CS 378 Computer Vision Problem Set 3

CHAO-YEH CHEN

CS account: chaoyeh

I. Programming problem: Image mosaics

1. Computing the homography parameters

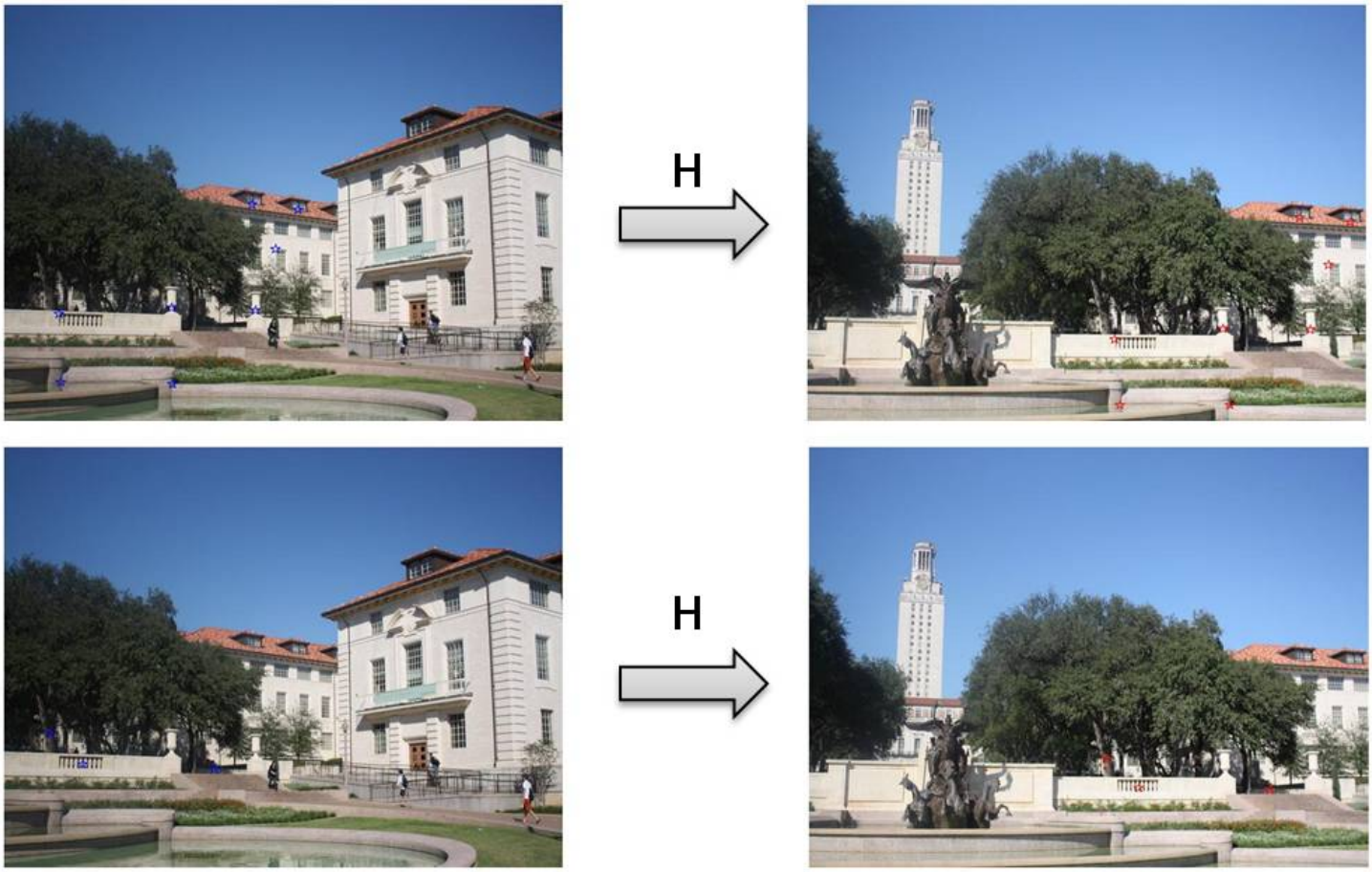


Fig. 1

1	2
3	4

1,2: Corresponding points for two images
3,4: Mapping the testing points by H

Explain:

To reduce the inaccuracy in corresponding points, we need more than 4 points. Here I apply 8 corresponding points. These points are widely spread in both images.

2. Warping between image planes

Fig. 2 Warped image

Explain:

After getting four related points by applying homography matrix to the corners of original image, we find out the pixel values inside the boundary by mapping these points back to the original image with inverse homography matrix.

The image is then shifted according to the axis of the 2nd image.

3. Apply system to the provided pair of images

Fig. 3 Output mosaic

Explain:

This is the best result among several trials. Increasing the number and the accuracy of corresponding points improves the quality of warping.

4. Additional examples of mosaic using other images

A. Swimming pool in Juniper Springs

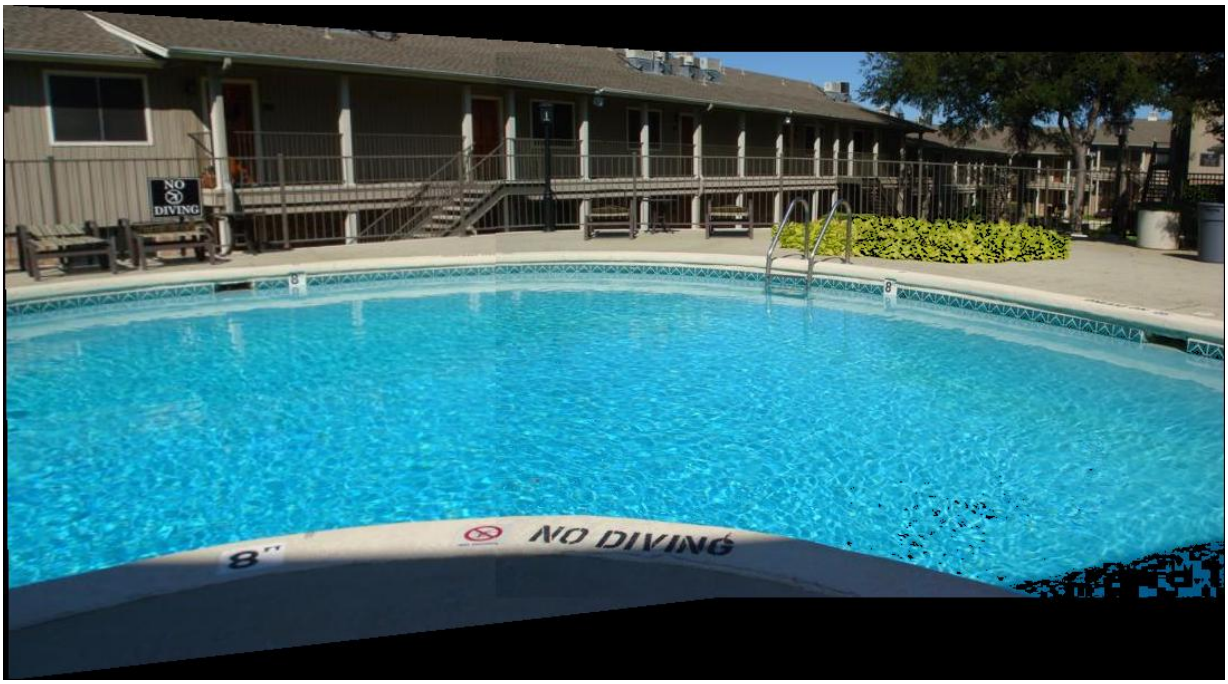


Fig. 4	
1	2
3	

1, 2: original image
3: mosaics

B. Far West

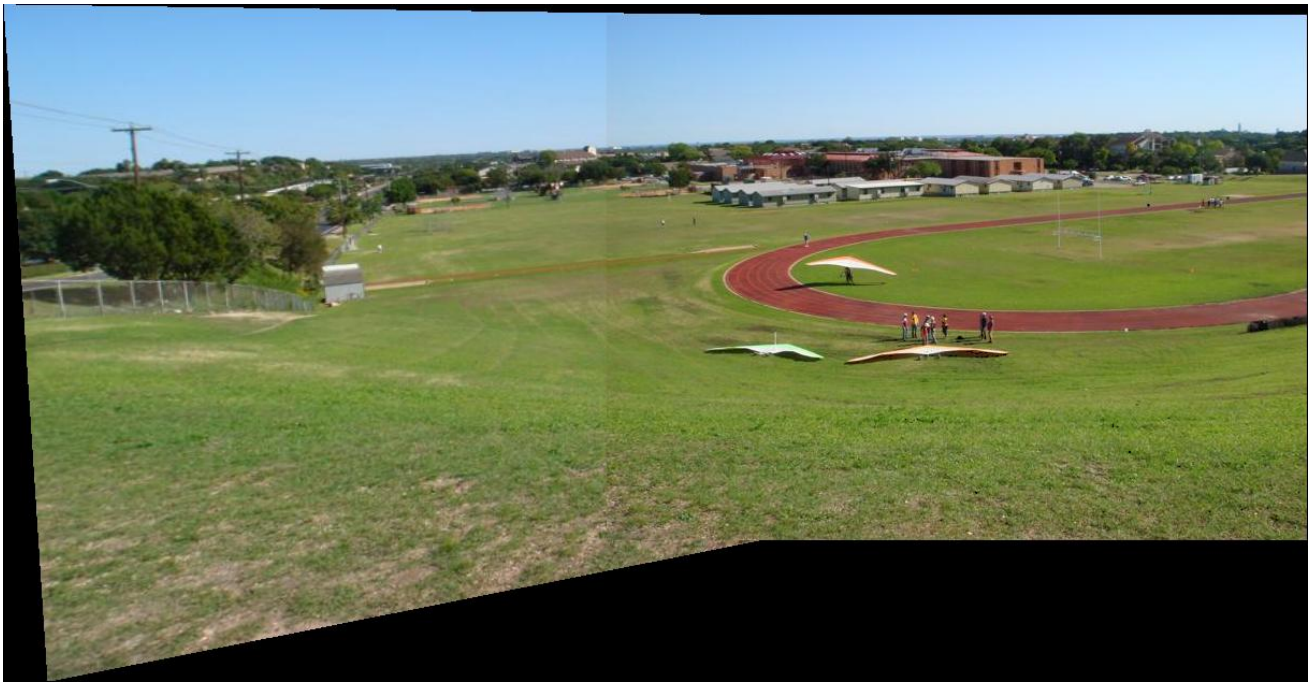
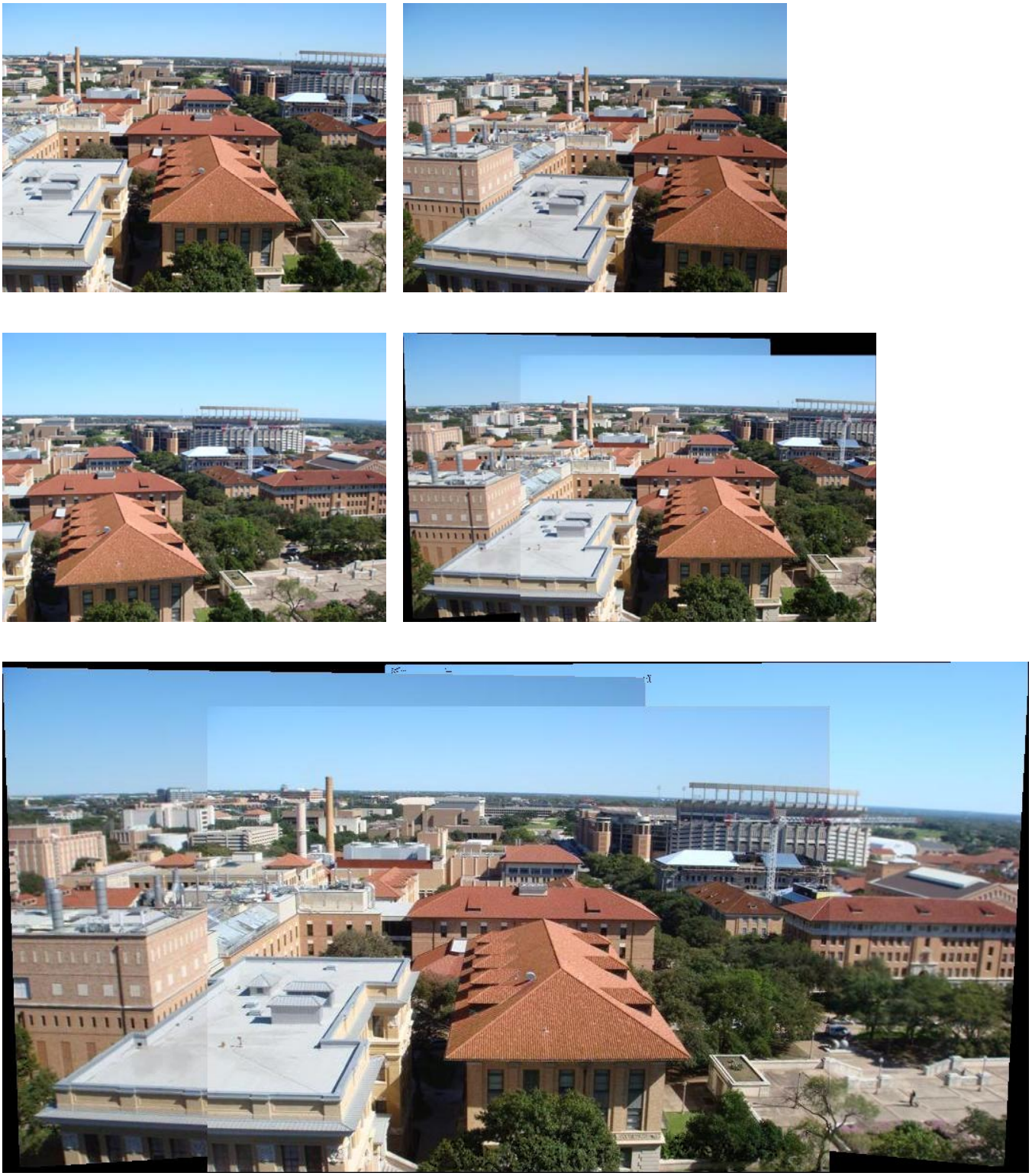


Fig. 5

1	2
3	

1, 2: original image
3: mosaics

C. Views of UT Tower



1	2
3	4
5	

Fig. 6

1, 2, 3: original image

3: mosaics of first two images

4: mosaics of three images

5. Warp one image into a frame region in the second image

A. Calligraphy on the ground

Fig. 7

1	2
3	

1, 2: original image

3: warp 2nd image into 1st image

B. Texas Rangers

Fig. 8

1	2
3	

1, 2: original image

3: warp 2nd image into 1st image

6. Warp one image into a frame region with transparency effect

When warping one image into the other, we can place a different weighting between two images in the overlapping areas. For example:

$I(x,y) = p*(I_1(x,y) + (1-p) *(I_2(x,y))$ where $0 <= p <= 1$

Then the resulting warping image has the property of transparency. The result is similar to project an image onto a wall.

A. UT Tower

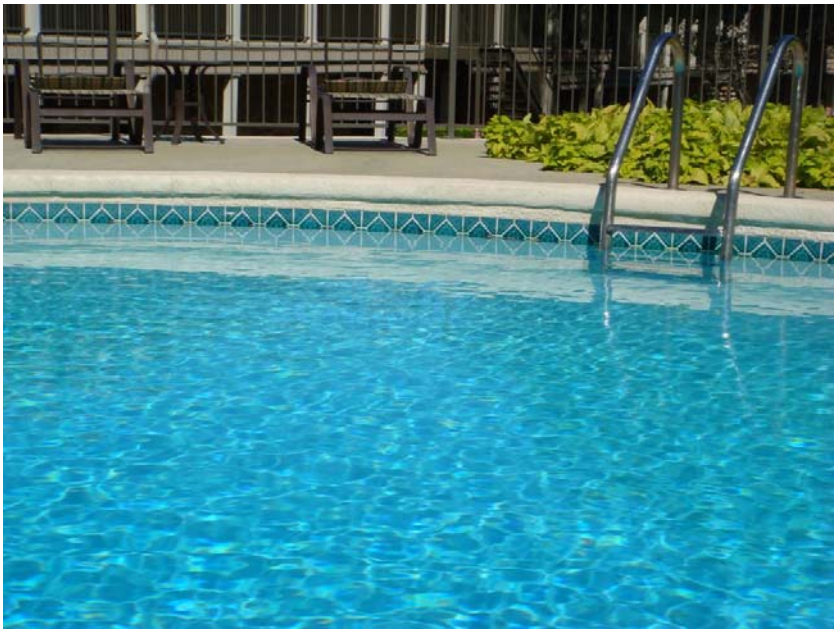
Fig. 9

1	2
3	

1, 2: original image

3: warp 2nd image into 1st image

B. Calligraphy on the swimming pool



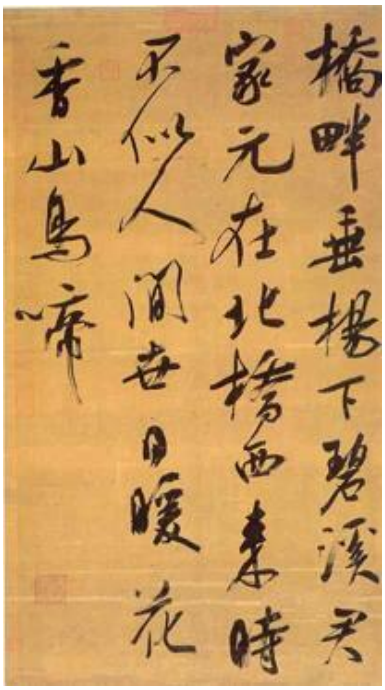


Fig. 10

1	2
3	

1, 2: original image

3: warp 2nd image into 1st image

C. Texture transformation

We can apply this method to change the texture of object with flat surface.

1. Bricks

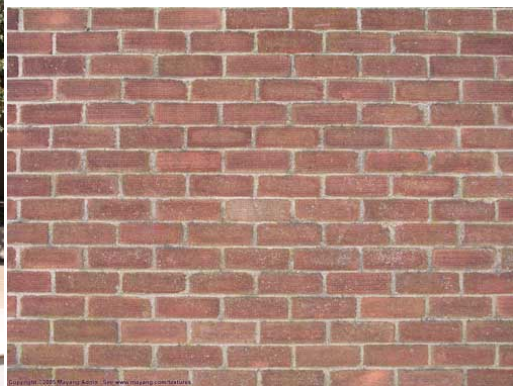


Fig. 11

1	2
3	

1: original image

2: Sample texture

3: warp texture into 1st image

2. Wood

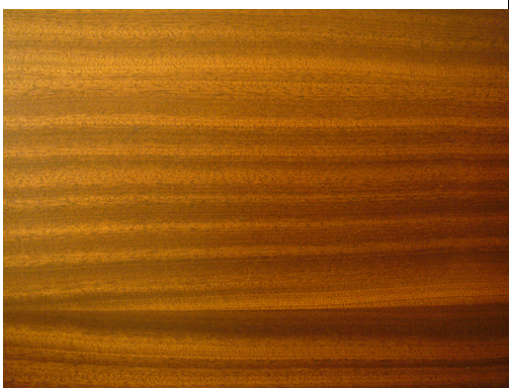


Fig. 12

1	2
3	

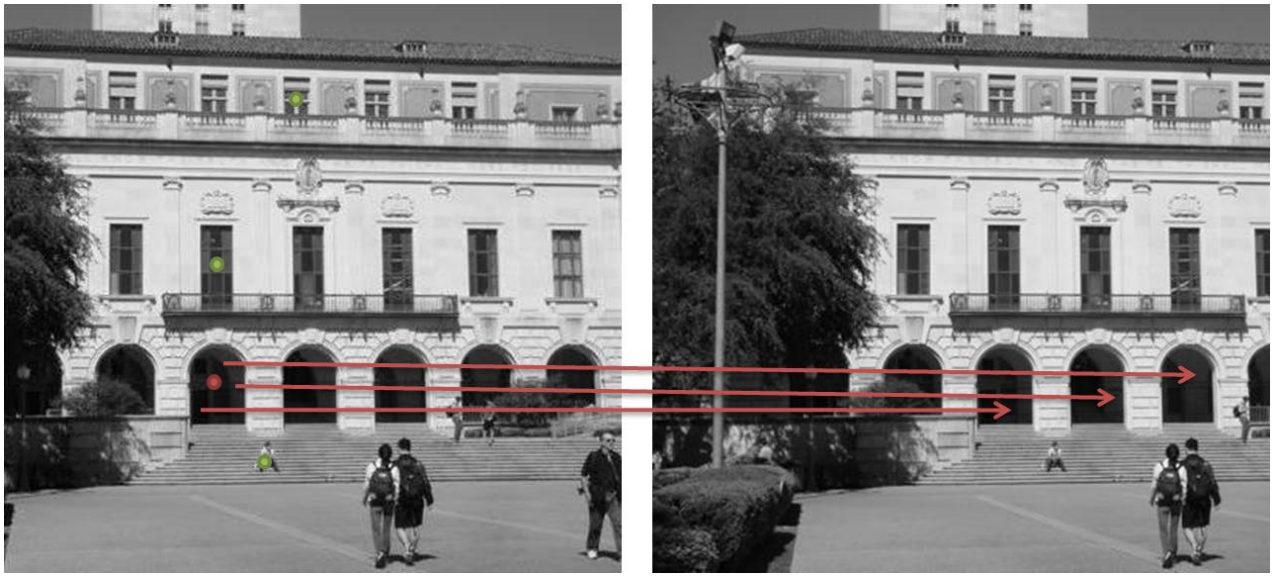
1: original image
2: Sample texture
3: warp texture into 1st image

II. Extra credie

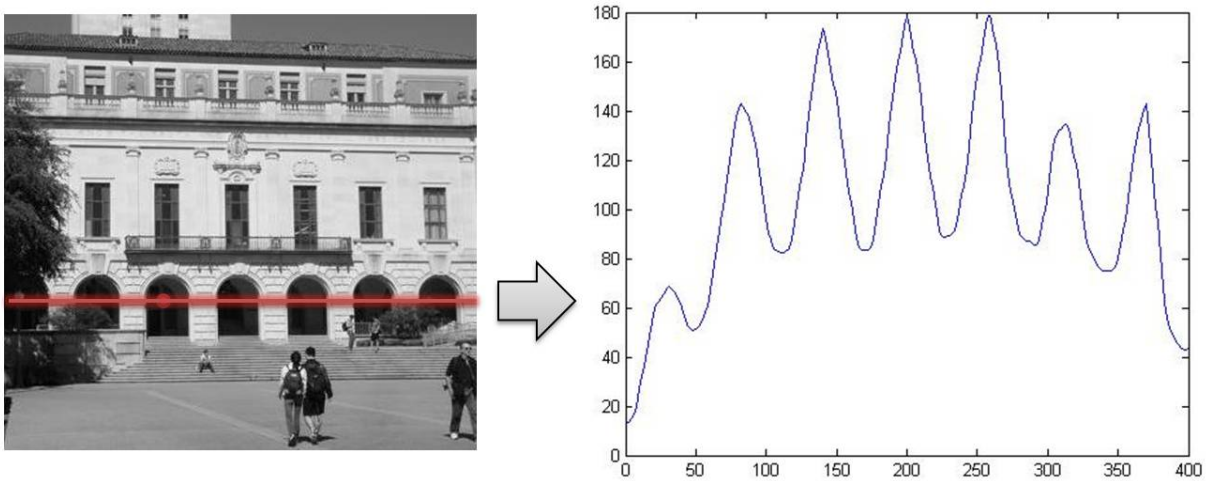
1. RANSAC

A. Problem statement: There is a horizontal shifting between two images, we want to estimate the shifting distance.

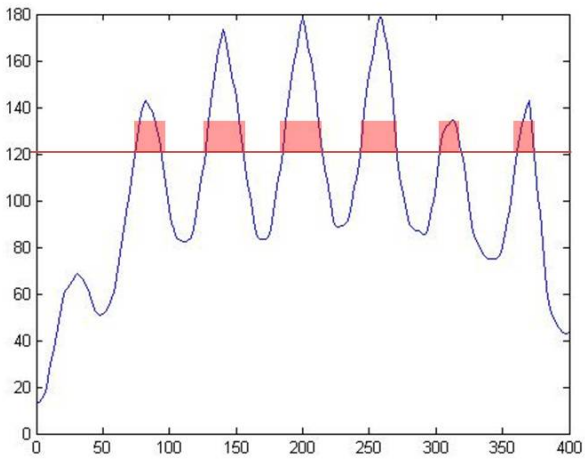
B. Take the red point as corresponding point in the 1st image and build a window with its neighborhoods. Then search along the horizontal line in the second image to find out the candidate corresponding points.



C. **There are six candidate corresponding points in the 2nd image. By plotting the filter response along the red line, we can see roughly six peaks.**

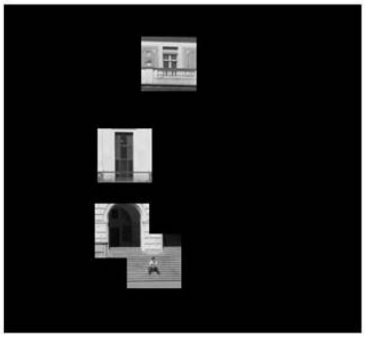


D. **To find the position of six peaks, take the 50 maximum response points (the points inside red interval), and then apply k-means method to find out the six clustering center.**



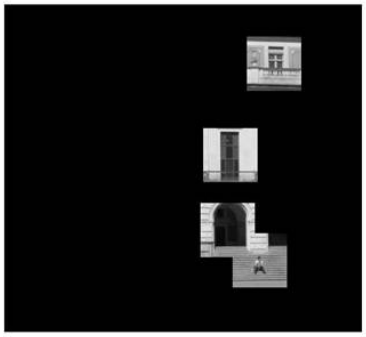
E. **Now we have six candidate corresponding points in the 2nd image. For every corresponding point, we shift some selected window in the 1st image according to related shifting distance and compute the correlation between shifted windows and 2nd image.**

Shift 1



*

Shift 2



*

F. **Determine the estimated shifting distance by the shifting distance related to max correlation between shifted windows and 2nd image.**

G. **The estimated shifting distance is different every time. The ideal result is -90. The experiment result here is – 89.**

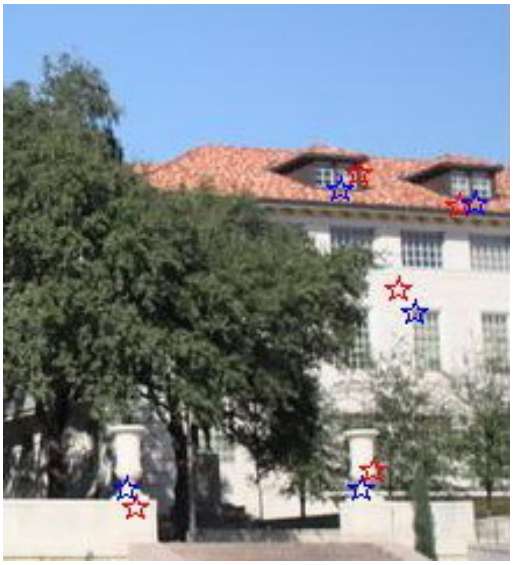
H. **Sometimes the estimated shifting distance is wrong due to wrong selected corresponding point.**

2. Automatic refinement of initial correspondences

The red star in the first image is the selected corresponding points.

The red stars in the 2nd image are the original selected corresponding points. They are slightly away from the expected position. The blue stars are the refinement corresponding points.

The window size is 19. The range of searching refinement points is also 19. If the window size is too large, the refinement location will not be located precisely. When the window size is too small, the chance of finding wrong location is high. In addition, if the range of searching is too large, it is possible that we will find another similar pattern in the image.



3. Image rectification

A. Bass Concert Hall

1: original image
2: rectified image

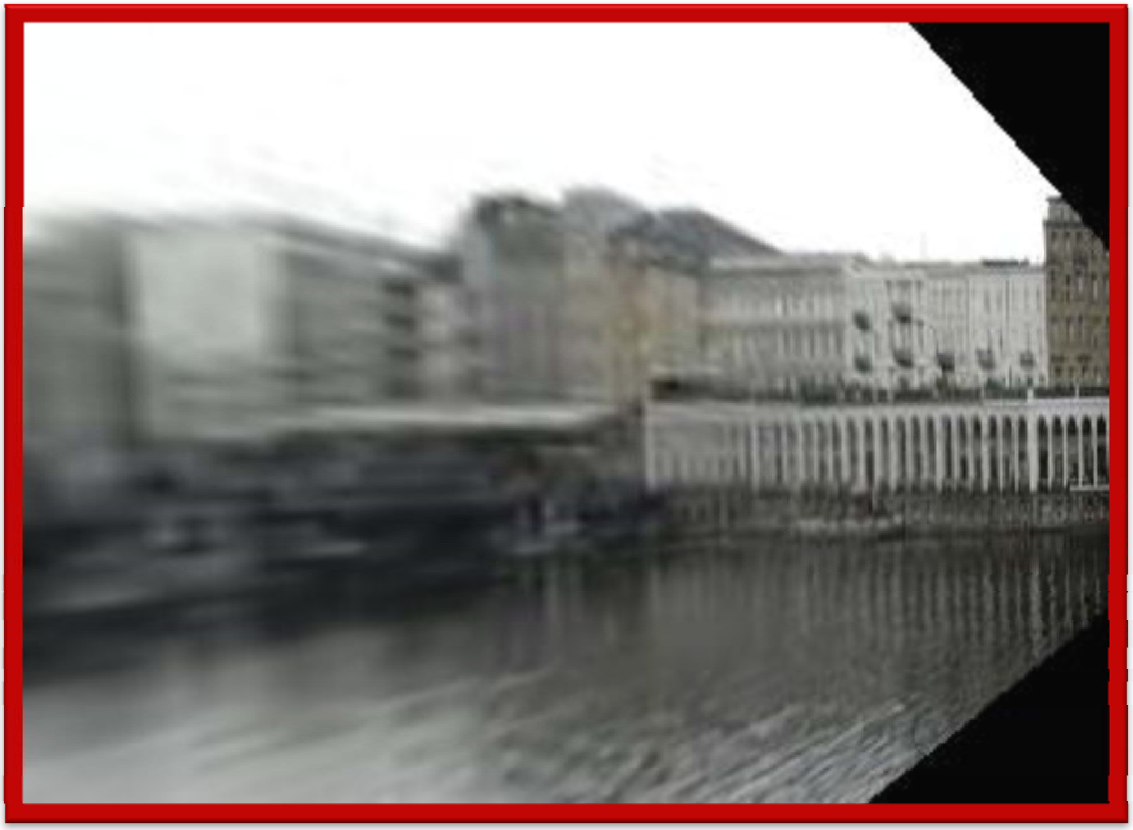
B. Gym

1: original image
2: rectified image

C. Venice

1	2
3	

1: original image
2: rectified image
3: enlarge of are inside red rectangular



Explain:

The size of rectified image is much larger than original image. We can find the clearly rectified part inside the rectangular. This result shows that we can't see an area clearly if there isn't enough information in original image. The rectified image is blurred due to warping a small area to a large area by interpolation.

4. Artificial boundary elimination

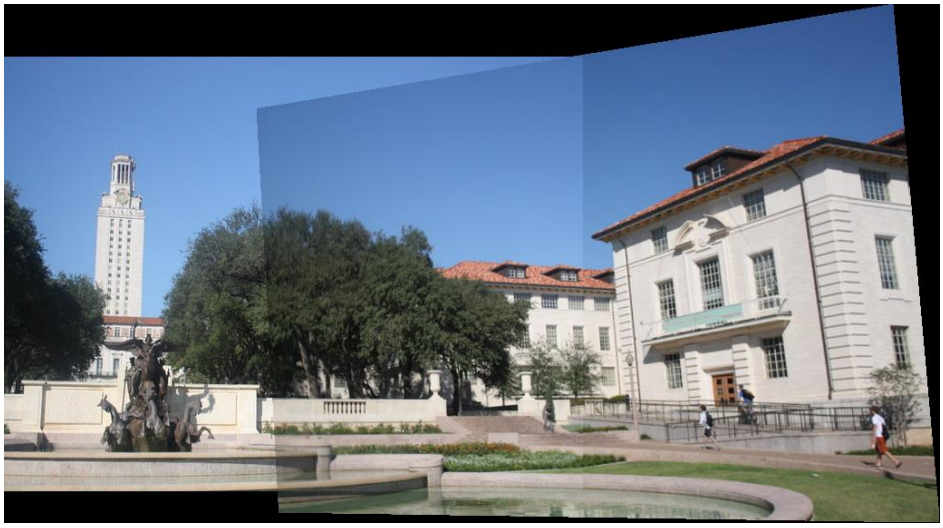
The original mosaic shows very clear boundary between two images. It can be eliminated by "blending" two images in the overlapped area.

For the original mosaic:

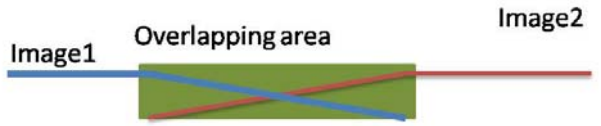
The pixel' value in the overlapping area is determined by image one

Now we overlap two images by averaging pixel values of two images

The pixel' value in the overlapping area is determined by averaging the pixel values from two images



The averaging method still cannot eliminate the artificial boundary. Now we apply the linear weighted averaging method.



The pixel' value in the overlapping area is determined by weighted average of pixel values from two images

The weighted average method eliminates the boundary successfully. The two images is blending together and there is no artificial boundary along the x direction. With similar procedire, boundary in the y direction can also be removed.

The cutted mosaic image with weighted average blending method

How to execute the code

1. pset3 % basic warping and mosaic
2. pset3_rectify % rectify
3. ransac % for ransac testing