Line and Polygon Clipping

Chapter 7

Clipping

Convexity

Set $C \subseteq \mathbb{R}^d$ is convex if for any two points $p, q \in C$ and any $\alpha \in [0, 1]$, $\alpha p + (1-\alpha)q \in C$

Convex

Non Convex

2D Projection of convex 3D shape is convex

Rendering Pipeline

Geometry Processing

Geometric Content

Model/View Transform

Lighting

Perspective Transform

Clipping

Scan Conversion

Texturing

Depth Test

Blending

Frame-buffer

- Discard geometry outside viewport window

Explicit Solution: Line Segments

Intersection of convex regions is convex

- Why?
  - $L$ & $D$ are convex - intersection is convex
  - $L$ & $D$ are convex - intersection is convex
  - single connected segment of $L$

Clipping uses intersections of $L$ with four boundary segments of window $D$

Line/Polygon Clipping (2D)

Problem:
Given a 2D line/polygon and a window, clip the line/polygon to their regions that are inside the window.

- Objectives
  - Efficiency
  - (Parallelization)
- Two approaches
  - Explicit (continuous setting)
  - Implicit (discrete setting) - part of scan conversion

Basic Method

Works, but inefficient for lines OUTSIDE $D$

- Four intersection tests
- Note: need special care for vertices ON window edges

Copyright Alla Sheffer UBC 2012
Line and Polygon Clipping

Segment-Segment Intersection

Intersection: $x$ & $y$ values equal in both representations - two linear equations in two unknowns $(x,y)$

$$ x' = x' = \frac{x_0 - x_1}{x_2 - x_1} \\
\frac{y_0 - y_1}{y_2 - y_1} \in [0,1] $$

Intersection with axis-aligned lines

$$ \begin{align*}
x' &= x' = \frac{x_0 - x_1}{x_2 - x_1} \\
y' &= y' = \frac{y_0 - y_1}{y_2 - y_1} \\
&\quad \text{if } 0 < t < 1 \text{ no intersection} \\
y' &= y' = \frac{y_0 - y_1}{y_2 - y_1} + (y_2 - y_1) t \quad (\text{relevant only for segments})
\end{align*} $$

Line Clipping

Cohen-Sutherland Algorithm (cont’d)

Given $L$ from $(x_0, y_0)$ to $(x_1, y_1)$ & rectangle $D$.

If bitwise and of the codes of $(x_0, y_0)$ and $(x_1, y_1)$ is not zero, or the bitwise or is zero,

then $L$ can be trivially handled (it is either totally outside or totally inside $D$).

Why?

Copyright Alla Sheffer UBC 2012
Line and Polygon Clipping

Cohen-Sutherland Algorithm (cont’d)

Determine portion of line inside axis-aligned box (viewing frustum in NDC)
- Simple extension of 2D algorithms
- After projection transform
  - clipping volume always the same
    - xmin=xmin=xmin=-1, xmax=xmax=zmax=1
  - boundary lines become boundary planes
    - but bit-codes still work the same way

3D clipping

Triangle Clipping

- How does intersection of rectangle & triangle looks like?
  - How many sides?
  - How to expand clipping to triangles?
    - Hint: it is convex
    - Will sketch on the board...

Copyright Alla Sheffer UBC 2012