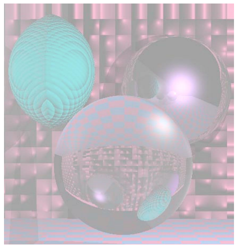


CS 4300
Computer Graphics

Prof. Harriet Fell

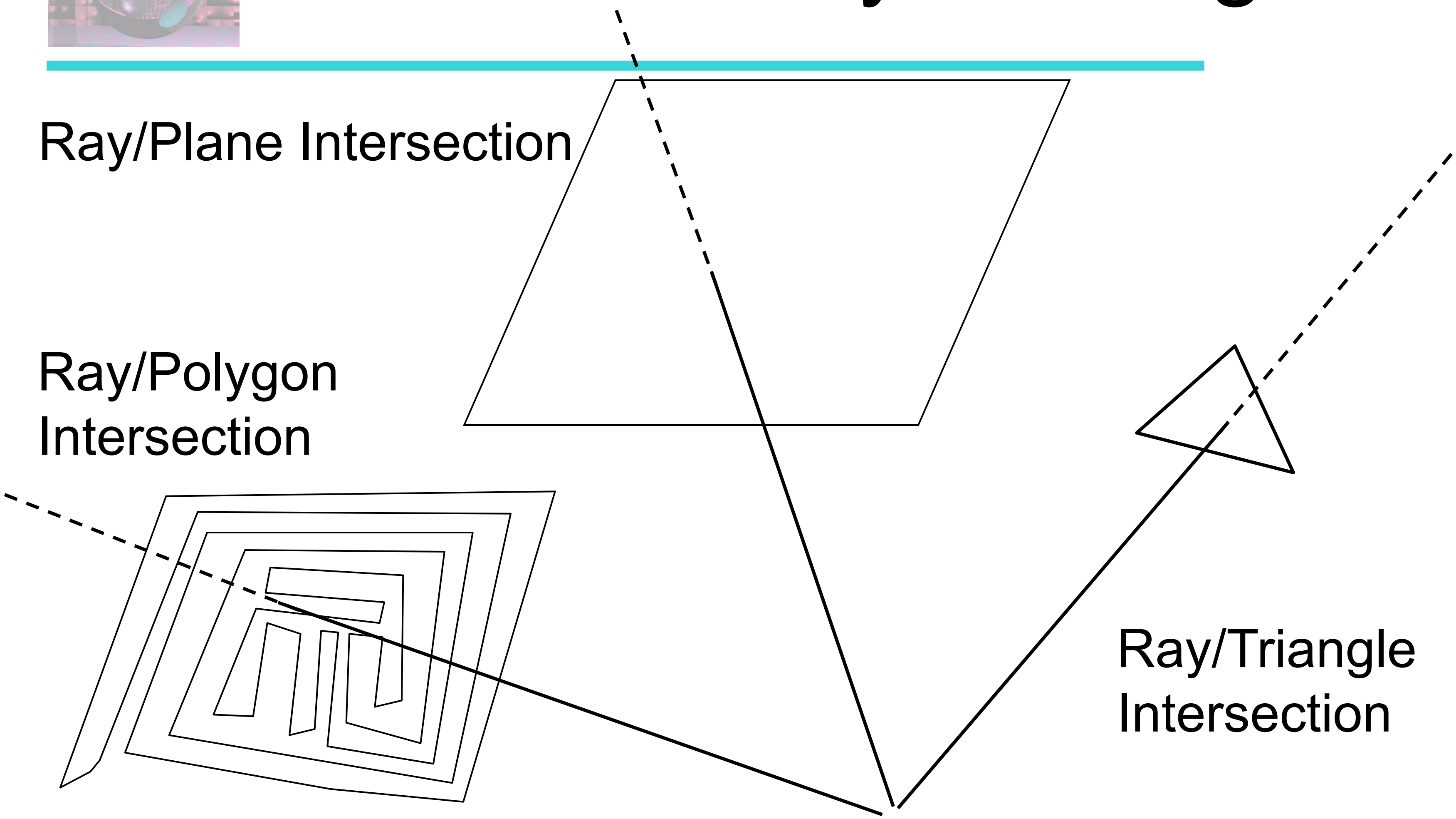
Fall 2011

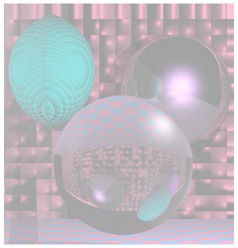
Lecture 27 – November 9, 2011

Topics

- Ray intersections with
  – plane
  – triangle
  – quadrics
- Recursive Ray Tracing

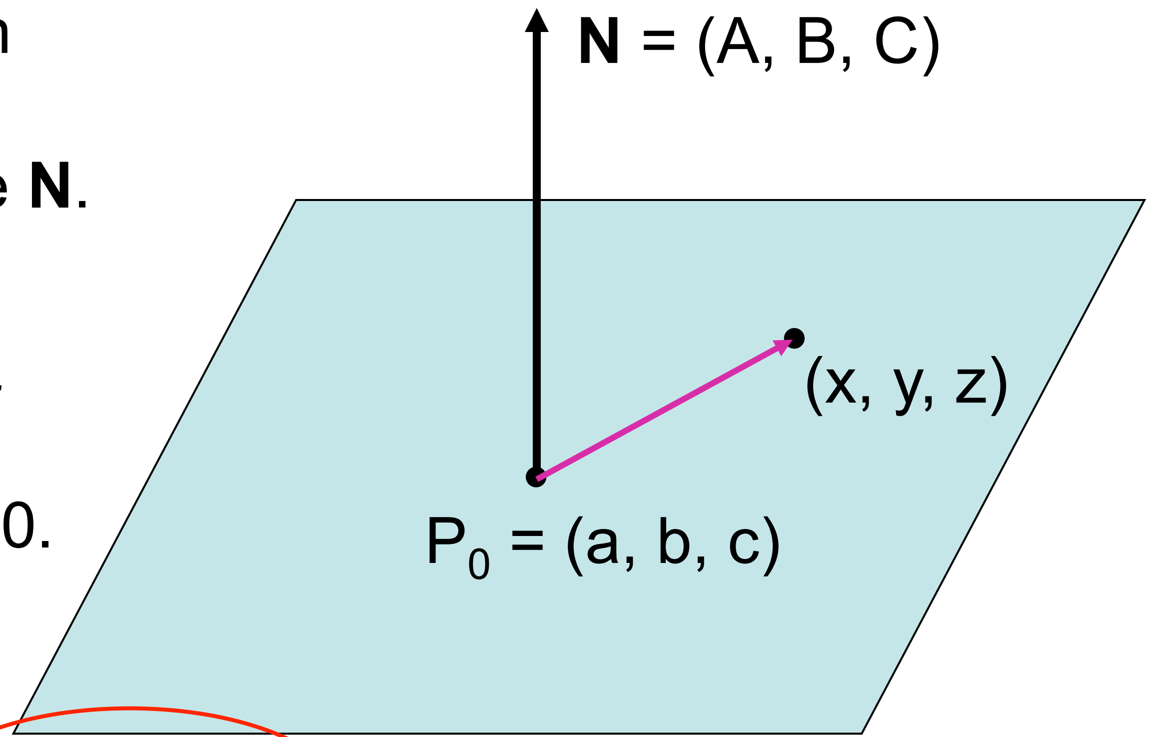
More Ray-Tracing





Equation of a Plane

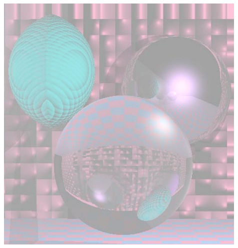
Given a point P_0 on the plane and a normal to the plane **N**.



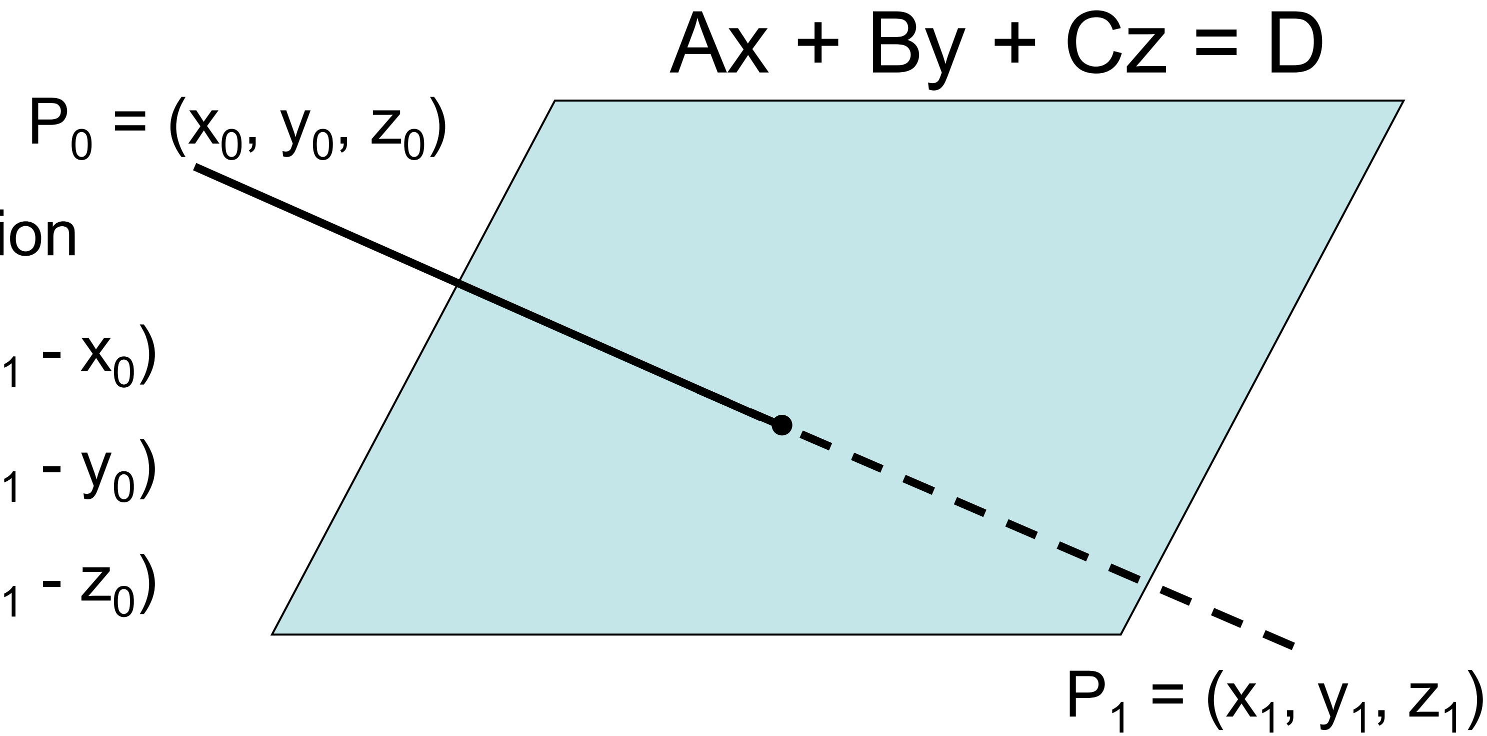
(x, y, z) is on the plane if and only if

$(x-a, y-b, z-c) \cdot \mathbf{N} = 0.$

$$Ax + By + Cz - \underbrace{(Aa + Bb + Cc)}_{D} = 0$$



Ray/Plane Intersection



Ray Equation

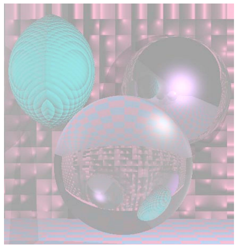
$x = x_0 + t(x_1 - x_0)$

$y = y_0 + t(y_1 - y_0)$

$z = z_0 + t(z_1 - z_0)$

$A(x_0 + t(x_1 - x_0)) + B(y_0 + t(y_1 - y_0)) + C(z_0 + t(z_1 - z_0)) = D$

Solve for t. Find x, y, z.



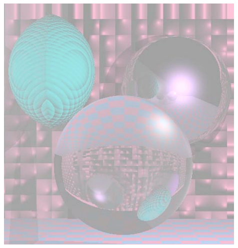
Planes in Your Scenes

- Planes are specified by
  - A, B, C, D or by **N** and P
  - Color and other coefficients are as for spheres
- To search for the nearest object, go through all the spheres and planes and find the smallest t.
- A plane will not be visible if the normal vector (A, B, C) points away from the light.
  - or we see the back of the plane

Ray/Triangle Intersection

Using the Ray/Plane intersection:

– Given the three vertices of the triangle,
  - Find **N**, the normal to the plane containing the triangle.
  - Use **N** and one of the triangle vertices to describe the plane, i.e. Find A, B, C, and D.
  - If the Ray intersects the Plane, find the intersection point and its β and γ.
  - If $0 \leq \beta$ and $0 \leq \gamma$ and $\beta + \gamma \leq 1$, the Ray hits the Triangle.



Ray/Triangle Intersection

Using barycentric coordinates directly: (Shirley pp. 206-208)

Solve

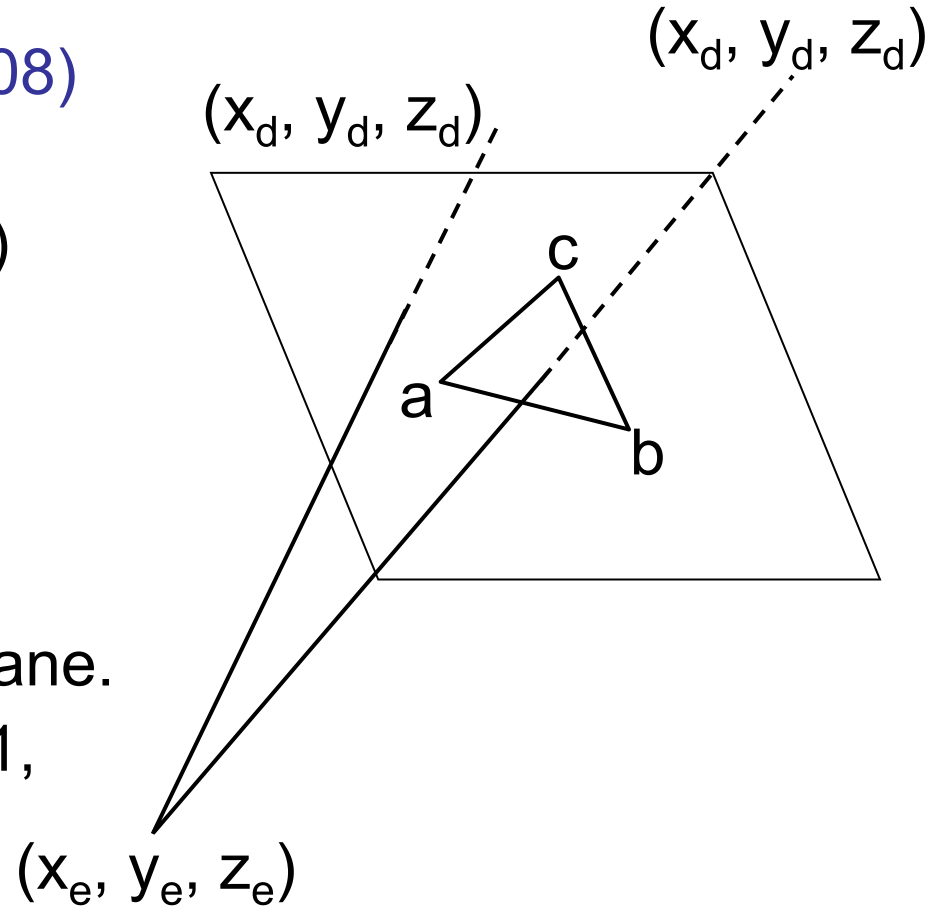
$\mathbf{e} + t\mathbf{d} = \mathbf{a} + \beta(\mathbf{b}-\mathbf{a}) + \gamma(\mathbf{c}-\mathbf{a})$

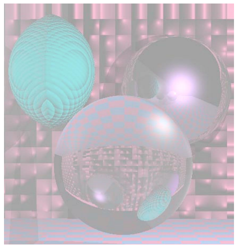
for t, γ, and β.

The x, y, and z components give you 3 linear equations in 3 unknowns.

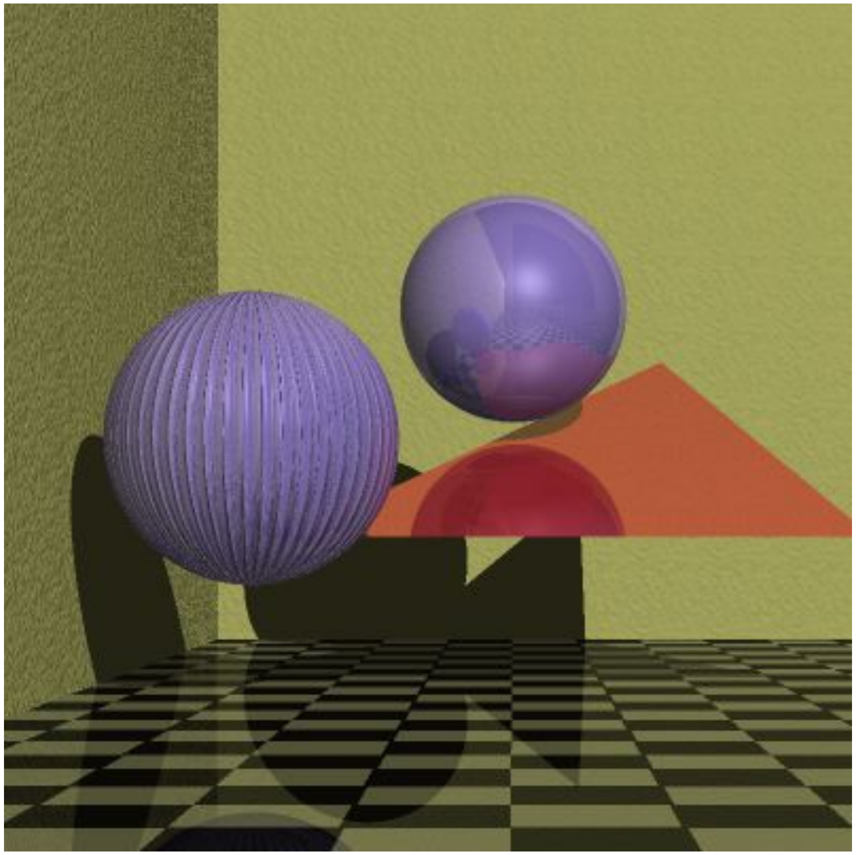
If $0 \leq t \leq 1$, the Ray hits the Plane.

If $0 \leq \beta$ and $0 \leq \gamma$ and $\beta + \gamma \leq 1$, the Ray hits the Triangle.

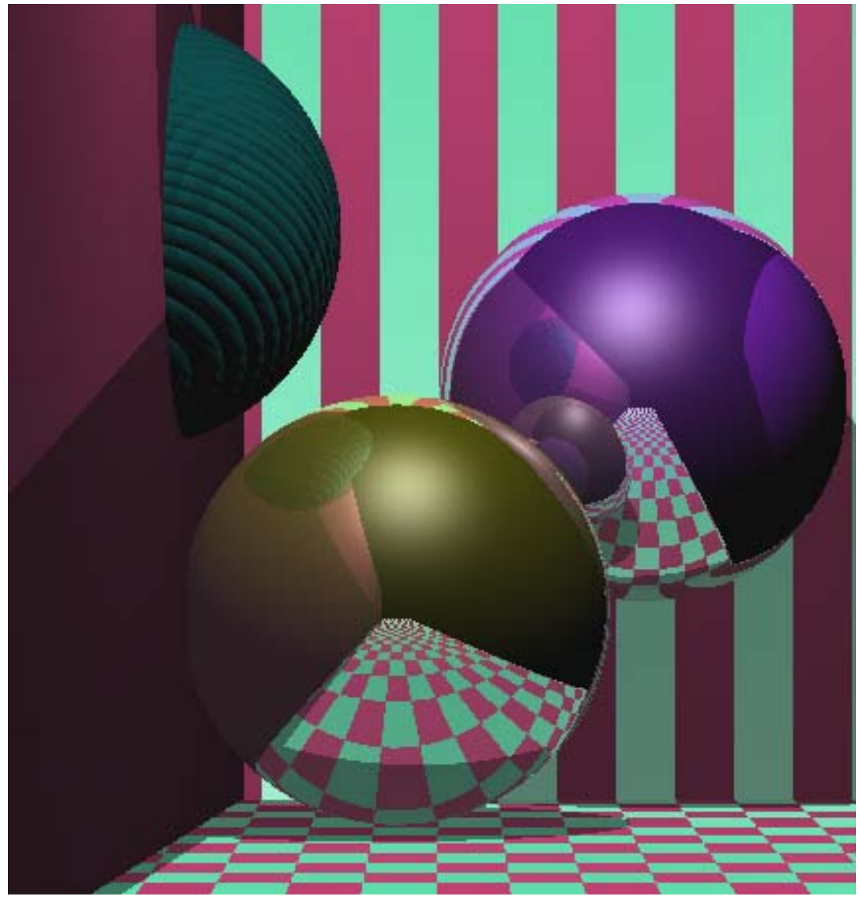




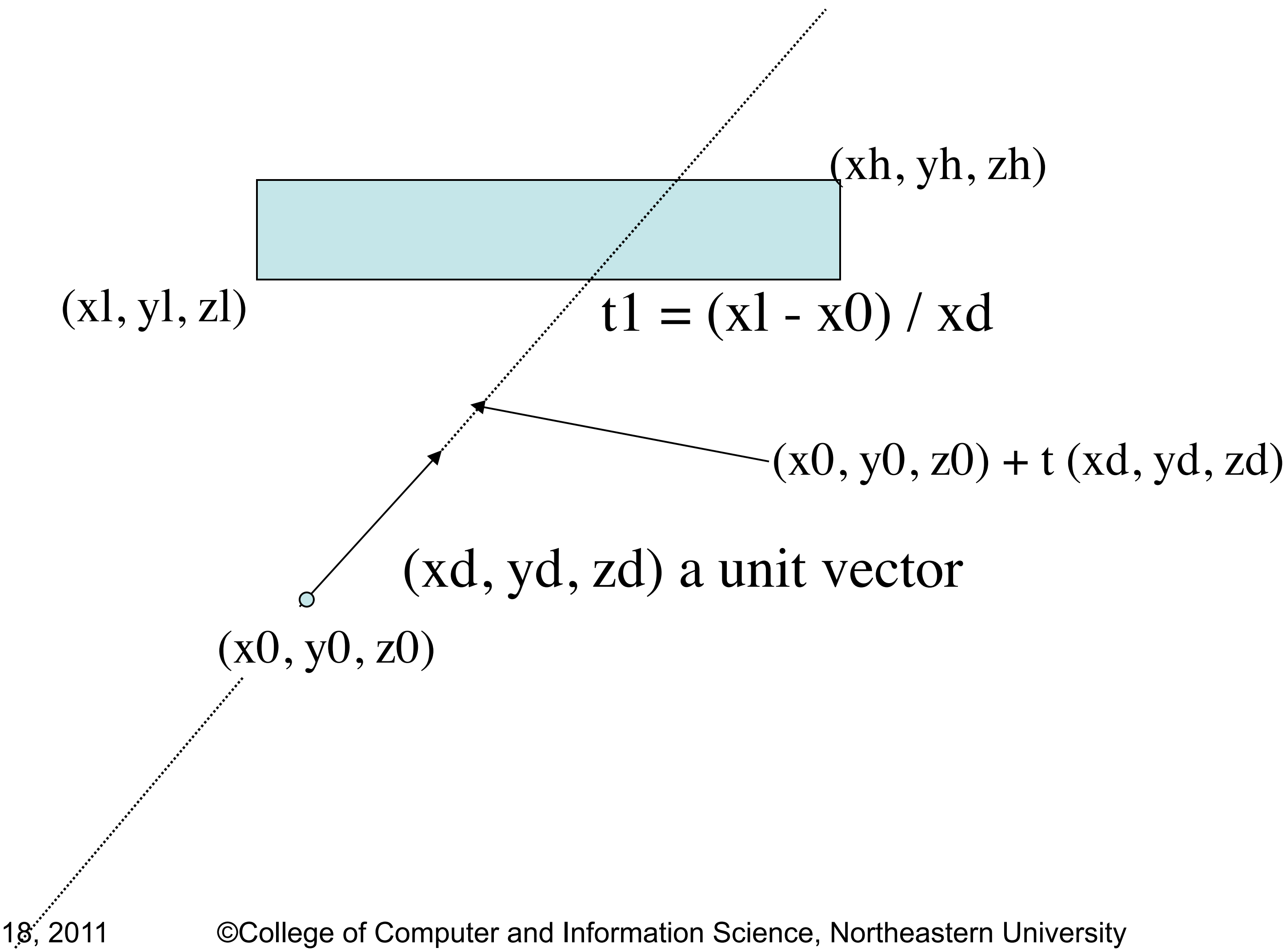
Images with
Planes and Polygons

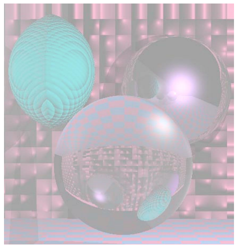


Images with Planes and Polygons



Ray Box Intersection





Ray Box Intersection

http://courses.csusm.edu/cs697exz/ray_box.htm

or see Watt pages 21-22

Box: minimum extent Bl = (xl, yl, zl) maximum extent Bh = (xh, yh, zh)

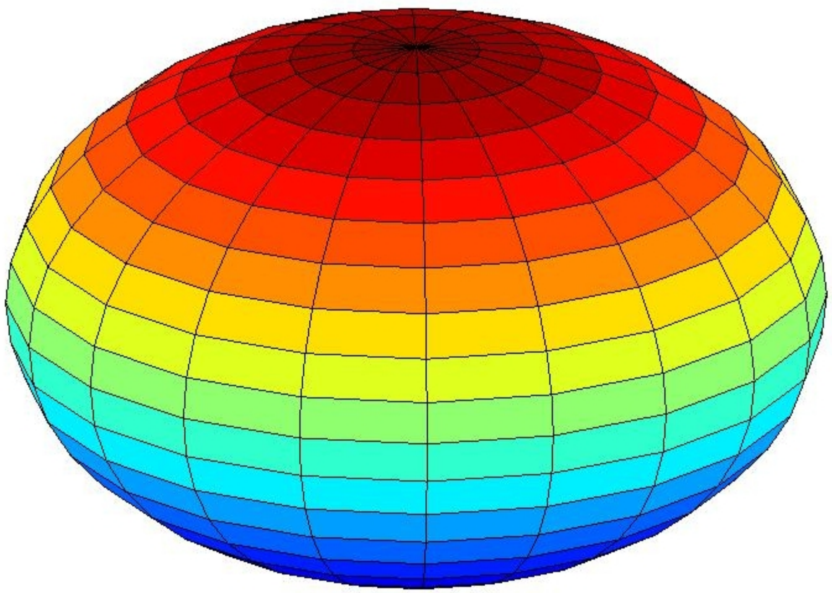
Ray: R0 = (x0, y0, z0) , Rd= (xd, yd, zd) ray is R0 + tRd

Algorithm:

1. Set tnear = -INFINITY , tfar = +INFINITY
2. For the pair of X planes
    1. if zd = 0, the ray is parallel to the planes so:
        - if x0 < x1 or x0 > xh return FALSE (origin not between planes)
    2. else the ray is not parallel to the planes, so calculate intersection distances of planes
        - t1 = (xl - x0) / xd (time at which ray intersects minimum X plane)
        - t2 = (xh - x0) / xd (time at which ray intersects maximum X plane)
        - if t1 > t2 , swap t1 and t2
        - if t1 > tnear , set tnear = t1
        - if t2 < tfar, set tfar= t2
        - if tnear > tfar, box is missed so return FALSE
        - if tfar < 0 , box is behind ray so return FALSE
3. Repeat step 2 for Y, then Z
4. All tests were survived, so return TRUE

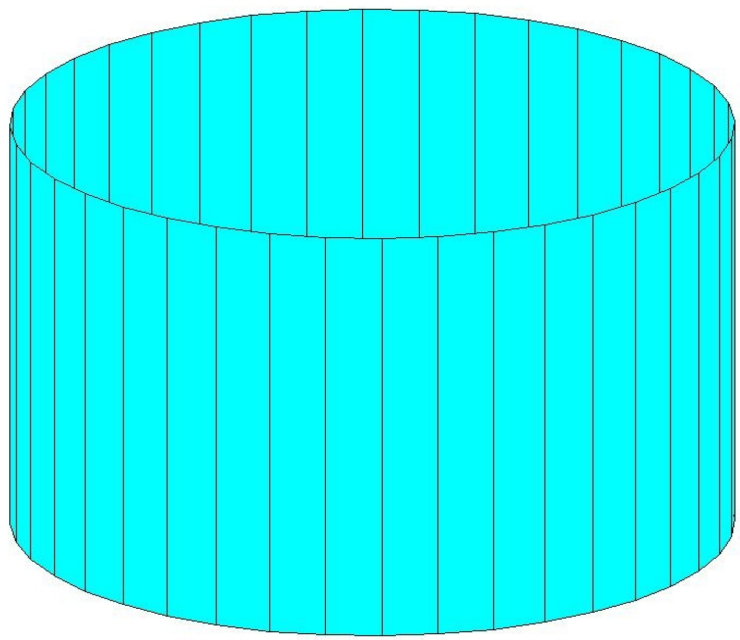
Quadric Surfaces

ellipsoid



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$$

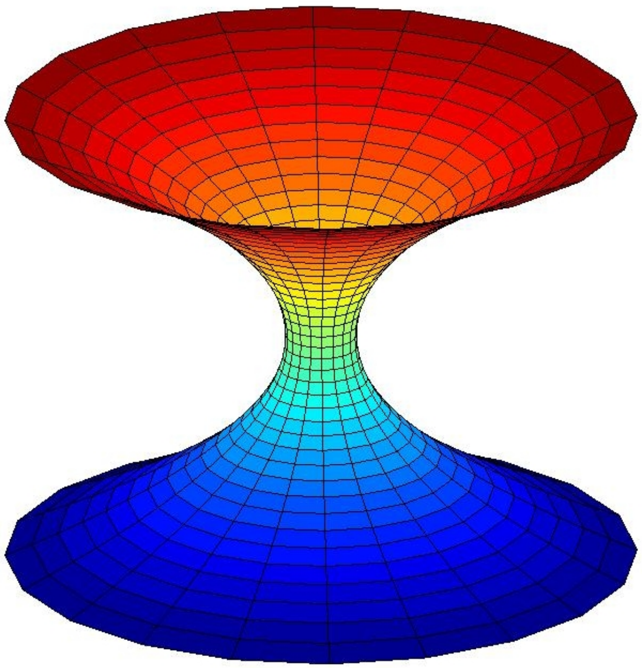
elliptic cylinder



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$$

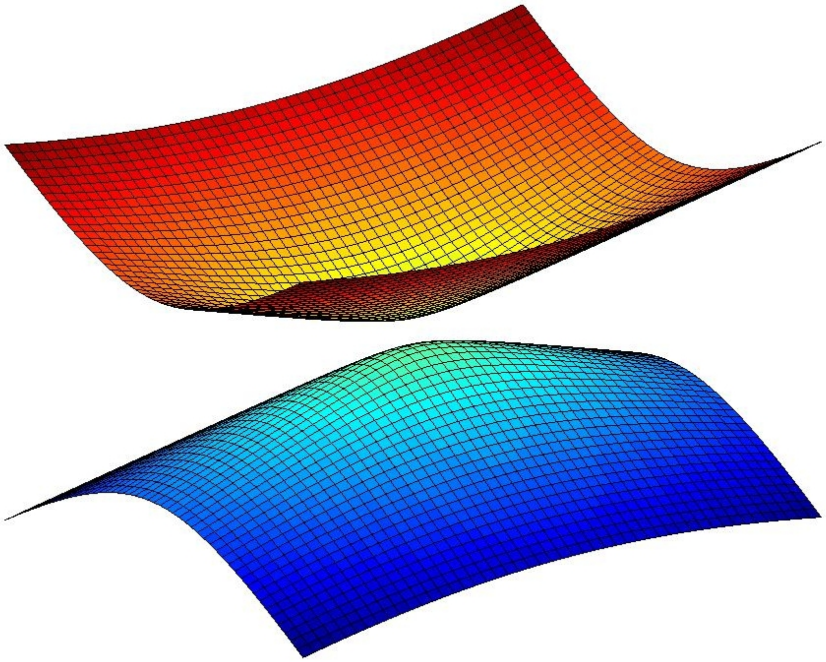
Quadric Surfaces

1-sheet hyperboloid



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$$

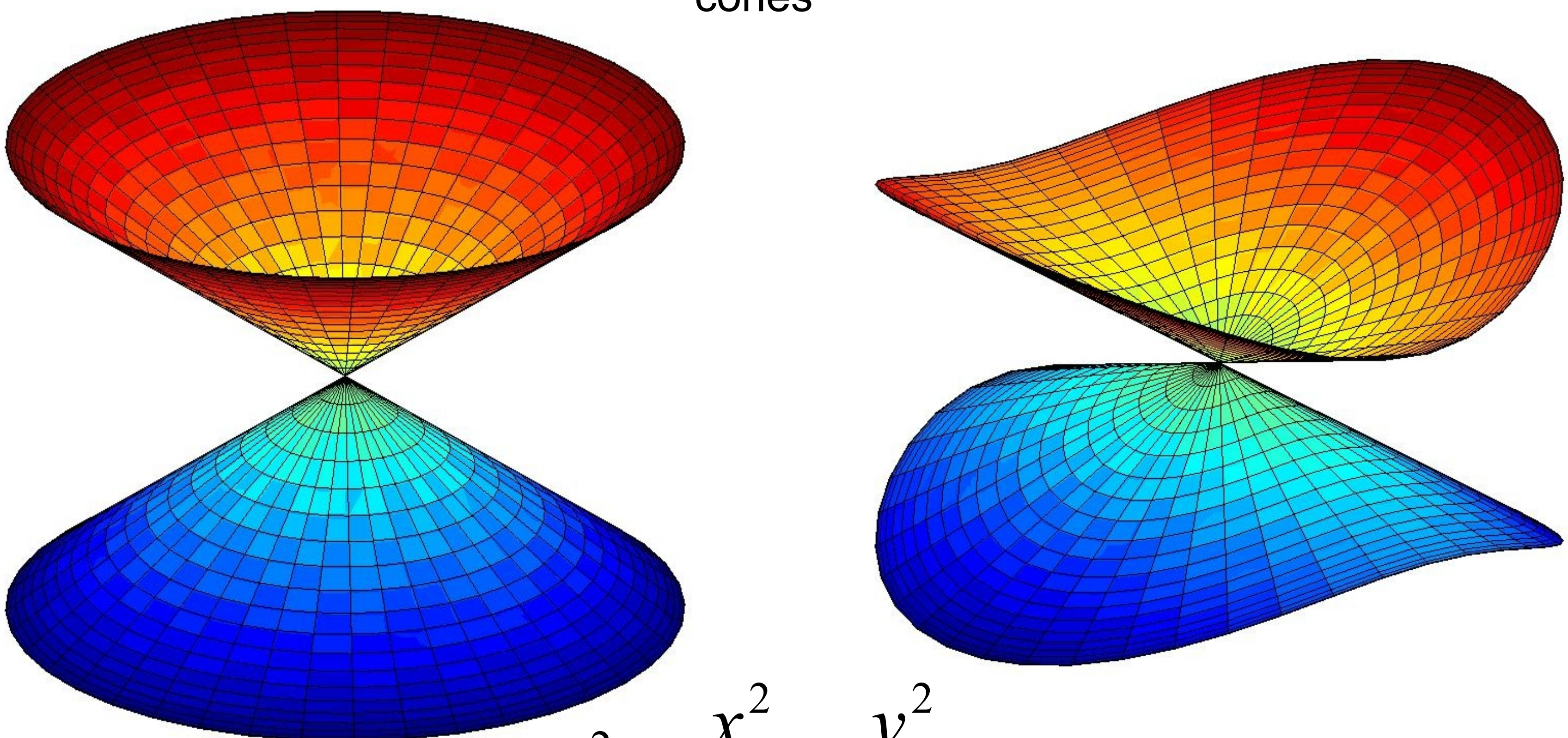
2-sheet hyperboloid



$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=-1$$

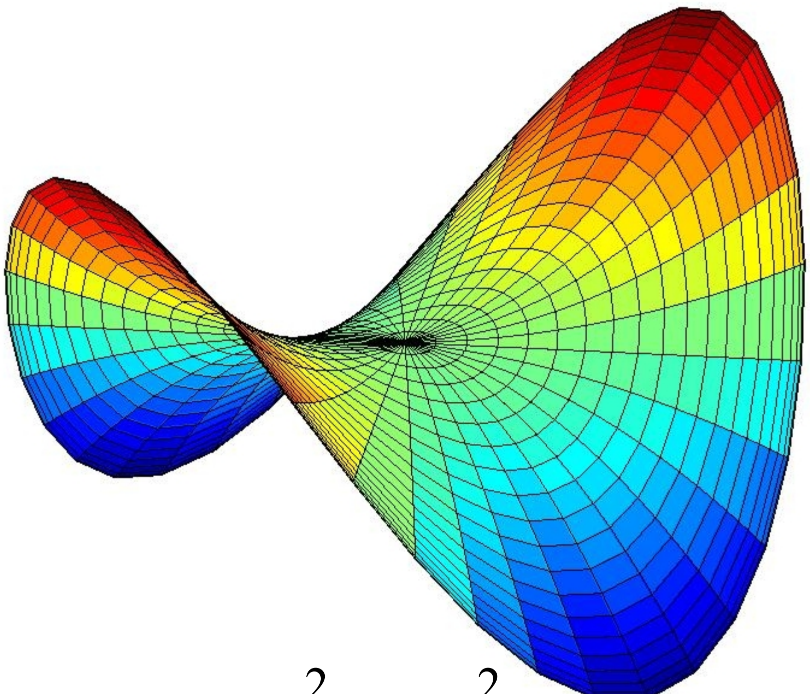
Quadric Surfaces

cones



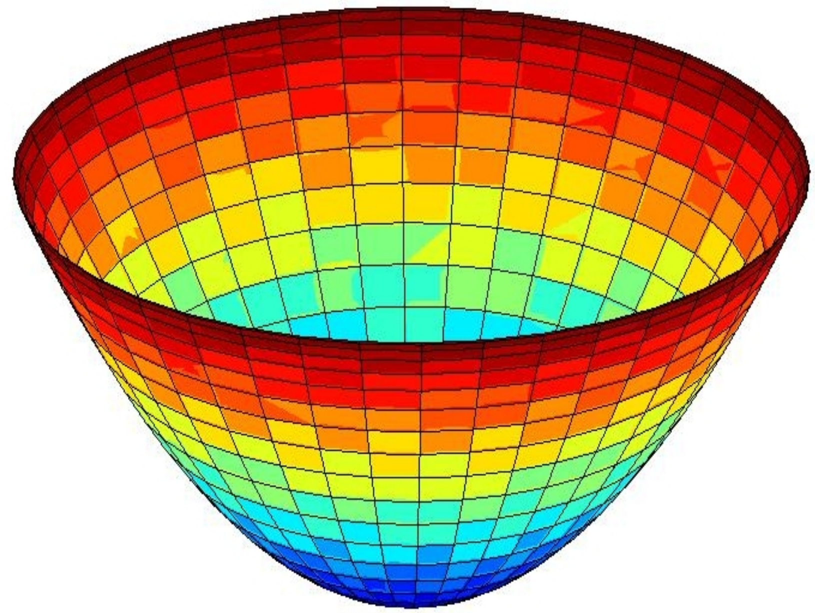
$$z^2 = \frac{x^2}{a^2} + \frac{y^2}{b^2}$$

hyperbolic parabaloid



$$z = \frac{y^2}{b^2} - \frac{x^2}{a^2}$$

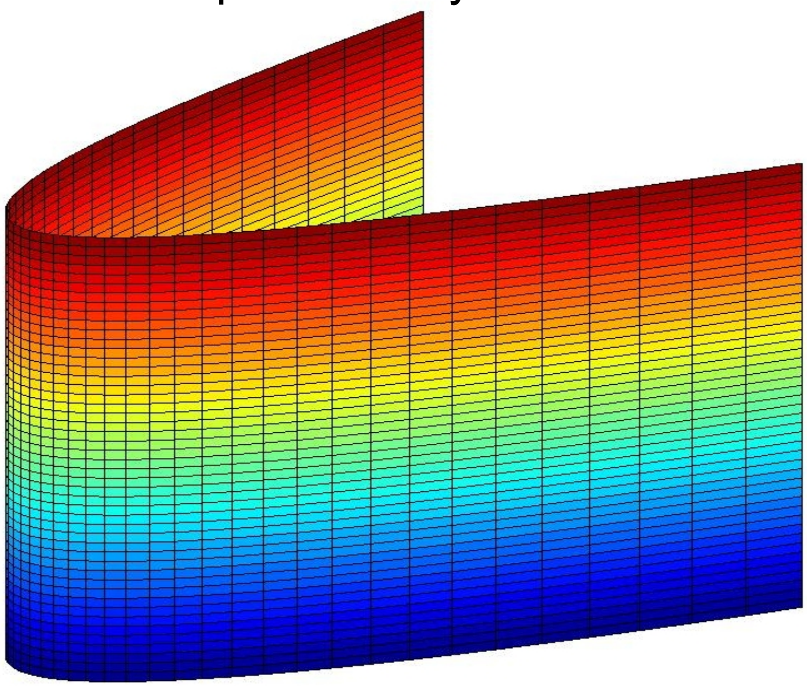
elliptic parabaloid



$$z = \frac{x^2}{a^2} + \frac{y^2}{b^2}$$

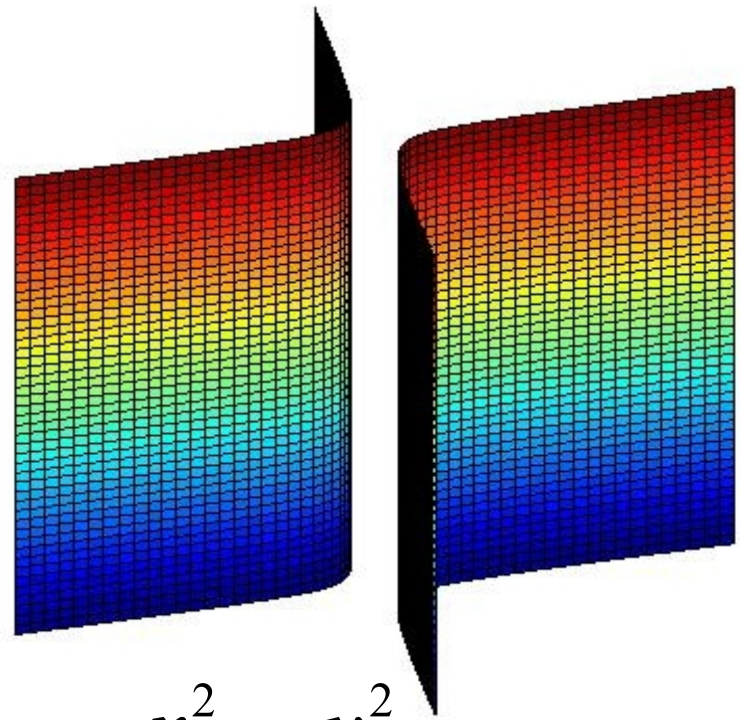
Quadric Surfaces

parabolic cylinder

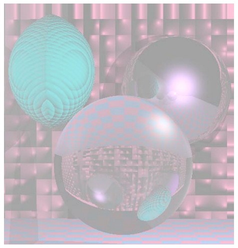


$$x^2 + 2y = 0$$

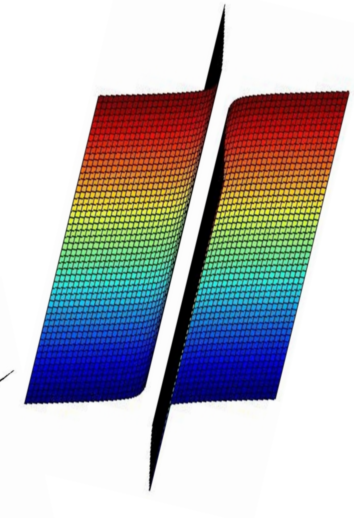
hyperbolic cylinder

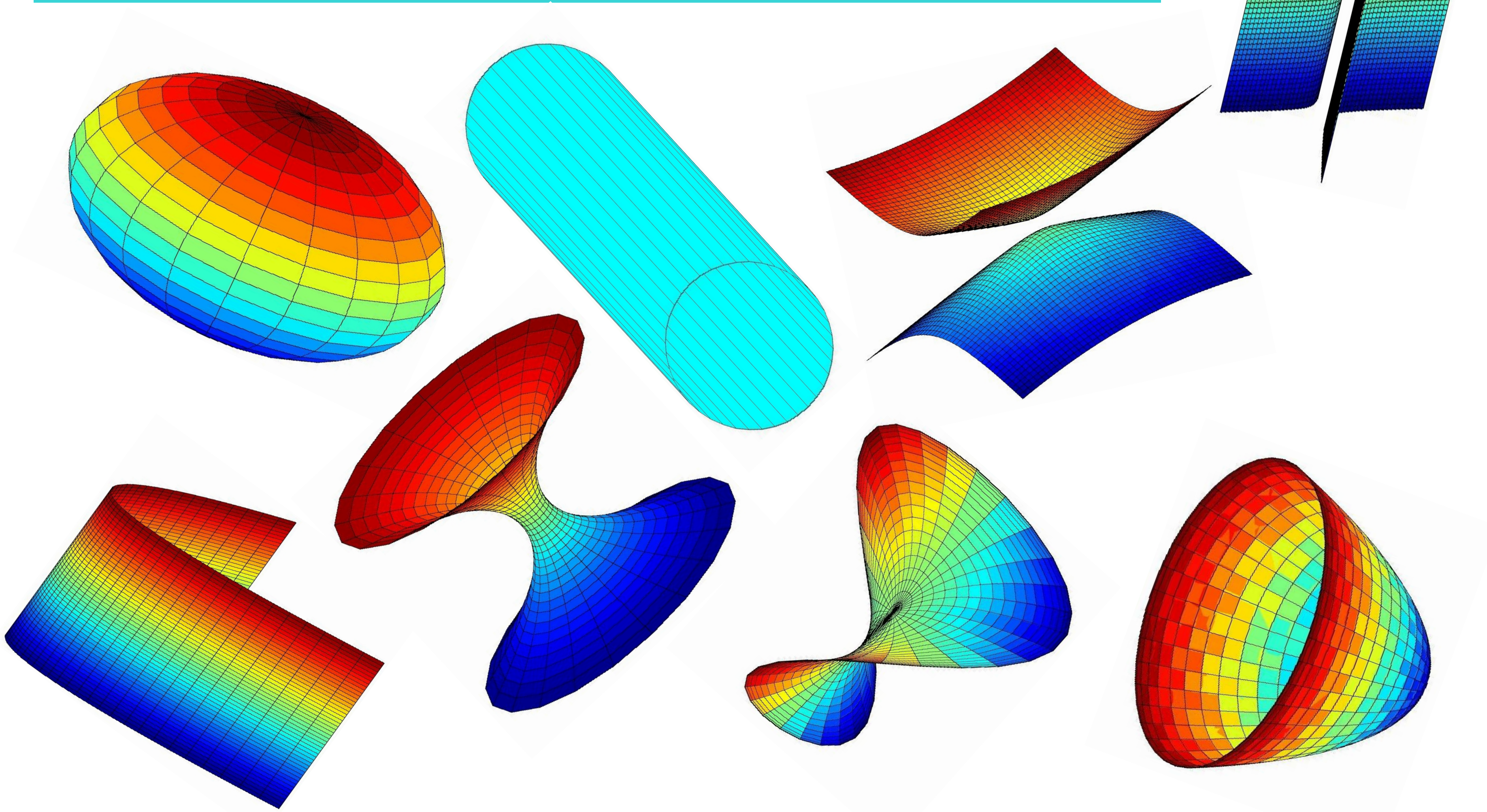


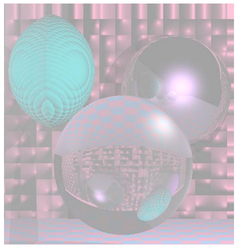
$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$



General Quadrics in General Position







General Quadric Equation

$$ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k = 0$$

Ray Equations

$$x(t) = x_0 + tx_d \qquad x_d = x_1 - x_0$$

$$y(t) = y_0 + ty_d \qquad y_d = y_1 - y_0$$

$$z(t) = z_0 + tz_d \qquad z_d = z_1 - z_0$$

$$a(x_0 + tx_d)^2 + b(y_0 + ty_d)^2 + c(z_0 + tz_d)^2$$
$$+ 2d(x_0 + tx_d)(y_0 + ty_d) + 2e(y_0 + ty_d)(z_0 + tz_d) + 2f(x_0 + tx_d)(z_0 + tz_d)$$
$$+ 2g(x_0 + tx_d) + 2h(y_0 + ty_d) + 2j(z_0 + tz_d) + k = 0$$

$$A = ax_d^2 + by_d^2 + cz_d^2 + 2dx_d y_d + 2ey_d z_d + 2fx_d z_d$$

$$B = 2ax_0 x_d + 2by_0 y_d + 2cz_0 z_d$$
$$+ 2d(x_0 y_d + x_d y_0) + 2e(y_0 z_d + y_d z_0) + 2f(x_0 z_d + x_d z_0) + 2gx_d + 2hy_d + 2jz_d$$

$$C = ax_0^2 + by_0^2 + cz_0^2 + 2dx_0 y_0 + 2ey_0 z_0 + 2fx_0 z_0 + 2gx_0 + 2hy_0 + 2jz_0 + k$$

Ray Quadric Intersection Quadratic Coefficients

A = a*xd*xd + b*yd*yd + c*zd*zd

+ 2[d*xd*yd + e*yd*zd + f*xd*zd

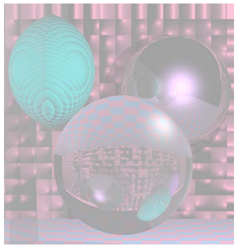
B = 2*[a*x0*xd + b*y0*yd + c*z0*zd

+ d*(x0*yd + xd*y0) + e*(y0*zd + yd*z0) + f*(x0*zd + xd*z0)

+ g*xd + h*yd + j*zd]

C = a*x0*x0 + b*y0*y0 + c*z0*z0

+ 2*[d*x0*y0 + e*y0*z0 + f*x0*z0 + g*x0 + h*y0 + j*z0] + k



Quadric Normals

$$Q(x,y,z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k$$

$$\frac{\partial Q}{\partial x} = 2ax + 2dy + 2fz + 2g = 2(ax + dy + fz + g)$$

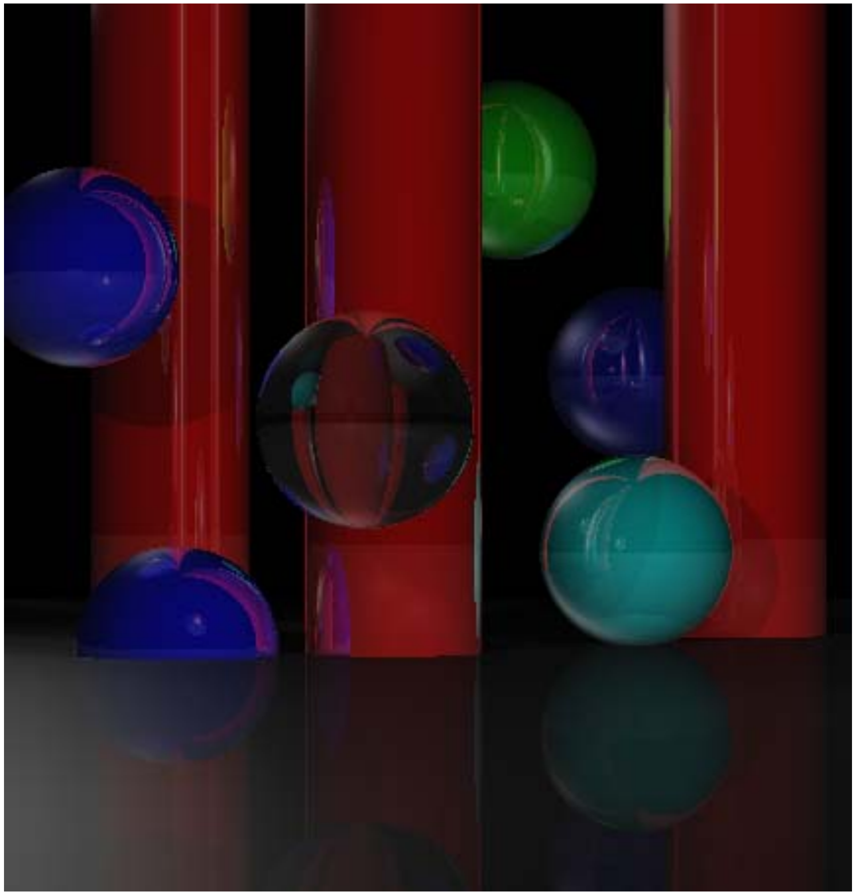
$$\frac{\partial Q}{\partial y} = 2by + 2dx + 2ez + 2h = 2(by + dx + ez + h)$$

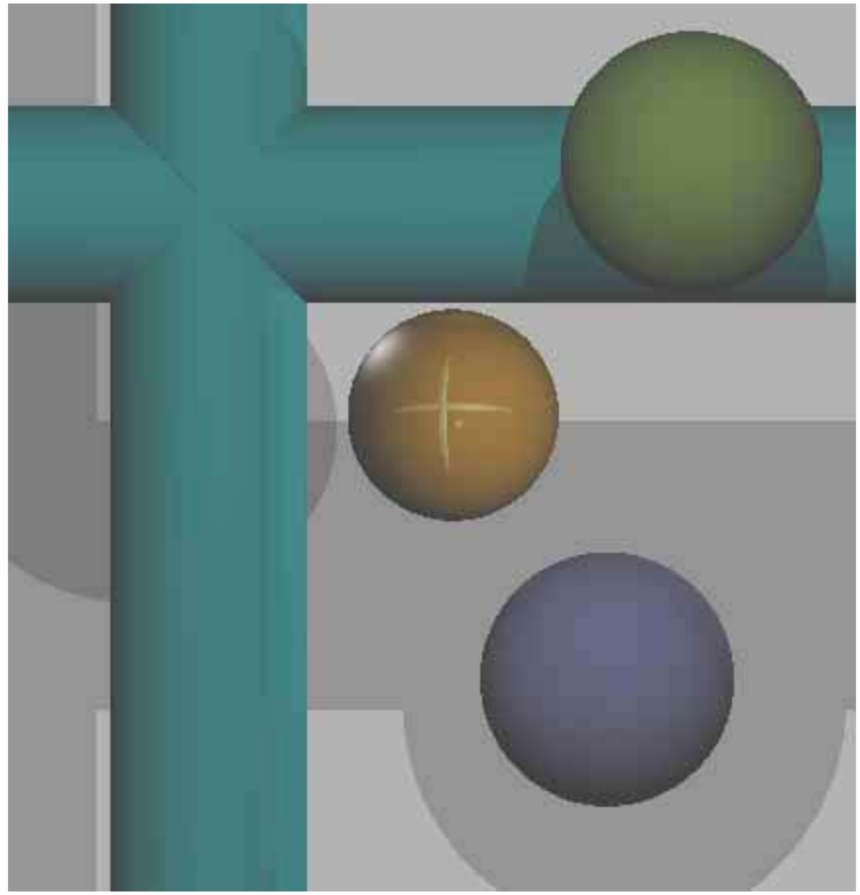
$$\frac{\partial Q}{\partial z} = 2cz + 2ey + 2fx + 2j = 2(cz + ey + fx + j)$$

$$N = \left(\frac{\partial Q}{\partial x}, \frac{\partial Q}{\partial y}, \frac{\partial Q}{\partial z} \right)$$

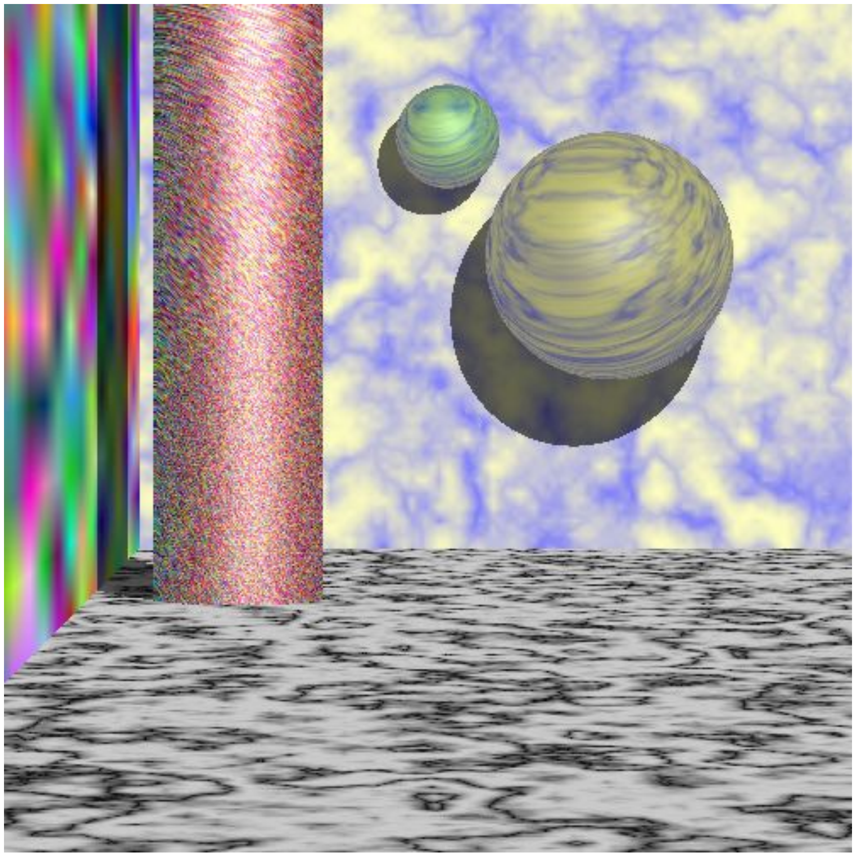
Normalize *N* and change its sign if necessary.

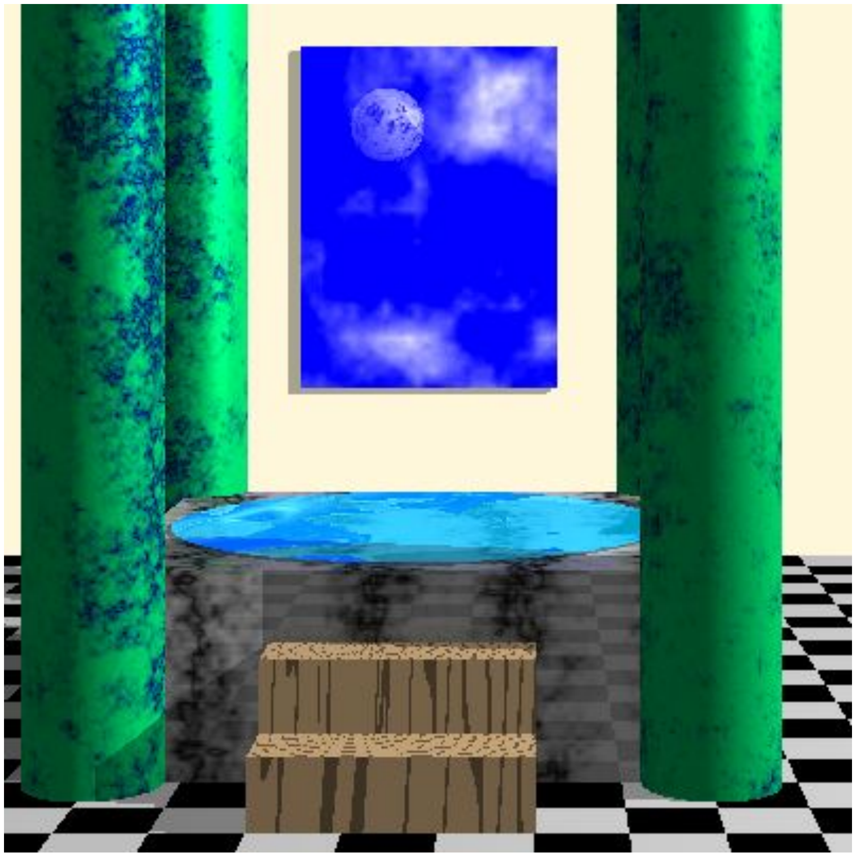
MyCylinders



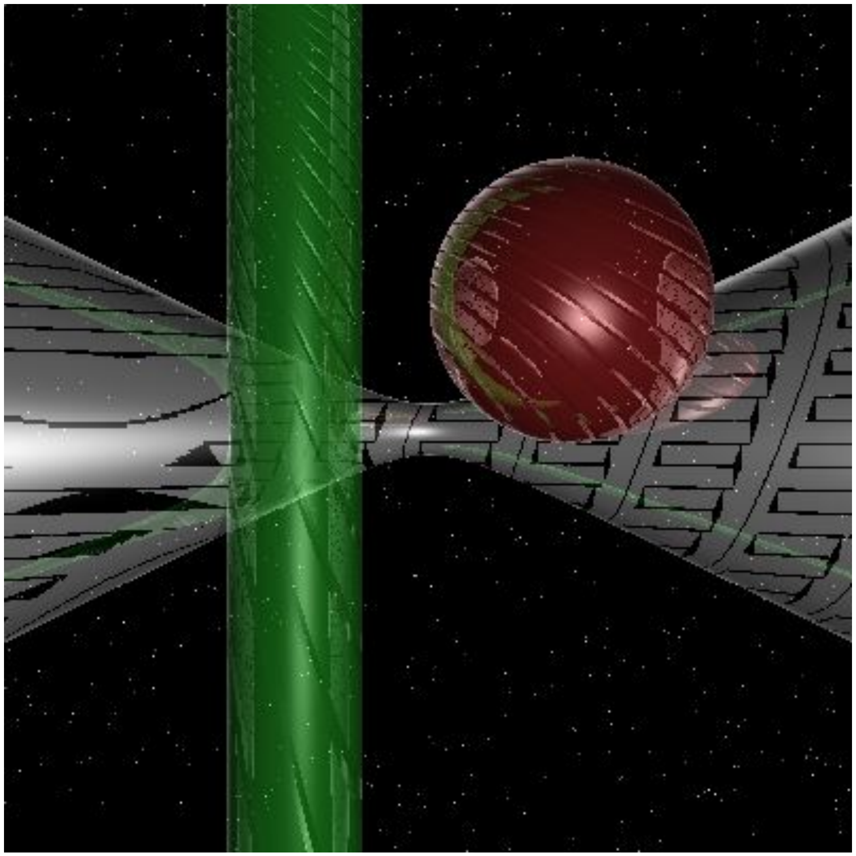


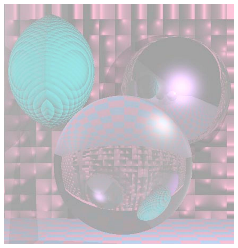
Student Images





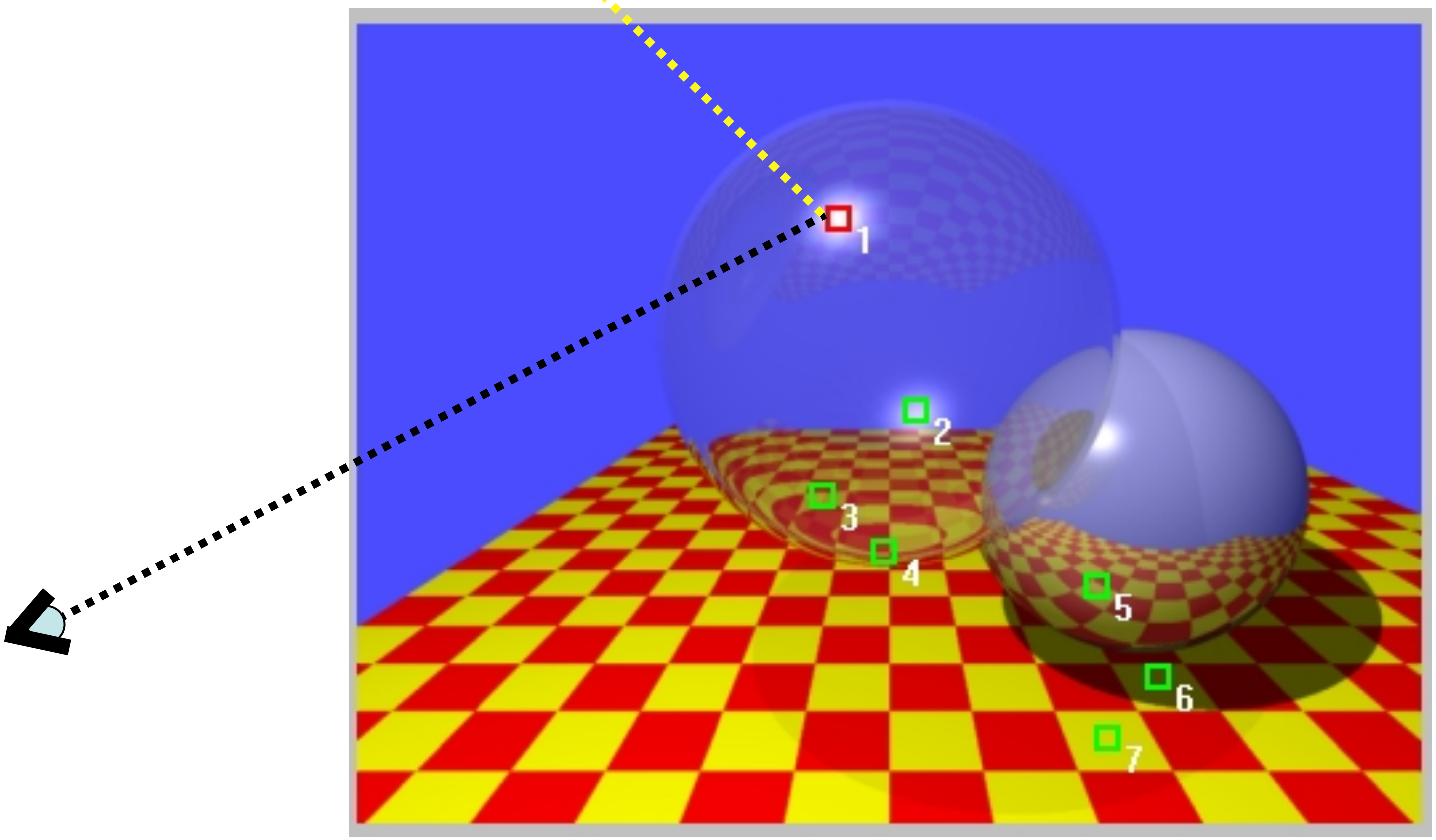
Student Images





Recursive Ray Tracing

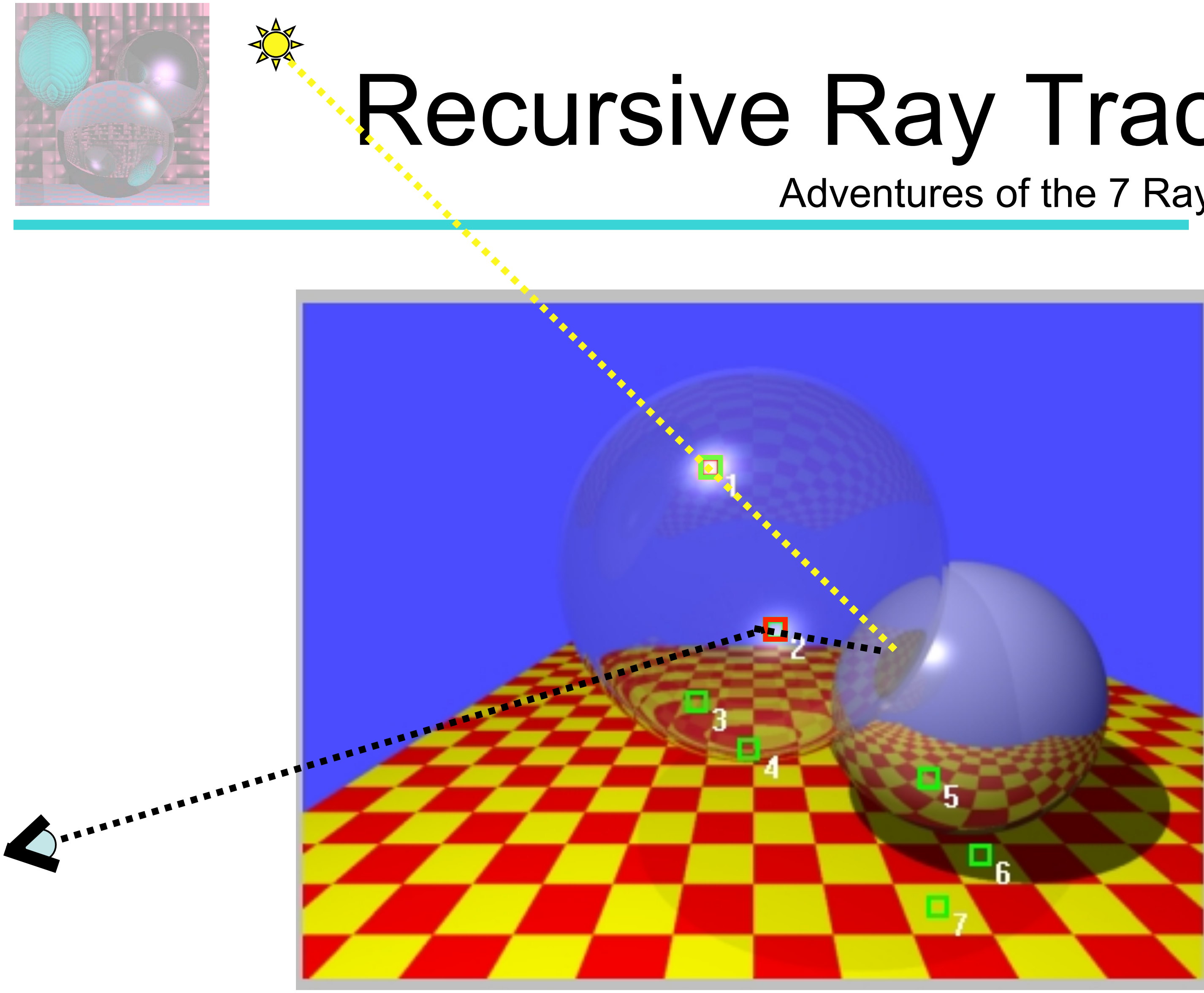
Adventures of the 7 Rays - Watt



Specular Highlight on Outside of Shere

Recursive Ray Tracing

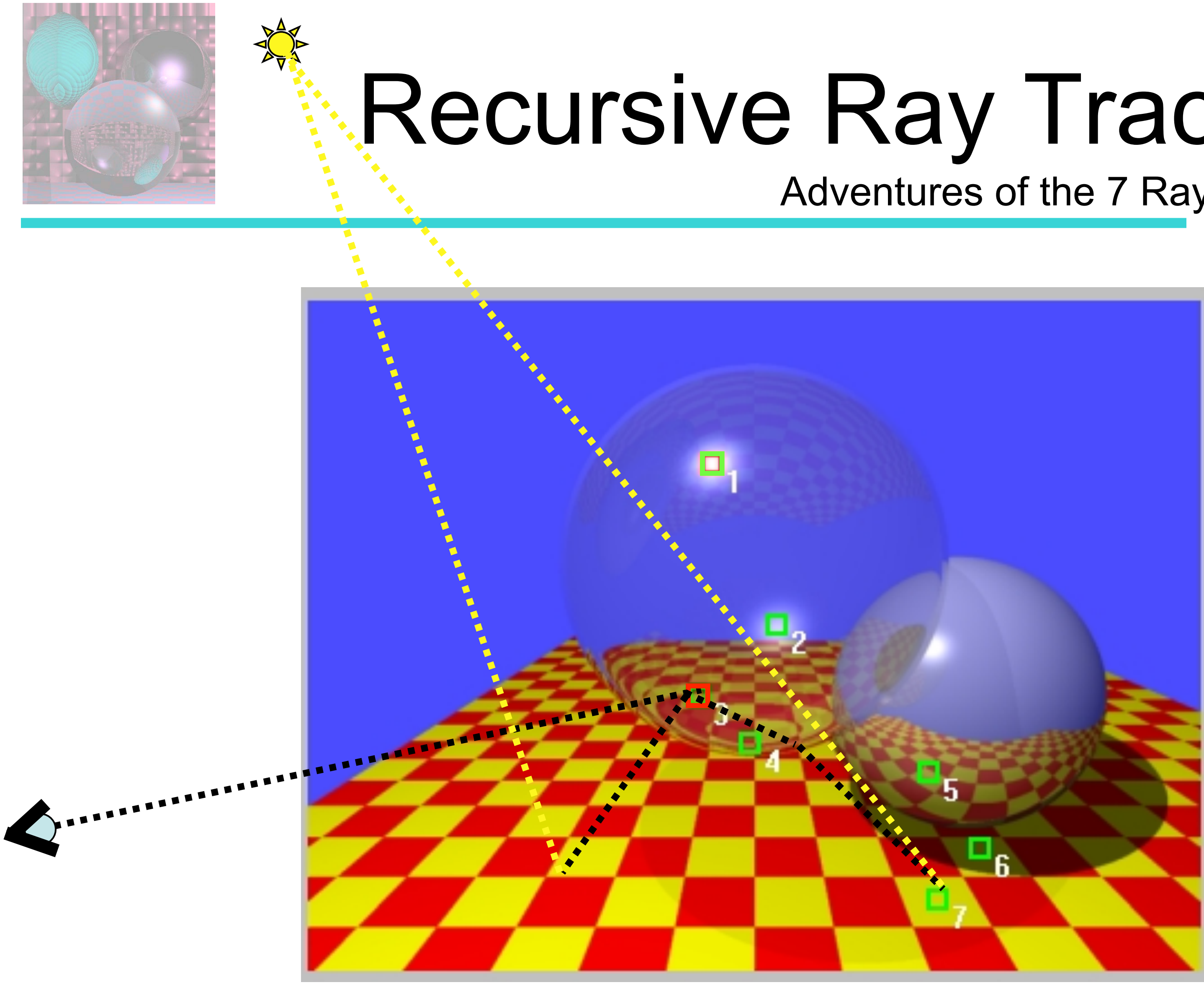
Adventures of the 7 Rays - Watt



Specular Highlight on Inside of Sphere

Recursive Ray Tracing

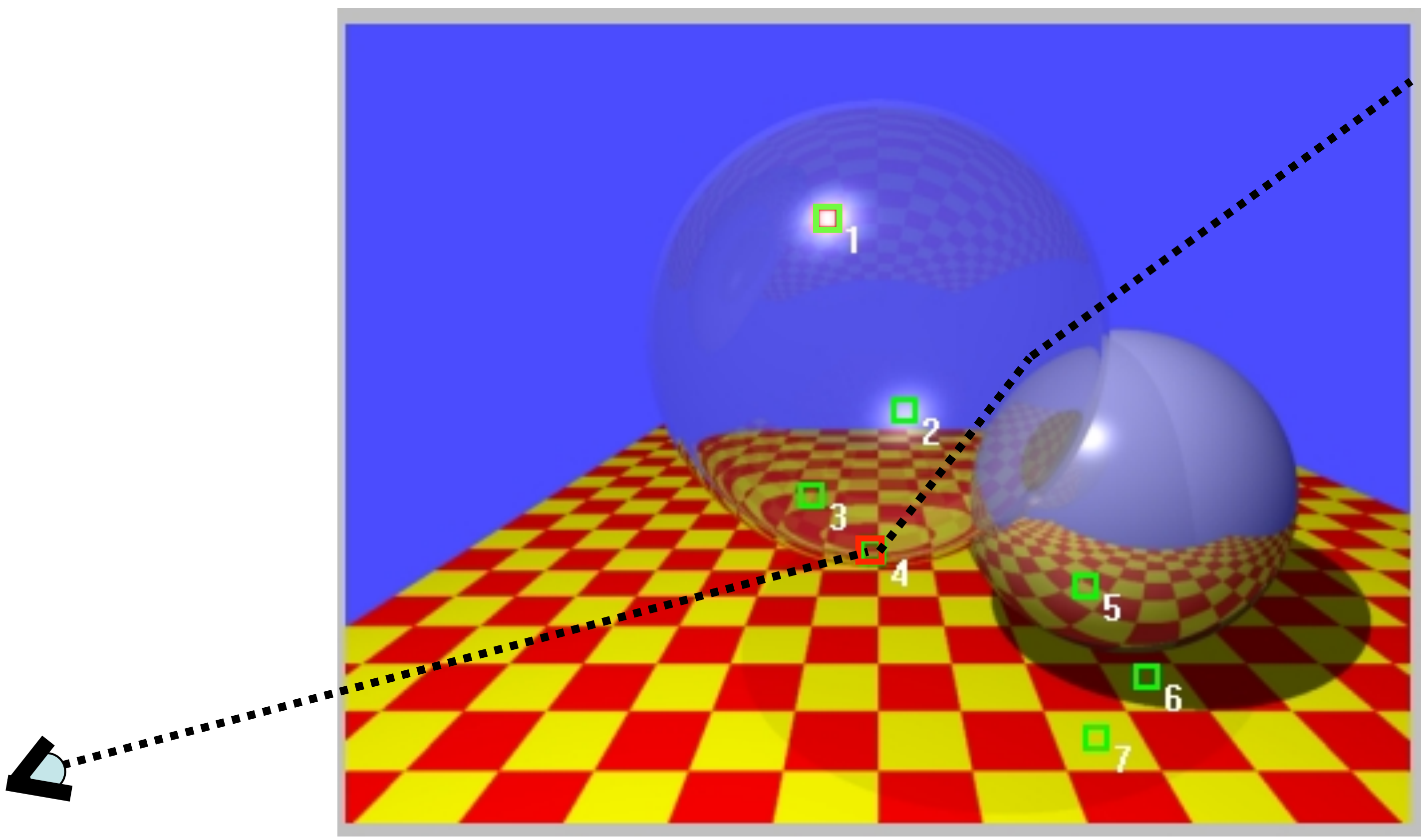
Adventures of the 7 Rays - Watt



Reflection and Refraction of Checkerboard

Recursive Ray Tracing

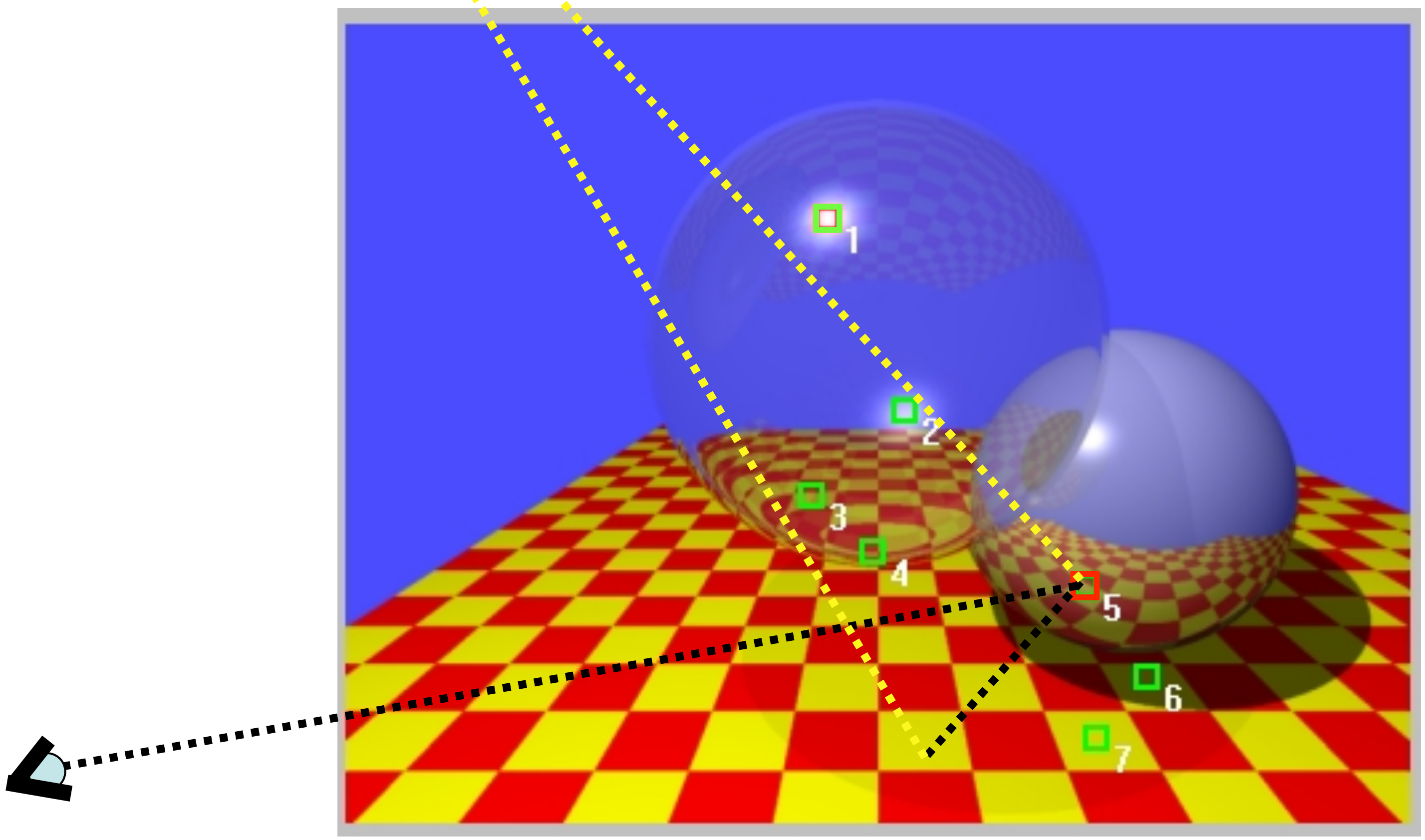
Adventures of the 7 Rays - Watt



Refraction Hitting Background

Recursive Ray Tracing

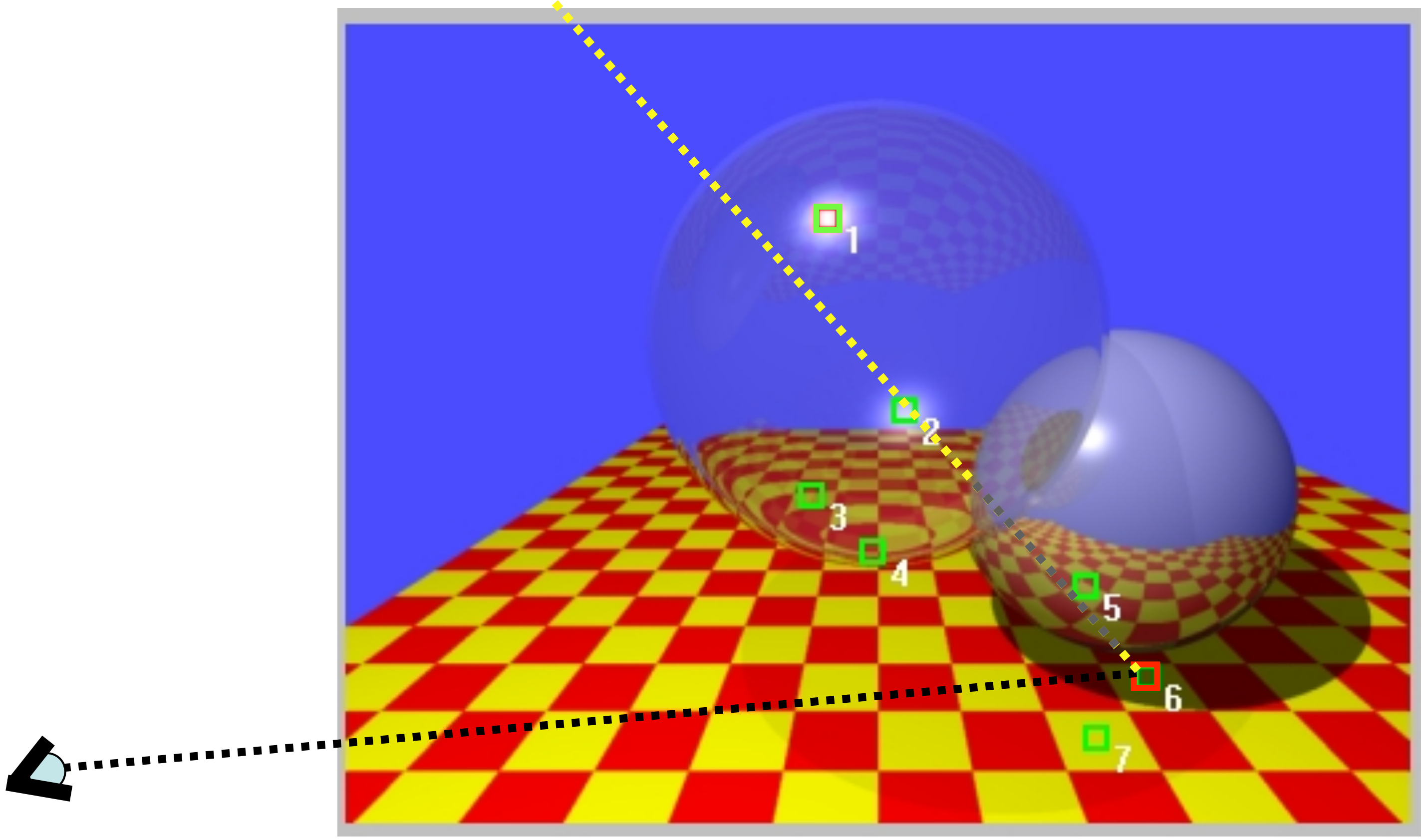
Adventures of the 7 Rays - Watt



Local Diffuse Plus Reflection from Checkerboard

Recursive Ray Tracing

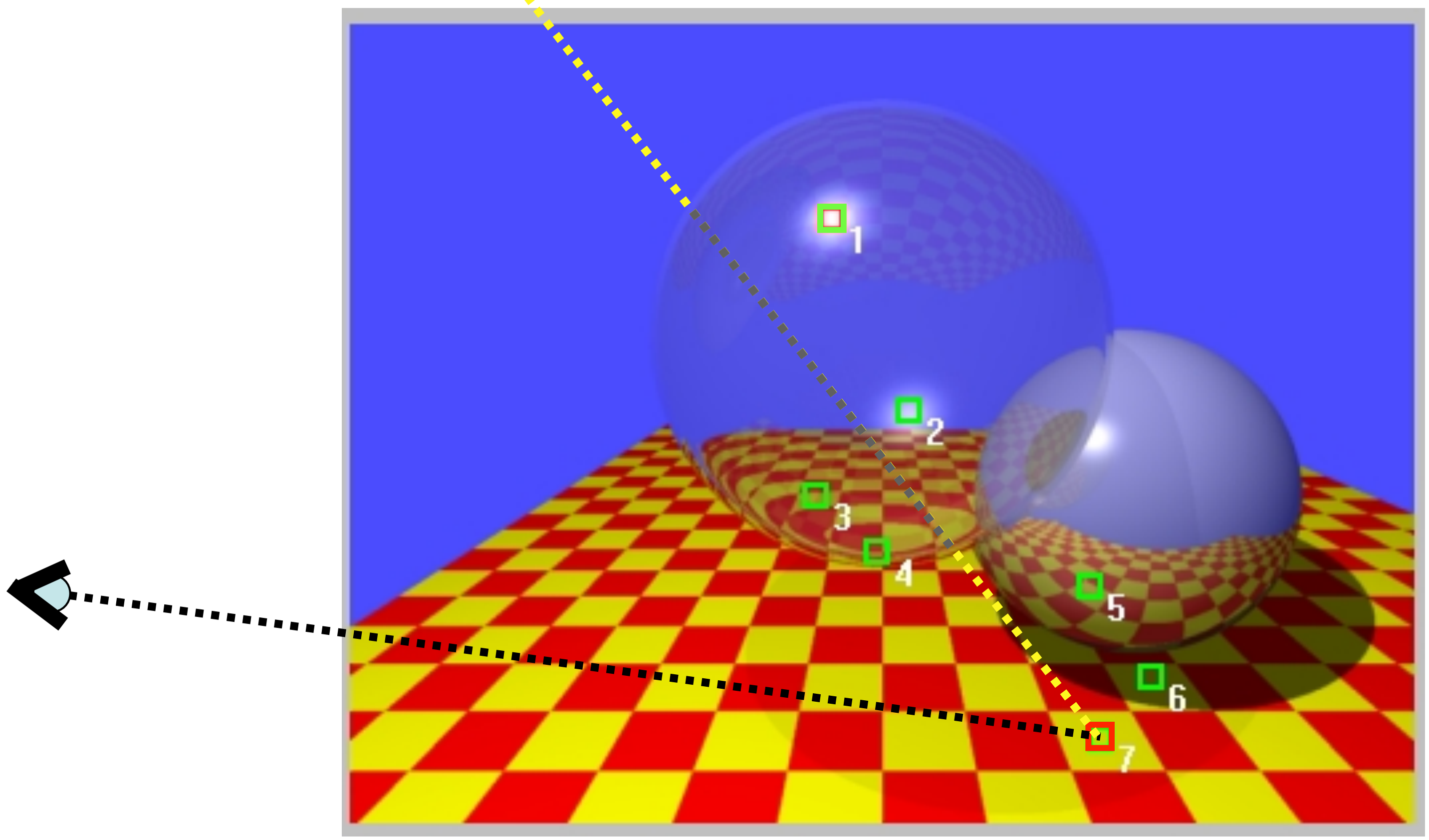
Adventures of the 7 Rays - Watt



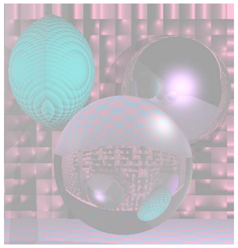
Local Diffuse in Complete Shadow

Recursive Ray Tracing

Adventures of the 7 Rays - Watt



Local Diffuse in Shadow from Transparent Sphere



Recursive Ray-Tracing

- How do we know which rays to follow?
- How do we compute those rays?
- How do we organize code so we can follow all those different rays?

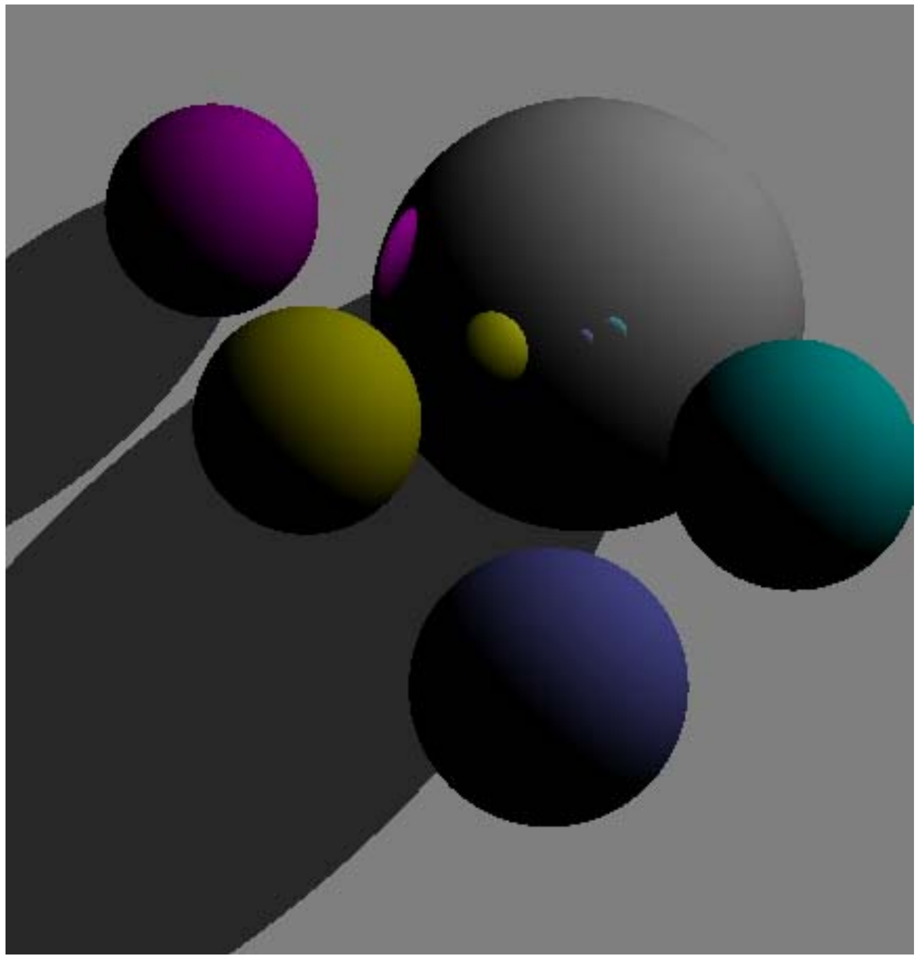
```
select center of projection(cp) and window on view plane;
for (each scan line in the image ) {
  for (each pixel in scan line ) {
    determine  ray from the cp through the pixel;
    pixel = RT_trace(ray, 1);}}

// intersect ray with objects; compute shade at closest intersection
// depth is current depth in ray tree
RT_color RT_trace (RT_ray ray; int depth){
  determine closest intersection of ray with an object;
  if (object hit) {
    compute normal at intersection;
    return RT_shade (closest object hit, ray, intersection, normal,
                               depth);}
  else
    return BACKGROUND_VALUE;
}
```

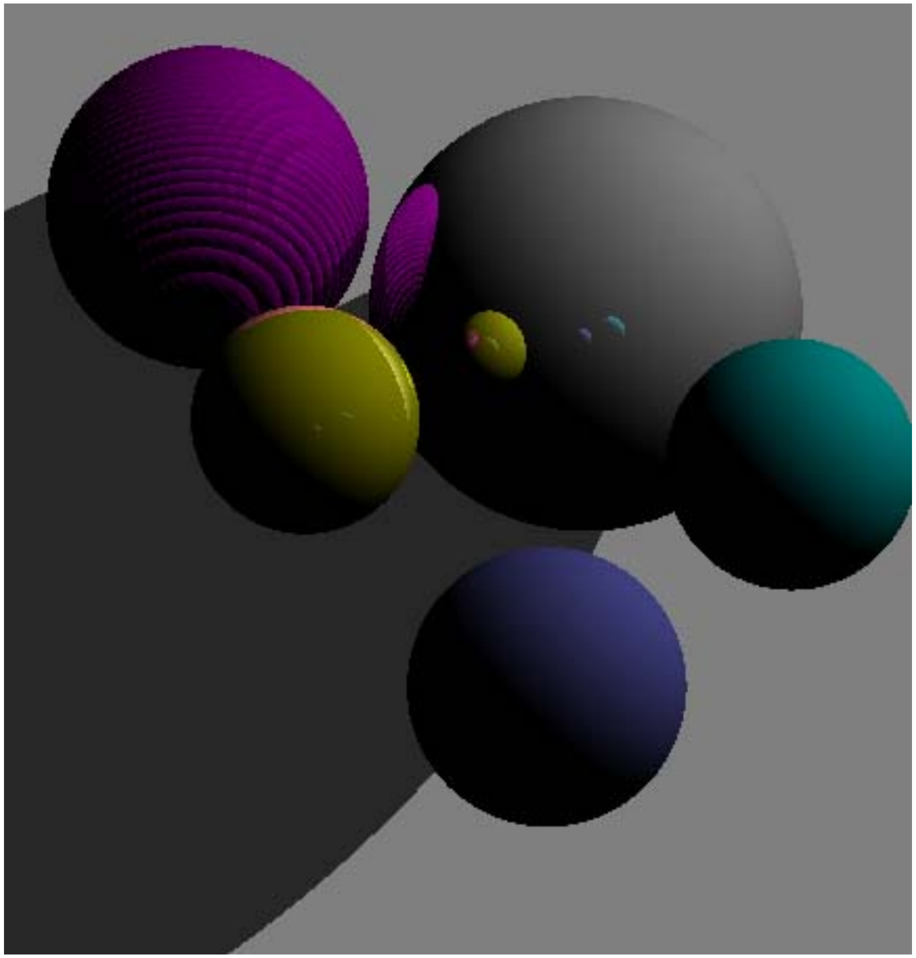
```
// Compute shade at point on object,
// tracing rays for shadows, reflection, refraction.
RT_color RT_shade (
   RT_object object,   // Object intersected
   RT_ray ray,         // Incident ray
   RT_point point,     // Point of intersection to shade
   RT_normal normal,// Normal at point
   int depth )         // Depth in ray tree
{
RT_color  color;       // Color of ray
RT_ray rRay, tRay, sRay;// Reflected, refracted, and shadow ray
   color = ambient term ;
   for ( each light  ) {
      sRay = ray from point to light ;
      if (  dot product of normal and direction to light is positive  ){
         compute how much light is blocked by opaque and
         transparent surfaces, and use to scale diffuse and specular
         terms before adding them to color;}}
```

```
    if ( depth < maxDepth ) {       // return if depth is too deep
        if ( object is reflective ) {
            rRay = ray in reflection direction from point;
            rColor = RT_trace(rRay, depth + 1);
            scale rColor by specular coefficient and add to color;
        }
        if ( object  is transparent  ) {
            tRay = ray in refraction direction from point;
            if ( total internal reflection does not occur  ) {
                tColor = RT_trace(tRay, depth + 1);
                scale tColor by transmission coefficient
                and add to color;
            }
        }
    }
    return color;   // Return the color of the ray
}
```

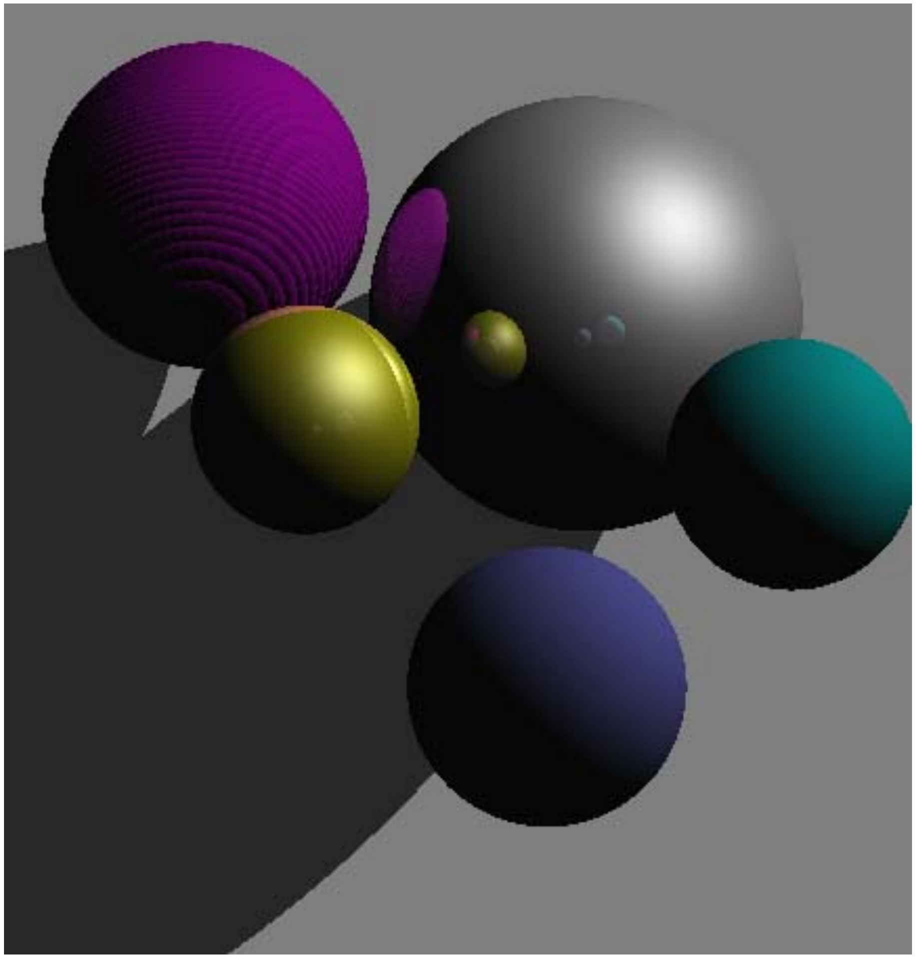
Reflections, no Highlight



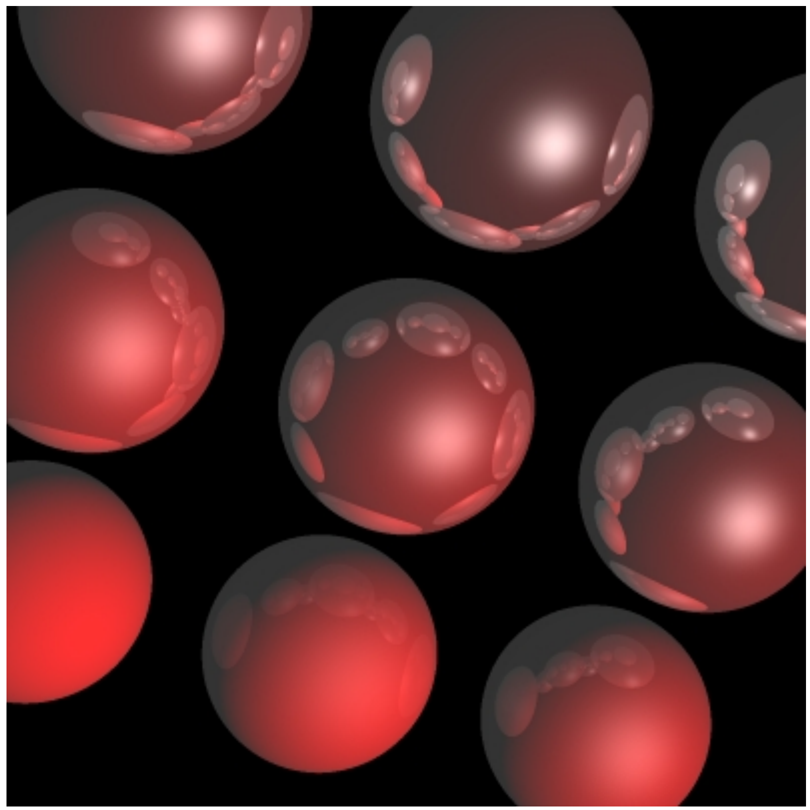
Second Order Reflection

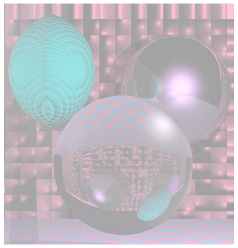


Refelction with Highlight

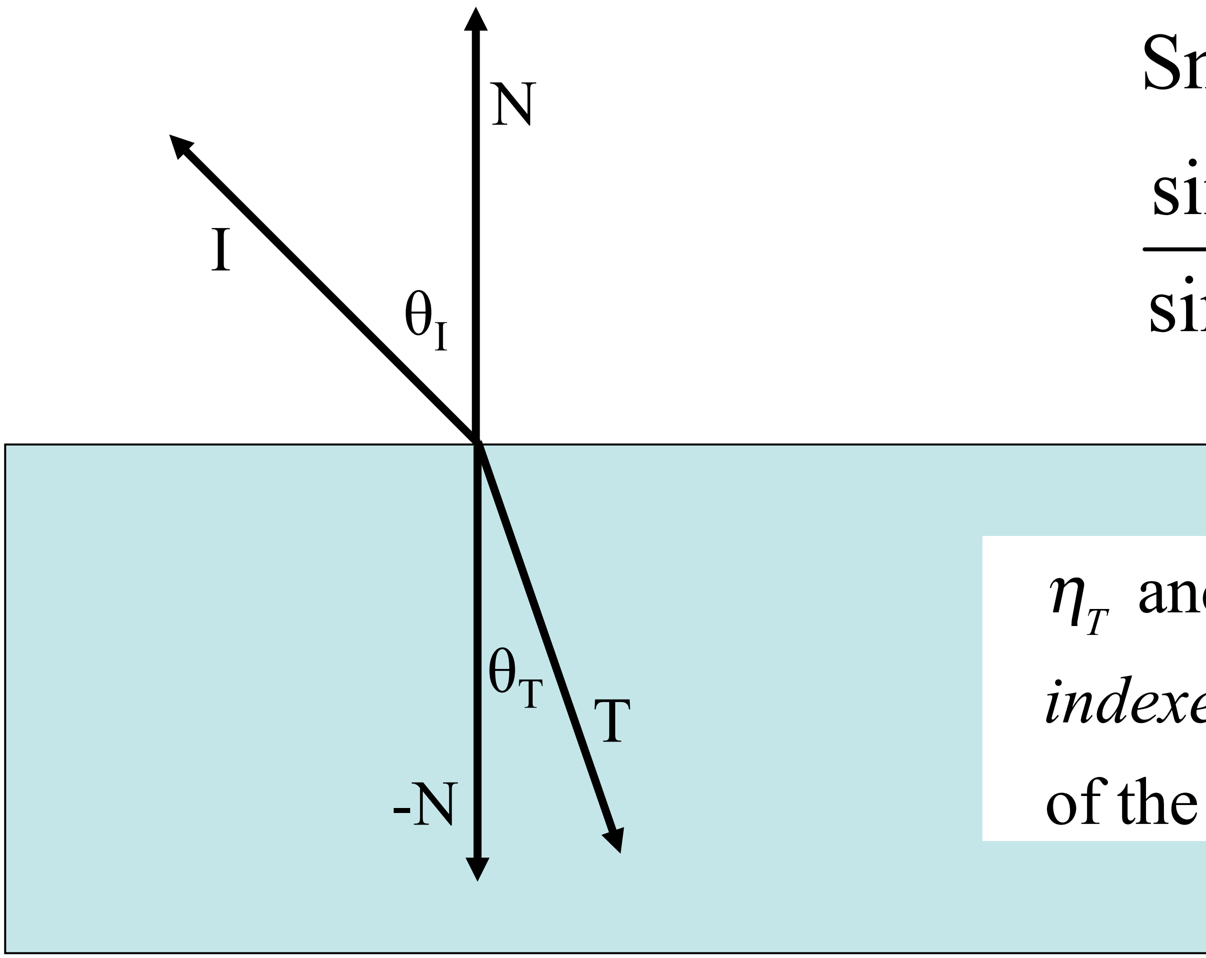


Nine Red Balls





Refraction



Snell's Law

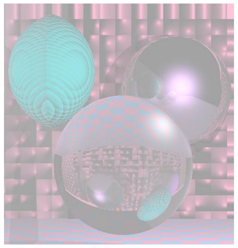
$$\frac{\sin(\theta_I)}{\sin(\theta_T)} = \frac{\eta_T}{\eta_I}$$

η_T and η_I are the *indexes of refraction* of the two mediums.

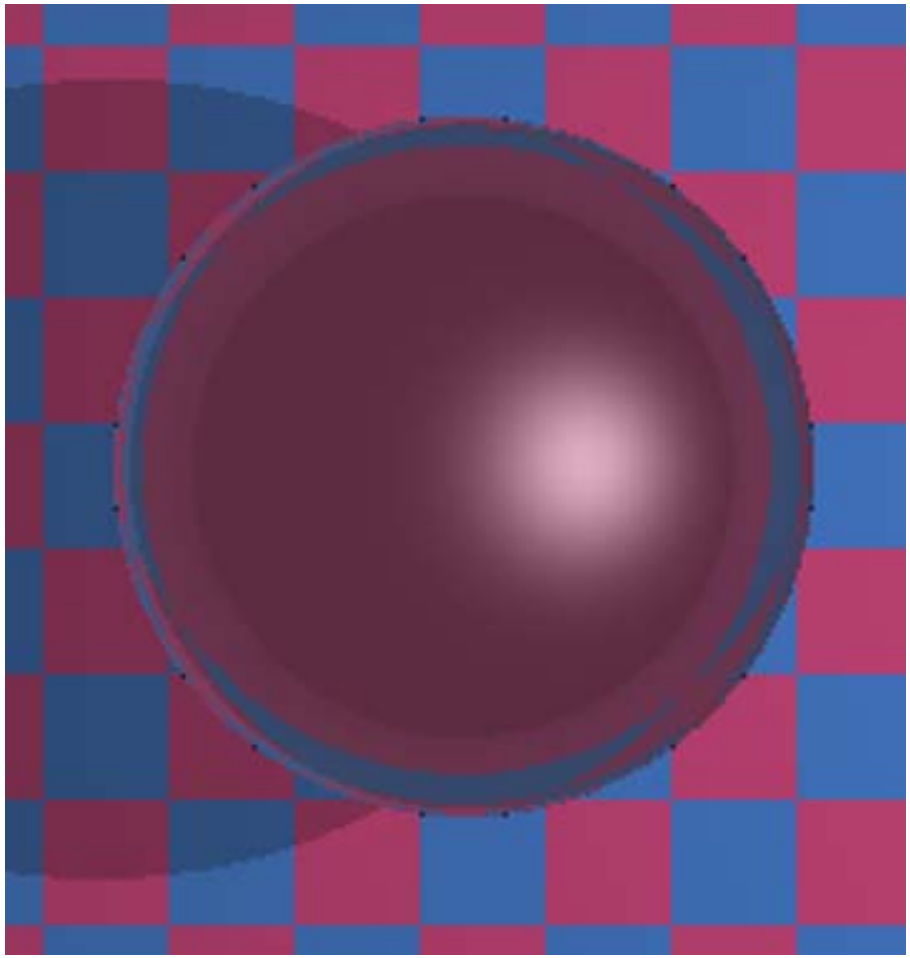
Indices of Refraction

Material	**η at λ=589.3 nm**
vacuum	1 (exactly)
helium	1.000036
air at STP	1.0002926
water ice	1.31
liquid water (20°C)	1.333
ethanol	1.36
glycerine	1.4729
rock salt	1.516
glass (typical)	1.5 to 1.9
cubic zirconia	2.15 to 2.18
diamond	2.419

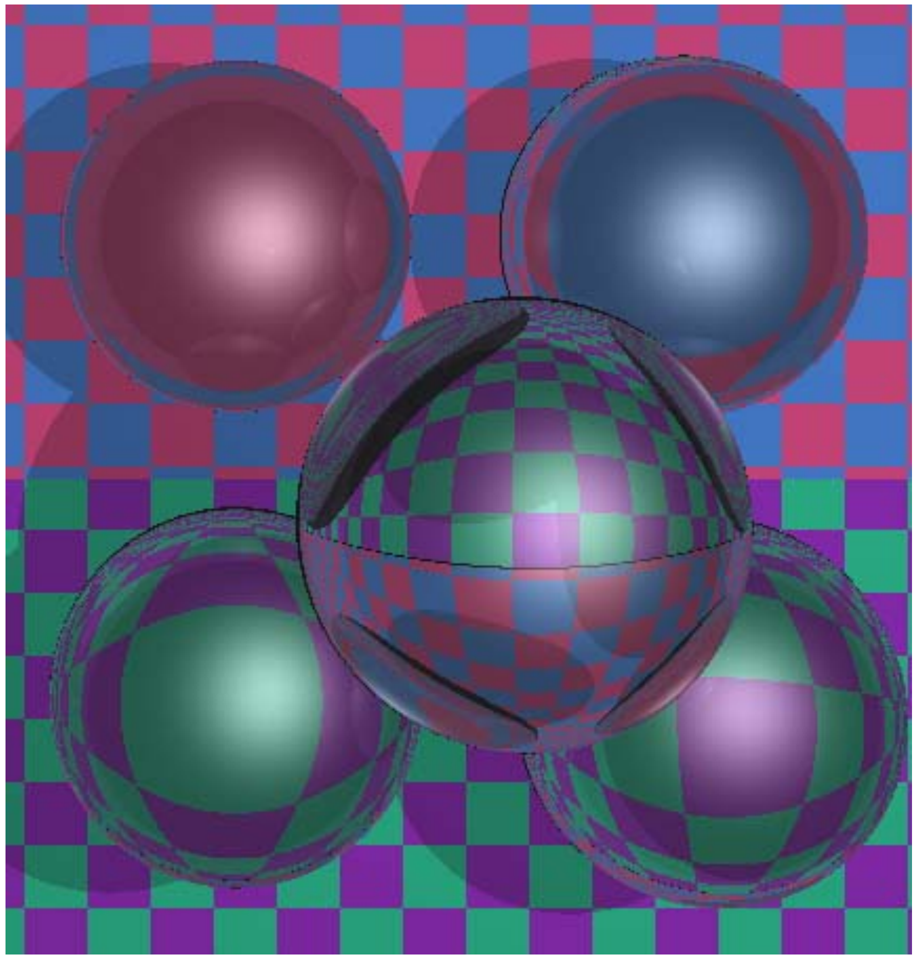
http://en.wikipedia.org/wiki/List_of_indices_of_refraction



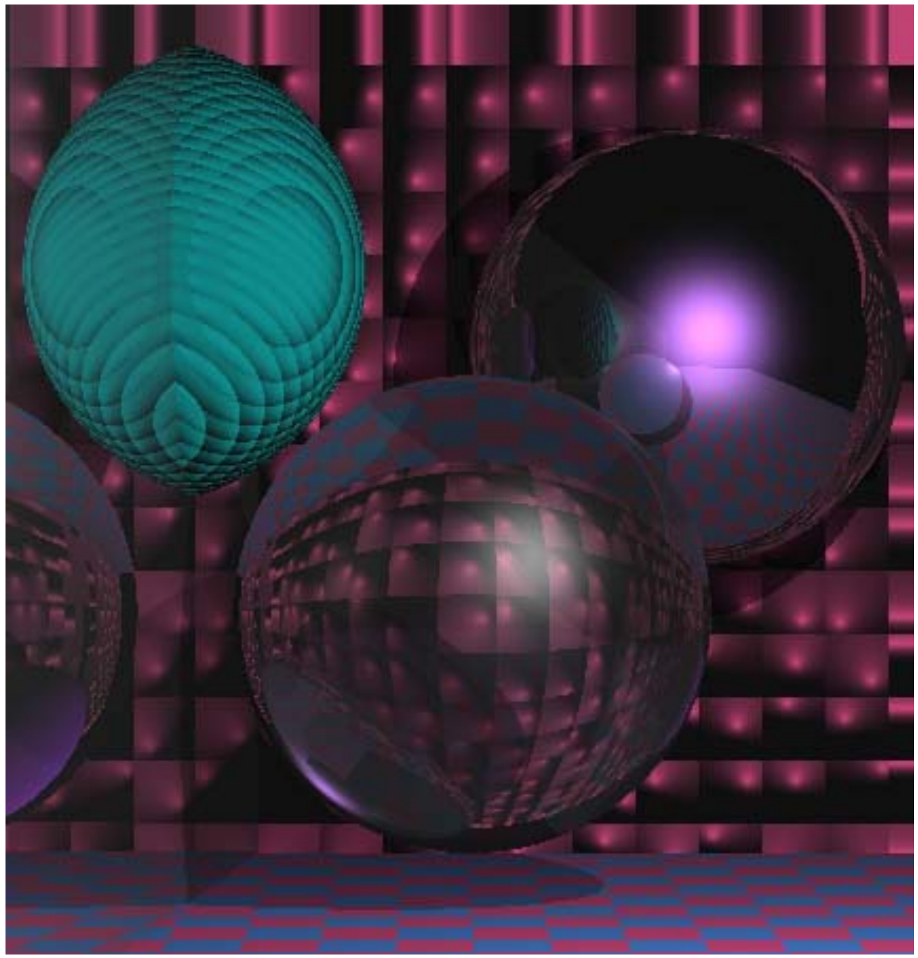
One Glass Sphere



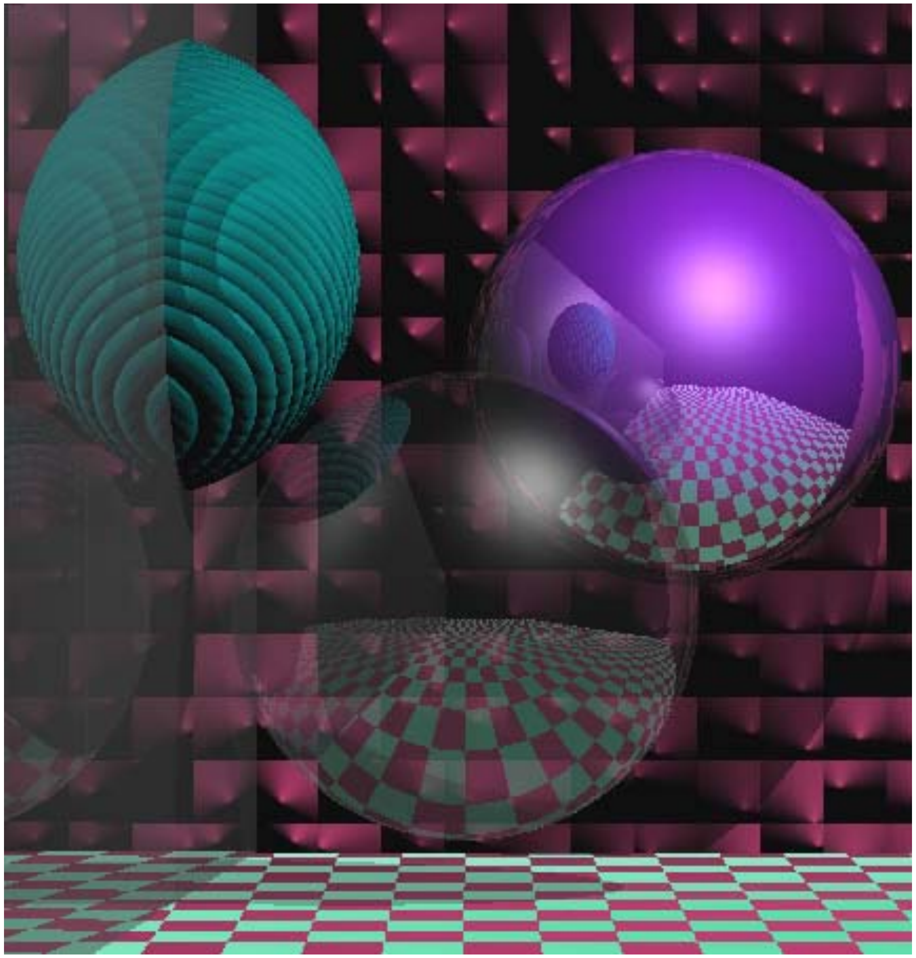
Five Glass Balls



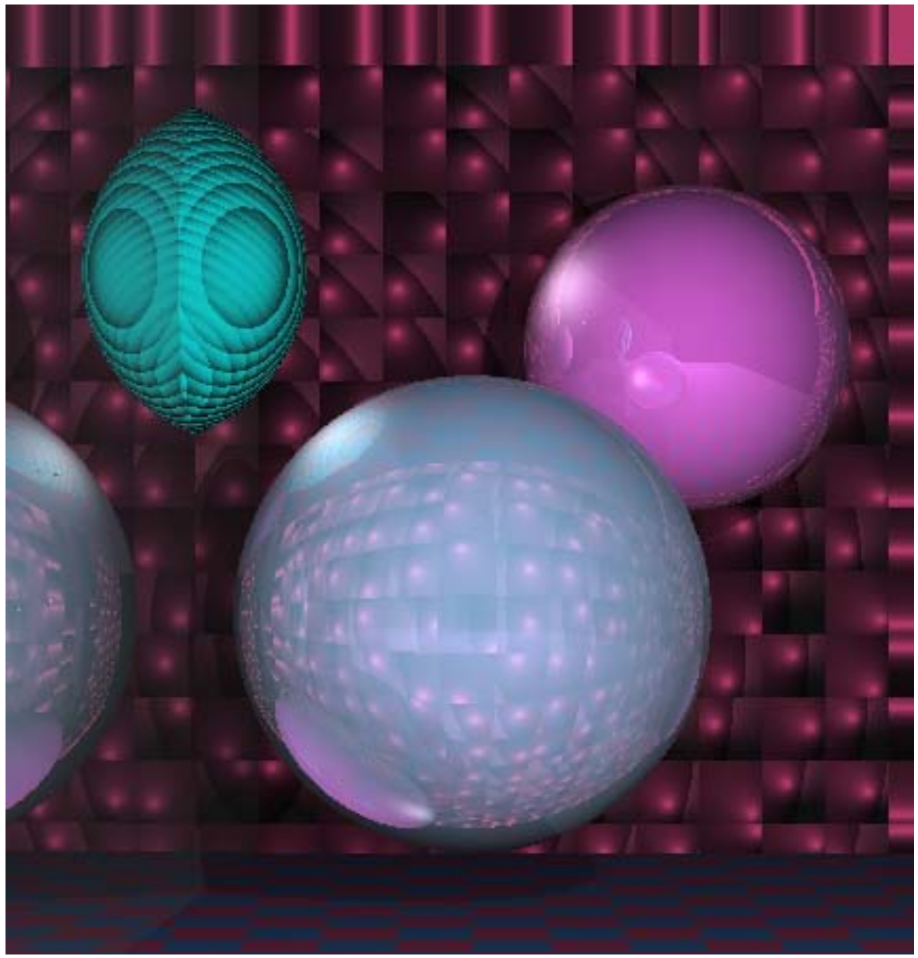
A Familiar Scene



Bubble



Milky Sphere



Lens - Carl Andrews 1999