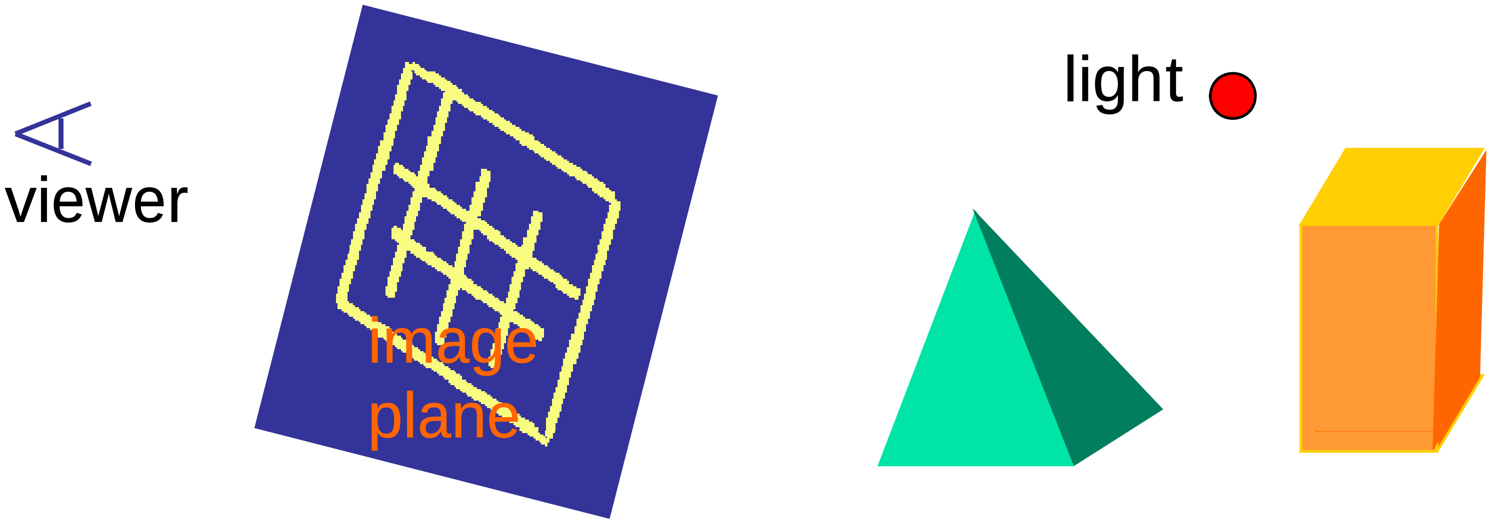
Illumination and Shading

Illumination and Shading

- Given: scene specification (object positions, optical properties of the surface, viewer position, viewing direction, ...)
- Find: intensity for each pixel



Illumination Models (lighting model, shading model)

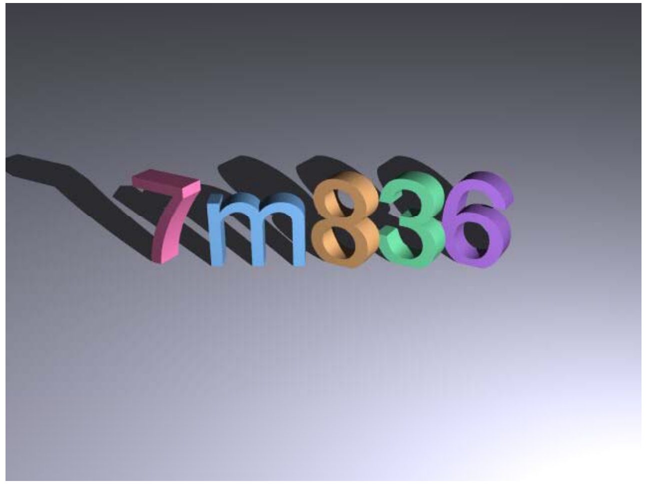
- Photorealism in computer graphics involves
  - Accurate representations of surface properties, and
  - Good physical descriptions of the lighting effects
- Rendering needs a model for how light interacts with objects.
  - Physically more correct model: the intensity reflected from every point depends on the intensity from every other points

    – global illumination
  - Simplistic model: the intensity depends only on the direct illumination due to light sources

    – local illumination

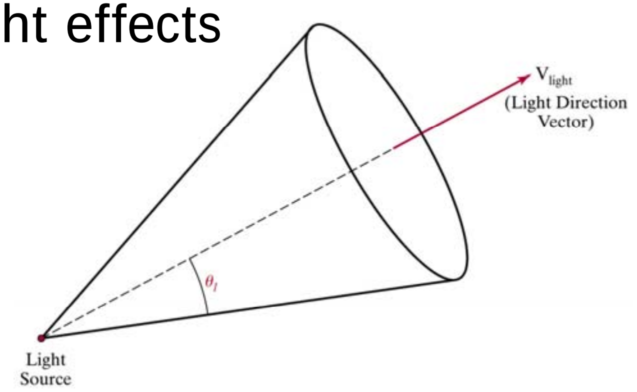
Light Sources

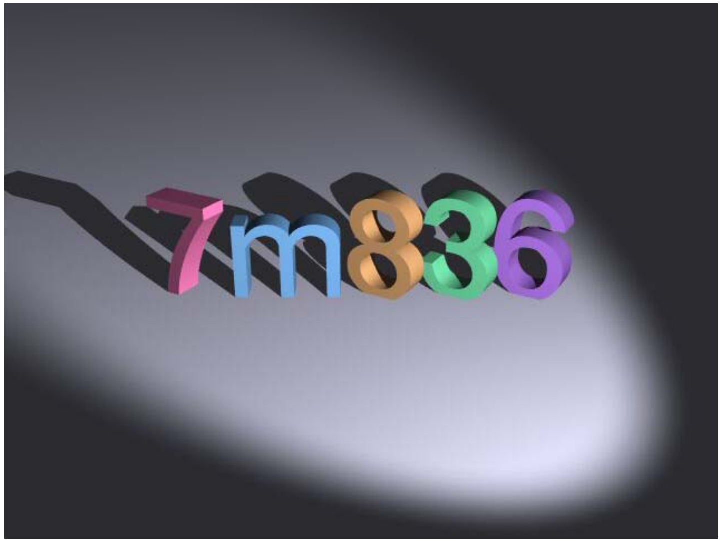
- Point light sources
  - Emitting radiant energy at a single point
  - Specified with its position and the color of the emitted light
- Infinitely distant light sources
  - A large light source, such as sun, that is very far from a scene
  - Little variation in its directional effects
  - Specified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



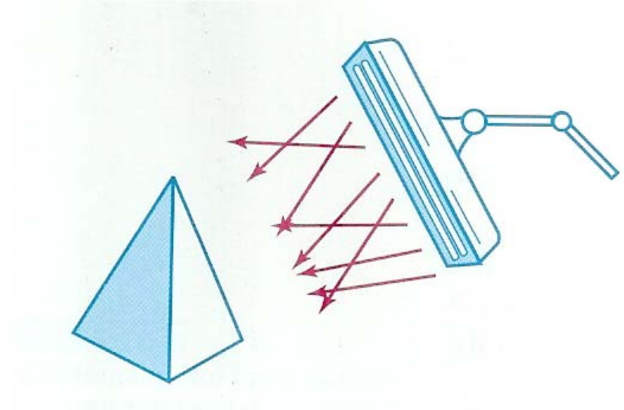
Light Sources

- 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

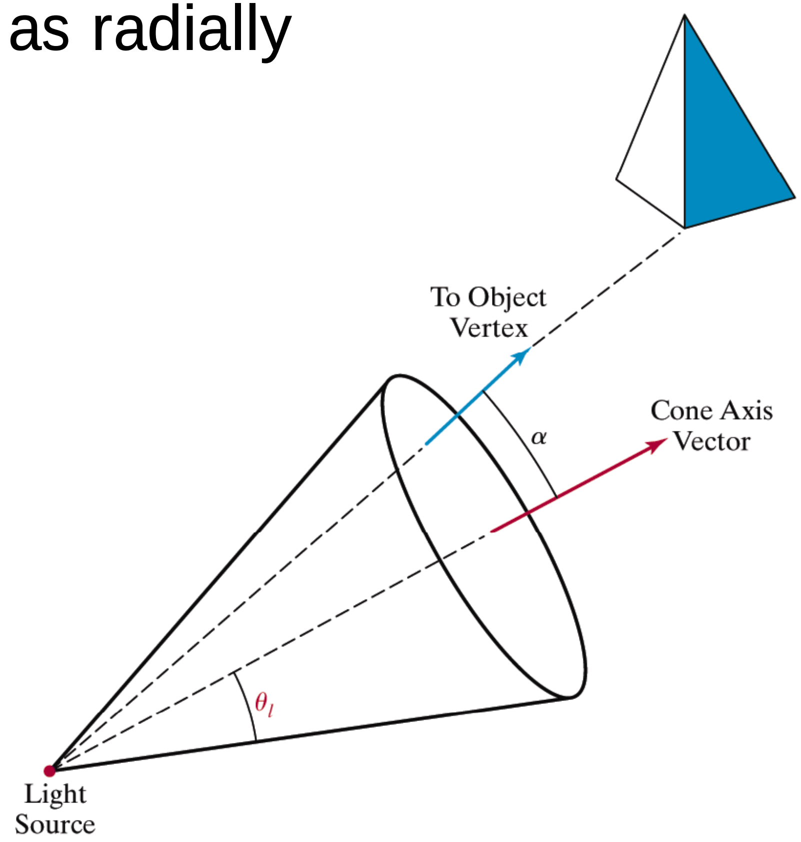
$$f = \begin{cases} 1.0 & \text{if source is at infinity} \\ \dfrac{1}{a_0 + a_1 d + a_2 d^2} & \text{if source is local} \end{cases}$$

  - 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 model describes the way incident light reflects from an opaque surface
  - Diffuse, ambient, specular reflections
  - Simple approximation of actual physical models
  - This model is known as:
    - shading model
    - lighting model
    - light reflection model
    - local illumination model
    - Phong illumination model
    - reflectance model

Ambient Light

- Multiple reflection of nearby objects (light-reflecting sources) yields a uniform illumination
- Ambient light is independent of light sources and viewer position
- Ambient illumination is constant for an object, without regard to directions of faces

$$I = k_a I_a$$

I_a : the incident ambient intensity

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

- Indirect illumination can be computed much better: e.g. radiosity

Wavelength dependence

k_a and I_a are function over all wavelengths λ .

Ideally, we need

$$I(\lambda) = k_a(\lambda) I_a(\lambda)$$

Calculate RGB component separately:

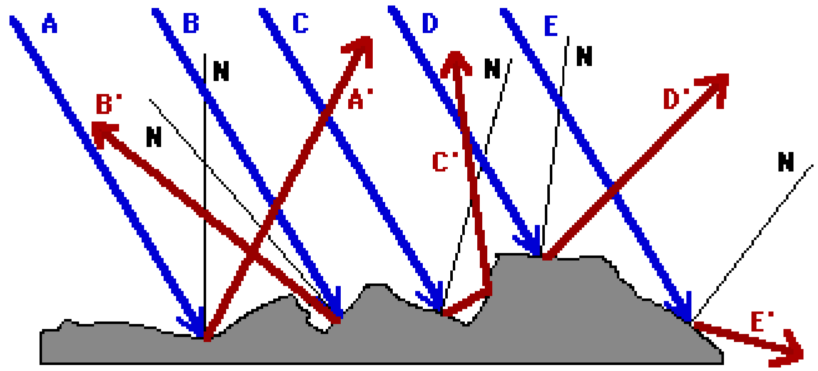
$$I_{red} = k_{a,red} \; I_{a,red}$$

$$I_{green} = k_{a,green} \; I_{a,green}$$

$$I_{blue} = k_{a,blue} \; I_{a,blue}$$

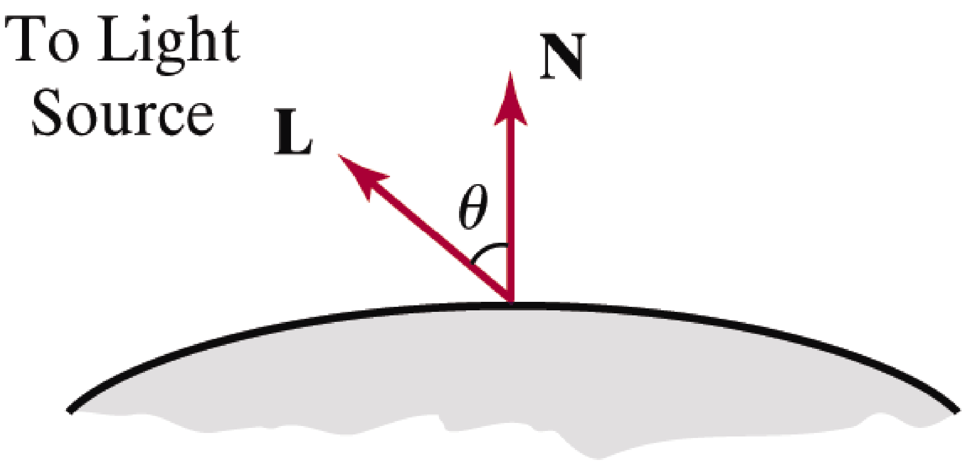
Diffuse Reflection

- Matte (dull) surfaces diffuse incident light uniformly to every direction
  (light intensity is independent of angle of reflection)
- Such surfaces are called *ideal diffuse reflectors* (also referred to as *Lambertian reflectors*)



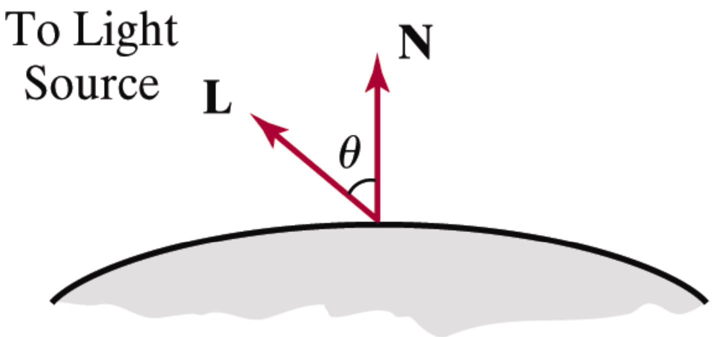
Diffuse Reflection

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



Diffuse Reflection

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



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

I_l : the intensity of the light source

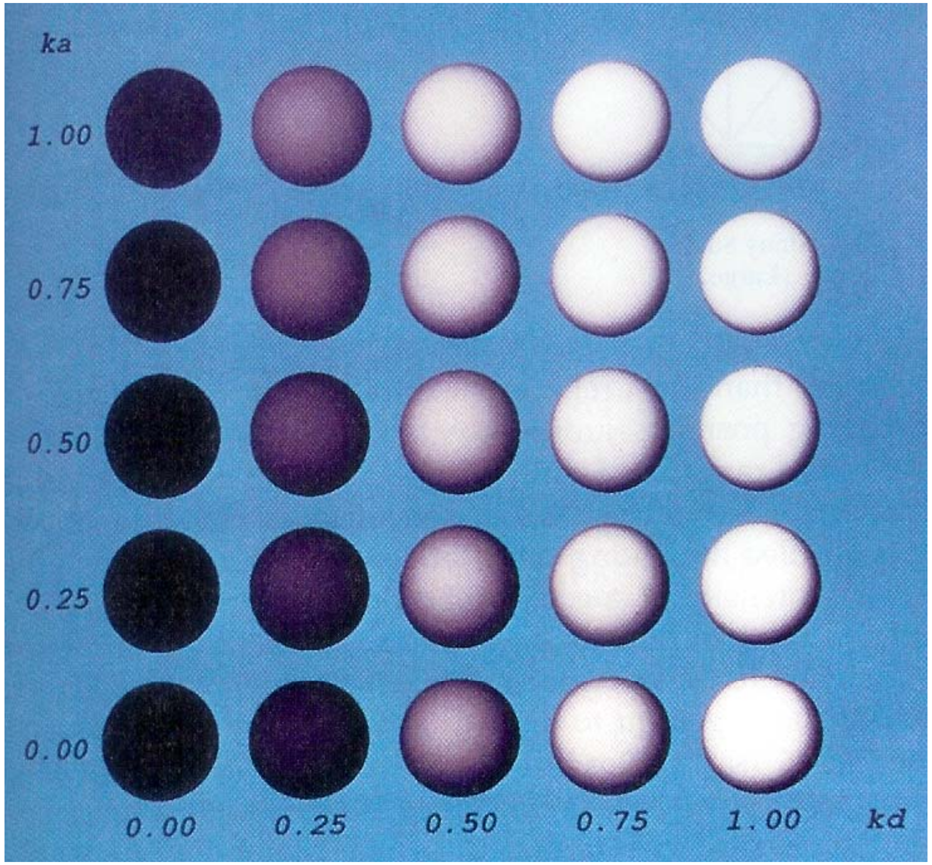
k_d: diffuse reflection coefficient,

N : the surface normal (unit vector)

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

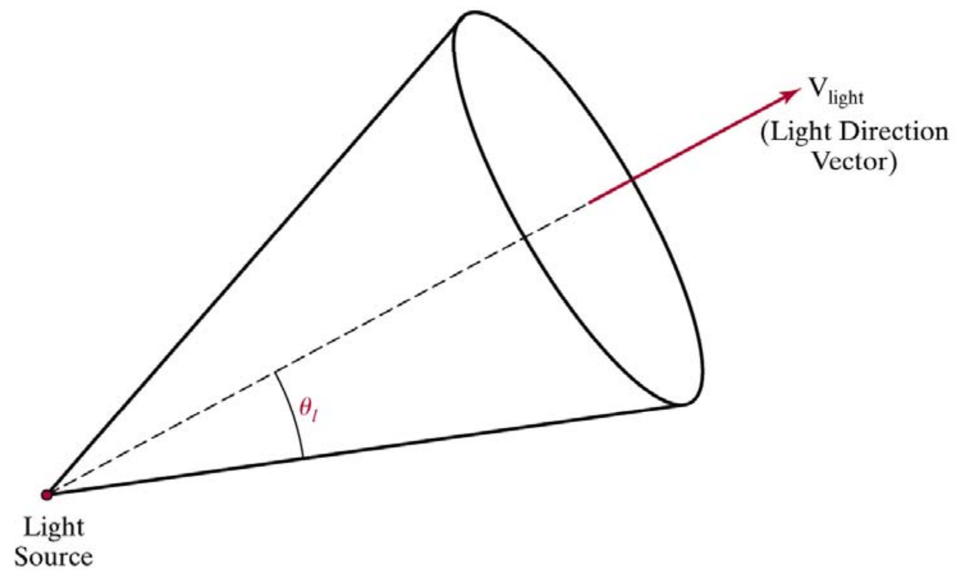
Ambient + Diffuse

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



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

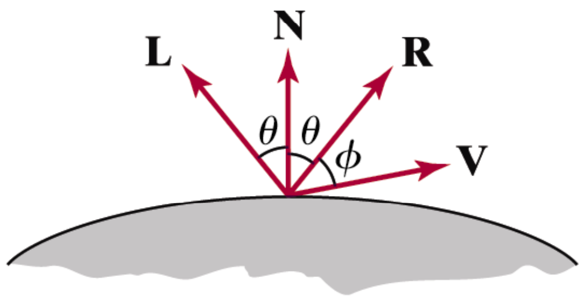


Figure 10-16

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

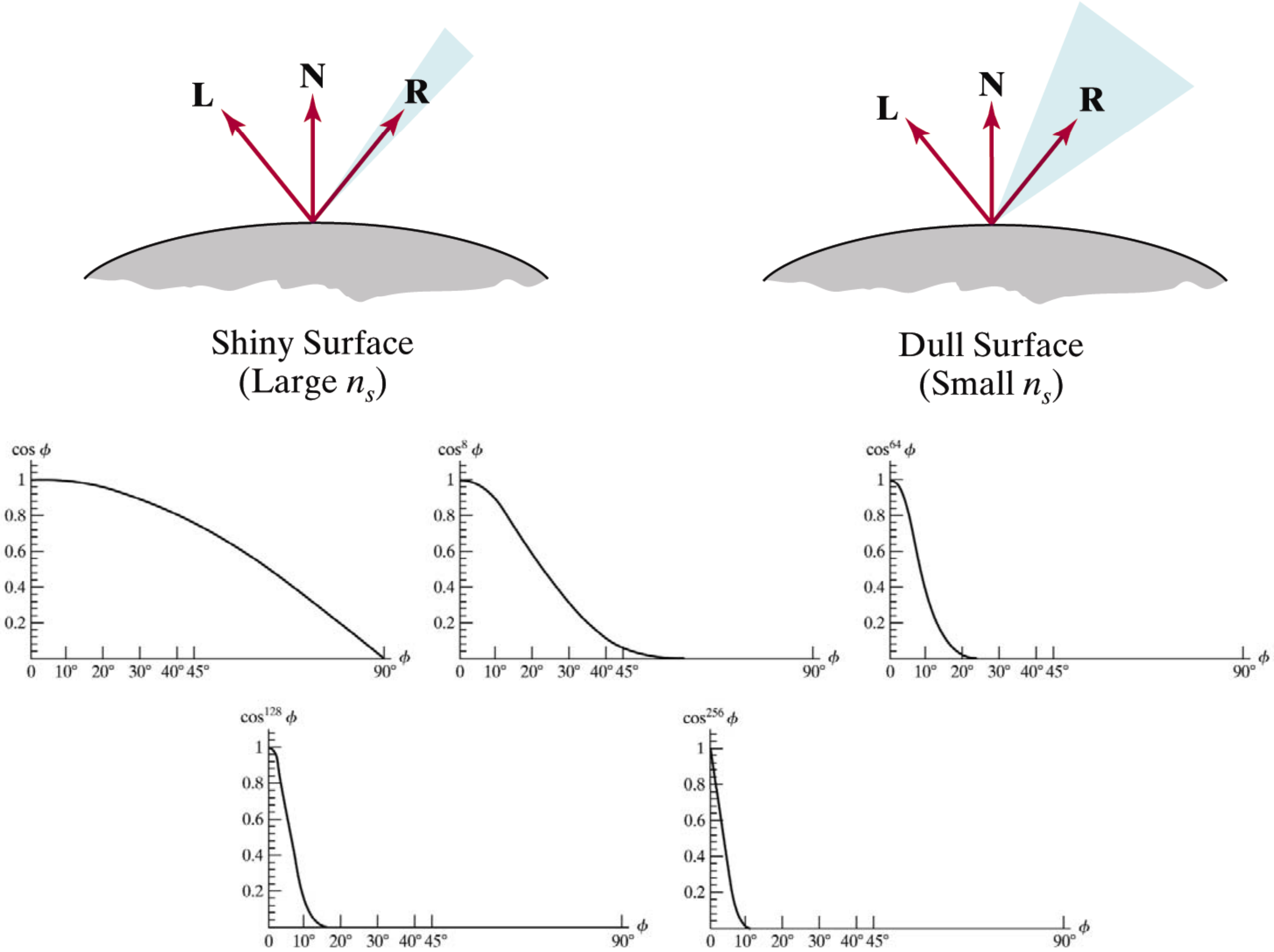
k_s : color-independent specular-reflection coefficient

n : the gloss of the surface

Specular Reflection

- Glossiness of surfaces

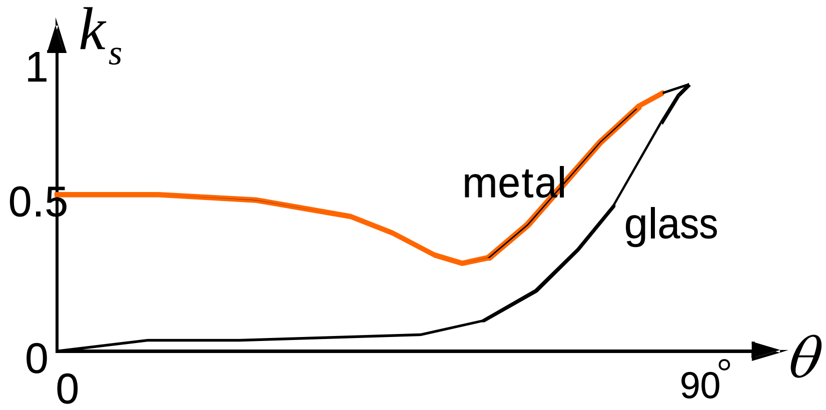
$$I = k_s I_l \cos^n \phi = k_s I_l (R \cdot V)^n$$



Specular Reflection

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

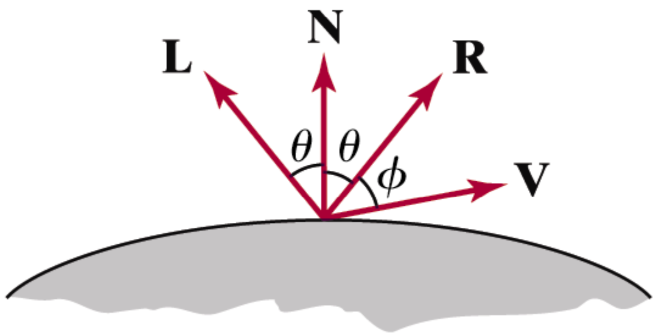
- Specular-reflection coefficient k_s is a material property
  - For some material, k_s varies depending on θ



- Calculating the reflection vector R

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

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



Specular Reflection

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

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

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

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

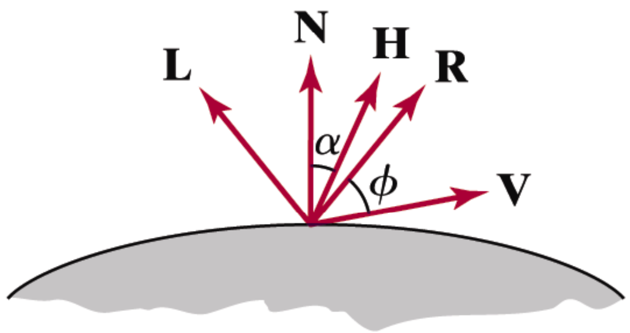
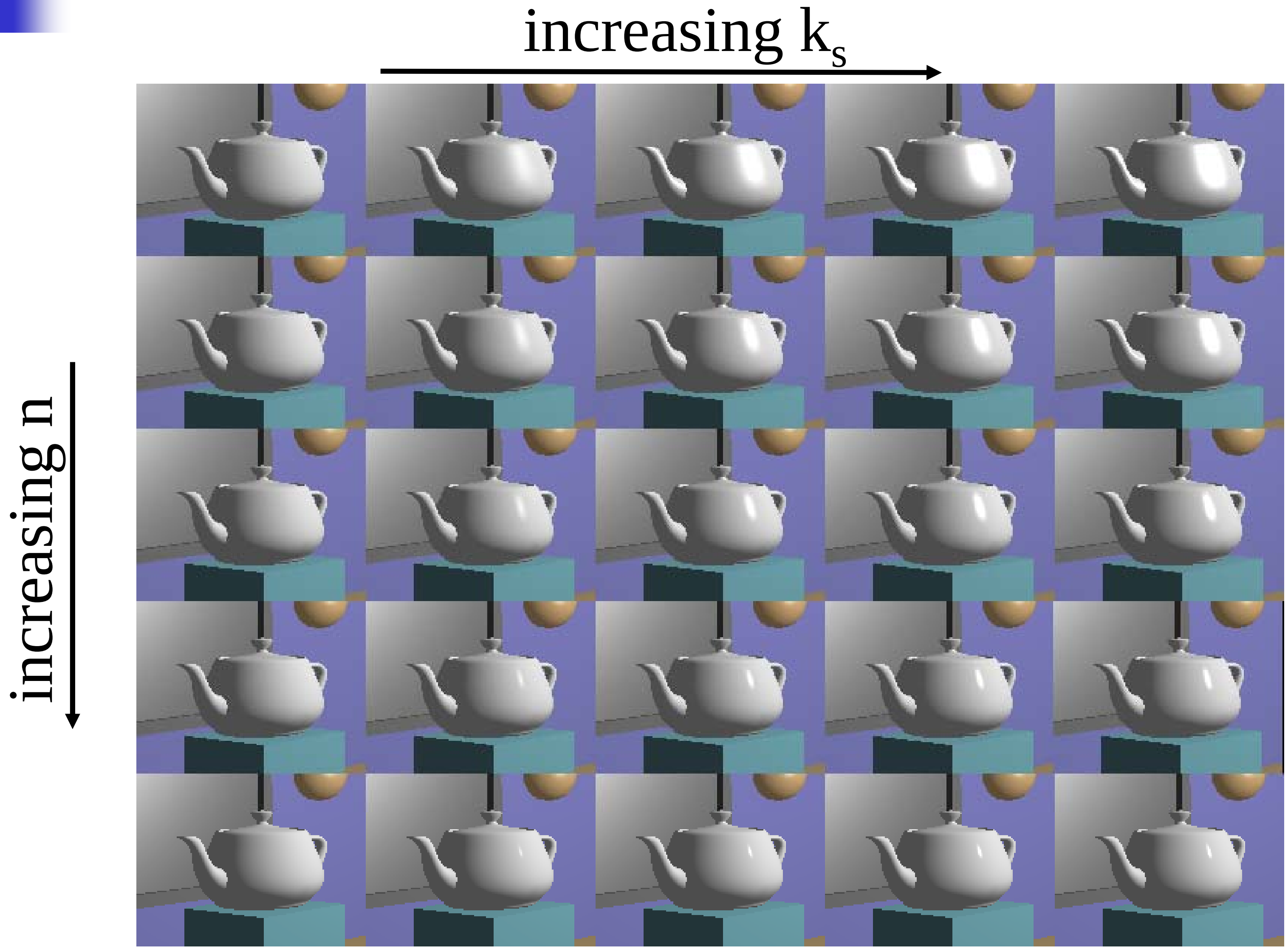


Figure 10-22

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

Specular reflection / Phong



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

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

- 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: $n=90,\ k_a=0.1,\ k_d=0.2,\ k_s=0.5$

Only emission and ambient

Including diffuse reflection

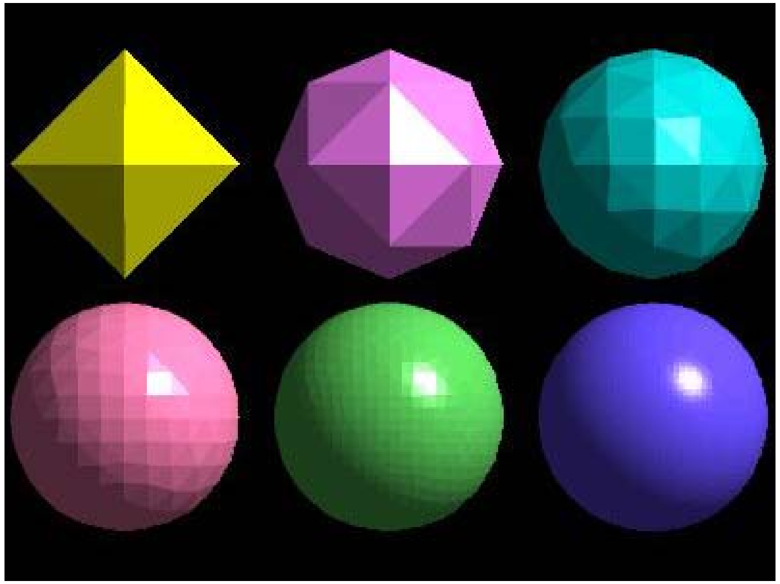
Including specular reflection

Polygon Rendering Methods

- We could use an illumination model to determine the surface intensity at every projected pixel position
- Or, we could apply the illumination model to a few selected points and approximate the intensity at the other surface positions
- Curved surfaces are often approximated by polygonal surfaces. So, polygonal (piecewise planar) surfaces often need to be rendered as if they are smooth
- Constant shading, Gouraud shading, Phong shading

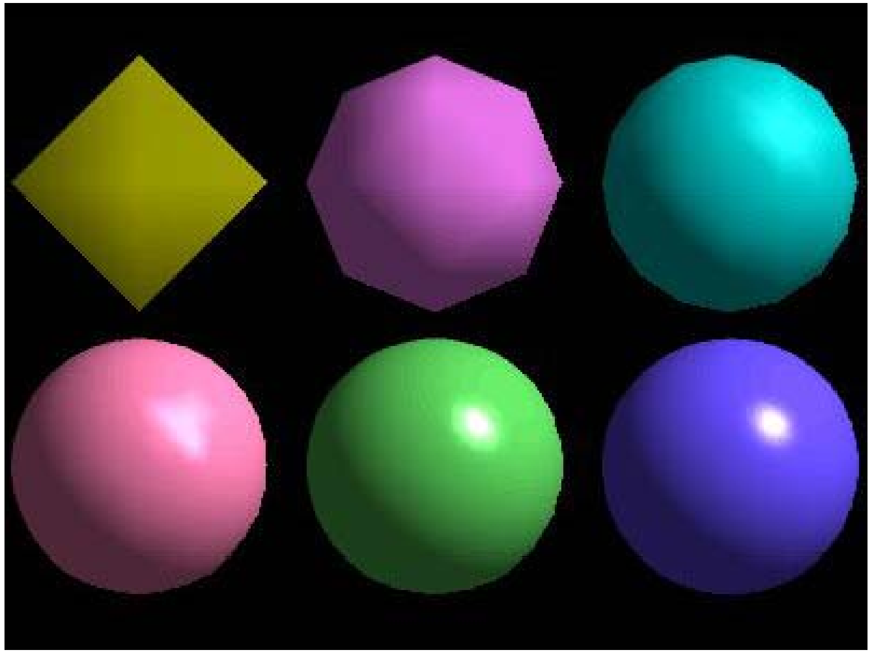
Constant Shading (Flat Shading)

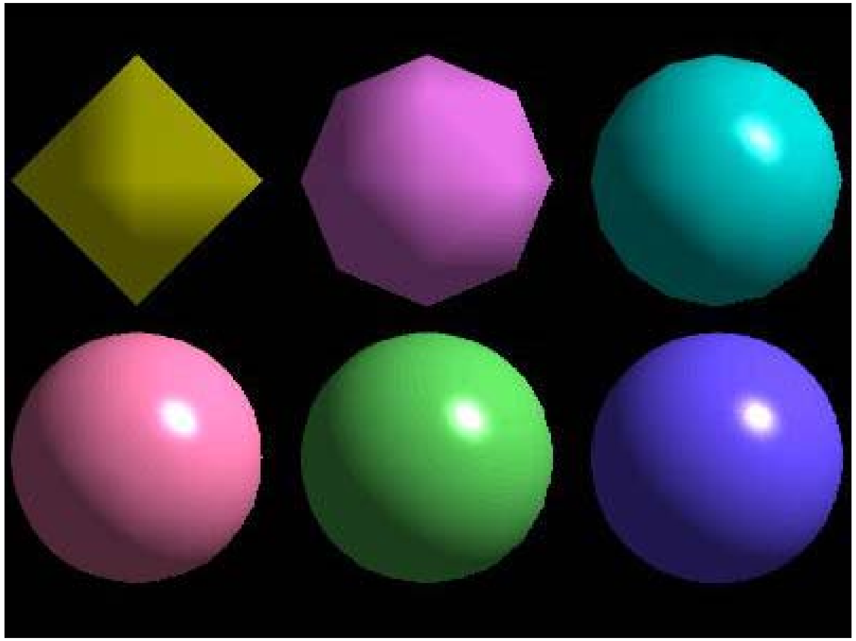
- Infinitely distant light source (constant $N \cdot L$) result in constant diffuse reflection
- Constant $N \cdot L$ and infinitely distant viewpoint (constant $V \cdot R$) result in constant specular reflection
- Abrupt change in surface orientation of adjacent surfaces produce an unrealistic effect



Smooth shading

- Two popular methods:
  - Gouraud shading
  - Phong shading





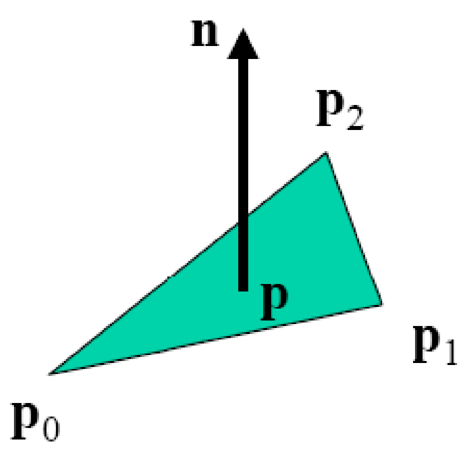
Normal vector of a vertex

Plane Normal

plane $\mathbf{n} \cdot (\mathbf{p} - \mathbf{p}_0) = 0$

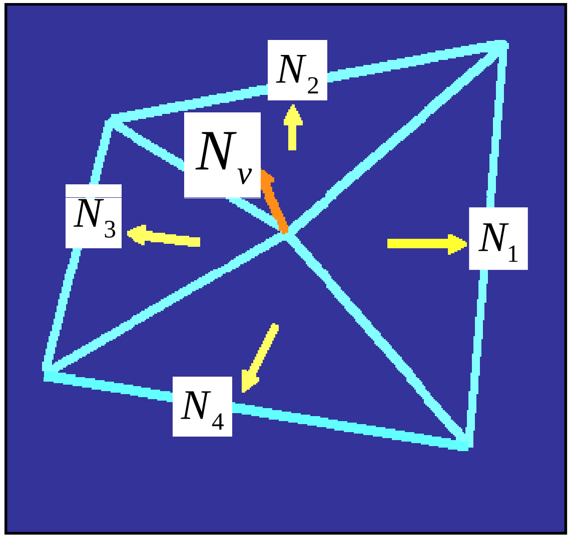
$$\mathbf{n} = (\mathbf{p}_2 - \mathbf{p}_0) \times (\mathbf{p}_1 - \mathbf{p}_0)$$

normalize $\mathbf{n} \leftarrow \mathbf{n} / |\mathbf{n}|$



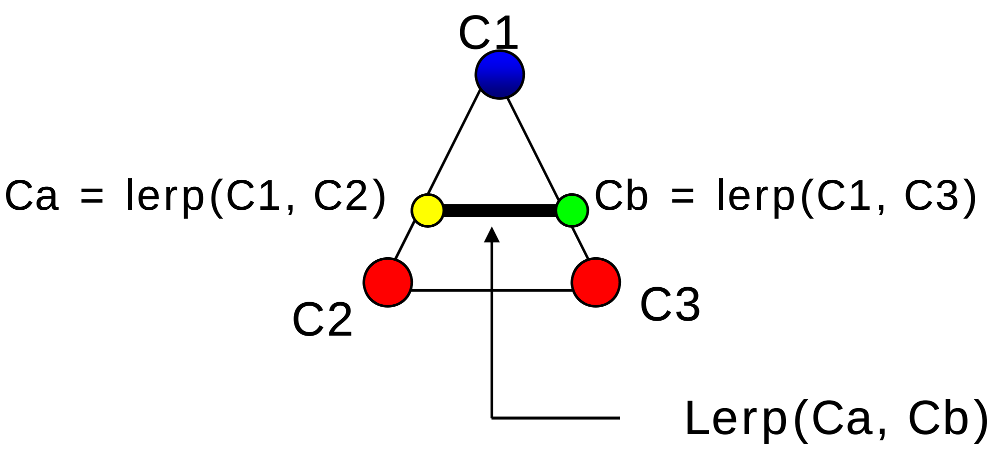
Vertex Normal

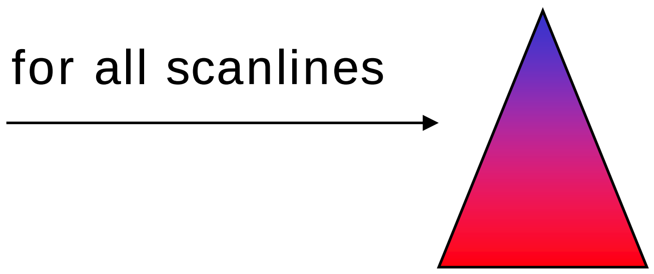
$$N_v = \frac{(N_1 + N_2 + N_3 + N_4)}{\|N_1 + N_2 + N_3 + N_4\|}$$



Gouraud Shading

- Compute vertex illumination (color) before the projection transformation
- Shade interior pixels: color interpolation (normals are not needed)

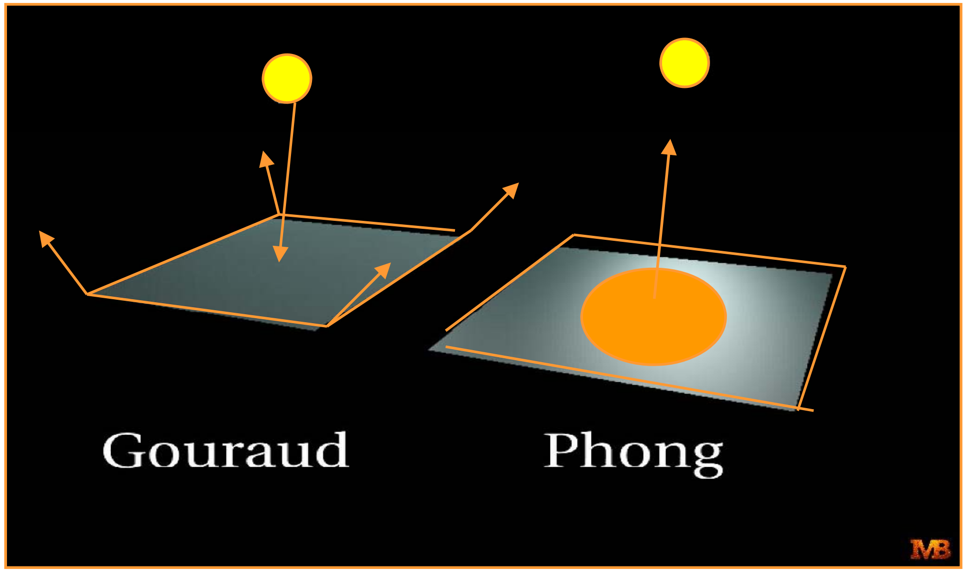


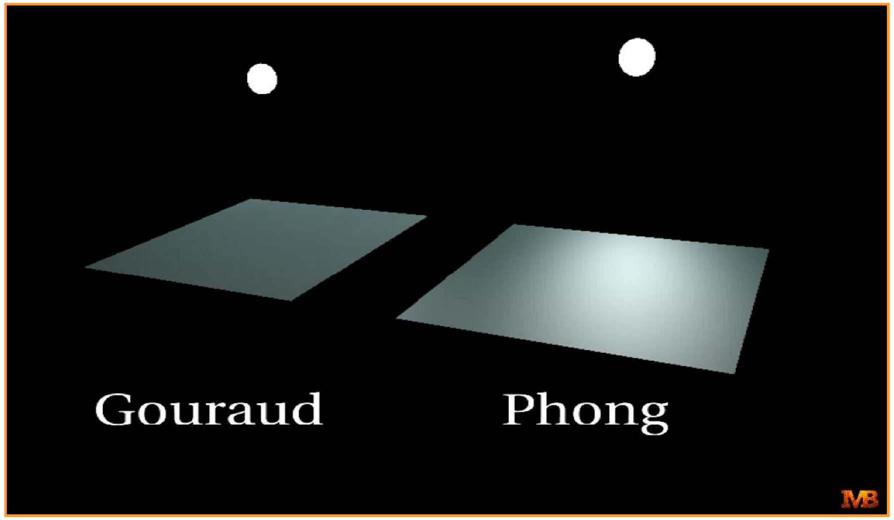


* lerp: linear interpolation

Gouraud Shading Problem

- Lighting in the polygon interior can be inaccurate



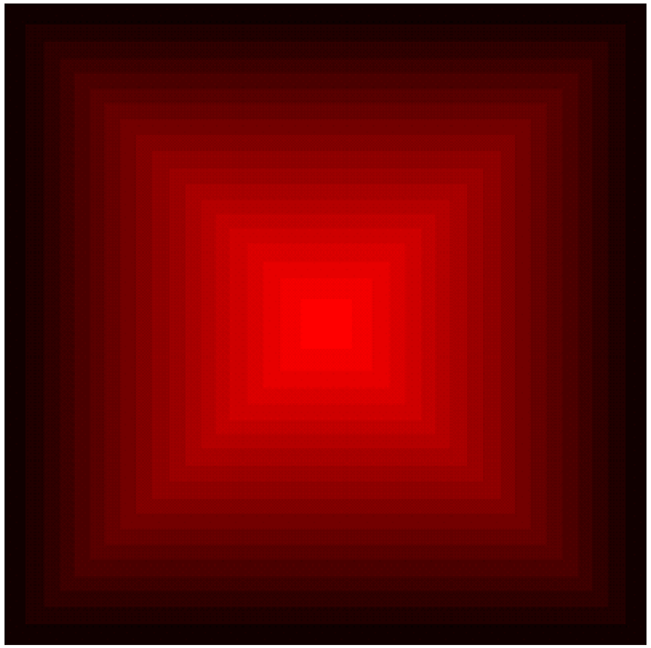


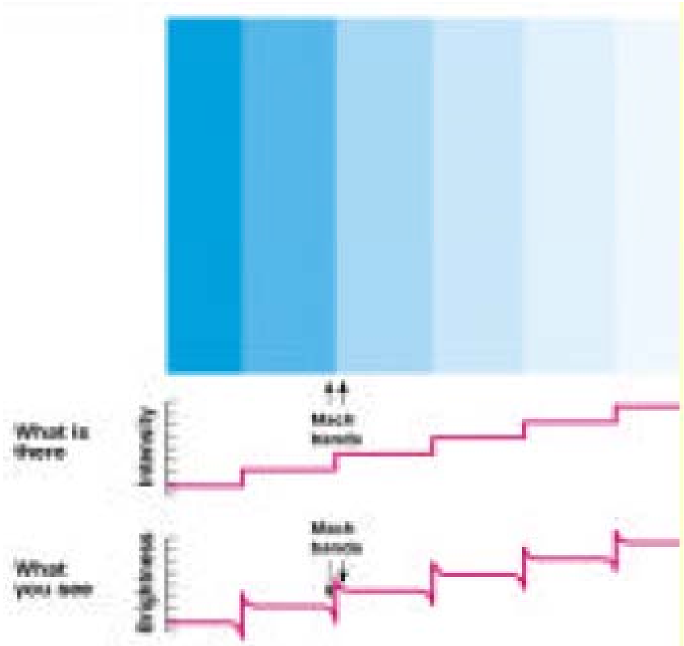
Gouraud Shading Problem - Mach band effect

These “Mach Bands” are not physically there. Instead, they are illusions due to excitation and inhibition in our neural processing.

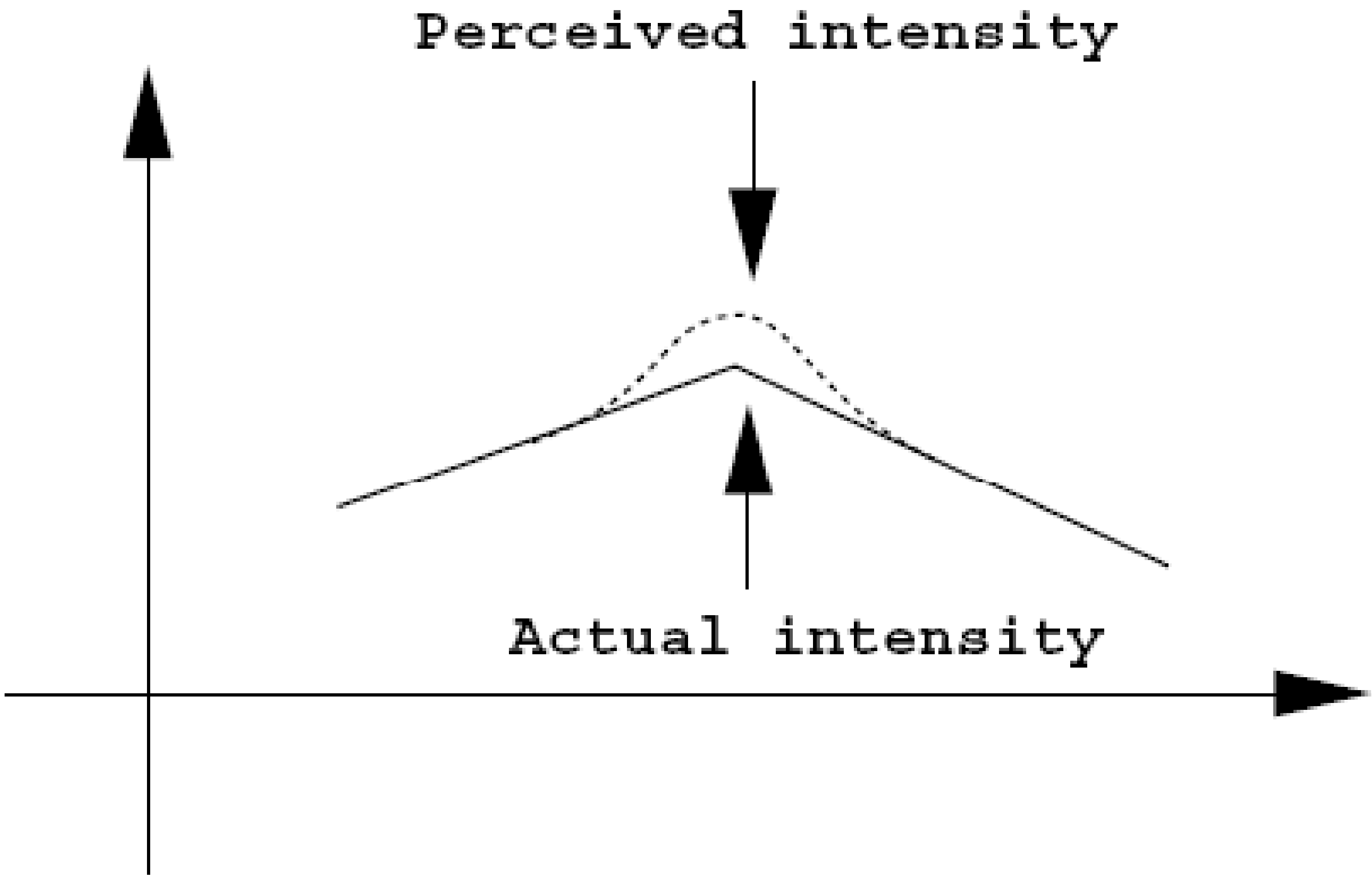
The bright bands at 45 degrees (and 135 degrees) are illusory.

The intensity of each square is the same.



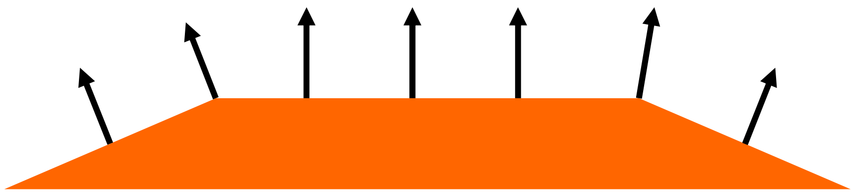


Mach band effect



Phong Shading

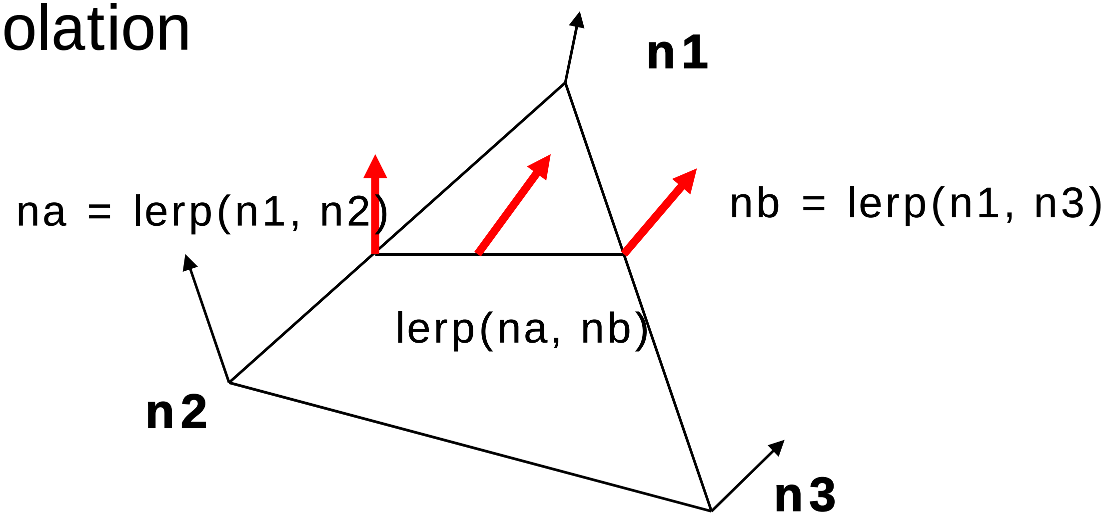
- Surface normals are interpolated



- Shades are computed at each point using the interpolated normal vector
- The shading computed by Phong shading is C_1 continuous.
  - Fix the mach band effect : remove edge discontinuity

Phong Shading

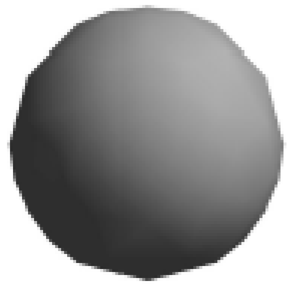
- Normal interpolation

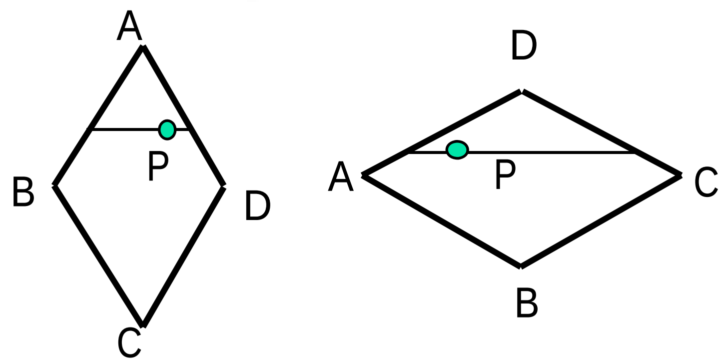


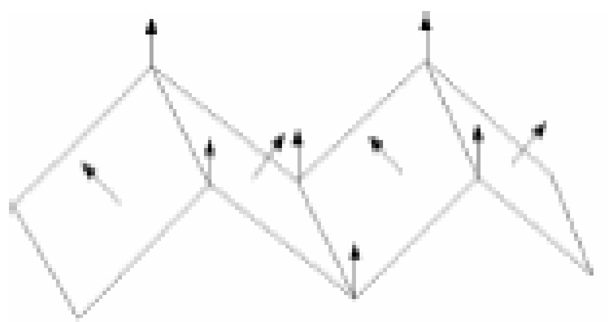
- Slow – not supported by OpenGL and most graphics hardware
- Modern programmable lighting hardware can implement full phong shading (and much more!), but you have to do this yourself.

Problems with interpolated shading

- Polygon silhouette
- Perspective distortion
- Orientation dependence
- Shared edges
- Unrepresentative vertex normals





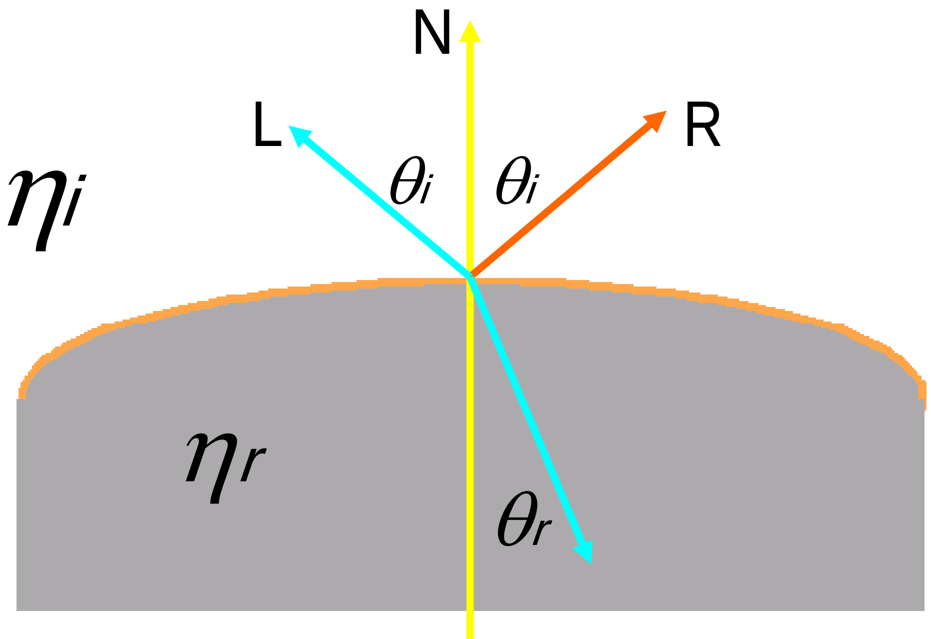


Refractions (Transparent Surfaces)

▷ diffuse refraction

- Partially transparent object (e.g. frosted glass) penetrating light is diffused
- Decrease light intensity, spread intensity contribution of each point onto a finite area on the refracting surface
- Expensive, seldom used

Specular refraction (Snell's law)



$$\sin\theta_r = \frac{\eta_i}{\eta_r}\sin\theta_i$$

η_i, η_r : index of refraction of each material
(averaged over wavelengths)

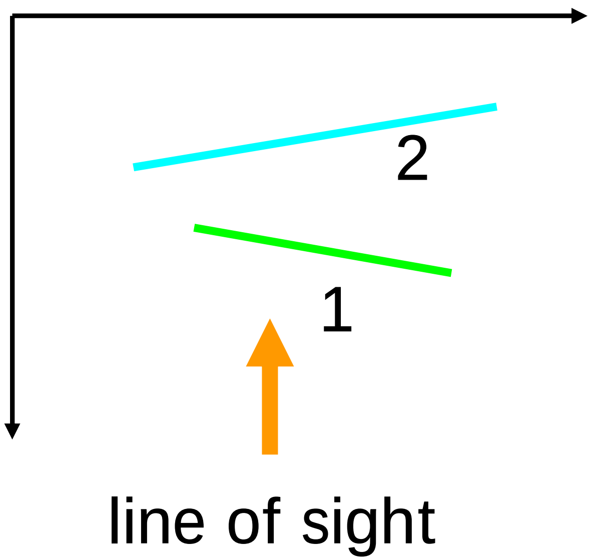
Specular refraction

- Path shifts are ignored for thin objects
- From Snell's law, we can obtain the unit transmission vector T in the direction θ_r

$$T = \left(\frac{\eta_i}{\eta_r}\cos\theta_i - \cos\theta_r\right)N - \frac{\eta_i}{\eta_r}L$$

Interpolated transparency

$$I = (1 - k_{t1}) I_{\lambda 1} + k_{t1} I_{\lambda 2}$$



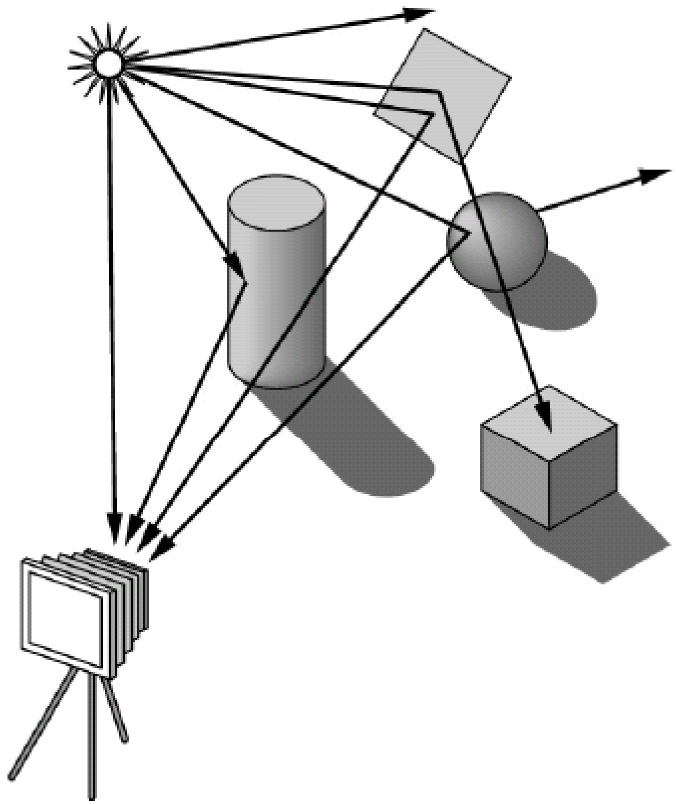
k_{t1} : transmission coefficient
(0 for opaque objects,
1 for totally transparent
objects)

Local vs. Global Illumination Models

- Local illumination models
  - Object illuminations are independent
  - No light scattering between objects
  - No real shadows, reflection, transmission
  - Phong Illumination model
- Global illumination models
  - Ray tracing (highlights, reflection, transmission)
  - Radiosity (Surface inter-reflections)
  - Photon mapping

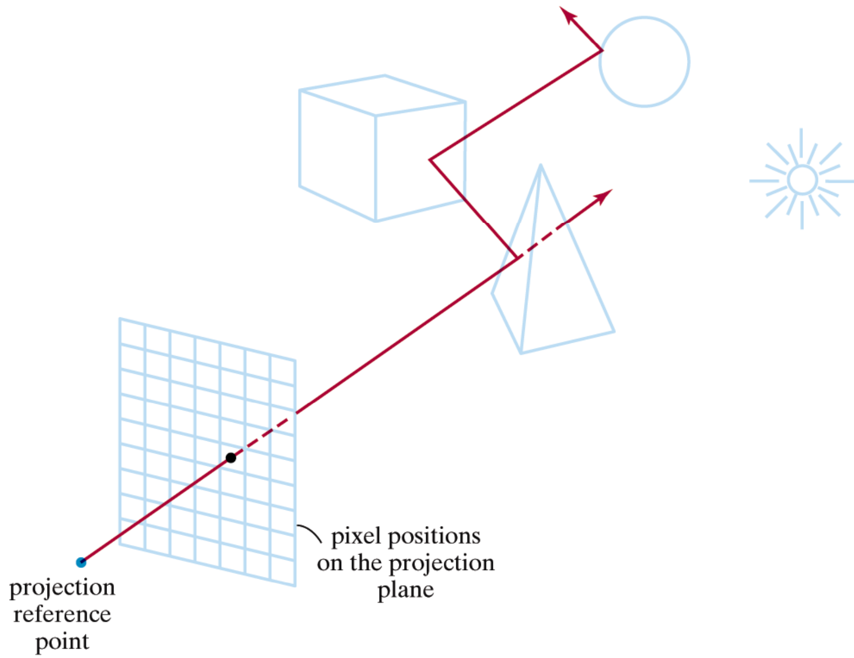
Forward Ray Tracing

- Rays as paths of photons in world space
- Follow photon from light sources to viewer
- Problem: Many rays will not contribute to image



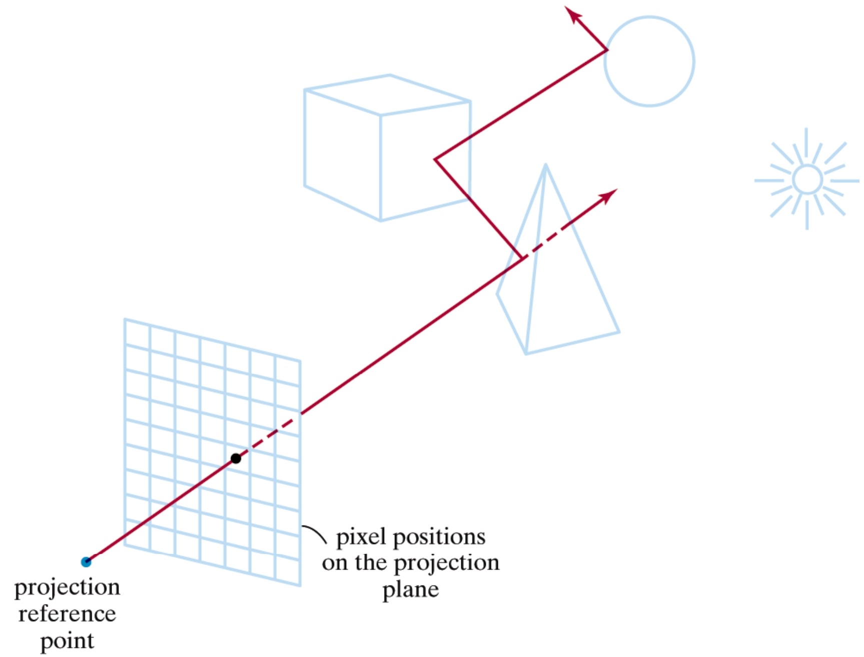
Backward Ray Tracing

- Trace rays backward from viewer to light sources
- One ray from center of projection through each pixel in image plane
- Ray casting
  - Simplest form of ray tracing
  - No recursion



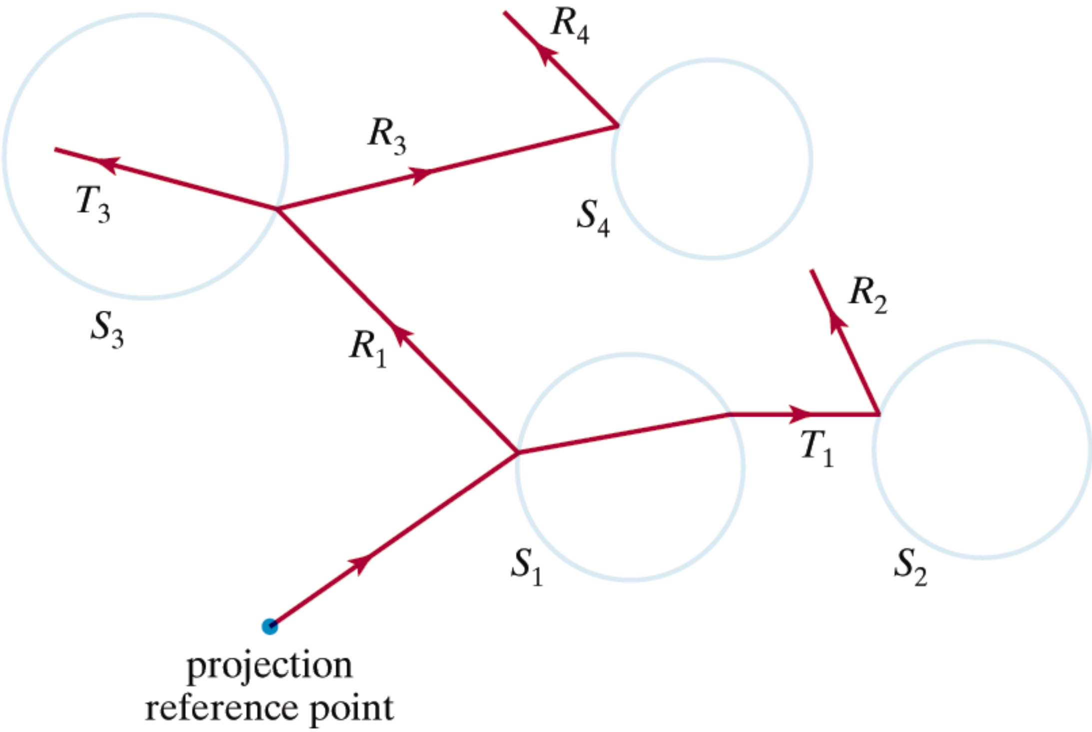
Backward Ray Tracing

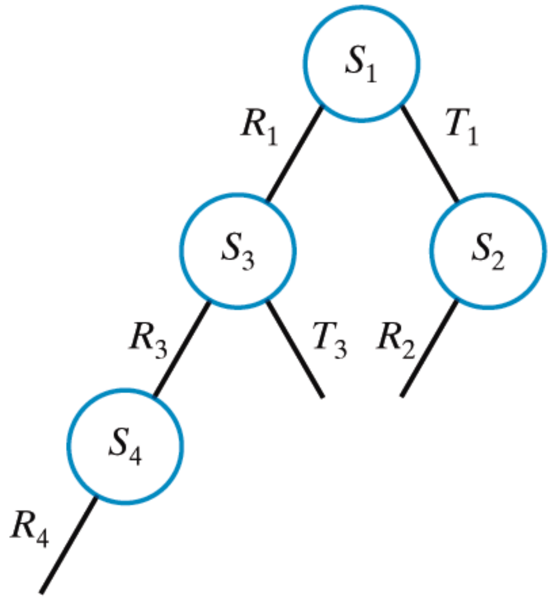
- Illumination
  - Phong illumination
  - Shadow rays
  - Specular reflection
  - Specular refraction



- Specular reflection and refraction are recursive

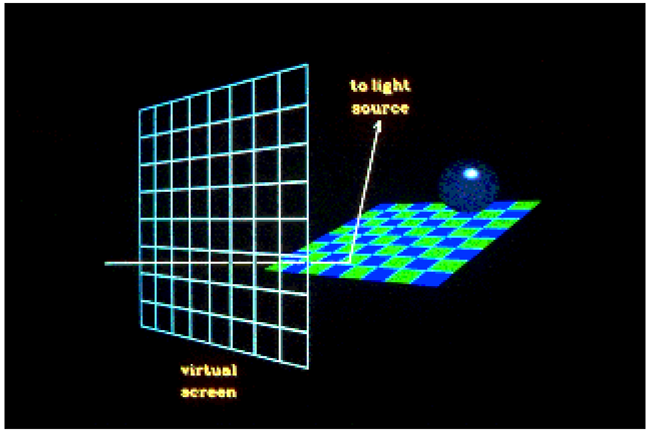
Binary Ray-Tracing Tree

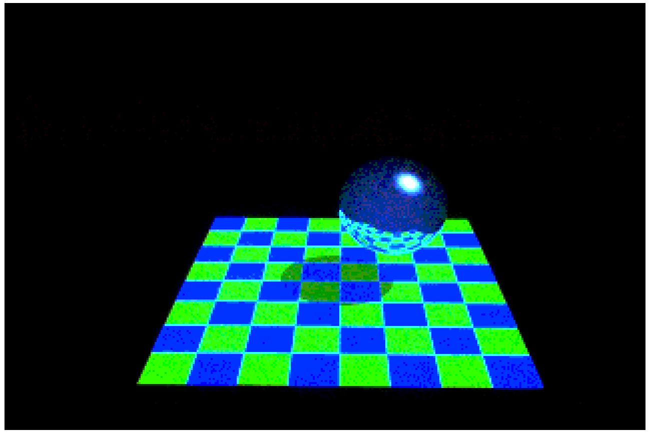




Shadow Rays

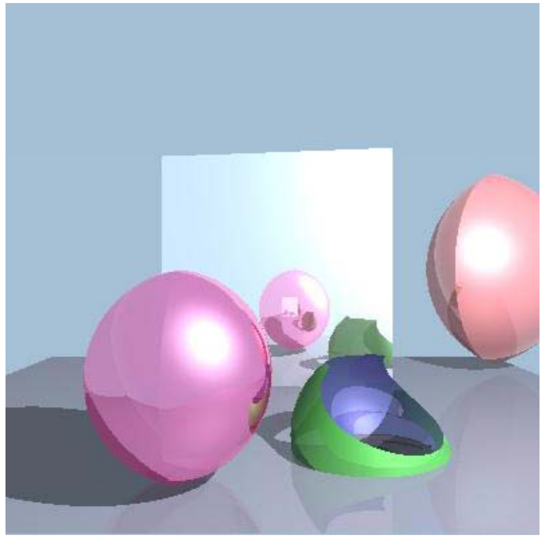
- If the ray hits an object, we want to know if that point on the object is in a shadow. So, when the ray hits an object, a secondary ray, called a "shadow" ray, is shot towards the light sources.
- If this shadow ray hits another object before it hits a light source, then the first intersection point is in the shadow of the second object.
- For a simple illumination model this means that we only apply the ambient term for that light source





http://www.siggraph.org/education/materials/HyperGraph/raytrace

Recursive Ray Tracing



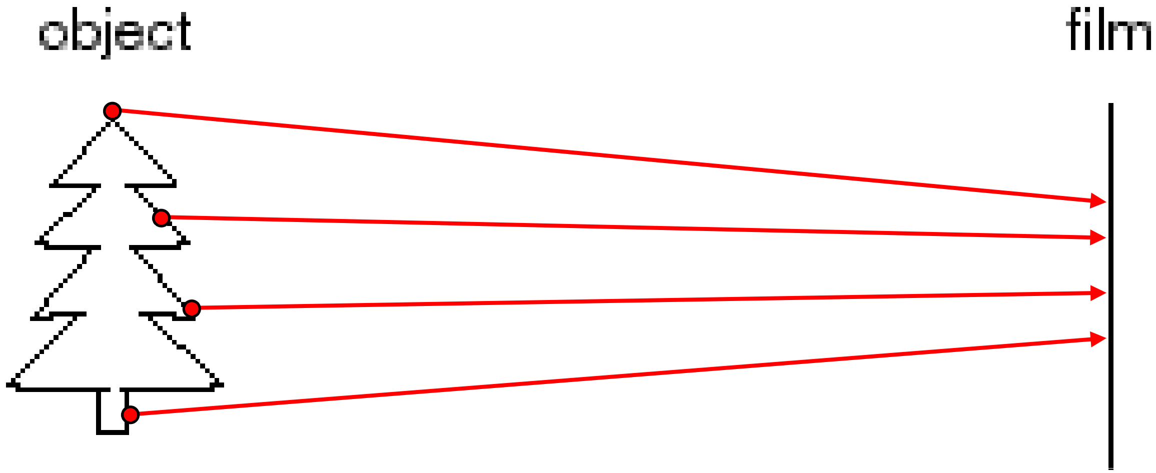
Pros

- very high quality images
- fast for very complex scenes

Cons

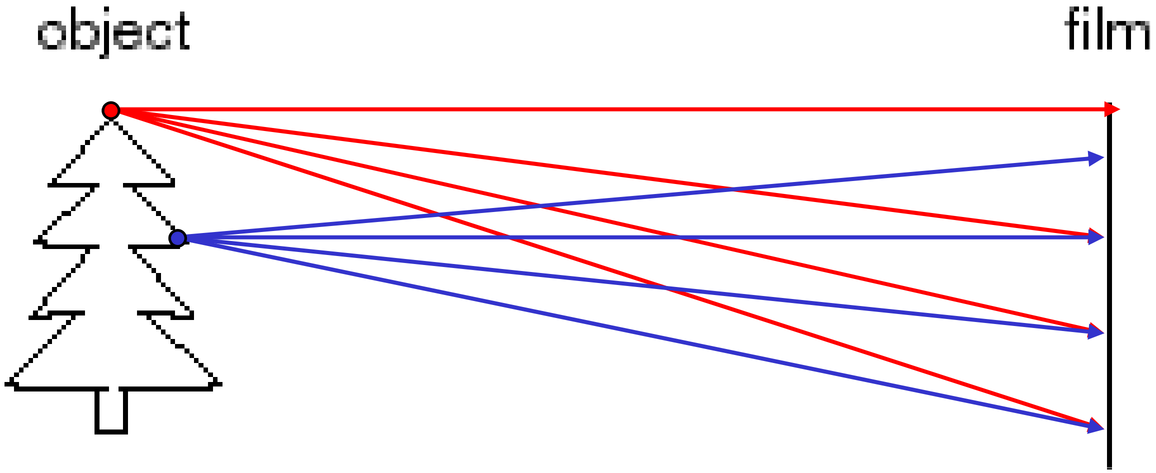
- slow
- complexity hinders hardware implementation

How do we see the world?



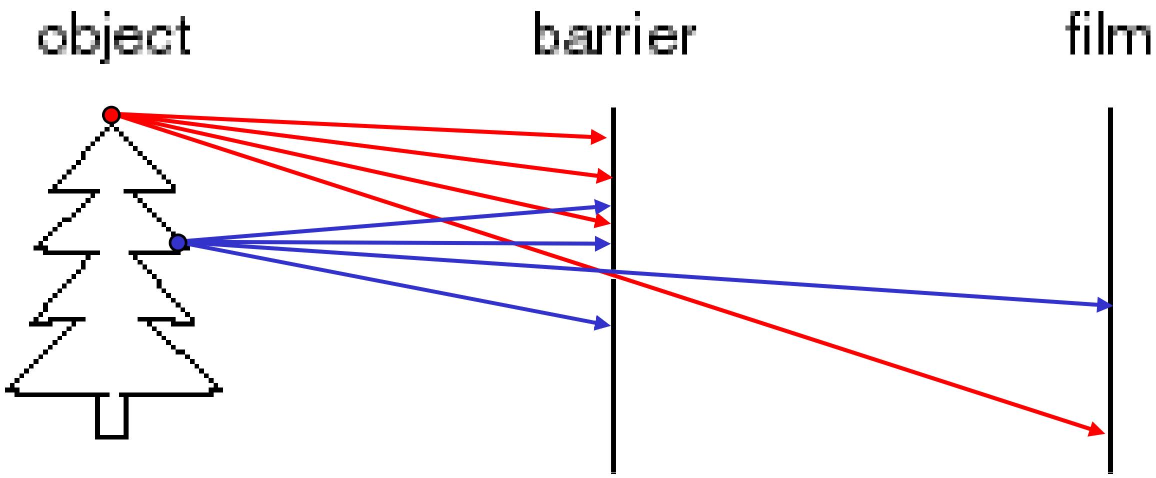
- In computer graphics, we assumed that rays are projected onto the image plane

How do we see the world?



- In physics, that is not true
- Rays are scattered in all directions

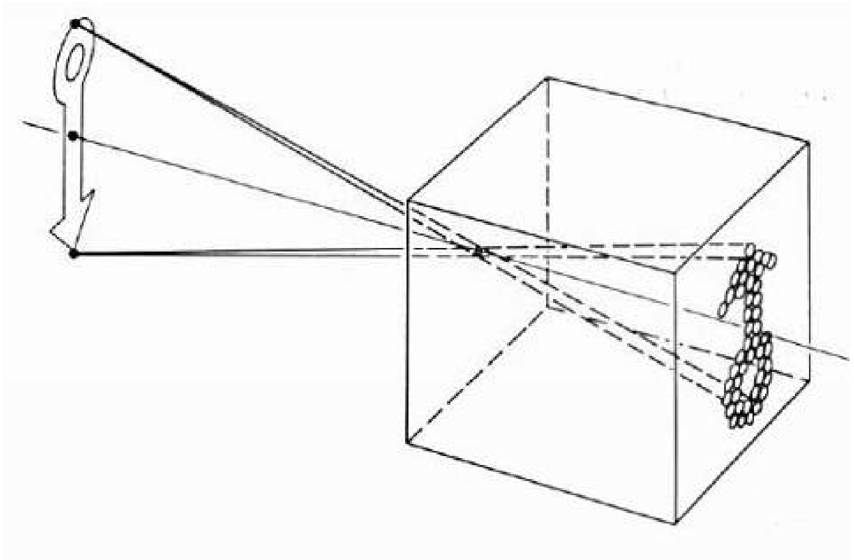
Pinhole camera



- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the **aperture**
  - How does this transform the image?

Slide by Steve Seitz

Pinhole camera model



- Pinhole model:
  - Captures **pencil of rays** – all rays through a single point
  - The point is called **Center of Projection (COP)**
  - The image is formed on the **Image Plane**
  - **Effective focal length *f*** is distance from COP to Image Plane

Slide by Steve Seitz

Problems with Pinholes

- Pinhole size (aperture) must be "very small" to obtain a clear image.
- However, as pinhole size is made smaller, less light is received by image plane.
- If pinhole is comparable to wavelength of incoming light, DIFFRACTION effects blur the image!
- Sharpest image is obtained when:

pinhole diameter $$d = 2\sqrt{f'\lambda}$$

Example: If $f' = 50mm$,

$\lambda = 600nm$ (red)

then $d = 0.36mm$

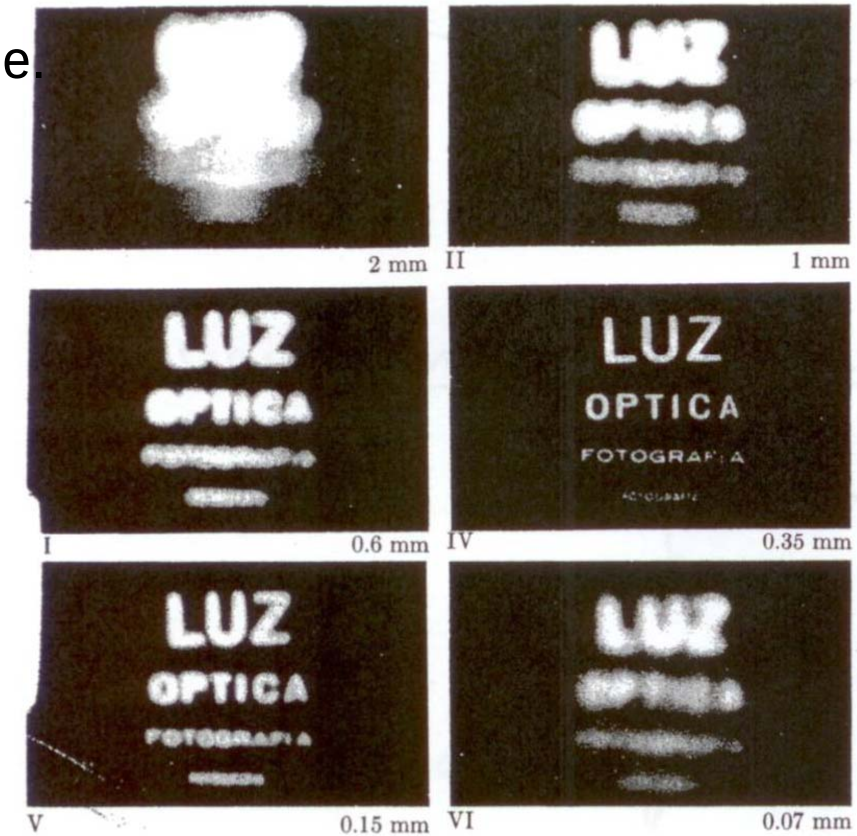
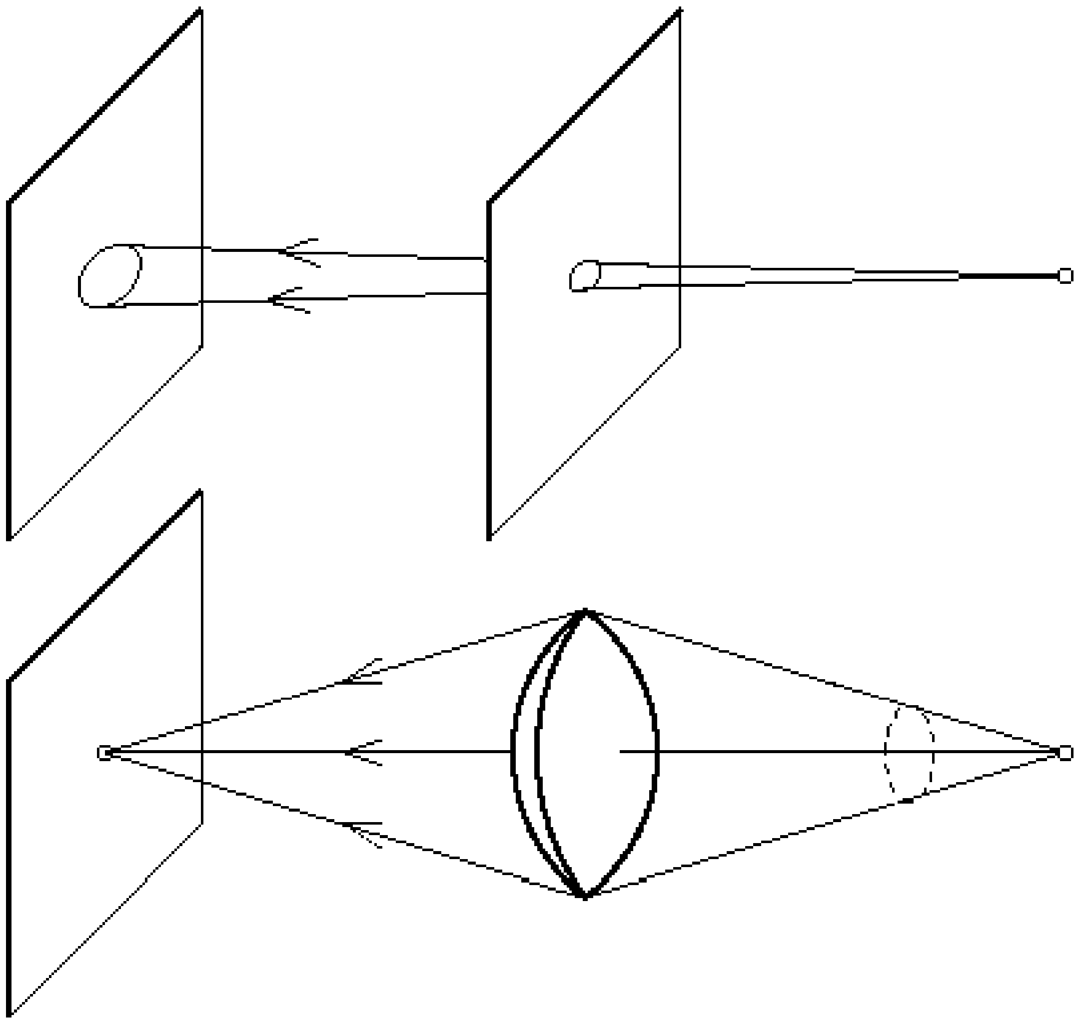


Fig. 5.96 The pinhole camera. Note the variation in image clarity as the hole diameter decreases. [Photos courtesy Dr. N. Joel, UNESCO.]

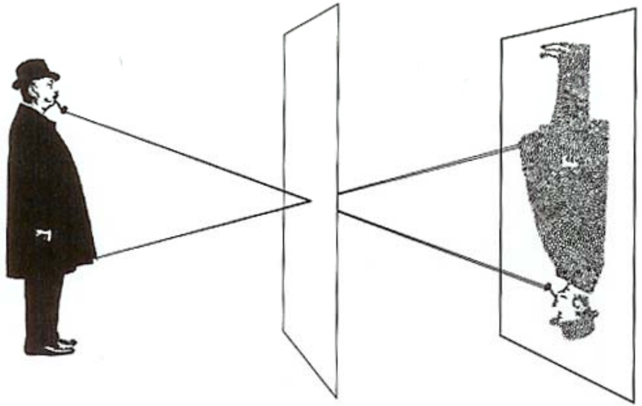
The reason for lenses



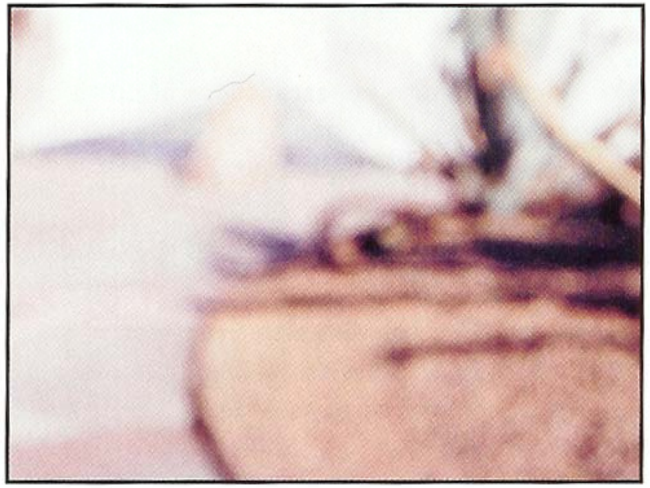
Slide by Steve Seitz

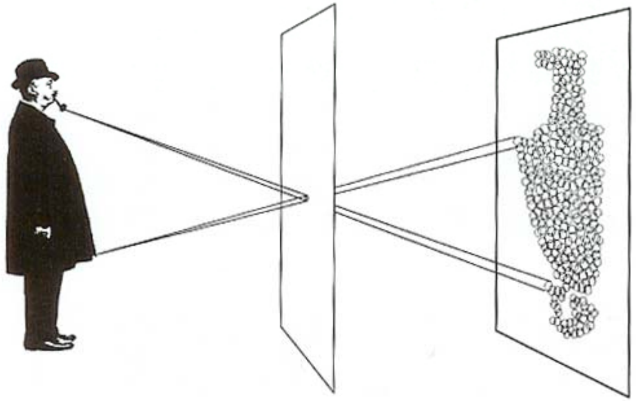
Small vs. Large Pinholes

Photograph made with small pinhole



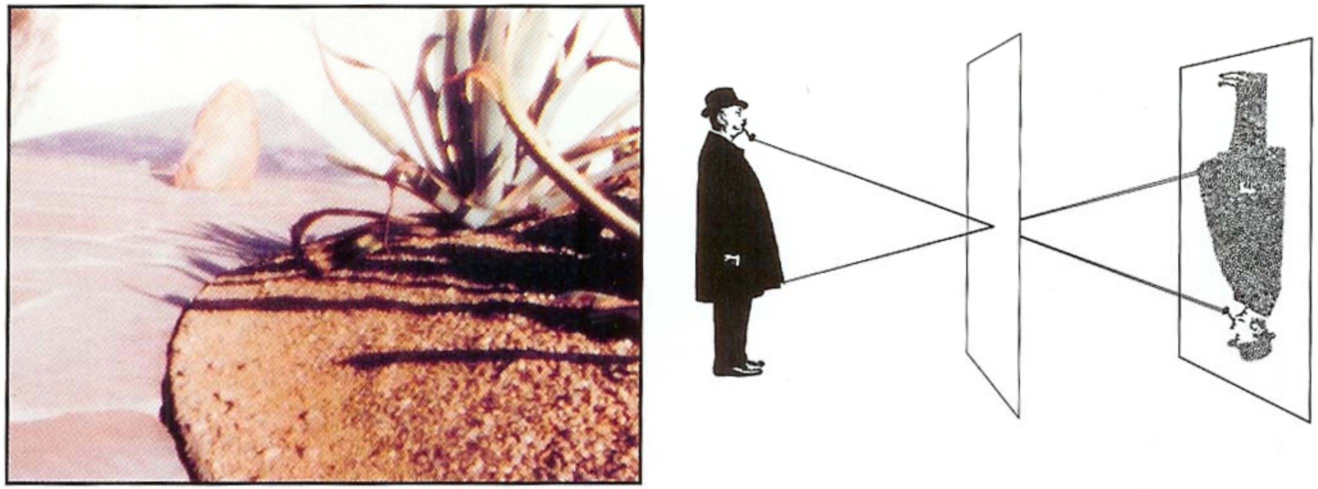
Photograph made with larger pinhole



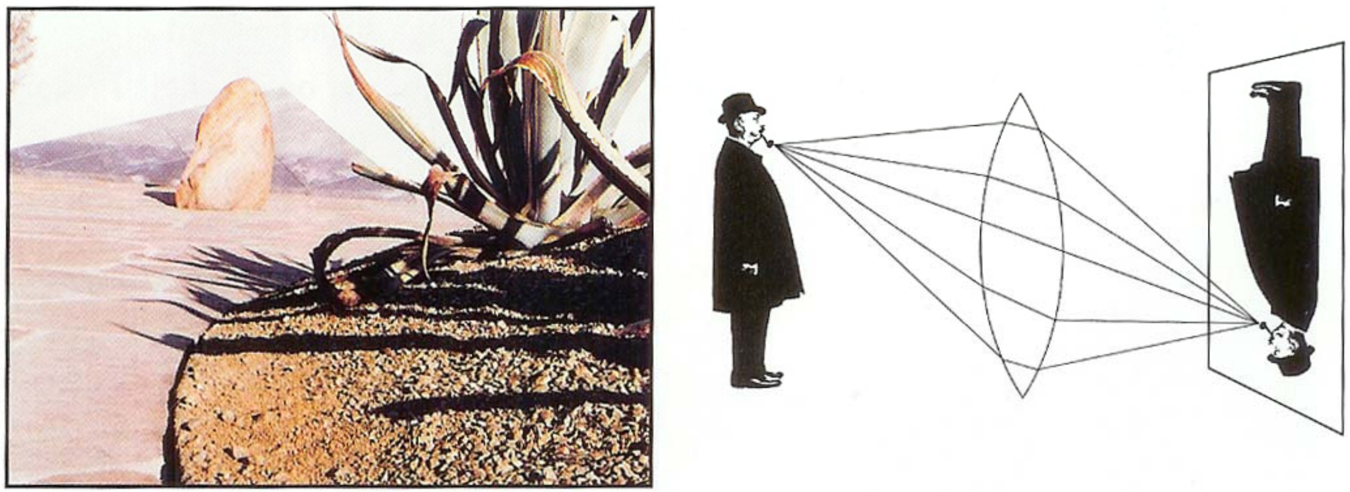


Pinhole vs. Lens

Photograph made with small pinhole

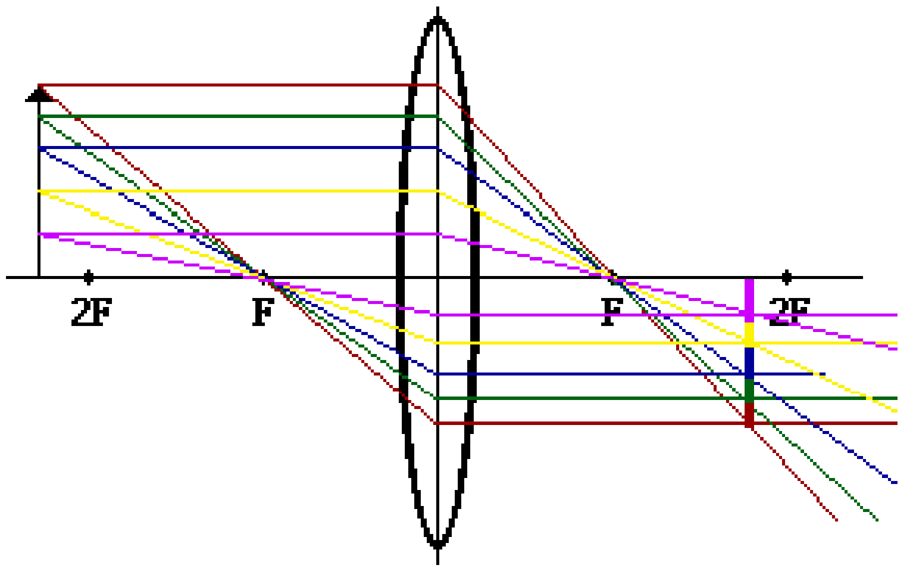


Photograph made with lens



- Ideal Lens: Same projection as pinhole but gathers more light!

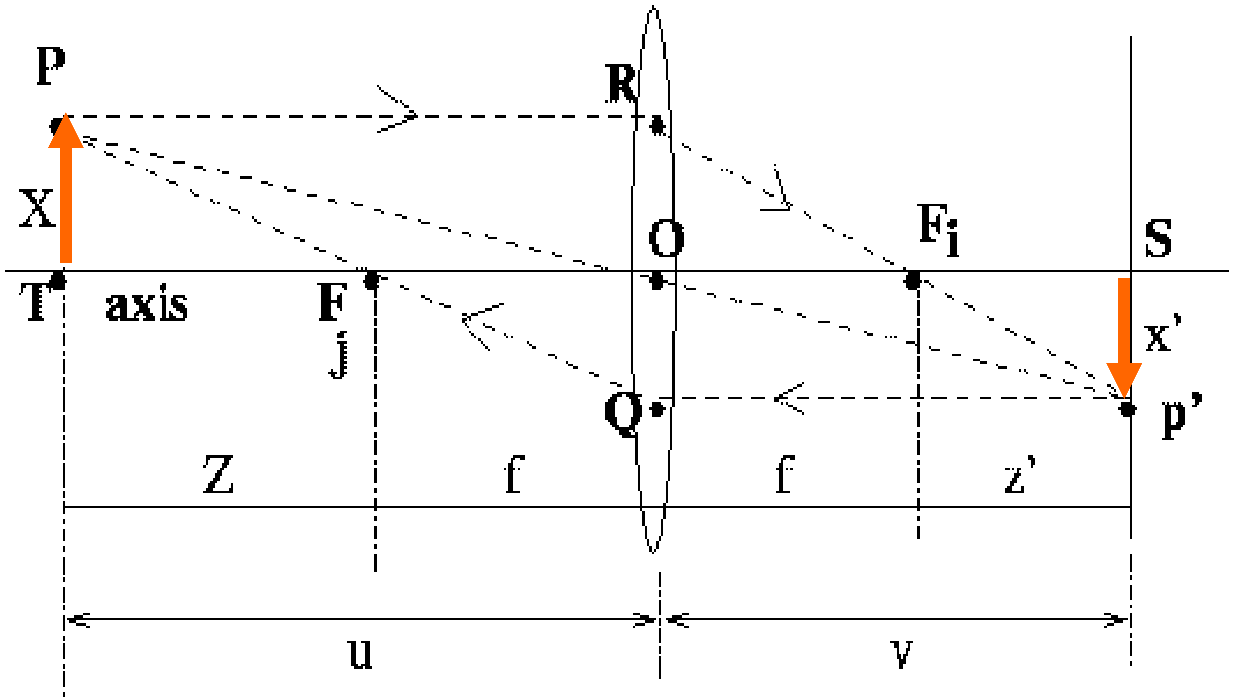
Image Formation using Lenses



When a set of incident and refracted rays are drawn for several points upon a vertical object, each reflected ray intersects at locations which form a vertical line - i.e., the image of a vertical object is an image which is also vertical.

- *F* is the focal length of the lens – determines the lens's ability to bend (refract) light

Thin lens equation relates object depth to image plane via *f*



For world point P in focus, then the thin lens equation is: $1/f = 1/u + 1/v$

Derivation of thin lens equation from geometry

Distance X is the same as the distance from **R** to **O**, similar triangles $\mathbf{ROF_i}$ and $\mathbf{Sp'F_i}$ give the following equation.

$$\frac{X}{f} = \frac{x'}{z'} \tag{5}$$

Using similar triangles **POT** and **p′OS** we obtain a second equation.

$$\frac{X}{f+Z} = \frac{x'}{f+z'} \tag{6}$$

Substituting the value of X from the first equation into the second yields

$$f^2 = Zz' \tag{7}$$

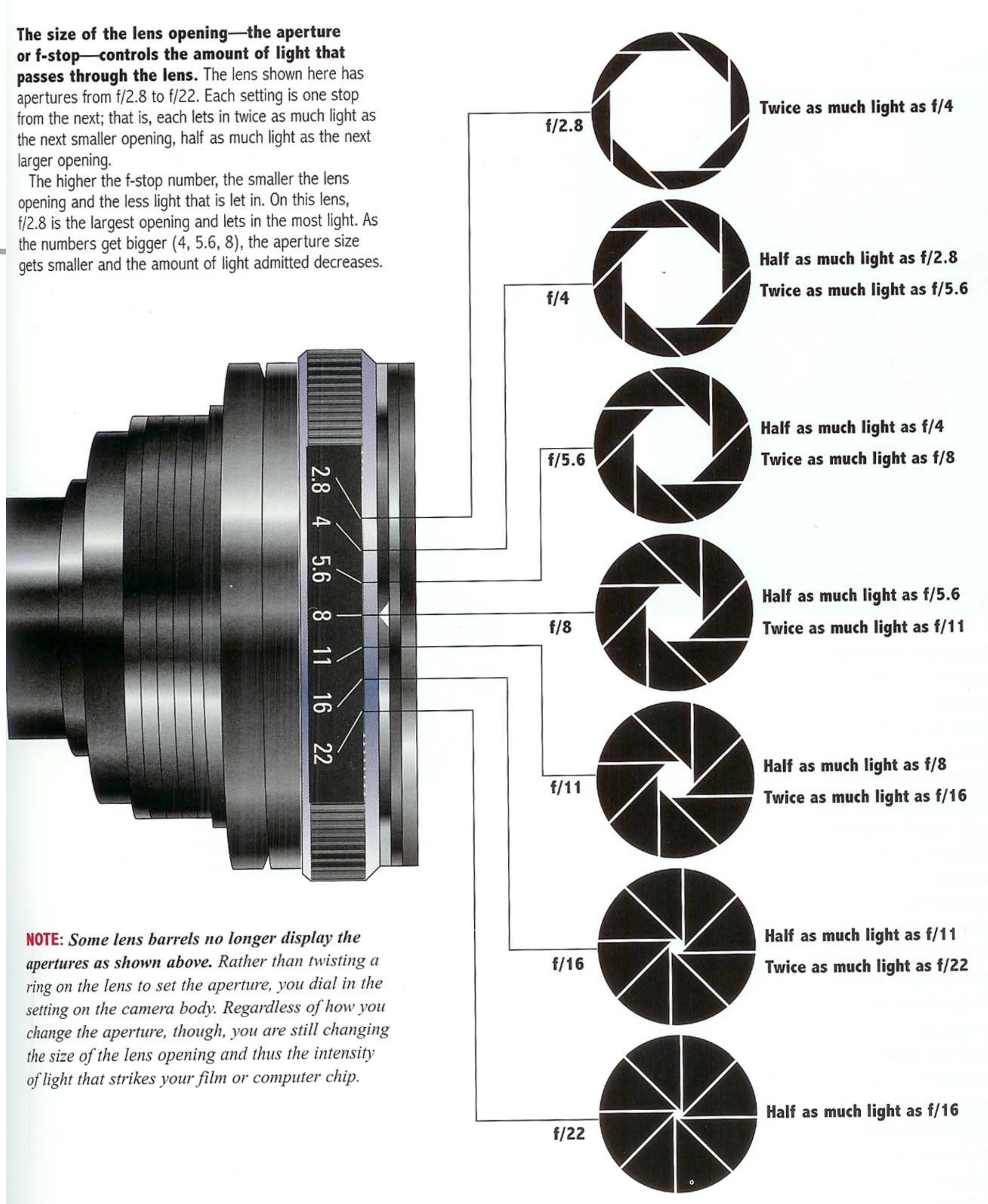
Substituting $u - f$ for Z and $v - f$ for z' yields

$$uv = f(u+v) \tag{8}$$

and finally dividing both sides by (uvf) yields the most common form of the lens equation, which relates the focal length to the object distance u from the lens center and the image distance v from the lens center.

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{9}$$

Aperture



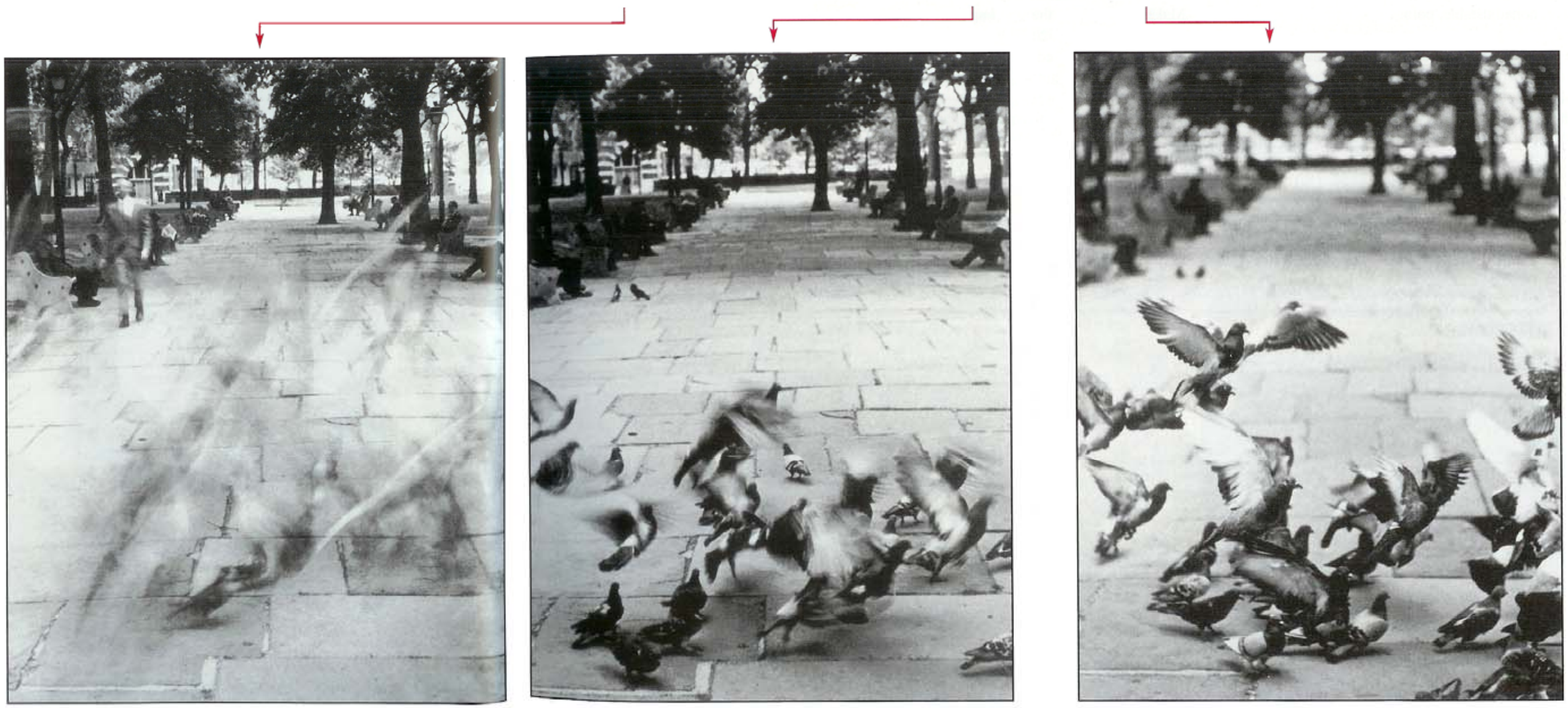
The size of the lens opening—the aperture or f-stop—controls the amount of light that passes through the lens. The lens shown here has apertures from f/2.8 to f/22. Each setting is one stop from the next; that is, each lets in twice as much light as the next smaller opening, half as much light as the next larger opening.

The higher the f-stop number, the smaller the lens opening and the less light that is let in. On this lens, f/2.8 is the largest opening and lets in the most light. As the numbers get bigger (4, 5.6, 8), the aperture size gets smaller and the amount of light admitted decreases.

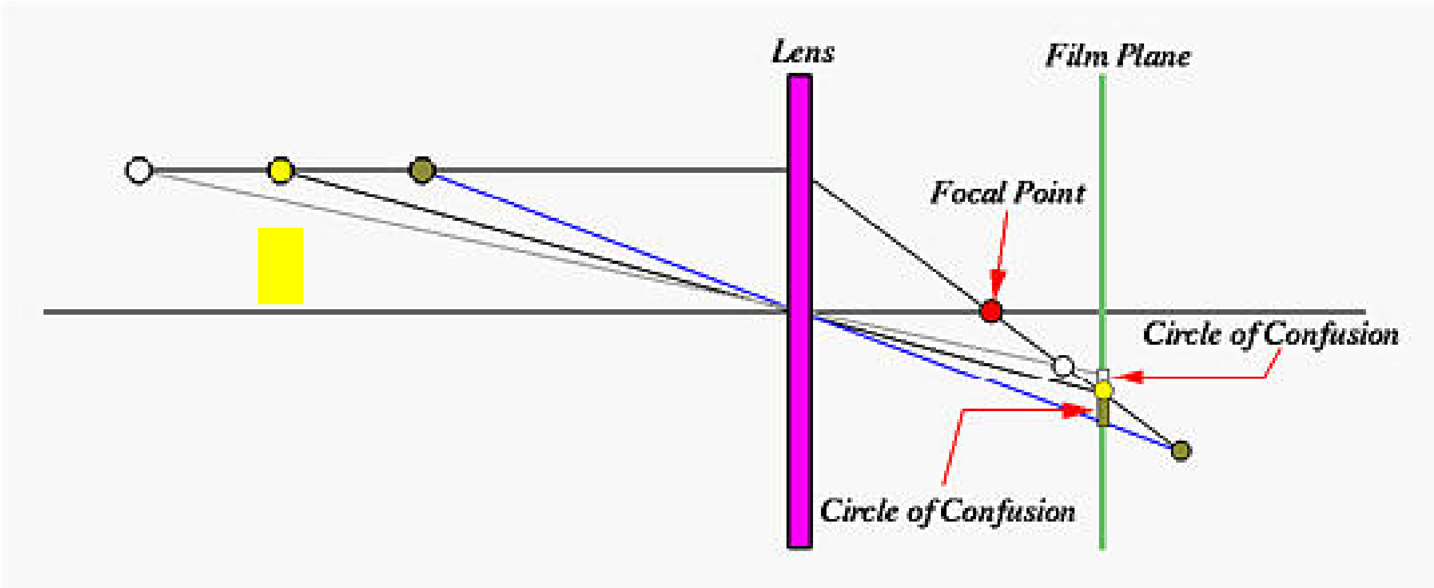
NOTE: ***Some lens barrels no longer display the apertures as shown above.*** *Rather than twisting a ring on the lens to set the aperture, you dial in the setting on the camera body. Regardless of how you change the aperture, though, you are still changing the size of the lens opening and thus the intensity of light that strikes your film or computer chip.*

Shutter

Equivalent exposures. *Each combination here of f-stop and shutter speed produces the equivalent exposure (lets in the same amount of light) but produces differences in depth of field and motion.*



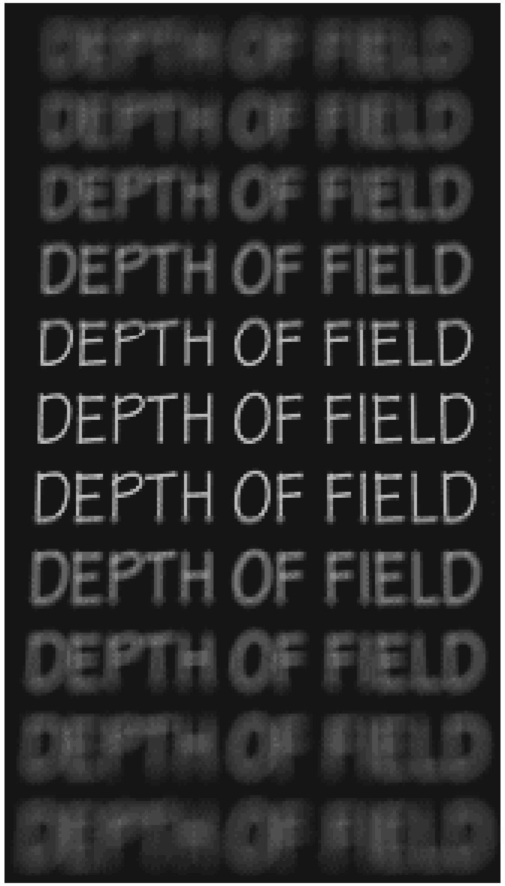
Depth of Field



- Lens focuses on the yellow dot
- The image of the white dot is a circle of confusion
- Circles of confusion are actually out of focus images of subjects
- Smaller (*resp.*, larger) circles of confusion will be formed if less (*resp.*, more) light can pass through

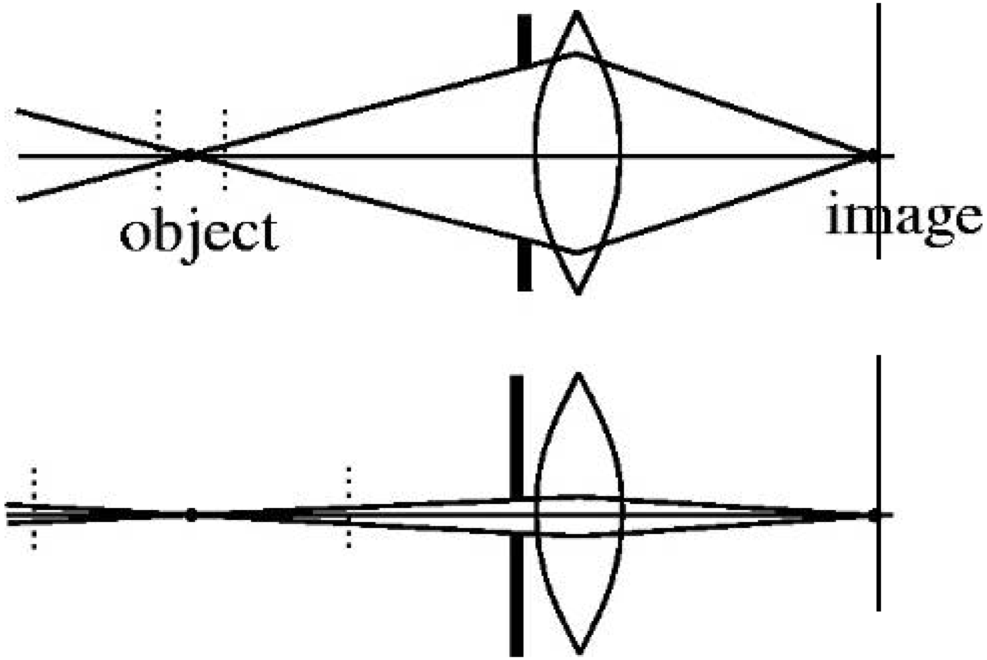
  ← aperture values

Depth of Field



http://www.cambridgeincolour.com/tutorials/depth-of-field.htm

Aperture controls Depth of Field



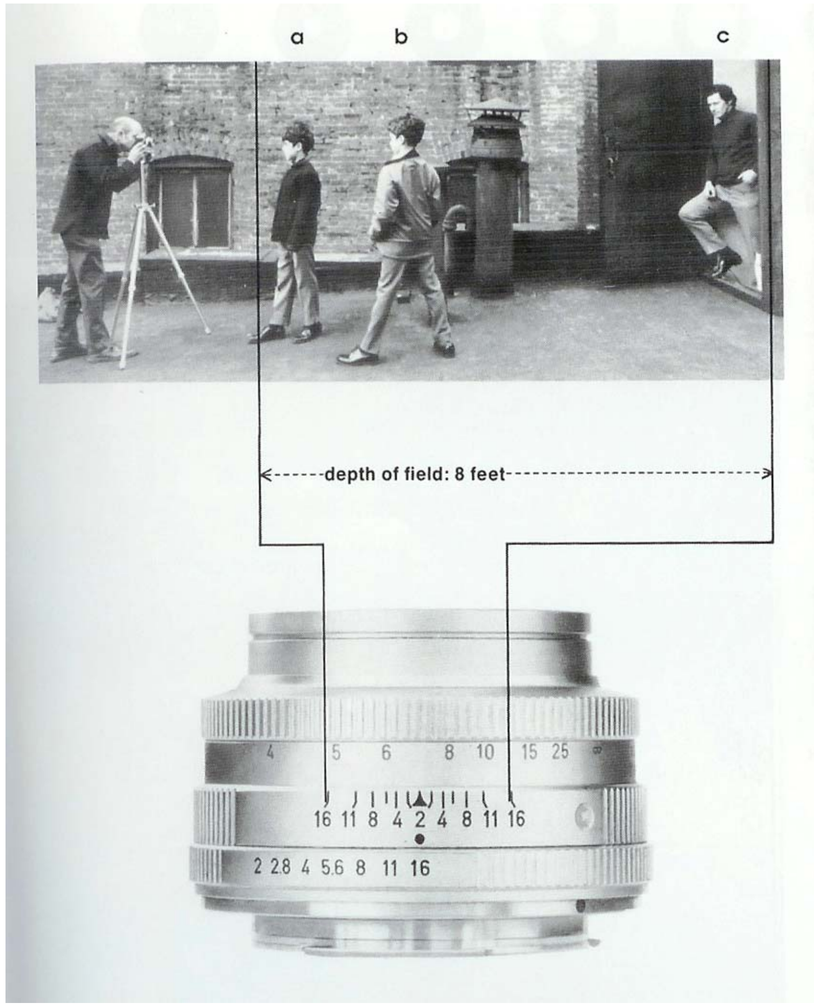
- Changing the aperture size affects the depth of field
  - A smaller aperture increases the range in which the object is approximately in focus
  - But small aperture reduces amount of light – need to increase exposure

Varying the aperture

LARGE APERTURE, LESS DEPTH OF FIELD

F/2.0

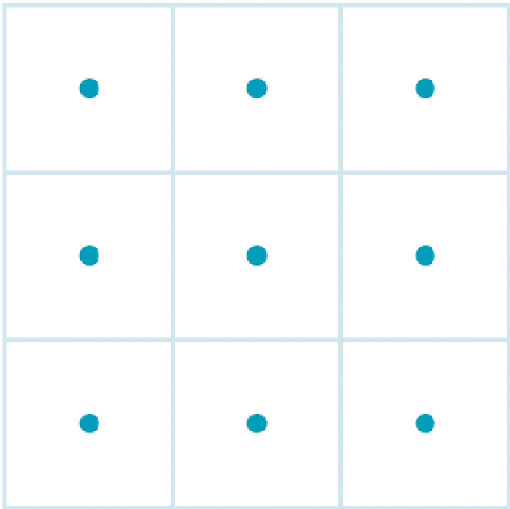
Varying the aperture

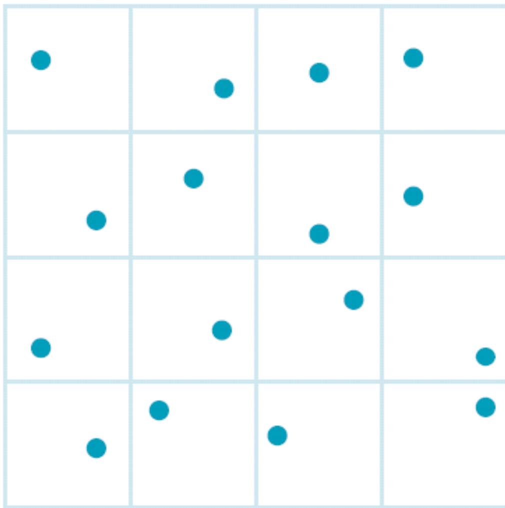


F/16.0

Antialising – Super sampling

- Antialiasing
  - Oversampling rays in each pixel
  - Regular vs. jittered sampling





Distributed Ray Tracing

- Stochastic sampling that randomly distribute rays according to the various parameters
- Monte Carlo evaluation of the multiple integrals that occur in an accurate physical description of surface lighting

Distributed Ray Tracing

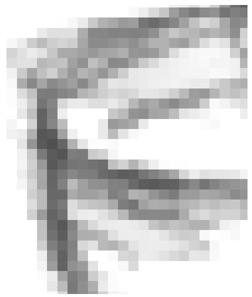
- Depth of Field
  - Distributing rays over the circle of confusion

Distributed Ray Tracing

- Motion blur
  - Distributing rays over time

Halftoning

- Simulate the look of a continuous tone image with limited colors
- How to make gray using tiny dots of black ink

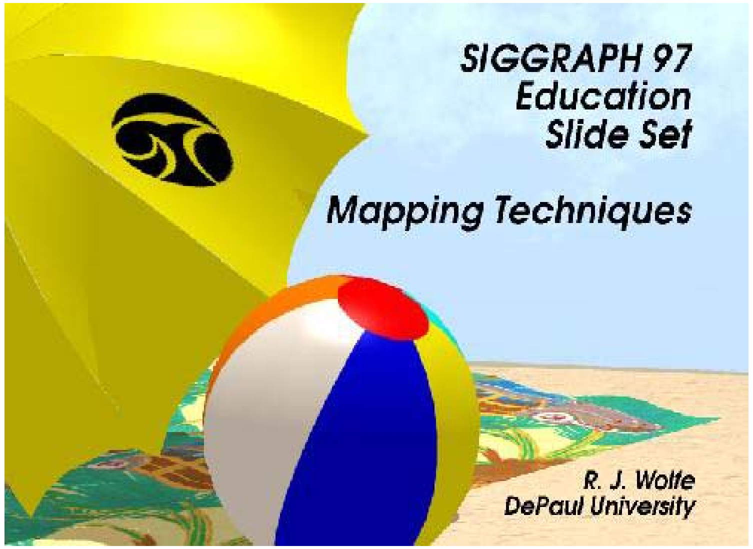


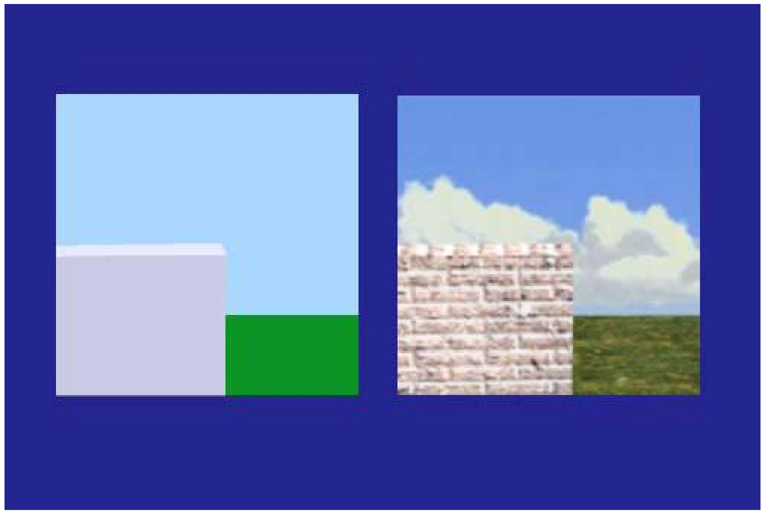
Halftoning

- for displaying intensity levels more than display device capability (clustered-dot ordered dither)
- intensity $\rightarrow n \times n$ spatial frequency of pixels or varying dot size
- Grid $\Rightarrow$ $n^2(above\,0) + 1(0)$ intensity levels for bilevel display devices
- symmetrical pixel arrangements are to be avoided
- multi-level device : increase the number of intensity level
- spatial resolution vs. # of intensity level tradeoff

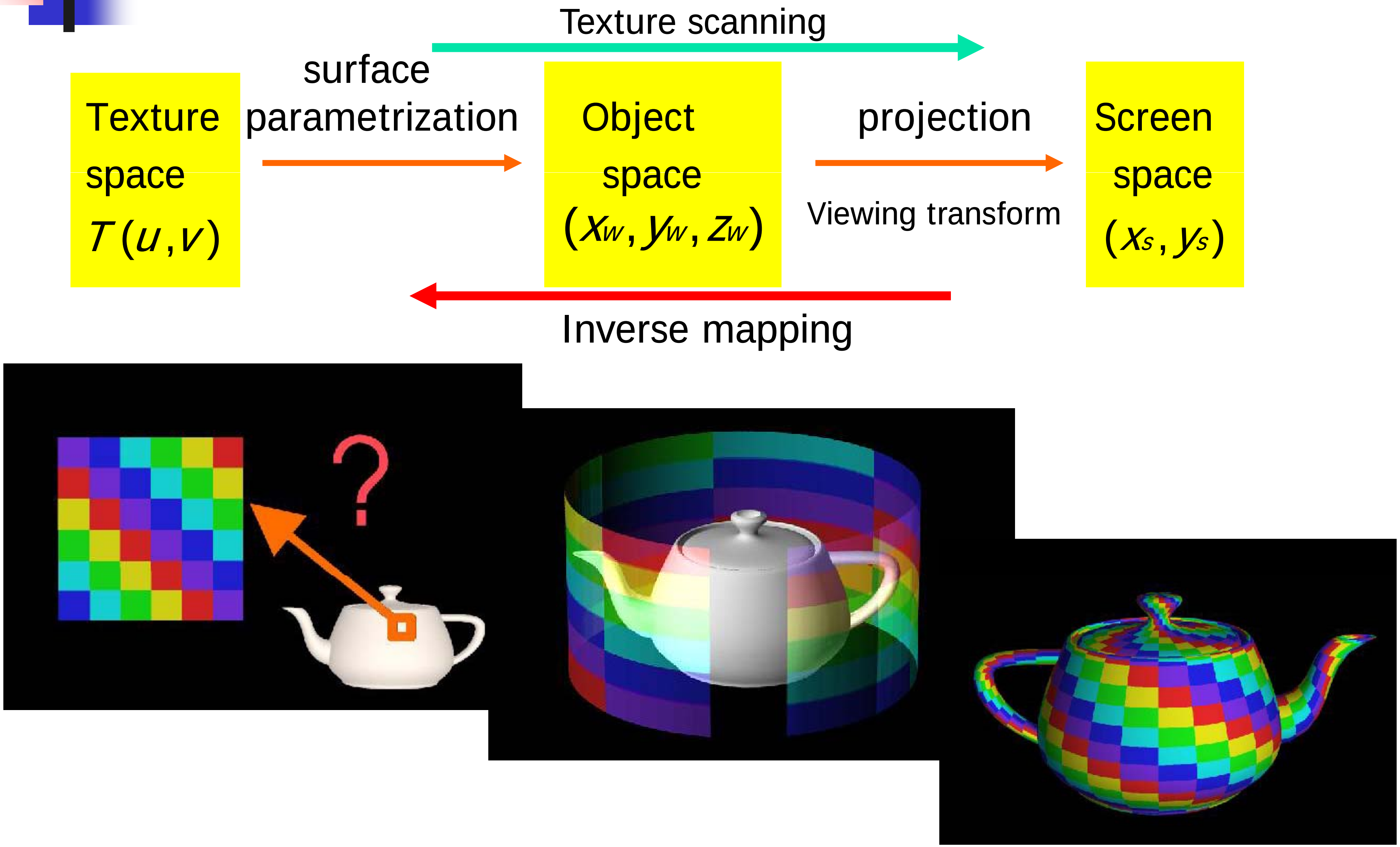
Texture Mapping

- Mapping techniques add realism and interest to computer graphics images.
- Texture mapping applies a pattern of color to an object - umbrella, background, beach ball, beach blanket.
- Bump mapping alters the surface of an object so that it appears rough, dented or pitted - sand



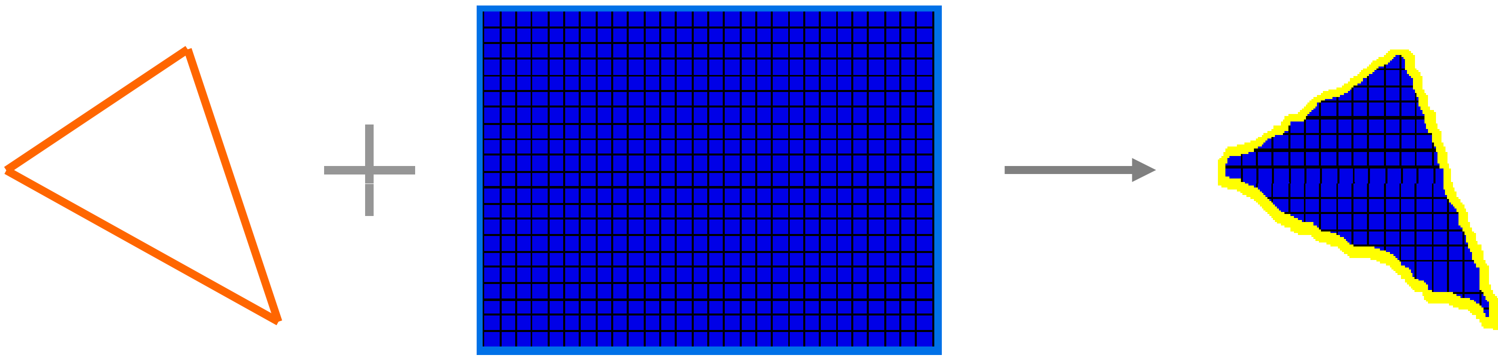


Texture Mapping



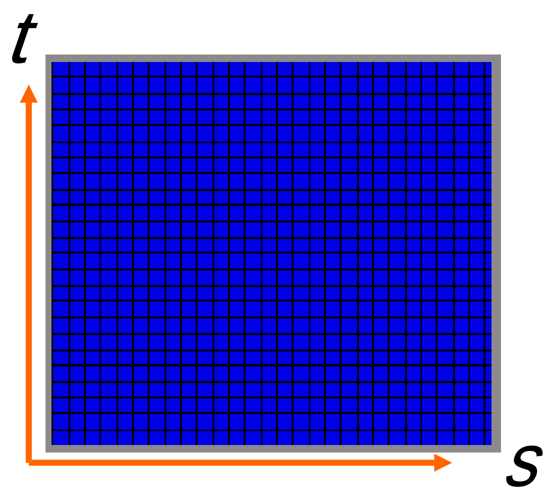
Non-Parametric Texture Mapping

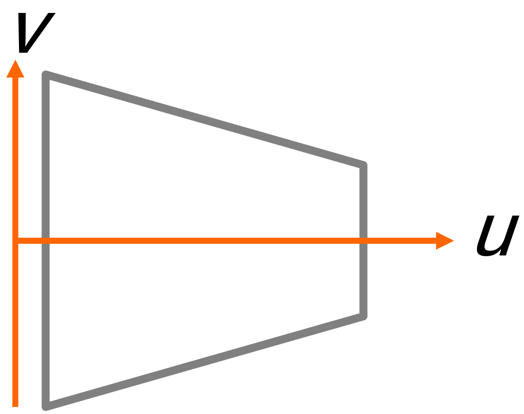
- Gives cookie-cutter effect
- Texture size and orientation fixed, not tied to the size and orientation of the polygon



Parametric Texture Mapping

- Separate texture space and screen space.
- Texture the polygon as before, but in texture space.
- Deform the textured polygon into screen space.





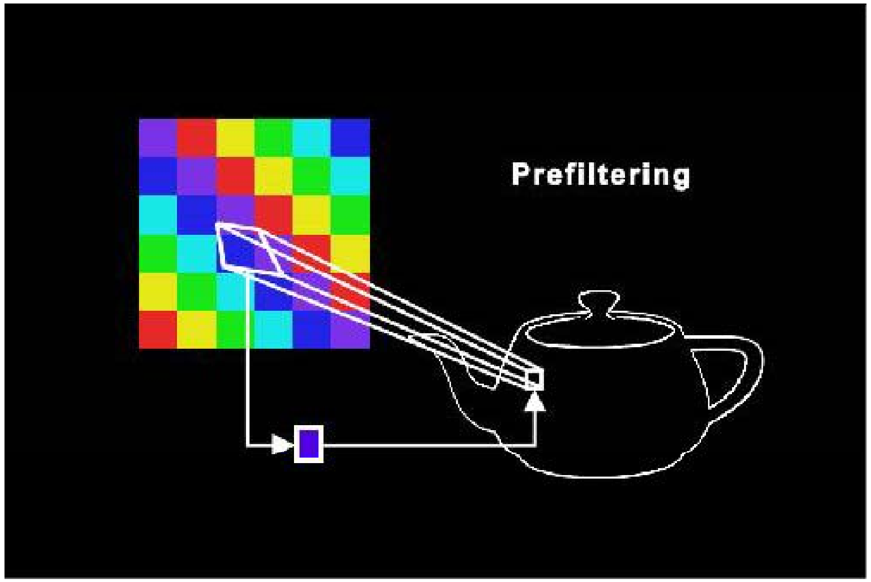
Parametric Texture Mapping

Two major difficulties : inventing a suitable surfaces;
parameterization and antialiasing

Use inverse mapping : find ‘pre-image’ of the current pixel in the texture domain (Fig. 10-106)

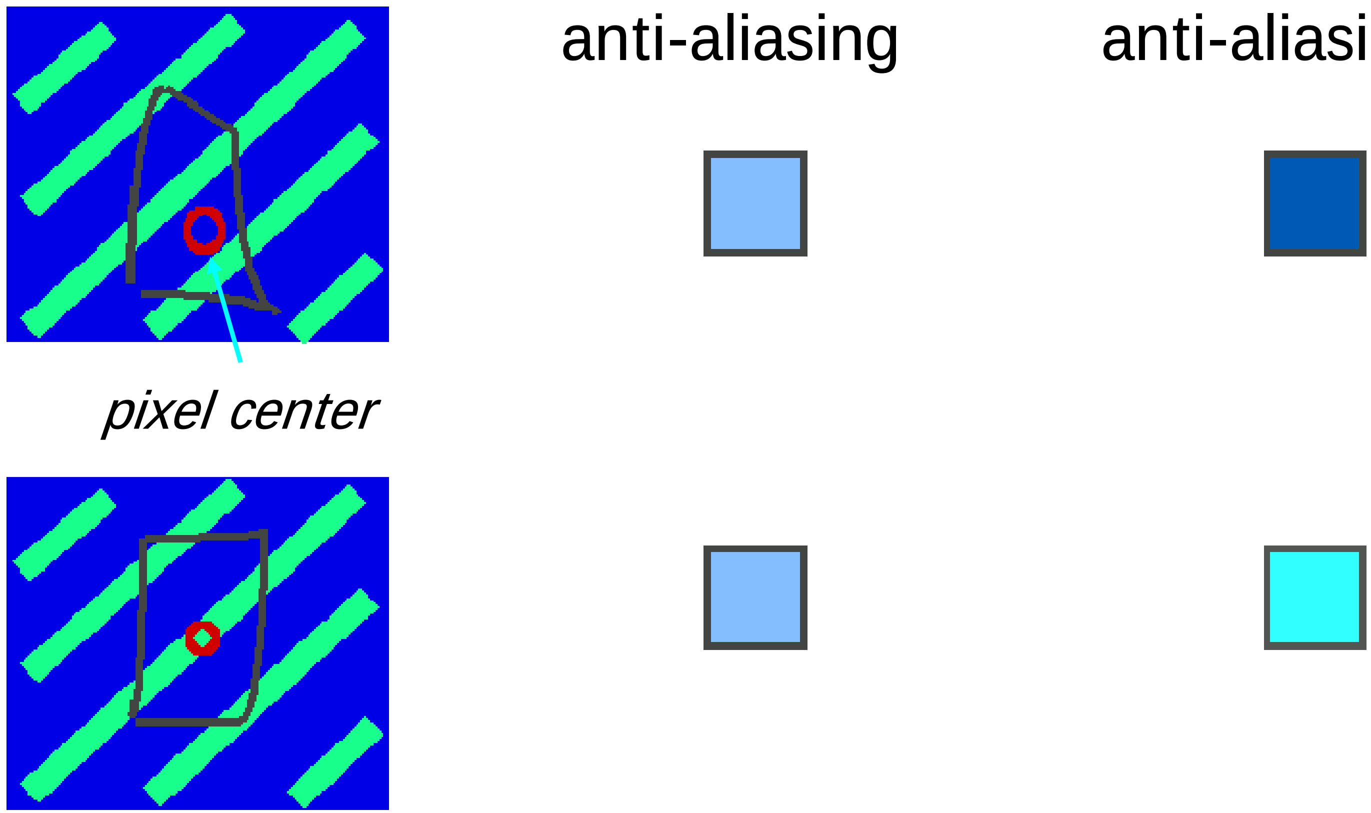
1. take the four pixel corner points
2. invert the object to screen space transformation
3. invert the surface parameterization

Object size decreases ⇒
pre-image of a pixel in texture space increases
: need anti-aliasing



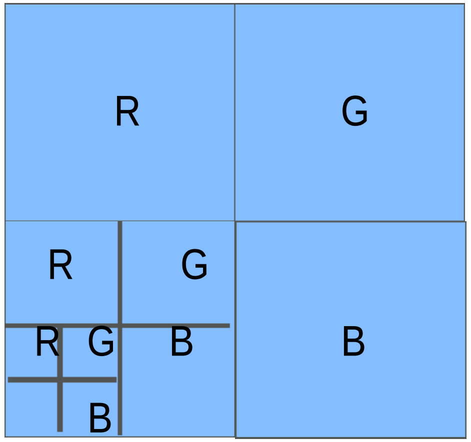
Aliasing and Antialiasing

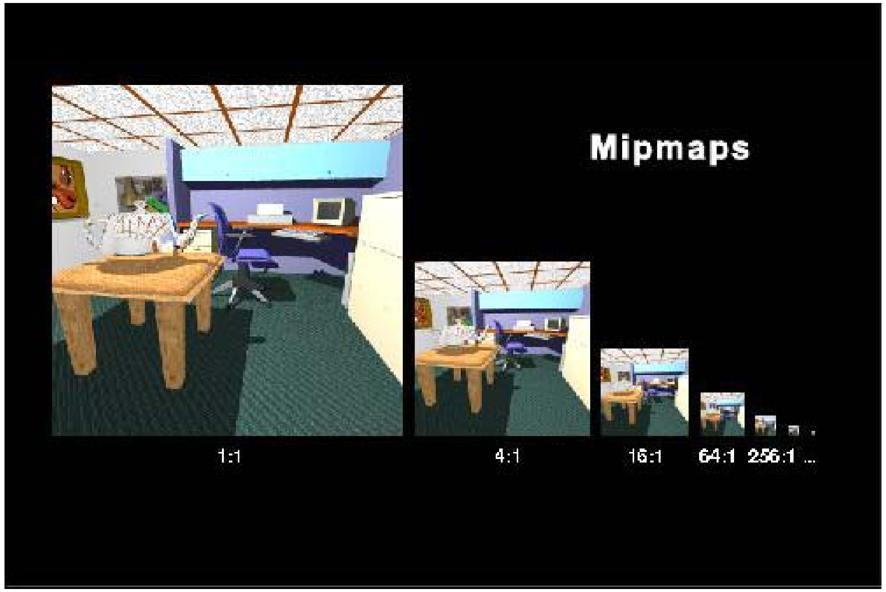
pre-image | pixel with anti-aliasing | pixel w/o anti-aliasing



Texture Filtering Technique

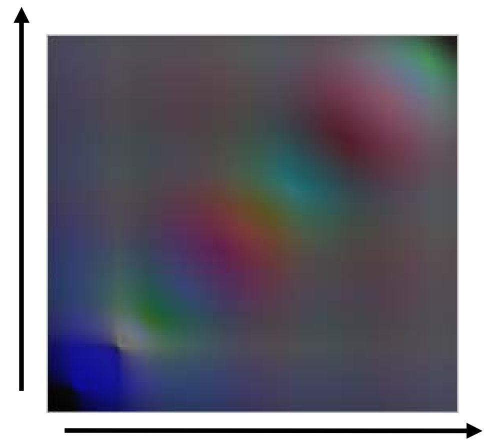
- Method I: Brute Force
  - Just sum
- Method II: Mip Maps
  - a sequence of textures prefiltered at multiple resolutions
  - Lance Willims, ’83
  - stands for multum in parvo, that is many things in a small place
  - makes distant objects look better

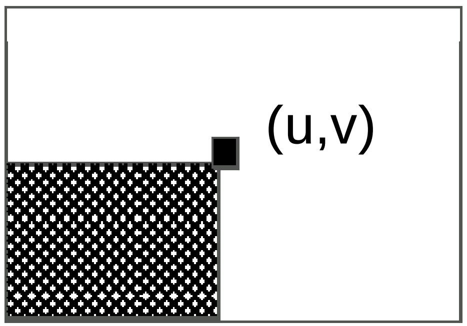


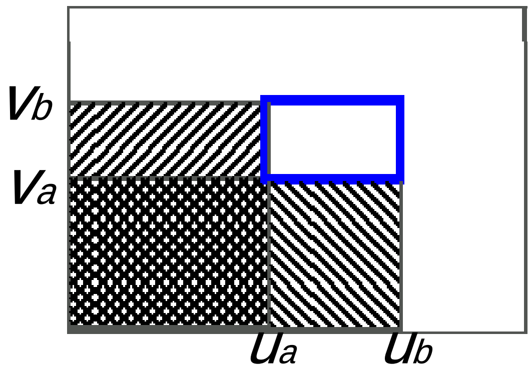


Texture Filtering Technique

- Method III: Summed Area Tables
  - Frank Crow, 1984
  - Keep sum of everything below and to the left
  - Uses four table lookup
  - Requires more memory
  - Can give excessively blurry textures







$$= T(u_b, v_b) - T(u_a, v_b) - T(u_b, v_a) + T(u_a, v_a)$$

Texture Filtering Technique

- Method IV: Stochastic Sampling
  - Gives noisy images, unless enough samples are taken
  - An extremely simple and robust technique
  - Poisson
    - Completely random
    - Add points at random until area is full.
    - Uniform distribution: some neighboring samples close together, some distant

Texture Filtering Technique

- Poisson Disc
  - Poisson distribution, with minimum-distance constraint between samples
  - Add points at random, removing again if they are too close to any previous points
  - Very even-looking distribution
- Jittered
  - Start with regular grid of samples
  - Perturb each sample slightly in a random direction
  - More “clumpy” or granular in appearance

Compositing

Composite different images possibly rendered using different techniques into one image

- Overlay where not zero
  - The first image is overlayed onto the second one where the second image is not black
- Alpha blending
  - The blending of two colors based on a transparency value. Each pixel of both images contains an "alpha" value of the image opacity.

Compositing

- Chroma-keying
  - real time video operation
  - overlay when not a predetermined color method
  - e.g., the weather man in front of a flat blue wall