Ray Tracing Polygons

Overview

- Barycentric Coordinates
- Ray-Polygon Intersection Test

- In this part, we will talk about ray tracing *one* polygon
- Next part, we will work on large objects

Line Equation

- Recall that given p_1 and p_2 in 3D space, the straight line that passes between is:

$$p(t) = (1 - t)p_1 + tp_2$$

for any real number t

- This is a simple example of a ***barycentric combination***

Barycentric Combinations

- A barycentric combination is: a weighted sum of points, where the weights sum to 1.
  - Let $p_1, p_2, \ldots, p_n$ be points
  - Let $a_1, a_2, \ldots, a_n$ be weights

$$p = \sum_{i=i}^{n} a_i p_i$$

$$\sum_{i=1}^{n} a_i = 1$$

Implications

- If $p_1, p_2, \ldots, p_n$ are co-planar points then p as defined will be inside the polygon (convex hull) defined by the points, if and only if

$$0 \leq a_i \quad \forall i$$

- Proof of this is out of scope, but a few diagrams should convince you of the outline of a proof …

Ray Tracing a Polygon

- Three steps
  - Does the ray intersect the plane of the polygon?
    - i.e., is the ray not orthogonal to the plane normal
  - Intersect ray with plane
  - Test whether intersection point lies within polygon on the plane

Does the Ray Intersect the Plane?

- Ray equation is: $r(t) = p_0 + t{*}d$
- Plane equation is: $n.(x,y,z) = k$

- Then test is $n.d \mathrel{!=} 0$

  → ray does intersect plane (ray direction and plane are not parallel)

Where Does It Intersect?

- Substitute line equation into plane equation

$$n \cdot \left(x_0 + td_x \quad y_0 + td_y \quad z_0 + td_z \right) = k$$

- Solve for t

$$t = \frac{k - (n \cdot p_0)}{n \cdot d}$$

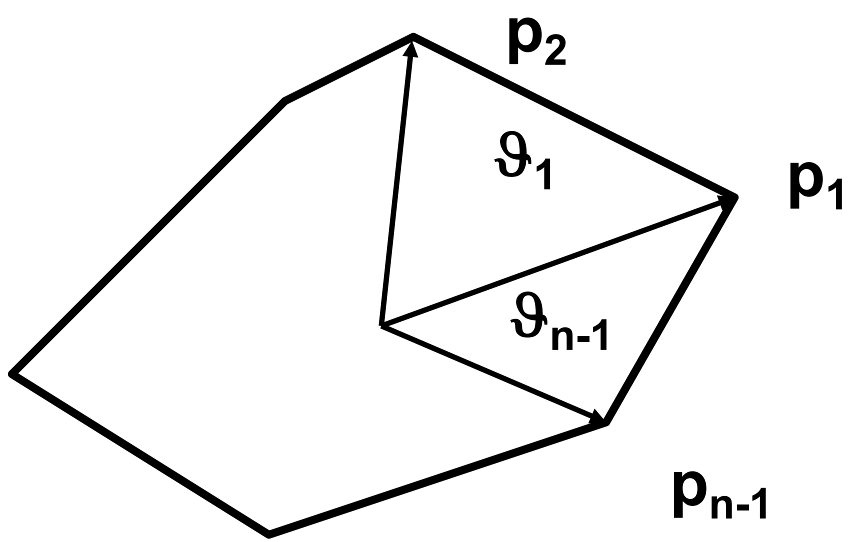
- Intersection: $p_{\text{int}} = p_0 + t * d$

Is This Point Inside the Polygon?

- If it is then it can be represented in barycentric co-ordinates with $0 \le a_i \quad \forall i$
- There are potentially several barycentric combinations (polygon with > 3 vertices)
- Many tests are possible
  - Winding number (can be done in 3D)
  - Infinite ray test (done in 2D)
  - Half-space test (done in 2D for convex poylgons)
  - Barycentric coordinates (in 3D, good for triangles)

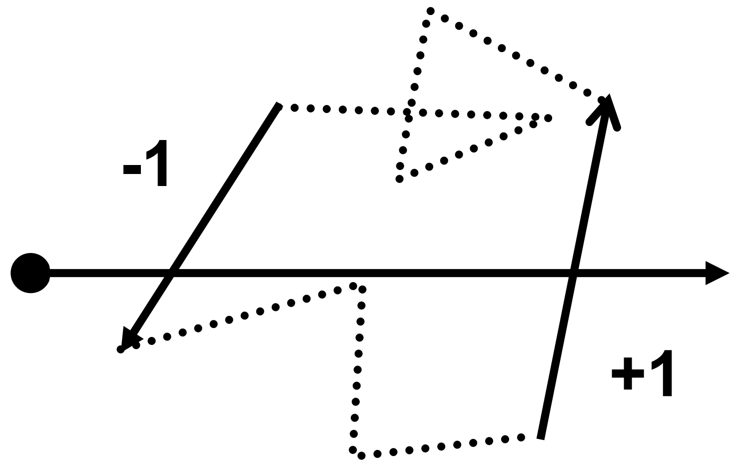
Method 1: Winding number test

- Sum the **signed** angles subtended by the vertices. If sum is zero, then outside. If sum is +/-2Π, inside.
- With non-convex shapes, can get +/-4Π, +/-6Π, etc…
- Q: How to get signed angles?



Method 2: Infinite Ray Test

- Draw a line from the test point to the outside
  - Count +1 if you cross an edge in an anti-clockwise sense
  - Count -1 if you cross and edge in a clockwise sense
- For convex polygons you can just count *the number of crossings*, ignoring the sign
- If total is even then point is outside, otherwise inside

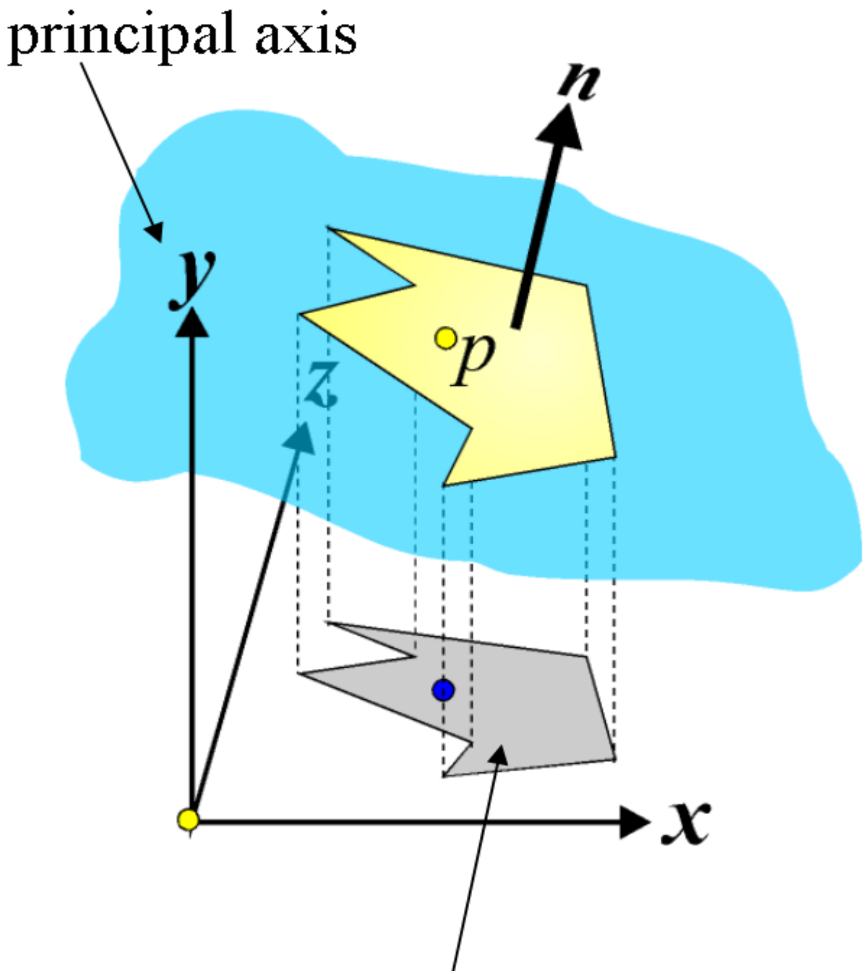


Infinite Ray Test

- How do you actually implement this?
- 2D
  - Pick ray along x-axis
  - Test if edges y-coordinates are above and below ray's y-coordinate
    - If so: count as intersection
- 3D
  - ?

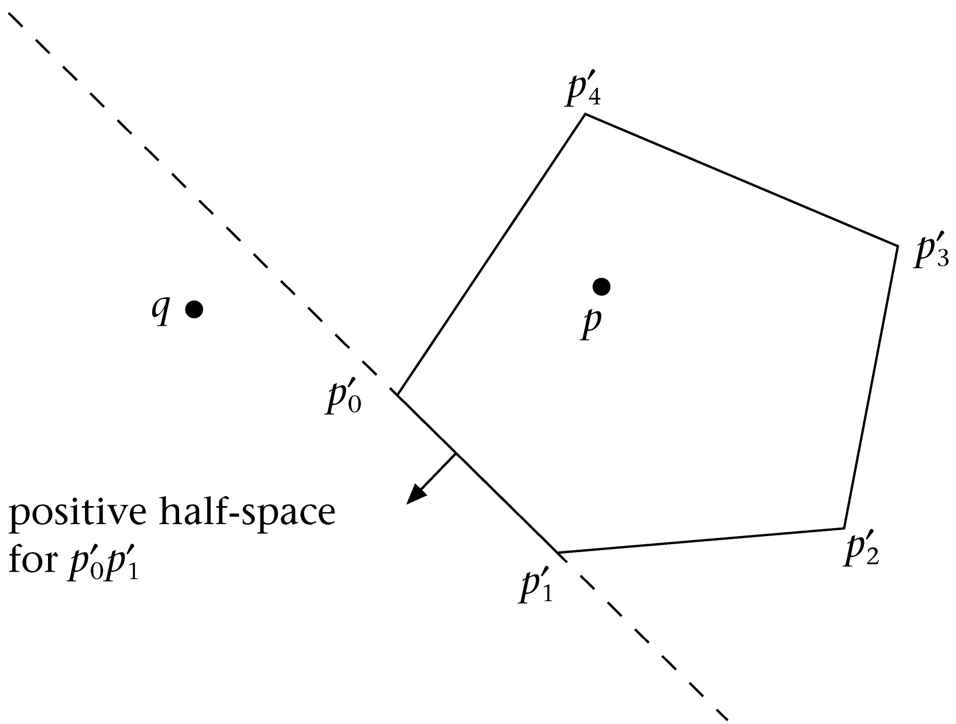
One Way to Project to 2D

- Select one of the principle planes
  - If the normal of the polygon is $n=(n_x, n_y, n_z)$ then we chose the one corresponding to the biggest of the three n_i values



Method 3: Half-Space Test (Convex)

- A point p is inside a polygon if it is in the negative half-space of all the line segments



Half-Space Test (Convex Polygons)

- 2D:
  - Check against line equations
- 3D:
  - Check against planes at each edge
    (plane orthgonal to polygon containing edge)
  - Or: ?

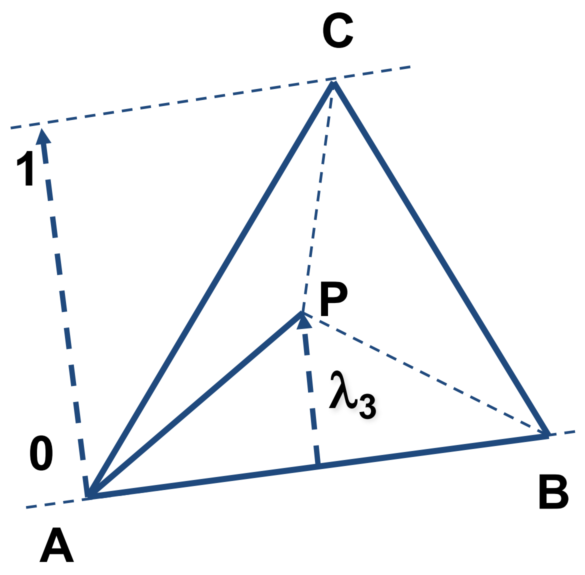
Method 4. Triangle inside/outside

- Compute barycentric coordinates, and check if all

$$0 \leq \lambda_i \quad \forall i$$

- Compute barycentric coords with:
  - $\lambda_1 = \Delta(PBC) / \Delta(ABC)$
  - $\lambda_2 = \Delta(APC) / \Delta(ABC)$
  - $\lambda_3 = \Delta(ABP) / \Delta(ABC)$
  - Note: **Δ is signed area,** computed with determinant:

$$\Delta(\mathbf{ABC}) = \tfrac{1}{2}\begin{vmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \\ 1 & 1 & 1 \end{vmatrix}$$



Note

- That the winding angle and infinite ray tests only tell you **if** the point is inside the polygon, they do not get you a barycentric combination
- With some minor extensions, its easy to show that the half-space test finds a barycentric combination.
- Baryc. coord test obviously finds a barycentric combination

Recap

- We have seen how to ray-trace polygons, by turning the problem into a 2D problem
- We saw that we have to be clear what is inside a polygon
- The different tests are suitable in different situations: whether or not you need to know if the polygon was hit or not
  - E.G. if are doing collision detection you don't need to know where, but if you are doing you need texture coordinates
- The different algorithms have different efficiencies depending on whether you expect the ray to hit