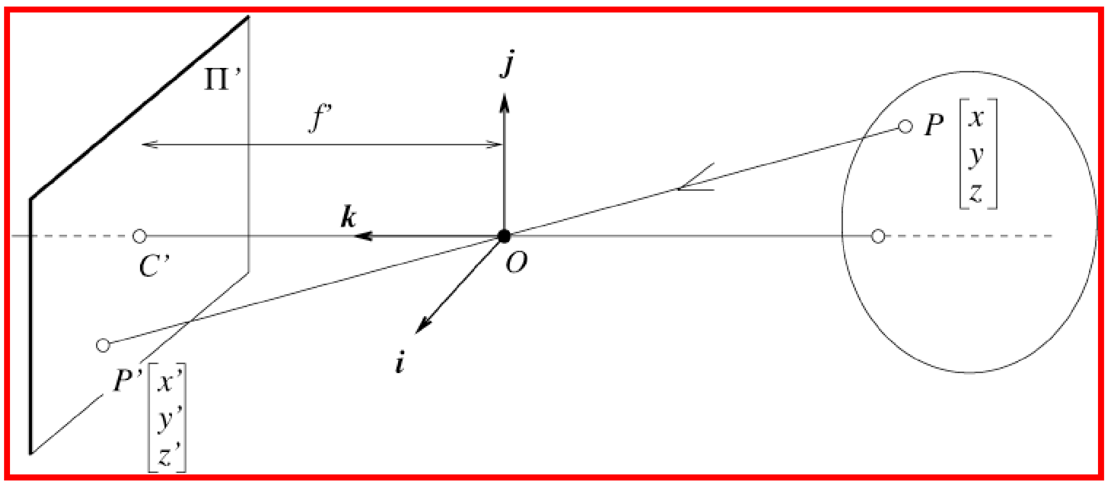
Radiometry

Nuno Vasconcelos
UCSD

Image formation

- two components: geometry and radiometry
- geometry:
  - pinhole camera
  - point (x,y,z) in 3D scene projected into image pixel of coordinates (x’, y’)
  - according to the perspective projection equation:

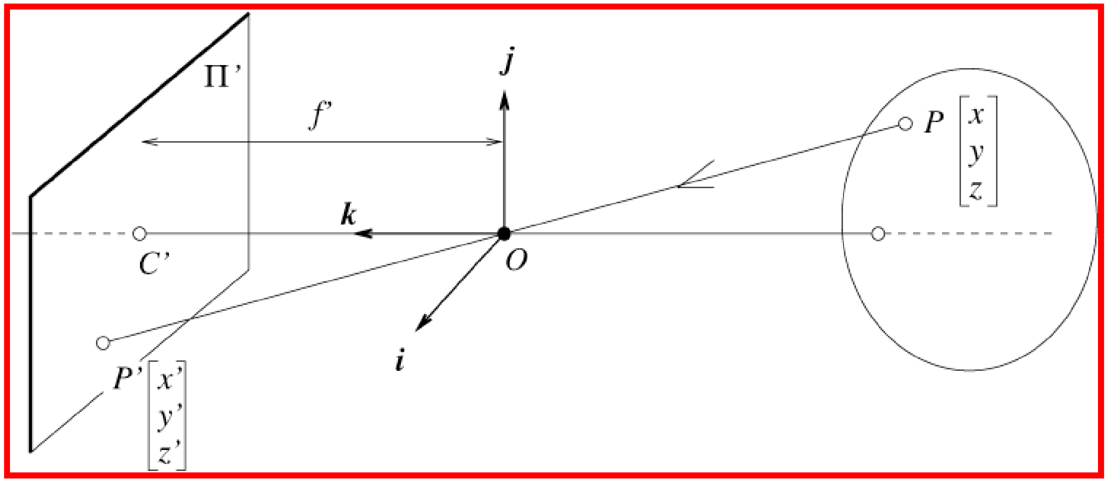


$$\begin{pmatrix} x' \\ y' \end{pmatrix} = f(X, Y, Z)$$

What is f(X,Y,Z)?

Image formation

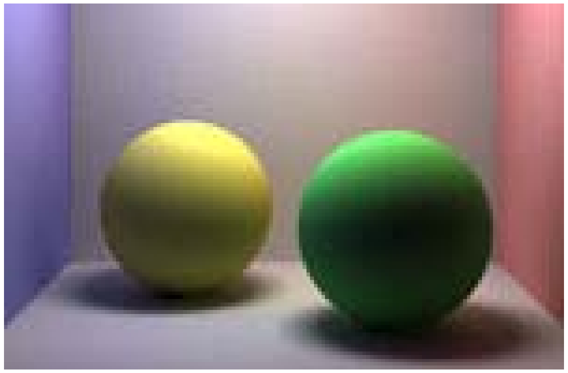
- two components: geometry and radiometry
- geometry:
  - pinhole camera
  - point (x,y,z) in 3D scene projected into image pixel of coordinates (x', y')
  - according to the perspective projection equation:



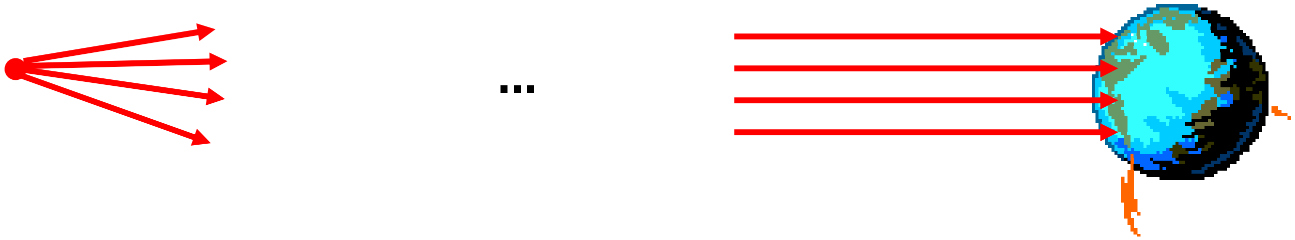
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = f \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

Radiometry

- simplifying assumptions
- objects:
  - Lambertian surfaces: reflect light equally in all directions
  - reflections determined by albedo: ratio of reflected/incident light



- light sources:
  - point source @ infinity

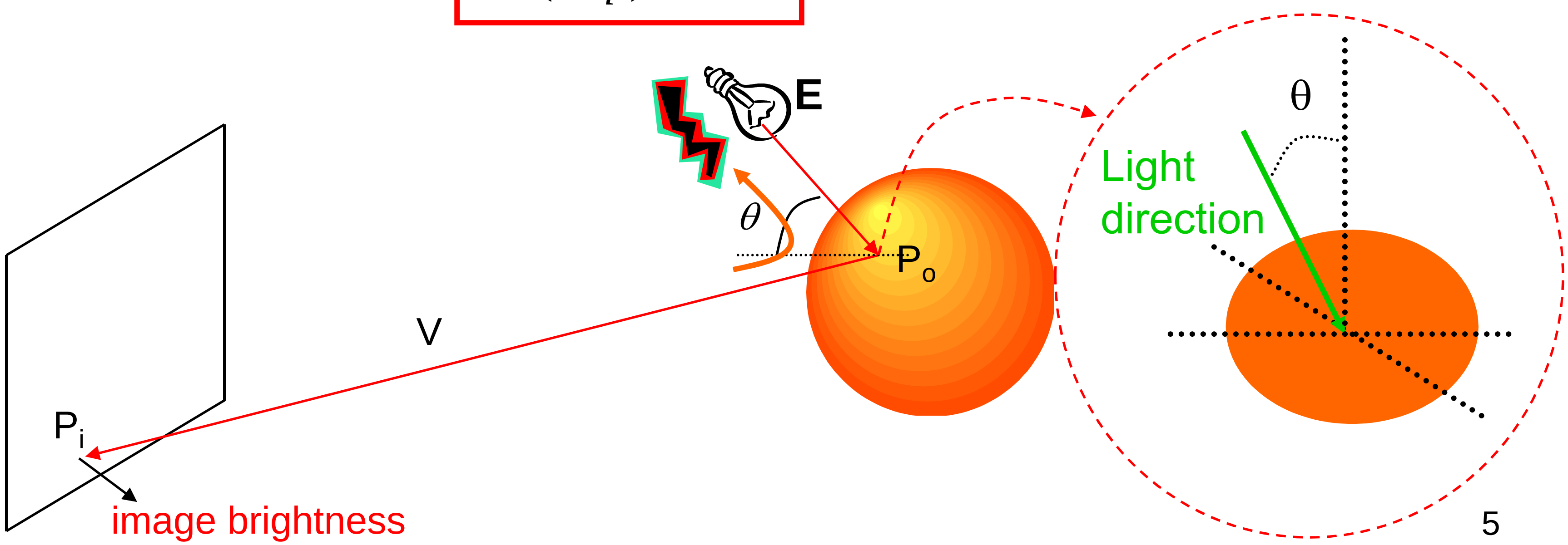


  - by the time they hit the object all rays are parallel
  - single direction ***s*** of light for the whole scene

Radiometry

- radiometry equation:
- the power at pixel P_i is a function of
  - object properties (which?)
  - light source (which)
  - interaction between the two?

$$P(P_i) = ?$$

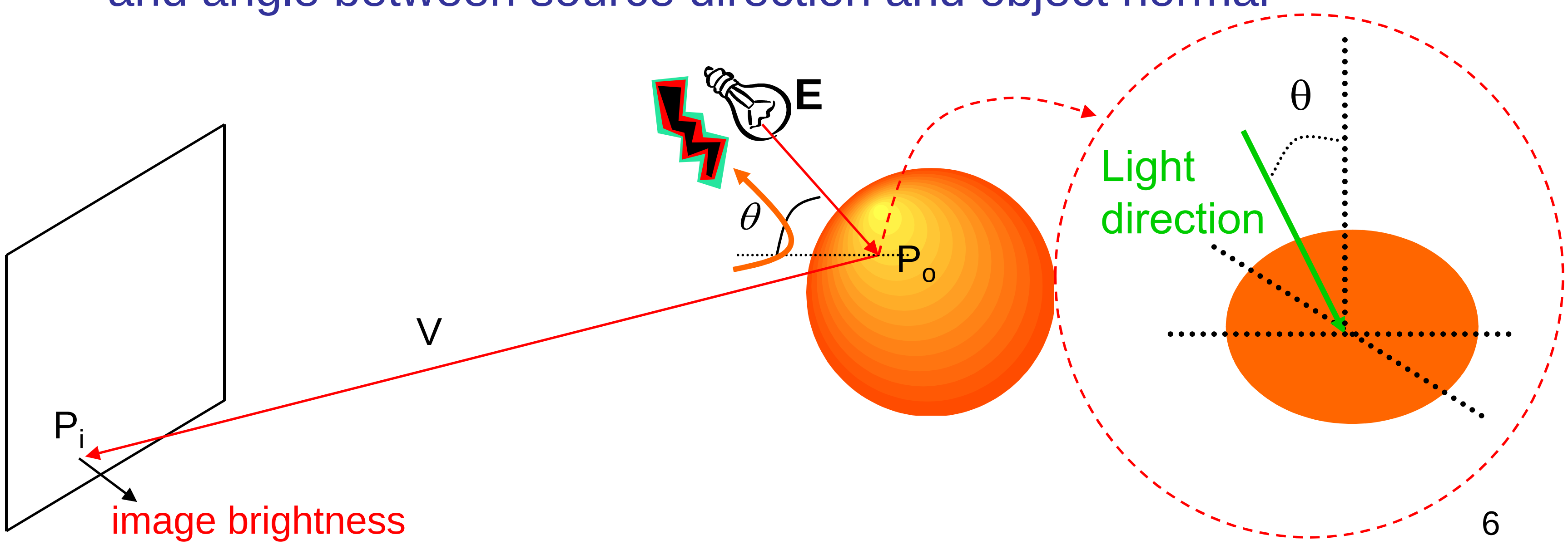


Radiometry

- radiometry equation:

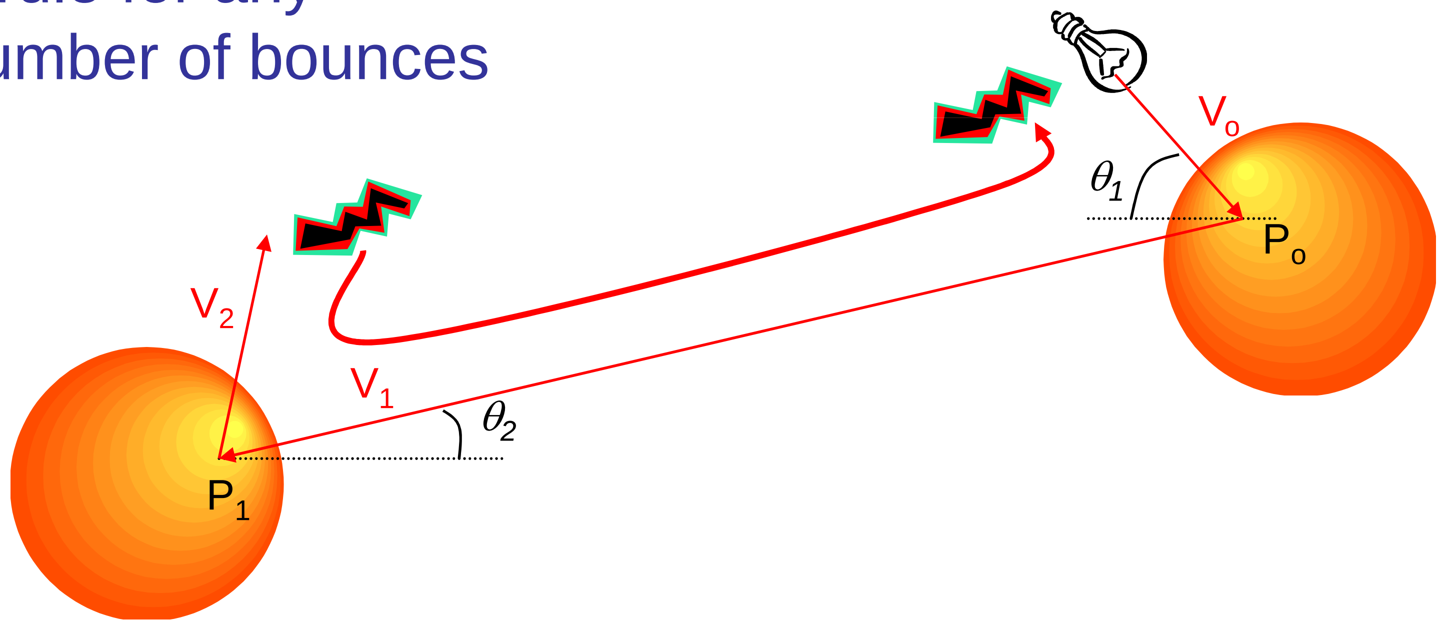
$$P(P_i) = E\rho(P_0)\cos\theta$$

- the power at pixel P_i is the product of
  - source power E,
  - albedo of the object at reflection point,
  - and angle between source direction and object normal



Radiometry

- a rule for any number of bounces



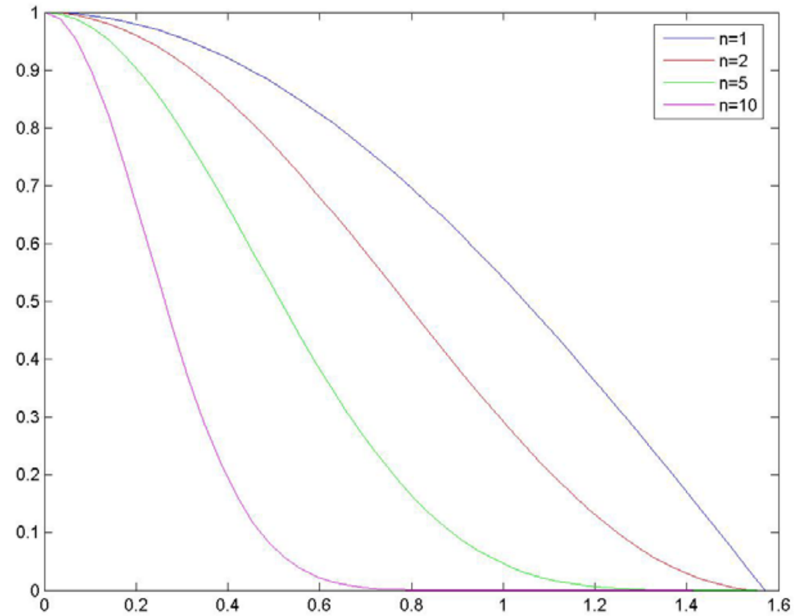
$$P(P_n) = E\left[\prod_{i=0}^{n} \rho(P_i)\right]\left[\prod_{i=1}^{n+1} \cos\theta_i\right]$$

Lambertian surfaces

- note that on

$$P(P_n) = E\left[\prod_{i=0}^{n} \rho(P_i)\right]\left[\prod_{i=1}^{n+1} \cos\theta_i\right]$$

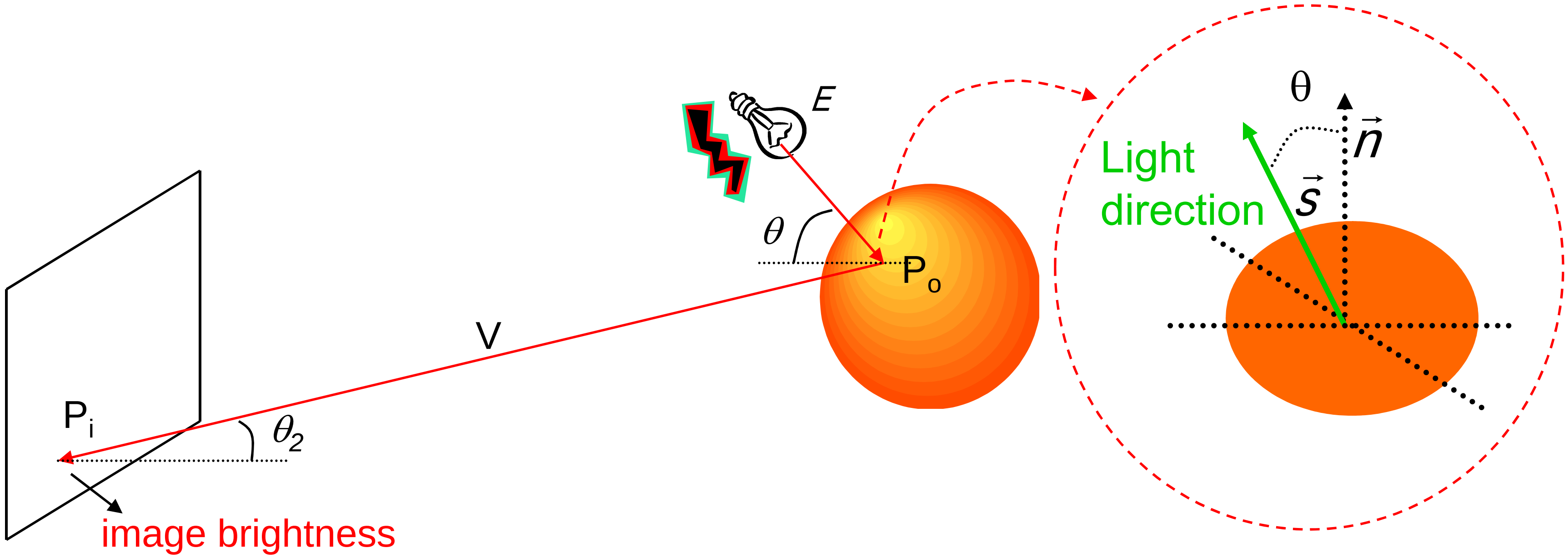
  - unless all cosines are close to 1
  - their product goes to zero quickly
  - e.g. see decay of $\cos^n(\theta)$ with n
  - this means that only light that arrives frontally to all the bounces gets propagated very far
  - such an alignment is very unlikely
  - we don't really have to worry about many bounces
  - the process becomes tractable



In vector form

- note that
  - if ***n*** is the surface normal and ***s*** the light direction
  - the two vectors have unit norm
  - then *cos* θ = ***n . s*** and

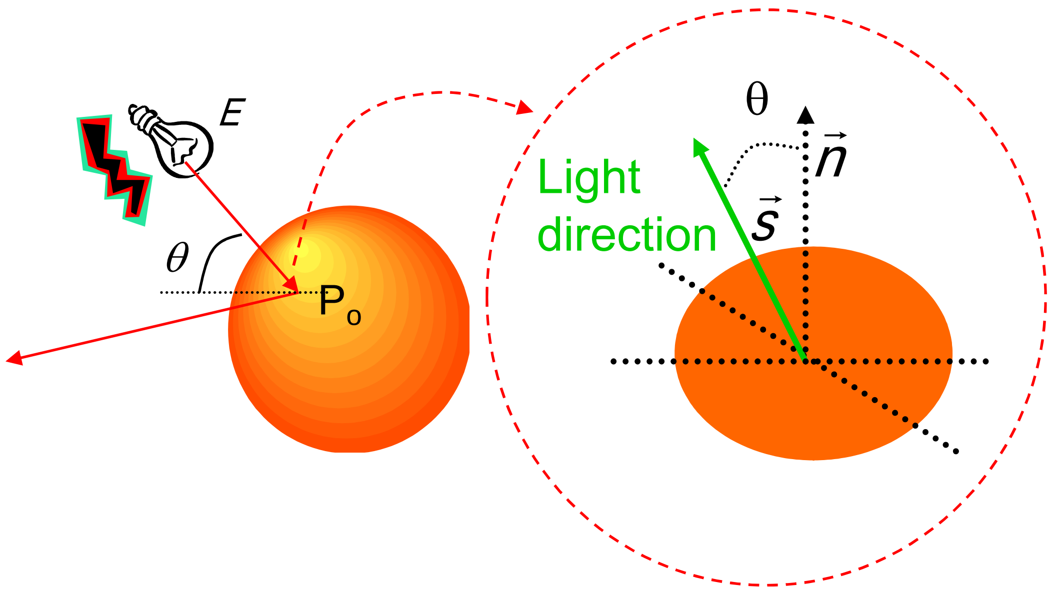
$$P(P_i) = E\rho(P_0)\vec{n}(P_0).\vec{s}$$

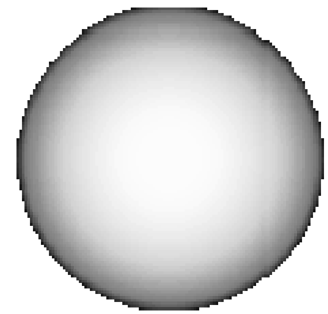


Lambertian surfaces

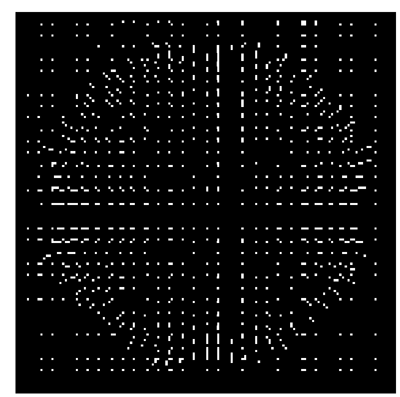
- note that
  - light direction ***s*** is constant
  - but the surface normal ***n*** and the albedo ρ are functions on the object surface

$$P(P_i) = E\rho(P_0)\vec{n}(P_0).\vec{s}$$





surface normal map



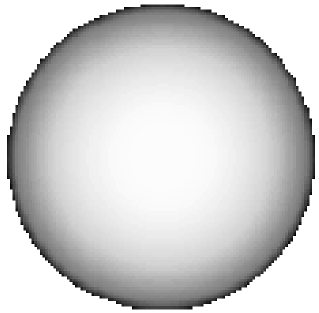
Normals

- the amount of reflected light at *P* depends on the surface normal *n(P)*
- how do I compute this?
- first we note that a surface is a function of three variables

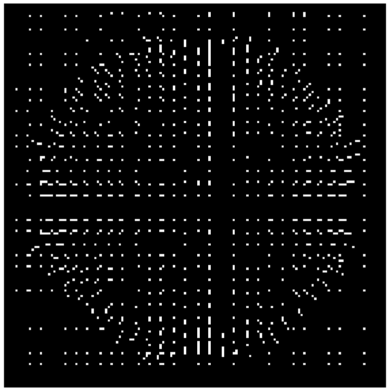
$$f(x,y,z) = 0$$

- e.g., the sphere of radius r and center (x_0,y_0,z_0) is described by

$$(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 - r^2 = 0$$



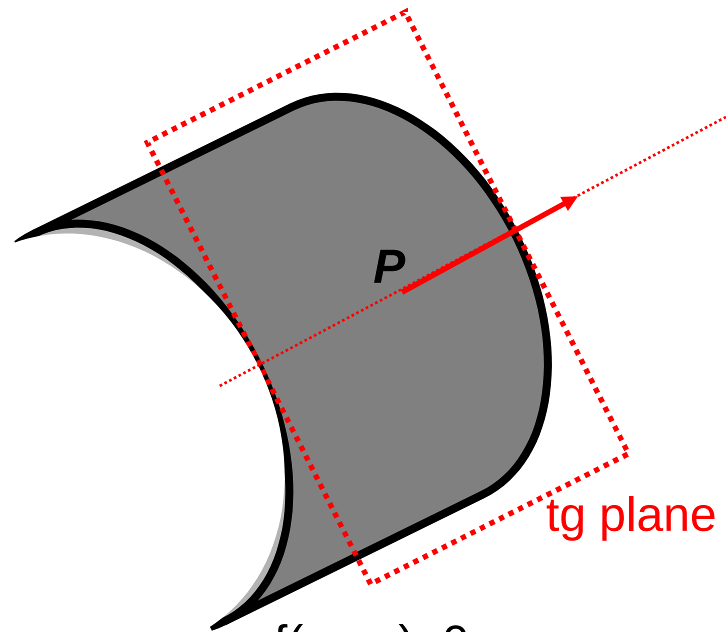
surface normal map



Normals

- the key is to note that
  - the normal is orthogonal to the surface
  - the surface is the set of points where the function f(x,y,z) is constant
  - to "walk" along the surface I "walk" along the set of points where f(x,y,z) stays constant

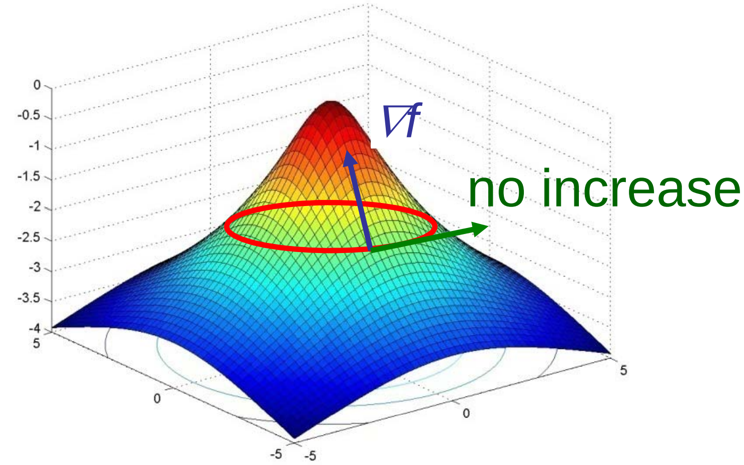
    (infinitesimally, this is the plane tangent to the surface at P)
  - to "walk" along the normal, I have to "walk" in the direction along which f(x,y,z) grows most quickly
  - the key is then to find this direction of largest growth



Normals

- recall that
  - a scalar function grows the most when its derivative is largest
- the generalization to a multivariate function is the gradient

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)^T$$



- *f* increases the most in the direction of its gradient
- hence,

"the normal to surface f(x,y,z) = 0 at point P is the gradient of f(x,y,z) evaluated at P"

Normals

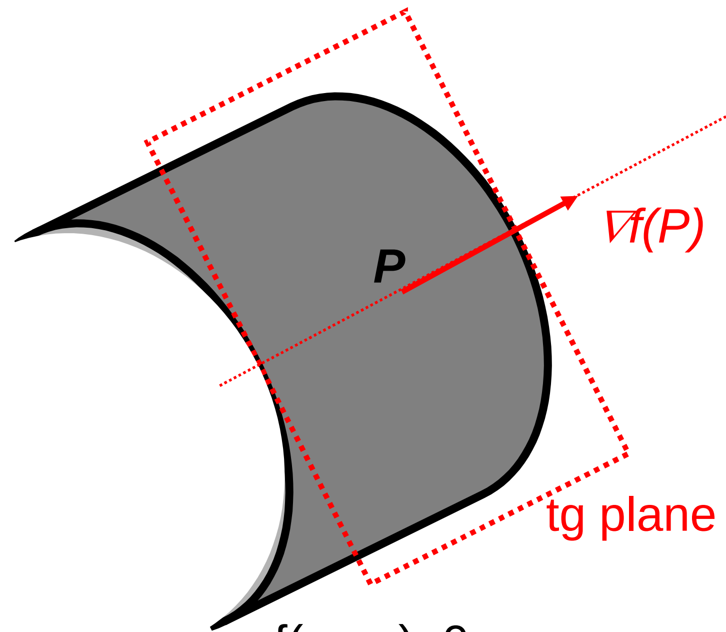
- this is an important result that we will use many times in the course
- e.g. for the sphere of center (0,0,0)

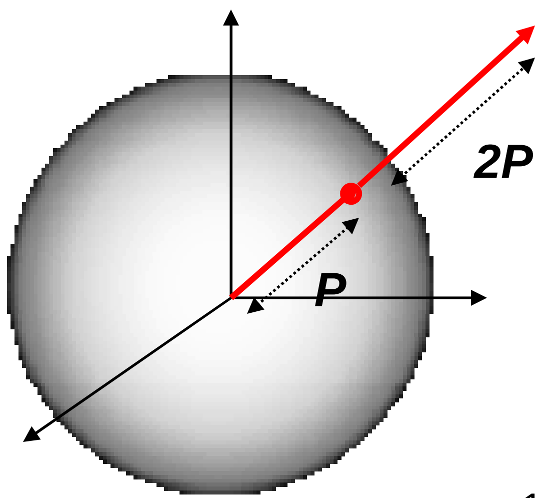
$$\nabla f = 2(x, y, z)^T$$

and “the normal at point *P* is simply *2P*”

- e.g.

n(1,0,0) = (2,0,0)

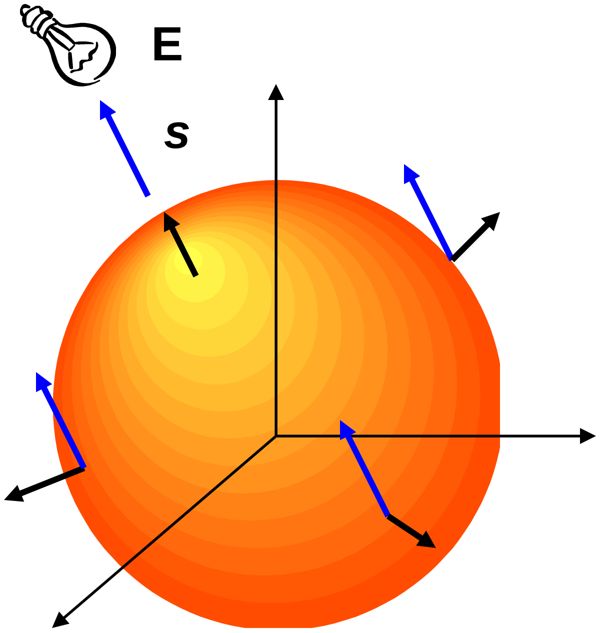




Normals

- note that this explains the shading of a sphere
  - we have seen a light source of direction ***s*** generates an image

$$I(P_0) = E\rho\langle \vec{n}(P_0), \vec{s}\rangle \propto \langle \vec{n}(P_0), \vec{s}\rangle$$



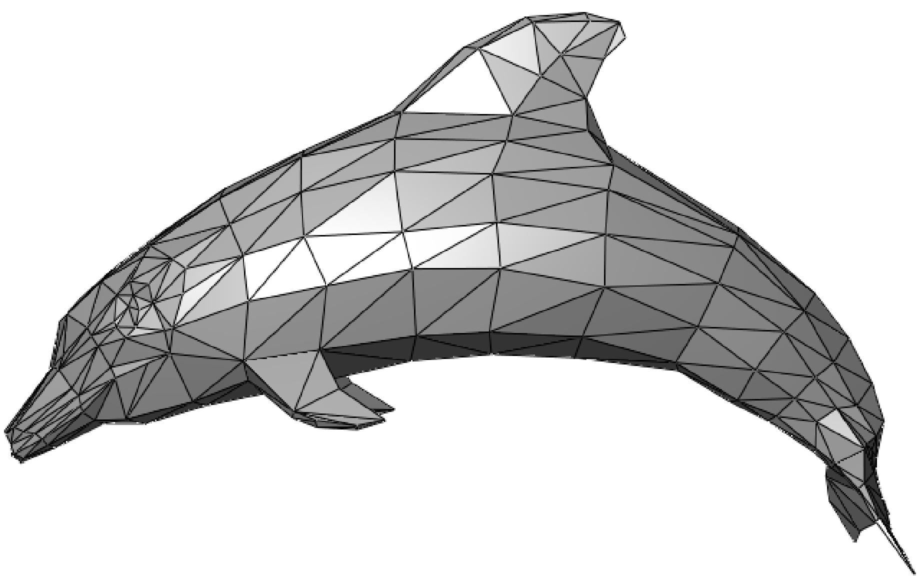
  - for the sphere this is just

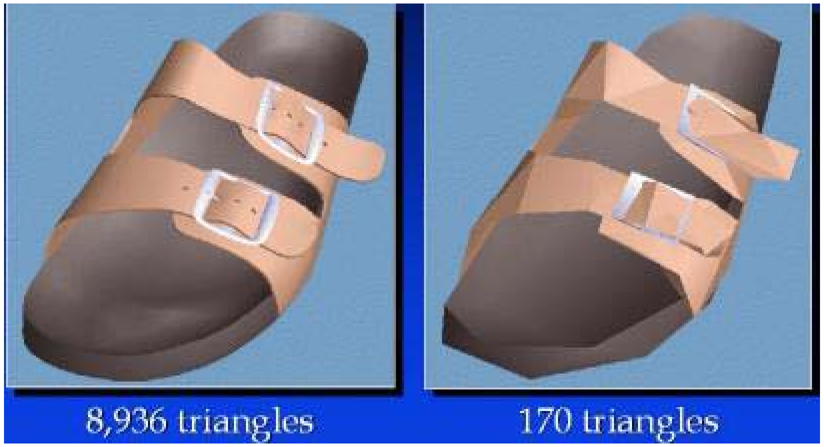
$$I(P_0) \propto \langle P_0, \vec{s}\rangle$$

"the dot product of ***s*** with ***P₀*** itself"

Normals

- what about more complex objects?
  - we approximate by a triangle mesh
  - sample a number of points on the surface
  - there are computer graphics algorithms for doing this
  - approximate surface by the triangles that connect those points
  - the more triangles you use, the better the approximation





Normals

- how do I compute the normal to a triangle?
- use the fact that a triangle is a patch of a plane
- how do I
  - compute the normal to a plane?
  - know what is the plane associated with my triangle?
- Q1: a plane is a function of the form

$$ax + by + cz + d = 0$$

  - this is the plane of parameters (a,b,c,d)
  - in this case

$$f(x, y, z) = ax + by + cz + d$$

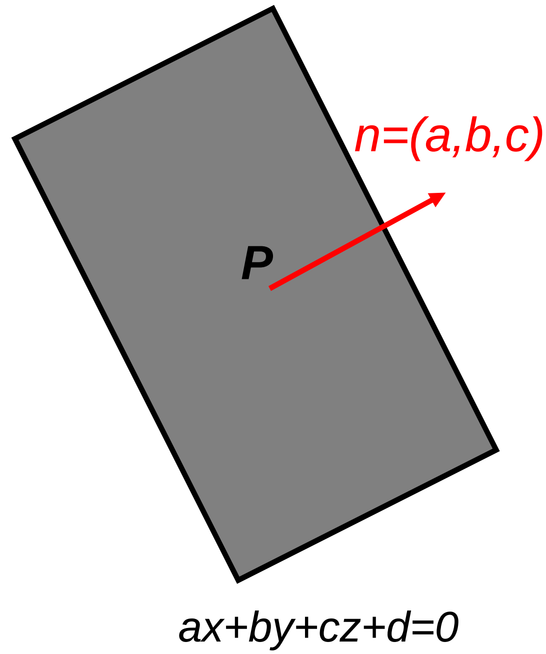
Normals

- and

$$\nabla f(x, y, z) = (a, b, c)^T$$

- note that this tells us various things:
  - all points in the plane have the same normal
  - this is called the "normal to the plane"

$$\vec{n} = (a, b, c)^T$$

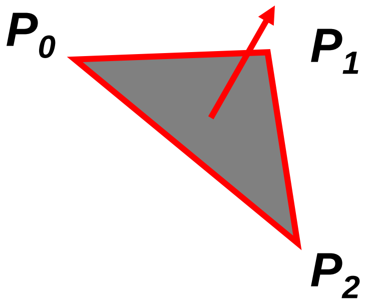


  - the parameter d only has to do with the distance to the origin
  - note that if (0,0,0) is on the plane, then d = 0, otherwise $d \neq 0$
  - the normal and *d* fully specify the plane

$$plane = \{P \mid \langle n, P \rangle = -d\}$$

Normals

- Q2: how do I know the plane associated with my triangle
  - note that triangle = 3 points, P_0, P_1, P_2
  - three points define a plane
  - we just have to solve the system of equations



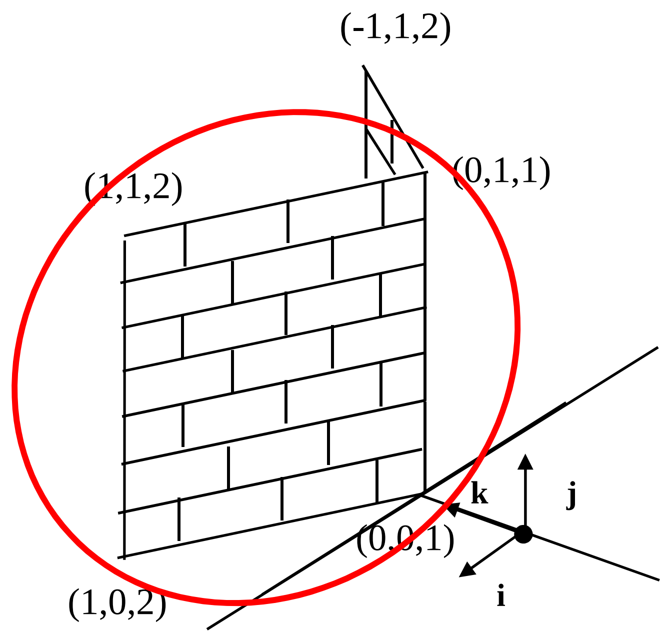
$$\begin{cases} \langle n, P_0 \rangle = -d \\ \langle n, P_1 \rangle = -d \\ \langle n, P_2 \rangle = -d \end{cases}$$

  - note that multiplying *n* and *d* by the same number does not change anything.
  - need to enforce extra constraint that $||n|| = 1$

example (from hw)

- three points

$$\begin{cases} \langle n, (0,1,1) \rangle = -d \\ \langle n, (0,0,1) \rangle = -d \\ \langle n, (1,0,2) \rangle = -d \end{cases}$$



$$\begin{cases} b = -d - c \\ c = -d \\ a = -d - 2c \end{cases} \Leftrightarrow \begin{cases} b = 0 \\ c = -d \\ a = d \end{cases}$$

- leads to the plane equation

$$dx - dz = -d \Leftrightarrow \boxed{z = x + 1}$$

Planes

- for a plane our radiometry equation

$$P(P_i) = E\rho(P_0)\vec{n}(P_0).\vec{s}$$

- simplifies to

$$P(P_i) = E\langle \vec{n}, \vec{s} \rangle \rho(P_0) \propto \rho(P_0)$$

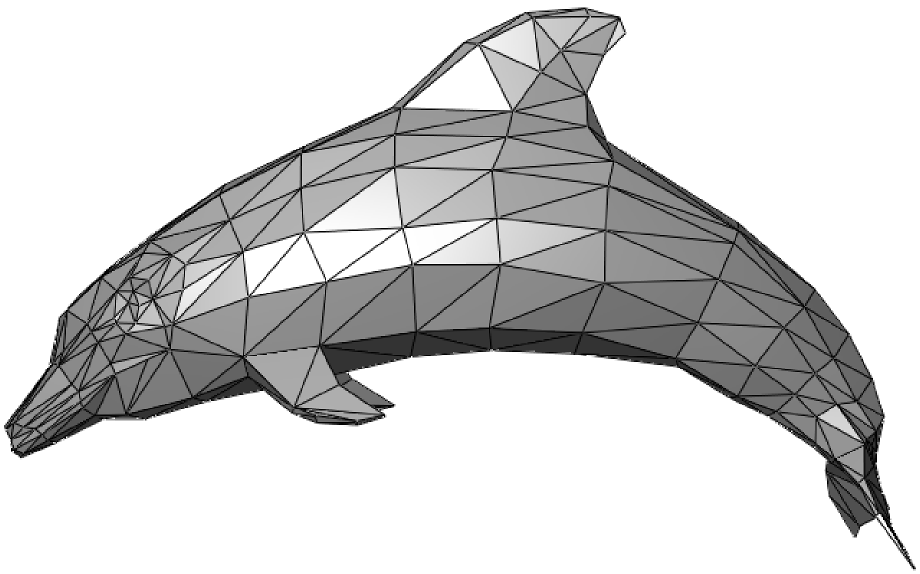
- this means that (constant) shading is uniform
- the variations that we see are variations in albedo

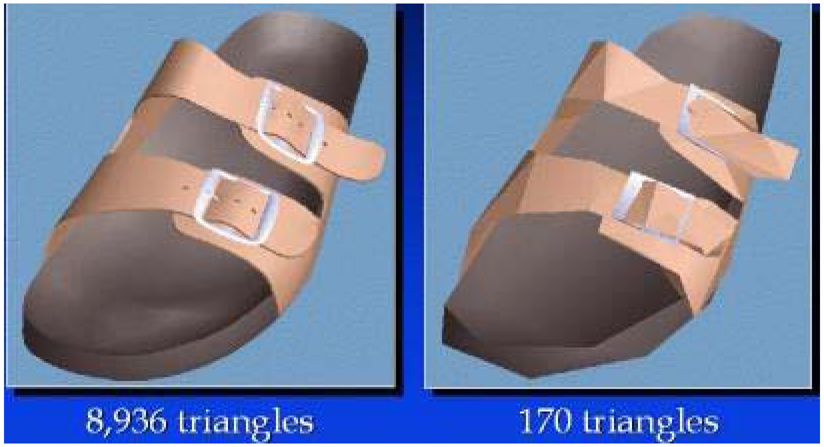
Meshes

- for a mesh
  - this holds for each triangle

$$P(P_i) = E\langle \vec{n}, \vec{s} \rangle \rho(P_0)$$

  - but <*n,s*> changes from triangle to triangle (different normals)
  - hence, we can still have complex shading effects as the number of triangles increases

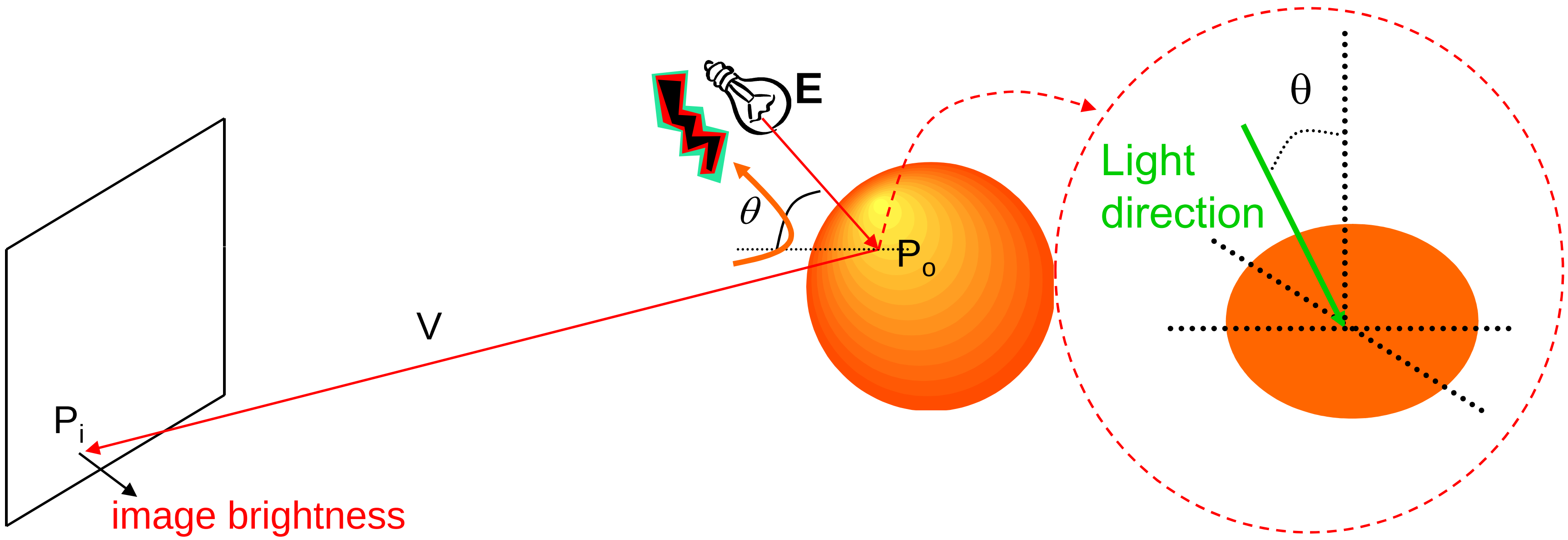




More complexity

- in summary, we can do a lot with our simplistic model
  - Lambertian surfaces
  - point source @ infinity

$$P(P_i) = E\rho(P_0)\cos\theta$$



  - turns out that we can work with much more complex light sources

Multiple light sources

- the key insight that the equation is linear on **s**

$$P(P_i) = E\rho(P_0)\langle \vec{n}(P_0), \vec{s}\,\rangle$$

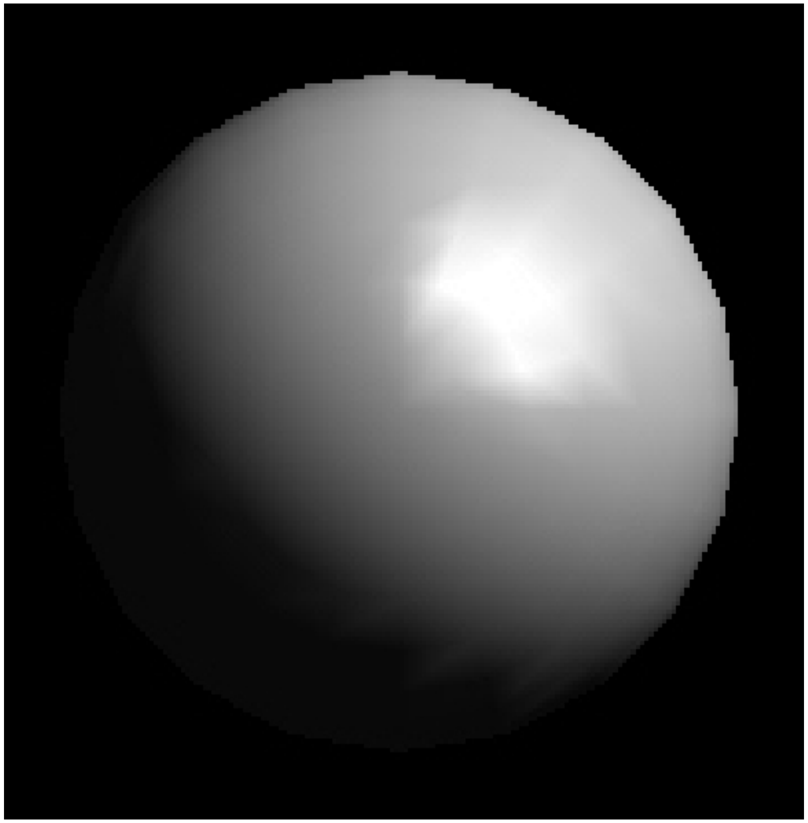
- if we have *n* PS @ infinity, we can just assume that **s** = $\mathbf{s}_1 + \ldots + \mathbf{s}_n$

$$P = E\rho(P_0)\left\langle \vec{n}(P_0), \sum_k \vec{s}_k \right\rangle$$
$$= \sum_k E\rho(P_0)\langle \vec{n}(P_0), \vec{s}_k \rangle = \sum_i P_k$$

- resulting image is sum of the images due to each source

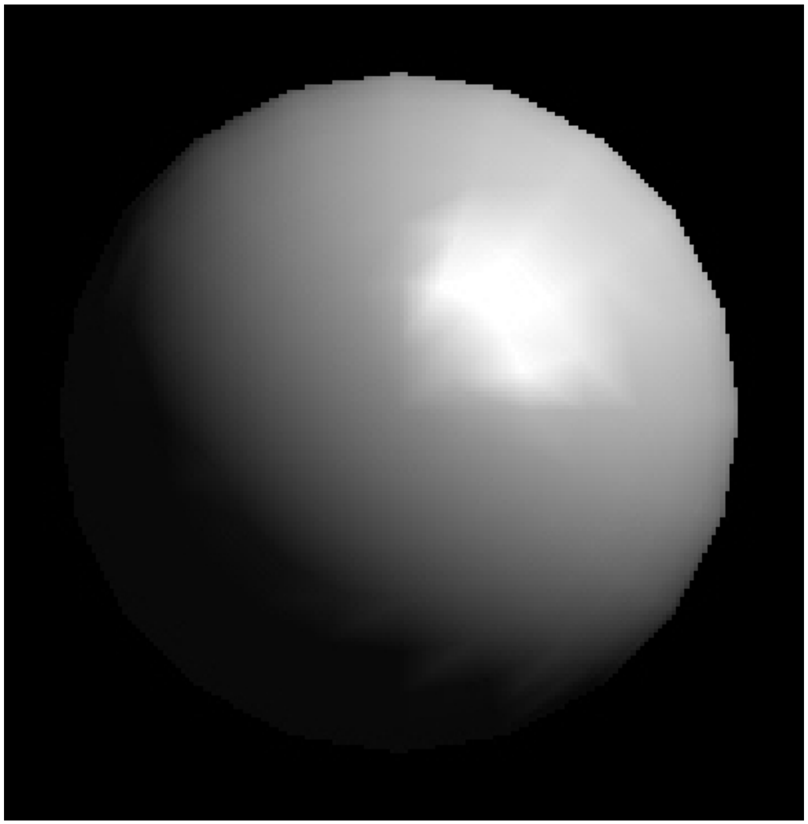
Demo

- we start from this image

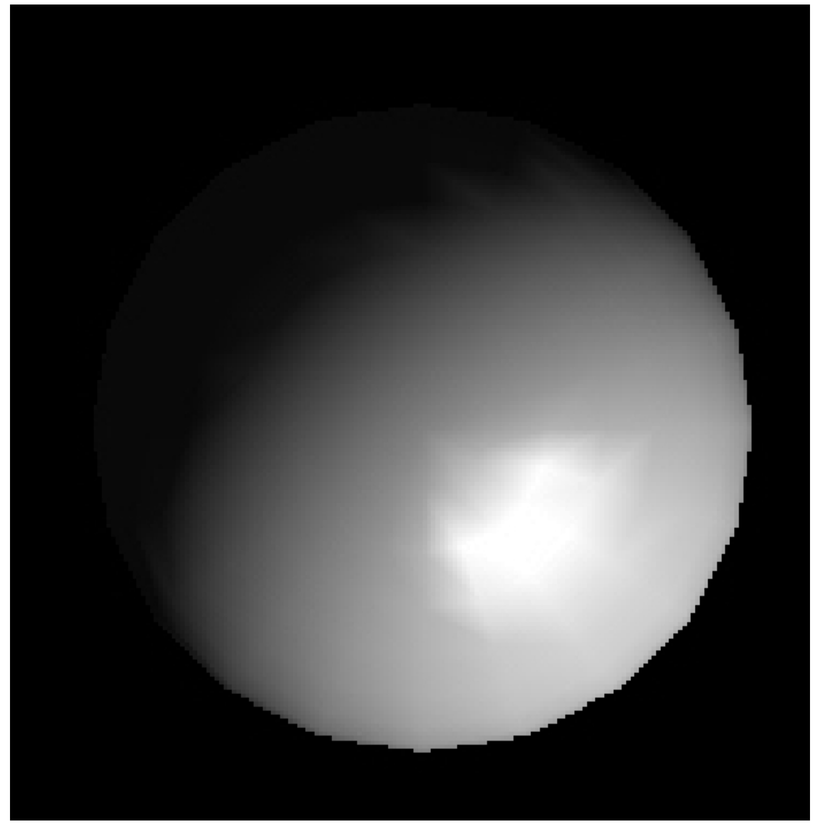


Demo

- adding this

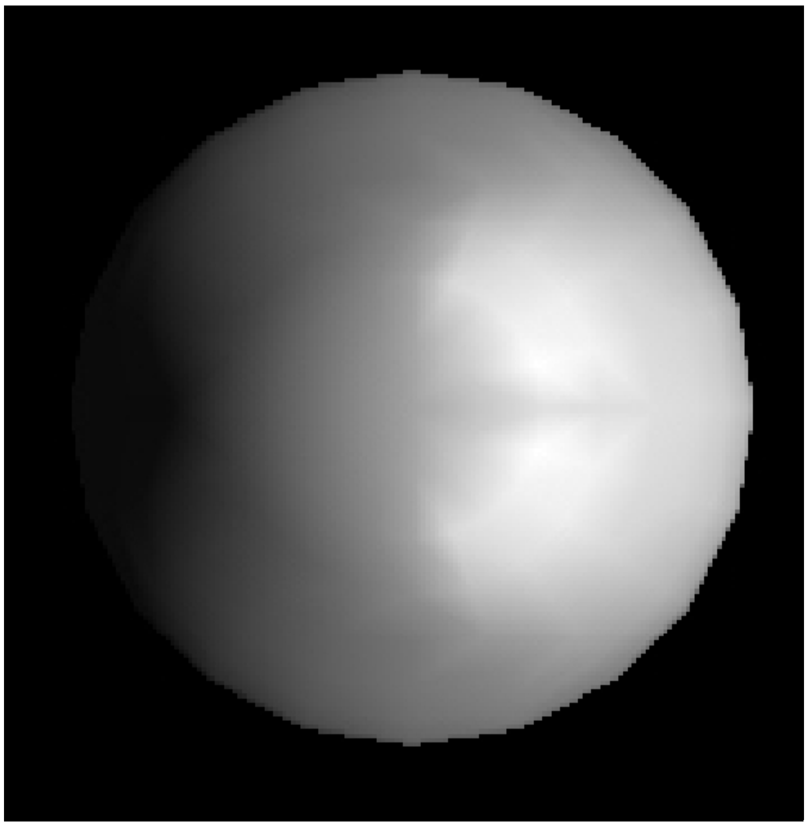


+



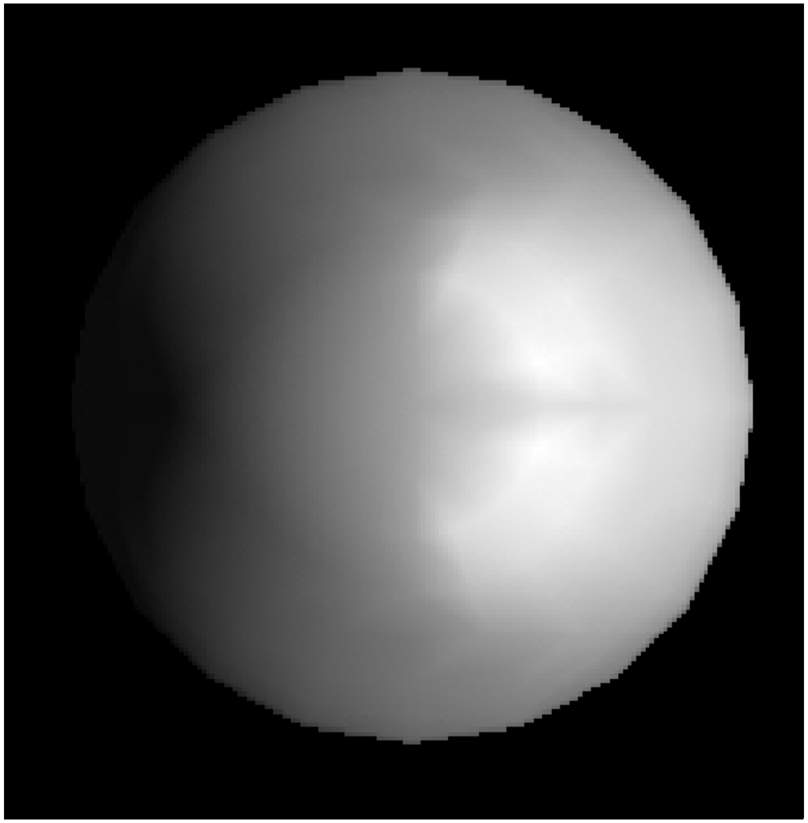
Demo

- we get the image lit by two light sources

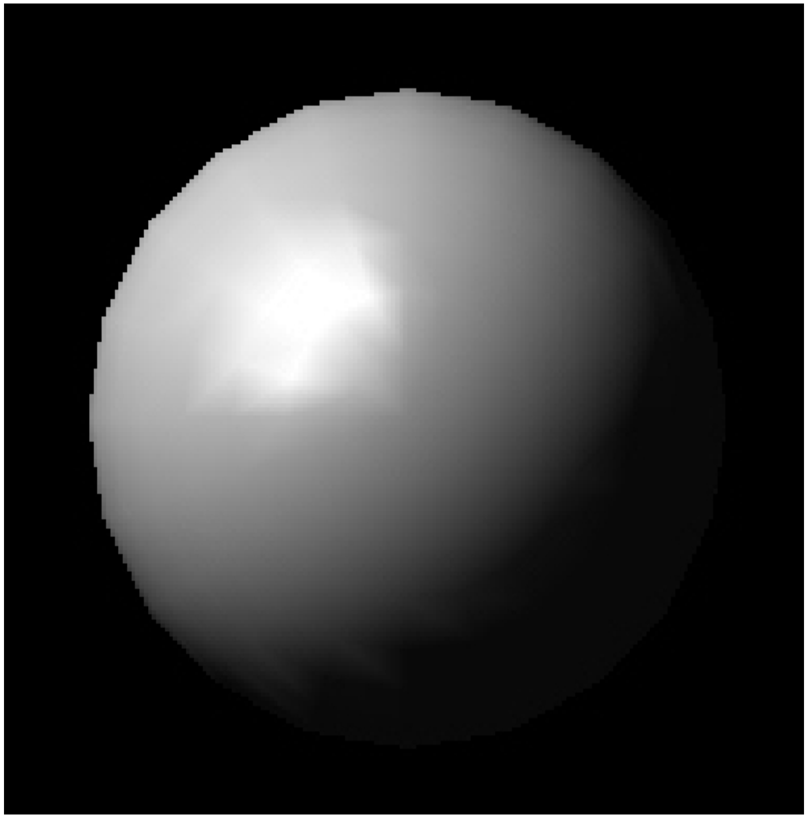


Demo

- adding this

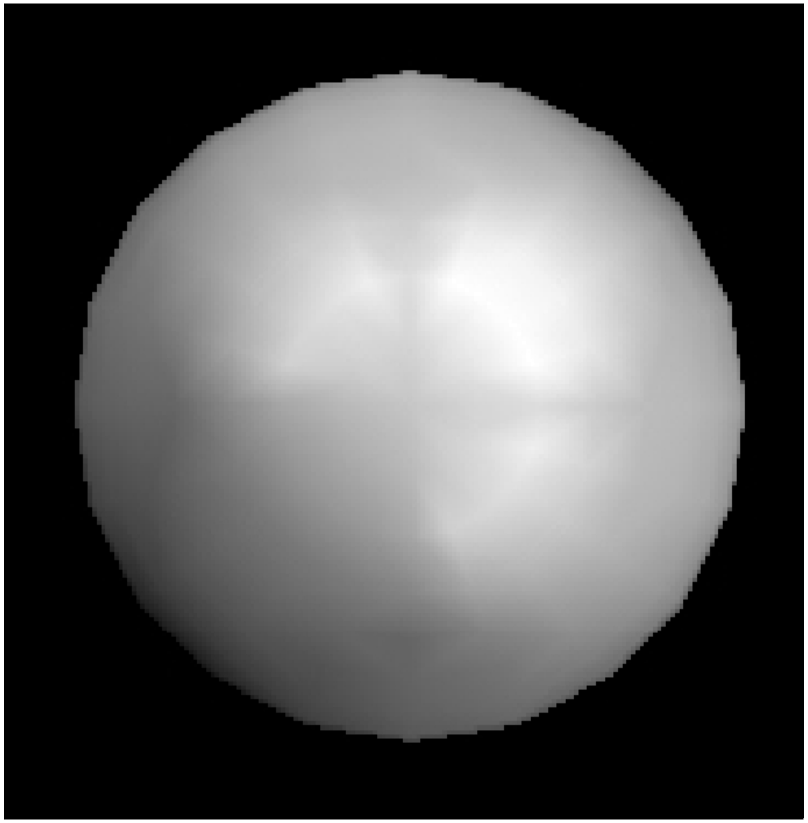


+



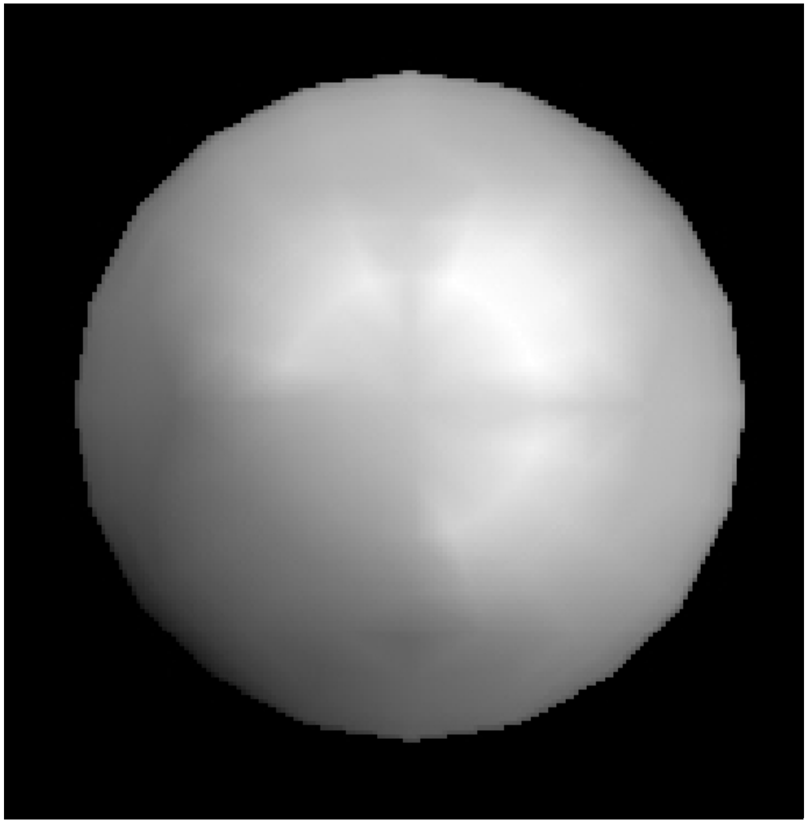
Demo

- we get the image lit by the three light sources

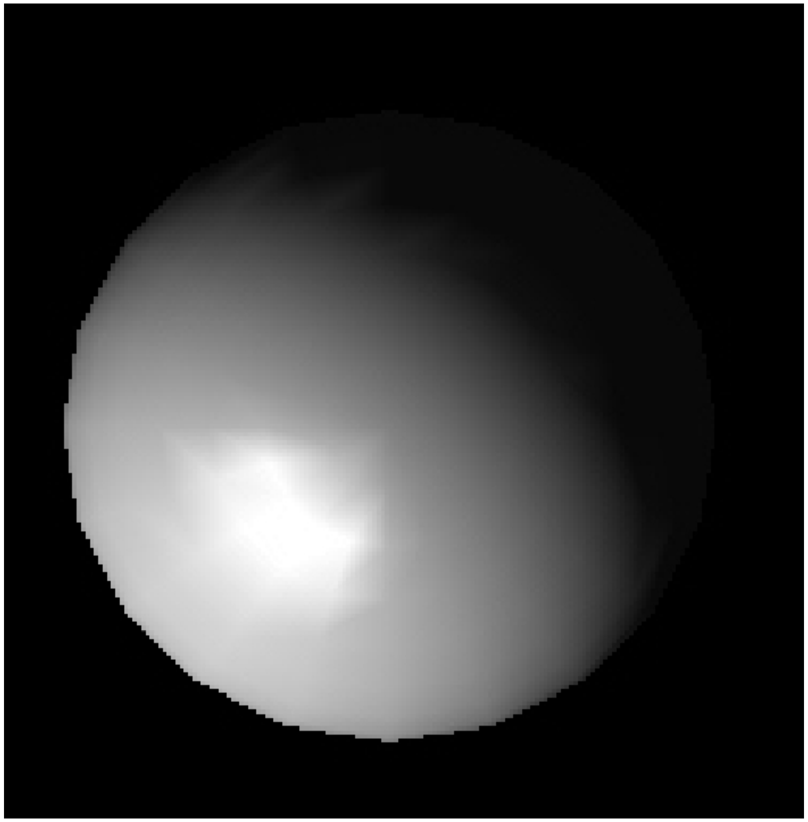


Demo

- adding this

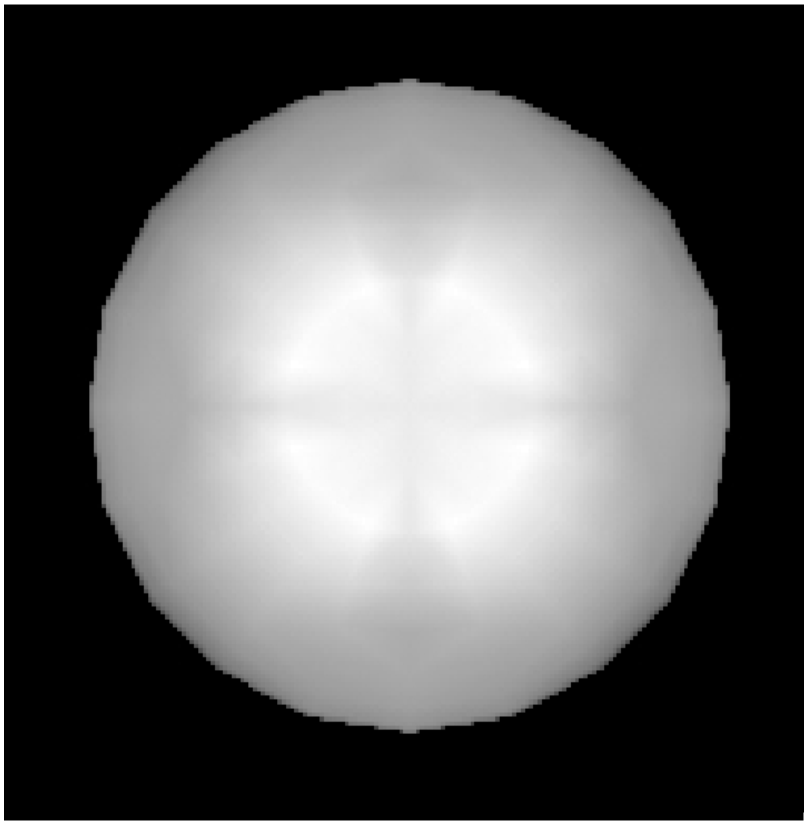


+



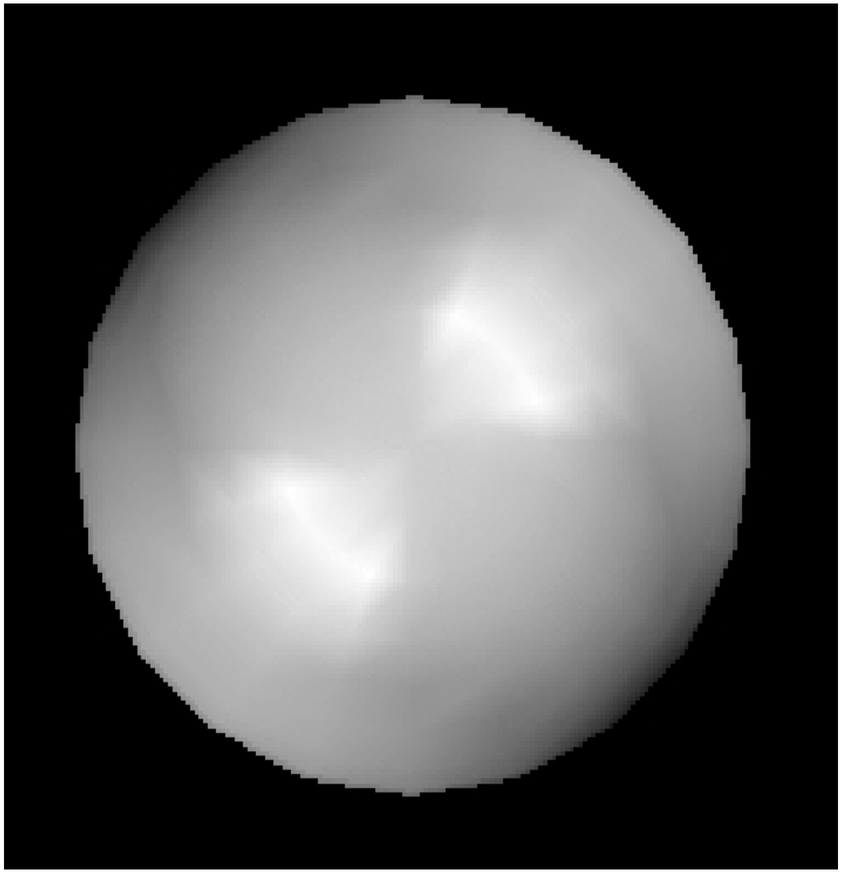
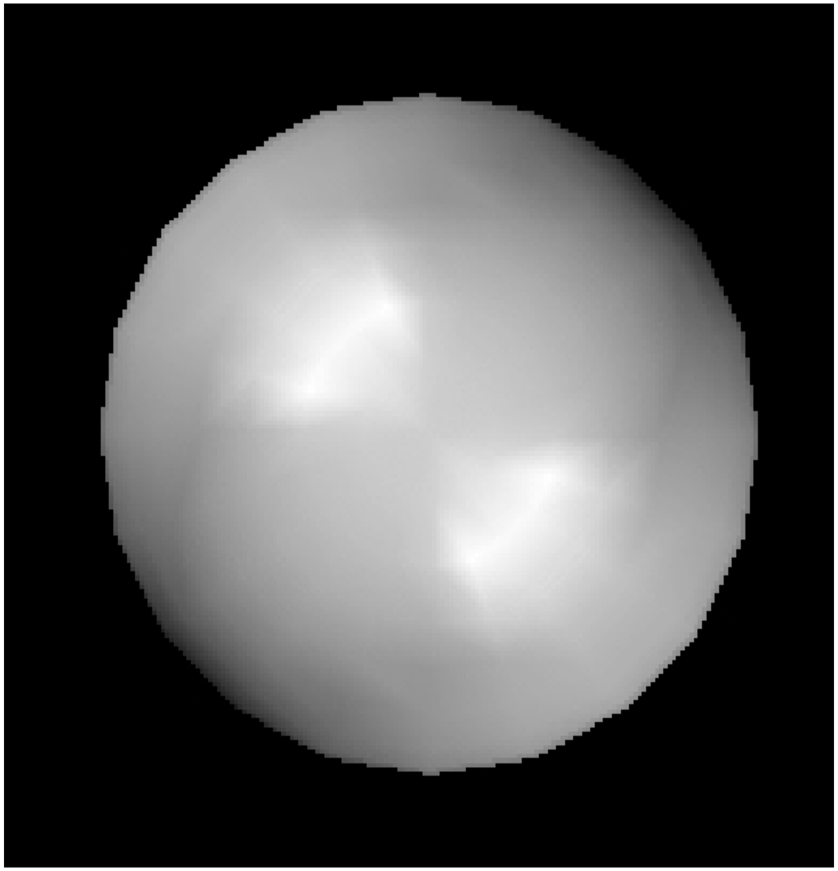
Demo

- we get the image lit by the four light sources



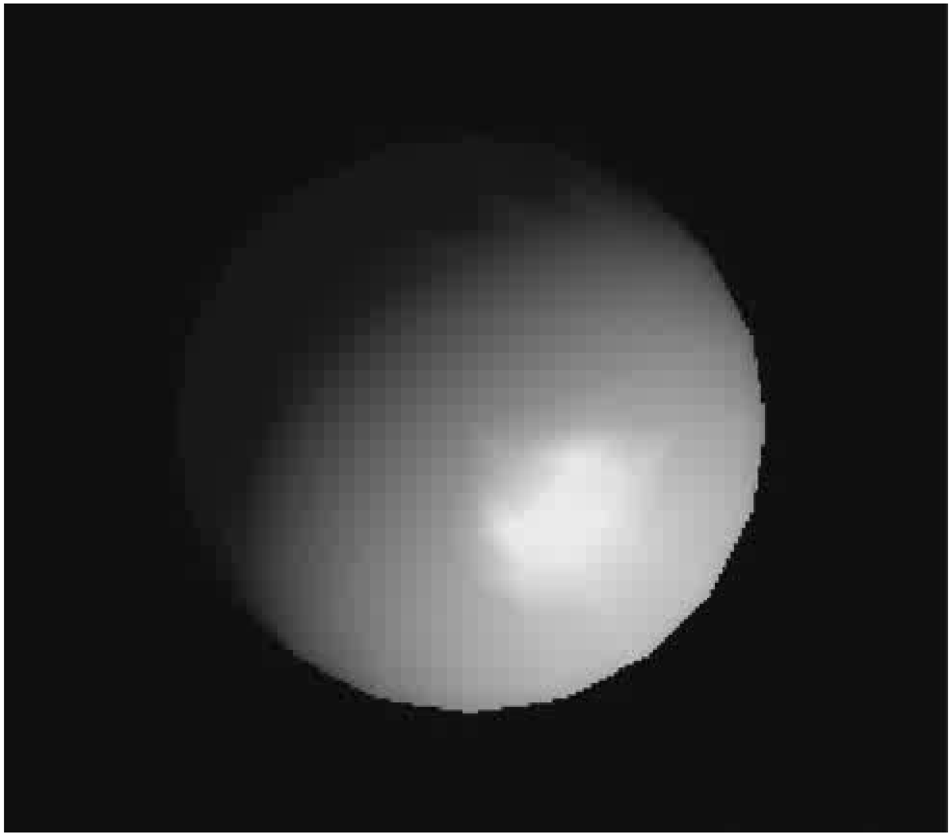
Demos

- other possible patterns

Demos

- From these I can create a movie

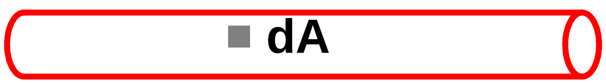


Demo

- and this happens in the real world too (note combo of geo-metry and radio-metry)

More complexity

- and what about fluorescent lights, etc.?
- is this still a point source?
- no, but an infinitesimal patch is



- the patch dA is a point source
- we work with the power density, instead of power

- e.g. dA, centered at x, emits density *E(x)dA* in its normal direction *s(x)*

More complexity

- the contribution of patch centered at *x* to the power that hits object point P_0 is

$$P(x) = E(x)\rho(P_0)\langle \vec{n}(P_0), s(x)\rangle dA$$

to compute the overall power, due to all patches, we simply integrate

$$P = \int_A E(x)\rho(P_0)\langle \vec{n}(P_0), s(x)\rangle dA$$

- and by linearity of the dot product

$$P = \rho(P_0)\left\langle \vec{n}(P_0), \int_A E(x)s(x)dA \right\rangle$$

More complexity

- we thus have

$$P = \rho(P_0)\left\langle \vec{n}(P_0), \int_A E(x)s(x)dA \right\rangle$$

- note that the integral does not depend on P_0
- this is the same as computing

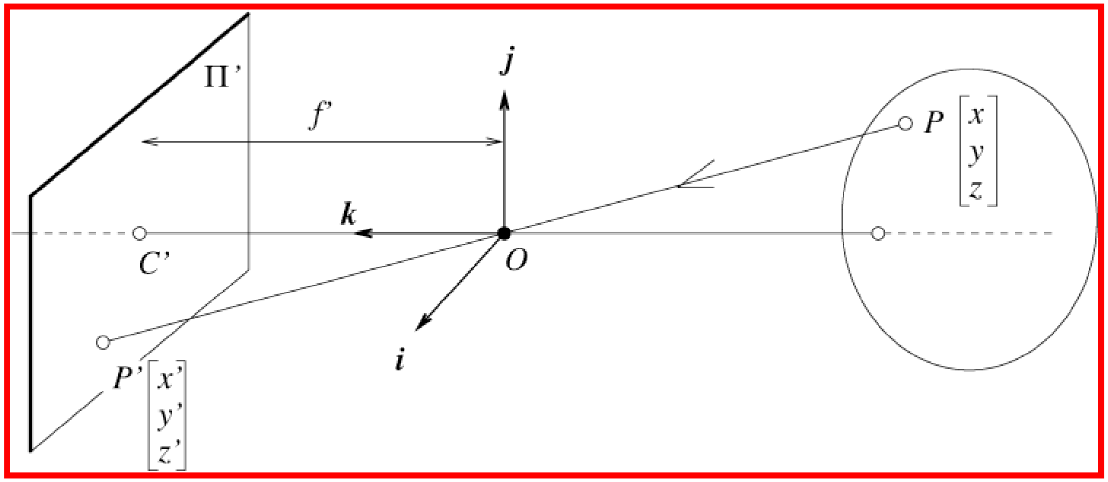
$$\vec{E} = \int_A E(x,y,z)\vec{s}(x,y,x)dxdydz$$

(which is a 3D vector) and assuming a point source of magnitude E' = ||E|| and unit direction s' = E/||E||

In summary

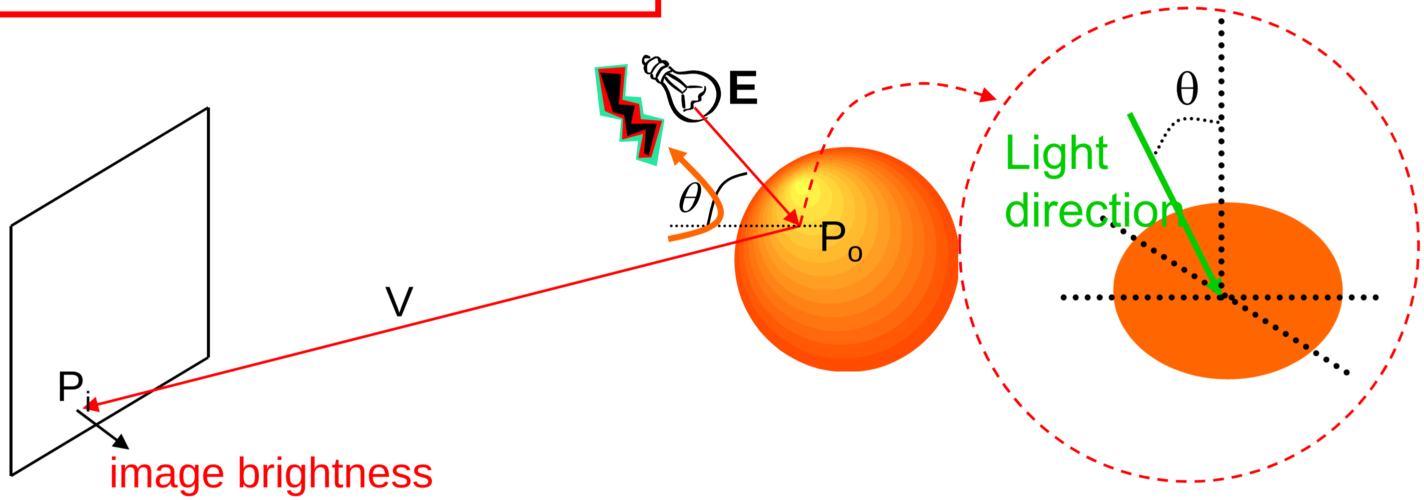
- geometry

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = f \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$



- radiometry

$$P(P_i) = E\rho(P_0)\cos\theta$$



Any questions?