Disparity for Stereo Vision
Block Matching and Dynamic Programming

Team members:

Mugdha Milind Ansarwadekar	50207606
Sindhu Madhuri Morapakala	50207349

Final Report Due: 6:00 pm Friday December 16, 2016

Literature Review:

Stereo vision is of enormous importance to obtain three dimensional information from two dimensional images with slight difference in the perception of the object. The perception of depth is based on the disparity in the left and right images. Disparity of an image is defined as the difference in the location of a pixel under the projection of two perceptions of the left and right eye.

One of the standard methods used to compute the disparity map is by using a block matching technique.This is done by taking a small block of pixels in the left image, computing its cost and finding a matching block in the right image and vice versa. Although, the matching pixel could be translated horizontally or vertically in the right image when compared to the left image, the epipolar constraint for correspondence limits the displacement to within the same epipolar line. The epipole is the point of intersection of the line joining the optical centres, that is the baseline, with the image plane. In other words, the test images are rectified ensuring a horizontal search for pixels within the same row. Some of the common similarity metrics chosen in literature are Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC). In this project, the similarity measure used is SSD which is an estimate of of the measured squares of differences or deviations. Due to the high presence of noise while considering pixel comparisons, the block size plays an important role in determining the disparity map. Two block sizes considered in this implementation of the project is 3×3 and 9×9.

Second method implemented to compute disparity map is Dynamic Programming. As it is a global correspondence method, it exploits non-local constraints in order to reduce sensitivity to local regions in the image that fail to match, for example, due to occlusion. This technique uses the ordering and smoothness constraints to optimize correspondences in each scan-line.

Introduction:

In this project we computed disparity map by using Block Matching (Sum of Squared Differences - SSD) and Dynamic Programming (DP) techniques. Disparity maps provide necessary information for estimating the depth information of the image which is used in reconstruction of the 3D scene from 2D images.

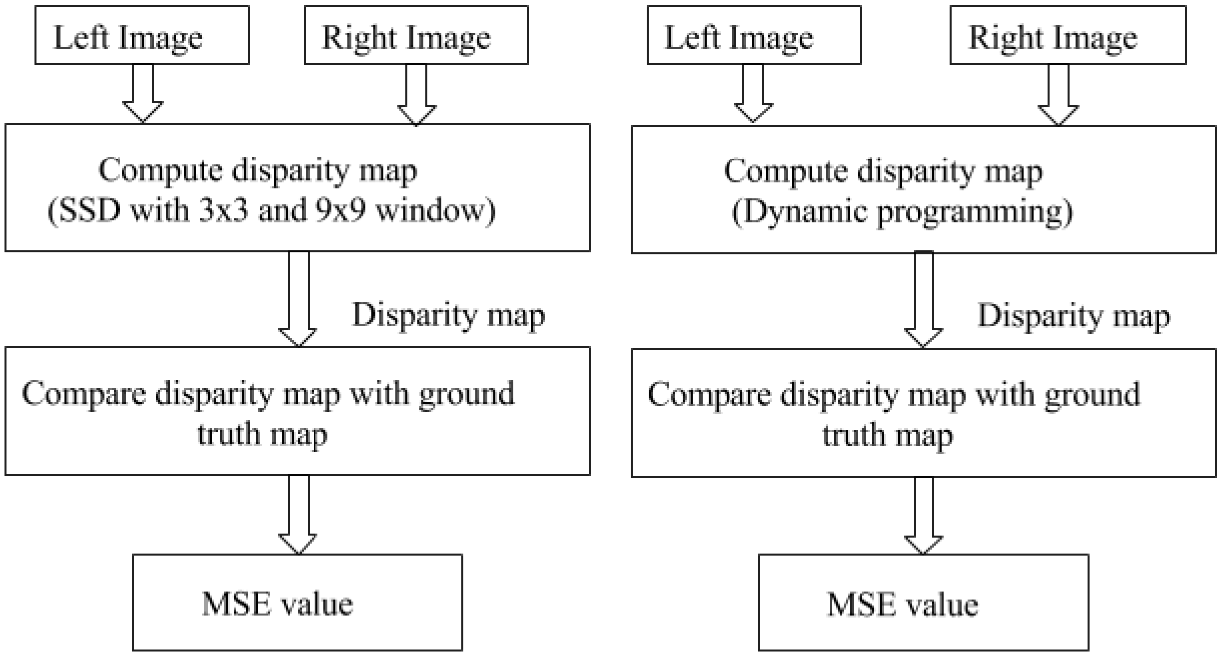
Window based neighborhood matching is one of the computer vision techniques used such that, given a block of pixels in the left image, the best match in the right image has to be determined as done using Sum of Squared Differences.

Dynamic programming is a mathematical method that reduces the computational complexity of optimization problems by decomposing them into smaller and simpler subproblems. We used this approach for stereo matching to find best match in right image for every pixel in the left image and vice-versa.

Project outline:

We used two methods for the implementation : Block matching using SSD method and Dynamic Programming. Block matching technique uses SSD as a measure to determine the best match to map the pixel in the right image to the pixel in the left and store the difference in the disparity map. Based on optimizing the maximum likelihood function, dynamic programming offers a better alternative to estimate the disparity maps. The key agenda of this implementation is to implement two images a left and a right perception image. Compute the disparity maps based on two different approaches. Comparing the disparity maps with the ground truth disparity maps provided, the MSE values are obtained to estimate the degree of error. The flow chart for this is presented below:

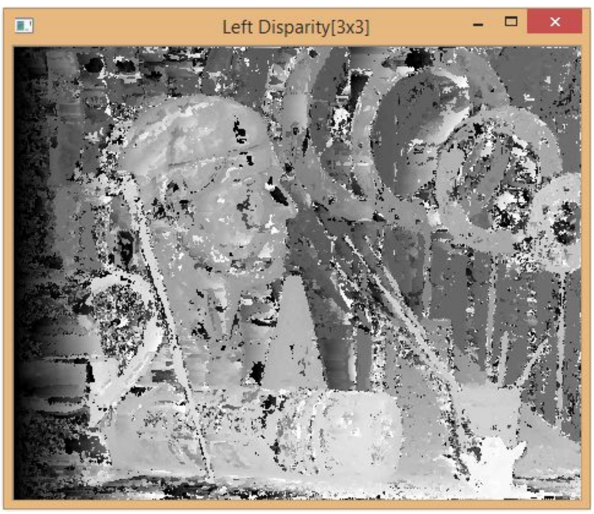
Flow chart:

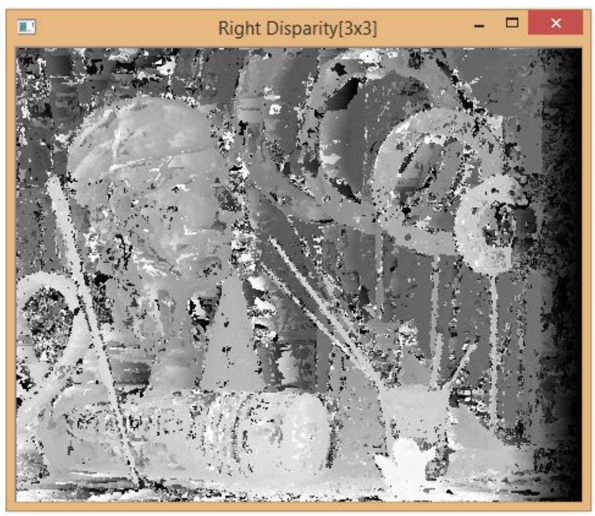


<u>Project Outcome :</u>

Block Matching approach:

1. **Block size = 3 x 3**
   MSE of Left disparity : 391
   MSE of Right disparity: 266

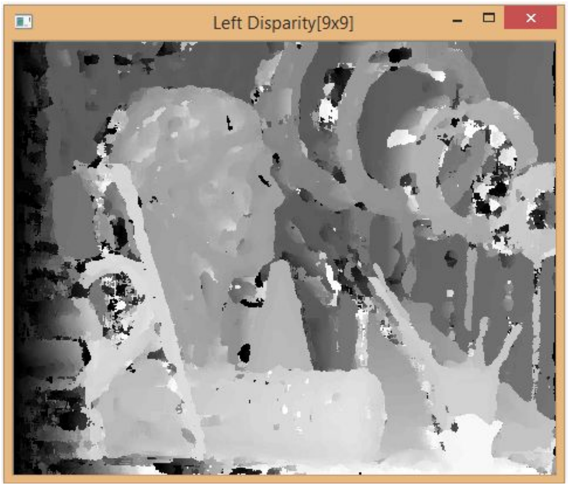


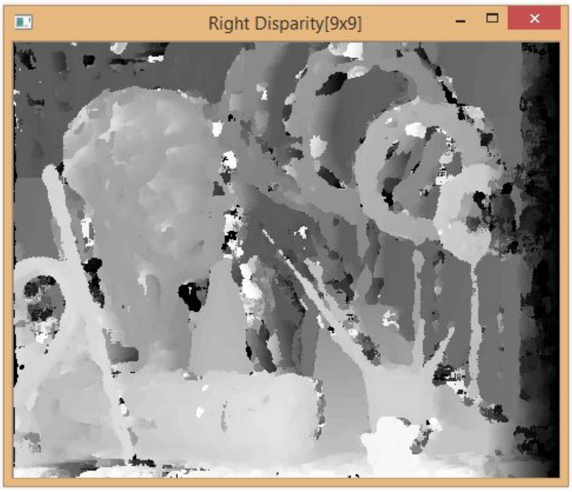


2. Block size = 9 x 9

MSE of Left disparity : 305

MSE of Right disparity: 184





Discussion of outcomes:

The MSE value can be attributed to the assumptions made in the project. Firstly, the fixed window size assumes constant depth within the window. This implicit assumption violates at depth discontinuities and slanted surfaces. This leads to wrong values in the matching costs. Also, uniform and repetitive areas leads to many minima of the SSD value easily resulting in wrong matches.

The block size has an impact on the MSE. When the size is 3 x 3 the MSE values for the left and right disparity were found to be high with a value of 391 and 266 respectively. For a 9 x 9 window, the MSE was found to be comparatively lower 305 and 184 for left and right disparity maps.

Dynamic Programming approach:

1. **Occlusion cost = 20**

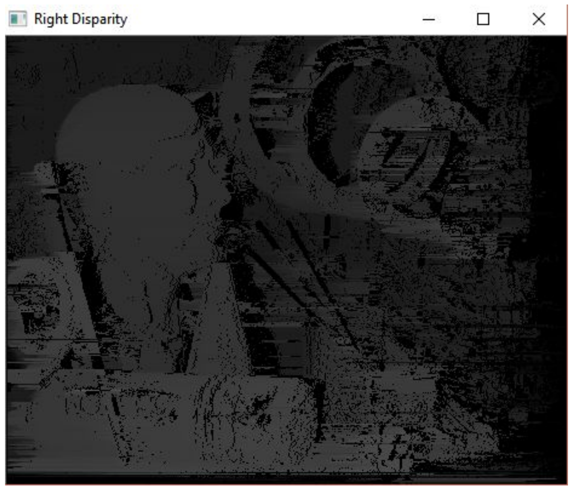
MSE values for the occlusion cost for left image: 78

MSE values for the occlusion cost for right image: 77

2. Occlusion Cost = 40

MSE values for the occlusion cost for left image: 128

MSE values for the occlusion cost for right image: 129



Discussion of outcomes:

The output images represent the disparity map which approximates the shape of the image and gives depth information. Respective MSE values indicate the similarity of the result with ground truth images.

Two important factors affecting the outcome of this implementation are cost function and occlusion function. The local cost function used considers difference in the intensity values only. Additionally, occlusion cost is fixed to a constant value. Because of these two reasons, the output presented here differs from the standard output.

The appearance of horizontal streaks in the output images are possibly because of propagation of local errors along the scanline which corrupted potential good matches.

Lessons learnt from algorithm development :

The block size plays a role in determining the deviation of the values in the pixels. The 3x3 window was found to have a higher MSE value when compared to the 9X9 values. Finally, it has been observed that the dynamic programming approach to obtain the right and left disparity maps has been relatively more efficient with respect to the block matching approach based on the analysis of the MSE values.

<u>Software and Program Development:</u>

Sum of Squared Differences (SSD) Matching:

To correspond which pixel in the right image correspond to each pixel in the rest image, the displacement of these two pixels are calculated using the Sum of Squared Differences to estimate the disparity. To reduce the noise occurrence in the estimation due to a poor signal to noise ratio, a window based cost estimation approach is chosen. SSD is an example of area based cost aggregation which calculates the sum of the squares of the differences between the window around the pixel of interest in left image to its corresponding row in the right image. The pixel displacements are assumed to be only along the row satisfying the epipolar constraint in which the search space is reduced to a difference of 75 pixels. For a right image I_R and for a left image I_L, the SSD over a window of size W is calculated by:

$$SSD = \sum_{(x,y)\in W} \left(I_R(x,y) - I_L(x,y)\right)$$

Implementation details:

- The left image I_L and the right image I_R were converted to grayscale from RGB.
- I_L and I_R are padded based on the window size chosen W which can be 3 or 9.
- The window estimates the SSD of the left image and loops over till a count of 75 in the same row of the right image in the direction of the shift.
- The intensity of the pixel with minimum SSD value is identified and the displacement of this pixel is noted in the corresponding disparity map.
- This is repeated over all the pixels in the row for each row to obtain the left disparity map.
- The Mean Square Error (MSE) of the disparity map is calculated with respect to the ground truth images provided.
- Finally the steps are repeated to estimate the right disparity map and its MSE.

Dynamic programming approach :

Dynamic programming approach works by minimizing cost function to produce optimal solution. In our implementation, local cost is defined as the difference of the pixel intensities in specific interval. Minimizing this cost will result in an optimal match for a pixel in one view with a pixel in another view.

$$LocalCost = I_R(x,y) - I_L(x,y)$$

Global search using dynamic programming resolves the problem of occlusion. In this project, matching costs at occlusion boundaries is minimized by defining a small fixed occlusion cost (OCost). In the next section, output with two occlusion costs of 20 and 40 is presented.

The final cost of matching pixel x in one view with pixel y in other view is calculated using following equation:

$$Cost(x,y) = min(\ Cost(x-1,y-1) + LocalCost,\ Cost(x-1,y) + OCost,\ Cost(x,y-1) + OCost)$$

Implementation Details :

- Left view image and right view image were converted from RGB to grayscale.
- For each pixel, the cost of matching is calculated using above equation
- The positions of minimum cost are traced and difference in indexes of left and right image are stored as disparity values.
- For all other positions where the cost was not minimum, zero value is stored which indicates that those positions were occluded.
- Mean square value of the disparity map is calculated with respect to ground truth.

Lessons learned from programing codes development :

Dynamic programming when incorporated into stereo vision disparity map estimation has been the major concept learnt in developing the code by dividing the main problem into subproblems using the same values without any recomputation.

Summary :

Disparity maps, key to perception of depth, from the two rectified images have been obtained using two methods of approach, block matching and dynamic programming. In block matching the implementation with the 9x9 window gave better MSE values when compared to the ground truth images. Overall, dynamic programming had a key advantage over block matching method due to its lower MSE values.

Besides learning in depth about stereo vision through this project, this course has provided a platform to build on image processing concepts and integrating them with concepts from computer vision. Learning more about image processing helped us to appreciate the nuances of viewing an image and modifying it based on different filters. Unlike usual homework assignments, this course enabled us to focus on integral concepts and learn more about its practical applications rather than emphasizing on purely theoretical knowledge.

Acknowledgement :

We are grateful to Professor Chang Wen Chen for taking this course on Computer Vision and Image Processing very lucidly and helping us grasp concepts well. We are also thankful to the teaching assistants for their timely help and guidance in this project. We thank the University at Buffalo for giving us an opportunity to have in-depth analysis through this course.

References :

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

[2]Vision, R. (2016). *Real Time Obstacle Depth Perception Using Stereo Vision*. [online] Ufdc.ufl.edu. Available at: http://ufdc.ufl.edu/UFE0046766/00001

[3]https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect16.pdf

[4]http://campar.in.tum.de/twiki/pub/Chair/TeachingWs09Cv2/3D_CV2_WS_2009_Stereo.pdf