Projection and Rasterization Redux

Dinesh K. Pai

Partly from
Textbook Chapters 10 and 12

1

Midterm 2 update

- Textbook. Read ALL of these, except as noted
  - Ch 14 Materials (shading and lighting)
  - Ch 15 Texture Mapping
  - Ch 3.6 (transformation of normals)
  - Ch 9 Interpolation. Skip 9.2 and 9.3
  - Ch 10 Projection
  - Ch 12 From Vertex to Pixel
  - ~~Ch 11:~~ We'll cover this AFTER midterm, so Wed. will be review

2

C^3 Review: Interpolation

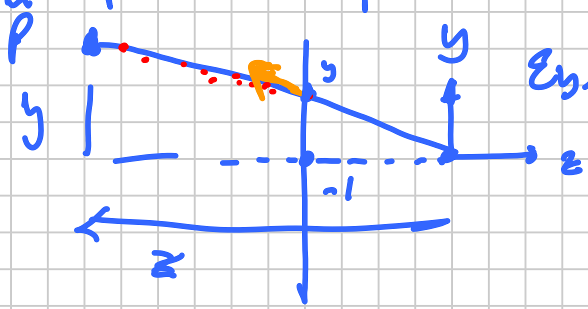
- How many control points are there for a segment of a Bezier curve of degree 3?
  a) 1
  b) 2
  c) 3
  d) 4
  e) None of the above

C^3 Review: Interpolation

- If you use 4 points C0 = (0,0,0), C1 = (1,0,0), C2 = (0,1,0), C3 = (0,0,1) as the control points for a piece of Bezier curve, what is its tangent direction at C0?
  a) (1,0,0)
  b) (0,1,0)
  c) (0,0,1)
  d) None of the above

Projection

§ Recap basic pinhole projection



Basic Projection Matrix

$$P_b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Check

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \xrightarrow{P_b} \begin{pmatrix} x \\ y \\ z \\ -z \end{pmatrix} \xrightarrow[\text{"normalize"}]{\text{Homogenize } H} \begin{pmatrix} -x/z \\ -y/z \\ -1 \\ 1 \end{pmatrix}$$

§ Our initial conception of 4x4 matrices & homogeneous coords. was a way to combine points $\begin{pmatrix} \vdots \\ 1 \end{pmatrix}$ and vectors $\begin{pmatrix} \vdots \\ 0 \end{pmatrix}$

Now we are using all 4 coords, esp. w $\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$

[Note: book often writes $\begin{pmatrix} xw \\ yw \\ zw \\ w \end{pmatrix}$

What we have really done is to model 3D ($\mathbb{R}^3$) as a **Projective** space ($\mathbb{P}^3$)

A projective transform is any non-singular 4×4 matrix.

Let's make a better projection

$$P_p = \left[\begin{array}{ccc|c} 1 & & & 0 \\ & 1 & & 0 \\ & & 0 & 1 \\ \hline & & -1 & 0 \end{array}\right]$$

Non-singular!

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \xrightarrow{P_p} \begin{pmatrix} x \\ y \\ 1 \\ -z \end{pmatrix} \xrightarrow{H} \begin{pmatrix} -x/z \\ -y/z \\ -1/z \\ 1 \end{pmatrix}$$

So e.g.

$$\begin{pmatrix} 0 \\ 1 \\ -1 \\ 1 \end{pmatrix} \dashrightarrow \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

origin

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ \infty \\ 1 \end{pmatrix}$$

became a vector!

Huh? Solution: think of this as a point at infinity

vector

$$\begin{pmatrix} 0 \\ y \\ -1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ y \\ -1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ y \\ 0 \\ 1 \end{pmatrix}$$

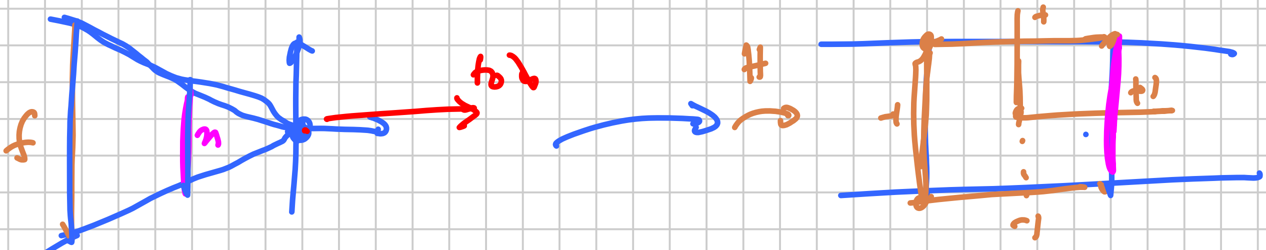
became a point!

In a projective space, $\begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix}$ is a point at infinity in the direction of that ~~f~~ vector.

Simplifies a lot of things.

eg. any two lines intersect at a point, perhaps at the

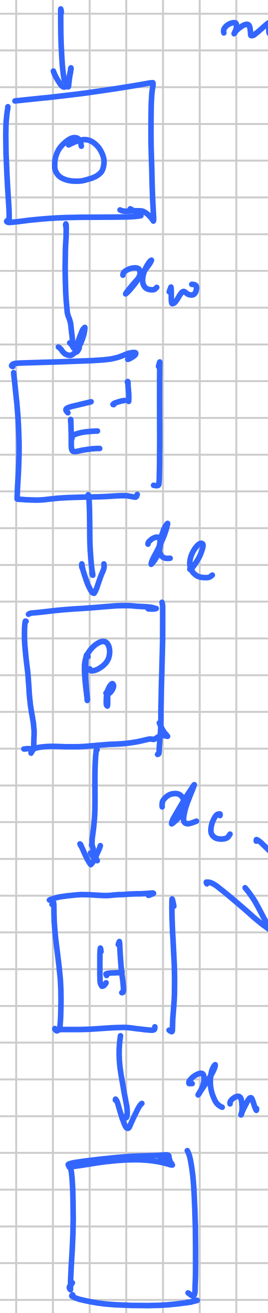
What this does to 3D space:



What glm::Perspective( ) does.

Normalized device coordinates

Pipeline



model coord

Model

x_w world coords

View

x_e eye coords

Projection

x_c Clip coords $\begin{pmatrix} x_n w_n \\ y_n w_n \\ z_n w_n \\ w_n \end{pmatrix}$

w-component is non-trivial

x_n clipping $-1 \leq x_n \leq 1$

Instead we clip

$-w_c \leq x_c \leq w_c$