CS 498 VR

Lecture 8 - 2/12/18

go.illinois.edu/VRlect8

Last Time on CS 498

- Why are quaternions useful? What can they do that euler angles can't?
  - Exam question detected.

- Give the matrix that would transform a 3D object by:
  - Rotating ***θ*** degrees around the Z axis
  - Then translating by ***(x, y, z)***.
    - Would the answer change if the steps were reversed?

Homogeneous Transformation Matrices

- Translate by ***t = (3, 4, 5)***
- Then rotate by

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

- … What matrix combines these two transformations?

Homogeneous Transformation Matrices

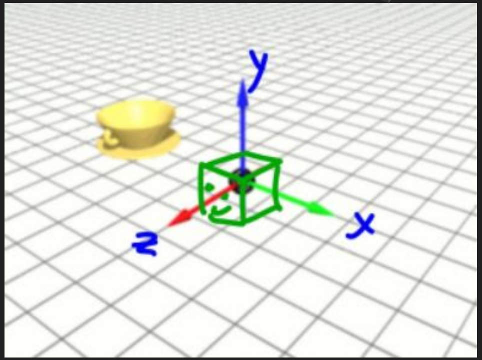
What is the geometric interpretation of the following matrices?

Case #1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case #2

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Homogeneous Transformation Matrices

Case #1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case #2

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow$$

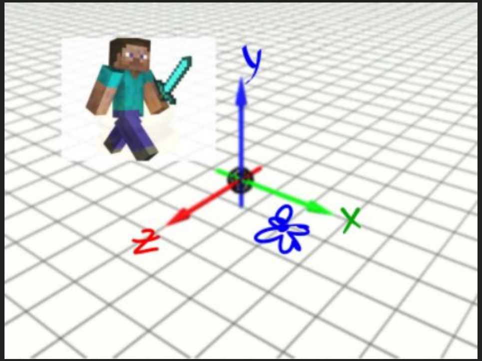
Homogen. matrix for R and t

Homogeneous Transformation Matrix Inverse

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & 0 \\ \cdot & R & \cdot & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cdot & \cdot & \cdot & 0 \\ \cdot & R & \cdot & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Switching Coordinate Frames



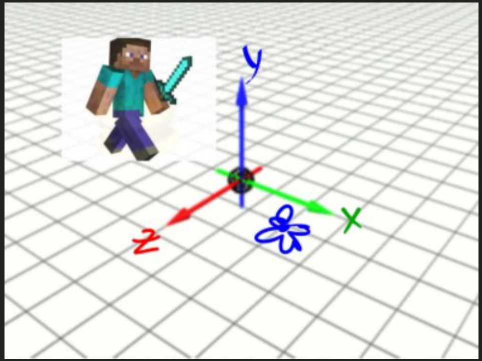
Flower: ***F = (2, 0, 1)***

Pupil: ***P = (1, 0, 3)***

Steve's local coordinate system coincides with the global system. Then Steve rotates by ***pi/2*** yaw and translates by ***(-10, 10, 0)***.

- Write Steve's homogeneous transformation matrix.
- Find the coordinates of the flower from Steve's (LCF) perspective after the transformation

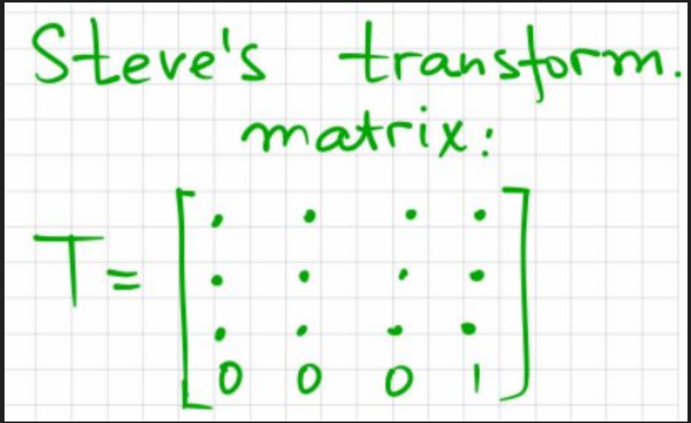
Switching Coordinate Frames



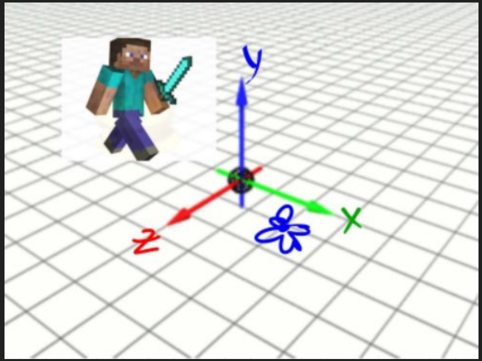
Flower: ***F* = (2, 0, 1)**

Pupil: ***P* = (1, 0, 3)**

Steve's local coordinate system coincides with the global system. Then Steve rotates by ***pi/2*** yaw and translates by ***(-10, 10, 0)***.



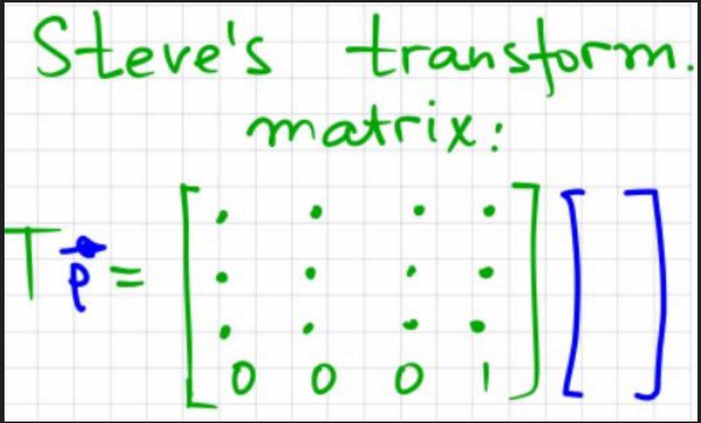
Switching Coordinate Frames

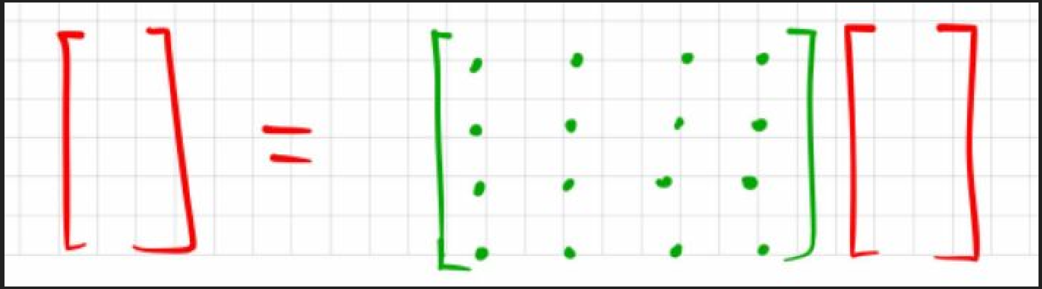


Flower: ***F* = (2, 0, 1)**

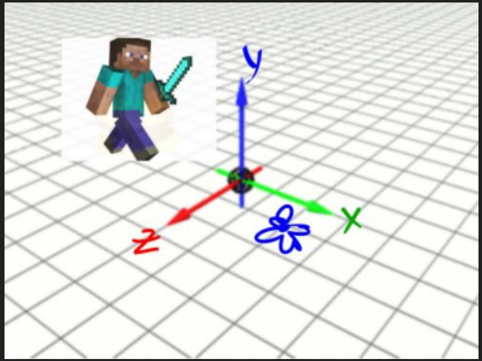
Pupil: ***P* = (1, 0, 3)**

Steve's local coordinate system coincides with the global system. Then Steve rotates by ***pi/2*** yaw and translates by ***(-10, 10, 0)***.





Switching Coordinate Frames



Flower: ***F = (2, 0, 1)***

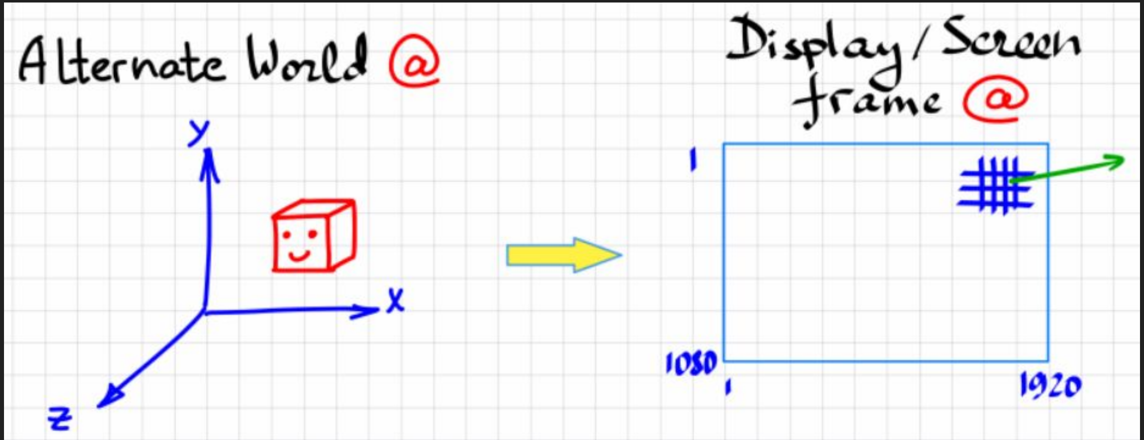
Pupil: ***P = (1, 0, 3)***

Steve's local coordinate system coincides with the global system. Then Steve rotates by ***pi/2*** yaw and translates by ***(-10, 10, 0)***.

The inverse of ***T*** represents the switch of the "view point" from global coordinate frame to local coordinate frame.

Viewing Transformations

From World Coordinate Frame to Pixels on Screen

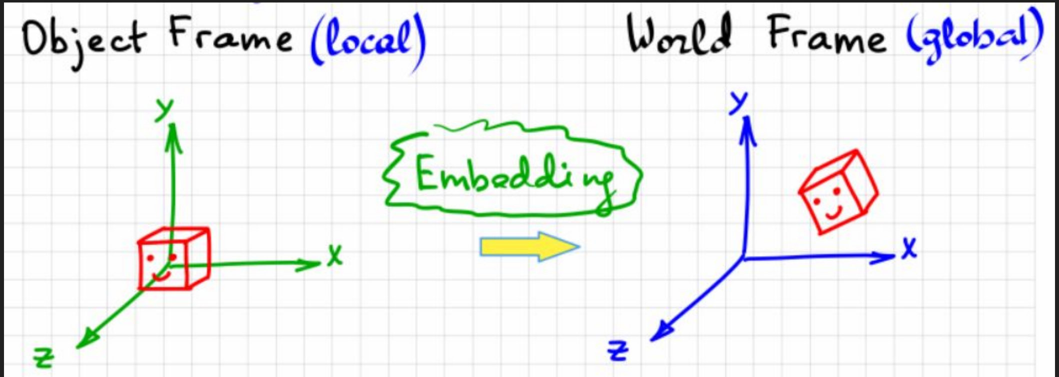


Goal:

Ignore for now:

What is different from previous geometric transformations?

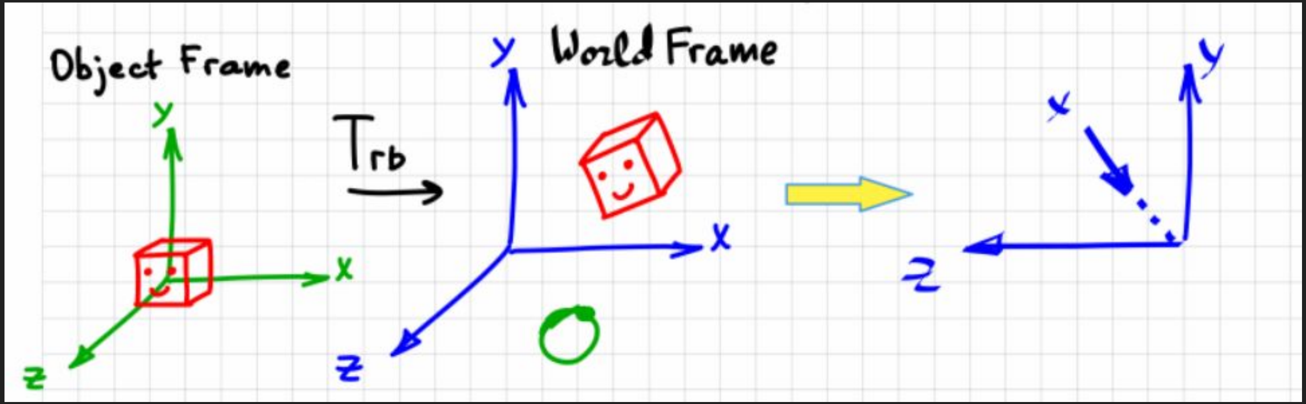
Object Frame to World Frame



The chain of transformations starts with:

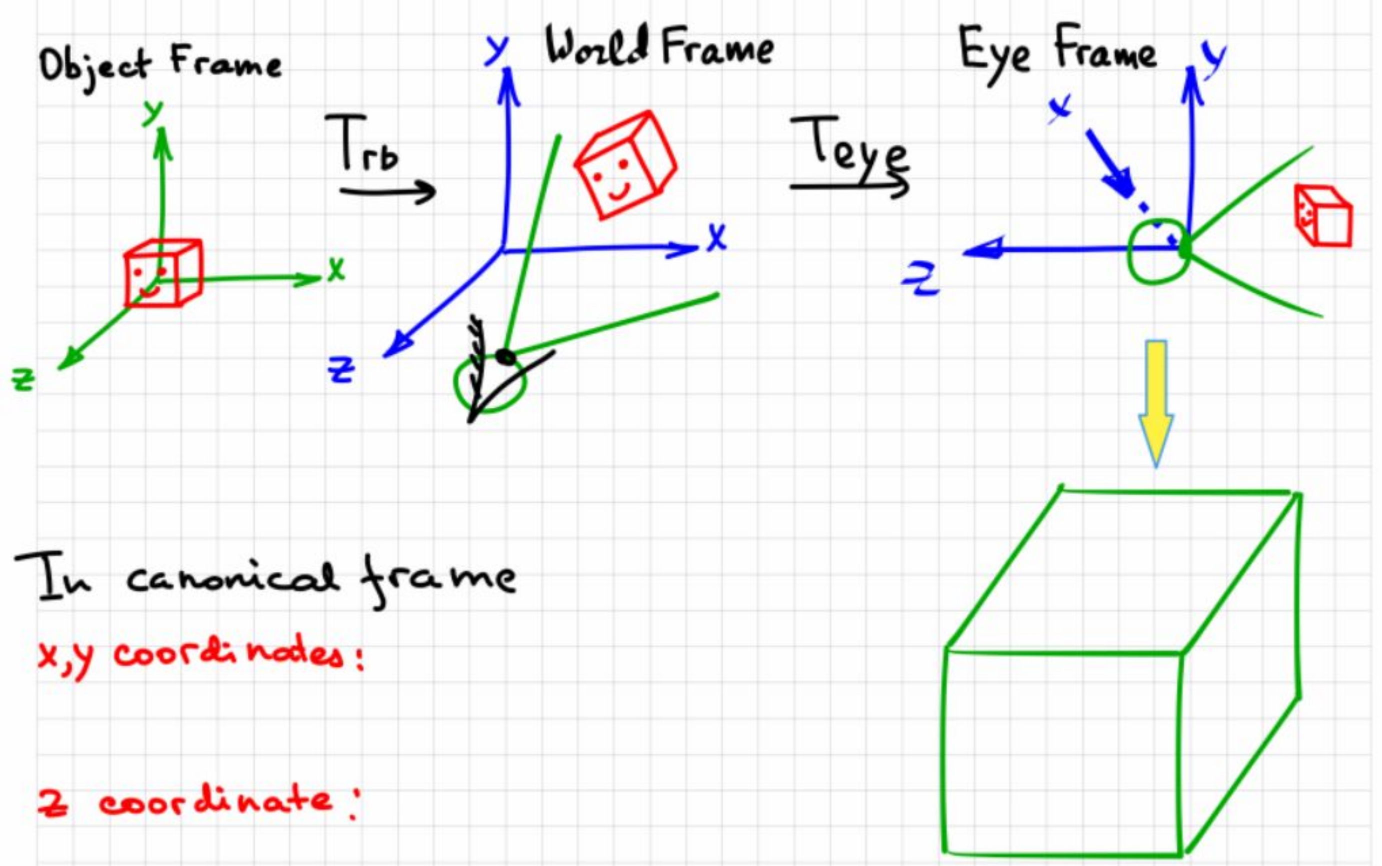
- for moving objects.
- for stationary objects.

World Frame to Eye Frame

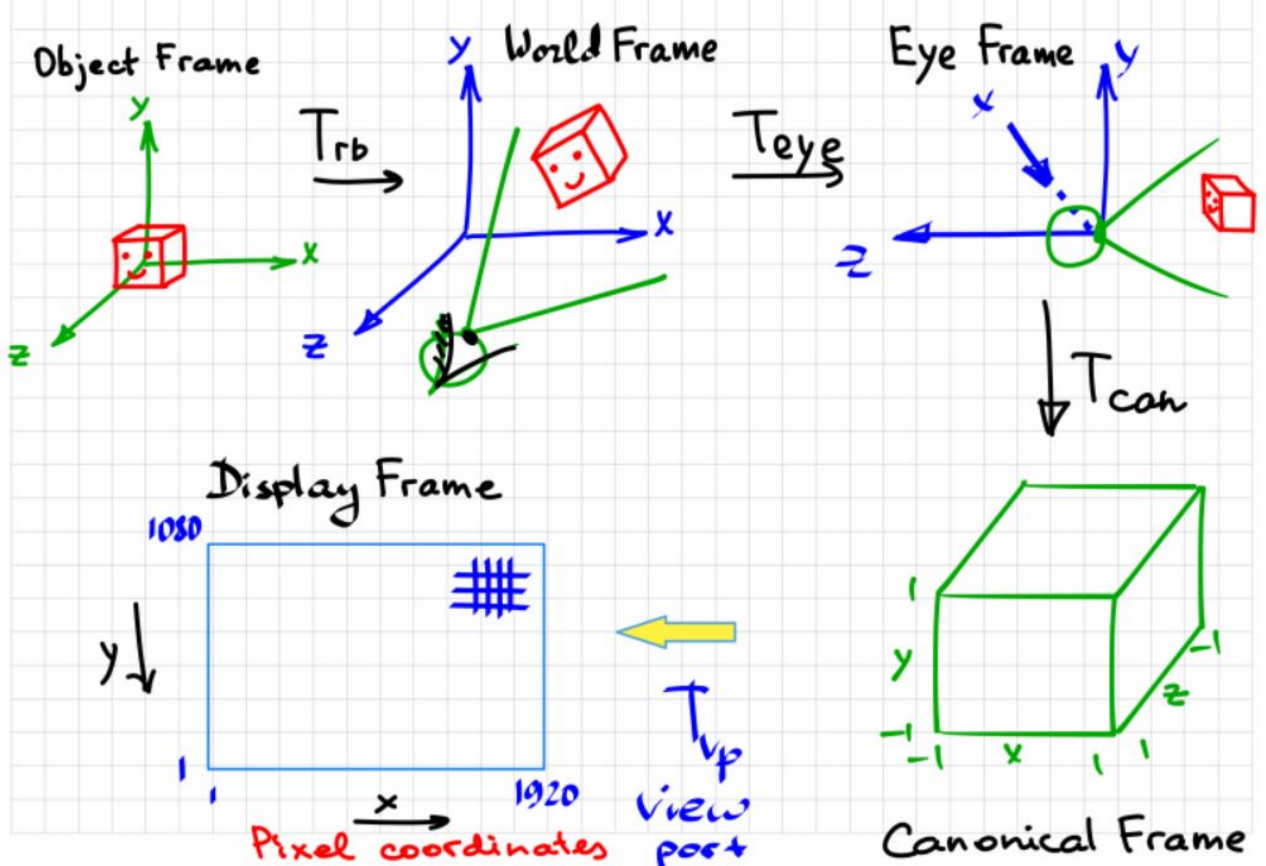


The eye is a rigid body, too.

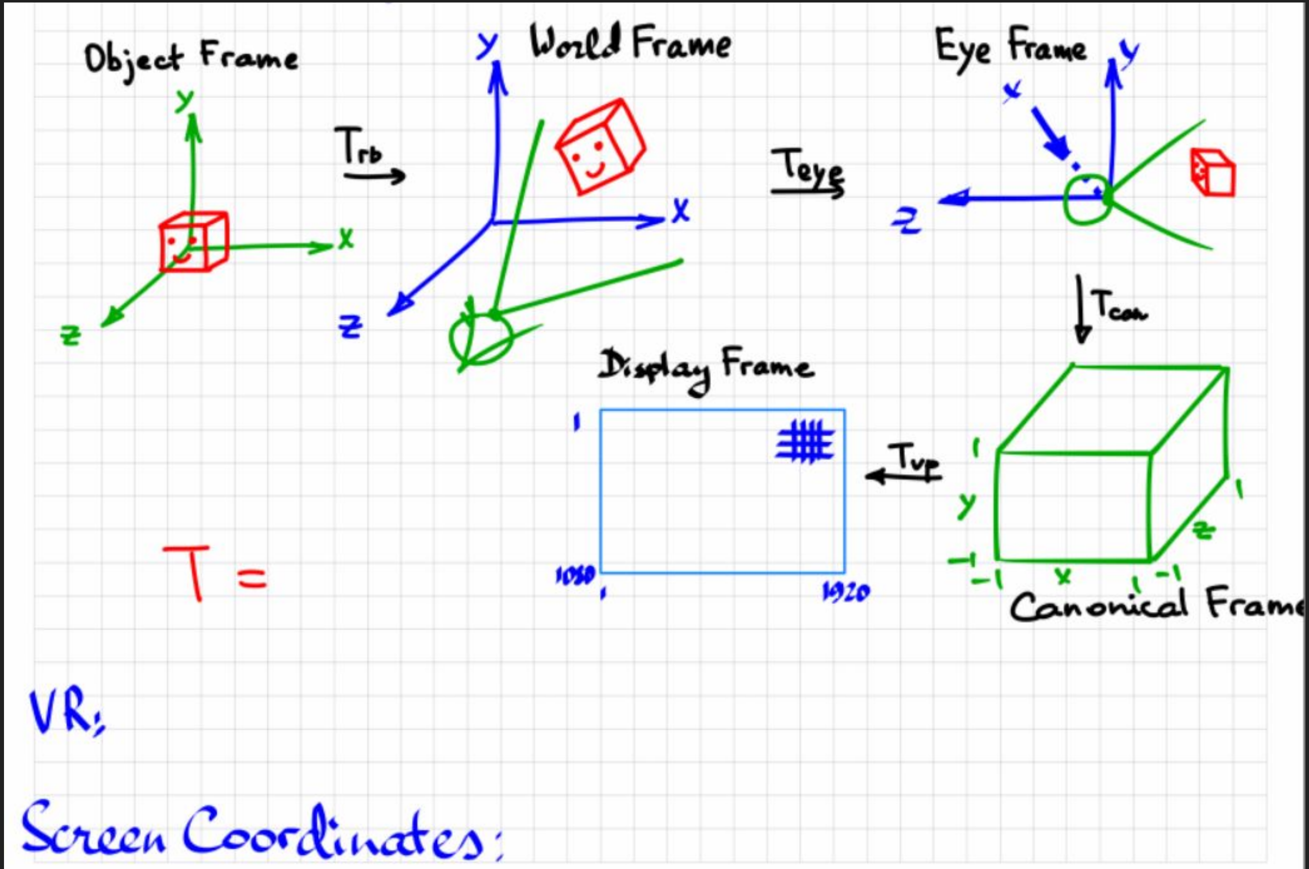
Eye Frame to Canonical Frame



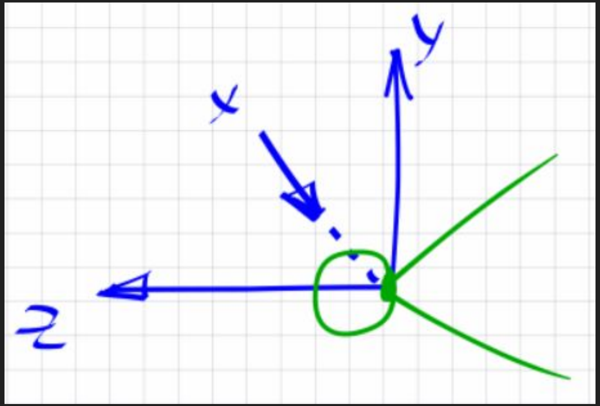
Canonical Frame to Viewport Frame



Algebraic Representation



Algebraic Representation: Cyclopean Eye Transformation



Consider a “look at”:

1.

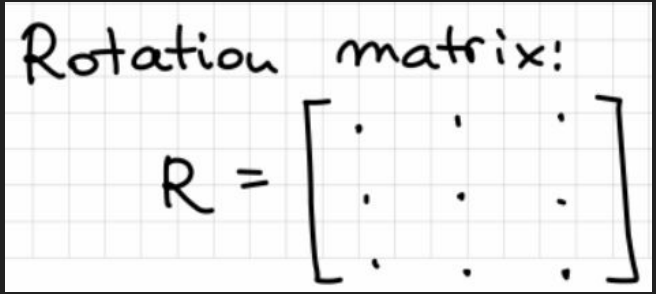
2.

3.

Coordinate axis for the eye in the world:

In Graphics:

In VR:



Algebraic Representation: Cyclopean Eye Transformation

To place the eye in the world:

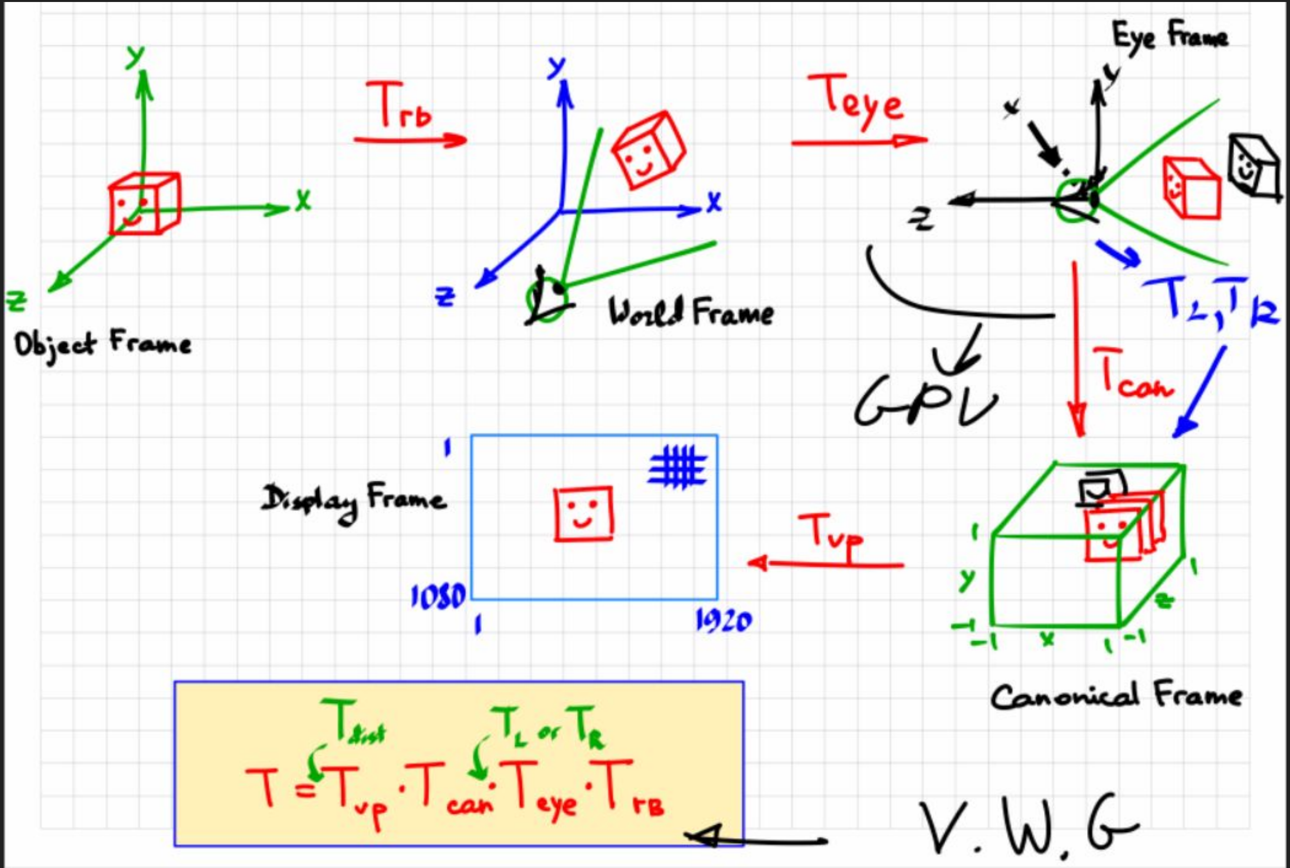
To convert from world frame to eye frame:

Algebraic Representation: Left Eye Transformation

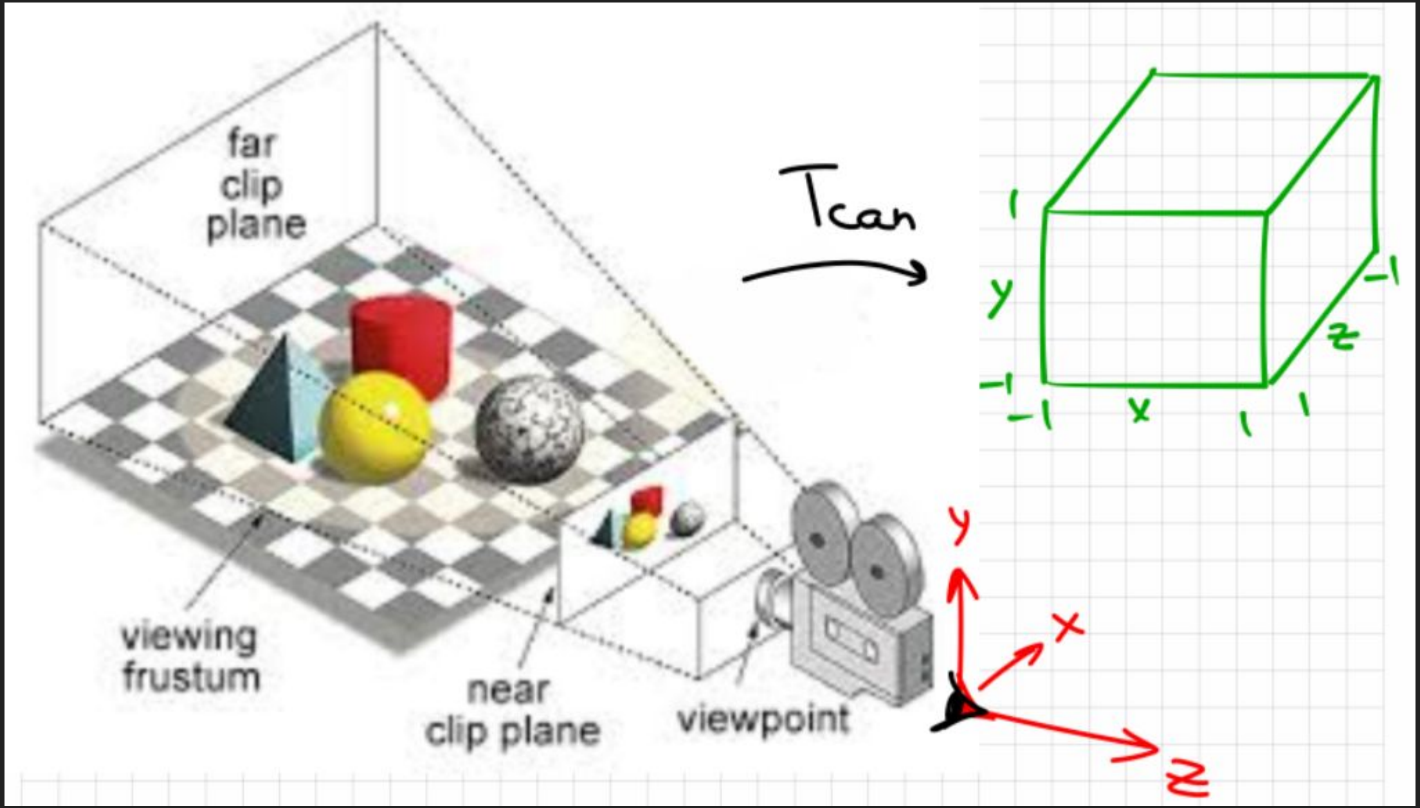
To place the left eye in the world:

To convert from world frame to left eye frame:

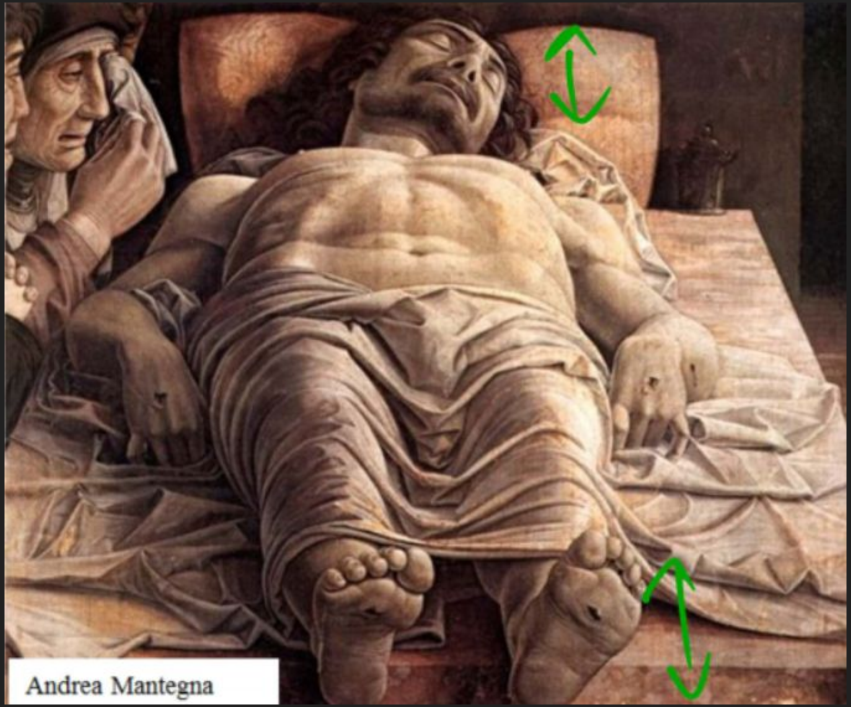
From Alternate World Generator to GPU



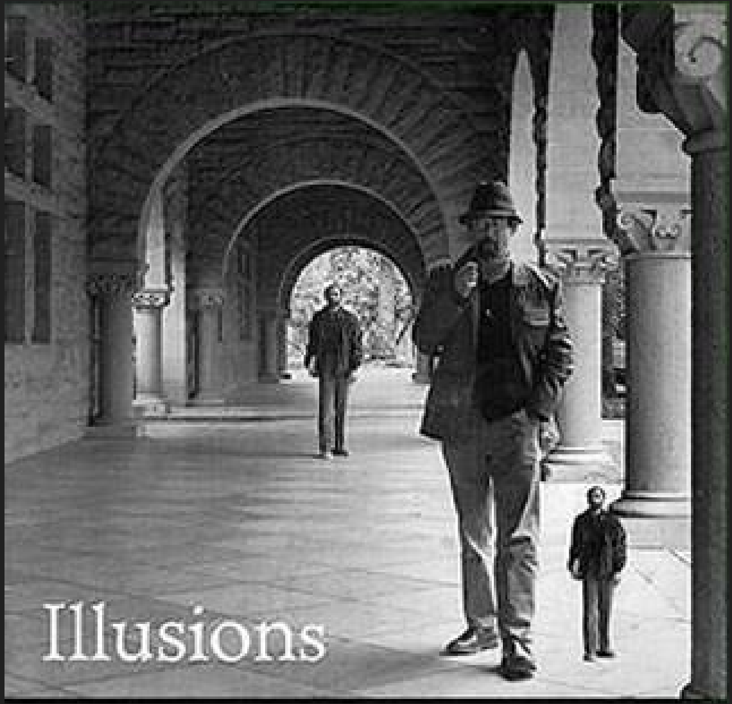
Canonical Transformation



Canonical Transformation

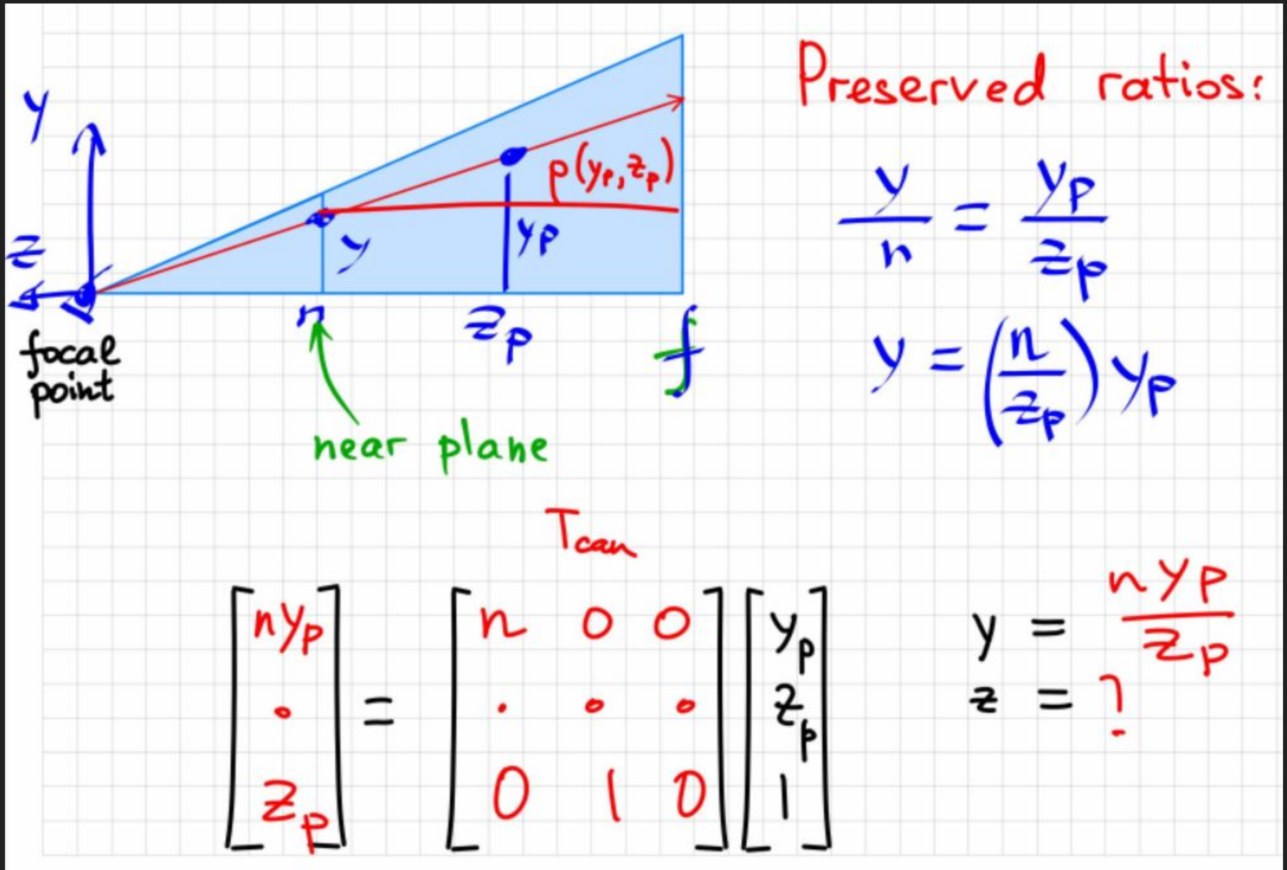


Incorrect perspective

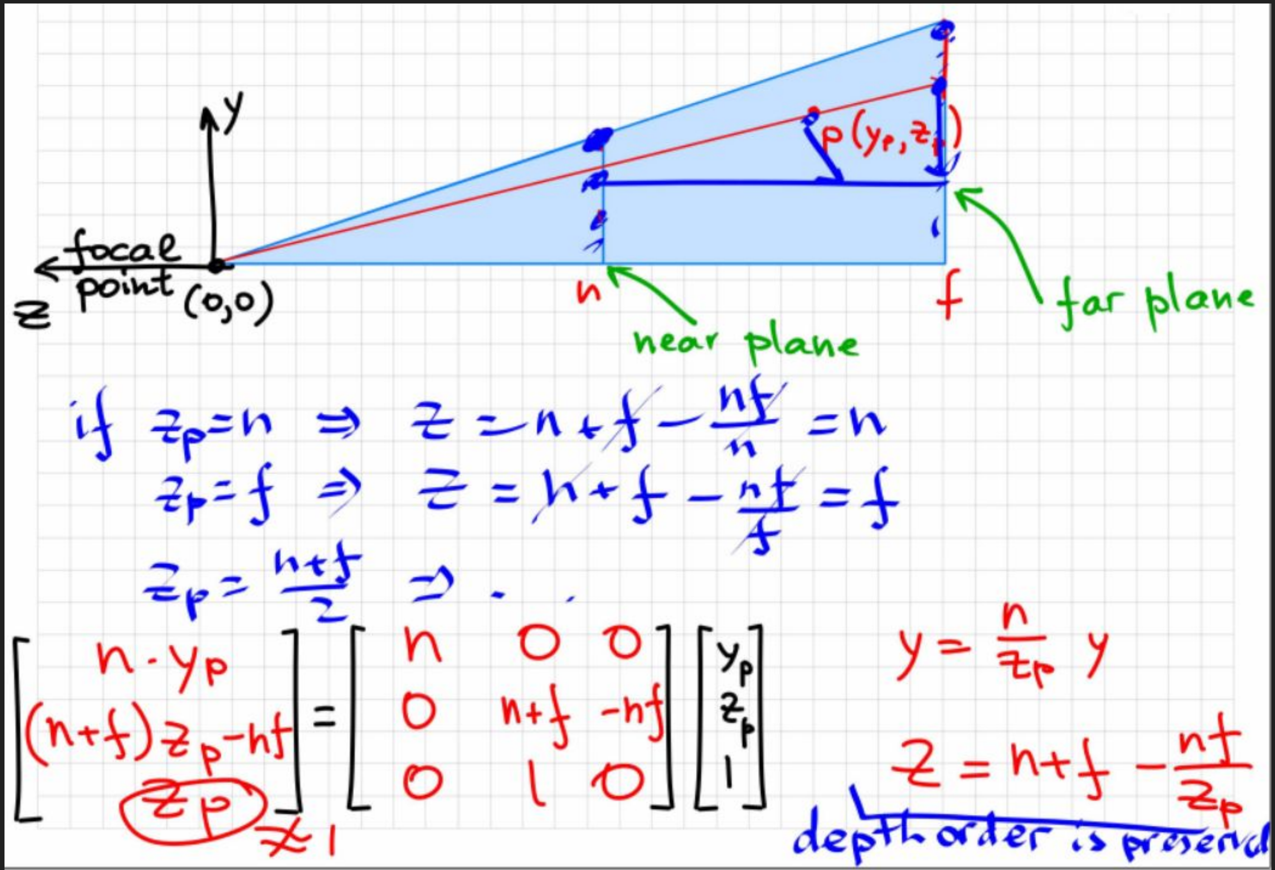


Correct perspective

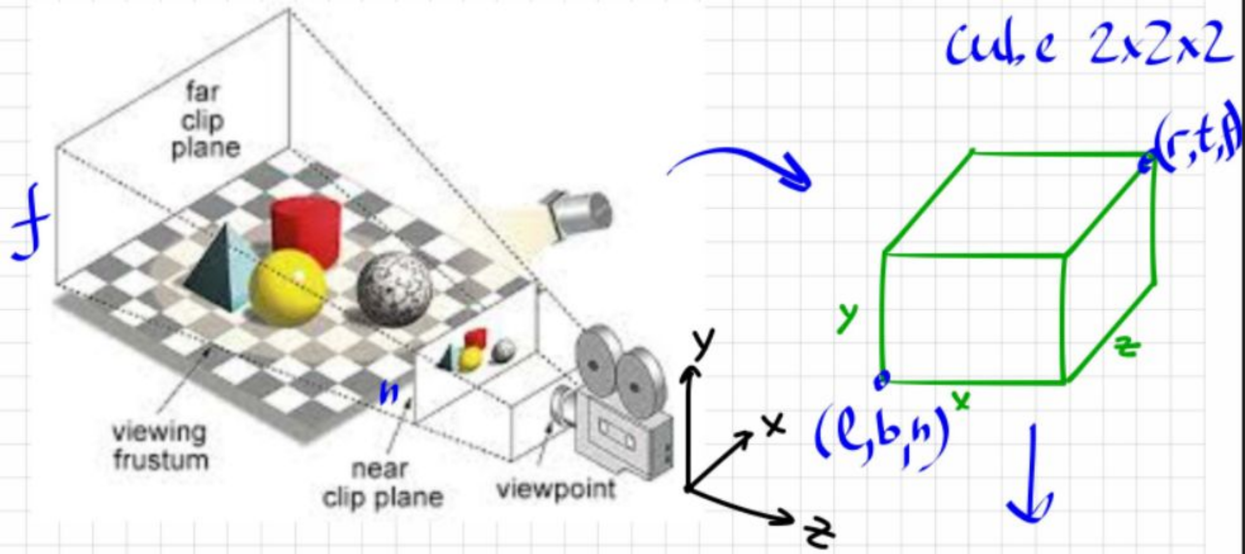
Canonical Transformation: 2D Analogy



Canonical Transformation: 2D Analogy

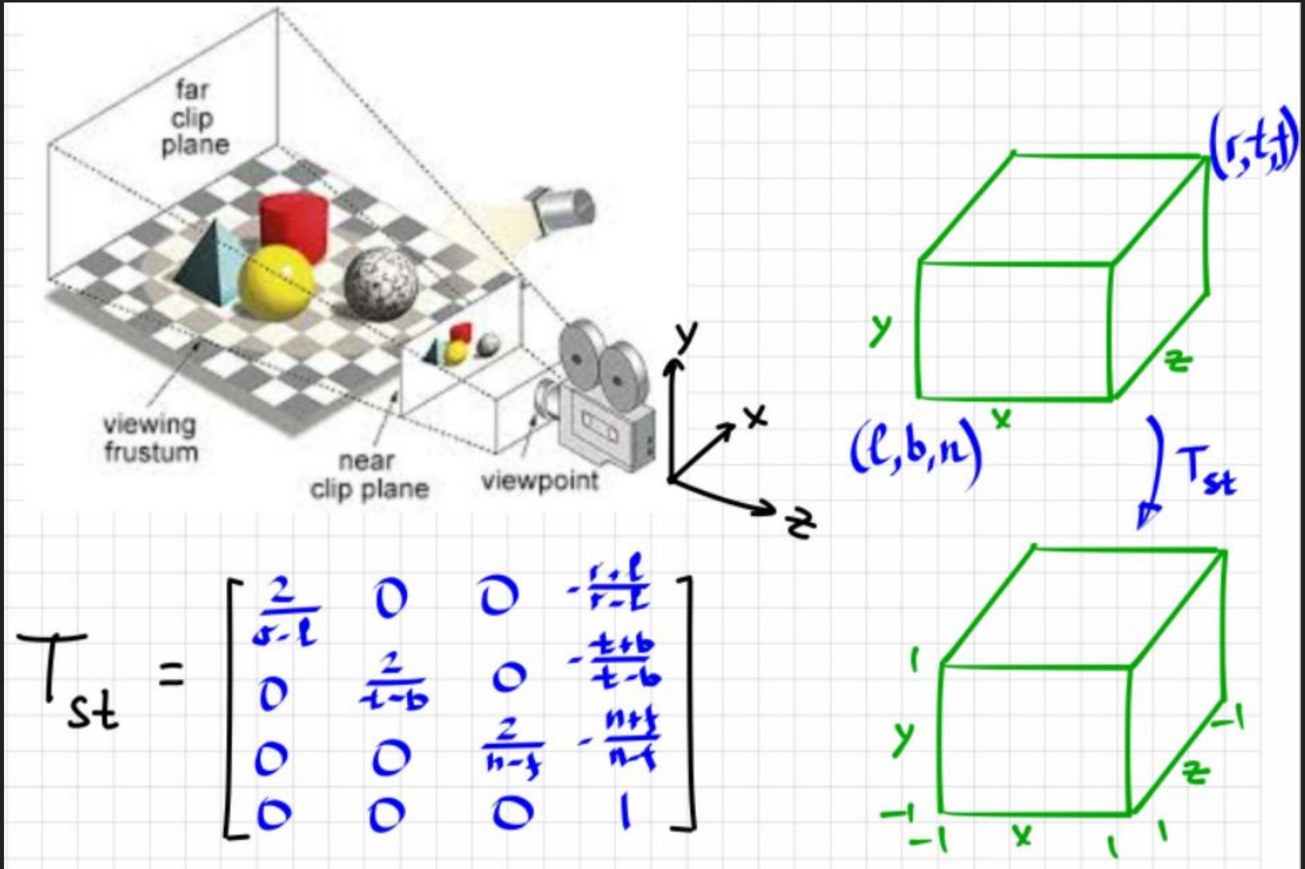


Canonical Transformation



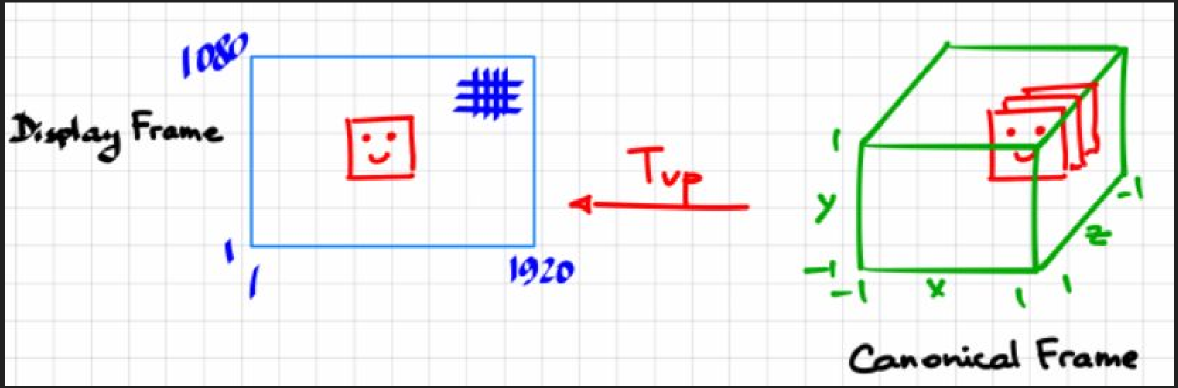
$$T_p = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -nf \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Canonical Transformation



$$T_{st} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Viewport Transformation



T_{VP} converts -1 .. 1 range to pixel coordinates:

n_x = # horizontal pixels

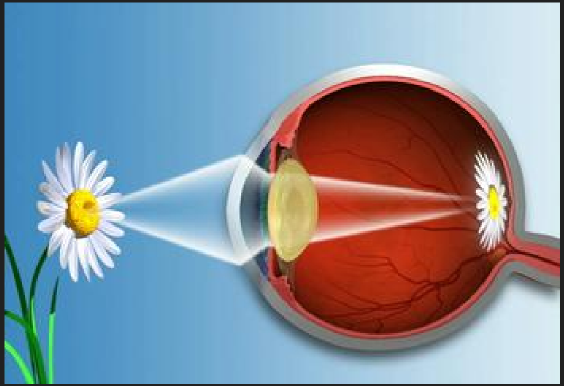
n_y = # vertical pixels

$$T_{vp} = \begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y - 1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next Time on CS 498: Light and Optical Systems

Alternate World Generator: proper lighting and shadows.

Lens: proper correction for the lens distortion.



Review

- How could a matrix that is the product of several homogeneous transformations be inverted?
  - What would the inverted matrix “mean”?

- Does Unity store global object coordinates or local object coordinates?
  - Why is Unity’s choice the “natural” choice? Does it make your life easier?

Announcements

- MP 2.1 & Team Formation was due!
  - But you already did them so it's fine.

- MP 2.2-2.4 is due **next Monday (02/19).**

- Read Ch. 3.4 & 3.5