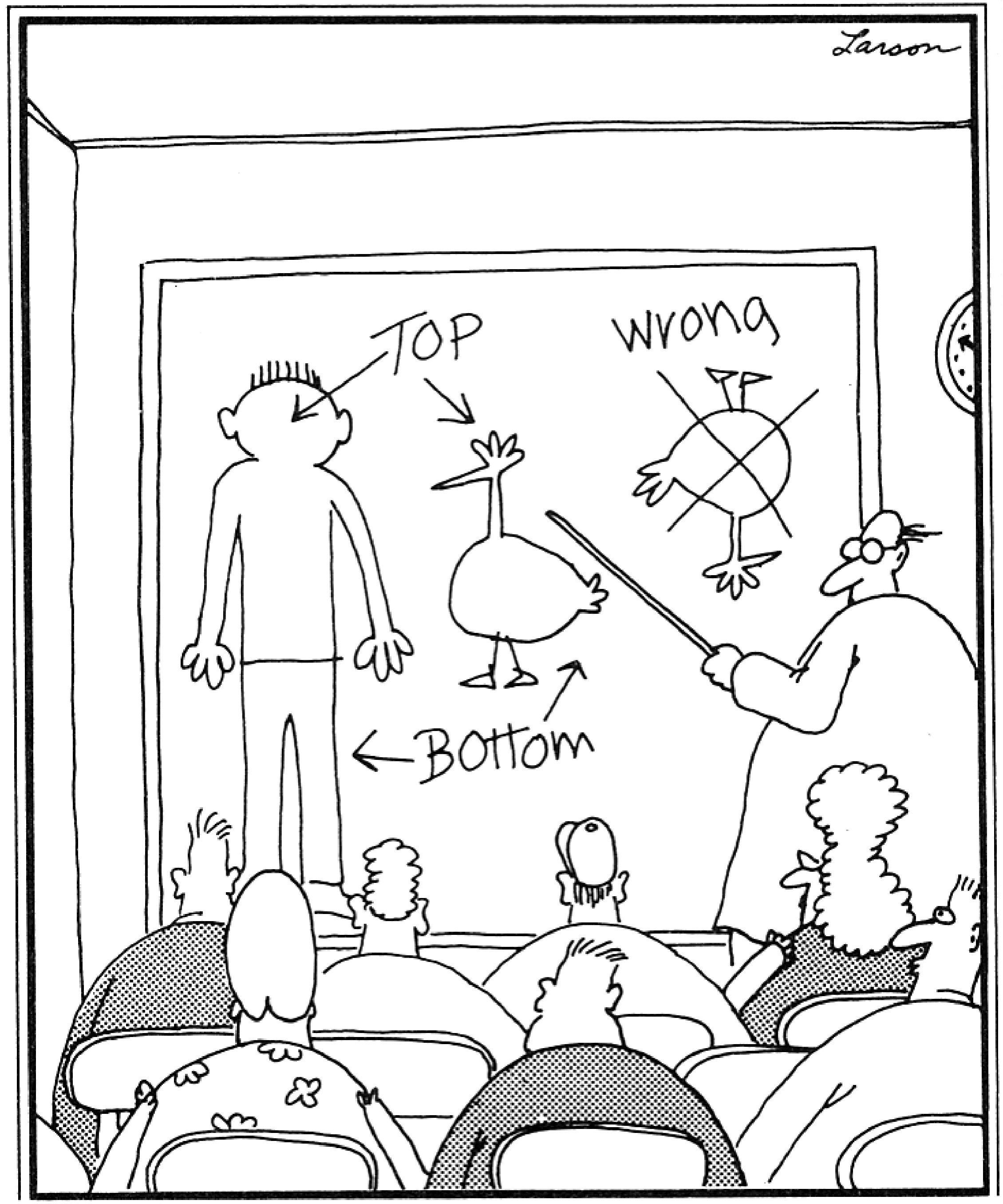
Transformations
&
Local
Illumination

MIT



People who don't know which end is up.

Last Time?

- Transformations
  - Rigid body, affine, similitude, linear, projective
- Linearity
  - $f(x+y)=f(x)+f(y); \quad f(ax) = a\, f(x)$
- Homogeneous coordinates
  - $(x, y, z, w) \sim (x/w, y/w, z/w)$
  - Translation in a matrix
  - Projective transforms
- Non-commutativity
- Transformations in modeling

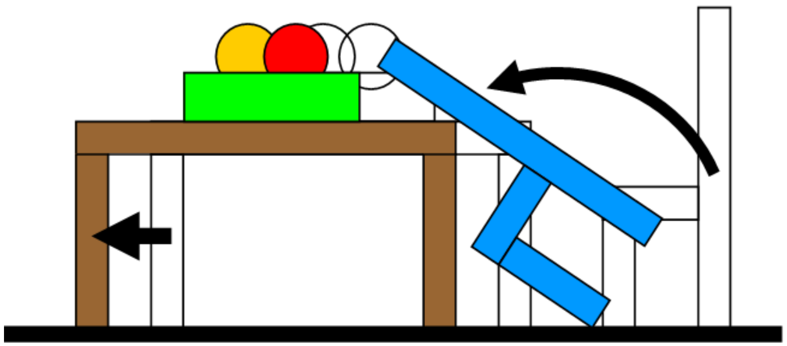
Today

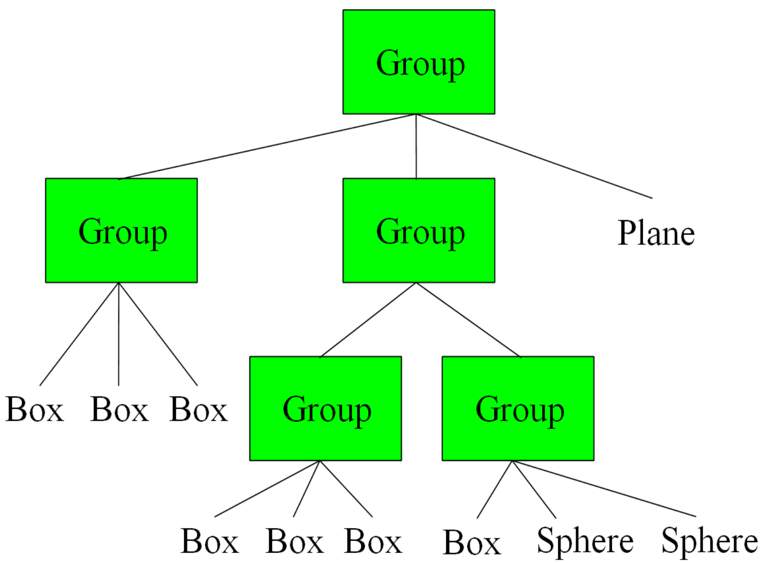
- Intro to Transformations
- Classes of Transformations
- Representing Transformations
- Combining Transformations
- Transformations in Modeling
- Adding Transformations to our Ray Tracer

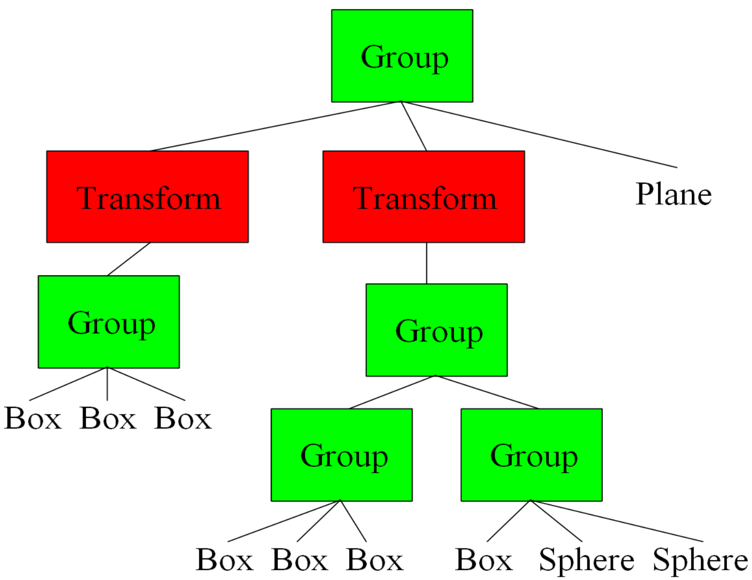
- Local illumination and shading

Why is a Transform an Object3D?

- To position the logical groupings of objects within the scene

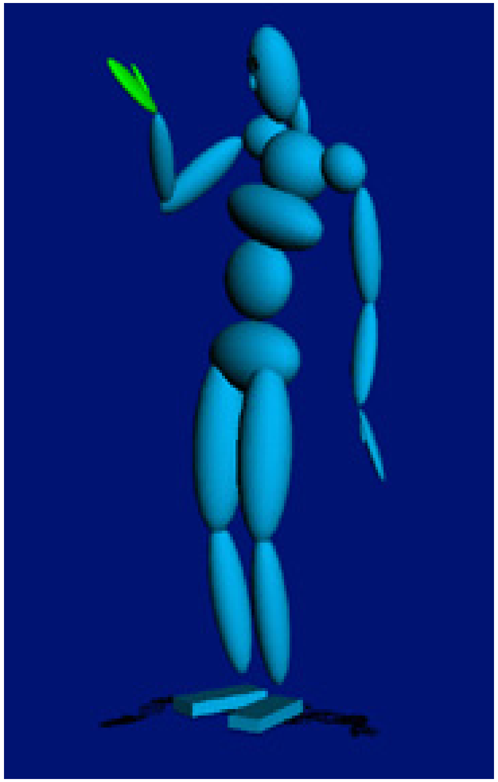






Recursive call and composition

- Recursive call tree: leaves are evaluated first
- Apply matrix from right to left
- Natural composition of transformations from object space to world space
  - First put finger in hand frame
  - Then apply elbow transform
  - Then shoulder transform
  - etc.



Questions?

Today

- Intro to Transformations
- Classes of Transformations
- Representing Transformations
- Combining Transformations
- Transformations in Modeling
- Adding Transformations to our Ray Tracer

Incorporating Transforms

1. Make each primitive handle any applied transformations

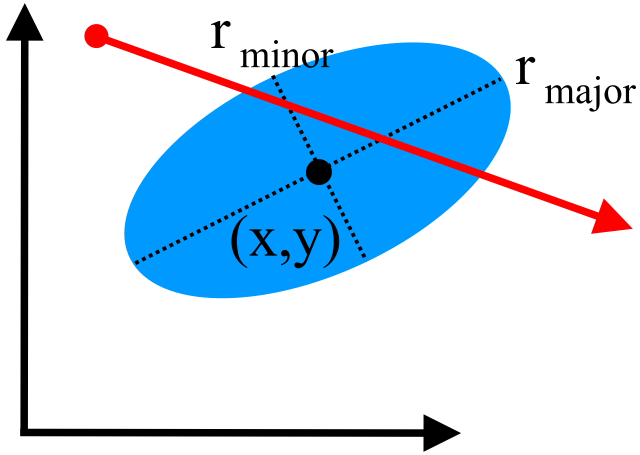
```
Sphere {
    center 1 0.5 0
    radius 2
}
```

2. Transform the Rays

```
Transform {
    Translate { 1 0.5 0 }
    Scale { 2 2 2 }
    Sphere {
        center 0 0 0
        radius 1
    }
}
```

Primitives handle Transforms

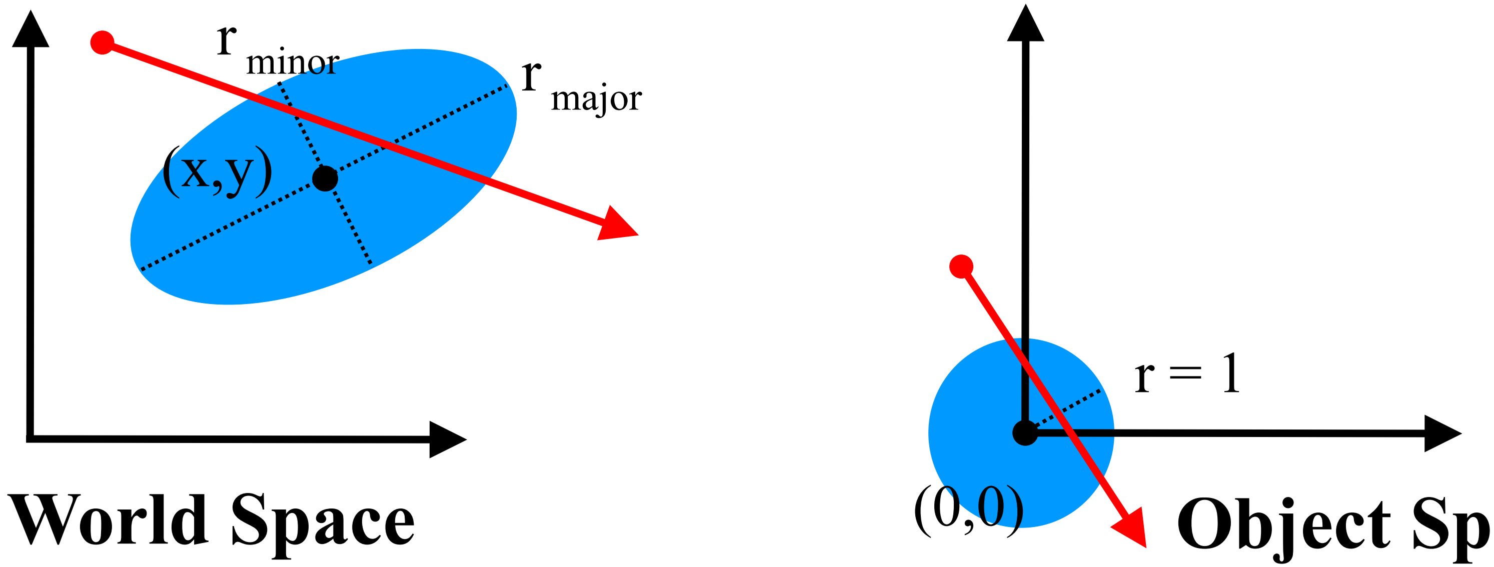
```
Sphere {
    center 3 2 0
    z_rotation 30
    r_major 2
    r_minor 1
}
```



- Complicated for many primitives

Transform the Ray

- Move the ray from *World Space* to *Object Space*



$$p_{WS} = \mathbf{M} \; p_{OS}$$

$$p_{OS} = \mathbf{M}^{-1} \; p_{WS}$$

Transform Ray

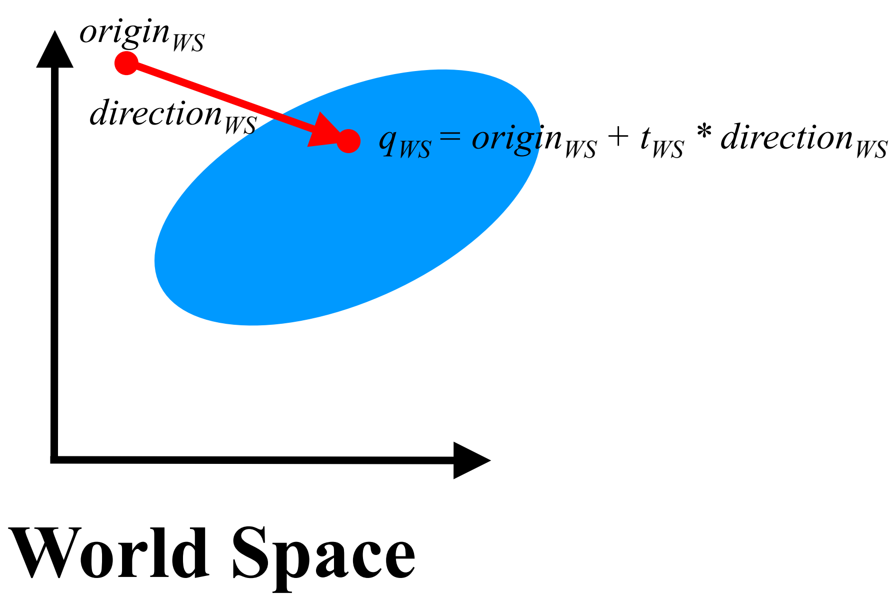
- New origin:

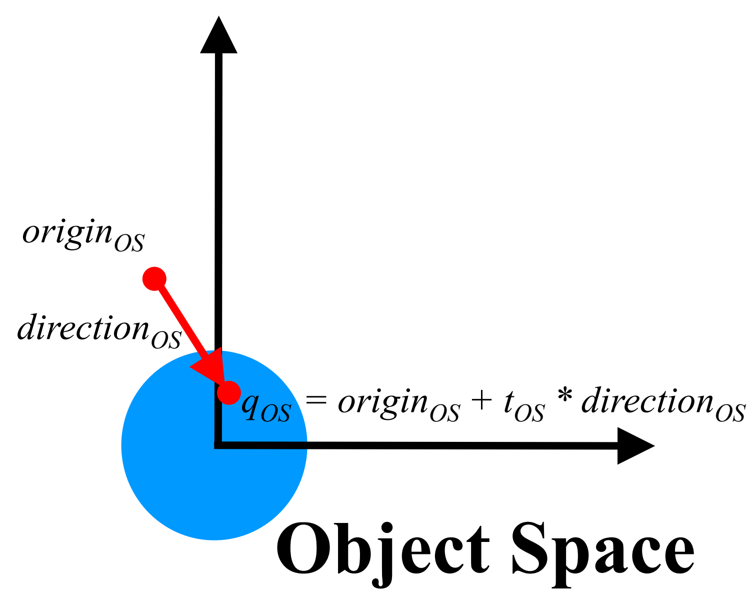
$$origin_{OS} = \mathbf{M^{-1}}\ origin_{WS}$$

- New direction:

$$direction_{OS} = \mathbf{M^{-1}}\ (origin_{WS} + 1 * direction_{WS}) \ - \ \mathbf{M^{-1}}\ origin_{WS}$$

$$direction_{OS} = \mathbf{M^{-1}}\ \ direction_{WS}$$





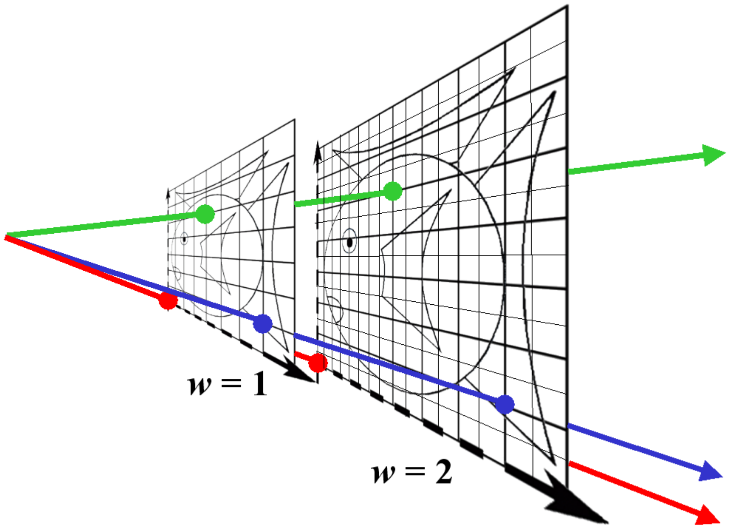
Transforming Points & Directions

- Transform point

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} ax+by+cz+d \\ ex+fy+gz+h \\ ix+jy+kz+l \\ 1 \end{pmatrix}$$

- Transform direction

$$\begin{pmatrix} x' \\ y' \\ z' \\ 0 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix} = \begin{pmatrix} ax+by+cz \\ ex+fy+gz \\ ix+jy+kz \\ 0 \end{pmatrix}$$



Homogeneous Coordinates: *(x,y,z,w)*

w = 0 is a point at infinity (direction)

- With the usual storage strategy (no *w*) you need different routines to apply M to a point and to a direction

What to do about the depth, t

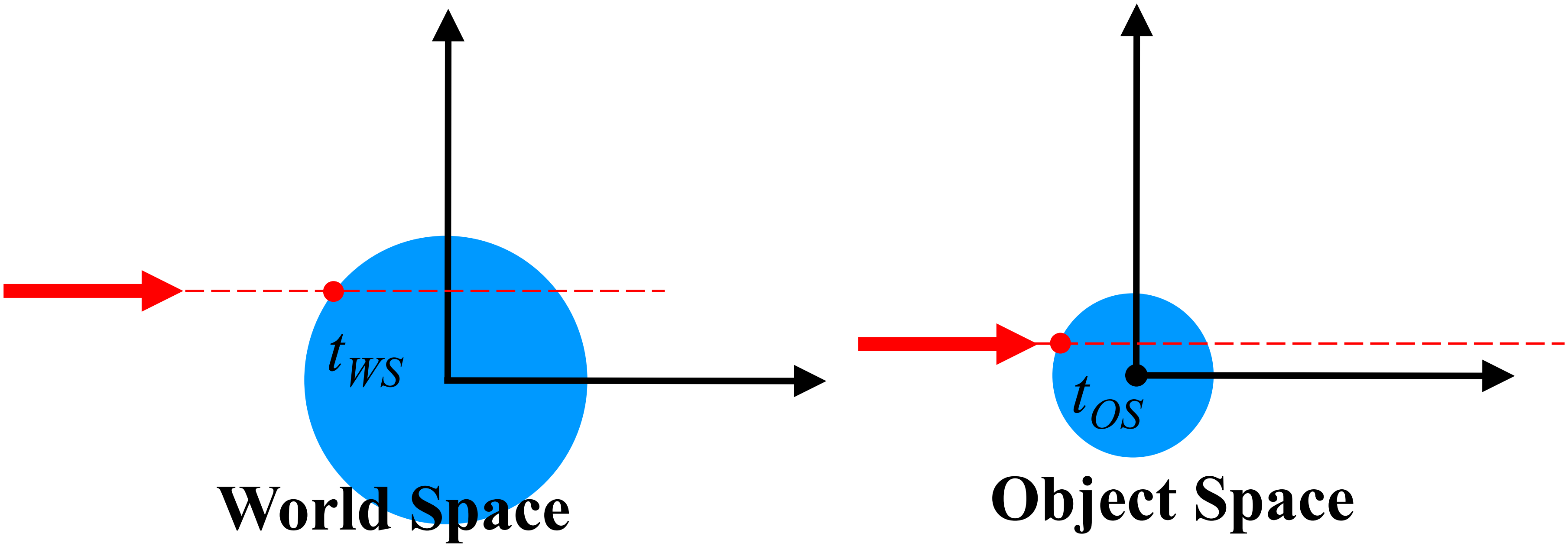
If **M** includes scaling, $direction_{OS}$ will NOT be normalized

1. Normalize the direction

2. Don't normalize the direction

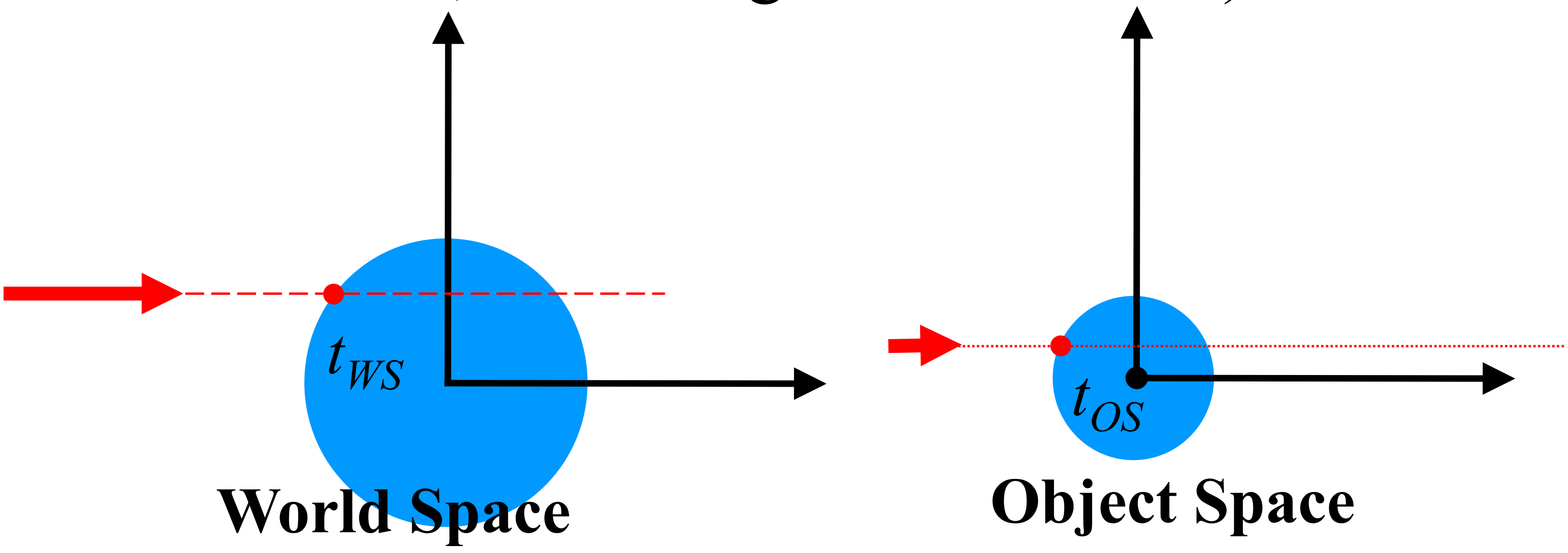
1. Normalize direction

- $t_{OS} \neq t_{WS}$
  and must be rescaled after intersection



2. Don't normalize direction

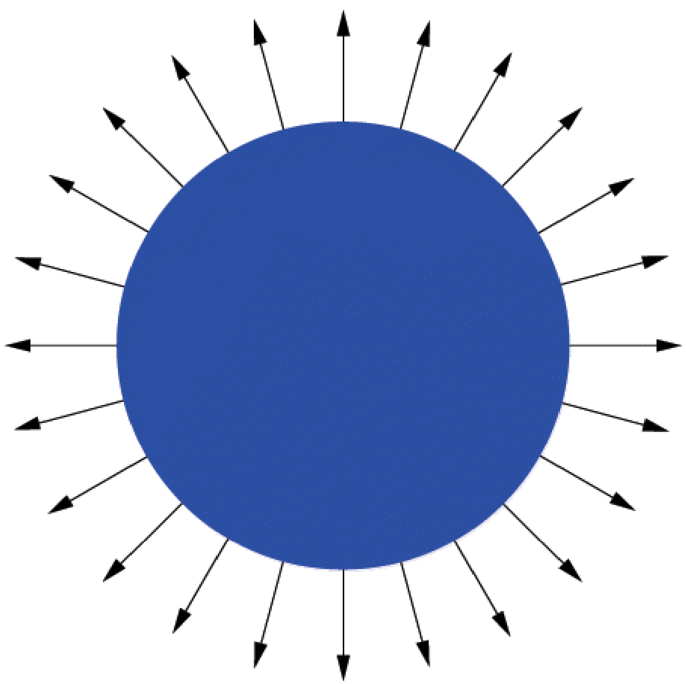
- $t_{OS} = t_{WS}$
- Don't rely on t_{OS} being true distance during intersection routines (e.g. geometric ray-sphere intersection, a≠1 in algebraic solution)

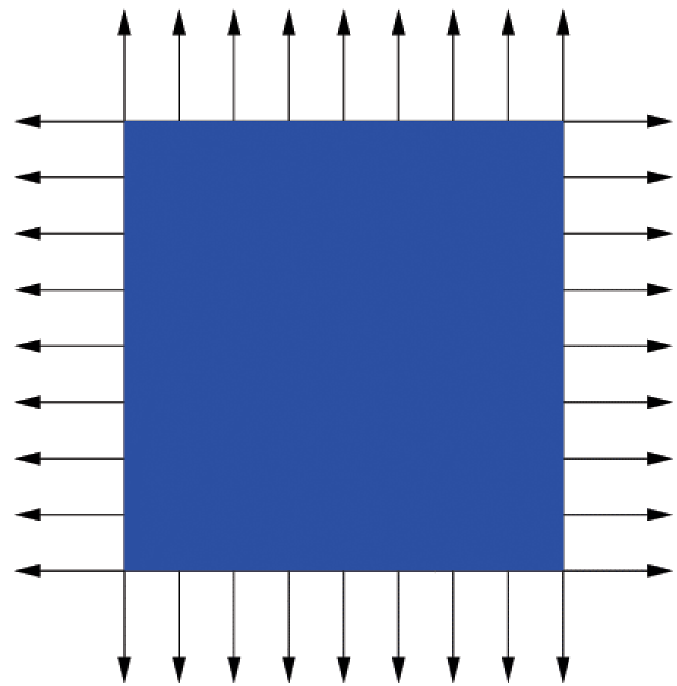


Questions?

New component of the Hit class

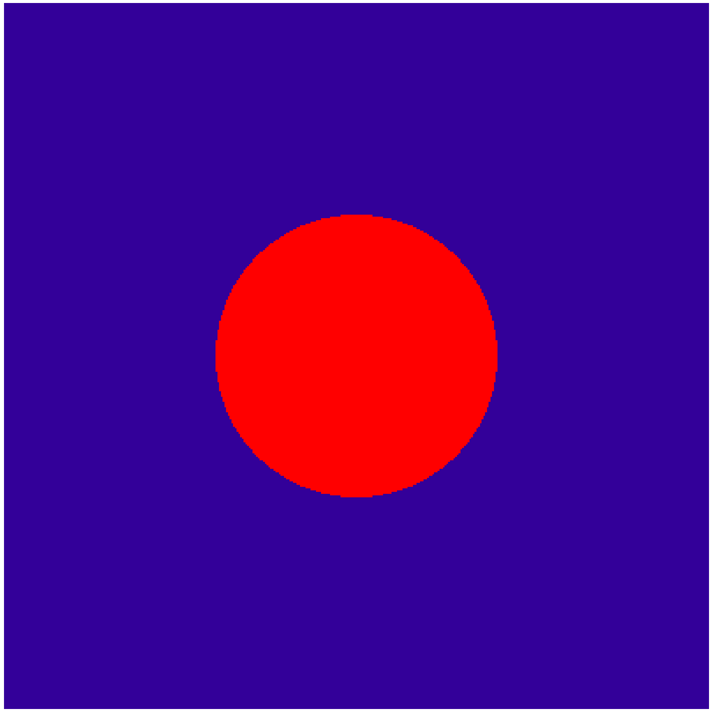
- Surface Normal: unit vector that is locally perpendicular to the surface



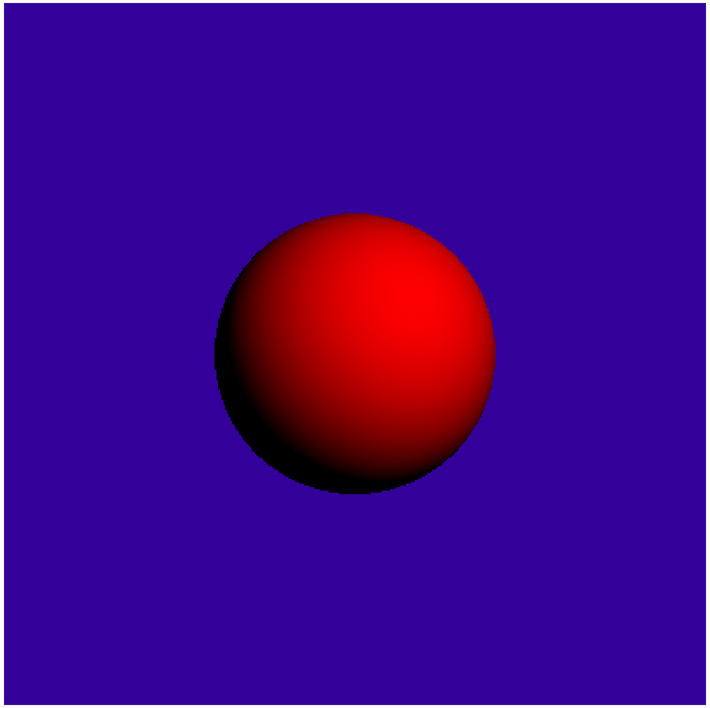


Why is the Normal important?

- It's used for shading — makes things look 3D!

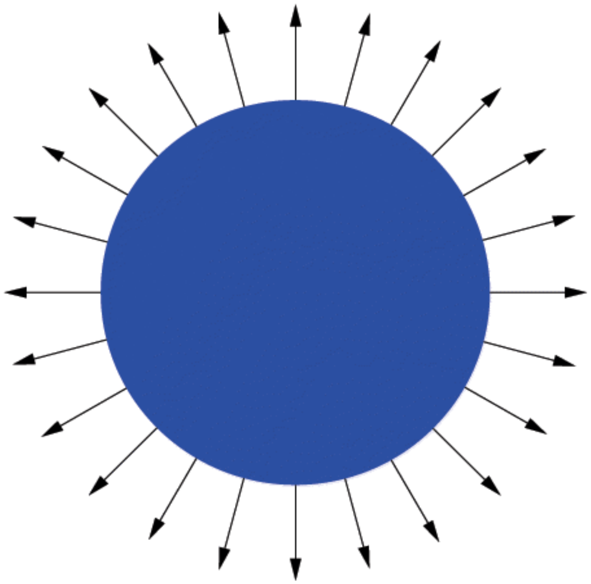


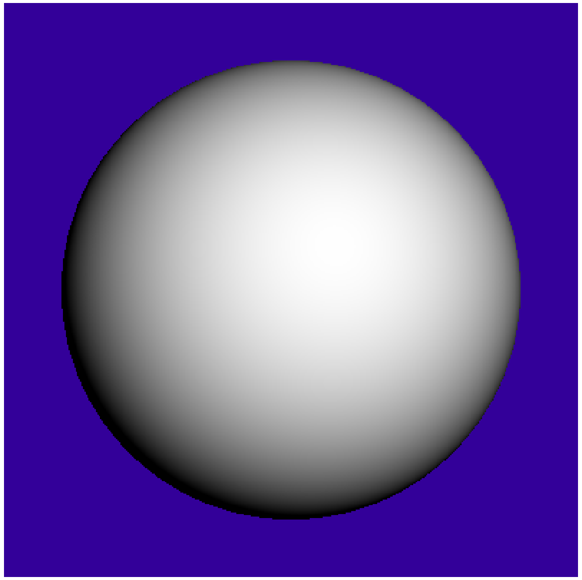
object color only
(Assignment 1)

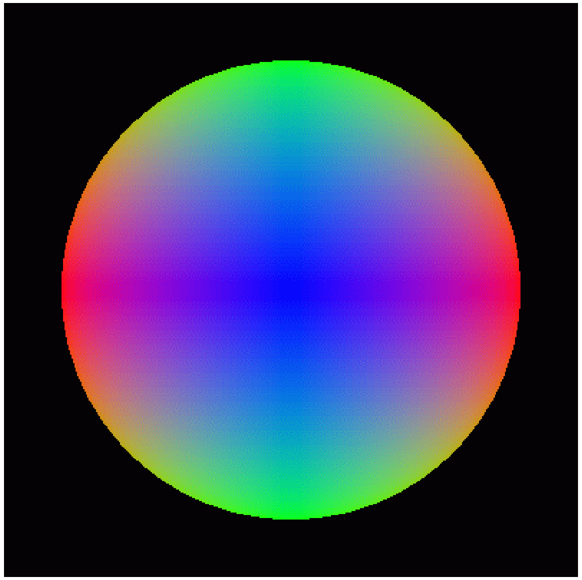


Diffuse Shading
(Assignment 2)

Visualization of Surface Normal





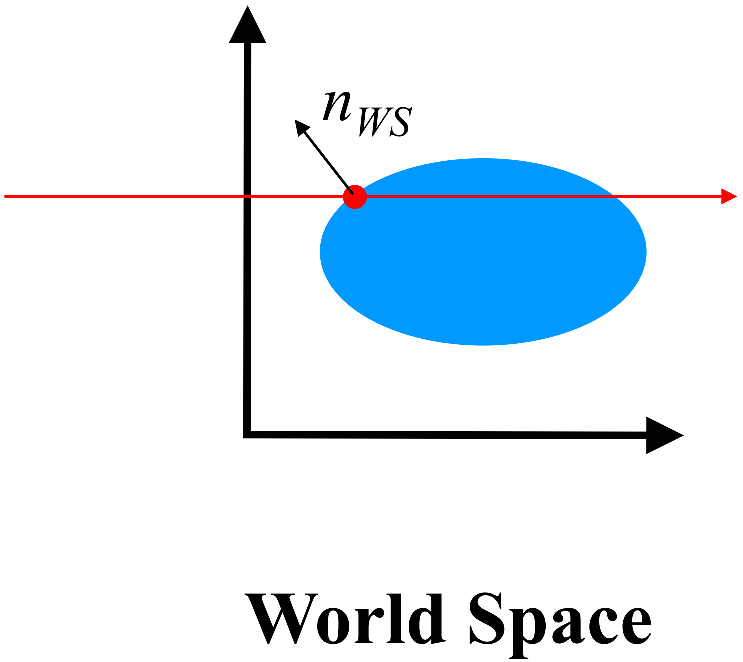


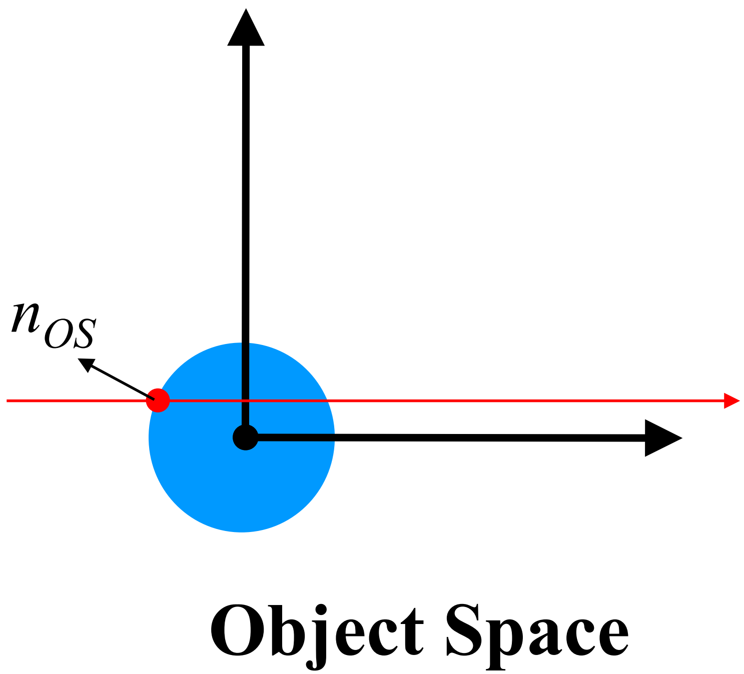
$\pm x = \text{Red}$

$\pm y = \text{Green}$

$\pm z = \text{Blue}$

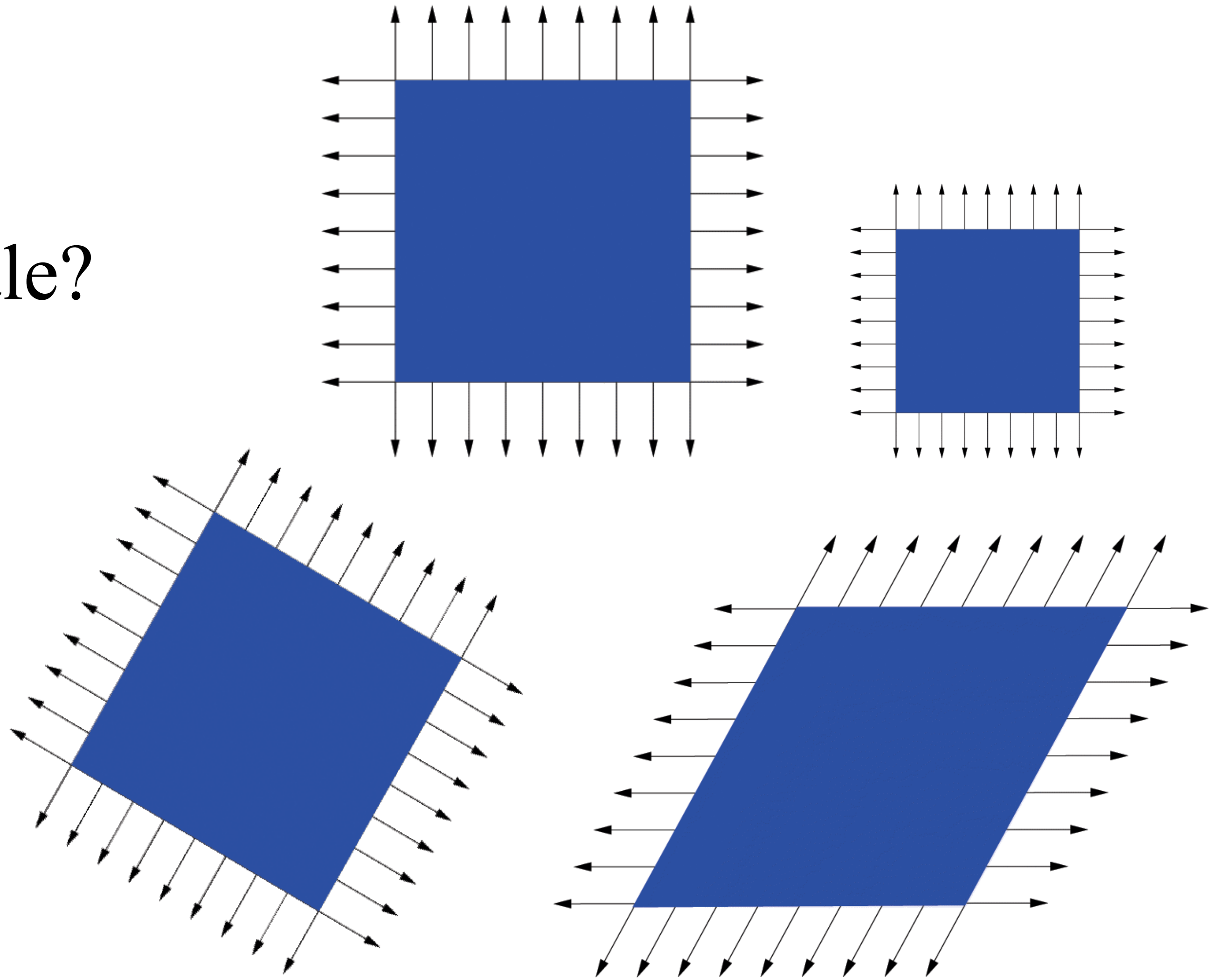
How do we transform normals?





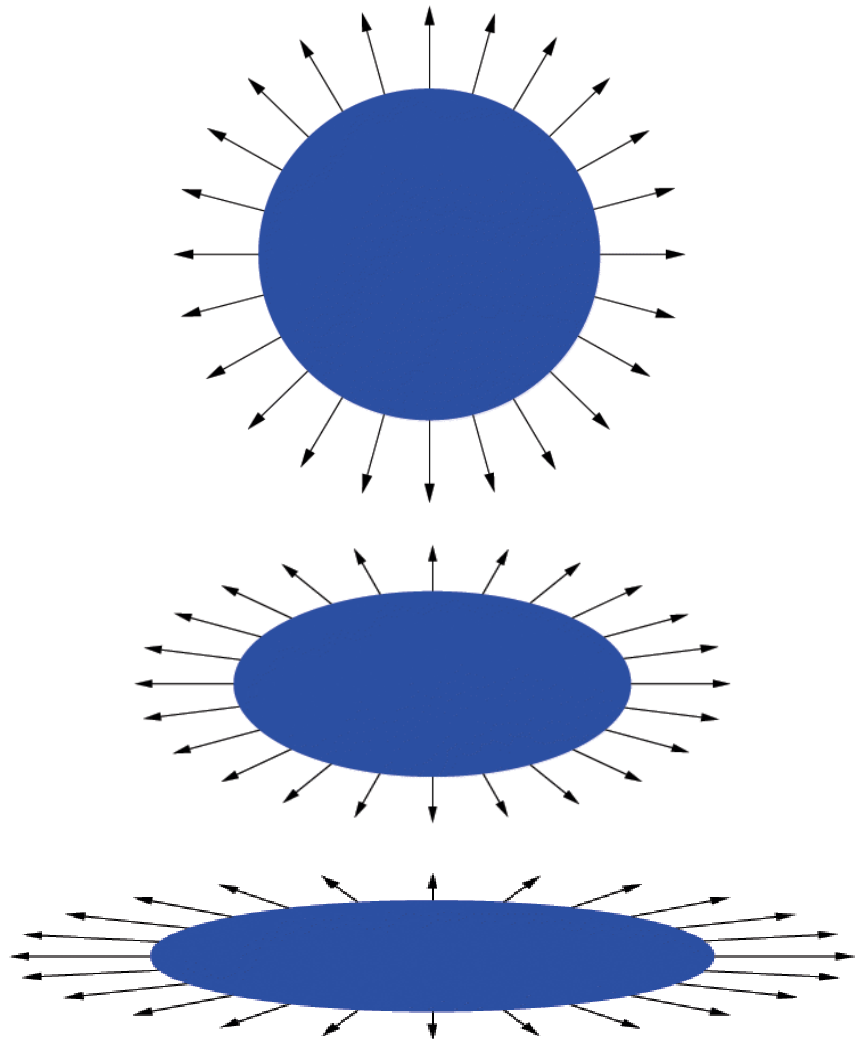
Transform the Normal like the Ray?

- translation?
- rotation?
- isotropic scale?
- scale?
- reflection?
- shear?
- perspective?

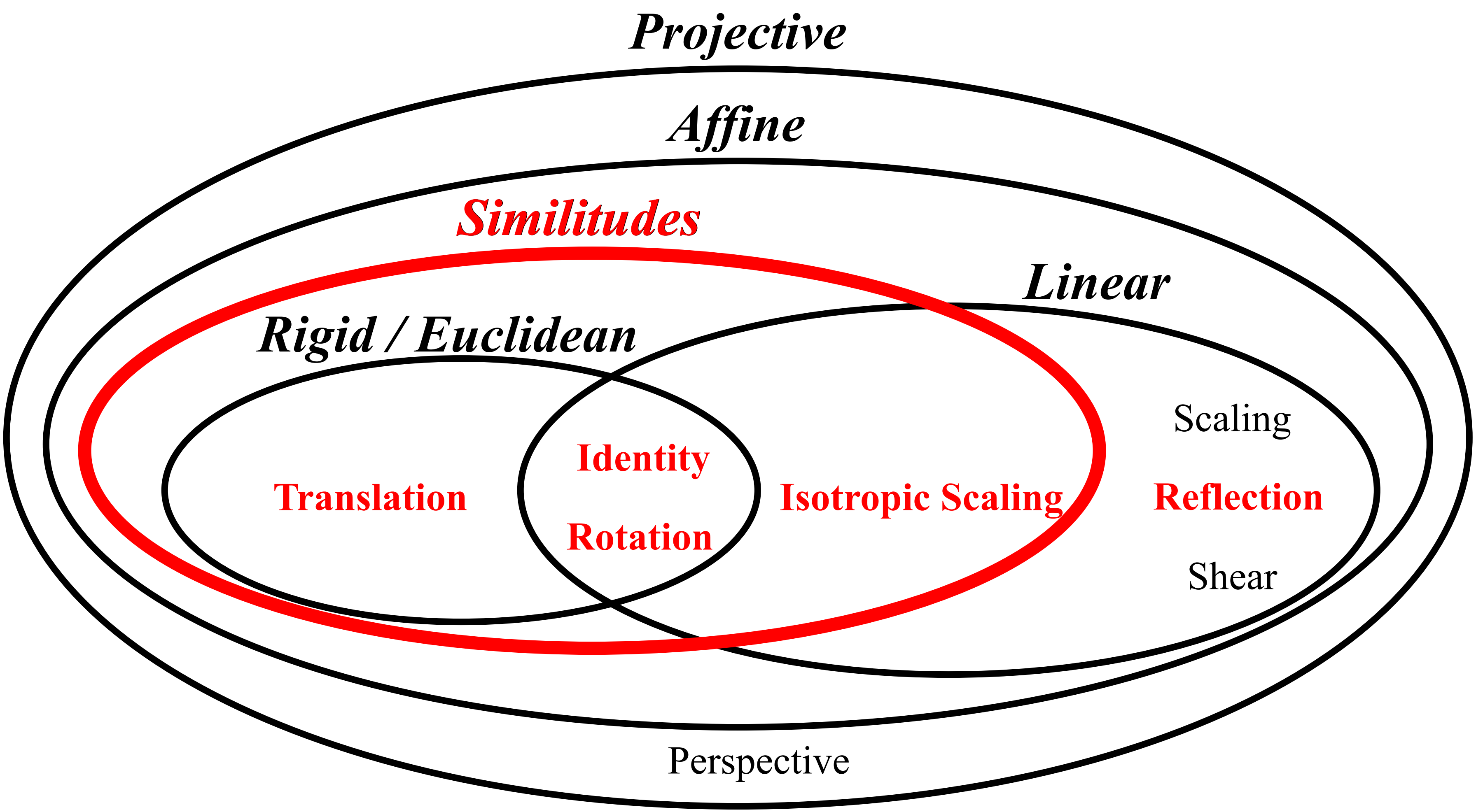


Transform the Normal like the Ray?

- translation?
- rotation?
- isotropic scale?
- scale?
- reflection?
- shear?
- perspective?



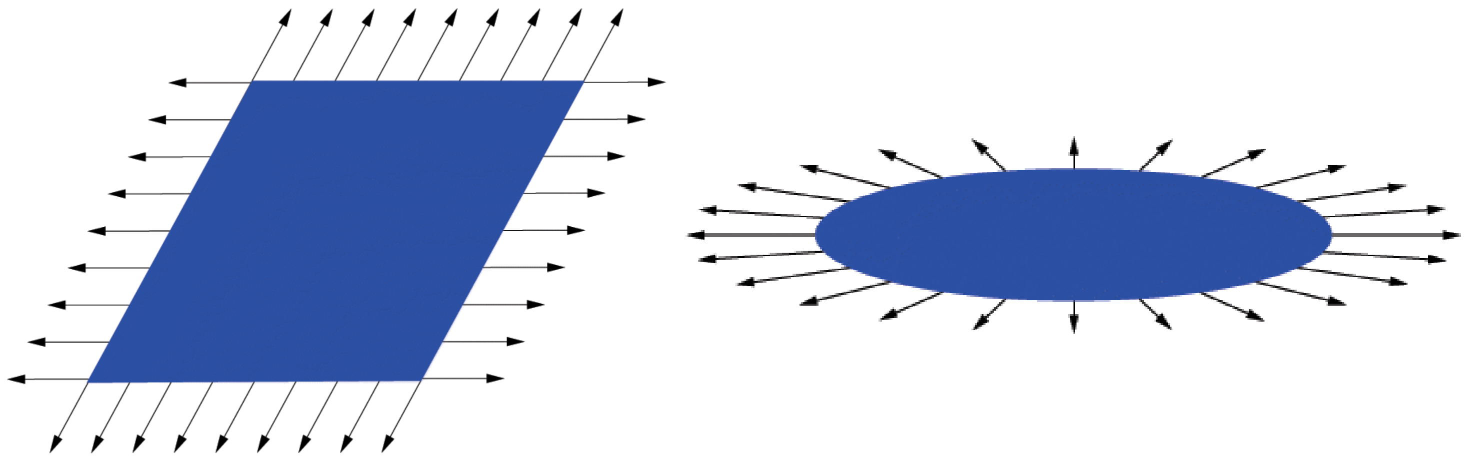
What class of transforms?



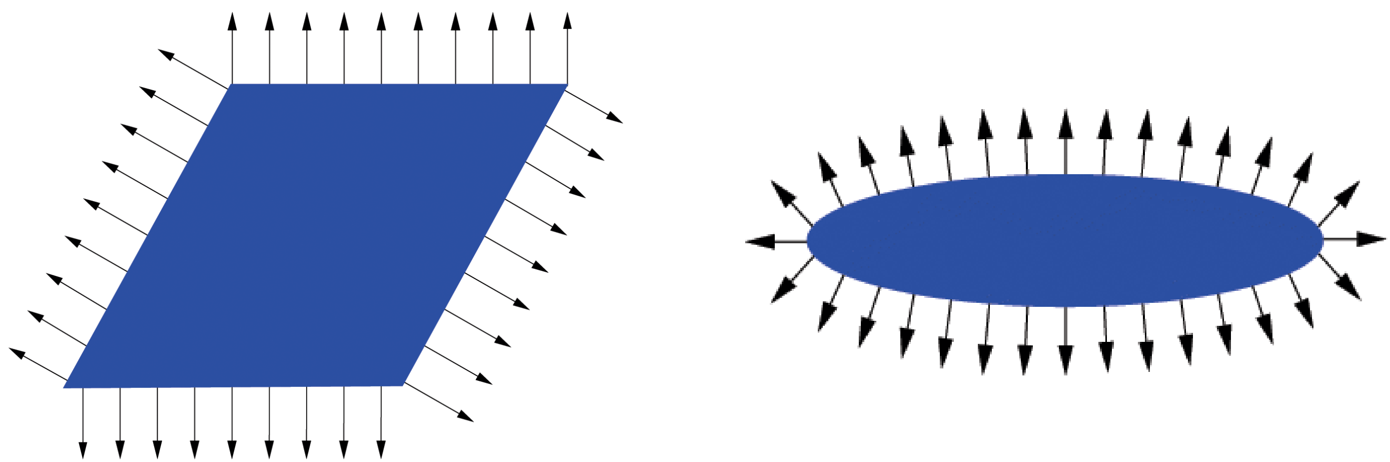
a.k.a. Orthogonal Transforms

Transformation for shear and scale

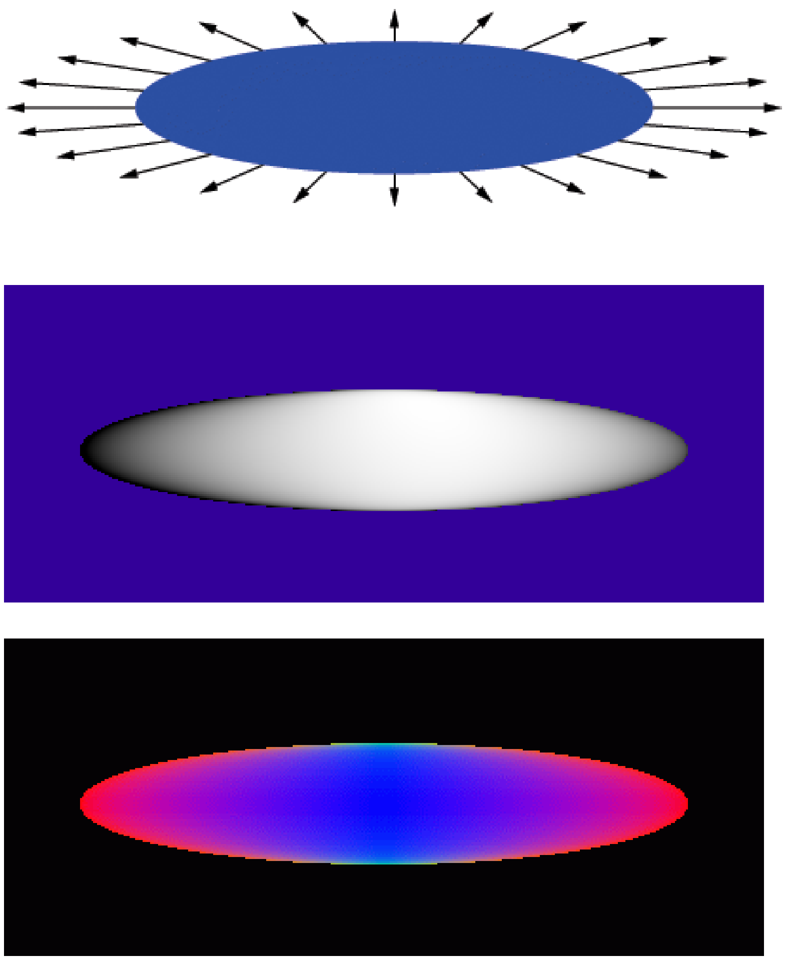
Incorrect
Normal
Transformation



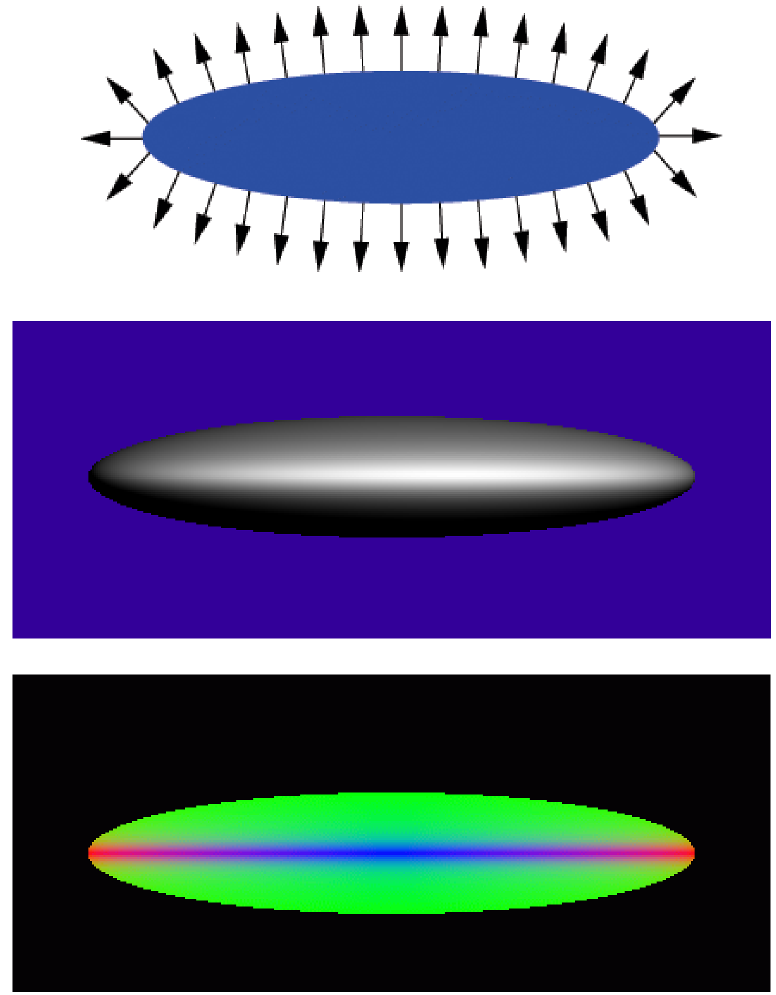
Correct
Normal
Transformation



More Normal Visualizations



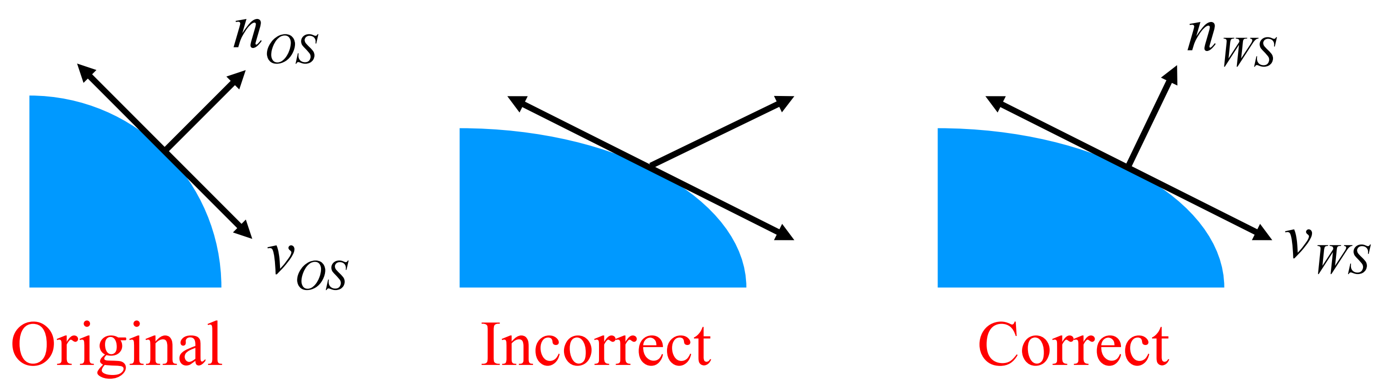
Incorrect Normal Transformation



Correct Normal Transformation

So how do we do it right?

- Think about transforming the *tangent plane* to the normal, not the normal *vector*



Pick any vector v_{OS} in the tangent plane, how is it transformed by matrix **M**?

$$v_{WS} = \mathbf{M} \; v_{OS}$$

Transform tangent vector v

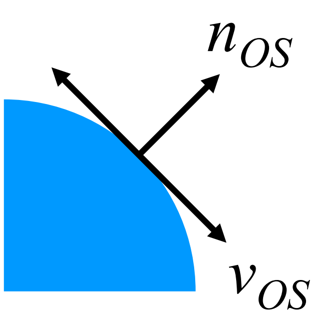
v is perpendicular to normal n:

Dot product $$n_{OS}^{\mathbf{T}}\, v_{OS} = 0$$

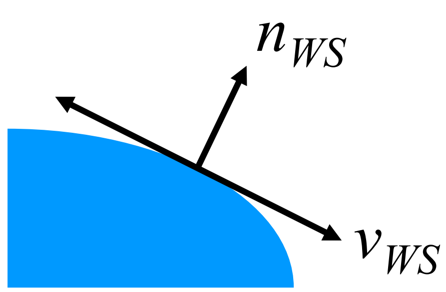
$$n_{OS}^{\mathbf{T}}\, (\mathbf{M}^{-1}\ \mathbf{M})\, v_{OS} = 0$$

$$(n_{OS}^{\mathbf{T}}\ \mathbf{M}^{-1})\, (\mathbf{M}\ \ v_{OS}) = 0$$

$$(n_{OS}^{\mathbf{T}}\ \mathbf{M}^{-1})\, v_{WS} = 0$$



v_{WS} is perpendicular to normal n_{WS}:



$$n_{WS}^{\mathbf{T}} = n_{OS}^{\mathbf{T}}\, (\mathbf{M}^{-1})$$

$$\boxed{n_{WS} = (\mathbf{M}^{-1})^{\mathbf{T}}\, n_{OS}}$$

$$n_{WS}^{\mathbf{T}}\, v_{WS} = 0$$

Comment

- So the correct way to transform normals is:

$$n_{WS} = (\mathbf{M}^{-1})^{\mathrm{T}}\, n_{OS}$$

Sometimes noted M^{-T}

- But why did $n_{WS} = \mathbf{M}\, n_{OS}$ work for similitudes?
- Because for similitude / similarity transforms,

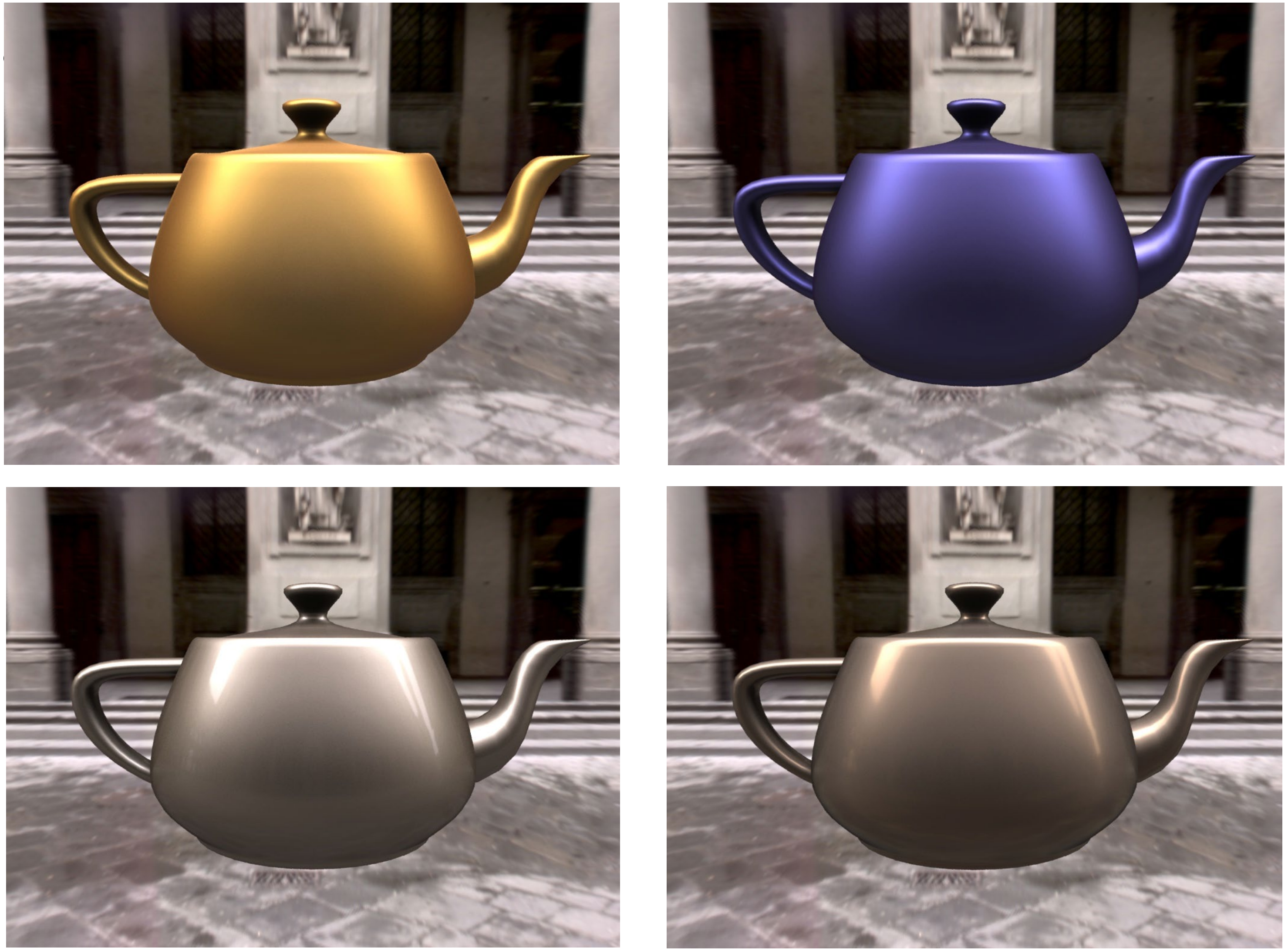
$$(\mathbf{M}^{-1})^{\mathrm{T}} = \lambda\, \mathbf{M}$$

- e.g. for orthonormal basis:

$$\mathbf{M} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \qquad \mathbf{M}^{-1} = \begin{pmatrix} x_u & x_v & x_n \\ y_u & y_v & y_n \\ z_u & z_v & z_n \end{pmatrix}$$

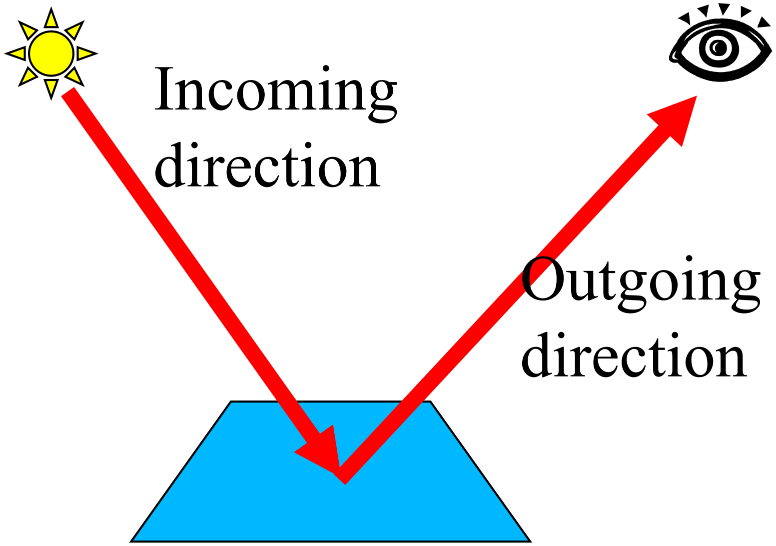
Questions?

Local Illumination



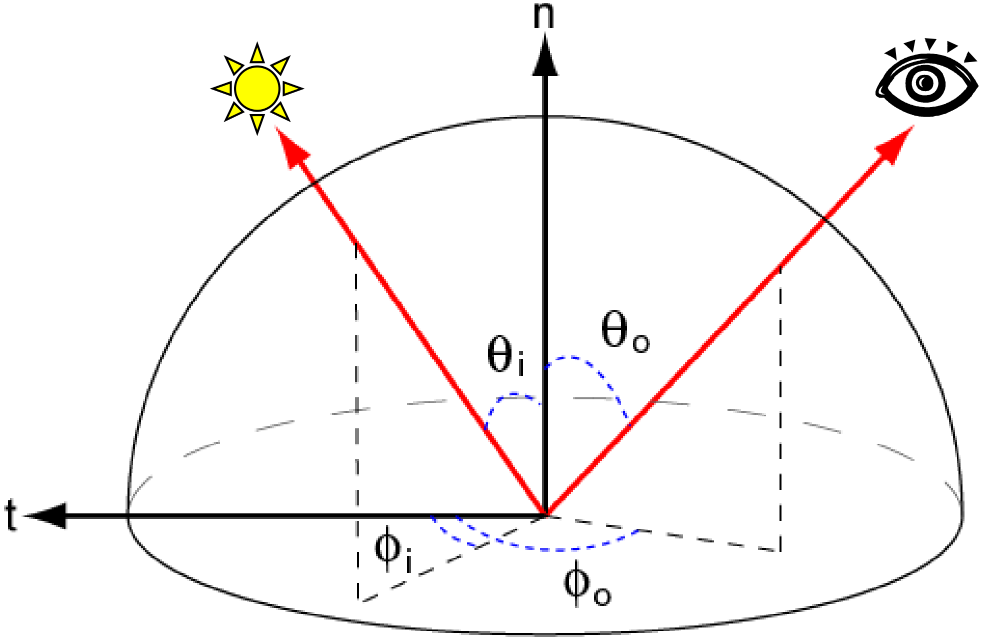
BRDF

- Ratio of light coming from one direction that gets reflected in another direction
- Bidirectional Reflectance Distribution Function



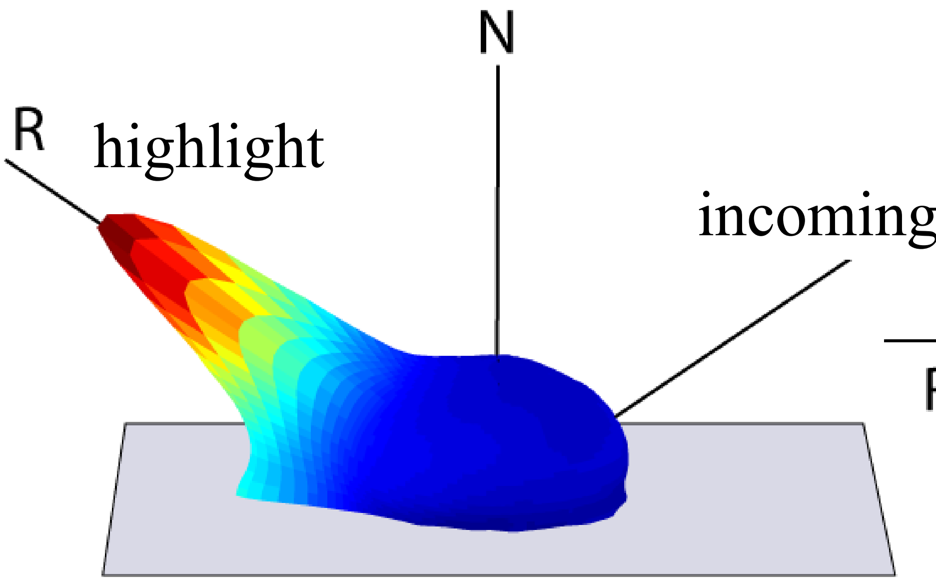
BRDF

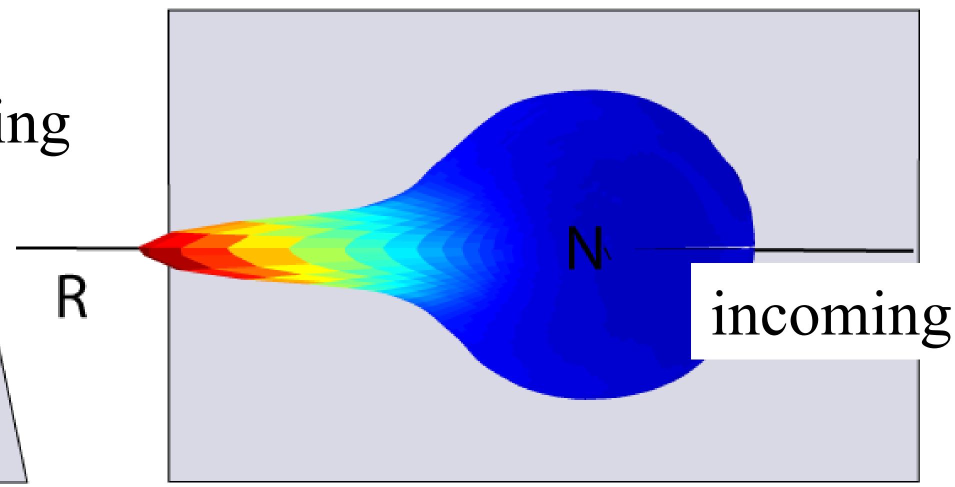
- Bidirectional Reflectance Distribution Function
  - 4D
    - 2 angles for each direction
  - $R(\theta_i, \phi_i; \theta_o, \phi_o)$



Slice at constant incidence

- 2D spherical function





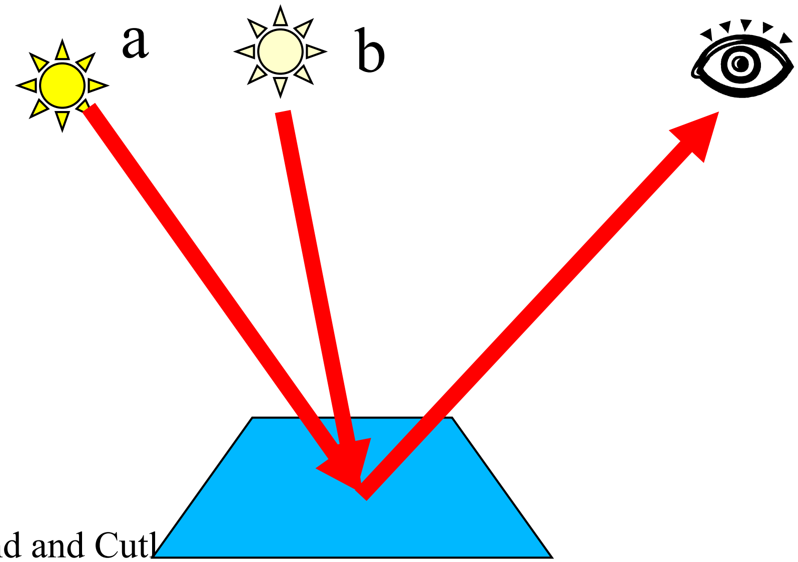
Example: Plot of “PVC” BRDF at 55° incidence

Unit issues - radiometry

- We will not be too formal in this lecture
- Typical issues:
  - Directional quantities vs. integrated over all directions
  - Differential terms: per solid angle, per area, per time
  - Power, intensity, flux

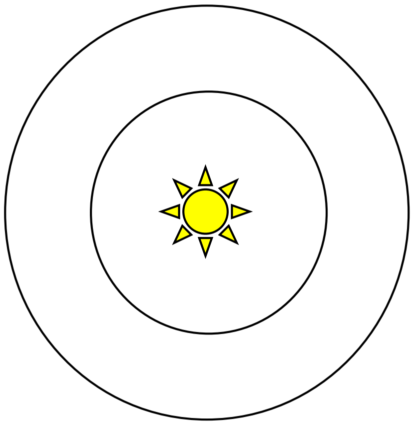
Light sources

- Today, we only consider point light sources
- For multiple light sources, use linearity
  - We can add the solutions for two light sources
    - I(a+b)=I(a)+I(b)
  - We simply multiply the solution when we scale the light intensity
    - I(s a) = s I(a)



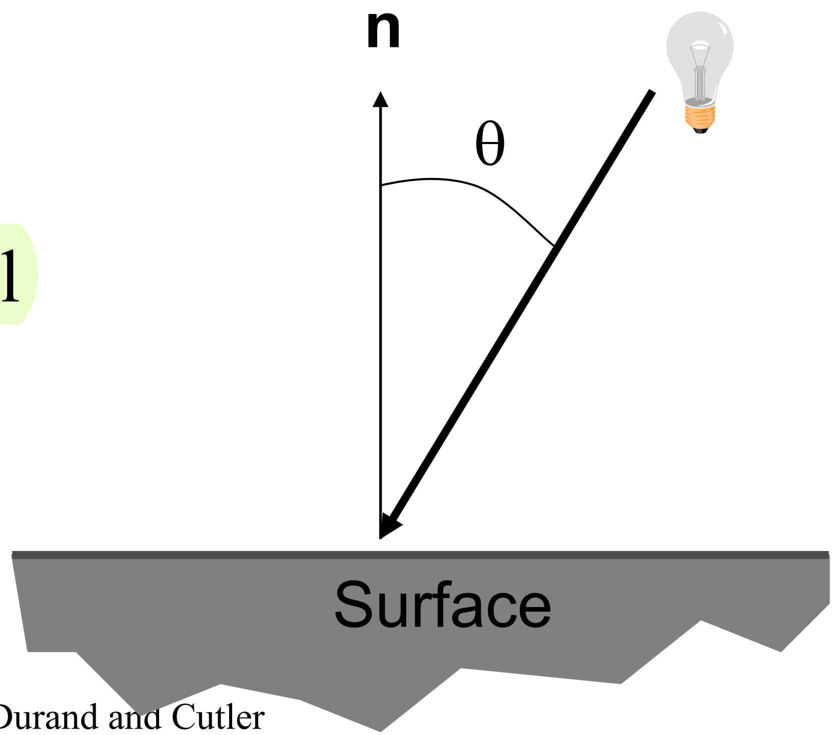
Light intensity

- $1/r^2$ falloff
  - Why?
  - Same power in all concentric circles



Incoming radiance

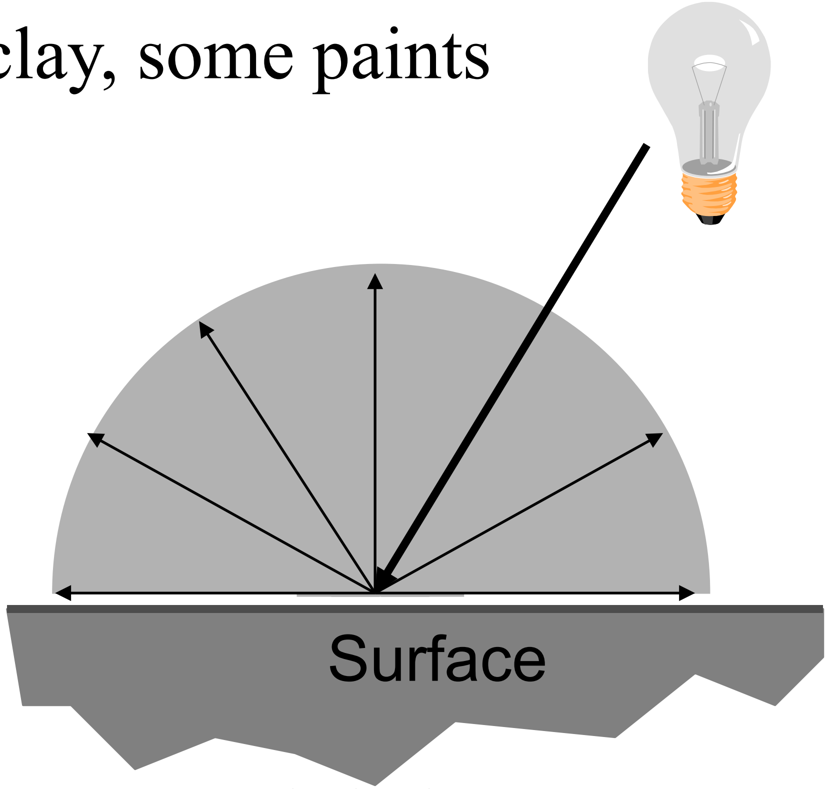
- The amount of light received by a surface depends on incoming angle
  - Bigger at normal incidence
    - Similar to Winter/Summer difference
- By how much?
  - Cos θ law
  - Dot product with normal
  - This term is sometimes included in the BRDF, sometimes not



Questions?

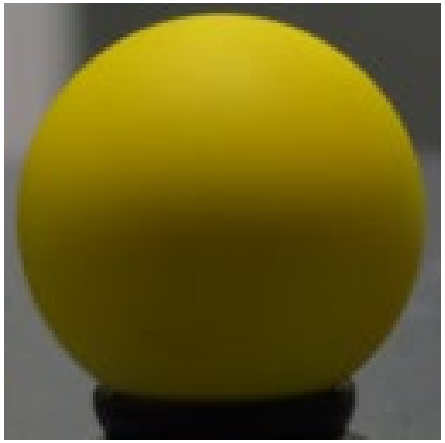
Ideal Diffuse Reflectance

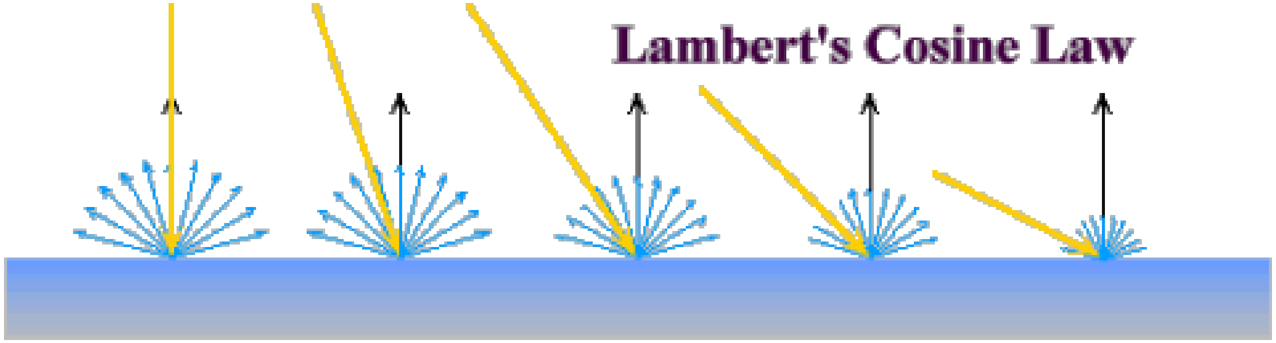
- Assume surface reflects equally in all directions.
- An ideal diffuse surface is, at the microscopic level, a very rough surface.
  - Example: chalk, clay, some paints



Ideal Diffuse Reflectance

- Ideal diffuse reflectors reflect light according to Lambert's cosine law.

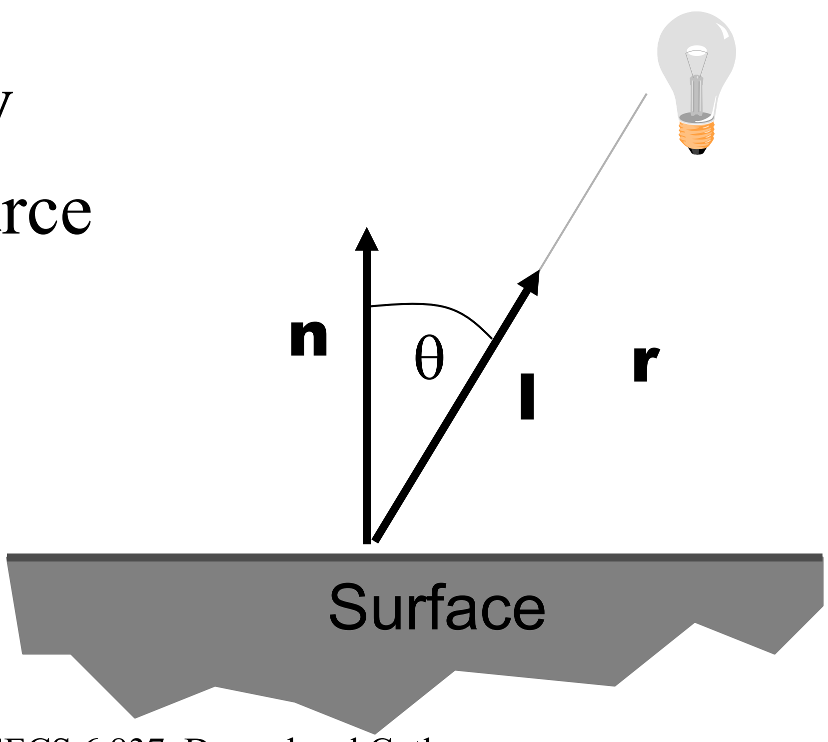




Ideal Diffuse Reflectance

- Single Point Light Source
  - k_d: diffuse coefficient.
  - **n**: Surface normal.
  - **l**: Light direction.
  - L_i: Light intensity
  - r: Distance to source

$$L_o = k_d (\mathbf{n} \cdot \mathbf{l}) \frac{L_i}{r^2}$$



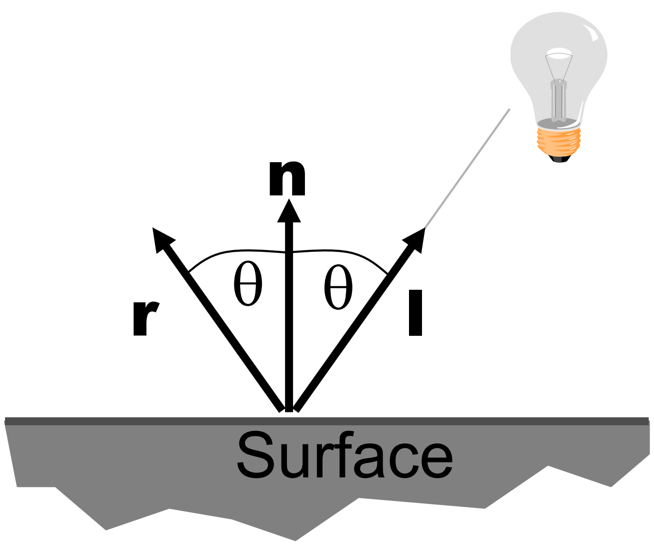
Ideal Diffuse Reflectance – More Details

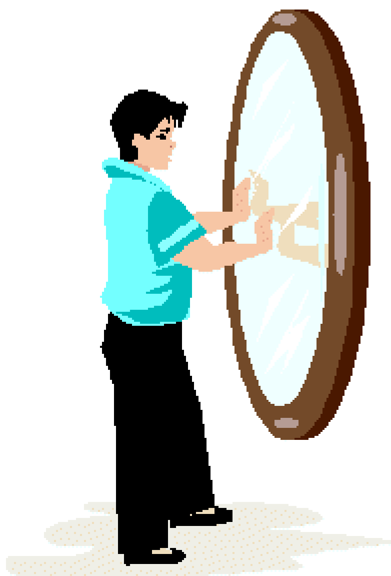
- If **n** and **l** are facing away from each other, **n • l** becomes negative.
- Using max((**n • l**),0) makes sure that the result is zero.
  - From now on, we mean max() when we write •.
- Do not forget to **normalize your vectors** for the dot product!

Questions?

Ideal Specular Reflectance

- Reflection is only at mirror angle.
  - View dependent
  - Microscopic surface elements are usually oriented in the same direction as the surface itself.
  - Examples: mirrors, highly polished metals



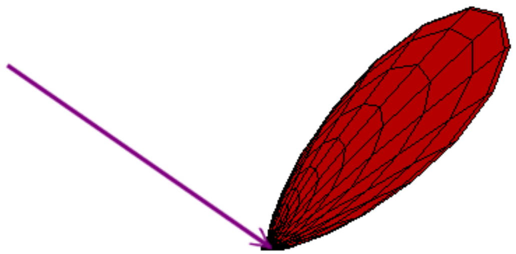


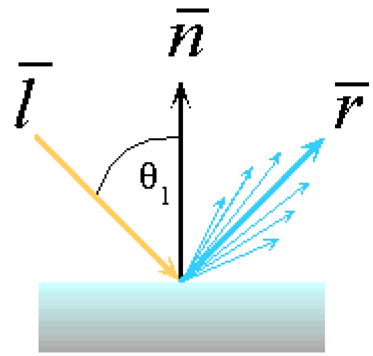
Non-ideal Reflectors

- Real materials tend to deviate significantly from ideal mirror reflectors.
- Highlight is blurry
- They are not ideal diffuse surfaces either …

Non-ideal Reflectors

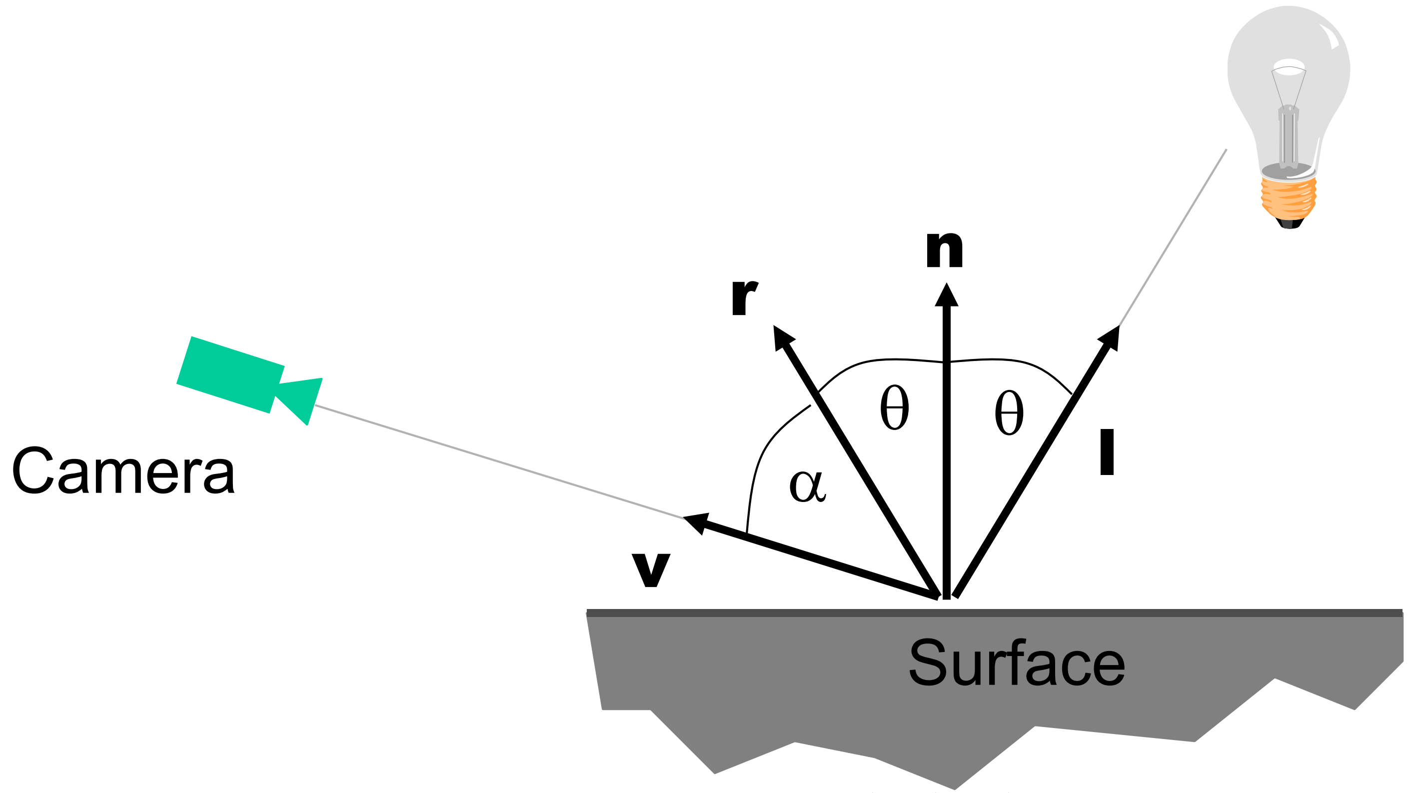
- Simple Empirical Model:
  - We expect most of the reflected light to travel in the direction of the ideal ray.
  - However, because of microscopic surface variations we might expect some of the light to be reflected just slightly offset from the ideal reflected ray.
  - As we move farther and farther, in the angular sense, from the reflected ray we expect to see less light reflected.





The Phong Model

- How much light is reflected?
  - Depends on the angle between the ideal reflection direction and the viewer direction α.

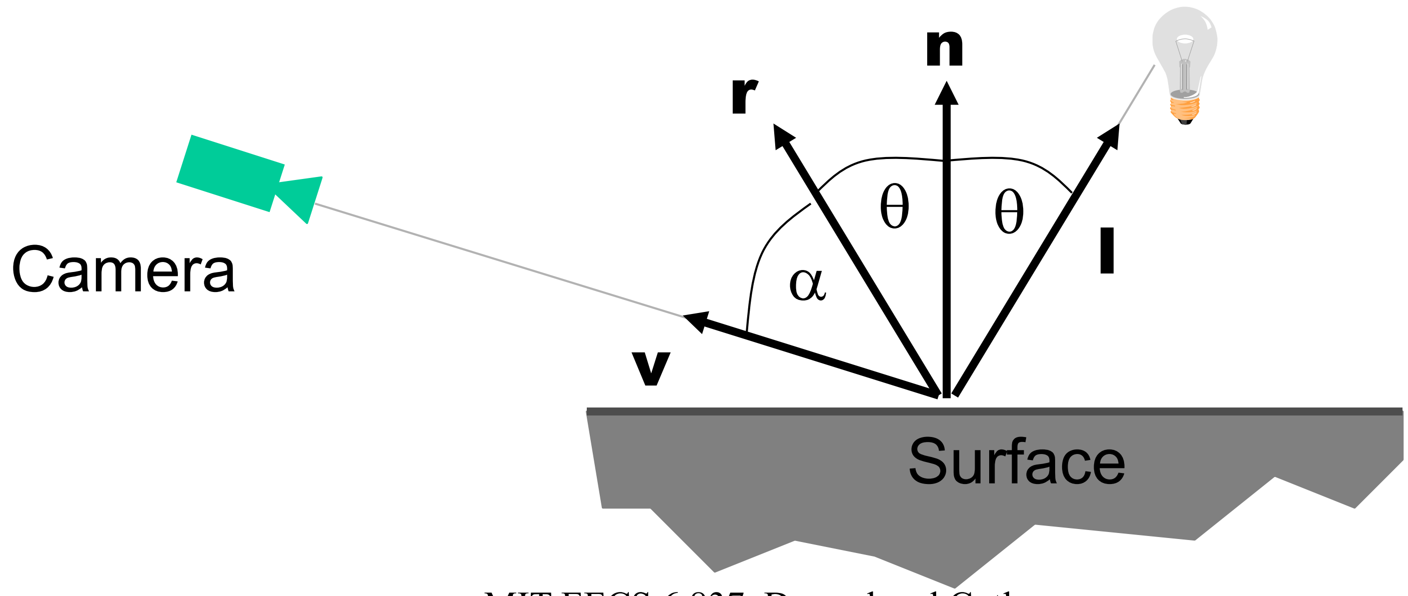


The Phong Model

- Parameters
  - k_s: specular reflection coefficient
  - q : specular reflection exponent

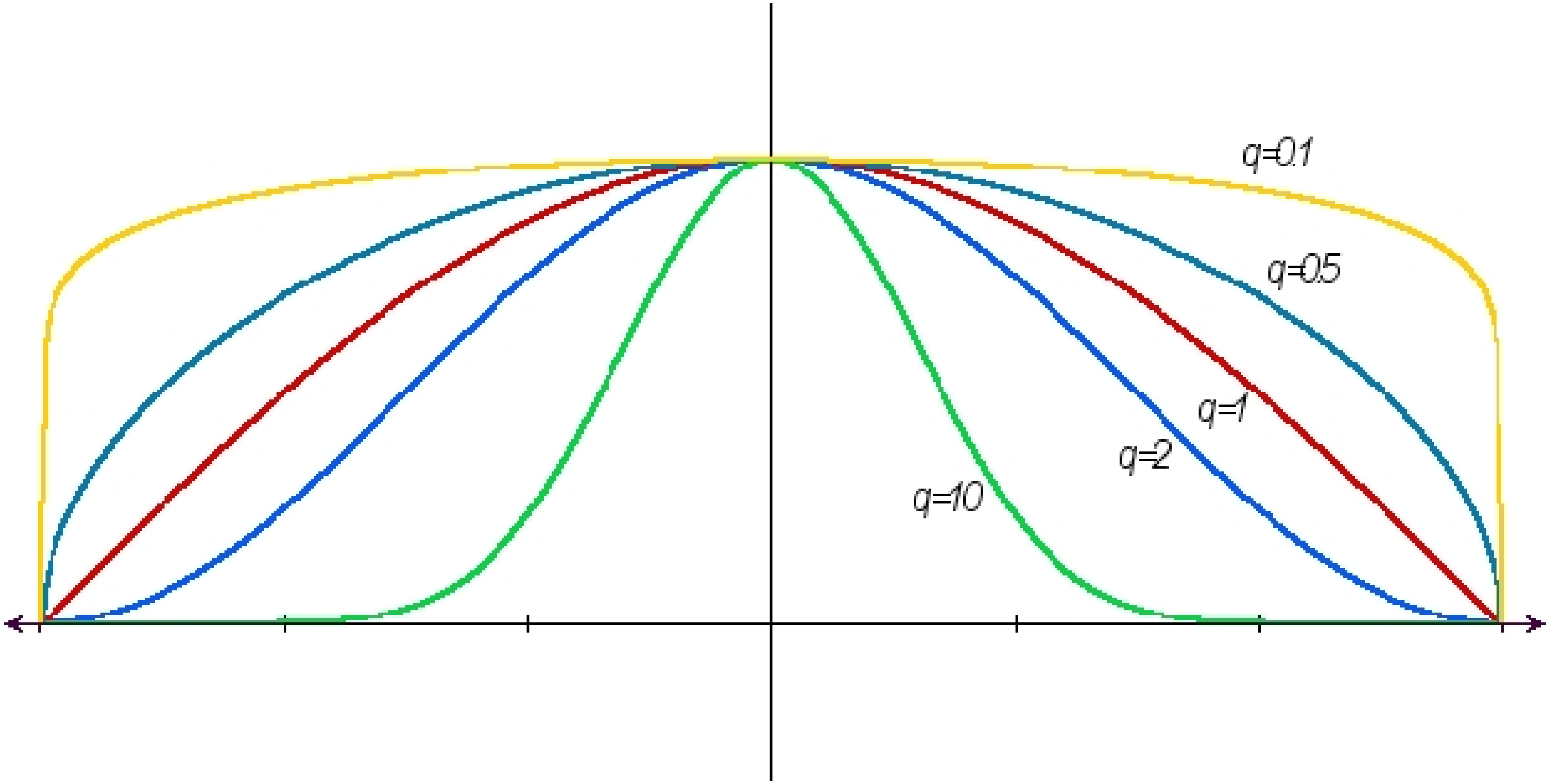
$$L_o = k_s (\cos\alpha)^q \frac{L_i}{r^2}$$

$$L_o = k_s (\mathbf{v} \cdot \mathbf{r})^q \frac{L_i}{r^2}$$



The Phong Model

- Effect of the q coefficient



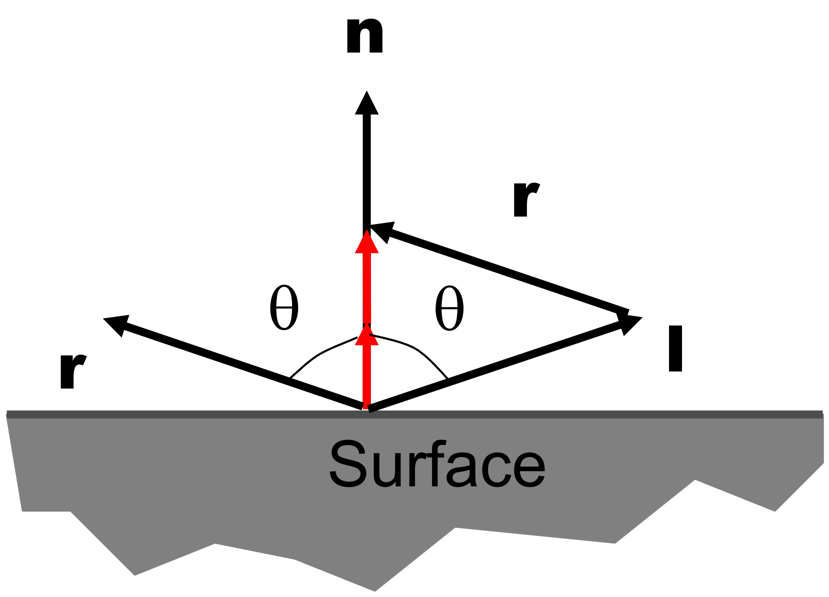
How to get the mirror direction?

$$\mathbf{r} + \mathbf{l} = 2\ \cos\theta\,\mathbf{n}$$

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

$$L_o = k_s(\mathbf{v} \cdot \mathbf{r})^q \frac{L_i}{r^2} =$$

$$= k_s(\mathbf{v} \cdot (2(\mathbf{n} \cdot \mathbf{l})\mathbf{n} - \mathbf{l}))^q \frac{L_i}{r^2}$$

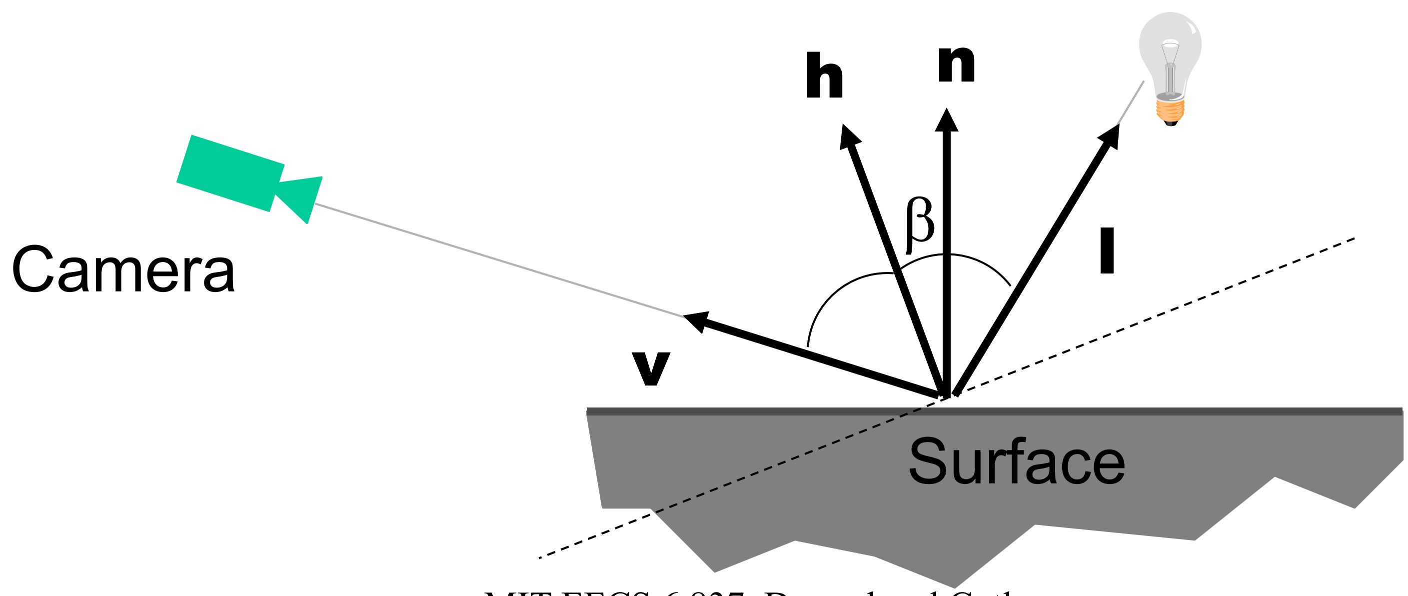


Blinn-Torrance Variation

- Uses the halfway vector **h** between **l** and **v**.

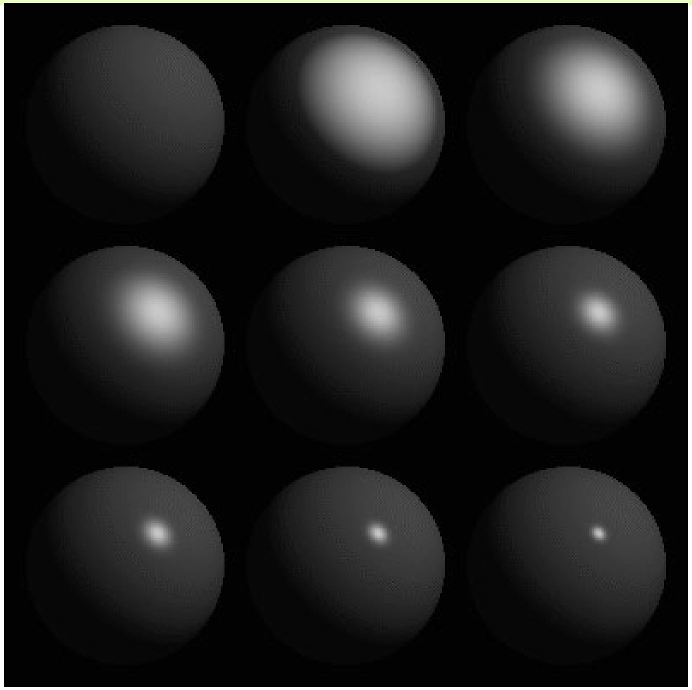
$$\mathbf{h} = \frac{\mathbf{l}+\mathbf{v}}{\|\mathbf{l}+\mathbf{v}\|}$$

$$L_o = k_s (\cos\beta)^q \frac{L_i}{r^2} = k_s (\mathbf{n}\cdot\mathbf{h})^q \frac{L_i}{r^2}$$

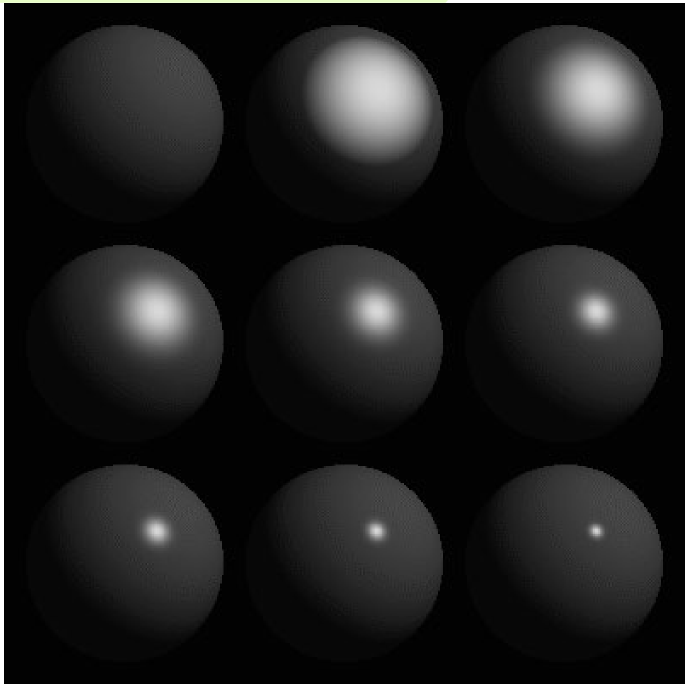


Phong Examples

- The following spheres illustrate specular reflections as the direction of the light source and the coefficient of shininess is varied.



Phong



Blinn-Torrance

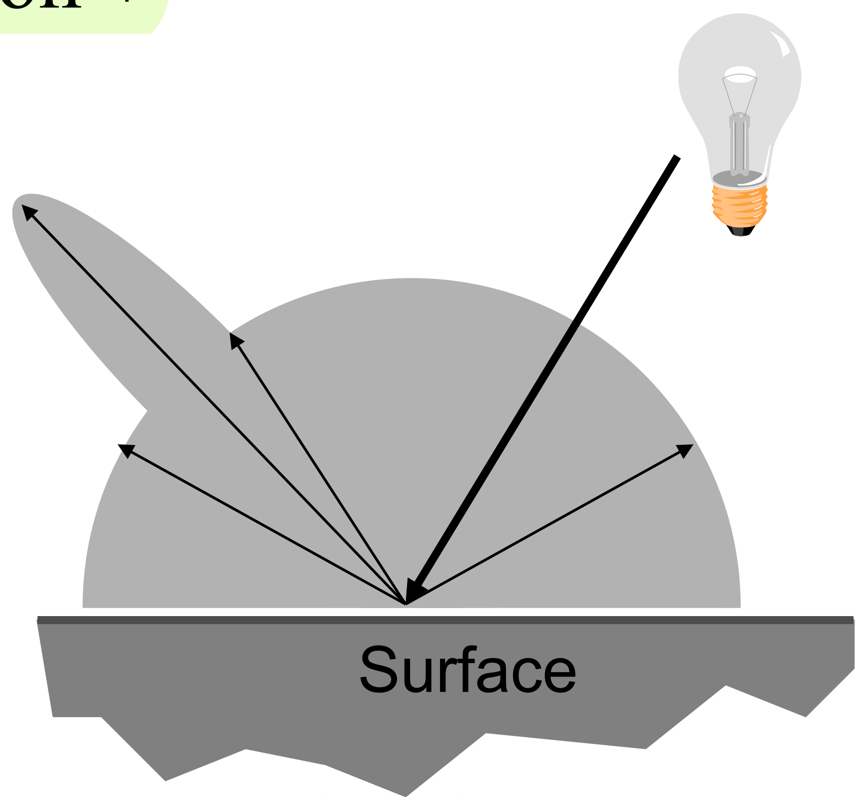
The Phong Model

- Sum of three components:

  diffuse reflection +

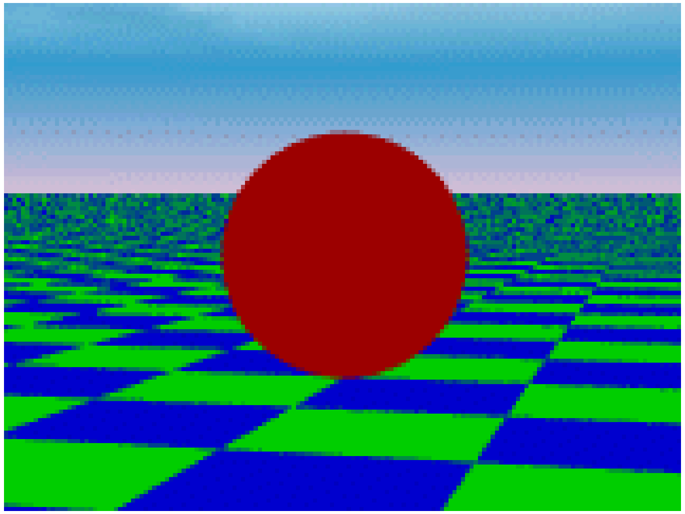
  specular reflection +

  "ambient".



Ambient Illumination

- Represents the reflection of all indirect illumination.
- This is a total hack!
- Avoids the complexity of global illumination.

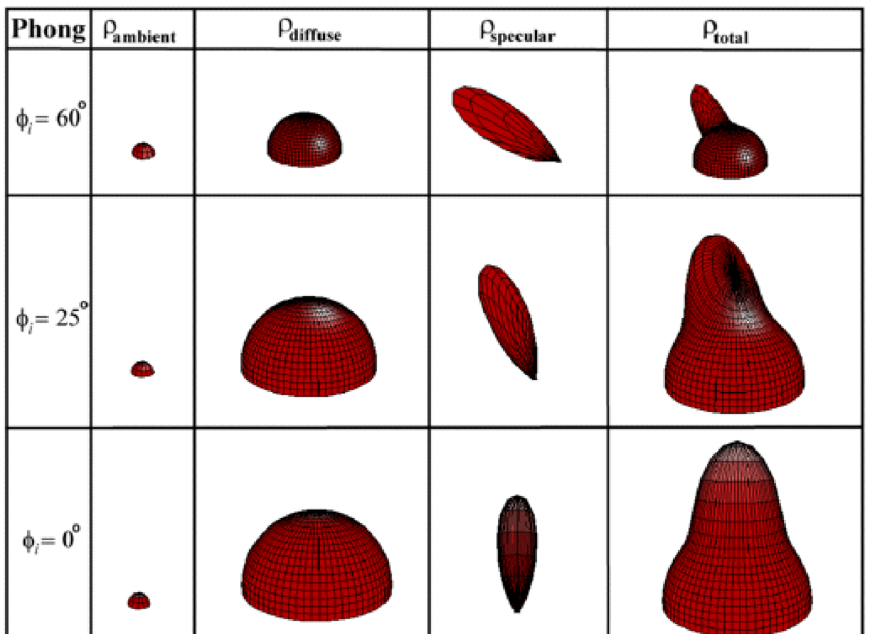


$$L(\omega_r) = k_a$$

Putting it all together

- Phong Illumination Model

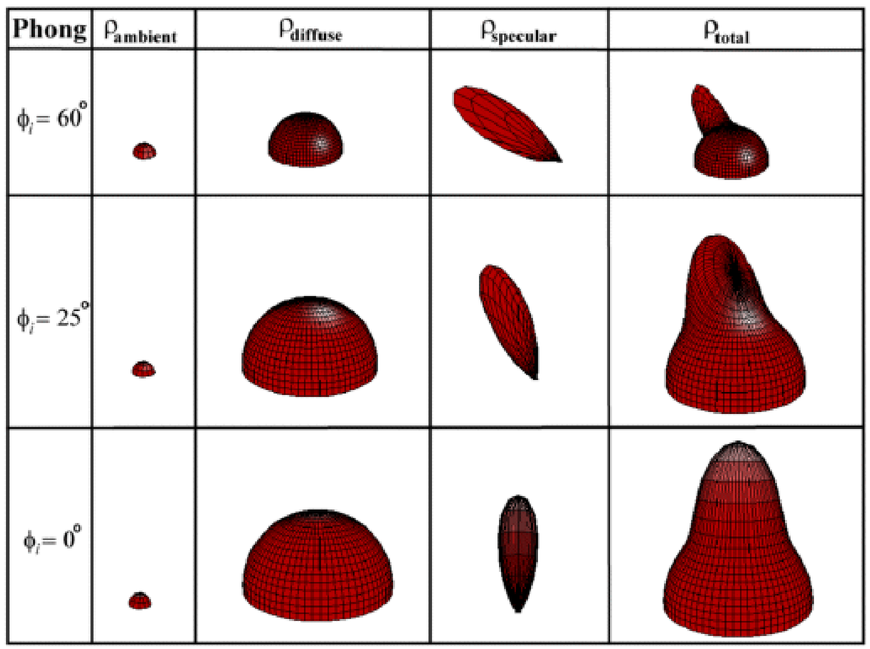
$$L_o = k_a + \left(k_d (\mathbf{n} \cdot \mathbf{l}) + k_s (\mathbf{v} \cdot \mathbf{r})^q \right) \frac{L_i}{r^2}$$



For Assignment 3

- Variation on Phong Illumination Model

$$L_o = k_a L_a + \left(k_d (\mathbf{n} \cdot \mathbf{l}) + k_s (\mathbf{v} \cdot \mathbf{r})^q \right) \frac{L_i}{r^2}$$



Adding color

- Diffuse coefficients:
  - $k_{d\text{-}red}$, $k_{d\text{-}green}$, $k_{d\text{-}blue}$
- Specular coefficients:
  - $k_{s\text{-}red}$, $k_{s\text{-}green}$, $k_{s\text{-}blue}$
- Specular exponent:

  q

Questions?

Shaders (`Material` class)

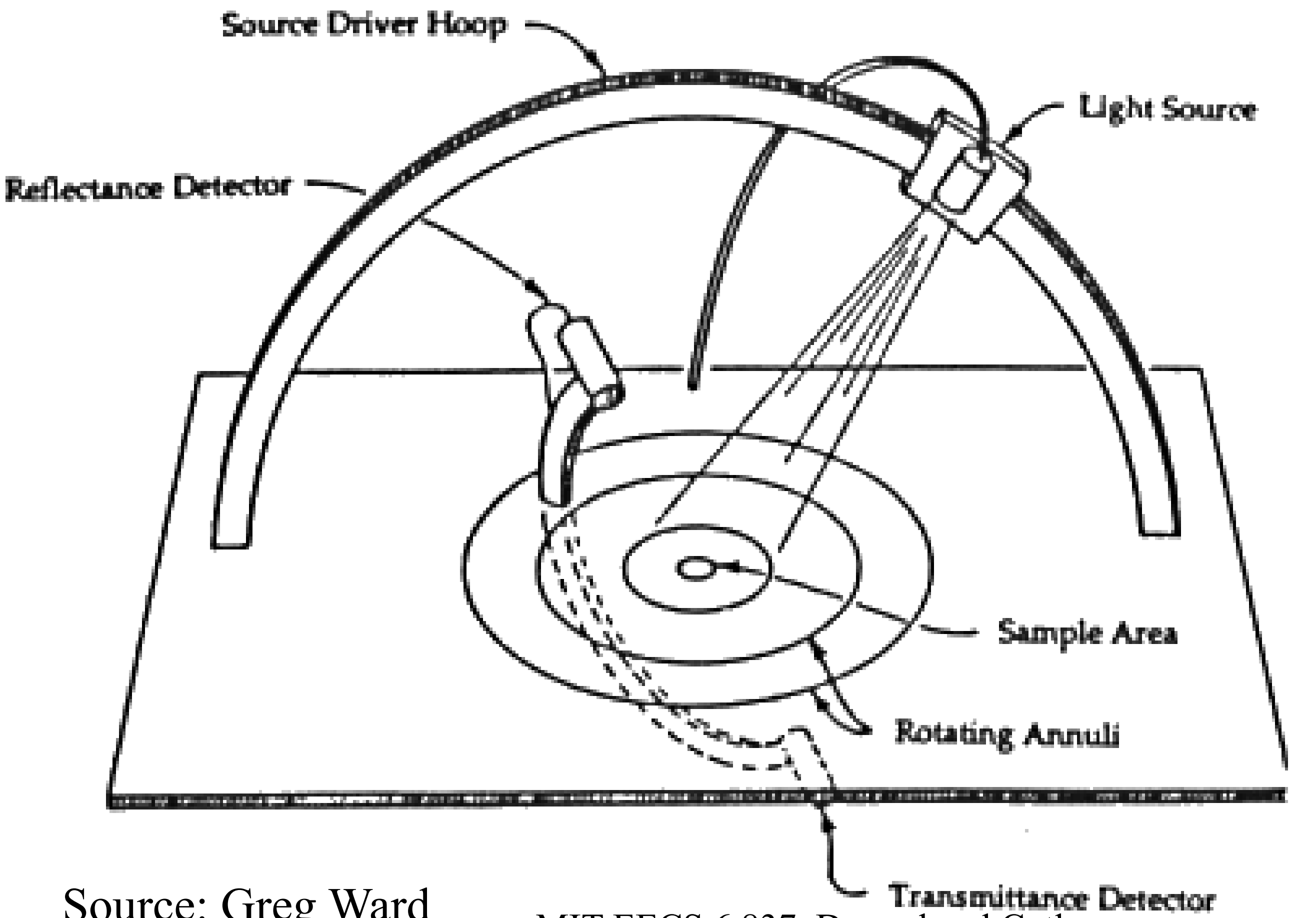
- Functions executed when light interacts with a surface
- Constructor:
  - set shader parameters
- Inputs:
  - Incident radiance
  - Incident & reflected light directions
  - surface tangent (anisotropic shaders only)
- Output:
  - Reflected radiance

BRDFs in the movie industry

- http://www.virtualcinematography.org/publications/acrobat/BRDF-s2003.pdf
- For the Matrix movies
- Clothes of the agent Smith are CG, with measured BRDF

How do we obtain BRDFs?

- Gonioreflectometer
  - 4 degrees of freedom



Source: Greg Ward

BRDFs in the movie industry

- http://www.virtualcinematography.org/publications/acrobat/BRDF-s2003.pdf
- For the Matrix movies

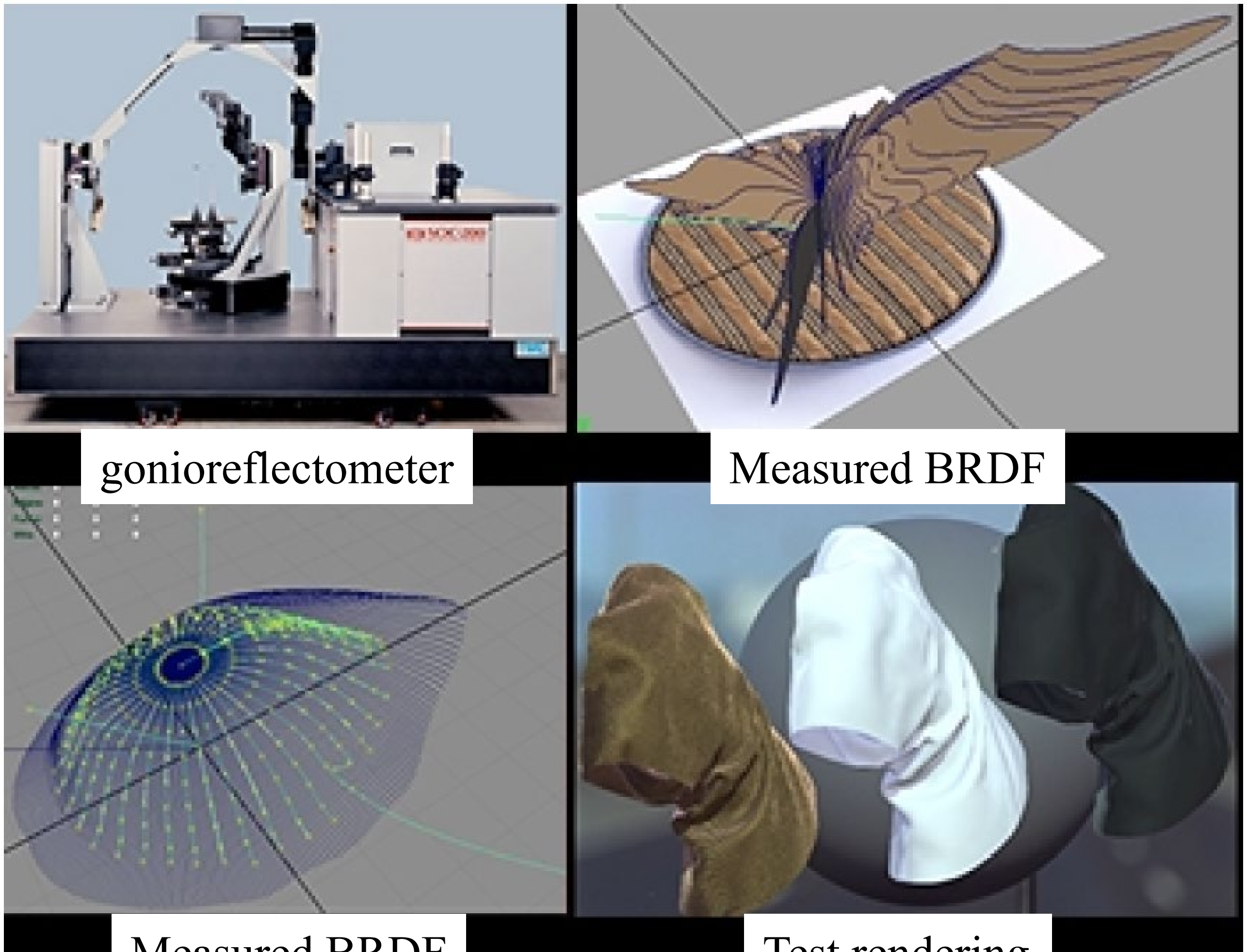
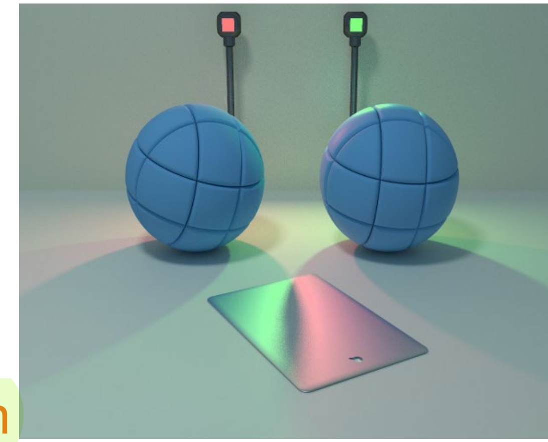


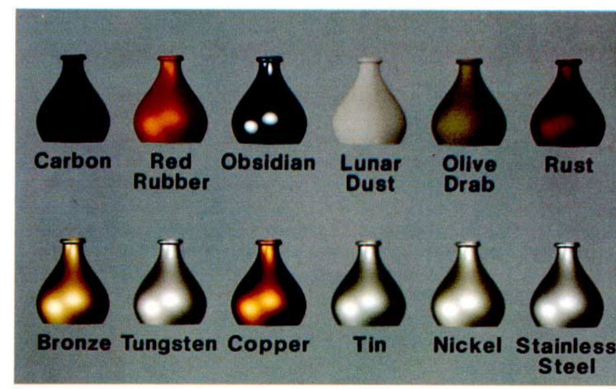
Photo CG Photo CG

BRDF Models

- Phenomenological
  - Phong [75]
    - Blinn-Phong [77]
  - Ward [92]
  - Lafortune et al. [97]
  - Ashikhmin et al. [00]
- Physical
  - Cook-Torrance [81]
  - He et al. [91]

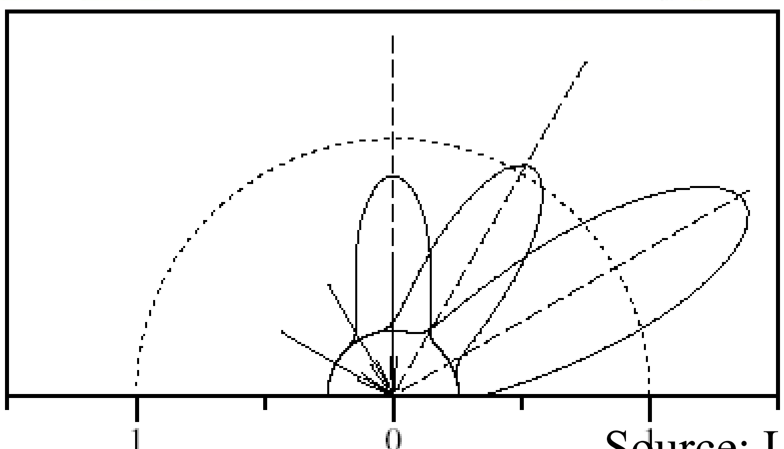
Roughly increasing computation time





Fresnel Reflection

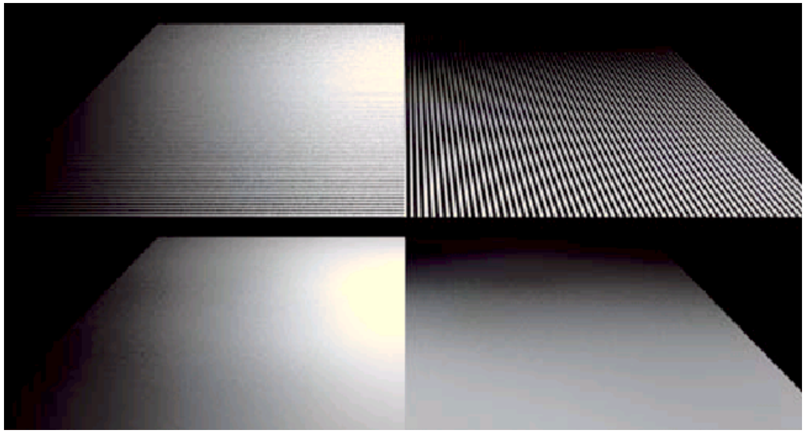
- Increasing specularity near grazing angles.



Source: Lafortune et al. 97

Anisotropic BRDFs

- Surfaces with strongly oriented microgeometry elements
- Examples:
  - brushed metals,
  - hair, fur, cloth, velvet



Source: Westin et.al 92

Off-specular & Retro-reflection

- Off-specular reflection
  - Peak is not centered at the reflection direction
- Retro-reflection:
  - Reflection in the direction of incident illumination
  - Examples: Moon, road markings

Questions?