

CS-184: Computer Graphics

Lecture #10: Clipping and Hidden Surfaces

Prof. James O'Brien
University of California, Berkeley

V2013.S.10-1.0

1

Today

- Clipping
 - Clipping to view volume
 - Clipping arbitrary polygons
- Hidden Surface Removal
 - Z-Buffer
 - BSP Trees
 - Others

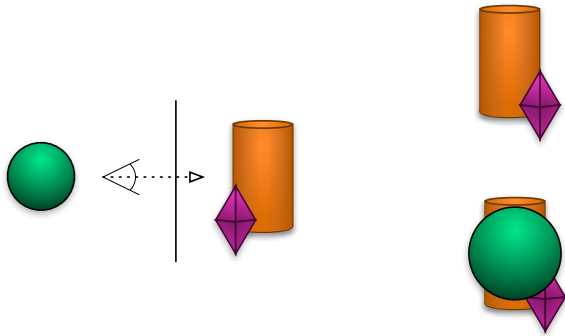
2

2

Sunday, March 3, 13

Clipping

- Stuff outside view volume should not be drawn
 - Too close: obscures view



3

3

Clipping

- Stuff outside view volume should not be drawn
 - Too close: obscures view
 - Too far:
 - Complexity
 - Z-buffer problems
 - Too high/low/right/left:
 - Memory errors
 - Broken algorithms
 - Complexity

4

4

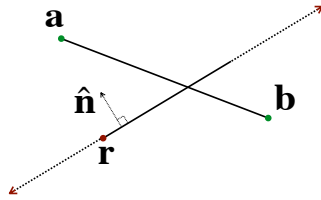
Clipping Line to Line/Plane

Line segment to be clipped

$$\mathbf{x}(t) = \mathbf{a} + t(\mathbf{b} - \mathbf{a})$$

Line/plane that clips it

$$\hat{\mathbf{n}} \cdot \mathbf{x} - \hat{\mathbf{n}} \cdot \mathbf{r} = 0$$



5

5

Clipping Line to Line/Plane

Line segment to be clipped

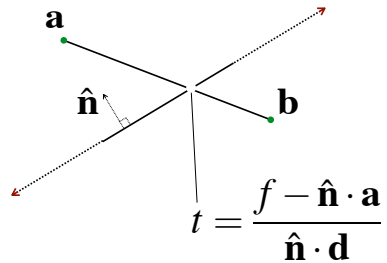
$$\mathbf{x}(t) = \mathbf{a} + t(\mathbf{b} - \mathbf{a})$$

Line/plane that clips it

$$\hat{\mathbf{n}} \cdot \mathbf{x} - f = 0$$

$$\hat{\mathbf{n}} \cdot (\mathbf{a} + t(\mathbf{b} - \mathbf{a})) - f = 0$$

$$\hat{\mathbf{n}} \cdot \mathbf{a} + t(\hat{\mathbf{n}} \cdot (\mathbf{b} - \mathbf{a})) - f = 0$$



6

6

Clipping Line to Line/Plane

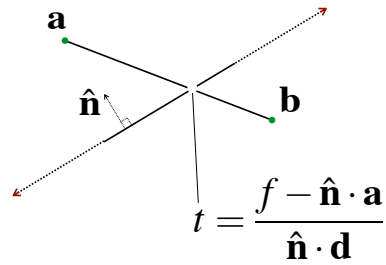
- Segment may be on one side

$$t \notin [0 \dots 1]$$

- Lines may be parallel

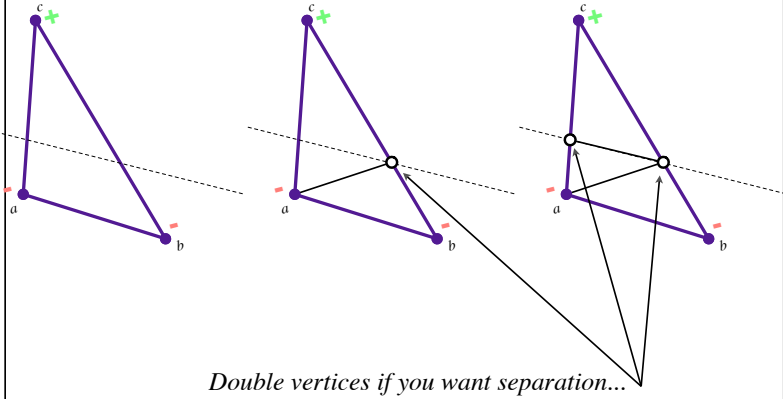
$$\hat{\mathbf{n}} \cdot \mathbf{d} = 0$$

$$|\hat{\mathbf{n}} \cdot \mathbf{d}| \leq \epsilon \quad (\text{Recall comments about numerical issues})$$



7

Triangle Clip/Split



8

Polygon Clip to Convex Domain

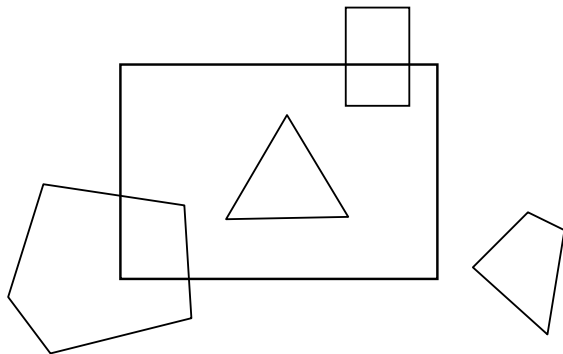
- Convex domain defined by collection of planes (or lines or hyper-planes)
- Planes have outward pointing normals
- Clip against each plane in turn
- Check for early/trivial rejection

9

9

Polygon Clipping

- Find the part of a polygon inside the clip window?

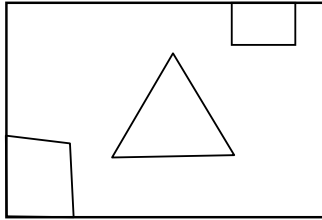


Before Clipping

10

Polygon Clipping

- Find the part of a polygon inside the clip window?

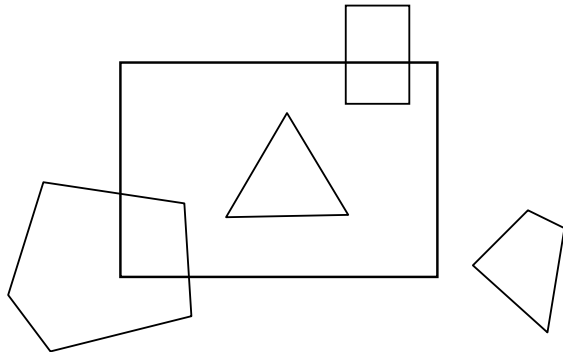


After Clipping

11

Sutherland-Hodgman Clipping

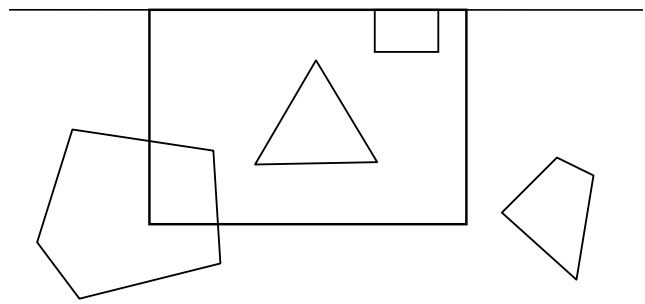
- Clip to each window boundary one at a time



12

Sutherland-Hodgman Clipping

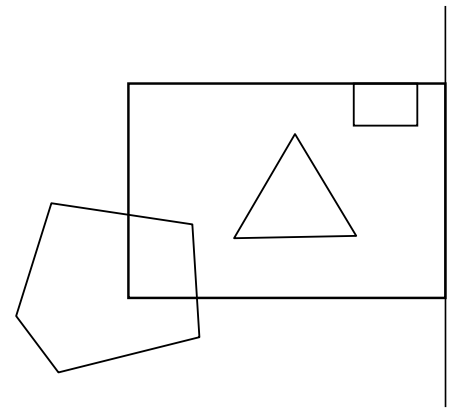
- Clip to each window boundary one at a time



13

Sutherland-Hodgman Clipping

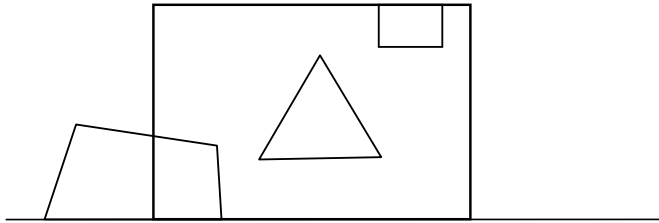
- Clip to each window boundary one at a time



14

Sutherland-Hodgman Clipping

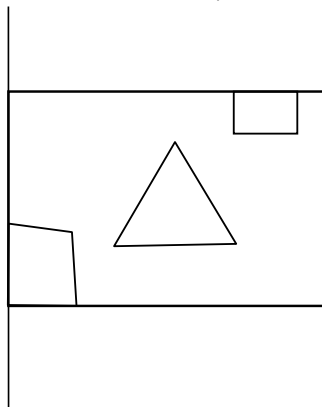
- Clip to each window boundary one at a time



15

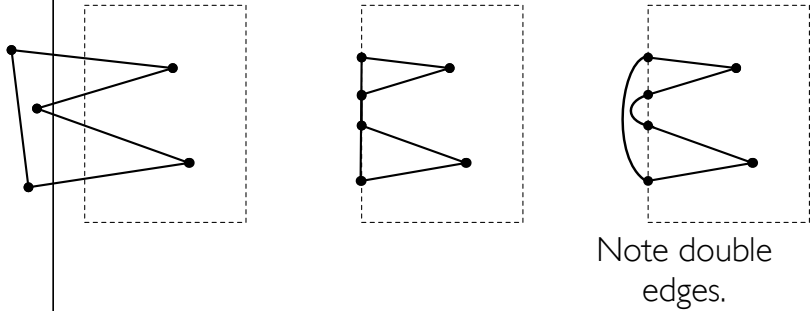
Sutherland-Hodgman Clipping

- Clip to each window boundary one at a time



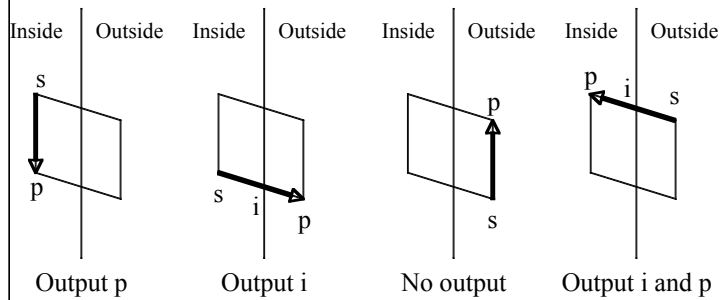
16

Polygon Clip to Convex Domain



17

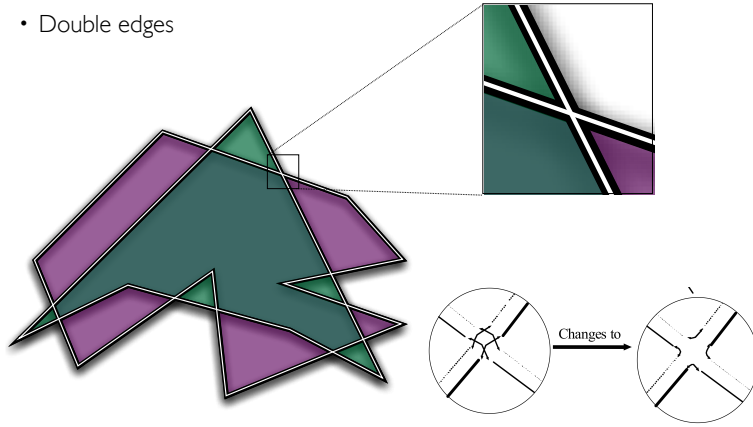
Polygon Clip to Convex Domain



18

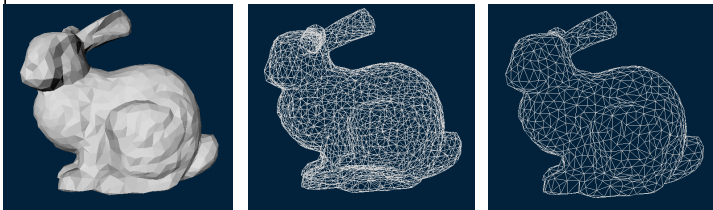
General Polygon Clipping

- Weiler Algorithm
 - Double edges



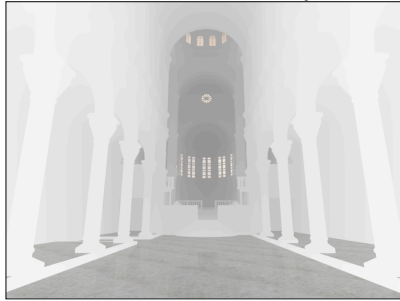
Hidden Surface Removal

- True 3D to 2D projection would put every thing overlapping into the view plane.
- We need to determine what's in front and display only that.



Z-Buffers

- Add extra depth channel to image
- Write Z values when writing pixels
- Test Z values before writing



23

Z-Buffers

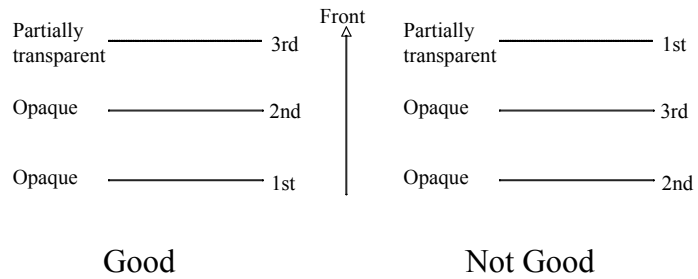
- Benefits
 - Easy to implement
 - Works for most any geometric primitive
 - Parallel operation in hardware
- Limitations
 - Quantization and aliasing artifacts
 - Overfill
 - Transparency does not work well

24

24

Z-Buffers

- Transparency requires partial sorting:



25

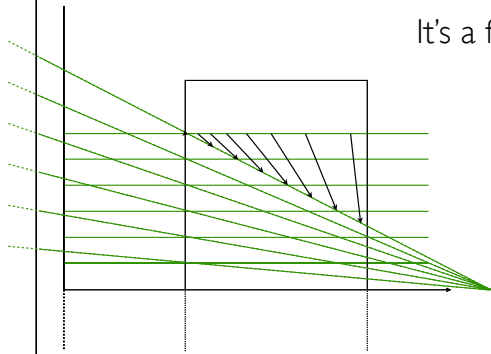
25

Z-Buffers

Recall depth-value distortions.

It's a feature...

More resolution near viewer
Best use of limited precision



26

26

A-Buffers

- Store sorted list of "fragments" at each pixel
- Draw all opaque stuff first then transparent
- Stuff behind full opacity gets ignored

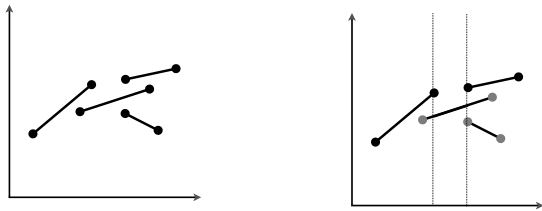
- Nice for antialiasing...

27

27

Scan-line Algorithm

- Assume polygons don't intersect
- Each time an edge is crossed determine who's on top

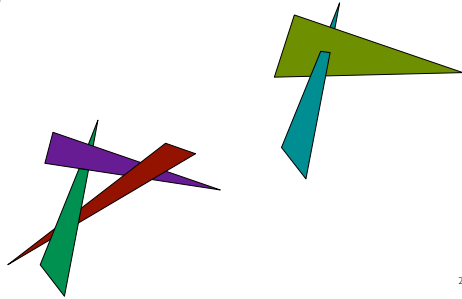


28

28

Painter's Algorithm

- Sort Polygons Front-to-Back
 - Draw in order
 - Back-to-Front works also, but wasteful
- How to sort quickly?
- Intersecting polygons?
- Cycles?



29

29

BSP-Trees

- Binary Space Partition Trees
 - Split space along planes
 - Allows fast queries of some spatial relations
- Draw Front-to-Back
 - Draw same-side polygons first
 - Draw root node polygon (if any)
 - Draw other-side polygons last

30

30
