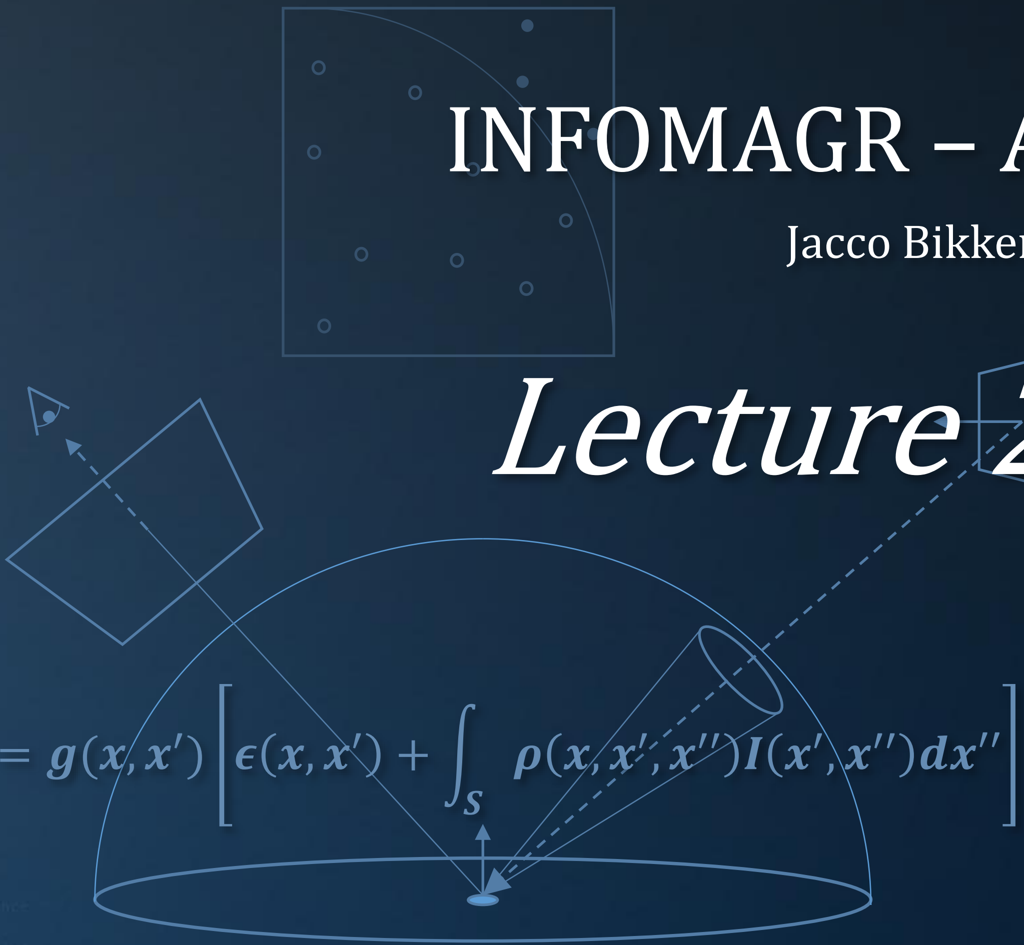
# INFOMAGR – Advanced Graphics

Jacco Bikker - November 2021 - February 2022

# *Lecture 2 - “Whitted”*

## Welcome!



# Today's Agenda:

- Introduction: Appel
- Whitted
- Cook

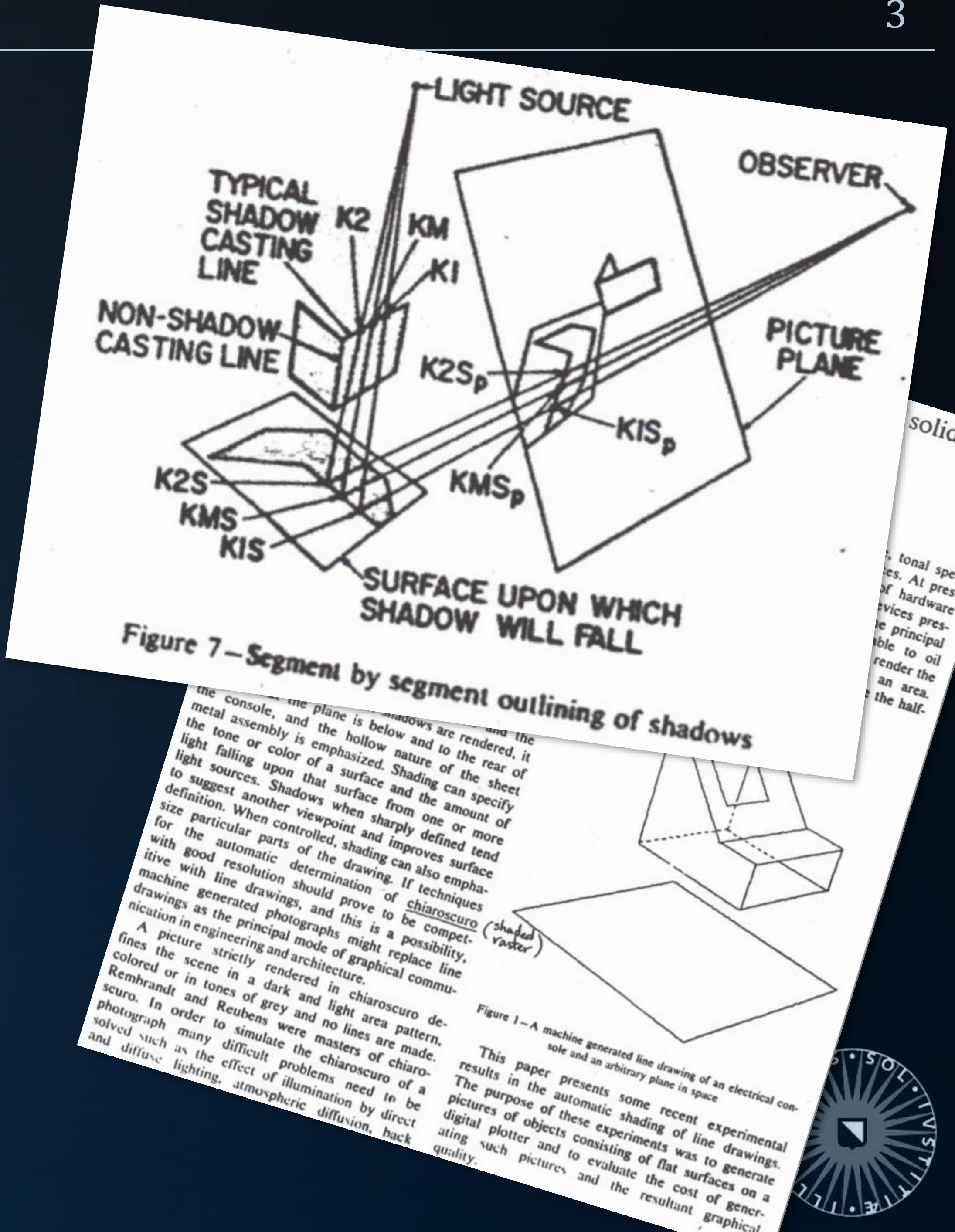
# Appel

Some Techniques for Shading Machine Renderings of Solids, Arthur Appel, 1964.

Idea: use rays to find geometry and shadows.

"Graphics" for games in 1964:
https://en.wikipedia.org/wiki/The_Sumerian_Game

IBM 7090



# Appel

Some Techniques for Shading Machine Renderings of Solids, Arthur Appel, 1964.

Idea: use rays to find geometry and shadows.

This method is very time consuming, usually requiring for useful results several thousand times as much calculation time as a wire frame drawing. About one half of this time is devoted to determining the

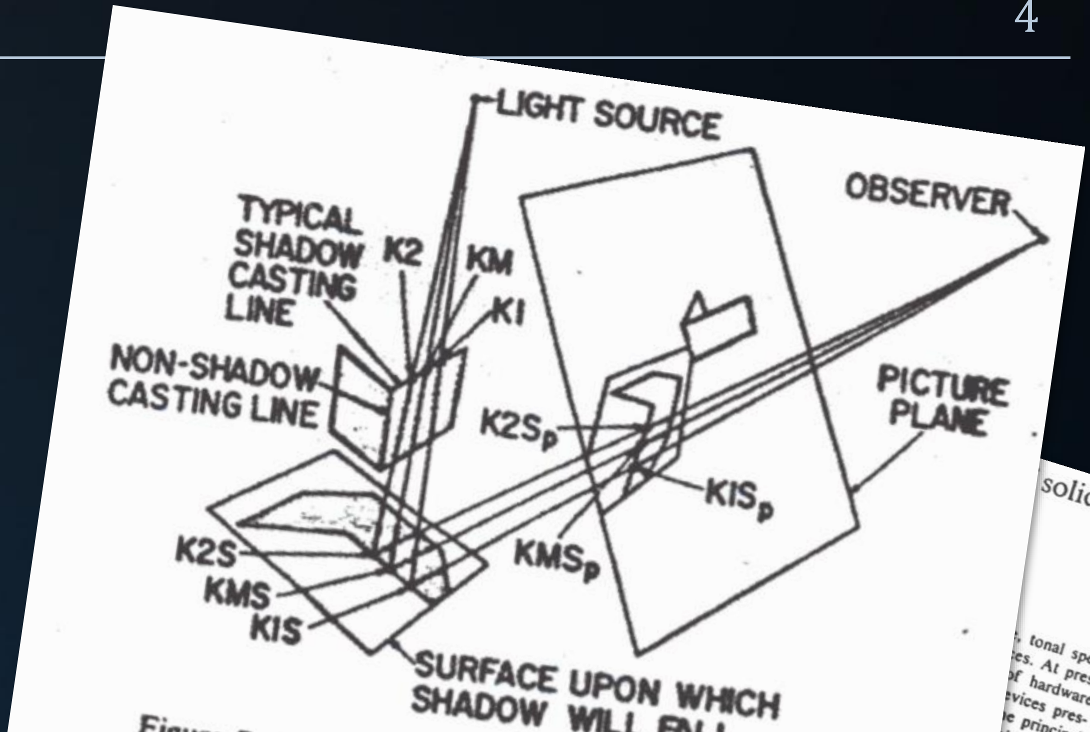


Figure 7—Segment by segment outlining of shadows

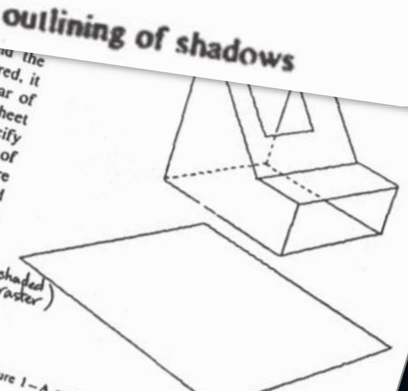
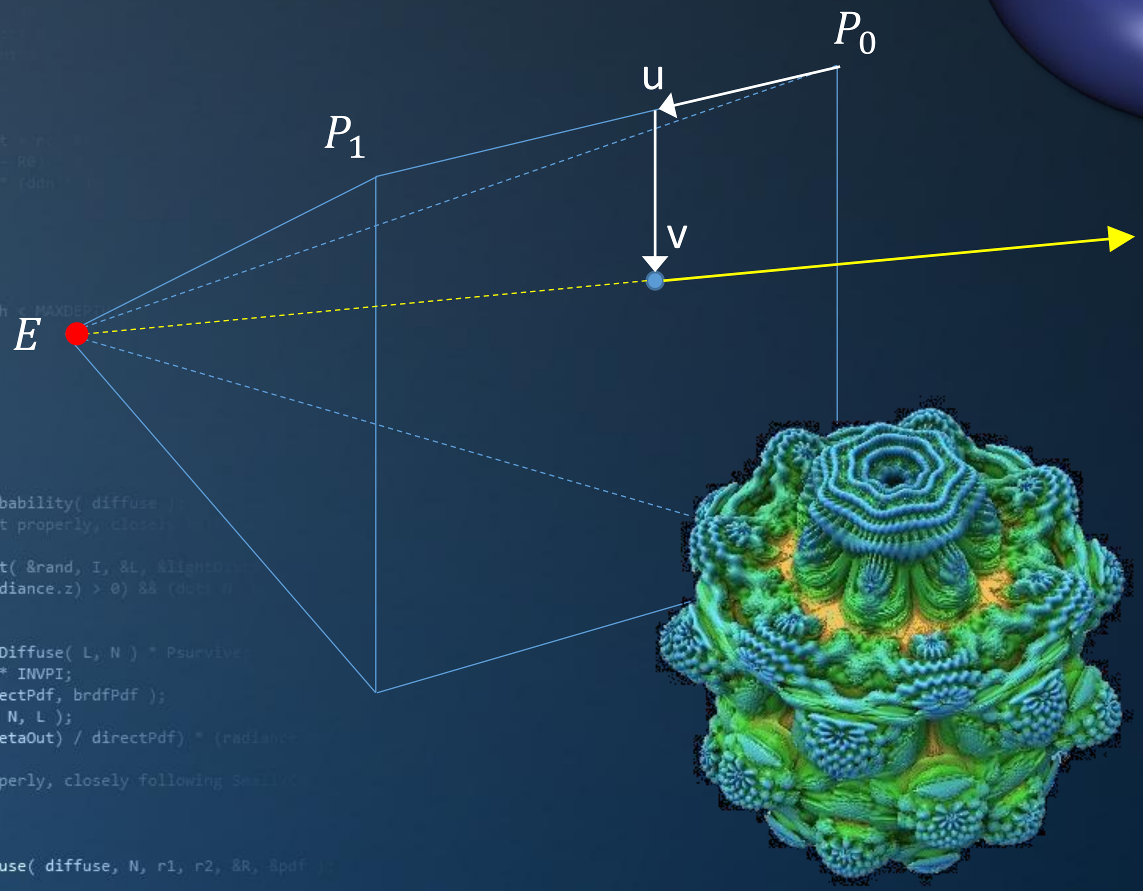


Figure 1—A machine generated line drawing of an electrical console and an arbitrary plane in space

# Appel

Recap




Plane: $P \cdot \vec{N} + d = 0$

Ray: $P(t) = O + t\vec{D}$

Substituting for $P(t)$, we get

$$(O + t\vec{D}) \cdot \vec{N} + d = 0$$
$$t = -(O \cdot \vec{N} + d)/(\vec{D} \cdot \vec{N})$$
$$P = O + t\vec{D}$$

Sphere: $(P - C) \cdot (P - C) - r^2 = 0$

Substituting for $P(t)$, we get

$$(O + t\vec{D} - C) \cdot (O + t\vec{D} - C) - r^2 = 0$$
$$\vec{D} \cdot \vec{D}\ t^2 + 2\vec{D} \cdot (O - C)\ t + (O - C)^2 - r^2 = 0$$

$$at^2 + bt + c = 0 \ \rightarrow t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$a = \vec{D} \cdot \vec{D}$$
$$b = 2\vec{D} \cdot (O - C)$$
$$c = (O - C) \cdot (O - C) - r^2$$

# Appel

Ray: $P(t) = O + t\vec{D}$

# Appel

# Today's Agenda:

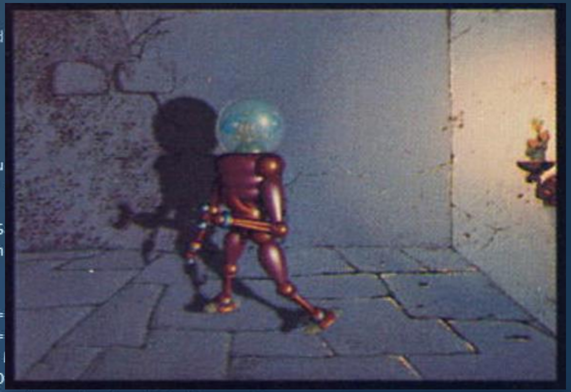
- Introduction: Appel
- Whitted
- Cook

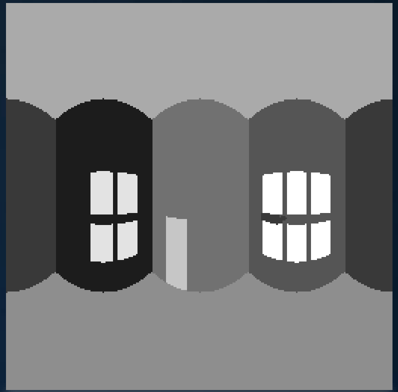
# Whitted

An Improved Illumination Model for Shaded Display

In 1980, “State of the Art” consisted of:

- Rasterization
- Shading: either diffuse (N · L) or specular ($(N \cdot H)^n$), both not taking into account fall-off (Phong)
- Reflection, using environment maps (Blinn & Newell *)
- Stencil shadows (Williams **)





* : Blinn, J. and Newell, M. 1976. Texture and Reflection in Computer Generated Images.

** : Williams, L. 1978. Casting curved shadows on curved surfaces..

# Whitted

An Improved Illumination Model for Shaded Display

In 1980, "State of the Art" consisted of:

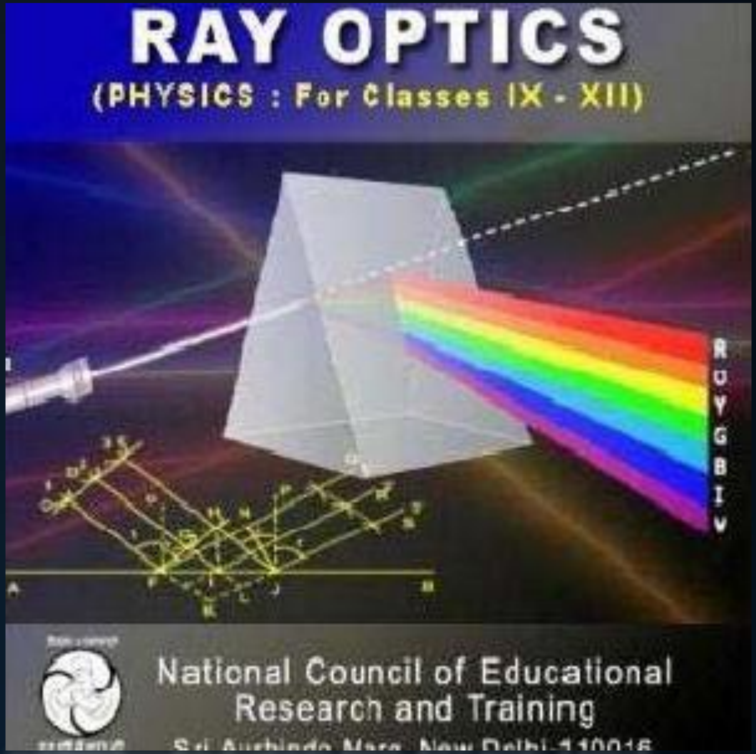
- Rasterization
- Shading: either diffuse $(N \cdot L)$ or specular $((N \cdot H)^n)$, both not taking into account fall-off (Phong)
- Reflection, using environment maps (Blinn & Newell)
- Stencil shadows (Williams)

Goal:

- Solve reflection and refraction

Improved model:

- Based on classical ray optics



# Whitted

An Improved Illumination Model for Shaded Display*

Physical basis of Whitted-style ray tracing:

Light paths are generated (backwards) from the camera to the light sources, using rays to simulate optics.

```
Color Trace( ray r )
   I, N, mat = NearestIntersection( scene, r )
   return mat.color * DirectIllumination( I, N )
```

* : T. Whitted. An Improved Illumination Model for Shaded Display. Commun. ACM, 23(6):343–349, 1980.

# Whitted

An Improved Illumination Model for Shaded Display

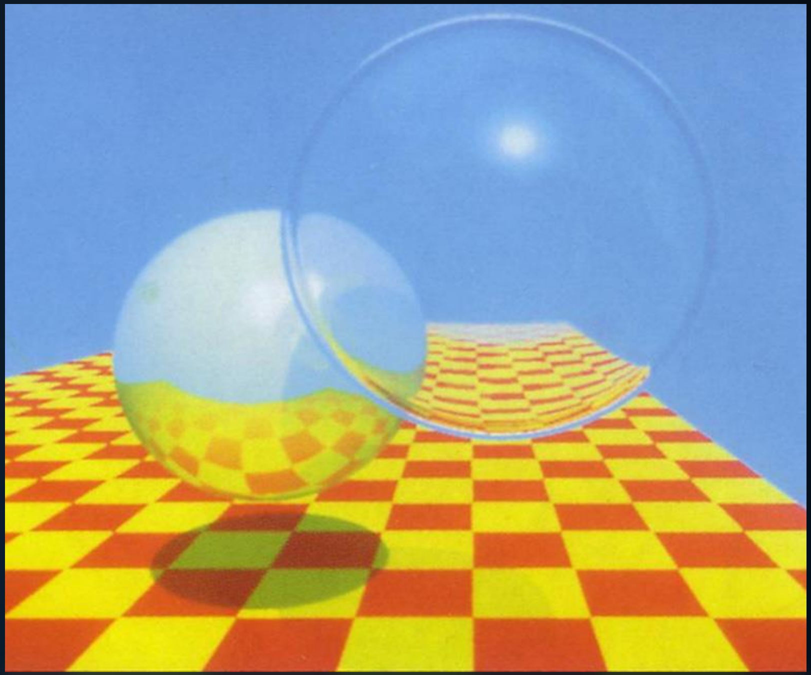
```
Color Trace( ray r )
   I, N, mat = NearestIntersection( scene, r )
   return mat.color * DirectIllumination( I, N )
```

Direct illumination:

*Summed* contribution of *unoccluded* point light sources, taking into account:

- Distance to I
- Angle between $\vec{L}$ and $\vec{N}$
- Intensity of light source

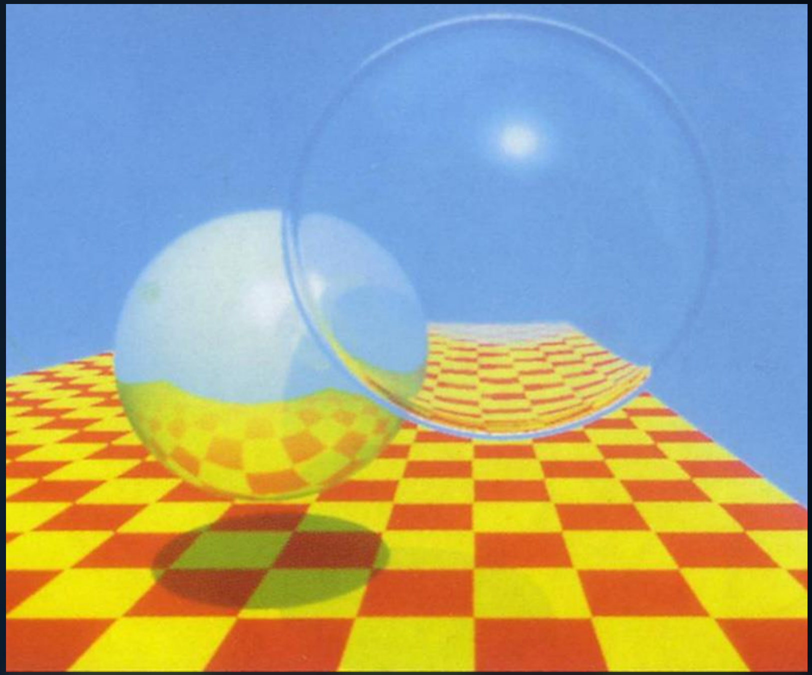
Note that this requires a ray per light source.



# Whitted

An Improved Illumination Model for Shaded Display

```
Color Trace( ray r )
   I, N⃗, mat = NearestIntersection( scene, r )
   if (mat == DIFFUSE)
      return mat.color * DirectIllumination( I, N⃗ )
   if (mat == MIRROR)
      return mat.color * Trace( I, reflect( r.D⃗, N⃗ )
```



Indirect illumination:

For perfect specular object (mirrors) we extend the primary ray with an extension ray:

- We still modulate transport with the material color
- We do not apply $\vec{N} \cdot \vec{L}$
- We do not calculate direct illumination

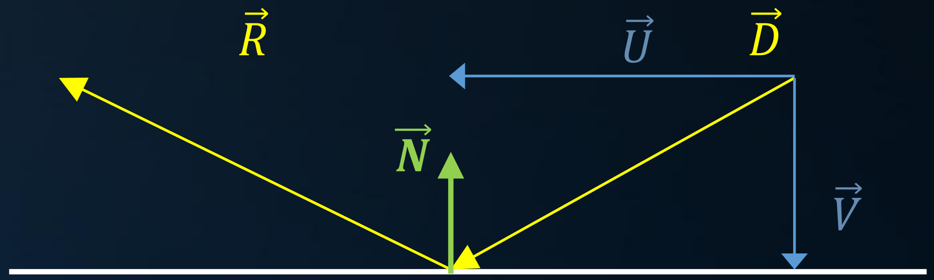
# Whitted

## Reflection

Given a ray direction $\vec{D}$ and a normalized surface normal $\vec{N}$, the reflected vector $\vec{R} = \vec{D} - 2(\vec{D} \cdot \vec{N})\vec{N}$.

Derivation:

$$\vec{V} = \vec{N}(\vec{D} \cdot \vec{N})$$
$$\vec{U} = \vec{D} - \vec{V}$$
$$\vec{R} = \vec{U} + (-\vec{V})$$
$$\vec{R} = \vec{D} - \vec{N}\left(\vec{D} \cdot \vec{N}\right) - \vec{N}(\vec{D} \cdot \vec{N})$$
$$\vec{R} = \vec{D} - 2(\vec{D} \cdot \vec{N})\vec{N}$$



# Whitted

<u>Question 1</u>: For direct illumination, we take into account:

- Material color
- Distance to light source
- $\vec{N} \cdot \vec{L}$

Why?

<u>Question 2</u>: We use the summed contribution of all light sources.

Is this correct?

<u>Question 3</u>: Why do we not sample the light sources for a pure specular surface? *(can you cast a shadow on a bathroom mirror?)*

<u>Question 4</u>: Show geometrically that, for normalized vectors $\vec{D}$ and $\vec{N}$, $\vec{R} = \vec{D} - 2(\vec{D} \cdot \vec{N})\vec{N}$ yields a normalized vector.

# Whitted

An Improved Illumination Model for Shaded Display

Handling partially reflective materials:

```
Color Trace( ray r )
   I, N⃗, mat = NearestIntersection( scene, r )
   s = mat.specularity
   d = 1 – mat.specularity
   return mat.color * (
      s * Trace( ray( I, reflect( r.D⃗, N⃗ ) ) ) +
      d * DirectIllumination( I, N⃗ ) )
```

Note: this is not efficient.  (why not?)

# Whitted

An Improved Illumination Model for Shaded Display

Dielectrics

```
Color Trace( ray r )
   I, N⃗, mat = NearestIntersection( scene, r )
   if (mat == DIFFUSE)
      return mat.color * DirectIllumination( I, N⃗ )
   if (mat == MIRROR)
      return mat.color * Trace( I, reflect( r.D⃗, N⃗ )
   if (mat == GLASS)
      return mat.color * ?
```

# Whitted

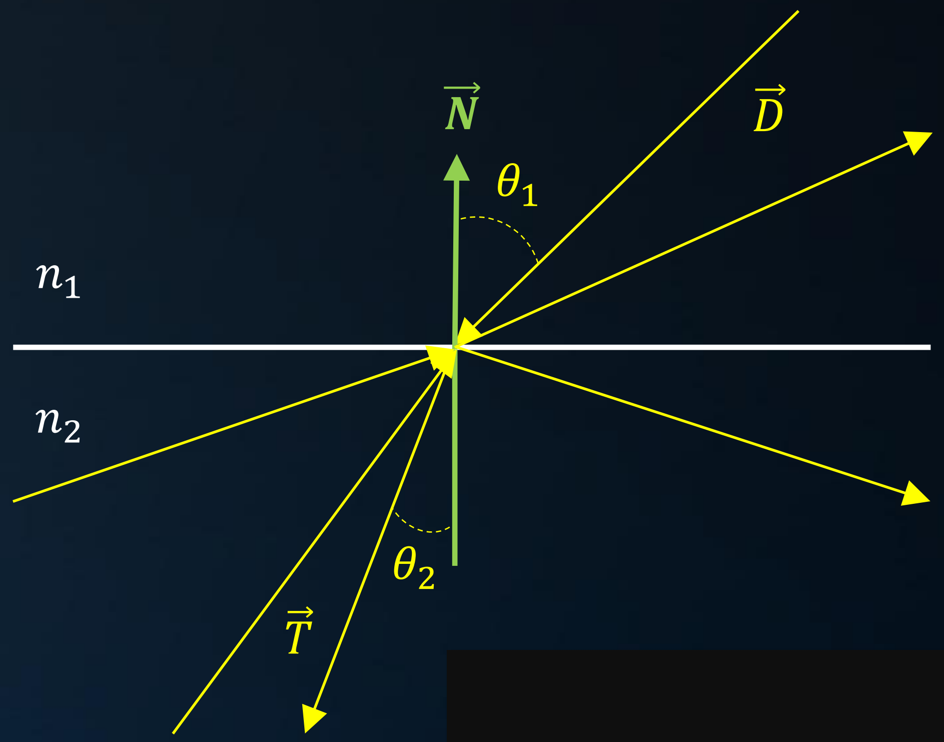
## Dielectrics

The direction of the transmitted vector $\vec{T}$ depends on the refraction indices $n_1, n_2$ of the media separated by the surface. According to Snell's Law:

$$n_1 sin\theta_1 = n_2 sin\theta_2$$

or

$$\frac{n_1}{n_2} sin\theta_1 = sin\theta_2$$




Note: left term may exceed 1, in which case $\theta_2$ cannot be computed. Therefore:

$$\frac{n_1}{n_2} sin\theta_1 = sin\theta_2 \Leftrightarrow sin\theta_1 \leq \frac{n2}{n1} \rightarrow \theta_{critical} = \arcsin\left(\frac{n_2}{n_1}\sin\theta_2\right)$$

# Whitted

# Whitted

https://en.wikipedia.org/wiki/Snell%27s_window

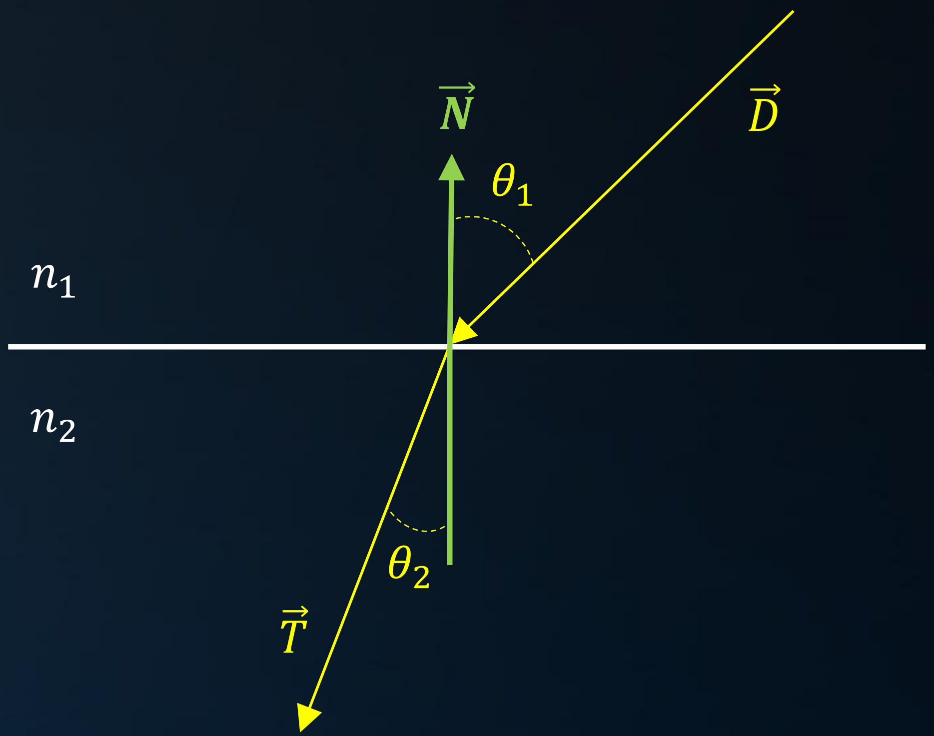
# Whitted

## Dielectrics

$$\frac{n_1}{n_2} sin\theta_1 = sin\theta_2 \Leftrightarrow sin\theta_1 \le \frac{n2}{n1}$$

$$k = 1 - \left(\frac{n_1}{n_2}\right)^2 \left(1 - cos{\theta_1}^2\right)$$

$$\vec{T} = \begin{cases} TIR, & for\ k < 0 \\ \frac{n_1}{n_2}\vec{D} + \vec{N}\left(\frac{n_1}{n_2} cos\theta_1 - \sqrt{k}\right), & for\ k \ge 0 \end{cases}$$



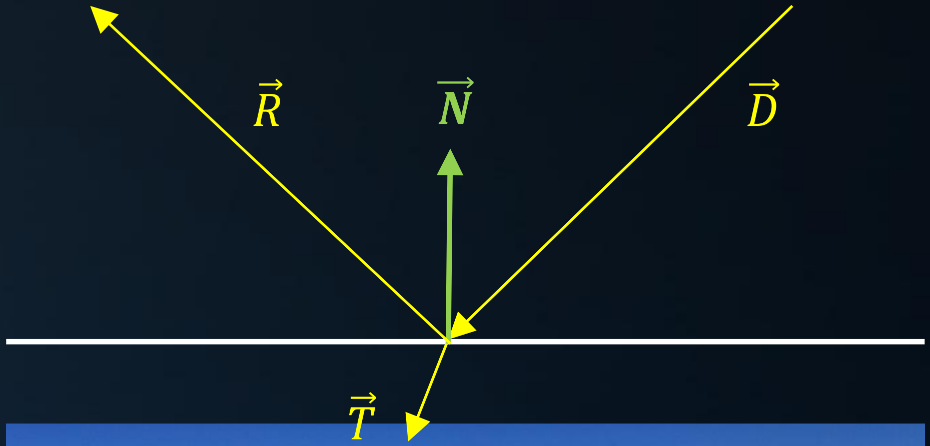

Note: $cos\theta_1 = \vec{N} \cdot -\vec{D}$, and $\frac{n_1}{n_2}$ should be calculated only once.

* For a full derivation, see http://www.flipcode.com/archives/reflection_transmission.pdf

# Whitted

## Dielectrics

A typical dielectric transmits *and* reflects light.




# Whitted

## Dielectrics

A typical dielectric transmits *and* reflects light.

Based on the Fresnel equations, the reflectivity of the surface for non-polarized light is formulated as:

$$F_r = \underbrace{\frac{1}{2}\Bigg( \underbrace{\left(\frac{n_1 cos\theta_i - n_2\cos\theta_t}{n_1 cos\theta_i + n_2\cos\theta_t}\right)^2}_{\text{Reflectance for s-polarized light}} + \underbrace{\left(\frac{n_1\cos\theta_t - n_2\cos\theta_i}{n_1\cos\theta_t + n_2\cos\theta_i}\right)^2}_{\text{Reflectance for p-polarized light}} \Bigg)}_{\text{Reflectance for unpolarized light}}$$

Where: $\cos\theta_t = \sqrt{1 - \left(\frac{n_1}{n_2}\sin\theta_i\right)^2}$

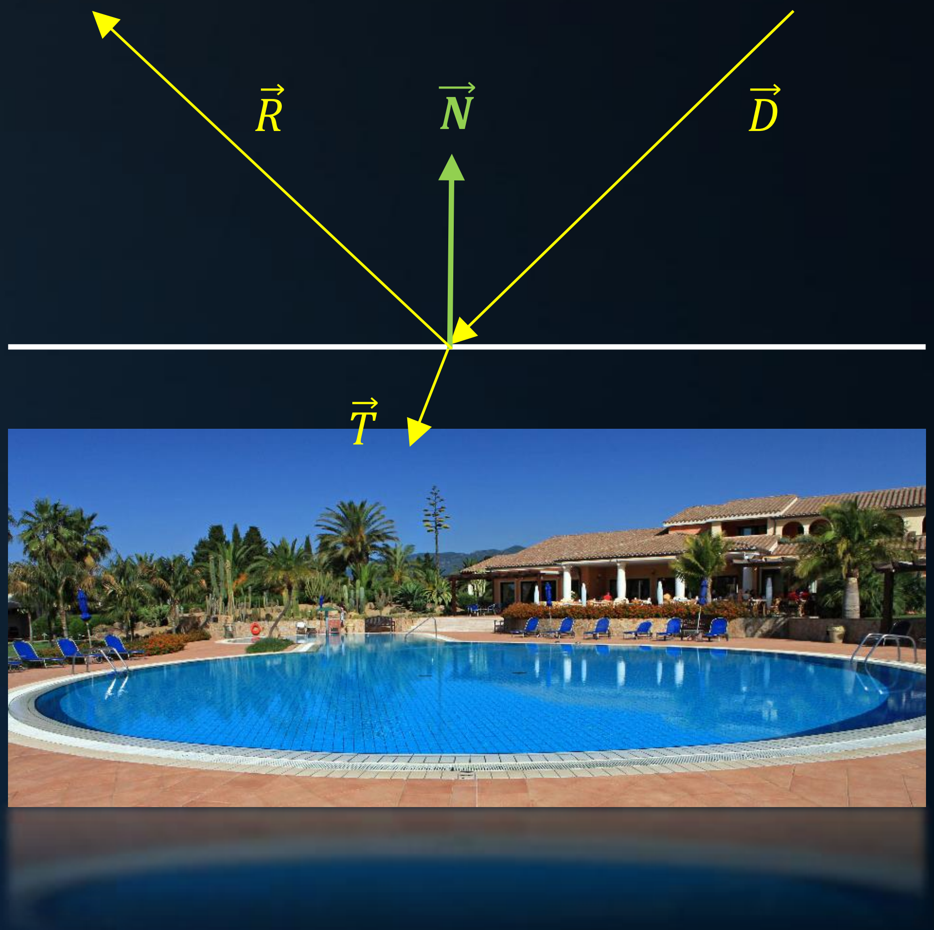
# Whitted

Dielectrics

$F_r = \cdots$

Based on the law of conservation of energy:

$F_t = 1 - F_r$



# Whitted

Ray Tracing

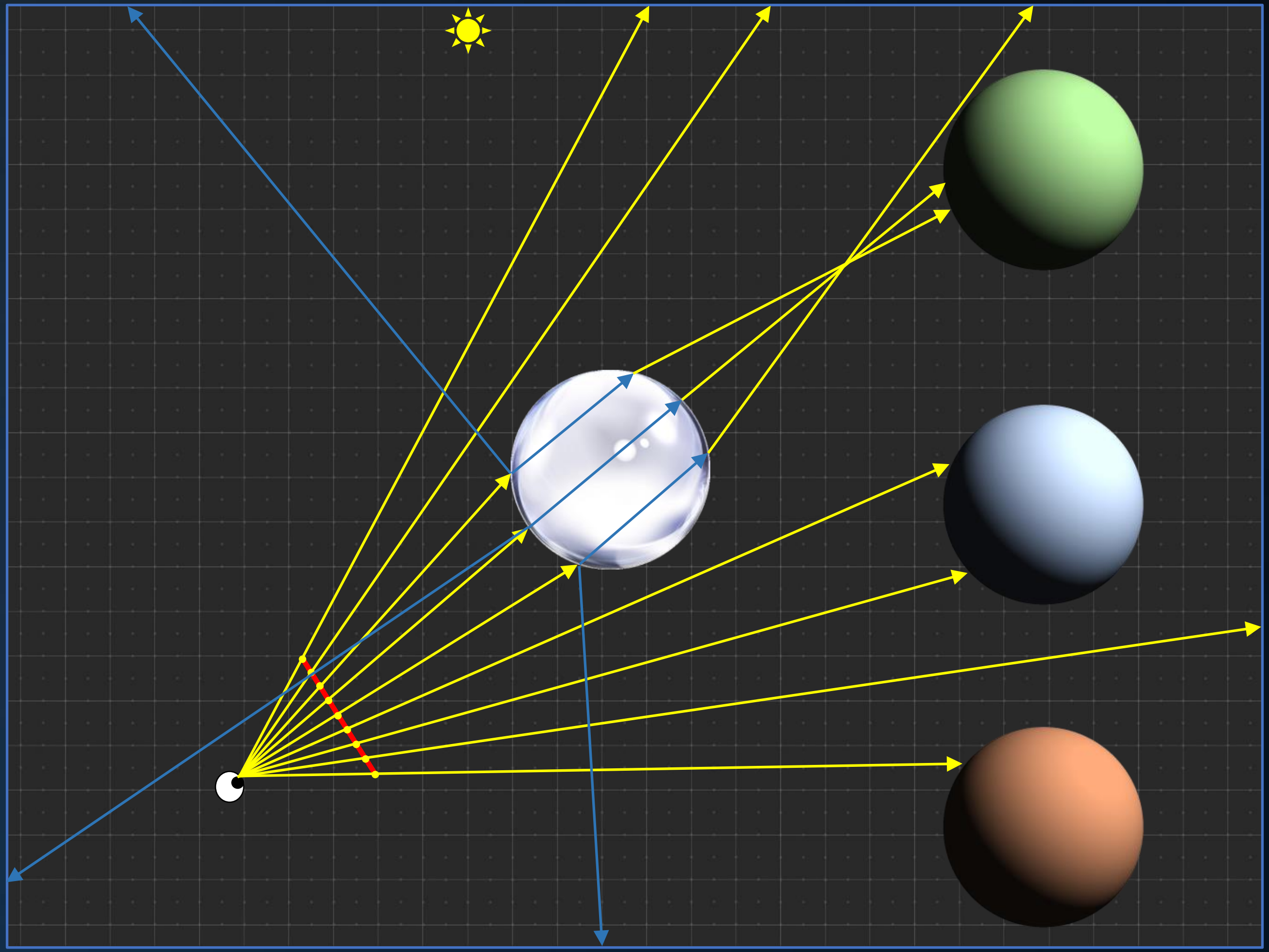
*World space*

- Geometry
- Eye
- Screen plane
- Screen pixels
- Primary rays
- Intersections
- Point light
- Shadow rays

Light transport

- Extension rays

Light transport



# Whitted

## Ray Tree

Using Whitted-style ray tracing, hitting a surface point may spawn:

- a shadow ray for each light source;
- a reflection ray;
- a ray transmitted into the material.

The reflected and transmitted rays may hit another object with the same material.

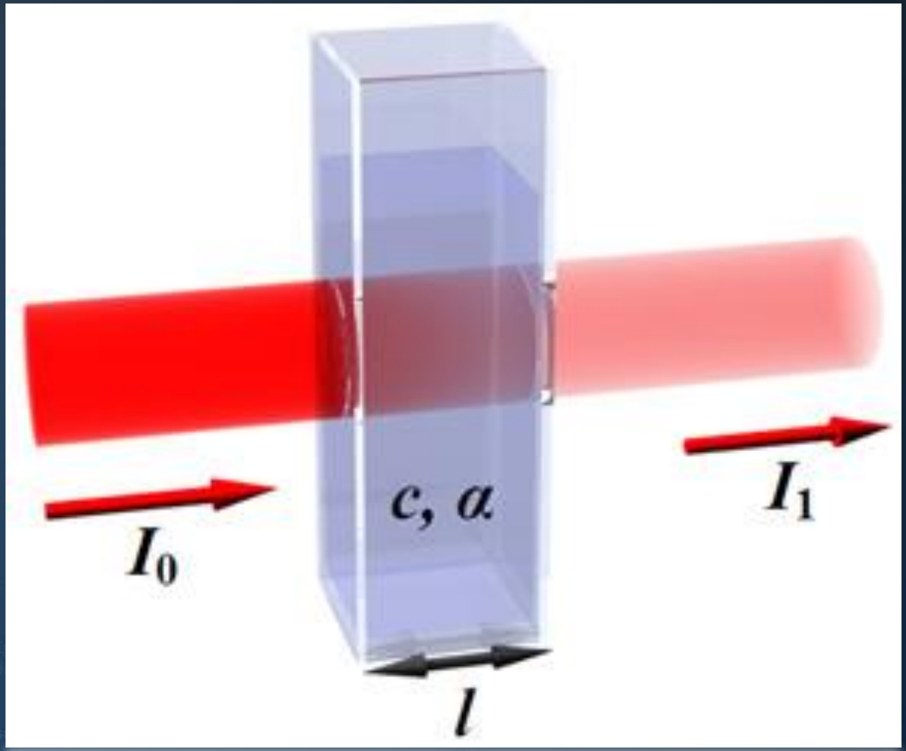
➔ A single primary ray may lead to a very large number of ray queries.

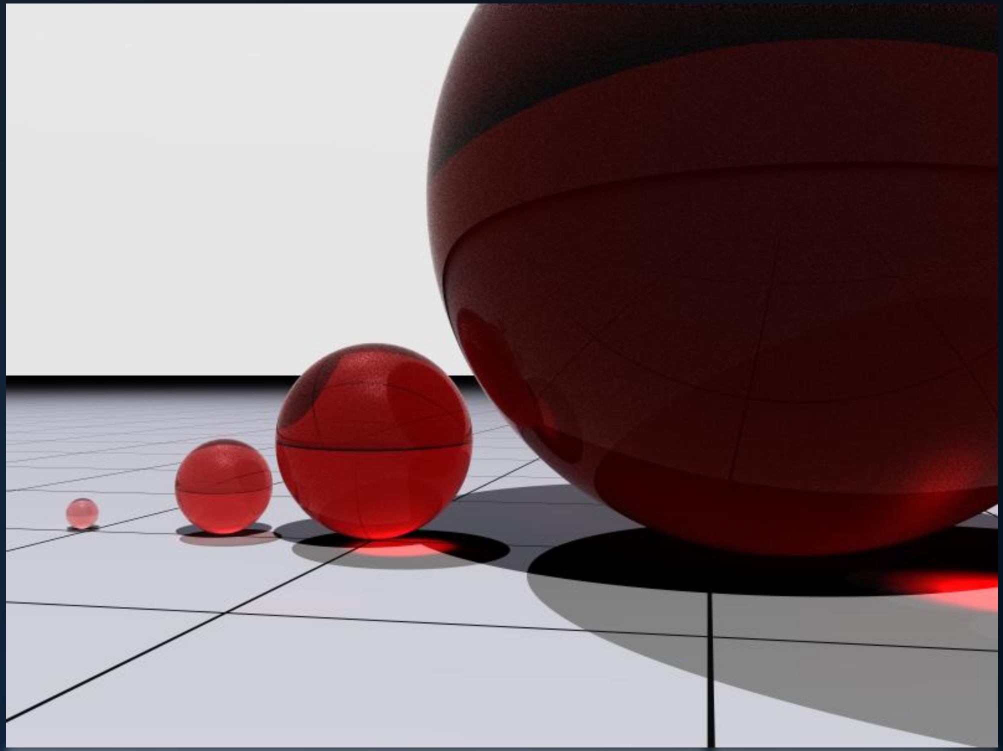
# Whitted

Question 5: imagine a scene with several point lights and dielectric materials. Considering the law of conservation of energy, what can you say about the energy transported by each individual ray?

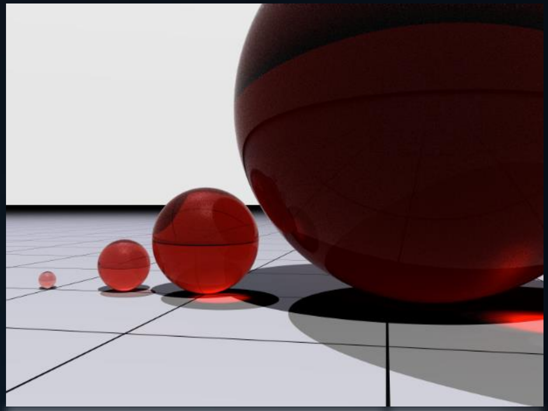
# Whitted

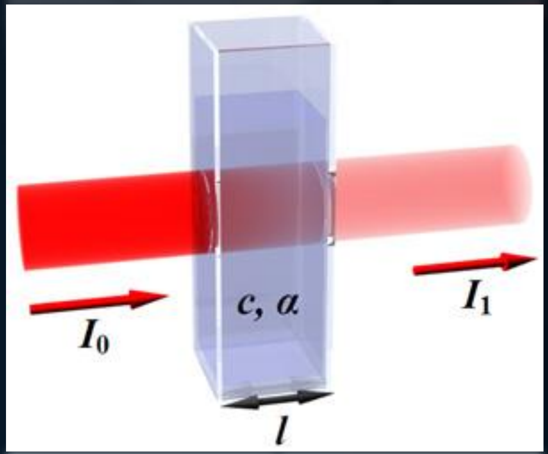
Beer's Law



# Whitted






## Beer's Law

Light travelling through a medium loses intensity due to absorption.

The intensity $I(d)$ that remains after travelling $d$ units through a substance with absorption $a$ is:

$$I(d) = I(0)e^{-\ln(a)d}$$

In pseudocode:

```
I.r *= exp( -a.r * d );
I.g *= exp( -a.g * d );
I.b *= exp( -a.b * d );
```

# Whitted

## Whitted - Summary

A Whitted-style ray tracer implements the following optical phenomena:

- Direct illumination of multiple light sources, taking into account
  - Visibility
  - Distance attenuation
  - A shading model: N dot L for diffuse
- Pure specular reflections, with recursion
- Dielectrics, with Fresnel, with recursion
- Beer’s Law

The ray tracer supports any primitive for which a ray/primitive intersection can be determined.

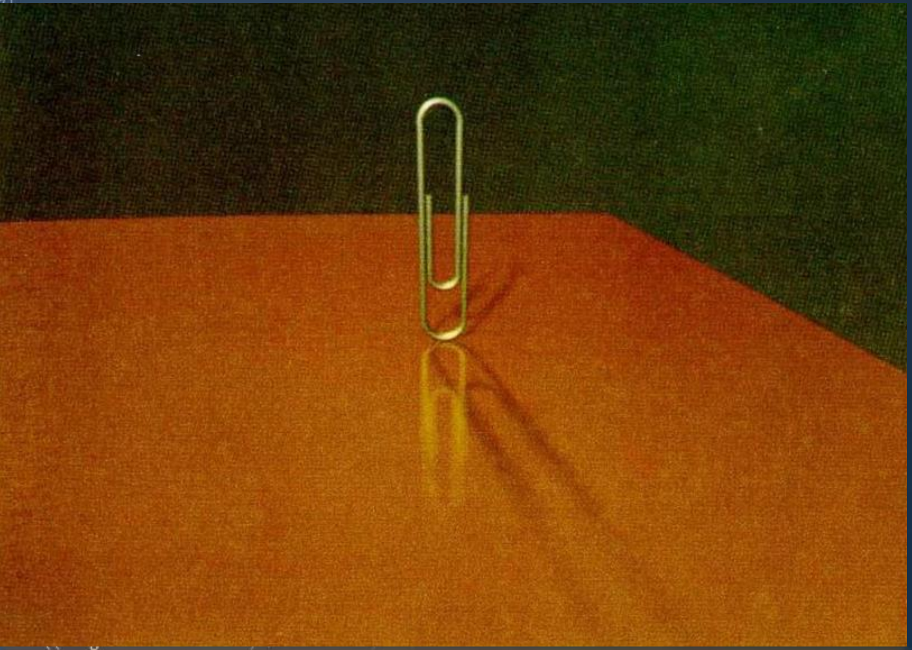
# Today's Agenda:

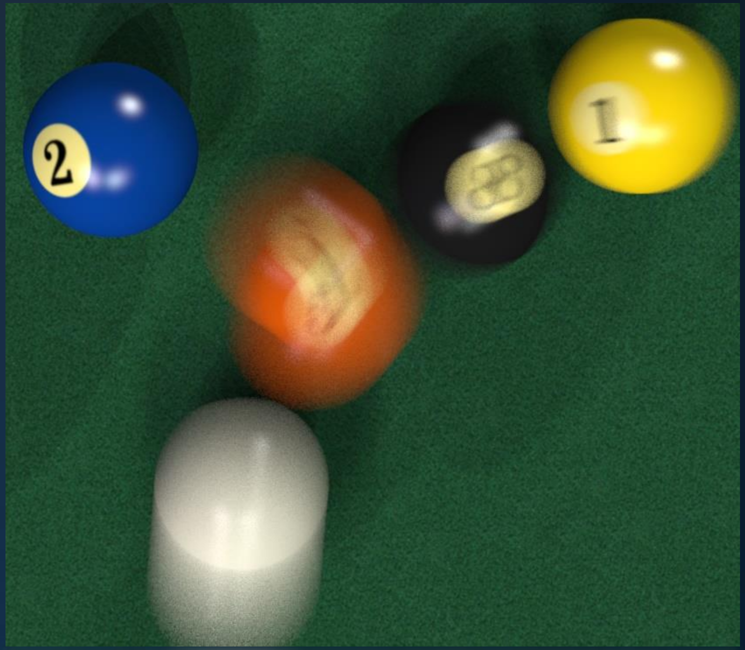
- Introduction: Appel
- Whitted
- Cook

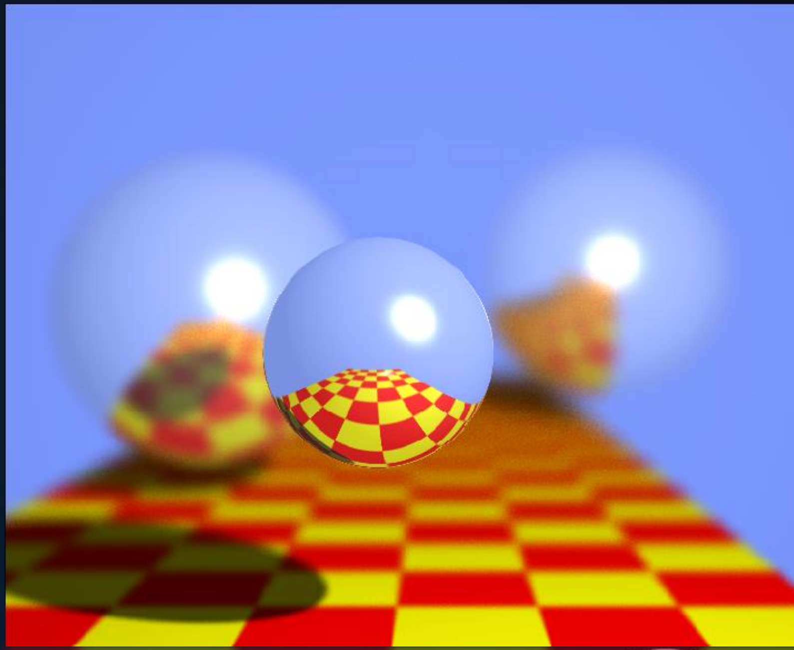
# Cook

’Distributed Ray Tracing’

Whitted-style ray tracing does not handle glossy reflections, depth of field, motion blur.







# Cook

'Distributed Ray Tracing'

Whitted-style ray tracing does not handle glossy reflections, depth of field, motion blur:

Ray tracing is a point sampling process.

Cook et al.*:

Replace point sampling by integrals:

- Perform anti-aliasing by integrating over the pixel
- Add motion blur by integrating over time
- Calculate depth of field by integrating over the aperture.

* : Cook et al., 1984. Distributed Ray Tracing.

# Today's Agenda:

- Introduction: Appel
- Whitted
- Cook

# INFOMAGR – Advanced Graphics

Jacco Bikker - November 2021 – February 2022

# END of “Whitted”

next lecture: “Acceleration Structures”