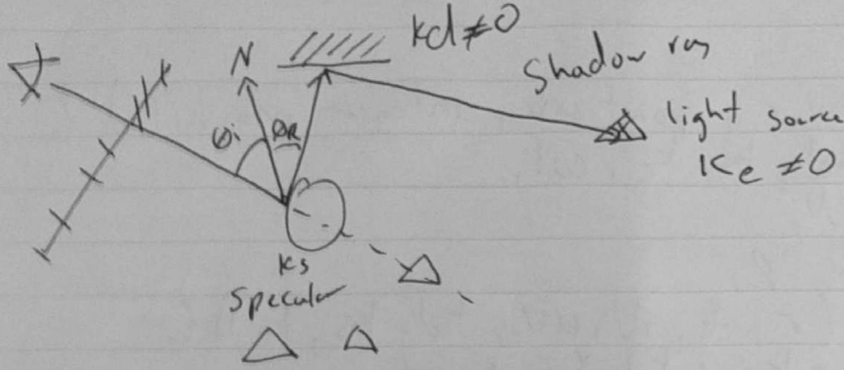
# Ray Tracing



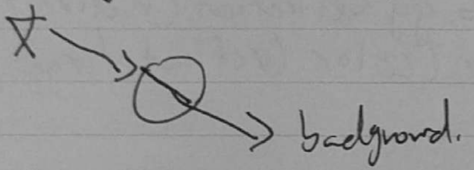
June 13th




$K_a$ = ambient

Specular object is getting ligh from a diffuse object. The diffuse object sends light out everywhere so we just create a shadow ray back to the light source as that is all we care about.

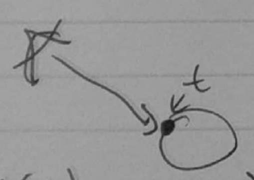
If things are transparent, we may hit a background



# Pseudocode

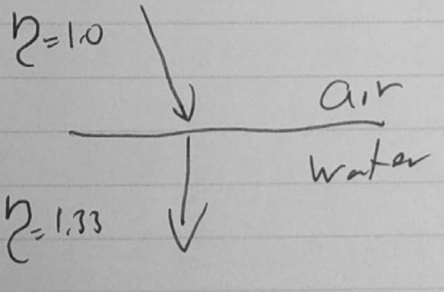
```
rayColor (ray r, point uv, integer maxhits) {
    color kd, ks, ke, col;
    vector N;
    point3D p;
    if (hit(r, t, N, uv, kd, ks, ke)) {
        col = ke + kd * ambient
        p = r.point.at Parameter (t);
        if (kd != 0) {                         -> see if objects are blockin the lightsource.
            col = kd * directLight (p, N, uv)
        }
        if (ks != 0) && (maxhits < T) {        -> to stop infinite hits.
            maxhits++;
            reflected.ray = ggReflection (r.direction, N)
            col += ks * ray (color (reflected.ray, uv, maxhits)
        }
        return col;
    }
    return Background
}
```

$t$ is the closest point that the ray hits an object




$N(\lambda) = \eta(\lambda) + jk(j)$

↳ refractive index



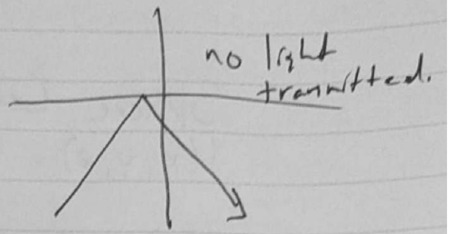

$k(\lambda) = \frac{(\text{Specific absorbtion } cc)\lambda}{4\pi}$

Water is blue due to absorbtion of light in water, light reflects many times internally.

# Raycasting Reflection

when $\sqrt{1-\left(\frac{n_i}{n_t}\right)^2\left(1-(\vec{V}\cdot\vec{N})^2\right)} < 0$

we get total internal reflection




Is $\eta$ always positive?

Digression: 1968 Russian Physicist Veselago

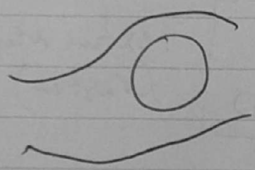
$\eta$ negative. but in nature $\eta$ is always positive.

1999 → John Pendry looked at it again

→ Can we make these materials.

→ metamaterials.

→ Can create a superlens which magnifies indefinitely.



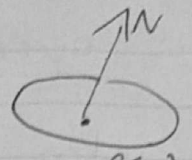
Can bend light around these materials.

Cloaking device.

## Surface Normals

Partial derivatives

$$N = \nabla f(p) = \left( \frac{\partial f(p)}{\partial (x)}, \frac{\partial f(p)}{\partial y}, \frac{\partial f(p)}{\partial z} \right)$$



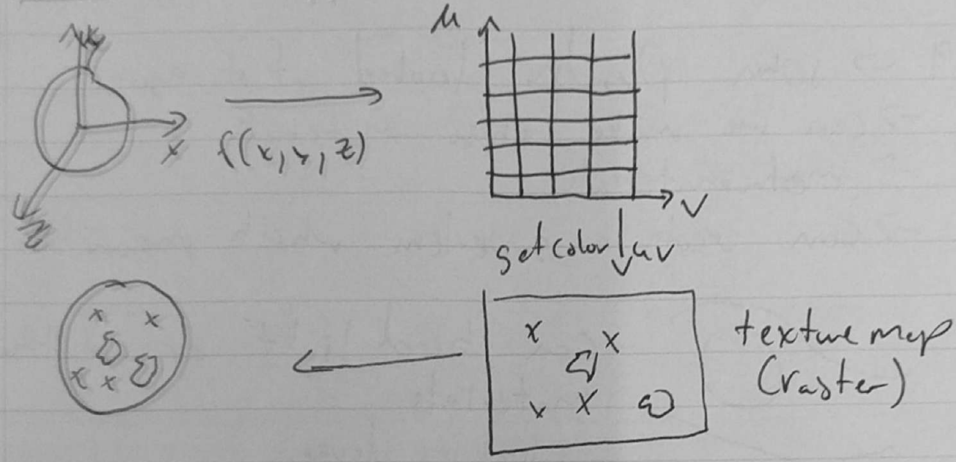

$f(p) = (x, y, z)$

Sphere Center = $C_x, C_y, C_z$ and radius $R$

$$f(x, y, z) = (x - C_x)^2 + (y - C_y)^2 + (z - C_z)^2 - R^2$$

$$\nabla f = (2(x - C_x), 2(y - C_y), 2(z - C_z))$$

## Texture Mapping (Blinn & Newel 1976)



*ggReflection* computes $\vec{\mathbf{r}}$ using:

$$\vec{\mathbf{r}} = \vec{\mathbf{v}} + 2\vec{\mathbf{N}}\cos\theta_i = \vec{\mathbf{v}} - 2\vec{\mathbf{N}}(\vec{\mathbf{v}}\cdot\vec{\mathbf{N}})$$

*ggRefract* computes $\vec{\mathbf{t}}$ using:

$$\vec{\mathbf{t}} = -\vec{\mathbf{N}}\cos\theta_t + \vec{\mathbf{m}}\sin\theta_t$$

eventually yields:

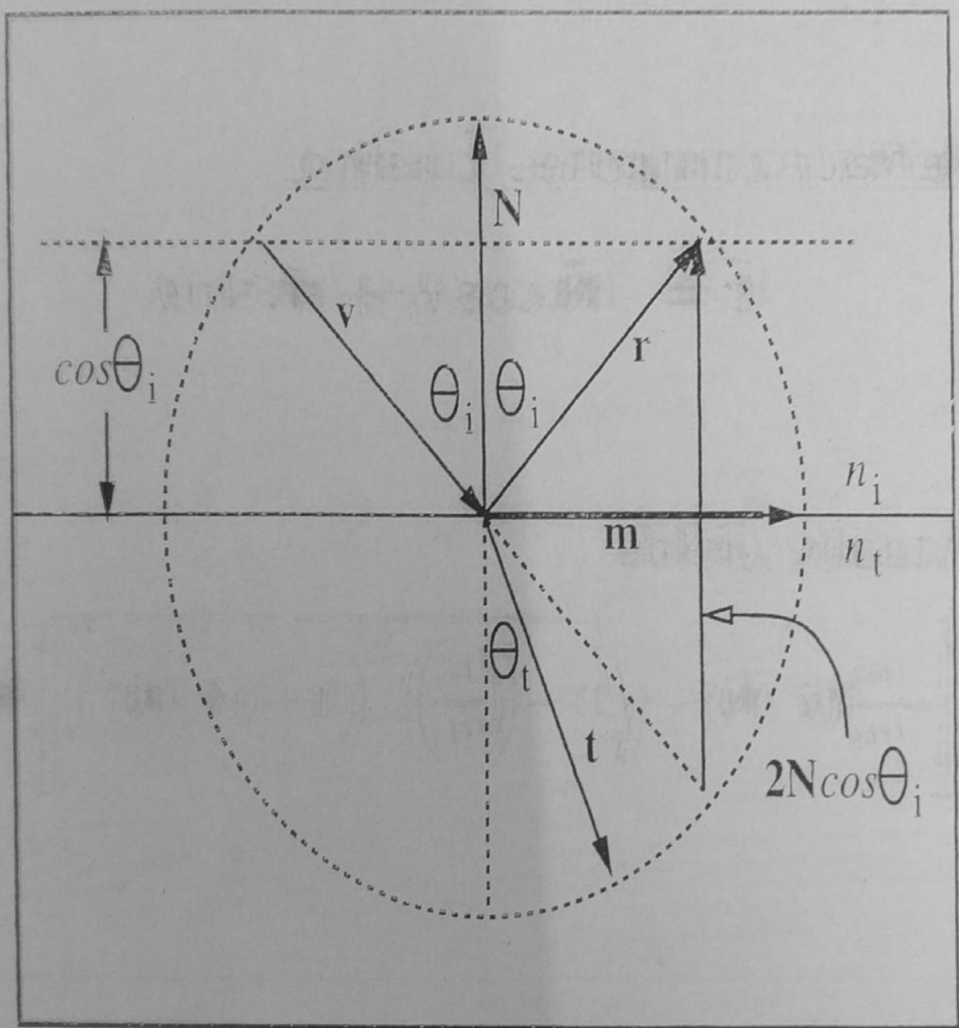
$$\vec{\mathbf{t}} = \left[-\frac{n_i}{n_t}(\vec{\mathbf{v}}\cdot\vec{\mathbf{N}}) - \sqrt{1-\left(\frac{n_i}{n_t}\right)^2\left(1-(\vec{\mathbf{v}}\cdot\vec{\mathbf{N}})^2\right)}\right]\vec{\mathbf{N}} + \frac{n_i}{n_t}\vec{\mathbf{v}}$$

Reflection Law:

$$\theta_i = \theta_r$$

Snell's Law:

$$n_i \sin\theta_i = n_t \sin\theta_t$$



# Texture Mapping

## Psuedocode for a rectangular region

- for a point $(u,v)$ find the coresponding $d_i, d_j$ in the raster file $d_i = (w-1) \times u$

  $d_j = (h-1) \times v$
- find $u_p, v_p$ for that pixel

$i = \mathrm{int}(d_i)$ $\quad u_p = d_i - i$

$j = \mathrm{int}(d_j)$ $\quad v_p = d_j - j$

- Interpolate color from surrounding pixels using the parametric equation of the rectangle

$C_{00} = \mathrm{Color}(i, j)$

$C_{01} = \mathrm{Color}(i, j+1)$

$C_{10} = \mathrm{Color}(i+1, j)$

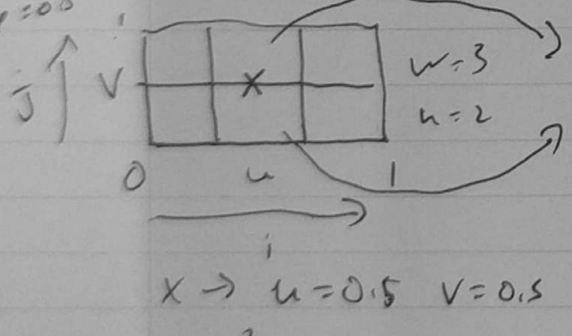
$C_{11} = \mathrm{Color}(i+1, j+1)$

$\mathrm{Color}(u_p, v_p) =$
$C_{00}(1-u_p)(1-v_p) +$
$C_{01}(1-u_p)v_p +$
$C_{10}(u_p)(1-v_p) +$
$C_{11}(u_p)(v_p)$

- clamp at 1.0.

$d_i = 1$
$d_j = 0.5$
$i = 1$
$j = 0$
$u_p = 0$
$v_p = 0.5$

**Example:**




$w = 3$
$h = 2$

$C_{01} = \mathrm{Color}(i, j+1)$

$C_{00} = \mathrm{Color}(i, j)$

$x \to u = 0.5 \quad v = 0.5$

$u_p = ?$

$v_p = ?$

$\mathrm{Color}(u_p, v_p) = \mathrm{Color}(0, 0.5)$
$= C_{00}(1)(0.5) + C_{01}(1)(0.5)$
$= \dfrac{C_{00} + C_{01}}{2}$