HOMEWORK ASSIGNMENT #4
DUE: Tuesday, November 11 2014
CSCI 574: Computer Vision, Prof. Nevatia
Fall Semester, 2014

This assignment is to work through an example of affine reconstruction from multiple images with uncalibrated cameras using the Affine ("Tomasi-Kanade") factorization algorithm (FP Algorithm 8.2). The assignment requires use of a computer and software to compute SVD and multiply matrices; however, you are NOT asked to write a program that takes an arbitrary set of points as input and need not implement a sophisticated user interface. Any method of solving the given numerical problem is sufficient.

The file **point.txt** gives coordinates of 8 points in six images. Assume that the images were acquired from an affine camera. Compute the 3-D positions of these points using the method described in FP Algorithm 8.2. Note that you are not being asked to also compute the Euclidean shape.
You may use any numerical analysis software that you find convenient. OpenCV, does include the needed functions as described below.

1. **SVD::compute**
http://docs.opencv.org/modules/core/doc/operations_on_arrays.html?highlight=svd#svd-compute
The function decomposes matrix A into the product of a diagonal matrix and two orthogonal matrices, and stores the results to the user-provided matrices.

$$\boldsymbol{A = UWV^T}$$

2. **mulTransposed**
http://docs.opencv.org/modules/core/doc/operations_on_arrays.html?highlight=multransposed#multransposed
The function calculates the product of a matrix and its transposition.

$$\text{dst} = \text{scale}(src - delta)(src - delta)^T$$

3. **mul**
[http://docs.opencv.org/modules/core/doc/basic_structures.html?highlight=mul#MatExpr Mat::mul(InputArray m, double scale) const](http://docs.opencv.org/modules/core/doc/basic_structures.html?highlight=mul#MatExpr)
This function performs an element-wise multiplication or division of the two matrices.

Suppose that the points above have the following connectivity pattern: (1, 2), (2, 3), (3, 4), (4, 1), (5, 6), (6, 7), (7, 8), (8, 5), (1, 5), (2, 6), (3, 7) and (4, 8). This should allow you to visualize the reconstructed points as forming a wire-frame object.

The point images given in this example are synthetic projections of a rectangular solid by a perspective camera with a long focal length and viewing directions and scale. Verify that the computed reconstruction is, in fact, approximately an affine transformation of a rectangular solid (*i.e.* parallel lines remain parallel).

What to Submit?

You should turn in the following (electronic submissions only):

1. A brief description of any programs you write
2. The source listing.
3. Numerical results of the 3-D positions of the eight points. Also include intermediate results where applicable.
4. A visualization of the resulting object. You need not construct an elaborate graphics program for this: easiest may be to orthographically project the object from some non-accidental viewpoint (*i.e.* where viewing direction is not parallel to any object lines).
5. A verification of whether the reconstruction is approximately an affine transform of a rectangular solid or not.