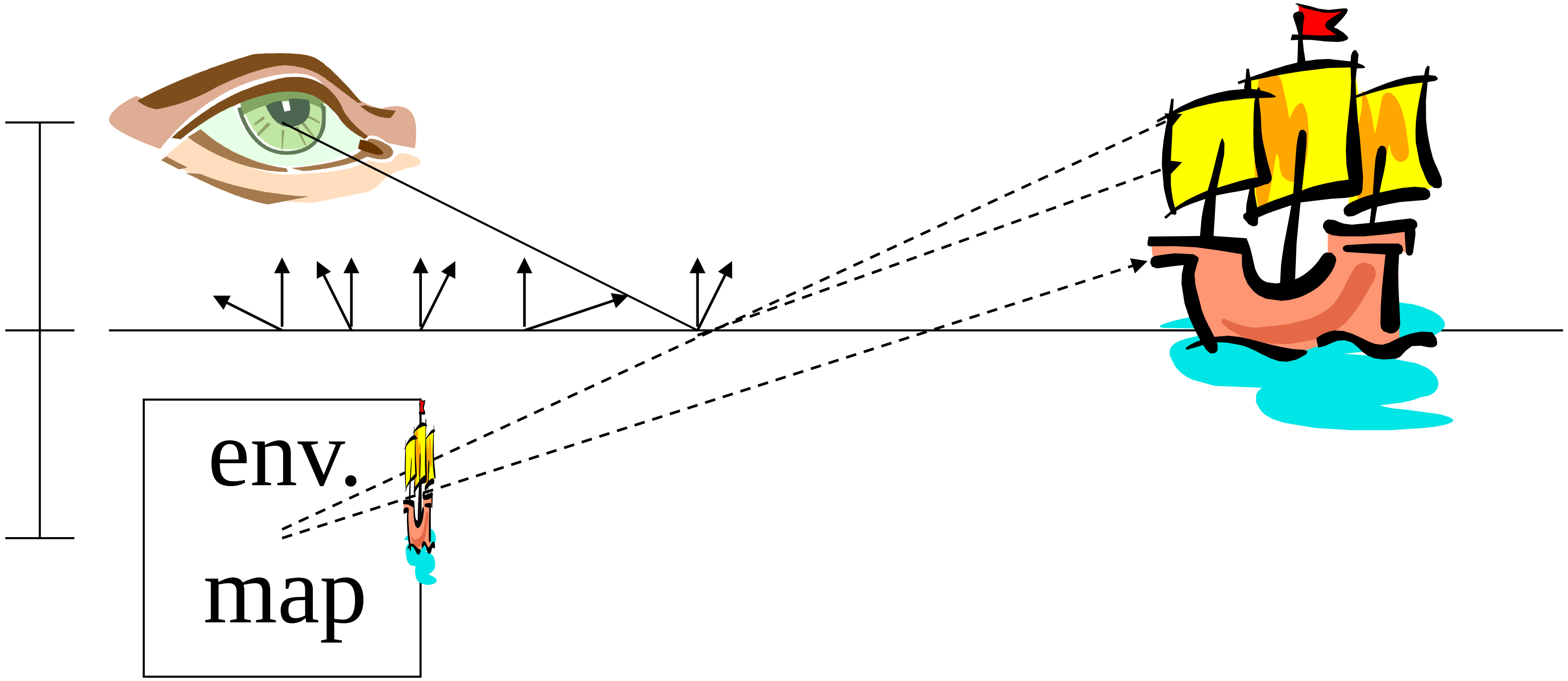
Ray Tracing

John C. Hart

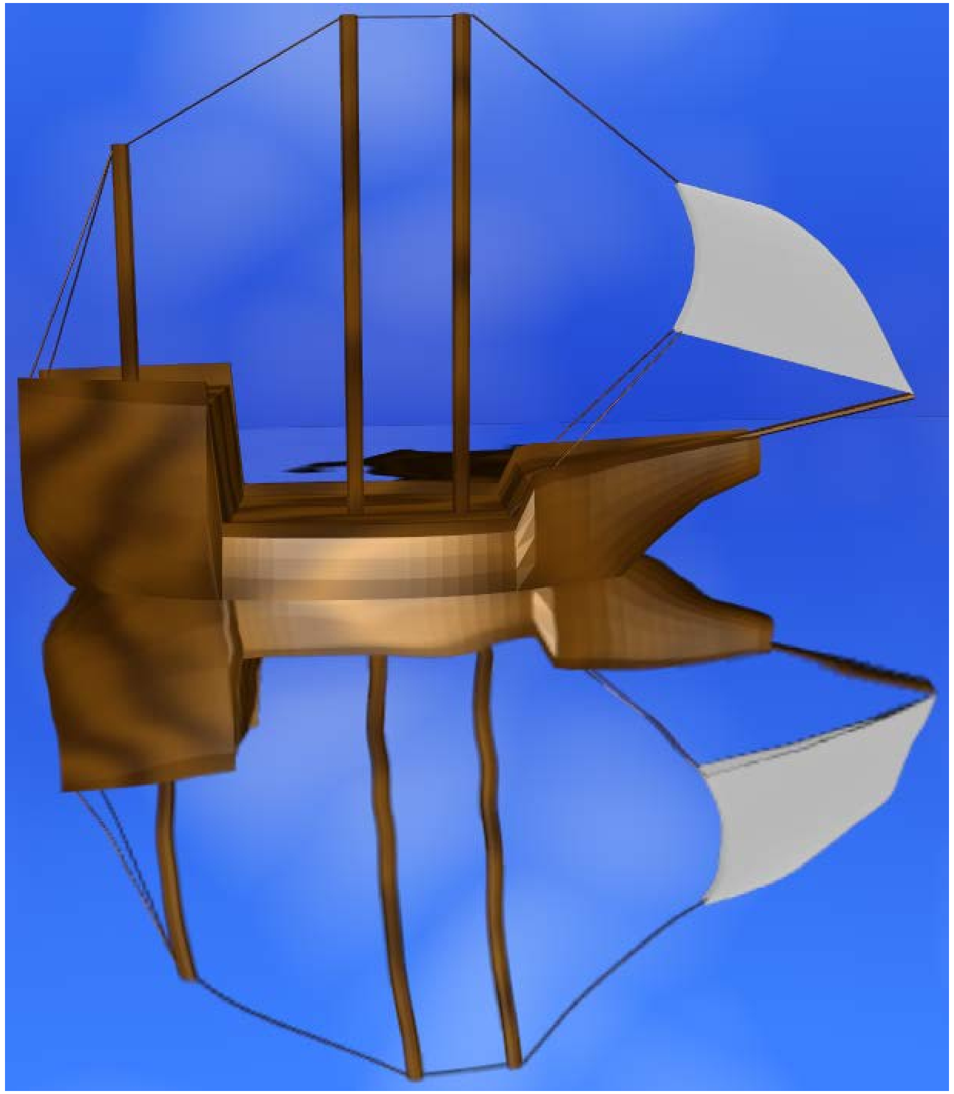
CS 418

Interactive Computer Graphics

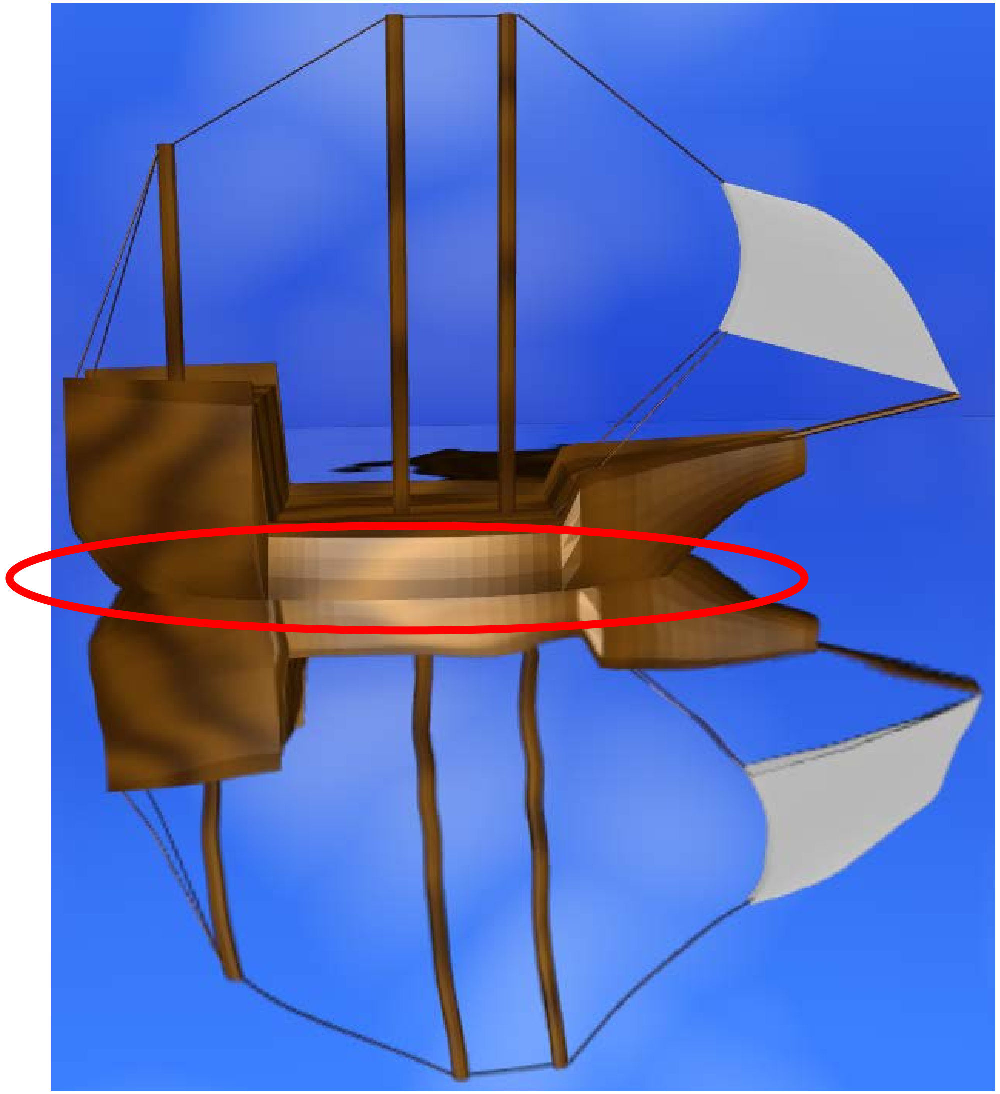
Environment Mapped Bump Mapping



What’s Wrong with this Picture?

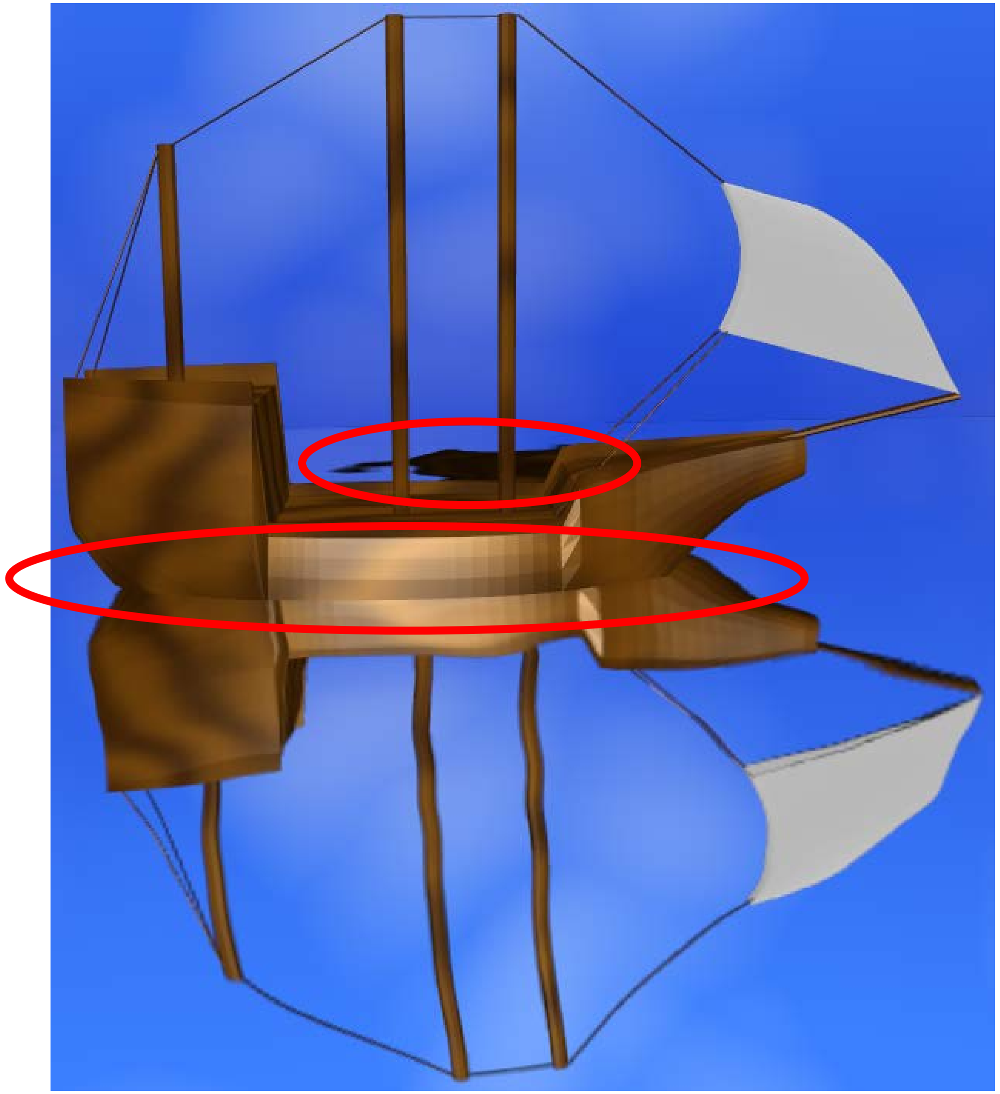


What’s Wrong with this Picture?



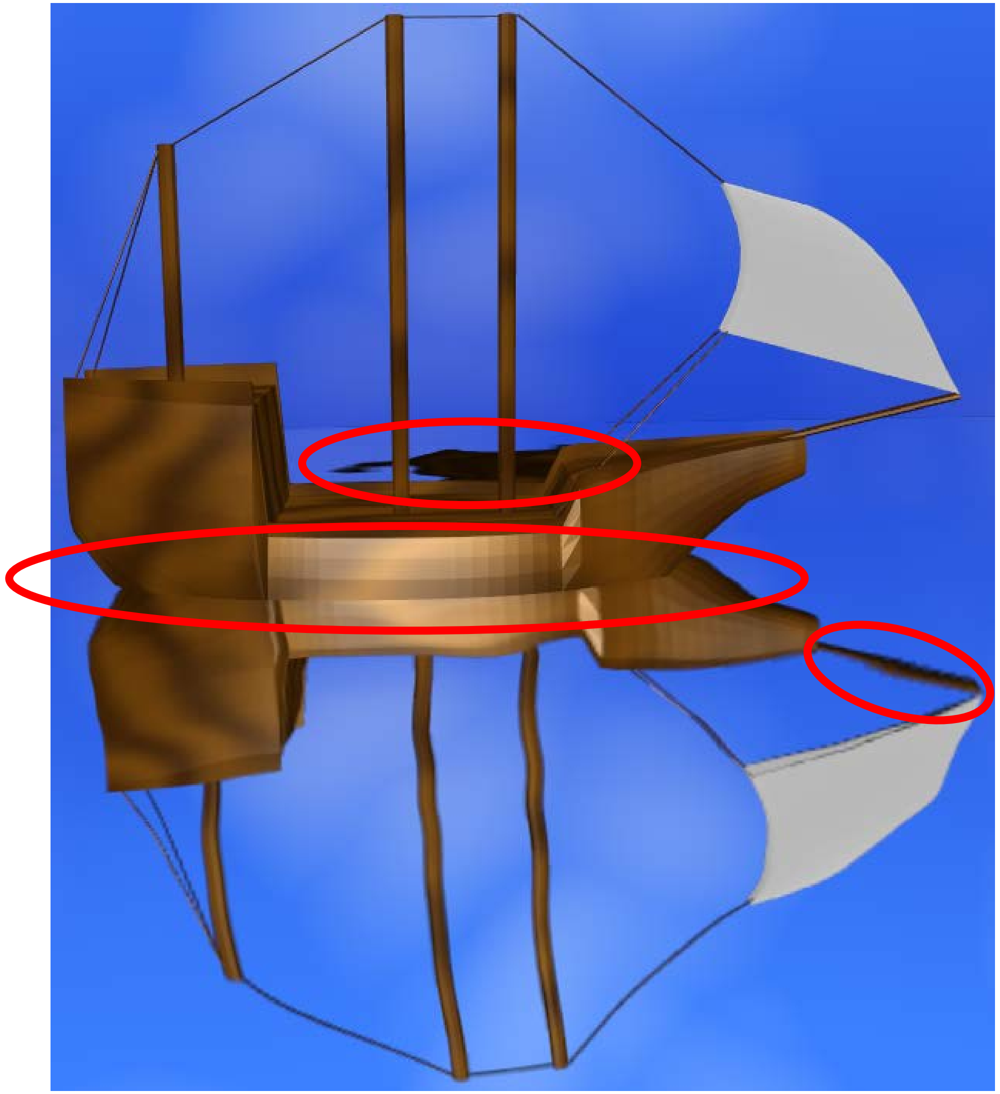
1.Reflection doesn’t meet boat

What’s Wrong with this Picture?



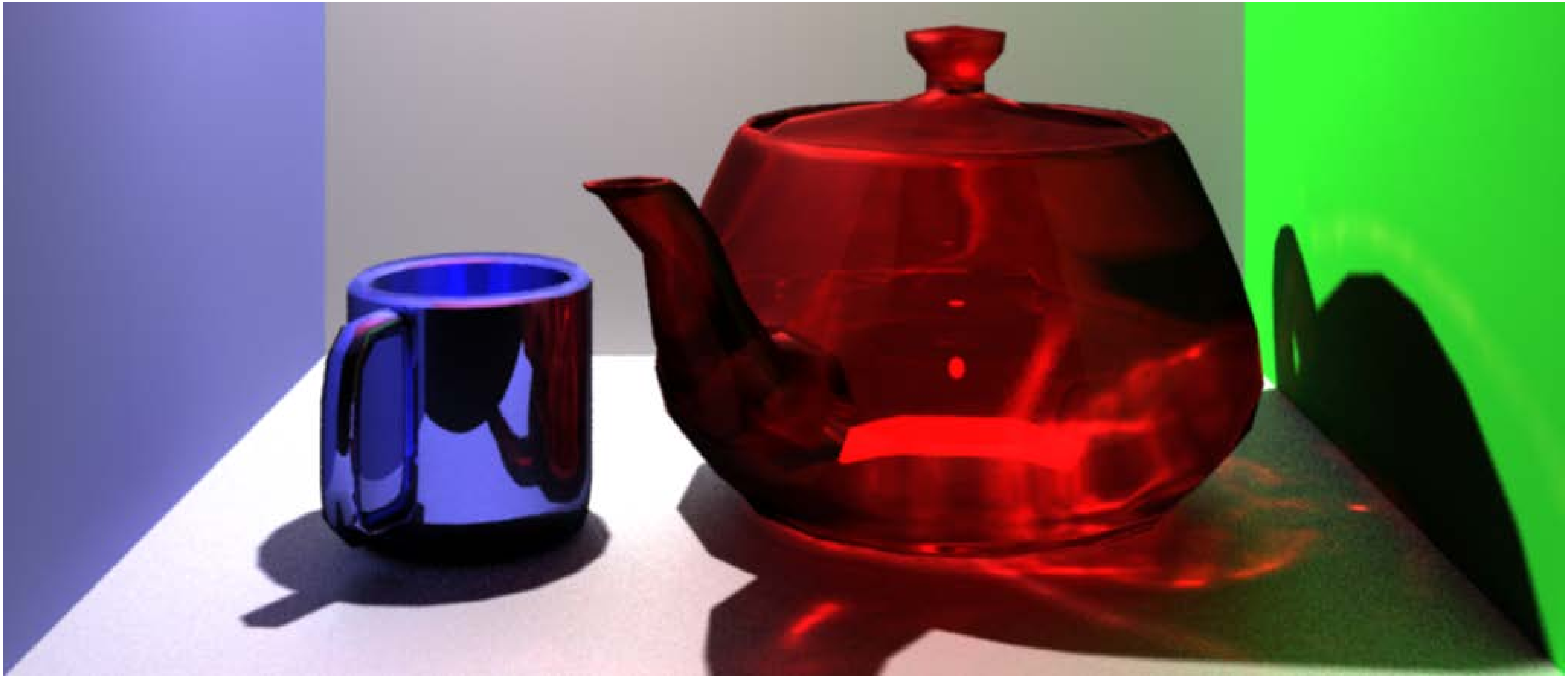
1. Reflection doesn’t meet boat
2. Reflection behind the boat

What’s Wrong with this Picture?

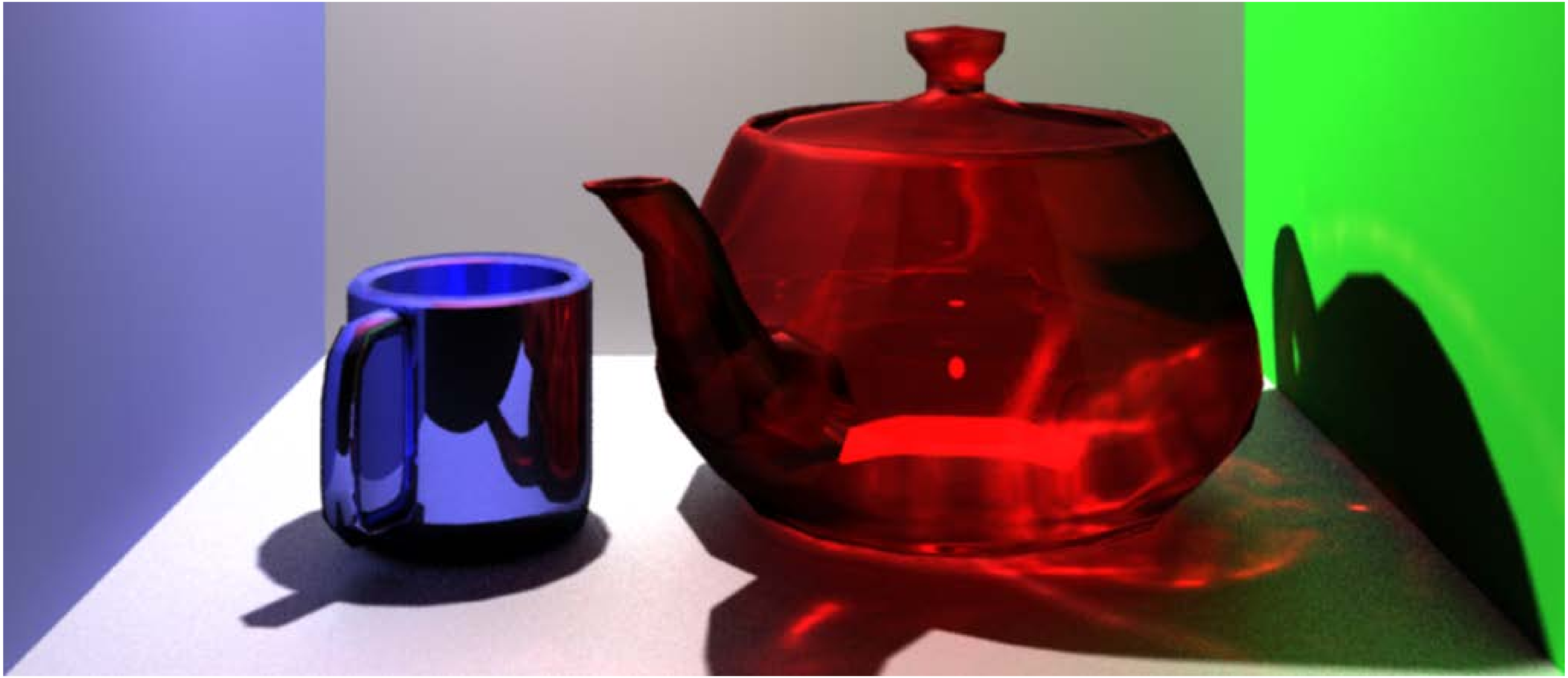


1. Reflection doesn’t meet boat
2. Reflection behind the boat
3. Environment map magnified

How Can We Do This?



How Can We Do This? *Ray Tracing*



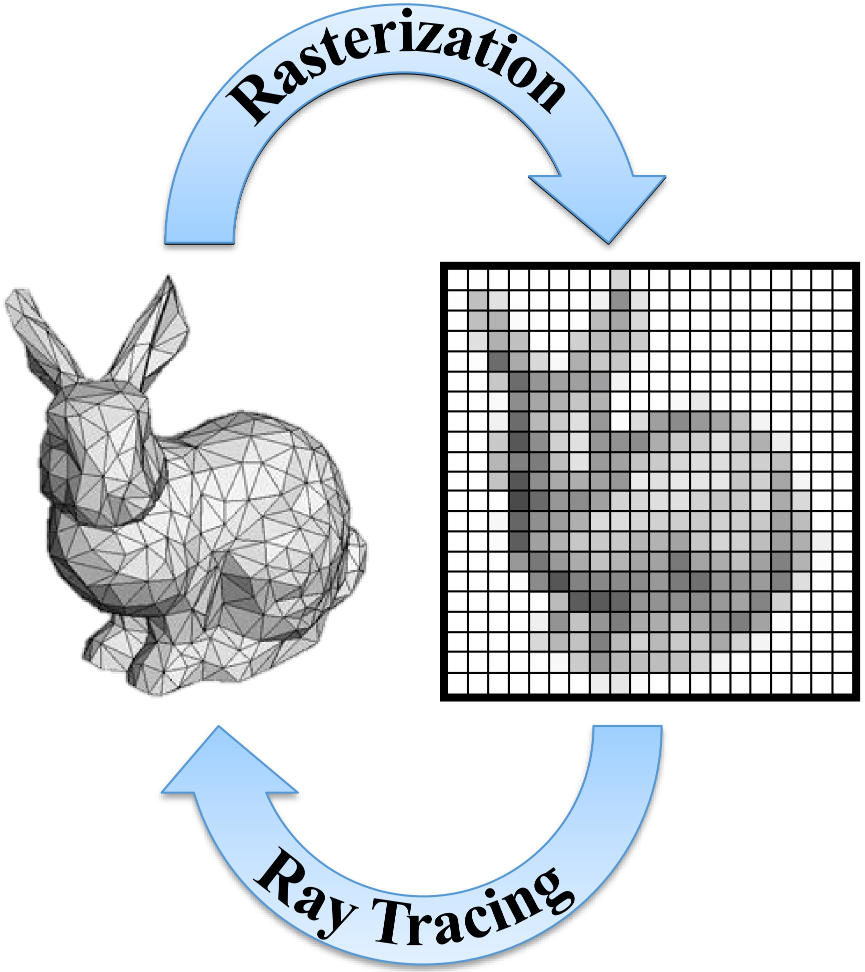
Ray Tracing v. Rasterization

- **Rasterization**

```
For each primitive
    For each pixel
        Render pixel
```

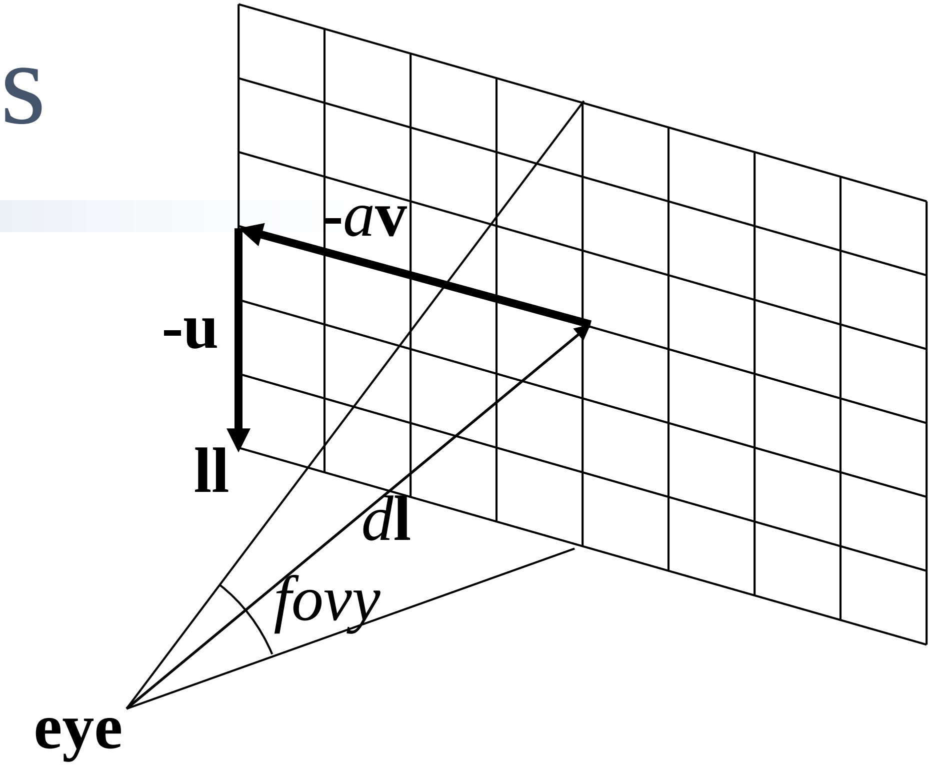
- **Ray Tracing**

```
For each pixel
    For each primitive
        Render pixel
```



Pixels in World Coords

- aspect ratio $a = w/h$
- focal length $d = 1/\tan(\text{fovy}/2)$

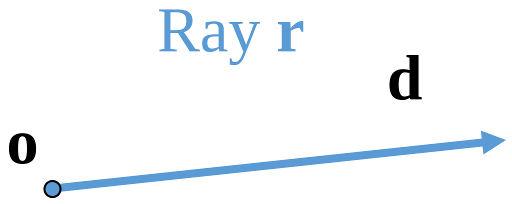


$$\mathbf{ll} = \mathbf{eye} + d\,\mathbf{l} - a\,\mathbf{v} - \mathbf{u}$$

```
for (j = 0; j < VRES; j++) {
   for (i = 0; i < HRES; i++) {
        p = ll + 2av (double)i/HRES + 2u (double)j/VRES;
        color = TraceRay(Ray(eye, p - eye));
        plot(i,j,color);
   }
}
```

TraceRay

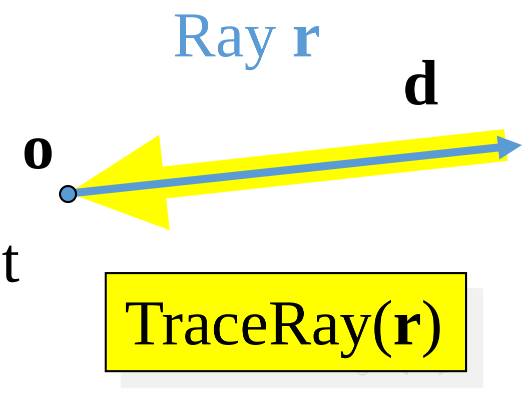
- TraceRay(**r** = (**o**,**d**)) returns the intensity of light arriving at the ray anchor **o** in the opposite direction (**-d**)
- Invoked with ray parameter only
  - Better if object database is global
  - Best if TraceRay is a member function of object database
- Returns intensity across the visible spectrum
  - e.g. an RGB triple



```
Color TraceRay(Ray r, int depth) {
  Color c = background;
  if (!depth) return c;
  if ((hit = Intersect(r)) != NULL) {
    hit->depth = depth – 1;
    c = hit->Shade();
  }
  return c;
}
```

TraceRay

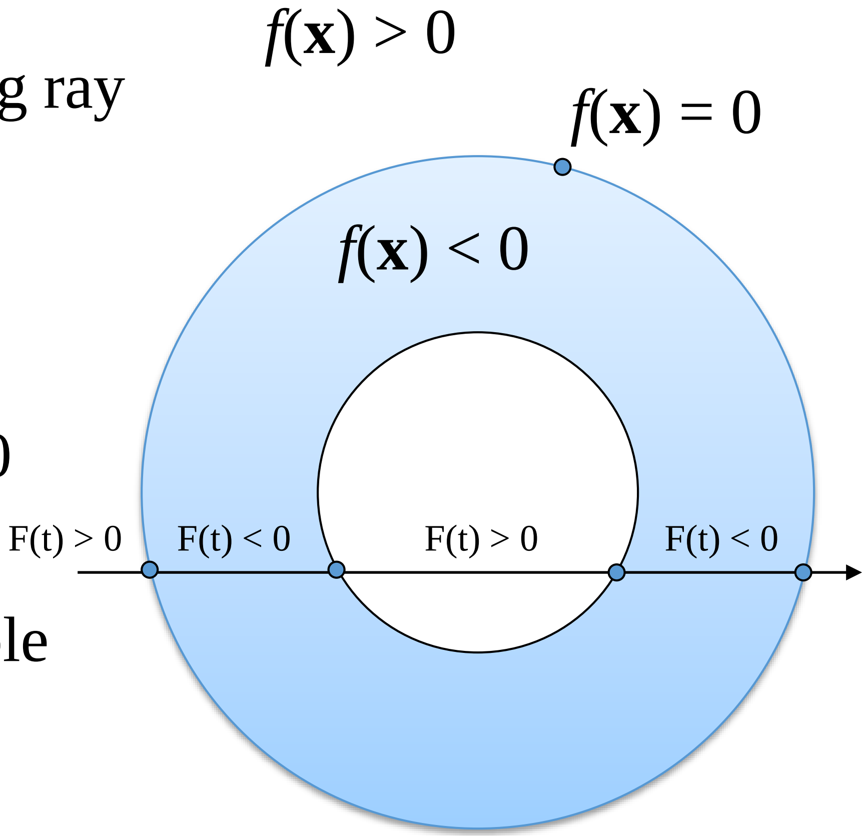
- TraceRay(**r** = (**o**,**d**)) returns the intensity of light arriving at the ray anchor **o** in the opposite direction (**-d**)
- Invoked with ray parameter only
  - Better if object database is global
  - Best if TraceRay is a member function of object database
- Returns intensity across the visible spectrum
  - e.g. an RGB triple



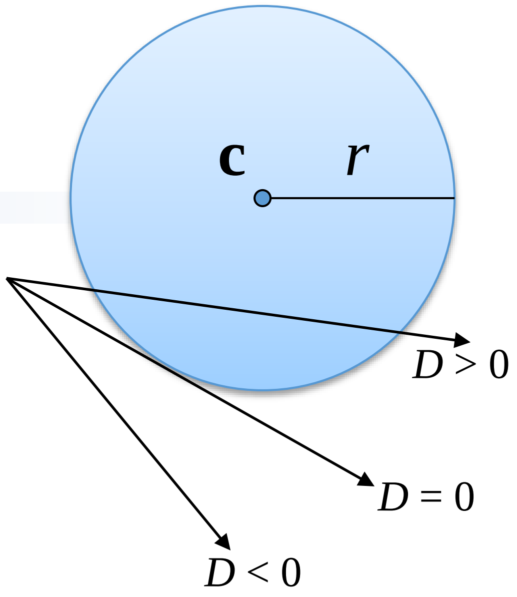
```
Color TraceRay(Ray r, int depth) {
  Color c = background;
  if (!depth) return c;
  if ((hit = Intersect(r)) != NULL) {
    hit->depth = depth – 1;
    c = hit->Shade();
  }
  return c;
}
```

Intersection Computation

- Parametric ray: $\mathbf{r}(t) = \mathbf{o} + t\,\mathbf{d}$
  - $t \geq 0$
  - Since $\|\mathbf{d}\| = 1$, t is distance along ray
- Implicit object: $f(\mathbf{x}) = 0$
  - $f(\mathbf{x}) > 0$ outside, $f(\mathbf{x}) < 0$ inside
  - Or vice-verse, doesn't matter
- Intersection occurs when $f(\mathbf{r}(t)) = 0$
  - Let $F(t) = f(\mathbf{r}(t))$
  - Real function of one real variable
  - Intersection ≡ root finding



Sphere Intersection



$$f(\mathbf{x})=(\mathbf{x} - \mathbf{c})\cdot(\mathbf{x} - \mathbf{c}) - r^2$$

$$f(\mathbf{r}(t)) = (\mathbf{o} + t\,\mathbf{d} - \mathbf{c})\cdot(\mathbf{o} + t\,\mathbf{d} - \mathbf{c}) - r^2$$

$$= \mathbf{d}\cdot\mathbf{d}\,t^2 + 2\,(\mathbf{o}\text{-}\mathbf{c})\cdot\mathbf{d}\,t + (\mathbf{o}\text{-}\mathbf{c})\cdot(\mathbf{o}\text{-}\mathbf{c}) - r^2$$

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

$A = \mathbf{d}\cdot\mathbf{d}$ (= 1 if $\mathbf{d}$ unit length)

$B = 2\,(\mathbf{o}\text{-}\mathbf{c})\cdot\mathbf{d}$

$C = (\mathbf{o}\text{-}\mathbf{c})\cdot(\mathbf{o}\text{-}\mathbf{c}) - r^2$

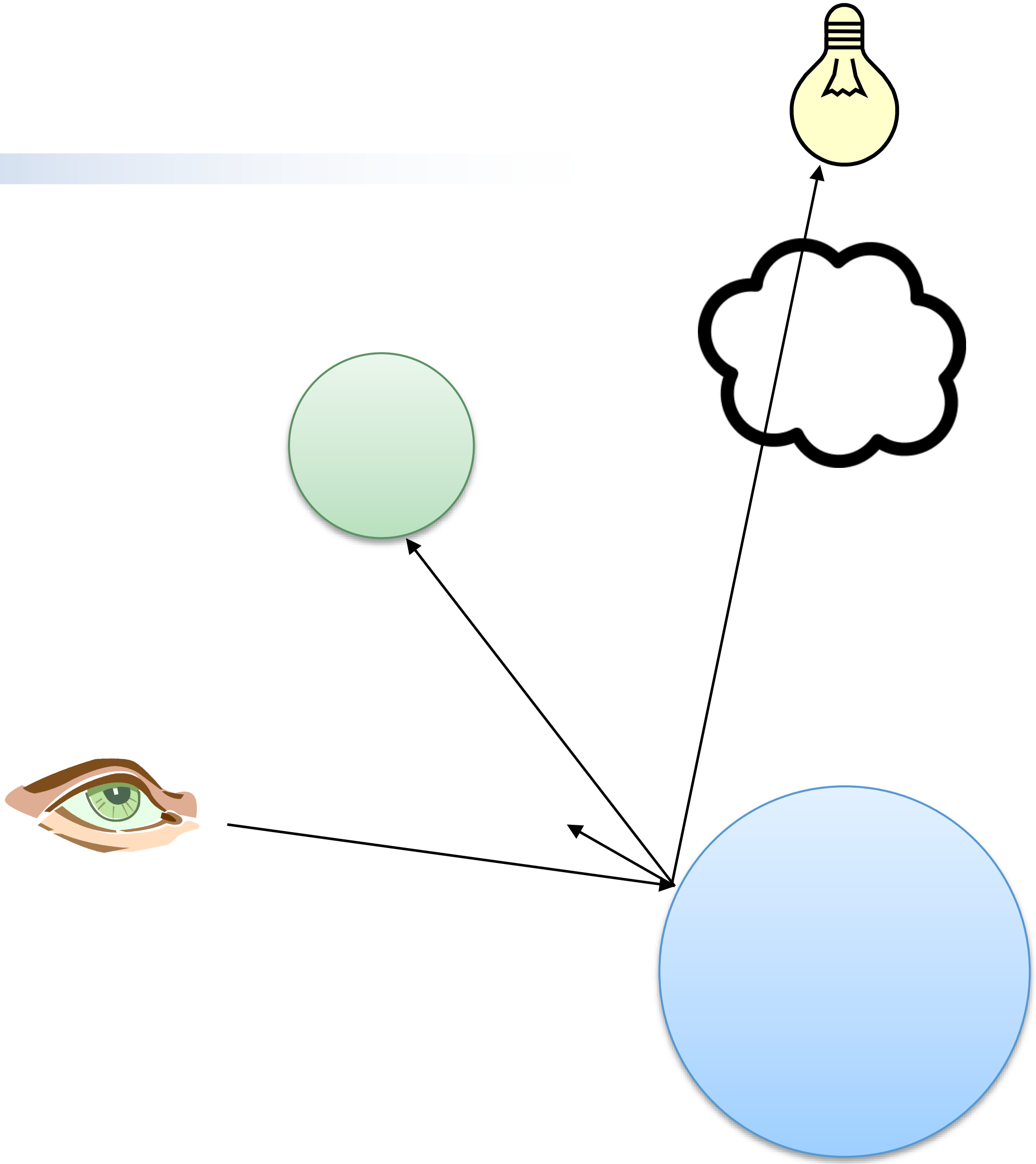
Hit position $\mathbf{x} = \mathbf{o} + t\,\mathbf{d}$

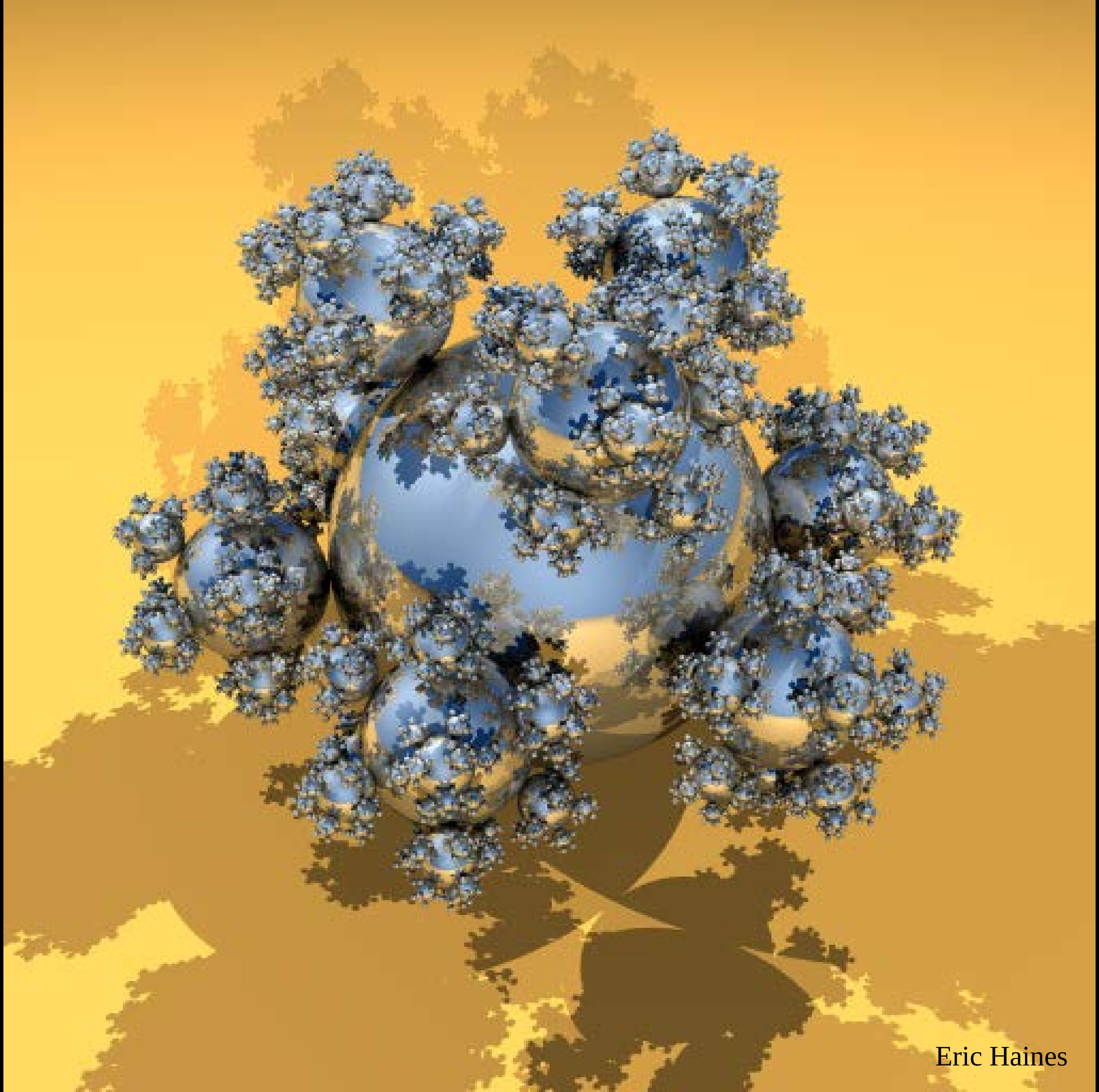
Hit normal $\mathbf{n} = (\mathbf{x} - \mathbf{c})/\|\mathbf{x} - \mathbf{c}\|$

```
D = B*B - 4*AC;
if (D < 0.0) return NULL;
rootD = sqrt(D);
t0 = 0.5*(-B - rootD)/A;
t1 = 0.5*(-B + rootD)/A;
if (t0 >= 0)
  hit->t = t0, return hit;
if (t1 >= 0)
  hit->t = t1, return hit;
return NULL;
```

Shading

- Hit point **x** = **o** + *t****d**
- Shadows
  - Create light vector
  - Trace ray to light source
- Mirrors
  - Reflect a ray about normal
  - Trace reflected ray to determine reflection color
- Glass
  - Use normal to refract ray
  - Trace refracted ray to determine transmitted color





Eric Haines

Nate Carr, Jared Hoberock, Keenan Crane, John C. Hart

© 2006 Board of Trustees of the University of Illinois