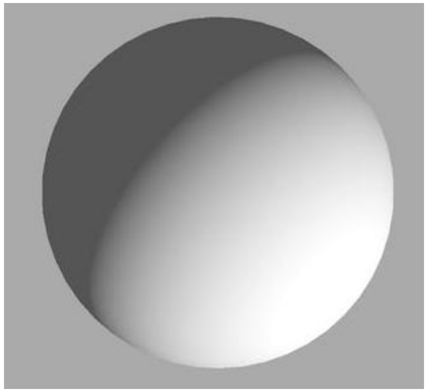
CSE4203: Computer Graphics
Chapter – 10
Surface Shading

Mohammad Imrul Jubair

Shading

- To make objects appear to have more volume, it can help to use *shading*
  - i.e., the surface is "painted" with light.
- This chapter presents the most common heuristic shading methods.



Outline

– Diffuse Shading
– Lambertian Model
– Phong Model

Diffuse Shading (1/2)

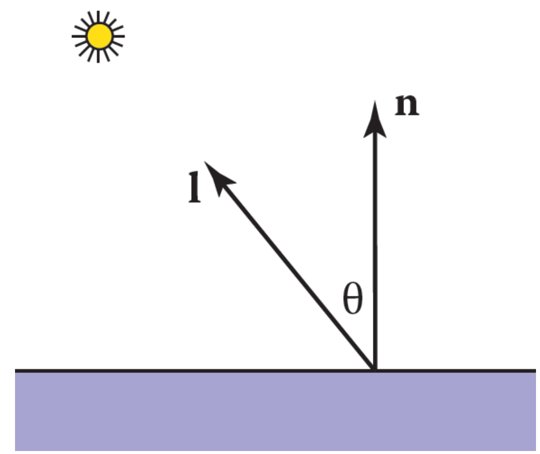
- Many objects in the world have a surface appearance loosely described as “matte,” indicating that the object is not at all shiny.
  - Examples include paper, unfinished wood, and dry, unpolished stones.
- To a large degree, such objects do not have a color change with a change in viewpoint.

Diffuse Shading (2/2)

- For example, if you stare at a particular point on a piece of paper
  - move while keeping your gaze fixed on that point, the color at that point will stay relatively constant.
- Such matte objects can be considered as behaving as *Lambertian* objects.

Lambertian Shading Model (1/10)

- A Lambertian object obeys *Lambert's cosine law.*
  - color *c* of a surface is proportional to the cosine of the angle between the surface normal *(n)* and the direction to the light source *(l)*.

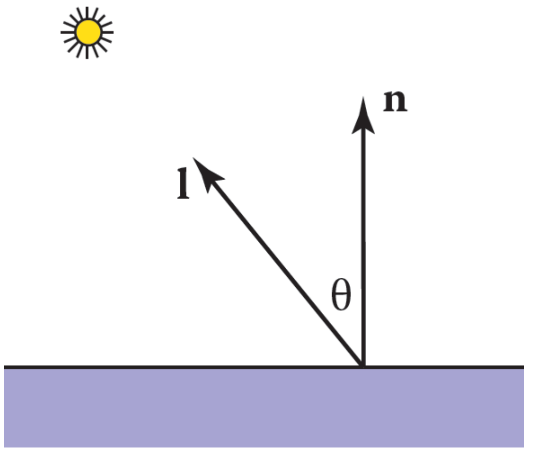


Lambertian Shading Model (2/10)

$$c \propto \cos\theta,$$

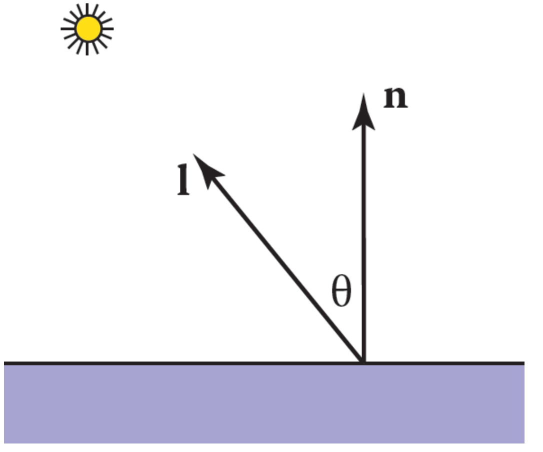
$$c \propto \mathbf{n} \cdot \mathbf{l},$$

Color on the surface will vary according to the cosine of the angle between the surface normal and the light direction



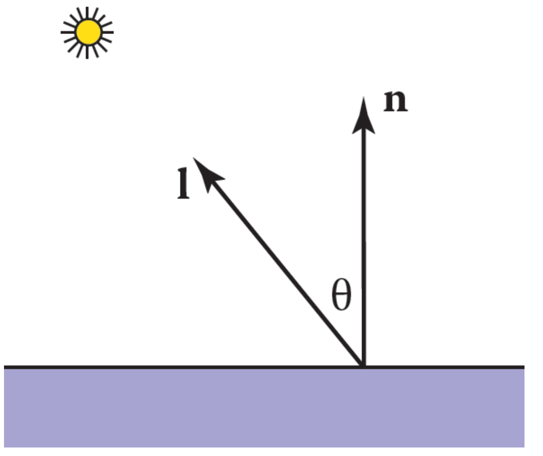
Lambertian Shading Model (3/10)

- Vector **l** is typically assumed not to depend on the location of the object.
  - light is "distant".
- Such a "distant" light is often called a *directional light*
  - because its position is specified only by a direction.



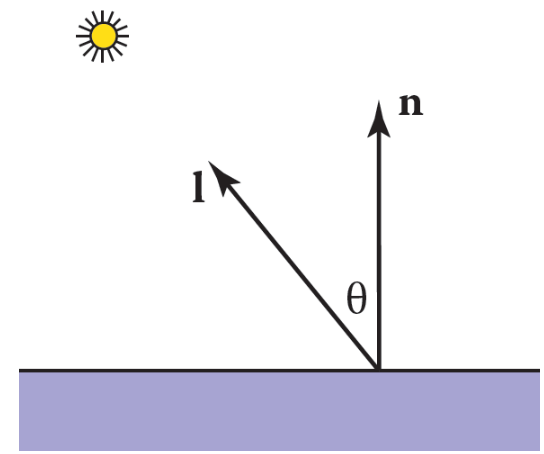
Lambertian Shading Model (4/10)

- A surface can be made lighter or darker by changing the intensity of:
  - the reflectance of the surface.
  - light source



Lambertian Shading Model (5/10)

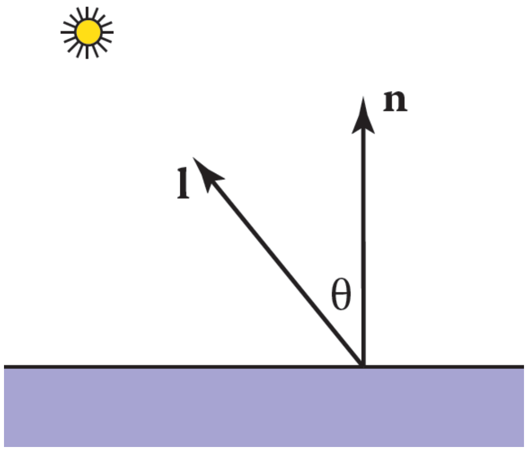
- Diffuse reflectance:
  - c_r is the fraction of light reflected by the surface.
  - will be different for different color components.
    - For example, a surface is red if it reflects a higher fraction of red incident light.



Lambertian Shading Model (6/10)

- Diffuse reflectance:
  - an RGB color
  - The diffuse reflectance c_r must also be included:

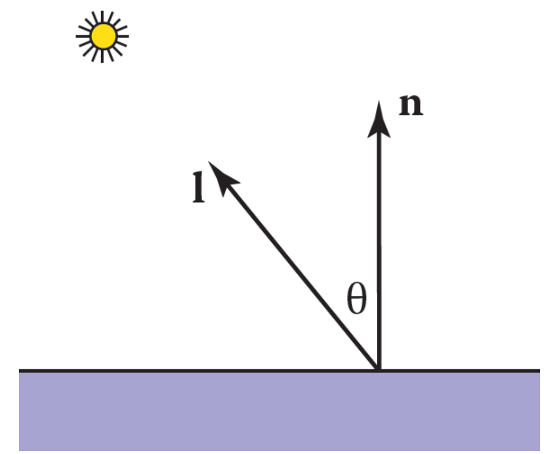
$$c \propto c_r \mathbf{n} \cdot \mathbf{l}.$$



Lambertian Shading Model (7/10)

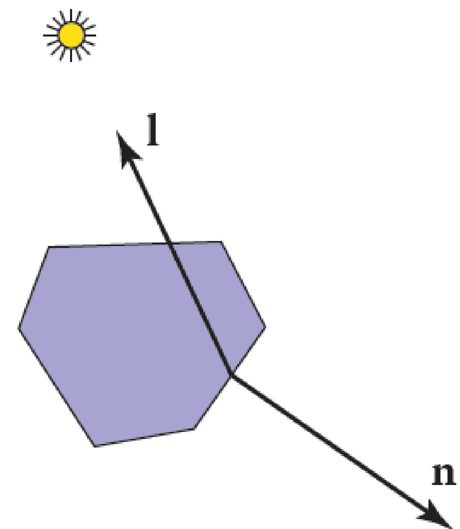
- Light intensity:
  - an RGB color

$$c = c_r c_l \mathbf{n} \cdot \mathbf{l}.$$



Lambertian Shading Model (8/10)

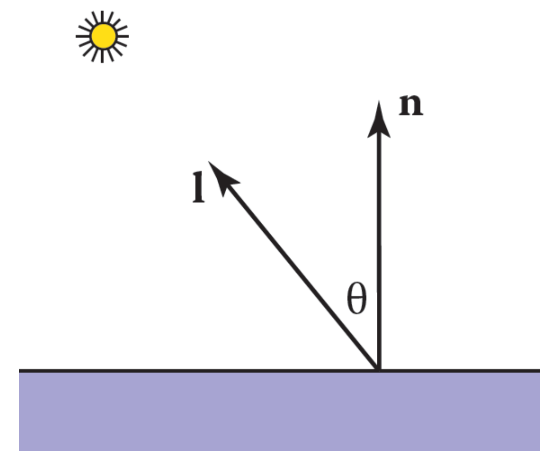
- Light intensity:
  - an RGB color
- it can produce RGB components for *c* that are outside the range [0, 1]
  - because the dot product can be negative.



Lambertian Shading Model (9/10)

- Light intensity:
  - an RGB color

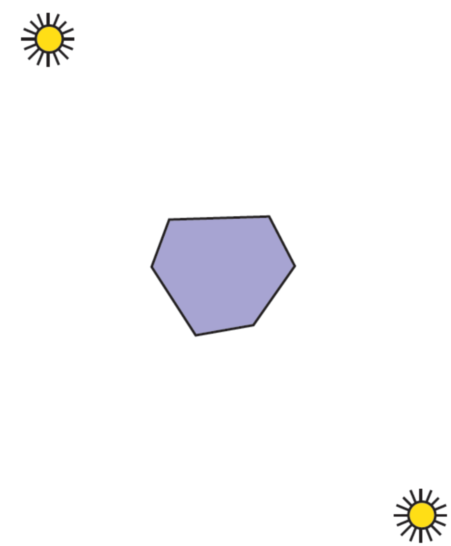
$$c = c_r c_l \max(0, \mathbf{n} \cdot \mathbf{l})$$



Lambertian Shading Model (10/10)

- Another way to deal with the "negative" light is to use an absolute value:

$$c = c_r c_l |\mathbf{n} \cdot \mathbf{l}|$$



- may seem physically implausible
  - it actually corresponds with two lights in opposite directions.

$$c = c_r c_l \max(0, \mathbf{n} \cdot \mathbf{l})$$

  - For this reason it is often called *two-sided* lighting.

Disadvantages of Diffuse Shading (1/2)

- One problem with the diffuse shading:
  - any point whose normal faces away from the light will be black.
- In real life, light is reflected all over, and some light is incident from every direction.

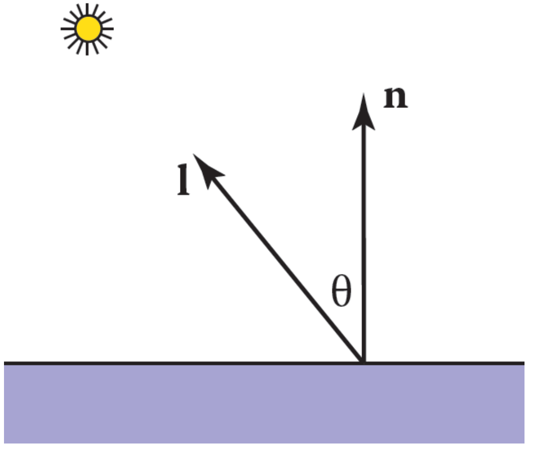
Disadvantages of Diffuse Shading (2/2)

- One way to handle this:
  1. Use several light sources.
  2. Always put a dim source at the eye so that all visible points will receive some light.
  3. Use two-sided lighting

Ambient Shading (1/2)

- A more common approach is to add an ambient term.

$$c = c_r\left(c_a + c_l \max\left(0, \mathbf{n}\cdot\mathbf{l}\right)\right)$$



Ambient Shading (2/2)

- A more common approach is to add an ambient term.

$$c = c_r \left(c_a + c_l \max\left(0, \mathbf{n} \cdot \mathbf{l}\right)\right)$$

- If you want to ensure that the computed RGB color stays in the range $[0, 1]^3$

$$c_a + c_l \leq (1, 1, 1)$$

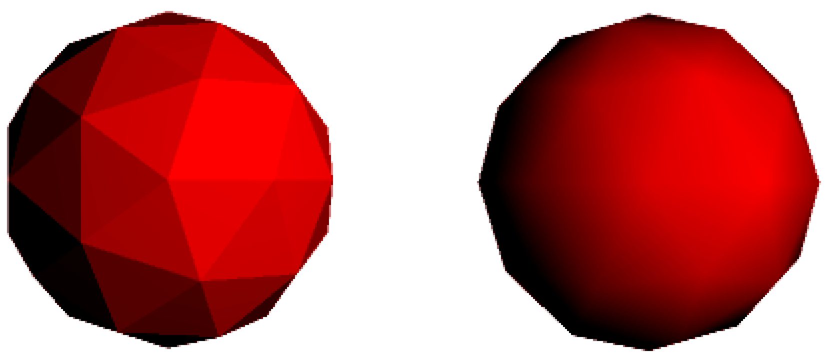
- Otherwise your code should "clamp" RGB values above one to have the value one.

Vertex-Based Diffuse Shading (1/5)

If we apply equation $c = c_r \left(c_a + c_l \max \left(0, \mathbf{n} \cdot \mathbf{l} \right) \right)$

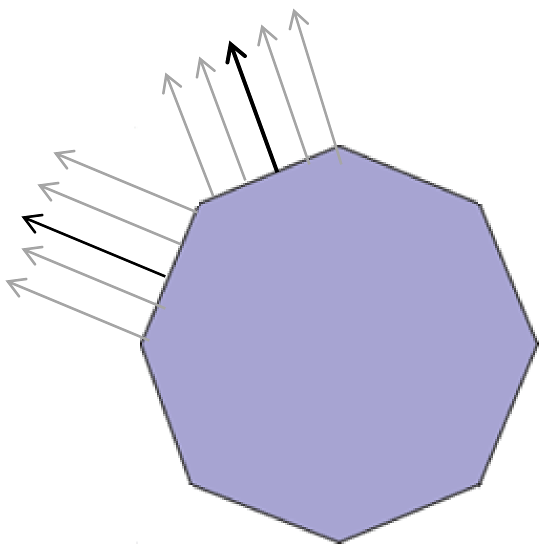
to an object made up of triangles:

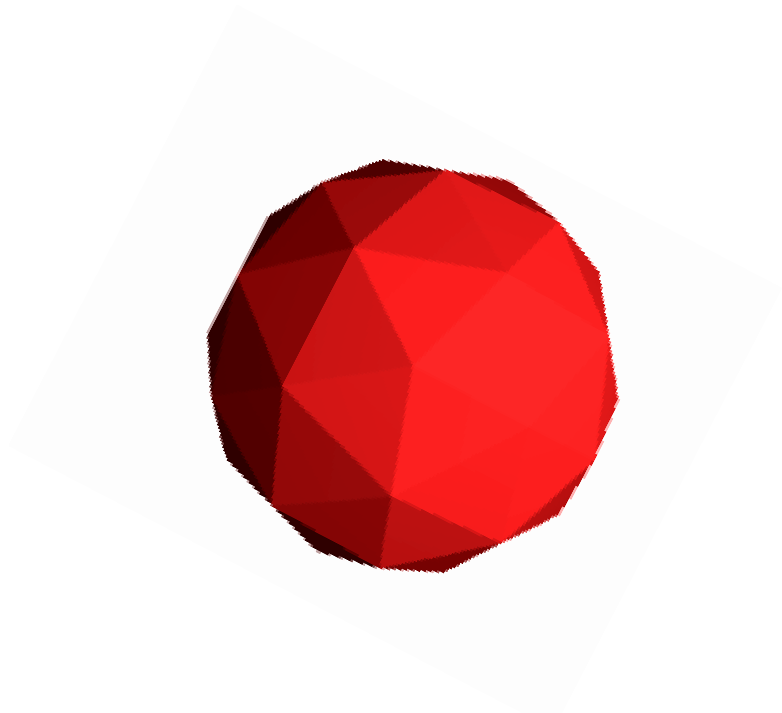
- it will typically have a faceted appearance.



Vertex-Based Diffuse Shading (2/5)

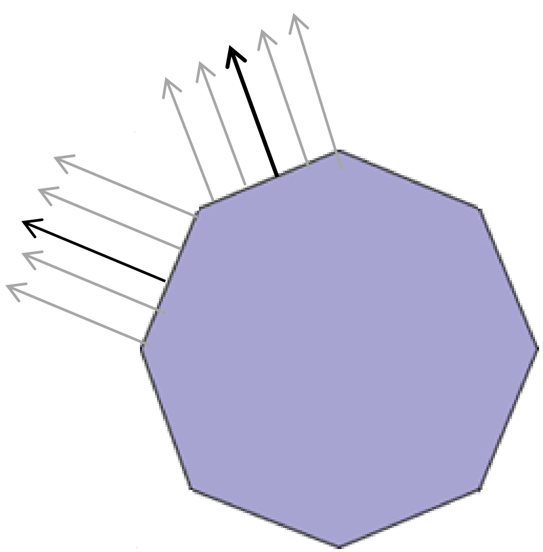
- Drastic changes of normals from surface to surface.

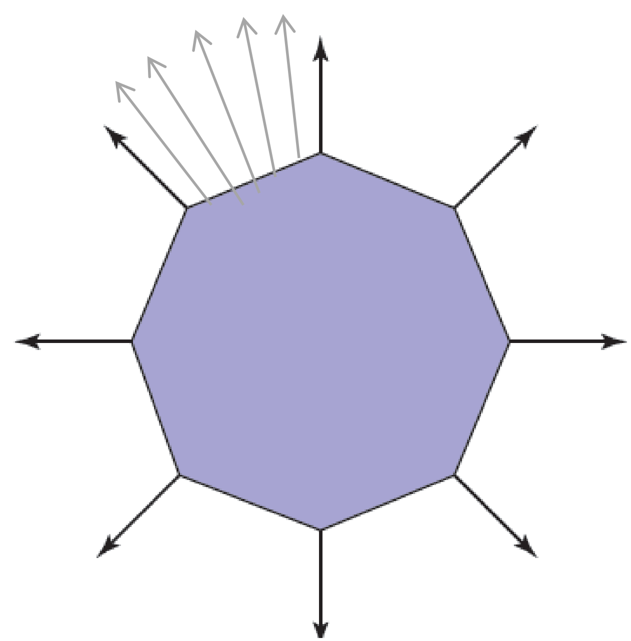




Vertex-Based Diffuse Shading (3/5)

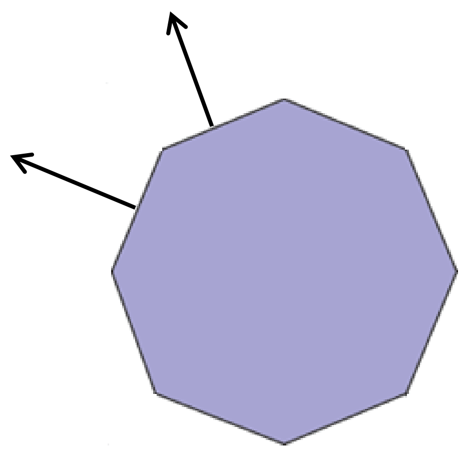
- We can place surface normal vectors at the vertices of the triangles and interpolate.





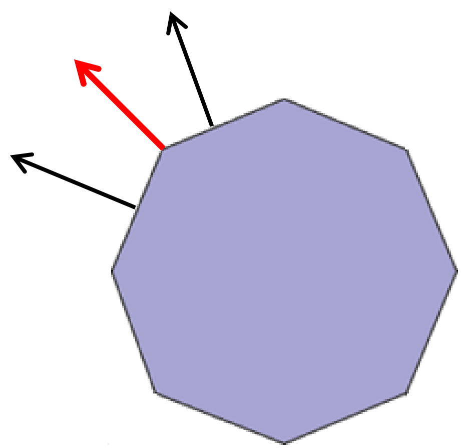
Vertex-Based Diffuse Shading (4/5)

- Problem:
  - Many models will come with normals.
  - compute normals by a variety of heuristic methods.



Vertex-Based Diffuse Shading (5/5)

- Solution:
  - average the normals of the triangles that share each vertex and use this average normal at the vertex.
  - should convert it to a unit vector before using it for shading.

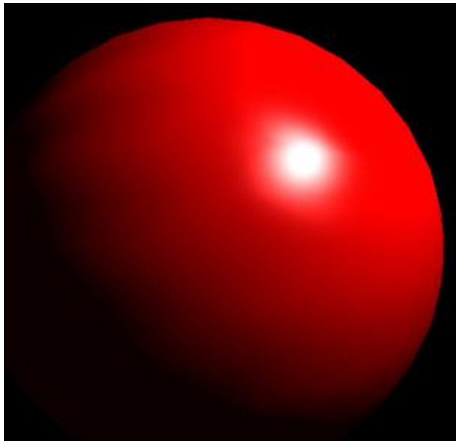


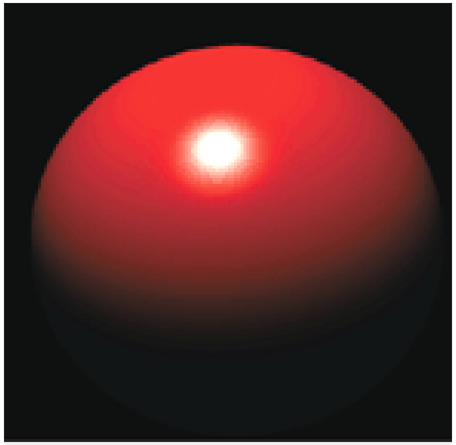
Phong Shading (1/20)

- Some surfaces are essentially like matte surfaces, but they have ***highlights***.
- Examples
  - surfaces include polished tile floors, gloss paint, and whiteboards.

Phong Shading (2/20)

- ***Highlights*** move across a surface as the viewpoint moves.
  - *So, eye matters!*



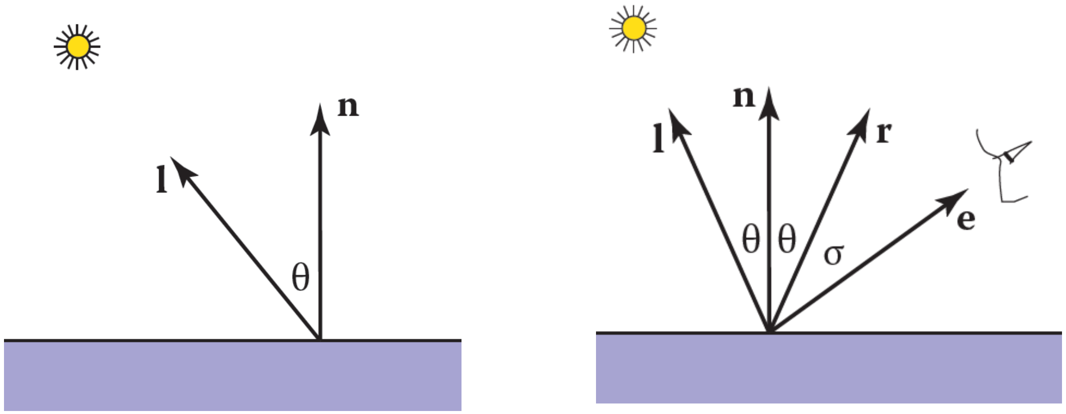


Phong Shading (3/20)

- Some surfaces are essentially like matte surfaces, but they have highlights.

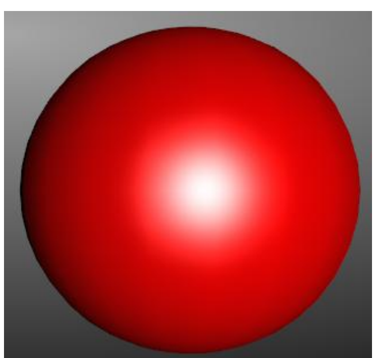
$$c = c_r \left(c_a + c_l \max\left(0, \mathbf{n} \cdot \mathbf{l}\right)\right)$$

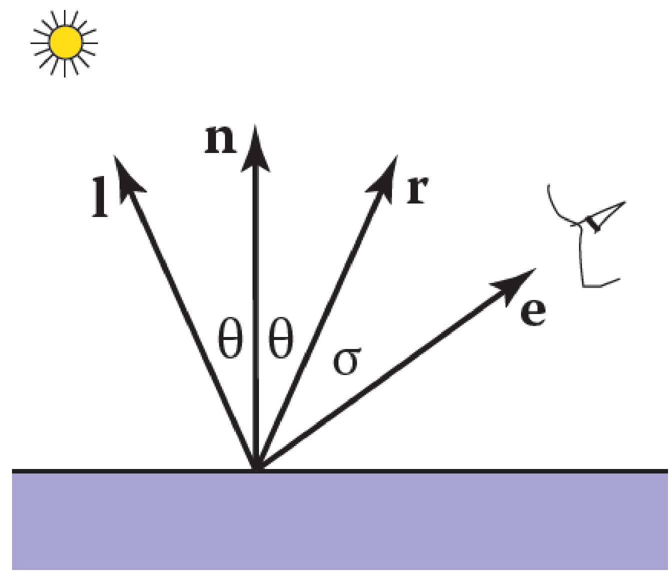
- This means that we must **add** a unit vector **e** toward the eye into our equations.



Phong Shading (4/20)

- We want to add a fuzzy “spot” the same color as the light source in the right place.
  - The center of the spot should be drawn where **e** “lines” up with the natural direction of reflection **r**

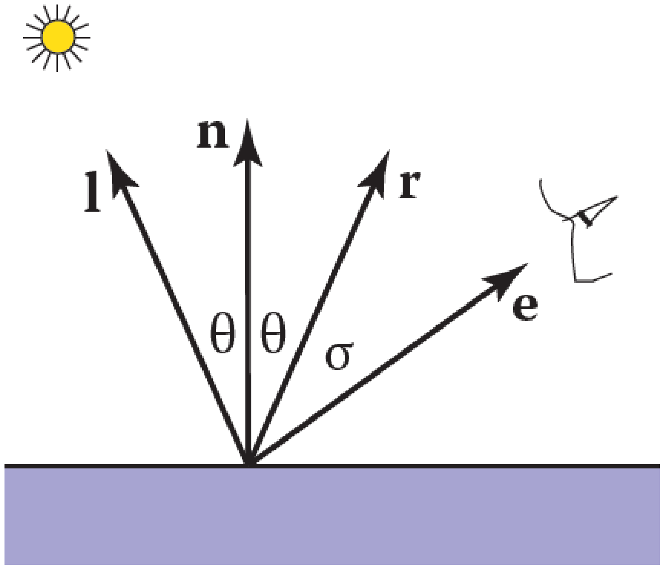


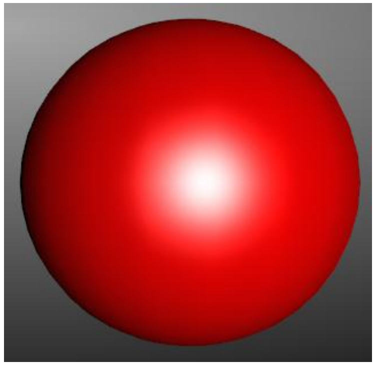


Phong Shading (5/20)

We would like to have the *highlight*

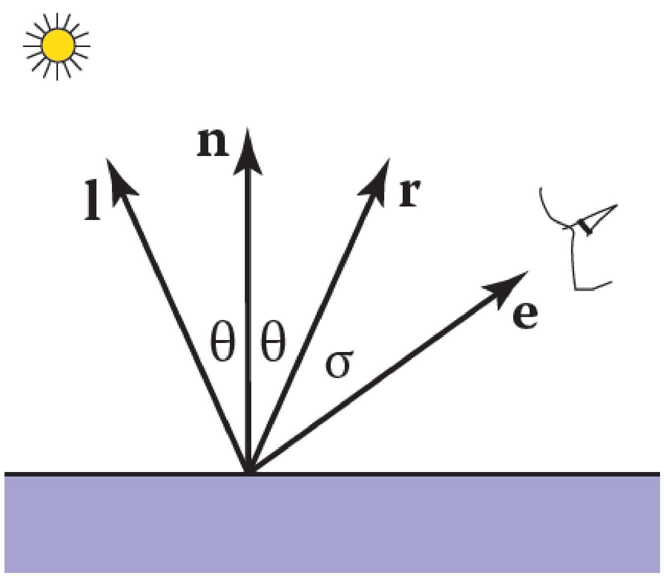
- so that the eye sees some ***highlight*** wherever ***σ* is small**.





Phong Shading (6/20)

- *c* is bright when ***e = r*** and falls off gradually when ***e*** moves away from ***r*.**



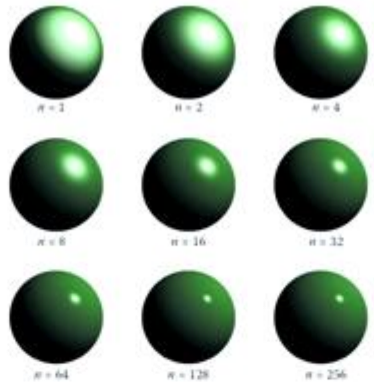
Highlight c :

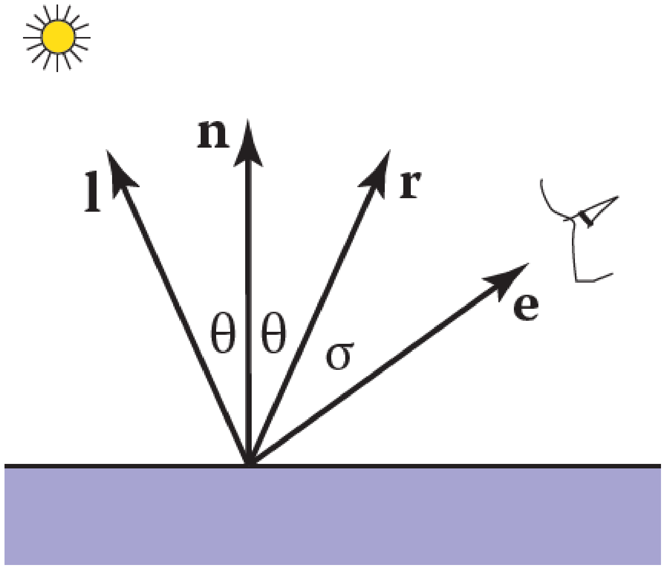
$$c = c_l(\mathbf{e} \cdot \mathbf{r})$$

$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

Phong Shading (7/20)

– Here p is called the *Phong exponent*; it is a positive real number.





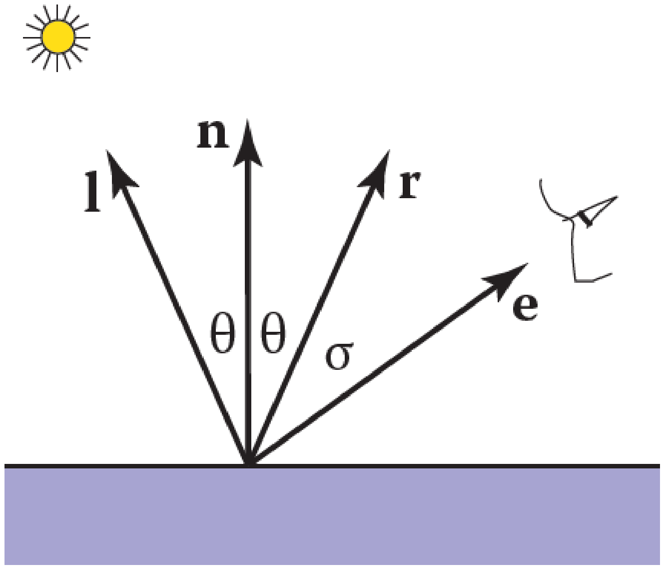
Highlight c :

$$c = c_l(\mathbf{e} \cdot \mathbf{r})$$

$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

Phong Shading (8/20)

Q: Which values are given?

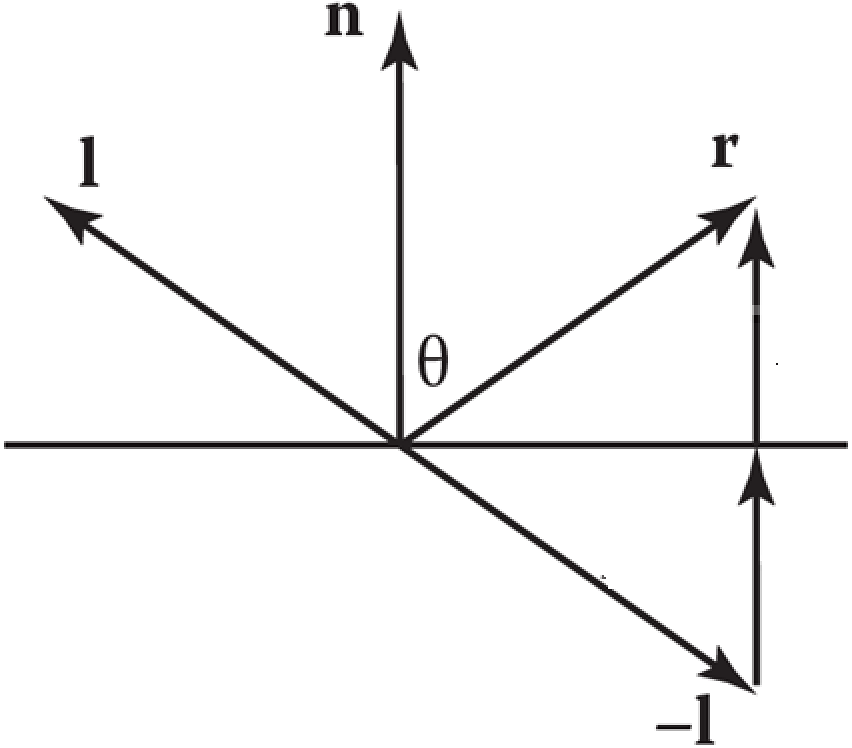


Highlight c :

$$c = c_l(\mathbf{e} \cdot \mathbf{r})$$

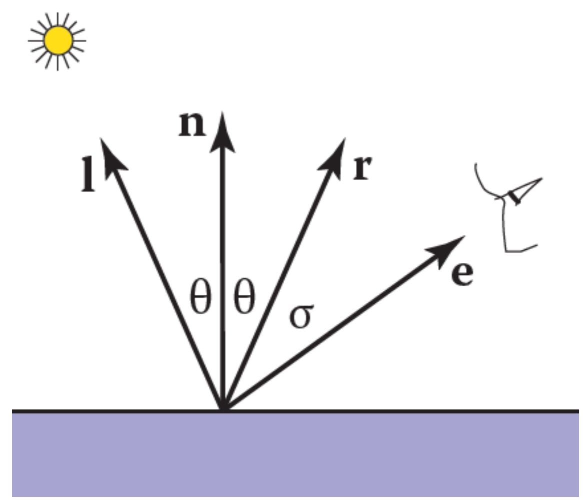
$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

Phong Shading (9/20)

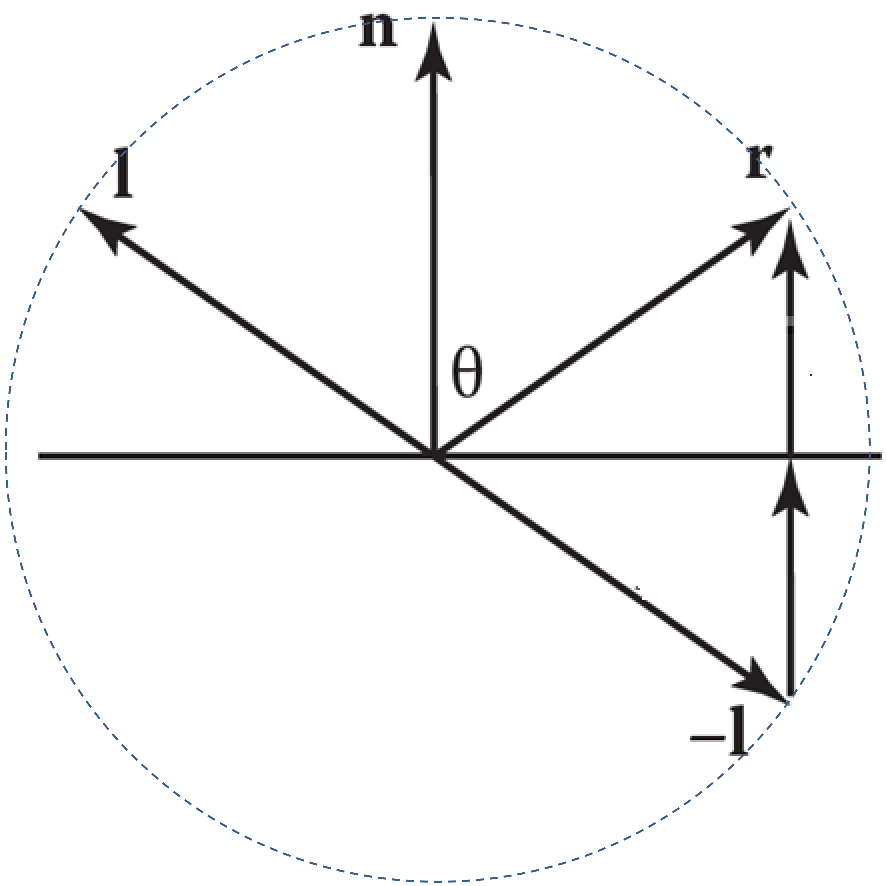


$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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

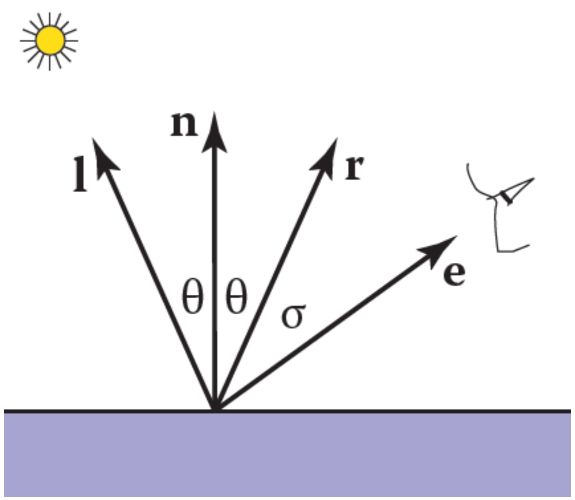


Phong Shading (10/20)

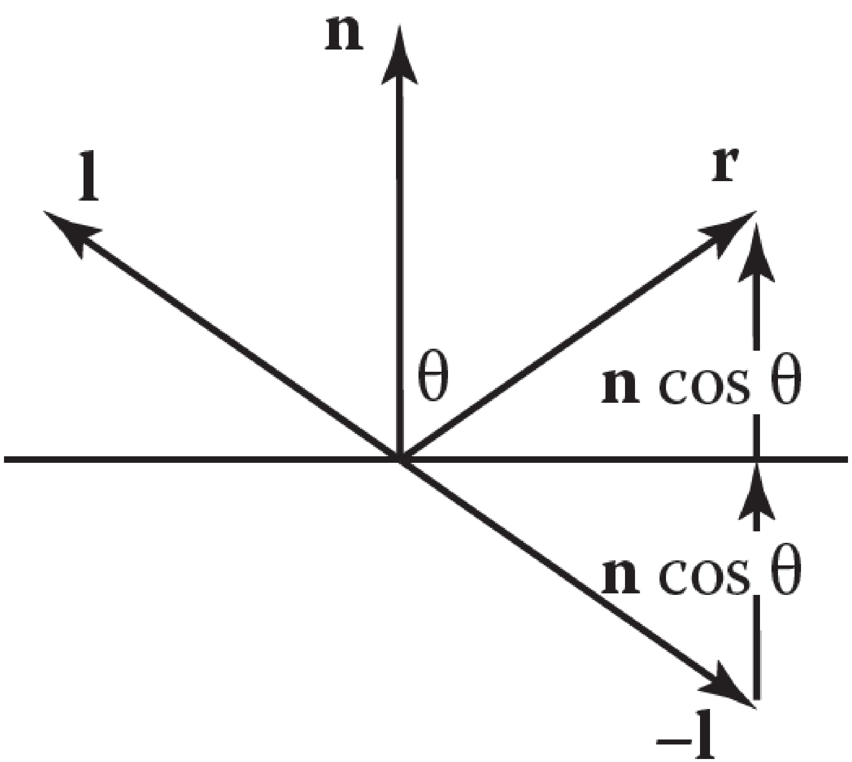


$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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

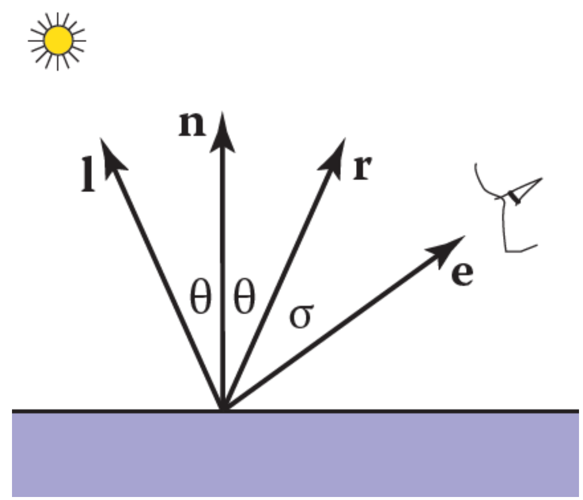


Phong Shading (11/20)



$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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

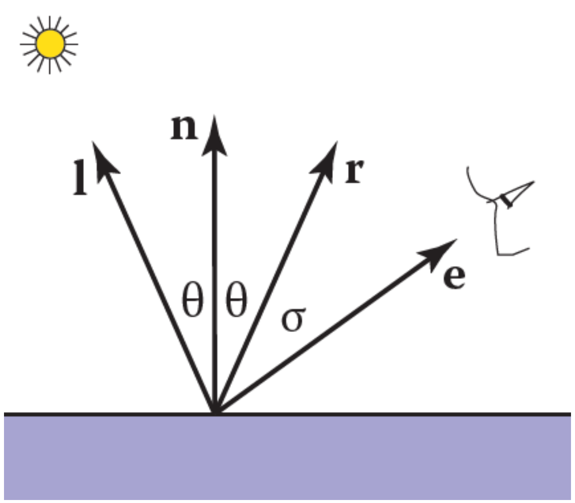


Phong Shading (12/20)

- Therefore, basic approach for *highlight:*

$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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



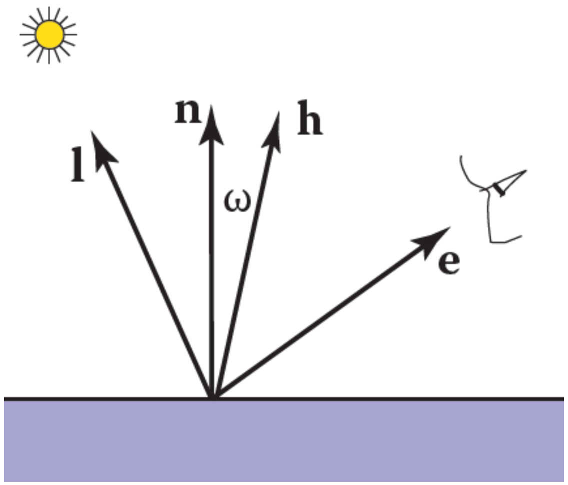
Phong Shading (13/20)

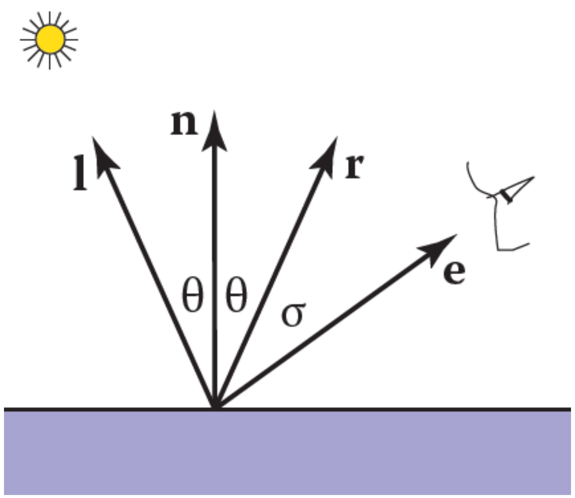
<u>Alternative Approach:</u>

[**h** = unit vector halfway between **l** and **e**]

- The highlight occurs when **h** is near **n**, i.e., when **cos(*ω*) = *h* · *n*** is near ***1***
  - Angle between n and h is near to zero

$$c = c_l(\mathbf{h} \cdot \mathbf{n})^p$$





Phong Shading (14/20)

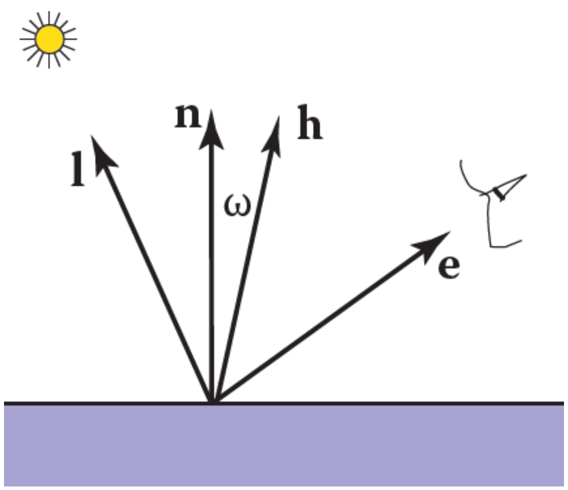
<u>Alternative Approach:</u>

[**h** = unit vector halfway between **l** and **e**]

- The highlight occurs when **h** is near **n**, i.e., when ***cos(ω) = h · n*** is near ***1***
  - Angle between n and h is near to zero

$$c = c_l(\mathbf{h} \cdot \mathbf{n})^p$$

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



Phong Shading (15/20)

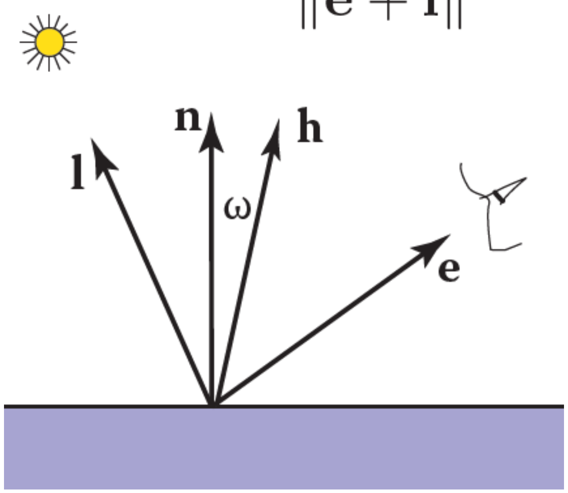
- **angle (*h, n*)** is half of the angle between **angle (*e, r*)**
  - so the details will be slightly different.

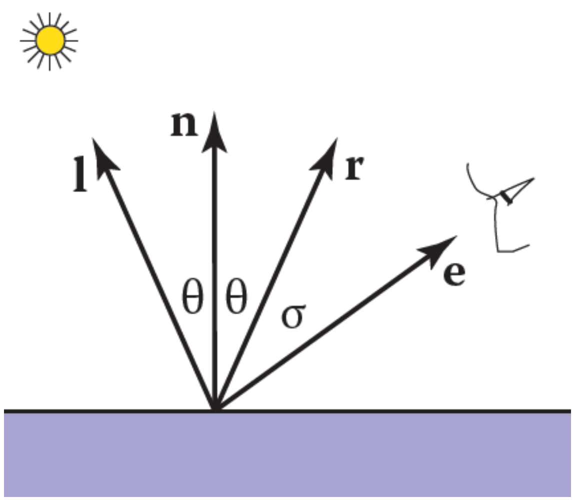
$$c = c_l(\mathbf{h} \cdot \mathbf{n})^p$$

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

$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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



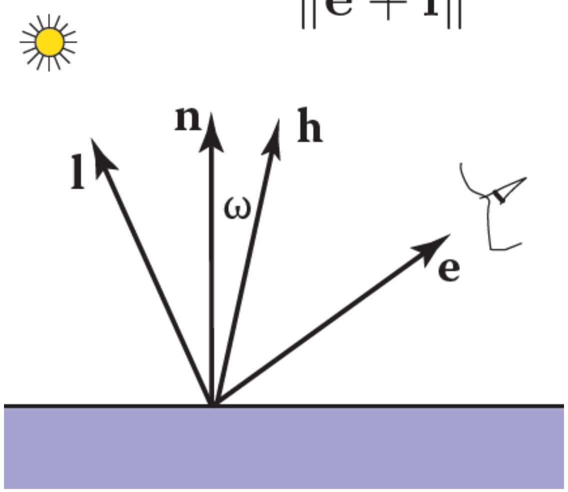


Phong Shading (16/20)

- Cosine between ***n*** and ***h*** is positive for eye and light above the plane.

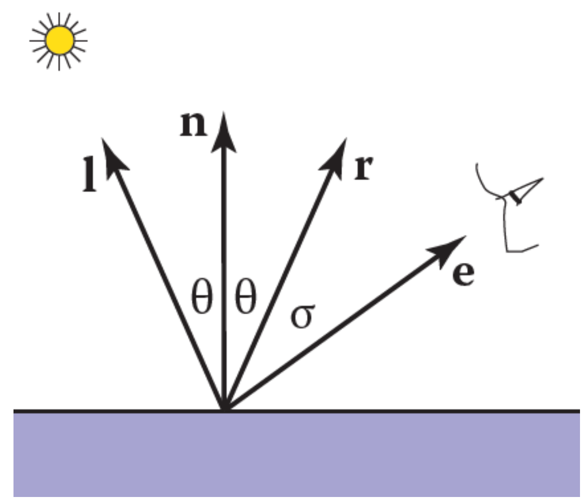
$$c = c_l(\mathbf{h} \cdot \mathbf{n})^p$$

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



$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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

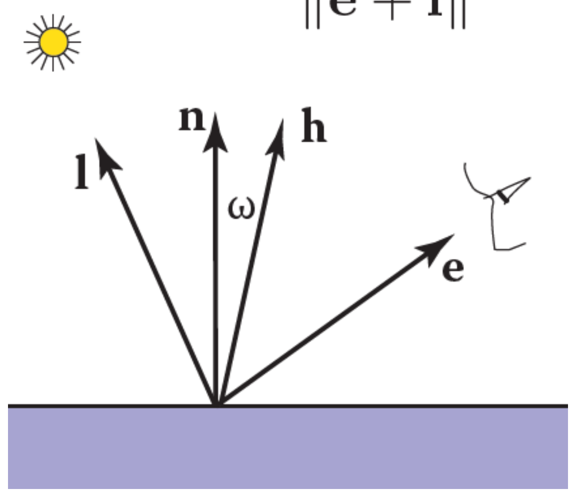


Phong Shading (17/20)

- Square root and divide is needed to compute ***h***.

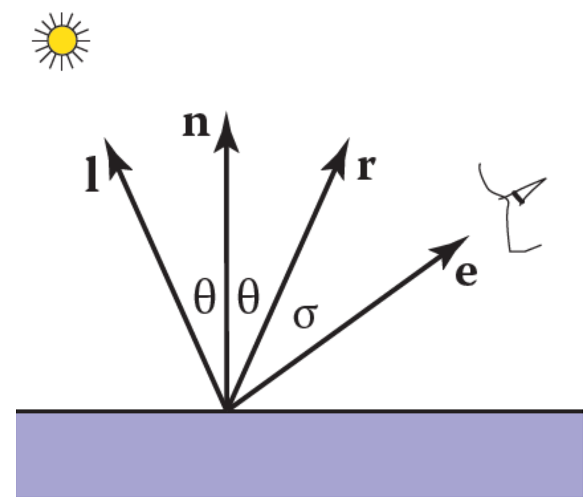
$$c = c_l(\mathbf{h} \cdot \mathbf{n})^p$$

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



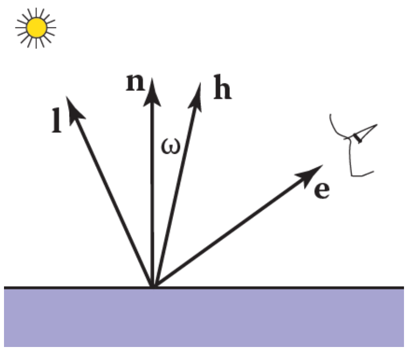
$$c = c_l \max(0, \mathbf{e} \cdot \mathbf{r})^p$$

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



Phong Shading (18/20)

- In practice, we want most materials to have a diffuse appearance in addition to a ***highlight***.

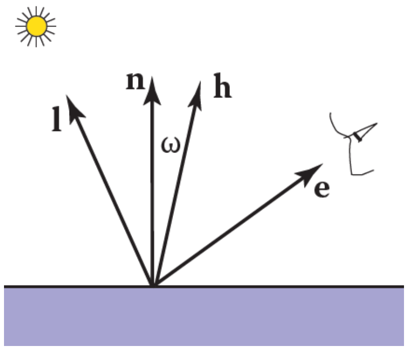


shade = ambient + diffuse

$$c = c_r \left(c_a + c_l \max\left(0, \mathbf{n} \cdot \mathbf{l}\right)\right)$$

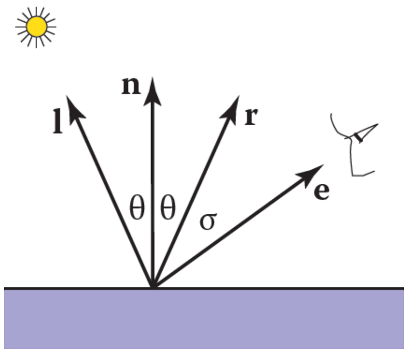
Phong Shading (19/20)

- In practice, we want most materials to have a diffuse appearance in addition to a ***highlight***.



shade = ambient + diffuse ***+ highlights***

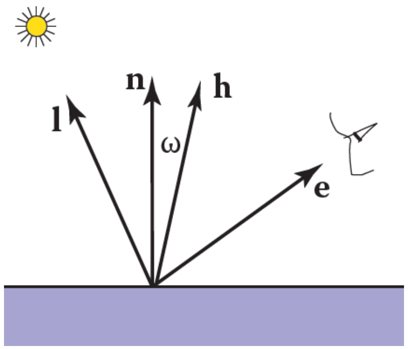
$$c = c_r\left(c_a + c_l \max\left(0, \mathbf{n}\cdot\mathbf{l}\right)\right) + c_l(\mathbf{h}\cdot\mathbf{n})^p$$



$$c = c_r\left(c_a + c_l \max\left(0, \mathbf{n}\cdot\mathbf{l}\right)\right) + c_l \max(0, \mathbf{e}\cdot\mathbf{r})^p$$

Phong Shading (20/20)

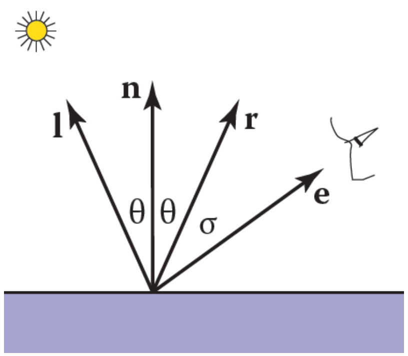
- If we want to allow the user to dim the highlight, we can add a control term $\boldsymbol{c_p}$:



shade = ambient + diffuse ***+ highlights***

$$c = c_r\,(c_a + c_l \max\,(0, \mathbf{n}\cdot\mathbf{l})) + c_l(\mathbf{h}\cdot\mathbf{n})^p$$

$$c = c_r\,(c_a + c_l \max\,(0, \mathbf{n}\cdot\mathbf{l})) + c_l c_p(\mathbf{h}\cdot\mathbf{n})^p$$



$$c = c_r\,(c_a + c_l \max\,(0, \mathbf{n}\cdot\mathbf{l})) + c_l \max(0, \mathbf{e}\cdot\mathbf{r})^p$$

$$c = c_r\,(c_a + c_l \max\,(0, \mathbf{n}\cdot\mathbf{l})) + c_l c_p \max(0, \mathbf{e}\cdot\mathbf{r})^p$$

Additional Reading

- 10.2.2: Surface Normal Vector Interpolation
- 10.3.2: Cool-to-Warm Shading

Exercise

- Textbook ex. 1 and 2
  - Answer hint: *https://www.quora.com/The-color-velvet-and-the-moon-are-poorly-approximated-by-diffuse-or-Phong-shading-Why*