Stereo Disparity Estimation

Introduction

To solve the problem of stereo disparity matching, I plan to use OpenCV's StereoSGBM implementation as a starting point. I also plan to explore the Pyramid Stereo Matching Network (PSMNet), which is a deep learning-based approach introduced in CVPR 2018 that has shown promising results in stereo matching tasks. Additionally, I will also experiment with some interpolation methods to further improve the disparity map and achieve a higher PSNR score.

In this specific problem, the two images are already rectified, which simplifies the process by reducing the problem to a one-dimensional horizontal search for corresponding pixels in the two images.

Throughout this report, I will be illustrating my experiments with the Reindeer images. Performance of each method will be evaluated based on average PSNR scores of the 3 images (Art, Dolls, Reindeer).

Approach 1: Stereo Semi-Global Block Matching (SGBM)

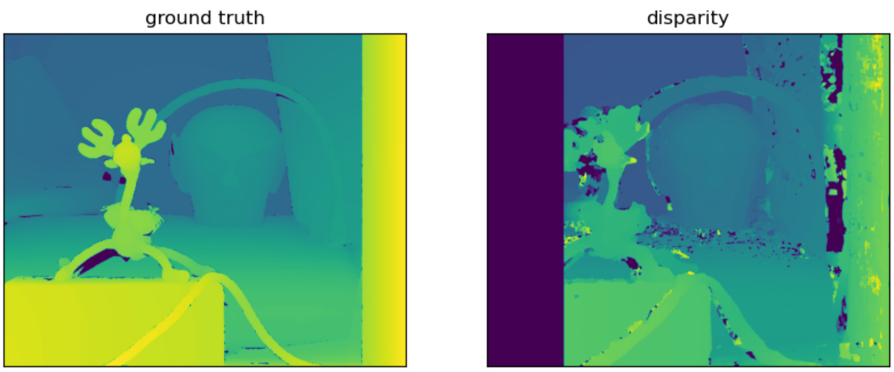

Figure 1.1, PSNR: 12.45

StereoSGBM works by computing a matching cost for each pixel in the left and right images, and then finding the disparity that minimizes this cost. It uses a semi-global approach to aggregate the matching cost over multiple paths to generate a dense disparity map.

From the results shown in figure 1.1, the disparity map generated has a few problems:

1. **Vertical black bars** on the left, where **intensity values are unknown**. These are areas where there is no corresponding point in the other image, and therefore, no disparity value can be calculated. In such cases, StereoSGBM assigns a default invalid disparity value, which can result in the vertical black bars in the disparity map.

2. The disparity map is **noisy** with many **black spots** which may be to textureless regions where there are no distinctive visual features, which makes it difficult for StereoSGBM to find corresponding pixels in the left image.

Through experimentation, I set an upper bound for intensity values to be classified as "unknown". Intensity values lower than 70 are considered unknown and will be handled by post-processing methods.

Post-processing methods

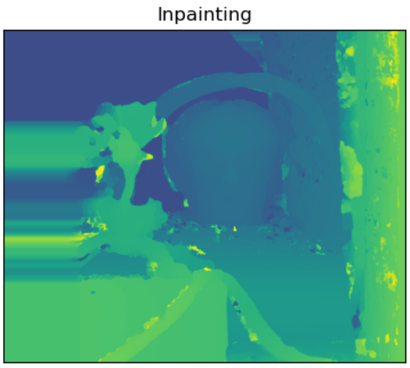

Figure 1.2.1, PSNR: 18.57

Method 1: Inpainting using Navier Stokes Equation

To address this issue of vertical black bars, we can apply post-processing methods such as inpainting to fill in the unknown areas. Navier-Stokes inpainting fills in missing regions of an image using fluid flow modelling, and can improve the accuracy and completeness of disparity maps. This method increased PSNR score by 6.12 points but also results in unwanted artifacts as shown in figure 1.2.1 where the column of pixels on the leftmost are pulled to the left side of the image.

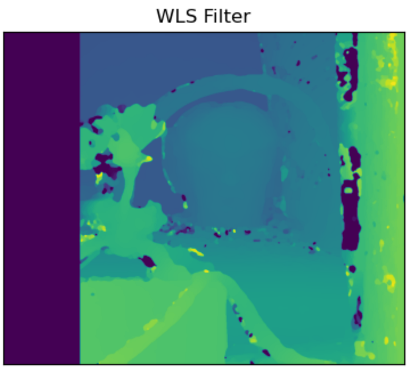

Figure 1.2.2, PSNR: 12.61

Method 2: Weighted Least Squares (WLS) Filter

To remove noise in the disparity map, we can apply post-processing methods like weighted least squares filtering to enhance accuracy of disparity estimation. The weighted least squares filter removes noise in a disparity map by assigning weights to the neighboring pixels based on their similarity, and computing the weighted average of the neighboring pixel values. This method improved PSNR score slightly by about 0.16 points.

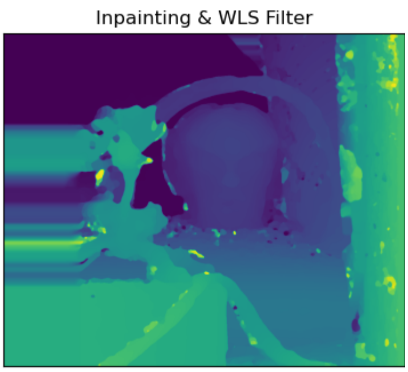

Figure 1.2.3, PSNR: 18.69

Method 3: Combination of Inpainting and WLS filter

I combined the first two post-processing methods by first applying inpainting to fill in larger areas of unknown disparities, and then removing noisy areas with WLS filtering. This method achieved a higher PSNR score than either method 1 or method 2 of post-processing.

StereoSGBM achieved the best results with post-processing method 3, achieving an average PSNR score of 19.74.

Approach 2: Pyramid Stereo Matching Network

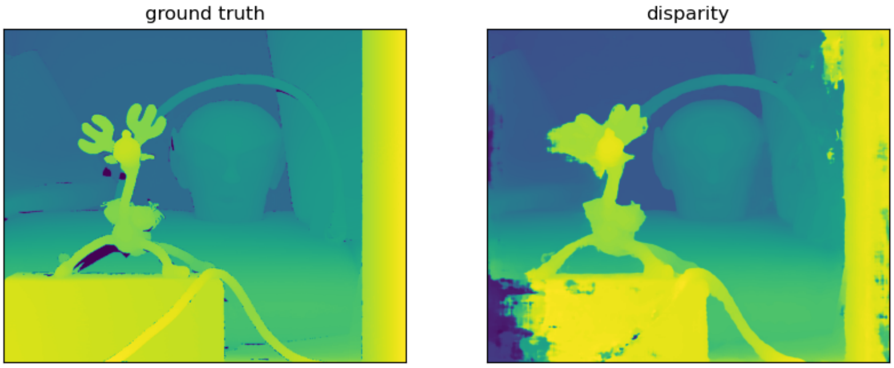

Figure 2.1, PSNR: 19.28

My second approach is utilizing Pyramid Stereo Matching Network (PSM-Net), a deep learning architecture that uses a spatial pyramid pooling module to extract multi-scale features from the input images. The network then uses a 3D convolutional network to estimate the disparity map, which is refined using a stacked hourglass network. The PSM-Net has been shown to achieve state-of-the-art performance on various stereo matching benchmarks, and is able to handle challenging scenes with textureless regions, occlusions, and large disparities [1].

The pretrained model provided in PSMNet's github repo was pretrained on Scene Flow dataset [2]. I followed the steps in the github repo, fine-tuning the PSMNet on KITTI 2015 dataset for 250 epochs, which took about 24 hours to complete on an Nvidia GTX 970.

Running PSMNet without further post-processing gives us an average PSNR score of 20.05, which outperforms StereoSGBM with post-processing applied. As you can see from figure 2.1 that there are still some unknown intensities in the disparity map, which may be reduced by further fine-tuning the model with more data, but due to time constraints I decided not to. Instead, I have tried a few post-processing methods to fill in the unknown disparities and improve performance.

Post-processing methods

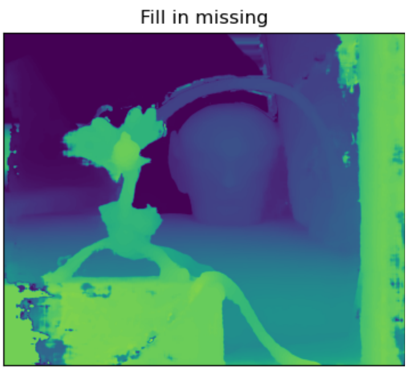

Figure 2.2.1, PSNR: 21.93

Method 1: fill in missing values from StereoSGBM

Here we fill in the unknown intensity values in PSMNet with values from the StereoSGBM implementation, increasing the PSNR score by 2.65 points.

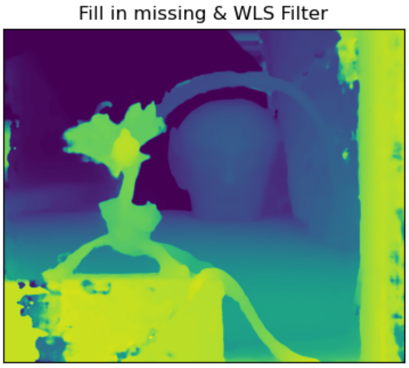

Figure 2.2.2, PSNR: 22.11

Method 2: combination of method 1 and WLS filter

Here we apply a weighted least squares filter to further remove noise from the disparity map generated by method 1, which resulted in a slightly better PSNR score.

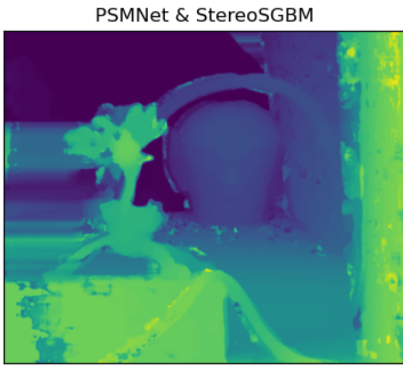

Figure 2.2.3, PSNR: 21.29

Method 3: average between StereoSGBM and PSMNet

Taking average between disparity maps generated by StereoSGBM and PSMNet resulted in poorer performance than method 2. Artifacts from inpainting in StereoSGBM are also carried over here.

PSMNet achieved best performance with post-processing method 2 applied, achieving an average PSNR score = 21.76.

Conclusion

For Stereo SGBM, I have found that post-processing methods such as inpainting and weighted least squares filtering can improve the accuracy and completeness of the disparity maps, and achieve the best results with a combination of these methods.

Overall, PSMNet with post-processing achieved the best performance with an average PSNR score of 21.76, outperforming Stereo SGBM with post-processing. However, further fine-tuning and experimentation may be necessary to achieve even better results.

If given more time to do this assignment, I will try exploring more recently published papers such as "StereoDRNet++," "Hierarchical Neural Stereo Matching via Adaptive Cost Volume Filtering," and "Neural Stereo Matching with Global Context Aggregation and Local Sub-networks," as these papers have achieved state-of-the-art results on stereo disparity matching and were published in recent years

References

[1] J.-R. Chang and Y.-S. Chen, "Pyramid Stereo Matching Network," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5410-5418.

[2] J.-R. Chang, "PSMNet: Pyramid Stereo Matching Network," GitHub repository, 2018. [Online]. Available: https://github.com/JiaRenChang/PSMNet.