



DCGI

DEPARTMENT OF COMPUTER GRAPHICS AND INTERACTION

APG Homework Assignment II

Jakub Hendrich, Daniel Meister

Outline

- Filling
- Z-buffer
- Perspective projection



Filling

- Scanline filling
- Area mode
 - SGL_POINT, SGL_LINE, SGL_FILL
- Triangle, Polygon
 - glBegin, glEnd
 - (SGL_TRIANGLES,) SGL_POLYGON
- Circle, Ellipse, Arc
 - glBegin, glEnd, glArc



Z-Buffer

■ Context scope

- Depth value for each pixel
- Similar to the color buffer

■ Clear

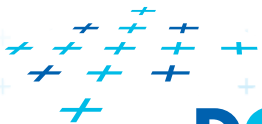
- `sglClear` sets depth of each pixel to the maximum value
 - with `SGL_DEPTH_BUFFER_BIT` set

■ Enable/Disable

- `sglEnable`, `sglDisable`
 - `SGL_DEPTH_TEST`

■ Viewport Transformation

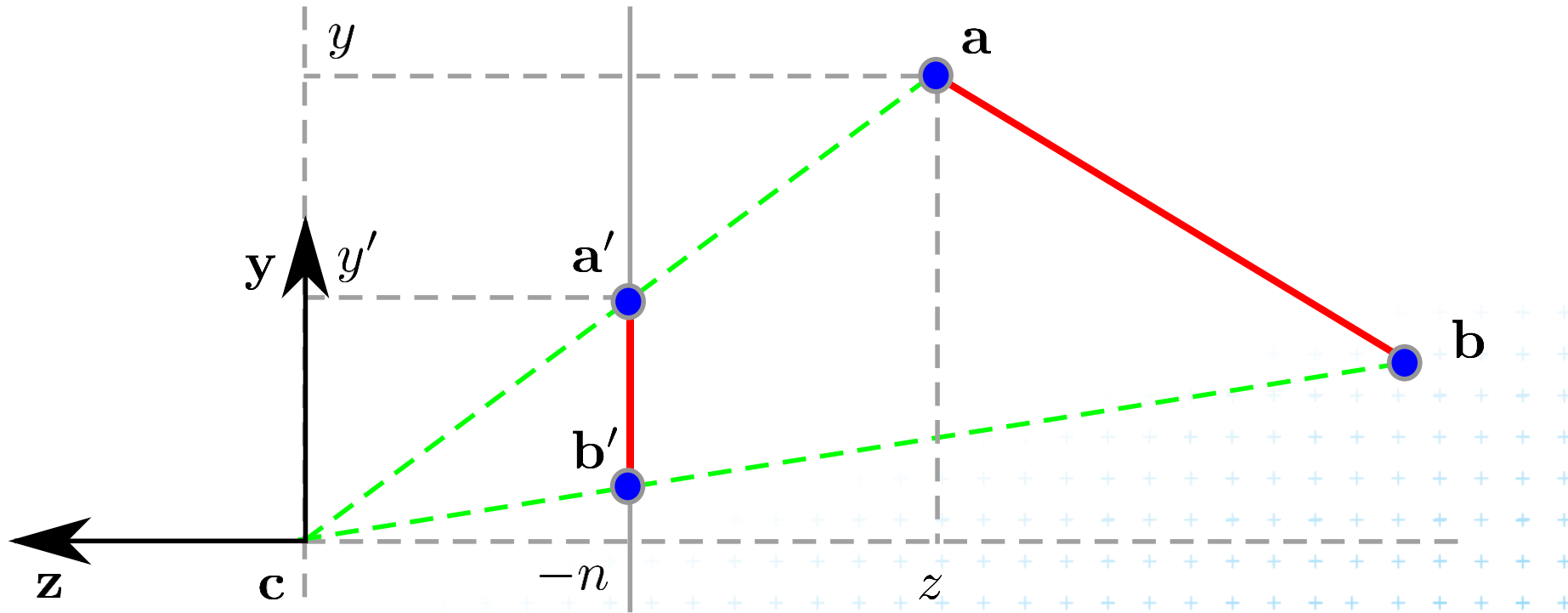
$$\mathbf{V} = \begin{bmatrix} \frac{w}{2} & 0 & 0 & x + \frac{w}{2} \\ 0 & \frac{h}{2} & 0 & y + \frac{h}{2} \\ 0 & 0 & \frac{d}{2} & z + \frac{d}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Perspective Projection

$$\frac{x'}{-n} = \frac{x}{z} \Rightarrow x' = -\frac{nx}{z}$$

$$\frac{y'}{-n} = \frac{y}{z} \Rightarrow y' = -\frac{ny}{z}$$



Perspective Projection

■ Depth interpolation

- Bilinear interpolation in the screen space!

■ Transformation Stack

- `sglFrustum` multiplies the current matrix by perspective matrix $\mathbf{P}$

■ Color interpolation (1 bonus point)

- Color specified in vertices

$$\mathbf{P} = \mathbf{P}_{\parallel} \mathbf{P}_{\angle} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & f+n & fn \\ 0 & 0 & -1 & 0 \end{bmatrix}$$



Thank you for your attention!

Jakub Hendrich

20.10.2025

