**Assignment 2:**

**Stereo Matching**

Ben Solomon 204454615

Din Carmon 209325026

1. Calculating ssdd for the below recrified images and dsp\_range=2:



|  |  |  |
| --- | --- | --- |
| 2-1 | 3-1 | 4-1 |
| 7-2 | 8-2 | 9-2 |
| 12-3 | 13-3 | 14-3 |

1. For , disparity value = 0. Differences:
2. For , disparity value = 1. Differences:

|  |  |  |
| --- | --- | --- |
| 2-1 | 3-1 | 4-1 |
| 7-2 | 8-2 | 9-2 |
| 12-3 | 13-3 | 14-3 |

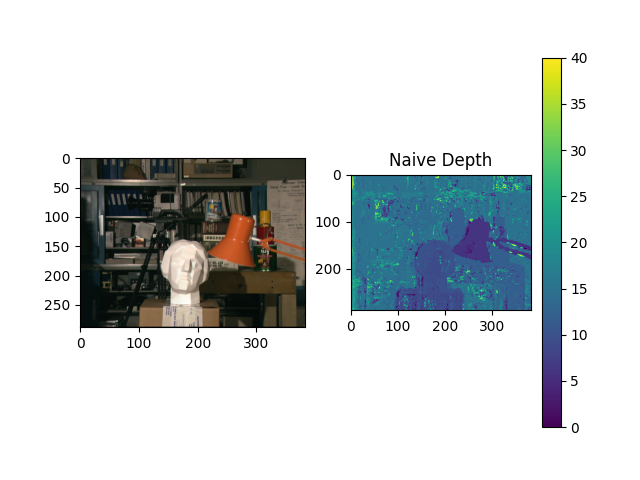
1. For , disparity value = -2. Differences:

|  |  |  |
| --- | --- | --- |
| 8-2 | 9-2 | 10-2 |
| 13-3 | 14-3 | 15-3 |
| 18-4 | 19-4 | 20-4 |

1. For , disparity value = -1. Differences:

|  |  |  |
| --- | --- | --- |
| 8-2 | 9-2 | 10-2 |
| 13-3 | 14-3 | 15-3 |
| 18-4 | 19-4 | 20-4 |

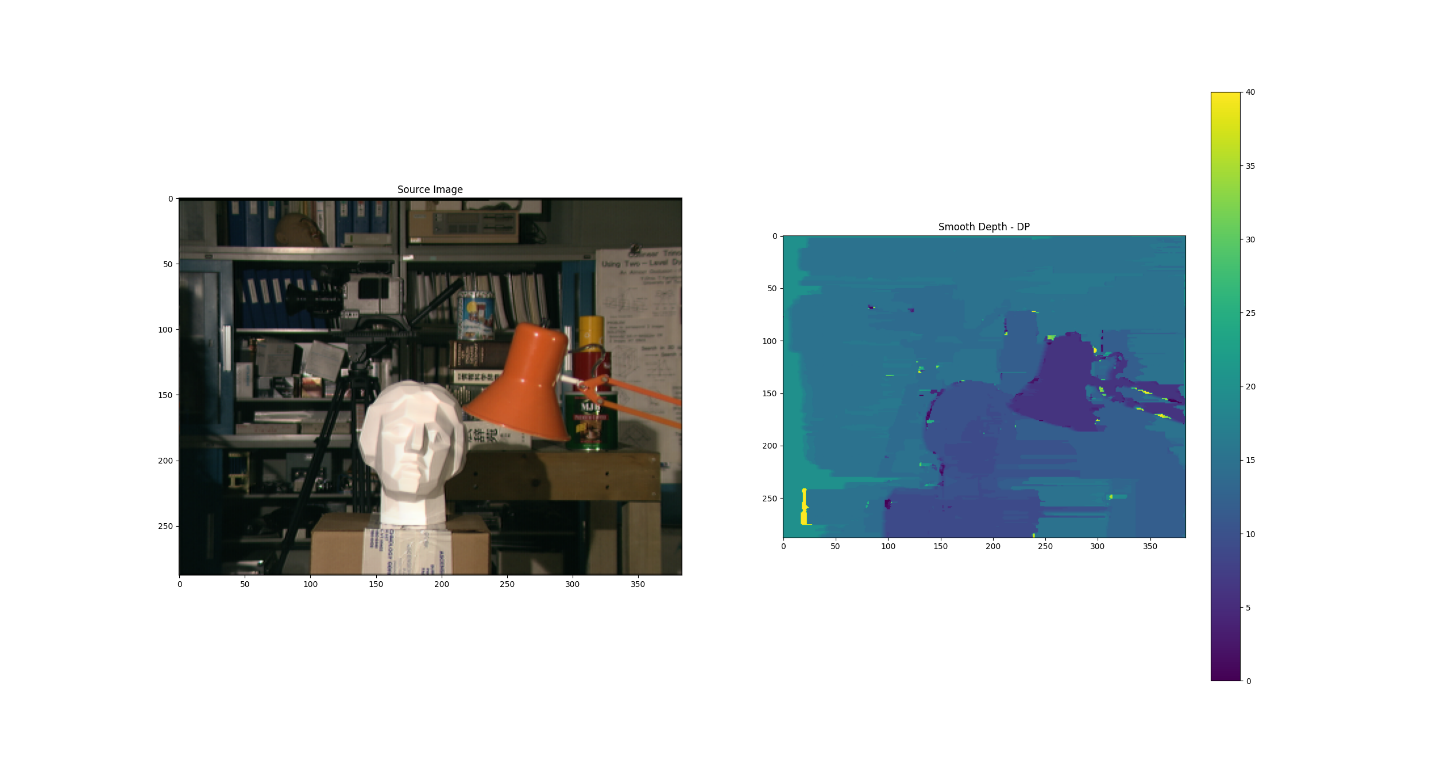
1. The function ssd\_distance is implemented and can be viewed in the python src files.
2. Naive labeling output from the distances tensor:



1. The result of the naive depth labeling using the SSDD tensor highlights several issues. Firstly, in areas with repetitive or low-texture patterns (e.g., the background books or monotonic surfaces), multiple local minima arise along the label dimension, leading to incorrect label selection. Naively selecting the minimum often results in suboptimal outcomes, especially in regions with similar objects or lacking distinctive textures. The lack of a smoothness prior worsens the issue, as the method relies solely on local minima, resulting in noisy and inconsistent labels for objects with uniform depth, as seen in Section 3 of the depth map. Additionally, fixed parameters, such as window size, restrict the algorithm's adaptability and reduce accuracy. This can be observed below, where pixels are incorrectly mapped, introducing noise into the forward-mapped image.



1. The function dp\_grade\_slice is implemented and can be viewed in the python src files.
2. Dynamic Programming labeling output:



1. The depth map from Section 6 is much smoother than the naive map because it solves each line optimally and incorporates the smoothness prior used in the dynamic programming method. This results in fewer mismatches and less noise in the forward-mapped image, aligning it more closely with the right image, as can be seen below. However, since the DP method solves each line independently rather than all lines together, some artifacts and inconsistencies between lines still cause mismatches in the forward map.