Emailed October 5, 2000
Due back October 11 2000

Camera calibration using a linear method

We want to calibrate the camera of a robot vehicle. We place a large cubic frame of size 4 meters on the road several meters in front of the vehicle. The positions of the eight corners of the cubic frame are defined with respect to a world coordinate system with its axes parallel to the cube edges and with its origin at the center of the cube. The world coordinates of the cube vertices are

```
    2       2       2
   -2       2       2
   -2       2      -2
    2       2      -2
    2      -2       2
   -2      -2       2
   -2      -2      -2
    2      -2      -2
```

There are bright light sources at the corners of the cube that are all visible and are easy to detect in the camera image.

We detect the cube corners at the following pixel positions in the camera image:

```
   422     323
   178     323
   118     483
   482     483
   438      73
   162      73
    78     117
   522     117
```

For each image point, the left coordinate defines its horizontal position (pixel column) and the right coordinate defines its vertical position. We were able to find out the correspondences between image points and world cube corners, so that the first set of image coordinates corresponds to the first set of world coordinates, and so on.

1. Draw the image points, using small circles for each image point.

2. Write a Matlab function that takes as argument the homogeneous coordinates of one cube corner and the homogeneous coordinates of its image, and returns 3 rows of the matrix A (slide 25 and slide 26 of the Camera Calibration pdf document). This matrix A will be used to compute the 12 elements of the projection matrix P.

NOTE: The slides 25 and 26 showed zero vectors with 3 elements (03) in matrix A. This was wrong. The zero vectors should have 4 elements (04). This has been corrected (but your browser may still be caching the older version).

3. Use this Matlab function to generate 3 rows of the matrix A for each cube corner and its image and obtain a matrix with 24 rows and 12 columns. Print matrix A.

4. Find the singular value decomposition of matrix A, `[U,S,V] =svd(A).`
The 12 elements of the last column of vector V obtained by svd(A) are the 12 elements in row order of the projection matrix that transformed the cube corner coordinates into their images (slide 27). Print matrix P.

5. Find the singular value decomposition of matrix P, `[U2,S2,V2] = svd(P).`
The 4 elements of the last column of vector V2 are the homogeneous coordinates of the position of the camera center of projection in the frame of reference of the cube (slide 21). Print the corresponding 3 Euclidean coordinates of the camera center in the frame of reference of the cube.

6. Consider the 3x3 matrix M composed of the first 3 columns of matrix P.
Rescale the elements of this matrix so that its element m33 becomes equal to 1.
Print matrix M.

7. Perform the RQ factorization of M in several steps. First, find a rotation matrix Rx (slide 23) that sets the term at position (3,2) to zero when multiplied to M. The cosine and sine used in this matrix are of the form

```
cosX = m33/sqrt(m33*m33+m32*m32)

sinX = -m32/sqrt(m33*m33+m32*m32)
```

Note that the term at position (3,2) would also be set to zero if the signs of `cosX` and `sinX` were reversed (as shown in the slides), but this would lead to finding a negative focal length for the camera.

Compute the angle ax of this rotation in degrees. Compute matrix `N = M * Rx.`
Print `Rx, ax` and N.

8. The element n31 of N is small enough so that there is no need for a rotation Ry. However, element n21 is large and a rotation matrix Rz is needed to set it to zero. Compute the rotation matrix Rz using cosine and sine of the form

```
cosZ = n22/sqrt(n21*n21+n22*n22)

sinZ = -n21/sqrt(n21*n21+n22*n22)
```

Compute the rotation angle az in degrees. This angle is actually very small.
Compute the calibration matrix K = N * Rz.

Rescale K so that its element k33 is set to 1.
Print K.

What are the focal lengths of the camera in pixels?
What are the pixel coordinates of the image center of the camera?