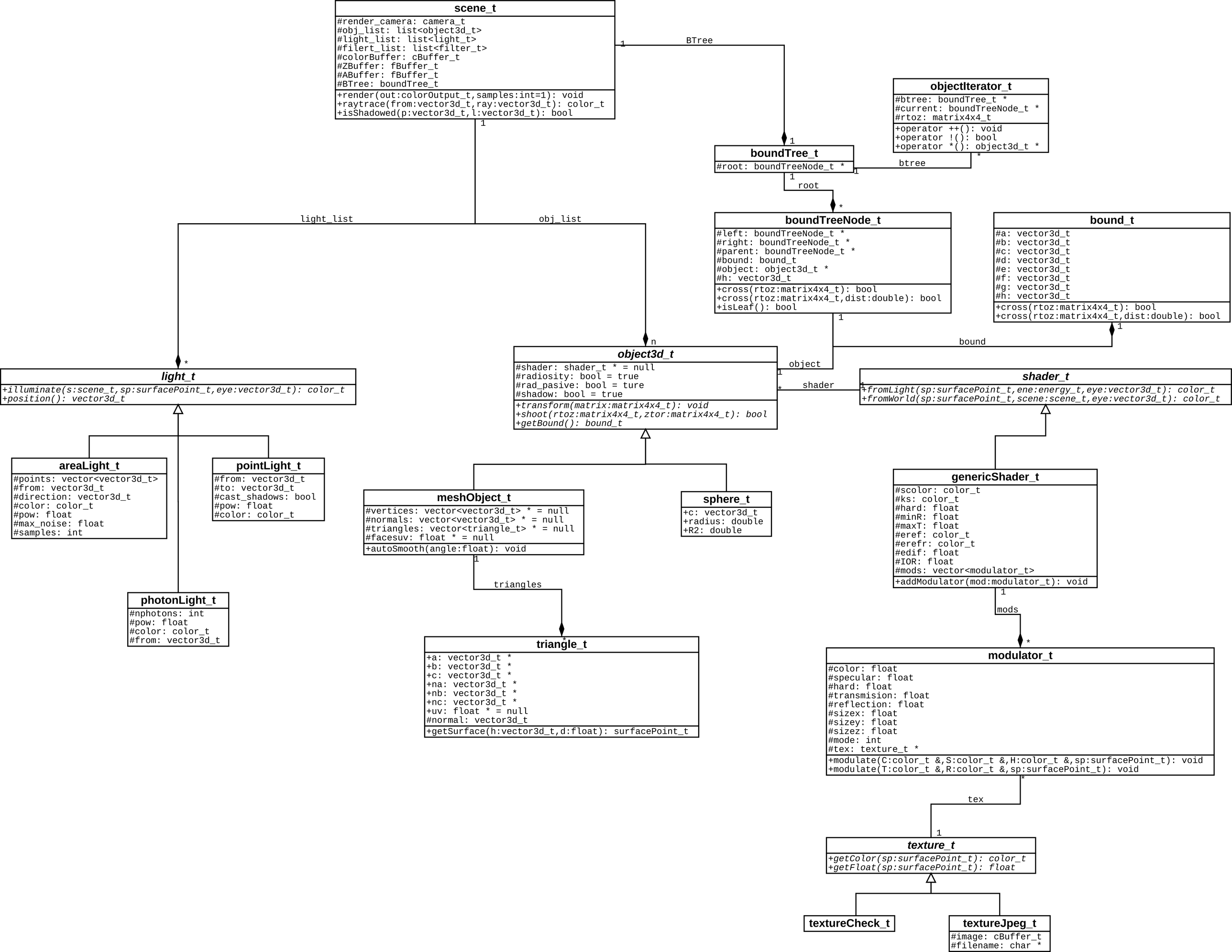
scene_t
#render_camera: camera_t
#obj_list: list<object3d_t>
#light_list: list<light_t>
#filert_list: list<filter_t>
#colorBuffer: cBuffer_t
#ZBuffer: fBuffer_t
#ABuffer: fBuffer_t
#BTree: boundTree_t
+render(out:colorOutput_t,samples:int=1): void
+raytrace(from:vector3d_t,ray:vector3d_t): color_t
+isShadowed(p:vector3d_t,l:vector3d_t): bool
BTree
1
1
objectIterator_t
#btree: boundTree_t *
#current: boundTreeNode_t *
#rtoz: matrix4x4_t
+operator ++(): void
+operator !(): bool
+operator *(): object3d_t *
*
btree
1
boundTree_t
#root: boundTreeNode_t *
1
root
*
light_list
obj_list
boundTreeNode_t
#left: boundTreeNode_t *
#right: boundTreeNode_t *
#parent: boundTreeNode_t *
#bound: bound_t
#object: object3d_t *
#h: vector3d_t
+cross(rtoz:matrix4x4_t): bool
+cross(rtoz:matrix4x4_t,dist:double): bool
+isLeaf(): bool
bound_t
#a: vector3d_t
#b: vector3d_t
#c: vector3d_t
#d: vector3d_t
#e: vector3d_t
#f: vector3d_t
#g: vector3d_t
#h: vector3d_t
+cross(rtoz:matrix4x4_t): bool
+cross(rtoz:matrix4x4_t,dist:double): bool
1
1
bound
n
*
object
1
shader
*
1
light_t
+illuminate(s:scene_t,sp:surfacePoint_t,eye:vector3d_t): color_t
+position(): vector3d_t
object3d_t
#shader: shader_t * = null
#radiosity: bool = true
#rad_pasive: bool = ture
#shadow: bool = true
+transform(matrix:matrix4x4_t): void
+shoot(rtoz:matrix4x4_t,ztor:matrix4x4_t): bool
+getBound(): bound_t
shader_t
+fromLight(sp:surfacePoint_t,ene:energy_t,eye:vector3d_t): color_t
+fromWorld(sp:surfacePoint_t,scene:scene_t,eye:vector3d_t): color_t
areaLight_t
#points: vector<vector3d_t>
#from: vector3d_t
#direction: vector3d_t
#color: color_t
#pow: float
#max_noise: float
#samples: int
pointLight_t
#from: vector3d_t
#to: vector3d_t
#cast_shadows: bool
#pow: float
#color: color_t
meshObject_t
#vertices: vector<vector3d_t> * = null
#normals: vector<vector3d_t> * = null
#triangles: vector<triangle_t> * = null
#facesuv: float * = null
+autoSmooth(angle:float): void
sphere_t
+c: vector3d_t
+radius: double
+R2: double
genericShader_t
#scolor: color_t
#ks: color_t
#hard: float
#minR: float
#maxT: float
#eref: color_t
#erefr: color_t
#edif: float
#IOR: float
#mods: vector<modulator_t>
+addModulator(mod:modulator_t): void
photonLight_t
#nphotons: int
#pow: float
#color: color_t
#from: vector3d_t
1
triangles
*
triangle_t
+a: vector3d_t *
+b: vector3d_t *
+c: vector3d_t *
+na: vector3d_t *
+nb: vector3d_t *
+nc: vector3d_t *
+uv: float * = null
#normal: vector3d_t
+getSurface(h:vector3d_t,d:float): surfacePoint_t
1
mods
*
modulator_t
#color: float
#specular: float
#hard: float
#transmision: float
#reflection: float
#sizex: float
#sizey: float
#sizez: float
#mode: int
#tex: texture_t *
+modulate(C:color_t &,S:color_t &,H:color_t &,sp:surfacePoint_t): void
+modulate(T:color_t &,R:color_t &,sp:surfacePoint_t): void
*
tex
1
texture_t
+getColor(sp:surfacePoint_t): color_t
+getFloat(sp:surfacePoint_t): float
textureCheck_t
textureJpeg_t
#image: cBuffer_t
#filename: char *