Vertex to Pixel

A brief introduction

Textbook Chapter 12

(some slides courtesy of Min Kim)

1

Today

- Announcements
  - Assignment 2 progress?
    Deadline extended to midnight Sunday!
    (but note that TAs and Instructor not available on weekends)
  - Signup sheet for grading is available
  - Quiz 2 is next week! Wednesday, in class
  - Final exam scheduled: DEC 19 2017 07:00 PM
    (sorry… I don't control the scheduling)
- Lecture
  - Quiz 2 prep
  - Wrap up projection and depth
  - Vertex to Pixel (Ch. 12)

2

Quiz 2 Preparation

- In class, Wednesday Oct 18. Please be on time.
- Review lecture notes, and assignments.
- Everything covered in lecture could be on the exam
- Everything covered in listed textbook chapters could be on the exam

3

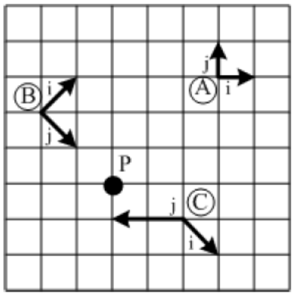
Quiz 2 Preparation

- Textbook. Read ALL of these except as noted
  - Ch 4 Respect
  - Ch 5, except 5.4 Hierarchies of transformations
  - Ch 10 Projection. Mainly focus on lecture notes
  - Ch 11 Depth. Mainly focus on lecture notes
  - Ch 12 From Vertex to Pixel
- Topics from Quiz 1 (especially Ch 3) will be assumed as pre-requisites (e.g., it is assumed you now know coordinate frames and how to transform them)

4

Quiz 2 Preparation Tips

- Format will be similar to Quiz 1
- Expect 2D problems. E.g., will ignore x coordinate and have just y and z basis vectors. Homogeneous coordinates will be 2+1 = 3 dimensions, etc.



E.g., What are the coordinates of point P in frame A, B, and C?
E.g., Define one coordinate frame with respect to another

5

★ Rotations

Quiz 2 Preparation Tips

- Section 5.2 is very important, since it uses transformations in the most common ways in computer graphics, e.g., different versions of doMtoOwrtA (see p. 46 of book). Make sure you understand this section.

Also for Assignment 2

- Know inverses of simple transformations, e.g.,
  - Translation by t
  - rotation about an axis by theta
  - scale by s

$\begin{pmatrix} \mathbf{\cdot} & t \\ 0 & 1 \end{pmatrix}^{-1} \rightarrow \begin{pmatrix} \cdot & -t \\ & 1 \end{pmatrix}$

$Rot(\hat{n}, \theta) \rightarrow Rot(\hat{n}, -\theta)$

$\begin{pmatrix} s_1 & & \\ & s_2 & \\ & & 1 \end{pmatrix}^{-1} \rightarrow \begin{pmatrix} 1/s_1 & & \\ & 1/s_2 & \\ & & 1 \end{pmatrix}$

- Review basic Three.js and GLSL functions you used in assignments

6

Wrap up Projection & Depth

October 13, 2017 10:19 AM

See previous class

Projective Transformation preserve

- Straightness of lines
- Intersections
- Order of points along $\vec{z}$

Put L13 & L14 together to produce P

$$P = P_o \cdot P_n$$

Almost done: But after P_n, top, bottom, near, far, left, right change.

top $= \tan\frac{\theta}{2}$ $\quad n \longrightarrow -\frac{1}{n}$

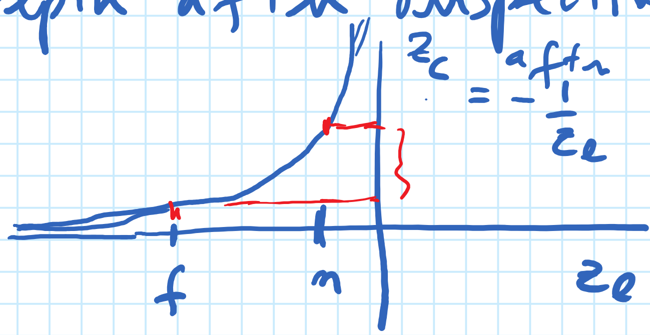
use those quantities in P_o

Try this at home.

I'll give you a handout to check.

§ Depth after Perspective Projection (again)

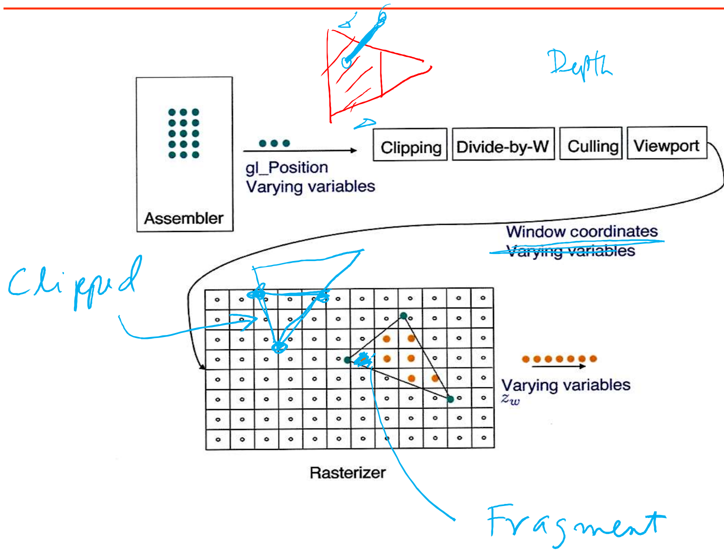
z_c after Projection $= -\frac{1}{z_e}$



Note: (1) Monotonic, so order is preserved

(2) Lower resolution as depth increases

Path from vertex to pixel



8

Rasterization

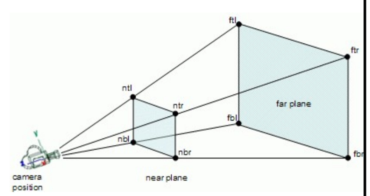
- This is part of the fixed function pipeline
- There are very clever and sophisticated algorithms underneath the hood, but most users just set a few knobs using OpenGL function calls
- We will speed through these issues for now, with the goal of getting to the fun topic of lighting asap!
- We may return to some of these issues at the end of the course, if we have time

9

Clipping coordinates

- Eye coordinates (projected) → clip coordinates → normalized device coordinates (NDCs)
- Dividing clip coordinates (x_c, y_c, z_c, w_c) by the $w_c (w_c = w_n)$ component (the fourth component in the homogeneous coordinates) yields normalized device coordinates (NDCs).

$$\begin{bmatrix} x_n w_n \\ y_n w_n \\ z_n w_n \\ w_n \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} s_x & 0 & -c_x & 0 \\ 0 & s_y & -c_y & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$$



10

Viewport matrix

- We need a transform that maps the lower left corner to $[-0.5, -0.5]^t$ and upper right corner to $[W-0.5, H-0.5]^t$
- The appropriate scale and shift can be done using the viewport matrix:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} W/2 & 0 & 0 & (W-1)/2 \\ 0 & H/2 & 0 & (H-1)/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$$

11