Project assignment: Dense Disparity Estimation

SGN-34006 2017-01 3D and Virtual Reality

Aleksi Korkee

240385

1.Introduction

The purpose of this assignment was to implement an algorithm for estimating disparity between two stereo images. Project was done in Matlab. Demo code was given by the course which demonstrated the functionality of the algorithm. The task was to implement functions which *demo_mandatory.m*-script tries to call.

In this submission of the project I have completed all three mandatory steps of the assignment.

Step	**Completed**
1,2,3	x
4	
5	
6	
7	

In the first step a cost volume calculation function was implemented from two stereo images. In the second step a cost aggregation was done with three different methods. In the third step a "winner-takes-all" disparity estimation function was implemented. Each implemented function is explained in the following sections of this report.

The two stereo pair images used in this project are shown in figure 1. Below those images are ground truth disparities to which the estimated disparities will be compared.

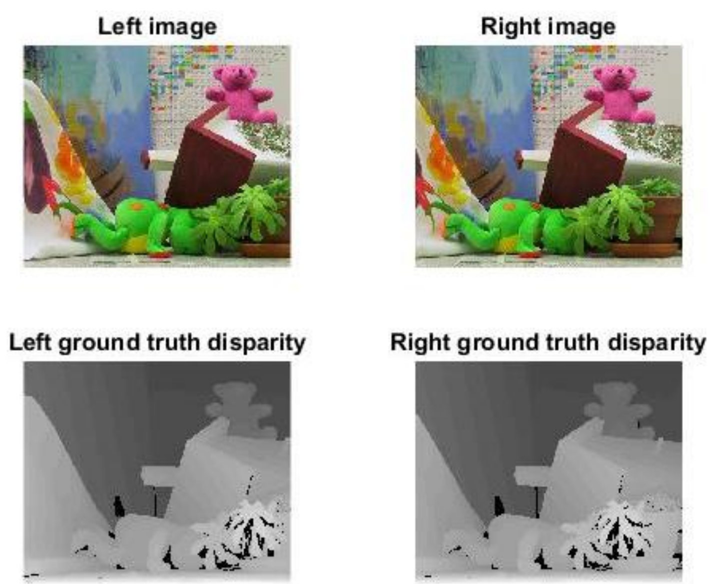

Figure 1: Stereo pair images and ground truth disparities

2. Steps

Cost Volume Calculation

The first step was to calculate cost volume for stereo pair image. This was done in *calculate_cost.m*-function. The function takes left and right images and largest disparity as input. The function returns two cost volumes. One is associated with left image and the other with right image. The function loops through all disparity levels and calculates cost for each pixel in that level. For a single pixel cost can be calculated by shifting other image sideways depending on which disparity level we are calculating. Equations 1 and 2 show a cost volume calculation for single pixel in left and right image.

$$CostL(y,x,d+1) = \sum_{c=1}^{3} |L_c(y,x) - R_c(y,x-d)|, \tag{1}$$

$$CostR(y,x,d+1) = \sum_{c=1}^{3} |R_c(y,x) - L_c(y,x+d)|, \tag{2}$$

where y and x are pixel locations, L and R are the images and index c is a color channel in RGB image. Absolute difference in pixel values in each color channel of the left image and shifted right image are summed to get the cost volume. Largest cost volume is capped to 150. In equations 1 and 2 $x - d$ and $x + d$ must not be outside of image boundaries. This is taken in account in the implemented function.

Cost aggregation

Cost aggregation was calculated for cost volume using three methods: box filtering, gaussian filtering and local color-weighted filtering. In box and gaussian filtering a matlab function *imfilter(A,h)* is used where A is one disparity level of cost volume and h is filter used.

In box filtering we go through all pixels and calculate an average of each pixel and its surrounding pixels under a window of wanted size. Here filter h is NxN-matrix where all elements are of value $1/(N*N)$ and N is the size of the window.

In gaussian filtering we get the filter h using matlab function fspecial('gaussian',w_size,sigma). We give as parameters the wanted filter type, filter size and standard deviation.

Local color-weighted filtering uses matlab function imguidedfilter. Each disparity level in cost volume is filtered using left color image as guide. Function takes also takes in filter size and smooth value which defines how much output will be smoothed.

Winner-Takes-All estimation

Winner-takes-all-function takes cost volume and finds for each image coordinate the disparity level that has the lowest cost. Output will be an estimated disparity map which will be compared to ground truth image in figure 1.

3. Results

The error calculation between disparity estimations and ground truth disparity was implemented in error_calculation.m-function. The function calculates the percentage of bad pixels in estimation. Pixel is called bad if absolute difference of estimated pixel and ground truth pixel is larger than 1. The number of bad pixels are then divided by the number of all pixels which gives us the error. There are 4 estimations of disparities with calculated error shown in figure 2.

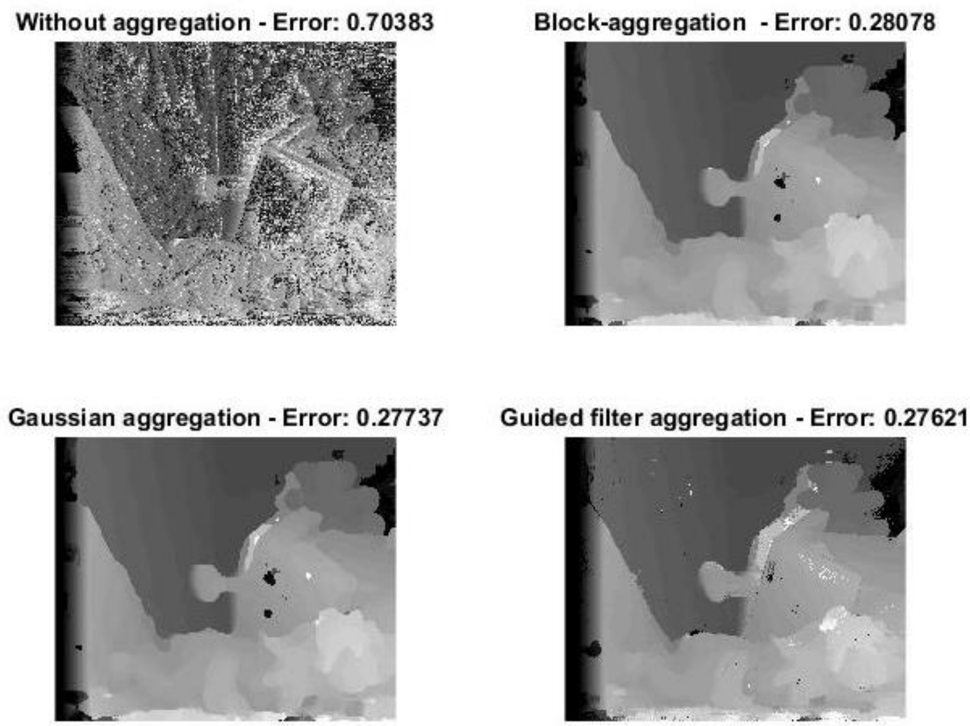

Figure 2: Disparity estimations with different methods

Without any aggregation the error between ground truth disparity is large. Visualized disparity is very noisy and it is hard to see shapes in it. When using any of the implemented aggregation methods the error gets a lot smaller. Disparities with aggregation also look visually a lot more similar with the ground truth disparity. The smallest error was attained with guided filter aggregation.

Three aggregation methods were compared by changing the window radius of filters. Error is shown as a function of window radius in figure 3.

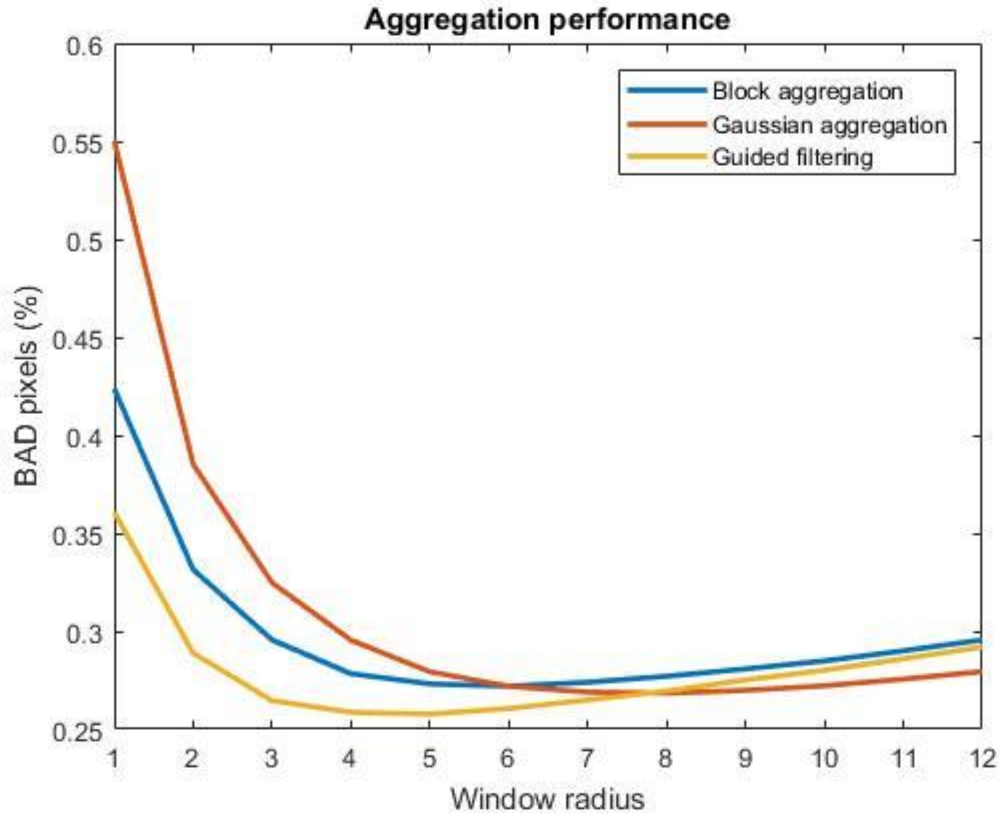

Figure 3: Aggregation performance as function of window radius.

Using larger window radius gives good results with each method. After a certain radius is reached the error start to rise again. Optimal radius for block aggregation is 6. For gaussian aggregation it is 8. The smallest error was attained with guided filtering with a window radius of 5. These are only optimal for the stereo pair images used in this assignment.

4. Conclusion

The three mandatory steps were completed in this submission. Guided filtering aggregation gave the best results. Block aggregation and gaussian aggregation gave also good results. Without aggregation the error is large so aggregation should be used when estimating disparity.

The functions implemented in this submission are *calculate_cost.m, calculate_error.m, aggregate_cost_block.m, aggregate_cost_gaussian.m, aggregate_cost_guided.m* and *winner_takes_all.m.*