

# IN4393-16 Computer Vision - Q3

## Lab Exercise 3

### Image Alignment and Stitching

March 15, 2018

Students should work on this lab exercise in groups of two people. The assignment should be completed before 26th March 2018.

In this assignment you will write a function that takes two images as input and computes the affine transform between them.

## 1 Image Alignment

You will work with supplied boat images. The overall scheme is as follows:

1. Detect **interest points** in each image.
2. Describe the **local appearance** of the regions around interest points.
3. Get the set of **possible matches** between descriptors in two image.
4. Implement **RANSAC** to discover the transformation between images.

The first three steps can be performed using **David Lowe's SIFT** (lab exercise 2). You can use the following functions to obtain the pairs for your RANSAC algorithm.

```
1 [frames, desc]=sift(im)
2 [matches] = vl_ubcmatch(desc1,desc2)
```

For step 4, RANSAC, should be implemented as follows:

1. Pick  $P$  matches at random from the total set of matches  $T$ .

```
1 perm = randperm(?);
2 seed = perm(1:P);
```

How many matches do we need to solve an affine transformation which can be formulated as shown in Fig.1?

2. Construct a matrix  $\mathbf{A}$  and vector  $\mathbf{b}$  using the  $P$  pairs of points to find the affine transformation parameters (Fig.1)  $(m_1, m_2, m_3, m_4, t_1, t_2)$  by solving the equation  $\mathbf{Ax} = \mathbf{b}$ , where

$$A = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix}, x = \begin{bmatrix} m1 \\ m2 \\ m3 \\ m4 \\ t1 \\ t2 \end{bmatrix}, b = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (1)$$

3. Fit the model

Hint: Equation  $\mathbf{Ax} = \mathbf{b}$  can be solved using pseudo-inverse. In Matlab use

```
1 x = pinv(A) * b'
```

4. Using the transform parameters, transform the locations of all  $T$  points in image 1. If the transform is correct, they should lie close to their counterparts in image 2. Plot the two images side by side with a line connecting the original  $T$  points image 1 and transformed  $T$  points over image 2 as shown in Fig.3.
5. Determine the inliers: First reform the matrix  $\mathbf{A}$  and then compute the new  $\mathbf{b}$

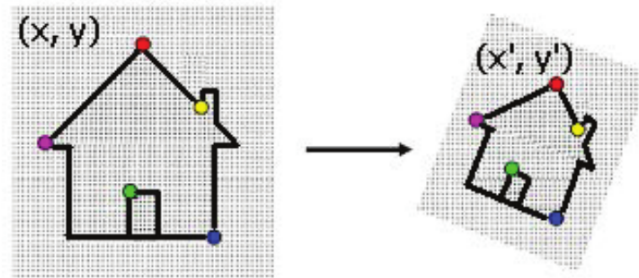


Figure 1: Affine transform

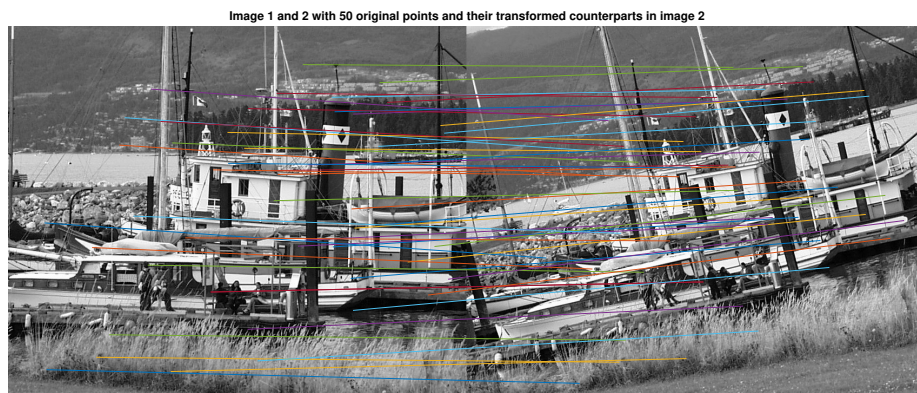


Figure 2: Correspondence in two image

```

1  bprim = A*x
2  ...
3  inliers = find(sqrt(sum((?).^2)) < threshold);

```

Count the numbers of inliers being defined as the number of trans-

formed points from image 1 that lie within a radius of 10 pixels of their pair in image 2.

6. Refit the model and save the best model (inliers and the parameters) so far and repeat previous steps  $N$  times. How do you choose  $N$ ?

Once the estimation of  $\mathbf{x}$  is finished, transform image 1 using this final set of transformation parameters. If you display this image you should find that the pose of the object in the scene should correspond to its pose in image 2. To transform the image, use the build-in Matlab functions as follows

```
1 T = [x(1) x(2) x(5); ...
2       x(3) x(4) x(6); ...
3       0    0    1   ];
4 tform = maketform('affine', T);
5 image1_transformed = imtransform(image1,tform, 'bicubic');
```

Do the same procedure for the second image as

```
1 tform = maketform('affine', inv(T) ');
2 image2_transformed = imtransform(image2,tform, 'bicubic');
```

The final images that you should plot for this exercise are depicted in Fig.4.

## 2 Image Stitching

In this assignment you will write a function that takes two images as input and stitch them together. You will work with supplied image *left.jpg* and *right.jpg*. The overall scheme can be summarized as follows:

1. Find the best transformation between input images, i.e., by taking the first image as original (no transform) and estimating the affine transformation of the second image with respect to the first image. Use the image alignment function that you developed in the previous task.
2. Note that images do not have the same size. You need to think about the size of your final panoramic image. How do we determine that? Hint: calculate the transformed coordinate of corners of the *right.jpg*



Figure 3: Transformed images

3. Finally, combine the *left.jpg* with the transformed *right.jpg* in one image. The final output should look like Fig.4.



Figure 4: Stitched image