The Two Dimensional Viewing Pipeline

CS116A
Chris Pollett
Nov. 8, 2004.
Outline

• The Two Dimensional Viewing Pipeline
• The Clipping Window
• Normalization and Viewport Transformations
The Two Dimensional Viewing Pipeline

- Clipping window -- the part of two dimensional scene that it to be displayed
- Viewport -- window where data from clipping window will be displayed
- Mapping between these two called 2D viewing transformation
More Pipeline

1. Construct World-coordinate Scene Using Modeling-Coordinate Transformations
2. Convert World Coordinates to Viewing Coordinates
3. Transform Viewing Coordinates to Normalized Coordinates
4. Map Normalized Coordinates to Device Coordinates
The Clipping Window

• Most graphics packages support rectangular clipping regions

• Some systems support rotated 2D viewing frames, but usually clipping window must be specified in world coordinates.
Viewing Coordinate Clipping Window

• Can set up a **viewing coordinate system** within the world-coordinate frame.

• This viewing frame provides a reference for specifying a rectangular clipping window with any specified orientation and position.

• Choose \( P_0 = (x_0, y_0) \) base position, and a vector \( V \) that defines the \( y_{\text{view}} \) direction.

• \( V \) is called the **view up** vector.

• Alternative we could have used a rotational angle.
Getting into the Viewing Frame

• Translate the viewing origin to the world origin.
• Rotate the viewing system to align with the world frame.
• $M_{WC,VC} = RT$
• Suppose $P_0 = (1,2)$ and $V = (4,3)$. Calculate $M_{WC,VC}$
• To specify clipping window now give its lower and upper corner positions.
Clipping Window into Normalized Viewport

• Before mapping to the actual viewport we map into a normalized viewport of sides between 0 and 1 in each axis.

• We clip as we map into this normalized region.

• Then we do a straightforward transformation from the normalized viewport to the actual viewport.
More Normalized Viewport Mapping

• Let \((x_w, y_w)\) be a point in world coordinates. To map it into normalized viewport need:

\[
\frac{x_v - x_{v_{\text{min}}}}{x_{v_{\text{max}}} - x_{v_{\text{min}}}} = \frac{x_w - x_{w_{\text{min}}}}{x_{w_{\text{max}}} - x_{w_{\text{min}}}}
\]

\[
\frac{y_v - y_{v_{\text{min}}}}{y_{v_{\text{max}}} - y_{v_{\text{min}}}} = \frac{y_w - y_{w_{\text{min}}}}{y_{w_{\text{max}}} - y_{w_{\text{min}}}}
\]

• Solving gives \(x_v = s_x x_w + t_x\) and \(y_v = s_y y_w + t_y\) where

\[
s_x = \frac{x_{v_{\text{max}}} - x_{v_{\text{min}}}}{x_{w_{\text{max}}} - x_{w_{\text{min}}}}, \quad t_x = \frac{x_{w_{\text{max}}} - x_{v_{\text{min}}}}{x_{w_{\text{max}}} - x_{w_{\text{min}}}}
\]

\[
s_y = \frac{y_{v_{\text{max}}} - y_{v_{\text{min}}}}{y_{w_{\text{max}}} - y_{w_{\text{min}}}}, \quad t_y = \frac{y_{w_{\text{max}}} - y_{v_{\text{min}}}}{y_{w_{\text{max}}} - y_{w_{\text{min}}}}
\]

\(s_y\) and \(t_y\) are similar but with \(y\)’s instead of \(x\)’s everywhere.
More Mapping

- As a matrix, this gives $M_{\text{window, normviewp}}$:

$$
\begin{bmatrix}
  s_x & 0 & t_x \\
  0 & s_y & t_y \\
  0 & 0 & 1
\end{bmatrix}
$$
Clipping Window into Normalized Square

- Another approach is to map into the square with \( x \) values between -1 and 1 and \( y \) values between -1 and 1. Get matrix:

\[
M_{\text{window,normsquare}} = \begin{bmatrix}
\frac{2}{(x_{\text{wmax}}-x_{\text{wmin}})} & 0 & -\frac{(x_{\text{wmax}}+x_{\text{wmin}})}{(x_{\text{wmax}}-x_{\text{wmin}})} \\
0 & \frac{2}{(y_{\text{wmax}}-y_{\text{wmin}})} & -\frac{(y_{\text{wmax}}+y_{\text{wmin}})}{(y_{\text{wmax}}-y_{\text{wmin}})} \\
0 & 0 & 1
\end{bmatrix}
\]

- After clipping, to map to viewport one applies:

\[
M_{\text{normsquare,viewsquare}} = \begin{bmatrix}
\frac{(x_{\text{vmax}}-x_{\text{vmin}})}{2} & 0 & \frac{(x_{\text{vmax}}+x_{\text{vmin}})}{2} \\
0 & \frac{(y_{\text{vmax}}-y_{\text{vmin}})}{2} & \frac{(y_{\text{vmax}}+y_{\text{vmin}})}{2} \\
0 & 0 & 1
\end{bmatrix}
\]