CS763 - Assignment 2

150050108 150050029 150070001

February 8, 2019

Problem 1

We work with exactly $N = 6$ points. Our camera calibration gives an $RMSE = 0.0097$.
The camera characteristics found are :

$$\hat{P} = \begin{bmatrix} 0.1338 & -0.3320 & 0.4064 & -0.0089 \\ -0.4983 & -0.2238 & -0.0221 & 0.0008 \\ -0.0287 & 0.0493 & 0.0516 & -0.6339 \end{bmatrix} \quad P = 1.0e+03 * \begin{bmatrix} 0.0022 & -0.0243 & 0.1262 & -1.5938 \\ -0.1120 & -0.0248 & 0.0166 & -0.2056 \\ -0.0000 & 0.0000 & 0.0000 & -0.0007 \end{bmatrix}$$

$$K = \begin{bmatrix} -108.9012 & -0.5936 & 68.2000 \\ 0 & 109.8376 & 36.9415 \\ 0 & 0 & 0.0312 \end{bmatrix} \quad R = \begin{bmatrix} -0.2489 & 0.6270 & -0.7382 \\ -0.8940 & -0.4419 & -0.0739 \\ -0.3726 & 0.6416 & 0.6705 \end{bmatrix} \quad X_0 = \begin{bmatrix} -3.0806 \\ 16.2032 \\ 15.8035 \end{bmatrix}$$

The reason for normalization of points is that doing so can avoid numerical precision issues. DLT method can work with arbitrary coordinate systems in both the real world and the image plane. Therefore, it can be the case that there is a high magnitude scaling factor between the two coordinate systems we adopt, and that can lead to magnification of numerical errors in the DLT algorithm, specially because we use involved internal routines like SVD. Because of this, it is advisable to handle such scale factors in pre-processing, and coordinate normalization is one way to do so. Note that the scale is indeed very different in the coordinate system we adopt. In the real world, unit dimension is chosen to be the side length of a single square. And in the real world, we work with pixel coordinates. There is an order of 1000 magnitude difference between these coordinate system, which comes up in the output also as we can see that the P matrix is about 1000 times order of magnitude higher than the $\hat{P}$ matrix.

The image is shown here. The coordinates of the 6 points (in the real world) are also marked. The corresponding 2d coordinates in the image plane are found using imtool(), as mentioned in the Q1/points_2d.m

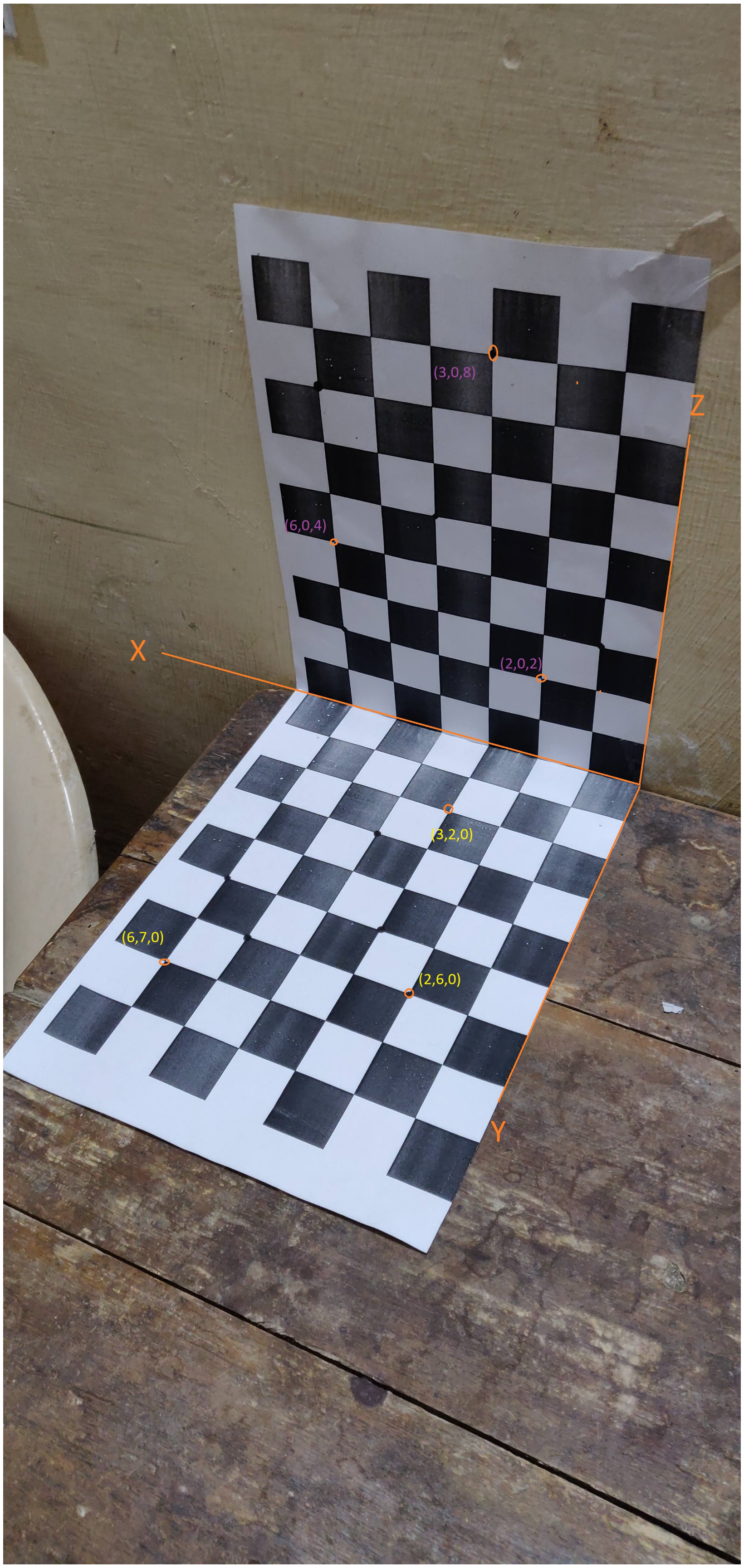

Figure 1: Clicker Checkerbox with 6 selected points and their coordinates in the real world

Problem 2

First we find the pixel points corresponding to the outer Dee (box) in the image (marked in red). Using this points, we find the 2D homography matrix which can transform the outer box to a rectangle of size $18yd \times 44yd$ (scaled in the code). Then using this homography matrix, we transform 3 points (marked in black) that correspond to the playing area. Using the transformed points we calculate the field dimensions which come out to be $125yd \times 74yd$. All the points marked and other data required (dimensions of the outer box) is saved in `Q2/input/Q2data.mat`.

Selected points for estimating homography and field dimensions

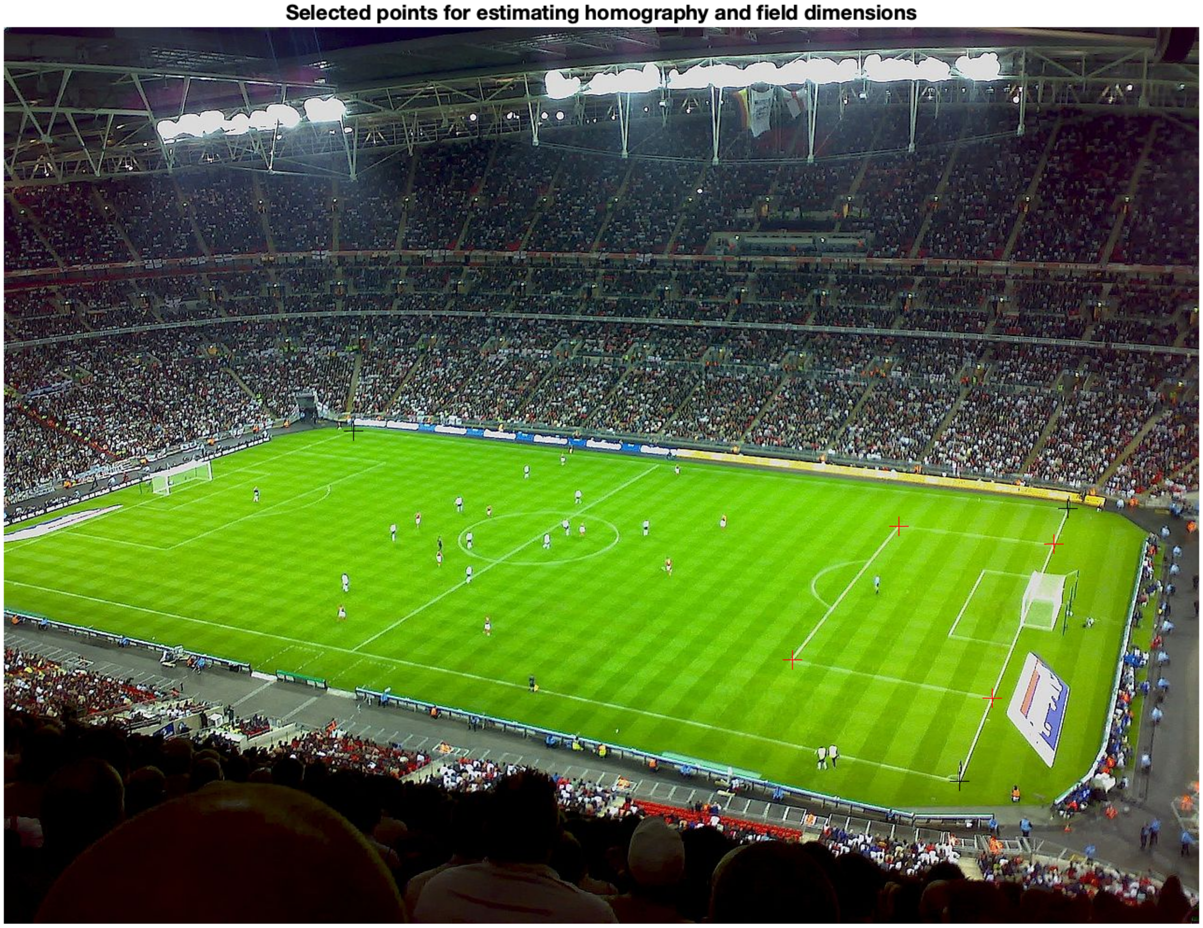

Figure 2: Marked points in the image

Problem 3

We use the property of pinhole projection that cross ratios remain invariant under projective tranformations. Consider the right baseline of the field which has been marked in the figure 3. We can mark four points on this line A,B,C,D starting from the top. So the cross ratio formed by these four collinear points will remain invariant. Thus, $\frac{AC \times BD}{BC \times AD} = \frac{A^{'}C^{'} \times B^{'}D^{'}}{B^{'}C^{'} \times A^{'}D^{'}}$, where ABCD is the line in actual physical distances and $A^{'}B^{'}C^{'}D^{'}$ is the one seen in image. The measurements (made through imtool Matlab) have been marked on the image. Let the distance $AB = CD = x$ yards in real life (football field is symmetric). This gives us $\frac{(44+x)^2}{44(44+2x)} = \frac{217.7 \times 274.8}{177 \times 315.5} \implies x = 15.3$. Hence we get the breadth of the football field is $44 + 2x = 74.6yd$

Similarly consider the line joining the shorter line of the outer D to the D on the other side as marked in the image. We know that these lie on a straight line in the physical world and thus will still be on a line as it is a straight line preserving transformation. Considering the points as ABCD from left to right, invariance of cross ratios gives us $\frac{AC \times BD}{BC \times AD} = \frac{A^{'}C^{'} \times B^{'}D^{'}}{B^{'}C^{'} \times A^{'}D^{'}}$. Assume $BC = y$ yards in real life. Then $\frac{(18+y)^2}{y(36+y)} = \frac{789.4 \times 894.7}{678.2 \times 1005.9} \implies y = 79.76$. Hence the length of the football field is $18 * 2 + x = 115.76yd$

Hence the size of the football field as found from the cross-ratio property is $115.76yd \times 74.6yd$

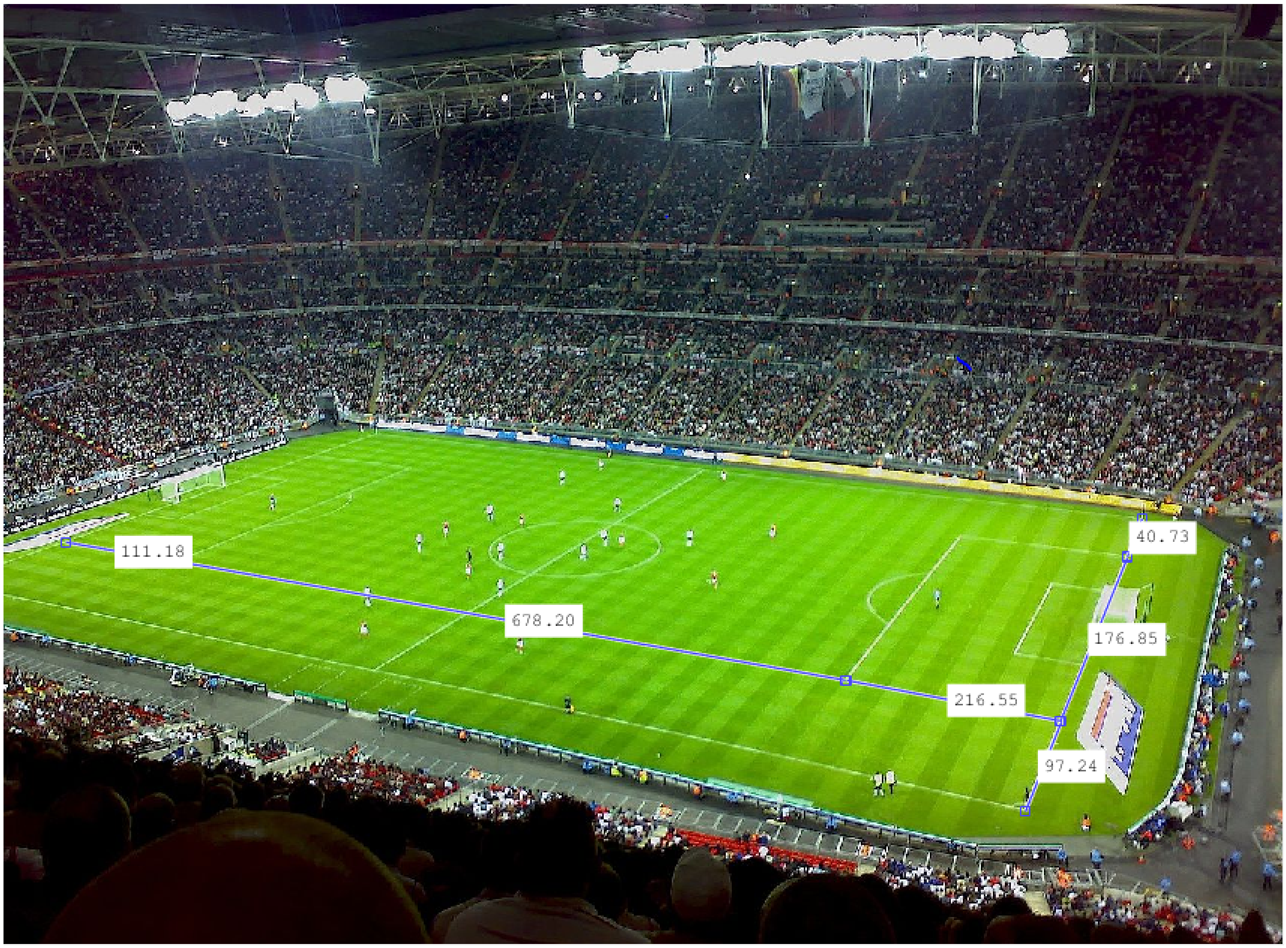

Figure 3: Wembley with pixel measurements

Problem 5

The method of minimizing joint entropy works well for the given set of images. The fixed image, moving image, and its corrected image formed by getting the parameters from minimizing joint entropy have been shown in Figures.

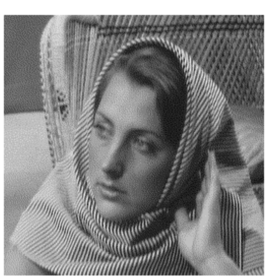

Figure 4: Original Barbara

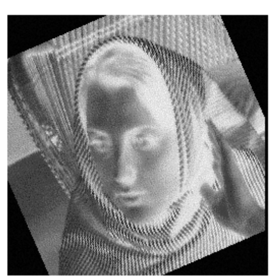

Figure 5: Moved Barbara(negative)

Figure 6: Corrected Barbara(negative)

The parameters for Barbara image come out to be -23 degrees rotation, 3 pixels translation which is exactly the ground truth. For the flash image we get -23 and 4 as the parameters for minimizing joint entropy. This is quite close

Figure 7: Original(flash)

Figure 8: Moved noflash

Figure 9: Corrected noflash

to the correct answer -23 and 3, even the values of joint entropy seen are 4.5967 and 4.5960 thus not very different.

The surface plots of the joint entropy for the images can also be seen in the attached figures. The translation from -12 to 12 has been represented on the surf plot as 0 to 25, and rotation from -60 to 60 degrees is shown as 0 to 121.

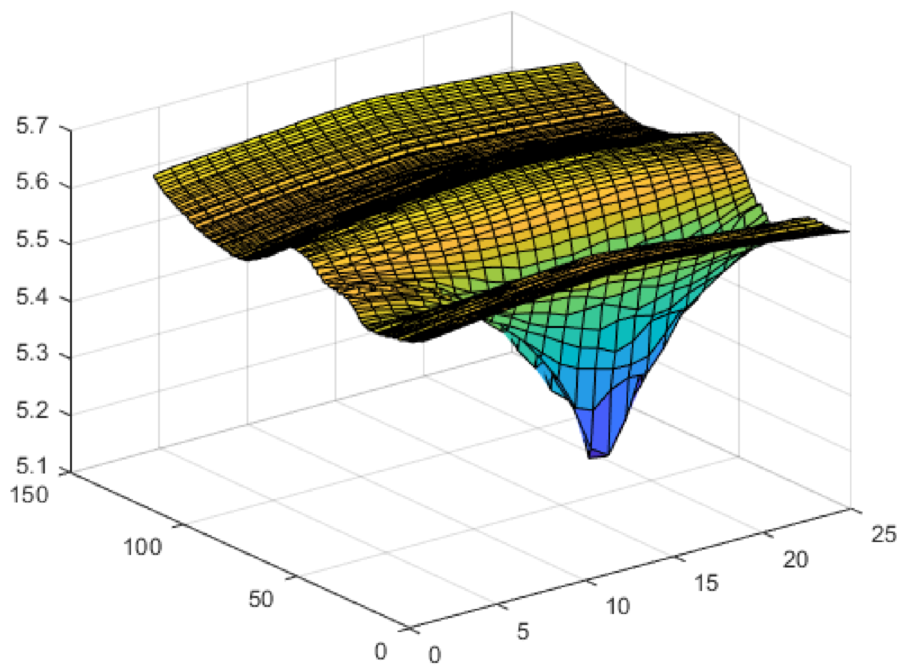

Figure 10: Barbara - joint entropy surface plot

The only difference between the surface plots is that in the barbara image the minima is quite steep, meaning it is quite distinct from its neighbouring points and thus easily find-able. In the flash image the region is more like a plateau, which could be due to more noise present in this image. But still we can say this method works really well on both the given images. The quality of alignment is good for both images.

A scenario for the Barbara image where there is a false minimum can be when the translation magnitude is large, say 30 pixels. Then this method does not work very well because there is very less area of overlap between the corrected moved image and the original image. It has been shown in the images below.

Barbara is rotated by 23.5 degrees and translated in X by a distance of -30 pixels. The minimization gives us (-37,40) which is far from the correct answer. The images are obviously misaligned.

The surf plot for this is also shown. Clearly it does not seem to have a well defined minima which might be the reason for the values coming out to be different from the expected parameters for rotation and translation.

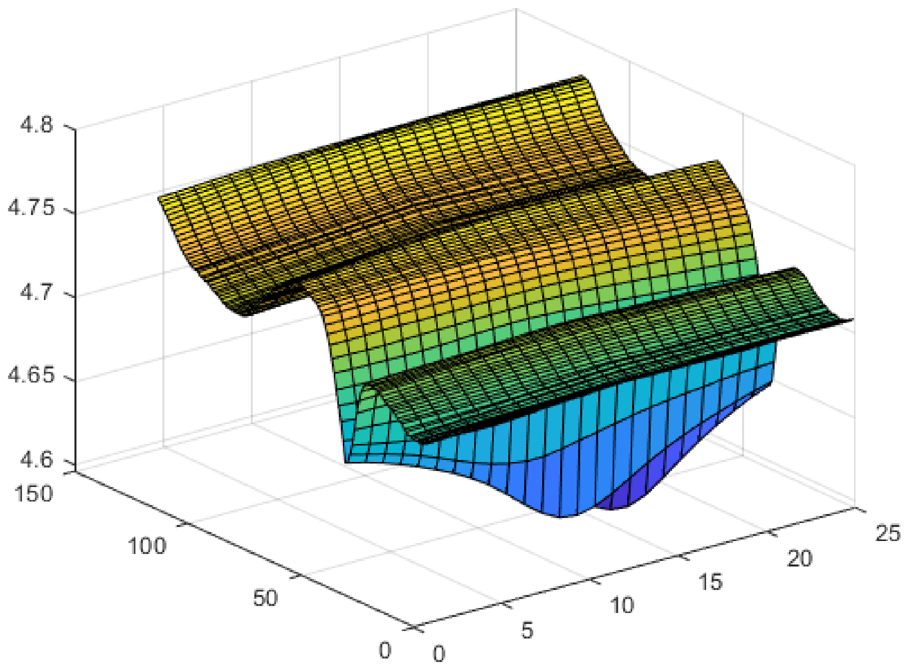

Figure 11: Flash image - joint entropy surface plot

Figure 12: Barbara image - joint entropy imshow plot

Figure 13: Flash image - joint entropy imshow plot

Figure 14: Original barbara

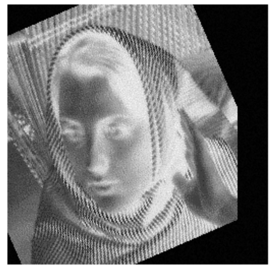

Figure 15: Moved(large translation)

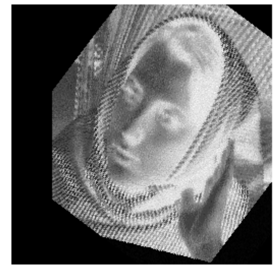

Figure 16: Corrected(according to min.)

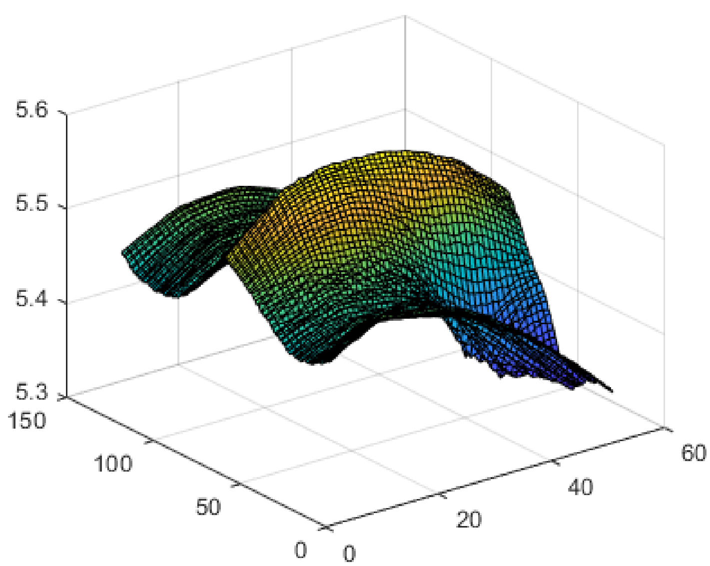

Figure 17: Barbara with large translation