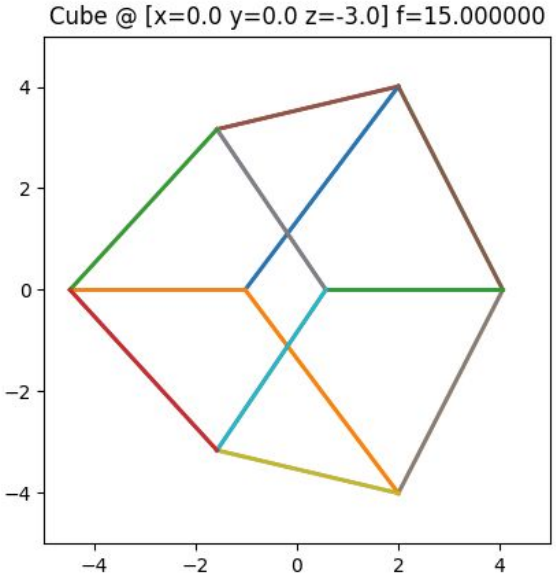
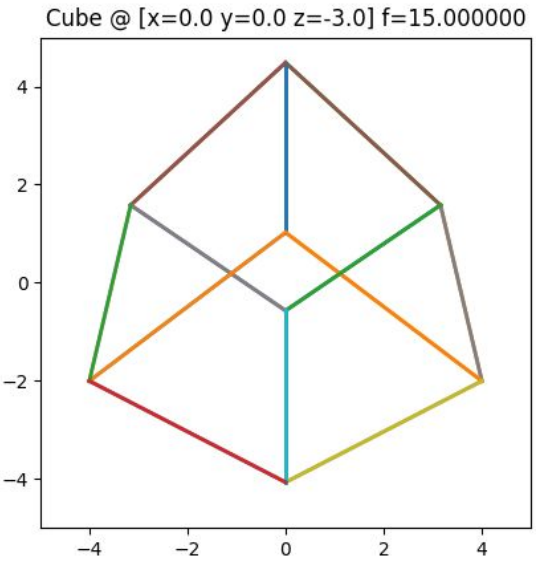
Camera Projection Matrix

1)

a) See submission details.

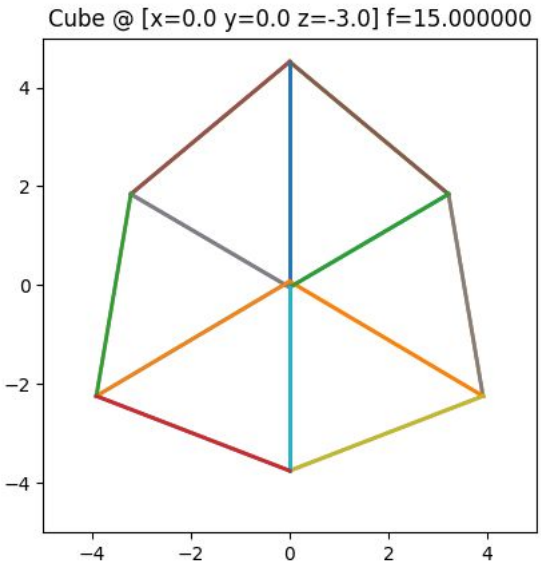
b) 3D rotation matrices are not commutative.



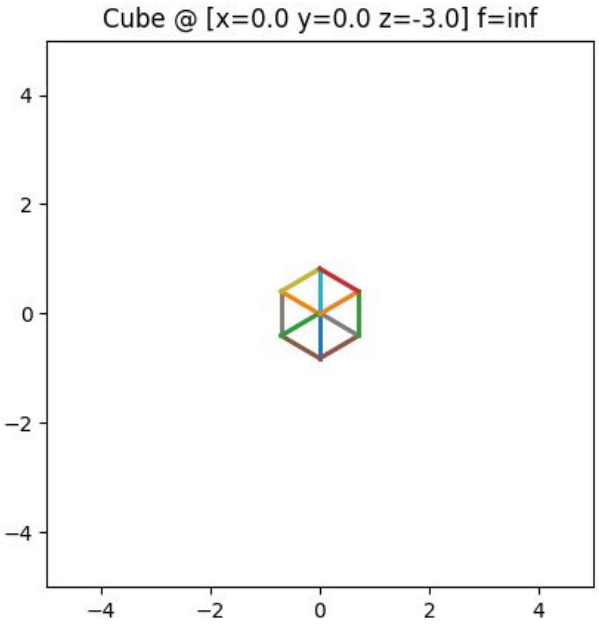


c) For rotX, I used pi/5 while for rotY I used pi/4. Additionally, here is the ordering for the function calls I made.

```
renderCube(R=rotX(math.pi / 5).dot(rotY(np.pi / 4)), file_name="1c")
```



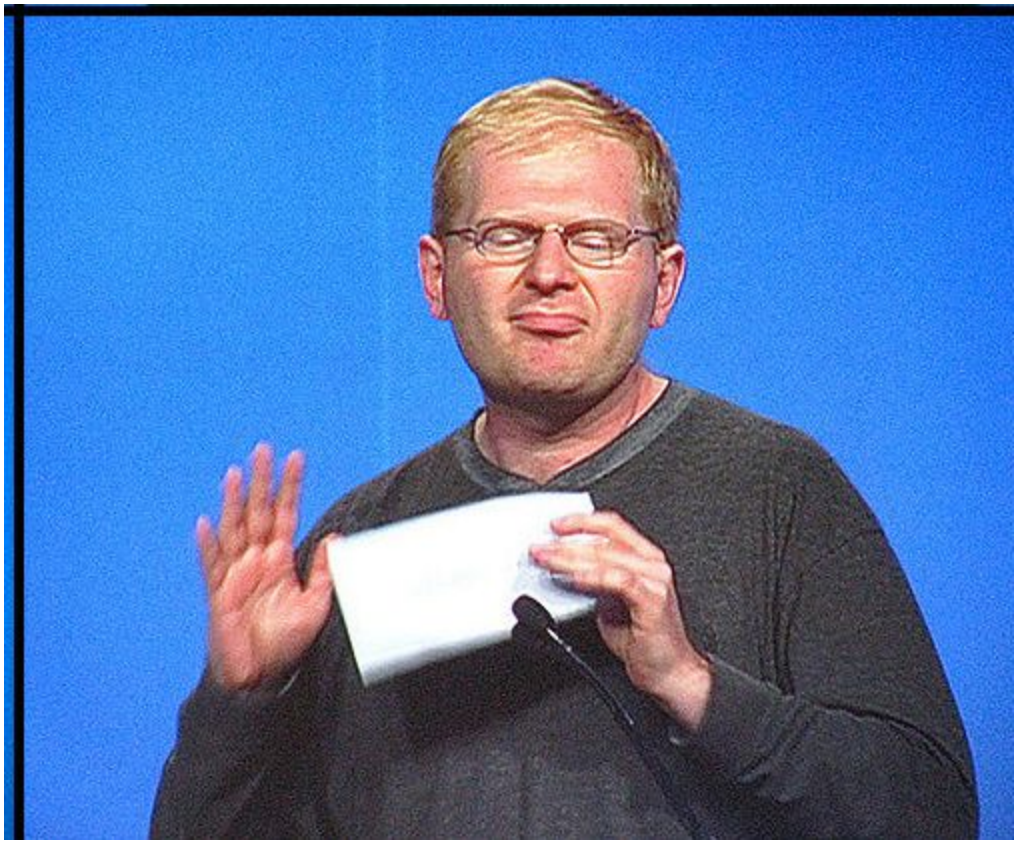
d) Orthographic project where f == inp.f



Prokudin-Gorskii

1) Combined image

2) Optimal offsets for each image are below. Important to note that image alignment was much better with normalization than without.



Optimal offsets

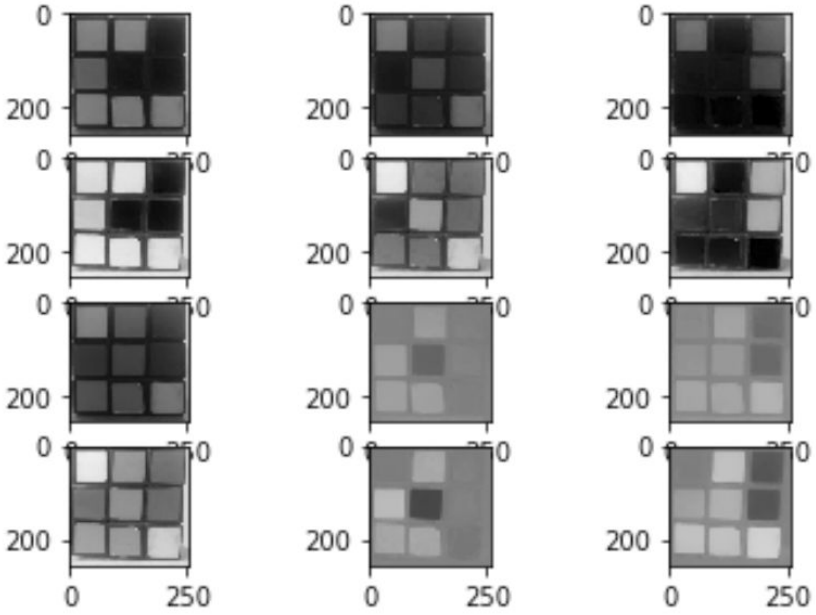
```
prokudin-gorskii/00125v.jpg
b --> r:  [12, 1]
g --> r:  [7, 1]
prokudin-gorskii/00149v.jpg
b --> r:  [13, 0]
g --> r:  [6, 0]
prokudin-gorskii/00153v.jpg
b --> r:  [0, -3]
g --> r:  [-1, -2]
prokudin-gorskii/00351v.jpg
b --> r:  [1, 0]
g --> r:  [-2, 0]
prokudin-gorskii/00398v.jpg
b --> r:  [6, 1]
g --> r:  [3, 0]
prokudin-gorskii/01112v.jpg
b --> r:  [13, -1]
g --> r:  [4, -1]
tableau/efros_tableau.jpg
b --> r:  [6, 5]
g --> r:  [3, 10]
```

Offsets without normalization

```
prokudin-gorskii/00125v.jpg
b --> r:  [14, -12]
g --> r:  [13, 12]
prokudin-gorskii/00149v.jpg
b --> r:  [7, -10]
g --> r:  [0, -9]
prokudin-gorskii/00153v.jpg
b --> r:  [11, -10]
g --> r:  [-6, -10]
prokudin-gorskii/00351v.jpg
b --> r:  [11, -10]
g --> r:  [9, 13]
prokudin-gorskii/00398v.jpg
b --> r:  [12, -6]
g --> r:  [3, -9]
prokudin-gorskii/01112v.jpg
b --> r:  [-11, 9]
g --> r:  [8, -6]
tableau/efros_tableau.jpg
b --> r:  [-15, 1]
g --> r:  [-14, 5]
```

Prokudin-Gorskii

1)



2) LAB better separates the illuminance change from other factors. This is largely because in LAB, one component, L, contains the luminance information. Making a pixel darker without changing the hue or saturation, will change the L component but not the A and B components. Similarly, if we change the color without making it darker or lighter, then L will still remain the same while A or B (possibly both) will change.

3)