Disparity for Stereo Vision - Block Matching and Dynamic Programming

Project Team:

Manjeet Singh | 50169105
Vipin Kumar | 50208397

Date of Submission : 16 Dec 2016

Introduction

Objective:

Depth information for a pair images of the same object can be calculated by computing the distance between the pixel position of the common features between 2 images by creating a depth disparity map of the pixels with respect to each other in the two images. Pixels with larger disparities are usually closer to the camera than those with lower disparities.

Our goal is to create and study the disparity estimation map between 2 images of the same object using block matching and dynamic programming based on the Maximum Likelihood Stereo Algorithm.

General Concepts to be used in this implementation:

- Disparity estimation in Stereo Vision (Concept of disparity = $X'_{l} - X'_{r}$).
- Finding MSE with respect to the ground truth image (concept of MSE)
- Dynamic programming concept (general approach for optimizing algorithms by memoization)
- Concept of image threshold to find the occlusion in the image as a function of threshold.

Analysis of Approaches

Block Matching based Disparity:

Block Matching based disparity focuses on finding the pixels which correspond to the minimum SSD for the 2 images on the same epipolar line. For computing the depth information for the two image views, we consider a small region of pixels (a block of 3*3 or 9*9) in one image and search for the closest matching region in the other image. While performing this search, we preselect some maximum distance **d** and search the block up-to this distance d in the other image. The similarity calculation is based on a simple computation we perform called SSD. For each block under consideration, we find a block with the least SSD value in the other image and select that value for the disparity map. We try to minimize the SSD since lower the SSD value, the higher is the similarity in the two blocks.

Dynamic Programming based Disparity :

Dynamic programming based disparity is based on the finding a local cost derived that consists of

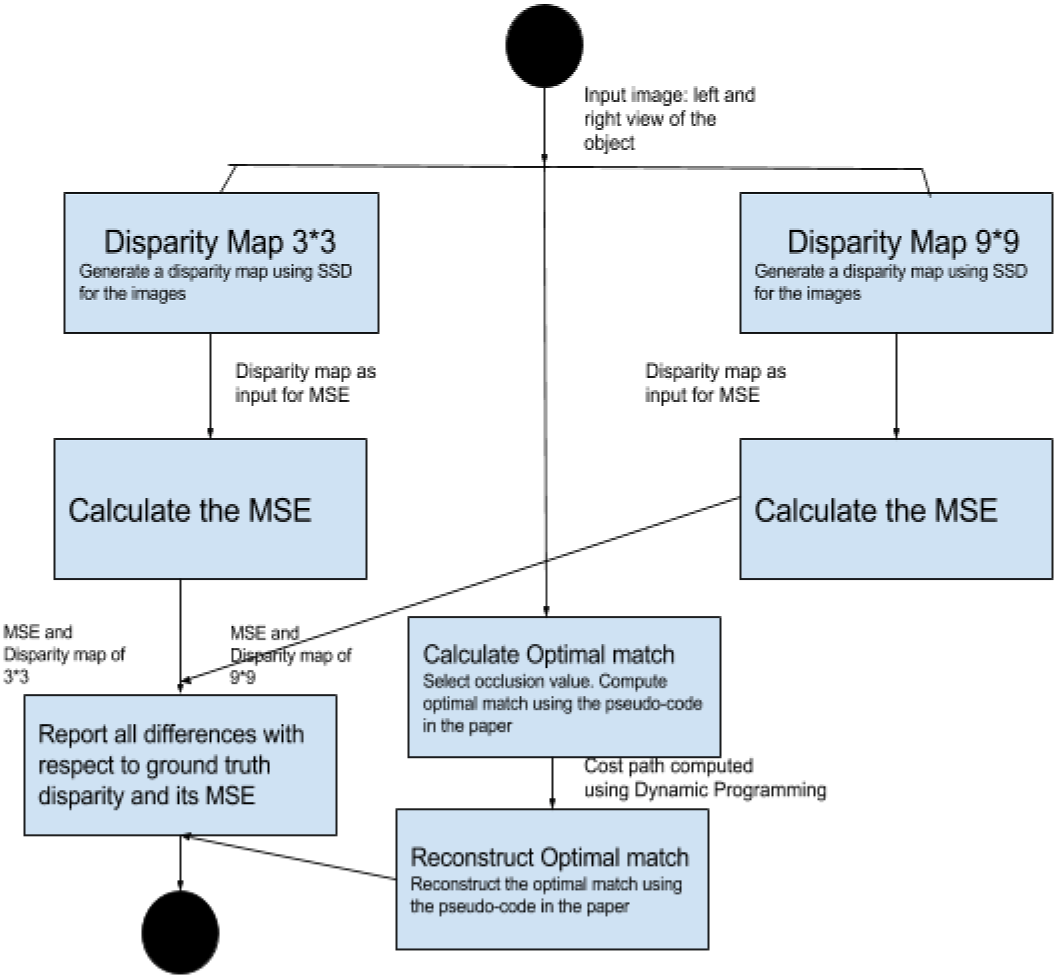
1) A normalized squared error term representing the cost of matching two feature

2) A fixed penalty for an unmatched measurement that is a function of the probability of occlusion

The paper presented an approach which is based on the above mentioned observations. We run our code on row basis, picking one row from left and right images to compute the optimal matching in two images. The paper also suggested that the dynamic approach is faster as compared to prior experiments conducted since no feature is extracted or adaptive windowing is needed prior to the algorithm

We then compute the MSE(mean squared error) for the generated disparity images with respect to the ground truth disparity to account for error in the disparity compared to the ground truth disparity provided for the images under consideration.

Project Flowchart


Input image: left and right view of the object
Disparity Map 3*3
Generate a disparity map using SSD for the images
Disparity Map 9*9
Generate a disparity map using SSD for the images
Disparity map as input for MSE
Disparity map as input for MSE
Calculate the MSE
Calculate the MSE
MSE and Disparity map of 3*3
MSE and Disparity map of 9*9
Calculate Optimal match
Select occlusion value. Compute optimal match using the pseudo-code in the paper
Cost path computed using Dynamic Programming
Report all differences with respect to ground truth disparity and its MSE
Reconstruct Optimal match
Reconstruct the optimal match using the pseudo-code in the paper

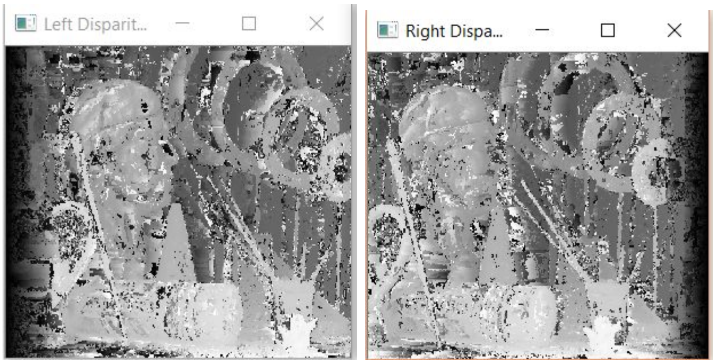
Results and Analysis

Block Matching based Disparity :

We ran 2 different experiments for generating disparity map. In the first experiment we varied the block sizes to study the effects of the window size on the output disparity map. In the second experiment, we varied the maximum distance **d** parameter to study its effects on the MSE and output disparity map

Results for block matching based disparity for window size 3x3:

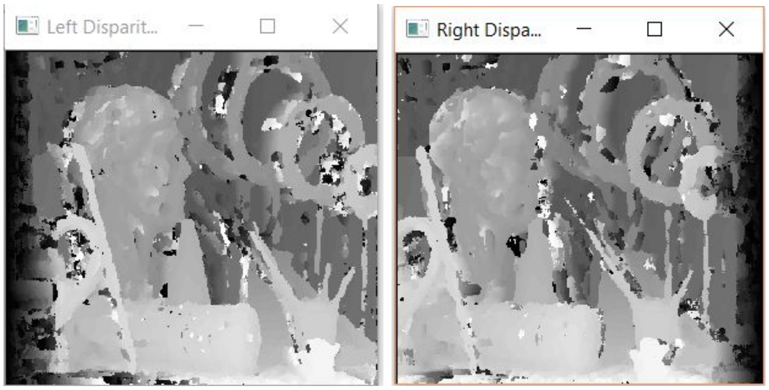
When the maximum distance parameter d is set to 75 (**Best result**)



MSE for left disparity result: 378.144644212
MSE for right disparity result: 258.107886288

Results for block matching based disparity for window size 9x9:

When maximum distance parameter is set to 75 (**Best Result**)

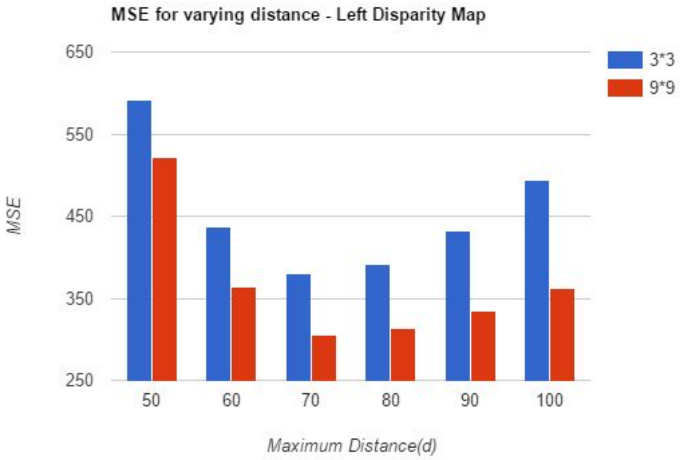


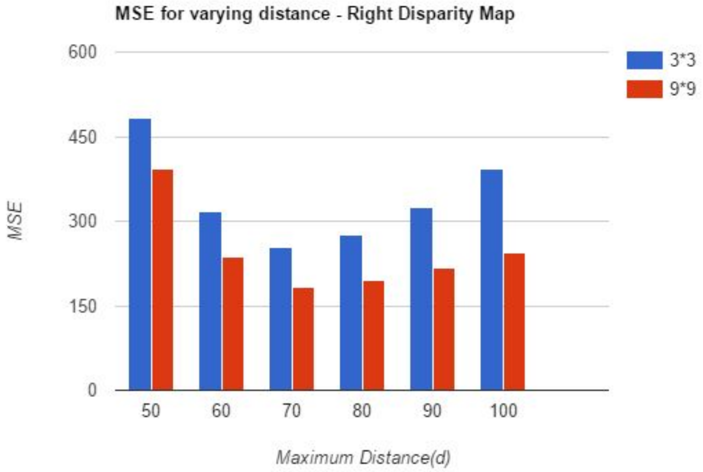
MSE for left disparity result: 305.984285798
MSE for right disparity result: 187.081040219

Discussion of the results for 3x3 and 9x9 block matching for disparity:

From the experiments conducted for block sizes of 3*3 and 9*9, we can see that increasing block size generates a better disparity map i.e. the disparity map is less noisy as we increase the window size for block matching. However this comes at a much higher computation cost.

Effect of varying the maximum distance d on MSE of the disparity result:



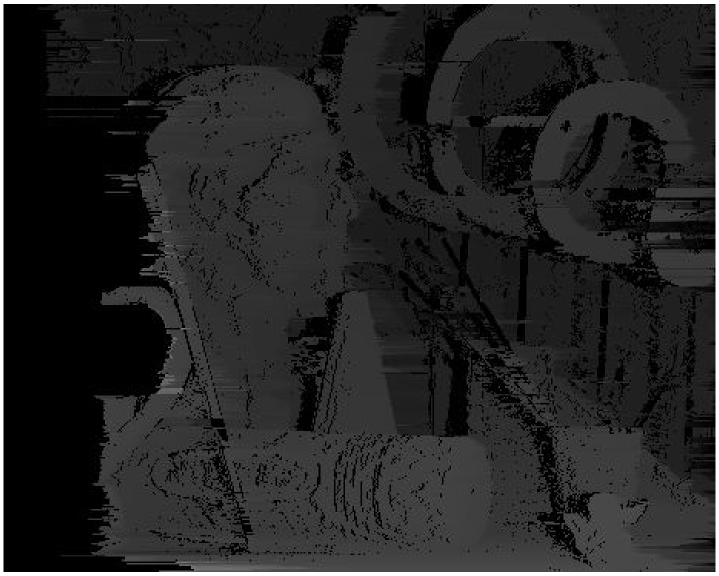


We can clearly see that a larger window gives a better disparity map compared to a smaller windows thus also resulting to a lower MSE value compared to a smaller window. In this case 9*9 provides better results compared to 3*3 block size.

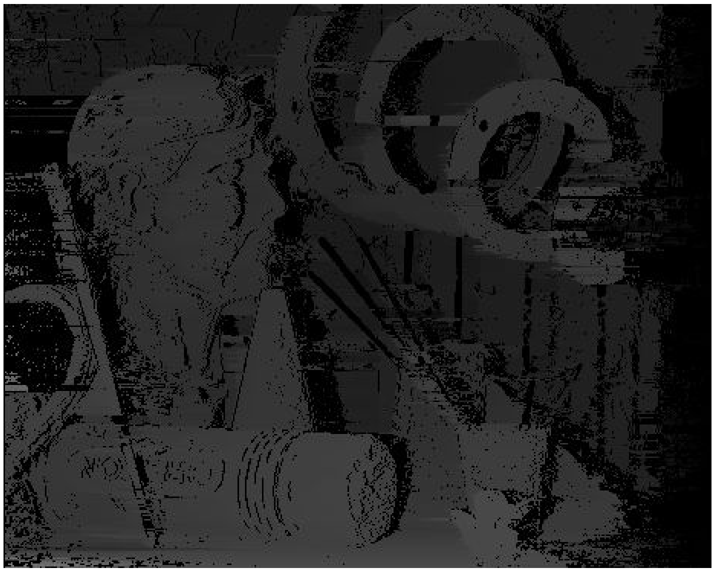
In the graph, we can see that as the maximum distance d approaches in the range 70-80 we get the best result and thereafter the MSE starts increasing again and the disparity image begins to blur as we increase the parameter d while performing block matching.

Dynamic Programming based Disparity :

Based on the dynamic programming we have observed that as we increase the penalty, occlusion, the MSE value increases and the resulting images become more and more blurred as compared to the ground truth . This is because as occlusion penalty increases giving a wider range to penality to occur.



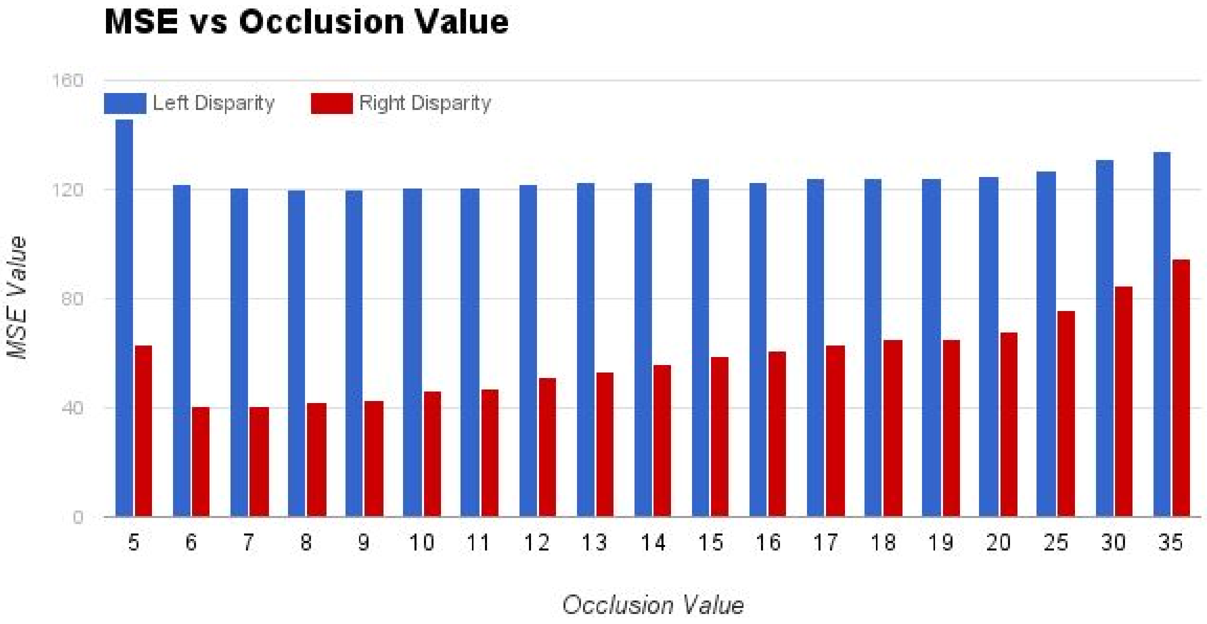
Left Disparity at Occlusion value : 7



Right Disparity at Occlusion value 7

Discussion of the results from Dynamic Programming for disparity:

As we can see that the image starts to blur and a lot of streaks comes into the output image. The effects gets more and more when higher value of occlusion is taken. When the occlusion value is low we get a very high MSE value. Following is a bar chart showing the effect of occlusion value with MSE value.



Lessons learnt from the outcome:

During the algorithm development, we understood the concept of disparity and how varying block sizes affects the results of the disparity map computations.

Software and Program Development

The following table describes the timeline, contribution and software technologies used.

Topic	Timeline (Week wise)	Team Member	Programming Language and IDE
Disparity Map 3*3 and 9*9 MSE for block 3*3 and 9*9	Week 1 (22 - 28 Nov)	Manjeet Singh	Language : Python IDE : Canopy IDE/PyChram IDE Libraries : Numpy, OpenCV, Matplotlib
Optimal match using Dynamic Programming and MSE	Week 2 (29 Nov - 5 Dec)	Vipin Kumar	
Final Report, Experiments and Optimization	Week 3 (11 - 16 Dec)	Manjeet Singh and Vipin Kumar	
Overall	**22Nov - 16 Dec**	**50% each**	

The programming phase provided a great opportunity and provided a very helpful insight as follows :

a) The collaboration and code optimization works best in given time frame for software development, just like in any organization
b) The distribution of the work was near perfect and project timeline was matched. We both came up with a basic solution approach for the block matching algorithm. Finally we collaborated the work to make use of the image slicing (idea for this approach was suggested by TA) to optimize the initial crude implementation of the block matching which was written using a naive pixel-by-pixel matching approach. This helped us reduce the runtime to just a few minutes from the previous code which took approximately 2 hours for 9*9 block matching. Both of us spent time reading the code presented in the paper and the details regarding the dynamic programming approach presented in the paper.
c) Provided a great insight in learning new IDE (Canopy), language (Python) and libraries(Numpy, OpenCV, Matplotlib) during the development phase.
d) The concepts taught in the classroom and recitation were of great help and helped for project code development and analysing results.

Summary

The above project is based on calculation of the disparity based on two approaches: block matching and dynamic programing. The two approaches varied in implementation and concepts but have some similarity when observed properly. One such observation is the penalty factor ,which in case of block matching was widow size and in dynamic programming occlusion value both of these values considered a range in which an element can be found in the two images.

How the two approaches are different?

Block matching is a more crude implementation which performs matching of pixels in a window and find the disparity based on the SSD values calculated. On the other hand, the dynamic programming approach is a better optimized version which makes use of memoization techniques while calculating the cost which is the normalized squared error term.

Lessons Learnt:

The course introduced us to Computer vision and Image processing techniques. We got a lot of opportunities through the homeworks and the projects to understand concepts such as Stereo Vision Imaging, use of different filters in image processing, need for segmentation and various approaches which can be used for it and through a homework got the chance to implement merging segmented regions into one.

The project gave us an opportunity to implement the basic block matching algorithm for stereo disparity maps which is the basis for stereo vision.

References

1) Cox, Ingemar J., et al. "A maximum likelihood stereo algorithm." Computer vision and image understanding 63.3 (1996): 542-567

2) http://mccormickml.com/2014/01/10/stereo-vision-tutorial-part-i/ (Simple block matching and block comparison, template size) - This reference helped us understand the concept of selecting the maximum distance value for block matching and the effects of various block sizes on the disparity map results.

3) Lecture6StereoVision.pdf shared on Piazza by TA - Gave us a basic approach and insight for block matching algorithm.