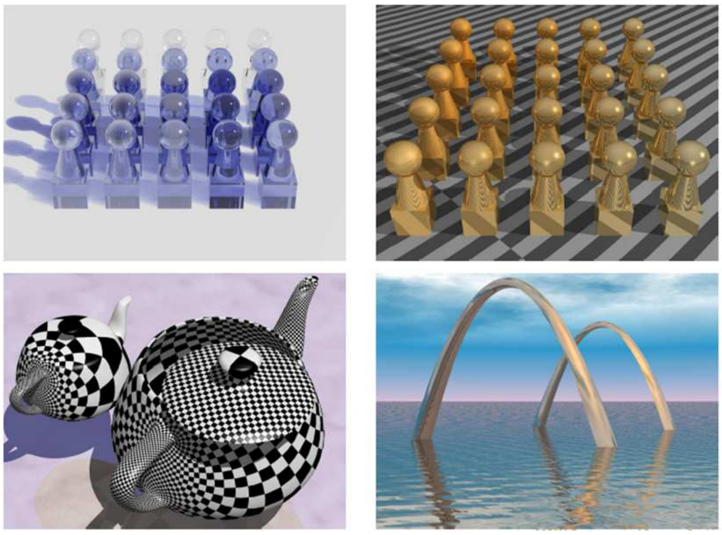
Ray Tracing

1



2

Ray Tracing – Global Illumination Model

- Simulate complex, photo-realistic illumination.
- Major algorithm in photo-realistic graphics.
- Apple A described the algorithm.
- Kay and Greenberg considered the refraction in 1979
- Whitted developed the first global illumination model, demonstrated the first general raytracer in 1981

3

Principle of Ray Tracing

- Light interaction among surfaces: the lights issued by the light sources reflect and refract when they reach the surface of the objects
- Fundamentals of ray tracing
  - Direct lights: lights issued by the light sources
  - Indirect lights: lights reflected and refracted by surfaces
  - Direct reflection and direct refraction: the reflection and refraction of the direct lights
  - Indirect reflection and indirect refraction: the reflection and refraction of the indirect lights

4

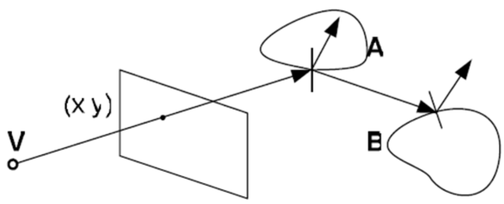
Ray Tracing Algorithm

- Basic algorithm traces the mirror reflection and refraction of light rays
- The light sources issue lights. The lights hit the surfaces and are reflected or refracted. The propagation directions of lights are changed. Lights continue the propagation along the new directions, until they hit new objects
- A small portion of the lights issued by the light sources will enter the eyes
- In reality, the propagation direction of lights are opposite to the direction of ray tracing

5

Ray Tracing Algorithm

- A ray is issued from the view point V through a pixel (x, y)
- The ray hits the first object along its propagation
- The reflection and refraction take place at the intersection point
- Continue to trace the reflected ray and refracted ray



6

Ray Tracing Algorithm

- There are 4 types of rays in the ray tracing algorithm

1. Viewing ray: the ray from view point and through a pixel
2. Shading test ray: the ray connecting a point on the surface and the light source
3. Reflection ray: reflected viewing ray
4. Refraction ray: refracted viewing ray

7

Ray Tracing Algorithm

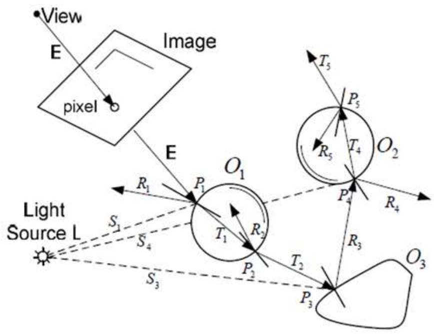
- Color intensity: A ray R intersects a surface at point p, there are 3 parts of lights at p contributing to R

1. The local light intensity caused by the light sources at p directly
2. The intensity of the reflection ray I_sK_s, I_s is obtained by recursively tracing the reflection ray
3. the intensity of the refraction ray I_tK_t, It is obtained by recursively tracing the refraction ray

8

Ray Tracing Algorithm

- The inverse process of real physical illumination process
- Simulate mirror reflection and refraction on ideal surfaces



9

Termination Criterion

- In practice, the tracing process is finite
- The termination condition for recursion:

1. The ray doesn’t hit any object
2. The ray goes to the ambient space
3. After many reflection and refractions, the intensity of the ray is too small
4. Recursive depth is beyond a predefined threshold

10

Algorithm

- Go through each pixel on the screen, the pixel and the view point determines a unique ray. Trace the ray using the following algorithm to compute the color of the pixel.


```
RayTracing (start, direction, weight, color) {
   if (weight < MinWeight)
      color = black;
   else {
      Compute the intersection point of the ray
      with all surfaces;
      select the intersection point P closest to start;
      if (no intersection point)
            color = black;
      else {
            Il, compute local intensity at the
            intersection point P;
            Compute reflection direction R;
            RayTracing (P, R, weight * Ws, Is);
            Compute refraction direction T;
            RayTracing (P, T, weight * Wt, It);
            color = Il + Ks Is + Kt It;
      }
   }
}
```

Ray Intersection

- Reflection and Refraction algorithm - Whitted algorithm.
- Ray tracing requires huge amount of intersection computation. The efficiency of intersection affects the efficiency of the whole system
- Intersection algorithm is the core of ray tracing
- Given a ray $\boldsymbol{e} + t\boldsymbol{d}$
  - Where $\boldsymbol{e}$ is the starting point, ***d*** is the direction and ***t*** is distance parameter $\boldsymbol{t} \in [\mathbf{0}, \infty]$
  - We want to find the intersection with any object such that $\boldsymbol{t} > \mathbf{0}$

13

Ray-Sphere Intersection

- Sphere is the most commonly used element in ray tracing
- Spheres are used as the bounding volumes of more complicated surfaces
- Assume (x_0, y_0, z_0) is the starting point of the ray; (x_d, y_d, z_d) is the normalized ray direction; (x_c, y_c, z_c) is the center of the sphere; R is the radius of the sphere
- Compute the conditions for the ray to intersect the sphere

14

Ray-Sphere Intersection

- Ray and the sphere equations

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = R^2, \begin{cases} x = x_0 + x_d t \\ y = y_0 + y_d t \\ z = z_0 + z_d t \end{cases}$$

- Plug in a merge to get

$$At^2 + Bt + C = 0$$

$$A = x_d^2 + y_d^2 + z_d^2 = 1$$
$$B = 2[x_d(x_0 - x_c) + y_d(y_0 - y_c) + z_d(z_0 - z_c)]$$
$$C = (x_0 - x_c)^2 + (x_0 - y_c)^2 + (x_0 - z_c)^2 - R^2$$

15

Ray-Sphere Intersection

- Solve the equation

$$t = \frac{-B \mp \sqrt{B^2 - 4C}}{2}$$

- $B^2 - 4C < 0$, no intersection
- $B^2 - 4C = 0$, the ray is tangent to the sphere
- $B^2 - 4C > 0$, there are two intersection point
- If $t < 0$ the intersection point is invalid
- Plug t in the ray equation to get the intersection point

- Intersection point at (x_i, y_i, z_i)
- Normal vector at the intersection point:

$$\left(\frac{x_i - x_c}{R}, \frac{y_i - y_c}{R}, \frac{z_i - z_c}{R}\right)$$

16

Ray-Polygon Intersection

- General algorithm has 2 steps:
- Compute the intersection of the ray with the **plane** of the polygon
- Verify whether the intersection point is inside or outside the polygon

17

Ray-Polygon Intersection

- The plane equation is $(p - p_0).n = 0$, where p_0 is a point in the plane and n is the plane normal vector
- $n = (p_1 - p_0) \times (p_2 - p_0)$
- Solving for t

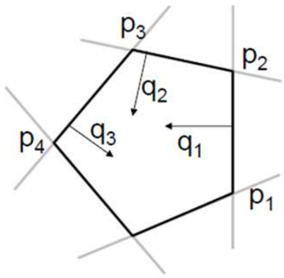
$$t = \frac{(p_0 - e).n}{d.n}$$

- if $d.n = 0$ (ray parallel to the plane) then no intersection

18

Ray-Polygon Intersection

- Assume convex polygon with N sides
  - Such polygons can be thought of as intersections of half-planes, each bounded by a line extending an edge.



q_i = perpendicular to (p_i, p_{i+1}) in the plane of the polygon;

$$q_i = n \times (p_{i+1} - p_i)$$

19

Ray-Polygon Intersection

- To be inside the polygon, a point p_t should be on the left side of all lines (p_i, p_{i+1})

- Testing for the side of a line (p_i, p_{i+1}):
  - $(p_t - p_i).\ q_i \geq 0$ means on the left (cosine of the angle between $p_t - p_i$ and q_i is positive

20

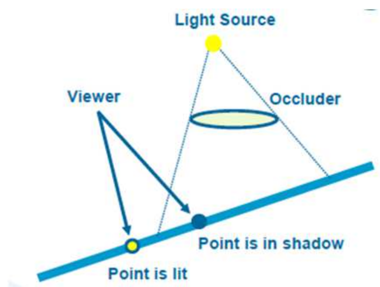
Ray Tracing with Textures

- Applied at local lighting computation
  - Once we know an intersection point
  - We know: Position, normal, color
- For textures
  - We know: Position, normal and texture coordinates, texture id
  - Apply texture retrieving to get the color
  - Then continue the local shading computation

21

Shadows

- A point is in shadow if the light got blocked between the light source and point



- Need information about blocked / non blocked pixels

22

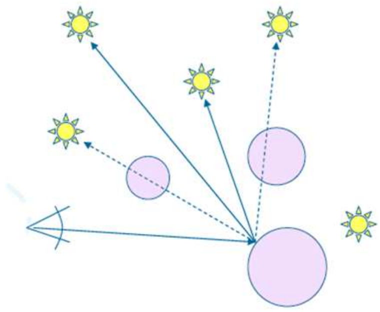
Shadow Rays

- Shadows are an important lighting effect that can easily be computed with ray tracing
- If we wish to compute the illumination with shadows for a point, we shoot a shadow ray from the point to every light source
- A light is only allowed to contribute to the final color if the ray doesn't hit anything in between the point and the light source

23

Shadow Rays

- We don't need to shoot a shadow ray to a light source if the dot product of the normal with the light direction is negative
- Shadow rays only need to know if something is hit or not
- You can include partially transparent objects with more computations



24

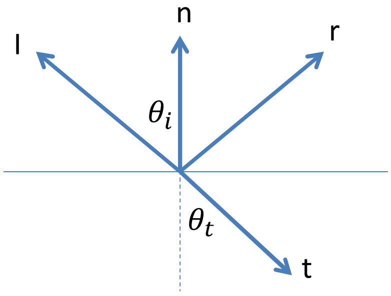
Refraction Direction

- Transparent objects refract light e.g. water, glass, diamonds
- Transparent surfaces possess a property called the index of refraction. According to Snell's law, the angle of incidence θ_i and the angle of transmission θ_T are related by the equation

$$\eta_i sin\theta_i = \eta_t sin\theta_t$$

η_i index of refraction of material light is leaving

η_t index of refraction of material light is entering



25

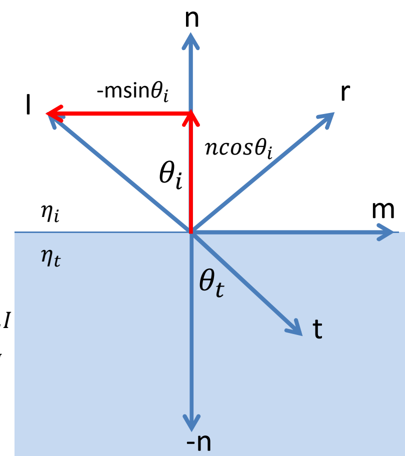
Refraction Direction Computation

$I = ncos\theta_i$ - msinθ_i

$m = (I - ncos\theta_i)/sin\theta_i$

$$
\begin{aligned}
t &= -ncos\theta_t + \text{msin}\theta_t \\
&= -ncos\theta_t + (I - ncos\theta_i)\text{sin}\theta_t/sin\theta_i \\
&= -ncos\theta_t + (I - ncos\theta_i)\eta_r \\
&= [\eta_r cos\theta_i - cos\theta_t]n - \eta_r I \\
&= [\eta_r cos\theta_i - \sqrt{1 - sin^2\theta_t}]n - \eta_r I \\
&= [\eta_r cos\theta_i - \sqrt{1 - \eta_r^2 sin^2\theta_i}]n - \eta_r I \\
&= [\eta_r cos\theta_i - \sqrt{1 - \eta_r^2(1 - cos^2\theta_i)}]n - \eta_r I
\end{aligned}
$$

$$t = [\eta_r(n.I) - \sqrt{1 - \eta_r^2(1 - (n.I)^2)}]n - \eta_r I$$



$$\frac{sin\theta_t}{sin\theta_i} = \frac{\eta_i}{\eta_t} = \eta_r$$

- Total internal reflection when the square root is imaginary
- Don't forget to normalize!

26