

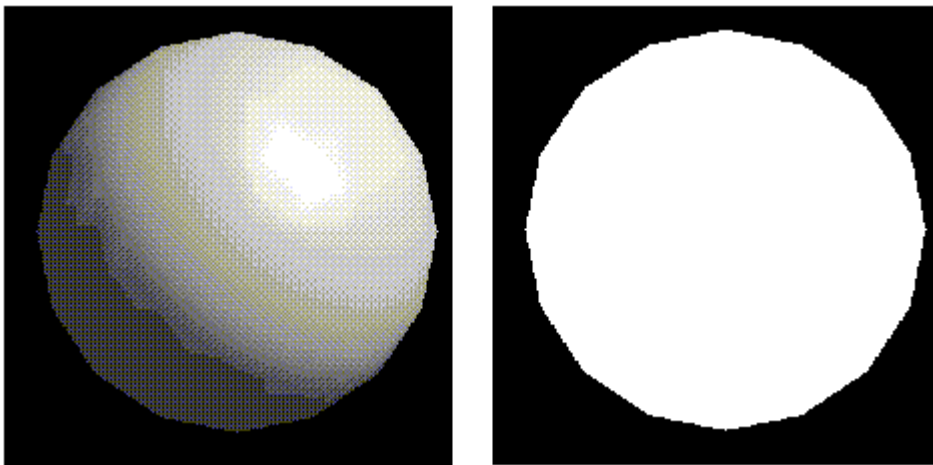
Course Progress:

Since Midterm:

- More detail on 3D viewing
- Polygonal and hierarchical modelling
- Silhouettes - Eric Schenk Guest Lecture
- Curves and Surfaces (Kori & John)
 - Parametric cubic curves
 - Hermite curves
 - Bezier curves
 - C & G continuity
 - de Casteljau Algorithm
 - parametric surfaces
 - Hermite surfaces
 - Bezier patches
- Visible Surface Determination - (Tom and Torsten)
 - problem of determining visible surfaces
 - image space vs. object space
 - Algorithms: painter's, reverse painter's, scanline, ray casting, z-buffer
 - Efficiency considerations
- Illumination and Shading (starting today)
- (Skipping Colour)

A critical step in the rendering of three-dimensional scenes is the final assignment of colors to pixels. The perspective projection process determines *where* a pixel is drawn on the screen. Visible surface computations determine *which* pixels are to be drawn. *How* the pixels are assigned colors is determined by the illumination models.

Must consider the interaction of objects in our world with lights. Why is this important?



In computer graphics, lighting can make all the difference when distinguishing 2D objects from 3D objects.

The colors that we perceive on the surfaces of objects result from complex interactions between light and matter. Illumination models can be either empirical or physically based. An empirical model results from the fitting of a function to observed data. Physically based models, on the other hand, attempt to simulate the interactions between light and various materials.

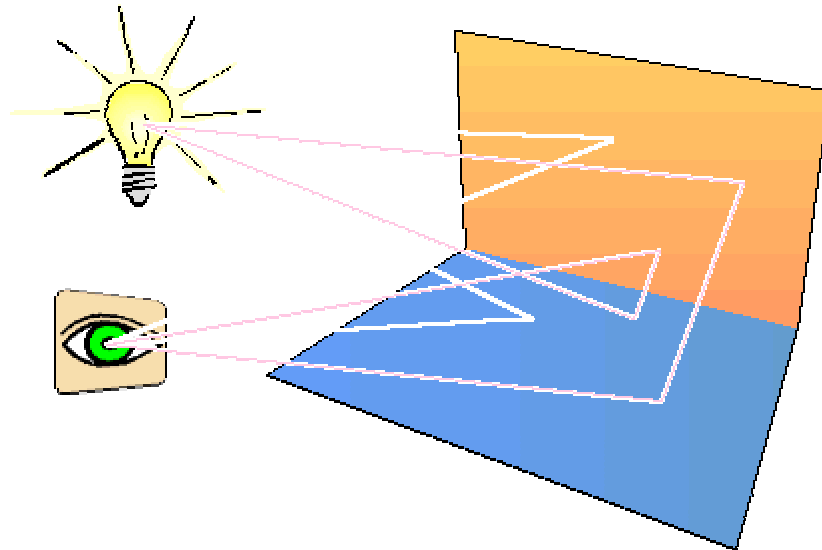
In computer graphics, we can consider two different yet closely related issues:

- lighting/illumination models: express factors related to light, which determine a surface's colour at a given point.
 - shading models: determine when an illumination model will be invoked and with which arguments. Some shading models invoke an illumination model for each pixel, whereas others use interpolation to shade pixels for which the illumination model was not explicitly used.
-
- computer graphics has traditionally employed a lot of simplifying assumptions, which are more or less accurate representations of the actual phenomena. Though many of the hacks employed

have little grounding in theory, they work well in practice and are easy to compute.

Light Sources

- Every object in a scene is potentially a source of light. Light may be either emitted or reflected from objects.
- The emitters are called light sources, and the reflectors are usually the objects being rendered.
- Light sources are characterized by their intensities while reflectors are characterized by their material properties.



Illumination Models

- factors that determine a surface's colour at a given point

Self-luminous objects:

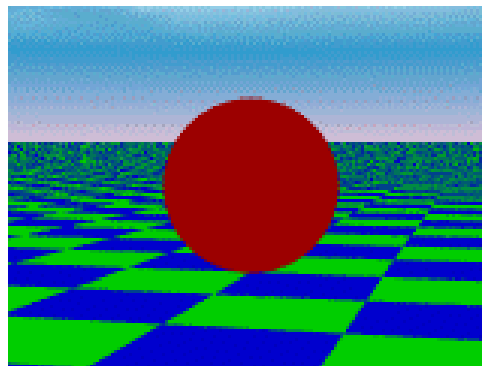
- each object is displayed using an intensity intrinsic to it
- I is resulting intensity and k_i is the object's intrinsic intensity
- $I = k_i$
- `intensity = intrinsic intensity of object`
- We only need to evaluate once for each object since the equation doesn't depend on the position of the point

Ambient light:

- Diffuse, nondirectional light source
- The product of multiple reflections of light from the many surfaces present in the environment
- We assume that ambient light impinges equally on all surfaces from all directions
- $I = I_a k_a$
- `intensity = (ambient light intensity) * (coefficient of ambient reflection)`
- I_a is the intensity of ambient light, assumed to be constant for all objects and k_a is ambient-

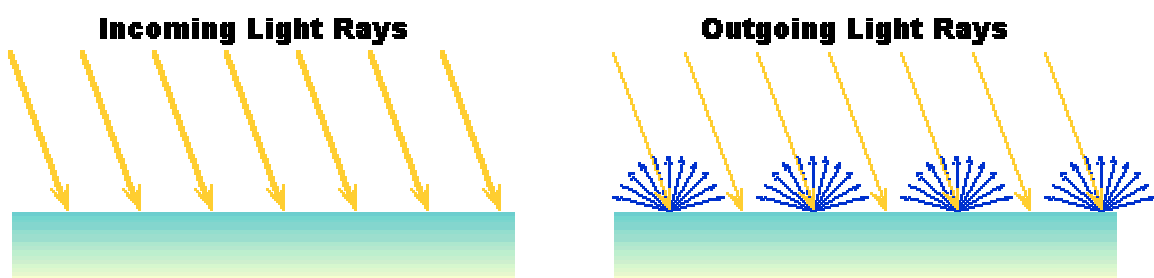
reflection coefficient, the amount of ambient light reflected from an object's surface. This value ranges from 0 to 1 and is a material property (but does not correspond directly to physical properties of real materials)

- Ambient light is often used to account for complex ways in which light can reach an object that are not otherwise addressed by the illumination equation.

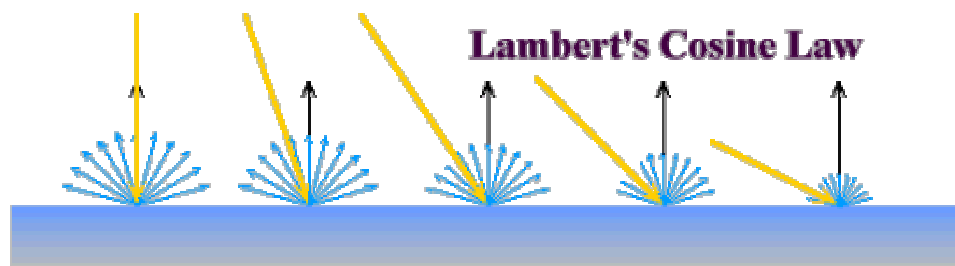


Diffuse Reflection

- Ambient light illuminates uniformly and does not account for the position nor the distance of the light source
- A more realistic illumination model locates light sources in the scene so their position and distance relative to objects can be determined.
- A simple light source which has this property is a point light source: located at a specific point, it emanates rays uniformly in all directions. Conceptually, we might consider it to be an incandescent bulb.
- Dull, matte surfaces such as chalk exhibit diffuse reflection also known as Lambertian reflection
- Their surfaces appear equally bright from all viewing angles because they reflect light with equal intensity in all directions



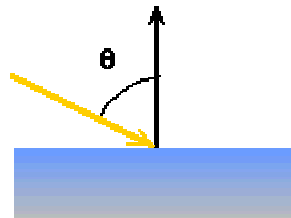
- Lambert's law states that the reflected energy from a small surface area in a particular direction is proportional to cosine of the angle between that direction and the surface normal.
- Lambert's law determines how much of the incoming light energy is reflected.



- remember that the amount energy that is reflected in any one direction is constant in this model. In other words the reflected intensity is independent of the viewing direction.
- The intensity does however depend on the light source's orientation relative to the surface, and it is this property that is governed by Lambert's law.

- The angle between the surface normal and the incoming light ray is called the angle of incidence and we can express a intensity of the light in terms of this angle.

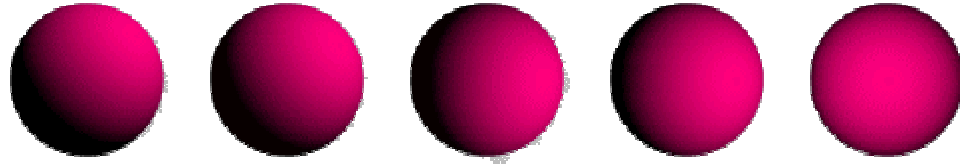
$$I_{diffuse} = k_d I_{light} \cos \theta$$



- I_{light} represents the intensity of the incoming light, k_d term represents the diffuse reflectivity of the surface
- we can take use vector analysis to compute this cosine term indirectly. If both the normal vector and the incoming light vector are normalized (unit length) then the diffuse shading component can be computed by:

$$I_{diffuse} = k_d I_{light} (\vec{n} \cdot \vec{l})$$

- In this equation we need only consider angles from 0 to 90 degrees. Greater angles are blocked by the surface, and the reflected energy is 0.



Examples with varying lighting examples

- If polygon normals are computed before modelling transformations, it is important that only rigid transformations be used. Shears or non-uniform scaling don't preserve angles so normals may no longer be perpendicular to their polygons.
- This illumination equation must be evaluated before normalization and projection since these transformations will modify the angle of light incidence.
- When a point light source is sufficiently far away from the objects in the scene, we consider it to be a directional light source and so the vector describing the direction to the light source is constant.
- It is more realistic to include ambient effects in the illumination calculations, which leads to:
- $I = I_a k_a + I_p k_d (\bar{N} \cdot \bar{L})$

```
intensity = (ambient light intensity) *
             (coefficient of ambient reflection) +
             (point light intensity) *
             (coefficient of diffuse reflection) *
             cosine (angle of light incidence)
```

Light-source attenuation

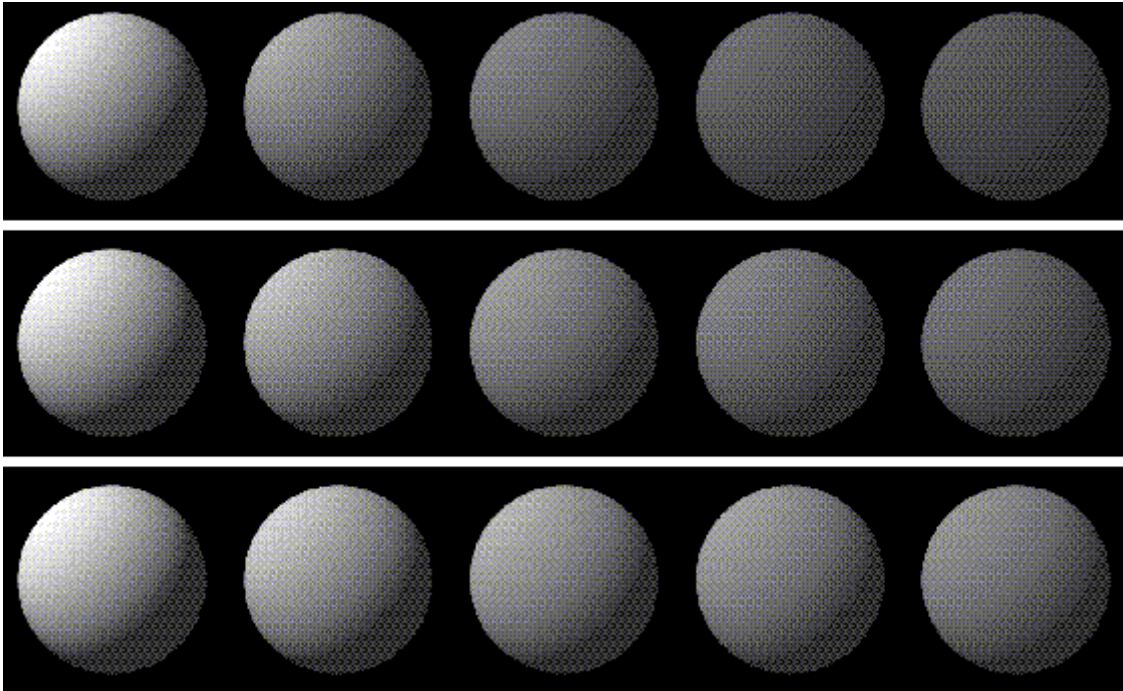
- Do light sources equally illuminate objects at different distances? There should be some notion of attenuation of light sources and we apply this term to scale the intensity derived from diffuse reflection from the object
- $I = I_a k_a + f_{att} I_p k_d (\bar{N} \bullet \bar{L})$

```
intensity = (ambient light intensity) *  
             (coefficient of ambient reflection) +  
             [light-source attenuation factor *  
              (point light intensity) *  
              (coefficient of diffuse reflection) *  
              cosine (angle of light incidence)]
```

- A reasonable choice is based on the inverse square of the distance ($1 / d_L^2$). However, this doesn't give the best results in practice (if light is far away, not much varies, if light is very close, it varies widely).
- A more robust formulation adds constant and linear terms to the denominator. We clamp the value to a maximum of 1, so it will never add energy.

$$f_{att} = \min \left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right)$$

c_1 , c_2 and c_3 are user defined constants for the light source



Example of spheres shaded using ambient and diffuse reflection with different light-source-attenuation constants

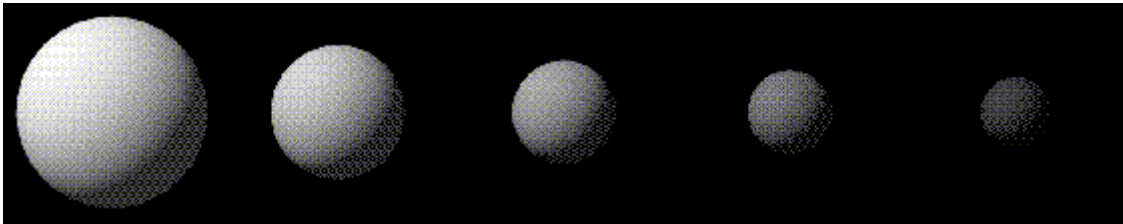
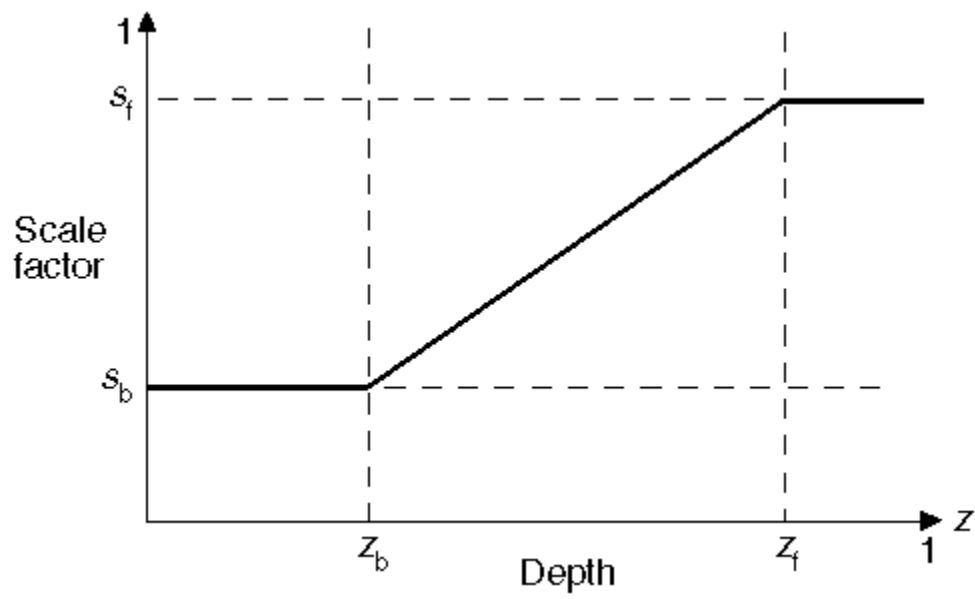
Coloured lights and surfaces

- coloured lights and surfaces are commonly treated by writing separate equations for each component of the colour model.
- An object's diffuse colour is represented by one value of O_d for each component.
- $$I_R = I_{aR}k_aO_{dR} + f_{att}I_{pR}k_dO_{dR} (\overline{N} \bullet \overline{L})$$

Atmospheric Attenuation

- depth cueing
- more distant objects are rendered with lower intensity than are closer ones
- front and back depth-cue reference planes are defined and each of these planes is associated with a scale factor, s_f and s_b , that ranges between 0 and 1.
- The scale factor determines the blending of the original intensity with that of a depth-cue colour
- $I' = s_o I + (1 - s_o) I_{dc}$
- If z_o is in front of the front depth-cue plane's z coordinate z_f , then $s_o = s_f$.
- If z_o is behind the back depth-cue plane's z coordinate z_b , then $s_o = s_b$.
- If z_o is between the two planes, then

$$s_o = s_b + \frac{(z_o - z_b)(s_f - s_b)}{(z_f - z_b)}$$

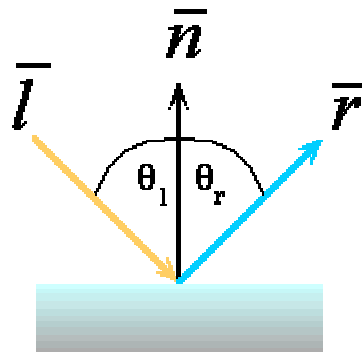


Example of spheres shaded using depth cueing.

Specular reflection

- specular reflection is the highlight that appears on a shiny surface when illuminated
- specular reflection is merely the mirror reflection of the light source in a surface
- In order to model specular reflection we need to understand the physics of reflection. Reflection behaves according to Snell's laws which state:
 - The incoming ray, the surface normal, and the reflected ray all lie in a common plane.
 - The angle that the reflected ray forms with the surface normal is determined by the angle that the incoming ray forms with the surface normal, and the relative speeds of light of the mediums in which the incident and reflected rays propagate according to the following expression:

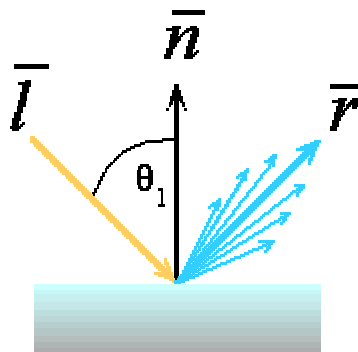
$$n_i \sin \theta_i = n_r \sin \theta_r$$



- Reflection is a very special case of Snell's Law where the incident light's medium and the reflected rays medium is the same. We can simplify the expression to:

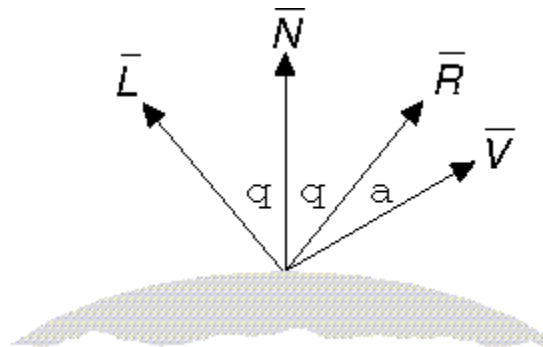
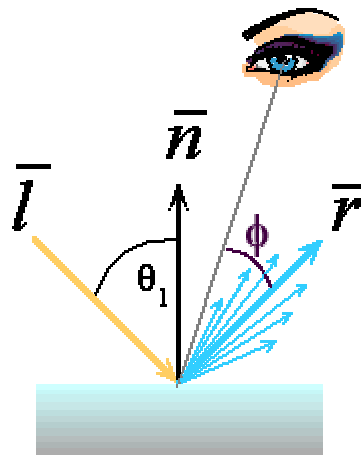
$$\theta_i = \theta_r$$

- Snell's law, however, applies only to ideal reflectors. Real materials, other than mirrors and chrome tend to deviate significantly from ideal reflectors.
- We will introduce an empirical model that is consistent with our experience, at least to a crude approximation.
- In general 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 reflect ray we expect to see less light reflected.



- One function that approximates this falloff is called the Phong Illumination model. This model is purely empirical and has no physical basis, yet it is one of the most commonly used illumination models in computer graphics.

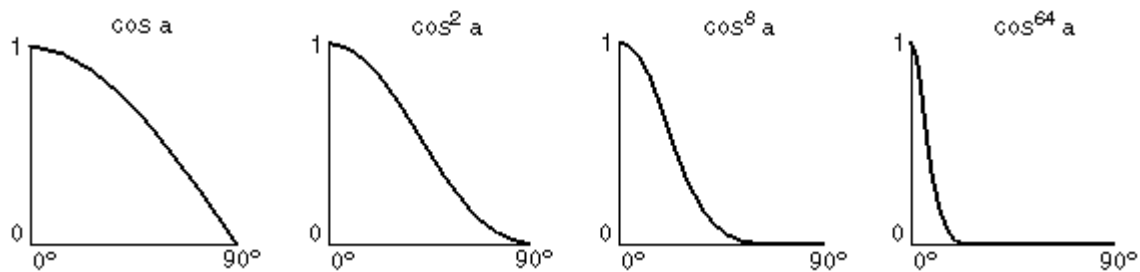
$$I_{\text{specular}} = I_{\text{light}} \cos^n \phi$$



- Maximum specular reflection occurs when ϕ is zero and falls off sharply as ϕ increases
- The rapid falloff is approximated by $\cos^n \phi$, where n is the material's specular-reflection exponent

(varying from 1 to several hundred depending on the material being simulated)

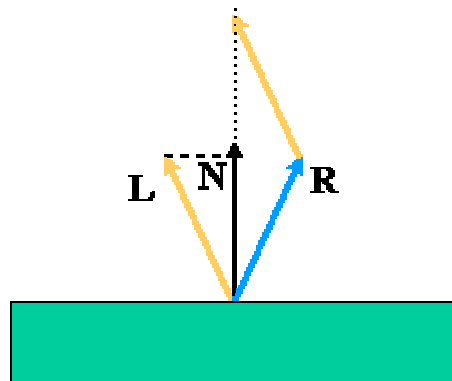
- For a perfect reflector, n would be infinite



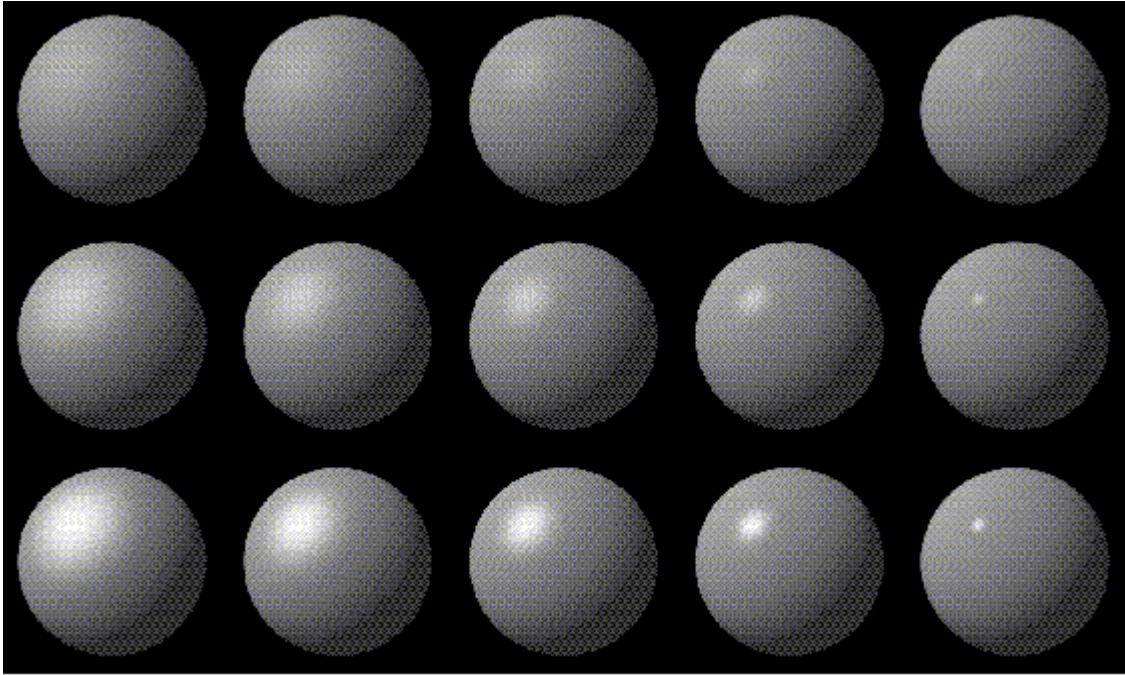
- If the direction of reflection \bar{R} and the viewpoint direction \bar{V} are normalized, then $\cos \phi = \bar{R} \cdot \bar{V}$
- The vector R can be computed from the incoming light direction and the surface normal as shown below.

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

$$\hat{R} + \hat{L} = (2(\hat{N} \cdot \hat{L}))\hat{N}$$



$$I = I_a k_a O_d + f_{att} I_p [k_d O_d (\bar{N} \cdot \bar{L}) + k_s O_s (\bar{R} \cdot \bar{V})^n]$$



From left to right, $n = 3, 5, 10, 27$. From top to bottom $k_s = 0.1, 0.25, 0.5$

Multiple light sources

- If there are m light sources, then the terms for each light source are summed:

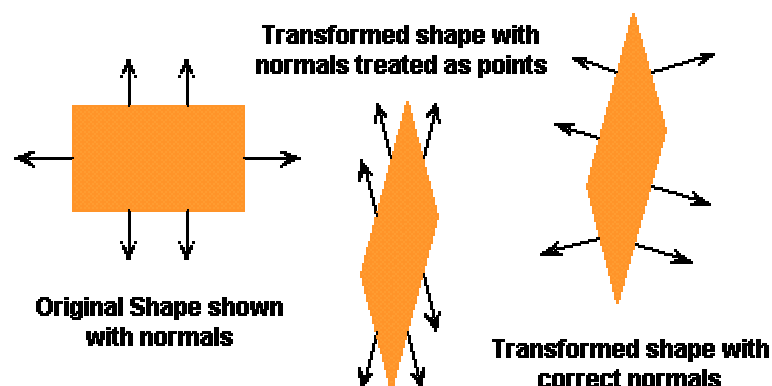
$$I = I_a k_a O_d + \sum f_{att} I_p [k_d O_d (\bar{N} \cdot \bar{L}) + k_s O_s (\bar{R} \cdot \bar{V})^n]$$

- Be careful that I doesn't exceed its maximum displayable pixel value.
- Easiest way is to clamp each I to its maximum value

- Another approach is to consider all of a pixel's RGB values together. If at least one is too big, each is divided by the largest to maintain hue and saturation at the expense of the value

Transforming surface normals

- Surface normals are the most important geometric surface characteristic used in computing computer graphics illumination models. The orientation of the surface normal is used in computing both the diffuse and specular components of reflection.
- If polygon normals are precomputed and transformed with the same matrix used for polygon vertices, it is important that nonrigid modelling transformations, such as shears or differential scaling, not be performed. These transformations DO NOT preserve angles and may cause some normals to no longer be perpendicular to their polygons



- For a parametric surface the three-space coordinates are determined by functions of two parameters, s and t in our case.

$$S = \begin{bmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{bmatrix}$$

- Two vectors in the tangent plane can be found by computing partial derivatives:

$$\overline{t_1} = \left[\frac{\partial X(s,t)}{\partial s}, \frac{\partial Y(s,t)}{\partial s}, \frac{\partial Z(s,t)}{\partial s} \right]$$

$$\overline{t_2} = \left[\frac{\partial X(s,t)}{\partial t}, \frac{\partial Y(s,t)}{\partial t}, \frac{\partial Z(s,t)}{\partial t} \right]$$

- Normals are always orthogonal to the tangent space at a point. Thus, given two tangent vectors we can compute the normal as follows:

$$\overline{n} = \overline{t_1} \times \overline{t_2}$$

Where in the rendering pipeline should we perform this calculation?

