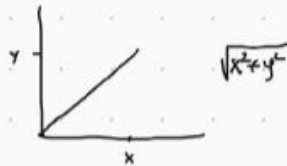


ray origin eyepoint (0,0,0)
dir (x₀, y₀, z₀)

$$\text{len} = \sqrt{x_0^2 + y_0^2 + z_0^2}$$

$$\text{normalized vector} = \left(\frac{x}{\text{len}}, \frac{y}{\text{len}}, \frac{z}{\text{len}} \right)$$



typedef struct {

VP-T origin;

VP-T dir;

}RAY-T;

VP-T;

main

set sphere (0,0,0) r=2

sphere_color 1 0 0

light loc 5, 10, 0

init the img file

for y

for x

set ray origin + dir (normalize)

intersect_sphere

true - red

false - black

PPM

P6
1000 1000
255

R5DR5DR5R...

intersection test
ray

$$A = 1$$

$$B = 2 * (dir_x * (or_x - ctr_x) + dir_y * (or_y - ctr_y) + dir_z * (or_z - ctr_z))$$

$$C = (or_x - ctr_x)^2 + (or_y - ctr_y)^2 + (or_z - ctr_z)^2 - rad^2$$

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

compute discrimin first

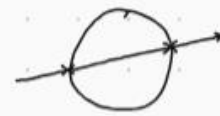
if discrimin ≤ 0, no intersection

smaller of 2 t's

return int

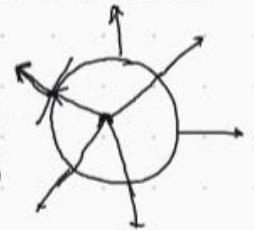
sphere origin radius

int intersect_sphere (ray, sphere, *t, *int_pt, *normal)



$$\text{int_pt} = or + t * dir$$

substant:
intpt - ctr =
normal
(normalization)



printf ("%c%c%c", (unsigned char) R,
(),
());

t > img.ppm