Problem Set 3
Assigned: 04/06/2004
Due: 04/22/2004

Please submit an electronic copy of your writeup and code for each problem to `6.891-submit@ai.mit.edu`.

Problem 1 *Structure From Motion*

Figure 1: Input images along with tracked features.

The goal of this problem is to recover the shape of an object (Xerox photocopier) and the movement between each frame. To achieve this goal you will implement the original structure-from-motion algorithm of Tomasi and Kanade.

Displayed in Figure 1 are four input images that we will be used in this problem to recover shape and motion. In the first picture, features have been detected using a sophisticated algorithm. Using a similar algorithm, we tracked these features between images. The MATLAB file `features.mat` contains the results of this tracking.

(a) For each frame, re-center the camera feature points to the object centroid by constraining them to have zero mean.

(b) Write a MATLAB function based on the factorization approach proposed by Tomasi and Kanade to compute the affine motion and the feature shapes given a set of tracked features. Your function should have the following syntax:

$$\texttt{function } [\texttt{A},\texttt{P}] = \texttt{TomasiKanade}(\texttt{D})$$

Test your function using the re-centered features. Re-project the feature shapes P onto the orthographic planes xy, yz, and zx.

(c) As described in Section 12.4.2 of Forsyth and Ponce, some affine transformation Q can be applied to A and P to upgrade from affine to Euclidean. Find the matrix C by imposing the orthographic constraints on the affine motion matrix A. First create an objective function that returns a residual vector F given an estimate of C. Then using the `lsqnonlin` MATLAB function minimize the orthographic constraints.

In our case, what displacement d should we apply?

Apply the *Euclidean upgrade* Q to the affine motion A and the feature shapes P. Re-project the new feature shapes P onto the orthographic planes xy, yz, and zx.

(d) After applying the Euclidean upgrade, the rows of the matrix A should represent the orientations of the horizontal and veritcal camera reference axes. How can you align the first camera reference system with the world reference system? Re-project the aligned feature shapes P.

Problem 2 *Eigenfaces for Recognition*

In this problem we explore the use of eigenfaces for person recognition. Download the MATLAB data file `faces.mat`, which contains the cropped, normalized 51×43 images of 34 individuals. Each individual has been photographed with two facial expressions: "neutral" and "smiling". The pixels of each person's image have been stored as columns of the `neutralFaces` and `smileFaces` matrices. To convert back and forth between image and vector representations, use the `reshape` command.

(a) To (approximately) account for illumination variations, normalize each of the face vectors so that it has zero mean and unit variance.

(b) Assume that we have a database of neutral face images, and want to determine the identity of the smiling individuals. For each of the 34 smiling faces, measure the norm of the difference between their normalized appearance vector and each of the 34 neutral faces. Classify each smiling face according to its nearest neighbor. What percentage of the smiling faces are correctly recognized?

(c) Using the MATLAB `svd` command, determine the principle components of the set of normalized neutral faces (do *not* include the smiling faces). Be sure to subtract the mean face before performing your PCA. Plot the mean face and the first three principle components (the "eigenfaces").

Hint: To avoid excessive memory usage when calculating SVD, use MATLAB's "economy size" option.

(d) Determine the number of principle components required to model 90% of the total variance in the neutral face set. Project each of the neutral and smiling faces onto the corresponding eigenfaces. Use the coefficients of these projections to classify each smiling face. Compute the percentage of correctly recognized faces, and compare to part (b).

(e) Repeat part (d) using the number of principal components required to model 80%, 70%, 60%, and 50% of the total neutral face variance. Plot the corresponding recognition rates.

Problem 3 *Eigenfaces for Detection*

(a) Construct the principle components (eigenfaces) which best model the combined variations of the neutral and smiling faces from Problem 2. Be sure to normalize each face vector as in 2(a). Plot the mean face and first three eignefaces, and compare to 2(c).

(b) Load the test image `mad.png`. Use MATLAB's `ginput` command to (approximately) locate the center's of this image's two (human) faces.

(c) Repeat this part for each of the faces from part (b). For each pixel in an 80×80 pixel square centered around the face, determine the best reconstruction of the 51×43 patch centered at that point using the first 10 principal components from part (a). Each image patch should be normalized as in 2(a) prior to this reconstruction procedure. Plot the distance between the input and reconstructed (normalized) patches as a function of patch location. For the smallest error shift, plot the input and reconstructed image patches.

(d) Repeat part (c), but this time search over the entire image. Make the same pair of plots as before.

(e) Are the eigenfaces suitable for face detection in general scenes? Why or why not? If not, are there controlled situations where they would prove more useful?