MULTIPLANE NERF-SUPERVISED DISENTANGLEMENT OF DEPTH AND CAMERA POSE FROM VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose to perform self-supervised disentanglement of depth and camera pose from large-scale videos. We introduce an Autoencoder-based method to reconstruct the input video frames for training, without using any ground-truth annotations of depth and camera. The model encoders estimate the monocular depth and the camera pose. The decoder then constructs a Multiplane NeRF representation based on the depth encoder feature, and renders the input frames with the estimated camera. The learning is supervised by the reconstruction error, based on the assumption that the scene structure does not change in short periods of time in videos. Once the model is learned, it can be applied to multiple applications including depth estimation, camera pose estimation, and single image novel view synthesis. We show substantial improvements over previous self-supervised approaches on all tasks and even better results than counterparts trained with camera ground-truths in some applications. Our code will be made publicly available. Video examples can be found in: https://anonymous-result.github.io/.

1 INTRODUCTION

The Autoencoder is a classical technique for visual representation learning. Besides learning representations for recognition tasks (Vincent et al., 2008; Rasmus et al., 2015; He et al., 2021), it has also been widely utilize to learn disentangled representations (Kulkarni et al., 2015; Park et al., 2020). For example, by introducing certain inductive biases, the Autoencoder can learn the disentanglement between structure and texture (Park et al., 2020), pose, light, and shape (Kulkarni et al., 2015) in a self-supervised manner. However, most methods using autoencoders focus on using 2D images.

We propose an Autoencoder that learns representations from videos and disentangles the scene structure and the camera pose. Our method learns such disentanglement in a self-supervised manner, without any supervision from ground-truth camera pose and depth. To achieve this goal, we utilize continuity and persistence in natural videos where the 3D structure remains static over short periods. Specifically, our Autoencoder model encodes the input frames into two separate intermediate representations of 3D scene structure and camera pose. These representations are used to decode the same video frames as the outputs. To ensure that the model learns disentangled representations of depth and pose, we propose a differentiable rendering-based decoder.

We use differentiable rendering with Neural Radiance Fields (NeRF) as our decoder, inspired by its recent success in view synthesis (Mildenhall et al., 2020). Despite its effectiveness in rendering high-quality images, constructing NeRF requires accurate ground-truth camera poses and the learned NeRF is specific to only one scene in most cases. This limits NeRF's applications in large-scale noisy real-world videos. Interestingly, real-world videos often come with slow camera changes (continuity) instead of presenting diverse viewpoints. Multiple continuous frames from the video can reconstruct multiplane images for a given view (Tucker & Snavely, 2020). With this observation, Li et al. (2021) propose to combine the discrete multiplane images into NeRF to create continuous multiplane neural radiance fields, which generalizes NeRF robustly and efficiently to synthesize diverse scenes instead of overfitting to one scene. However, ground-truth cameras computed from Structure-from-Motion (SfM) are still required for learning in (Li et al., 2021). Running SfM in training and testing can be time-consuming and it does not always succeed.

In this paper, we apply the Multiplane NeRF in our decoder and train our Autoencoder model end-to-end on large-scale video data without ground-truth camera pose. Given the input video frames,

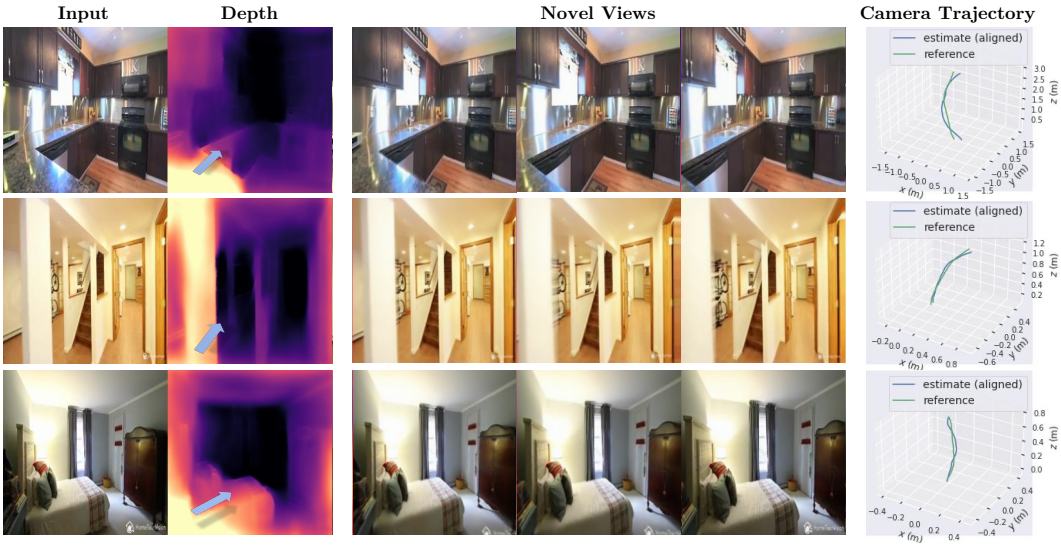

Figure 1: The disentangled representations learned in our model can be applied to depth estimation, novel view synthesis, and camera pose estimation.

our framework uses a depth encoder to perform monocular depth estimation for each frame (which is encouraged to be consistent), and a camera pose encoder to estimate the relative camera pose between every two consecutive frames. The **depth encoder feature** and the **camera pose** are the intermediate disentangled representations. For each input frame, we construct a Multiplane NeRF representation with the depth encoder feature and render it to decode another input frame based on the estimated camera pose. We train the model with the reconstruction loss between the rendered frames and the input frames. However, using a reconstruction loss alone can easily lead to a trivial solution as the estimated monocular depth, camera pose, and the NeRF representation are not necessarily on the same scale. One **key technical contribution** we propose is a novel scale calibration method during training to align these three representations. The advantages of our framework are: (i) Unlike NeRF, it does not need 3D camera pose annotations (e.g., computed via SfM); (ii) It generalizes training on a large-scale video dataset, which leads to better transfer.

At test time, the learned representations can be applied to multiple downstream tasks including: (i) monocular depth estimation from a single RGB image; (ii) camera pose estimation; (iii) single-image novel view synthesis. *We conduct all experiments on indoor scenes in this paper,* as shown in Figure 1. For depth estimation, we train on Scannet (Dai et al., 2017). Our method significantly improves over previous self-supervised depth estimation approaches not only on the Scannet test set and also generalizes to NYU Depth V2 (Nathan Silberman & Fergus, 2012) better. For camera pose estimation, we use RealEstate10K (Zhou et al., 2018) following (Lai et al., 2021) and consistently achieve much better performance compared to previous approaches. For novel view synthesis from a single image input, we estimate the monocular depth using the depth encoder, construct the multiplane NeRF, and then render another view with a given camera. On RealEstate10K (Zhou et al., 2018), our approach significantly improves over methods that learn without camera ground-truth and also outperform recent methods that learn with the ground-truth cameras (Wiles et al., 2020). To our knowledge, our method is **the first work that learns neural radiance fields on a large-scale dataset without camera ground truth**.

2 Related Work

Disentangled representations. Disentangled representations aim to decompose complex visual data into several lower-dimensional individual factors that control different types of attributes. Common approaches to achieve disentanglement include using Generative Adversarial Networks (Chen et al., 2016b; Huang et al., 2018; Karras et al., 2019; Lee et al., 2020; Zhu et al., 2018) and Autoencoders (Jha et al., 2018; Liu et al., 2020; Park et al., 2020; Pidhorskyi et al., 2020). For instance, Park et al. (2020) proposed an Autoencoder to disentangle texture from the structure by enforcing one component to encode co-occurrent patch statistics across different parts of the image. Besides learning from images, recently researchers have looked into using the temporal continuity in videos for learning disentangled representations (Denton et al., 2017; Minderer et al., 2019; Wiles et al.,

2018; Xue et al., 2016; Lai et al., 2021). The most related work to our method is (Lai et al., 2021), where a Video Autoencoder is proposed to disentangle the static 3D scene structure and camera motion from videos. However, their 3D structure is represented by deep voxel features, which cannot reveal the explicit scene geometric structure. Our work is able to directly infer depth as the scene representation, which can be directly used as a downstream application.

Novel View Synthesis. Learning-based novel view synthesis has been a long stand task. Researchers have studied on using explicit 3D representations including voxels (Jimenez Rezende et al., 2016; Kar et al., 2017; Tulsiani et al., 2017; Sitzmann et al., 2019a; Tung et al., 2019; Nguyen-Phuoc et al., 2019), depth maps (Wiles et al., 2020; Rockwell et al., 2021) and multiplane image (Zhou et al., 2018; Srinivasan et al., 2019; Tucker & Snavely, 2020; Li et al., 2021) for view synthesis. For example, Wiles et al. (2020) proposed to infer the depth map from the input image as an intermediate representation and perform rendering from another view for synthesis. Instead of using a single depth map, multiplane image (MPI) representation is utilized to explicitly model the occluded contents during view synthesis (Tucker & Snavely, 2020). Besides explicit 3D representations, recent work on using implicit representations have shown superior performance in view synthesis (Sitzmann et al., 2019b; Niemeyer et al., 2020). Following this line of research, NeRF and its subsequent works (Yu et al., 2021; Trevithick & Yang, 2021; Martin-Brualla et al., 2021; Schwarz et al., 2020; Wang et al., 2021; Meng et al., 2021; Chen et al., 2021) have even achieved photo-realistic rendering results. While the original formulation is restricted to one single instance with the provided camera, recent extensions have made it available to generalize to multiple instances with camera ground-truths (Yu et al., 2021; Trevithick & Yang, 2021; Li et al., 2021) or training on **a single scene** without camera ground-truths (Wang et al., 2021; Meng et al., 2021). For example, Wang et al. (2021) shows that the camera pose can be jointly optimized as learnable parameters with NeRF training. However, this approach only works on training NeRF for a single scene. None of the previous works can generalize to training on large-scale data and without cameras at the same time.

Self-Supervised Depth Estimation. Single image depth estimation has been widely studied in a supervised learning setting (Eigen et al., 2014; Laina et al., 2016; Kendall et al., 2017). However, with the absence of ground-truth depth or camera pose in most real-world data, self-supervised approaches using image reconstruction as the training signal without relying on neither depth nor camera annotations are proposed (Zhou et al., 2017; Vijayanarasimhan et al., 2017; Yin & Shi, 2018; Yang et al., 2018; Mahjourian et al., 2018; Gordon et al., 2019; Li et al., 2020). In this paper, we also follow the setting on learning without both depth and camera ground-truths and apply it on indoor scenes. Different from previous approaches, we show that depth can be learned by rendering with multiplane NeRF, which not only significantly improves depth estimation, but also allows better camera estimation and novel view synthesis results.

Self-supervised learning on video. Our work is also related to self-supervised learning of visual representations from videos (Agrawal et al., 2015; Han et al., 2019; Misra et al., 2016; Wang & Gupta, 2015; Wang et al., 2019; Jabri et al., 2020). However, instead of focusing on learning representations for recognition tasks, our work is more focused on scene geometric understanding for tasks including camera pose estimation, depth estimation, and novel view synthesis.

3 Proposed Method

In this work, we aim to learn disentangled 3D representations from videos in a self-supervised manner (**no camera pose and depth ground-truths**) in an autoencoder fashion. The inputs to our model are video frames (3 frames in our experiments) that are nearby in a short period of time. The video frames are processed with the depth encoder and the camera pose encoder for the depth estimation and camera trajectory estimation respectively. In the decoding process, we construct the multiplane NeRF representation using the depth encoder feature and render using the estimated cameras. We minimize the reconstruction loss between the rendered frames and the input frames to learn the full model. Our model learns disentanglement of the intermediate representations including the **depth feature** (which is used to predict depth) and the **camera pose**. We introduce the encoding process in sections 3.1 and 3.2, the decoding process in section 3.3, and the training details in section 3.4.

3.1 Camera Pose Encoder

The camera pose encoder predicts the relative camera transformation between two input frames as shown at the bottom of Fig 2 (blue box). Specifically, given a source frame I_s and a target frame

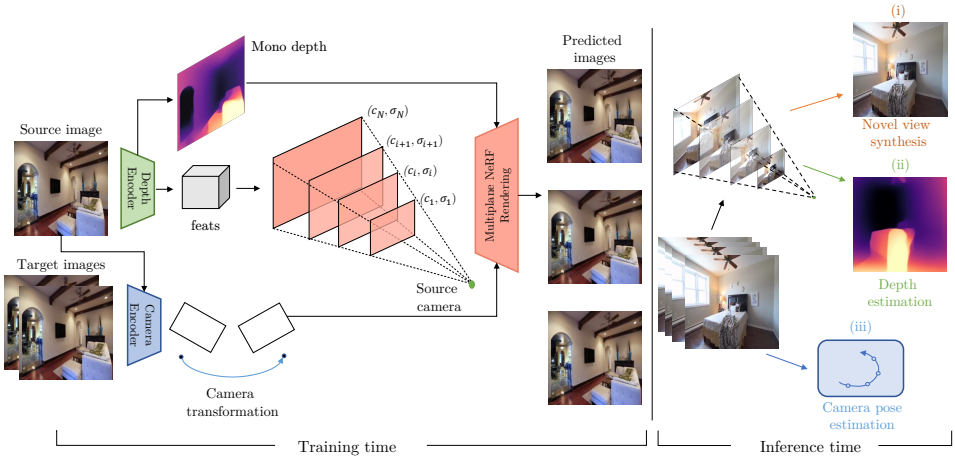

Figure 2: Method overview. Given a short clip of video, the camera encoder and depth encoder disentangle it into depth maps and relative camera trajectory. The Multiplane NeRF is utilized as the decoder to generate the target images according to the estimated camera pose. During training, the model is supervised via the reconstruction loss between the input frames and the generated ones. During testing, three downstream tasks, *i.e.*camera pose estimation, depth estimation, and novel view synthesis can be achieved within a single model.

I_t as inputs, it computes the rotation matrix and translation matrix w.r.t the source view image. For an input sequence during training, we use the middle frame as the source view image and take the remaining frames before and after as target images. We follow the ResNet (He et al., 2016) architecture to design our encoder, which takes both frames as inputs (*i.e.*, stacked along the channel dimension leading to six input channels) and outputs a 6-dim vector as the 3D rotation and translation parameters. We formulate the camera encoder as,

$$\mathbf{T}_{s\to t} := [R, \boldsymbol{t}] = \mathcal{F}_{\text{traj}}([I_s, I_t]) \tag{1}$$

The estimated camera poses for all target images can construct a trajectory and then be used for target view synthesis in the decoder which will be discussed later.

3.2 Monocular Depth Encoder

We design a separate encoder for monocular depth estimation from each single input frame, as shown in the upper part of Fig. 2 (green box). We adopt the network architecture from MnasNet (Tan et al., 2019) as the depth encoder network, which extracts feature maps with different resolution scales to predict the depth map. We formulate the depth encoder as,

$$\mathbf{D}_s = \mathcal{F}_{\text{dep}}(I_s) \tag{2}$$

Note that the raw output $\mathbf{D}_s$ is the disparity map and needs to be converted to the depth map. The output monocular depth map is used as the intermediate representation to guide the construction of Multiplane NeRF.

3.3 Multiplane NeRF based Decoder

The disentanglement is learned via back-propagation from the differentiable decoder. To enable the optimization, we assume that the input video frames are taken from a short range of time scales, and the scene structure remains the same. This assumption provides supervision for our method to construct a Multiplane NeRF representation from a single image in our decoder, and use this representation to render the outputs. We first introduce the multiplane image representation, and then illustrate how to combine it with NeRF to perform rendering in our framework.

Multiplane Images. We review Multiplane Images (MPIs) (Zhou et al., 2018), where an image is represented by a set of parallel planes of RGB-α, $\{(c_i, \alpha_i)\}_{i=1}^{D}$, where $c_i \in \mathbb{R}^{H\times W\times 3}$ are RGB values, $\alpha_i \in \mathbb{R}^{H\times W\times 1}$ are the alpha values and D is the number of planes. Each plane corresponds to a specific disparity (inverse of depth) value d_i uniformly sampled from a predefined range$[d_{\min}, d_{\max}]$. Given the rotation matrix R and translation matrix $\boldsymbol{t}$ from target to source view and the intrinsics matrix for source and target views K_s, K_t, we can generate the target-view image $\hat{\mathbf{I}}_t$ and the disparity map $\hat{\mathbf{D}}_s$ via the following steps. We use $\mathbf{D}$ to denote the monocular depth directly estimated

from the network, and $\hat{\mathbf{D}}$ to denote the depth generated by rendering from MPI. First, the warping operation for the i-th plane from target to source view can be formulated as the following,

$$\begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} \sim K_s\big(R - \boldsymbol{t}\boldsymbol{n}^T d_i\big)(K_t)^{-1} \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} \tag{3}$$

where $\boldsymbol{n}$ is the norm vector of the i-th plane and $[u_s, v_s]$, $[u_t, v_t]$ are coordinates in the source and target views respectively. The MPI representation of the target view can be obtained by warping each layer from the source viewpoint to the desired target viewpoint using Eq. 3. Then, the MPI representation under target view (c_i', α_i') can be described as,

$$c_i'(u_t, v_t) = c_i(u_s, v_s) \qquad \alpha_i'(u_t, v_t) = \alpha_i(u_s, v_s) \tag{4}$$

Finally, the RGB image and the disparity map under both the source view and target view can be obtained via the same compositing procedure proposed in (Zhou et al., 2018),

$$\begin{aligned} \hat{\mathbf{I}}_s &= \sum_{i=1}^{D}\big(c_i\alpha_i' \prod_{j=i+1}^{D}(1-\alpha_j)\big) \qquad & \hat{\mathbf{D}}_s &= \sum_{i=1}^{D}\big(d_i\alpha_i \prod_{j=i+1}^{D}(1-\alpha_j)\big) \\ \hat{\mathbf{I}}_t &= \sum_{i=1}^{D}\big(c_i'\alpha_i' \prod_{j=i+1}^{D}(1-\alpha_j')\big) \qquad & \hat{\mathbf{D}}_t &= \sum_{i=1}^{D}\big(d_i\alpha_i' \prod_{j=i+1}^{D}(1-\alpha_j')\big) \end{aligned} \tag{5}$$

Multiplane NeRF. Going beyond RGB images, we generalize the representations by introducing NeRF as (Li et al., 2021), namely Multiplane NeRF. Different from MPI which consists of multiple planes of RGB-α images at sparse and discrete depths, the Multiplane NeRF achieves continuous representation of 3D scenes by predicting RGB-α images at any arbitrary depth. 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. We follow a similar setting to construct the Multiplane NeRF representation as our decoder to generate the novel view images. Specifically, we extract the intermediate representation from the *monocular depth encoder* (gray cube in Fig. 2) as the image feature for $\mathbf{I}_s$. We combine this feature with a disparity level d_i as the inputs for an internal encoder-decoder module, which outputs the RGB image c_i and the density map σ_i as a 4-channel map $\{(c_i, \sigma_i)\}$ (multiple orange planes in Fig. 2). We have different planes of $\{(c_i, \sigma_i)\}$ given different disparity d_i, and we use positional encoding to encode each d_i. The i-th plane for the Multiplane NeRF representation is formulated as,

$$\{c_i, \sigma_i\} = \mathcal{F}_{\text{mpi}}\big(\mathbf{I}_s, \text{PE}(d_i)\big) \tag{6}$$

Note we only need to run the depth encoder once to extract the image feature for $\mathbf{I}_s$. To reconstruct one target view, given the camera trajectory obtained from the *camera pose encoder* (blue module in Fig 2), we first compute the new RGB and density values on the target view (c_i', σ_i') using homography warpping described in Eq. 3, then replace the alpha map α and the compositing operation in Eq. 5 by the volume density σ and the naive rendering procedure used in (Mildenhall et al., 2020) to obtain the image and the disparity map. The advantages of multiplane NeRF over the vanilla NeRF include: (i) it builds the frustum from a single image; (ii) it has a better generalization ability allowing training on large-scale data, which makes it more feasible than NeRF as the decoder in our autoencoder-like architecture.

3.4 Supervision with RGB

Our model is trained in a self-supervised manner by reconstructing multiple video frames as shown in Fig. 2. During training, we select the center frame of N-frame clip ($N = 3$ in our experiments) as the source view image $\mathbf{I}_s$. We use the depth encoder to estimate the monocular depth $\mathbf{D}_s$ for the source view. We use the camera encoder taking the source view image $\mathbf{I}_s$ and the target view image $\mathbf{I}_t$ as the inputs to obtain the relative camera pose $(R, \boldsymbol{t})$. Together with the depth encoder feature (gray box in Fig. 2) and the estimated camera, we can construct the Multiplane NeRF representation and render the target view $(\hat{\mathbf{I}}_t, \hat{\mathbf{D}}_t)$ as the outputs. The autoencoder is supervised by comparing the rendered target image $\hat{\mathbf{I}}_t$ and the ground-truth target image $\mathbf{I}_t$. However, a direct reconstruction objective can easily lead to trivial solutions given both depth and camera ground truths are not provided in training. We propose **two key technical contributions** including auto-scale calibration and new loss functions to enable successful disentanglement of depth and camera pose.

3.4.1 AUTO SCALE CALIBRATION

Recall that our Multiplane NeRF is built upon a single image, this can lead to the scale ambiguity issue. As explained in (Tucker & Snavely, 2020; Li et al., 2021), each training sequence can be considered equally valid when we scale down or up the world coordinate by any constant value. To tackle this issue, Li et al. (2021) and Tucker & Snavely (2020) propose to use Structure-from-Motion (SfM) to compute **camera pose** and the **depth** (sparse point cloud), where both are at the same scale. The calibration procedure is to adjust the camera pose by comparing the depth from SfM and the rendered depth map from Multiplane NeRF. However, the requirement of running SfM in training and testing is time-consuming and it does not always succeed.

In this paper, we propose to overcome the limits of SfM, and use the encoders to estimate the camera pose $\mathbf{T}_{s\to t}$ (Eq. 1) and disparity map $\mathbf{D}_s$ (Eq. 2). In this case, none of the **camera pose** $\mathbf{T}_{s\to t}$, the **disparity map** $\mathbf{D}_s$ and the **NeRF rendered disparity map** $\hat{\mathbf{D}}_s$ are at the same scale initially. We need to calibrate all three together at the same time in the following two steps.

(i) First, we encourage the rendered disparity map $\hat{\mathbf{D}}_s$ to be consistent with the disparity prediction $\mathbf{D}_s$ by minimizing the $\mathbf{L}1$ distance between them. In detail, we first convert the disparity map into the depth map and then compute the pixel-wise L1 distance between them,

$$\mathcal{L}_{\text{consist}} = \frac{1}{HW}\sum |\frac{1}{\mathbf{D}_s} - \frac{1}{\hat{\mathbf{D}}_s}|_1 \tag{7}$$

The above step aligns the rendered depth result with the monocular depth estimation result.

(ii) Meanwhile, we need to guarantee that the monocular depth estimation and the estimated camera pose are on the same scale. We achieve this goal via applying a photometric reprojection loss (Godard et al., 2019) between the original source image $\mathbf{I}_s$ and the synthesized source image $\mathbf{I}_{t\to s}$, obtained by projecting pixels from $\mathbf{I}_t$ onto $\mathbf{I}_s$ given the predicted monocular depth $\mathbf{D}_s$, camera transformation $\mathbf{T}_{s\to t}$ and the camera intrinsic $\mathbf{K}_s$.

$$\mathcal{L}_{\text{reproj}} = \frac{1}{HW}\sum |\mathbf{I}_s - \mathbf{I}_{t\to s}|_1 \qquad \mathbf{I}_{t\to s} = \mathbf{I}_t\langle \text{proj}(\mathbf{D}_s, \mathbf{T}_{s\to t}, \mathbf{K}_s)\rangle \tag{8}$$

These two steps can achieve the calibration among the camera pose $\mathbf{T}_{s\to t}$, the disparity map $\mathbf{D}_s$ and the NeRF rendered disparity map $\hat{\mathbf{D}}_s$ by enforcing the alignment between the synthesized disparity map $\hat{\mathbf{D}}_s$ and the estimated disparity map $\mathbf{D}_s$ as well as the alignment between camera pose $\mathbf{T}_{s\to t}$ and the estimated disparity map $\mathbf{D}_s$ simultaneously.

3.4.2 LOSS FUNCTIONS

In addition to the calibration, we also adopt three loss functions: RGB L1 loss $\mathcal{L}_{\text{L1}}$, RGB SSIM loss $\mathcal{L}_{\text{ssim}}$ and edge-aware disparity map smoothness loss $\mathcal{L}_{\text{edge}}$ as described in Tucker & Snavely (2020). The RGB L1 loss and SSIM loss (Wang et al., 2004) are defined as,

$$\mathcal{L}_{\text{L1}} = \frac{1}{HW}\sum |\hat{\mathbf{I}}_t - \mathbf{I}_t| \qquad \mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(\hat{\mathbf{I}}_t, \mathbf{I}_t) \tag{9}$$

Both losses aim at matching the synthesized target image with the ground-truth one. Both $\hat{\mathbf{I}}_t$ and $\mathbf{I}_t$ are RGB images with the size of $H \times W$. Meanwhile, we impose an edge-aware smoothness loss on the synthesized disparity map to align the edge and smoothness region between the disparity map and the original image (Godard et al., 2017; 2019; Tucker & Snavely, 2020; Li et al., 2021),

$$\mathcal{L}_{\text{smooth}} = |\partial_x \frac{\hat{\mathbf{D}}_s}{\bar{\mathbf{D}}_s}| \exp^{-|\partial_x \mathbf{I}|} + |\partial_y \frac{\hat{\mathbf{D}}_s}{\bar{\mathbf{D}}_s}| \exp^{-|\partial_y \mathbf{I}|} \tag{10}$$

where ∂_x and ∂_y are image gradients and $\bar{\mathbf{D}}_s$ is the mean value of the disparity map $\mathbf{D}_s$. Overall, together with the scale calibration losses, the total is:

$$\mathcal{L} = \lambda_{\text{L1}}\mathcal{L}_{\text{L1}} + \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{consist}}\mathcal{L}_{\text{consist}} + \lambda_{\text{reproj}}\mathcal{L}_{\text{reproj}} \tag{11}$$

4 EXPERIMENTS

We empirically evaluate our method and compare it to the existing approaches on three different tasks: monocular depth estimation, camera pose estimation, and single image novel view synthesis. We perform evaluations on indoor scenes. Compared to outdoor street views, indoor scenes have more structural variance and are more commonly used for evaluating all three tasks together.

Methods	Sup	Cam_{ex}	Abs Rel↓	Abs Err ↓	Sq Rel↓	RMSE ↓	$\sigma 1$ ↑
MVDepthNet (Wang & Shen, 2018)	Depth	✓	0.098	0.191	0.061	0.293	89.6
GPMVS (Hou et al., 2019)	Depth	✓	0.130	0.239	0.339	0.472	90.6
DPSNet (Im et al., 2019)	Depth	✓	0.087	0.158	0.035	0.232	92.5
Atlas (Murez et al., 2020)	Depth	✓	0.065	0.123	0.045	0.251	93.6
MonodepthV2 (Godard et al., 2019)	RGB	✗	0.205	0.351	0.129	0.453	67.9
Ours	RGB	✗	**0.169**	**0.288**	**0.089**	**0.375**	**76.0**

Table 1: Comparison of depth estimation task on the Scannet (Dai et al., 2017) dataset. We measure the standard metrics on the whole test set released by (Dai et al., 2017).

4.1 Implementation Details

In the pre-processing step, we resize all images to the resolution of 256×256 for both training and testing. During training, we randomly sample 3 frames per sequence with the **interval of** 5 as the input to ensure the camera motion is large enough. The number of planes D is set to 64 and the range of camera frustum is predefined as $[0.2, 20]$. We train our model end-to-end using a batch size of 4 with an Adam optimizer for 10 epochs. The initial learning rate is set to 0.0001 and is halved at $4, 6, 8$ epochs. We empirically set the balance parameters $\lambda_{\text{L1}}, \lambda_{\text{ssim}}, \lambda_{\text{smooth}}, \lambda_{\text{consist}}$ and λ_{reproj} in Eq. 11 to $1.0, 1.0, 1.0, 0.01, 1.0$ and 30, respectively. All configurations and hyperparameters are shared for all experiments over three tasks unless specified.

4.2 Depth Estimation

We evaluate our depth estimation results on two standard benchmarks: ScanNet (Dai et al., 2017) and NYU-depth V2 (Nathan Silberman & Fergus, 2012). We use the synthesized (rendered) depth map as our prediction result and evaluated it by standard metrics introduced in (Eigen et al., 2014). Before evaluation, we first align predictions with the ground truths for scale ambiguity issue, which is a common strategy for monocular depth estimation (Tucker & Snavely, 2020; Yin et al., 2021).

For the experiment on the ScanNet (Dai et al., 2017), we train our framework with all training sequences and evaluate it on all testing sequences released in the official test split. We first compare our model with several fully supervised methods that trained with ground-truth depth supervision: MVDepthNet (Wang & Shen, 2018), GPMVS (Hou et al., 2019), DPSNet (Im et al., 2019) and Atlas (Murez et al., 2020). We directly borrow the performance reported in their paper and list them in Table 1. Note that most of these methods are based on MVS with at least two images as input while our work only requires a single image as input. Without any depth ground truths, our approach still achieves a comparable result with some state-of-the-art. Meanwhile, compared to MonodepthV2 (Godard et al., 2019) which also only requires RGB supervision as ours, our method achieves much better performance.

Beyond ScanNet (Dai et al., 2017), we also evaluate the depth estimation performance on NYU Depth V2 (Nathan Silberman & Fergus, 2012). For a fair comparison, we train the model only with RealEstate10K (Zhou et al., 2018) training data as suggested in (Tucker & Snavely, 2020; Li et al., 2021) and report the results in Table 2. We split the existing method into three groups: (i) the depth supervision model; (ii) the RGB supervision model with camera pose; and (iii) the RGB supervision model without camera pose. Notably, compared to MiDas (Ranftl et al., 2020) trained across 10 different datasets with depth supervision, we achieve comparable performance. Although our method is slightly worse than MINE (Li et al., 2021), they utilize ground-truth camera poses for training while we do not. Compared with the approaches without neither depth supervision nor camera poses, our approach significantly outperforms them by a large margin.

4.3 Camera Pose Estimation

We perform camera pose trajectory estimation and evaluate its performance on RealEstate10K (Zhou et al., 2018). Following (Lai et al., 2021), we use $1,000$ 30-frames video clips from RealEstate10K testing data to construct the testing set. For each video clip, we take a pair of images as input and estimate the relative pose between them and repeat this step sequentially through the whole video to obtain the full trajectory. Since the model only estimates the relative pose in the world coordinate defined in our model, we adopt a post-processing step for alignment between the predicted camera trajectory and the SfM trajectory provided by RealEstate10K (Zhou et al., 2018) via the Umeyama algorithm (Umeyama, 1991). We evaluate the Absolute Trajectory Error (ATE) over testing videos and compare it with the state-of-the-art methods in Table 3.

Methods	Sup	Dataset	Cam_{ex}	rel↓	log10↓	RMS↓	$\sigma 1$ ↑	$\sigma 2$ ↑
DIW (Chen et al., 2016a)	Depth	DIW	–	0.25	0.1	0.76	0.62	0.88
MegaDepth (Li & Snavely, 2018)	Depth	Mega	–	0.24	0.09	0.72	0.63	0.88
MiDaS (Ranftl et al., 2020)	Depth	MiDaS 10 datasets	–	0.16	0.06	0.50	0.80	0.95
MPI (Tucker & Snavely, 2020)	RGB†	RealEstate10K	✓	0.15	0.06	0.49	0.81	0.96
MINE (Li et al., 2021)	RGB†	RealEstate10K	✓	0.11	0.05	0.40	0.88	0.98
MonodepthV2 (Godard et al., 2019)	RGB	KITTI	✗	0.25	0.10	0.74	0.62	0.87
MonodepthV2* (Godard et al., 2019)	RGB	RealEstate10K	✗	0.31	0.12	0.82	0.51	0.83
Manydepth (Watson et al., 2021)	RGB	KITTI	✗	0.25	0.10	0.76	0.61	0.87
Ours	RGB	RealEstate10K	✗	**0.17**	**0.07**	**0.57**	**0.73**	**0.94**

Table 2: Comparison of depth estimation task on NYU Depth V2 dataset. We follow the standard metrics. "Sup" denotes the supervision signal used during training. "RGB† " means using both RGB image and sparse depth during training. "MonodepthV2*" is our reproduction of MonodepthV2 on RealEstate10K.

Methods	Cam_{ex}	PSNR↑	SSIM↑	Perc Sim↓
Dosovitsky *et al.* (Dosovitskiy et al., 2015)	✓	11.35	0.33	3.95
GQN (Eslami et al., 2018)	✓	16.94	0.56	3.33
Appearance Flow (Zhou et al., 2016)	✓	17.05	0.56	2.19
SynSin (Wiles et al., 2020)	✓	22.31	0.74	**1.18**
StereoMag† (Zhou et al., 2018)	✓	**25.34**	**0.82**	1.19
SSV (Mustikovela et al., 2020)	✗	7.95	0.19	4.12
SfMLearner (Zhou et al., 2017)	✗	15.82	0.46	2.39
MonoDepth2 (Godard et al., 2019)	✗	17.15	0.55	2.08
P^2Net (Yu et al., 2020)	✗	17.77	0.56	1.96
VideoAE (Lai et al., 2021)	✗	23.21	0.73	1.54
Ours	✗	**25.00**	**0.83**	**0.99**
results on the test split proposed in (Li et al., 2021)				
MPI‡ (Tucker & Snavely, 2020)	✓	27.05	0.87	0.097*
MINE‡ (Li et al., 2021)	✓	28.39	0.90	0.090*
Ours	✗	26.68	0.86	0.143*

Table 4: Comparison of novel view synthesis task on RealEstate10K. We follow the standard metrics of PSNR, SSIM, and Perc Sim (Wiles et al., 2020). The number xx* represents the LPIPS metric using the implementation of (Zhang et al., 2018).†StereoMag makes use of 2 images as input. ‡MPI and ‡MINE use sparse point clouds as the additional supervision signal during training.

Methods	Mean↓	RMSE↓	Max err. ↓
SSV (Mustikovela et al., 2020)	0.142	0.175	0.365
SfMLearner (Zhou et al., 2017)	0.048	0.055	0.1105
P^2Net (Yu et al., 2020)	0.059	0.068	0.1475
COLMAP (Schönberger et al., 2016)	0.024	0.030	0.0765
VideoAE (Lai et al., 2021)	0.017	0.019	0.0410
Ours	**0.009**	**0.011**	**0.0223**

Table 3: Comparison of camera pose estimation task on RealEstate10K.

SfMLearner (Zhou et al., 2017) and P^2Net (Yu et al., 2020) are two works related to ours, which borrow similar ideas from traditional SfM and optimize the camera trajectory and depth map jointly. Our approach outperforms them by a large margin. For instance, the RMSE is reduced from 0.055 to 0.011 which is about a 80% improvement. In addition, our approach is superior compared to the COLMAP (Schönberger et al., 2016) based on the SfM pipeline. Especially for the videos with slow and little camera movement, COLMAP (Schönberger et al., 2016) can hardly work well and always requires plenty of frames to process leading to a much longer inference time. Finally, a similar improvement can be also found when comparing to VideoAE (Lai et al., 2021), which is a recent work on the disentanglement of camera motion and 3D structure.

4.4 Novel View Synthesis

Our approach generates novel view images by rendering the Multiplane NeRF representation into target views. Following the setting of (Wiles et al., 2020; Lai et al., 2021), we evaluate the novel view synthesis on RealEstate10K (Zhou et al., 2018), which is a large-scale walkthrough video dataset with both indoor and outdoor scenes. During training, we follow the training split used in (Lai et al., 2021), while for the testing, we follow two test splits provided by (Lai et al., 2021) and (Li et al., 2021). For evaluation, we randomly sample 5 source frames from each testing sequence and sample target frames that are 5 frames apart from the source frames. We measure the similarity scores by PSNR, SSIM (Wang et al., 2004), and perceptual similarity with VGG (Simonyan & Zisserman, 2014) features. Note that there are two different implementations to calculate the perceptual simi-

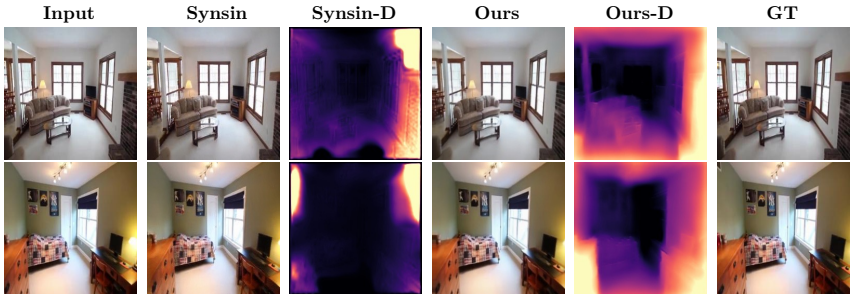

Figure 3: Visualization of depth and novel view images on RealEstate10K. We compare our method with Synsin. Despite they share similar quality of generated images, our depth output is much more accurate.

calib.	PSNR↑	SSIM↑	LPIPS↓
✗	21.46	0.677	0.289
✓	26.68	0.863	0.143

Table 5: Novel view synthesis on RealEstate10K w./w.o. auto scale calibration (Sec. 3.4.1).

#D	PSNR↑	SSIM↑	LPIPS↓
64	26.68	0.863	0.143
32	26.56	0.861	0.141
16	26.65	0.861	0.144

Table 6: Novel view synthesis on RealEstate10K with the different number of planes.

ratio	PSNR↑	SSIM↑	LPIPS↓
1.0	26.68	0.863	0.143
0.8	26.61	0.860	0.145
0.6	26.31	0.857	0.147
0.4	26.21	0.851	0.151
0.2	25.52	0.834	0.161

Table 7: Novel view synthesis on RealEstate10K with different ratios of training data.

larity used in SynSin (Wiles et al., 2020) and MINE (Li et al., 2021), the latter one is also known as LPIPS (Zhang et al., 2018). Table 4 summarizes the novel view synthesis performance over different methods. Compared to single-image view synthesis algorithms, for instance, Synsin (Wiles et al., 2020) , our method can achieve comparable or better performance, even though our method does not require camera pose ground truths while other methods do. Compared with MPI (Tucker & Snavely, 2020) and MINE (Li et al., 2021) where similar 3D representations are adopted, our approach is slightly worse on PSNR and SSIM. We believe this inferior performance is reasonable since they rely on the ground-truth camera pose and the sparse points obtained by COLMAP (Schönberger et al., 2016) during training and testing. On the other hand, our approach easily outperforms all existing methods of training without the camera pose. Some qualitative results are shown in Fig. 3 and more can be found in the supplementary material.

4.5 Ablation Study

We find the performance of three tasks are aligned in our experiments, thus we report the ablation based on the novel view synthesis task here.

Auto scale calibration. We show the effectiveness of auto-calibration (Sec. 3.4.1) by conducting an experiment w./w.o the calibration step. As shown in Table 5, the novel view synthesis performance drops dramatically without the auto scale calibration, *i.e.*, more than 5% on PSNR, which indicates this calibration step is beneficial to scale-invariant synthesis.

Number of planes. We compare our default model with different numbers of planes used in Multiplane NeRF as listed in Table 6. We found that our approach is not so sensitive to the number of planes, but in general, our default setting achieves the best performance.

Amount of training data. We analyze the effect of using different fractions of training data. We uniformly sample every 20% fraction of RealEstate10K (Zhou et al., 2018) training data and evaluate the performance on the same test set. As reported in Table 7, with more training data, the quality of generated images is getting better.

5 Conclusion

We present an autoencoder architecture that disentangles video into camera motion and depth map via the camera encoder and the depth encoder. And the Multiplane NeRF is utilized as the decoder to represent the 3D scene. We further introduce an auto-scale calibration strategy to learn the disentanglement representation even with the camera pose. With the powerful 3D representation, we show our model enables camera pose estimation, depth estimation, and novel view synthesis. Our model achieves on-par or even better results on three tasks compared to approaches with the ground-truth camera or depth during training.

6 Reproducibility Statement

All experiments reported in this paper are reproducible and we are committed to releasing the code once accepted.

References

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pp. 37–45, 2015. 3

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14124–14133, 2021. 3

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016a. 8

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016b. 2

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. 2, 7, 17, 18, 19

Emily L Denton et al. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, 30, 2017. 2

Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1538–1546, 2015. 8

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 3, 7

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 8

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279, 2017. 6

Clément Godard, Oisin 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. 6, 7, 8, 16

Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8977–8986, 2019. 3

Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 17

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019. 3

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. 4, 16

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 1

Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2651–2660, 2019. 7

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 172–189, 2018. 2

Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019. 7

Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020. 3

Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 805–820, 2018. 2

Danilo Jimenez Rezende, SM Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. *Advances in neural information processing systems*, 29, 2016. 3

Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *Advances in neural information processing systems*, 30, 2017. 3

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019. 2

Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pp. 66–75, 2017. 3

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. *Advances in neural information processing systems*, 28, 2015. 1

Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong 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. 2, 3, 7, 8, 17

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pp. 239–248. IEEE, 2016. 3

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128(10):2402–2417, 2020. 2

Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. *arXiv preprint arXiv:2010.16404*, 2020. 3

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. 1, 3, 5, 6, 7, 8, 9, 17, 18

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, 2018. 8

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. 16

Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A Efros, and Noah Snavely. Learning to factorize and relight a city. In *European Conference on Computer Vision*, pp. 544–561. Springer, 2020. 2

Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5667–5675, 2018. 3

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7210–7219, 2021. 3

Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6351–6361, 2021. 3

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. 1, 5, 16

Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. *Advances in Neural Information Processing Systems*, 32, 2019. 2

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pp. 527–544. Springer, 2016. 3

Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European Conference on Computer Vision*, pp. 414–431. Springer, 2020. 7

Siva Karthik Mustikovela, Varun Jampani, Shalini De Mello, Sifei Liu, Umar Iqbal, Carsten Rother, and Jan Kautz. Self-supervised viewpoint learning from image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3971–3981, 2020. 8

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 7

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7588–7597, 2019. 3

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020. 3

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020. 1, 2

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020. 2

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 7, 8

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015. 1

Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14104–14113, 2021. 3

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. 16

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518. Springer, 2016. 8, 9

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 3

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2437–2446, 2019a. 3

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019b. 3

Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 175–184, 2019. 3

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. 4, 16

Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15182–15192, 2021. 3

Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 551–560, 2020. 1, 3, 6, 7, 8, 9

Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2626–2634, 2017. 3

Hsiao-Yu Fish Tung, Ricson Cheng, and Katerina Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2595–2603, 2019. 3

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 7

Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 3

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008. 1

Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International conference on 3d vision (3DV)*, pp. 248–257. IEEE, 2018. 7

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2015. 3

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2566–2576, 2019. 3

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6, 8

Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 3

Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

Olivia Wiles, A Koepke, and Andrew Zisserman. Self-supervised learning of a facial attribute embedding from video. *arXiv preprint arXiv:1808.06882*, 2018. 2

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. 2, 3, 8, 9, 18

Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *Advances in neural information processing systems*, 29, 2016. 3

Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 225–234, 2018. 3

Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 204–213, 2021. 7

Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1983–1992, 2018. 3

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021. 3

Zehao Yu, Lei Jin, and Shenghua Gao. P^2net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *European Conference on Computer Vision*, pp. 206–222. Springer, 2020. 8

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 8, 9

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. 8, 18

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1851–1858, 2017. 3, 8

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. 2, 3, 4, 5, 7, 8, 9, 17, 18, 20, 21

Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. *Advances in neural information processing systems*, 31, 2018. 2

A NETWORK STRUCTURES

We show the details of network structures in Table 8, Table 9, and Table 10, including the camera encoder $\mathcal{F}_{\text{traj}}$, the depth encoder $\mathcal{F}_{\text{dep}}$, 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** ($\mathcal{F}_{\text{dep}}$): 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** ($\mathcal{F}_{\text{mpi}}$): 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,

$$\hat{\mathbf{I}}_s = \sum_{i=1}^{D} T_i\big(1 - \exp\left(-\sigma_i \delta_i\right)\big)c_i \qquad \hat{\mathbf{D}}_s = \sum_{i=1}^{D} T_i\big(1 - \exp\left(-\sigma_i \delta_i\right)\big)d_i \tag{12}$$

  where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$ 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.

B TRAINING & INFERENCE DETAILS

B.1 TRAINING DETAILS.

We adopt a multi-scale training strategy proposed in (Godard et al., 2019). More specifically, $\mathcal{L}_{\text{L1}}$ 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 output1, output2, and output3. Since the monocular depth is used to compute the 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 $\mathcal{L}_{\text{reproj}}$.

B.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

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 8: 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↑, fconv3
upconv3_0	3	1	64	upconv4_1
upconv3_1	3	1	64	upconv3_0↑, fconv2
disp3	3	1	1	upconv3_1
upconv2_0	3	1	32	upconv3_1
upconv2_1	3	1	32	upconv2_0↑, fconv1
disp2	3	1	1	upconv2_1
upconv1_0	3	1	16	upconv2_1
upconv1_1	3	1	16	upconv1_0↑
disp1	3	1	1	upconv1_1

Table 9: The depth encoder ($\mathcal{F}_{\text{dep}}$) architecture. The "↑" is the upsampling operation.

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.

C More Experiments

C.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 11, our model can achieve on par or even better results on both two tasks.

C.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. 4.

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, PE(d_i)
upconv4_1	3	1	128	upconv4_0↑, fconv3, PE(d_i)
upconv3_0	3	1	64	upconv4_1
upconv3_1	3	1	64	upconv3_0↑, fconv2, PE(d_i)
output3	3	1	4	upconv3_1
upconv2_0	3	1	32	upconv3_1
upconv2_1	3	1	32	upconv2_0↑, fconv1, PE(d_i)
output2	3	1	4	upconv2_1
upconv1_0	3	1	16	upconv2_1
upconv1_1	3	1	16	upconv1_0↑
output1	3	1	4	upconv1_1

Table 10: Multiplane NeRF ($\mathcal{F}_{\text{mpi}}$) architecture. The MnasNet+FPN* is shared by both depth encoder and Multiplane NeRF.

Methods	*novel view synthesis*			*depth estimation*		
	PSNR↑	SSIM ↑	Perc Sim↓	Abs Rel↓	Sq Rel↓	RMSE↓
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 11: 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).

Camera Pose Estimation. We also plot the camera pose trajectory in Fig. 5. 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. 6. 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.

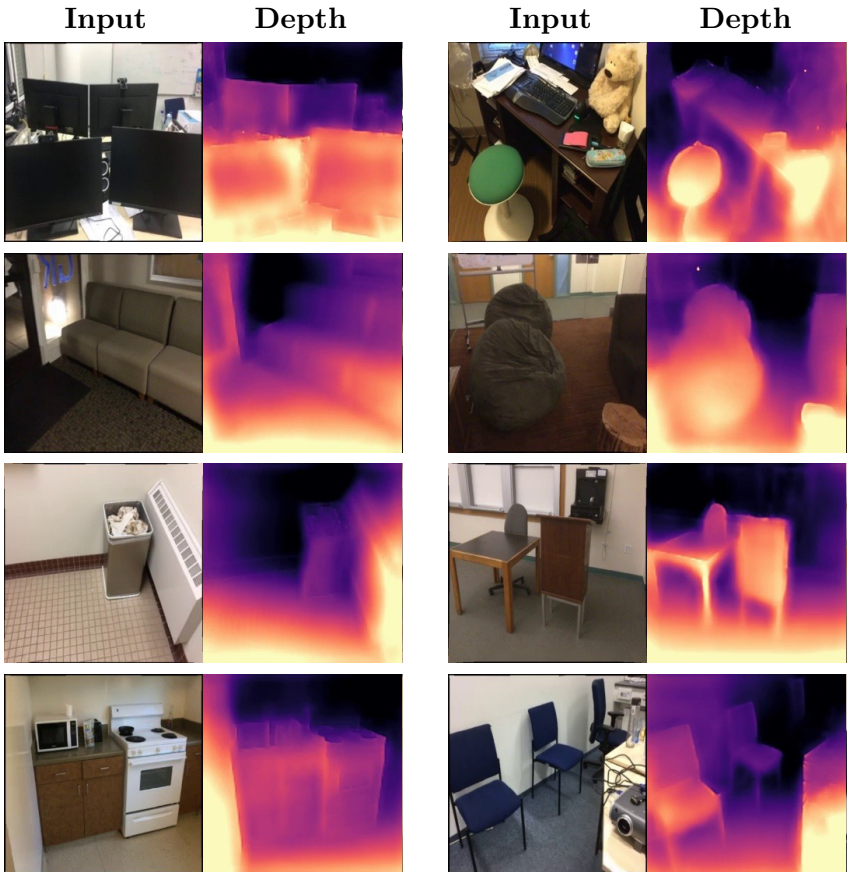

Figure 4: Visualization of depth map on ScanNet (Dai et al., 2017)

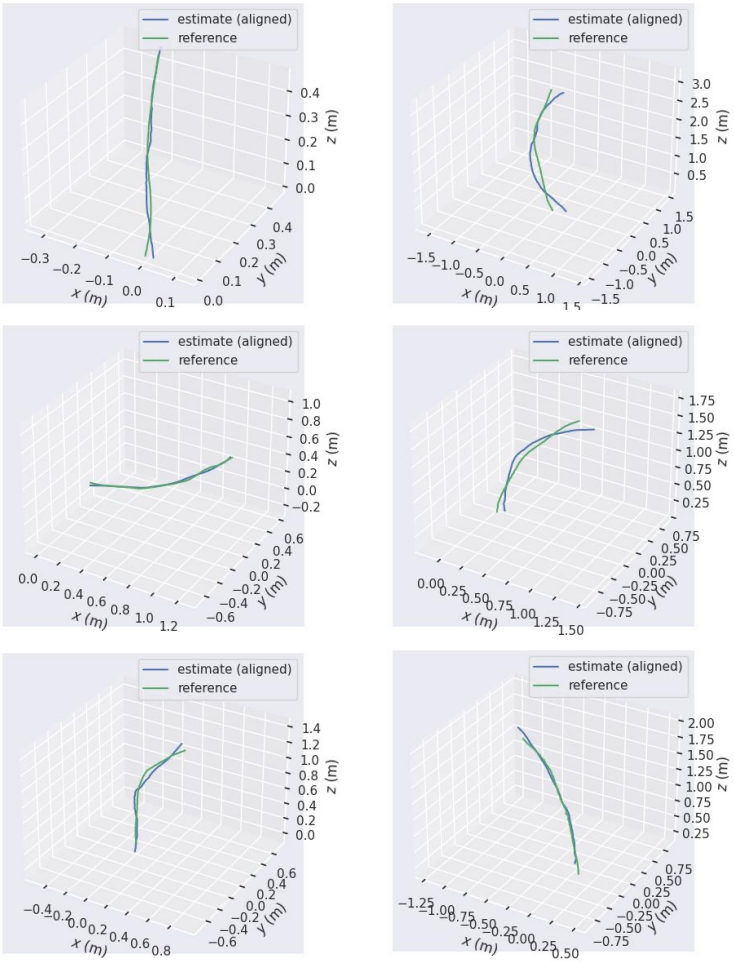

Figure 5: Visualization of estimated camera trajectory on RealEstate10K (Zhou et al., 2018). The green trajectory indicates the ground-truth camera poses while the blue one indicates the estimated poses.

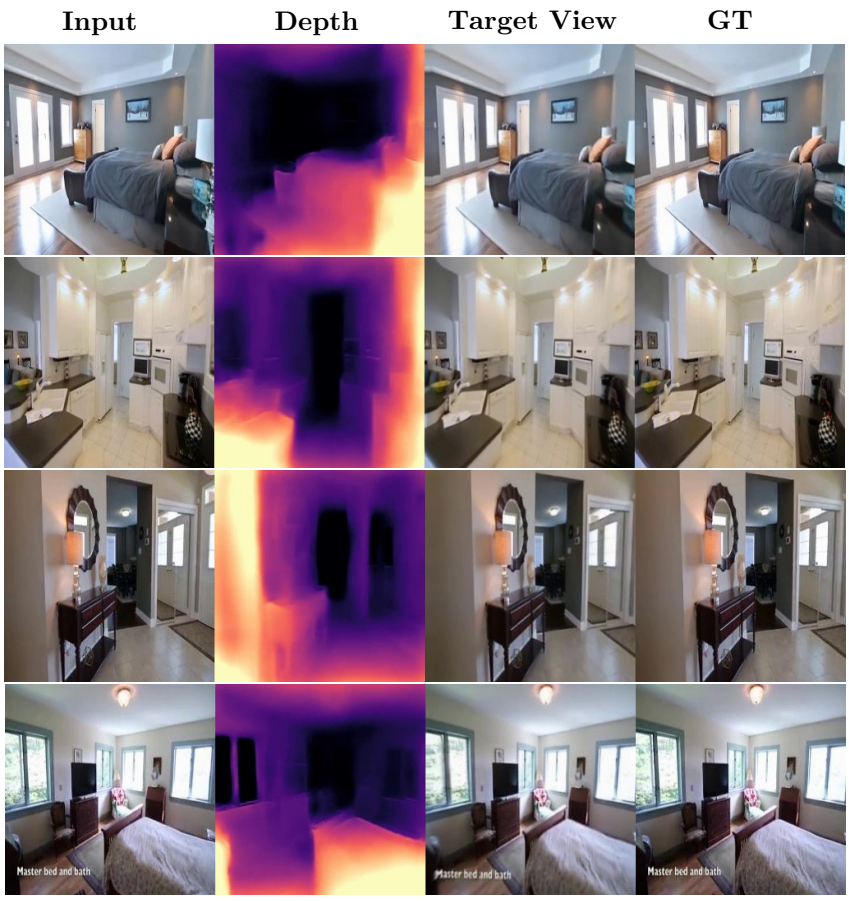

Figure 6: Visualization of depth map and novel view images on RealEstate10K (Zhou et al., 2018)