MAEG5720

Computer Vision in Practice

Mini-Project2

Date: 26th, November 2021

Due Date: 10th, December 2021

Aim: The field of view of the image is limited by your lens. However, we could solve this issue by combining multiple images together to form a panorama. The aim of this assignment is to automatically stitch the images acquired by a panning camera.

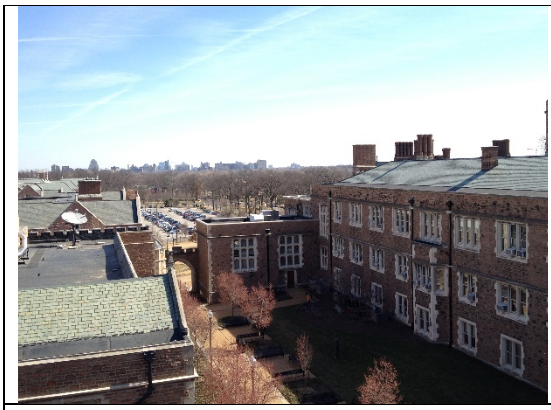
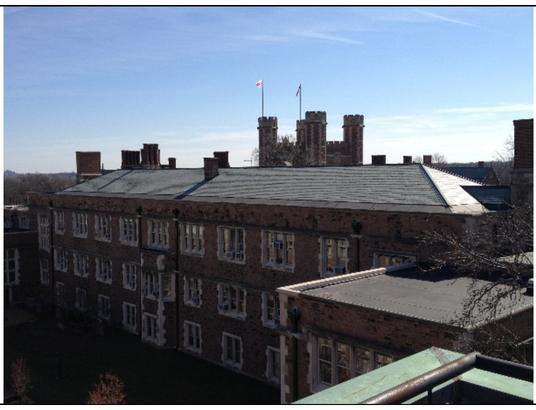
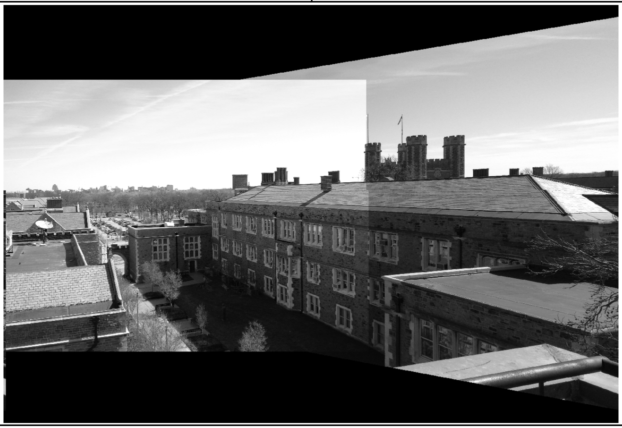
	

Image 1	Image 2
	
Stitched Image	

The Algorithm outline:

1. Choose one image as the reference frame
2. Estimate the homography between reference image and the other image
   - Detect the local features and extract the feature descriptor for each feature point in both images
   - Matches the feature descriptors between two images
   - Robustly estimate homography using RANSAC
   - Warp the image using the homography into the reference frame.

MAEG5720

Computer Vision in Practice

Tips and detailed description of the algorithm:

(1) Choose the reference image: Since we are working with two images, you are ***free*** to choose the reference frame from any image.

Possible extension (Not in this assignment): In case you would like to extend this algorithm to multiple images in your spare time, you can choose the middle frame as the reference frame to minimize the distortion of the images.

Also, to ensure you have larger area of overlapping, you can use chain homographies. For example: If you wish to calculate the homography between image 2 to image 4, you can separate it into two steps. i.e. $\mathrm{H}_{24} = \mathrm{H}_{23} * \mathrm{H}_{34}$

(2) Estimation of homography

(a) Detecting local features in each image

To extract the local features of the image, one of the common methods is to use SIFT descriptor. Since SIFT descriptor is a patented technology and it is not available in MATLAB, please download the library from VLFeat.org (https://www.vlfeat.org/download.html) and follow the instruction on the installation page (https://www.vlfeat.org/install-matlab.html).

Once the setup is done, you can use the following code for SIFT descriptor extraction.

```
I = single(rgb2gray(I));
[f,d] = vl_sift(I) ;
```

where

f is a $4 \times num\_descriptor$ matrix consists of x, y, σ, θ of each SIFT point
d is a $128 \times num\_descriptor$ matrix

please refer to the Shape Descriptor lecture notes for detail.

(b) Matching the descriptors between images

Once the SIFT descriptors are extracted, you have to find the pairs of descriptors between images that looks similar. This will probably a corresponding pair. One of the methods to do is to compare the sum of square distance between two descriptors and adopt the near neighbour distance ratio (NNDR) of (say 1.5)

$$NNDR = \frac{distance\ of\ Second\ Match}{distance\ of\ First\ Match}$$

The pseudo code below gives you some hints on the algorithm outline:

```
function [matches scores] = match_descriptor(d1, d2)

%Define the threshold (for example 1.5)
Num_of_matches =0

for each i in d1
    for j each d2
        calculate the SSD of descriptor i and j
            $SSD(pi,pj) = \sqrt{(D1_{pi} - D1_{pj})^2 + (D2_{pi} - D2_{pj})^2 + \ \dots + (D128_{p1} - D128_{pj})^2}$

        find the best match and second best match
        if NNDR>threshold
            add the i and j into matches[num_of_match]
            add the $SSD(pi,pj)$score into scores[num_of_match]
            num_of_match=num_of_matches+1
        end
    end
end

return matches and scores
```

(c) Robust Estimation of the homography by RANSAC

Random samples of 4-points to compute each homography hypothesis. You will need to write the following function

$$H = FindHomography(x1, y1, x2, y2, x3, y3, x4, y4, xp1, yp1, xp2, yp2, xp3, yp3, xp4, yp4)$$

Which takes the 4 SIFT points from image 1 and 4 corresponding SIFT points from image 2. In order to compute the elements of H, You will need to setup the linear system of equations (i.e. $\mathrm{A}h = 0$). There are totally 8 elements for the 3x3 homography matrix. Each pair of correspondence gives you 2 equations and therefore you need totally 4 pairs of correspondences. The A matrix is of the form below:

```
A=[
-x1  -y1  -1   0    0    0   x1*xp1   y1*xp1   xp1;
 0    0    0  -x1   -y1  -1   x1*yp1   y1*yp1   yp1;
                            :
                            :
                            :
-x4  -y4   -1  0    0    0   x4*xp4   y4*xp4   xp4;
 0    0    0  -x4  -y4  -1   x4*yp4   y4*yp4   yp4];
```

The solution can be obtained using SVD of A. The solution is the right singular vector corresponding to the least singular value.

```
[U,S,V] = svd(A);
H=V(:,end);
H=reshape(H,3,3)';
```

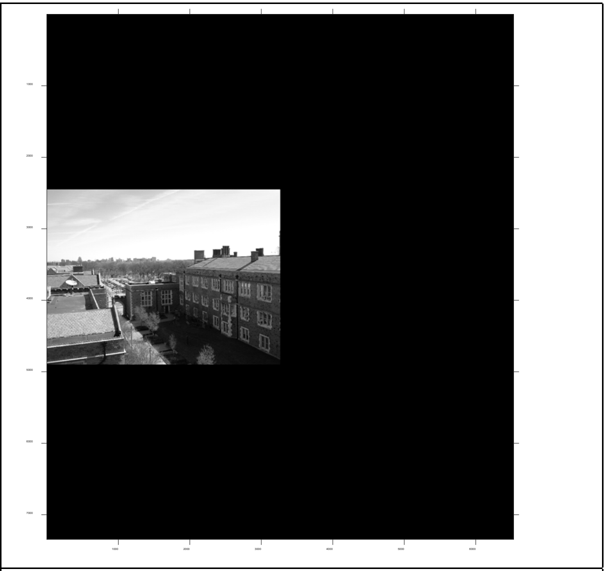
Please refer to Lecture note 9 PP60 for more detail.

For the RANSAC, you can just use a very simple implementation with a fixed number of sampling iteration ~~is sufficient~~. You should output a single homography matrix H with the greatest number of inliners. The pseudo code for the RANSAC is given below:

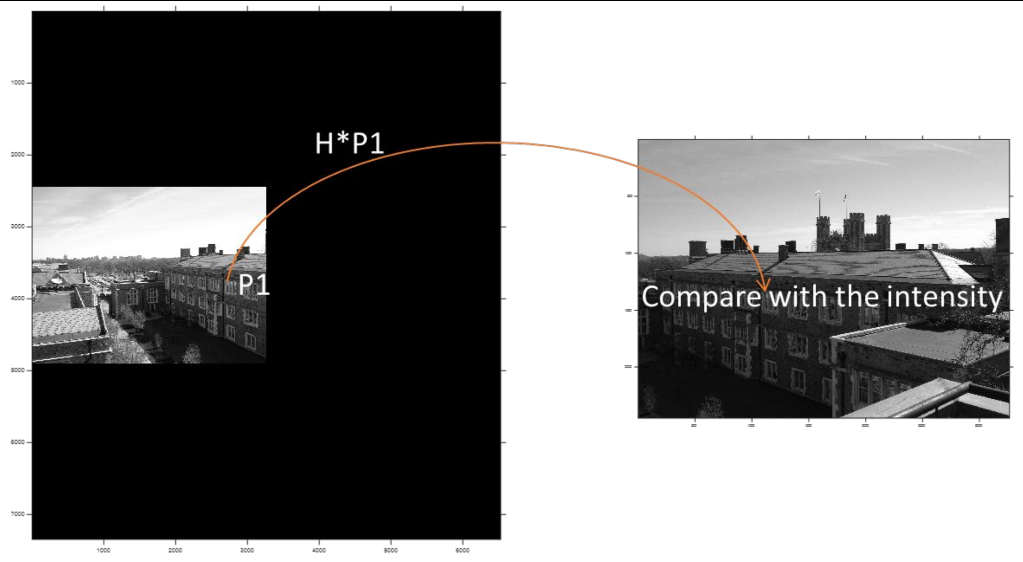
```
function [bestH, num_of_inliners] = RANSAC ( f1, f2, matches)
define the parameters (PixelError, num_Iteration, num_pts_cal_H=4)
max_inliner =0;

for each i in num_Iteration
    random select 4 points and compute H_hypothesis with $FindHomography$ function
        reset inliner;
        for each  pair in the matches list
            find_out the coordinate for each matches $P1$ and $P2$
            use the H_hypothesis to compute the warpped $P1 \rightarrow P2_{Hom}$
            if $distance(P2_{Hom}, P2) < Pixel\_Error$
                inliner =inliner+1
            end
        end
        if inlier>max_inliner
            bestH = H_hypothesis
            number_of_inliners = inliner
        end
end
```

(d) Warp the image using the best homography into the reference frame

Once you obtained the best homography, the last step is to warp the image and stich the image together. You will need to extend the images in the reference frame to accommodate the other image. One of the methods is to use padarray function in MATLAB to pad zeros into the reference frame.

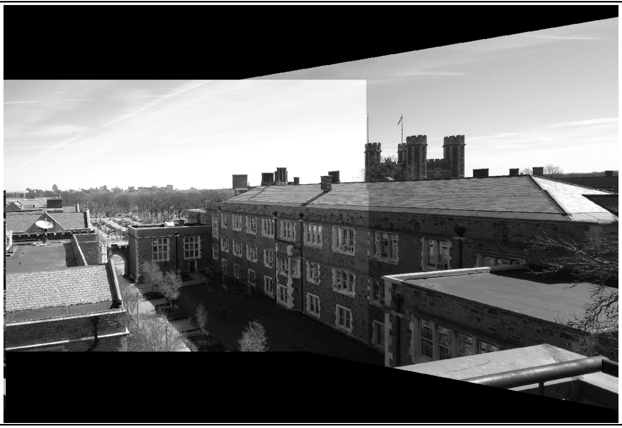
	

Then we use the similar approach by backward projection. This could avoid holes in the warpped image. For each pixel in the reference image, calculate the corresponding pixel location in the second image using the best homography obtained by the RANSAC function. You can compare the value of the wrapped pixel with the corresponding value in the other image and simply pick the pixel value with maximum value from both image. This tends to produce less artifacts than taking the average of the warped images.



```
function stitchedImage = stitch(im1, im2, homography)
copy the reference image (im1) to the stitchedImage
pad the stitchedImage with zeros
for each i column in stitchedImage)
    for each j row in stitchedImage
        calculate the corresponding position in im2 using homography
        p2 = homography* p1
        if (p2 is within range of stitchedimage)
            Take the higher intensity out of the p1 and p2 as the pixel value
        end
    end;
end;
Crop the image to remove extra boundary which are black(pixel value=0)
```

The final result should look like this.



What to submit:

(1) Please submit the following MATLAB .m files
- main.m for the main function and read in the image for processing. (20%)
- match_descriptor.m for matches the descriptors (20%)
- FindHomography.m for solving the homography using 4 correspondences (20%)
- RANSAC.m for robust estimation of the best homography between two images (20%)
- Stitch.m for stitching the images together.(20%)

- (Optional) You may wish to stitch more than 2 images. Please submit another function main_multiple_images.m and show the result (10% Bonus)

(2) A pdf file for the result and discussion you would like to share.

You will have two weeks for this assignment and tutorial will be arranged to help you do this assignment. The tutorial time will be announced by Mr. Li Bin soon. Good luck!