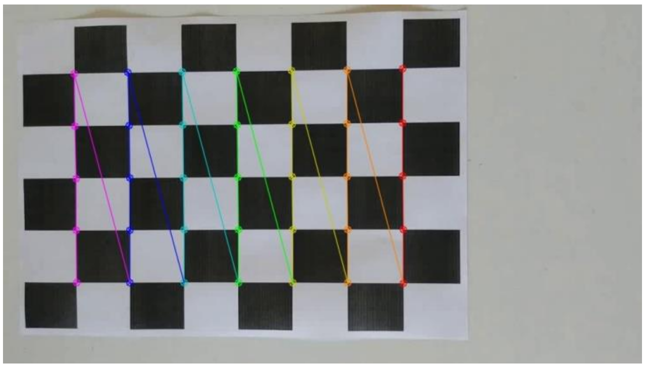
MACHINE VISION ASSIGNMENT 2

Note: Please refer to the output.txt file for the outputs generated from the execution of the python file. Also see the output folder.

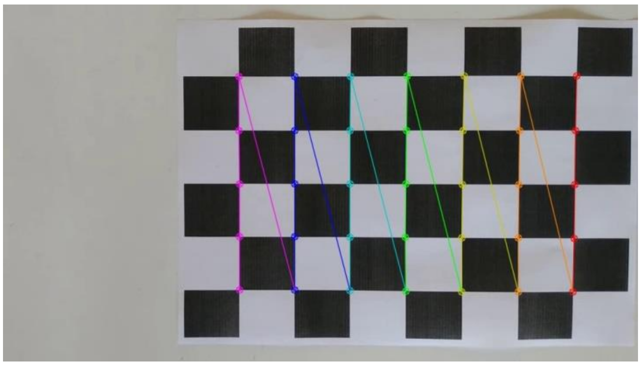
TASK 1 (pre-processing, 8 *points*)

Part (A): Download the file Assignment_MV_02_calibration.zip from Canvas and load all calibration images contained in this archive. Extract and display the checkerboard corners to subpixel accuracy in all images using the OpenCV calibration tools [3 points].

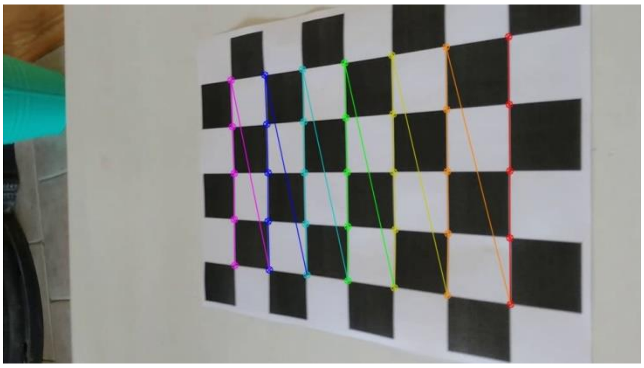
Solution (A): There was no pattern found for image 5 so we could not display the corners on that image. Below are the 6 images of checkerboard corners displayed to the sub-pixel level of accuracy.



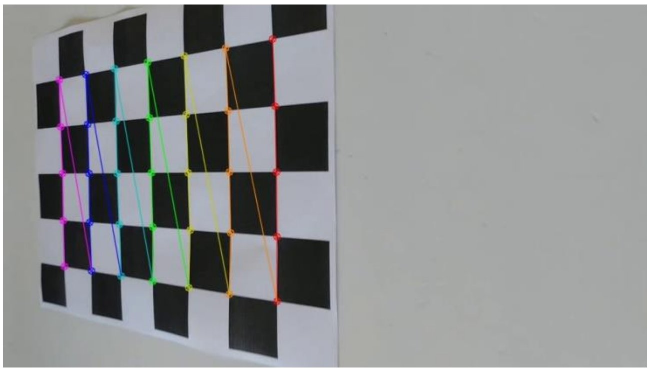
Corner Image 1



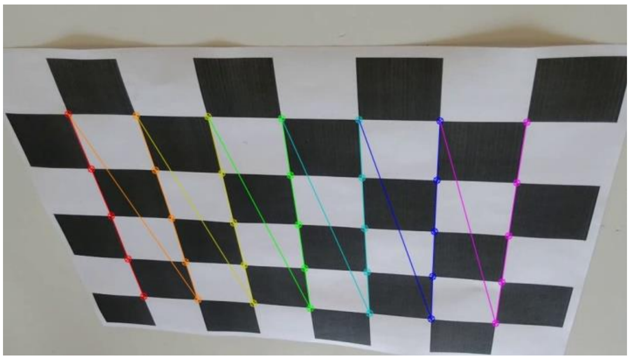
Corner Image 2



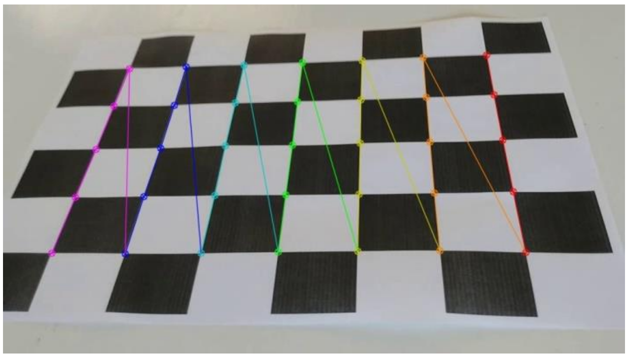
Corner Image 3



Corner Image 4



Corner Image 6



Corner Image 7

Part (B): Determine and output the camera calibration matrix $\boldsymbol{K}$ using the OpenCV calibration tools [1 point].

Solution (B): Below are the calibration matrices of each of the above 6 images.

```
Displaying corners for image 1.
press [ENTER] to print the matrix
Calibration matrix for image 1 :
[[2.99154501e+03 0.00000000e+00 3.58976138e+02]
 [0.00000000e+00 2.87743755e+03 6.35171443e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

```
Displaying corners for image 2.
press [ENTER] to print the matrix
Calibration matrix for image 2 :
[[4.80481114e+03 0.00000000e+00 3.00860407e+02]
 [0.00000000e+00 4.68566623e+03 2.62222745e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

```
Displaying corners for image 3.
press [ENTER] to print the matrix
Calibration matrix for image 3 :
[[1.14551835e+03 0.00000000e+00 1.98678425e+02]
 [0.00000000e+00 1.67348813e+03 4.38425262e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

```
Displaying corners for image 4.
press [ENTER] to print the matrix
Calibration matrix for image 4 :
[[1.81727910e+03 0.00000000e+00 2.18767424e+02]
 [0.00000000e+00 1.27614023e+03 4.96583778e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

```
No pattern found for image 5.


Displaying corners for image 6.
press [ENTER] to print the matrix
Calibration matrix for image 6 :
[[1.36579119e+03 0.00000000e+00 2.94398946e+02]
 [0.00000000e+00 1.67610958e+03 5.07682383e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

```
Displaying corners for image 7.
press [ENTER] to print the matrix
Calibration matrix for image 7 :
[[1.38409653e+03 0.00000000e+00 3.75904813e+02]
 [0.00000000e+00 2.66381838e+03 4.46710986e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

Part (C): Download the file Assignment_MV_02_video.mp4 from Canvas and open it for processing. Identify good features to track in the first frame [1 point] using the OpenCV feature extraction and tracking functions. Refine the feature point coordinates to sub-pixel accuracy [1 point].

Solution (C): Below is the image of first frame with the mark of good features.

Part (D): Use the OpenCV implementation of the KLT algorithm to track these features across the whole image sequence [1 point]. Make sure to refine the feature point coordinates to sub-pixel accuracy [1 point] in each step.

Solution (D): We have saved the video in the output folder in the zip file that shows the application of KLT algorithm to track these detected features across the whole image sequence and we have also refined the feature point co-ordinates to sub-pixel accuracy.

TASK 2 (Fundamental matrix, 24 points)

A. Extract and visualise the feature tracks calculated in task 1 which are visible in both the first and the last frame to establish correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i$ between the two images [2 points]. Use Euclidean normalised homogeneous vectors.

Solution (A): The output of this task is displayed but refer to the code and the **output.txt** file

B. Calculate the mean feature coordinates $\boldsymbol{\mu} = \frac{1}{N}\sum_i \boldsymbol{x}_i$ and $\boldsymbol{\mu}' = \frac{1}{N}\sum_i \boldsymbol{x}_i{}'$ in the first and the last frame [2 points]. Also calculate the corresponding standard deviations $\sigma = \sqrt{\frac{1}{N}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})^2}$ and $\sigma' = \sqrt{\frac{1}{N}\sum_i (\boldsymbol{x}'_i - \boldsymbol{\mu}')^2}$ (where $(\,)^2$ denotes the element-wise square) [2 points]. Normalise all feature coordinates and work with $\boldsymbol{y}_i = \boldsymbol{T}\boldsymbol{x}_i$ and $\boldsymbol{y}'_i = \boldsymbol{T}'\boldsymbol{x}'_i$ which are translated and scaled using the homographies [2 points]

$$T = \begin{pmatrix} \frac{1}{\sigma_x} & 0 & -\mu_x/\sigma_x \\ 0 & \frac{1}{\sigma_y} & -\mu_y/\sigma_y \\ 0 & 0 & 1 \end{pmatrix} \qquad T' = \begin{pmatrix} \frac{1}{\sigma'_x} & 0 & -\mu'_x/\sigma'_x \\ 0 & \frac{1}{\sigma'_y} & -\mu'_y/\sigma'_y \\ 0 & 0 & 1 \end{pmatrix}$$

Solution (B): For the mean and standard deviations, please refer to the **output.txt** file and the python console after running the python file (please note that the output in the txt file is recorded with the results at the time of execution so when the program will be executed again the output in the console will vary). For the normalisation of the feature co-ordinates, please refer the python code file.

C. Select eight feature correspondences at random [1 point] and build a matrix comprising the eight corresponding rows $\boldsymbol{a}_i^T = \boldsymbol{y}_i^T \otimes \boldsymbol{y}_i{}'$ to calculate the fundamental matrix using the 8-point DLT algorithm [1 point].

Solution (C): As evident from the code, we have selected the eight correspondences at random. The fundamental matrix is displayed in **output.txt** file and is displayed in console.

D. Use the 8-point DLT algorithm to calculate the fundamental matrix $\widehat{\boldsymbol{F}}$ for the eight selected normalised correspondences $\boldsymbol{y}_i \leftrightarrow \boldsymbol{y}_i'$ [1 point]. Make sure that $\widehat{\boldsymbol{F}}$ is singular [1 point]. Apply the normalisation homographies to $\widehat{\boldsymbol{F}}$ to obtain the fundamental matrix $\boldsymbol{F} = \boldsymbol{T}'^T \widehat{\boldsymbol{F}}\, \boldsymbol{T}$ [1 point].

Solution (D): For the output of this part, please refer the **output.txt** file and the code file.

E. For the remaining feature correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i'$ not used in the 8-point algorithm calculate the value of the model equation [1 point]

$$g_i = \boldsymbol{x}'^T_i \boldsymbol{F} \boldsymbol{x}_i$$

Also calculate the variance of the model equation [1 point]

$$\sigma_i^2 = \boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{C}_{xx} \boldsymbol{F}^T \boldsymbol{x}_i' + \boldsymbol{x}_i^T \boldsymbol{F}^T \boldsymbol{C}_{xx} \boldsymbol{F} \boldsymbol{x}_i$$

Use the following point observation covariance matrix of the homogeneous features

$$\boldsymbol{C}_{xx} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Solution (E): For the output of this part, please refer the **output.txt** file and the code file.

F. Determine for each of these correspondences if they are an outlier with respect to the selection of the eight points or not by calculating the test statistic [1 point]

$$T_i = \frac{g_i^2}{\sigma_i^2}$$

Use an outlier threshold of $T_i > 6.635$ [1 point]. Sum up the test statistics over all inliers [1 point].

Solution (F): For the output of this part, please refer the **output.txt** file and the code file.

G. Repeat the above procedure 10000 times for different random selections of correspondences [1 point]. Select the fundamental matrix and remove all outliers for the selection of eight points which yielded the least number of outliers [1 point]. Break ties by looking at the sum of the test statistic over the inliers [1 point].

Solution (G): For the output of this part, please refer the **output.txt** file and the code file.

H. Adapt the display of feature tracks implemented in subtask A to indicate which of these tracks are inliers and which tracks are outliers [1 point]. Also calculate and output the coordinates of the two epipoles? [2 points]

Solution (H): For the output of this part, please refer the **output.txt** file and the code file.

TASK 3 (Essential matrix and 3d points, *18 points*)

A. Use the fundamental matrix $\boldsymbol{F}$ determined in task 2 and the calibration matrix $\boldsymbol{K}$ determined in task 1 to calculate the essential matrix $\boldsymbol{E}$ [1 point]. Make sure that the non-zero singular values of $\boldsymbol{E}$ are identical [1 point]. Also make sure that the rotation matrices of the singular value decomposition have positive determinants [1 point].

Solution (A): Please refer to the output console after running the code file and also the **output.txt** file to see the output of this part.

B. Determine the four potential combinations of rotation matrices $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ between the first and the last frame [3 points]. Assume the camera was moving at 50km/h and that the video was taken at 30fps to determine the scale of the baseline $\boldsymbol{t}$ in meters [1 point].

Solution (B): Please refer to the output console after running the code file and also the **output.txt** file to see the output of this part.

C. Calculate for each inlier feature correspondence determined in task 2 and each potential solution calculated in the previous subtask the directions $\boldsymbol{m}$ and $\boldsymbol{m}'$ of the 3d lines originating from the centre of projection towards the 3d points [1 point]

$$\boldsymbol{X}[\lambda] = \lambda \boldsymbol{m}$$

and

$$\boldsymbol{X}[\mu] = \boldsymbol{t} + \mu \boldsymbol{R}\boldsymbol{m}'$$

Then calculate the unknown distances λ and μ by solving the linear equation system [1 point]

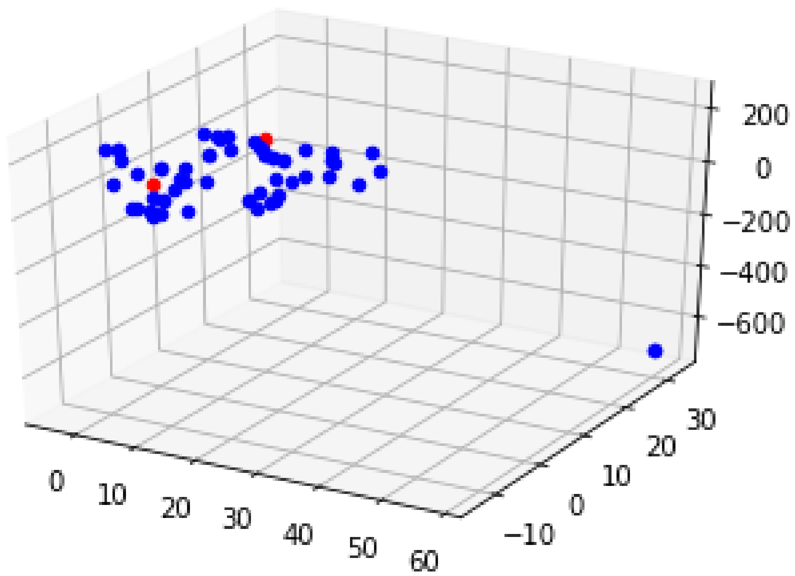
$$\begin{pmatrix} \boldsymbol{m}^T\boldsymbol{m} & -\boldsymbol{m}^T\boldsymbol{R}\boldsymbol{m}' \\ \boldsymbol{m}^T\boldsymbol{R}\boldsymbol{m}' & -\boldsymbol{m}'^T\boldsymbol{m}' \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} \boldsymbol{t}^T\boldsymbol{m} \\ \boldsymbol{t}^T\boldsymbol{R}\boldsymbol{m}' \end{pmatrix}$$

to obtain the 3d coordinates of the scene points [1 point]. Determine which of the four solutions calculated in the previous subtask is correct by selecting the one where most of the scene points are in front of both frames, i.e. where both distances $\lambda > 0$ and $\mu > 0$ [1 point]. Discard all points, which are behind either of the frames for this solution as outliers [1 point].

Solution (C): Please refer to the output console after running the code file and also the **output.txt** file to see the output of this part.

D. Create a 3d plot to show the two camera centres and all 3d points [2 points].

Solution (D): Please see the below 3D plot generated where red points are camera centres and blue points represents the 3D points.



E. Project the 3d points into the first and the last frame [2 points] and display their position in relation to the corresponding features to visualise the reprojection error [2 points].

Solution (E): The outputs for this part are depicted below