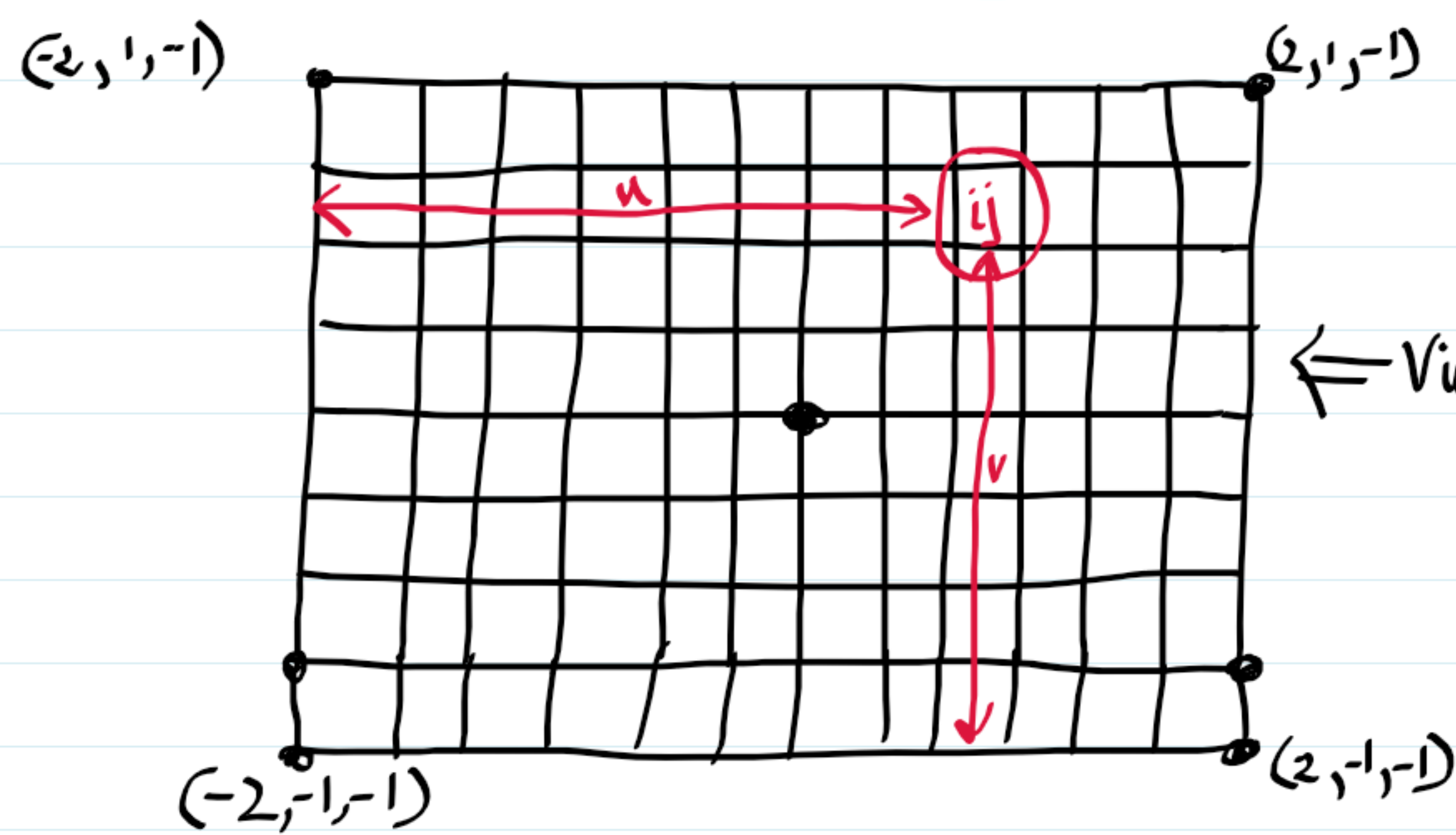


CG NOTES VOL. 1



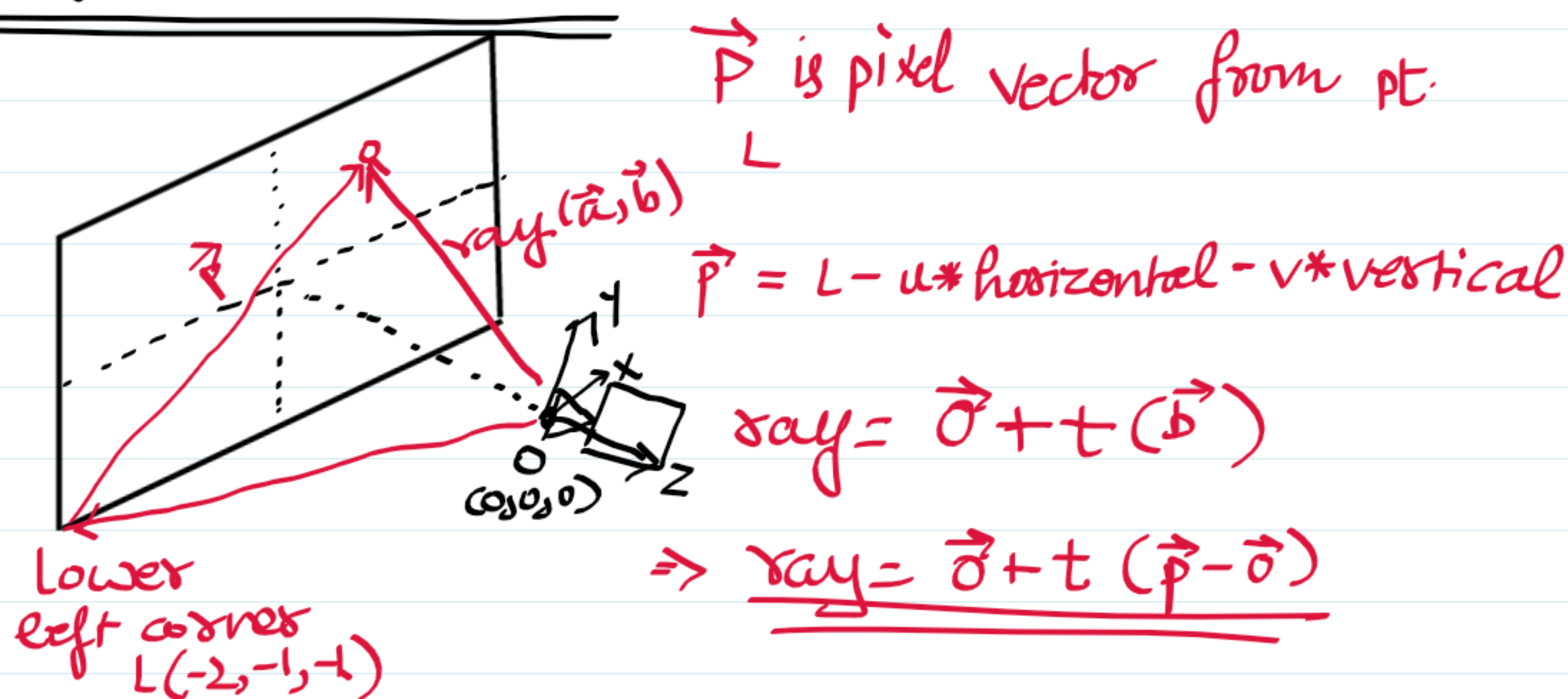
$$u = \frac{i}{width-1}; v = \frac{j}{height-1}$$

For multisampling

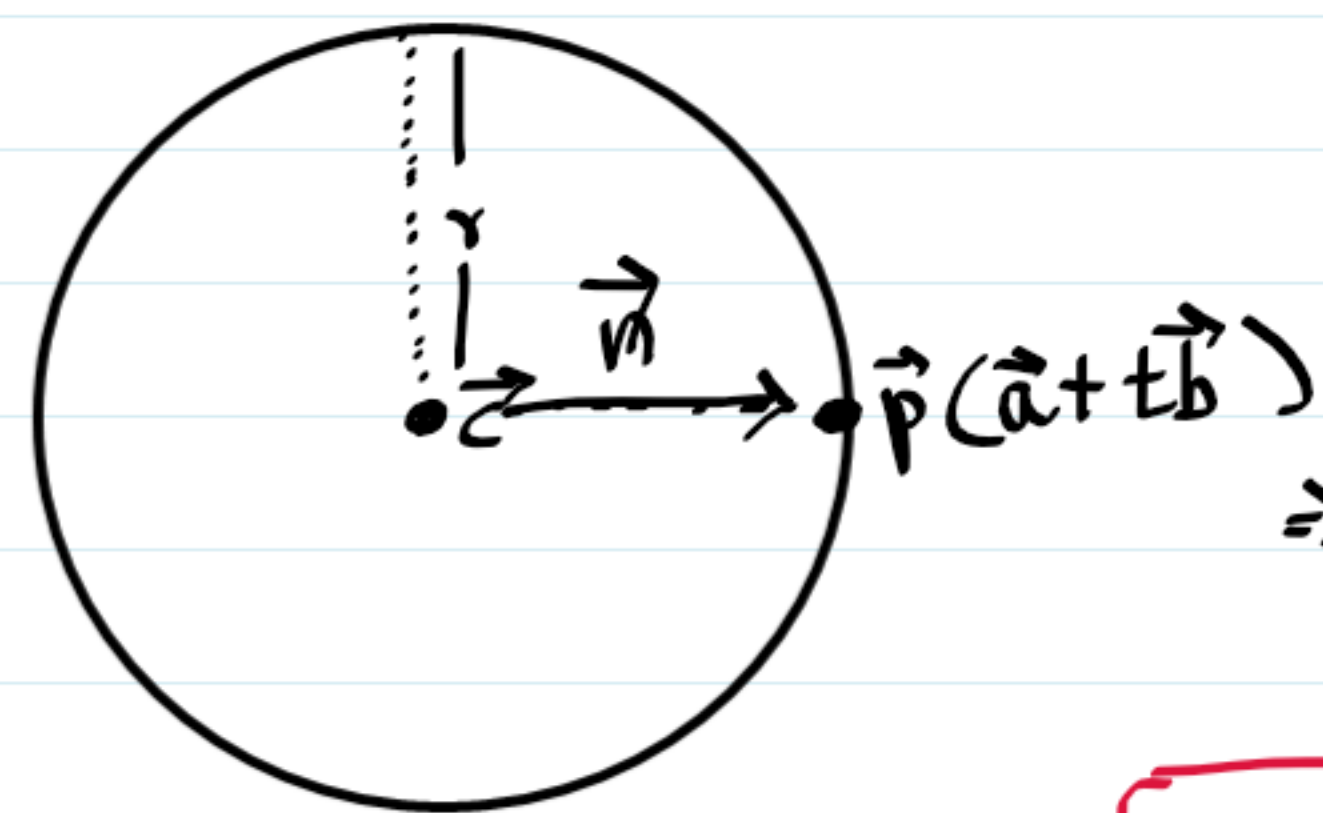
$$u = \frac{i + \text{random}(0,1)}{width-1}; v = \frac{j + \text{random}(0,1)}{height-1}$$

Viewport

Casting rays through every pixel:



Intersection of ray w/ a sphere



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

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

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

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2A}$$

$$\vec{n} = \vec{P} - \vec{C}$$

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

struct hit_record

{ \vec{p}, \vec{n}, t }

world::hit()

closest = max_t;

for each (obj: world)

if (obj->hit(ray, t_min, closest, temp_rec))

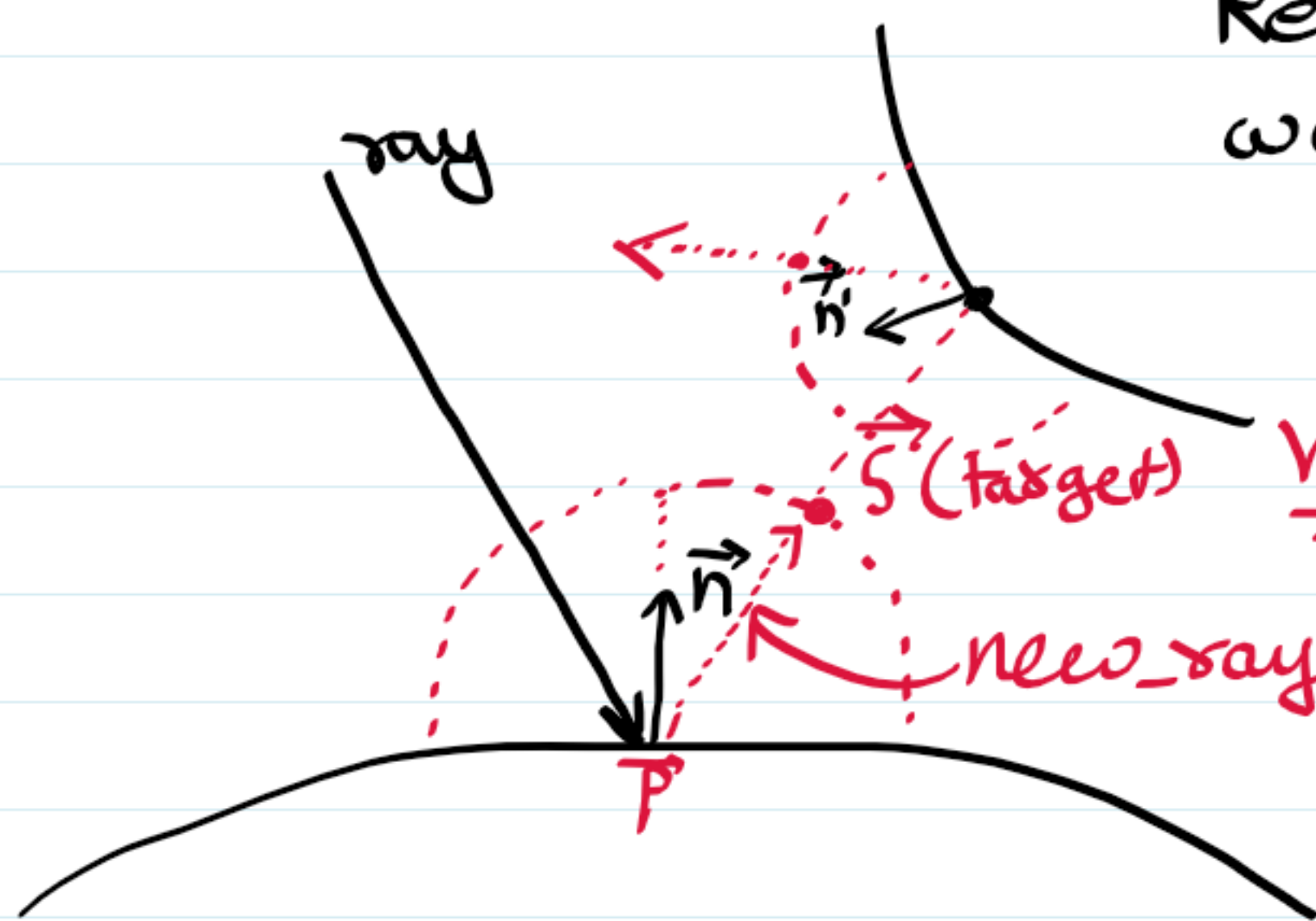
closest = temp_rec.t;

rec = temp_rec;

}

Diffuse Reflection:

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



$$\text{new_ray} = \vec{P} + t(\vec{S} - \vec{P})$$

$$\vec{S} = \vec{P} + \vec{n} + \text{random_vec}()$$