Delft University of Technology

3D Robot Vision

ME41030

Assignment Three: Stereo Matching

Authors:
Xin An (4735994)
Xingchen Liu (4739094)
Yuhao Xuan (4696514)

March 15, 2018

Contents

1 Basic of image filtering

1.1 Part a

What is this Gaussian filter often used for? Explain the underlying process of how such a kernel is applied.
Answer

Gaussian filter is often used for image processing, it can efficiently decrease the noise or reduce details which was distributed as gaussian function. The process is also called Gaussian blur or Gaussian smoothing. Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function.Assuming that A is the original image and B is the image after gaussian filtering, we can say that every point in B is calculated by the average of the point in A as well as the points surrounding them.

As mentoined in above, a Gaussian blur is conducted by convolving an image with a kernel of Gaussian values. In order to reduce the calculation, we take advantage of the Gaussian blur's separable property by dividing the process into two dimension paths,the horizontal and vertical direction. The resulting effect is the same as convolving with a two-dimensional kernel in a single pass.

1.2 Part b

Create a Gaussian filter and display it. Vary the hsize in 3 steps and do the same for sigma, such that you create clearly different filters. Display these images with subplot.
Answer

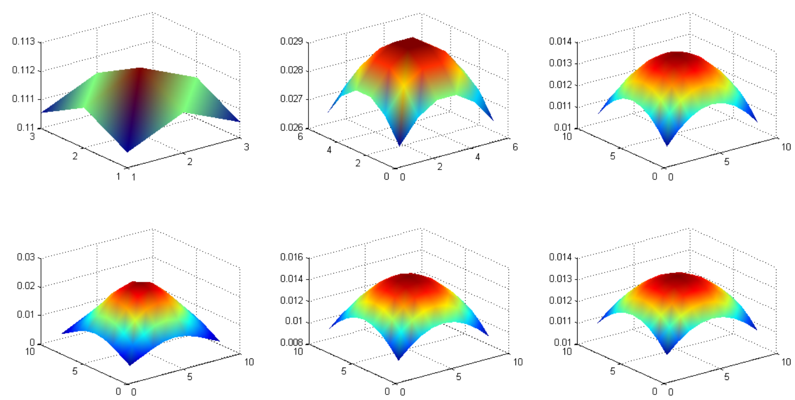


Figure 1: Gaussian filters with different values of hsize and sigma(from left to right,up to down);(a) hsize=3,sigma=8;(b) hsize=6,sigma=8;(c) hsize=9,sigma=8;(d) hsize=9,sigma=3;(e) hsize=9,sigma=6;(f) hsize=9,sigma=9

1.3 Part c

Load the image lena.jpg and apply the filters from 1b, display these images with a similar subplot. What is the effect of varying the hsize and sigma parameter on the image?

Answer

Figure 2: Lena processed by Gaussian filters with different values of hsize and sigma(from left to right,up to down);(a) hsize=3,sigma=8;(b) hsize=6,sigma=8;(c) hsize=9,sigma=8;(d) hsize=9,sigma=3;(e) hsize=9,sigma=6;(f) hsize=9,sigma=9

Matlab instruction fspecial('gaussian', hsize, sigma) returns a rotationally symmetric Gaussian lowpass filter of size hsize with standard deviation sigma. Hsize can be a vector specifying the number of rows and columns in h, or it can be a scalar, in which case h is a square matrix. The larger the hsize is, the larger the kernel is.Hence one image pixel is calculated by more pixels around it, which causing it to be more blur. From figure below, it can be concluded that the larger the sigma is, the more the blur level is.

1.4 Part d

Create an image pyramid that scales and blurs the Lena image. Use a factor of $\sqrt{2}$ for sigma and $1/\sqrt{2}$ for scale and create at least 4 levels. Display it with subplot. Explain how this is used in SIFT and why! (You could use a picture or plot to support your answer)

Answer

Gaussian pyramid has 4 blurred images, for example. DoG is difference of these images, so there are 3 images in DoG. Note that all these are in first octave! When the first pyramidis created, resize image and start to build new pyramids for second octave. Now for the matching of minima and maxima: Lets say you are looking maxima in first octave. You must use DoG pyramid and start from 2nd image. You take a pixel and calculate if it is maxima. In this calculation you should use 1st,2nd and 3rd images of DoG pyramid. If it is done go and find maxima in 3rd image by

Figure 3: Image pyramid of lena using a factor of $\sqrt{2}$ for sigma and $1/\sqrt{2}$ for scale (from up to down)

considering 2nd,3rd and 4th images. And lastly go find maxima in 4th image by considering 3rd,4th and 5th images.

Now finding mixama in first ocatave is completed, go to next octave and repeat these steps.

2 Dense stereo Matching

2.1 Part a

Implement the above described algorithm. Load the images left_rect_chessboard.jpg and right_rect_chessboard.jpg from the directory of exercise 2a. Define a reasonable search window 'W' and a reasonable range to search from the left to the right image. What values did you use and why? Show the disparity output.

Answer

First, the search range needs to be settled, choose the bottom corner in the left picture, the corner corresponds to the left corner shift right in the right picture with a range of $\frac{1}{3}$ of the total length. In that case, the search from 130-170 could be reasonable. As for the window size, small window size would cause error because the grids in the chessboard have different color and might be wrongly matched in that case, therefore a windowsize of 21 could be appropriate.

We test the search range from 130 to 170, and below are the result images.(130, 150, 170 are shown below, the rests are in the folder)

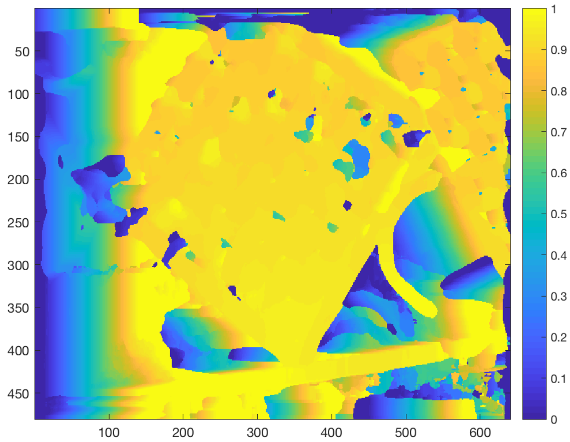


Figure 4: Window size: 21; Search range: 130;

As shown in the figure 4, 5, and 6, when search range reach 170, we can still see the difference between chessboard and the table, the chessboard became darker comparing to the other figures, the color of the table and the chessboard should be similar but distinguishable because the depth of the chessboard and the table should be very close, therefore, when search range is from 0-170, we can have a better matching.

In a word, we chose window size 21 and a search range from 0 to 170.

2.2 Part b

Load the images left_rect_hallway.jpg and right_rect_hallway.jpg from the directory of exercise 2b. Now there is also the stereo_calib.mat file, containing both the Essential (E), Fundamental (F) and the intrinsic camera matrices (Ml, Mr).

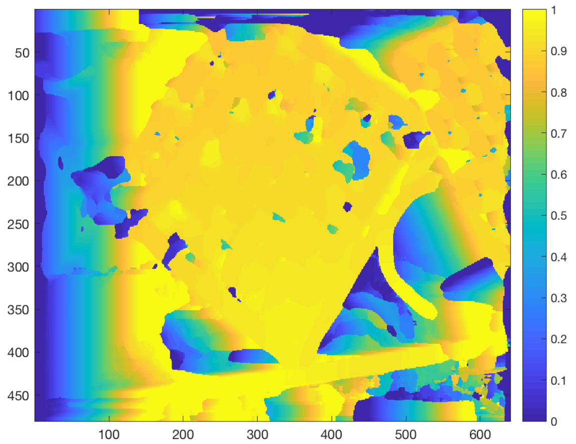


Figure 5: Window size:21; Search range: 150;

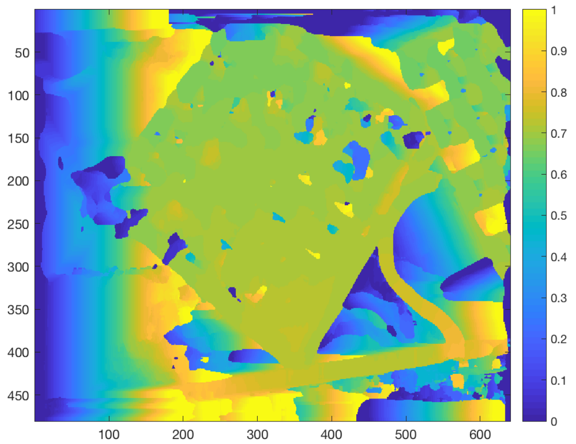


Figure 6: Window size:21; Search range: 170;

Calculate the search range for a distance 1-7m and show the disparity output. What values did you choose for the search window 'W'?

Answer According to the equation

$$Z = f\frac{b}{d}$$

if z range from 1-7m (1000-7000mm), with f(given in the intrinsic matrix). If we manage to get baseline b, we can obtain the disparity easily. According to the paper[1], the equation to recover baseline from essential matrix is shown below:

$$bb^T = \frac{1}{2}Trace(EE^T)I - EE^T,$$

with Matlab, we calculated the value of the bb^T, the length of b should be

$$|b| = \sqrt{b_1^2 + b_2^2 + b_3^2}$$

we can see that the sum of the diagonal elements of bb^t. therefore we have b = 121.5115, hence the d we calculated has a range from 9 to 64.

As for the window size, we first test 5 and 11, the result(in the folder) shows too many details, the people stand in the hallway could not be matched correctly, then we tried 21 and 31, the result are shown below: In the two figures(7 and 8),

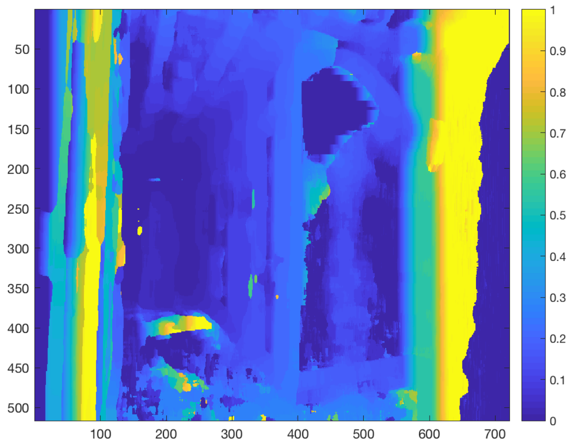


Figure 7: Window size:21; Search range: 9-64;

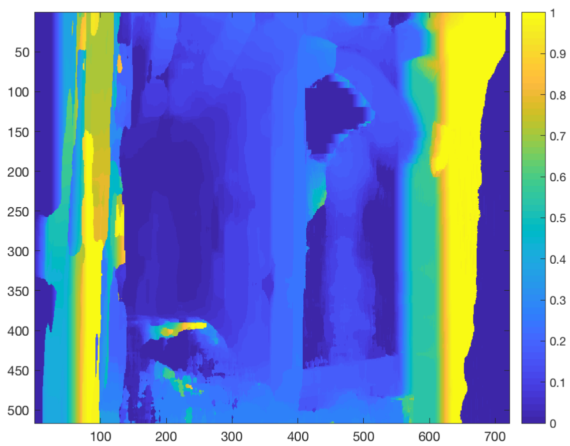


Figure 8: Window size:31; Search range: 9-64;

the person stand in the hallway is shown with a slightly lighter blue, however in the figure 5 we can see that with window size 31, some informations in the original picture are lost, such as the wall of the first room.

Therefore we chose 21 as our window size.

2.3 Part c

Open the file getCorrelation.m (On purpose the implementation of SAD is left out). Use the different correlation methods in your algorithm. Redo part a) and b),

what is the best method and parameters. Display your results.
Answer As for part a, when applying the getCorrelation.m, seven methods were taken into consideration and for each method, different window sizes should be applyed in order to find the best match.

First take a look at the SAD, we tried awindow size of 11, 21and a window size of 31, the results are stored in the exercise folder with a format $D\_METHOD\_MAP\_WIN-DMIN-DMAX\_$ for part a and $d\_METHOD\_MAP\_WIN-DMIN-DMAX\_$ for part b.

The result for part a shows that the SAD and the SSD have a better matching than the other methods, the result figure are shown below(Figure:9, 10):

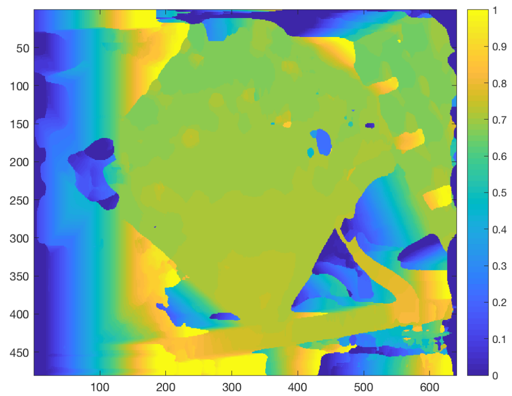


Figure 9: SAD with window size 31

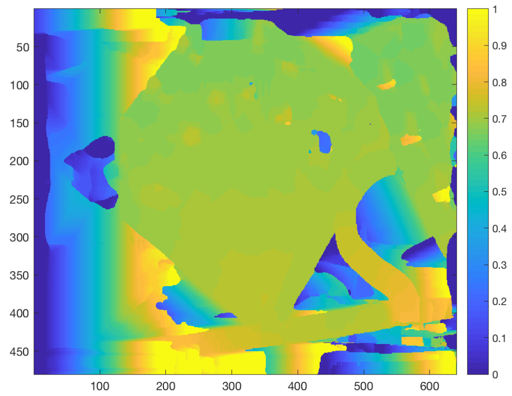


Figure 10: SSD with window size 31

The result for the part b shows that SAD and SSD have a better result the result are shown below(Figure11, 12):

Although the SAD and SSC are both good to use for part a and part b, the time consumption is different, we can see that the SAD is a little faster than sSSD, therefore we assume SAD is better.

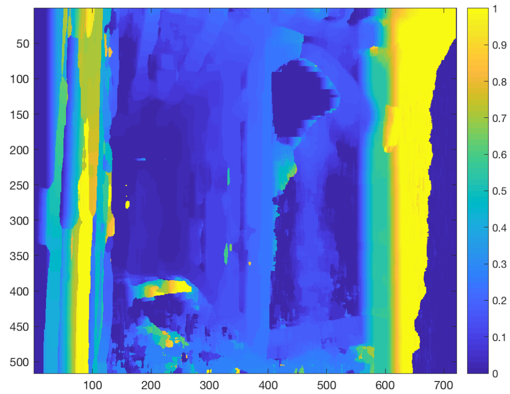


Figure 11: SAD with window size 21

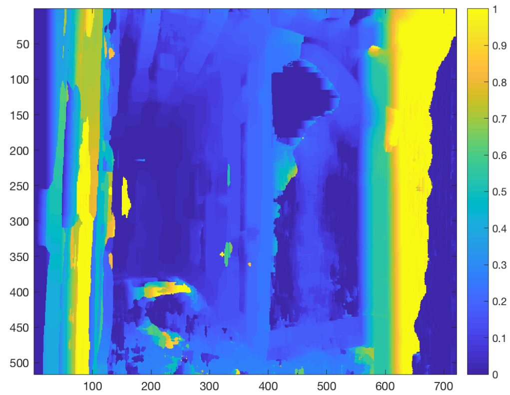


Figure 12: SSD with window size 21

3 Feature Matching

3.1 Part a

Load the images Left_SmallOffice.jpg and Right_SmallOffice.jpg and apply the SIFT method on them. What do the f and d values represent? Read the documentation of vl_sift and change the parameters. How many keypoints did you obtain with the default settings and how many with your own 'optimal' parameters? What parameters did you change and why?

Answer

Applying SIFT method, toolbox VL_FEAT is used here. For [f,d]= vl_sift(I) the input is image, and outputs using SIFT is f and d, where f contains X position, Y position, scale and orientation of all the keypoints; d contains information of descriptors. Each vector of matrix d represents a 3-D spatial histogram (8*4*4) of the image gradients in characterizing the appearance of a keypoint.

Using the default settings, we obtain 784 keypoints from Left_SmallOffice image, 787 keypoints from Right_SmallOffice image and 468 matches,as seen in figure

. After reading the documentation of vl_sift, we choose two parameters to optimize,

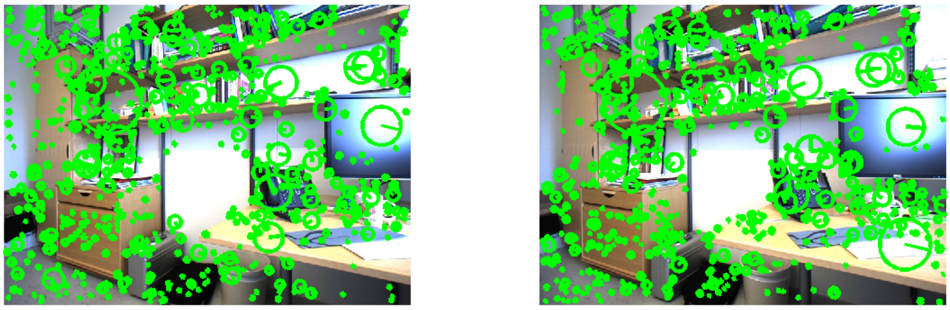


Figure 13: Keypoints obtained from SmallOffice with default settings

PeakThresh and Levels. The meaning, default and optimal values can be find in table . Using our optimal parameters, we obtain 516 keypoints from Left_SmallOffice image, 530 keypoints from Right_SmallOffice image and 340 matches,as seen in figure .

Figure 14: Keypoints obtained from SmallOffice with our optimal settings

For parameter PeakThresh, we set it change from 0 (default value) to 30, and we observe the number of match points and the ratio of number of match points to

Table 1: Default and optimal settings for vl_shift

Parameter	Meaning	Default	Optimal
PeakThresh	Set the peak selection threshold	0	4.5
Levels	Set the number of levels per octave of the DoG scale space	3	4

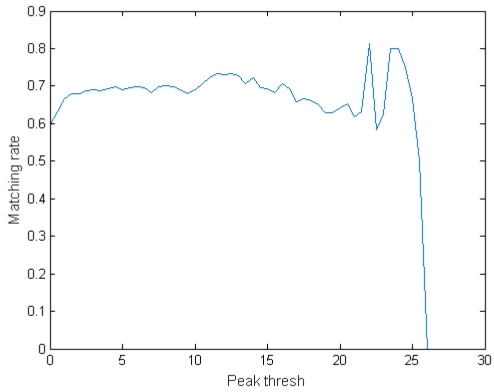


Figure 15: Match rate as function of PeakThresh

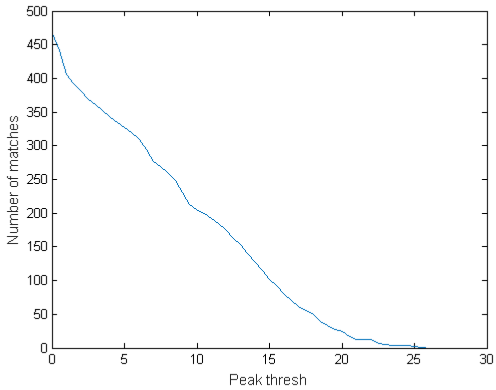


Figure 16: Number of match points as function of PeakThresh

number of keypoints.

As shown in figure and ,match rate oscillates around a certain value and then increases a little bit when PeakThresh is around 12. It should be noted that the sudden increase of match rate after PeakThresh increases to 22 is because both the number of keypoints and match points are very small. However, in figure it also shows that as the PeakThresh increases the number of match points decreases. Therefore we choose the optimal value of PeakThresh as 4.5 where we can have relatively large match rate without sacrificing too many match points.

For parameter Levels, it is the number of levels per octave of the DoG scale space. We set it change from 2 to 10 (default value is 3) , and we observe the number of match points and the ratio of number of match points to number of keypoints.

As shown in figure and ,match rate Keeps decreasing as the value of Levels increases. However, in figure it also shows that as the Levels increases the number of match points increase. It is possible that by increasing Levels, we introduce match

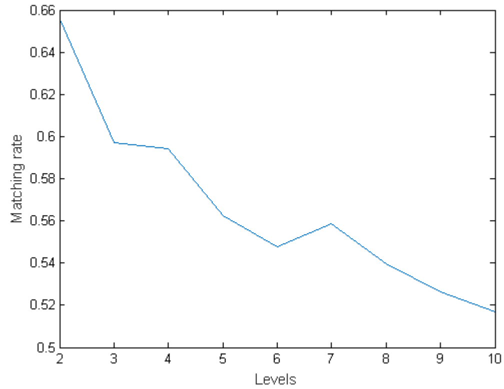


Figure 17: Number of match points as function of levels

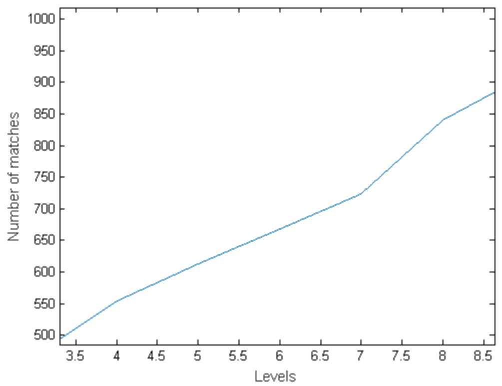


Figure 18: Number of match points as function of Levels

points and mismatch keypoints at the same time. Therefore we choose the optimal value of Levels as 4 where we can have almost the same match rate as with default settings while having more match keypoints.

Table 2: Match rate and other parameters with default settings and optimal settings

Parameter	PeakThresh	Levels	Keypoints	Match points	Match rate
Default	0	3	784	468	59.69%
Optimal	4.5	4	516	340	65.89%

3.2 Part b

Now do the matching (you can use vl_ubcmatch) Display your results.

Read the manual of the VL_FEAT toolbox and change the parameter. How many matches did you obtain with the default settings and how many with your own 'optimal' parameter? How and why did you change this parameter. Did you end up with only good matches?

Answer

Use vl_ubcmatch and plotmatches, we obtain the matches pictures with default settings and optimal settings in figure and , respectively. The parameter we change for matching is Thresh. A descriptor D1 is matched to a descriptor D2 only if the distance d(D1,D2) multiplied by Thresh is not greater than the distance of D1 to all other descriptors. The default value of Thresh is 1.5. The matches we obtain with default settings and optimal settings are shown in table In figure , number

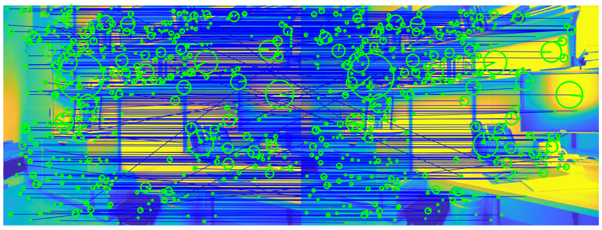


Figure 19: Matches pictures with default settings(Thresh=1.5)

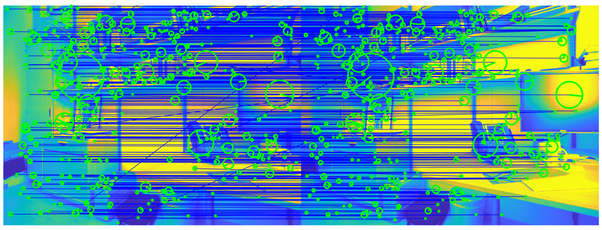


Figure 20: Matches pictures with optimal settings(Thresh=2)

Table 3: My caption

Parameter	Thresh	Match points
Default	1.5	468
Optimal	2	427

of match points as function of Thresh is shown. It is evident that the smaller the Thresh is, the more match points we obtain. However, it is not always a good thing to have a large number of matches because some of matches may not be the right matches. Figure is the matches picture with Thresh equals to 1. Compared to figure with Thresh equals to 1.5 and 2, the matches lines in figure with thresh equals to 1 increase but as the same time the unparallel lines also increase.

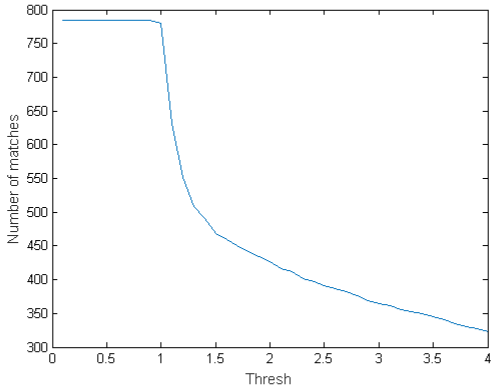


Figure 21: Number of match points as function of Thresh

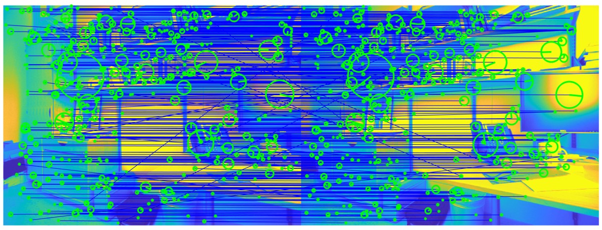


Figure 22: Matches pictures with optimal settings(Thresh=1)

4 Stereo Reconstruction

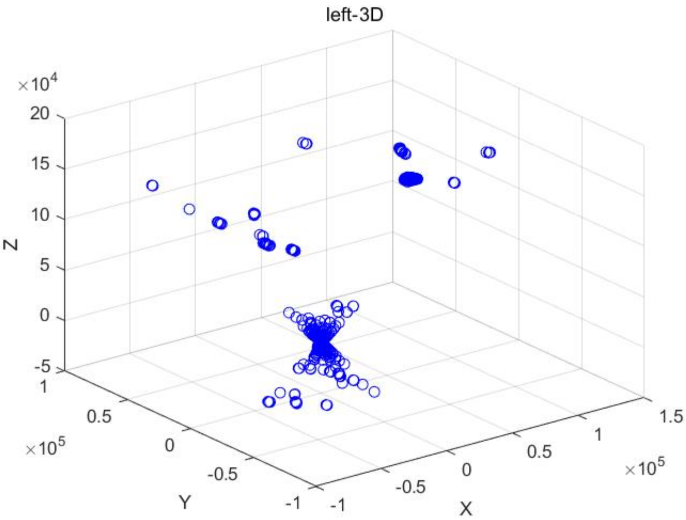
4.1 Part a

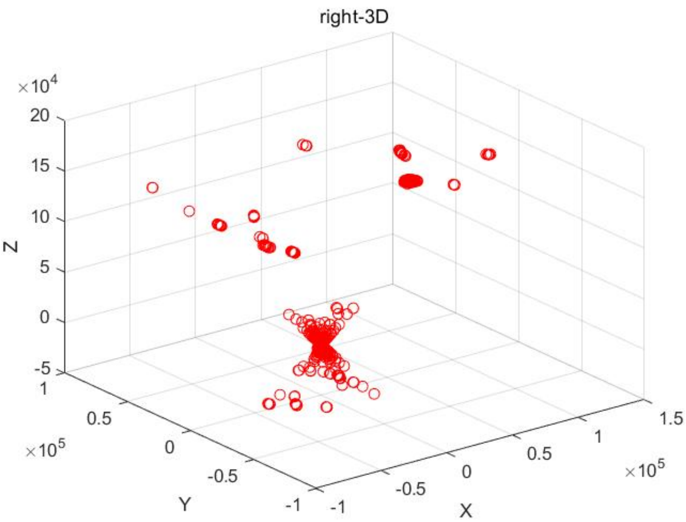
Perform a 3D reconstruction of corresponding points that you acquired using dense stereo matching techniques from exercise 2a. In order to perform reconstruction you need to find the relation between real world and disparity map that you obtained using stereo matching. For this part you will need the fundamental matrix and calibration data. These are provided in the file Calib_Results_stereo_rectified.mat
Answer

4.2 Part b

Load the data and your results from exercise 3. Reconstruct the corresponding points obtained using feature matching. Calibration data is provided in the file Calib_Results_stereo.mat. Discuss the difference between sparse matching and dense matching: In what situations would you use sparse matching and in what situations would you use dense matching?
Answer

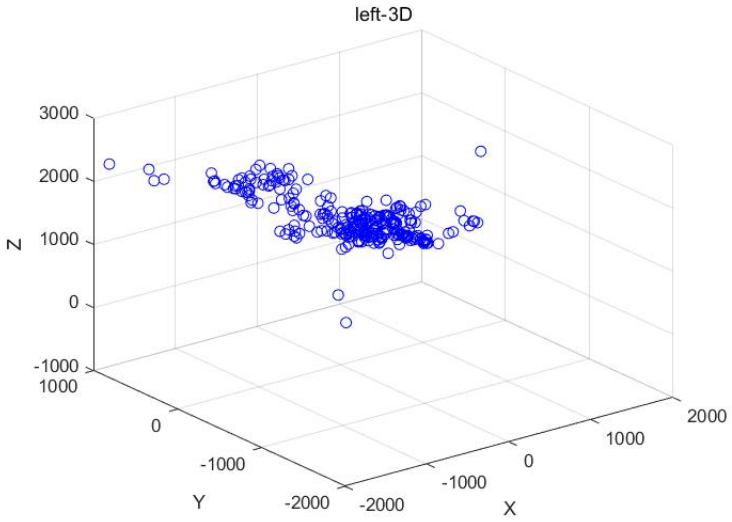
Sparse matching algorithms are used to establish a set of corresponding points

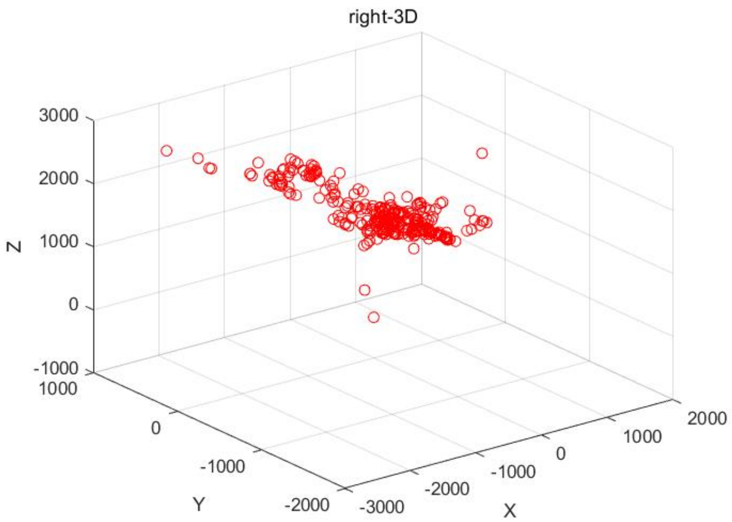




between two images. These sparse matches then can be used to compute the epipolar geometry. This algorithm is a feature based approach, only the necessary points are selected. The Computational complexity of this algorithm is lower than dense matching and it works with illumination and view point change. As a result, this algorithm can be used in large-range scenery 3D reconstruction.

Dense matching algorithms are used to find matches for all points in the images. For dense matching, we need to match as many pixels as possible. For each point in the left image, all the possible points in the right images can be found. As a result, the computation of this algorithm is complex. Also, this algorithm needs textured images and cannot work with illumination and viewpoint change. So this algorithm can be applied to 3D reconstruction in a small area scenery.





References

[1] Berthold K.P. Horn. Jan,1990, "Recovering Baseline and Orientation from 'Essential' Matrix".