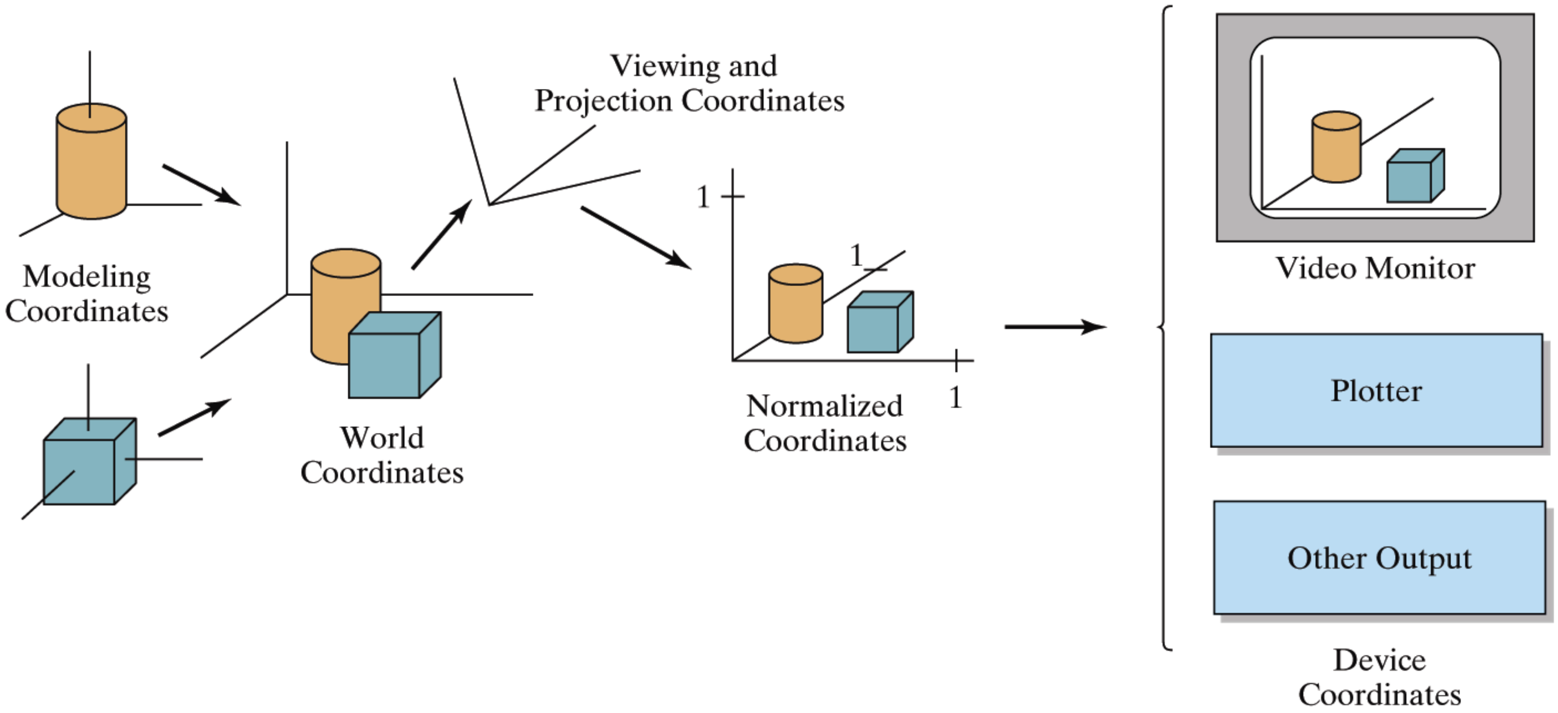
Projection Matrix and its implementation

Review : OpenGL Geometric Transformations

- `glMatrixMode(GL_MODELVIEW);`

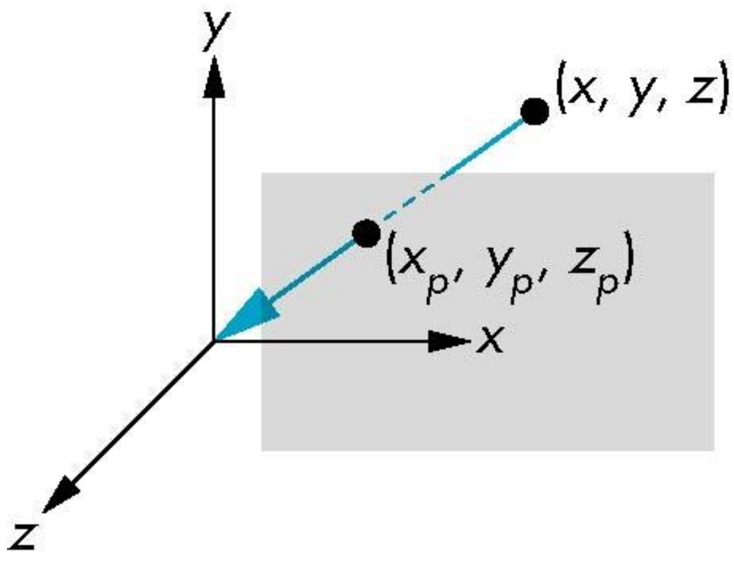


Review: Projection Transformation

- Simple Parallel Projections
- Simple Perspective Projections

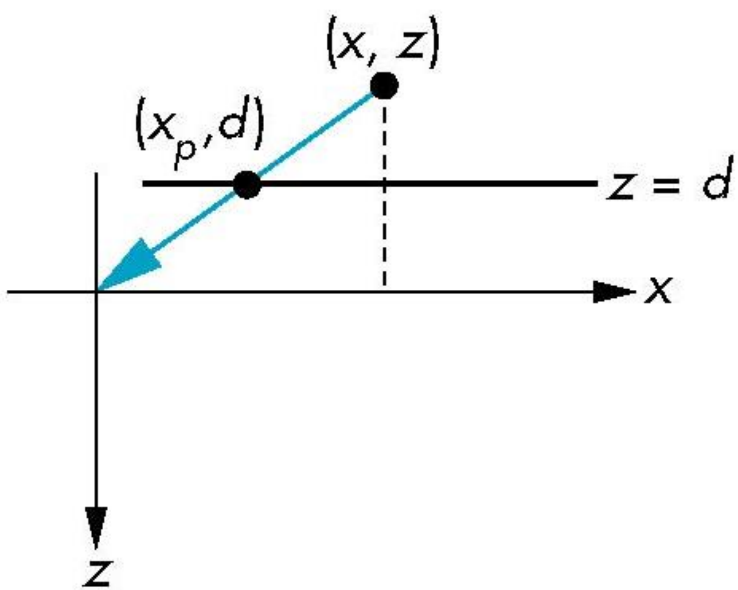
Simple Perspective

- Center of projection at the origin
- Projection plane $z = d$, $d < 0$

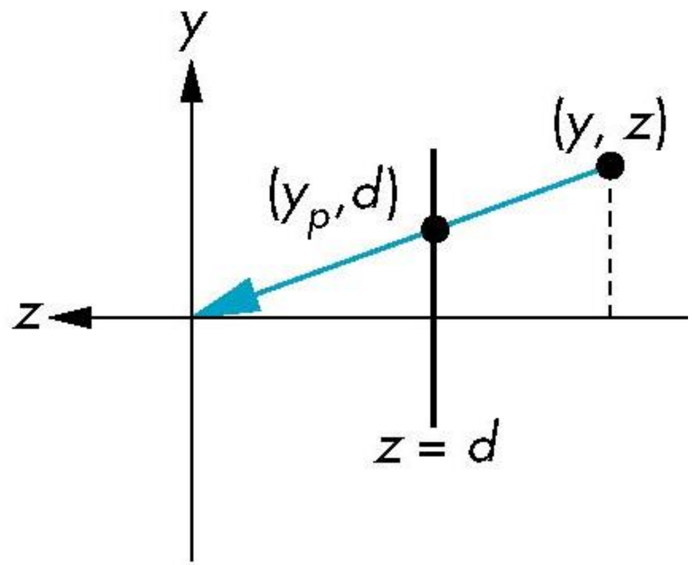


Perspective Equations

Consider top and side views



$$\frac{x}{z/d}$$



$$x_{\mathrm{p}} = \frac{x}{z/d} \qquad y_{\mathrm{p}} = \frac{y}{z/d} \qquad z_{\mathrm{p}} = d$$

Homogeneous Coordinate Form

consider $\mathbf{q} = \mathbf{Mp}$ where $\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{q} = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix}$$

Perspective Division

- However $w \neq 1$, so we must divide by w to return from homogeneous coordinates

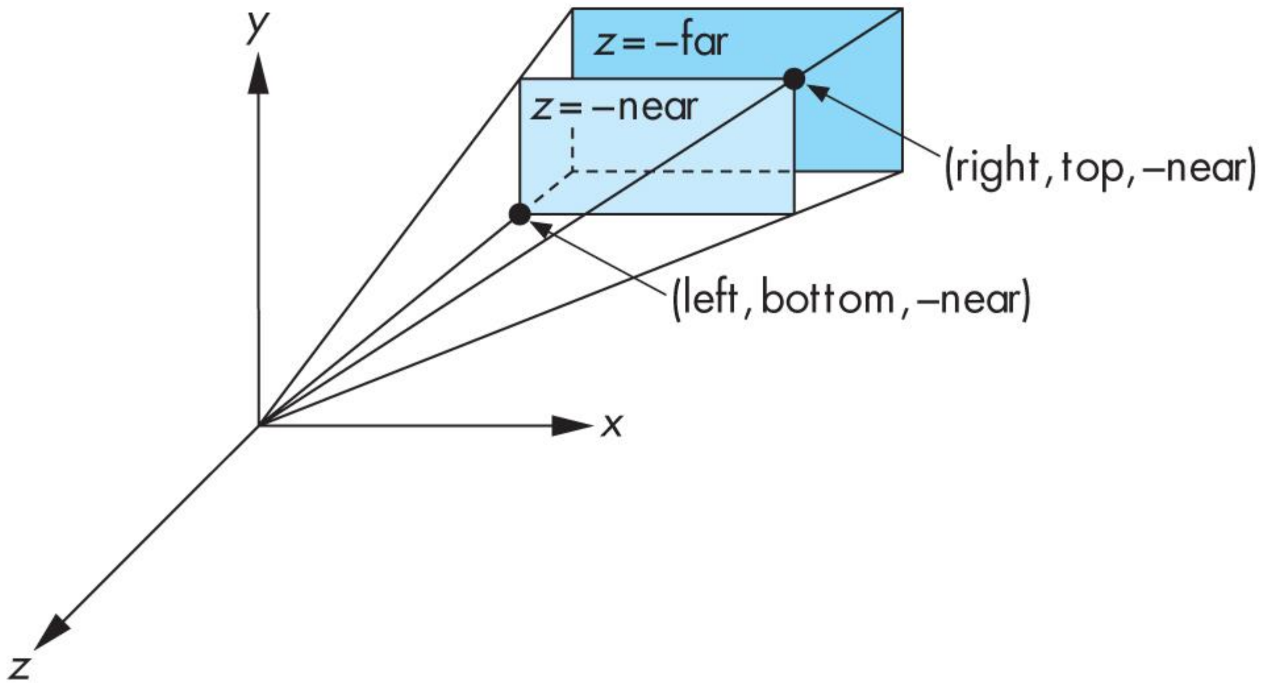
- This *perspective division* yields

$$x_p = \frac{x}{z/d} \qquad y_p = \frac{y}{z/d} \qquad z_p = d$$

the desired perspective equations

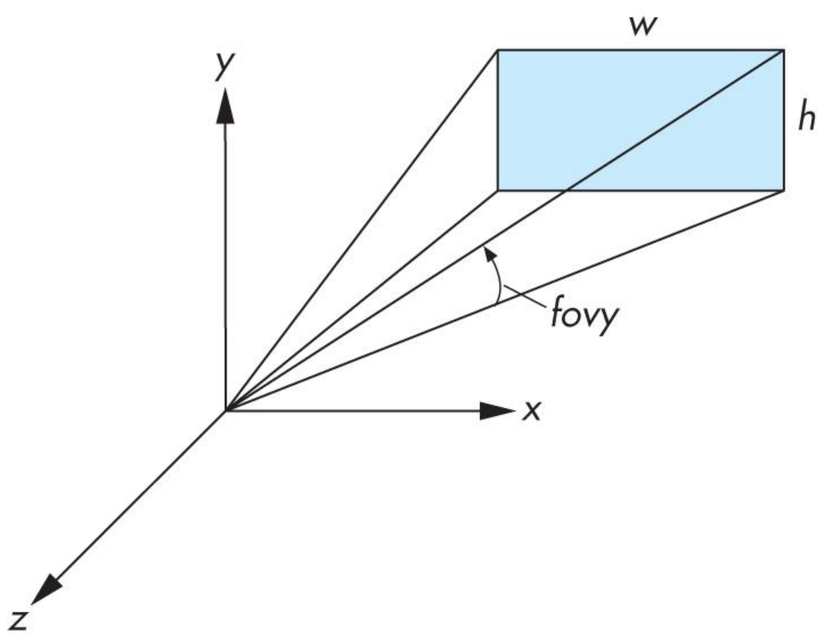
Perspective Viewing in Old OpenGL

- Two interfaces: glFrustum and gluPerspective
- glFrustum(xmin, xmax, ymin, ymax, near, far);



Field of View Interface in Old OpenGL

- gluPerspective(fovy, aspectRatio, near, far);
- aspectRatio = w / h
- fovy specifies field of view as height (y) angle

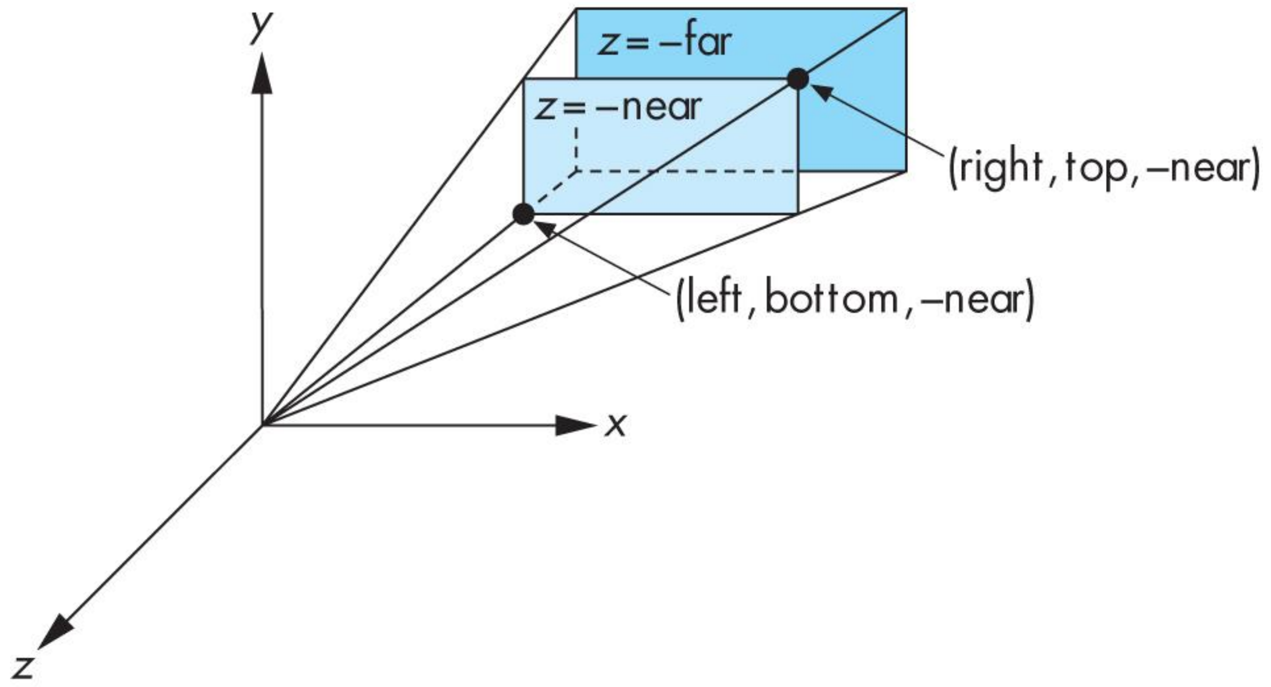


Old OpenGL code

```
void reshape(int x, int y)
{
      glViewport(0, 0, x, y);
      glMatrixMode(GL_PROJECTION);
      glLoadIdentity();
      gluPerspective(60.0, x/float(y), 0.01, 10.0);
}
```

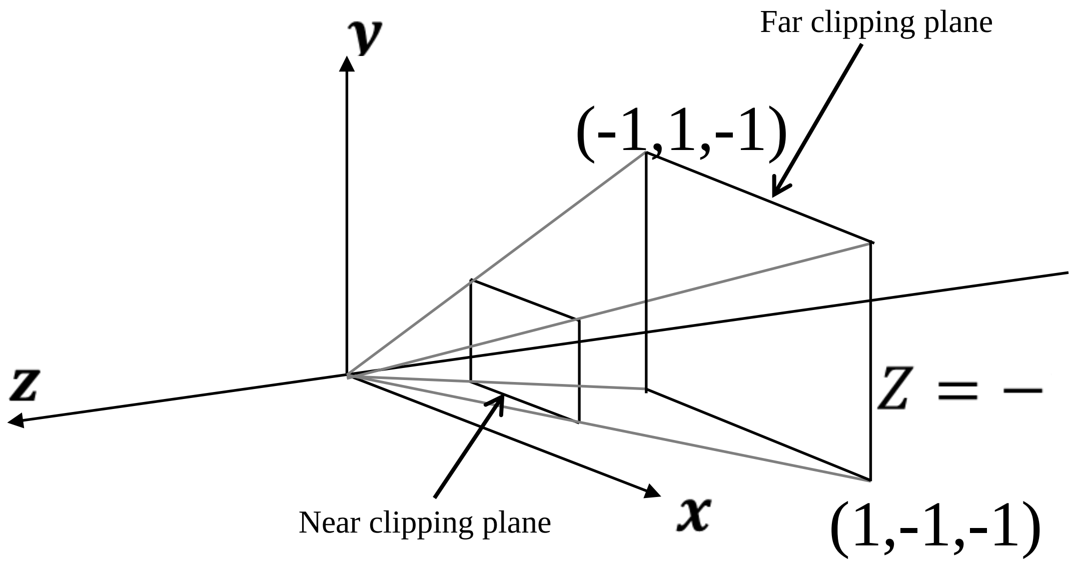
Implementing your own Frustum Function

- `glFrustum(xmin,xmax, ymin,ymax, near,far);`
- `gluPerspective(fovy,aspectRatio, near,far);`



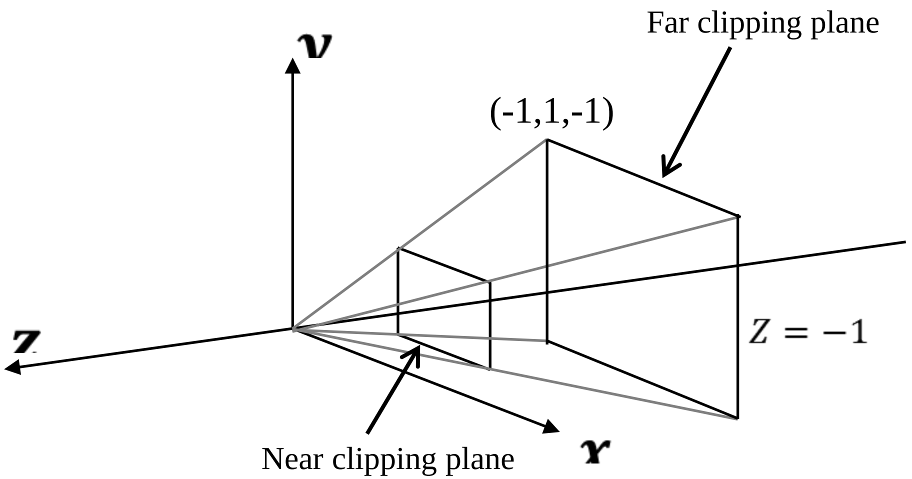
The Perspective View Volume

- Canonical view volume (frustum):



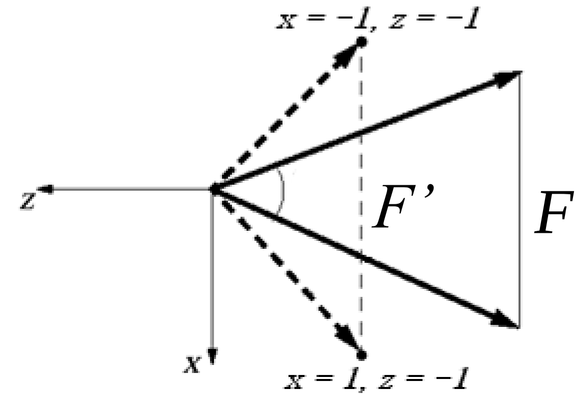
Properties of the canonical view volume

- Sits at origin:
  - Camera position = (0,0,0)
- Looks along negative *z*-axis:
  - Look Vector = (0,0,-1)
- Oriented upright:
  - Up Vector = (0,1,0)
- Near and far clipping planes:
  - Near plane at *z* =$c = -\frac{near}{far}$ (will prove this)
  - Far plane at *z* = -1
- Far clipping plane bounds:
  - (*x*, *y*) from -1 to 1

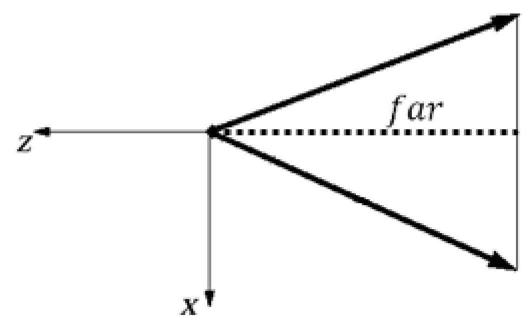


- Note: *The perspective canonical view volume is just like the parallel one except that the "film"/viewing window is more ambiguous here, so we bound just the far clipping plane for now*

Scaling the perspective view volume (1/4)



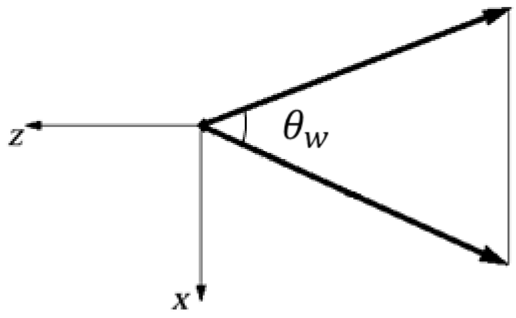
- Top-down view of the perspective view volume:
- Goal: scale the original volume so the solid arrows are transformed to the dotted arrows
  - Equivalently: Scale the original (solid) far plane cross-section F so it lines up with the canonical (dotted) far plane cross-section F'
- First, scale along Z direction
  - Want to scale so far plane lies at $z = -1$
  - Far plane originally lies at $z = -far$
  - Divide by far, since $\frac{-far}{far} = -1$
  - So $Scale_Z = \frac{1}{far}$

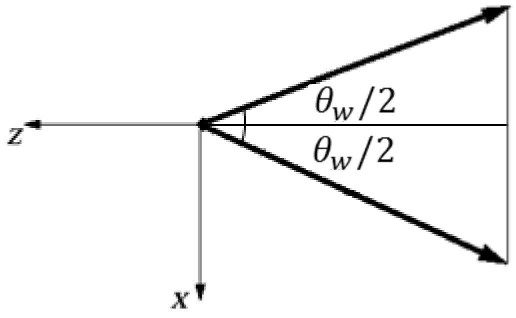


Scaling the perspective view volume (2/4)

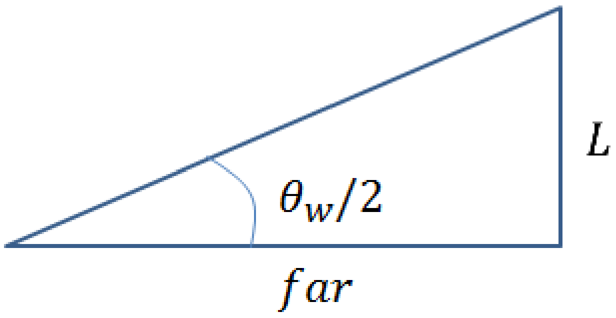
- Next, scale along X direction
  - Use the same trick: divide by size of volume along the X axis

- How long is the side of the volume along X? Find out using trig…
  - Start with the original volume
  - Cut in half along the Z axis





Scaling the perspective view volume (3/4)



- Consider just the top triangle
- Note that L equals the X coordinate of a corner of the perspective view volume's cross-section. Ultimately want to scale by $\frac{1}{L}$ to make $L \rightarrow 1$
- $\frac{L}{far} = \tan(\frac{\theta_w}{2})$ $\rightarrow$ $L = far\tan\left(\frac{\theta_w}{2}\right)$
- Conclude that $Scale_X = \frac{1}{far\tan(\frac{\theta_w}{2})}$

Scaling the perspective view volume (4/4)

- Finally, scale along Y direction
  - Use the same trig as X direction, but use the height angle θ_h instead of θ_w
  - Result: $Scale_Y = \frac{1}{far\ \tan(\frac{\theta_h}{2})}$

- The final result is this scale matrix:

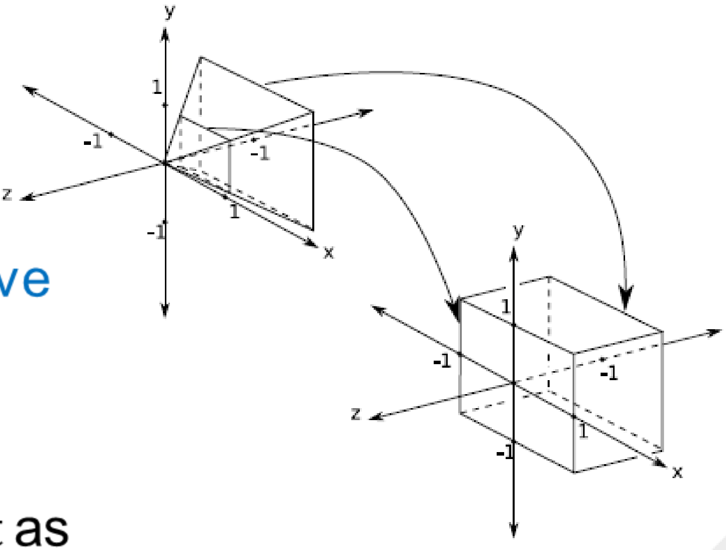
$$S_{xyz} = \begin{bmatrix} \frac{1}{\tan\left(\frac{\theta_w}{2}\right)far} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{\theta_h}{2}\right)far} & 0 & 0 \\ 0 & 0 & \frac{1}{far} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Perspective and Projection

- Now we have our canonical perspective view volume

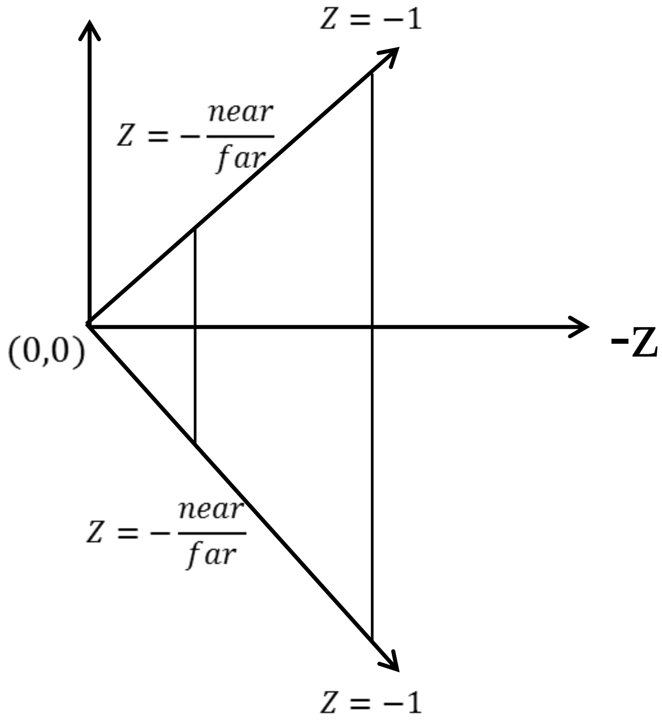
- The final step of our normalizing transformation, transforming the perspective view volume into a parallel one!

- Think of this perspective transformation $\boldsymbol{p}$t as the ***unhinging transformation***, represented by matrix $\boldsymbol{M_{pt}}$



Unhinging View Volume to Become a Parallel View Volume (1/4)

- Near clipping plane at $c = -\frac{near}{far}$ should transform to $z = 0$



Unhinging View Volume to Become a Parallel View Volume(2/4)

- The derivation of our unhinging transformation matrix is complex.
- Instead, we will give you the matrix and show that it works by example
- Our unhinging transformation matrix, $\boldsymbol{M_{pt}}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{c+1} & \frac{-c}{c+1} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Unhinging View Volume to Become a Parallel View Volume(3/4)

- Our perspective transformation does the following:
  - Sends all points on the z = -1 far clipping plane to themselves
    - We'll check top-left (-1, 1, -1, 1) and bottom-right (1, -1, -1, 1) corners
  - Sends all points on the z = c near clipping plane onto the $z = 0$ plane
    - Note that the corners of the cross section of the near clipping plane in the frustum are $(-c, c, c, 1)$, $(c, -c, c, 1), (c, c, c, 1)$ and $(-c, -c, c, 1)$
    - We'll check to see that $(-c, c, c, 1)$ gets sent to (-1, 1, 0, 1) and that $(c, -c, c, 1)$ gets sent to (1, -1, 0, 1)
  - Let's try $c = -\frac{1}{2}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{c+1} & \frac{-c}{c+1} \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Unhinging View Volume to Become a Parallel View Volume(4/4)

$$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 2 & 1\\0 & 0 & -1 & 0\end{bmatrix} \begin{bmatrix}-1\\1\\-1\\1\end{bmatrix} \longrightarrow \begin{bmatrix}-1\\1\\-1\\1\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 2 & 1\\0 & 0 & -1 & 0\end{bmatrix} \begin{bmatrix}1\\-1\\-1\\1\end{bmatrix} \longrightarrow \begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}$$

Don’t forget to homogenize!

$$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 2 & 1\\0 & 0 & -1 & 0\end{bmatrix} \begin{bmatrix}-1/2\\1/2\\-1/2\\1\end{bmatrix} \longrightarrow \begin{bmatrix}-1/2\\1/2\\0\\1/2\end{bmatrix} \longrightarrow \begin{bmatrix}-1\\1\\0\\1\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 2 & 1\\0 & 0 & -1 & 0\end{bmatrix} \begin{bmatrix}1/2\\-1/2\\-1/2\\1\end{bmatrix} \longrightarrow \begin{bmatrix}1/2\\-1/2\\0\\1/2\end{bmatrix} \longrightarrow \begin{bmatrix}1\\-1\\0\\1\end{bmatrix}$$

The normalizing transformation (perspective)

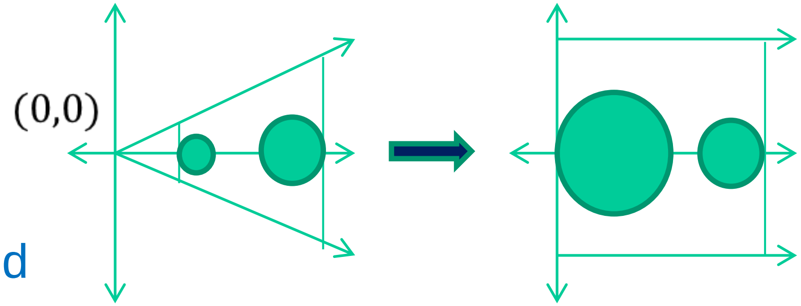
- $\boldsymbol{N_{perspective}} = \boldsymbol{M_{pt} S_{xyz}}$

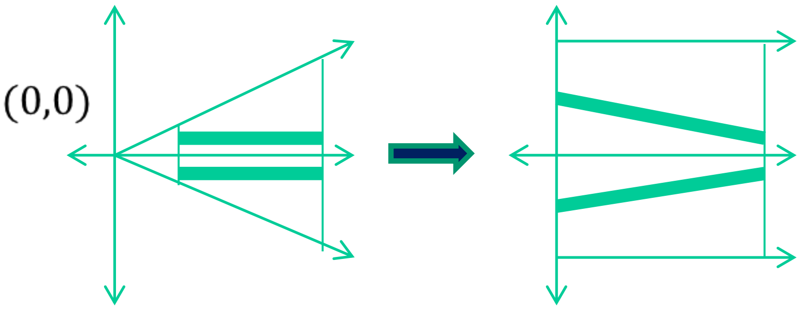
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{c+1} & \frac{-c}{c+1} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\tan\left(\frac{\theta_w}{2}\right) far} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{\theta_h}{2}\right) far} & 0 & 0 \\ 0 & 0 & 1/far & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Remember to homogenize your points after you apply this transformation

Why it works (1/2)

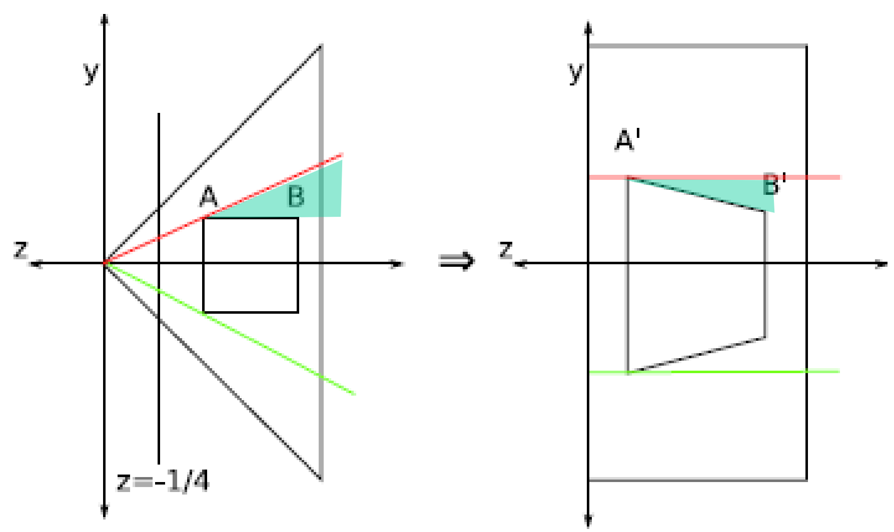
- The key is in the **unhinging step**
- We can take an intuitive approach to see this
  - The closer the object is to the near clipping plane, the more it is enlarged during the unhinging step
  - Thus, closer objects are larger and farther away objects are smaller, as is to be expected
- Another way to see it is to use the parallel lines
  - Draw parallel lines in a perspective volume
  - When we unhinge the volume, the lines fan out at the near clipping
  - The result is converging lines, the railroad track





Why it works (2/2)

- Yet another way to demonstrate how this works is to use occlusion (when elements in the scene are blocked by other elements)
- Looking at the top view of the frustum, we see a square
- Draw a line from your eye point to the left corner of the square, we can see that points behind this corner are obscured
- Now unhinge the perspective and draw a line again to the left corner, we can see that all points obscured before are still obscured and all points that were visible before are still visible



Chapter 5.
Lighting and Shading

Photorealism in Computer Graphics

- Photorealism in computer graphics involves
  - Accurate representations of surface properties, and
  - Good physical descriptions of the lighting effects

- Modeling the lighting effects that we see on an object is a complex process, involving principles of both physics and psychology

- Physical illumination models involve
  - Material properties, object position relative to light sources and other objects, the features of the light sources, and so on

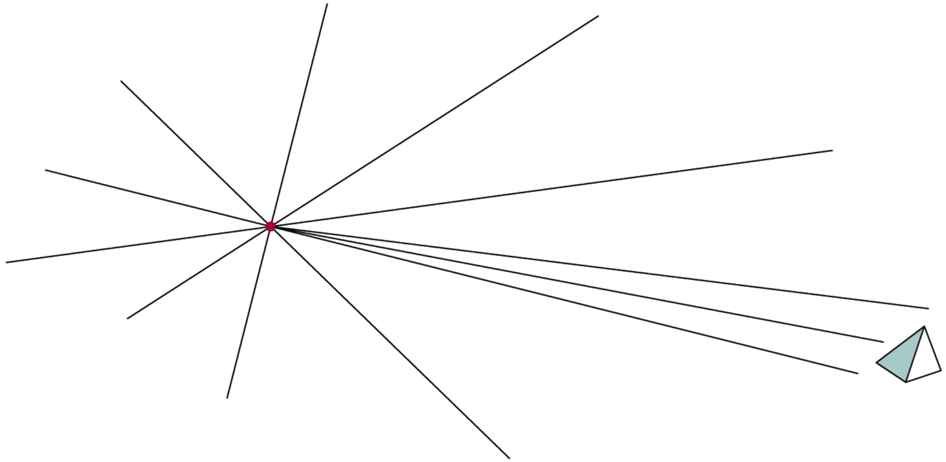
Illumination and Rendering

- An ***illumination model*** in computer graphics
  - also called a ***lighting model*** or a ***shading model***
  - used to calculate the color of an illuminated position on the surface of an object
  - Approximations of the physical laws

- A ***surface-rendering method*** determine the pixel colors for all projected positions in a scene

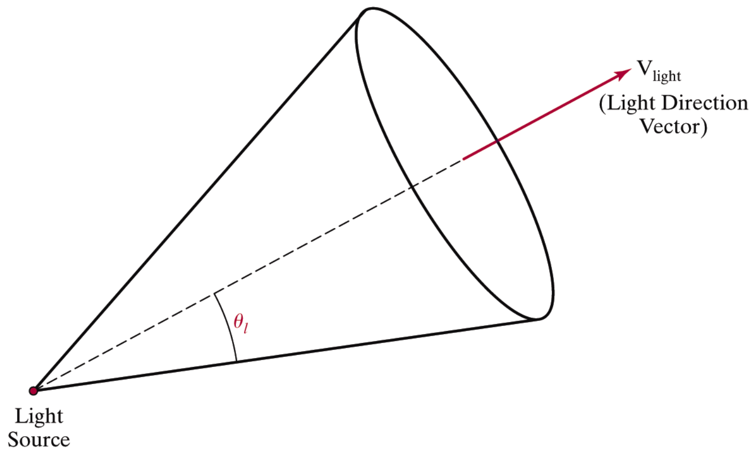
Light Sources

- Point light sources
  - Emitting radiant energy at a single point
  - Specified with its position and the color of the emitted light
- Infinitely distant light sources
  - A large light source, such as sun, that is very far from a scene
  - Little variation in its directional effects
  - Specified with its color value and a fixed direction for the light rays

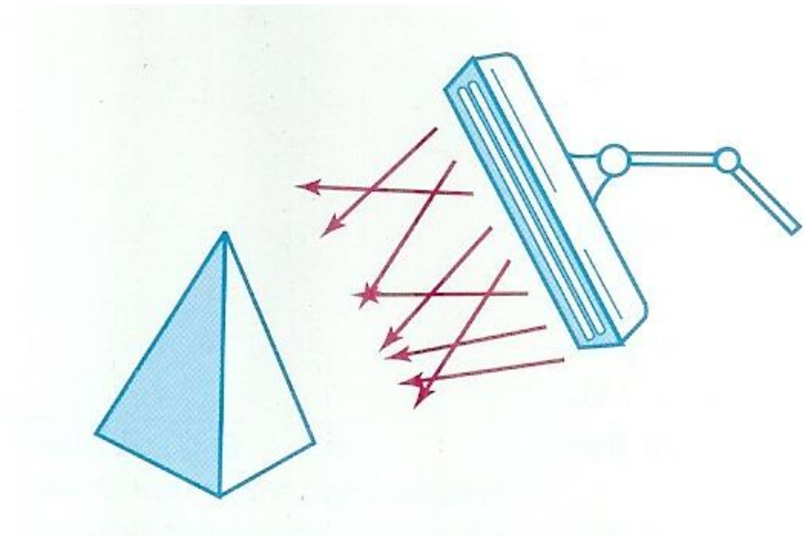


Light Sources

- Directional light sources
  - Produces a directional beam of light
  - Spotlight effects



- Area light sources



Light Sources

- Radial intensity attenuation
  - As radiant energy travels, its amplitude is attenuated by the factor $1/d^2$
  - Sometimes, more realistic attenuation effects can be obtained with an inverse quadratic function of distance

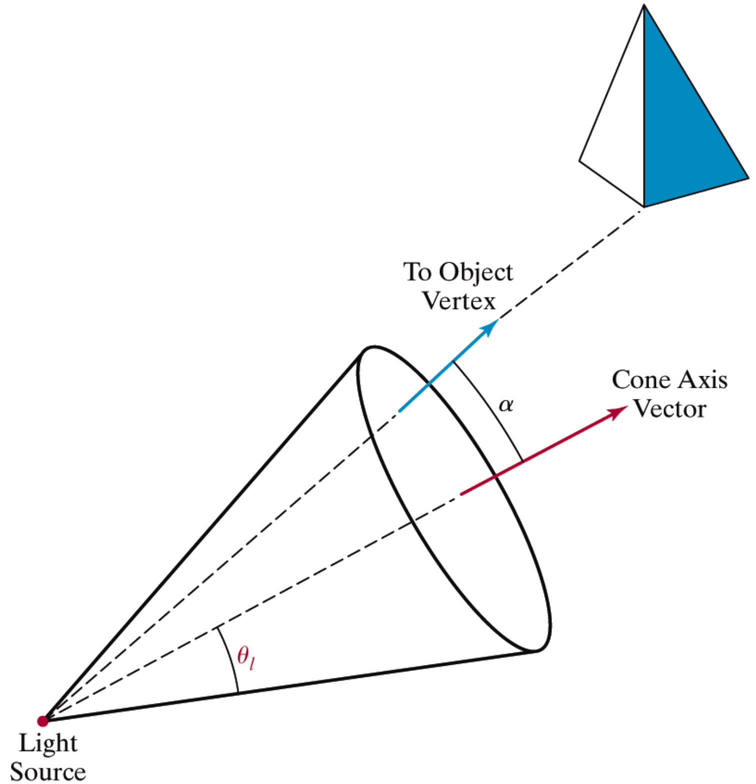
$$f = \begin{cases} 1.0 & \text{if source is at infinity} \\ \dfrac{1}{a_0 + a_1 d + a_2 d^2} & \text{if source is local} \end{cases}$$

  - The intensity attenuation is not applied to light sources at infinity because all points in the scene are at a nearly equal distance from a far-off source

Light Sources

- Angular intensity attenuation
  - For a directional light, we can attenuate the light intensity angularly as well as radially

$$f(\alpha) = \cos^n \alpha$$



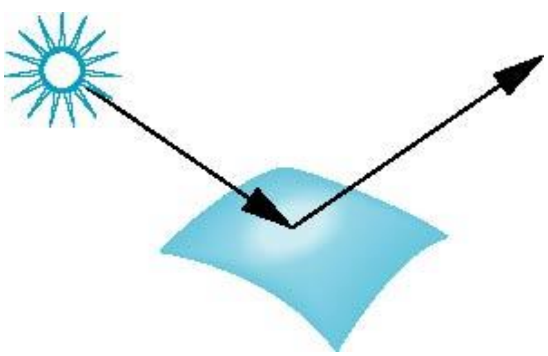
Surface Lighting Effects

- An illumination model computes the lighting effects for a surface using the various optical properties
  - Degree of transparency, color reflectance, surface texture

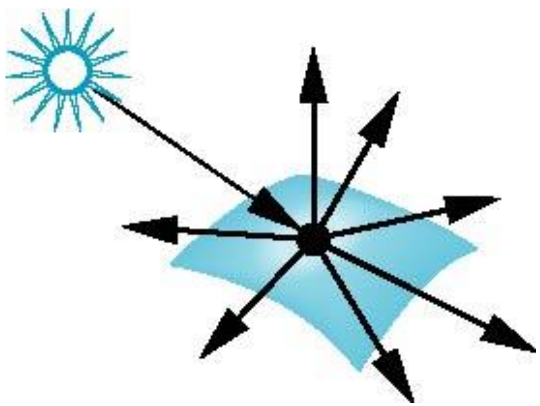
- The reflection (***phong illumination***) model describes the way incident light reflects from an opaque surface
  - Diffuse, ambient, specular reflections
  - Simple approximation of actual physical models

Surface Types

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



smooth surface



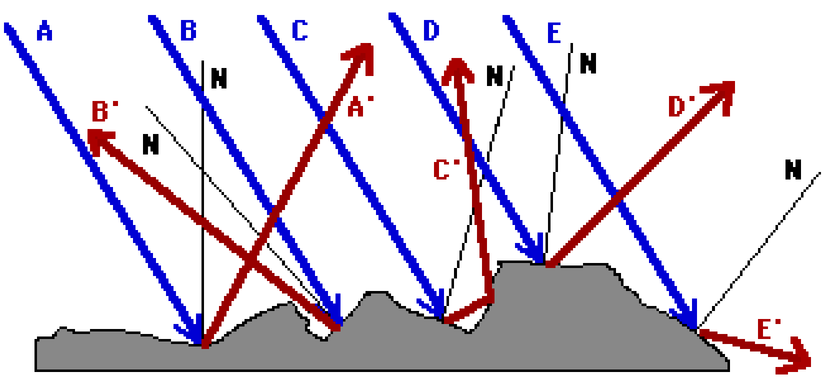
rough surface

Phong Model

- A simple model that can be computed rapidly
- Has three components
  - Diffuse
  - Specular
  - Ambient

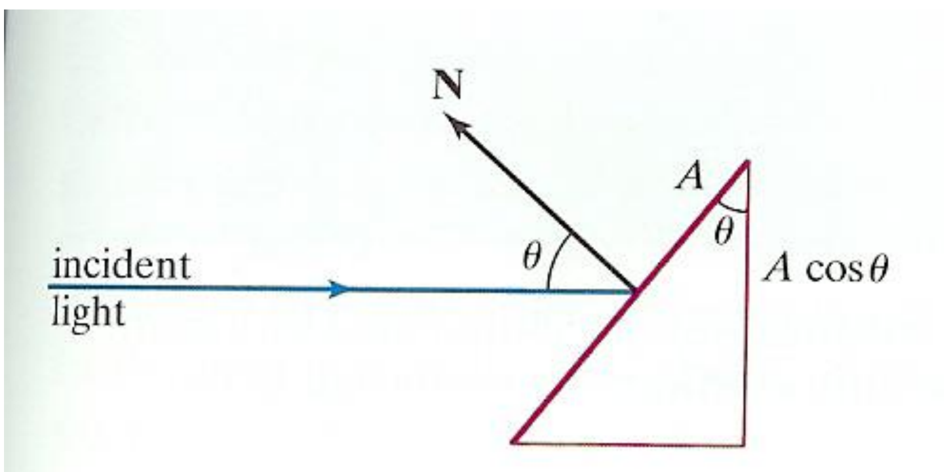
Diffuse Reflection

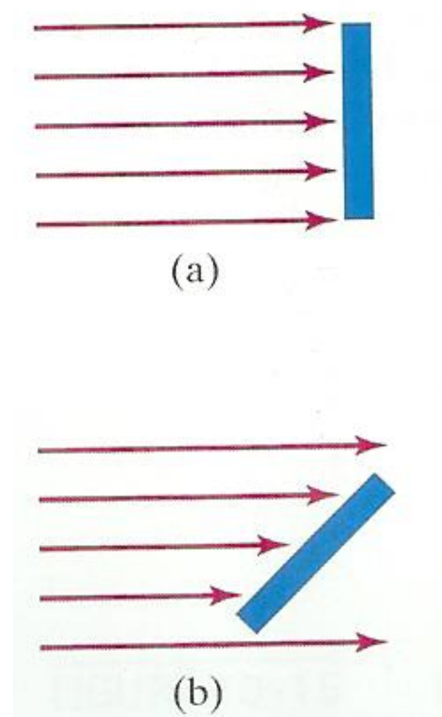
- Incident light is scattered with equal intensity in all directions
- Such surfaces are called ***ideal diffuse reflectors***
  (also referred to as ***Lambertian reflectors***)



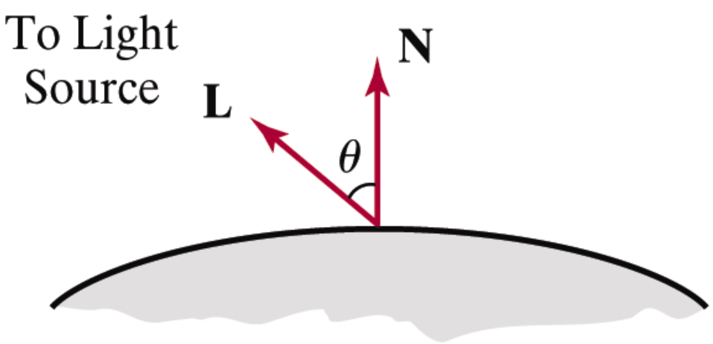
Diffuse Reflection

- Light intensity is independent of angle of reflection
- Light intensity depends on angle of incidence





Diffuse Reflection



$$I = k_d I_l \cos\theta = k_d I_l (N \cdot L)$$

I_l : the intensity of the light source

k_d : diffuse reflection coefficient,

N : the surface normal (unit vector)

L : the direction of light source, (unit vector)

Ambient Light

- Multiple reflection of nearby (light-reflecting) objects yields a uniform illumination
- A form of diffuse reflection independent of he viewing direction and the spatial orientation of a surface
- Ambient illumination is constant for an object

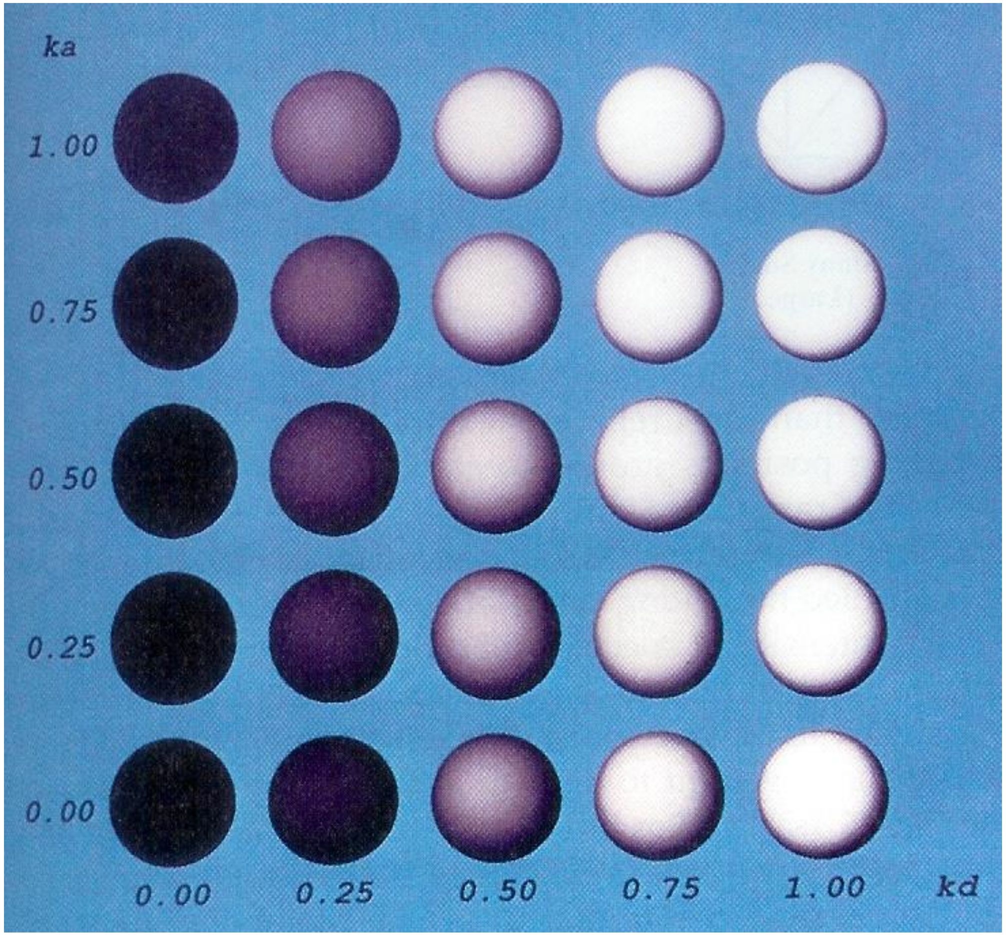
$$I = k_a I_a$$

I_a: the incident ambient intensity

k_a: ambient reflection coefficient, the proportion reflected away from the surface

Ambient + Diffuse

$$I = \begin{cases} k_a I_a + k_d I_l (N \cdot L) & \text{if} \quad N \cdot L > 0 \\ k_a I_a & \text{if} \quad N \cdot L \le 0 \end{cases}$$



Specular Reflection

- Perfect reflector (mirror) reflects all lights to the direction where angle of reflection is identical to the angle of incidence
- It accounts for the ***highlight***
- Near total reflector reflects most of light over a range of positions close to the direction

Specular Reflection

- Phong specular-reflection model
  - Note that N, L, and R are coplanar, but V may not be coplanar to the others

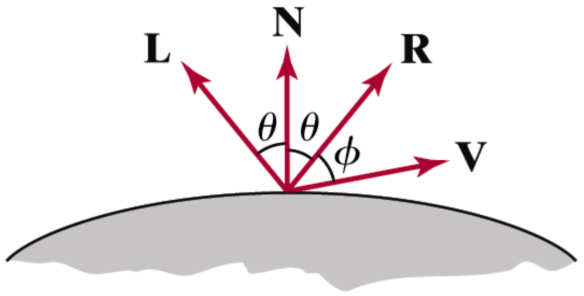


Figure 10-16

Specular reflection angle equals angle of incidence θ.

$$I = k_s I_l \cos^n \phi = k_s I_l (R \cdot V)^n$$

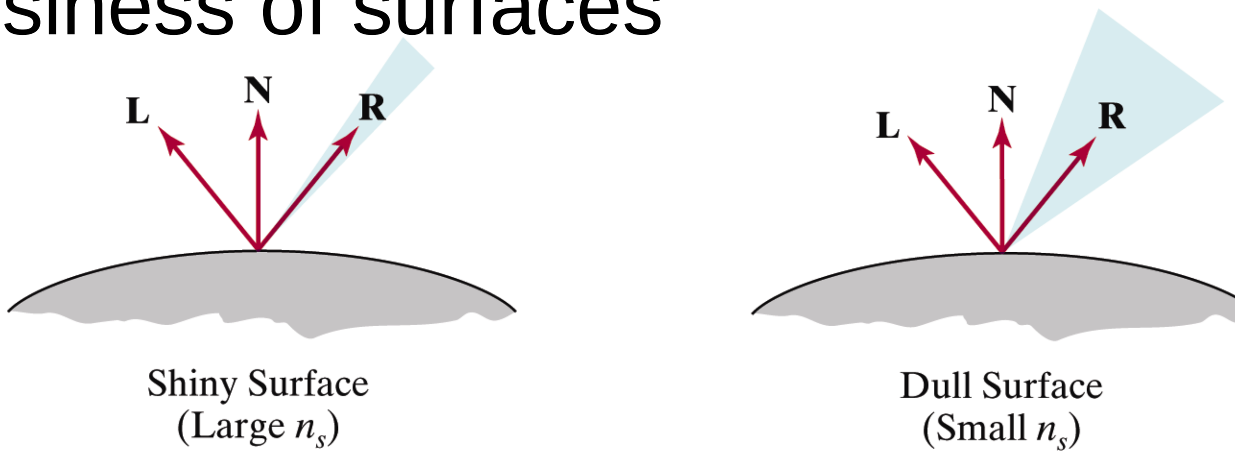
I_l : intensity of the incident light

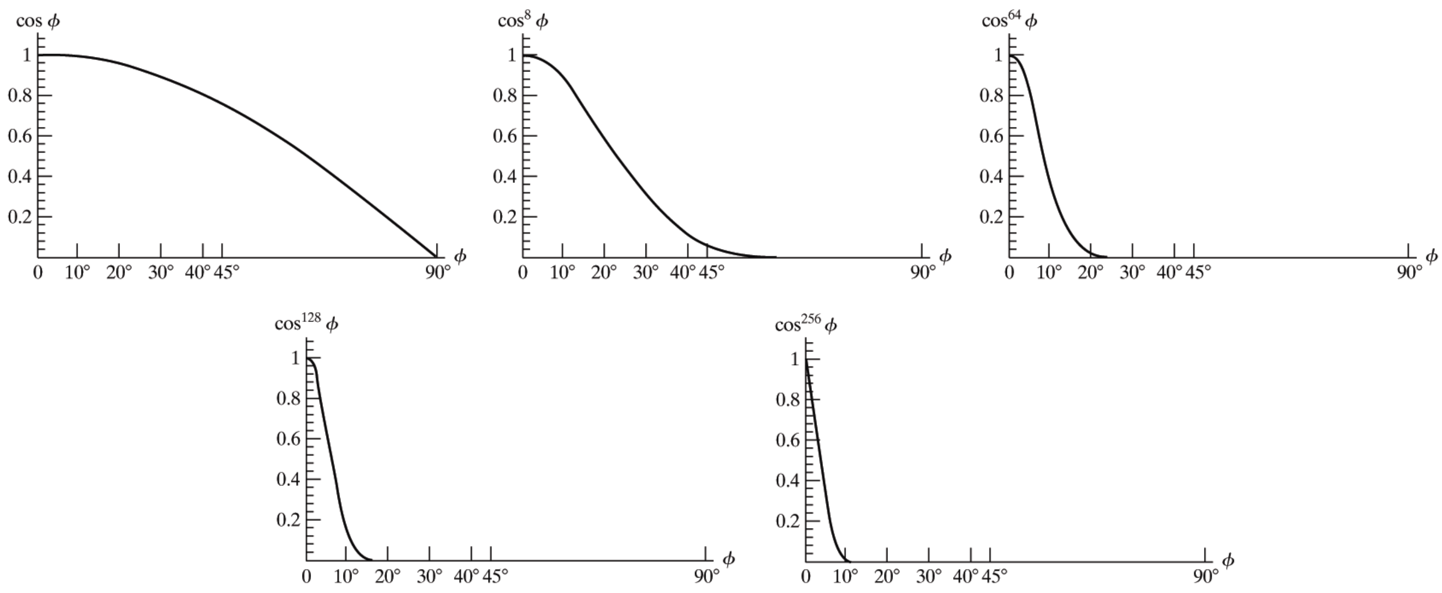
k_s : color-independent specular coefficient

n : the gloss of the surface

Specular Reflection

- Glossiness of surfaces





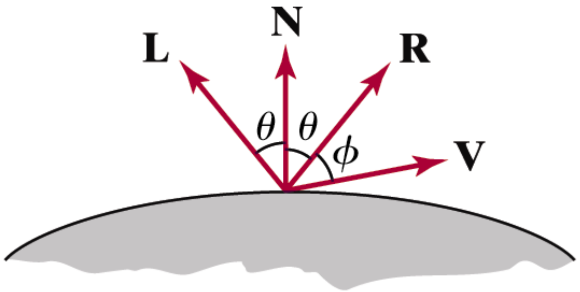
Specular Reflection

- Specular-reflection coefficient k_s is a material property
  - For some material, k_s varies depending on θ
  - $k_s = 1$ if $\theta = 90^\circ$

- Calculating the reflection vector R

$$R + L = (2L \cdot N)N$$

$$R = (2L \cdot N)N - L$$



Specular Reflection

- Simplified Phong model using halfway vector
  - H is constant if both viewer and the light source are sufficiently far from the surface

$$H = \frac{V + L}{|V + L|}$$

$$I = I_p k_s \cos^n \phi = I_p k_s (R \cdot V)^n$$

$$\approx I_p k_s \cos^n \alpha = I_p k_s (N \cdot H)^n$$

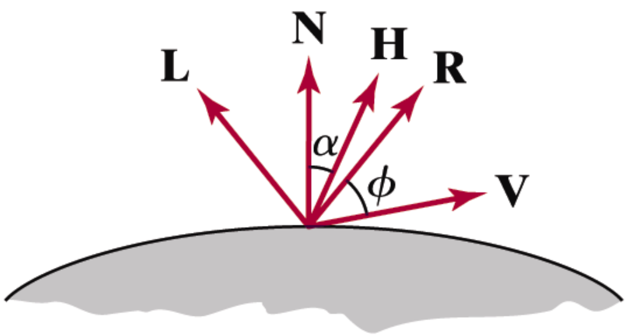


Figure 10-22

Halfway vector **H** along the bisector of the angle between **L** and **V**.

Ambient+Diffuse+Specular Reflections

- Single light source

$$I = k_a I_a + k_d I_l (N \cdot L) + k_s I_l (R \cdot V)^n$$

- Multiple light source

$$I = k_a I_a + \sum_l k_d I_l (N \cdot L) + k_s I_l (R \cdot V)^n$$

- Emission and attenuation

$$I = I_{emit} + k_a I_a$$
$$+ \sum_l f_{l,rad\_atten} f_{l,ang\_atten} \left(k_d I_l (N \cdot L) + k_s I_l (R \cdot V)^n \right)$$

Parameter Choosing Tips

- For a RGB color description, each intensity and reflectance specification is a three-element vector
- The sum of reflectance coefficients is usually smaller than one $k_a + k_d + k_s \leq 1$
- Try *n* in the range [0, 100]
- Use a small k_a (~0.1)
- Example
  - Metal: n=90, k_a=0.1, k_d=0.2, k_s=0.5

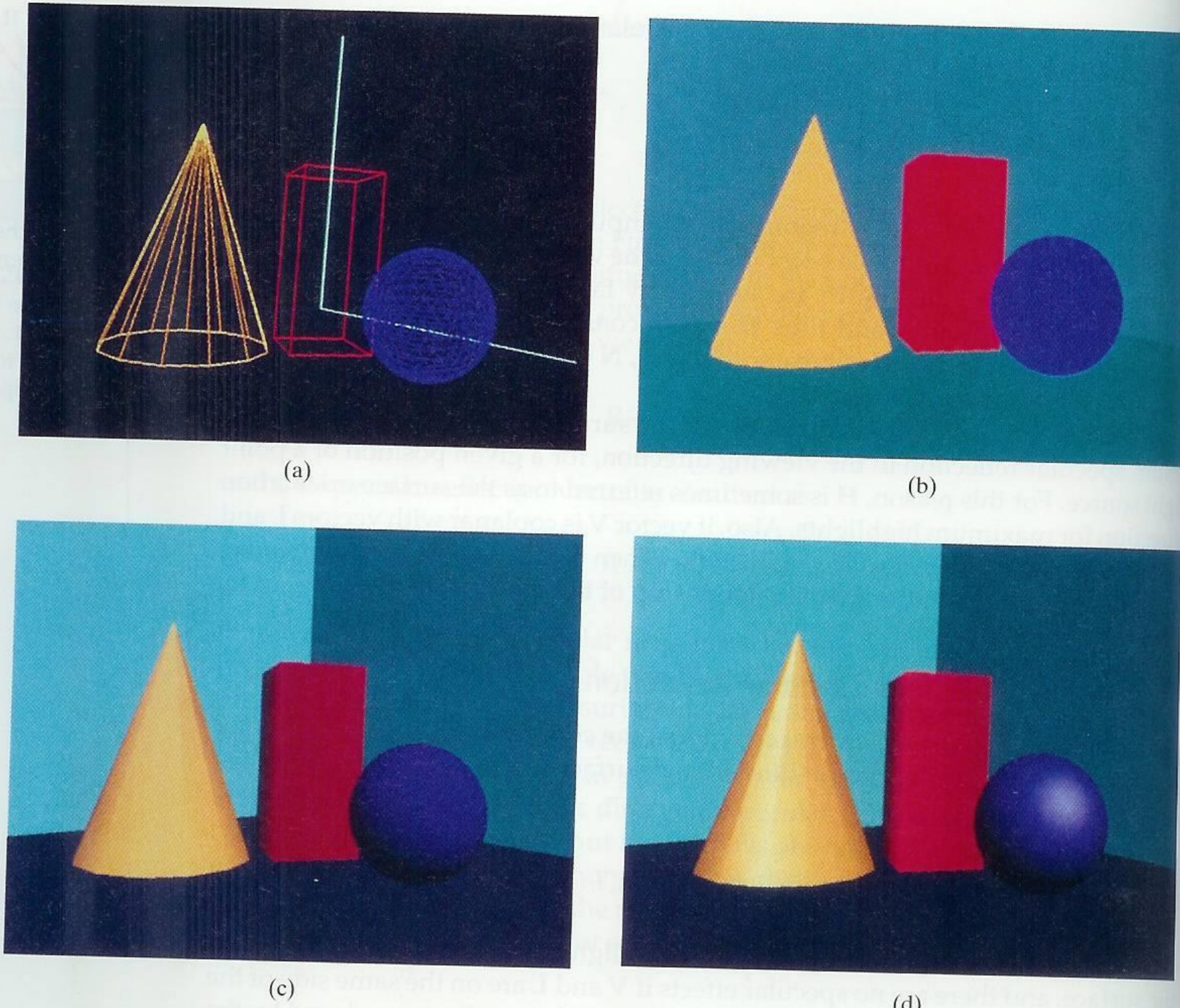


FIGURE 10-