

Novel View Synthesis with View-Dependent Effects from a Single Image

Juan Luis Gonzalez Bello

juanluisgb@kaist.ac.kr

Munchurl Kim

mkimee@kaist.ac.kr (corresponding author)

School of Electrical Engineering
 Korea Advanced Institute of Science and Technology (KAIST)

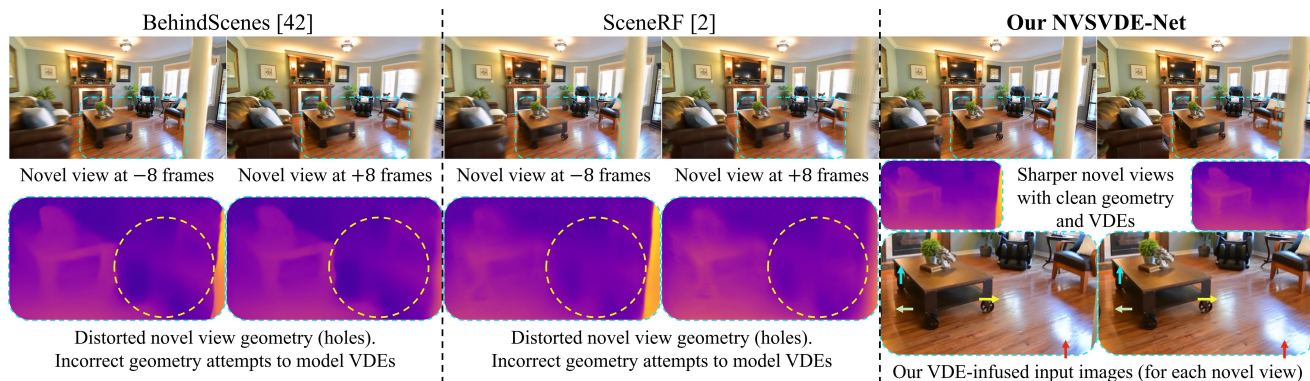


Figure 1. Comparison of single-view NVS. See Supplemental and <https://shorturl.at/ltJT7> for videos.

Abstract

In this paper, we address single image-based novel view synthesis (NVS) by firstly integrating view-dependent effects (VDE) into the process. Our approach leverages camera motion priors to model VDE, treating negative disparity as the representation of these effects in the scene. By identifying that specularities align with camera motion, we infuse VDEs into input images by aggregating pixel colors along the negative depth region of epipolar lines. Additionally, we introduce a ‘relaxed volumetric rendering’ approximation, enhancing efficiency by computing densities in a single pass for NVS from single images. Notably, our method learns single-image NVS from image sequences alone, making it a fully self-supervised learning approach that requires no depth or camera pose annotations. We present extensive experimental results and show that our proposed method can learn NVS with VDEs, outperforming the SOTA single-view NVS methods on the RealEstate10k and MannequinChallenge datasets. Visit our project site¹.

1. Introduction

Novel view synthesis (NVS) is a fundamental computer vision task that aims to generate new views of the input

scene from arbitrarily different camera positions. Recent advances in NVS have shown that view-dependent effects (VDE), which increase the realism and perceived quality of the novel views, can also be learned and incorporated into the rendering pipelines. Particularly, NeRF [32] has demonstrated that radiance fields can be effectively learned by a multi-layer perceptron (MLP) that allows rendering geometry and VDEs from multi-view captures.

NeRF relies purely on multi-view consistency and cannot exploit prior knowledge, such as textures and depth cues common across natural scenes. This limits NeRF when only a few or only one view is available. On the other hand, to leverage the 3D prior knowledge in multi-view datasets, PixelNeRF [46] proposed to train an MLP that takes as inputs the spatial locations, the viewing directions, and the pixel-aligned deep features to generate the colors and densities in the radiance fields. However, PixelNeRF is still limited in learning the ill-posed task of directly mapping input image pixels to VDEs. Other works, such as single-image MPIs [40], MINE [26], BehindTheScenes [42], and SceneNeRF [2] have also proposed single-view-based NVS, but poorly model VDEs.

VDEs depend on the material’s reflectance, which is a function of the material properties and the light’s angle of incidence. Learning such material properties and light sources from a single image is a very ill-posed problem.

¹<https://kaist-viclab.github.io/monovde-site>

While MLP [32] and spherical-harmonic-based [11, 43, 45] techniques to encode VDEs are effective when learning from multiple input images, they are still constrained when learning from single images. Instead, *for the first time*, to tackle the estimation of VDEs for single-view-based NVS, we propose to rely on the contents of the single images and estimated (during training) or user-defined (during test time) camera motions to estimate photo-metrically realistic view-dependent effects. Our main contributions are:

- We *firstly* propose a single view NVS method with VDEs, called NVSVDE-Net. By recognizing camera motion priors and negative disparities govern view-dependent appearance, we model VDEs as the negative disparities in the scene induced by the target camera motion.
- The NVSVDE-Net is the *first* to be trained in a completely self-supervised manner in the sense that neither depths nor pose annotations are required. While other methods rely on given depths and/or camera poses, the NVSVDE-Net learns *only* from image sequences. During test, only one single image input is required for the NVSVDE-Net to render novel views with VDEs.
- A novel ‘relaxed volumetric rendering’ approximation method is proposed, allowing fast and efficient rendering of novel views from a single image. The NVSVDE-Net’s rendering pipeline well approximates volumetric rendering by a single pass of a convolutional or transformer-based backbone and re-calibration and sampler MLP blocks.

Additionally, to better learn from image sequences only, we introduce a new coarse-to-fine camera pose estimation network as a secondary contribution.

2. Related Works

Unlike the classical techniques for single view NVS which require user input or rely on hard-coded domain-specific assumptions [21, 22], deep-learning-based approaches can leverage image representations that are common across scenes to render novel views automatically. We distinguish between two kinds of deep-learning-based NVS methods from single images: one kind requires an additional pre-computed depth map [18, 25, 33] and the other kind only uses a single image during test time [2, 26, 37, 39, 40, 42, 44, 46]. Our method lies in the second category, with the remarkable exception of not requiring ground truth (GT) depths and GT camera poses during training. We focus on static forward-facing scenes, not object-centric 360-degree rendering [29] or modeling motion of dynamic objects [28].

2.1. MPI-Based Single View NVS

The multi-plane image representation (MPI) [50] maps single or multiple images into a camera-centric layered 3D representation. Each input image pixel $p = (u, v)$ is mapped to D -number of colors and opacities. New views are rendered

by sampling the MPI colors and opacities from novel camera poses and by aggregating them via volumetric rendering [9]. In [40], Tucker and Snavely proposed single-view NVS with MPIs. Estimating densities and colors in MPIs is challenging and not well-posed to model view-dependent effects, as the MPI’s colors are not a function of the viewing directions. Furthermore, Sampling distances in MPIs depend on intersections with multi-image planes, potentially hindering high-quality rendering. On the other hand, MINE [26] borrows from [32] and [50] by adopting an MPI that is trained and queried via NeRF-like strategies, where the MPI representation is decoded one depth plane at a time. However, MINE still cannot model VDEs, and its sequential decoding increases computational complexity.

In contrast, our method explicitly models VDEs and approximates volumetric rendering by relaxing alpha compositing and obtaining refined sample distances at the target camera view for fine-grained rendering. Furthermore, we do not rely on ORB-SLAM2 predicted camera poses and sparse 3D point clouds for supervision.

2.2. NeRF-based Single View NVS

In Neural Radiance Fields (NeRF) [32], Mildenhall *et al.* introduced a method that effectively maps 3D coordinates and viewing directions to color and density values using an MLP for volumetric rendering. However, NeRF has limitations: Besides overfitting to a single scene, it needs several reference views with accurate camera poses and long processing times. Nevertheless, recent advances have dramatically reduced neural rendering train and test times [4, 24].

PixelNeRF [46] is a cross-scene generalizable variant of NeRF, sacrificing rendering quality for scene generalization. It maps not only 3D ray points but also projected deep features into color and opacity values in radiance fields. Even when viewing directions are incorporated into PixelNeRF’s MLP, the lack of a view-dependent inductive bias prevents PixelNeRF from generating VDEs.

Wimbauer *et al.* extended PixelNeRF in BehindScenes [42] by sampling colors from epipolar lines and processing densities with an MLP head for each point in the target ray. BehindScenes is designed to focus on 3D geometry but still lacks a mechanism for modeling VDEs as colors are projected between the minimum and maximum distance bounds in the target view. SceneRF [2] is also built on PixelNeRF but incorporates a probabilistic ray sampling strategy and a Spherical U-Net, along with depth penalties [14, 15]. Their sampling strategy uses Gaussian mixtures and requires multiple forward passes for rendering. Like NeRF[32], the MLPs in PixelNeRF [46], BehindScenes [42], and SceneRF [2] independently process each sample, amounting to a considerable computational complexity. In contrast, our approach with fine-grained ray sampling and relaxed volumetric rendering allows for all opacity estimations in a *single pass*, yielding efficiency gains.

2.3. Generative Single-View NVS

NVS from a single image has also benefited from new advances in generative models. In [35], Ren *et al.* used an autoregressive transformer with a VQ-GAN [10] decoder to model long-term future frames. Similarly, Rombach *et al.* proposed GeoGPT [36] where they showed that autoregressive transformers can implicitly learn 3D relations between source and target images. In [39], Tseng *et al.* proposed a diffusion model [20] with epipolar attention blocks at its bottleneck that learn to infuse source view information along the epipolar lines into the target synthetic view during the diffusion process. In VQ3D [37], Sargent *et al.* proposed to learn 3D-aware representations and generation from ImageNet [7] using transformer-based [8] autoencoders to map images and latent vectors into a triplane representation for neural rendering [3]. Pre-trained DPT [34] depths and various adversarial losses are used to train VQ3D. VQ3D is limited to render low-resolution views (256×256) with constrained viewpoints more related to object-centric rendering.

Even when generative models aid NVS from a single image in generating novel views with very large baselines, they suffer from severe inconsistencies due to the stochastic nature of generative models. In addition, they suffer from large computational requirements and inference times that still make them impractical for single-view NVS.

2.4. View Dependent Effects

View-dependent effects (VDE) in NVS from multiple images have been achieved by incorporating viewing directions into the scene representations. In the case of NeRFs [1, 32], viewing directions are fed into the late stages of the MLPs, while in other works [11, 11, 43] view-dependent effects are modeled by spherical harmonics which map viewing directions to intensity changes. Even though these methods show SOTA view-dependent effects, they have the severe limitation of not generalizing cross-scenes and requiring several reference frames for inference. Moreover, fitting new scenes is a time-consuming optimization process that can take several hours.

In contrast, we introduce a new rendering pipeline, the NVSVDE-Net, which learns to approximate volumetric rendering for estimating novel views and view-dependent effects (VDE) from videos without requiring camera pose or depth labels. Trained in a self-supervised manner from image sequences, our NVSVDE-Net can render novel views with VDEs during inference, even from unseen single-image inputs. This marks the first instance of showcasing VDEs estimated from single images, leveraging local context and camera motion priors.

3. Proposed Method

Volumetric rendering [9] synthesizes novel camera views by traversing rays r that originate in the target view camera

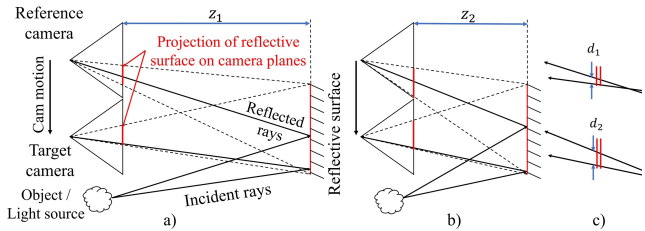


Figure 2. VDEs ‘follow’ camera motion. (a) VDEs have disparities w.r.t. their reflective surfaces in the opposite direction of the projection of the reflective surface itself. (b) and (c) the max VDE disparity is proportional to the reflective surface disparity. The closer the reflective surface, the larger the VDE disparity can be.

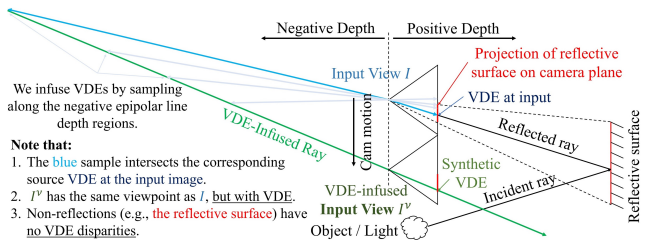


Figure 3. VDEs follow camera motion w.r.t. their reflective surfaces. We exploit this simple yet effective prior for simple glossy/diffuse specular reflections, which are plausible to estimate/render for single image inputs.

center into a 3D volume of colors c and densities σ . The continuous volumetric rendering equation (VRE) is

$$C(\mathbf{r}) = \int_{t_n}^{t_f} \mathcal{T}(t) \sigma(t) c(t) dt, \quad (1)$$

where the accumulated transmittance $\mathcal{T}(t)$ indicates the probability of \mathbf{r} traveling between the near distance bound t_n and ray distance t without hitting a particle. On the other hand, the density $\sigma(t)$ is understood as the probability of \mathbf{r} hitting a particle exactly at t . t_f is the far distance bound.

In NeRF [32], a large number of samples along the ray is considered to discretely approximate the VRE by Monte-Carlo sampling while the processing blocks, such as an MLP or convolutional layers, compute σ_i and c_i for each i^{th} sample along the ray. For this reason, the methods inspired in NeRF [2, 26, 42, 46] tend to yield very slow rendering times as each sample requires an independent forward pass.

3.1. Relaxed Volumetric Rendering

Estimating the c_i and σ_i for volumetric rendering is not a trivial task. The complexity of volumetric rendering increases when a few or a single observation is available, as it becomes a one(pixel color)-to-many(densities and colors) mapping. For this reason, recent works such as [42] ease the network burden by only predicting density values (by several passes of an MLP head) and using colors directly projected from the input image I .

To better incorporate a 3D inductive bias into our models, we propose to relax further the VRE Eq. (1) by directly

modeling the ‘ray point weights’ also known as $\mathcal{T}(t)\sigma(t)$. This allows NVS to be treated as a classification problem for geometry and VDE. Eq. (1) can further be relaxed by using projected colors from a VDE-infused input image I'_c instead of estimated colors c_i . c is the target VDE camera.

For each ray in our relaxed volumetric rendering, VDE-infused colors from I'_c and estimated ray point weights (or depth probabilities) D^P are sampled/perspective-projected at ray depths t_i into the target camera c and integrated to generate the final color estimate $I'_c(\mathbf{p})$ at pixel location $\mathbf{p} = (u, v)$, as given by

$$I'_c(\mathbf{p}) = \sum_{i=0}^{N-1} D_i^P(\mathbf{p}) I'_c(g(\mathbf{p}, t_i, R_c | \mathbf{t}_c, K)), \quad (2)$$

where R_c and \mathbf{t}_c are the target camera rotations and translations, K is the camera intrinsics, and N is the number of samples in the ray \mathbf{r} . Following the literature [12, 17], $t_i = t_n(t_f/t_n)^{1-i/(N-1)}$ exponentially distributes the sampling points in the target rays such that there are more sampling points for the closer depths and fewer for the far-away distances. $g(\cdot)$ is the perspective projection function that outputs the pixel coordinate $(u', v') = g(\cdot)$, allowing to sample colors from I'_c at (u', v') by bilinear interpolation. See *Supplemental* for more details on $g(\cdot)$. Finally, $D_i^P(\mathbf{p})$ is the i^{th} channel of the projected depth probability volume, which approximates $\mathcal{T}(t)\sigma(t)$ in Eq. (1) for our relaxed volumetric rendering, and is given by

$$D^P(\mathbf{p}) = \sigma(\{D_i^L(g(\mathbf{p}, t_i, R_c | \mathbf{t}_c, K))\}_{i=0}^{N-1}), \quad (3)$$

where $\sigma(\cdot)$ denotes the channel-wise softmax operator and $\{D_i^L(\cdot)\}_{i=0}^{N-1}$ denotes the channel-wise concatenation of $D_i^L(\cdot)$. D^L is the estimated depth logit volume, which describes the scene’s geometry seen from the input or source camera view. Additionally, a depth estimate \hat{D} can also be drawn from D^L by a dot-product with all t_i by

$$\hat{D}(\mathbf{p}) = \{t_i\}_{i=0}^{N-1} \cdot \sigma(D^L(\mathbf{p})). \quad (4)$$

Contrary to MPis [40, 50], our approximated volumetric rendering allows us to efficiently and uniformly sample the target view rays, instead of relying on the intersections of the target rays into the multi-image planes.

3.2. Synthesis of View-Dependent Effects

View-dependent effects (VDE), such as the reflection depicted in Fig. 2-(a), seem to ‘follow’ the camera motions relative to their reflective surfaces. While the reflective surface projection (red line) ‘moves’ upwards in the target camera, the reflected ray (or *view-dependent effect*) moves downwards *with respect to* the reflective surface. We propose to exploit this strong yet simple and effective prior to generating a target VDE by re-sampling the pixels that ‘follow’ the camera motion. This operation is equivalent to a

weighted sum of projected source colors along the negative depth region of the epipolar line², as shown in Fig. 3. As depicted in Fig. 3, the target VDE (green ray) intersects the reflection at the source camera in the negative depth region of the epipolar line (blue sample).

Following Fig. 3, we define the VDE-infused input image I'_c at the viewing direction of the target camera c as

$$I'_c(\mathbf{p}) = I_H + \sum_{j=0}^{N_\nu-1} \mathcal{V}_j^P(\mathbf{p}) I(g(\mathbf{p}, 1/\nu_j, \mathbf{I} | \mathbf{t}_c, K)), \quad (5)$$

where $I_H = I - I * k_{5 \times 5}$ roughly contains the high-frequency details of I by subtracting the low-pass box-kernel-filtered $I * k_{5 \times 5}$ from I . I_H aids in generating VDEs while preserving structural details. N_ν is the number of VDE samples from I . $\mathcal{V}_j^P(\mathbf{p})$ are the projected VDE probabilities or ‘weights’ at the location \mathbf{p} . Note that the identity matrix \mathbf{I} (no 3D rotation) in Eq. (5) is used instead of R_c , as a pure 3D rotation of the pinhole camera cannot induce VDEs. Finally, $1/\nu_j$ describes the hypothetical depth values used to sample I along the negative disparity region of the epipolar line.

We made a critical observation to define ν_j : relative VDE ‘motion’ cannot be larger than the corresponding rigid flow generated by the scene depth and camera translation. That is, the *max* disparity in VDEs is inversely proportional to the scene depths. This is further visualized in Fig. 2-(b) and (c), which display that the disparity relative to the reflective surface projection is larger for the closer distance z_2 than that at a distance z_1 in Fig. 2-(a). Then, instead of modeling $|\nu_j|$ to vary from 0 to $1/t_n$, we define it as

$$\nu_j = -\frac{j}{N_\nu-1} \left(\frac{1}{D} - \epsilon \right) - \epsilon, \quad (6)$$

where ϵ is a very small number. \mathcal{V}^P in Eq. (5) is the projected VDE probabilities from the VDE logits \mathcal{V}^L given by

$$\mathcal{V}^P(\mathbf{p}) = \sigma(\{\mathcal{V}_j^L(g(\mathbf{p}, \nu_j, \mathbf{I} | \mathbf{t}_c, K))\}_{j=0}^{N_\nu-1}). \quad (7)$$

Similar to \hat{D} in Eq. (4), a VDE activation map $\hat{\mathcal{V}}$, which is useful for visualizing the most reflective regions in an image, can then be obtained by

$$\hat{\mathcal{V}}(\mathbf{p}) = \{\nu_j\}_{j=0}^{N_\nu-1} \cdot \sigma(\mathcal{V}^L(\mathbf{p})). \quad (8)$$

3.3. NVSVDE-Net

With our relaxed volumetric rendering (Eq. 2) and VDE synthesis (Eq. 5), we can generate novel views with VDEs via the respective depth (geometry) and VDE logits, D^L and \mathcal{V}^L . Fig. 4 depicts the overall architecture of our NVSVDE-Net for joint learning of NVS and VDEs. The

²The same effect can be achieved by keeping a positive VDE depth and inverting the relative camera motion between the source and target views.

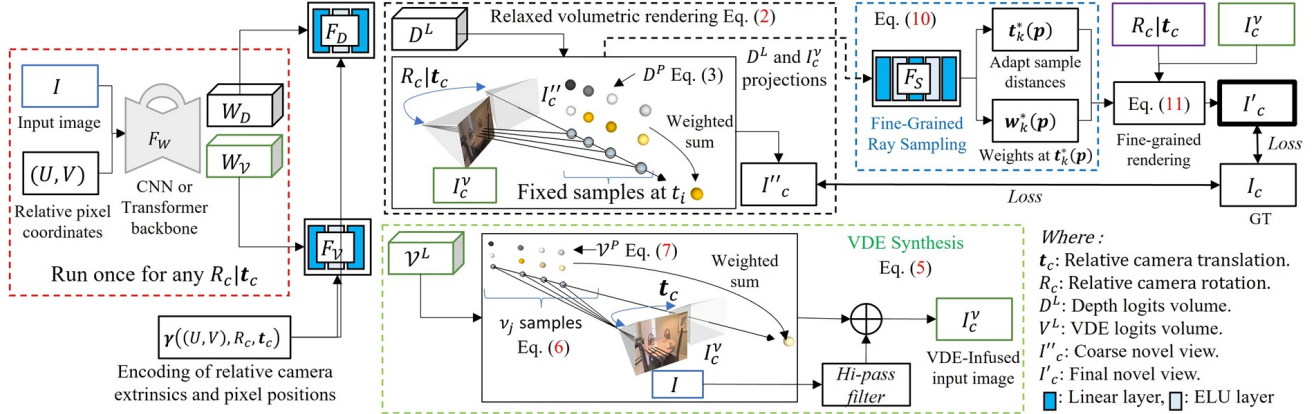


Figure 4. NVSVDE-Net architecture. NVSVDE-Net models VDEs as negative scene disparities under the target camera motion $R_c|t_c$. Novel views are estimated in two stages, firstly with coarse fixed ray samples t_i , then with refined adaptive sampling distances $t_k^*(p)$.

NVSVDE-Net predicts D^L and V^L from geometry and VDE pixel-aligned features in a *single forward pass* by

$$\begin{aligned} D^L(p) &= F_D(W_D(p), \gamma(p, R_c, t_c)), \\ V^L(p) &= F_V(W_V(p), \gamma(p, R_c, t_c)), \end{aligned} \quad (9)$$

where F_D and F_V are MLP heads (Linear-ELU-Linear) that re-calibrate the geometry and VDE pixel-aligned features, W_D and W_V respectively, for the target camera view c . Note that such calibration is needed as the samples at distances t_i (in Eq. 2) and $1/v_i$ (in Eq. 5) are measured from the target camera view I_c , not the input source view I . Pixel positional information p and relative camera extrinsics R_c and t_c are also fed into F_D and F_V via the positional encoding $\gamma(\cdot)$. We explored both a learnable γ_θ (fast and compact but can potentially overfit) and a sine-cosine [32] γ positional encoding (more general but slower due to requiring more channels) where there was little difference.

W_D and W_V enable cross-scene generalization and are simultaneously estimated from a CNN-based (or transformer-based) encoder-decoder backbone F_W as shown in Fig. 4. F_W is also fed with the single image input I and the relative pixel locations (U, V) to improve learning from random-resized and -cropped patches [16] as $[W_D, W_V] = F_W(I, (U, V))$.

Once D^L and V^L are computed, they can be used in Eqs. (2) and (5) to yield the VDE-infused synthetic view I_c^y , as shown in the center of Fig. 4. However, the quality of I_c^y is closely tied to the number of samples N in our relaxed volumetric rendering approximation. Naively increasing N can incur additional computational complexity. Instead, we incorporate a sampler block F_S to estimate fine-grained ray samples from projected depth probabilities and colors.

3.3.1 The Sampler Block

The sampler block F_S in our NVSVDE-Net (top left of Fig. 4) takes as input the projected probability logits D^P

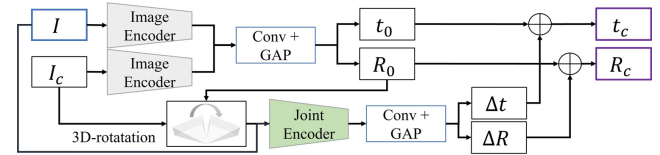


Figure 5. Improved PoseNet. Our PoseNet refines initial coarse extrinsics by predicting a residual between rotation-aligned views.

and projected VDE-infused colors from I_c^y . These are then mapped by a fully-connected network (Linear-ELU-Linear-ELU-Linear) into N^* refined per-pixel sampling distances $t^*(p)$ and soft-maxed weights $w^*(p)$ as given by

$$w^*(p), t^*(p) = F_S(D^P(p), \{I_c^y(g(p, t_i, R_c|t_c, K))\}_{i=0}^{N-1}). \quad (10)$$

$w_k^*(p)$ and $t_k^*(p)$ respectively replace D_i^P and t_i in Eq. (2) to yield the final synthetic image I_c' by a *fine-grained* relaxed volumetric rendering as

$$I_c'(p) = \sum_{k=1}^{N^*} w_k^*(p) I_c^y(g(p, t_k^*(p), R_c|t_c, K)). \quad (11)$$

Note the architecture in Fig. 4 only requires the backbone to be *run once per reference image*. Once W_D and W_V are estimated, the novel views are generated by running the computationally inexpensive re-calibration (F_D , F_V) and the sampler (F_S) blocks according to the relaxed VRE in Eq. (11). Contrary to previous works [2, 26, 42, 46] that also incorporate MLP heads for late-stage rendering, we only need to run them once instead of running them for each point in the target rays. Furthermore, previous single-view-based NVS methods [2, 26, 40, 42, 46] require either depths or pose GTs (or both), while our method learns from image sequences only in an entirely self-supervised manner with the aid from an improved camera pose estimation network.

3.4. Improved Camera Pose Estimation

Previously, camera-pose estimation networks [15, 49] have shown reasonable performance for the monocular depth es-

timization task in driving datasets [6, 13]. However, they can only handle relatively simple camera motions because the cameras mounted on cars are primarily exposed to Z-axis motion and small horizontal rotations. In contrast, indoor datasets [50] and hand-held captured scenes [27] contain arbitrary translations with considerable rotations on all axes.

We observe that based on perceptual features, it is much simpler to understand 3D translation between two images if they are first rotation-aligned. We incorporate this observation into our improved PoseNet as depicted in Fig. 5. In our PoseNet, from a pair of input images (I, I_c), coarse camera extrinsics (R_0, t_0) are estimated by aggregating their deep features (estimated by a shared encoder network) with a convolutional and global average pooling (Conv+GAP) layer. This coarse stage resembles the PoseNets in [15, 49]. R_0 is then used to rotation-align I_c to I . We then extract deep features from the rotation-aligned pair that contain much more relevant visual features, such as closer vanishing points and disparities. By an additional Conv + GAP layer, residual translation and rotation values, ΔR and Δt , are computed and added to R_0 and t_0 to yield the final relative extrinsic parameters R_c and t_c .

3.5. Loss Functions

From a training input image I , synthetic images are estimated for both the previous and next views (I_{-1} and I_{+1}) in the video sequence. We utilize a synthesis loss, l_{syn} , between each synthetic view, both coarse and fine (I''_{-1} and I''_{+1} , I'_{-1} and I'_{+1} , respectively) and their corresponding GT images (I_{-1} and I_{+1}). Additionally, we incorporate disparity smoothness l_{sm} that aids in regularizing \hat{D} . The total loss function is given by

$$l_{total} = l_{syn}(I''_c) + l_{syn}(I'_c) + \alpha_{sm}l_{sm}. \quad (12)$$

The synthesis loss, l_{syn} , is a combination of L1 and perceptual (VGG) loss [23] to enforce similar colors and structures between synthetic and GT views. The VGG loss penalizes deformations, textures, and lack of sharpness as it compares estimated and GT views in the deep feature space of a pre-trained image classification network. The L2 norm of the perceptual error of the first three max-pool VGG19 [38] layers (denoted by ϕ^l) was utilized. l_{syn} is given by

$$l_{syn} = \|I_c^o - I_c\|_1 + \alpha_p \sum_{l=1}^3 \|\phi^l(I_c^o) - \phi^l(I_c)\|_2^2, \quad (13)$$

where $\alpha_p = 0.01$ balances the contributions of the L1 and VGG terms. $I_c^o = (1 - O_c) \odot I_c + O_c \odot I'_c$ is the synthetic view with occluded contents replaced by those in I_c . \odot is the Hadamard product. We compute the occlusion mask O_c following [17, 31] as

$$O_c(\mathbf{p}) = \sum_{i=0}^{N-1} \sigma(D^L)_i(g(\mathbf{p}, t_i, R_c | t_c, K)), \quad (14)$$

The edge-aware smoothness loss l_{sm} with a weight of $\alpha_{sm} = 0.05$ regularizes \hat{D} to be smooth in homogenous image regions. See *Supplemental* for more details.

4. Experiments and Results

Extensive experimental results show our method can generate, for the first time, NVS with VDEs from single image inputs on real datasets that contain complex camera motions and scenes with no depth or pose annotations. Please see *Supplemental* for additional results and *videos*.

We train our NVSVDE-Net with the Adam optimizer ($\beta_1: 0.9, \beta_2: 0.999$) with a batch size of 6 for 50 epochs and 3k iterations per epoch. The initial learning rate is set to 10^{-4} and is halved at 50, 75, and 90% of the training for stability and convergence. We set $N = 32, N_\nu = 32$, and $N^* = 16$ for our relaxed volumetric rendering. We train all models on the RealEstate10k [50] and Mannequin-Challenge [27] datasets with random spatial data augmentations such as random resize and crops, random horizontal flip, and value-based data augmentations such as random gamma, brightness, and color shifts. Training patches are of 240×426 obtained from randomly resized images between 30% and 85% of their original resolution. Finally, 8-bit RGB images are normalized to the $[-0.5, 0.5]$ range.

We adopt the same ResNet34 [19] IMAGENet [7] pre-trained backbone in all models (in ours and [2, 42, 46]), otherwise specified. In particular, we set $N = 64, 48$, and 32 ray samples for PixelNeRF [46], BehindScenes [42], and SceneRF [2], respectively. To match NVSVDE-Net’s N^* , we use 4 Gaussians with four samples in SceneRF. For a fair comparison, we train [2, 26, 40, 42, 46] under the same self-supervised conditions as our NVSVDE-Net. We observed [2, 42, 46] are unstable in the early epochs when learning self-supervised, so they are trained from a fully-trained PoseNet from our best NVSVDE-Net.

4.1. Datasets

RealEstate10k (RE10) [50] is a dataset consisting of 10M frames from approximately 80k video clips from around 10k YouTube videos of indoor and outdoor human-made environments. After downloading, we account for 400k and 180k for training and testing, respectively. We randomly select the 4, 8, 12, or 16th previous or next frame for training. We use every 1k sample from the test set for testing.

The **MannequinChallenge (MC)** [27] dataset is a large, diverse, and challenging dataset that mainly contains humans in both indoor and outdoor scenarios. It contains 170K frames from 2k YouTube videos where people try to stay stationary while a recording camera moves around the scene. After downloading, we account for 60k samples for training and 9k samples for testing. We randomly select up to the 6th previous or next frame for training. We use every 20th frame from the test set for testing.

In both datasets, images have a full resolution of 720×1280 . We train with 1/3 resolution patches and test with 1/2 resolution inputs. We render two views from an input test view, the previous k -th frame and the next k -th frame for evaluation. k is set to 8 and 1 for RE10k and

Methods	VDE	MAE↓	PSNR↑	PSNR _{lf} ↑	SSIM↑	LPIPS↓
RealEstate10k (RE10k) Dataset [50]						
PixelNerf [46]	No	0.0417	22.8455	28.0945	0.7818	0.3256
BehindScenes [42]	No	0.0466	22.9949	28.5941	0.8068	0.2762
MINE [26]	No	0.0415	23.1657	27.8785	0.8041	0.2976
Single-view MPI [40]	No	0.0374	23.6260	28.9447	0.8112	0.2925
SceneRF [2]	No	0.0373	23.6087	28.9636	0.8130	0.2709
NVSVDE-Net (Ours)	Yes	0.0319	24.3131	30.2529	0.8397	0.2325
MannequinChallenge (MC) Dataset [27]						
PixelNerf [46]	No	0.0511	21.3047	25.2781	0.7580	0.3455
BHindScenes [42]	No	0.0463	21.4307	25.9280	0.7831	0.3101
SceneRF [2]	No	0.0467	21.5992	25.8119	0.7796	0.3080
NVSVDE-Net (Ours)	Yes	0.0405	22.4274	27.0263	0.8130	0.2733

Table 1. NVS results. ↓↑ denotes the lower/higher, the better.

Methods	VDE	MAE↓	PSNR↑	PSNR _{lf} ↑	SSIM↑	LPIPS↓
RealEstate10k (RE10k) Dataset [50]						
(w/o VDE)	No	0.0323	24.1679	30.1651	0.8362	0.2320
(VDE disabled)	No	0.0323	24.2092	30.1046	0.8383	0.2325
(w/o F_D & F_V)	No	0.0337	23.9710	29.8303	0.8291	0.2426
($N=48, N^*=0$)	Yes	0.0332	23.9613	29.8602	0.8322	0.2417
($N=32, N^*=32$)	Yes	0.0321	24.3007	30.2156	0.8389	0.2340
(I'_c)	Yes	0.0325	24.1020	29.9808	0.8343	0.2365
(σ -based VRE)	Yes	0.0325	24.1729	30.1202	0.8361	0.2375
(Periodic γ [32])	Yes	0.0323	24.2506	30.1848	0.8366	0.2350
(No resize-crop, test at 1/3)	Yes	0.0331	24.1030	30.4413	0.8213	0.2672
(R18)	Yes	0.0338	24.0340	29.9165	0.8289	0.2437
(Swin-t [30])	Yes	0.0322	24.1917	30.1424	0.8354	0.2365
([49]’s PoseNet)	Yes	0.0337	23.8736	29.8197	0.8259	0.2415
(Full)	Yes	0.0319	24.3131	30.2529	0.8397	0.2325
MannequinChallenge (MC) Dataset [27]						
([49]’s PoseNet)	Yes	0.0454	21.5004	26.2065	0.7787	0.2897
(Full)	Yes	0.0405	22.4274	27.0263	0.8130	0.2733

Table 2. NVSVDE-Net ablation studies.

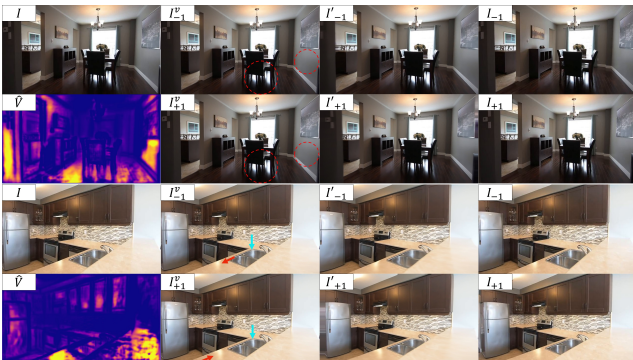


Figure 6. Our NVSVDEnet yields novel views with VDEs. See <https://shorturl.at/1tJT7> Fig5-video.

MC, respectively. Network outputs are bilinearly up-scaled to full resolution before measuring their quality metrics. We measure the quality of the rendered novel views with RMSE, PSNR, SSIM [41], and LPIPS [47] metrics.

4.2. Results on RealEstate10k (RE10k)

The scenes in RE10k contain a considerable amount of perceptually significant view-dependent effects. As shown in Fig. 6, our NVSVDE-Net learns single-view-based realistic NVS and VDEs whose activations map the most reflective regions in the corresponding input images well.

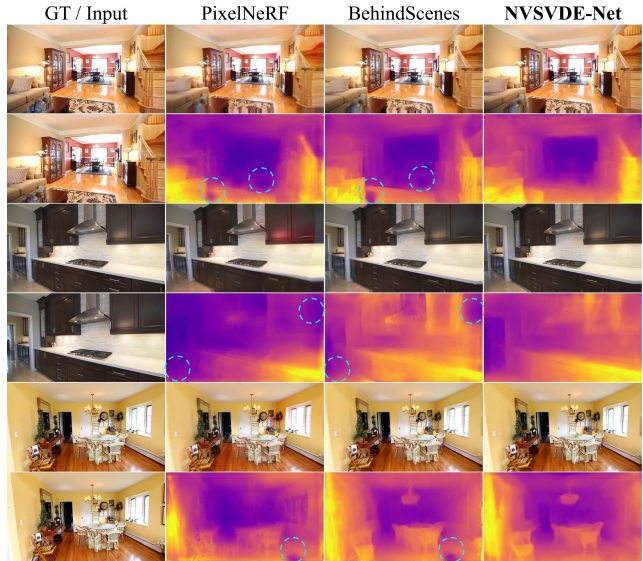


Figure 7. Comparison with previous methods on RE10 [50]. See <https://shorturl.at/1tJT7> Fig6-video

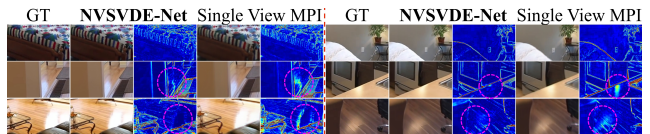


Figure 8. NVSVDE-Net VS MPI. Zoom in for details.

Measuring VDEs is difficult due to their sparse nature. We observed that most VDE that could be modeled from single images consist of low-frequency information, such as glossy reflections. Based on that assumption, we propose the PSNR_{lf} metric in Table 1. PSNR_{lf} takes the PSNR between an estimated image and its GT image, both filtered by a 21×21 -sized Gaussian kernel before comparison. Other metrics that focus on strong image structures, such as SSIM and LPIPS, showed transparency to the changes in VDEs, as VDEs rarely change the high-frequency information or high-level structures in the natural scenes.

Table 1 shows the results on RE10K [50]. Our NVSVDE-Net *significantly* outperforms SceneRF [2] and MPI [40] by a large margin of **0.7dB** in PSNR and **1.3dB** in PSNR_{lf} and previous methods [26, 42, 46] by more than **1.1dB** in PSNR and **1.6dB** in PSNR_{lf}. We qualitatively compare PixelNeRF, BehindScenes, and NVSVDE-Net in Fig. 7, which shows that our model yields more detailed synthetic views with fewer artifacts. Fig. 7 also shows that methods [42, 46] that do not model VDEs struggle to yield plausible geometries for the reflective regions.

Fig. 8 demonstrates cases where MPI [40] and NVSVDE-Net perform similarly (top row) and instances where MPI [40] fails due to reflective surfaces, as indicated by higher error maps (bottom 2 rows). This shows that modeling VDEs by incorrect geometries is insufficient for higher-quality rendering. Our method exhibits more realistic highlights and sharper novel views due to VDE synthesis and fine-grained relaxed VR.

4.2.1 Ablation Studies

Table 2 shows ablation studies for our NVSVDEnet. We study the effects of our proposed VDE synthesis in (w/o VDE), which yields 0.15dB lower PSNR than our full model. Even though the PSNR difference is not very large, a clear advantage in predicted geometries can be observed in Fig. 10, where our model w/o VDEs predicts holes for the reflective regions. We also ablate the effects of our VDEs by disabling the VDE estimation in our full model, denoted by (VDE disabled), which yields 0.15dB lower PSNR_{I_f}.

Next, we ablate design choices in our proposed relaxed volumetric rendering. (w/o F_D & F_V) in Table 2 shows the negative effects of not adjusting the depth and VDE logits volumes. ($N = 48, N^* = 0$) shows that the coarse synthetic output, even with more samples, is still 0.3dB in PSNR behind our full model. On the other hand, ($N = 32, N^* = 32$) shows that only a few adaptive $N^* = 16$ in our full model is enough for optimal results. (I''_c) is the coarse synthetic output of our full model. Interestingly, (I''_c) is 0.2dB lower than I'_c , but 0.14dB higher in PSNR than ($N = 48, N^* = 0$), showing that the joint supervision of I'_c benefits it. The fine-grained I'_c removes most double-edge artifacts due to discrete ray distance discretization as shown in Fig. 10. In our relaxed volumetric rendering, we also explore estimating density values σ instead of $\mathcal{T}(t)\sigma(t)$. However, estimating σ was not only computationally more expensive due to the computation of the accumulated transmittance for each ray sample but also yielded slightly worse quality metrics. A periodic encoding γ was also explored, but we observed slightly worse quality metrics.

We also provide metrics for our NVSVDE-Net with different backbones, such as ResNet18 (R18) and Swin transformer [30] (Swin-t) transformer backbone. Even when NVSVDE-Net (swint) quantitative metrics are not better our ResNet34 full model, its estimated depth and VDE maps are perceptually more consistent (see *Supplemental* for qualitative comparison).

Finally, we also ablate the effect of our improved PoseNet, with a synthesis quality impact of ~ 0.5 dB and ~ 0.9 dB higher PSNR than the widely adopted [49]’s PoseNet, on RE10k and MC respectively. Test poses are also estimated, reflecting the quality of our PoseNet.

4.3. Results on MannequinChallenge (MC)

The bottom part of Table 1 shows quantitative results for single-image-based NVS on the MC [27] validation set. The NVSVDE-Net outperforms the existing methods at least by **0.8dB** in PSNR and **~ 1.2 dB** in PSNR_{I_f}. Fig. 9 depicts qualitative comparisons between our NVSVDE-Net and SceneRF [2]. Again, our method yields sharper and more consistent single-view NVS outputs.

4.4. Runtime Analysis

Our NVSVDE-Net takes 42ms for a 640×360 image on an Nvidia A6000 GPU on vanilla PyTorch. 12ms are used for

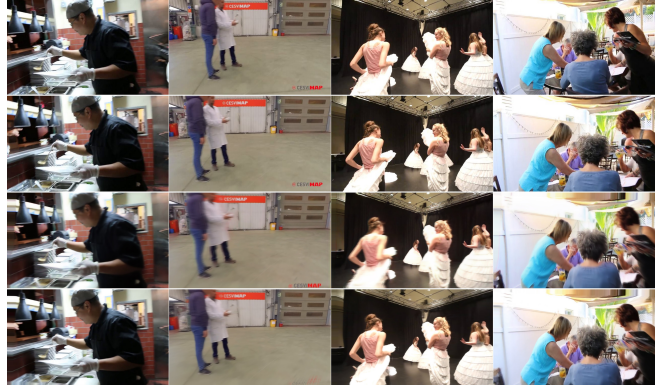


Figure 9. Results on MC [27]. Top to bottom: I , GT, SceneRF [2], Ours. See <https://shorturl.at/ltJT7> Fig7-video.

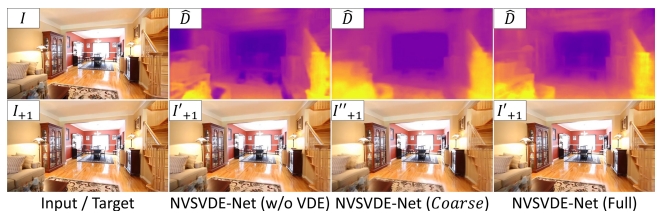


Figure 10. NVSVDE-Net ablation studies. See <https://shorturl.at/ltJT7> Fig8-video.

the backbone operation and 30ms for projections, sampler, and rendering. Since the backbone runs only once for each input image, it only takes 30ms to render novel views of the same scene. Even though our method is near real-time with 33FPS, further speed-ups (up to $3 \times$) are possible with optimized Python packages, such as TensorRT. In comparison, SceneRF, PixelNeRF, and BehindScenes in Table 1, which all share the same 25M-parameter backbone, take 160, 112, and 99ms for rendering a 640×360 image, respectively.

5. Conclusions

We firstly presented a novel method, the NVSVDE-Net, that can learn to perform NVS and VDE estimation on single images in a self-supervised manner from monocular image sequences. We showed that our method generalizes well on unseen test images and that it can generate plausible VDEs and depth maps from a single image. In addition, our NVSVDE-Net incorporates a relaxed approximation to volumetric rendering, which we further improve by incorporating a sampler module for fine-grained ray sampling and rendering. Our NVSVDE-Net yields more realistic NVS images with VDEs in comparison to the recent SOTA methods such as PixelNeRF, BHindScenes, and SceneRF on the RE10k and MC datasets.

Acknowledgement

This work was supported by IITP grant funded by the Korea government (MSIT) (No. RS2022-00144444, Deep Learning Based Visual Representational Learning and Rendering of Static and Dynamic Scenes).

References

- [1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021. [3](#)
- [2] Anh-Quan Cao and Raoul de Charette. Scenerf: Self-supervised monocular 3d scene reconstruction with radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9387–9398, 2023. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#), [4](#)
- [3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. [3](#)
- [4] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *The Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [2](#)
- [5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. [3](#)
- [6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. [6](#)
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [3](#), [6](#), [1](#)
- [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [3](#)
- [9] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4): 65–74, 1988. [2](#), [3](#)
- [10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. [3](#)
- [11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5491–5500. IEEE, 2022. [2](#), [3](#)
- [12] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. [4](#)
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. [6](#)
- [14] Clément Godard, Oisín 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*, pages 270–279, 2017. [2](#)
- [15] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3838, 2019. [2](#), [5](#), [6](#), [1](#)
- [16] Juan Luis Gonzalez and Munchurl Kim. Plade-net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6851–6860, 2021. [5](#)
- [17] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. In *Advances in Neural Information Processing Systems*, pages 12626–12637. Curran Associates, Inc., 2020. [4](#), [6](#)
- [18] Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–8, 2022. [2](#)
- [19] 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*, pages 770–778, 2016. [6](#), [1](#)
- [20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [3](#)
- [21] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584, New York, NY, USA, 2005. ACM. [2](#)
- [22] Youichi Horry, Ken Anjyo, and Kiyoshi Arai. Tour into the picture: Using spidery mesh interface to make animation from a single image”. pages 225–232, 1997. [2](#)
- [23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. [6](#)
- [24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. [2](#)
- [25] Johannes Kopf, Kevin Matzen, Suhil Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, et al. One shot 3d photography. *ACM Transactions on Graphics (TOG)*, 39(4): 76–1, 2020. [2](#)

- [26] 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 *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12558–12568. IEEE, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [4](#)
- [27] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [6](#), [7](#), [8](#), [4](#)
- [28] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023. [2](#)
- [29] Kai-En Lin, Yen-Chen Lin, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 806–815, 2023. [2](#)
- [30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. [7](#), [8](#)
- [31] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):29:1–29:14, 2019. [6](#)
- [32] 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 *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, pages 405–421. Springer, 2020. [1](#), [2](#), [3](#), [5](#), [7](#)
- [33] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (ToG)*, 38(6):1–15, 2019. [2](#)
- [34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. [3](#)
- [35] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3563–3573, 2022. [3](#)
- [36] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14356–14366, 2021. [3](#)
- [37] Kyle Sargent, Jing Yu Koh, Han Zhang, Huiwen Chang, Charles Herrmann, Pratul Srinivasan, Jiajun Wu, and Deqing Sun. VQ3D: Learning a 3D-aware generative model on ImageNet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [2](#), [3](#)
- [38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [6](#)
- [39] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhub Alsisan, Jiabin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16773–16783, 2023. [2](#), [3](#)
- [40] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
- [41] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. [7](#)
- [42] F. Wimbauer, N. Yang, C. Rupprecht, and D. Cremers. Behind the scenes: Density fields for single view reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9076–9086, Los Alamitos, CA, USA, 2023. IEEE Computer Society. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [4](#)
- [43] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8534–8543. Computer Vision Foundation / IEEE, 2021. [2](#), [3](#)
- [44] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016. [2](#)
- [45] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5732–5741. IEEE, 2021. [2](#)
- [46] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4578–4587. Computer Vision Foundation / IEEE, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [4](#)
- [47] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. [7](#)
- [48] Kaichen Zhou, Lanqing Hong, Changhao Chen, Hang Xu, Chaoqiang Ye, Qingyong Hu, and Zhenguo Li. Devnet: Self-supervised monocular depth learning via density volume construction. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, page 125–142, Berlin, Heidelberg, 2022. Springer-Verlag. [2](#)

- [49] 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*, pages 1851–1858, 2017. [5](#), [6](#), [7](#), [8](#)
- [50] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. [2](#), [4](#), [6](#), [7](#), [5](#)