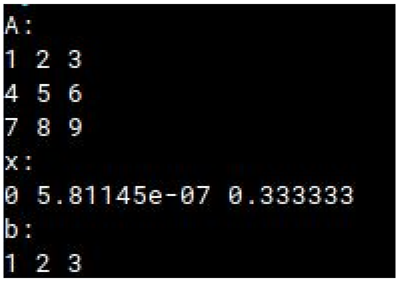
Jonathon Hewitt

PA2 Report

Face normalization:
I was unable to complete the project due to difficulty with correctly solving SVD on the constructed matrices. Both the given code on the course website and the code from opencv4 for solving these systems of equations gave inconsistent, and incorrect solutions. I confirmed that I was using the code from the course website correctly by solving a simple system of equations that can be easily verified as getting the correct solution. Examples of getting correct solutions, and some of the incorrect solutions I was getting for my images are shown below. Also shown below is an example image showing that my transform application function is correct by doing a simple rotation affine transformation.

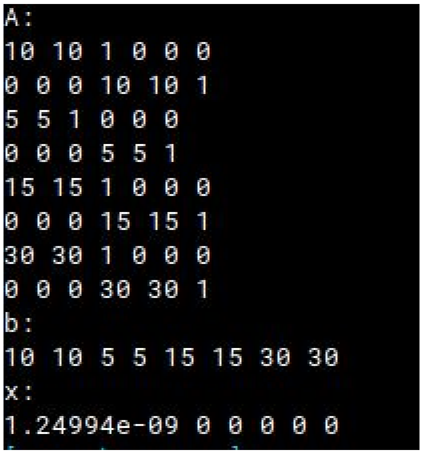
Simple SVD Solution



```
A:
1 2 3
4 5 6
7 8 9
x:
0 5.81145e-07 0.333333
b:
1 2 3
```

As we can see, the solution found by the solve_system function on a simple 3x3 matrix is correct. The same solution was found every time the program was ran.

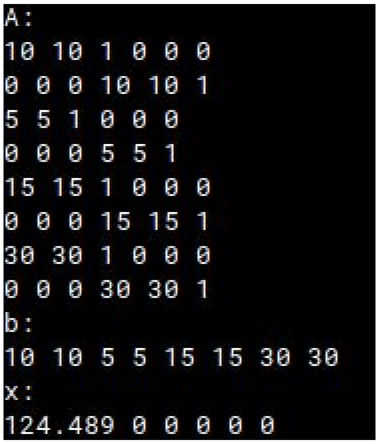
Incorrect SVD Solution on an easily verifiable matrix



```
A:
10 10 1 0 0 0
0 0 0 10 10 1
5 5 1 0 0 0
0 0 0 5 5 1
15 15 1 0 0 0
0 0 0 15 15 1
30 30 1 0 0 0
0 0 0 30 30 1
b:
10 10 5 5 15 15 30 30
x:
1.24994e-09 0 0 0 0 0
```

Because the transformation doesn't move any of the positions, we'd expect to see x=[1,0,0,0,1,0] so that we'd have
X_1^f = 1x + 0y + 0
Y_1^f = 0x + 1y +

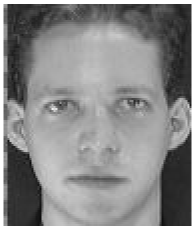
However, that is not the case. In addition to this, the value of the solution changes every time the program is run. Here is another example solution with the same exact input:



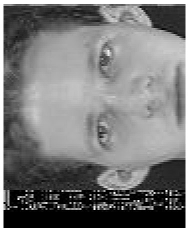
```
A:
10 10 1 0 0 0
0 0 0 10 10 1
5 5 1 0 0 0
0 0 0 5 5 1
15 15 1 0 0 0
0 0 0 15 15 1
30 30 1 0 0 0
0 0 0 30 30 1
b:
10 10 5 5 15 15 30 30
x:
124.489 0 0 0 0 0
```

Because I was unable to get valid transformations for the face normalization, I was unable to produce any valid output images. I would like to show that despite the fact that I was unable to get any good values for my transformations, that my code for transformation application is valid and works well. Below are a few example inputs and outputs from my transformation function with different transformation matrices.

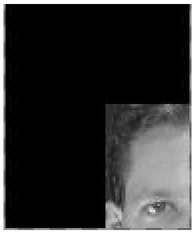
Input Image:



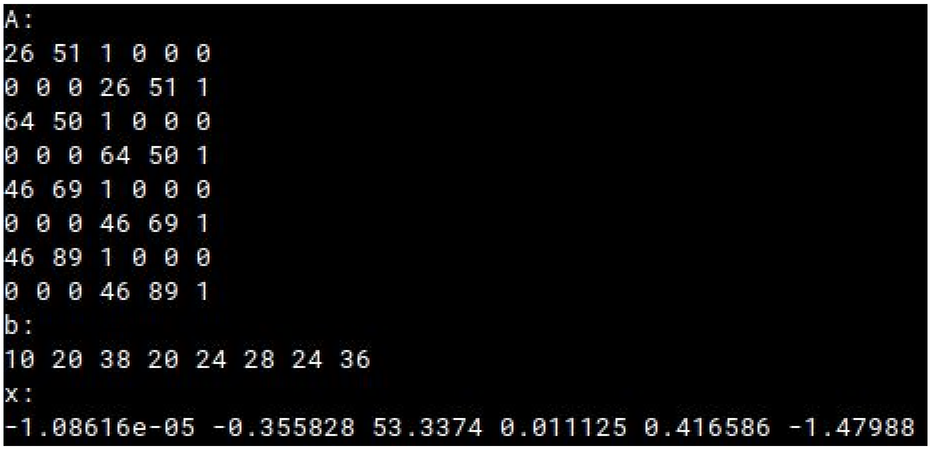
Rotation:
X = [0,1,0,1,0,0]



Translation:
X = [1,0,50,0,1,50]



Normalized:
X = [-1.08616e-5,-0.355828,53.3374,0.011125,0.416586,-1.47988]



```
A:
26 51 1 0 0 0
0 0 0 26 51 1
64 50 1 0 0 0
0 0 0 64 50 1
46 69 1 0 0 0
0 0 0 46 69 1
46 89 1 0 0 0
0 0 0 46 89 1
b:
10 20 38 20 24 28 24 36
x:
-1.08616e-05 -0.355828 53.3374 0.011125 0.416586 -1.47988
```