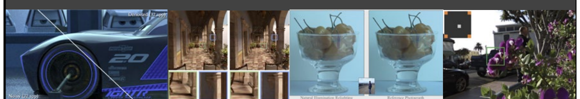
Sampling and Reconstruction of Visual Appearance: From Denoising to View Synthesis

CSE 274 [Fall 2021], Lecture 2

Ravi Ramamoorthi

http://www.cs.ucsd.edu/~ravir



Motivation: BRDFs, Radiometry

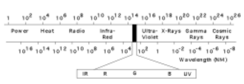
- Basics of Illumination, Reflection
- Formal radiometric analysis (not ad-hoc)
- Reflection Equation
- Ray Tracing and Rendering Equation 2nd half
- Monte Carlo Rendering next week

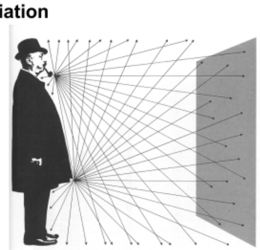
- Appreciate formal analysis in a graduate course, even if not absolutely essential in practice
- *Please e-mail re papers you want to present (by Th)*

Light

Visible electromagnetic radiation

Power spectrum





From London and Upton

Polarization

Photon (quantum effects)

Wave (interference, diffraction)

CS348B Lecture 4 Pat Hanrahan, 2009

Radiometry

- Physical measurement of electromagnetic energy

- *Measure spatial (and angular) properties of light*
  - *Radiance, Irradiance*
  - Reflection functions: Bi-Directional Reflectance Distribution Function or BRDF
  - Reflection Equation
  - Simple BRDF models

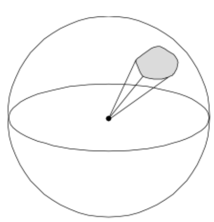
Angles and Solid Angles

- **Angle** $\theta = \frac{l}{r}$

  $\Rightarrow$ circle has 2π radians

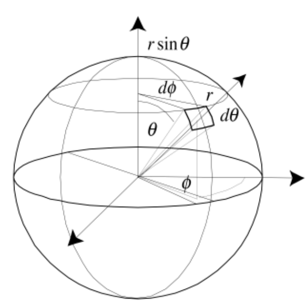
- **Solid angle** $\Omega = \frac{A}{R^2}$

  $\Rightarrow$ sphere has 4π steradians



CS348B Lecture 4 Pat Hanrahan, 2009

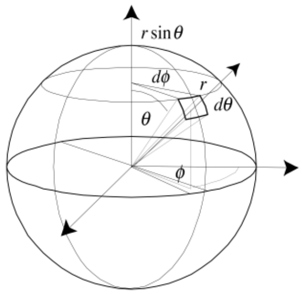
Differential Solid Angles



$$dA = (r\,d\theta)(r\sin\theta\,d\phi)$$
$$= r^2 \sin\theta\,d\theta\,d\phi$$

CS348B Lecture 4 Pat Hanrahan, 2009

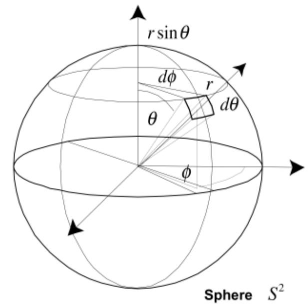
Differential Solid Angles



$$dA = (r\,d\theta)(r\sin\theta\,d\phi) = r^2\sin\theta\,d\theta\,d\phi$$

$$d\omega = \frac{dA}{r^2} = \sin\theta\,d\theta\,d\phi$$

Differential Solid Angles



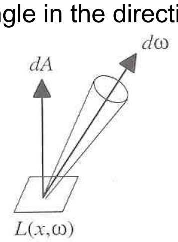
$$d\omega = \sin\theta\,d\theta\,d\phi$$

$$\Omega = \int_{S^2} d\omega = \int_0^{\pi}\int_0^{2\pi} \sin\theta\,d\theta\,d\phi = \int_{-1}^{1}\int_0^{2\pi} d\cos\theta\;d\phi = 4\pi$$

Sphere S^2

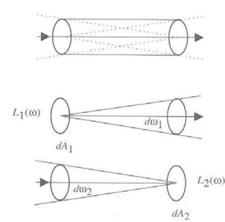
Radiance

- Power per unit projected area perpendicular to the ray per unit solid angle in the direction of the ray
- Symbol: L(x,ω) (W/m^2 sr)
- Flux given by
  dΦ = L(x,ω) cos θ dω dA



Radiance properties

- Radiance constant as propagates along ray
  - Derived from conservation of flux
  - Fundamental in Light Transport.



$$d\Phi_1 = L_1 d\omega_1 dA_1 = L_2 d\omega_2 dA_2 = d\Phi_2$$

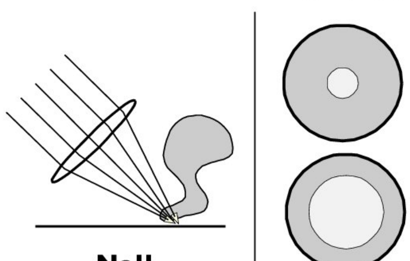
$$d\omega_1 = dA_2/r^2 \qquad d\omega_2 = dA_1/r^2$$

$$d\omega_1 dA_1 = \frac{dA_1 dA_2}{r^2} = d\omega_2 dA_2$$

$$\therefore L_1 = L_2$$

Quiz

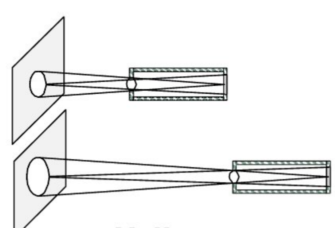
Does radiance increase under a magnifying glass?



No!!

Quiz

Does the brightness that a wall appears to the eye depend on the distance of the viewer to the wall?



No!!

Radiance properties

- Sensor response proportional to radiance (constant of proportionality is throughput)
  - Far away surface: See more, but subtends smaller angle
  - Wall equally bright across viewing distances

Consequences
  - Radiance associated with rays in a ray tracer
  - Other radiometric quants derived from radiance

Irradiance, Radiosity

- Irradiance E is radiant power per unit area
- Integrate incoming radiance over hemisphere
  - Projected solid angle (cos θ dω)
  - Uniform illumination: Irradiance = π [CW 24,25]
  - Units: W/m^2
- Radiant Exitance (radiosity)
  - Power per unit area leaving surface (like irradiance)

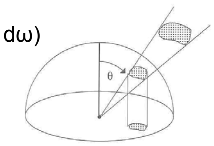
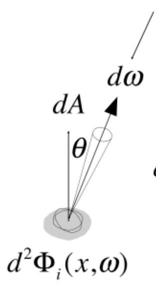


Figure 2.8: Projection of differential area.

Directional Power Arriving at a Surface



$$d^2\Phi_i(x,\omega) = L_i(x,\omega)\cos\theta\, dA\, d\omega$$

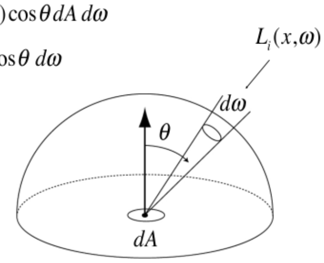
CS348B Lecture 4 — Pat Hanrahan, 2007

Irradiance from the Environment

$$d^2\Phi_i(x,\omega) = L_i(x,\omega)\cos\theta\, dA\, d\omega$$

$$dE(x,\omega) = L_i(x,\omega)\cos\theta\, d\omega$$

Light meter

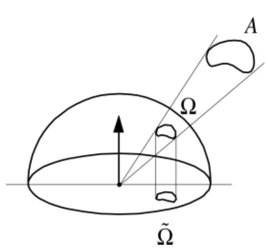


$$E(x) = \int_{H^2} L_i(x,\omega)\cos\theta\, d\omega$$

CS348B Lecture 4 — Pat Hanrahan, 2007

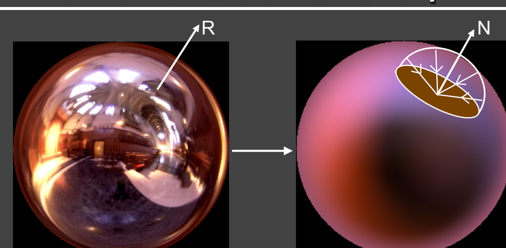
Uniform Area Source

$$E(x) = \int_{H^2} L\cos\theta\, d\omega$$
$$= L\int_{\Omega} \cos\theta\, d\omega$$
$$= L\tilde{\Omega}$$



CS348B Lecture 5 — Pat Hanrahan, 2009

Irradiance Environment Maps



Incident Radiance (Illumination Environment Map) — Irradiance Environment Map

Radiometry

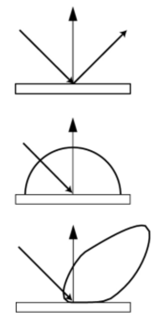
- Physical measurement of electromagnetic energy
- *Measure spatial (and angular) properties of light*
  - Radiance, Irradiance
  - *Reflection functions: Bi-Directional Reflectance Distribution Function or BRDF*
  - *Reflection Equation*
  - Simple BRDF models

Types of Reflection Functions

Ideal Specular
- **Reflection Law**
- **Mirror**

Ideal Diffuse
- **Lambert's Law**
- **Matte**

Specular
- **Glossy**
- **Directional diffuse**

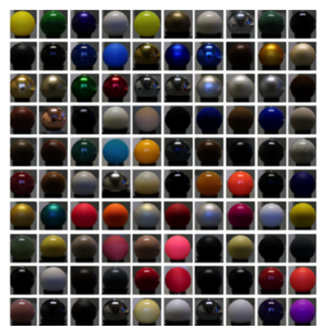


Materials

Plastic **Metal** **Matte**

From Apodaca and Gritz, *Advanced RenderMan*

Spheres [Matusik et al.]

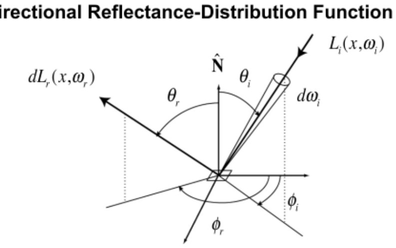


Building up the BRDF

- Bi-Directional Reflectance Distribution Function [Nicodemus 77]
- Function based on incident, view direction
- Relates incoming light energy to outgoing
- Unifying framework for many materials

The BRDF

Bidirectional Reflectance-Distribution Function



$$f_r(\omega_i \to \omega_r) \equiv \frac{dL_r(\omega_i \to \omega_r)}{dE_i} \quad \left[\frac{1}{sr}\right]$$

BRDF

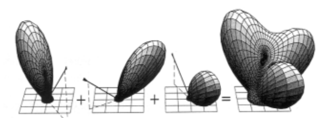
- Reflected Radiance proportional Irradiance
- Constant proportionality: BRDF
- Ratio of outgoing light (radiance) to incoming light (irradiance)
  - Bidirectional Reflection Distribution Function
  - (4 Vars) units 1/sr

$$f(\omega_i,\omega_r) = \frac{L_r(\omega_r)}{L_i(\omega_i)\cos\theta_i d\omega_i}$$

$$L_r(\omega_r) = L_i(\omega_i) f(\omega_i,\omega_r)\cos\theta_i d\omega_i$$

Properties of BRDF's

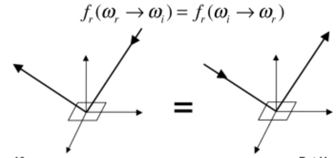
1. Linearity



From Sillion, Arvo, Westin, Greenberg

2. Reciprocity principle

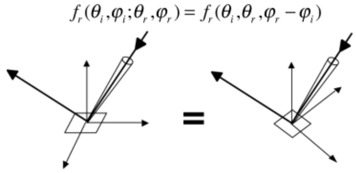
$$f_r(\omega_r \to \omega_i) = f_r(\omega_i \to \omega_r)$$



Properties of BRDF's

3. Isotropic vs. anisotropic

$$f_r(\theta_i,\varphi_i;\theta_r,\varphi_r) = f_r(\theta_i,\theta_r,\varphi_r-\varphi_i)$$



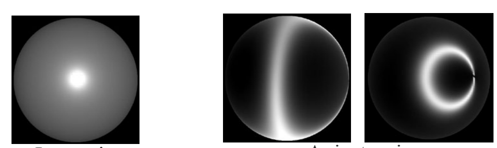
Reciprocity *and* isotropy

$$f_r(\theta_i,\theta_r,\varphi_r-\varphi_i) = f_r(\theta_r,\theta_i,\varphi_i-\varphi_r) = f_r(\theta_i,\theta_r,|\varphi_r-\varphi_i|)$$

4. Energy conservation

Isotropic vs Anisotropic

- Isotropic: Most materials (you can rotate about normal without changing reflections)
- Anisotropic: brushed metal etc. preferred tangential direction



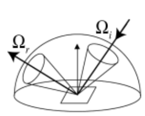
Isotropic Anisotropic

Energy Conservation

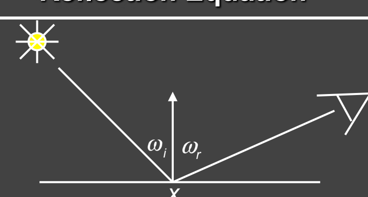
$$\frac{d\Phi_r}{d\Phi_i} = \frac{\int_{\Omega_r} L_r(\omega_r)\cos\theta_r\, d\omega_r}{\int_{\Omega_i} L_i(\omega_i)\cos\theta_i\, d\omega_i}$$

$$= \frac{\int_{\Omega_r}\int_{\Omega_i} f_r(\omega_i \to \omega_r) L_i(\omega_i)\cos\theta_i\, d\omega_i \cos\theta_r\, d\omega_r}{\int_{\Omega_i} L_i(\omega_i)\cos\theta_i\, d\omega_i}$$

$$\le 1$$



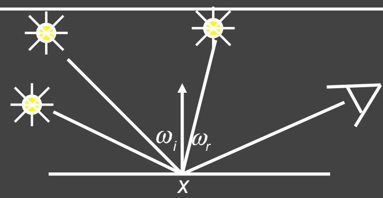
Reflection Equation



$$L_r(x,\omega_r) = L_e(x,\omega_r) + L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

Reflection Equation

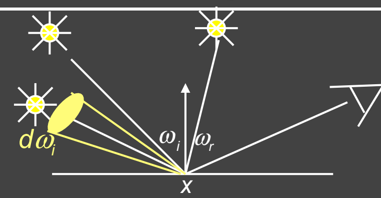


Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \sum L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

Reflection Equation



Replace sum with integral

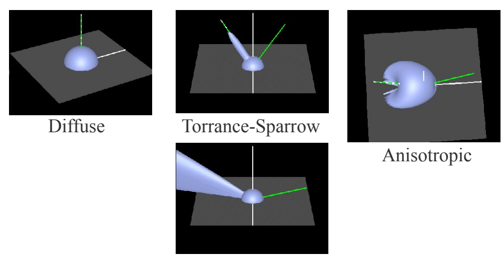
$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\Omega} L_i(x,\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

Radiometry

- Physical measurement of electromagnetic energy
- *Measure spatial (and angular) properties of light*
  - Radiance, Irradiance
  - Reflection functions: Bi-Directional Reflectance Distribution Function or BRDF
  - Reflection Equation
  - *Simple BRDF models*

Brdf Viewer plots

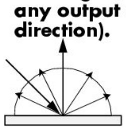


Diffuse — Torrance-Sparrow — Anisotropic

bv written by Szymon Rusinkiewicz

Ideal Diffuse Reflection

Assume light is equally likely to be reflected in any output direction (independent of input direction).



$$L_{r,d}(\omega_r) = \int f_{r,d} L_i(\omega_i) \cos\theta_i \, d\omega_i$$
$$= f_{r,d} \int L_i(\omega_i) \cos\theta_i \, d\omega_i$$
$$= f_{r,d} E$$

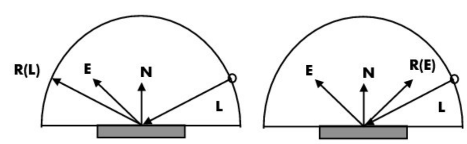
$$M = \int L_r(\omega_r) \cos\theta_r \, d\omega_r = L_r \int \cos\theta_r \, d\omega_r = \pi L_r$$

$$\rho_d = \frac{M}{E} = \frac{\pi L_r}{E} = \frac{\pi f_{r,d} E}{E} = \pi f_{r,d} \quad \Rightarrow \quad f_{r,d} = \frac{\rho_d}{\pi}$$

Lambert's Cosine Law $M = \rho_d E = \rho_d E_s \cos\theta_s$

CS348B Lecture 10 — Pat Hanrahan, Spring 2002

Phong Model



$$(\hat{\mathbf{E}} \bullet \mathbf{R}(\hat{\mathbf{L}}))^s \qquad (\hat{\mathbf{L}} \bullet \mathbf{R}(\hat{\mathbf{E}}))^s$$

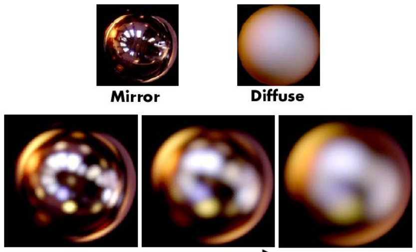
Reciprocity: $(\hat{\mathbf{E}} \bullet \mathbf{R}(\hat{\mathbf{L}}))^s = (\hat{\mathbf{L}} \bullet \mathbf{R}(\hat{\mathbf{E}}))^s$

Distributed light source!

CS348B Lecture 10 — Pat Hanrahan, Spring 2002

Specular Term (Phong)

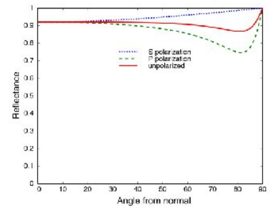
Phong Model

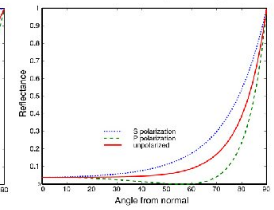


CS348B Lecture 10 — Pat Hanrahan, Spring 2002

Fresnel Reflectance

Metal (Aluminum) — Dielectric (N=1.5)





Gold F(0)=0.82
Silver F(0)=0.95

Glass n=1.5 F(0)=0.04
Diamond n=2.4 F(0)=0.15

Schlick Approximation $F(\theta) = F(0) + (1 - F(0))(1 - \cos\theta)^5$

CS348B Lecture 10 — Pat Hanrahan, Spring 2002

Experiment

Reflections from a shiny floor

From Lafortune, Foo, Torrance, Greenberg, SIGGRAPH 97

CS348B Lecture 10 — Pat Hanrahan, Spring 2002

Torrance-Sparrow

Fresnel term: allows for wavelength dependency

Geometric Attenuation: reduces the output based on the amount of shadowing or masking that occurs.

$$f = \frac{F(\theta_i)G(\omega_i,\omega_r)D(\theta_h)}{4\cos(\theta_i)\cos(\theta_r)}$$

Distribution: distribution function determines what percentage of microfacets are oriented to reflect in the viewer direction.

How much of the macroscopic surface is visible to the light source

How much of the macroscopic surface is visible to the viewer

Other BRDF models

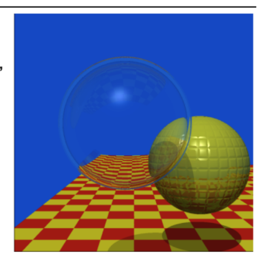
- Empirical: Measure and build a 4D table
- Anisotropic models for hair, brushed steel
- Cartoon shaders, funky BRDFs
- Capturing spatial variation
- Very active area of research

Ray Tracing History

Ray Tracing in Computer Graphics

"An improved Illumination model for shaded display," T. Whitted, CACM 1980

Resolution:
512 x 512
Time:
VAX 11/780 (1979)
74 min.
PC (2006)
6 sec.



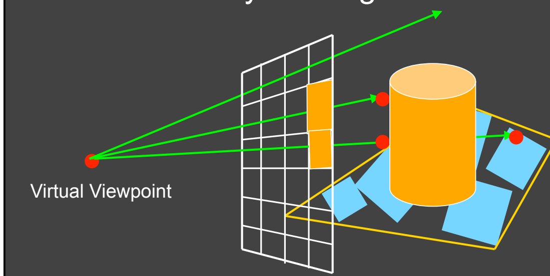
Spheres and Checkerboard, T. Whitted, 1979

CS348B Lecture 2 — Pat Hanrahan, Spring 2009

From SIGGRAPH 18

Real Photo: Instructor and Turner Whitted at SIGGRAPH 18

Ray Casting



Outline in Code

```
Image Raytrace (Camera cam, Scene scene, int width, int height)
{
   Image image = new Image (width, height) ;
   for (int i = 0 ; i < height ; i++)
        for (int j = 0 ; j < width ; j++) {
                Ray ray = RayThruPixel (cam, i, j) ;
                Intersection hit = Intersect (ray, scene) ;
                image[i][j] = FindColor (hit) ;
                }
   return image ;
}
```

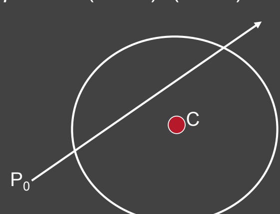
Ray/Object Intersections

- Heart of Ray Tracer
  - One of the main initial research areas
  - Optimized routines for wide variety of primitives
- Various types of info
  - Shadow rays: Intersection/No Intersection
  - Primary rays: Point of intersection, material, normals
  - Texture coordinates
- Work out examples
  - Triangle, sphere, polygon, general implicit surface

Ray-Sphere Intersection

$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$sphere \equiv (\vec{P} - \vec{C}) \cdot (\vec{P} - \vec{C}) - r^2 = 0$$



Ray-Sphere Intersection

$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$sphere \equiv (\vec{P} - \vec{C}) \cdot (\vec{P} - \vec{C}) - r^2 = 0$$

Substitute

$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$sphere \equiv (\vec{P}_0 + \vec{P}_1 t - \vec{C}) \cdot (\vec{P}_0 + \vec{P}_1 t - \vec{C}) - r^2 = 0$$

Simplify

$$t^2(\vec{P}_1 \cdot \vec{P}_1) + 2t\, \vec{P}_1 \cdot (\vec{P}_0 - \vec{C}) + (\vec{P}_0 - \vec{C}) \cdot (\vec{P}_0 - \vec{C}) - r^2 = 0$$

Ray-Sphere Intersection

$$t^2(\vec{P}_1 \bullet \vec{P}_1) + 2t\,\vec{P}_1 \bullet (\vec{P}_0 - \vec{C}) + (\vec{P}_0 - \vec{C}) \bullet (\vec{P}_0 - \vec{C}) - r^2 = 0$$

Solve quadratic equations for t

- 2 real positive roots: pick smaller root
- Both roots same: tangent to sphere
- One positive, one negative root: ray origin inside sphere (pick + root)
- Complex roots: no intersection (check discriminant of equation first)

Ray-Sphere Intersection

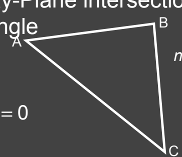
- Intersection point: $ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$
- Normal (for sphere, this is same as coordinates in sphere frame of reference, useful other tasks)

$$normal = \frac{\vec{P} - \vec{C}}{\left|\vec{P} - \vec{C}\right|}$$

Ray-Triangle Intersection

- One approach: Ray-Plane intersection, then check if inside triangle
- Plane equation:

$$plane \equiv \vec{P} \bullet \vec{n} - \vec{A} \bullet \vec{n} = 0$$

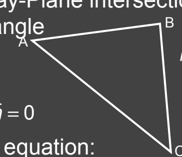


$$n = \frac{(C-A)\times(B-A)}{\left|(C-A)\times(B-A)\right|}$$

Ray-Triangle Intersection

- One approach: Ray-Plane intersection, then check if inside triangle
- Plane equation:

$$plane \equiv \vec{P} \bullet \vec{n} - \vec{A} \bullet \vec{n} = 0$$



$$n = \frac{(C-A)\times(B-A)}{\left|(C-A)\times(B-A)\right|}$$

- Combine with ray equation:

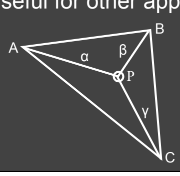
$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$(\vec{P}_0 + \vec{P}_1 t) \bullet \vec{n} = \vec{A} \bullet \vec{n}$$

$$t = \frac{\vec{A} \bullet \vec{n} - \vec{P}_0 \bullet \vec{n}}{\vec{P}_1 \bullet \vec{n}}$$

Ray inside Triangle

- Once intersect with plane, still need to find if in triangle
- Many possibilities for triangles, general polygons (point in polygon tests)
- We find parametrically [barycentric coordinates]. Also useful for other applications (texture mapping)

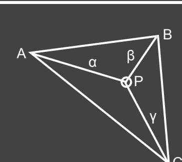


$P = \alpha A + \beta B + \gamma C$

$\alpha \geq 0, \beta \geq 0, \gamma \geq 0$

$\alpha + \beta + \gamma = 1$

Ray inside Triangle



$P = \alpha A + \beta B + \gamma C$

$\alpha \geq 0, \beta \geq 0, \gamma \geq 0$

$\alpha + \beta + \gamma = 1$

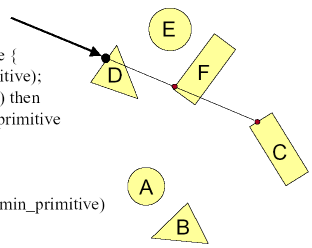
$P - A = \beta(B - A) + \gamma(C - A)$

$0 \leq \beta \leq 1$, $0 \leq \gamma \leq 1$

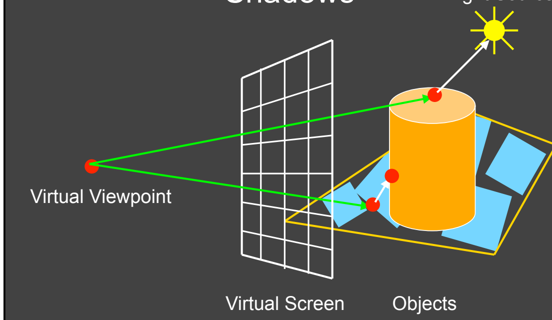
$\beta + \gamma \leq 1$

Ray Scene Intersection

```
Intersection FindIntersection(Ray ray, Scene scene)
{
    min_t = infinity
    min_primitive = NULL
    For each primitive in scene {
        t = Intersect(ray, primitive);
        if (t > 0 && t < min_t) then
            min_primitive = primitive
            min_t = t
        }
    }
    return Intersection(min_t, min_primitive)
}
```



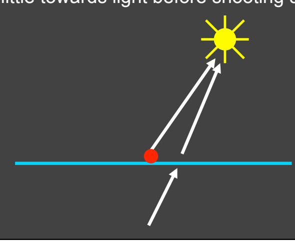
Shadows



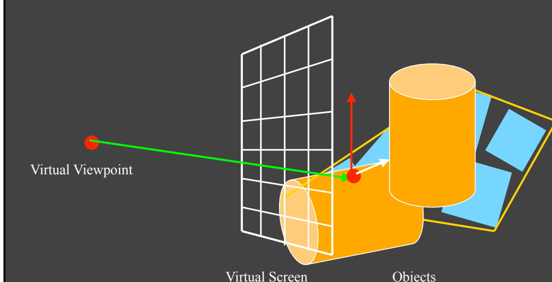
Shadow ray to light is blocked: object in shadow

Shadows: Numerical Issues

- Numerical inaccuracy may cause intersection to be below surface (effect exaggerated in figure)
- Causing surface to incorrectly shadow itself
- Move a little towards light before shooting shadow ray



Mirror Reflections/Refractions



Generate reflected ray in mirror direction,
Get reflections and refractions of objects

Recursive Ray Tracing

For each pixel
- Trace Primary Eye Ray, find intersection
- Trace Secondary Shadow Ray(s) to all light(s)
  - Color = Visible ? Illumination Model : 0 ;
- Trace Reflected Ray
  - Color += reflectivity * Color of reflected ray

- Need acceleration structure for performance

Interactive Raytracing

- Ray tracing historically slow
- Now viable alternative for complex scenes
  - Key is sublinear complexity with acceleration; need not process all triangles in scene
- Allows many effects hard in hardware
- Today graphics hardware and software (NVIDIA Optix 6, RTX chips 10G+ rays per second). Video