NA-MVSNet: A Normal-Aware MVSNet with View-Consistency for Depth Estimation

Chenhao Yang, Dijing Zhang*

Carnegie Mellon University

Abstract

Accurate stereo depth estimation plays a critical role in various 3D tasks in both indoor and outdoor environments. In this project, we introduce a deep multi-view stereo (MVS) system that couples the depth estimation module and surface normal esitmation module, which we call NA-MVSNet. The main idea of our approach is to restore the depths and normal of source image sets from estimation of reference image. Specifically, the algorithm differentiable transforms the estimation of reference view to source views given the known intrinsic and extrinsic matrix and based on these, we hope to maintain the view consistency along every input instead of just compare the difference of reference image. What's more, we compute the error between predicted depth, normal and their respective ground truth. Consistency loss between depth and normal are also taken into consideration. The whole system can be trained end-to-end, tackling the challenging problem of matching pixels within textureless regions. Experimental results on ETH3D demonstrate the better performance over the results of MVSNet. Code is avavilable at www.github.com/16889-team/Normal-Aware-MVSNet

*Authors' names listed in alphabetical order

1 Introduction

Within recent years, depth map has shown wide application in many areas such as autonomous driving, robotics and virtual/augmented reality. However, existing depth sensors, including LiDARs, structured-light-based depth sensors, and stereo cameras, all have their own limitations. For instance, depth map captured are usually sparse because of hardware or power limitations, so that delicate details are abandoned in exchange for computational efficiency [8], [16], [3]. An alternative pipeline for depth inference is using stereo matching by computing correlation or sum of squared differences(SSD) from paired images [4], [5] and generate disparity map. Meanwhile, deep learning has also been widely proven to be highly efficient to solve depth inference using Multi-view stereo (MVS) [13][14][1][2]. However, depth inference still remain a challenge for both classical and learning-based method under non-ideal Lambertian scenarios such as low-textured, specular and reflective regions of the scene make dense matching intractable and thus lead to incomplete estimations. Within the scope of this project, we are focusing on depth inference for scenes including common low-textured, even textureless surfaces (walls, ceilings, floors, etc.) using deep learning method. Meanwhile, estimating surface normal may well be one idea to improve the results since normal is related to geomoetric information instead of texture information.

Inspired by the work [6][15], we proposed a normal-aware MVSNet, named as NA-MVSNet. The algorithm jointly esitamte the normal maps and depth maps of reference and then differentiable transform the view from reference to source by the given transformation information. Then apply the loss between ground truth and estimation of depth and normal. Depth-normal consistency is also counted by comparing the results from depth module and normal module. The proposed system combines the benefits of state-of-art depth module and take normal maps into consideration, which predict some geometric information. Besides, the information from source image sets are also merged.

2 Related Work

Multi-view stereo (MVS) algorithms are able to reconstruct 3D models from images under the assumptions of known materials, view points and light conditions[10]. Traditional methods using hand-crafted features have achieved impressive results while tend to fail for thin objects, non-diffuse surfaces or the objects with insufficient features. So, recently some learning-based methods achieve compelling results in depth estimation. Yao, et al.[13] proposed end-to-end learning algorithm, MVSNet, by computing 3D cost volume via differential homography warping and 3D CNN. It became the SOTA at many datasets in 2018 and lead the spring-up of plenty of variants. In order to reduce the amount of demanded memory, the team replaced 3D CNNs with RNN and it improves the scalability of MVS methods.[14] Coarse-to-fine pattern is another solution of reducing massive memory consumption.[2] Coarse depth map is predicted and finer results are based on coarse ones. Cas-MVSNet is one of the example, which regenerates cost volumes by warping feature maps with a smaller depth range around previous coarse prediction. Rui, et al. [1] propsed a novel point-base deep framewrok for MVS. Point-MVSNet directly processes the target scene as point clouds while others focus on disparity maps. It leverages 3D geometry priors and 2D texture information jointly and effectively by fusing them into a feature-augmented point cloud. However, a very typical problem for learning-based MVS algorithms is lack of valid depth values at low-texture regions and that is the reason why SOTA methods of ETH3D are still non-learning ones. Some attempts use different a priori information, such as surface normal [6][7] to overcome planar areas. But these attempts highly rely on post-process and are still far from end-to-end.

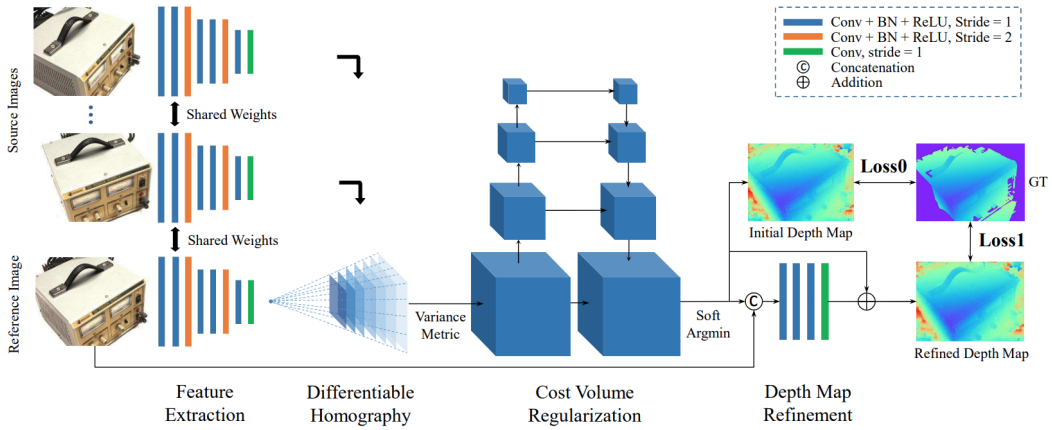


Figure 1: The network design of MVSNet.

3 Benchmark

Our benchmark is the Multi-View Stereo Net (MVSNet) [13]. Having one reference image and multiple 2D source images as the input, MVSNet goes through feature extraction, differentiable homography warping to generate cost volume by the given camera information. The final depth map output is regressed from the regularized probability volume and refined with the reference image. MVSNet applies eight convolutional layers to get N (number of input images) feature maps with 32 channels. The cost volume stores the loss of different views, in order to generate the probability map. Then with the loss between the initial depth map and ground truth, it estimates and refines the depth map. The visualization of MVSNet architecture is shown below in Fig. 1.

ETH3D is a comprehensive benchmark for both SLAM [11] and stereo [12] tasks. Considering MVS, it contains 25 high-resolution scenes and 10 low-resolution scenes. ETH3D is widely acknowledged as the most difficult MVS task since it contains many low-textured regions such as white walls and reflective floor. Traditional MVS methods based on broadcasting valid depth values perform better in this case.

4 Approach

We propose an end-to-end pipeline for multi-view depth and normal estimation as shown in Fig. 2. The entire pipeline can be viewed as three modules. The first one consists of joint estimation of depth and normal maps from the cost volume built from multi-view image features. The second one refine the predicted depth by enforcing consistency between the predicted depth and normal maps. The last one differentiable transform the view from reference to source and apply the same loss to the transformed results.

4.1 Data Pre-Processing

4.1.1 Paired Images

As mentioned in 3, ETH3D dataset contains depth and multi-view pictures of various scenes. Our model requires additional relationships of paired reference and source images. Original author of MVSNet provided paired reference and source images by computing view selection score according to the sparse points[13].

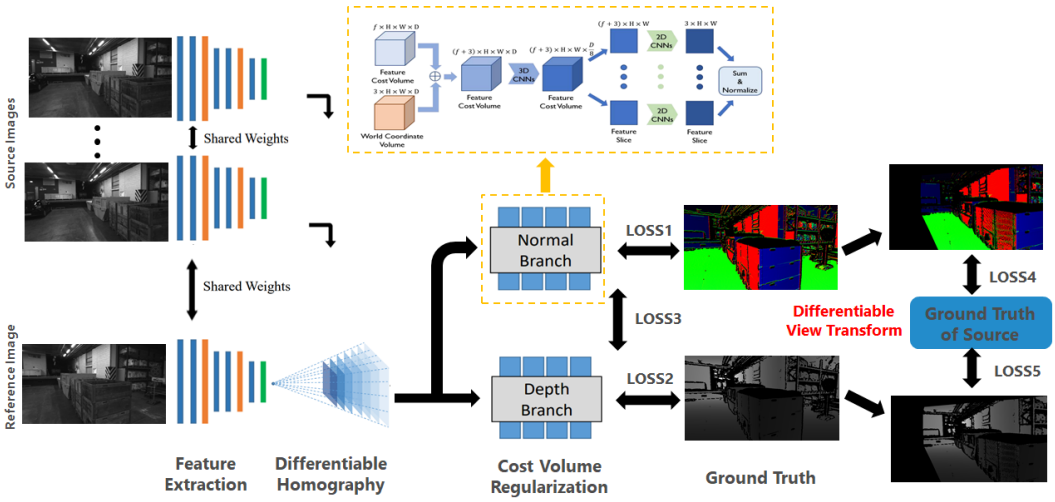
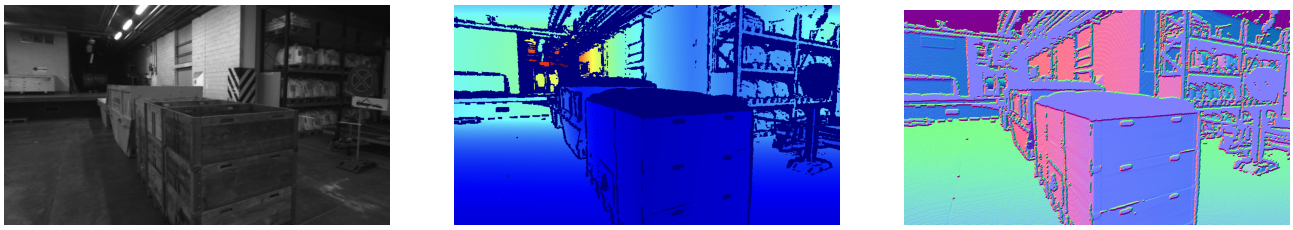


Figure 2: **Illustration of the pipeline of our method.** We first extract deep image features from viewed images and build a feature cost volume by feature wrapping. The depth and normal are jointly learned in a supervised fashion. Further we differentiably transform views from reference to source and apply a consistency loss.



(a) Gray Image (b) Depth Map (c) Normal Map

Figure 3: **Illustration of dataset.** Normal maps are generated from depth map gradients. The scene is delivery area in ETH3D dataset. [12]

4.1.2 Normal Map Generation

In addition to depth map, we utilized normal maps for supervised training in our model. Therefore, we pre-generated normal maps from depth image gradients. The method can be described as using adjacent pixels and compute the normal at each pixel on a depth image by following three steps. First, image gradients on a depth image are computed with a 2D differential filtering. Next, two 3D gradient vectors are computed from horizontal and vertical depth image gradients. Finally, the normal vector is obtained from the cross product of the 3D gradient vectors.[9] A sample result can be found in Fig. 3.

4.2 MVSNet based Depth Estimation

This section describes the depth estimation module. It is highly based on the original MVSNet[13]. The design of MVSNet strongly follows the rules of camera geometry. The first step of MVSNet is to extract the deep features $[F_i]_{i=1}^N$ using shared weights and then build a 3D cost volume from the extracted feature maps and input cameras. For simplicity, in the following we denote I_1 as the reference image, $[I_i]_{i=2}^N$ as the source images, and K_i, R_i, t_i as the camera intrinsics, rotations and translations that correspond to the feature maps.

Differentiable Homorgraphy. All feature maps are warped into different fronto-parallel palnes of the reference camera to form N feature volume. The coordinate mapping from the warped feature map to F_i at depth d is determined by the planar transformation $x' \sim H_i(d) \cdot x$, where '$\sim$' denotes the projective equality and $H_i(d)$ the homography between the i^th feature

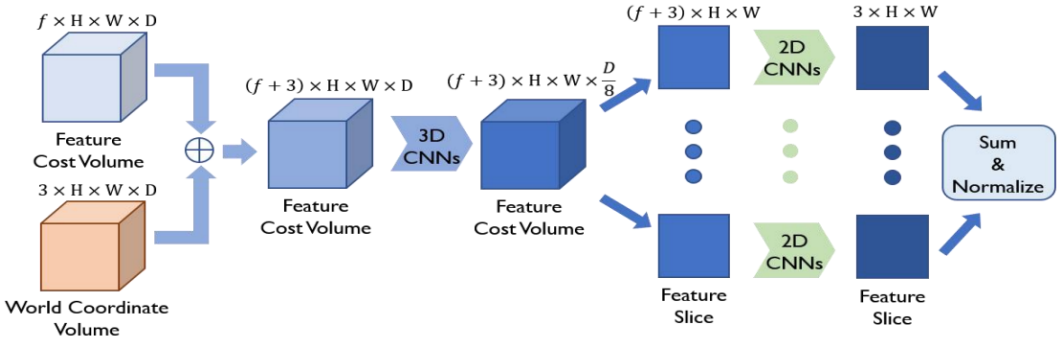


Figure 4: **Details of normal estimation module**

map and the reference feature map at depth d. Let n_1 be the principle axis of the reference camera, the homography is expressed by a 3×3 matrix:

$$H_i(d) = K_i \cdot R_i \cdot (I - \frac{(t_1 - t_i) \cdot n_1^T}{d}) \cdot R_1^T \cdot K_1^{-1} \tag{1}$$

As the core step to bridge the 2D feature extraction and the 3D regularization netwokrs, the warping operation is implemented in differentiable manner, which enables end-to-end training of depth map inference.

Cost Metric Next, we aggregate multiple feature volume to one cost volume C. To adapth arbitrary number of input views, we propose a variance-based cost metric $\mathcal{M}$ for N-view similarity measurement. Let N, H, D, F be the input image width, height, depth sample number and the channel number of the feature map, the $V = \frac{W}{4} \cdot \frac{H}{4} \cdot D \cdot F$ the feature volume size, our cost metric defines the mapping $\mathbf{M}$ that:

$$\boldsymbol{C} = \boldsymbol{\mathcal{M}}(\boldsymbol{V_1}, ..., \boldsymbol{V_N}) = \frac{\sum_{i=1}^{N} (\boldsymbol{V_i} - \bar{\boldsymbol{V}}_{\boldsymbol{i}})^2}{N} \tag{2}$$

Where $\bar{\boldsymbol{V}}_{\boldsymbol{i}}$ is the average volume among all feature volumes, and all operations above are element-wise.

4.3 Cost Volume based Surface Normal Estimation

This section describes the module of cost volume based surface normal estimation as shown in Fig. 4. The cost volume contains all the spatial information in the scene as well as image features in it. This motivates us to use the cost volume C_1 which also contains the image-level features to estimate the surface normal map of the underlying scene.

Given the cost volume C_1 we concatenate the world coordinates of every voxel to its features. Then use three layers of 2-stride convolution along the depth channel to reduce the size to $((f+3) \times H \times W \times D \times \frac{D}{8})$ and call this C_n. We pass each slice through a normal-estimation network and add the output of all slices and normalize the sum to obtain the estimate of the normal map.

4.4 Geometric based View Transformation

In this section, we will describe the view transformation from reference view to source view, which aims at restoring the depth and normal maps for source image sets at the same time. The core idea is to find the corresponding pixel between reference image I_1 and source images $[I_i]_{i=2}^N$.

Assuming the reference pixel is P_1 and the source pixel is $[P_i]_{i=2}^N$. Taking T_1 as projection matrix for reference and $[T_i]_{i=2}^N$ for source. The view transformation can be expressed as:

$$P_i = \mathcal{N}(T_i \cdot (T_1^{-1} \cdot P_1)) \tag{3}$$

Where $\mathcal{N}$ denotes the normalization procedure, which includes the depth normalization and size normalization.

After given the flow-field grid and the estimation results from respectively branches, we apply bi-linear interpolation to get the output, which is implemented in differentiable manner and enable back-propagation. The pixels out of the range will be counted into the mask so that we can erase these ineffective information.

The view transformation with normal maps will be little different since surface normal will change due to rotation between reference camera and source cameras. Taking N_1 as normal for reference and $[N_i]_{i=2}^N$ for source and the procedure can be expressed as:

$$N_i = \mathcal{N}'(R_i \cdot (R_1^{-1} \cdot N_1)) \tag{4}$$

Where $\mathcal{N}'$ is normalization along normal direction to make it unit.

4.5 Depth-Normal Consistency

In addition to estimating depth and normal jointly, we also use a consistency loss [6] directly to enforce consistency between estimated normal and depth maps. We utilize the camera model to estimate the spatial gradeint of the depth map in the pixel coordinate space using the depth and normal map. We compute two estimates for $(\frac{\delta Z}{\delta u}, \frac{\delta Z}{\delta v})$ and enforce them to be consistent.

From the camera model, we can yield:

$$\begin{aligned} X = \frac{Z(u-u_c)}{f_x} &\longrightarrow \frac{\delta X}{\delta u} = \frac{u-u_c}{f_x}\frac{\delta Z}{\delta u} + \frac{Z}{f_x} \\ Y = \frac{Z(v-v_c)}{f_y} &\longrightarrow \frac{\delta Y}{\delta u} = \frac{v-v_c}{f_y}\frac{\delta Z}{\delta x} \end{aligned} \tag{5}$$

Estimate 1: The spatial gradient of the depth map can first be computed from the depth map by a Sobel filter:

$$(\frac{\delta Z}{\delta u}, \frac{\delta Z}{\delta u})_1 = (\frac{\Delta Z}{\Delta u}, \frac{\Delta Z}{\Delta v}) \tag{6}$$

Estimate 2: Assume the underlying scene to be of a smooth surface which can be expressed as an implicit function $F(X, Y, Z) = 0$. The normal map $\vec{n}$ is an estimate of the gradient of this surface.

$$\vec{n} = (n_x, n_y, n_z) = (\frac{\delta F}{\delta X}, \frac{\delta F}{\delta Y}, \frac{\delta F}{\delta Z}) \longrightarrow \frac{\delta Z}{\delta X} = \frac{-n_x}{n_z}, \frac{\delta Z}{\delta Y} = \frac{-n_y}{n_z} \tag{7}$$

Thus, we can derive the second estimate of the depth spatial gradient by:

$$(\frac{\delta Z}{\delta u}, \frac{\delta Z}{\delta u})_2 = (\frac{\delta Z}{\delta X}\frac{\delta X}{\delta u} + \frac{\delta Z}{\delta Y}\frac{\delta Y}{\delta u}, \frac{\delta Z}{\delta X}\frac{\delta X}{\delta v} + \frac{\delta Z}{\delta Y}\frac{\delta Y}{\delta v}) \tag{8}$$

4.6 Loss

This section will conclude all the loss we apply for our pipeline. Original (reference) loss and view-consistency (source) loss are taken into consideration, for depth and normal. Depth-Normal consistency loss is added to loss function. The overall loss is expressed as:

$$\mathcal{L} = \mathcal{L}_{rd} + \beta \cdot \mathcal{L}_{rn} + \alpha \cdot \mathcal{L}_{vc} + \gamma \cdot \mathcal{L}_c \tag{9}$$

Where α, β, γ are the hyper-parameters which can balance the proportion of separate loss. $\mathcal{L}_c$ is given as the Huber norm of the deviation between two gradient estimates and $\mathcal{L}_{vc}$ is given as the view consistency loss, which also denotes as $L_{sd} + \beta \cdot \mathcal{L}_{sn}$. L_d and L_n are the Huber norm of the depth and normal, known as smooth L1 loss.

$$
\begin{aligned}
\mathcal{L}_c &= |(\frac{\delta Z}{\delta u}, \frac{\delta Z}{\delta u})_1 - (\frac{\delta Z}{\delta u}, \frac{\delta Z}{\delta u})_2|_{\mathbf{H}} \\
L_{rd} &= |D_{1pred} - D_{1gt}|_{\mathbf{H}} \\
L_{rn} &= |N_{1pred} - N_{1gt}|_{\mathbf{H}} \\
L_{rd} &= \sum_{i=2}^{N} |D_{ipred} - D_{igt}|_{\mathbf{H}} \\
L_{rn} &= \sum_{i=2}^{N} |N_{ipred} - N_{igt}|_{\mathbf{H}}
\end{aligned}
\tag{10}
$$

Where D_1 and $[D_i]_{i=2}^N$ are the depth estimation for reference and source images while N_1 and $[N_i]_{i=2}^N$ are normal estimation.

5 Experiments

5.1 Datasets

We use ETH3D[12] dataset for training our end-to-end pipeline from scratch. To save the computer resource, we utilize low-resolution image dataset whose size is 225 × 125. View number N is set to be 4, which contains one reference pair and three source pairs. Training dataset contains 1588 image pairs from three scenes *(delivery area, electro, forest)*. We evaluate the model on 810 image pairs from two scenes *(playground, terrains)*. Except the experiment on our method, we set MVSNet [13] as our baseline and mainly make comparison and report the common quantitative measures of depth quality: absolute difference error (Abs diff) and metric threshold error below three different measurement $(2mm, 4mm, 8mm)$.

5.2 Implementation Details

We use 64 levels of depth/disparity while building the cost volume. The hyperparameters α, β, γ are of vital importance since the loss are needed to balance.To save the computer sources, we set the batch size to be 16 and training the pipeline end-to-end for only for 16 epochs with a learning rate of 1×10^{-3}. To find out which parameter setting is the best, we do ablation study by setting other two hyperparameters to be zero to find out the best value for α, β, γ. The study result is shown in Table. 1 and Fig. 5.

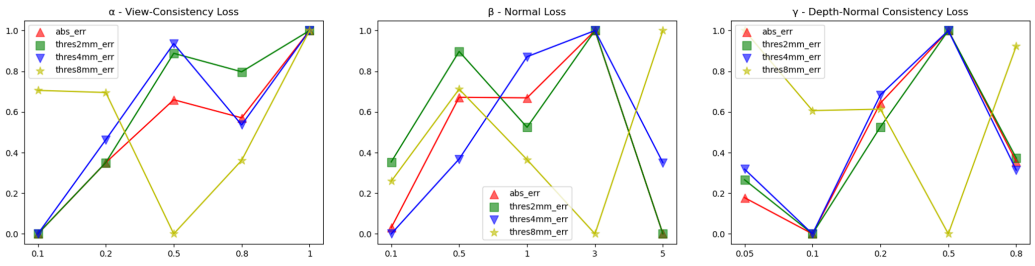


Figure 5: **Plots of ablation study on each hyperparameter.** X-axis is parameter value and Y-axis is normalized result $\frac{Y - Y_{MIN}}{Y_{MAX} - Y_{MIN}}$

α	**abs err**	**thres2mm err**	**thres4mm err**	**thres8mm err**
0.1	1.1439336	0.1595291	0.0587183	0.0183237
0.2	1.2232659	0.1802036	0.0676603	0.0182831
0.5	1.2931906	0.2118712	0.0767898	0.0156022
0.8	1.2730515	0.2065175	0.0690675	0.0195440
1	1.3701903	0.2184631	0.0780311	0.0194583
β	**abs err**	**thres2mm err**	**thres4mm err**	**thres8mm err**
0.1	1.2126009	0.1753789	0.0650039	0.0199515
0.5	1.2302525	0.1796702	0.0673513	0.0206652
1	1.2301798	0.1767233	0.0705596	0.0201190
3	1.2392914	0.1804805	0.0713791	0.0195440
5	1.2117639	0.1725851	0.0672355	0.0211191
γ	**abs err**	**thres2mm err**	**thres4mm err**	**thres8mm err**
.05	1.2220918	0.1751215	0.0663291	0.0206981
0.1	1.2064987	0.1673891	0.0611906	0.0197326
0.2	1.2627815	0.1826546	0.722244	0.0197496
0.5	1.2940516	0.1964299	0.0773323	0.0182451
0.8	1.2377207	0.1782329	0.0662541	0.0205105

Table 1: **Statistical result of ablation study on each hyperparameter.** α: View Consistency, β: Normal, γ: Dpeth-Normal Consistency.

As we can conclude from the ablations, $\alpha = 0.1$, $\beta = 0.1$ and $\gamma = 0.1/0.5$ are the optimal value setting. The experiment with $\alpha = 0.1$ gives us a extremely great result, compared with MVSNet, which indicates our view consistency idea is on the right way.

5.3 Comparison with MVSNet

For comparison with the performance of the baseline (MVSNet) and our method on ETH3D, we set same training parameters for MVSNet as our method and set the optimal hyperparameter setting for NA-MVSNet. Given the reason that our depth estimation module is inherited from MVSNet, so when we set all three parameters to be zero, the metric result should be the same. After tens of ablabtion study, we found $\alpha = 0.1, \beta = 0.1, \gamma = 0.5$ gives the optimal result. Comparison is expressed in Table. 2

Model	**abs err**	**thres2mm err**	**thres4mm err**	**thres8mm err**
Baseline	1.23625	0.17813	0.06987	0.01935
Only View-Consistency	1.14393	0.15953	0.05872	0.01832
Ours(optimal)	1.22034	0.17642	0.06174	0.01585

Table 2: **Comparative evaluation on validation dataset between MVSNet and our method.** View-Consistency setting is $\alpha = 0.1, \beta = 0, \gamma = 0$ and Optimal setting is $\alpha = 0.1, \beta = 0.1, \gamma = 0.5$

According to the quantitative results, best result increases around 7.5%, 10.4%, 16.0% and 18.0% respectively in four different metrics. We can find that our view-consistency module does a great job in predicting depth maps, which tremendously increase the first three metrics.

But when it comes to other two modules, like normal estimation and depth-normal consistency module, it preforms a little worse than our method with only view-consistency, except it increases around 18.0% in thres8mm error, which is impressive. We believe it has something to do with the poor performance of our normal estimation module.

5.4 Visualization

In this section, we will provide the visualization results. Comparative results between MVSNet and our method are shown in Fig. 6

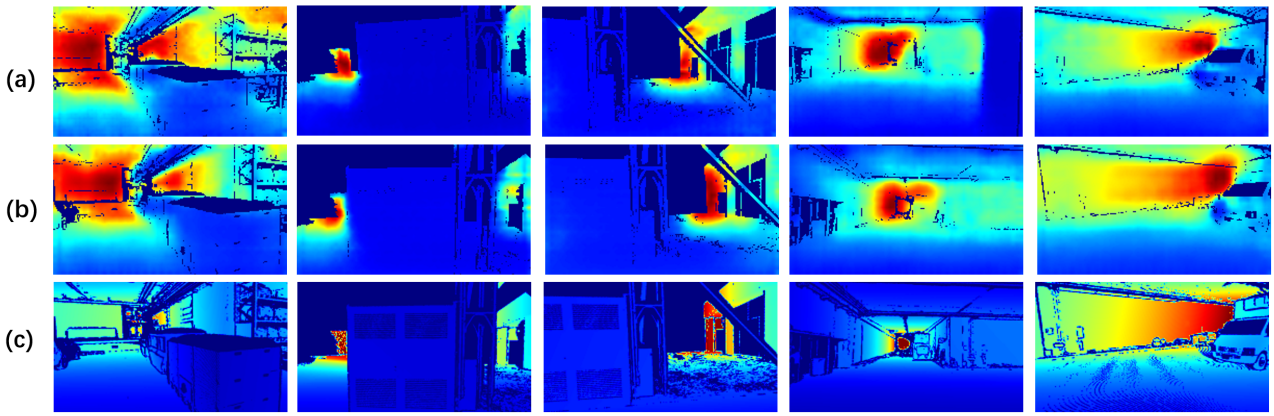


Figure 6: **Visualization of depth maps.** (a):baseline (b):ours (c):ground truth

Conclusion can be made that our method yield fine results for depth estimation, and for some textureless regions, like walls, pillars, it can also be tackled slightly better. However for both our model and MVSNet there are the lack of details in trade for smoothness. We also noticed that they did not yield as good results for estimating objects at far distances as those at near distances. It can be explained by the principles of perspective, as there are reduction of the projected size on image plane for objects in distanced place, therefore it's harder to estimate the depth.

6 Conclusions

In this project, we proposed a method to improve MVSNet for depth map inference. We merged normal information into depth estimation pipeline and applied geometric-based transformation from reference to source to maintain view consistency. The geometry constraints between surface normal and depth at training time were also considered to improve the stereo depth estimation. Experimental results showed that our NA-MVSNet performs better than MVSNet in depth estimation, especially in quantitative evaluation metric. We also presented visualization results of the depth predicted by our model and MVSNet, it showed slight improvement from our method. We jointly predicted the depth and normal based on multi-view cost volume and transform view from reference to source. The module designed for normal estimation can help us to better predict depth at textureless regions, where lacks of feature, since normal map contains geometry information that depth map doesn't have. The view consistency can merge much more information at one time than MVSNet itself, which ignores the ground truth information from source data.

References

[1] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1538–1547, 2019.

[2] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020.

[3] Mohit Gupta, Qi Yin, and Shree K. Nayar. Structured light in sunlight. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[4] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE transactions on pattern analysis and machine intelligence*, 16(9):920–932, 1994.

[5] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 508–515. IEEE, 2001.

[6] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2189–2199, 2020.

[7] Hongmin Liu, Xincheng Tang, and Shuhan Shen. Depth-map completion for large indoor scene reconstruction. *Pattern Recognition*, 99:107112, 2020.

[8] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018.

[9] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rin-Ichiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *2015 International Conference on 3D Vision*, pages 640–647, 2015.

[10] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[11] T. Schöps, T. Sattler, and M. Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. CVPR, 2019.

[12] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. CVPR, 2017.

[13] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *arXiv preprint arXiv:1804.02505*, 2018.

[14] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

[15] Wang Zhao, Shaohui Liu, Yi Wei, Hengkai Guo, and Yong-Jin Liu. A confidence-based iterative solver of depths and surface normals for deep multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6168–6177, 2021.

[16] Qingtian Zhu, Chen Min, Zizhuang Wei, Yisong Chen, and Guoping Wang. Deep learning for multi-view stereo via plane sweep: A survey. *arXiv preprint arXiv:2106.15328*, 2021.