

SUPPLEMENTARY MATERIAL: MULTIPLANE NeRF-SUPERVISED DISENTANGLEMENT OF DEPTH AND CAMERA POSE FROM VIDEOS

Anonymous authors

Paper under double-blind review

A APPENDIX

B NETWORK STRUCTURES

We show the details of network structures in Table 1, Table 2, and Table 3, including the camera encoder $\mathcal{F}_{\text{traj}}$, the depth encoder \mathcal{F}_{dep} , and the Multiplane NeRF. More specifically,

- **Camera encoder** ($\mathcal{F}_{\text{traj}}$): given a pair of frames as input, we first use the ResNet50 (He et al., 2016) to extract the RGB feature, which is modified to accept a pair of frames (6-channel input), then we use several convolutional layers to predict the camera pose. Note that we represent the camera pose by axis-angle, hence the output is a 6-channel vector.
- **Depth encoder** (\mathcal{F}_{dep}): given the raw RGB image, we instead use the MnasNet (Tan et al., 2019) followed with a FPN (Lin et al., 2017) to obtain the multi-stage features, then the U-Net (Ronneberger et al., 2015) like structure with skip-connections is utilized to predict the monocular depth map at different resolution scales.
- **Multiplane NeRF** (\mathcal{F}_{mpi}): as described in the method section, the Multiplane NeRF is construct upon the raw RGB image and a position embedding of a specific disparity value d_i . Given the shared image feature from MnasNet (Tan et al., 2019) and FPN (Lin et al., 2017), it first concatenates together with the positional embedding and then feed into the similar U-Net (Ronneberger et al., 2015) structure used in depth encoder, except that we add two additional downsampling blocks and two upsampling blocks. The output is the 4-channel image with RGB color c and the density value σ .
- **Multiplane NeRF rendering**: Multiplane NeRF is a continuous depth generalization of the MPIs by introducing the neural radiance fields. Formally, the image is represented by $\{(c_i, \sigma_i)\}_{i=1}^D$, where σ_i is the volume density of the i -th plane. Unlike the vanilla NeRF Mildenhall et al. (2020), it represents a camera frustum using planes instead of rays. Then, we follow the naive setting of rendering mechanism used in NeRF Mildenhall et al. (2020) to obtain the image and the disparity map under the source view,

$$\hat{\mathbf{I}}_s = \sum_{i=1}^D T_i (1 - \exp(-\sigma_i \delta_i)) c_i \quad \hat{\mathbf{D}}_s = \sum_{i=1}^D T_i (1 - \exp(-\sigma_i \delta_i)) d_i \quad (1)$$

where $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ denotes the probability of a ray travels from the first plane to i -th plane without hitting any object and the δ_i is the distance map between the i -th plane and $i+1$ -th plane.

C TRAINING & INFERENCE DETAILS

C.1 TRAINING DETAILS.

We adopt a multi-scale training strategy proposed in (Godard et al., 2019). More specifically, \mathcal{L}_{L1} and $\mathcal{L}_{\text{SSIM}}$ are applied on the output1 while the remaining term $\mathcal{L}_{\text{smooth}}$, $\mathcal{L}_{\text{consist}}$, and $\mathcal{L}_{\text{reproj}}$ are applied on output2, and output3. Since the monocular depth is used to compute the consistency loss $\mathcal{L}_{\text{consist}}$, we detach the monocular depth estimation part to stop the gradient flow from $\mathcal{L}_{\text{consist}}$ and the monocular depth estimation is only supervised by the reprojection loss $\mathcal{L}_{\text{reproj}}$.

Layer	k	s	c	input
Resnet50	–	–	2048	Concat($\mathbf{I}_s, \mathbf{I}_t$)
pconv0	1	1	256	econv5
pconv1	3	1	256	pconv0
pconv2	3	1	256	pconv1
pconv3	1	1	6	pconv2
avgpool	–	–	–	pconv3

Table 1: The camera encoder ($\mathcal{F}_{\text{traj}}$) architecture.

Layer	k	s	c	input
MnasNet+FPN	–	–	32	\mathbf{I}_s
upconv4_0	3	1	128	fconv4
upconv4_1	3	1	128	upconv4_0 \uparrow , fconv3
upconv3_0	3	1	64	upconv4_1
upconv3_1	3	1	64	upconv3_0 \uparrow , fconv2
disp3	3	1	1	upconv3_1
upconv2_0	3	1	32	upconv3_1
upconv2_1	3	1	32	upconv2_0 \uparrow , fconv1
disp2	3	1	1	upconv2_1
upconv1_0	3	1	16	upconv2_1
upconv1_1	3	1	16	upconv1_0 \uparrow
disp1	3	1	1	upconv1_1

Table 2: The depth encoder (\mathcal{F}_{dep}) architecture. The “ \uparrow ” is the upsampling operation.

C.1.1 INFERENCE DETAILS.

We evaluate our model on three different tasks: depth estimation, camera pose estimation, and novel view synthesis with different inference procedures. We describe the inference procedure for each task in details as following:

- Depth estimation: given a testing frame, instead of using the monocular depth estimation results, we utilize the Multiplane NeRF to obtain the depth map via rendering. Comparing with the monocular depth predictions, the rendered depth maps are always more smooth. To address the scale ambiguity issue, we adopt a scale alignment method by least squares optimization before evaluation.
- Camera pose estimation: given a short video clip *i.e.*, 30 frames, we take a pair of two frames as input sequentially. Each pair of frames is concatenated together and fed into the camera encoder $\mathcal{F}_{\text{traj}}$ to obtain the relative pose between two frames. Then, the camera trajectory can be constructed upon estimated relative poses. Next, both estimated camera trajectory and the ground-truth one are converted into the same coordinate with the same origin and the Absolute Trajectory Error (ATE) is evaluated via the public evo package (Grupp, 2017).
- Novel view synthesis: given a pair of two frames, *i.e.*, one is the source view image and the other is the target view image, we first compute the relative camera pose between two frames and then construct the Multiplane NeRF upon the source image and utilize the estimated camera transformation to obtain the RGB image under the target view. We follow two different test split released by VideoAE (Lai et al., 2021) and MINE (Li et al., 2021) and the interval between source and target view is set to 5.

Layer	k	s	c	input
MnasNet+FPN*	–	–	32	\mathbf{I}_s
downconv1	1	1	512	fconv4
downconv2	3	1	256	downconv1
upconv_0	3	1	256	downconv2
upconv_1	1	1	32	upconv_0
upconv4_0	3	1	128	upconv_1, $\text{PE}(d_i)$
upconv4_1	3	1	128	upconv4_0 \uparrow , fconv3, $\text{PE}(d_i)$
upconv3_0	3	1	64	upconv4_1
upconv3_1	3	1	64	upconv3_0 \uparrow , fconv2, $\text{PE}(d_i)$
output3	3	1	4	upconv3_1
upconv2_0	3	1	32	upconv3_1
upconv2_1	3	1	32	upconv2_0 \uparrow , fconv1, $\text{PE}(d_i)$
output2	3	1	4	upconv2_1
upconv1_0	3	1	16	upconv2_1
upconv1_1	3	1	16	upconv1_0 \uparrow
output1	3	1	4	upconv1_1

Table 3: Multiplane NeRF (\mathcal{F}_{mpi}) architecture. The MnasNet+FPN* is shared by both depth encoder and Multiplane NeRF.

Methods	<i>novel view synthesis</i>			<i>depth estimation</i>		
	PSNR \uparrow	SSIM \uparrow	Perc Sim \downarrow	Abs Rel \downarrow	Sq Rel \downarrow	RMSE \downarrow
Appearance Flow (Zhou et al., 2016)	14.8	0.48	3.13	–	–	–
SynSin (Wiles et al., 2020)	15.7	0.47	2.76	0.91	1.81	2.08
MINE (Li et al., 2021)	19.3	0.71	1.69	0.19	0.18	0.34
Ours	18.0	0.61	2.11	0.17	0.09	0.39

Table 4: Generalization ability of novel view synthesis task and depth estimation task. We pretrain our model on the RealEstate10K (Zhou et al., 2018) and evaluate on the 100 30-frames clips of ScanNet (Dai et al., 2017).

D MORE EXPERIMENTS

D.1 GENERALIZATION ABILITY

To show the generalization ability of our model, we utilize the model pretrained on RealEstate10K (Zhou et al., 2018) and evaluate the performance of novel view synthesis and depth estimation on ScanNet (Dai et al., 2017). As illustrated in Table 4, our model can achieve on par or even better results on both two tasks.

D.2 ADDITIONAL QUALITATIVE RESULTS

We highly recommend you to check the supplementary video which contains more video results.

Depth Estimation. More depth estimation visualizations on ScanNet (Dai et al., 2017) are shown in Fig. 1.

Camera Pose Estimation. We also plot the camera pose trajectory in Fig. 2. The ground-truth trajectory is marked by green color while the estimated one is marked by blue. Note that we first adopt the alignment before visualization.

Novel View Synthesis. We provide more qualitative results of novel view synthesis in Fig. 3. We present the input RGB image, synthesised RGB image under target view and the the ground-truth RGB image. The synthesised depth maps are also shown as the reference.

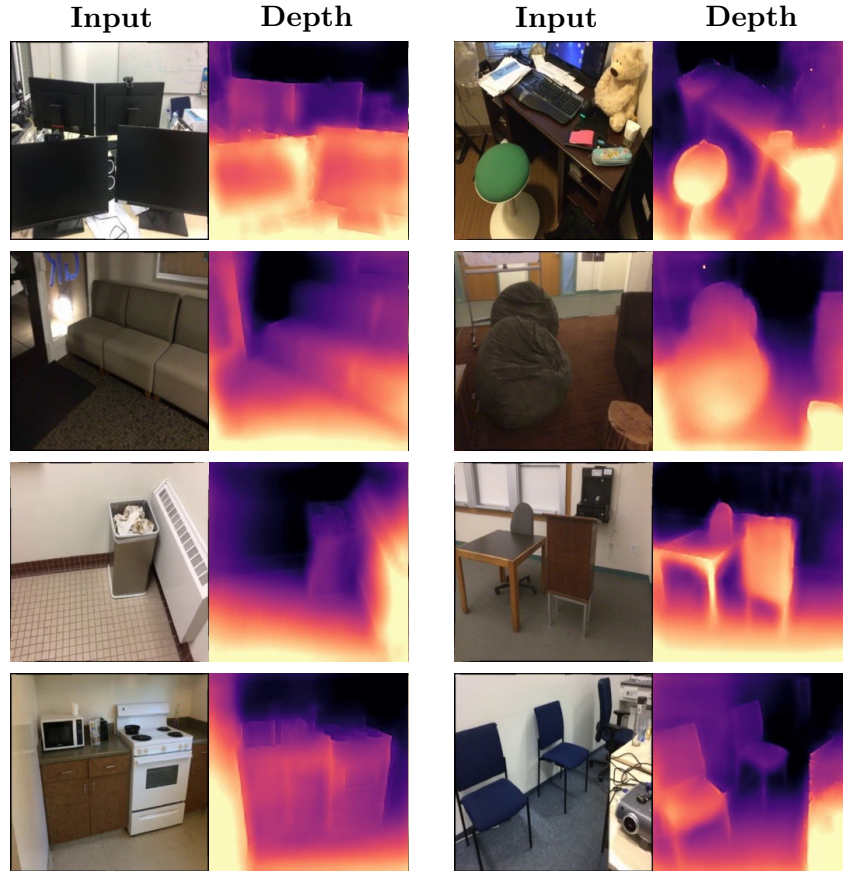


Figure 1: Visualization of depth map on ScanNet (Dai et al., 2017)

REFERENCES

- Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.
- Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828–3838, 2019.
- Michael Grupp. evo: Python package for the evaluation of odometry and slam. <https://github.com/MichaelGrupp/evo>, 2017.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Zihang Lai, Sifei Liu, Alexei A Efros, and Xiao Long Wang. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9730–9740, 2021.
- Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12578–12588, 2021.
- Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

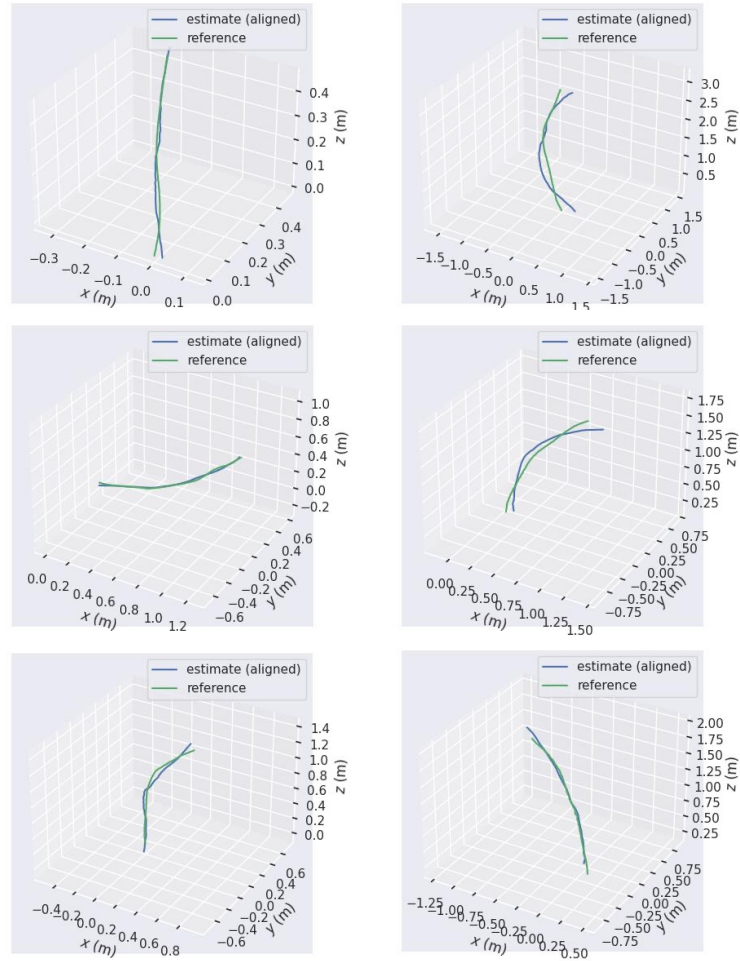


Figure 2: Visualization of estimated camera trajectory on RealEstate10K (Zhou et al., 2018). The green trajectory indicates the ground-truth camera poses while the blue one indicates the estimated poses.

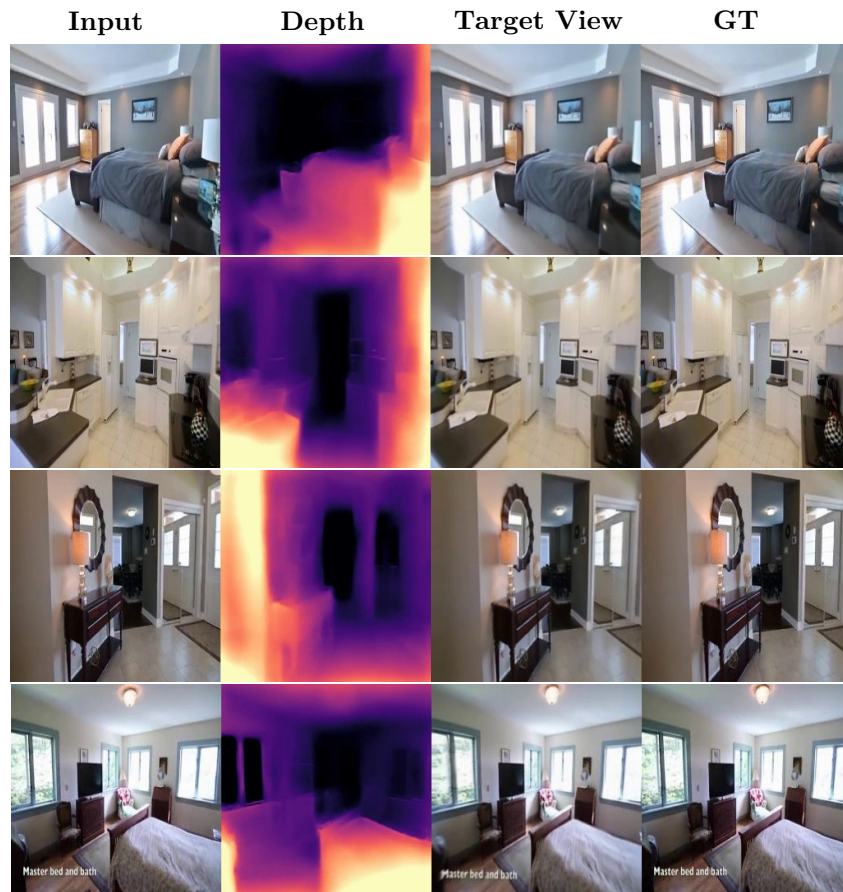


Figure 3: Visualization of depth map and novel view images on RealEstate10K (Zhou et al., 2018)

- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pp. 405–421. Springer, 2020.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.
- Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.
- Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7467–7477, 2020.
- Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pp. 286–301. Springer, 2016.
- Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.