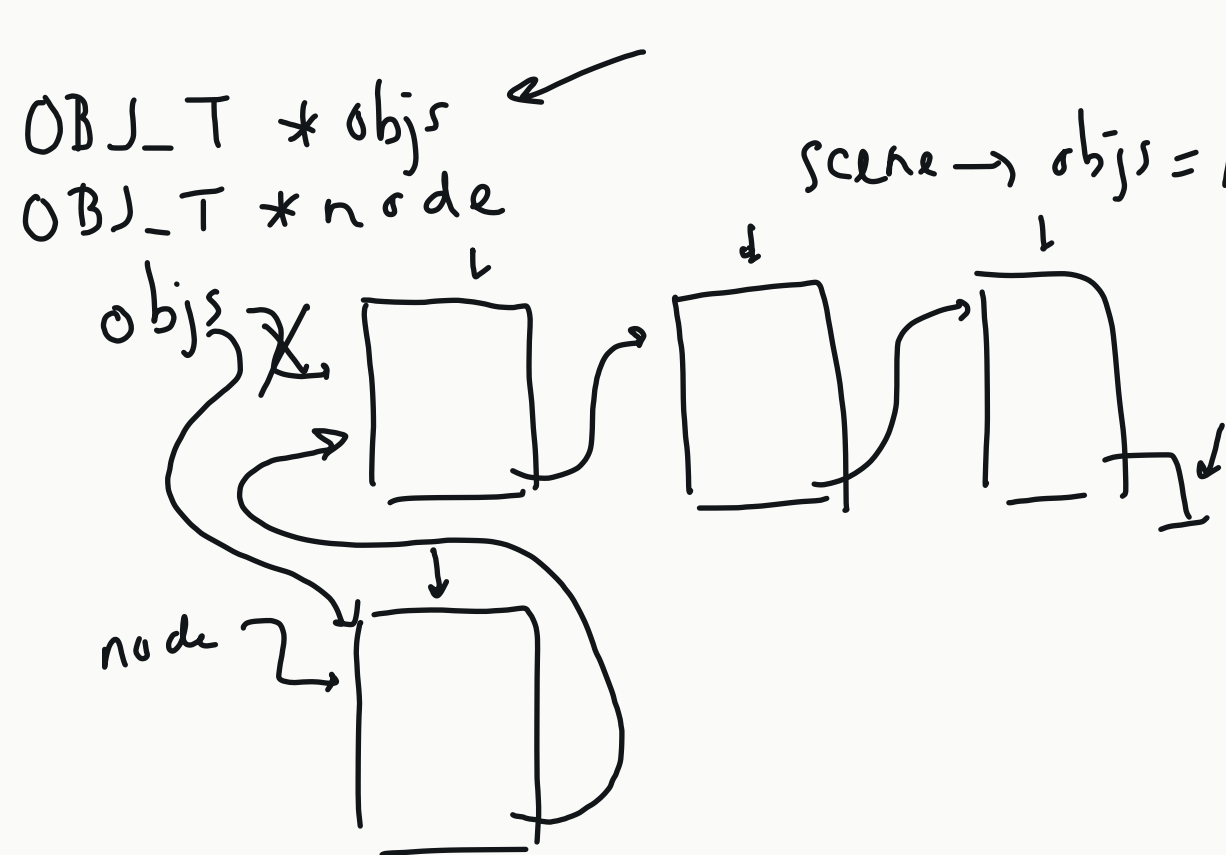


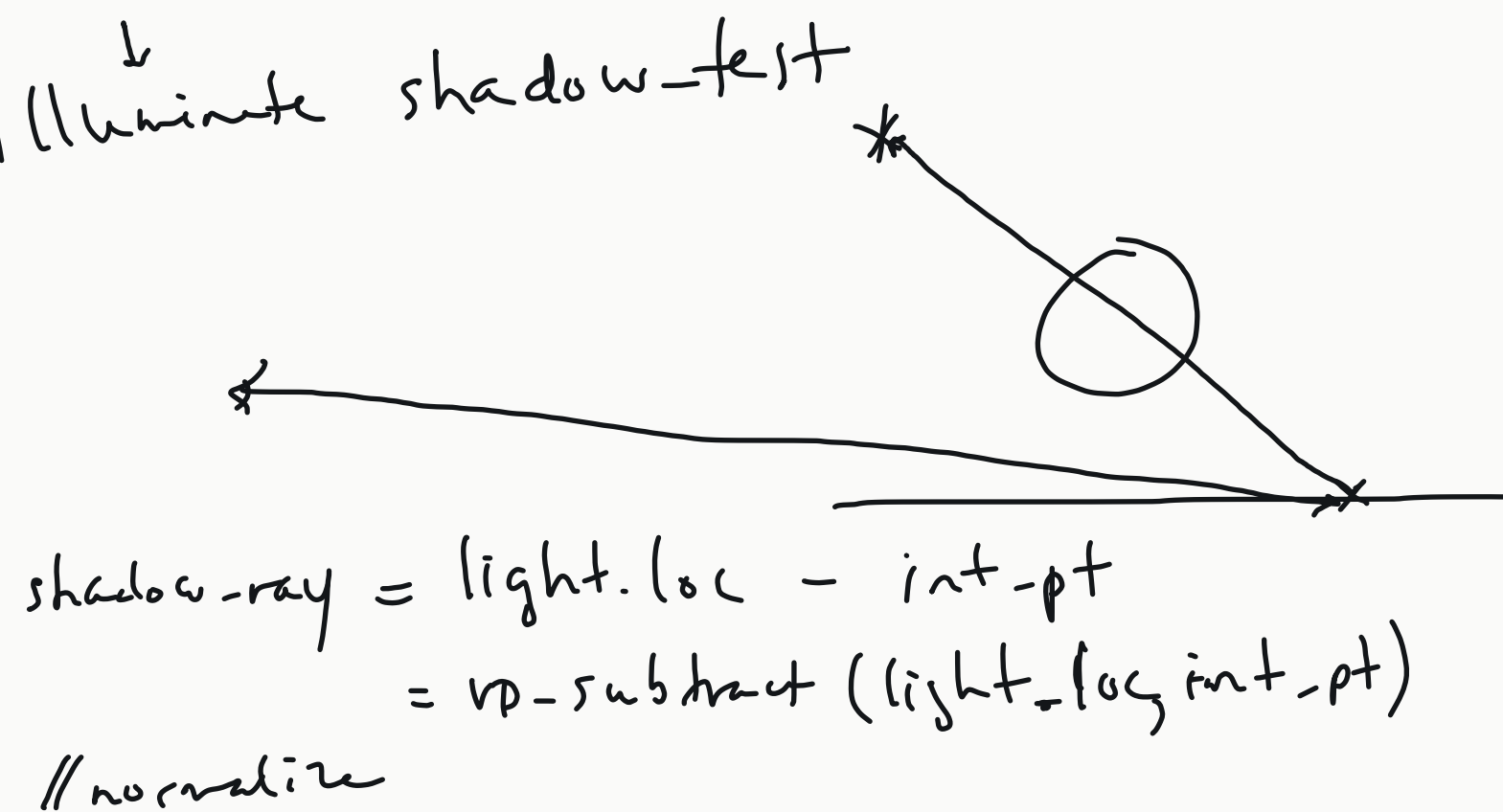


node → next = objs
objs = node

node = (OBJ_T *) malloc (sizeof (OBJ_T))

node → sphere.ctr.x =

vp.h
vp.c
rt.h
rt.c main trace init
sphere.h prototype
sphere.c intersect-sphere
plane.h
plane.c
light.h
light.c illuminate shadow-test



void init (scene)

// open input file
while (fscanf (fp, "%c", &type))
if (type == 's')
// sphere

scene → objs

SCENE_T

OBJ_T * objs
LIGHT_T light
start-x
start-y
pixel size

OBJ_T

struct OBJ * next

trace

scene

OBJ_T * curr;

for (curr = scene.objs; curr != NULL;
curr = curr → next)

if ((*curr → intersect) (...)
if (t < closest_t)

File I/O

stdio.h
stdlib.h

FILE * fp filename
fp = fopen (" " , " ")

fprintf (fp, "%c%c%c", ...)

illuminate

call shadow-test
after ambient
before diffuse + specular