



Faculty of Engineering

EE5731 Visual Computing
Assignment 1: Panoramic Image Stitching

Name: Liu Xingyu
Matric No: A0116430W

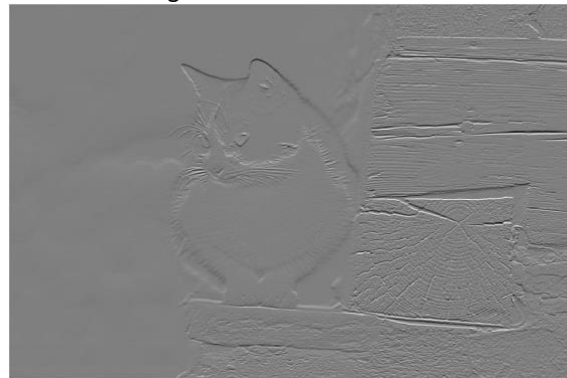
Part 1: 2D Convolution

Original image:



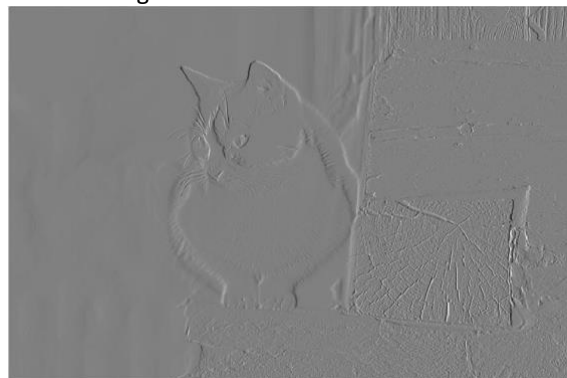
Sobel horizontal kernel: $[-1, -2, -1; 0, 0, 0; 1, 2, 1]$

This kernel is for detecting the horizontal edges. See below.



Sobel vertical kernel: $[-1, 0, 1; -2, 0, 2; -1, 0, 1]$

This kernel is for detecting the vertical edges. See below.



Combined above sobel vertical and horizontal kernels:

By combining the sobel vertical and horizontal kernels, it is used for edges detecting. See below.



Gaussian kernel 1: [1, 2, 1; 2, 4, 2; 1, 2, 1]

This kernel is for blurring the image. See below.



Gaussian kernel 2: [1, 4, 7, 4, 1; ...
4, 16, 26, 16, 4; ...
7, 26, 41, 26, 7; ...
4, 16, 26, 16, 4; ...
1, 4, 7, 4, 1]

This kernel is also for blurring the image, but more blurring effect than gaussian kernel 1. See below.



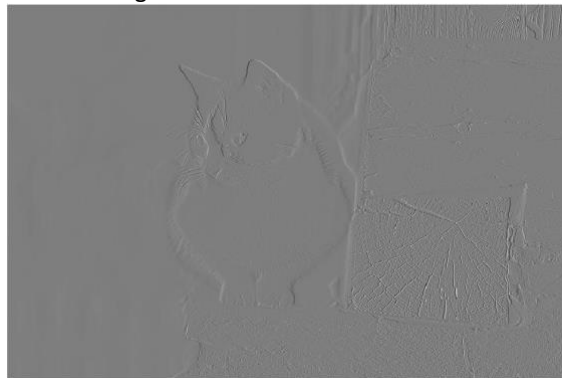
Gaussian kernel 3: [0, 0, 0, 5, 0, 0, 0;...
0, 5, 18, 32, 18, 5, 0;...
0, 18, 64, 100, 64, 18, 0;...
5, 32, 100, 100, 100, 32, 5;...
0, 18, 64, 100, 64, 18, 0;...
0, 5, 18, 32, 18, 5, 0;...
0, 0, 0, 5, 0, 0, 0]

This kernel is also for blurring the image, but more blurring effect than gaussian kernel 1 and gaussian kernel 2. We can conclude that when increasing the size of the gaussian kernel, the blurring effect also is increasing. See below.



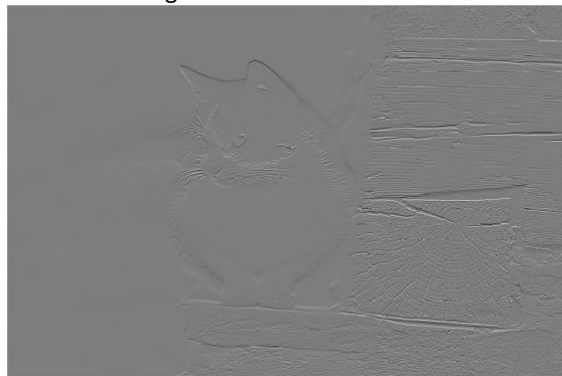
Haar kernel 1: [-1, 1]

This kernel is for detecting the vertical edges. See below.



Haar kernel 2: [-1; 1]

This kernel is for detecting the horizontal edges. See below.



Haar kernel 3: $[1, -1, 1]$

This kernel is for detecting vertical line edges that has strong color difference. See below.



Haar kernel 4: $[1; -1; 1]$

This kernel is for detecting horizontal line edges that has strong color difference. See below.



Haar kernel 5: $[-1, 1; 1, -1]$

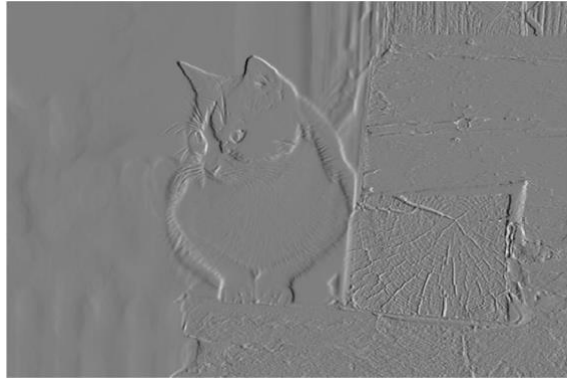
This kernel is for detecting slanted edges. See below.



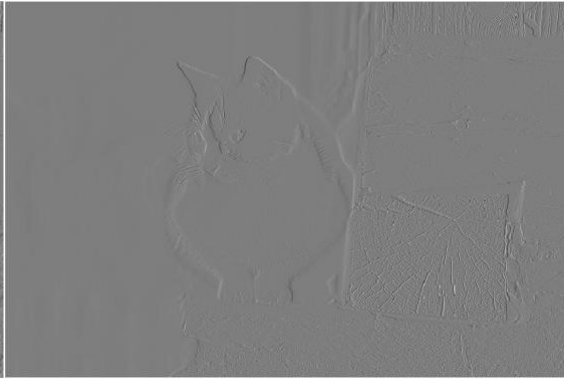
Haar kernel 6: [-1, -1, -1, 1, 1, 1;...
-1, -1, -1, 1, 1, 1;...
-1, -1, -1, 1, 1, 1]

This kernel is also for detecting vertical edges comparing to haar kernel 1 [-1, 1] but with larger kernel size. We can see that when increasing the size of haar kernel, it focuses more on the edges and the contrast between the edges and the rest is more obvious. See below for comparing.

Haar kernel 6



Haar kernel 1

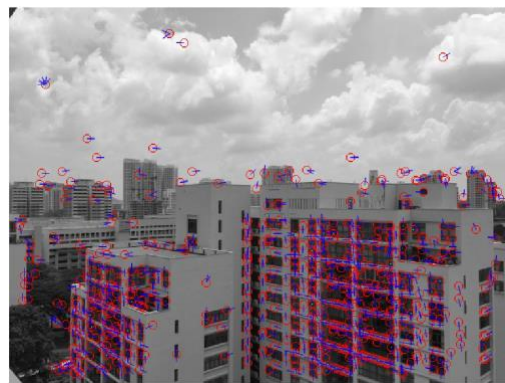


Part 2: SIFT Features and Descriptors

Original image:



Key points on the image:



This SIFT code main function takes gray scale image, number of octaves, number of scales per octaves and initial value for sigma as input. And returns key points inside a cell array as output.

Every key point is created and stored in a cell array which is an object. And each key point contains below:

coordinates () : [x, y] coordinate of the key point on image.

direction () : general direction of the key point.

magnitude () : magnitude of general direction vector.

octave () : number of the octave which the key point extracted from.

scale () : sigma value which the image is convolved with.

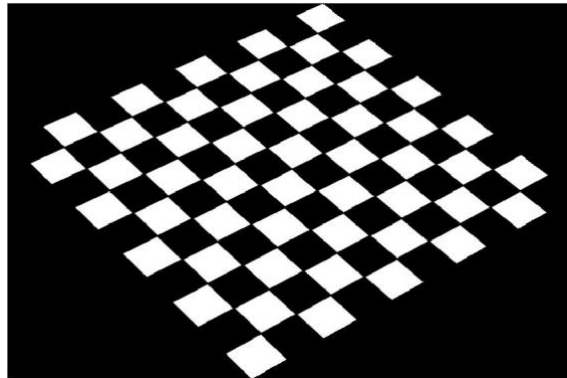
descriptor () : a vector containing the descriptor.

SIFT keypoint code reference:

<https://github.com/aminzabardast/SIFT-on-MATLAB>

Part 3: Homography

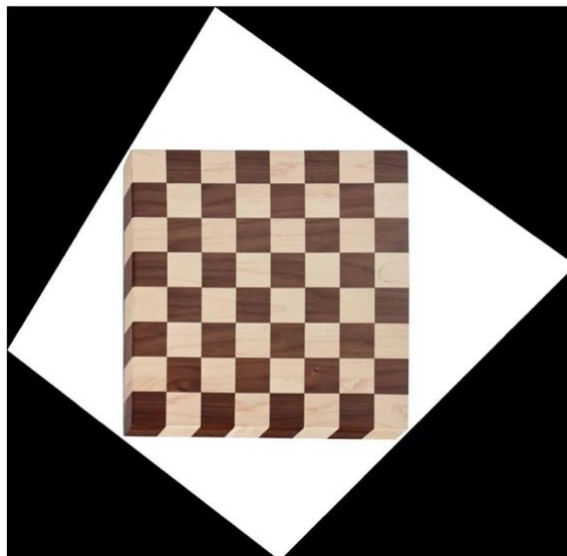
Transform from h1 to h2:



Homography matrix:

| | | |
|-----------------------|-----------------------|---------------------|
| 0.00203167906083710 | 0.00122619927482271 | 0.0671747345696785 |
| -0.000791701851303065 | 0.00110243037291573 | 0.997725988229764 |
| 4.71748268068702e-07 | -4.77231045817034e-07 | 0.00478894737583979 |

Transform from h2 to h1:



Homography matrix:

| | | |
|-----------------------|----------------------|---------------------|
| 0.00242170779215301 | -0.00248424660616874 | 0.483596840860406 |
| 0.00179329663469115 | 0.00408044437089656 | -0.875271597043633 |
| -5.98502355062378e-08 | 6.51344340126909e-07 | 0.00135086238428662 |

Part 4: Manual Homography + Sticthing

Original images im01 and im02:



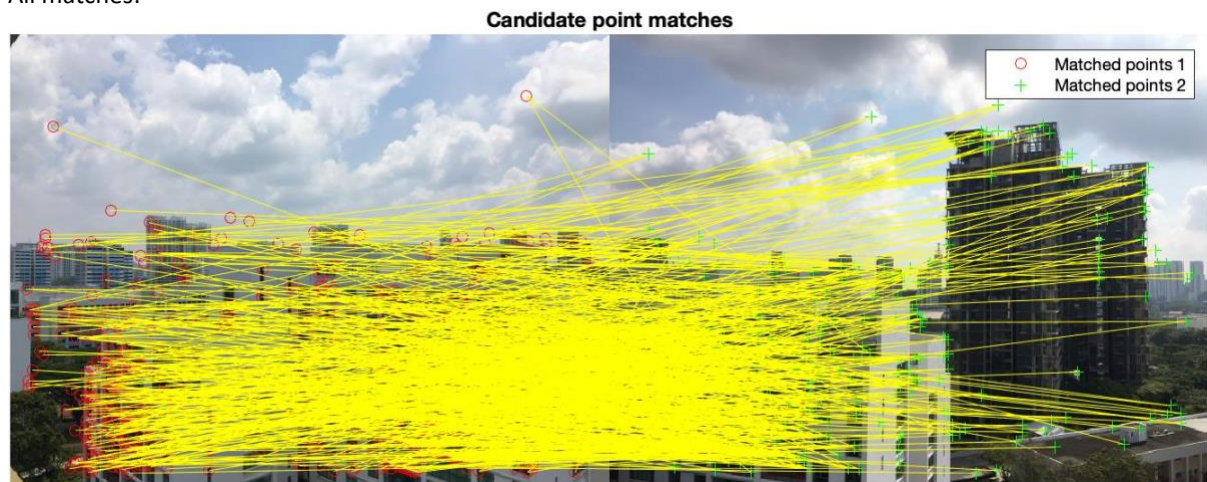
Stitched image:



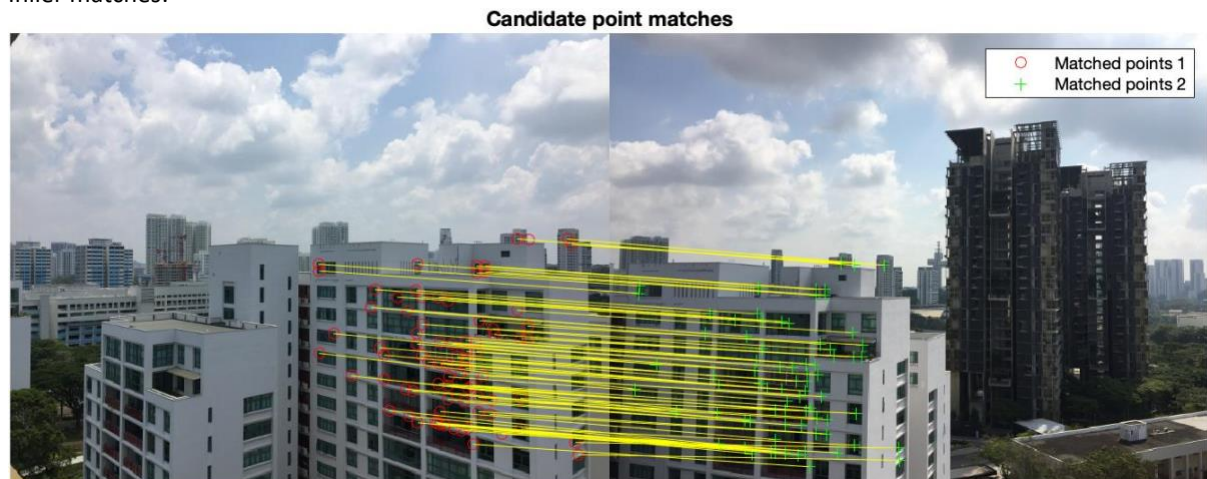
The effect of double edges is due to that the middle part is the overlapping of the two images. The two images were taken from different perspective. So when stitch these two images, the two images cannot be 100% overlapped and there will be double edges shown on the stitched image.

Part 5: Homography + RANSAC

All matches:



Inlier matches:

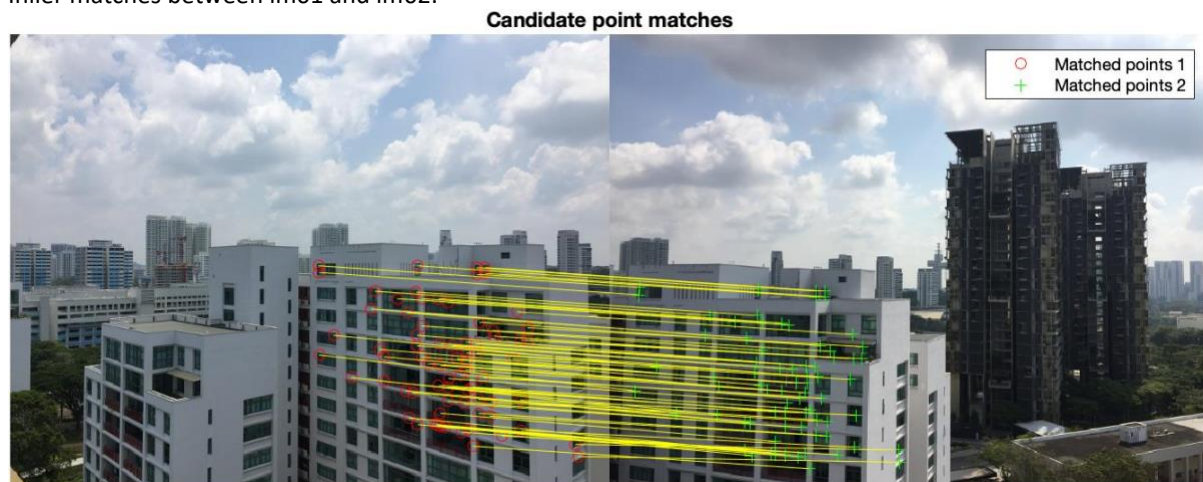


Stitched image:

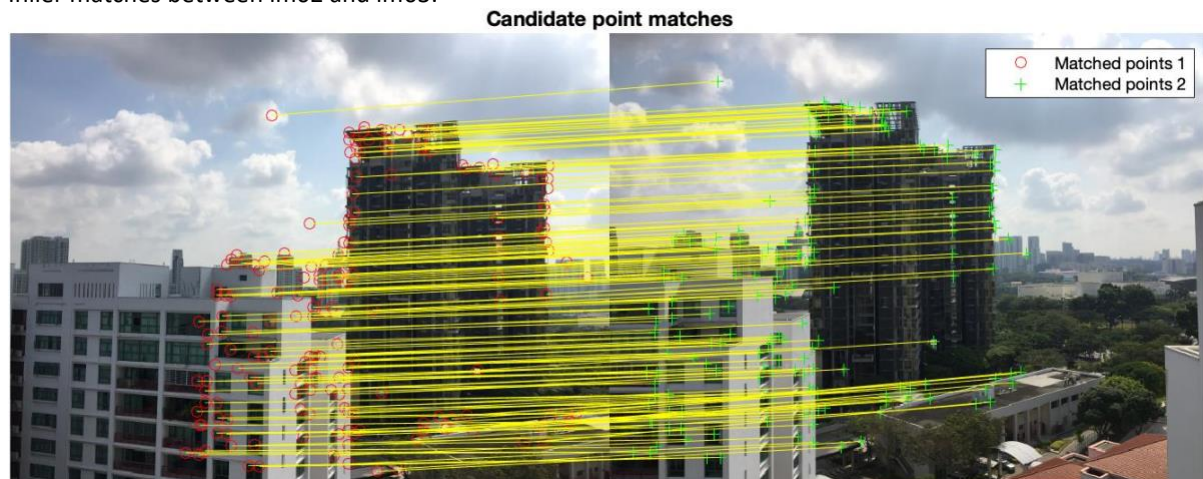


Part 6: Basic Panoramic Image

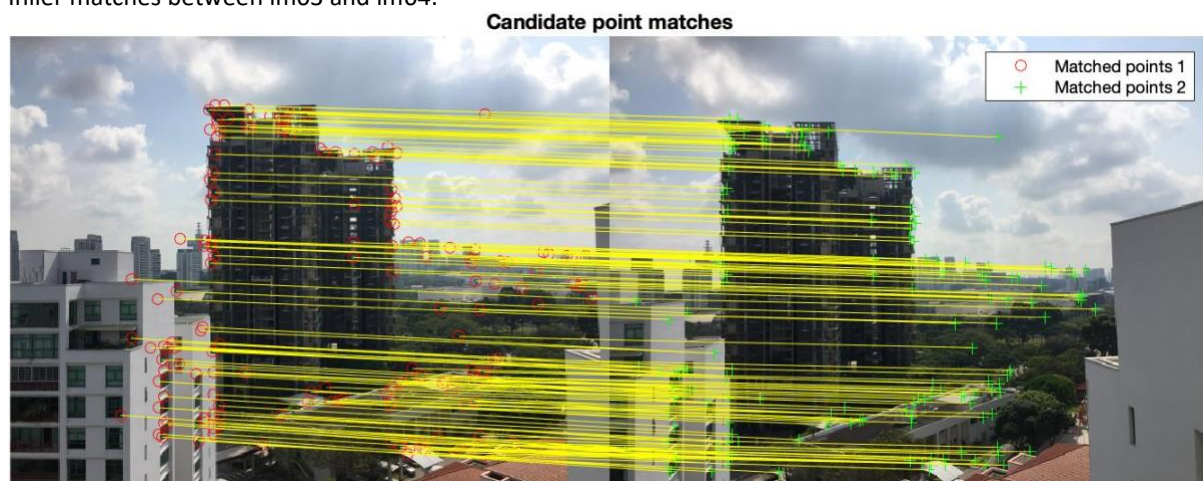
Inlier matches between im01 and im02:



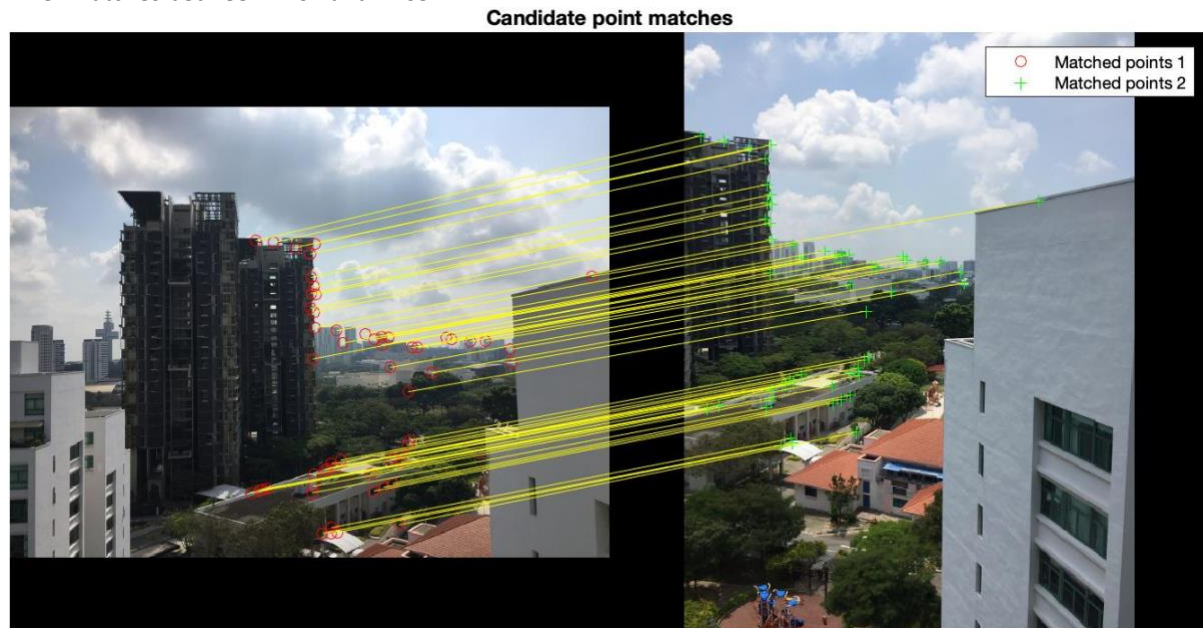
Inlier matches between im02 and im03:



Inlier matches between im03 and im04:



Inlier matches between im04 and im05:



Final stitched image of im01, im02, im03, im04, im05:



Reference:

SIFT keypoint code reference: <https://github.com/aminzabardast/SIFT-on-MATLAB>