Project assignment: Additional Tasks

SGN-34006 2017-01 3D and Virtual Reality

Aleksi Korkee

240385

1.Introduction

This submission contains the additional tasks of the project assignment. Tasks can be run by executing *demo_4_5_6_7.m.*

In this submission I have returned these additional tasks:

Step	Completed
4	x
5	x
6	x
7	x

In figure 1 is the left and right stereo pair images and ground thruth disparities.

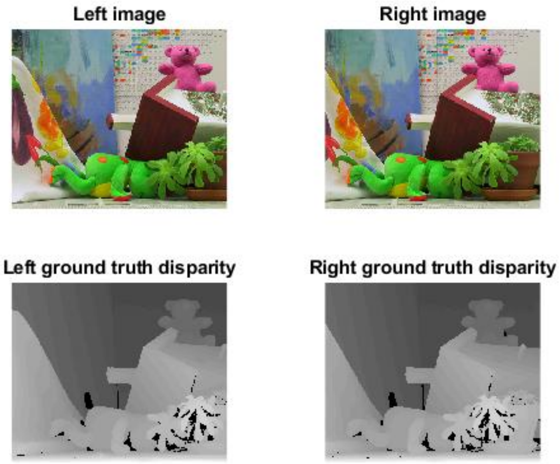

Figure 1: Stereo pair images and ground thruth disparities

In the 4th step a left-to-right correspondance check is performed. In the 5th step we calculate a confidence for cost volume. Using the result of step 4 we can fill blanks in disparity map by median filtering in step 6. In step 7 a cost aggregation with bilateral filtering is performed.

2.Steps

Correspondance Checking

In step 4 we detect outliers by doing correspondance checking for both left and right disparity estimates. This is implemented in the function *consistency_check.m.* For each pixel in left image we find the corresponding pixel in the right image and compare the absolute difference of their values. The corresponding pixel is found at pixels x coordinate minus the disparity value. If the difference is larger than 1 we define the pixel as invalid. The same is done for the right image. In this case the corresponding pixel is found from x plus disparity value. The function returns a black and white image, where white means that the pixel is invalid.

In figure 2 is plotted the cost aggregated disparity from guided filter and the results from task 4,5 and 6.

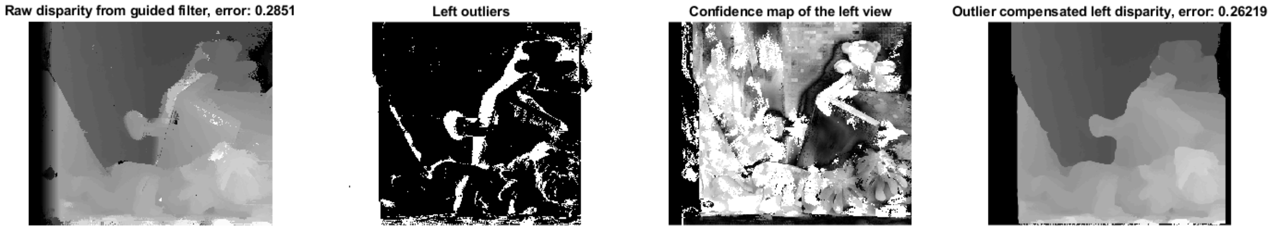
Raw disparity from guided filter, error: 0.2851
Left outliers
Confidence map of the left view
Outlier compensated left disparity, error: 0.26219

Figure 2: Estimated disparity, outliers, confidence map and outlier compensated disparity.

When looking at the disparity estimation and ground thruth disparity in figure 1 we can see that the estimate’s shapes are more round and the estimate has some noise left in it. These occluded areas can be seen as white areas in ”Left outliers” map in figure 2.

Confidence calculation

In step 5 a confidence map is calculated which tells how reliable each value of the aggregated cost volume is. I implemented the function in *comp_confidence.m*. Calculated confidence map is shown in the figure 2. For each coordinate in cost volume a peak ratio is calculated.

Outlier filling

The disparity estimate has some noise and doesn’t take occlusion in account. Using the ouliers map from task 4 occluded areas can be filled in disparity map. The occluded areas are removed from disparity map by turning them NaN-values. The disparity map is then median filtered to fill these areas. Using nanmedian-function we can take a median that ignores NaN-values. The filtering needs to be done several times to remove all occluded pixels. In figure 3 is visualised the filtering process.

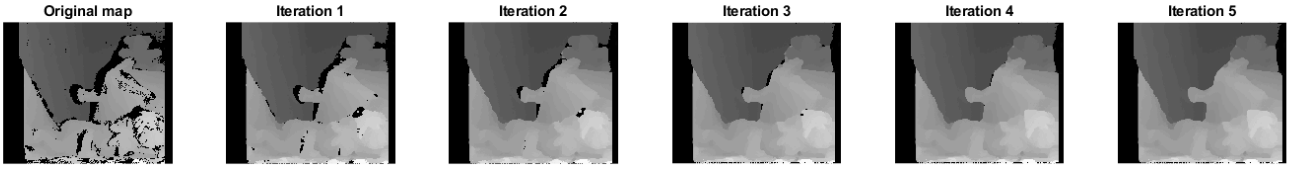

Figure 3: Outliers added to disparity estimation and iteratively filling the outliers.

The first iterations gets rid of small blank areas and larger areas get smaller. In the fifth iterations all blank areas in the middle of the imaige are filled. In figure 2 is the resulting disparity with an error of 0.26219. It also looks more like the ground thruth disparity than before calculating outlier compensation.

Bilateral Filtering

In task 7 a cost aggregation is calculated with bilateral filtering. In bilateral filtering both pixel distance and color value difference is taken in account when calculating weights in filtering. Weights are calculated from color image and the filtering is done to cost volume. Resulting image from bilateral filtering is shown in figure 4.

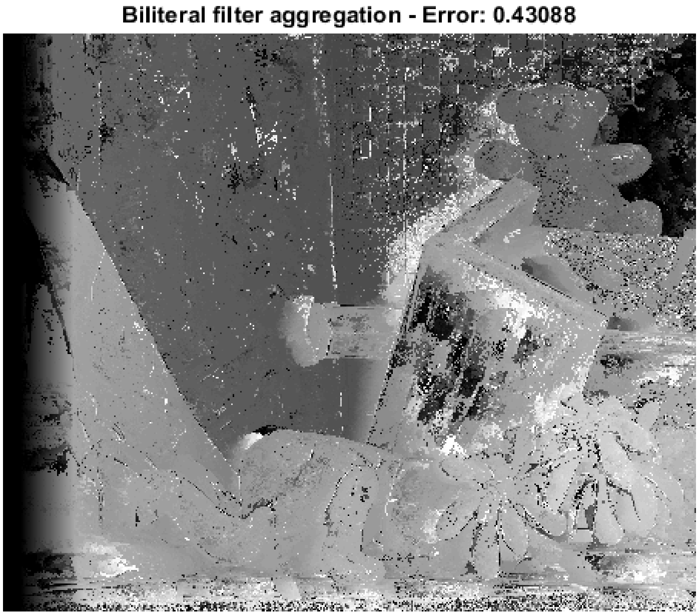

Figure 4: Disparity estimation with bilateral filtering

This is my second submission of the bilateral filtering. The first time I calculated the weights from cost volume instead of the color image. In figure 5 is shown my previous attempt of the filter.

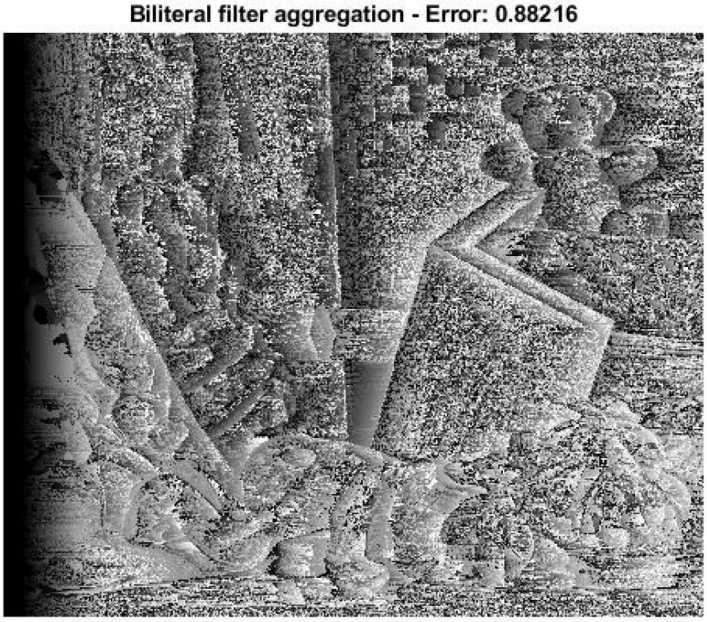

Figure 5: Disparity estimation with wrongly implemented bilateral fitering.

By calculating the weights from color image the error gets to 0.43 which is a lot better than in my first attempt of the filter in figure 5.