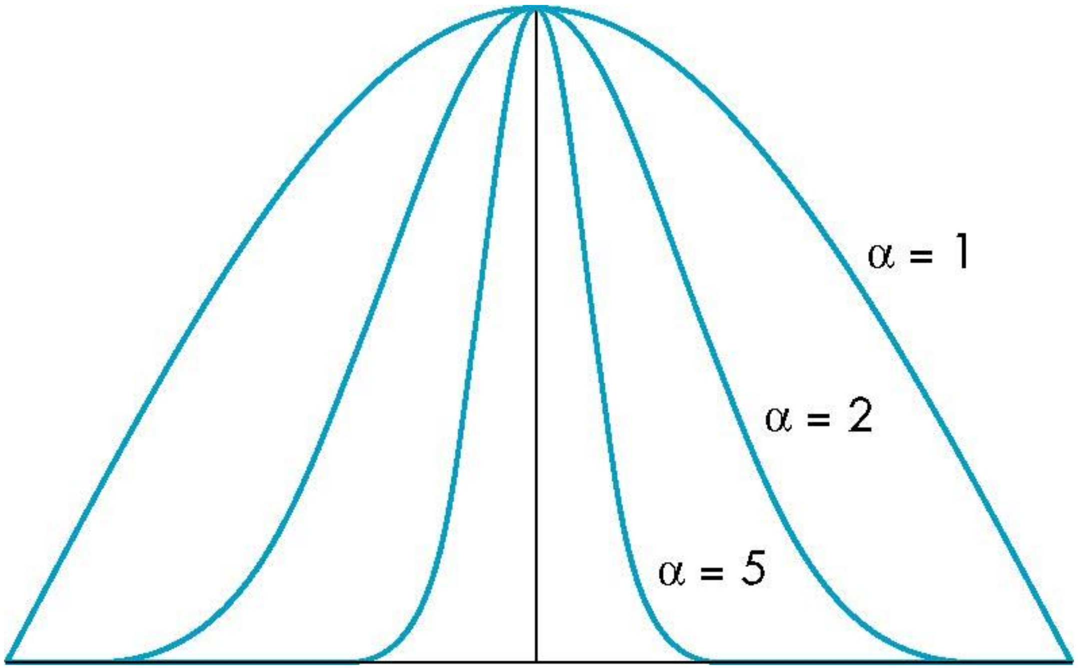






The shininess coefficient

- Values of a between 100 and 200 correspond to metals
- Values between 5 and 10 give surface that look like plastic



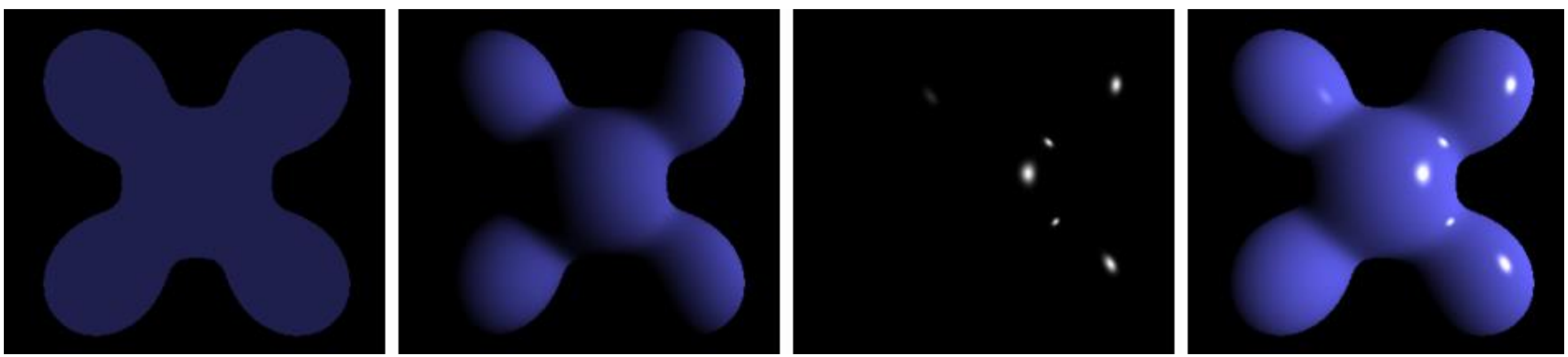
Coefficients

- 9 coefficients for each point light source
  - $I_{dr}, I_{dg}, I_{db}, I_{sr}, I_{sg}, I_{sb}, I_{ar}, I_{ag}, I_{ab}$
- Material properties
  - Nine absorbtion coefficients
    - $k_{dr}, k_{dg}, k_{db}, k_{sr}, k_{sg}, k_{sb}, k_{ar}, k_{ag}, k_{ab}$
  - Shininess coefficient a

Adding up the components

- A primitive virtual world with lighting can be shaded by combining the three light components .
- $I = I_{ambient} + I_{diffuse} + I_{specular}$

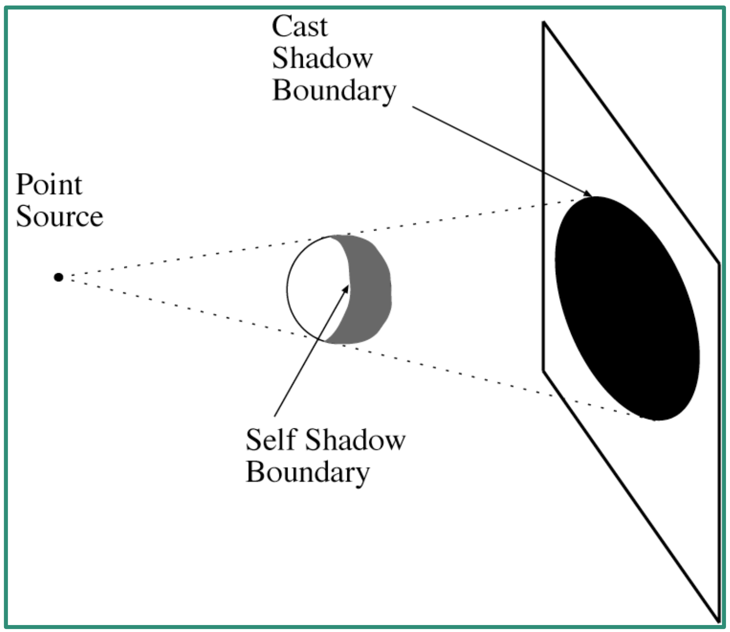
  $= k_a I_a + k_d I_d(l \cdot n) + k_s I_s (v \cdot r)^a$

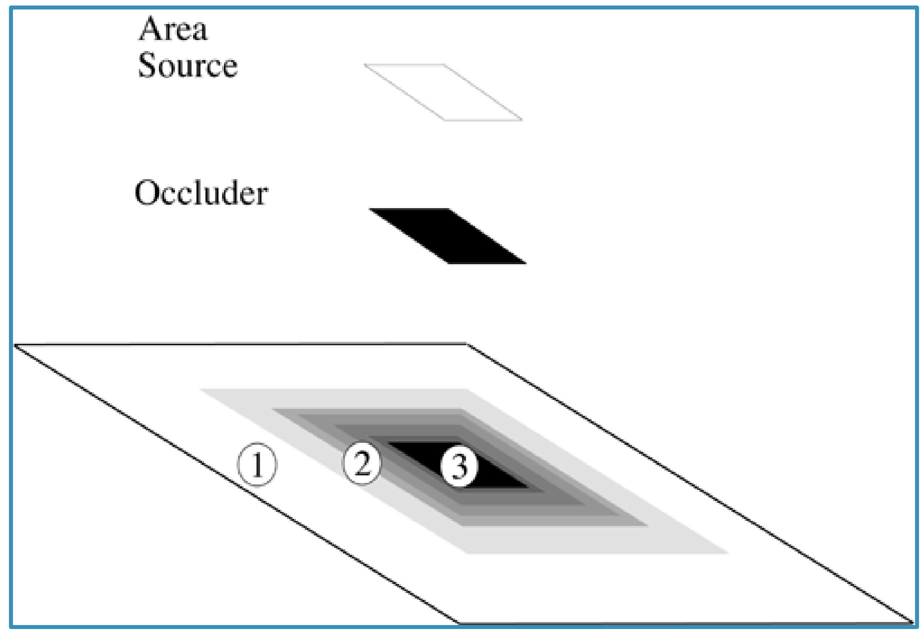


Ambient + Diffuse + Specular = Phong reflection

Shadows

- To calculate shadows, we must take into account visibility and occlusion.

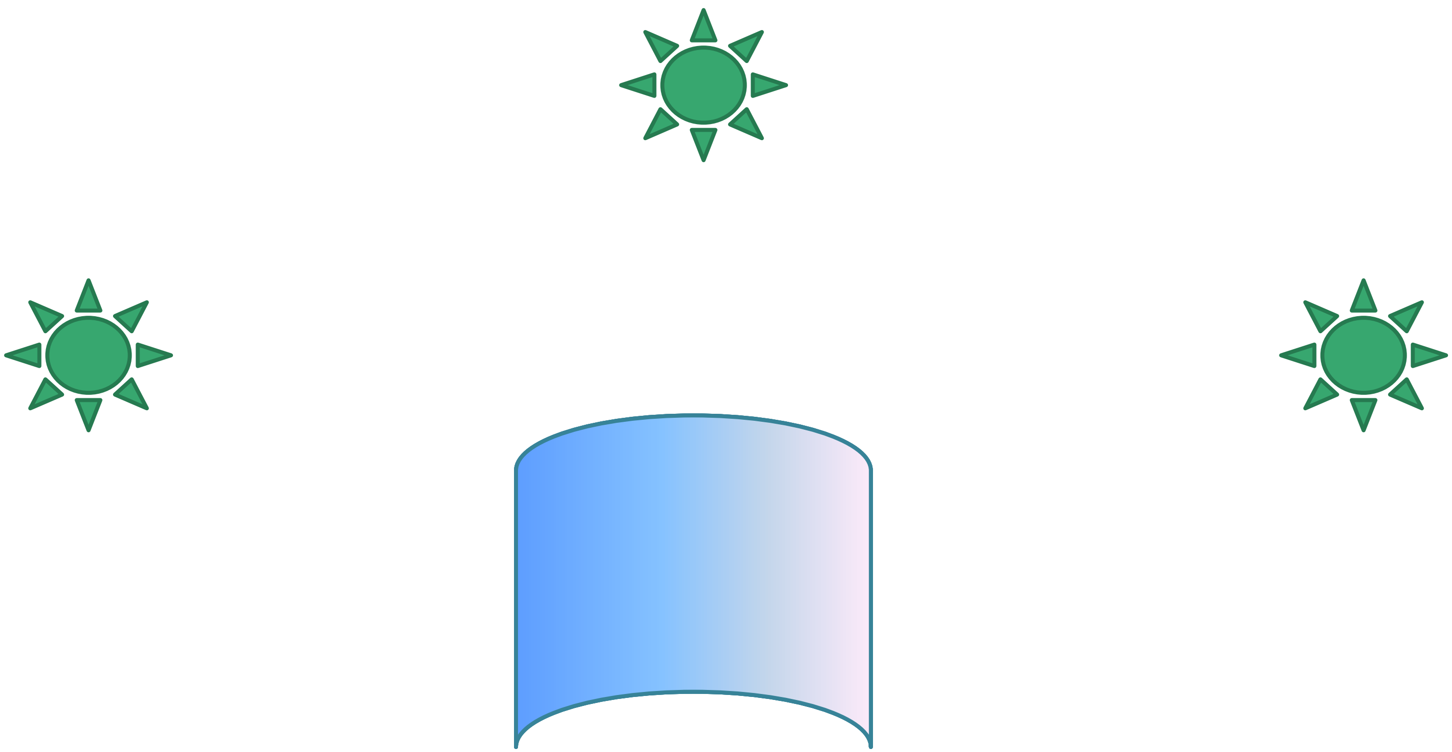




Penumbra and umbra

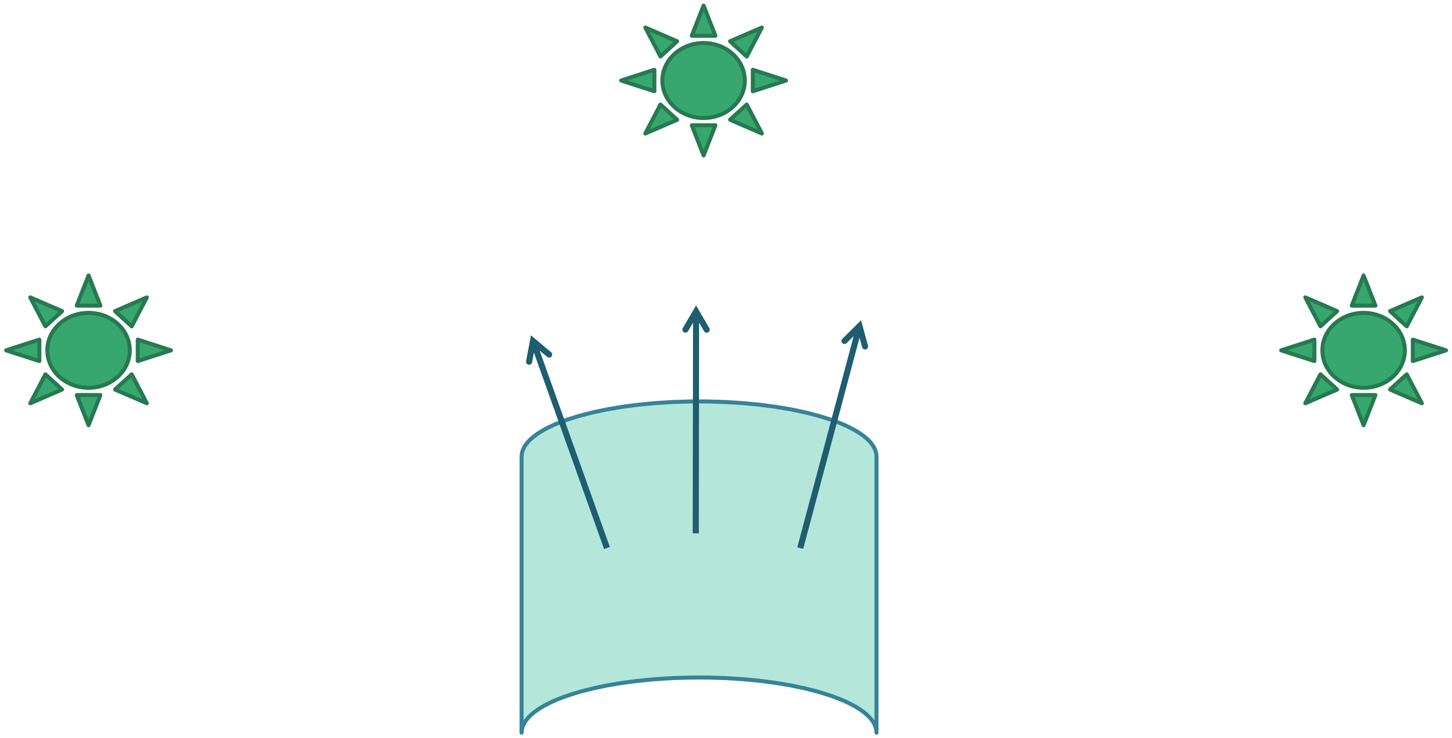
Photometric stereo

- Given multiple images of the same surface under different known lighting conditions, can we recover the surface shape?



Photometric stereo (cont.)

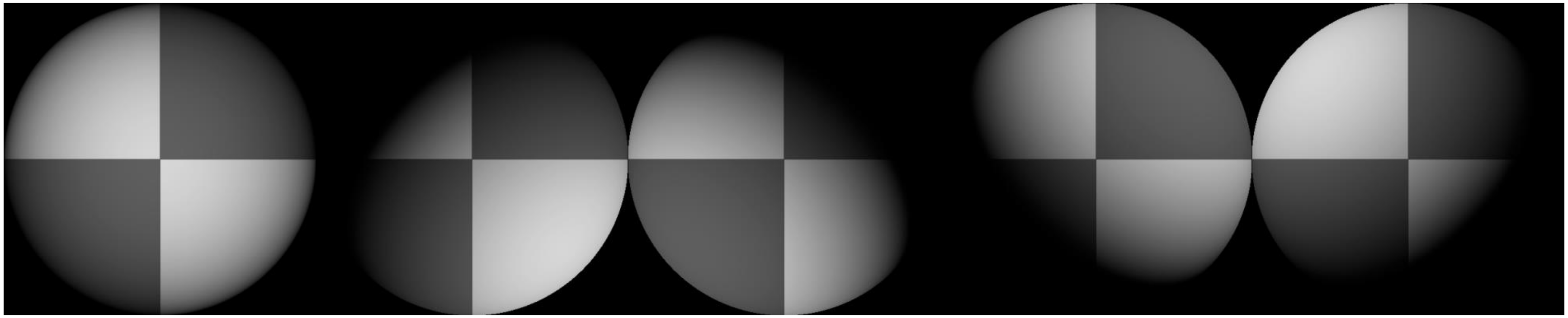
- Reminder: surface reflection is related to surface normal N and light source L (and view direction V in specular reflection)



Photometric stereo (cont.)

- Using a local shading model
- A set of point sources that are infinitely distant
- A set of pictures of an object, obtained in exactly the same camera/object configuration but using different sources
- A Lambertian (diffuse) object for simplification
  - (or the specular component has been identified and removed)

$$\boxed{k_d\, I_d(\boldsymbol{l} \cdot \boldsymbol{n})}$$



Photometric stereo (cont.)

- For pixel (x, y) at image i,

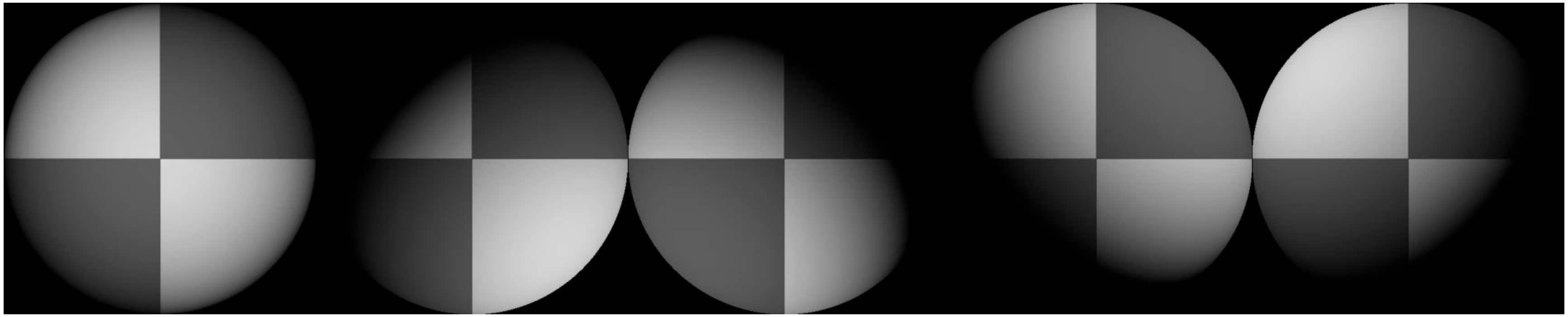
$$I_i(x,y) = \rho(x,y) S_i \cdot N(x,y) \qquad b(x,y) = \rho(x,y) N(x,y)$$

, where ρ is the albedo (k_d), and S_i is the light source vector.

$$\begin{bmatrix} I_1(x,y) \\ I_2(x,y) \\ \vdots \\ I_n(x,y) \end{bmatrix} = \begin{bmatrix} S_1^T \\ S_2^T \\ \vdots \\ S_n^T \end{bmatrix} b(x,y)$$

An over-determined linear system, for n>3
Can be solved by pseudo-inverse or other methods.

$$\boxed{\begin{aligned} Ux &= y \\ x &= \left(U^T U\right)^{-1} U^T y \end{aligned}}$$



Photometric stereo (cont.)

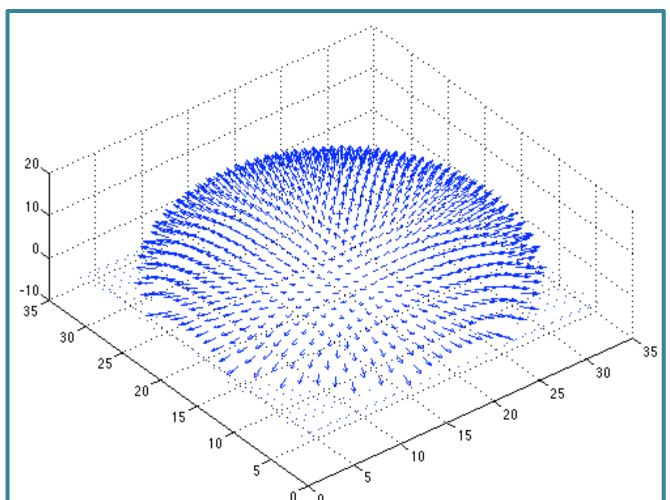
- Pre-multiplying by a thresholded weight matrix zeros the contributions from shadowed pixels

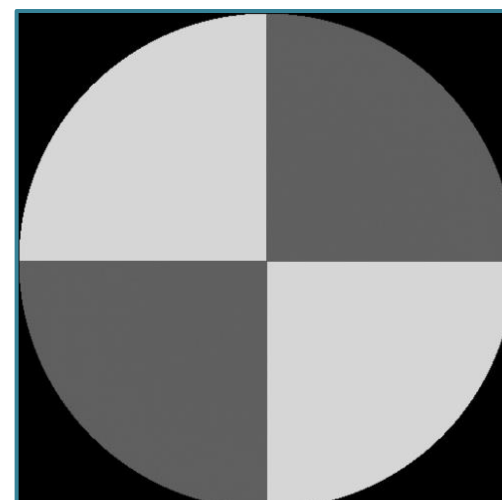
$$\begin{bmatrix} w_1(x,y) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n(x,y) \end{bmatrix} \begin{bmatrix} I_1(x,y) \\ \vdots \\ I_n(x,y) \end{bmatrix} = \begin{bmatrix} w_1(x,y) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n(x,y) \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_n^T \end{bmatrix} b(x,y)$$

- Recovering normals and albedos by

$$N(x,y) = \frac{b(x,y)}{\|b(x,y)\|}$$

$$\rho(x,y) = \|b(x,y)\|$$

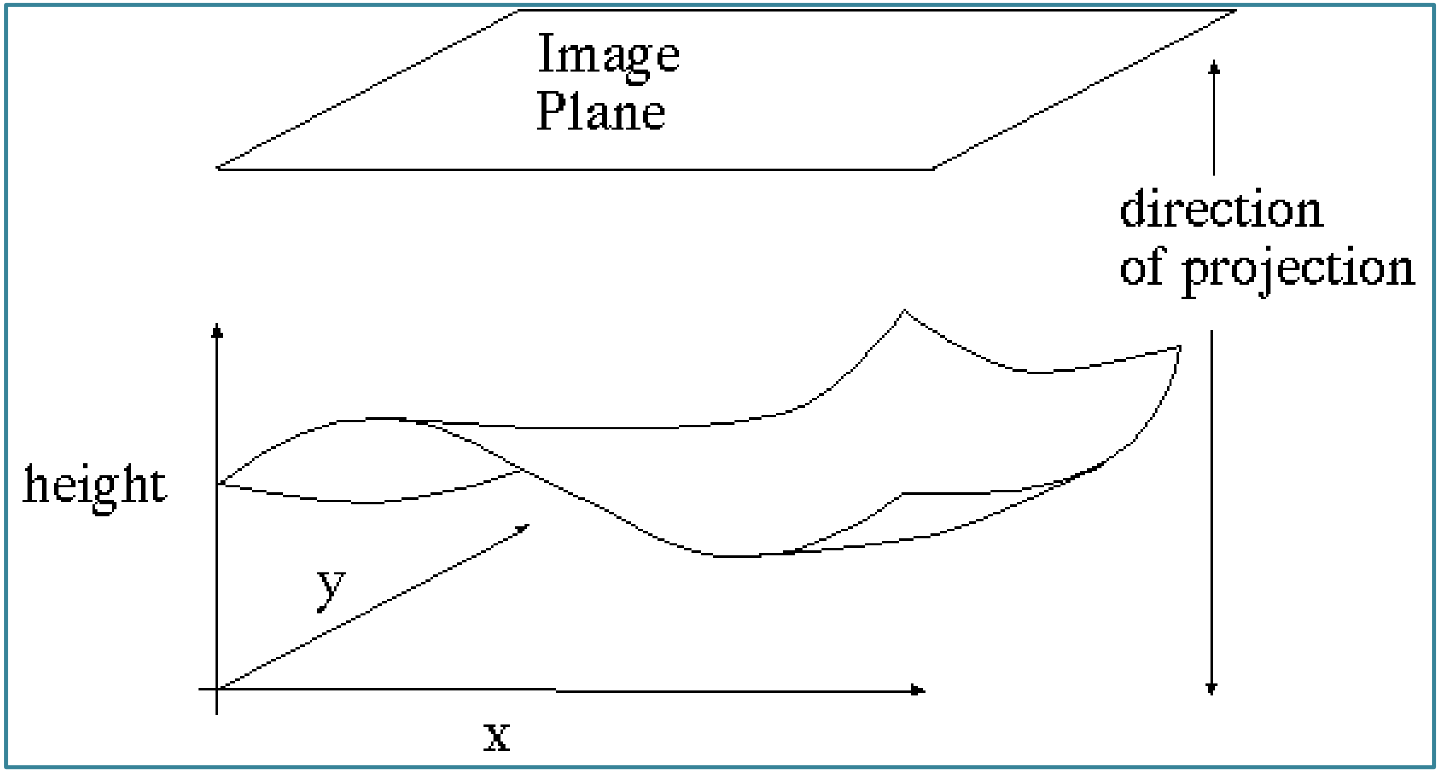




Recovering the surface (depth)

- Depth map:

  $z = f(x, y)$



Recovering the surface (depth)

- The surface can be represented as (x, y, $f(x,y)$).
- From the surface gradient vectors, we can evaluate the surface normal as:

$$\boxed{N(x,y)=\frac{\left(-\frac{\partial f}{\partial x},\ -\frac{\partial f}{\partial y},\ 1\right)^T}{\sqrt{1+\frac{\partial f}{\partial x}^2+\frac{\partial f}{\partial y}^2}}}$$

- Therefore, if estimated $N(x,y)$ is ($N_a(x,y)$, $N_b(x,y)$, $N_c(x,y)$)^T, we get the partial derivative:

$$\boxed{\frac{\partial f}{\partial x}=\frac{-N_a(x,y)}{N_c(x,y)}} \qquad \boxed{\frac{\partial f}{\partial y}=\frac{-N_b(x,y)}{N_c(x,y)}}$$

Check the derivatives

Since $$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$$

- We have to check whether the numerical 2nd order derivatives are close to each other

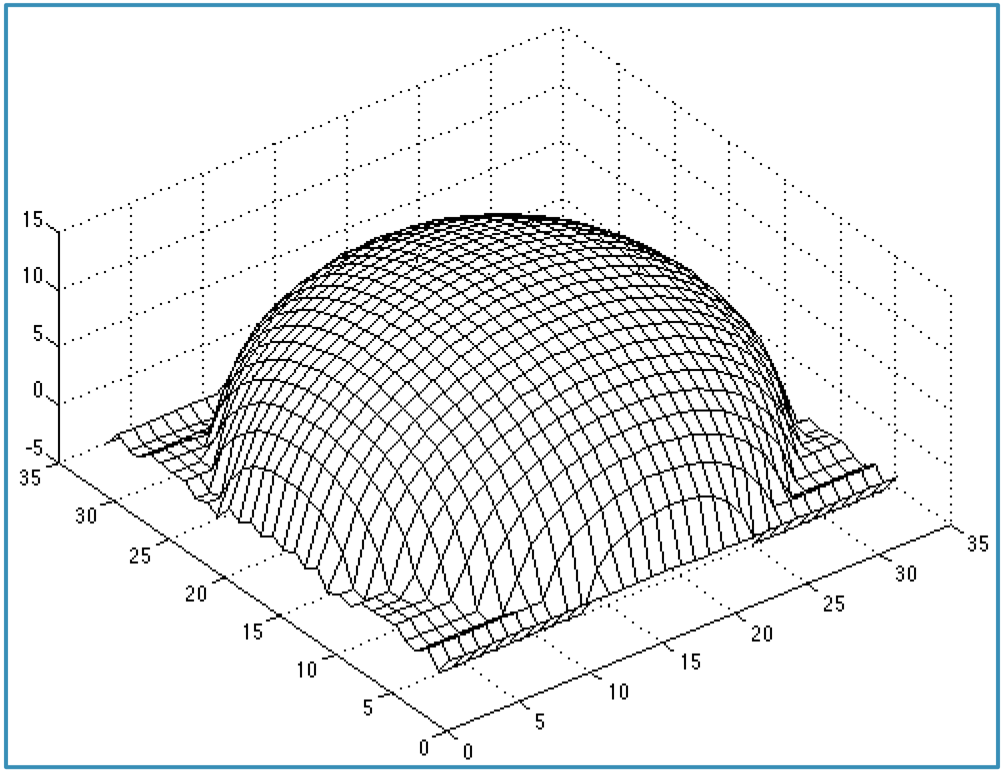
$$\left(\frac{\partial \left(\frac{-N_a(x,y)}{N_c(x,y)} \right)}{\partial y} - \frac{\partial \left(\frac{-N_b(x,y)}{N_c(x,y)} \right)}{\partial x} \right)^2$$

- Then, we can reconstruct the surface by integration along some paths, e.g:

$$f(u,v) = \int_0^v \frac{\partial f}{\partial y}(0,y)dy + \int_0^u \frac{\partial f}{\partial x}(x,v)dx + c$$

Recovering the shape

- Is there any problem or difficulty for real objects?



Recovering the shape (appendix)

- Problems:
  - Different integral paths may result in different surfaces.
  - Noise from digitization, sampling, etc.

- Modern research usually formulates the problem as an optimization process for depth *z* with smoothness penalties.

  For instance, specify *z* values around the image boundary, and find the depths within the image by optimization.

$$\arg\min_z O = \sum_p \left(\frac{\partial f_p}{\partial x} - \nabla_x z_p \right)^2 + \sum_p \left(\frac{\partial f_p}{\partial y} - \nabla_y z_p \right)^2 + \lambda \sum_p \left(z_p - \frac{1}{\|q\|} \sum_{q \in p\_Neighbor} z_q \right)^2$$

$\frac{\partial f_p}{\partial x}$ and $\frac{\partial f_p}{\partial y}$ are evaluated from normals.