Vertex to Pixel

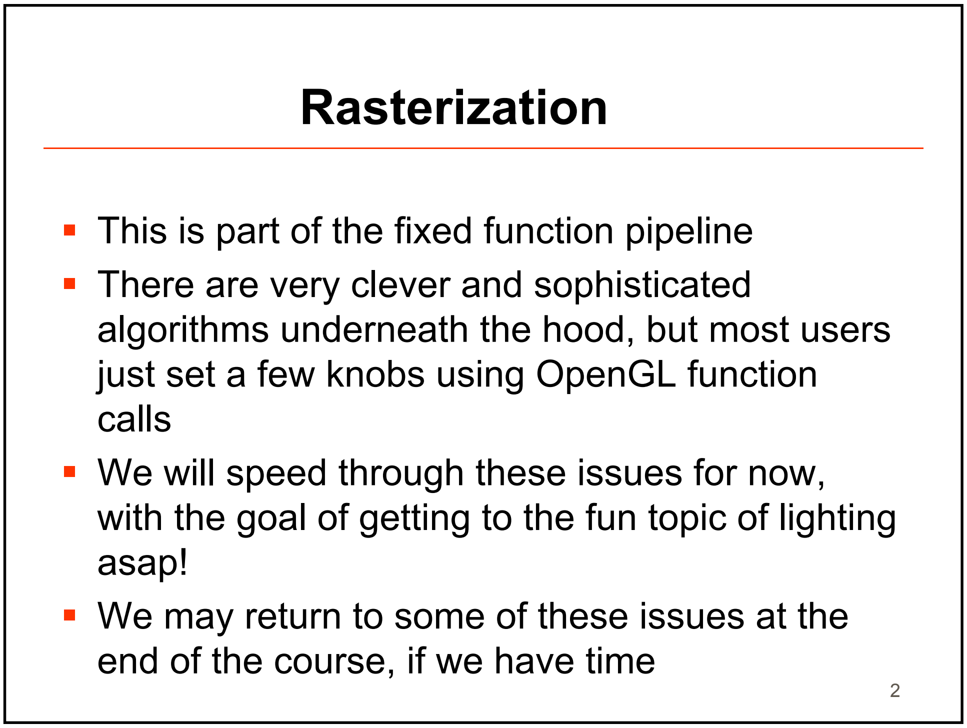
A brief introduction

Textbook Chapter 12

Guest Lecture by Prof. van de Panne

(some slides courtesy of Min Kim)

1

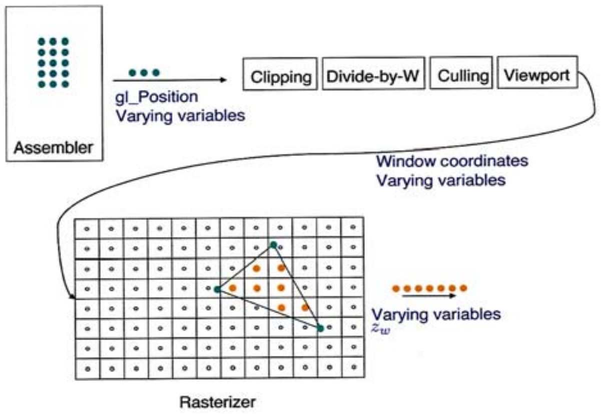


Rasterization

- 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

2

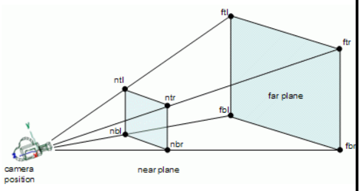
Path from vertex to pixel



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).

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

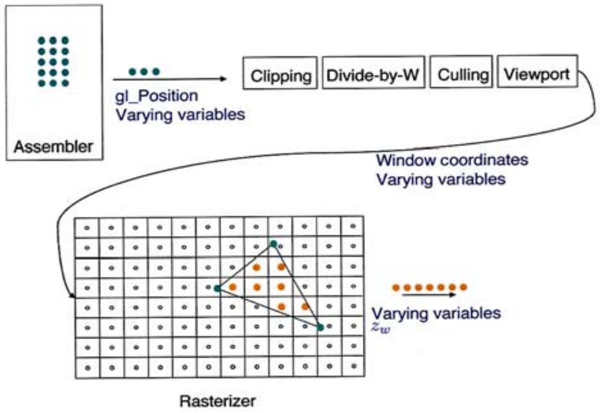


Viewport matrix

- 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:

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

Path from vertex to pixel

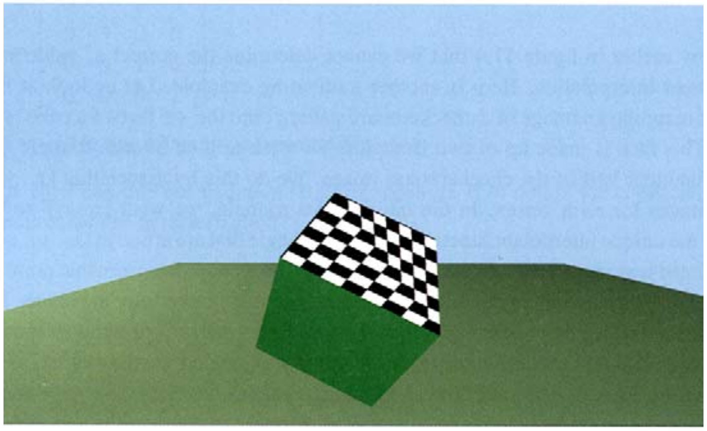


Interpolation of varying variables

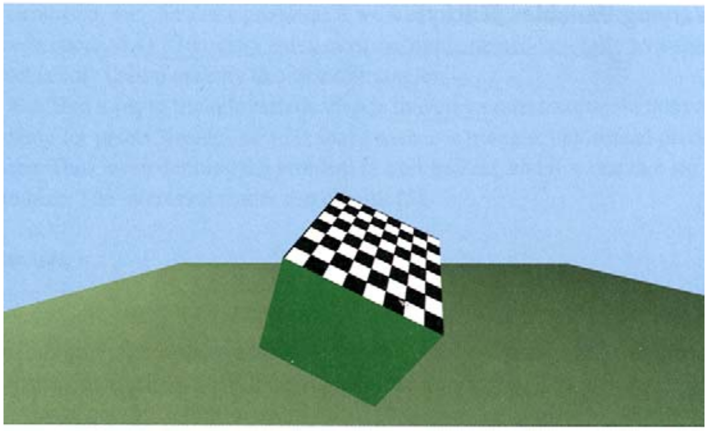
- Topic of Chapter 13. Optional for this course, but please remember that there is a subtle issue.
- In between the vertex and fragment shader, we need to interpolate the values of the varying variables.
- This is surprisingly subtle (called “perspective correct interpolation”).

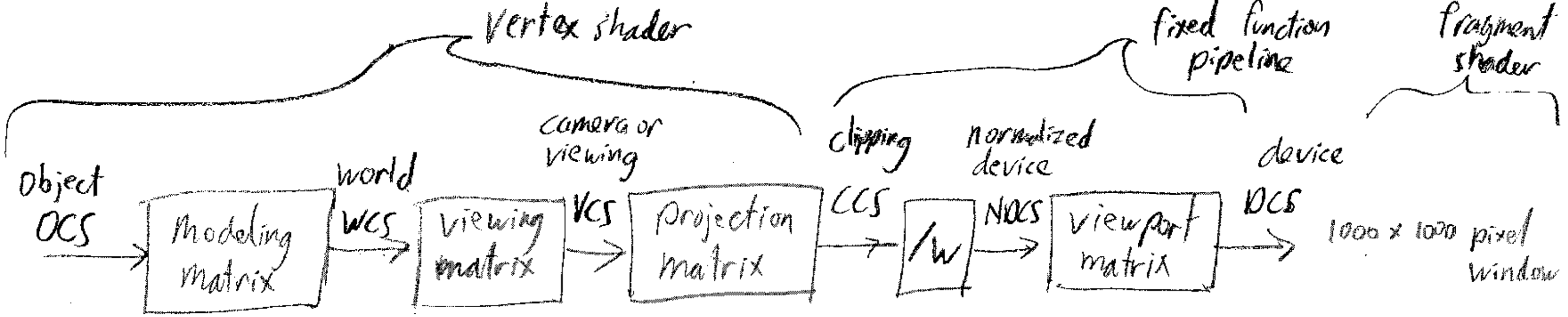
Wrong representation of texture

When texture coordinates are linearly interpolated in window coordinates, an incorrect image results.



Correct representation of texture

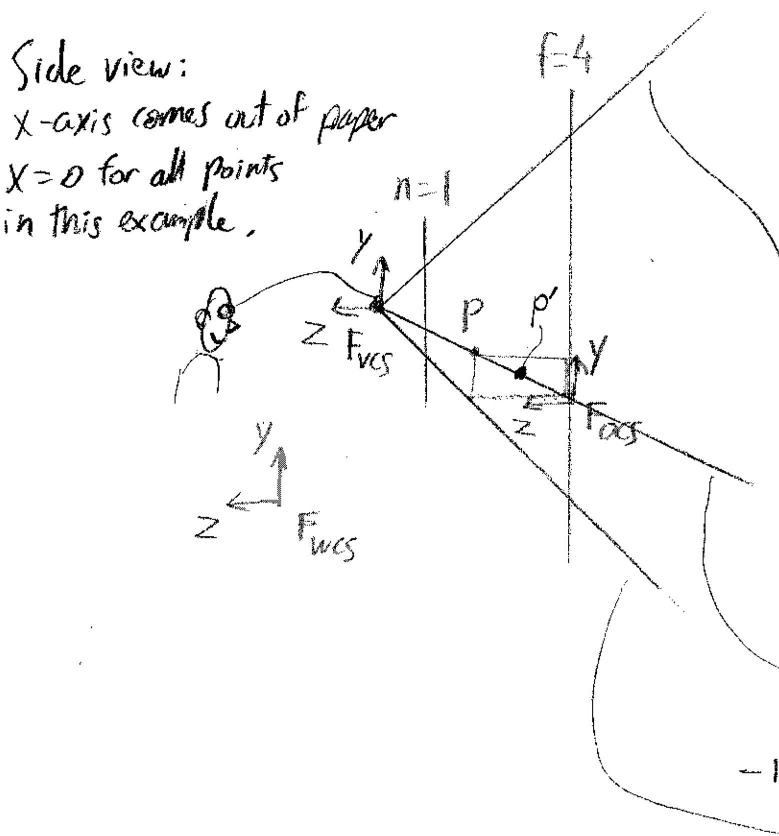


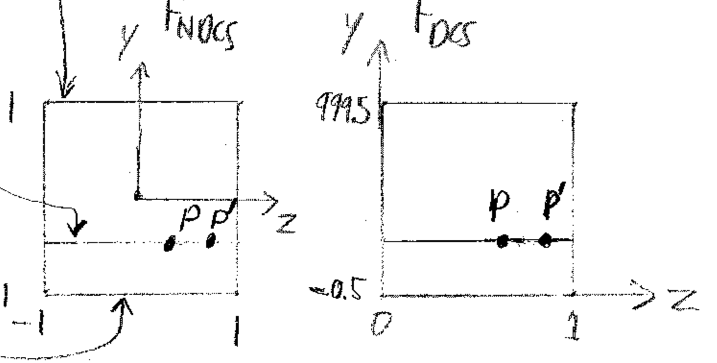


$$\begin{bmatrix}1&&&0\\&1&&2\\&&1&-6\\&&&1\end{bmatrix} \qquad \begin{bmatrix}1&&&0\\&1&&-4\\&&1&2\\&&&1\end{bmatrix} \qquad \begin{bmatrix}1&&0&\\&1&0&\\&&-\frac{5}{3}&-\frac{8}{3}\\&&-1&\end{bmatrix} \qquad \begin{bmatrix}500&&&499.5\\&500&&499.5\\&&0.5&0.5\\&&&1\end{bmatrix}$$

$$T_M \qquad T_V \qquad T_P \qquad T_{VP}$$

$$\begin{array}{ccccccc} & P_{OCS} & P_{WCS} & P_{VCS} & P_{CCS} & P_{NDCS} & P_{DCS}\\ \begin{bmatrix}x\\y\\z\\w\end{bmatrix} & \begin{bmatrix}0\\1\\2\\1\end{bmatrix} & \begin{bmatrix}0\\3\\-4\\1\end{bmatrix} & \begin{bmatrix}0\\-1\\-2\\1\end{bmatrix} & \begin{bmatrix}0\\-1\\2/3\\2\end{bmatrix} & \begin{bmatrix}0\\-0.5\\1/3\\1\end{bmatrix} & \begin{bmatrix}499.5\\249.5\\0.667\\1\end{bmatrix}\end{array}$$


Side view:
X-axis comes out of paper
X=0 for all points
in this example.


y
F_NDCS
1
-1
-1
1
z
P
P'
y
F_DCS
999.5
-0.5
0
1
z
P
P'