2020 Computer Vision
Assignment 1: Disparity Estimation

Vandit Jyotindra Gajjar
School of Computer Science, The University of Adelaide
vanditjyotindra.gajjar@student.adelaide.edu.au
Student ID: a1779153

Abstract

Image filtering is one of the core computational tasks in today's computer vision tasks both conventional and current deep learning tasks. It is essentially a task in which a particular patch-size or kernel size is selected and then slide through the entire image(s) to get certain outputs such as Image blurring, sharpening, edge detection, etc. In this assignment, we use these types of operations for disparity estimation between two images. This report focuses to solve the assignment questions with detailed analysis and in-depth context relating to disparity estimation, image matching, and a variant of the Patch Match algorithm [1]. To evaluate our results, we have used the Middlebury Stereo dataset [2], which contains a large number of stereo pairs. The source code is publicly available at *https://github.com/Vanditg/COMP-SCI-7315---Computer-Vision*

Task 1
Answer (a)

Figure 1 contains different pairs of images taken from the Middlebury Stereo dataset. The reason for choosing these pairs have been answered in the answer (b).

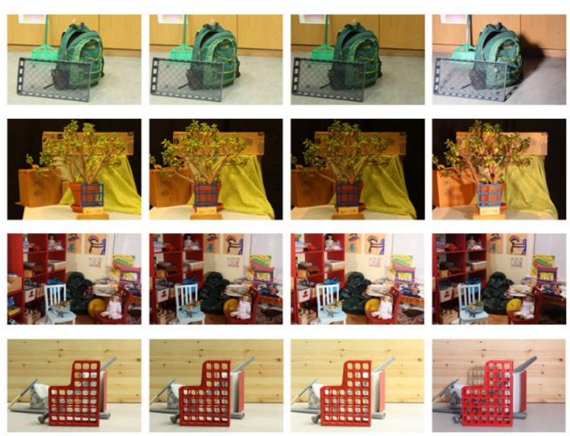

Figure 1. Selected image pairs from the Middlebury Stereo dataset. From left to right images are in the given sequence: Image Left, Image Right, Image Exposed, and Image Light condition changed.

Answer (b)

The following are the main reasons to choose these pairs to give a better in-depth understanding.

- The Middlebury Stereo dataset contains so many image pairs in which most of the images are stationary (Not moving cameras), but these image pairs have been carefully selected as there has been a significant change in the camera, considering that these images might have been taken using moving camera.
- The selected image pairs are having very dense textures i.e. lots of information, objects, etc. This will help to understand how the window size will impact when we estimate disparity for these pairs. Also, because of the rich information on these pairs, we can get the best window size for all other pairs as well (According to our experiments we found that window size 3 works very well on other image pairs).
- For the stereo correspondence problem, because of significantly changed camera conditions, we'll be able to identify the most scene points because they are visible in both the images, also the matching regions are similar in this case.
- To get a better understanding that these images are slightly given an advantage over other image pairs, we have provided disparity estimation for one of the pairs in Figure 2.

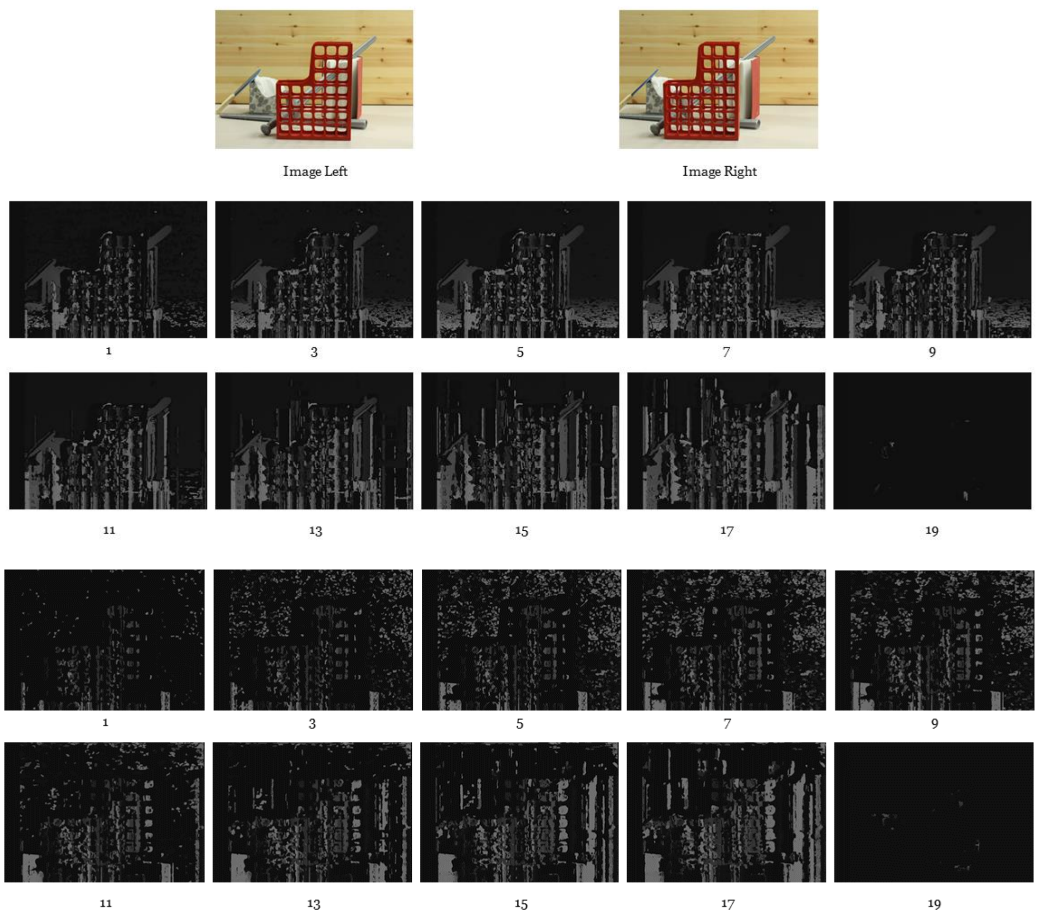

Figure 2. The first row shows input images. The second and third-row shows disparity images from left to right with different window sizes. The last two row shows disparity images from right to left with different window sizes.

- While we haven't performed the experiments when we have written this section, we believe that the following will be the advantage of this image pairs for the given tasks:
    - While computing the SSD, because of the significantly changed image pairs, it will be interesting to see how the score distribution is affected.
    - For normalized cross-correlation image transformation, the result might affect due to dense textures in image pairs.
    - It will also be interesting to run patch-match for reconstruction, because of the changed camera conditions it seems that the reconstruction either will not be successful or might give good results for some of the pairs.

Computer Vision (7315)

Task 2

Answer (a)

First of all, for computing SSD, we're considering two image pairs, a normal left image, and the normal right image. After that we consider another image pair consists of the normal left images and enhanced right images, and finally, we consider the normal left images and changed light condition right images. Figure 3 shows the result of our experiment for the normal left and normal right image pairs for different image pairs.

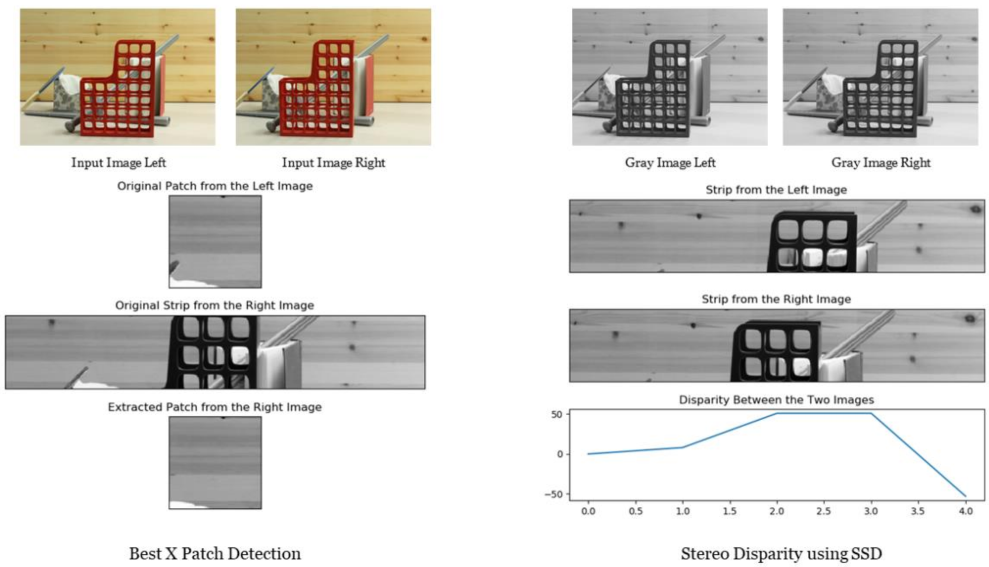

Figure 3. Stereo Disparity estimation using SSD. First of all, for normal input images, we convert them into grayscale images. After that, for the selected manual patch we find the best patch. Finally, we compute the disparity from the left and right strip using SSD to show the disparity estimation in the graph.

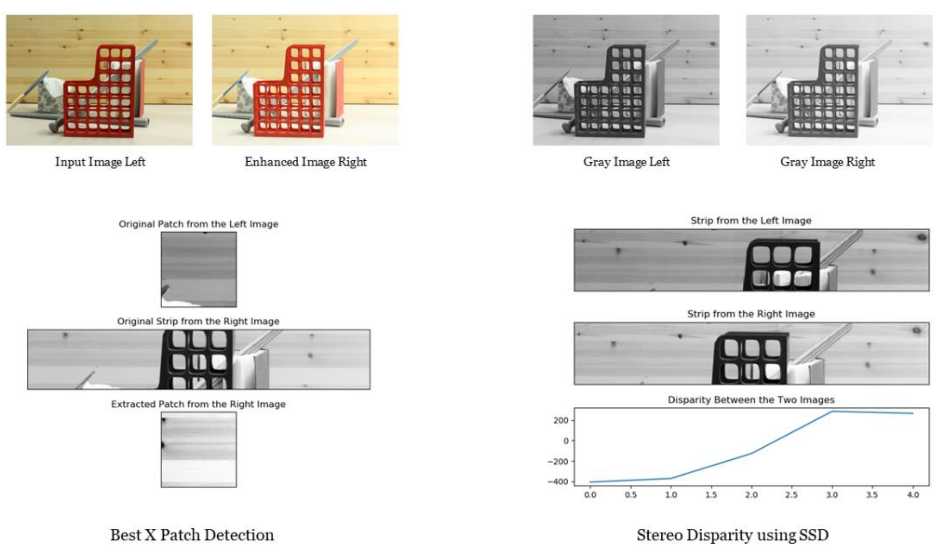

Figure 4. Stereo Disparity estimation using SSD. First of all, for normal input image and enhanced right image, we convert them into grayscale images. After that, for the selected manual patch we find the best patch. Finally, we compute the disparity from the left and right strip using SSD to show the disparity estimation in the graph.

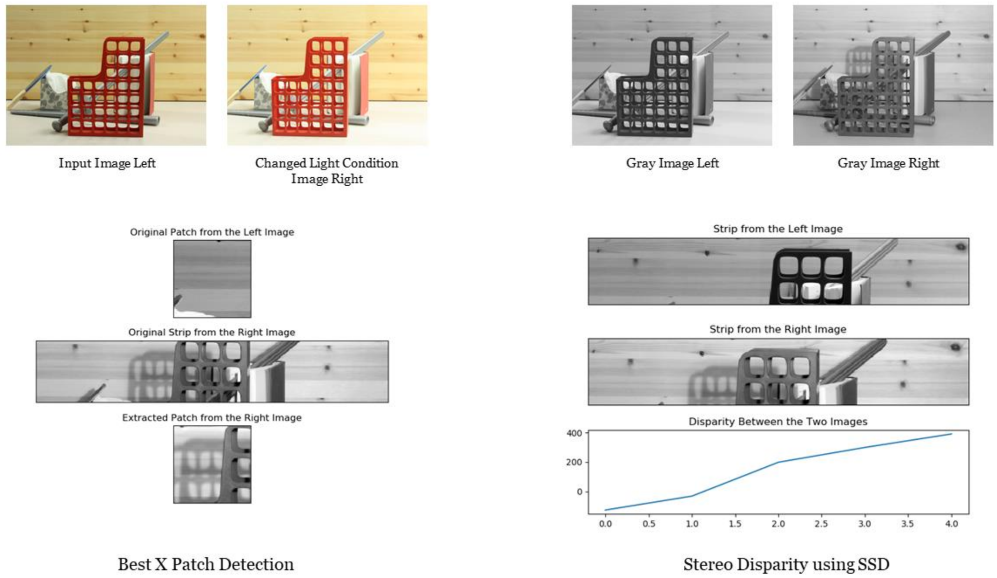

Figure 5. Stereo Disparity estimation using SSD. First of all, for normal input image and changed light condition right image, we convert them into grayscale images. After that, for the selected manual patch we find the best patch. Finally, we compute the disparity from the left and right strip using SSD to show the disparity estimation in the graph.

From figures 3, 4, and 5, we can see that for different scene points and for different changed image condition our score distribution changes significantly as shown in the graph. The reason for this case might be the change of pixel value as SSD is eventually finding the given manual point from the first image to the point in the second image with a good score of similarity. Thus for enhanced and changed light condition images, the extracted patch is different corresponding to our experiments and thus leading to significant changes in our score distribution. Also, considering the scene points, they do not have the same characteristic as their RGB pixel values are different or the grayscale value if we are converting them. The result shown here is only for one of the pairs from the dataset.

Answer (b)

Considering changing the patch size for image pairs this will affect the score distribution. We provide a set of experimental images to prove this hypothesis. From the result, we have gained from the experiment we were not able to find the single best patch size for all scene points. Also by varying the patch sizes the change in score distribution shows that smaller patch size has more advantage rather than bigger patch sizes as smaller patch size gives the good depth result and provides good textures in pairs while bigger patch size fails to do so.

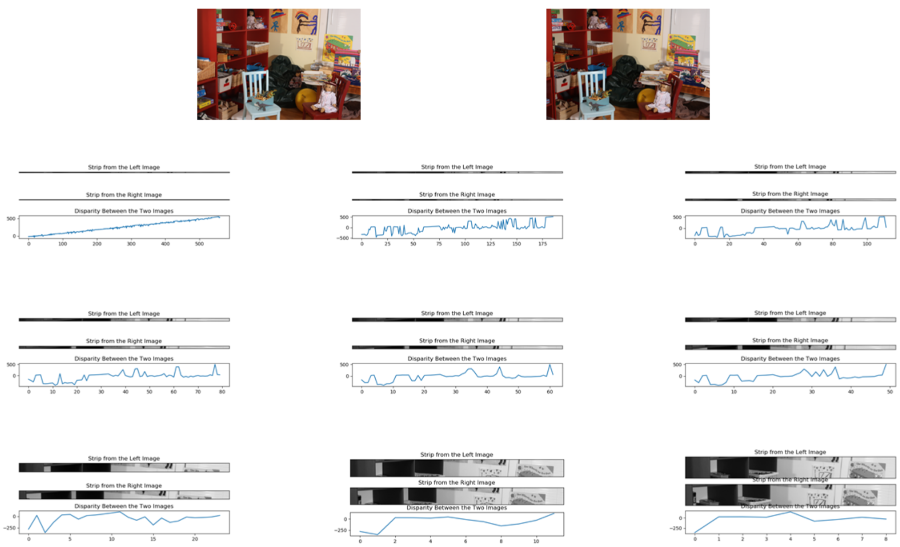

Figure 6. Different patch sizes for images to computing the score distribution. We can see that the score distribution changes significantly.

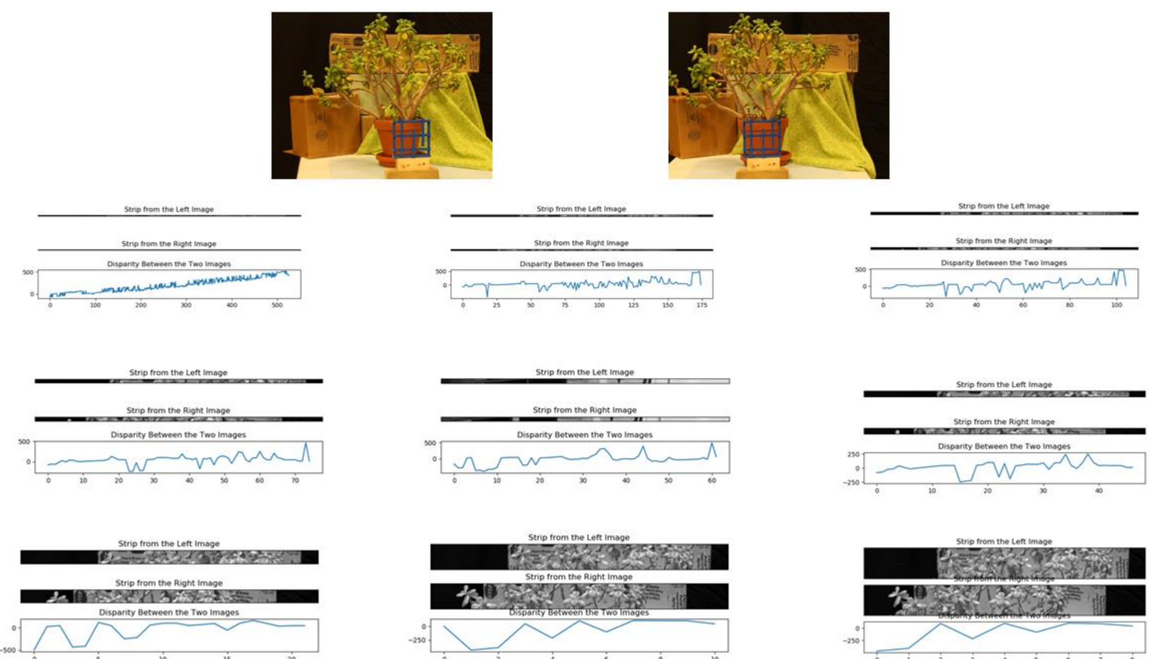

Figure 7. Different patch sizes for images to computing the score distribution. We can see that the score distribution changes significantly.

Answer (c)

A variant given in the assignment is Normalized Cross-Correlation, in which a high score coincides with the correct match in the image. We try to incorporate this image transformation into our experiment where we take a source image and a target image to extract the correct patch in the target image. For some of the experiments, we get a closer match but, in most cases, we get failure cases, which are not close enough or either very far from the original patch. We believe that by doing this experiment, we able to get the intuition that incorporating NCC gives the correct match in the source image. We back up this claim by providing experiments provided in figure 8, 9 and at *https://github.com/Vanditg/COMP-SCI-7315---Computer-Vision*

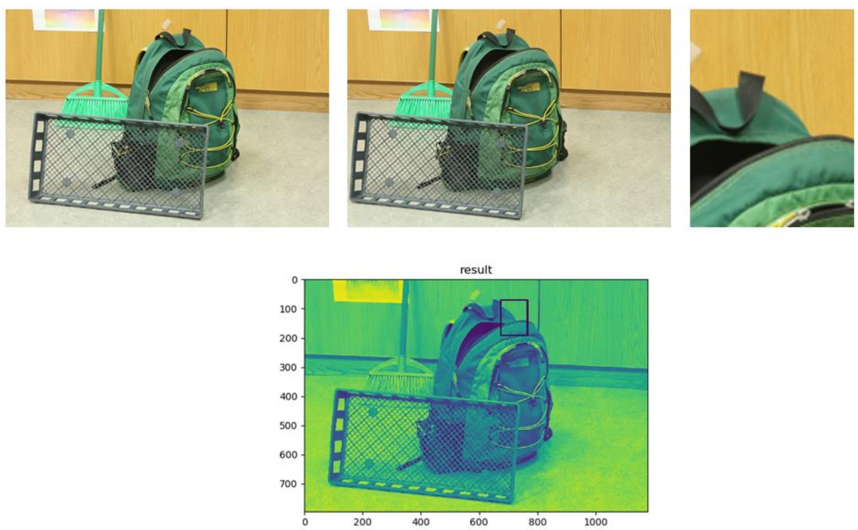

Figure 8. We incorporated NCC image transformation in our experiment to correctly match the target image patch to the source image. Here detected result shown in the bounding box.

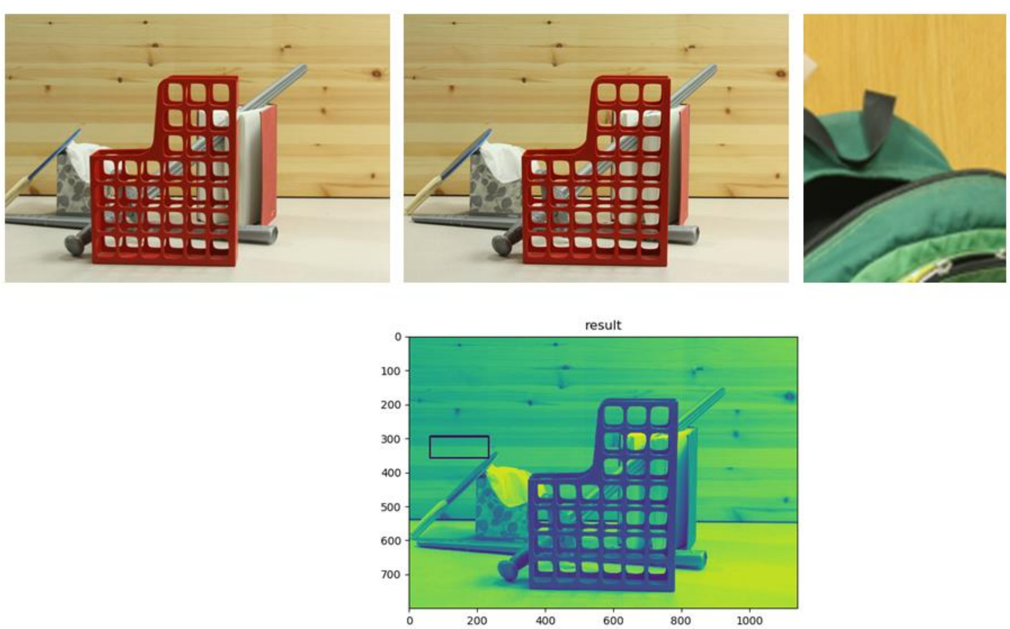

Figure 9 We incorporated NCC image transformation in our experiment to correctly match the target image patch to the source image. Here detected result shown in the bounding box which is a failure case.

From this figure 10 and 11, we can see that certain areas are smoother, and some areas are having loss of finer details while computing NCC.

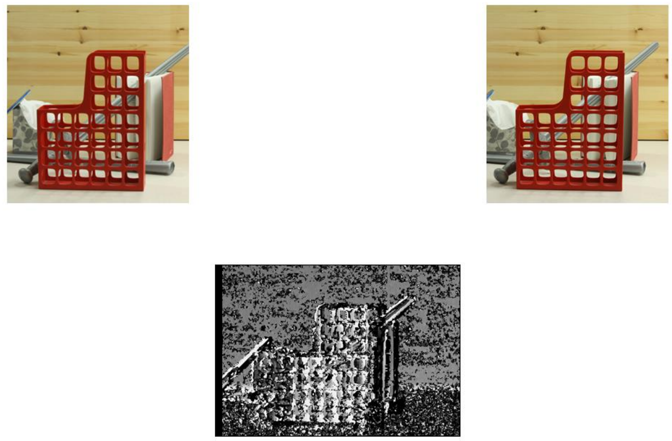

Figure 10. NCC disparity computation for one of the pairs from the dataset.

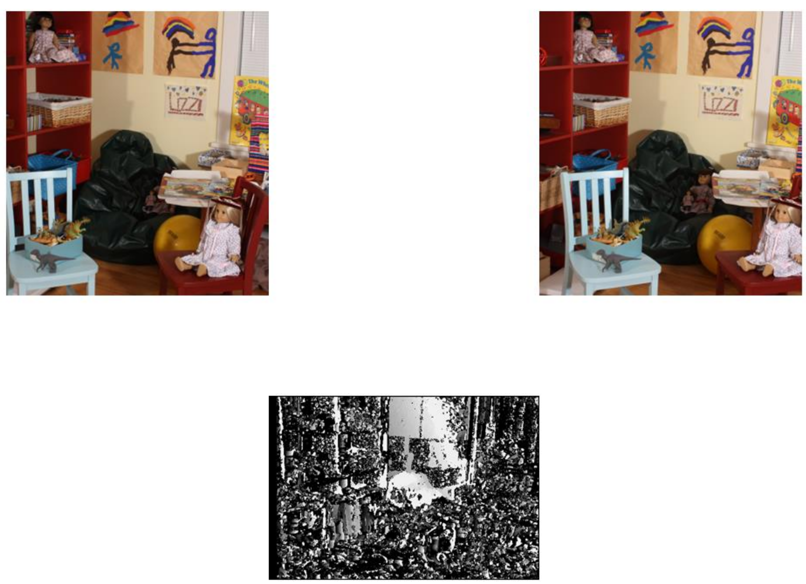

Figure 11. NCC disparity computation for one of the pairs from the dataset.

Computer Vision (7315)

Task 3

Answer (a)

We have implemented the Patch Match algorithm [1] as described in the assignment to see that for the scene points whether it is improvement or disadvantage. The figure 12 and 13 shows the corresponding scene points match using a sparse set of lines.

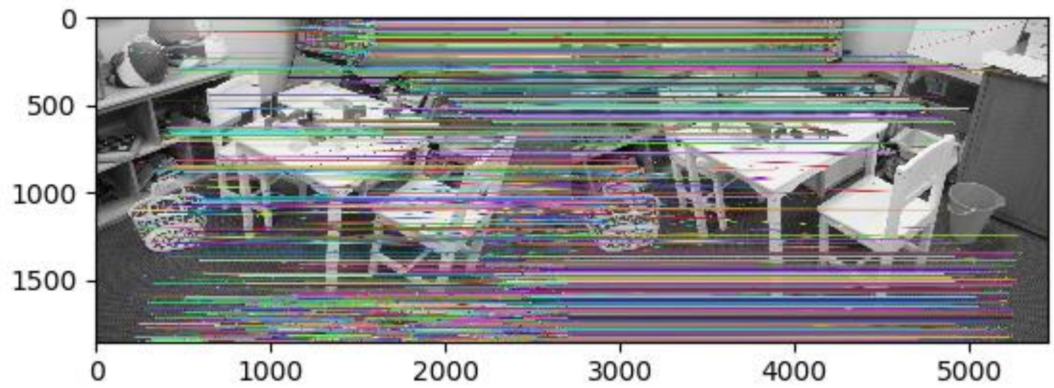

Figure 12 Dense correspondences between the two images by plotting them side-by-side and drawing a sparse set of lines between correspondences.

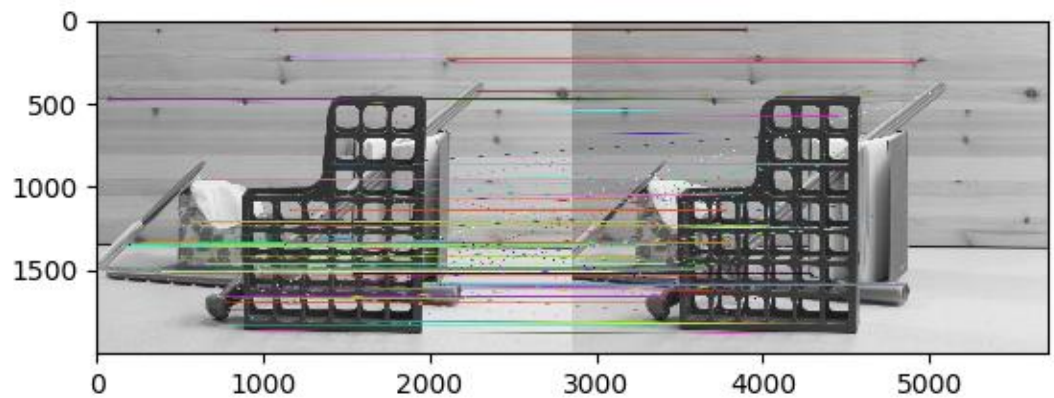

Figure 13 Dense correspondences between the two images by plotting them side-by-side and drawing a sparse set of lines between correspondences.

The goal of the Patch Match algorithm is to find the correspondence by defining a nearest-neighbor field which is overall possible matches of the patch in the left image, for some distance function of two patches. So for a given patch coordinate of left images and its corresponding nearest neighbor in the right image. The Patch Match algorithm can significantly reduce the complexity dependency on the search space size, thus leading to a faster result, and from this experiment, we learned that it has improved the search faster and leading to good results.

Answer (b)

We have tried to incorporate disparity magnitude in our experiment in which we can see that the left image is a disparity estimated image while the right is the magnitude disparity image. We can see that the correlation score is higher only when darker parts of the image and brighter parts of the image overlap brighter parts of the image. We provide results in the figure 14, 15, and at *https://github.com/Vanditg/COMP-SCI-7315---Computer-Vision*

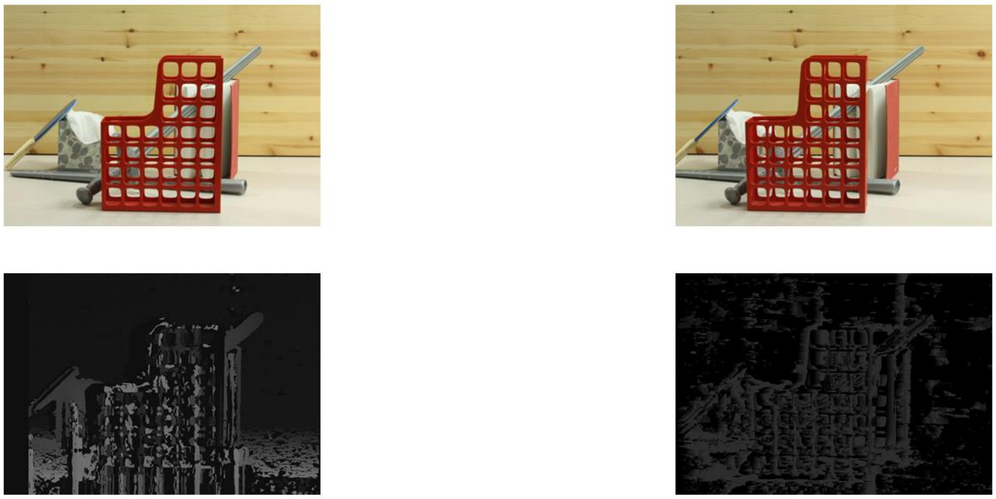

Figure 14. Disparity magnitude for given image pair. The first row contains image pairs left and right. While the second raw contains disparity estimated image and disparity magnitude image.

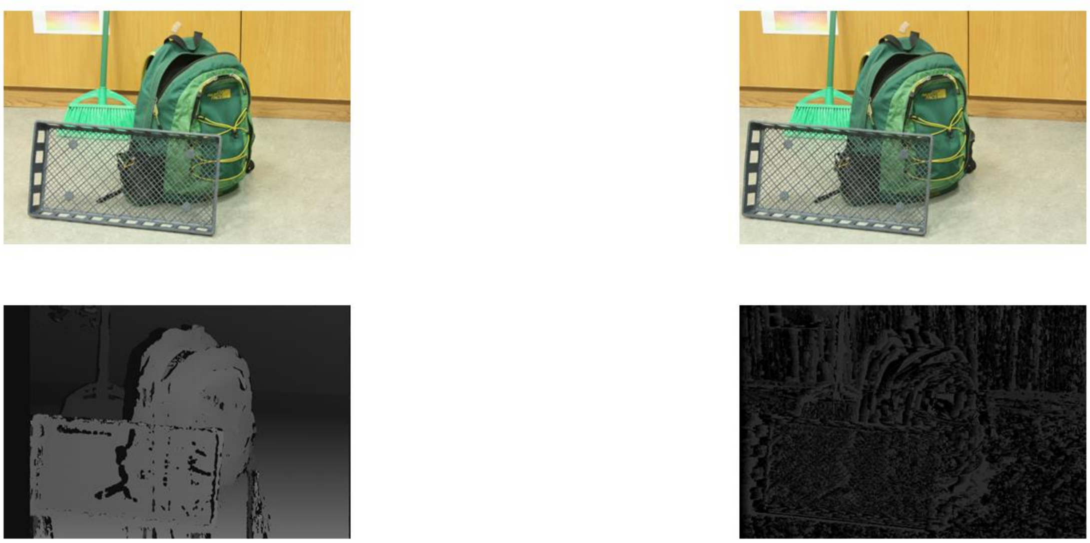

Figure 15 Disparity magnitude for given image pair. The first row contains image pairs left and right. While the second raw contains disparity estimated image and disparity magnitude image.

Answer (c)

We have implemented Patch Match algorithm described in the assignment and produced some of the images shown in the figure 14, 15, and at https://github.com/Vanditg/COMP-SCI-7315---Computer-Vision

- As we can see here that the images are reconstructed with most information but as we can also see the texture difference between the original image and the reconstructed image. The reconstructed image looks a little less dense than the original one, the reason might be that as the process is random for finding the best displacement, thus might leading to poor results.

Figure 14. Reconstruction using the Patch Match algorithm. The first raw contains input images – left image and right image. The last row contains the reconstructed image.

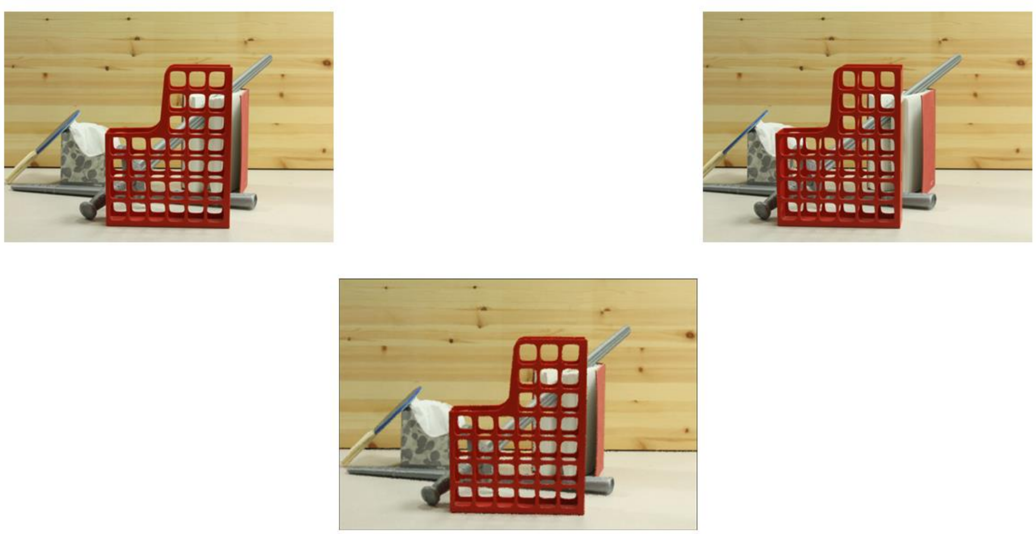

Figure 15 Reconstruction using the Patch Match algorithm. The first raw contains input images – left image and right image. The last row contains the reconstructed image.

Task 4
Answer (a)

The Median filter is non-linear. Unlike linear filters, median filters replace the pixel values with the median value available in the local neighborhood (say, 5x5 or 3x3 pixels around the central pixel value). Also, the median filter is edge-preserving (the median value must be the value of one of the pixels in the neighborhood). Eventually, Median filtering is a technique that is sometimes useful as it can preserve sharp features (e.g. lines) in an image whilst filtering noise. For our case from the experiment, we can see that because of median filtering the images are smoothed and reduces the noise and giving some good output images. We have provided some results in the figure 16, 17, and at *https://github.com/Vanditg/COMP-SCI-7315---Computer-Vision*

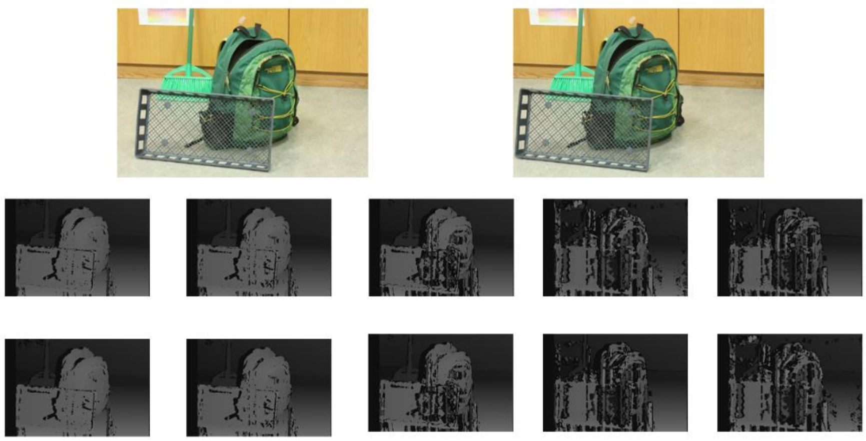

Figure 16. Applying median filtering on different patch sized disparity maps. We can see that the images are smoothed and contain less noise compare to actual disparity maps. The first raw contains input images. The second raw contains disparity images of different window sizes in given order 1, 5, 9, 13, 17. The last raw contains output images of median filtered images.

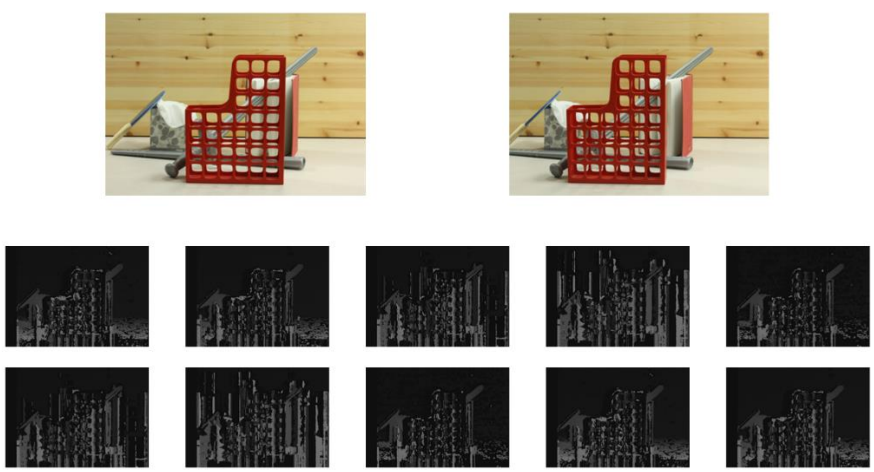

Figure 17 Applying median filtering on different patch sized disparity maps. We can see that the images are smoothed and contain less noise compare to actual disparity maps. The first raw contains input images. The second raw contains disparity images of different window sizes in given order 1, 5, 9, 13, 17. The last raw contains output images of median filtered images.

References:

[1] Barnes, Connelly, et al. "PatchMatch: A randomized correspondence algorithm for structural image editing." ACM Transactions on Graphics (ToG). Vol. 28. No. 3. ACM, 2009.

[2] Scharstein, Daniel, and Richard Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms." International journal of computer vision 47.1-3 (2002): 7-42.

[3] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[4] Walt, Stéfan van der, S. Chris Colbert, and Gael Varoquaux. "The NumPy array: a structure for efficient numerical computation." Computing in Science & Engineering 13.2 (2011): 22-30.