**Code Documentation:**

Part 1: Basic Alignment

1. First we set a variable to the name of the file we are going to read: file\_name
2. Then we open and read the file using **fopen**
3. We then set the directory of the image files using **dir**
4. Next we read in the landmarks file using **dlmread**
5. **Imagesc** displays matrices as a color image.
6. We then set the colormap to gray using **colormap**
7. Then we created our standard face triangle matrix with three different variables: x, y, z
8. We then put these into one standard face matrix: stdface
9. Next we perform ten transformations for ten images using the information from the muct files. We do this by multiplying the standard face matrix by the inverse of the image matrix.
10. We then set new variables for each image and using **imperspectivewarp** to warp the images to the standard matrix we created.
11. We use **imshow** next to print out the ten images that we transformed.
12. Lastly we close the file by using **fclose**.

Part 2: Least-Squares Alignment

1. First we set a variable to the name of the file we are going to read: file\_name
2. Then we open and read the file using **fopen**
3. Next we read in the landmarks file using **dlmread**
4. We then set the directory of the image files using **dir**
5. Then we created our standard matrix with two different variables: x, y
6. We then put these into one standard matrix: stdface
7. Next we formed the ten matrices for ten different images. We got the values from the muct page. We chose 10 different landmarks to use.
8. We then solve for the unknown U in order to get the least squares solution.

Part 3: Full alignment Morph

1. First we defined a triangulation in the variables: x,y
2. Then we used the **Delaunay** function to get the triangle.
3. We then plot the triangular mesh using **triplot**.
4. Next, we use **tetramesh** to display the tetrahedrons.
5. Then, we defined a standard matrix: stdref.

Part 4: Face Morph

1. First we read in the images that we are going to blend together using **imread**.
2. We use **imfuse** to mesh the pictures together (referenced from <http://www.mathworks.com/help/images/ref/imfuse.htm>) – does not work in Octave
3. **Imwrite** is used to write a matrix into a file.
4. Display the image using **imshow**