**Introduction**

For this homework assignment we were given a Principle Component Analysis function. We were first tasked to use it to investigate the reconstruction error and we built a graph of the reconstruction vs. the number of principal components. Next, we replicated the plot of the eigenvector plot shown in the textbook using three dimensions (three eigenvectors). Finally, we were instructed to plot the mean and the 8 principal components as image.

**Methods**

Below are algorithms that are critical to the implementation of the algorithm:

1. function [xd, xx, xxmse, evec, eval, b] = pca(data, class\_labels)
   1. Performs Principal Component Analysis. Returns the set of components of the data in the direction of the eigenvectors (xx), reconstructed dataset (xd), reconstruction error (xxmse), the calculated eigenvectors (evac), eitenvalues (eval), and the mean of the input
2. function plotEval(data)
   1. Plots the Proportion of the variance vs. the eigenvector.
3. function plotPCA(data, class\_label)
   1. Plots the first three eigenvectors on a three-dimensional grid
4. function eigenfaces
   1. Displays the mean of the eigendigits and the first 8 principal components as images.

**Results**

A close up of a map

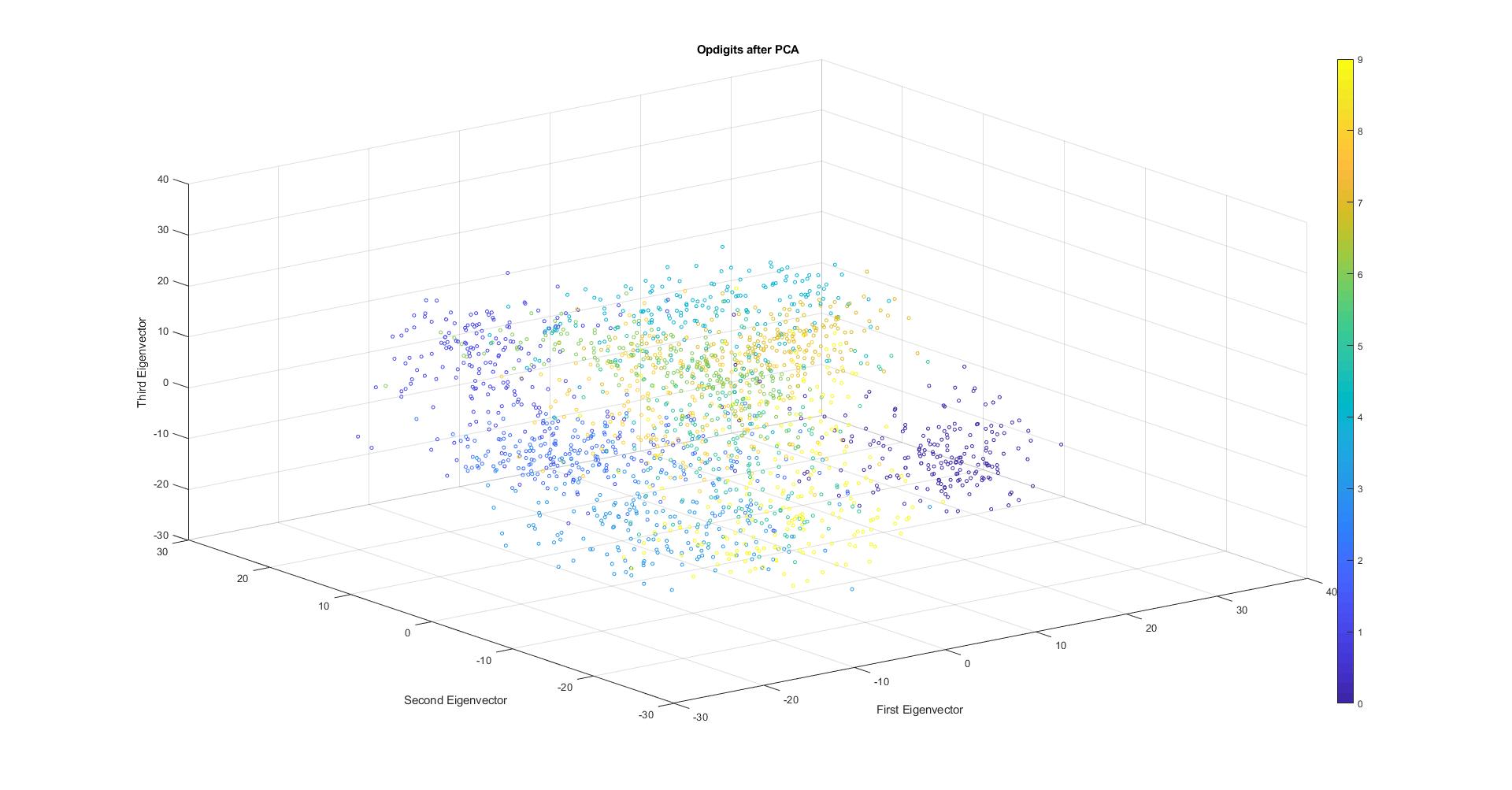
Description automatically generated

**Figure 1**. Reconstruction Error Vs. Eigenvector count

A close up of a map

Description automatically generated

**Figure 2**. Proportion of Variance with Increased Number of Eigenvectors



**Figure 3**. Three-dimensional scatterplot of the first three eigenvectors

A close up of a logo

Description automatically generated

**Figure 4**. The mean of the eigendigits

A screenshot of a cell phone

Description automatically generated

**Figure 5**. First 8 Principle Components Displayed As Images

**Discussion**

In figure 1 you can see the reconstruction error given varying eigenvector counts. According to the textbook, “…if we discard some eigenvectors with nonzero eigenvalues, there will be a reconstruction error and its magnitude will depend on the discarded eigenvalues” (p. 127). The plot in figure 1 follows this trend. As the number of eigenvectors increases, the reconstruction error decreases.

Figure 2 displays the proportion of the variance. This demonstrates how a few of the eigenvectors are responsible for most of the variance. This is especially true for the datasets where the dimensions are highly correlated. The variance is determined by the magnitude of the eigenvalues; therefore, eigenvectors with large eigenvalues a higher variance. If the dimensions are highly correlated, we will be able to make a large reduction in dimensionality (p. 124).

Figure 3 shows the optdigits data plotted in the space of three principal components with the greatest eigenvalues. This allows us to see how well the dimensions grouped the digits. For the most part, the digits are grouped correctly; however, there are a few outliers. This is to be expected since, while the first three principle components are responsible for most of the variance, there are other lesser components that responsible for the variance. As a result, the digits would be grouped more accurately if more principle components were shown.

Figure 4 contains the mean of the eigendigits. Here the eigenvector is displayed as an image. This gives us an idea of what the mean digit looks like. When performing PCA, this mean eigenvector is subtracted from each image. This allows us to get the variance or see how each image differs from the mean.

Figure 5 shows the first eight principle component as images. These eight images can be used to represent the covariance matrix of the optdigits. In other words, any digit in the dataset can be created by combining the principle components. This allows us to perform expensive operations such as face recognition using only a few dimensions.

**Software listing and executable software**

Start the program by pressing “Run”. The program will pause and wait for input after each task.