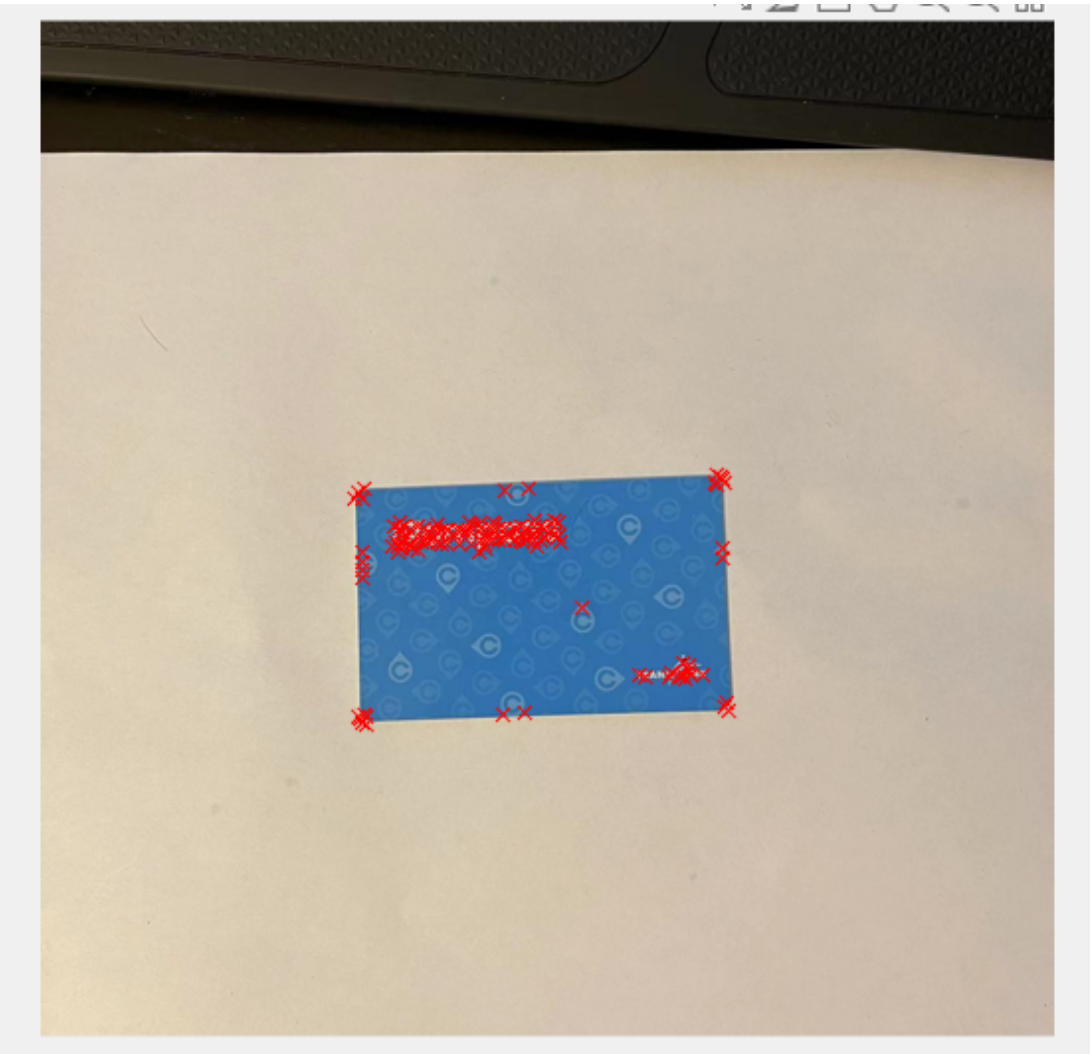
Report

Part 2:

To run the code, click on run, and enter a number from 1 to 3 at the command line to select an image. The main code is part2.m. You can change images at line 11, 22, and 34.
The basic images of different sets:

card1.jpg: dimension = 512*512; threshold = 0.01

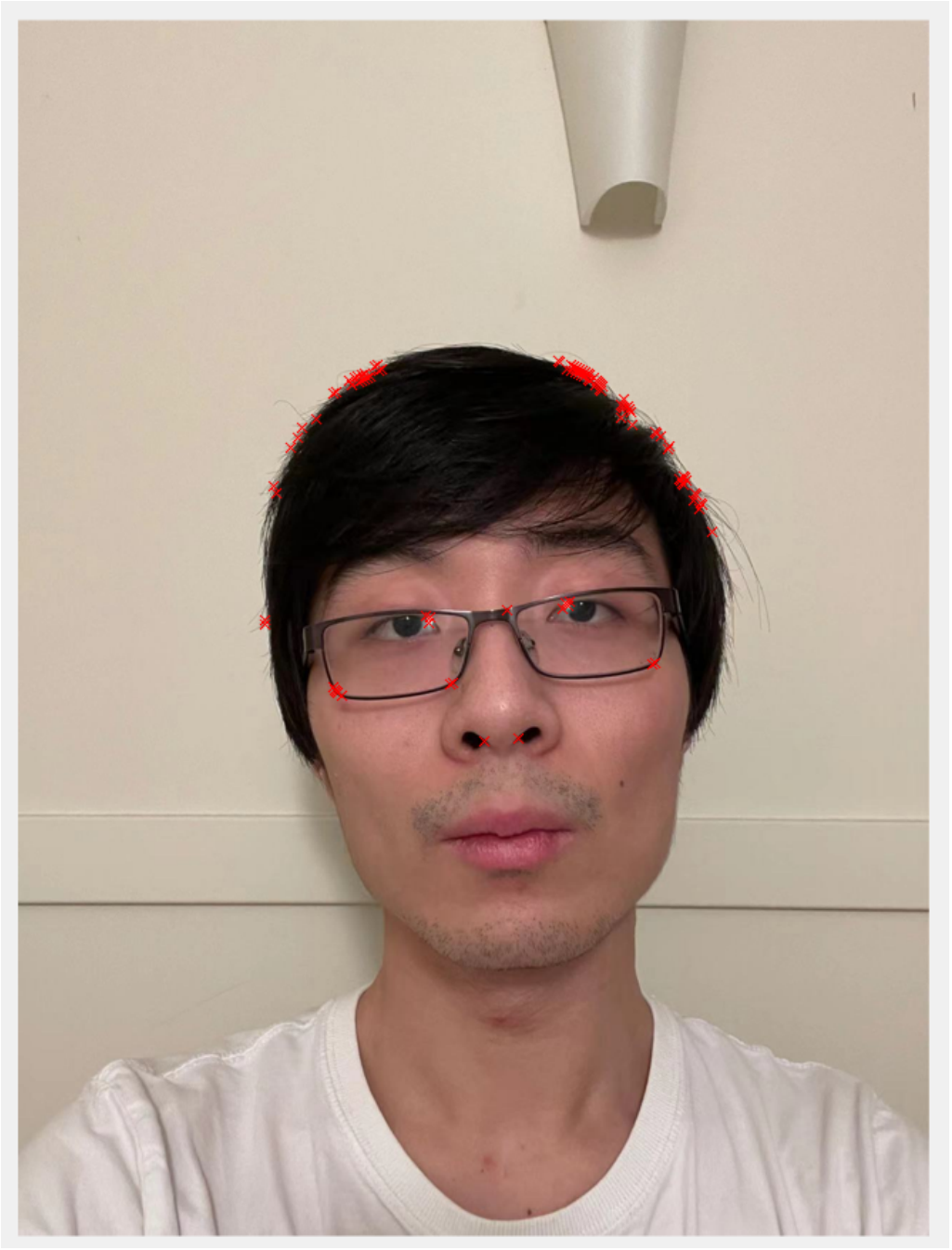


Harris Corner detected all of the corners of the card. However, there is some noisiness around the corners, and much more on the surface of the card. I expected there to be only 4 detected corners, but the actual number of detected corners is way beyond my expectations.

mask1.jpg: dimension=original dimension/2, original dimension/2; threshold=0.1

This one is perfect in my opinion. The 8 corners (4 outer corners, 4 corners at the black and white boundaries) are clearly detected. The dense clusters of detected corners on the sides are holes on the mask. Other places on the surface do not have any detected corners.

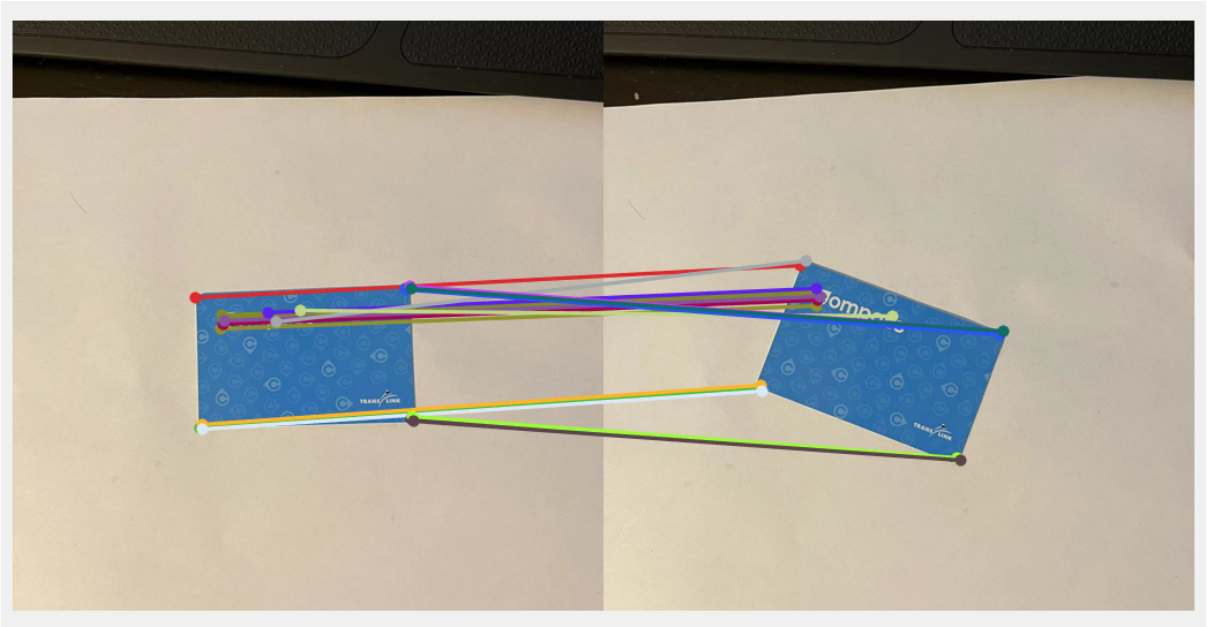
me1.jpg: dimension=original dimension/2, original dimension/2; threshold=0.05



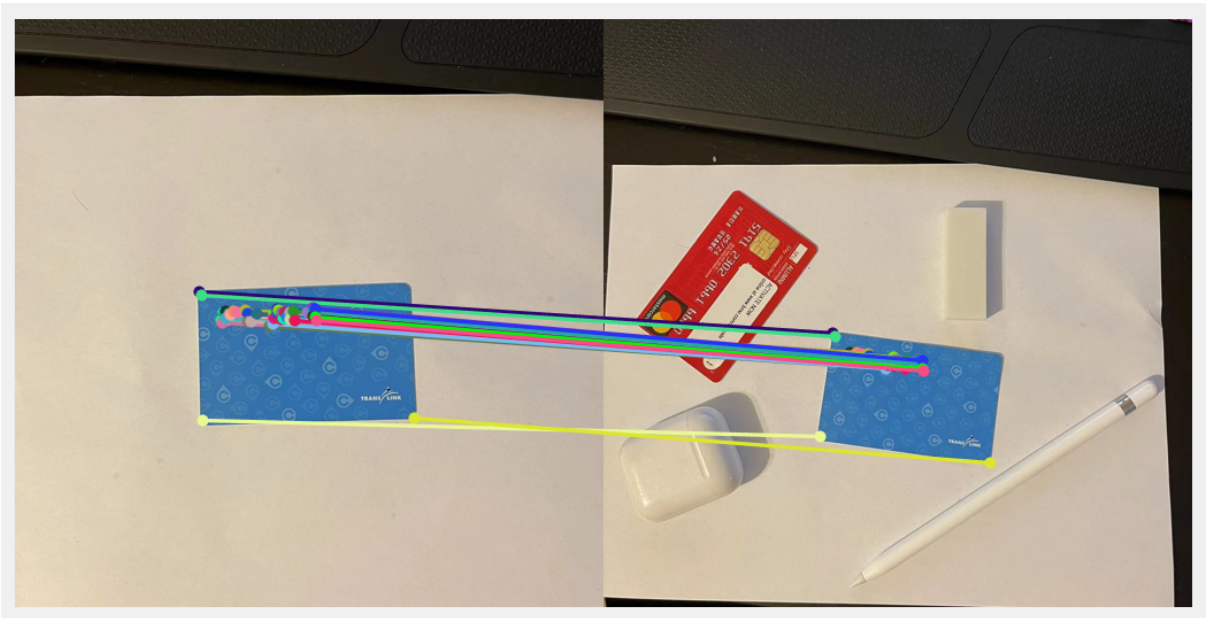
This one is not good. I expected there to be corners around my chin and my shoulder, but there are none.

Part 3:
To run the code, click on run and enter a number between 1 to 3 (represents difficulty) in the command line. The main code is part3.m. To change images, change at line 11 12, 45 46, 75 76

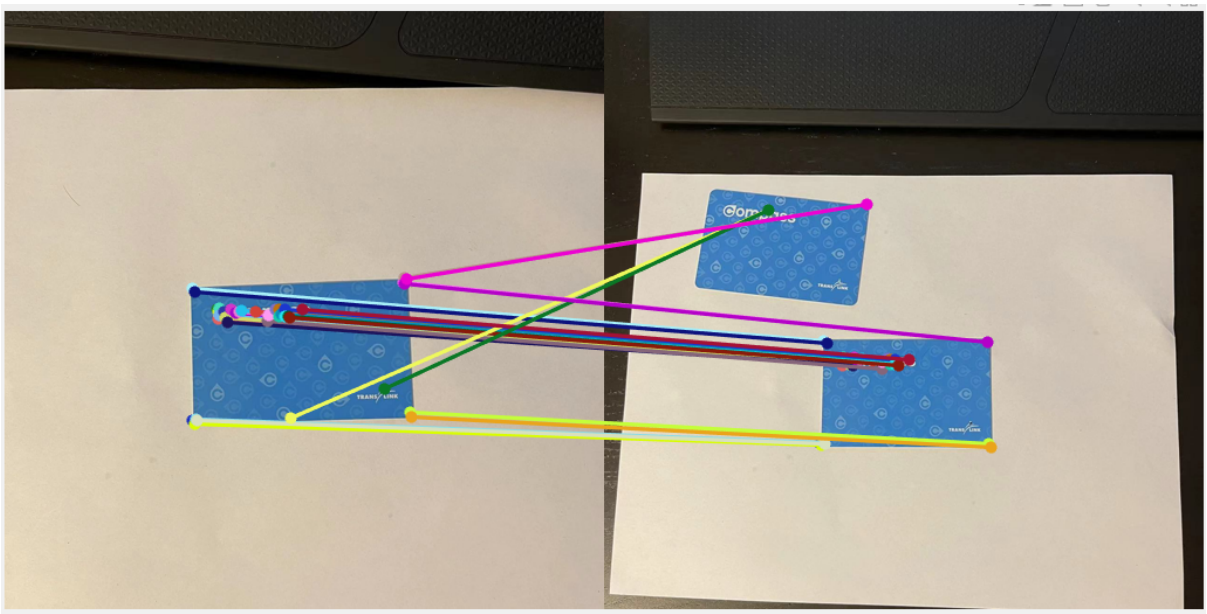
Difficulty 1: card1 vs card2
dimension = 512*512; Harris threshold = 0.01; feature matching threshold:0.88
precision: 0.145; recall:0.039; F score: 0.0615



Difficulty 2: card1 vs card3
dimension = 512*512; Harris threshold = 0.01; feature matching threshold:0.88
Precision = 0.3359; Recall=0.0656; F score=0.1097



Difficulty 3: card1 vs card4
dimension = 512*512; Harris threshold = 0.01; feature matching threshold:0.88
Precision= 0.2901; recall: 0.2879; F score=0.2890



The precision, recall and f score are all very low because lots of the correct data were filtered away by the threshold. If I use the data without filtering, the accuracy should increase a lot.

Part4:

To run the code, click on run and enter a number between 1 to 3 (represents difficulty) in the command line. The main code is part4.m. To change photos, change at 11 12, 45 46, 73 74.

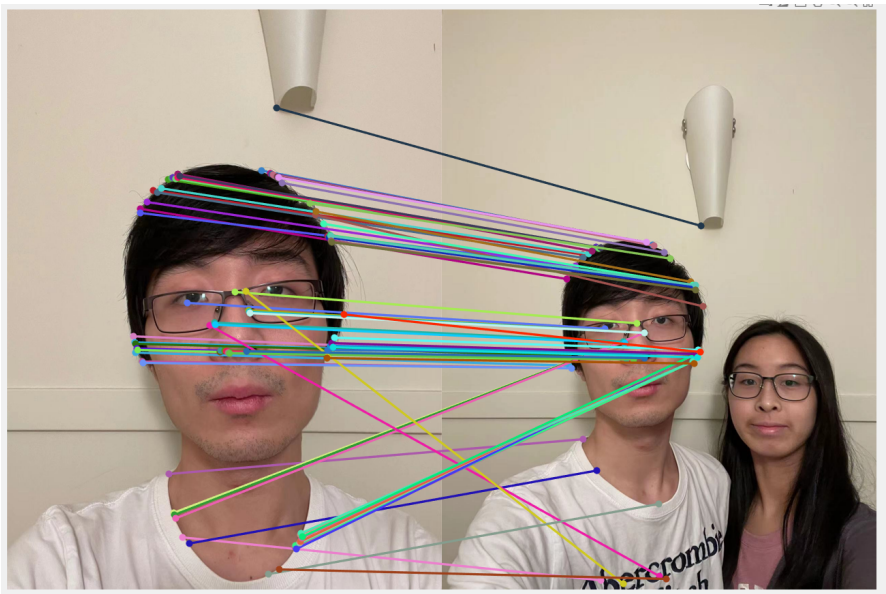
The difference between my basic descriptor and advanced descriptor is that I applied linear interpolation. In the basic descriptor, I don't have linear interpolation so I alway put the angle of every pixel into their upper bound bins. This is not accurate because for example, 60 degrees is between 45 and 90 degrees, and it is obviously closer to 45 degrees, but I put it in the upper bound bin: 90 degrees. In the advanced descriptor, I use linear interpolation to determine how much portion is an angle in its lower bound and upper bound bins.

To me more specific, here is my logic for the advanced descriptor:

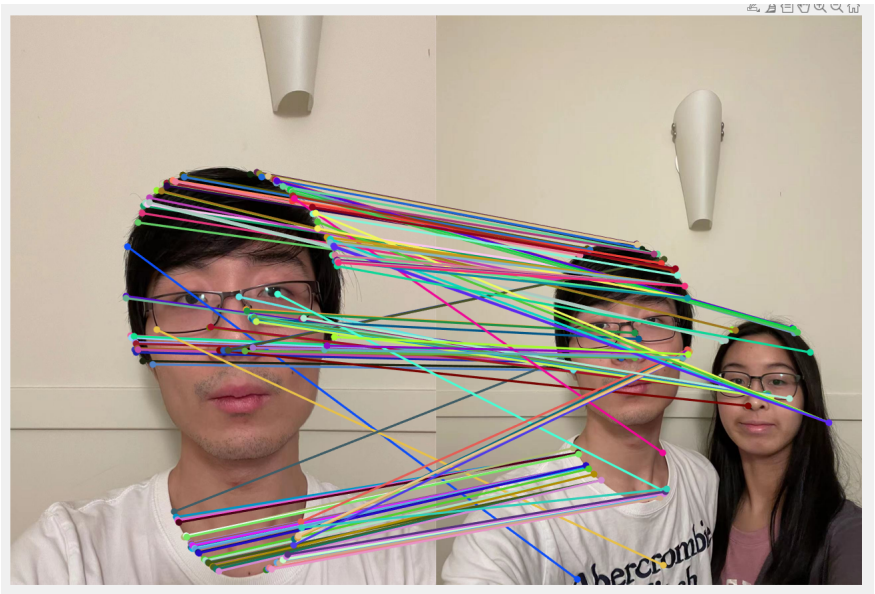
```
for every angle
        for index=1 to 8
                if angle < current_bin_degree and angle>prev_bin_degree
                        angle_ratio=(angle-prev_bin_degree)/45
```

When distributing values into the histograms:
magnitude_in_lower_bound=magnitude*（1-angle_ratio)
magnitude_in_higher_bound=magnitude* angle_ratio
lower_hitogram=lower_hitogram+magnitude_in_lower_bound
higher_histogram=higher_histogram+magnitude_in_higher_bound

basic descriptor



advanced descriptor



With all dimension and threshold controlled, The advanced descritor turns out to have more matching lines, but the accuracy is lower than the basic descriptor.