Clipping

- Usage
  - Visualization
  - Others
- Geometry
  - Line
  - Polygon
  - Patches
- Types
  - Culling
  - Approximate
  - Exact

Problem

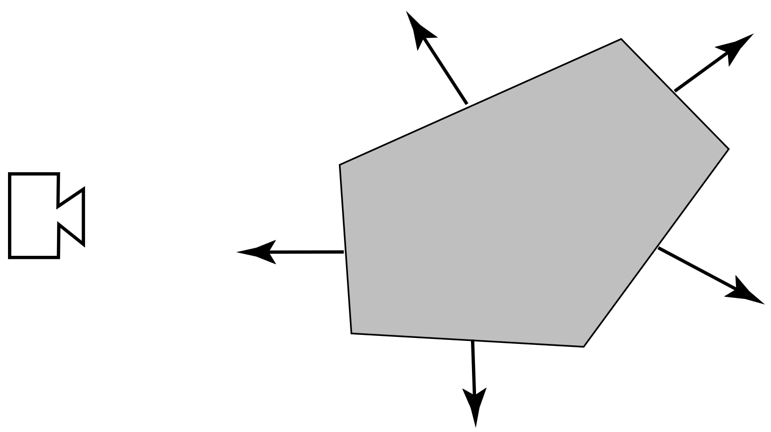
- Space Partition
  - Local
  - GLobal
- Separability

Clipping for Visualization

- Reasons
  - Avoid Projection Errors
  - Efficiency
- Strategies
  - Eliminate Simple Cases
  - Canonical Situation
- Clipping Volume (Pyramid)
  - Normalized
  - Convex
- Coordinate System
  - Euclidean Space
  - Projective Space

Backface Culling

- Discard Polygons Facing Away from Camera
- Dot Product with Viewing Direction



Hither Clipping

- Front Face of Clip Volume
- Types
  - Analytic
  - Subdivision
- Geometry
  - Generic

Line Clipping

- Problem
  - Split Line Segment
  - Intersect with Principal Directions
- Algorithms
  - Mid-Point Subdivision
  - Sutherland-Cohen

Sutherland-Cohen Clipping

```
LineClip (p0, p1)
    Classify p0 and p1
    while endpoints not inside
      if endpoints outside
        return out
      p = endpoint inside
      if p in left side
        split left
      else if p in right side
        split right
      else if p in above
        split top
      else if p in below
        split bottom
      Reclassify p
```

Endpoint Classification

- Strategy
  - Binary Code
  - Logical Operations

1001	1000	1010
0001	0000	0010
0101	0100	0110

$L = 0001$

$R = 0010$

$B = 0100$

$T = 1000$

Polygon Clipping

- Edge Schedule
  - Each Edge
  - All Planes
- Plane Schedule
  - Entire Polygon
  - Each Plane
- Polygon Types
  - Concave / Convex
  - Planar / Non Planar
- Clipping Region
  - Convex
- Pipeline
- Storage

Sutherland-Hodgman Clipping

```
PolygonClip(plane, pIn, pOut)
    for each edge
        Compute distances from plane
        if product of dist negative
            Split edge w/ plane
        Copy inside segment to pOut
    if last plane or num edges < 3
        return
    else
        PolygonClip(next plane, pOut, pIn)
```

Separable Clipping

- Two Step
  - Hither Clip
  - 2D Screen Clip

Homogeneous Clipping

- Avoid Wrap Around
  - Detect Crossing $w = 0$