# Ray Tracing



## Recursive Ray Tracing

Gather light from various directions by tracing rays

Each pixel shows light at a surface

- trace ray from eye to surface

Each surface illuminated by lights and other surfaces

- trace rays from surface to other surfaces

And so on...



## Types of Rays

Eye/pixel rays

Illumination/shadow rays

Reflection rays

Transmission/transparency rays



## Eye Rays

Same as in ray casting

Effectively determine visible surfaces

For perspective view, trace from eye through pixel

For orthogonal view, trace parallel rays through pixels in direction of projection

Stop at nearest intersection with surface



## Illumination Rays

From surface point towards light source

Intervening surfaces may block or attenuate direct illumination

Light reaching surface applied using local illumination model



## Reflection Rays

Gather non-local illumination reflecting (specularly) towards eye (incident ray)

From current point towards reflection direction

Contribute illumination from closest surface intersection

Assume perfect (sharp) specular reflection

Attenuated by specular coefficient

## Transmission Rays

- Gather light transmitted through current surface
- Incident ray refracted according to index of refraction and Snell's law
- May use a single index of refraction (rather than per wavelength)
- Attenuated by transmission coefficient

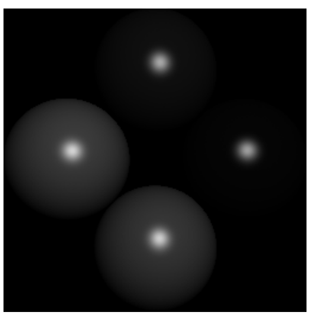
Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen

## Trace Algorithm

```
Trace(ray)
    Foreach object in scene
        Intersect(ray, object)
    If no intersections
        return BackgroundColor
    For each light
        Foreach object in scene
            Intersect(ShadowRay, object)
        Accumulate local illumination
    Trace(ReflectionRay)
    Trace(TransmissionRay)
    Accumulate global illumination
    Return illumination
```

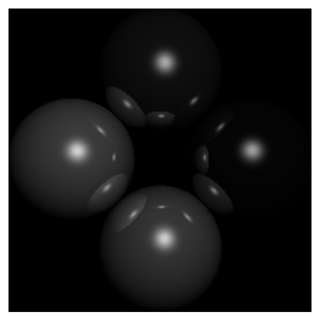
Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen

## No Recursion



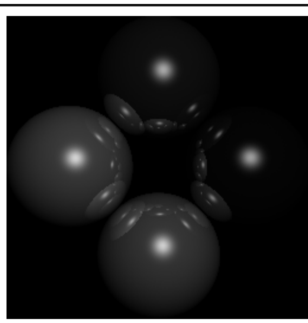
Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen

## 1 Bounce (level of recursion)



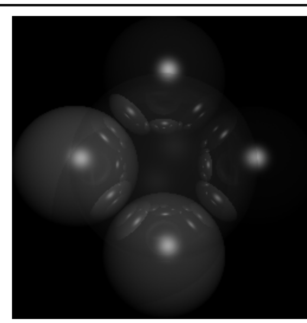
Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen

## 2 bounces



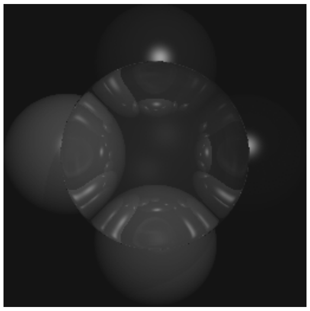
Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen

## What's different here?



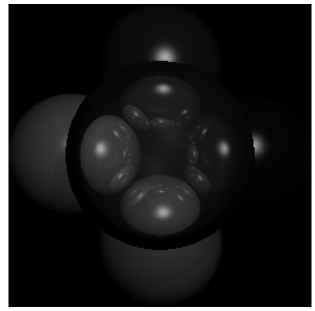
Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen

## Index of refraction > 1





## Index of refraction < 1



Notice *total internal reflection*



## Spheres and Checkerboard

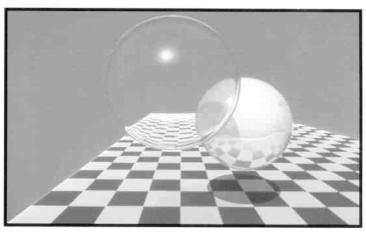
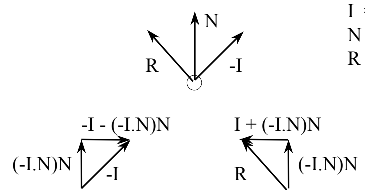


**Plate III.10** Spheres and checkerboard. An early image produced with recursive ray tracing (Section 16.12). (Courtesy of Turner Whitted, Bell Laboratories.)

**Turned Whitted, 1980 (Foley/vanDam III.10)**



## Computing Reflection Ray

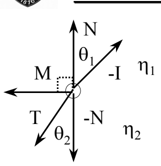



I = incident ray
N = normal vector
R = reflected ray

$R = (-I.N)N + I + (-I.N)N = I - 2(I.N)N$



## Computing Transmission Ray

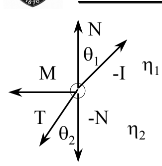


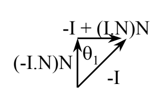

**Given N, I, $\eta_1$, and $\eta_2$**
**M is unit vector perpendicular to N**
**N, I, M, and T all lie in same plane**

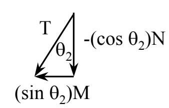
**Use Snell's law plus trigonometry to compute T…**



## Computing Transmission Ray




$-I + (I.N)N = -M \sin\theta_1$
$M = [I - (I.N)N] / \sin\theta_1$

$T = (\sin\theta_2)M - (\cos\theta_2)N$

Snell's Law
$\eta_{12} = \eta_1 / \eta_2 = \sin\theta_2 / \sin\theta_1$

$$
\begin{aligned}
T &= (\sin\theta_2 / \sin\theta_1)[I - (I.N)N] - (\cos\theta_2)N \\
&= \eta_{12} I - [\eta_{12}(I.N) - (\cos\theta_2)]N \\
&= \eta_{12} I - [\eta_{12}(I.N) - \mathrm{sqrt}(1 - \sin^2\theta_2)]N \\
&= \eta_{12} \mathbf{I} - [\eta_{12}(I.N) - \mathrm{sqrt}(1 - \eta_{12}^2\sin^2\theta_1)]\mathbf{N}
\end{aligned}
$$

## Accumulating Light Contributions at a Point

I = Σ local illumination (attenuated illum. rays)
+ attenuated reflection ray
+ attenuated transmission ray

Illumination rays attenuated by transmission coefficients of light occluders

Reflection ray attenuated by specular coefficient

Transmission ray attenuated by trans. coefficient

All rays *may* be attenuated by $1/r^2$



## Sampling Issues

Currently using only a single sample for each

- Pixel
- Reflection
- Transmission
- Frame time
- Eye point

All of these can cause forms of *aliasing*



## Sampling Theory

Aliasing: a high frequency signal masquerading as a lower frequency

Nyquist limit: maximum frequency signal that may be adequately sampled

(1/2 sampling frequency)

Aliasing may occur when we take regularly-spaced samples of frequencies above the Nyquist limit



## Examples of Aliasing

Reverse-rotating wagon wheels in films (temporal aliasing)

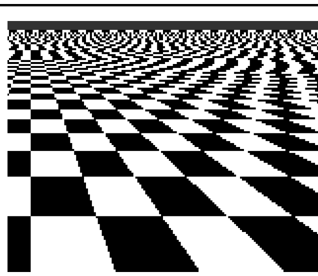
http://flowers.ofthenight.org/wagonWheel/wagonWheel.html

“Jaggies” on polygon edges or specular highlights (spatial aliasing)

Creeping effects (jaggies in motion)



## Jaggies and Moire pattern





## Aliasing Illustration

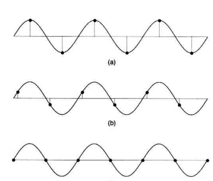



**Fig. 14.16** Sampling at the Nyquist rate (a) at peaks, (b) between peaks, (c) at zero crossings. (Courtesy of George Wolberg, Columbia University.)

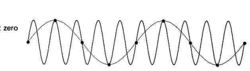
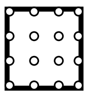


**Fig. 14.17** Sampling below the Nyquist rate. (Courtesy of George Wolberg, Columbia University.)

**Figures by George Wolberg, from Foley, vanDam, et al., *Computer Graphics: Principle and Practice*, p. 627-628**

## Anti-aliasing - Supersampling



Shoot multiple rays through each pixel

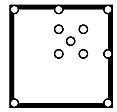
Aim through centers of a regular grid (e.g. 3x3 or 4x4 grid of samples)

Average resulting intensity values (box filter reconstruction)

Reduces spatial aliasing effects



## Adaptive Supersampling



Sample at corners and center

Add samples if gradient is more than some threshold

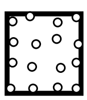
Add samples only in necessary regions

Reduces aliasing effects more efficiently

Still may not sample adequately



## Jittering



Apply random perturbations to sample positions of uniform or adaptive grid

Does not increase sampling rate, but removes regularity

Converts aliasing to *noise*, which is perceptually more tolerable



## Aliasing vs. Noise

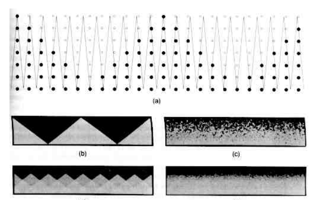



**Fig. 16.62** Aliasing vs. noise. (a) A comb with regularly spaced triangles, each $(n+1)/n$ pixels wide, sampled with one sample per pixel. o = samples that fall outside comb; ● = samples that fall inside comb. (b) A comb with 200 triangles, each 1.01 pixels wide and 50 pixels high. 1 sample/pixel, regular grid. (c) 1 sample/pixel, jittered $\pm\frac{1}{2}$ pixel. (d) 16 samples/pixel, regular grid. (e) 16 samples/pixel, jittered $\pm\frac{1}{8}$ pixel. (Images (b)–(e) by Robert Cook, Lucasfilm Ltd.)

Figures by Robert Cook, from Foley, vanDam, et al., *Computer Graphics: Principles and Practice*, p. 791.



## Distributed Ray Tracing

Apply distribution-based sampling to many parts of the ray-tracing algorithm

Fuzzy reflection/transmission
- perturb directions reflection/transmission, with distribution based on angle from ideal ray

Motion blur
- perturb eye ray samples in time

Depth of field
- perturb eye position on lens

Fuzzy shadows/penumbra
- sample illumination rays across area light

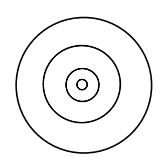


## Importance Sampling

Divide sample space into blocks of equal area under weighting function

Assign each pixel sample a different (random) block

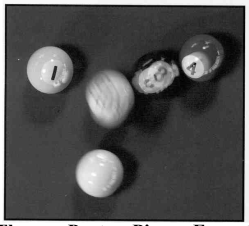
Perturb randomly within block (ideally, points in block have nearly equal weight…)



possible block arrangement for reflection ray direction

## Motion Blur Illustration

**Plate III.16** *1984.* Rendered using distributed ray tracing (Section 16.12.4) at 4096 × 3550 pixels with 16 samples per pixel. Note the motion-blurred reflections and shadows with penumbrae cast by extended light sources. (By Thomas Porter. © Pixar 1984. All Rights Reserved.)



**"1984," by Thomas Porter, Pixar. Exerpted from Foley, vanDam, et al., *Computer Graphics: Principles and Practice,* plate III.16**



## Depth of Field Illustration

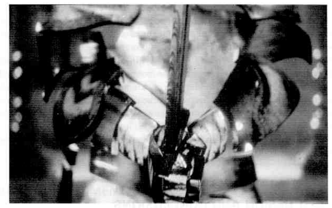


Fig. 17. Example of depth of field from *Young Sherlock Holmes*. Copyright 1985, Paramount Pictures Corp.

**from *Young Sherlock Holmes*, Paramount Pictures Corp, 1985. Exerpted from *An Introduction to Ray Tracing*, Andrew Glassner, ed., p. 195**



## Penumbras and Blurry Reflection

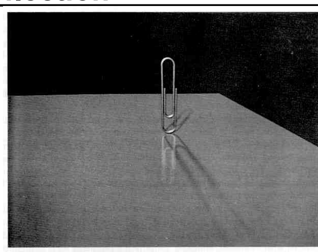


Fig. 18. Example of penumbras and blurry reflection.

**Exerpted from "Stochastic Sampling and Distributed Ray Tracing," by Robert L. Cook, in *An Introduction to Ray Tracing*, Andrew Glassner, ed., p. 195**