

## Assignment 2 Mobile Robotics:

Camera Calibration : Finding Intrinsic Parameters of the camera using zhang's method :-

PATH to webcam images

1.4 non-colinear points needed for estimating homography H for each image – taken 4 corner points.

2. exploiting properties of rotation matrix zhangs gives relation between  $\text{inv}(K)'\text{inv}(K)$  and elements of H.

3. b has six elements but each H gives two equations in six elements of b .So atleast 3 homographies needed.

4. Homography of 1,3,5 images taken-

`detectCheckerBoardPoints()` - function used. This gives 48x2x5 matrix of image points .`worldPoints` – this matrix contains corresponding world points same for every image . The corner points are – 1,6,43,48 – rows of `imagePoints`.

To obtain H – take svd of a matrix made of rows of the form  $(-X, -Y, 0, 0, \dots)$  .

For each H find  $v_{12}, v_{11}$  and  $v_{22}$  and concatenate them vertically, take svd on this matrix ,the last column vector of V give the vector b.

B is constructed from b .

take  $B = (B+B')/2$ ;

it is not positive semidefinite – run a loop for 1000 times with if -else blocks to check if the eigenvalue smallest of B is negative or zero add identity multiplied by  $\text{eigenvalue}^2$  or small positive respectively.

Do `chol(B)`

the matrix it returns is K inverse

invert it this is K.

BONUS 1: multiply each homography (also the two not used) with k inverse . Take the cross product (using cross function) of first two columns to obtain a vector normal to both, normalize these (using norm –  $r1 = r1/\text{norm}(r1)$ );

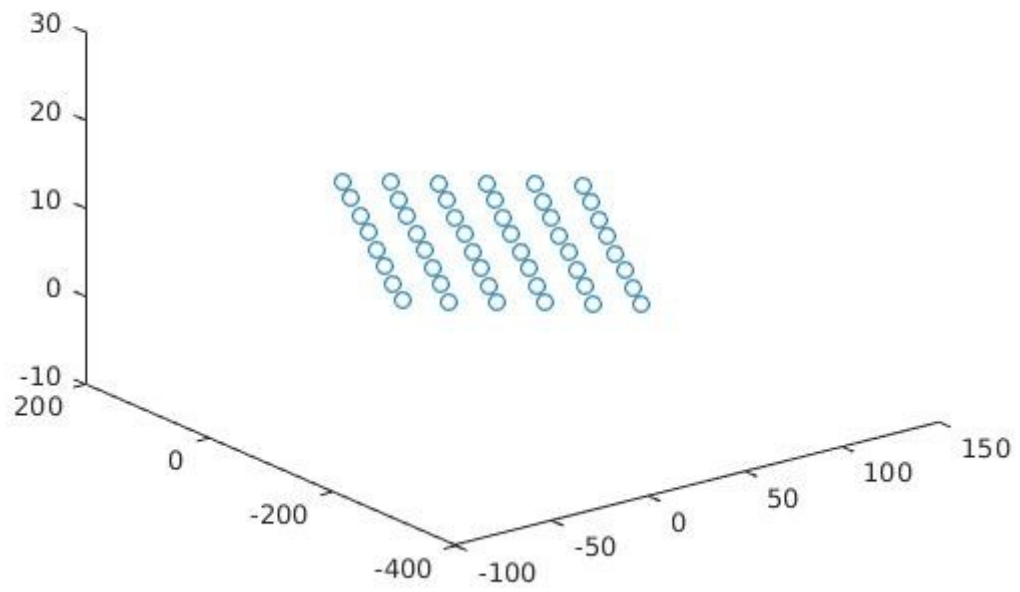
concatenate into one matrix this is the rotation matrix for that homography .normalize the third column of h after multiplication with K inverse this gives translation vector.

Check if rotation matrix proper – dot product ( using dot function) gives zero for every column pair of R and norm of every column is 1.  $\det(R)$  is also one .Values obtained for all five Rs quite close to 1 or 0 respectively.

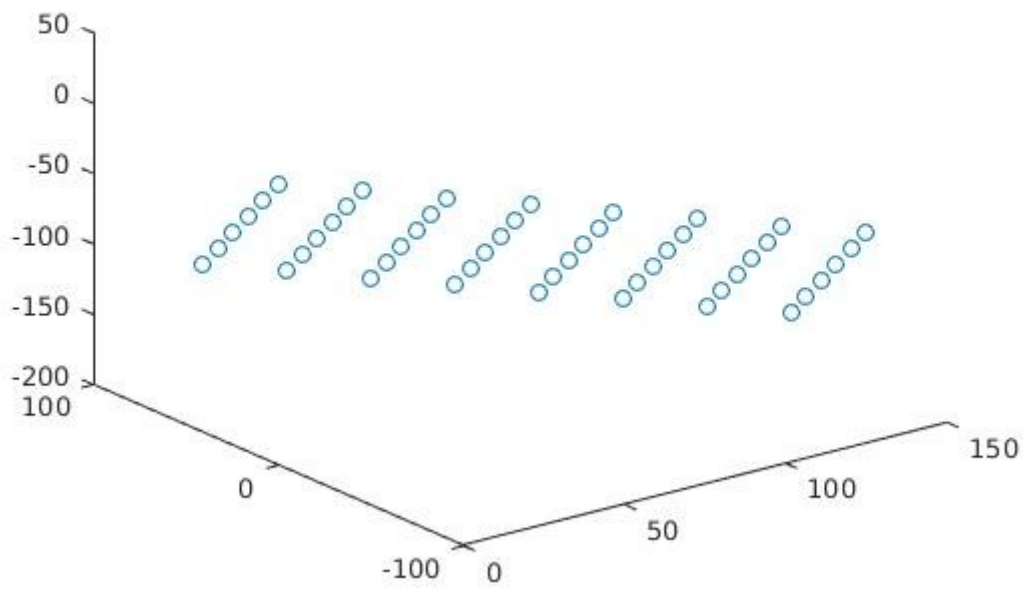
Obtain camera coordinates as -

`camcord = [ R t; 0 0 0 1]*Worldcoords;`

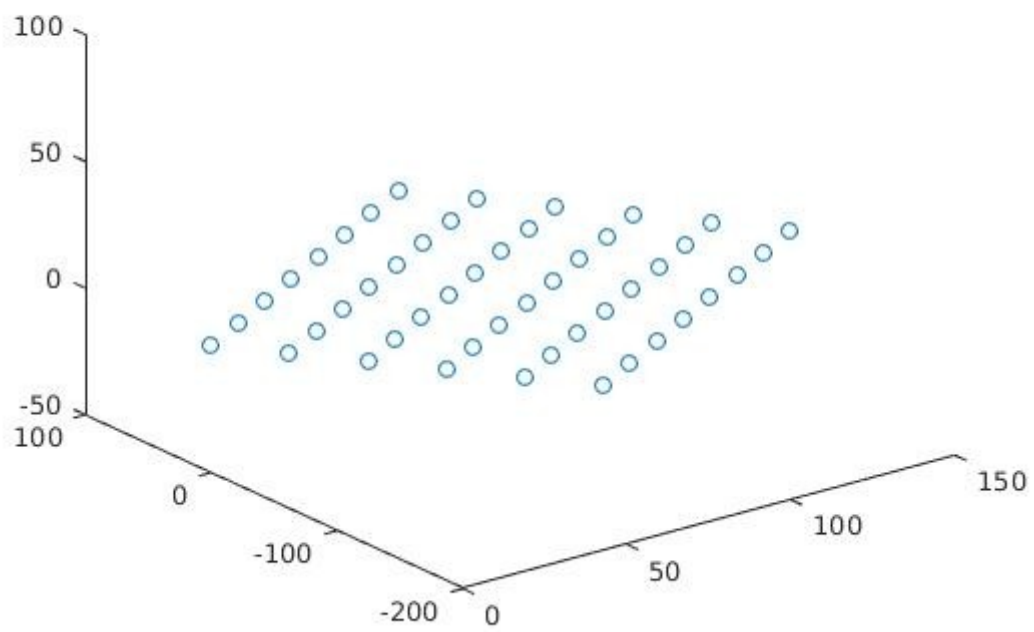
plot in 3d camera coords -x,y,z – they are rectangular in shape. and size/number of points equal to checker board.



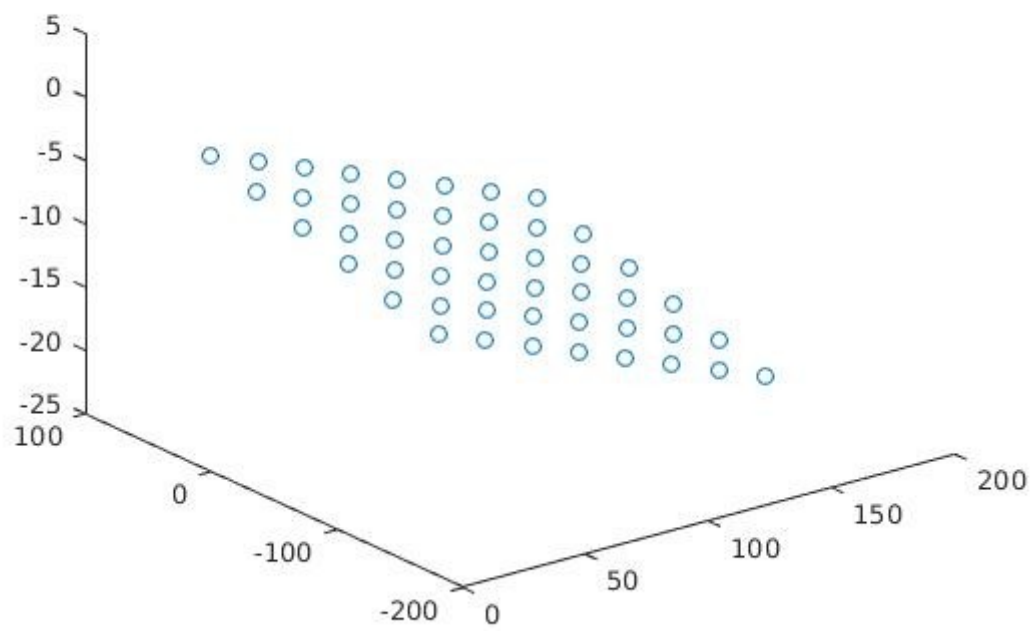
Corresponding to Img1



to img3



to img 4



img5