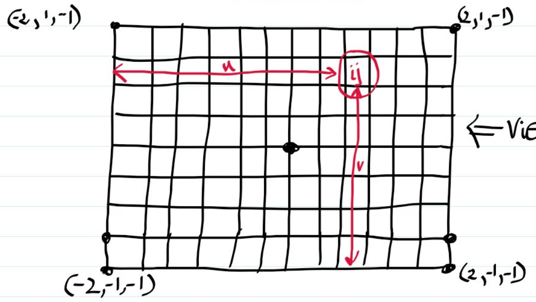
CG NOTES VOL·1

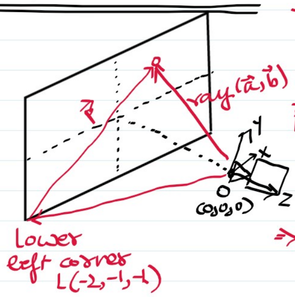


$u = \frac{i}{width-1}$; $v = \frac{j}{height-1}$

For multisampling

$u = \frac{i+random(0,1)}{width-1}$; $v = \frac{j+random(0,1)}{height-1}$

Casting rays through every pixel:



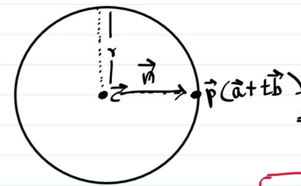
$\vec{p}$ is pixel vector from pt. L

$\vec{p} = L - u*horizontal - v*vertical$

$ray = \vec{o} + t(\vec{b})$

$\Rightarrow \underline{ray = \vec{o} + t(\vec{p}-\vec{o})}$

Intersection of ray w/ a sphere



$(\vec{p}-\vec{c})\cdot(\vec{p}-\vec{c}) = r^2$

$(\vec{a}+t\vec{b}-\vec{c})\cdot(\vec{a}+t\vec{b}-\vec{c}) = r^2$

$$\Rightarrow \boxed{t^2\overbrace{(\vec{b}\cdot\vec{b})}^{A} + \overbrace{2t\vec{b}\cdot(\vec{a}-\vec{c})}^{B} + \underbrace{(\vec{a}-\vec{c})\cdot(\vec{a}-\vec{c}) - r^2}_{C}}$$

$$\therefore \boxed{t = \frac{-B \pm \sqrt{B^2-4AC}}{2A}}$$

$\vec{n} = \vec{p} - \vec{c}$

Recording Ray hits: In a world w/ multiple objects, a ray trace returns color of nearest hit recorded.

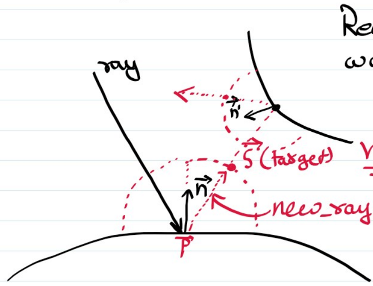
```
struct hit_record
{ p, n, t }

world :: hit(
{
    closest = max_t;
    foreach (obj : world)
    {
        if (obj->hit (ray, t_min, closest, temp_rec)
        {
            closest = temp_rec.t;
            rec = temp.rec;
        }
    }
}
```

*

Diffuse Reflection:

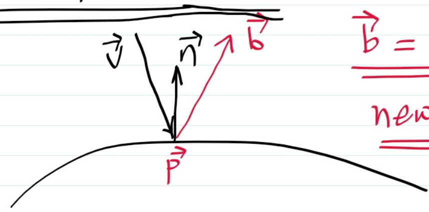
Recursive testing. Breaks if max depth reached or new_ray does not intersect w/ any objects in world.



$\underline{new\_ray = \vec{p} + t(\vec{s}-\vec{p})}$

$\underline{\vec{s} = \vec{p} + \vec{n} + random\_vec(\ )}$

METALLIC REFLECTION :

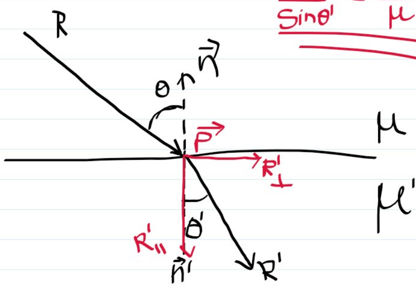


$$\vec{b} = \vec{v} - 2(\vec{v}\cdot\vec{n})\vec{n}$$

$$new\_ray = \vec{p} + t\vec{b}$$

Refraction :

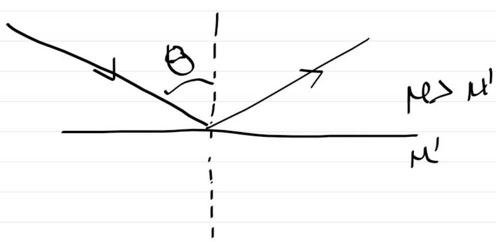
$$\frac{\sin\theta}{\sin\theta'} = \frac{\mu'}{\mu} \Rightarrow \sin\theta' = \frac{\mu}{\mu'} \times \sin\theta$$



$$\boxed{\begin{array}{l} R'_\perp = \frac{\mu}{\mu'}\left(R + \vec{n}\cos\theta\right) \\ R'_{\parallel} = -\sqrt{1 - |R_\perp|^2}\ \vec{n} \end{array}} \Bigg\} R = R'_\perp + R'_{\parallel}$$

$$new\_ray = \vec{p} + t\vec{R'}$$

Total Internal Reflection :



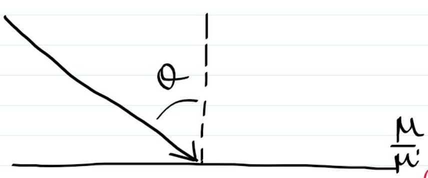
ray reflects because, $\sin\theta \in [0,1]$

$\Rightarrow \sin\theta' = \frac{\mu}{\mu'}\sin\theta > 1 \quad \because \frac{\mu}{\mu'} > 1$

$\therefore$ TIR happens when $\frac{\mu}{\mu'} \times \sin\theta > 1.0$

Schlick's approximation :

Reflectance varies w/ angle. Schlick's approxmn' helps approximate this phenomenon & the calculation.

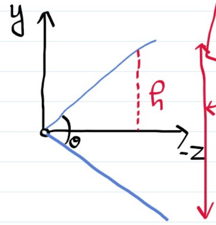


$$r_0 = \left[\frac{1 - \frac{\mu}{\mu'}}{1 + \frac{\mu}{\mu'}}\right]^2$$

$$\boxed{r = r_0 + (1 - r_0)\left[1 - \cos\theta\right]^5} *$$

Positionable Camera :

1) FOV -

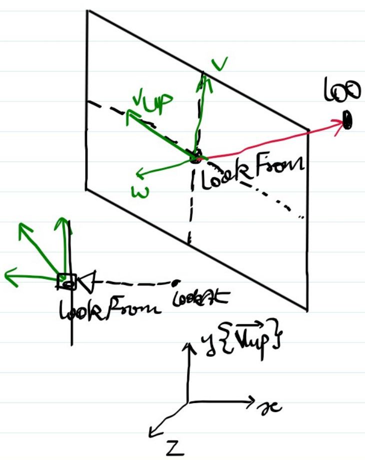


$\boxed{h = \tan\left(\frac{\theta}{2}\right)}$ *

viewport_Height = $2.0 * h$

Viewport_Height = aspect ratio * $h_{viewport}$

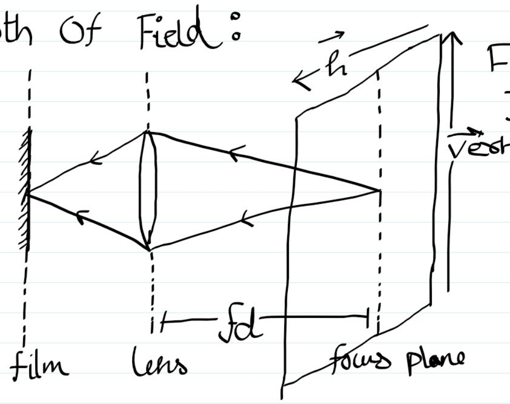
2) Position & Orientation :



$\vec{v}_{up} = \vec{y}$

$$\boxed{\begin{aligned} \vec{w} &= \text{lookFrom} - \text{LookAt} \\ \vec{u} &= \frac{(\vec{V}_{up} \times \vec{w})}{|\vec{V}_{up} \times \vec{w}|} \\ \vec{v} &= \vec{w} \times \vec{u} \end{aligned}}$$

3) Depth Of Field :



From above, ↑

$$\left.\begin{aligned} \vec{h} &= fd * \text{viewportHeight} * \vec{u} \\ \vec{vert} &= fd * \text{viewportWidth} * \vec{v} \end{aligned}\right\}$$