Chapter 6

Clipping

The Rendering Pipeline

Geometry Database -> Model/View Transform. -> Lighting -> Perspective Transform. -> Clipping

Scan Conversion -> Texturing -> Depth Test -> Blending -> Frame-buffer

Copyright Alla Sheffer
UBC 2004
Problem:
Given a set of 2D lines/polygons and a window, clip the lines/polygons to their regions that are inside the window.

Objectives
• Efficiency
• Display in portion of screen (rectangular window)

Analytic Solution
- *Intersection* of convex regions is convex
  - Why?
    - \( L \) & \( D \) are *convex* - intersection is convex
      - single connected segment of \( L \)
    - **Question**: Can boundary of two convex shapes intersect more than twice?
    - Clipping - compute intersection of \( L \) with four boundary segments of window \( D \)
**Line-Line Intersection**

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

\[
G_1 = \begin{cases} 
  x^1(t) = x^1_0 + (x^1_1 - x^1_0)t \\
  y^1(t) = y^1_0 + (y^1_1 - y^1_0)t 
\end{cases} \quad t \in [0,1] \\
G_2 = \begin{cases} 
  x^2(r) = x^2_0 + (x^2_1 - x^2_0)r \\
  y^2(r) = y^2_0 + (y^2_1 - y^2_0)r 
\end{cases} \quad r \in [0,1]
\]

\[
x^1_0 + (x^1_1 - x^1_0)t = x^2_0 + (x^2_1 - x^2_0)r
\]

\[
y^1_0 + (y^1_1 - y^1_0)t = y^2_0 + (y^2_1 - y^2_0)r
\]

---

**Intersection with vertical/horizontal lines**

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

\[
G_1 = \begin{cases} 
  x^1(t) = x^1_0 + (x^1_1 - x^1_0)t \\
  y^1(t) = y^1_0 + (y^1_1 - y^1_0)t 
\end{cases} \quad t \in [0,1] \\
G_2 = \begin{cases} 
  x^2(r) = x^2_0 + (x^2_1 - x^2_0)r \\
  y^2(r) = y^2_0 + (y^2_1 - y^2_0)r 
\end{cases} \quad r \in [0,1]
\]

\[
x^1_0 + (x^1_1 - x^1_0)t = x^2_0
\]

\[
t = \frac{x^2_0 - x^1_0}{x^1_1 - x^1_0}
\]

\[
y^1_0 + (y^1_1 - y^1_0)t = y^2_0 + (y^2_1 - y^2_0)r
\]
Purpose:
Fast treatment of line segments that are trivially inside/outside window.

\[ P = (x, y) \] - point to be classified against window \( D \)

Idea: Assign to \( P \) a binary code consisting of a bit for each edge of \( D \), using lookup table:

<table>
<thead>
<tr>
<th>bit</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( y &lt; y_{\min} )</td>
<td>( y \geq y_{\min} )</td>
</tr>
<tr>
<td>2</td>
<td>( y &gt; y_{\max} )</td>
<td>( y \leq y_{\max} )</td>
</tr>
<tr>
<td>3</td>
<td>( x &gt; x_{\max} )</td>
<td>( x \leq x_{\max} )</td>
</tr>
<tr>
<td>4</td>
<td>( x &lt; x_{\min} )</td>
<td>( x \geq x_{\min} )</td>
</tr>
</tbody>
</table>

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?
Cohen-Sutherland Algorithm (cont’d)

\[ C - S - \text{Clip}(P_i, P_j, x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}) \]

\[ C_i \triangleq \text{code}(P_i); \quad C_j \triangleq \text{code}(P_j); \]

if \((C_i \text{ and } C_j) \neq 0 \) then return;

if \((C_i \text{ or } C_j) = 0 \) then draw \((P_i, P_j)\);

else if \((\text{OutsideWindow}(P_j)) \) then begin

\[ \text{Edge} \triangleq \text{Window boundary of leftmost non-zero bit of } C_j; \]

\[ P_i \triangleq P_i \cap \text{Edge}; \]

\[ C - S - \text{Clip}(P_i, P_j, x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}); \]

end

else

\[ \text{Edge} \triangleq \text{Window boundary of leftmost non-zero bit of } C_i; \]

\[ P_j \triangleq P_j \cap \text{Edge}; \]

\[ C - S - \text{Clip}(P_i, P_j, x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}); \]

end

Triangle Clipping

- How does intersection of rectangle & triangle looks like?
  - How to expand clipping to triangles?
    - Hint: it is convex
Questions: How can these ideas be used to design an algorithm for checking if:

- a point is inside a (convex) polygon?
- a (convex) polygon is inside/intersects/outside a (convex) polygon?