The Two Dimensional Viewing Pipeline

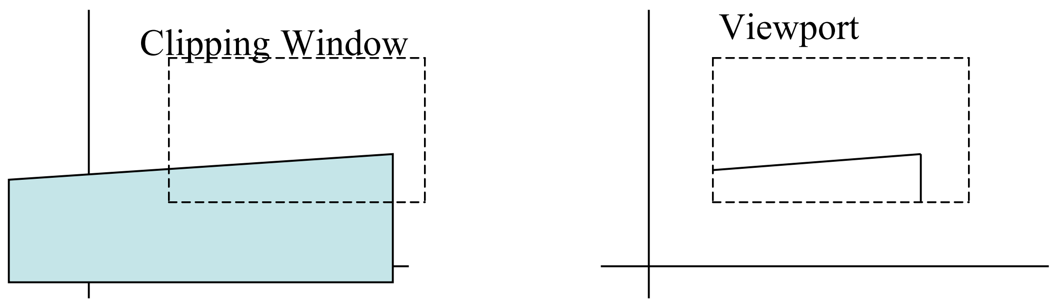
CS116A
Chris Pollett
Nov. 8, 2004.

Outline

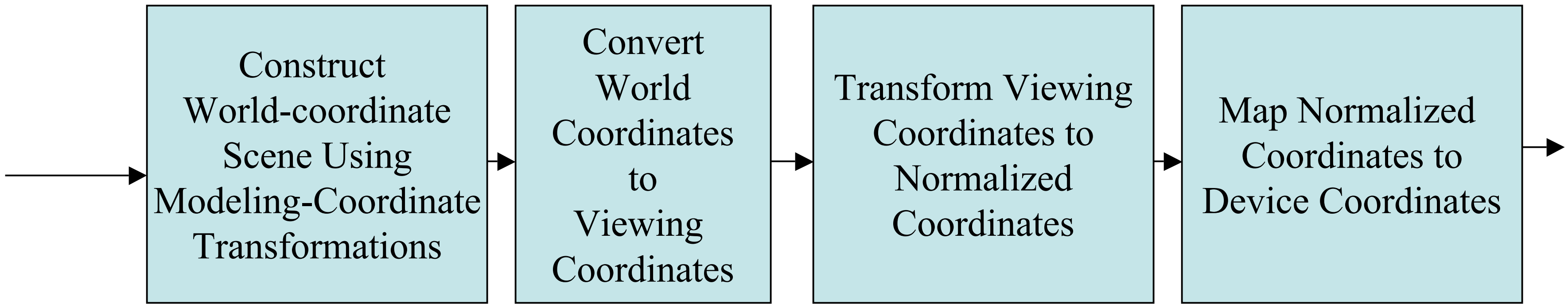
- The Two Dimensional Viewing Pipeline
- The Clipping Window
- Normalization and Viewport Transformations

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



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 $\mathbf{P}_0$=(x0,y0) base position, and a vector **V** that defines the y_{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.
- $\mathbf{M}_{WC,VC} = \mathbf{RT}$
- Suppose $\mathbf{P_0}$=(1,2) and **V**=(4,3). Calculate $\mathbf{M}_{WC,VC}$
- 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 (xw, yw) be a point in world coordinates. To map it into normalized viewport need:

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

- Solving gives $xv = s_x xw + t_x$ and $yv = s_y yw + t_y$ where $s_x = (xv_{max} - xv_{min})/(xw_{max} - xw_{min})$, $t_x = (xw_{max} - xv_{min} - xw_{min} xv_{max})/(xw_{max} - xw_{min})$. s_y and t_y are similar but with y's instead of x's everywhere.

More Mapping

- As a matrix, this gives $\mathbf{M}_{\text{window, normviewp}}$:

$$\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

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:

$$\mathbf{M}_{window,normsquare} = \begin{bmatrix} 2/(xw_{max}-xw_{min}) & 0 & -(xw_{max}+xw_{min})/(xw_{max}-xw_{min}) \\ 0 & 2/(yw_{max}-yw_{min}) & -(yw_{max}+yw_{min})/(yw_{max}-yw_{min}) \\ 0 & 0 & 1 \end{bmatrix}$$

- After clipping, to map to viewport one applies:

$$\mathbf{M}_{normsquare,viewsquare} = \begin{bmatrix} (xv_{max}-xv_{min})/2 & 0 & (xv_{max}+xv_{min})/2 \\ 0 & (yv_{max}-yv_{min})/2 & (yv_{max}+ yv_{min})/2 \\ 0 & 0 & 1 \end{bmatrix}$$