Single Noisy Image Depth Estimation with Multi-Scale Feature Fusion Module

Yuxiao Chen
Stanford University
yuxiaoc@stanford.edu

Abstract

Single image depth estimation has become critical for lots of applications. However, most existing methods only focus on images without noise and they perform poorly for noisy images. In this work, I propose a learning based depth estimation method combined with denoise filtering. Specifically, I explore a network architecture consisting of an encoder, decoder, multi-scale feature fusion module, and refinement module. For the denoise filtering, I explore the median filter and bilateral filter. Experimental results show that the bilateral filter improves the performance of single noisy image depth estimation while median filter has negative effect on it.

1. Introduction

Depth estimation has become critical for autonomous driving[5], scene recognition[20], and human computer interaction[11]. Traditional depth estimation methods, like structure from motion and stereo vision matching, are built on feature correspondences of multiple viewpoints. Inferring depth information from a single image (monocular depth estimation) is an ill-posed problem since lots of stereoscopic information is lost. Therefore, many applications of depth estimation rely on lots of different data besides a single image.

One of most popular approach is to recover the 3D structures from a couple of images based on geometric constraints, and it has been widely investigated in recent forty years. In addition, sensors like RGB-D cameras and LIDAR are commonly used to get more depth information of the corresponding image. Although those methods achieve very good performances [9], these methods usually depend on image pairs or image sequences which are very difficult to collect in some scenarios. Moreover, the large size and power consumption of these depth sensors (RGB-D cameras and LIDAR) restricts their applications from small robotics, like drones, which is not even practical in some cases.

2. Related Work

With the rapid development of deep neural networks, monocular depth estimation based on deep learning has been widely studied recently and achieved promising performance in accuracy [6]. A variety of neural networks have manifested their effectiveness to address the monocular depth estimation, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), variational auto-encoders (VAEs) and generative adversarial networks (GANs) [4].

However, most of the work based on deep learning focus on images without much measurement noise. Although there is work exploring the robustness of the network on noisy images [15], the performance in general is not very well since the model is still trained on images without noise. There are some works using an end-to-end network model to estimate the depth from single noise image. The work [1] explores and designs an end-to-end CNN to estimate the depth from monocular blurry image. The work [10] designs an huge end-to-end model to estimate the depth from a night image in very bad light condition. However, the end-to-end models in those work are extremely large and diffcult to train since in general it is a very hard task to distinguish edges with noise without any prior knowledge of the noise.

In this work, I try to explore a two-stage process for single image depth estimation task. I will denoise the image first before feeding into the network. With the help of denoising from a simple denoiser, it will be easier for the network to learn and train.

3. Approach

In summary, the problem in my task can be split into two stages. In the first stage, I explore several different common denoising techniques with the assumption and prior knowledge that gaussian noise is added to RGB image. In the second stage, I will use a medium size neural network to estimate the depth from the denoised image. For the neural network, I use multi-scale feature fusion and encoder-decoder structure. In addition, I am motivated by the previous studies on statistical properties of range images of natural scenes

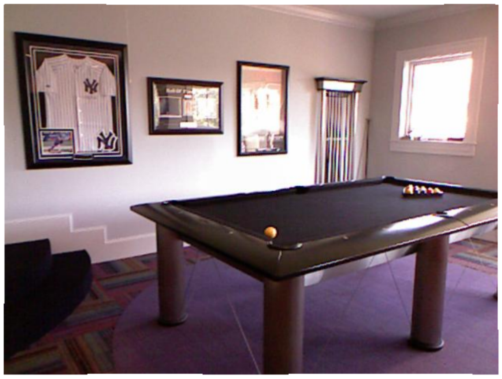


Figure 1. Ground Truth RGB image

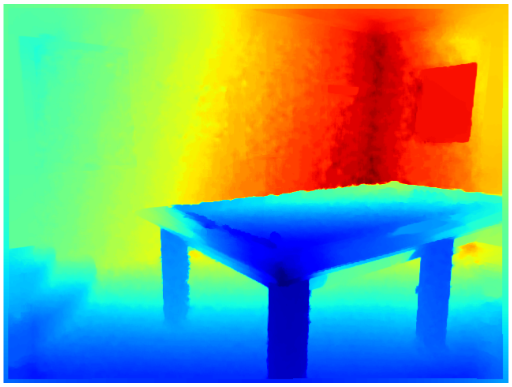


Figure 2. Ground Truth Viridis Depth Map

[16], and use the depth difference, norm difference and gradients altogether to form the loss.

For the input, I will use the NYU-Depth V2 dataset [12] which consists of a variety of indoor scenes, and is the most widely used dataset for the task of single view depth prediction. An example of RGB and depth pair of NYU-Depth V2 dataset is shown in figure 1 and figure 2.

Specifically, I use the official splits for 464 scenes, i.e., 249 scenes for training and 215 scenes for testing. Following suggestion in work [7], I downsample images from original size (640 × 480) to 320 × 240 pixels using bilinear interpolation, and then crop their central parts to obtain images with 304 × 228 pixels. For training, the depth maps are downsampled to 114 × 152 to fit the size of output.

For evaluation, I use the following accuracy measures that are commonly employed in the previous studies [7]:

Root mean square error (RMSE)

$$\sqrt{\frac{1}{P}\sum_{i=1}^{P}(d_i - g_i)^2}$$

Mean square error (MSE)

$$\frac{1}{P}\sum_{i=1}^{P}(d_i - g_i)^2$$

Mean absolute error (MAE)

$$\frac{1}{P}\sum_{i=1}^{P}\mid d_i - g_i \mid_1$$

Mean Relative Error (REL)

$$\frac{1}{P}\sum_{i=1}^{P}\frac{\mid d_i - g_i \mid_1}{g_i}$$

Mean Log 10 Error (log 10)

$$\frac{1}{P}\sum_{i=1}^{P}\mid log_{10}d_i - log_{10}g_i \mid_1$$

Threshold accuracy: percentage of d_i such that

$$max(\frac{d_i}{g_i}, \frac{g_i}{d_i}) < threshold$$

Note that the total number of pixels used in all evaluated images is denoted by P.

For the denoising task, I will explore two very commonly used denoising technique. The first denoising method is the median filter [2], which is a non-linear filter often used to remove noise from an image with good result. The median filter is very simple to implement but it is not aware of the spatial location of each pixel. In addition, another big disadvantage is that it is difficult write an analytical equation to represent a median filter. The second denoising method I explore is a more advanced technique called bilateral filter [14] which is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. The bilateral filter weights the intensity of each pixel with a weighted average of the intensity values from nearby pixels f_r. In my implementation, I use the Gaussian distribution for this weight g_s. The bilateral filter is defined as

$$I^{\text{filtered}} = \frac{1}{W_p}\sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

Both filters have been shown to perform good results with efficient denoising time [14]. Although there are more advanced denoising filters, those techiniques with long denoising time is not suitable for my application.

For the depth estimation part, inspired by the work [8], I use a network architecture consists of four modules: an encoder (E), a decoder (D), a multi-scale feature fusion module (MFF), and a refinement module (R). As shown in the

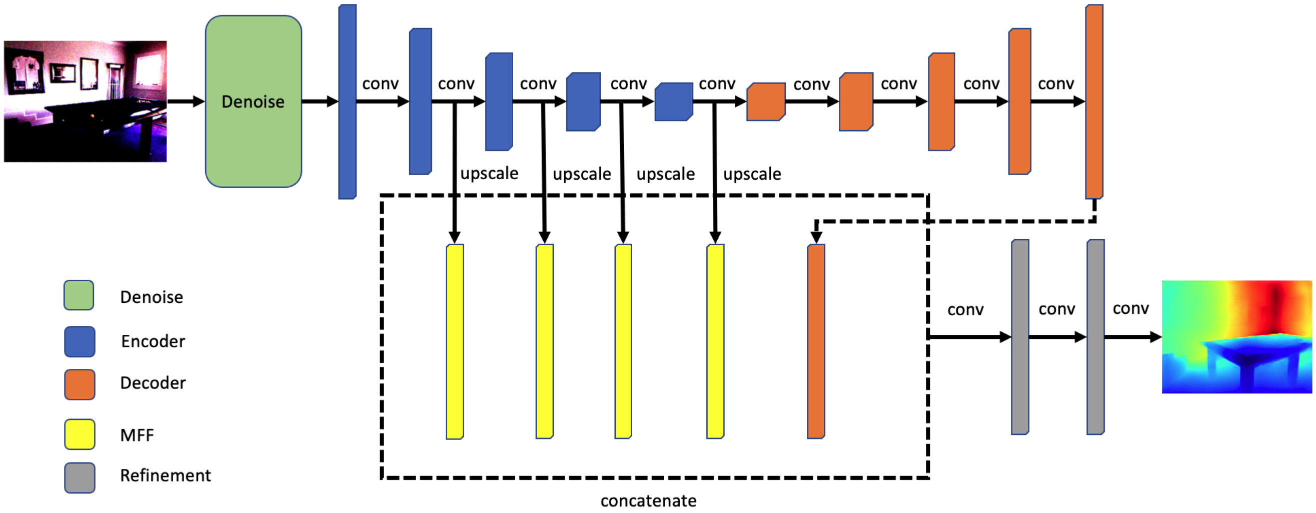


Figure 3. An illustration of the overall model with denoiser and MFF

figure 3, the encoder (E) extracts features at multiple scales and levels. In my implementation, I use the ResNet-50 [13] as the encoder. The decoder uses four up-sampling blocks to gradually up-scale the output from the encoder while decreasing the number of channels. In this work, I also explore the usage of the MFF module, which merge four different scale features from the encoder by performing channel-wise concatenation. I expect that the MFF module could help the model to better learn the features at different scales as shown in the work [8]. In the end, a refinement module consisted of three convolutional layers is used to give the final prediction for the depth map.

For the loss of my model, I am largely inspired by the work [8]. Specifically, I define the loss as the summation of three parts: depth errors, gradients of depth, and normal errors. Assume that the depth estimate is d_i and its ground truth is g_i. For the depth errors, I use the logarithm of depth errors:

$$l_{depth} = \frac{1}{n}\sum_{i=1}^{n}\log(\|d_i - g_i\|_1 + \alpha)$$

The α is a constant. I use the logarithm scale to put more importance on nearby points and less importance on distant points [16]. I use the l_1 norm but I could also use l_2 norm as shown in the work [15].

For the loss of the gradients of depth, I am inspired by the work [17] and define it as:

$$l_{gradient} = \frac{1}{n}\sum_{i=1}^{n}\nabla_x(\|d_i - g_i\|_1) + \nabla_y(\|d_i - g_i\|_1)$$

It has been shown in the work [8] that this loss is very efficient to remove errors around edges.

For the normal loss, I am inspired by the work [16] and define it as:

$$l_{normal} = \frac{1}{n}\sum_{i=1}^{n}(1 - \frac{< n_i^d, n_i^g >}{\sqrt{< n_i^d, n_i^d >}\sqrt{< n_i^g, n_i^g >}})$$

where $n_i^d = [-\nabla_x(d_i), -\nabla_y(d_i)]^T$, $n_i^g = [-\nabla_x(g_i), -\nabla_y(g_i)]^T$ are the surface normals of the estimated depth map and its ground truth. As shown by the previous work [8], this loss term measures the angle between two surface normals, and is very useful for detecting small depth structures.

4. Experiments and Results

I use the 8000 uniformly sampled rgb-depth pair of images as training samples to train all four models: without denoise and without MFF, without denoise and with MFF, with median filter denoise and with MFF, with bilateral filter and with MFF. For each model, I add a Gaussian noise level of σ=0.1 to the RGB image input. An example of figure with added noise is shown in figure 4. For all models, I use batch size 8 and learning rate 0.0001. I train for 5 epoches.

With each trained model, I evaluate the model with 150 unseen rgb-depth image pairs. I use the evaluation metrics mentioned in the problem statement to measure the accuracy of each image, and take the average among all 150 test samples. Each model's test result is shown in figure 5.

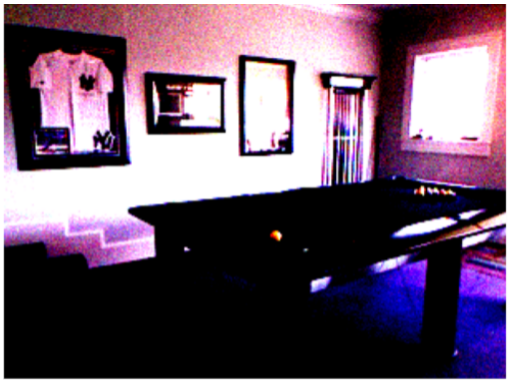


Figure 4. Input RGB image with noise level σ=0.1

4.1. Without denoise and without MFF

One sample output trained with this model, and the training loss are shown in figure 6 and 7. I observe that the training loss is steadily decreasing.

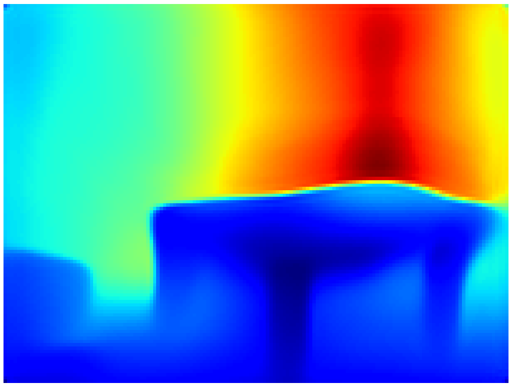


Figure 6. Predicted viridis depth map on noise level σ=0.1 without denoise without MFF trained with 8000 training samples for 5 epoches

4.2. Without denoise and with MFF

One sample output trained with this model, and the training loss are shown in figure 8 and 9. I observe that the training loss is steadily decreasing. Compared to the loss curve for the model without MFF, I could see that the loss for the model with MFF drops more quickly with smaller oscillation.

In comparison to the model without MFF in figure 6, the model with MFF is much better at capture small details. For example, the legs of the table could be clearly visualized in figure 8 while they are hard to distinguish from the background in figure 6. In addition, the door in figure 8 has better shapes and structure. However, compared with the ground truth in figure 2, I observe that there are still a lot more rooms for improvement. Although the shape of the table and space structure of the room are learned by the models, many small details are lost. For example, the window in figure 8 is almost lost which may be caused by the depth of the window being too close to the depth of the background wall.

From the quantitative test result, I could observe that the model with MFF outperforms the model without MFF in all metrics. This demonstrates that MFF positively improves the performance of the network by integrating information from all scales in the encoder.

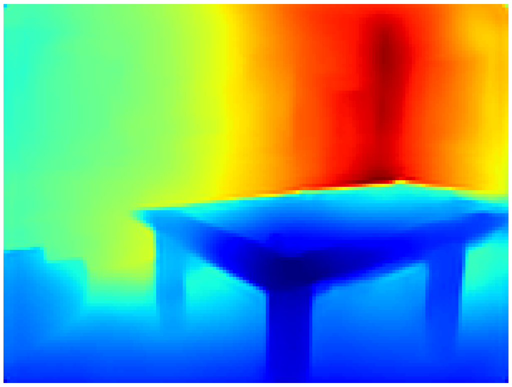


Figure 8. Predicted viridis depth map on noise level σ=0.1 without denoise with MFF trained with 8000 training samples for 5 epoches

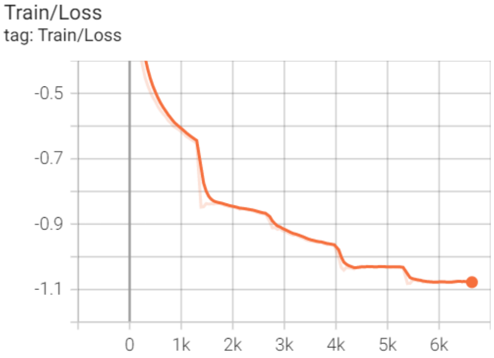
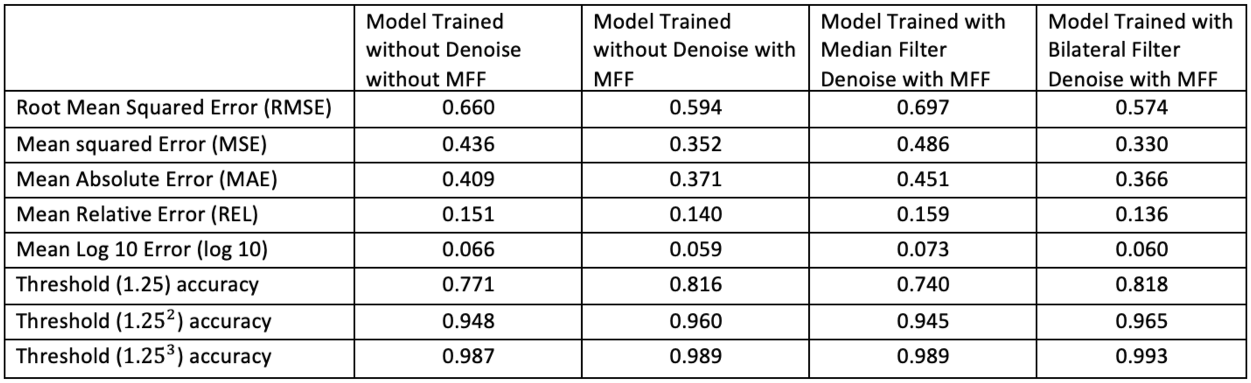


Figure 9. Training loss on noise level σ=0.1 without denoise with MFF trained with 8000 training samples for 5 epoches

4.3. With median filter denoise and with MFF

One sample output trained with this model, and the training loss are shown in figure 10 and 11. I observe that the



	Model Trained without Denoise without MFF	Model Trained without Denoise with MFF	Model Trained with Median Filter Denoise with MFF	Model Trained with Bilateral Filter Denoise with MFF
Root Mean Squared Error (RMSE)	0.660	0.594	0.697	0.574
Mean squared Error (MSE)	0.436	0.352	0.486	0.330
Mean Absolute Error (MAE)	0.409	0.371	0.451	0.366
Mean Relative Error (REL)	0.151	0.140	0.159	0.136
Mean Log 10 Error (log 10)	0.066	0.059	0.073	0.060
Threshold (1.25) accuracy	0.771	0.816	0.740	0.818
Threshold (1.25^2) accuracy	0.948	0.960	0.945	0.965
Threshold (1.25^3) accuracy	0.987	0.989	0.989	0.993

Figure 5. Trained Model Test Result

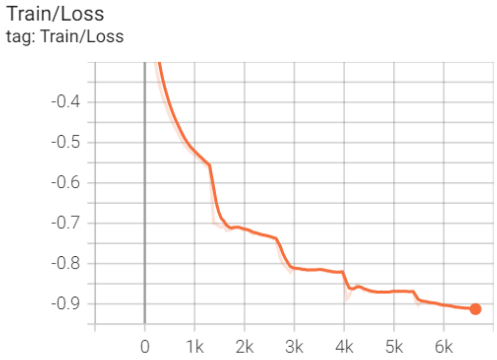


Figure 7. Training loss on noise level σ=0.1 without denoise without MFF trained with 8000 training samples for 5 epoches

training loss is steadily decreasing.

From the predicted depth map I observe that the model is still unable to capture the shape of the window on the top right corner. The model is also unable to capture the correct depth estimation of the table. Compared with the ground truth and the model trained without denoise with MFF, I observe that the back left and back right part of table are supposed to be both lighter blue, indicating it is further into the image plane. However, in this model's estimation, the back left part of the table is darker blue, indicating it is closer to the image plane, which is inaccurate. In summary, I could obviously observe that the model with median filter has worse depth map than the models without denoise.

From the quantitative test result, this model achieves the worst performance among all the models. One reason may be that the median filter smooth out the edges in the image, so it increases the level of difficulty to recover depth estimation.

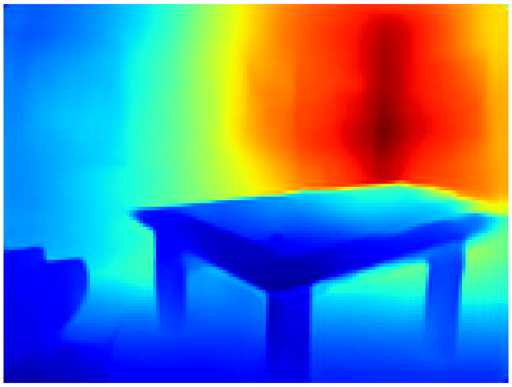


Figure 10. Predicted viridis depth map on noise level σ=0.1 with median denoise filter with MFF trained with 8000 training samples for 5 epoches

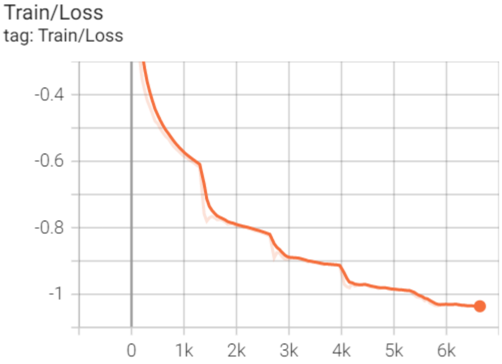


Figure 11. Training loss on noise level σ=0.1 with median filter denoise with MFF trained with 8000 training samples for 5 epoches

4.4. With bilateral filter denoise and with MFF

One sample output trained with this model, and the training loss are shown in figure 12 and 13. I observe that the training loss is steadily decreasing. Among all of model tested, I could see that the loss for the model with bilateral filter has the largest drop rate with smallest oscillation.

From the predicted depth map I observe that the model is able to capture some contour of the window on the top right corner. The model is also able to capture the depth information of the table, and the contour of the small sofa.

From the quantitative test result, this model achieves the best performance among all the models. One reason bilateral filter outperforms the median filter may be the bilateral filter not only depends on the distance among pixels but also the intensity among pixels, and therefore is better at preserving edges.

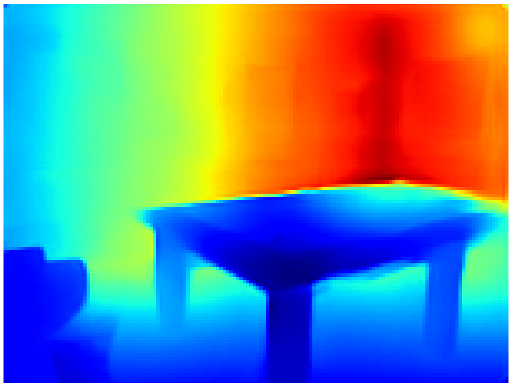


Figure 12. Predicted viridis depth map on noise level σ=0.1 with bilateral denoise with MFF trained with 8000 training samples for 5 epoches

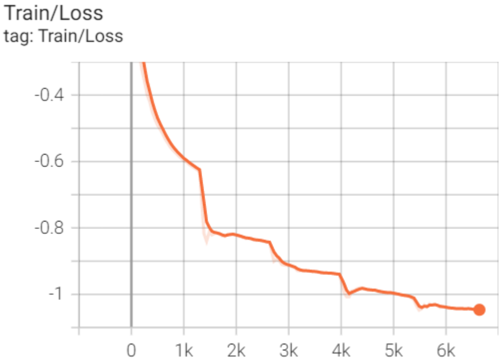


Figure 13. Training loss on noise level σ=0.1 with bilateral filter denoise with MFF trained with 8000 training samples for 5 epoches

5. Conclusion and Future Work

In this work, I explore the effect of 2 different denoise techniques on the single noisy image depth estimation using learning based approach. I also explore the effect of multi-feature fusion model for depth estimation. For the network, it consists of 4 parts: encoder, decoder, MMF and refinement layer. I observe that the bilateral filter is more suitable for depth estimation than the median filter since it is better at preserving edges in the image while the median filter tends to smooth out the edges. I also observe that the multi-feature fusion model could help the model to train and learn better by integrating information from multiple scales.

There are several limitation in this project. First of all, I assume that the noise is Gaussian distributed which may not hold true in real applications. In many applications, I may not have any prior knowledge about the distribution of the noise. Moreover, although the denoise filter may produce an image that visually looks better, it does smooth out small details or even objects in the image which could not be recovered by the depth estimation network. Although the final depth map may have good values on the evaluation metrics, the lost object could has very serious impact in real world application like autonomous driving. In addition, it takes longer time to train the model once the denoise filter becomes more computationally heavy.

For the future work, I want to explore more advanced denoise techniques such as the family of non-local mean filters like BM3D [3]. However, one potential challenge of those advanced denoising techniques is that they are much more computationally expensive than the median filter and bilateral filter. It is also worth to explore blurry images and the effect of deblurring for depth estimation. In addition, I also want to explore the neural network based denoising techniques such as DnCNN [18].

In this work, I assumed using a single image with no prior knowledge of the previous and the next frame. For future work,I would like to explore self-supervised learning [19] method that relies on using sequential frames and structure of motion to find depth estimation of each frame.

In summary, I observe that the multi-feature fusion module significantly improves the quality of depth map. The bilateral filter could improve the performance of depth estimation while median filter has negative effect. The bilateral filter denoise with multi-feature fusion module achieves the best performance for single noisy image depth estimation.

References

[1] S. Anwar, Z. Hayder, and F. Porikli. Deblur and deep depth from single defocus image. *Machine vision and applications*, 32(1):1–13, 2021. 1

[2] D. R. Brownrigg. The weighted median filter. *Communications of the ACM*, 27(8):807–818, 1984. 2

[3] A. Danielyan, V. Katkovnik, and K. Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on image processing*, 21(4):1715–1728, 2011. 6

[4] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 1

[5] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency, 2017. 1

[6] C. Godard, O. M. Aodha, M. Firman, and G. Brostow. Digging into self-supervised monocular depth estimation, 2019. 1

[7] V. Guizilini, R. Ambrus, W. Burgard, and A. Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11078–11088, 2021. 2

[8] J. Hu, M. Ozay, Y. Zhang, and T. Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 2, 3

[9] M. M. Johari, C. Carta, and F. Fleuret. Depthinspace: Exploitation and fusion of multiple video frames for structured-light depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6039–6048, 2021. 1

[10] Y. Lu and G. Lu. An alternative of lidar in nighttime: Unsupervised depth estimation based on single thermal image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3833–3843, 2021. 1

[11] Z. Ren, J. Meng, and J. Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction, 2011. 1

[12] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[13] S. Targ, D. Almeida, and K. Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016. 3

[14] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998. 2

[15] W. Williem and I. K. Park. Robust light field depth estimation for noisy scene with occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4396–4404, 2016. 1, 3

[16] X. Xu, Z. Chen, and F. Yin. Monocular depth estimation with multi-scale feature fusion. *IEEE Signal Processing Letters*, 28:678–682, 2021. 2, 3

[17] J. Yang, Q. Zhang, and Z. Cao. Multi-attribute statistics histograms for accurate and robust pairwise registration of range images. *Neurocomputing*, 251:54–67, 2017. 3

[18] Y. Zheng*, Y. Yuan, and X. Si. The improved dncnn for linear noise attenuation. In *SEG 2019 Workshop: Mathematical Geophysics: Traditional vs Learning, Beijing, China, 5-7 November 2019*, pages 56–59. Society of Exploration Geophysicists, 2020. 6

[19] H. Zhou, D. Greenwood, and S. Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021. 6

[20] S. Zia, B. Yüksel, D. Yüret, and Y. Yemez. Rgb-d object recognition using deep convolutional neural networks, 2017. 1