Computer Vision | Assignment 2

Mohd Safwan, Kushagra Juneja, Saksham Khandelwal
17D070047, 170050041, 170070024

February 4, 2019

Question 1

We used two images of the checkerboard on the orthogonal walls for our purpose. We took around 48 points in total for the callibration of the camera. The dataset of points can be found on files xy.mat and xyz.mat files enclosed. We created a function for the normalization of the 2D and 3D points each which normalises the points as mentioned in the question. Next, we estimated the DLT matrix on these normalized points by flattening the 3×4 transformation matrix to be determined and then making equations using the 2D-3D correspondences obtained. Then we obtained the least squares solution to this problem by doing the SVD decomposition of the matrix. Then we decomposed the DLT matrix to find the value of X_0, and did Q-R decomposition to find the value of the K,R matrices. Here are the results of our callibration :-

$^{s}\mathbf{x}$	$^{s}\mathbf{y}$
1958	648
1961	877
1970	1098
1972	1317
1979	1531
.	.
.	.
.	.
2877	1682
3074	642
3075	858
3074	1070
3076	1699

Table 1: Image Plane Coordinates in px

X	**Y**	**Z**
8.7	0.0	14.5
8.7	0.0	11.6
8.7	0.0	8.7
8.7	0.0	5.8
8.7	0.0	2.9
.	.	.
.	.	.
.	.	.
0.0	11.6	11.6
0.0	11.6	8.7
0.0	11.6	5.8
0.0	11.6	2.9
0.0	11.6	0.0

Table 2: World Coordinates in *cm*

$$P = \begin{bmatrix} -67.06 & 43.53 & -45.62 & 2332.76 \\ -12.90 & -7.94 & -67.06 & 1646.23 \\ 0.01 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$X_0 = \begin{bmatrix} 54.68 \\ 31.94 \\ 10.24 \end{bmatrix} \tag{2}$$

$$R = \begin{bmatrix} -0.4463 & 0.8947 & -0.0201 \\ -0.0950 & -0.0250 & 0.9952 \\ 0.8898 & 0.4460 & 0.0962 \end{bmatrix} \tag{3}$$

$$K = \begin{bmatrix} 68.9654 & 0.7430 & -40.6947 \\ 0 & -65.3124 & -21.4752 \\ 0 & 0 & -0.0157 \end{bmatrix} \tag{4}$$

As demonstrated by the result figure, the 2D projections calculated by the estimated projection matrix are equal to the original 2D points within experimental errors. The RMSE between the points is **5.213**. The reason normalisation is important before estimating the DLT matrix is that if the scale of different inputs is different then the scale of their errors would also be different. This would mean that the least-square error estimation algorithm in case of noisy observations would give different weightage to both the inputs. Hence, we need to normalise them so that both the errors are treated alike by our least-square error estimation algorithm.

Question 2

We select four points of the 'D' (pixel values, using `imtool()` function of MATLAB), and four corresponding points of the rectangle from the top view of dimension 44yd * 18yd. These points, when fed into the `homography()` function give the corresponding transformation matrix, then we multiply the corresponding corners of the play area with the transformation matrix to get the corresponding points on the top view projection, code for which is given in the `myMainScript.m`, which converted into the Euclidean coordinates, then give us the dimensions of the play area, which comes out to be **74 yards * 114.3 yards**.

Question 3

The question requires us to find the dimensions of the play area, using just the dimensions of the Dee and without using Homography transforms. We use the property of cross ratios, which remain invariant under projective transformations.

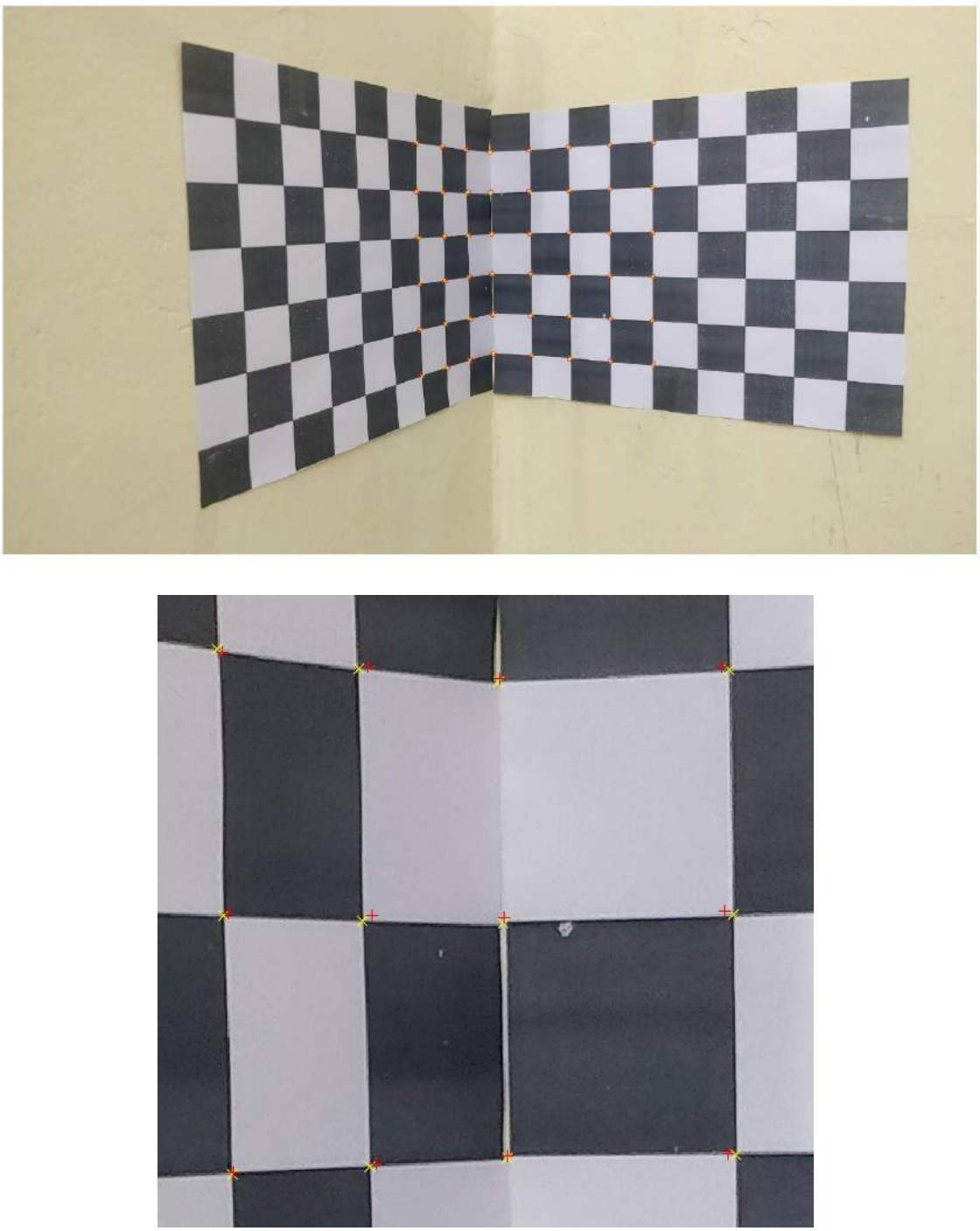


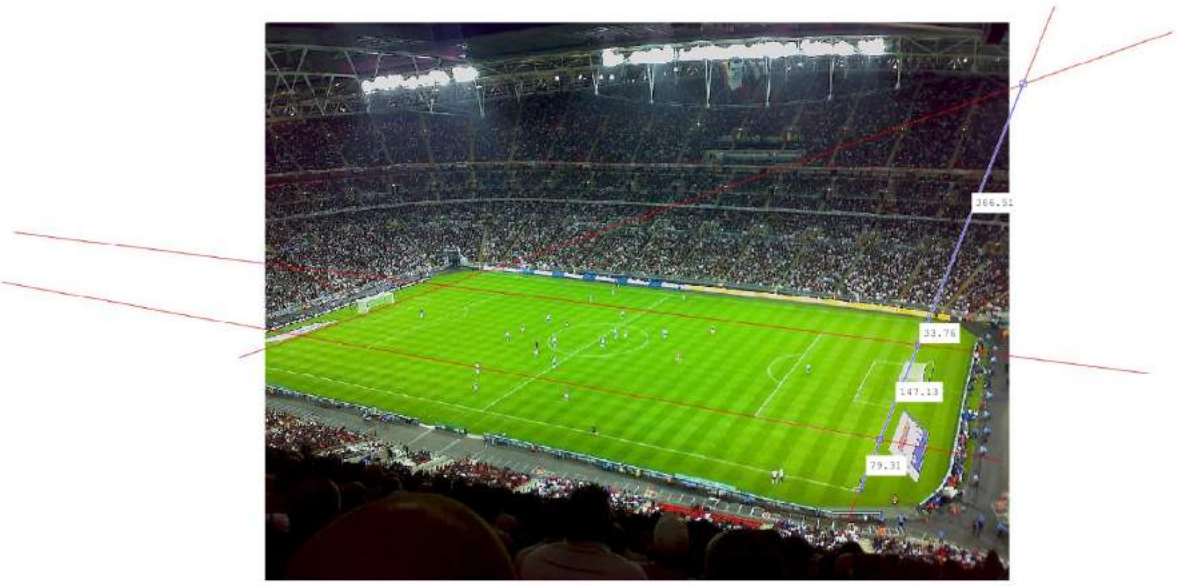
Figure 1: Visualisation of estimated 2D projections, yellow `x` are manually annotated points, red `+` are calculated from the homography

For this, we take two projections of the field, the first, given in the picture and the other taken from the top, and four points on the same line in the two projections to apply the property of cross ratios.

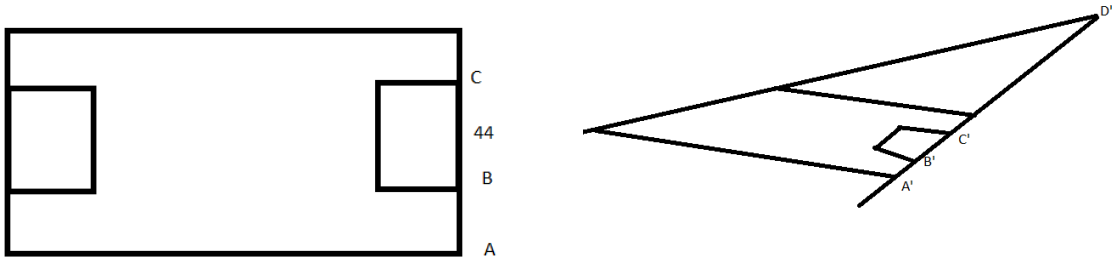
In the projection given in the picture, we extend two parallel for each dimension of the field to get the point at the horizon, where the parallel lines meet, and subsequently, we calculate the distance in pixels for either dimension, between the points A', B', C', D'.
For the projection from the top, we know that one of the point resides at infinity, and hence, the property of cross ratios reduces to:

$$\frac{A'C'}{A'D'} : \frac{B'C'}{B'D'} = \frac{AC}{BC}$$

The extended lines look like:



For the shorter width of the play area, top view show:



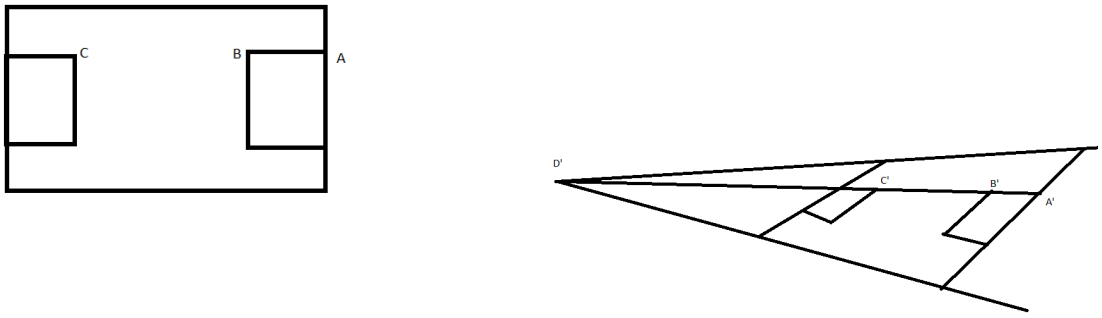
((a)) View 1 ((b)) View 2

Figure 2: Points used for Cross Ratio calculation

Plugging in the values of lengths :

$$\frac{226}{626} : \frac{147}{547} = \frac{44 + AB}{44}$$

This gives, AB = 15.1yd
Similarly the width on the other side of the Dee is = 15.2 yd
Hence, the total width of the playing area is 15.1 + 44 + 15.2 = 74.3yards
For the length, consider:
Plugging in values:



((a)) View 1 ((b)) View 2

Figure 3: Points used for Cross Ratio calculation

$$\frac{600}{1744} : \frac{461}{1605} = \frac{18 + BC}{BC}$$

Hence, BC comes out to be = 91 yards
The total length is 18 + 91 + 18 = 127 yards
By rounding off, the final dimensions are **127 yards * 74 yards**

Question 4: Image Stitching

We extracted the SURF feature-points from each image using `detectSURFFeatures()`, `extractFeatures()` and matched them using `matchFeatures()`. We use the RANSAC algorithm to compute the best 2D homography between images by reusing the `homography()` function from Question 2. The images were warped using the computed 2D homography transforms and are stitched together using `AlphaBlender` using Binary Masking instead on Blending.

The obvious problem in the above results in different image intensities of the source images. We believe normalizing them using histogram transfer might solve the issue, but that is beyond the scope of the assignment. For the next part of the question, due to some

((a)) Ledge

((b)) Hill

((c)) Monument

((d)) Pier

Figure 4: Points used for Cross Ratio calculation

logistical issues, we could not click an image from the phone camera. We used another image publicly available on the internet.

((a)) Left Side ((b)) Center ((c)) Right Side

Figure 5: Some random building in Cornell University found in Public Domain

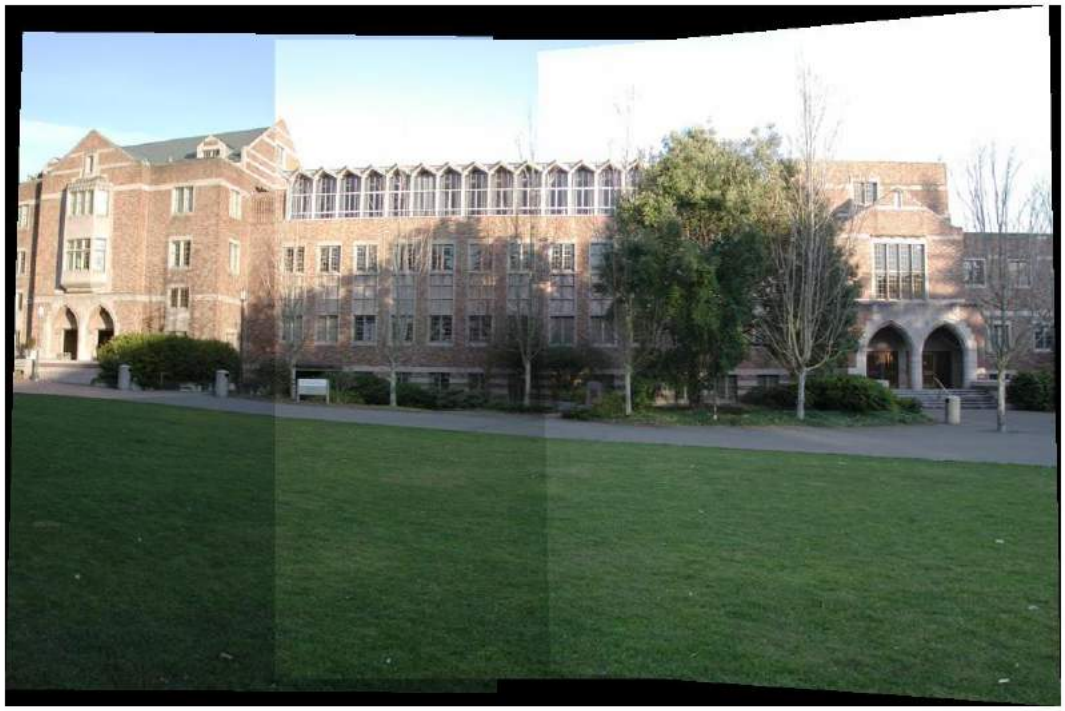
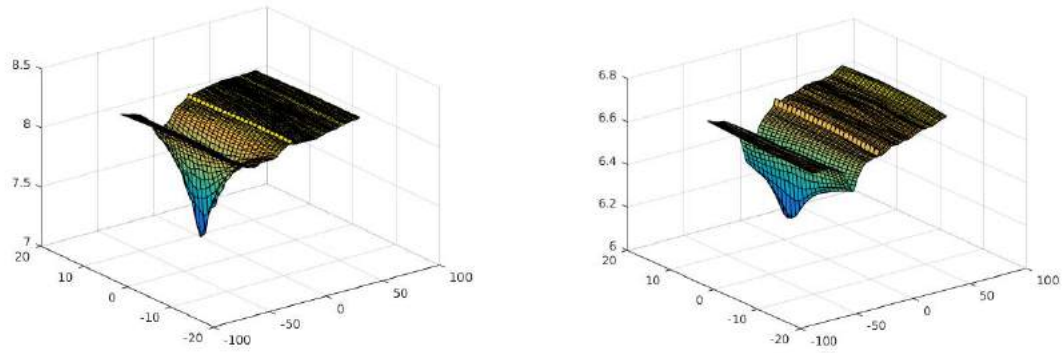


Figure 6: Stitched image with optimized `threshold`

Question 5

The Barbara image was already in grayscale so we didn't modify that but the flash image was in RGB so we first converted it into grayscale. Then, we downsampled the flash image by a factor of 2 on both the dimensions. We created a function called 'calc_entropy' which calculates the joint entropy of the two images over only those pixels at which both the images have non-zero intensity by making a joint histogram of bin-size 10.
The following results were observed:-
For Barbara image, min joint entropy value 7.16 occurs at theta = -23, tx = 3
For flash image, min joint entropy value 6.20 occurs at theta = -23, tx = 3



((a)) Barbara ((b)) Flash

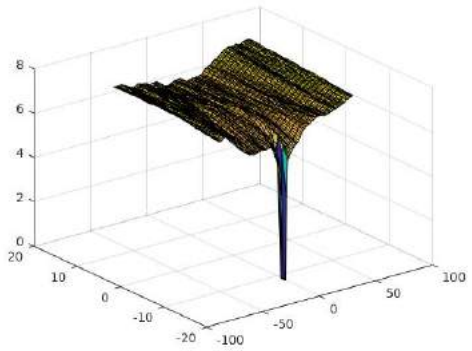
Figure 7: Surface plot of joint-entropy

((a)) Barbara ((b)) Flash

Figure 8: Plot of joint-entropy interpreted as grayscale image using imshow

Hence, even though the alignment was correct on both the images(barbara and flash) but still by looking at the surface plot of joint-entropy we can comment that barbara image has a better alignment since the minima is steeper in that case. This is because the flash image has different brightness of different parts of image and also both the images have different brightness levels.(one is with flash and the other is without flash).

The case of undesirable minima can occur when the image is translated so much such that most of the region in the translated image is black hence the algorithm cannot compare properly. E.g. in the figure, we have shown the output of the surface plot when we originally translated the image by -245 pixels rather than -3 pixels. The minima should have occured



((a)) Undesirable minima

Figure 9: Surface plot of joint-entropy for Barbara image showing undesirable minima

at 12 pixels since that is the maximum we can go towards the original image based on our search space. But the actual minima occurs at -12 since at that value the algorithm can completely avoid any joint-entropy since the image is black hence leading to minima at zero joint-entropy.