Viewing Transformations

CS425: Computer Graphics I

Khairi Reda

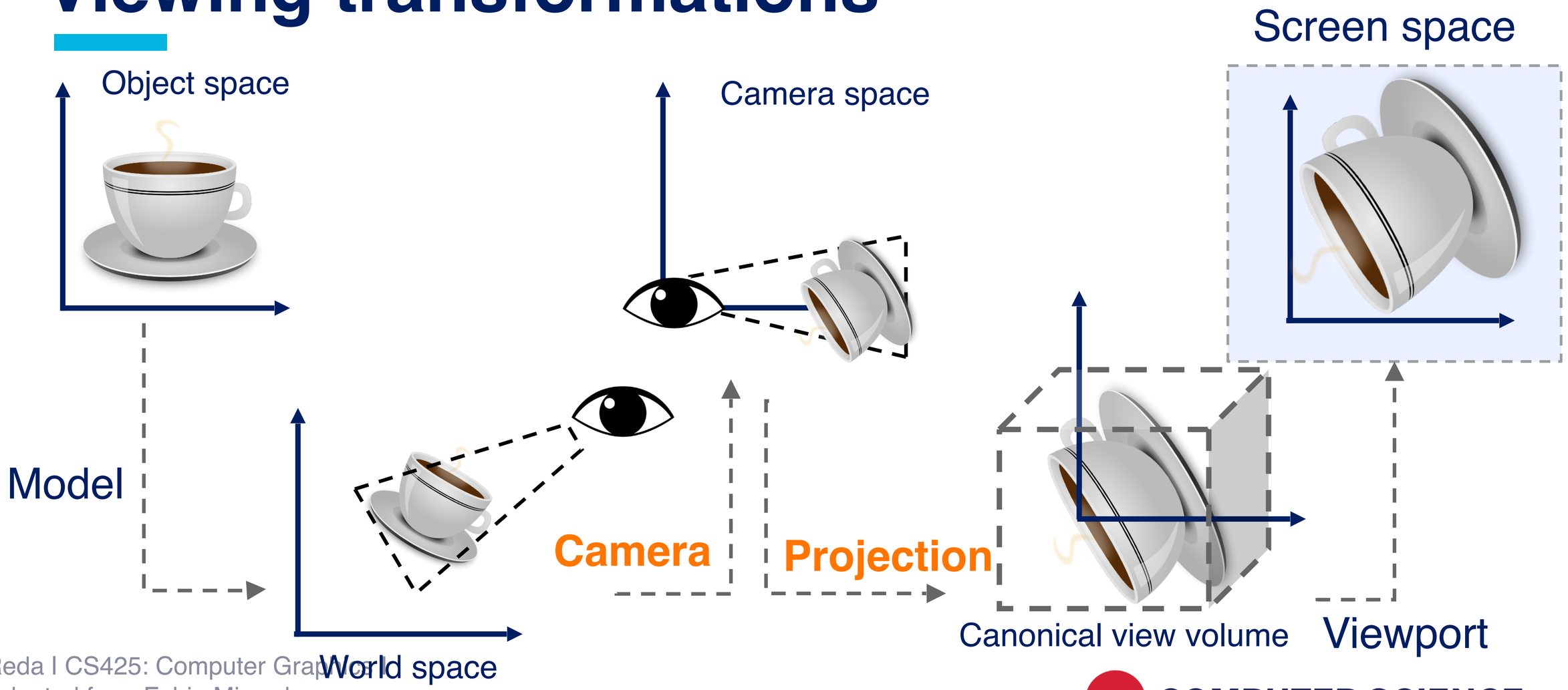
Wednesday — Administrativia

- If you have not accepted assignment 2, please do so today
- Make a small commit ASAP to test whether have write access to your private repo
- **Reminder on code-use / collaboration policy**
  - Re-use of labs and starter code is OK
  - You can search for examples (e.g., on how to use lookAt() or perspective projection).
  - The code you submit must be yours and yours alone. You must be able to explain every bit of code that you submit
  - **No GenAI code allowed!** (We'll check)
  - Post assignment quiz will be used to inform participation and assignment grades
  - No collaboration on assignment (code must be yours alone); studying together is OK

Today: lab

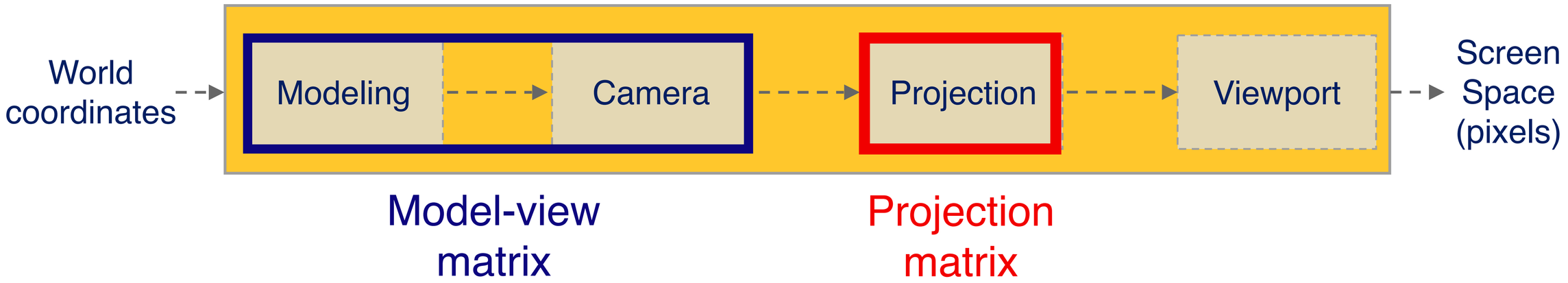
- Get used to performing transformations
- Camera / viewing transformations
- Upload model-view and projection matrices to GPU and perform vertex transform in GLSL shader

Viewing transformations

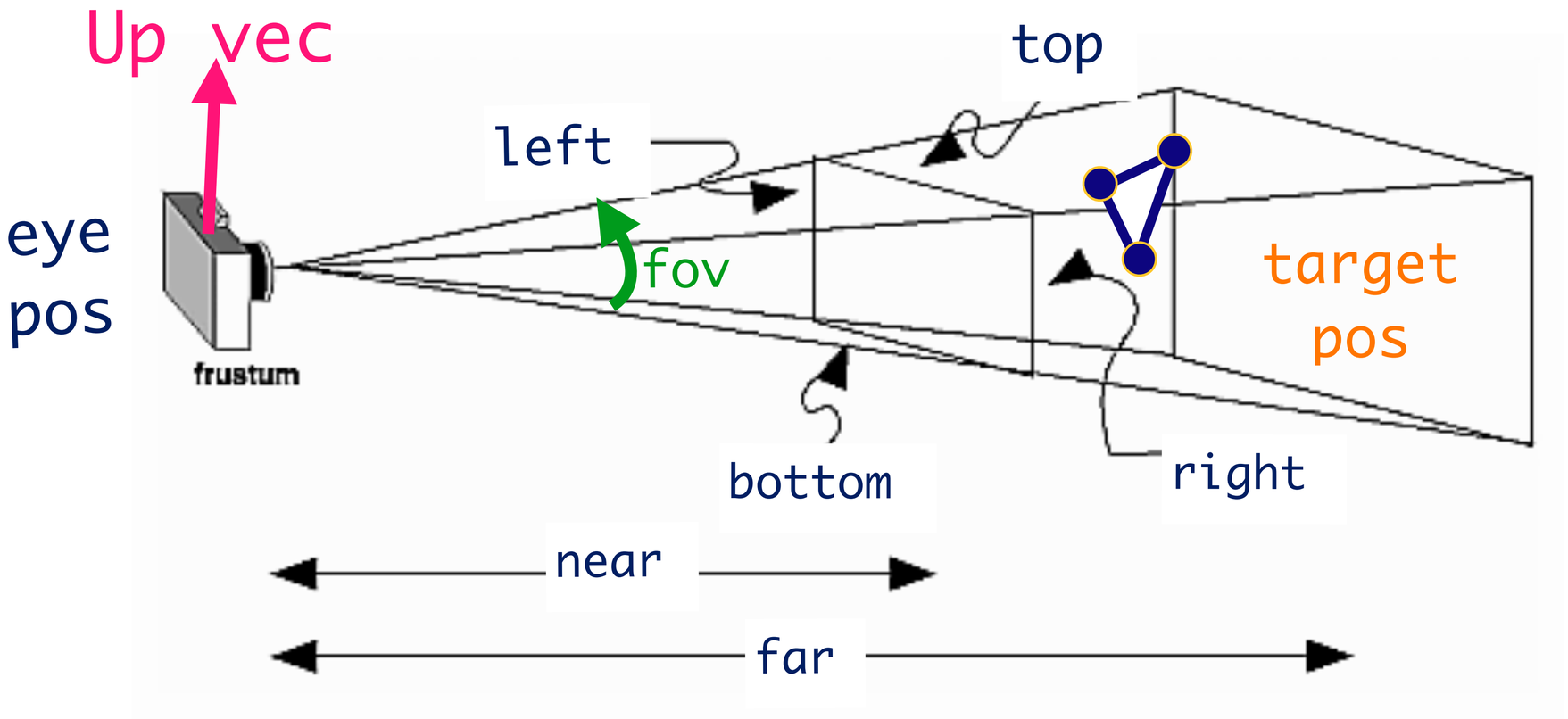


Viewing transformations

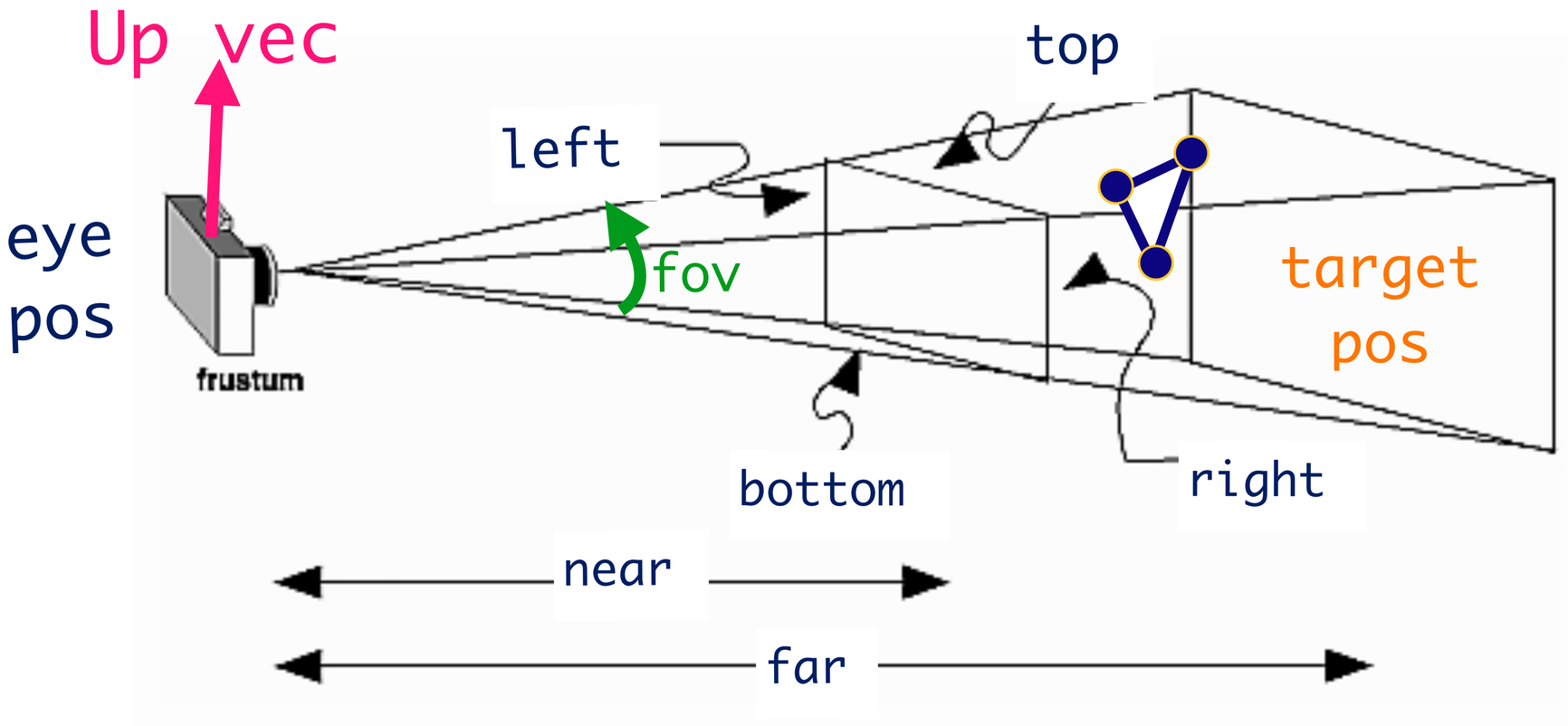
- Viewing transformation is the mapping of coordinates of points and lines from world coordinates into screen space pixels.



Anatomy of viewing + projection transform



Anatomy of viewing + projection transform



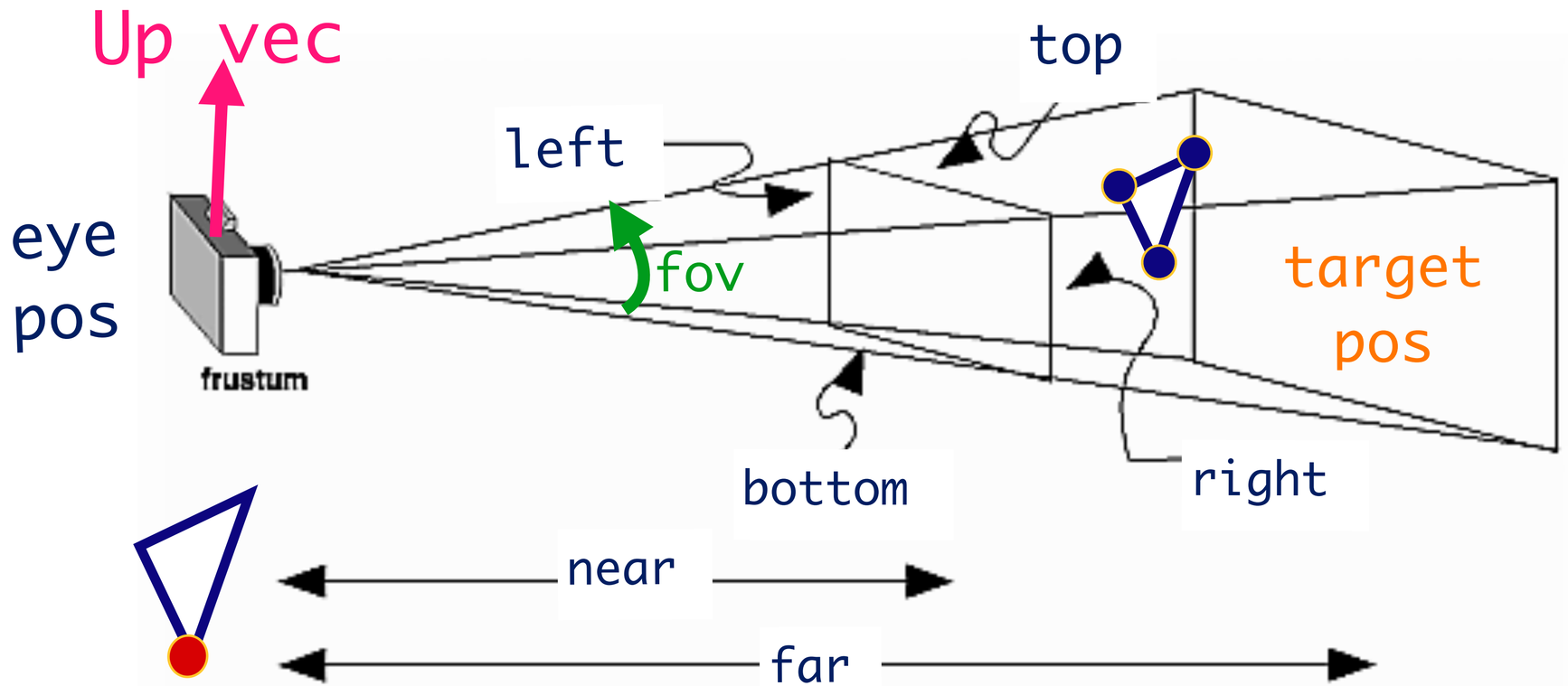
camera (view matrix)

```
lookat(eye, target, up)
```

projection matrix

```
perspective(fov, w/h,
near, far)
```

Anatomy of viewing + projection transform



projection * view * model * v

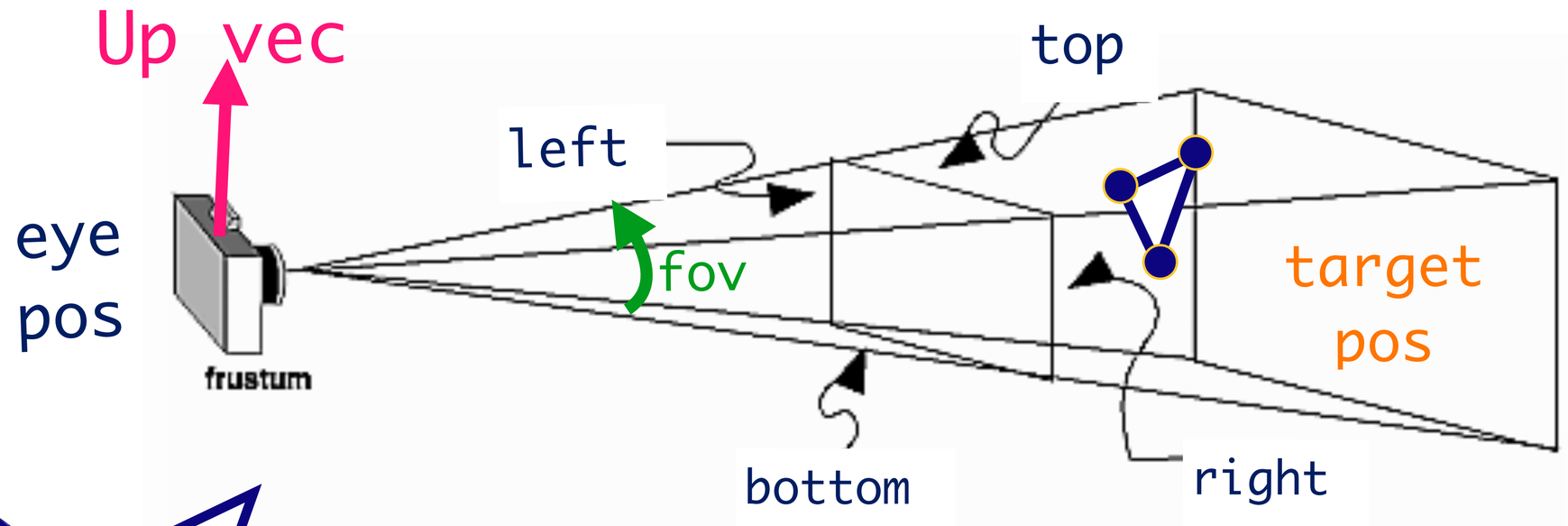
camera (view matrix)

```
lookat(eye, target, up)
```

projection matrix

```
perspective(fov, w/h,
near, far)
```

Anatomy of viewing + projection transform



Any transforms you want to do to your model (scaling, translation, rotation)

projection * view * model * V

camera (view matrix)

```
lookat(eye, target, up)
```

projection matrix

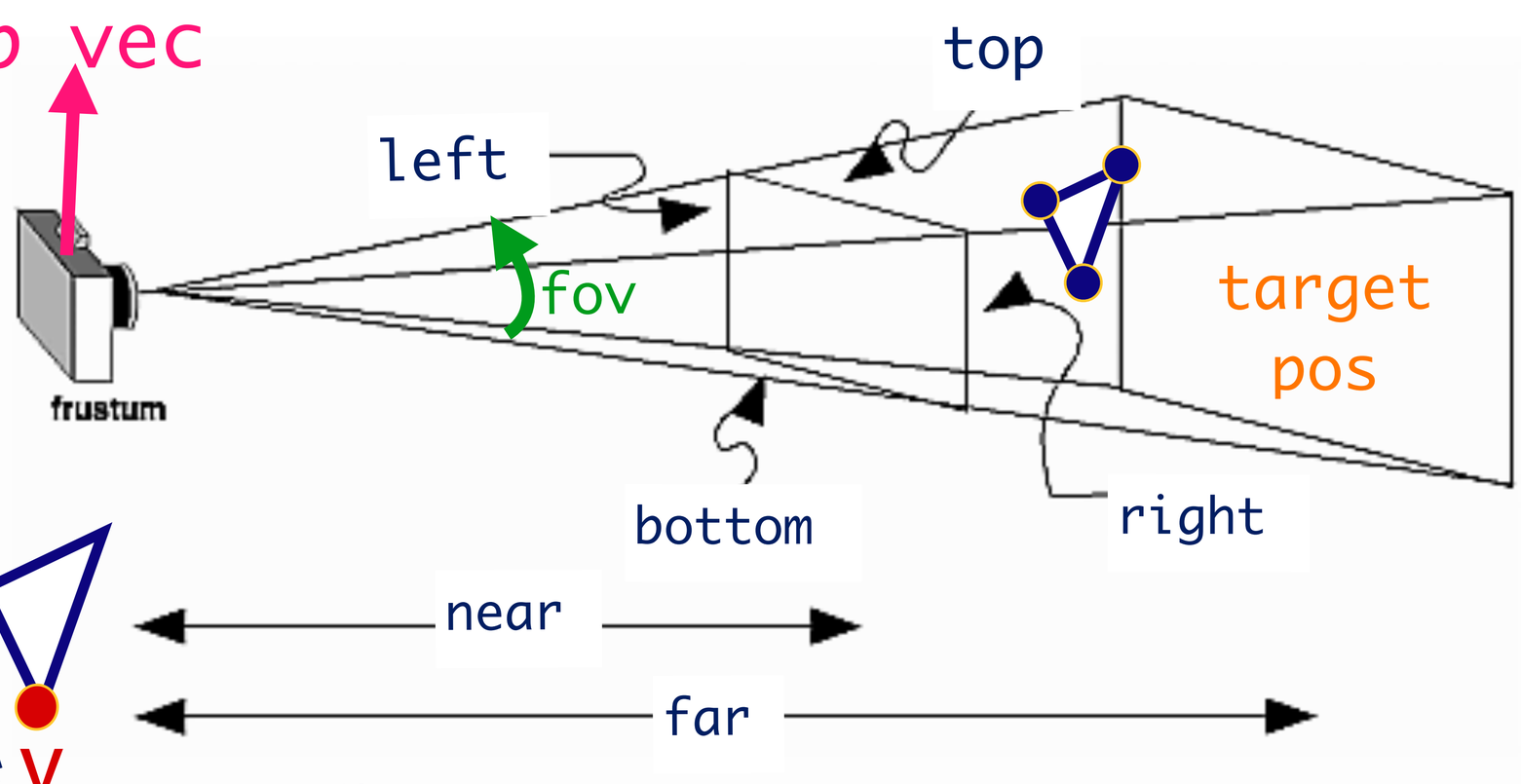
```
perspective(fov, w/h,
near, far)
```

Anatomy of viewing + projection transform

Typically **model-view** matrices are combined into one

Any transforms you want to do to your model (scaling, translation, rotation)

projection * view * model * v



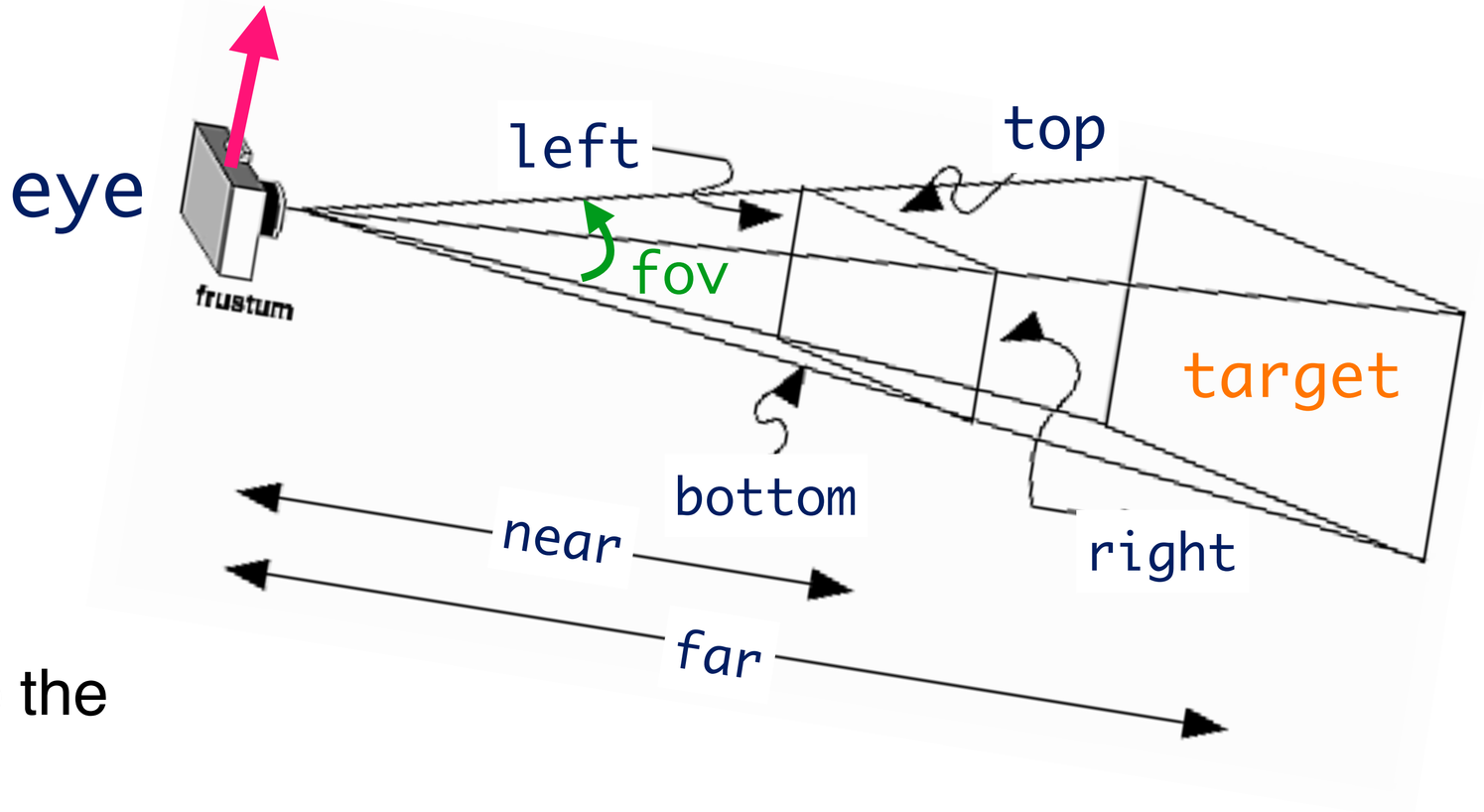
camera (view matrix)

```
lookat(eye, target, up)
```

projection matrix

```
perspective(fov, w/h, near, far)
```

Lab



- Building on **lab-3 (in Piazza)**
  - Apply transformations to rotate the **model** (boxes) about Y and X
  - Switch from a perspective matrix to orthographic matrix
  - Bonus: add zoom slider to get closer/ further from the model

view

```
lookat(eye, target, up)
```

projection

```
perspective(fov, w/h,
near, far)
```