****

|  |
| --- |
| **Computer Vision** |
| **Feature Extraction** |

Name: hanxue Liang

Student ID: 16-912-156

Catalog

[1. Feature extraction 3](#_Toc527020205)

[2. Feature descriptor 4](#_Toc527020206)

[3. SSD Feature matching 4](#_Toc527020207)

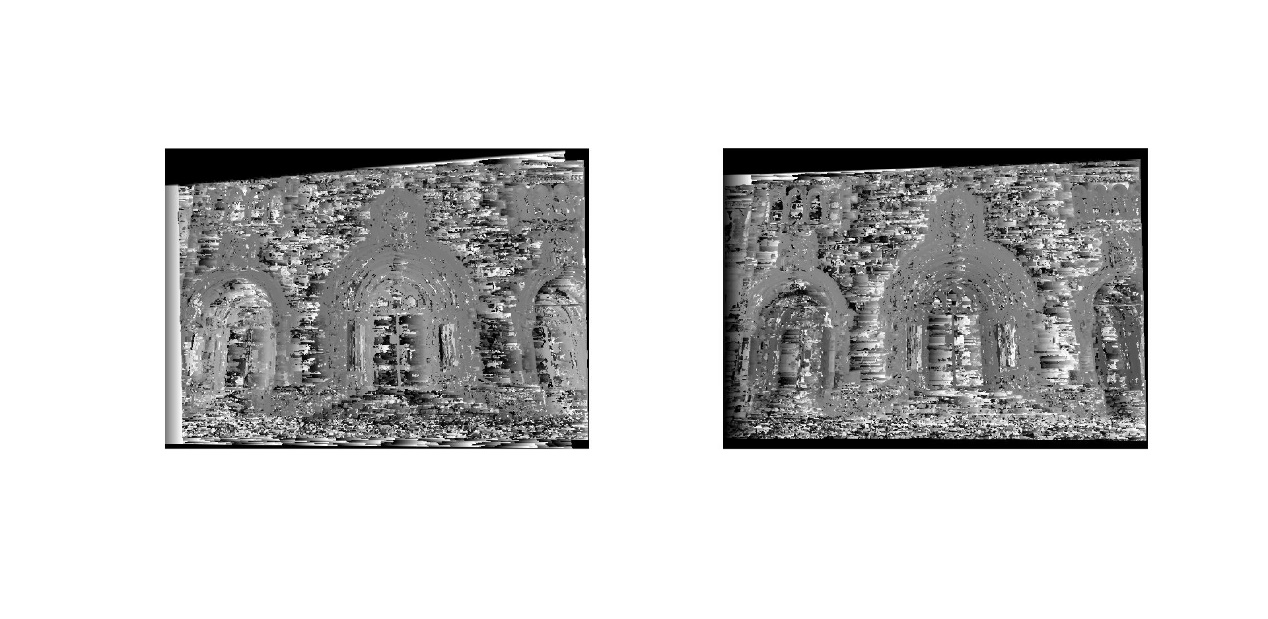
[4. SIFT features 5](#_Toc527020208)

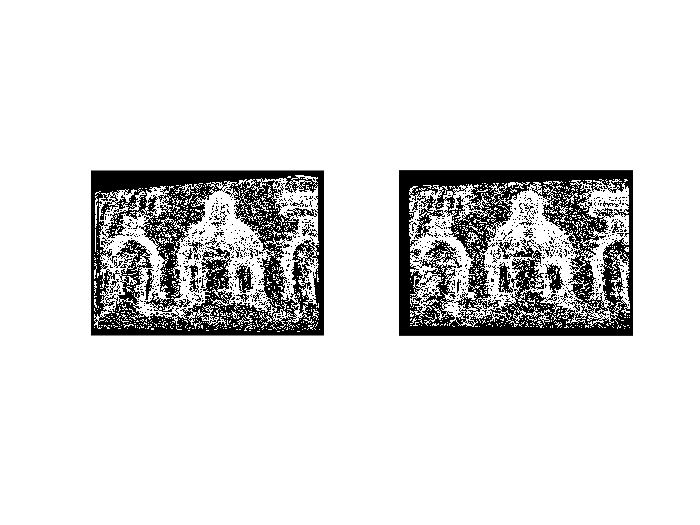
[5. Conclusion 6](#_Toc527020209)

In this exercise, I have implemented a winner-takes-all and a graph cut algorithm to compute the disparity map. I also realize the automatically compute the disparity range by using sift matches method. The textured 3D models are constructed using Meshlab.

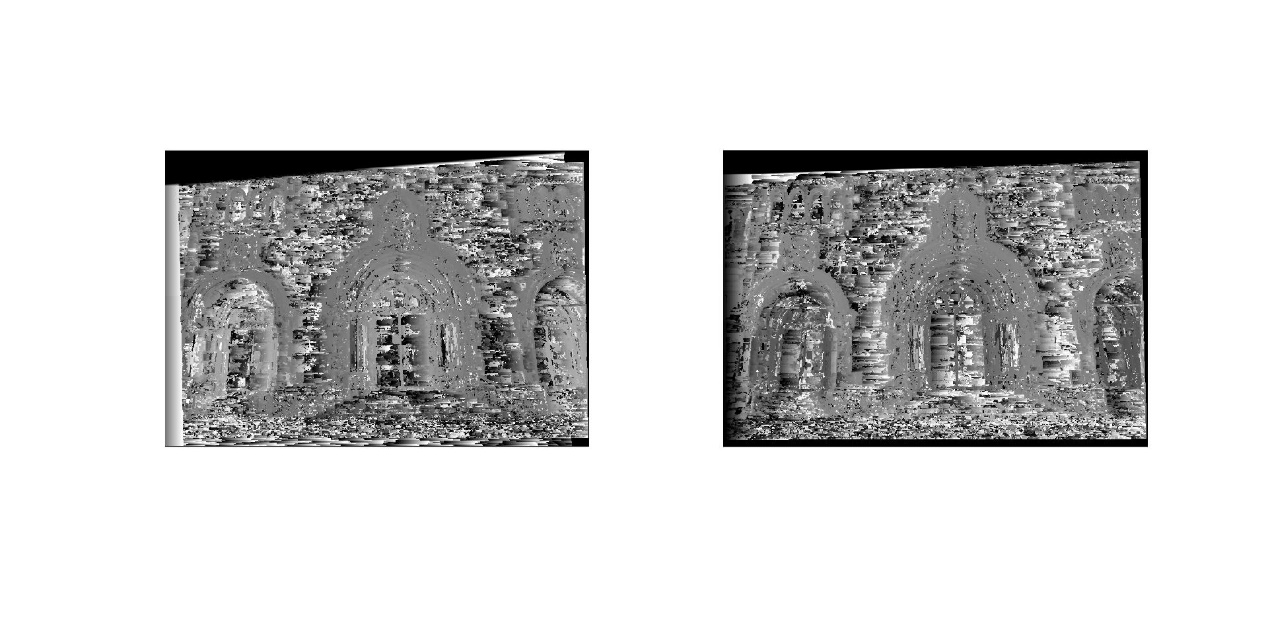
## Disparity computation

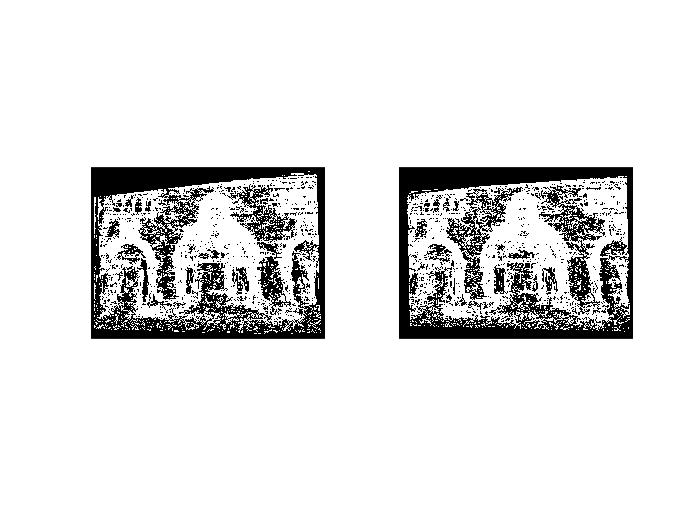
In this part, I implement winner-takes-all stereo using SSD. First I decide the disparity range by manually clicking the points. Then within this range, I use a loop function over all possible disparity value and find the ‘winner’ disparity value for each pixel. In each iteration, I shift image1 by the respective disparity value, and the calculate SSD error between the shifted image and image2. Then I convolve the result with an average filter. The ‘winner’ disparity value for a pixel is chosen as the one which provide the smallest SSD error for that pixel. And the disparity maps are shown below. Here, I will compare the disparity map computing from different convolving filter.



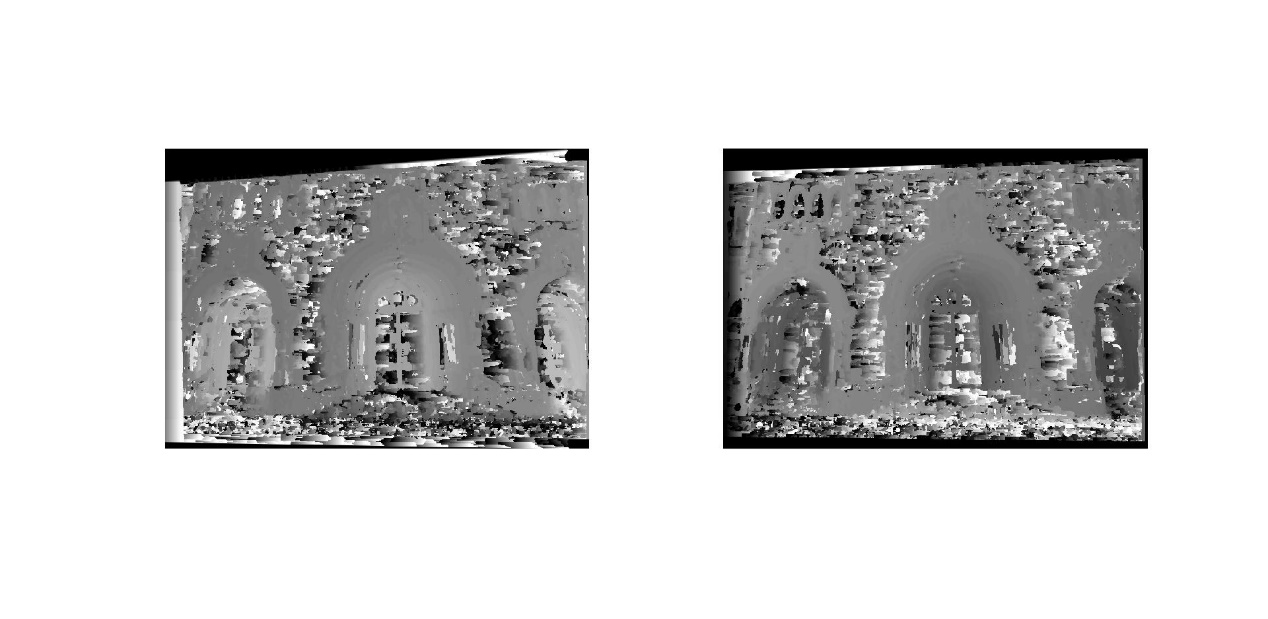


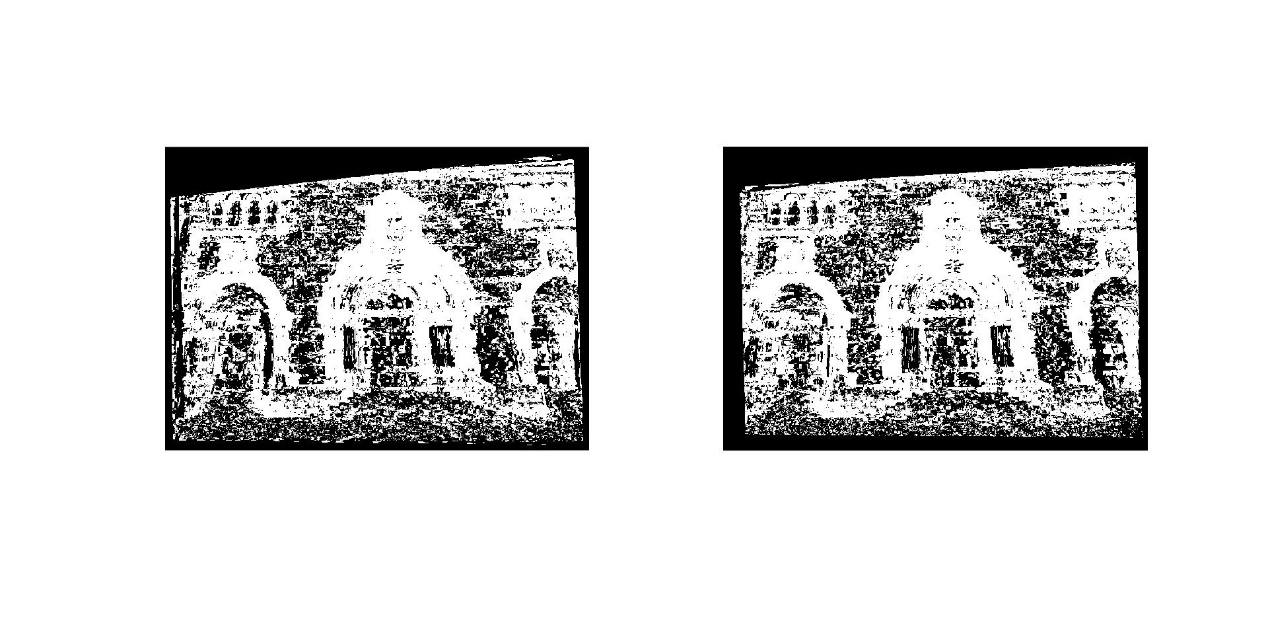
WTA result with 3\*3 window





WTA result with 5\*5 window





WTA result with 7\*7 window

Figure1. Disparity map using WTA method

It can be found that WTA method could provide a good disparity map. When we improve the size of the average filter, a bigger part of neighboring pixels is forced to take the same disparity value. We could clearly see the smoothness effect of growing window size and consequently, the result becomes more accurate for the overall image, although we lost some fine details.

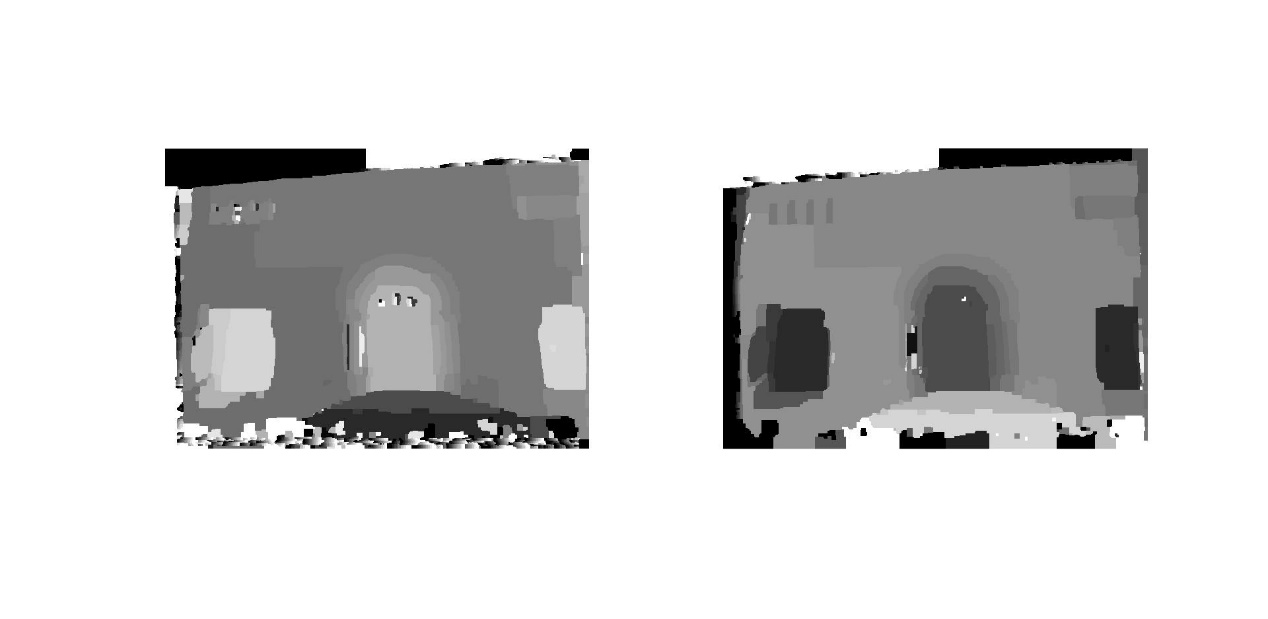
## Graph-cut

In this part, computing disparity is formulated as a graph labeling problem. The goal is to find a labeling f which map pixel to disparity. The objective function is defined to penalizing the neighboring pixels having different disparity values. I put my code for calculating SSD error of the first part in the file diffGC.m and also compute the disparity values with different size of average filter. And the maps are shown below:



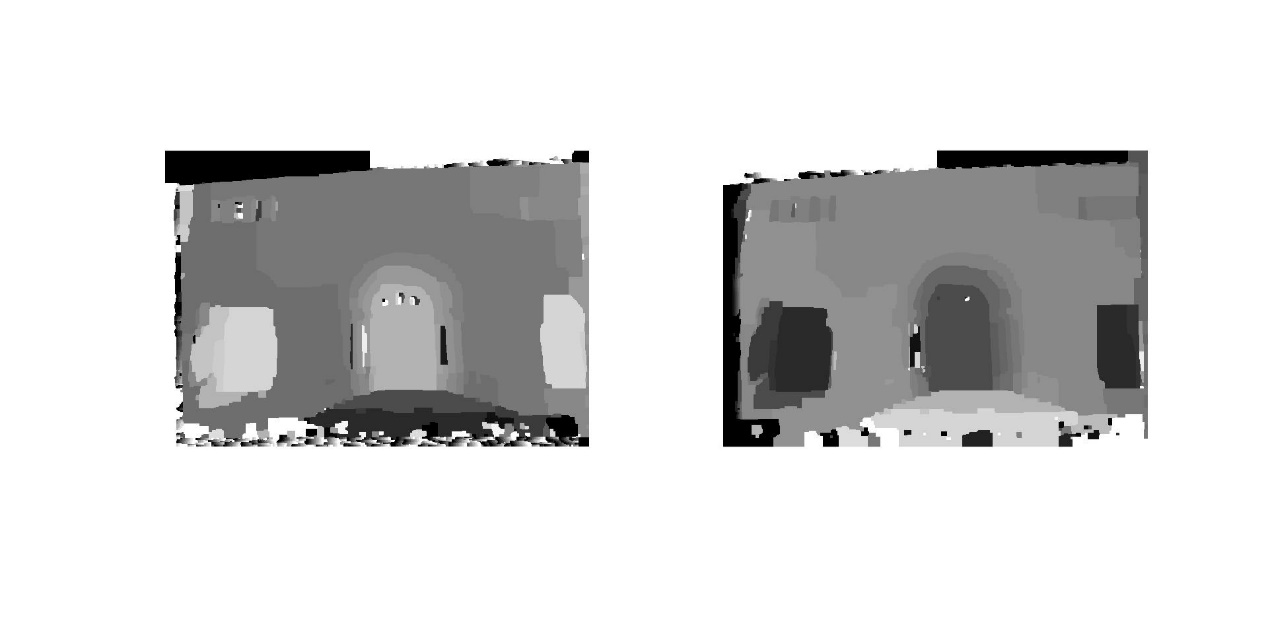


Graphcut result with 3\*3 window





Graphcut result with 5\*5 window





Graphcut result with 7\*7 window

Figure2. Disparity map using Graphcut method

For graphcut algorithm, we could find that when we change the size of the average filter, the results don’t change a lot. But compared with WTA method, there is a good improvement and the smoothness effect is obvious.

## Textured 3D model

In this part, I used the result of dense stereo algorithm and the camera parameters,

generate a textured 3D model.

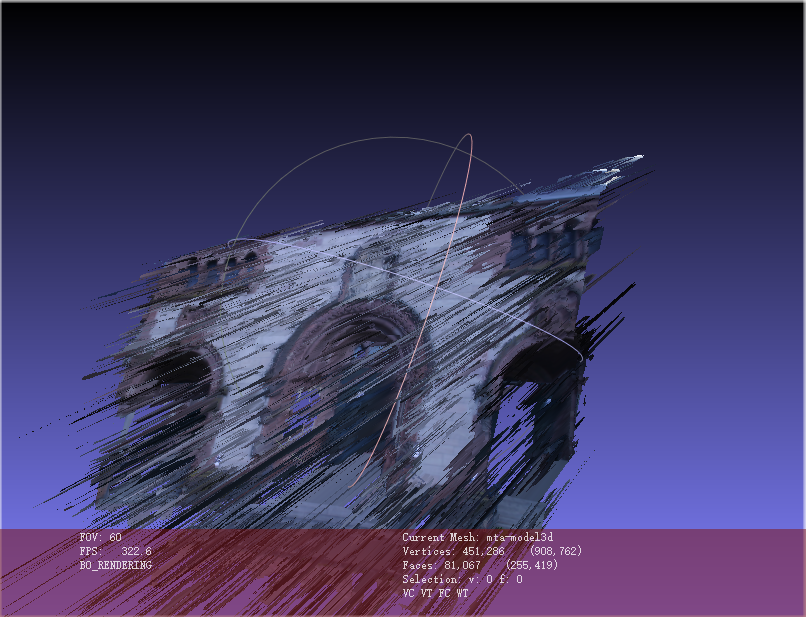


Figure3: 3D model using WTA algorithm



Figure8: Match two images (size 1) by vlfeat lib.

It is found that if we choose descriptors as patch of 9\*9 radius, some matches are obviously not correct. So vlfeat has better performance than Harris corner detector. Even though both harris corner detector and Sift have rotation-invariant property, sift method actually has the advantage of scale-invariant. So changing the scale of the images would not affect the detection of the corner as which happens with Harris corner detector.

## Conclusion

By doing this exercise, I realize feature extraction by implementing harris corner detection, descriptor match as well as the vlfeat library. By comparison, I could find that SIFT algorithm is a more powerful than Harris corner detection.