Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
1a-)

Figure 1: Average Face

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
1b-)

Figure 2: Eigenface 1

Figure 3: Eigenface 2

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

Figure 4: Eigenface 3

Figure 5: Eigenface 4

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

Figure 6: Eigenface 5

Figure 7: Eigenface 6

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
1c-)

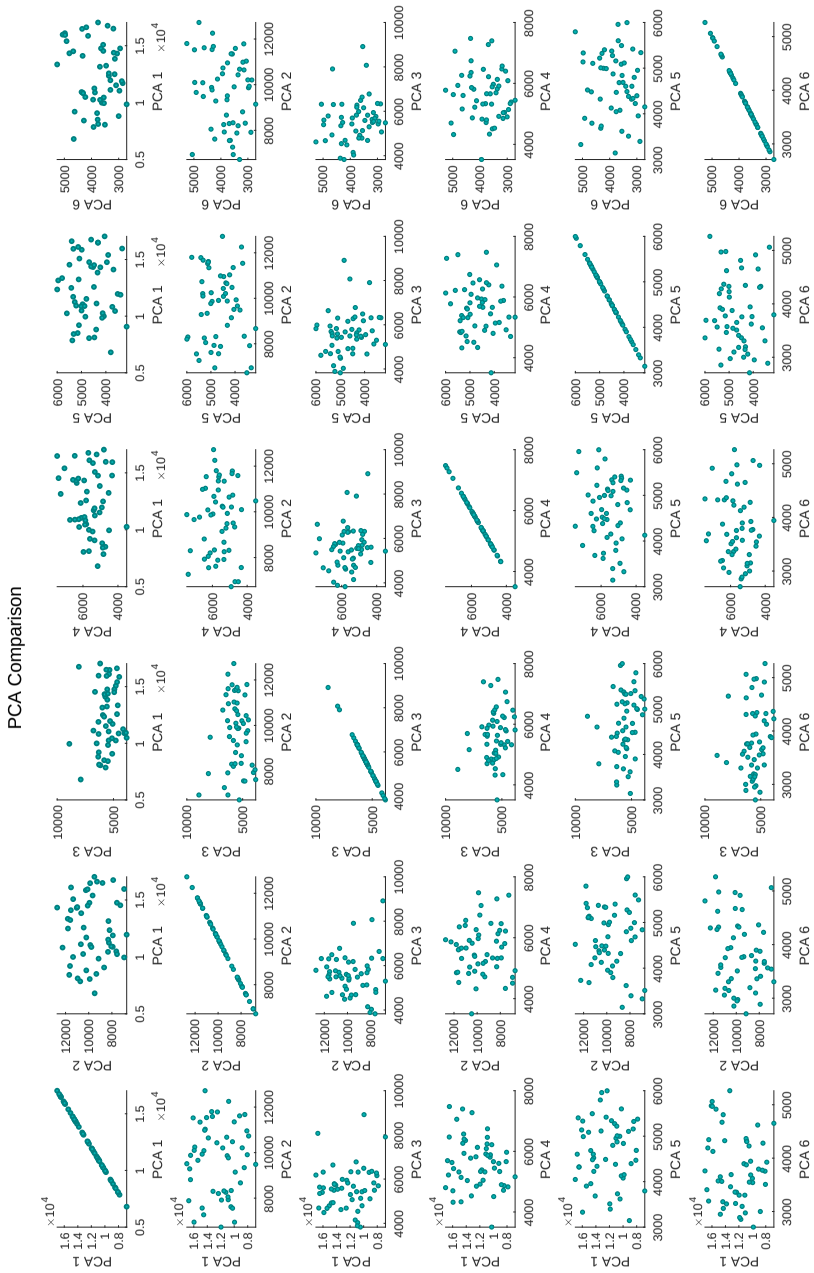

Figure 8: PCA Plots

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

1d-)

The primary reason of conducting a PCA snapshot is that when processing images, the covariance matrix itself is very large and takes a lot of computational power. The trick was developed to get as many “eigenfaces” as there are images, which in this case we got 53 because we have 53 images of faculty. And we needed the average image to center each face in order to get an accurate representation. The eigenfaces themselves, from my understanding, form a basis of every image and so every image can be represented as a combo of such eigenfaces and the more significant eigenfaces, such as those shown in (Figures 2-7) are able to capture more significant features. The first two eigenfaces illuminate the left a right facial regions, the third eigenface (Figure 4) captures more of the cheek region, while eigenfaces four and five capture the eye region, and eigenface 6 (Figure 7) the teeth. We can also see from (Figure 8) that each face falls into a different area in the PCA plots, these plots basically show which eigenfaces provide more information for that specific image because each image is a combo of these basis vectors.

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

2a-)

You only need two dimensions to capture this manifold

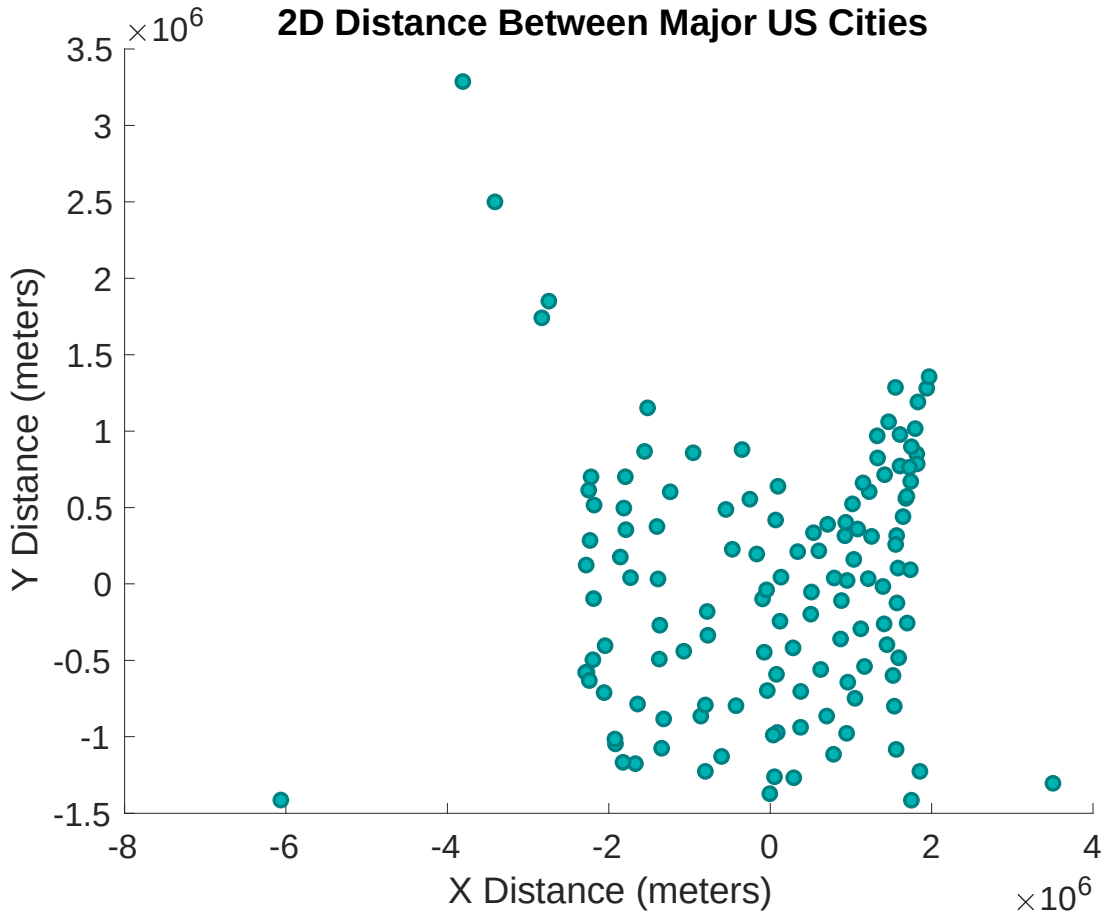

Figure 9: Distance between Major U.S. Cities 2D

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

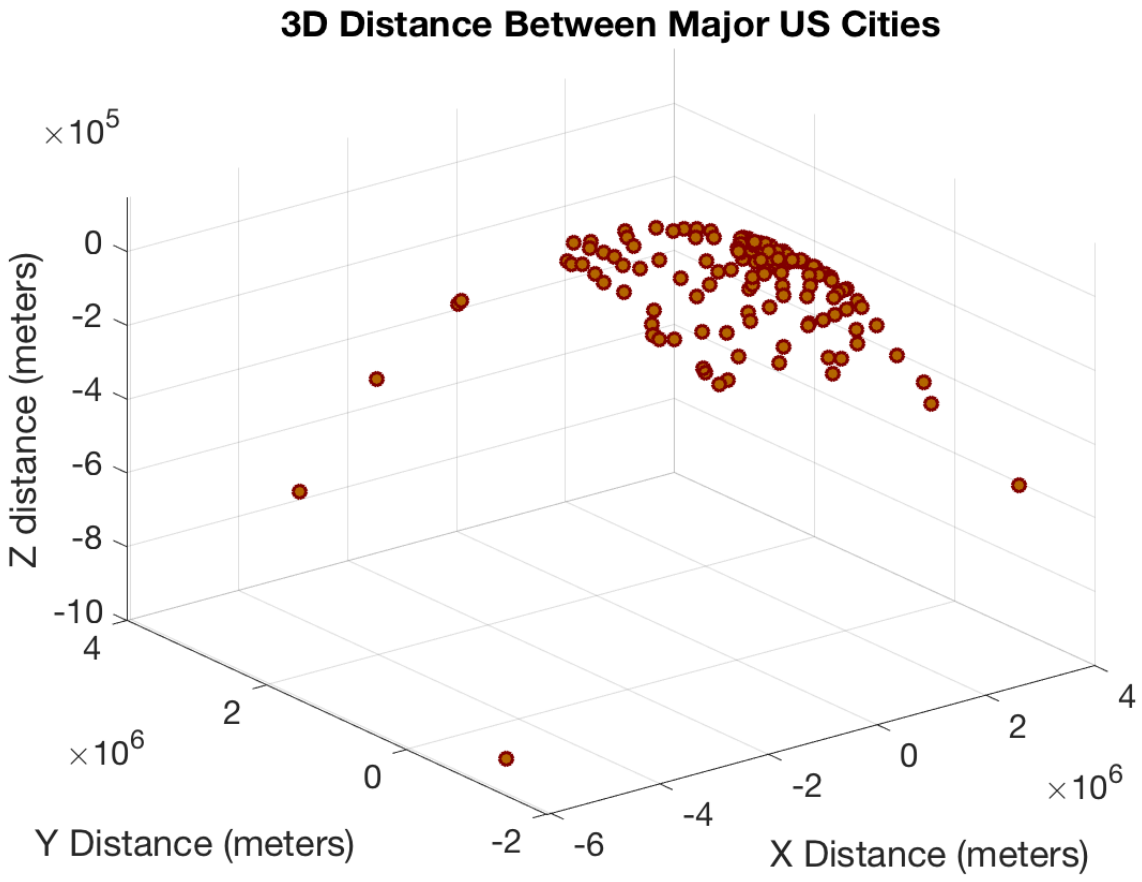

Figure 10: Distance between Major U.S. Cities 3D, First View

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

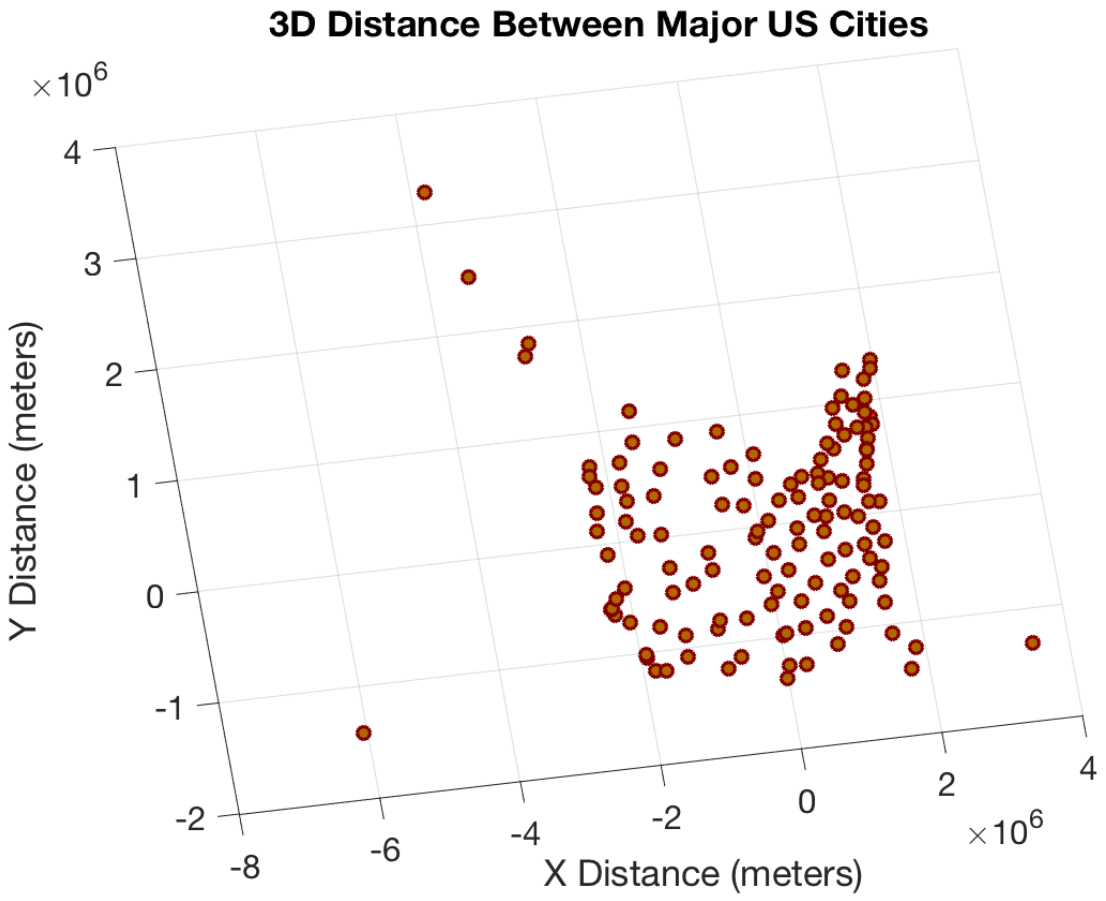

Figure 11: Distance between Major U.S. Cities 3D, Second View

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
2b-)

You need three dimensions to get a good representation of this manifold!

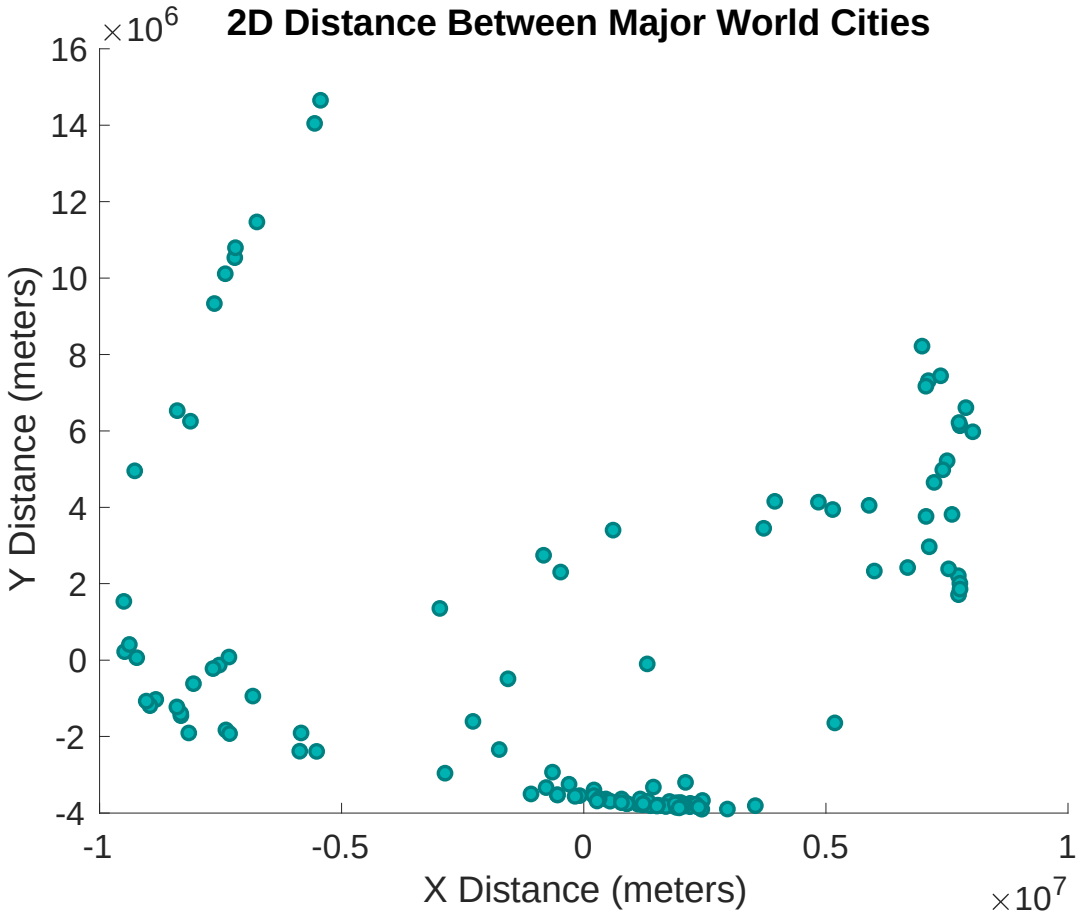

Figure 12: Distance between Major World Cities 2D

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

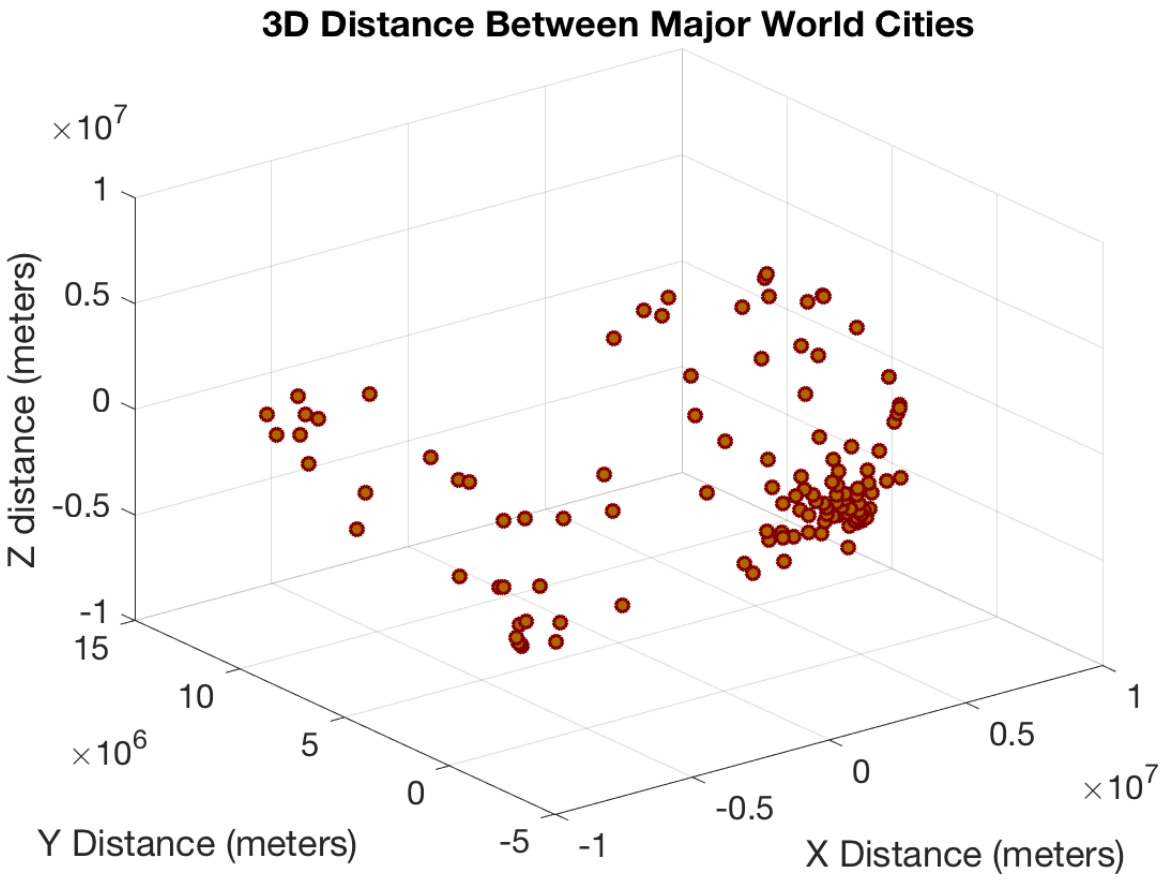

Figure 13: Distance between Major World Cities 3D, View 1

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

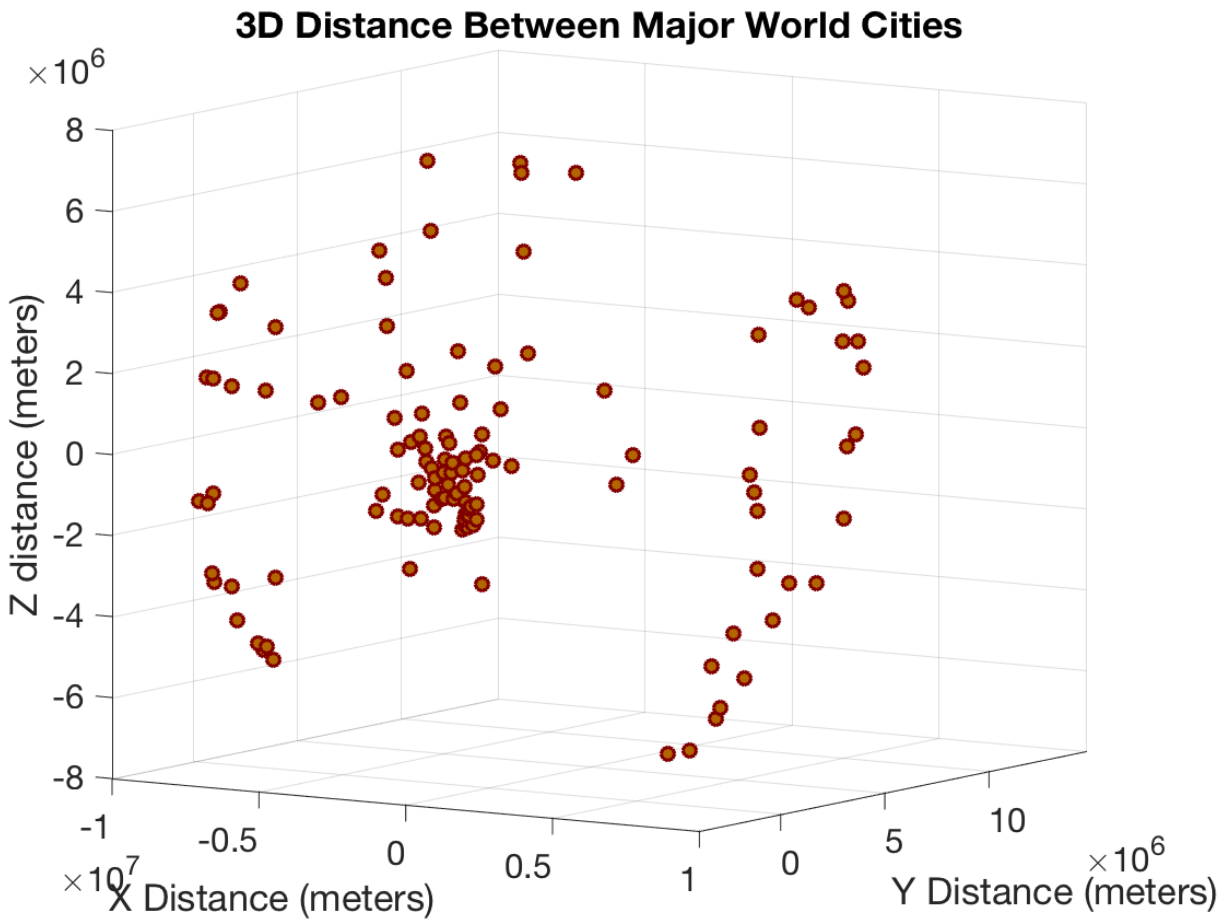

Figure 14: Distance between Major World Cities 3D, View 2

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
2c-)

We can see from the results that the two manifolds require a different number of dimensions to be captured, this all ties into the amount of variance captured by the eigenvalues when creating the manifold. In the US city case, the first two eigenvectors alone capture 99 percent of the variance, that is why the (Figure 9) displays a pattern of the United States that is recognizable, and the same for the 3D plot as well (Figure 10), which looks like the US on a sphere. However, in the case of the World cities, the first two eigenvectors capture 78.42 percent of the variance (a.k.a not good enough); so we can see in (Figure 12) that the information in two dimensions is not that discernable. Nonetheless, if we take the three largest eigenvalues they can capture 91.75 percent of the manifold, and thus when looking at (Figure 14) we can see a spherical structure with the shape of continents, not as clear as the US cities example, but still good enough to be perceived. This example shows how linear manifolds embed information on different dimensions depending on the eigenvalues!

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
3a-c)

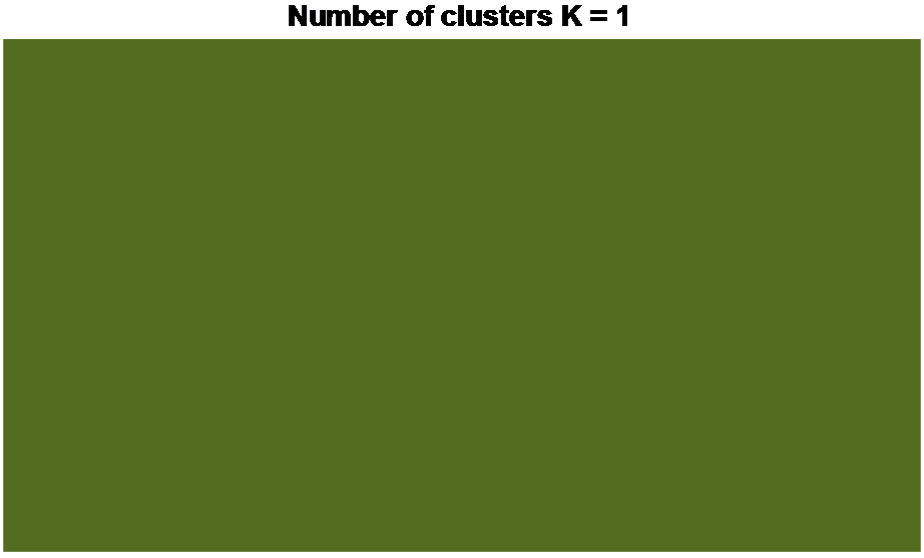

Figure 15: Image for K=1

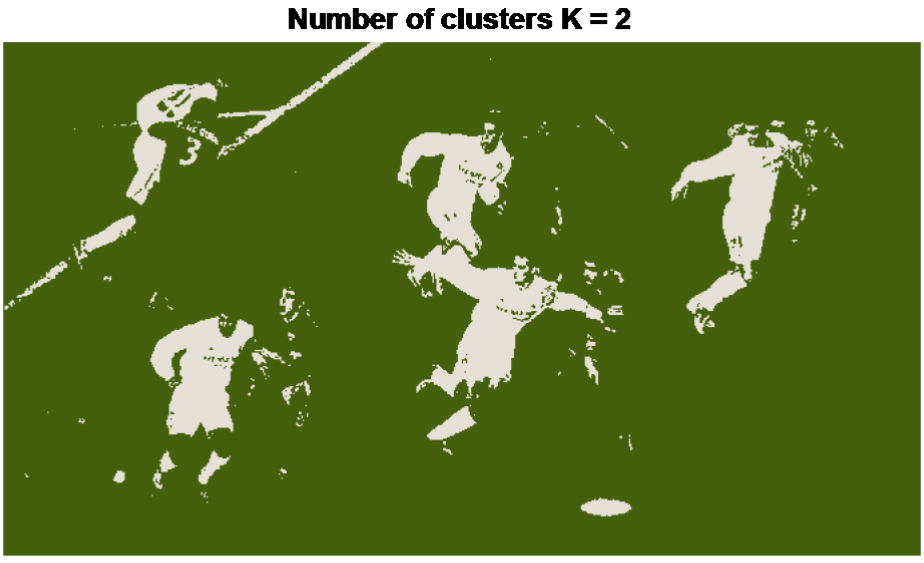

Figure 16: Image for K=2

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

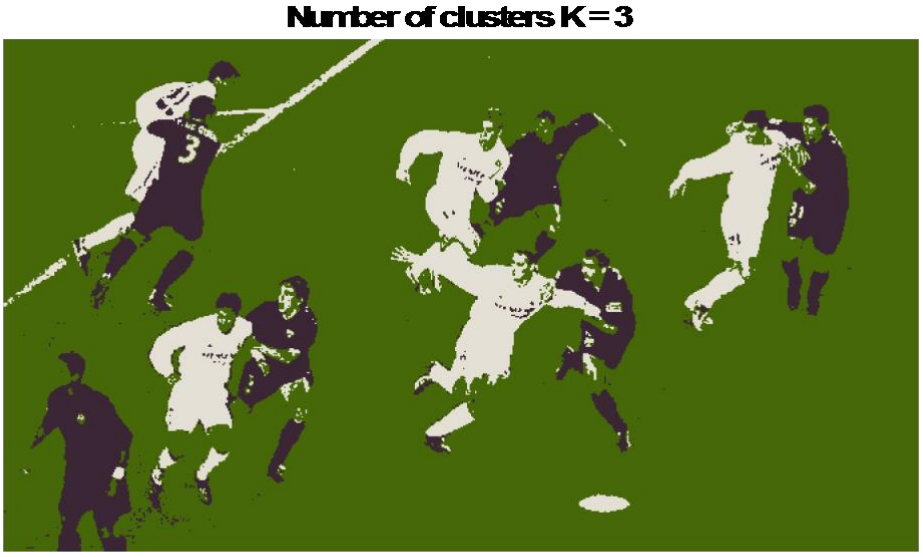

Figure 17: Image for K=3

The images were generated with the codebooks and saved into the zip file. I will only show the first three here because showing the rest in this document will just be redundant. As seen (Fig 15-17), when you progress from the first to the last image you will notice that a new shade (and RGB color) is being added consecutively. This is due to the fact that per K increase we get a new color, because the codebook in this case is simply codes of RBG colors. Every vector in the codebook has three values, an R, G, and a B. It is also important to note that the first "code" in the codebook is the most dominant color in the image, which is the grass. That is why the first image is all green, because it is only composed of one color from the codebook and that color happens to be the average pixel color! If you go to K=2 you see that the color white is next due to the prominence of white on the jerseys and on the lines in the field. So basically each code is the next most significant color, and whatever color on the image is closest to that code will get mapped to that color in the code book; this is what we are seeing.

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020
3d-)

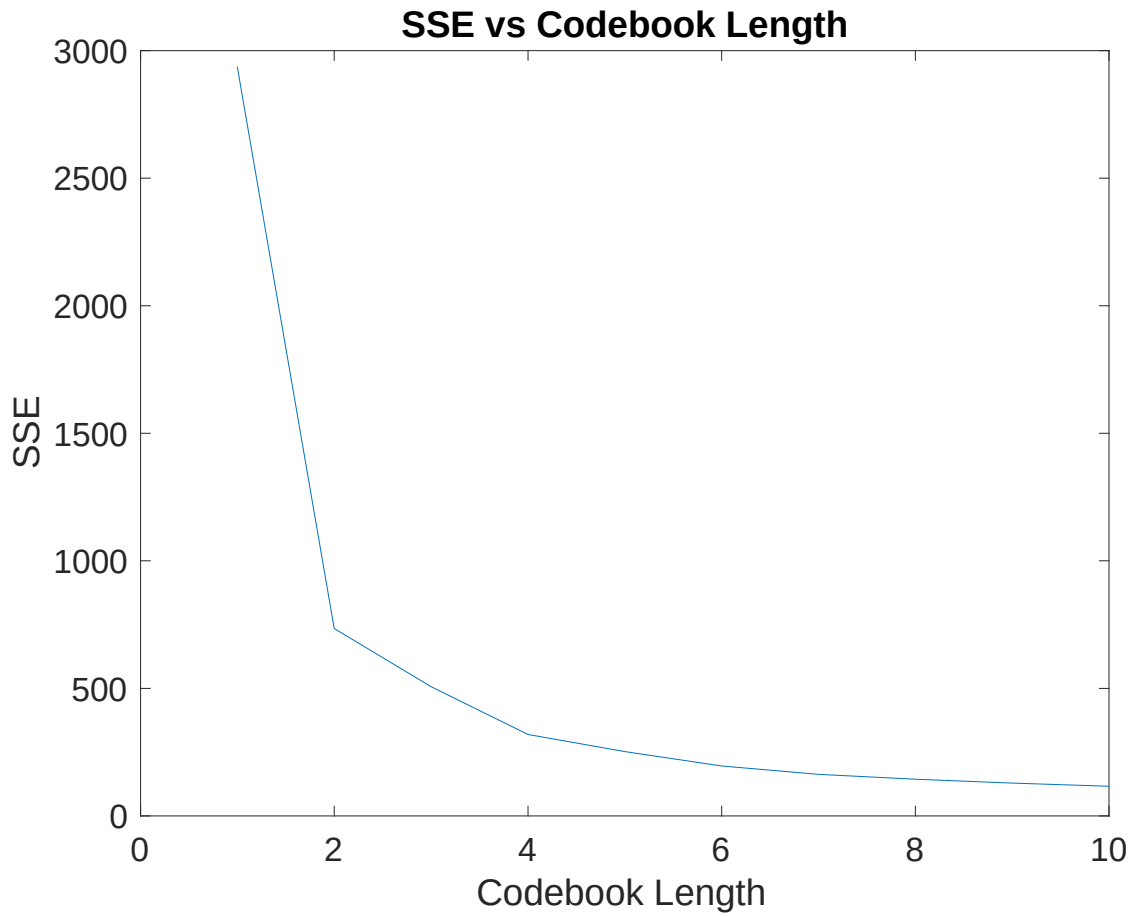

Figure 18: SSE vs K

3e-)
The findings in this problem have taught me that the codebook is simply a collection of pixel values which are used to reconstruct the image. As K gets larger you can get more colors to recreate the picture and thus produce a more accurate reconstruction, that is why in (Figure 18) it is seen that as K increases, the SSE between the reconstructed image to the original image decreases exponentially. This makes sense because you are continuously getting a more accurate image as you go along.

Assignment 3
Mahmood Mustafa Shilleh
04/03/2020

4a-)

I selected features9 in this problem, that are: Manganese, Lycopene, Copper, Thiamin, Pantothenic Acid, Vitamin B6, Riboflavin, Folic Acid, Vitamin E. The process of selecting these is in part C

4b-)

Given as code

4c-)

I initially filled in the missing data by taking the average of each class. I started off my approach with an SFS that used a high dimensional KNN, I got about 68 percent when using a single set as the test and training sets. I decided that I had to use a cross validation to get a more accurate result. I wrote a code that used a 10-fold cross validation as recommended in the notes and I started to get classification rates of 74 percent at best, that is how my features were selected. I did the same thing as but instead I tried a BFS, which happened to perform very poorly with classification rates between 50 and 60 percent, I think this is because the optimal set of features in this problem is small, BFS works better when its large, and it was clear to see that. To summarize this, I used a High Dimensional KNN with SFS and K-fold cross validation. There are definitely more optimal feature selectors, classifiers, and validation techniques, but due to the amount of time I had I was not able to test more.