Remainder of the Course

Hidden Surface Calculations

- **Clipping**
- **Z-buffering**
- **Painter's algorithm**
- **Object-space hidden line elimination**

Splines

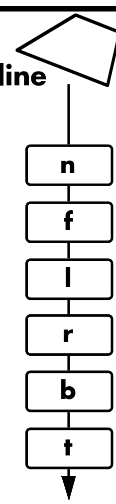
Animation

Reflection models and shading

Illumination algorithms and ray tracing

Clipping

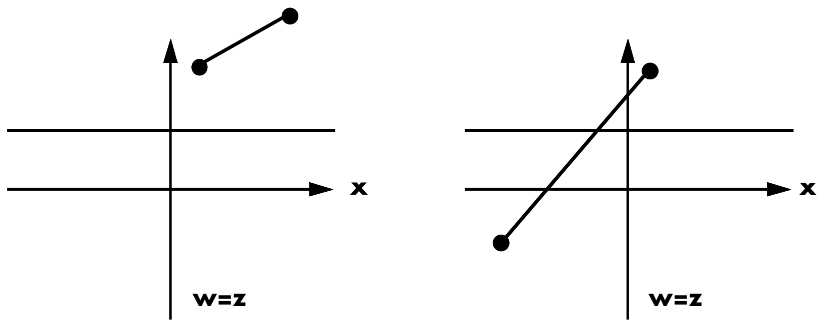
Clip polygon to 6 planes of the frustum in a pipeline




```
pt = input();
side = sign(pl.A o pt.X);
if( inside(side) )
  output(pt);
if( side != prevside ) {
  pt = intersect(prevpt,pt,pl);
  output(pt);
}
```

May add clipping planes, clip to convex volume

Clip BEFORE Projection



Visibility Analysis

Hidden/Visible surface/line elimination

Issues

- Handle diverse set of geometric primitives
- Handle large number of geometric primitives O(N)

Classification: Sutherland, Sproull, Schumacher (1974):

- Object Space
  - Geometric calculations involving polygons
  - Floating point precision: Exact
  - Object order
- Image Space
  - Visibility at pixel centers
  - Integer precision
  - Image order

1. Z-buffering

Maintain depth or range image

Zbuffer drawing mode

```
...
if( p.z < i[p.x][p.y].z ) {
  i[p.x][p.y].z = p.z;
  i[p.x][p.y].c = p.c;
}
p.z += p.dz;
p.c += p.dc;
...
```


Why So Popular?

Advantage
- **Simple to implement in hardware**
  - **Add additional z interpolator**
  - **Memory for z-buffer**
- **Diversity of primitives**
- **Unlimited complexity**

Disadvantage
- **Memory still too expensive for personal computers**
- **Point sampling causing jaggies**
- **Waste time drawing hidden objects**

Rendering Tricks: Decals

Graphics packages

May draw directly in the z-buffer

May mask bits to be written to framebuffers

Decal

1. **Draw base polygon using z-buffer compare,**
   but, *don't update z-buffer*
2. **Draw decal polygon using z-buffer compare,**
   but, *don't update z-buffer*
3. **Draw base polygon using z-buffer compare,**
   and, *update z-buffer*, but, don't *update color buffer*

Rendering Tricks: Hidden Lines

"Hollow" Polygons: Same pixels as normal polygons

Modes: Z-GE, 24-bit msb, z-range remaining 23-bits

1. **Draw polygon into z-buffer,**
   but only setting msb-bit in the z-buffer
2. **Draw outline into z-buffer,**
   clearing msb-bit in the z-buffer
   msb-bit is now set in the interior of the polygon
3. **Draw polygon into color-buffer using outline color**
   only perimeter values are near, thus outline is drawn
4. **Draw polygon into the z-buffer clearing the msb**

2. Painter's Algorithm

Draw surfaces in back to front order

Front surfaces are drawn over back surfaces, hiding them

Draw surfaces in front to back order using under operator

Previously drawn surfaces mask drawing of those behind

Key problem is priority determination

- **Manually**
- **Exploit spatial properties of the model**
- **BSP (Binary Space Partition) Tree**

3. Object-Space Algorithm

Robert's Algorithm

Input: Set of convex polyhedra

Output: Set of visible line segments

```
for each polyhedron
  form shadow volume
    discard back-facing faces
    keep front-facing faces
    create silhoutte planes
for each edge
  for each shadow volume
    clip edge to shadow volume
  draw visible edge
```