Assignment 2: Projective Geometry and RANSAC

Michael Shepherd, 19059019

August 2019

Question 1

1a

We would like to represent:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s\left(\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) \tag{1}$$

In the form:

$$\underline{x}' = \mathbf{H}\underline{x}, \text{ with } \mathbf{H} = \begin{bmatrix} sR & \underline{t} \\ \underline{0} & 1 \end{bmatrix} \tag{2}$$

By multiplying out equation 1, we find that:

$$\underline{x}' = \begin{bmatrix} s\cos\theta(x - x_0) - s\sin\theta(y - y_0) + x_0 \\ s\sin\theta(x - x_0) + s\cos\theta(y - y_0) + y_0 \\ 1 \end{bmatrix} \tag{3}$$

By multiplying out equation 2, we find that:

$$\underline{x}' = \begin{bmatrix} s\cos\theta - s\sin\theta + t_1 \\ s\sin\theta + s\cos\theta + t_2 \\ 1 \end{bmatrix} \tag{4}$$

We can then solve for $\underline{t}$ using these two equations, giving us:

$$t_0 = -(\cos\theta x_0 + \sin\theta y_0 - x_0) \tag{5}$$

$$t_1 = -(\sin\theta x_0 + \cos\theta y_0 - x_0) \tag{6}$$

This gives us:

$$\underline{x}' = \begin{bmatrix} s\cos\theta & -s\sin\theta & -(\cos\theta x_0 + \sin\theta y_0 - x_0) \\ s\sin\theta & s\cos\theta & -(\sin\theta x_0 + \cos\theta y_0 - x_0) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{7}$$

1b

To complete this question, I took the apply homography function that was given to us and translated it into Python to suit my needs. I then changed the H matrix from the previous question to re-size and rotate an image in various different ways, as is illustrated below.

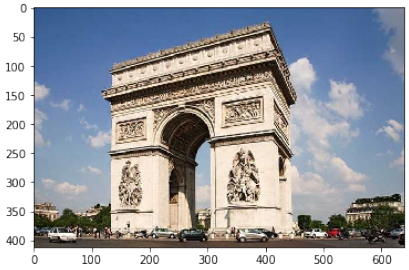

Figure 1: Base Image

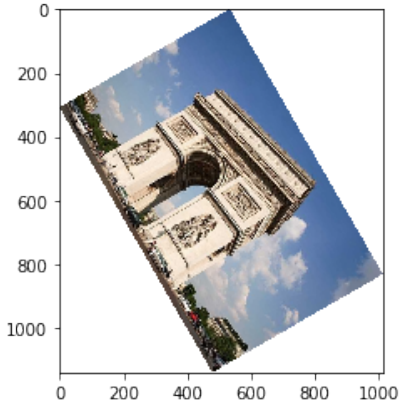

Figure 2: $\theta = \frac{\pi}{3}$, $s = 1.5$

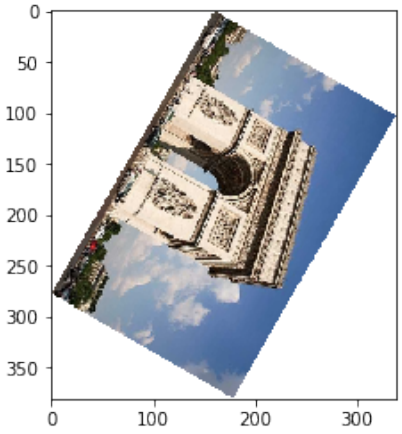

Figure 3: $\theta = 2\frac{\pi}{3}$, $s = 1.5$

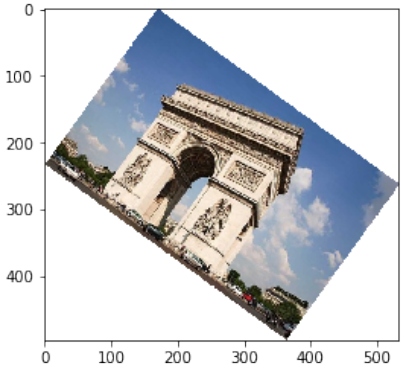

Figure 4: $\theta = 0.2\pi$, $s = 0.7$

1c

The procedure in the document does not actually shift the image in a new space, it only performs the ration and the updated origin is stored in the form of minx and miny. This means that the image will be rotated and warped, but will still use 0,0 as its origin.

The image is rotated in an anticlockwise direction on a standard Cartesian plane, but we are not using a standard Cartesian plane, we are using image coordinates which points the x coordinates rightwards and the y coordinates downwards. This results in us seeing what appears to be a clockwise rotation, but is in fact an anticlockwise rotation on a standard Cartesian plane.

Question 2

Figure 5: Poster

Figure 6: Building

Above is the poster that I have chose and the building that it will be projected onto. This was achieved by first finding the homography that transforms the poster into a shape that will fit onto the building and then using that to place it.

To find the homography, I used the instructions for determining H from point correspondences from the lecture slides. This gave me a warped image that was on the incorrect axis. To shift the warped poster, I created an empty image that was the size of the building image and manually shifted every pixel in the poster the their new position in relation to the shifted origin as calculated by the apply homography function.

To then put the poster on top of the building, I created a boolean mask of the new shifted and warped poster image that was true where the poster was and false where it wasn't. I then used this mask to create a new image that contained the poster where it needed to be and the building everywhere else.

Down-scaling large images can cause aliasing, because condensing a large amount of information into a small area can cause some of the information to be lost or distorted. This can lead to jagged edges and unclear objects in the down scaled image. This was clearly visible in the previous assignment.

To try and illustrate the issues of aliasing, I both down-scaled and up-scaled the poster before projecting it onto the building. this is illustrated

on the small poster and large poster in the figures. Due to my image being around the same size as the building, up-scaling and then projecting the image had almost zero effects. This is because you cannot create information by increasing the size of an image with bilinear interpolation. The aliasing is made more clear in the small poster image, but this is due to the poster being re-scaled before being projected, not because of the transformation itself. From this, I can conclude that if the original poster was a high resolution image that was far larger than the building, it would have a similar effect to the small poster image.

1. Identify at least 4 point correspondences $(x_i, y_i) \leftrightarrow (x_i', y_i')$
2. Create matrix A (slide 6)
3. Calculate SVD of A: $A = U\Sigma V^T$
4. Let $\underline{h}$ be the last column of V, then $H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$

Figure 7: Determine H from point correspondence

Figure 8: Small Poster

Figure 9: Medium Poster

Figure 10: Large Poster

Question 3

This question was solved using the RANSAC idea and homography estimation algorithms given in the lecture slides.

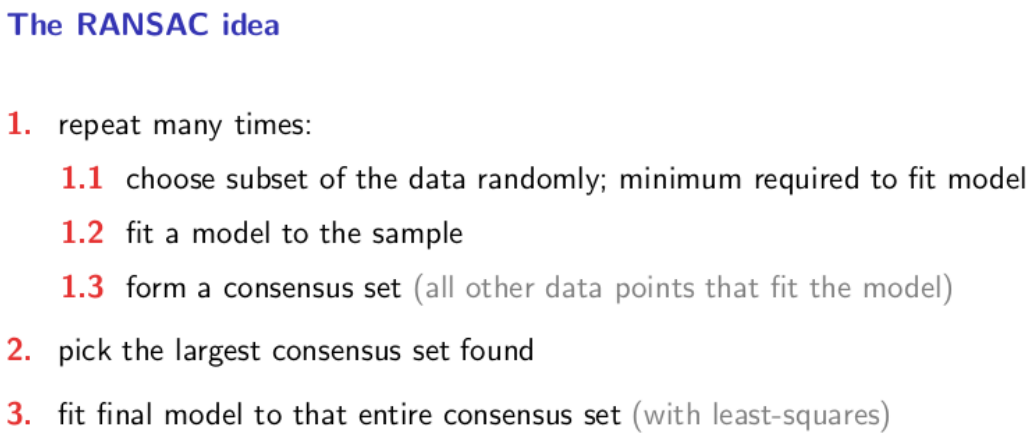

Figure 11: RANSAC

For the RANSAC, I found a consensus set from the given matches where the end points and the start points were both within the threshold distance of their respective model, with the model being a line between two matches. This allowed me to have a larger consensus set as well as a more accurate consensus set. I repeated the process 1000 times, or until the consensus set was 75% of the size of the set of matches.

For the homography estimation, I ran through 1000 homographies that were generated using four random points from the consensus set. To check the accuracy of the homography, I tested if the transform of every single given match start point was in a similar area to the corresponding match endpoint and counted the amount of correct transformations to find the best homography. Once the best homography was found, I used the correct transformations that I had previously counted to re-estimate H to find the best possible Homography.

Estimating a homography from image feature matches

The data we have: matches $(x_i, y_i) \leftrightarrow (x_i', y_i')$

1. repeat many times:
   - 1.1 choose 4 matches randomly
   - 1.2 calculate H (as we did in Lecture 10)
   - 1.3 map every (x_i, y_i) with H, and compare with its match (x_i', y_i'); those close enough form the consensus set
2. pick the largest consensus set found: this is our set of inliers
3. re-estimate H in a least-squares sense, using the entire set of inliers

Figure 12: Estimate H

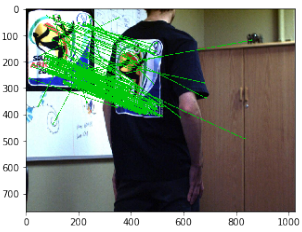

Figure 13: Matches

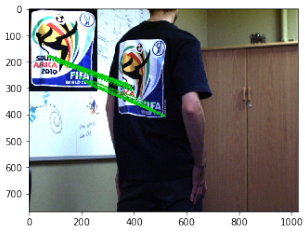

Figure 14: Inliers

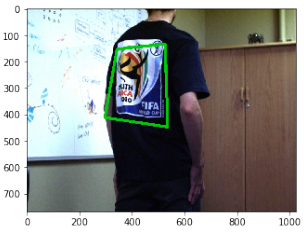

Figure 15: Feature

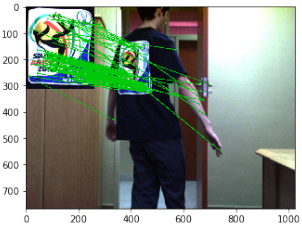

Figure 16: Matches

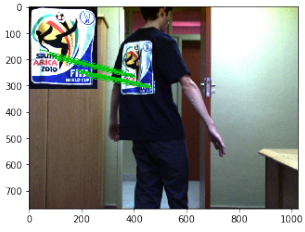

Figure 17: Inliers

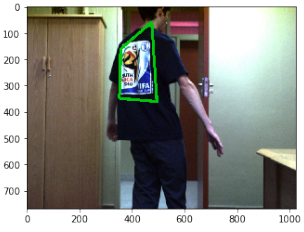

Figure 18: Feature

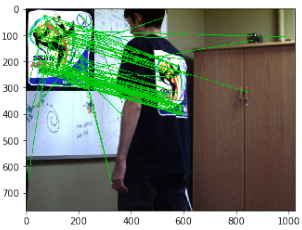

Figure 19: Matches

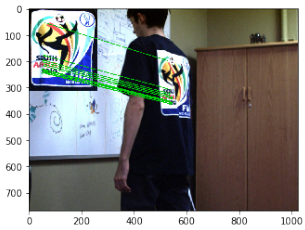

Figure 20: Inliers

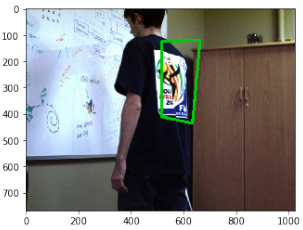

Figure 21: Feature

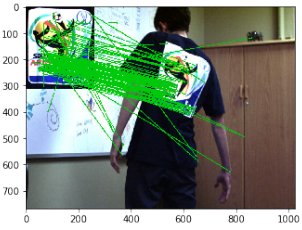

Figure 22: Matches

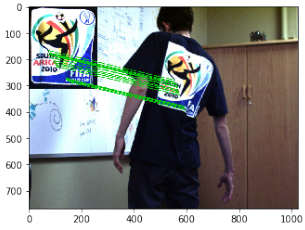

Figure 23: Inliers

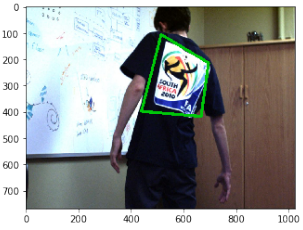

Figure 24: Feature

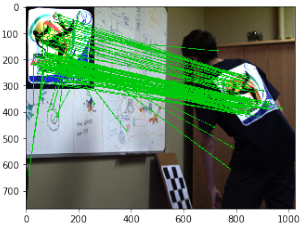

Figure 25: Matches

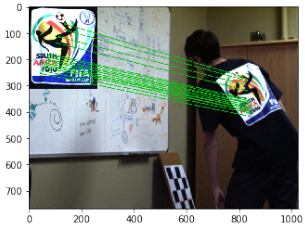

Figure 26: Inliers

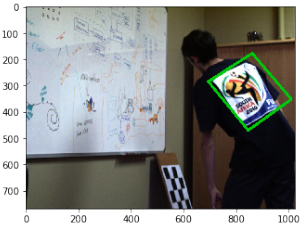

Figure 27: Feature

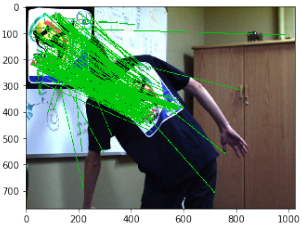

Figure 28: Matches

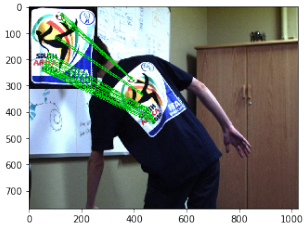

Figure 29: Inliers

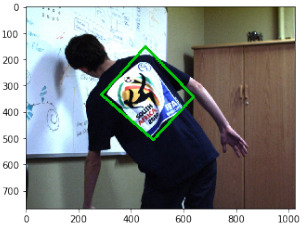

Figure 30: Feature

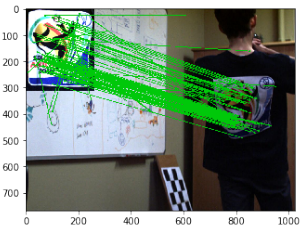

Figure 31: Matches

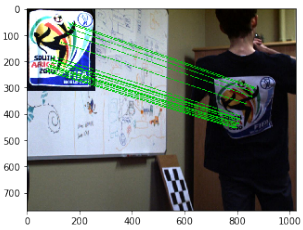

Figure 32: Inliers

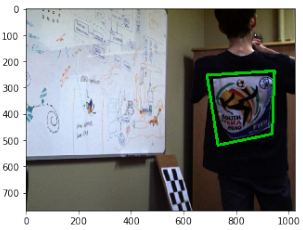

Figure 33: Feature

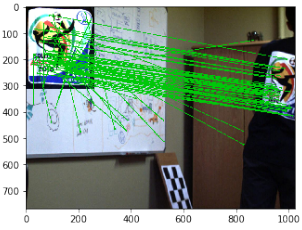

Figure 34: Matches

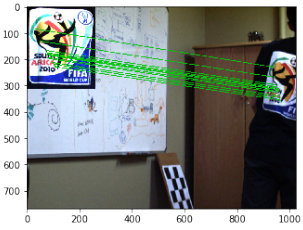

Figure 35: Inliers

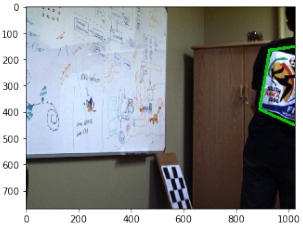

Figure 36: Feature

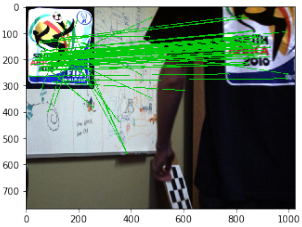

Figure 37: Matches

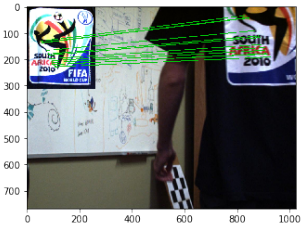

Figure 38: Inliers

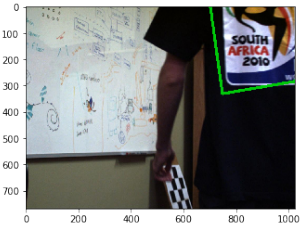

Figure 39: Feature

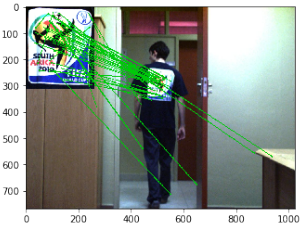

Figure 40: Matches

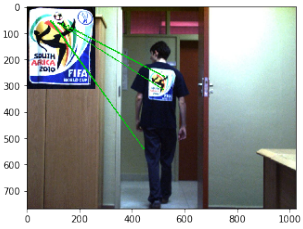

Figure 41: Inliers

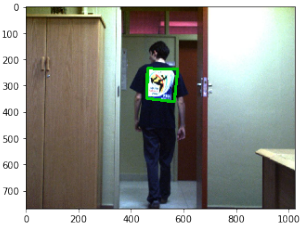

Figure 42: Feature

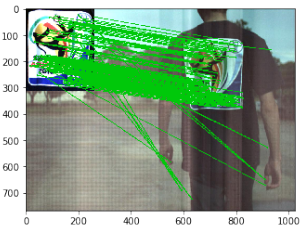

Figure 43: Matches

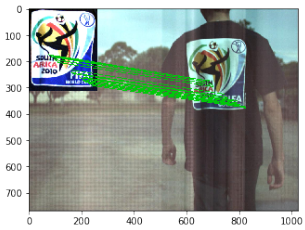

Figure 44: Inliers

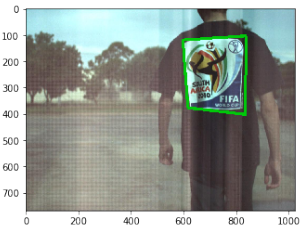

Figure 45: Feature

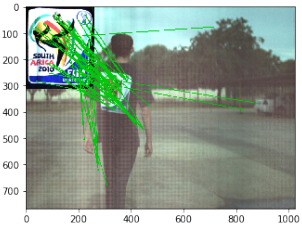

Figure 46: Matches

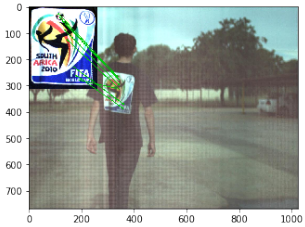

Figure 47: Inliers

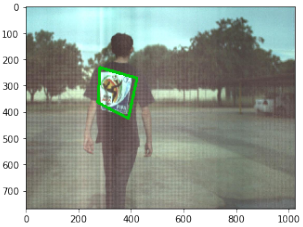

Figure 48: Feature