Machine Vision - Assignment 1

Qi Wang

January 2018

1 Homographies Part I

The aim of practical 1 is to calculate the homography that best maps two sets of points to one another.

(A)

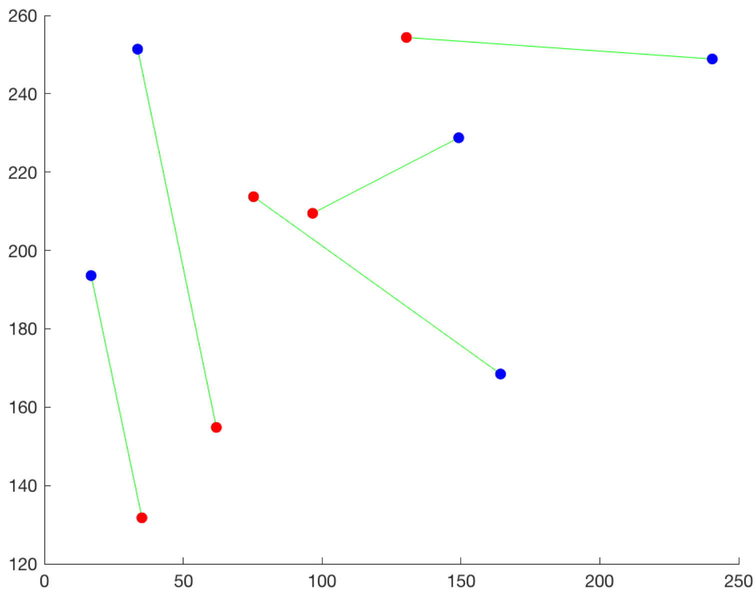

Figure 1: two sets of Cartesian points

Figure 1 shows how a set of Cartesian points $w = [u, v]'$ on the plane are converted into another set of Cartesian points $X = [x, y]'$ to which are projected. The $X = [x, y]'$ is generated by applying homography to the original points $w = [u, v]'$, As can be seen that the original Cartesian points $w = [u, v]'$ are marked by blue points, while the projective points are marked by red points. Additionally, there is a green line connecting two sets of corresponding points. According to the code, the elements in projective transformation matrix are pre-defined values, so figure 2 will show a new mapping relationship between $w = [u, v]'$ and $X = [x, y]'$ by calculating the best homography matrix.

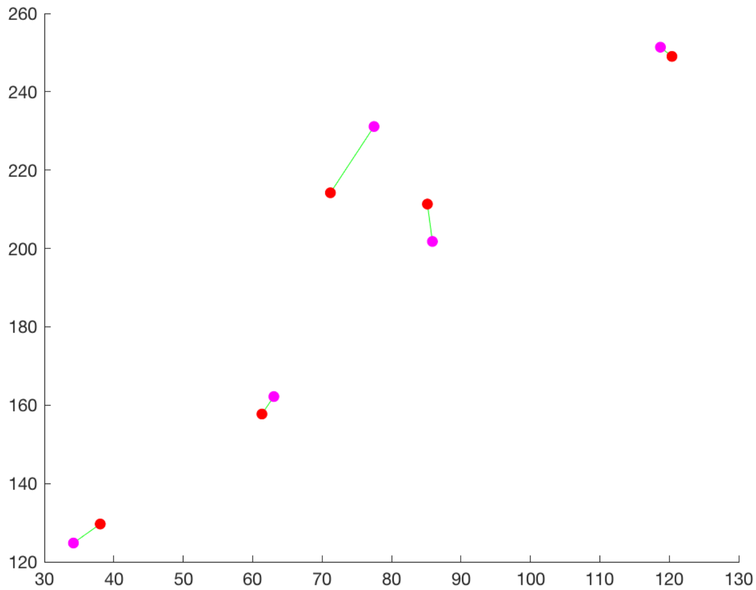

Figure 2: two sets of Cartesian points

Figure 2 shows a new mapping relationship between $w = [u, v]'$ and $X = [x, y]'$. The homography matrix is generated by function H = calcBestHomography(pts1Cart, pts2Cart). This function applies direct linear transform (DLT) algorithm to calculate best homography that maps the points in pts1Cart to their corresonding match in pts2Cart

Scale ambiguity: homography is ambiguous up to scale, as the Cartesian points $w = [u, v]'$ is firstly converted into a set of Homogenous points and then converted into projective points $X = [x, y]'$, the scaling factor is stored into the Homogenous matrix. In the revised process, the scaling factor is removed in the process of converting from Homogenous matrix back to Cartesian points. So by multiplying it by a constant factor will not affect the process of transformation.

Exact mapping of pairs of four points: The projective transformation or homography is parameterized by a 3*3 matrixΦ which is ambiguous up to scale, giving a total of eight degrees of freedom. Consequently, we need a minimum of I = 4 pairs of corresponding points for a unique solution. This neatly matches our expectations: a homography can map any four points in the plane to any other four points, and so it is reasonable that we should need at least four pairs of points to determine it.

******TO DO******

function H = calcBestHomography(pts1Cart, pts2Cart)

This function is a direct linear transformation or DLT algorithm. The DLTalgorithm uses homogeneous coordinates where the homography is a linear transformation.

$$\lambda \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \tag{1}$$

Each homogeneous coordinate can be considered as a direction in 3D space. So, the equation 1 states that the left-hand side x_i represents the same direction in space as the right-hand side, Φw. If this is the case, their cross product must be zero, so that $x \times \Phi w = 0$.

To find the solution, we compute the $SVDA = ULV^T$ and choose $\Phi$to be the last column of V.

(B)

Figure 3: panorama of several images

There are three images and two sets of matching points initially. The first thing is to calculate the homograpghy matrices between these matching points in order to get the panorama. pts1 and pts2 are matching points between image1 and image2. pts1b and pts3 are matching points between image 1 and image 3. Fortunately, the algorithm to calculate the homography matrix has been given in the part A. The next step is to write an algorithm which can copy pixel colour from image 2 pixel to current position in image 1, same method can be used between image 3 and image 1.

The algorithm can be concluded as follows:

```
Algorithm: copy pixel color
for every pixel in image 1

    transform this pixel position with your homography to find
    where it is in the coordinates of image 2
    if it the transformed position is within the boundary of image 2 then
        copy pixel colour from image 2 pixel to current position in image 1
        draw new image1
    end
end
```

Same algorithm can be used to copy pixel colour from image 3 pixel to current position in image 1. To be more specific, each pixel in the center image (img1) is transformed with homography to obtain a position on the two sides after each homography matrix is calculated. Finally, the current pixel in the central image is replaced with the colour(RGB value) of the associated pixel in the left and right images.

2 Homographies Part II

(C)

******TO DO******

xImCart = projectiveCamera(K,T,XCart);

Goal of function is to project points in XCart through projective camera defined by intrinsic matrix K and extrinsic matrix T. The relationship between a point $w = [u; v; 0]^T$ on the plane and the position $x = [x; y]^T$ to which it is projected is

$$\lambda \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \Phi_x & \gamma & \delta_x \\ 0 & \Phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{11} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ \omega_{31} & \omega_{32} & \tau_z \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \tag{2}$$

The first step is to convert Cartesian 3d points in the plane to homogeneous coordinates, then apply extrinsic matrix to homogeneous coordinates to move to frame of reference of camera. After that, the intrinsic matrix is applied in order to move points to image coordinates. The last step is to convert points back to Cartesian coordinates.

```
T = [ 0.9851   -0.0492   0.1619   46.00;...
     -0.1623   -0.5520   0.8181   70.00;...
      0.0490   -0.8324  -0.5518   500.89;...
      0         0         0         1]
```

Figure 4: correct extrinsic matrix

```
TEst =

    0.9646   -0.0536    0.2581   45.2760
   -0.2449   -0.5449    0.8020   71.4241
    0.0977   -0.8368   -0.5387  508.7924
         0         0         0    1.0000
```

Figure 5: estimated extrinsic matrix

I set the noise level to be 1, which means that a normal noise with the stand deviation 1 is added in the process. So the results vary during each simulation, this gives the reason why the elements in the correct extrinsic matrix are different from estimated extrinsic matrix.

(D)

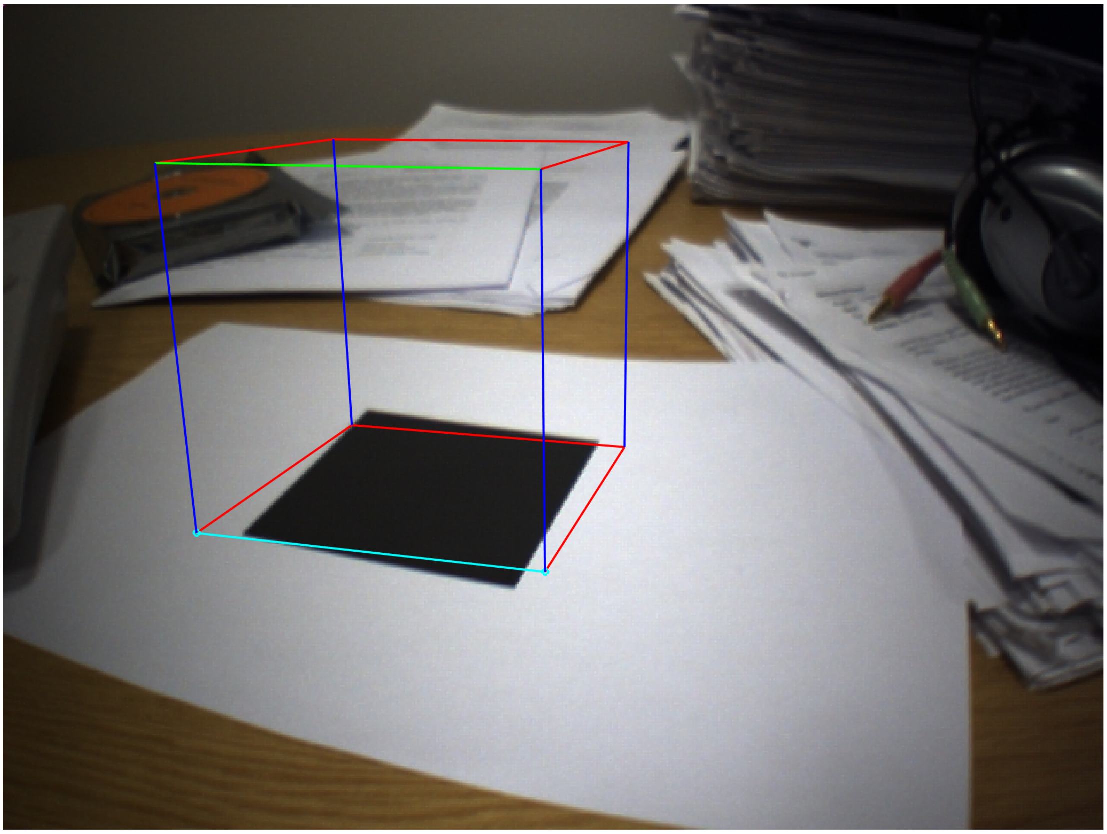

Figure 6: estimated extrinsic matrix

Practice B aims to draw a wire frame cube, by projecting the vertices of a 3D cube through the projective camera and drawing lines between the resulting 2d image points.

As there are eight points are given as the coordinates in the plane, the corresponding coordinates in the image are also given. This is quite easy process to draw the cube after the projective transformation mapping is known. Different colors of the lines refer to there is a relationship between two points and thus they should be connected together. Lines with same color refers to they are plotted at the same time. I plot the line in order. However, same results can be got by different approaches. Another method would be connect the points when the distance between them is 100.

As can be seen the cube does not fit the black square in the plane perfectly.

3 Condensation

(E)

******TO DO******

Normalize the weights:

The weight is normalized by divided by its sum (the total weights).

******TO DO****** Compute the cumulative sume of the weights:

the MATLAB built-in function 'cumsum' is used here to calculated the cumulative sum of the of the weights.

******TO DO******

Incorporate some noise:

The old particles are incorporated into Gaussian noise with standard deviation 10 to produce the new particles.

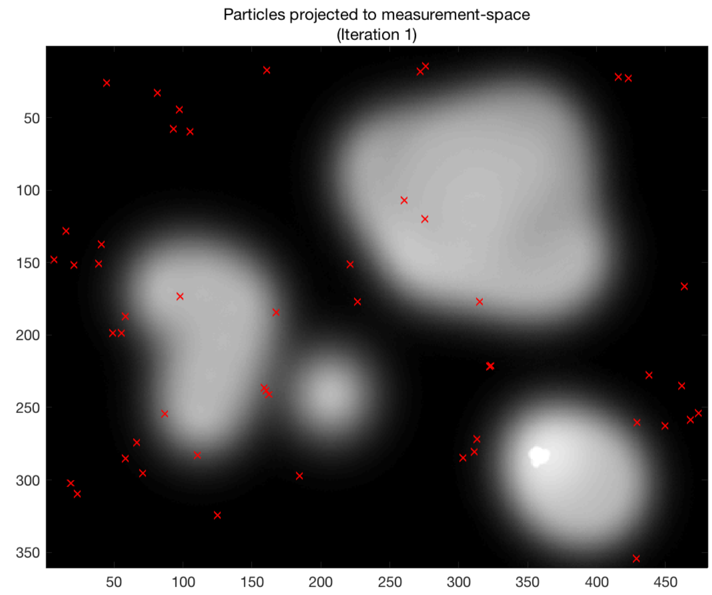

Figure 7: particles distribution in the iteration 1

In the first iteration, the particles are distributed randomly, because the threshold value is random number and what we are looking for is which sample in the ordered set is the first one to push the cumulative probability above that threshold.

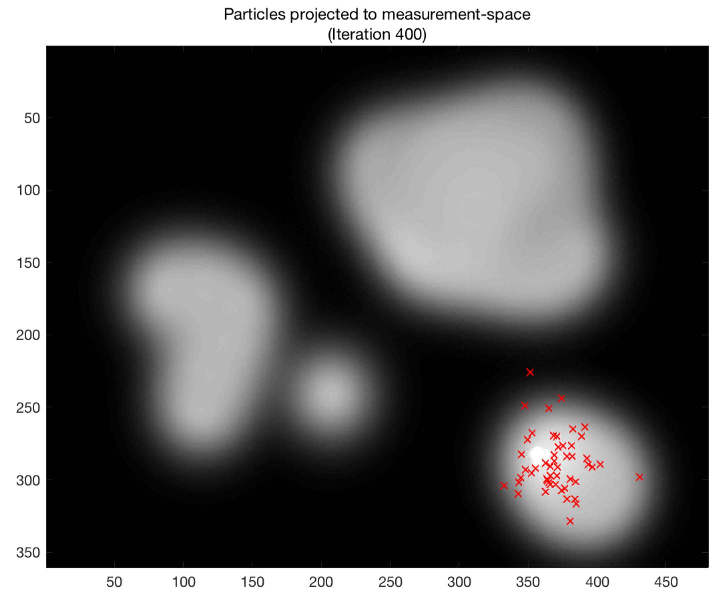

Figure 8: particles distribution in the iteration 400

TURN OVER

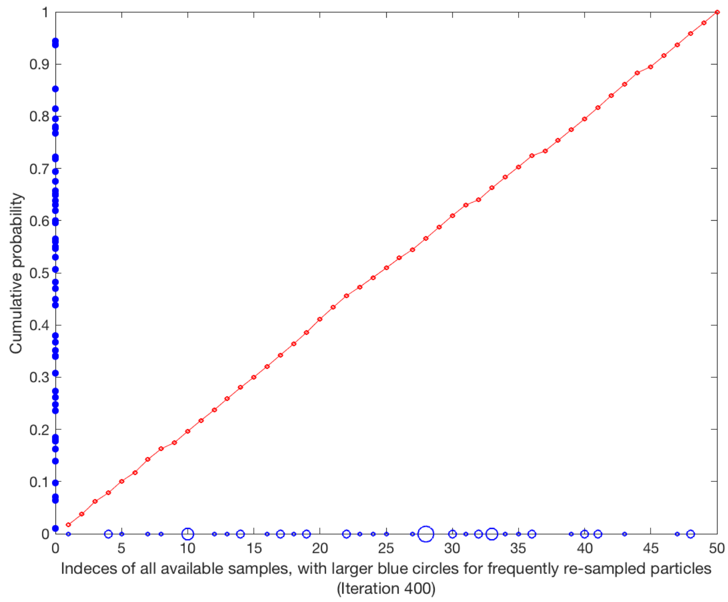

Figure 9: Indeces of all available samples, with larger blue circles for frequently re-sampled particles

After 400 iterations, the likelihood of the each sample is evaluated after the random noise is incorporated into the into the current location, to give a Brownian motion model. As can be seen from figure 9, The samples propagate to the location with high likelihood gradually.

(F)

Figure 10: particles distribution in the iteration 400

In this section we track a given shape as it moves in a sequence of frames, there are 1000 particles distributed in the frames. As can be seen from figure 10, There is some initial uncertainty at iteration 5, several cluster of particles are placed in some similar places as the

 CONTINUED

geometrical information of those places are similar as the given shape.

Figure 11: Indeces of all available samples, with larger blue circles for frequently re-sampled particles

After 22 iterations, particles are propagated to the position of the car, and they will track the car in every frame.

4 Combining Tracking and Homographies

(G)

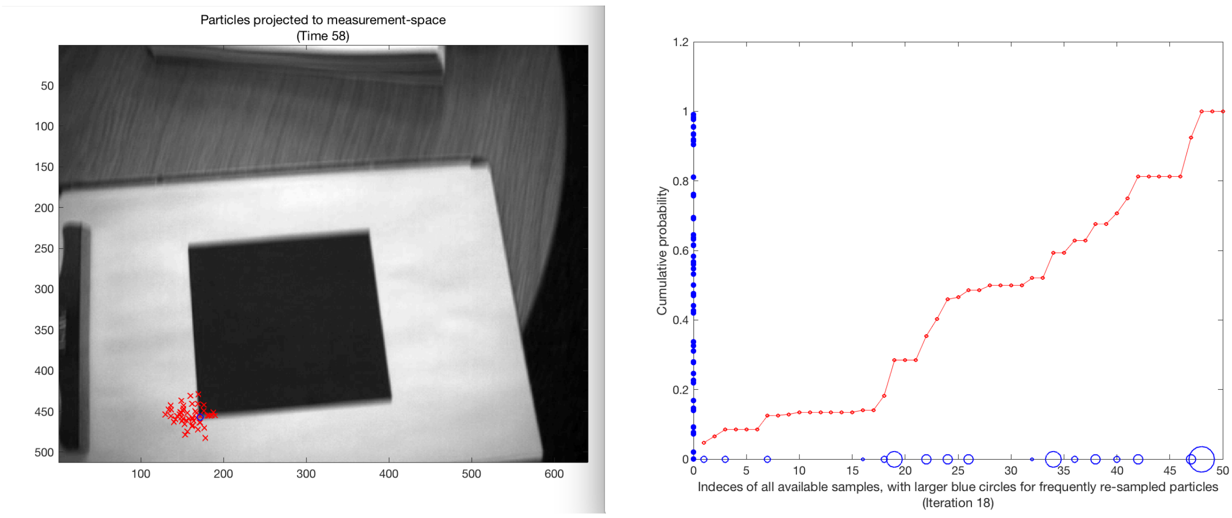

Figure 12: low left corner

TURN OVER

Figure 12 is one of the tracking frame among 150 frames. As can be seen, particles are tracking the low Left corner of the black square. The coordinate of the best location is expressed as a blue circle, which is the MAP of middle position of a patch on the frame. That 'middle' is the patch's interest point in the current frame. The particles are gathered around the blue circle from time to time. However, the situation which particles deviate away from the corner can not be avoided due to the existence of noise.

******TO DO******

Compute the coordinate of the "best" (i.e. MAP) location by computing the weighted average of all the particles:

weightedAve = weight-of-samples'* particles-new;

(H)

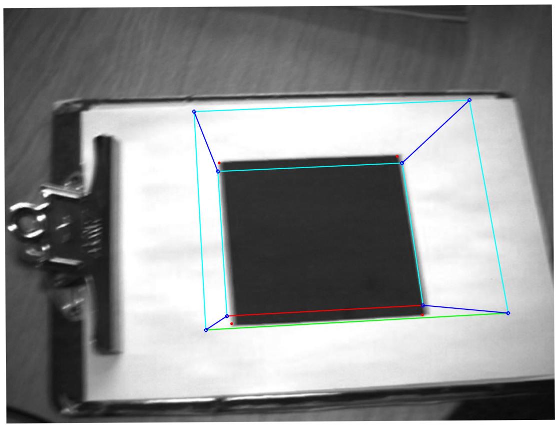

Figure 13: cube frame has been projected on the img

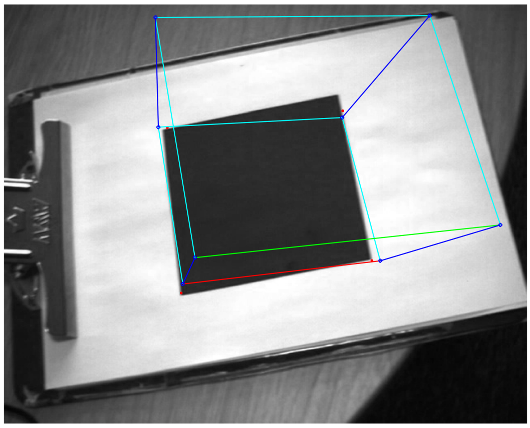

Figure 14: cube frame has been projected on the img

Figure 13 and 14 show the particles are tracking four corners of the black square. Same as what (G) did. After the best location of four corners of the black square has been tracked, a 3D cube frame will be projected onto the image, this procedure is the same as what (D) was doing. Figure 13 and 14 give two good examples on tracking the corners. However, the figures below will show how the noise will disturb the the process of homography and thus how badly the cube are projected on the image.

CONTINUED

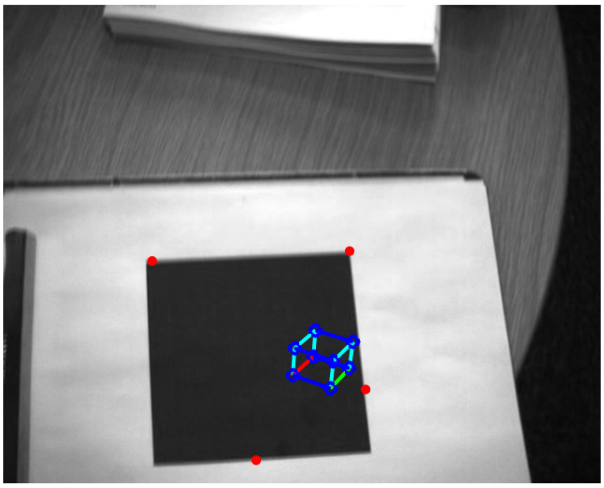

Figure 15: cube frame has been projected on the img badly

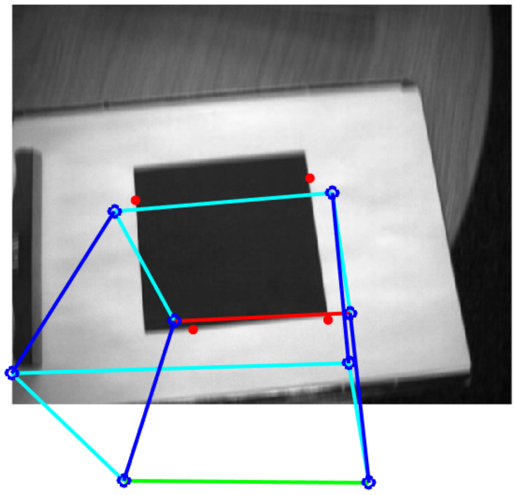

Figure 16: cube frame has been projected on the img badly

Two actions or changes we could make to improve the result:

a. Directed model for chains: take the posterior of the particles from last frame into account

b. Use multiple camera instead of one camera.

END OF COURSEWORK