REPORT

Camera Calibration

Steps:

1. Convert world coordinate 27 X 16.5 cm to pixel using 1 cm = 37.795 pixel
2. Generate Homography Matrix between each image with world coordinate. Homography2d.m is a function for homography. Here each input image is used to compute a separate homography.
3. Compute K matrix by converting Homography matrix to vertex then to symmentric Positive matrix i.e. B = $\begin{matrix} B11 & B12 & B13 \\ B12 & B22 & B23 \\ B13 & B23 & B33 \end{matrix}$ B=K^{-T}*K^{-1} from which calibration matrix K has been recovered using matrix square root and inversion. Matrix B is the image of absolute conic in projective geometry

4. K= $\begin{matrix} \alpha & \gamma & u \\ 0 & \beta & v \\ 0 & 0 & 1 \end{matrix}$ is calculated using matrix B

5. Compute R (Rotation matrix) and t (translation matrix) using Homography and K matrix .

6. Rotation matrix is orthogonal, but after computing R from H and K it is not orthogonal. So AlternateR function is used to make it orthogonal.

7. Compute camera position using R and T since C=-R’*t. MATLAB has “ extrinsicsToCameraPose” function to compute Camera Position for each image.

Result:

```
R_better_2 =

   0.0205   -0.0879   -0.9959
   0.9985   -0.0487    0.0249
   0.0507    0.9949   -0.0868

t_2 =

  1.0e+03 *

   0.6012
  -0.4156
   1.8737

R_better_3 =

   0.0994   -0.0494   -0.9938
   0.9236   -0.3671    0.1106
   0.3703    0.9288   -0.0092
```

```
t_3 =

   1.0e+03 *

    0.6775
   -0.1497
    1.8919

R_better_4 =

    0.0922   -0.0252    0.9954
   -0.9402    0.3269    0.0954
    0.3278    0.9447   -0.0064

t_4 =

   1.0e+03 *

    0.4202
   -0.4992
    1.5181


 camera_pose_2 =

   1.0e+03 *

    1.8172   -0.6671    0.5457

camera_pose_3 =

   1.0e+03 *

    1.8055   -0.8899   -0.0945

camera_pose_4 =

   1.0e+03 *

   -1.5625    0.4134    0.3436
```

Accuracy:

The results are not completely accurate since value of u and v in H matrix should be ideally W/2 and H/2 of image but after calculation it is more than that also B is not completely positive value as it should be, therefore computed camera position is nearly accurate.

Reference:

1. Textbook “Computer Vision” by Richard Szeliski
2. reference: http://ais.informatik.uni-freiburg.de/teaching/ws11/robotics2/pdfs/rob2-08-camera-calibration.pdf