Ray Casting

Outline
- Ray casting algorithm
- Construct ray from viewpoint to pixel
- Intersection calculation
  - Ray-triangle
  - Ray-polygon
  - Ray-box
  - Ray-sphere
  - Ray-cylinder

Ray Casting Algorithm
- An image space visibility determination algorithm
- Shoot rays through each of the pixels
- Find the closest surface intersected by the ray
- Compute the radiance emitting from the intersection

Psudeocode for the Algorithm
```plaintext
for ( v=0 ; v<height ; v++ )
  for ( u=0 ; u<width ; u++ ) {
    construct a ray from viewpoint to pixel (u,v);
    set min_dist = ∞, min_obj = null;
    for ( each object k in scene ) {
      if ( the ray intersects object k )
        if ( distance to k < min_dist )
          update min_dist & min_obj;
    }
    compute the color of (u,v) based on min_obj;
  }
```

Construct Ray (2D Example)
- Known:
  - Center of projection: O
  - View direction: D
  - Right direction: R
  - Focal length: f
  - Image size: 2w
  - Image resolution: width
- Approach:
  - \( P = O + f*D + \left(\frac{w*2-1}{w*2}\right)*w*R \)
  - \( V = (P-O) / |P-O| \)

Ray-Triangle Intersection
- Assume an intersection exist, which is P
- \( O + t*V = C + P \)
- Intersection is on the plane:
  - \( P \cdot N = 0 \)
- Therefore:
  - \( (O + t*V - C) \cdot N = 0 \)
  - \( t = \frac{(C-O) \cdot N}{V \cdot N} \)
Ray-Triangle Solution Cases
- If $\mathbf{V} \cdot \mathbf{N} = 0$
  - Ray parallel with triangle
  - No intersection
- If $t < 0$
  - Intersection is behind viewpoint
  - Is intersection inside triangle?
    - Calculate Areal coordinates $(\alpha, \beta, \gamma)$
    - Test whether $0 \leq \alpha, \beta, \gamma \leq 1$

Ray-Polygon Intersect
- Only difference with ray-triangle calculation is intersection-in-polygon test
- Project intersection & polygon onto a 2D plane
  - Test whether the projection of intersection is inside the projection of polygon
  - Convert to a 2D raster graphics problem

Which Plane to Use
- Projecting to a general 3D plane is difficult
- Easier to use one of the XY, XZ, YZ planes
- Polygon may degenerate into a line in one or two of the planes
- Cannot do point-in-polygon test
- Should use the one on which the projection is the largest

Ray-Box Intersect
- Instead of intersecting 6 individual polygons, we can intersect the ray with 6 planes
- Each pair of parallel plane gives one parameter interval
  - $O_x + t_x \mathbf{V}_x = A_x$
  - $O_x + t_x \mathbf{V}_x = B_x$
  - $O_y + t_y \mathbf{V}_y = A_y$
  - $O_y + t_y \mathbf{V}_y = B_y$
  - $O_z + t_z \mathbf{V}_z = A_z$
  - $O_z + t_z \mathbf{V}_z = B_z$

Ray-Box Intersect (Cont’d)
- Calculation of entering & leaving intersections is similar to parametric line clipping algorithm
  - $X_{in} = \min(t_{x1}, t_{x2})$
  - $X_{out} = \max(t_{x1}, t_{x2})$
  - $Y_{in} = \min(t_{y1}, t_{y2})$
  - $Y_{out} = \max(t_{y1}, t_{y2})$
  - $Z_{in} = \min(t_{z1}, t_{z2})$
  - $Z_{out} = \max(t_{z1}, t_{z2})$
  - $t_{in} = \max(X_{in}, Y_{in}, Z_{in})$
  - $t_{out} = \min(X_{out}, Y_{out}, Z_{out})$

Ray-Box Solution Cases
- If $t_{in} > t_{out}$
  - The ray leaves the box before it enters
  - No intersection
- If $t_{in} < t_{out} < 0$
  - Box is behind viewpoint
- If $0 < t_{in} < t_{out}$
  - Box is in front
    - Use $t_{in}$
  - If $t_{in} < 0 < t_{out}$
    - Viewpoint is in the box
      - Use $t_{out}$
Ray-Sphere Equation

- Assume an intersection exist, which is \( P \)
  - \( O + t*V = C + P \)
- Intersection is on the surface of the sphere:
  - \(|P| = r\)
  - \(P\cdot P = r^2\)
- Quadratic equation is established in \( t \):
  - \(t^2 + 2*V\cdot (O-C)*t - r^2 + |O-C|^2 = 0\)

Ray-Sphere Solution Cases

- If no root exists:
  - No intersection
- If both roots < 0:
  - Sphere is behind
- If both roots > 0:
  - Sphere is in front
  - Use the smaller root
- If one root < 0 & the other root > 0:
  - Viewpoint is inside the sphere
  - Use the positive root

Ray-Cylinder Intersection

- Assume an intersection exist
  - \( O + t*V = C + k*H + P \)
- \( P \) is perpendicular to \( H \)
  - \(O\cdot H + t*V\cdot H = C\cdot H + k\)
  - \(k = O\cdot H + t*V\cdot H - C\cdot H\)
- Substitute \( k \)
  - \(P = O-C + t*V - (O\cdot H + t*V\cdot H - C\cdot H)*H\)
- Length of \( P \) is \( r \):
  - \(P\cdot P = r^2\)

Ray-Cylinder Solution Cases

- If no root exists:
  - No intersection
- If both roots < 0:
  - Cylinder is behind viewpoint
- If the corresponding \( k \) > \(|D-C|\) or \( k < 0 \):
  - Ray intersects the extension of the cylinder
  - No intersection

Intersect with CSG Models

- First intersect the ray with each leaf node
  - The number of intersections is even
  - Each intersection pair specifies a portion of ray inside the solid
- Then determine the portion inside the combined solid based on the Boolean operator
- Finally find the closest intersection