Comp. 4900C: Assignment #3
Due: March 19, 2008

The goal of this assignment is to implement some code that performs calibration using the two methods described in the Trucco and Verri. You can do this assignment in any language you like, but I believe that using Ch is the easiest solution for two reasons; first it is easy to manipulate arrays and it has built in math routines, and second I give you the code for the main routine and some subroutines in Ch, and your task is to write the code for a number of subroutines.

You are given the routines to create an R, T and K matrix, and some random 3d points. These are the routines named eulerangletorotmatrix, setkmatrix, settrans, and computerandom3dpoints and the main program itself. You need to create a projection matrix from the given R, T and K, and use this projection matrix to create a new set of projected 2d points from the given 3d points. Then with these projected 2d points as input use the Tsai calibration approach and the Projection Calibration approach to compute the intrinsic and extrinsic parameters.

Since you know the original intrinsic and extrinsic parameters, the ones you extract from the 2d projections should be the same parameters as the ones used to create the 2d data in the first place. You need to use the eigenvector or SVD routine of Ch, so I on the web site is an example of the eigenvector routine, called eigenvector-2.ch.

First install Ch on your computer, and then ChScite. The ChScite editor allows you to edit Ch programs, and to execute them using the run command in the tools menu. Ch and ChsCite are on the CD that I gave you at the beginning of the year, and also on the class web site.

The first thing to do after installing Ch and ChScite is to run eigenvector-2.ch. This example uses the same matrix that was discussed in the lecture on the review of linear algebra. Notice that the eigenvectors are returned in column order, and not row order.

Your job is to write the following routines:

```
/* take the R, T, and K and create the projection matrix P as output */
void composeprojectionmatrix(array double r[3][3], array double t[3], array double
K[3][3], array double P[3][4])

/* apply the projection matrix to the 3d points to get a set of 2d feature points as output */
void applyprojectionmatrix(array double threedpoints[NUM_FEATURES][3],
                           array double twodprojections[NUM_FEATURES][2],
                           array double projectionmatrix[3][4])
```

```
/* given a set of correspondences from 3d to 2d points compute the projection matrix
   that produces these correspondence values. Output is projectionmatrix */
   Use the method described in 6.3 of Trucco and Verri */
void computeprojectionmatrix(array double threedpoints[NUM_FEATURES][3],
                             array double twodprojections[NUM_FEATURES][2],
                             array double projectionmatrix[3][4])


/* take the projection matrix and decompose to find the rotation, translation and
calibration. taken from 6.3.2 of Trucco and Verri, Output is r, t, and K */
void decomposeprojectionmatrix(array double P[3][4], array double r[3][3], array double
t[3], array double K[3][3])


/* given a set of correspondences from 3d to 2d compute the aspect ratio,
   which is fx/fy and the translation in x, and y. Just print out these three values.
   using the method described in 6.2.2 of Trucco and Verri */
void computetsaicalibration(array double threedpoints[NUM_FEATURES][3],
                            array double twodprojections[NUM_FEATURES][2],
                            array double r[3][3], array double t[3], array double K[3][3])
```

You just need to run the programs with print statements enclosed. You should see from the print statements that the original R, K and t should match the computed R, K and t. For the Tsai approach you need only match and print tx, ty, and the aspect ratio fx/fy. If the signs are different for the Tsai approach that is ok, in other words by match I mean within a sign flip.

Hand in the Ch program, so the TA can simply run your program to see that it works.