Depth Self-Supervision for Single Image Novel View Synthesis

Depth self-supervision for single image novel view synthesis

Giovanni Minelli[1] Matteo Poggi[1] Samuele Salti[1]

***Abstract*— In this paper, we tackle the problem of generating a novel image from an arbitrary viewpoint given a single frame as input. While existing methods operating in this setup aim at predicting the target view depth map to guide the synthesis, without explicit supervision over such a task, we jointly optimize our framework for both novel view synthesis and depth estimation to unleash the synergy between the two at its best. Specifically, a shared depth decoder is trained in a self-supervised manner to predict depth maps that are consistent across the source and target views. Our results demonstrate the effectiveness of our approach in addressing the challenges of both tasks allowing for higher-quality generated images, as well as more accurate depth for the target viewpoint.**

I. Introduction

In many fields, data is a necessary burden. It plays a crucial role in the industry, helping in decision-making, progress monitoring, and gaining insights. It becomes essential in applications involving deep learning, such as most computer vision tasks. However, obtaining data in large quantities is often challenging, specifically when physical space is limited and does not allow for deploying multiple sensors, as often happens in robotic systems, or when cost and time are significant constraints. Ongoing research is trying to discover alternative ways to enable realistic data generation to compensate for this lack. Among them, we can find Novel View Synthesis (NVS) in computer vision, as the task of generating images framing unseen or occluded parts in a scene. It can be used for applications such as image editing, enhancement of visual experiences or virtual reality [27], where it would be desirable to generate new frames on the fly out of a scene, or even in robotic navigation, potentially allowing an agent to infer and predict a dangerous situation before it could happen [55].

To tackle the NVS task, it is generally necessary to reason about the 3D structure of the scene, which enables to infer the relative motion of objects represented under a view transformation. While recent methods have made significant progress using multiple views to reconstruct 3D scene geometry [29], [30], [24], [35], [41], [13], [25], they cannot be seamlessly extended to work with a single image. When working with a single 2D image, estimating the underlying geometry of the scene becomes an ill-posed problem. Additional prior information in the form of depth [54], [26], [59] or light fields [46] can be used to aid in this estimation and ensure consistency in the geometric structure. However, acquiring such additional data is challenging and requires ad-hoc sensors.

To relax these constraints, recently some methods [8], [20] proposed to implicitly encode knowledge about the scene geometry in a compact latent embedding extracted from the source image and to reconstruct the desired novel view by transforming such a representation according to the relative pose between the two viewpoints. We dub them *source-to-target* approaches. Although such a paradigm is appealing, some limitations dampen its overall effectiveness, where existing frameworks learn the NVS task with direct supervision being limited to the image generation process itself. However, the synthesis process is a direct consequence of the scene geometry which is implicitly modeled during inference – e.g., in the form of depth maps [8], [20] – yet never explicitly optimized. We argue this lack of supervision on geometry yields sub-optimal results, dampening the great potential of source-to-target approaches.

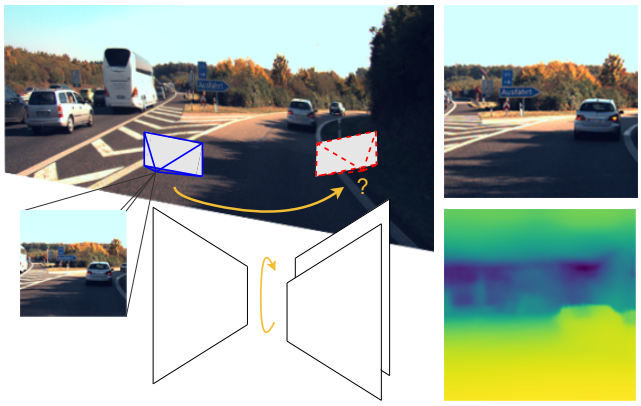

Fig. 1: **Framework overview.** Our model generates a novel, target image and its depth map out of a single, source frame.

In this paper, we propose a new pipeline for source-to-target NVS that explicitly reasons on a latent representation of the scene at the geometry level. The key point of our approach consists in exploiting self-supervised depth estimation to better guide the network in learning such underlying geometry, and to allow for the proper generation of novel views. The synthesis task is performed by two decoders, parsing the encoded embeddings, correcting the distortion effects and producing highly detailed output, as those sketched in Figure 1. The synergy between NVS and depth estimation yields higher quality results in comparison to existing approaches [8], [20], both on synthetic images and on real data, for both generated novel images and estimated depth maps. Our code is available at `https://github.com/johnMinelli/TwoWaySynth/`.

[1]Alma Mater Studiorum University of Bologna

II. Related Work

In this section, we review the research trends in NVS being most relevant to our work.

A. View synthesis from multi-view images.

This family of approaches can be used in a variety of applications, including 3D reconstruction, object recognition, and scene understanding. It can be performed by collecting images from multiple cameras or by taking multiple pictures from different angles with a single camera and then reconstructing the 3D structure of the scene by exploiting the consistency between views. Traditionally, this was done using depth maps [6], or multi-view geometry methods [11], [24], [35], [41], [42], [13], [25], but those approaches often suffer from unreliable photometric consistency and artifacts in the reconstructed images. Then, the use of neural networks – CNNs in particular – has become a popular approach for NVS, by exploiting deep features instead of using explicit images. This strategy leverages geometrical and optical properties such as depth and occlusions to generate novel views [34], [18], [58], [32], [2], [27]. Furthermore, the fully differentiable frameworks often involved allows for merging geometry estimation and novel view synthesis for better results. [48] proposed a framework to combine flow-based predictions from multiple input views and then do pixel generation prediction via confidence maps. [43] reasons about the visibility of the pixels in the different images to implement a consensus volume and then determine the depth of the scene usable to warp source pixels into target views. [4] employs Transformers to fuse 3D point clouds relative to the target viewpoint, extracted from a sparse set of input images with estimated depths.

A very popular and novel paradigm for NVS from multiple images is Neural Radiance Fields (NeRF) [29], using MLPs to infer the volume density and view-dependent emitted radiance from 5D input coordinates – spatial locations and viewpoint directions. Images are generated by querying the MLPs for a set of 3D points along the camera ray of each pixel, through volumetric rendering. As a downside, NeRF necessitates millions of queries to the MLP network and does not generalise across different scenes – i.e., it requires per-scene training. Some variants [56], [7], [52], [49] address the generalisation issue of NeRF showing comparable performance on selected testing scenes, while others focus on training and rendering speed [47], [1], [30].

The underlying 3D geometry of the scene is crucial to properly render novel views. A variety of representations (both implicit and explicit) have been used for NVS. Some methods approaching NVS from multi-view images estimate it by means of 3D representations like voxels [51], [10], [45], [17] or meshes [36], [37], [23], [21], but these methods can be computationally expensive. Some approaches [36], [37] obtain photorealistic results using a mixed approach: starting from a possibly incomplete depth estimation obtained through a structure-from-motion method [40], they build a point cloud used as starting point for the meshing process. Other works with higher computational and memory efficiency rely instead on point clouds directly [54], [2], [4], [38]. However, point-based representations can yield artifacts or holes between points after projection on the image plane, unless complex generative methods are included in the pipeline to fill the gap with plausible generated content [54], [28], [50].

All the approaches introduced so far obtain impressive NVS results, yet require multiple images as input, a hard-to-meet constraint in most applications. In contrast, our method only requires a source image to render a novel view.

B. Source-to-target view synthesis

A different approach to NVS aims at generating a target novel view given a single source image. The lack of geometry knowledge due to the absence of multiple images is compensated by exploiting additional cues, often in the form of ground truth geometry or semantic information to train the 3D representation [31], [44], [51]. We mention in particular the use of depth maps [54], [26], [59], surface normals [26] and light field images [46]. However, collecting large quantities of such data in real-world settings can be challenging. As a result, synthetic environments are often used for training, not excluding the possibility of later refining the results using real-world data.

Image-based rendering techniques use geometry information to generate new views using the pixels from the source images. By projecting one or more images onto a target view and blending the results, these methods can generalize to unseen data creating free-viewpoint images [9], [34], [18]. [33] performs a transformation on the 3D latent space of an encoder–decoder network used to predict an occlusion-aware flow and then refine the transformed image with a completion network. [3] shows how depth estimation and novel view synthesis tasks are tightly linked by generating new data to supervise the first task. [8] obtain novel views using inverse warping in target position by using a 3D transforming autoencoder [19] to predict the depth map needed. Generally, directly generated images may suffer from blurriness, lack of texture details or inconsistency of identity, and [20] solve that by introducing a decoding step to obtain the NVS output.

Among these methods, [8], [20] allows for generating target views without any additional supervision, by implicitly modeling the geometry of the scene in a compact latent representation, that is transformed according to the relative camera poses between the source and the target images to generate the latter. However, these frameworks are supervised at the image level only, without any direct optimization of the latent representation. Our work shares the main objective with these approaches, yet overcomes this latter limitation by explicitly supervising the process at the geometry level by means of self-supervised depth estimation.

III. Proposed Method

In this section, we introduce our framework designed to tackle the source-to-target NVS task by processing a single RGB image, while assisted by depth estimation as an auxiliary task. We consider as input a 255×255 RGB

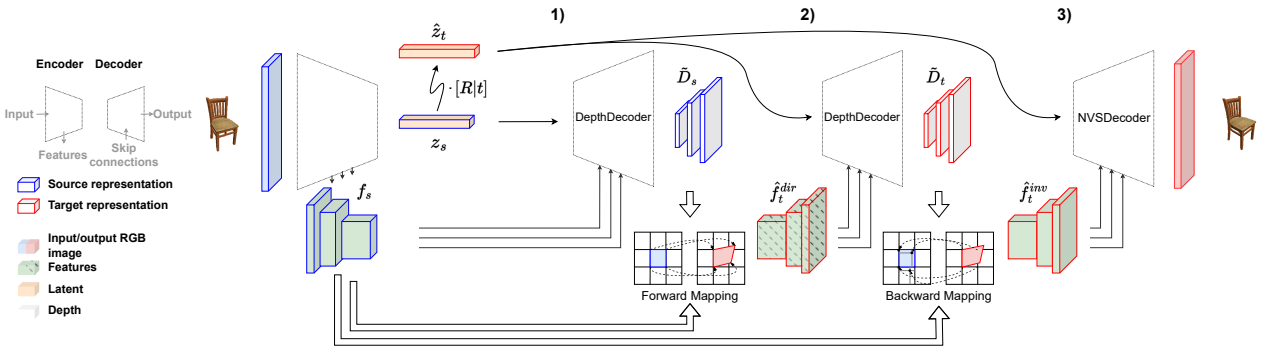

Fig. 2: **Proposed architecture for joint NVS and depth estimation.** A source image I_s is forwarded to a feature encoder to obtain a compact embedding z_s and multi-scale features f_s. Then, z_s and f_s are processed by three decoders for different purposes, respectively 1) by a DepthDecoder to produce a set of depth maps $\tilde{D}_s$ (blue) at different resolutions, 2) by a second DepthDecoder, sharing weights with the previous one, after having been aligned to the target view according to $[R|t]_{s\to t}$, to produce a second set of depth maps $\tilde{D}_t$ (red), and 3) by the NVSDecoder, again after being aligned to the target viewpoint, to generate the novel view $\tilde{I}_t$. To align z_s to the target view, we apply a latent space transformation (Sec. III-B), while f_s are warped by means of forward or backward warping, according to estimated depths $\tilde{D}_s$ and $\tilde{D}_t$ respectively (Sec. III-C).

frame, i.e. the *source* image I_s. During training, it is paired with a second, *target* image – i.e., the novel view I_t we wish to generate starting from the source one – and the relative transformation between the two, expressed by a 4×4 matrix $[R|t]_{s\to t}$. The framework is trained to learn to generate the latter view, i.e., a view $\tilde{I}_t$ that closely resembles I_t. Synthesis is performed by aligning a compact latent space and the intermediate multi-scale features to the target viewpoint. Accordingly, we will refer to $\hat{\varepsilon}$ as the transformed counterpart of a generic ε input from now on. We will now introduce our architecture, depicted in Fig. 2, for joint NVS and depth estimation, as well as the two key working principles implemented inside it.

A. Architecture

The source image I_s is forwarded through an encoder-decoder pipeline to obtain an estimation of the view at target position $\tilde{I}_t$. The encoder reduces the resolution of the input image to a 1×1 multidimensional tensor which embeds geometrical information, z_s, while a set of 5 multi-scale feature maps f_s is obtained during this encoding process, respectively from half to $\frac{1}{32}$ of the original resolution. The generation step will then be performed by the NVSDecoder, the rightmost in Figure 2, which predicts the image from an embedding aligned with the desired viewpoints. Such latent representation can be obtained, following [8], [20], by directly manipulating the source embeddings, as detailed in the remainder. We used ResNet-18 convolutional blocks with pre-trained weights for the encoder, replacing the last pooling stage with an FC layer and a U-Net like architecture for the decoders.

B. Latent Space Transformation

In order to decode an image from the compact embedding forwarded to the NVSDecoder, such an embedding should be geometrically aligned with the desired, target viewpoint. Purposely, inspired by [8], [20] we train our encoder-decoder architecture to learn a compact latent representation that is equivariant to 3D transformations. This can be achieved by directly applying a 3D geometric transformation to the latent code itself. Specifically, the embedding z_s is reshaped into a $N \times 3$ structure and then multiplied by the transformation matrix $T_{s\to t} = [R|t]_{s\to t}$, that describes the roto-translation movement between source and target views. This produces a new latent code $\hat{z}_t$:

$$\hat{z}_t = z_s \times R_{s\to t} + t_{s\to t} \tag{1}$$

aligned to the viewpoint of the target image.

This operation is fully differentiable, thus back-propagating through the whole framework will encourage the features encoder to extract embeddings meaningful to the 3D geometry of the scene. As a consequence, the transformed latent code can be used as a coarse 3D structure by the decoders to synthesize either the new view $\tilde{I}_t$ or its corresponding depth map $\tilde{D}_t$.

C. Direct and inverse warping

While applying the 3D transformation in the latent space is sufficient for producing an embedding aligned with the target view, this is not enough for decoding accurate depth maps and novel views. Indeed, feeding the decoder with features coming from the encoder is crucial for preserving fine-grained details in the final results predicted by U-Net like architectures [39], and this is usually achieved by implementing skip connections. Therefore, to obtain features properly aligned with the target viewpoint, image-warping operators are used to establish a relationship between the pixel (homogeneous) coordinates p_a of a generic image I_a and those p_b of a second frame I_b. This is achieved by using known intrinsics K, pose $[R|t]_{b\to a}$, and depth $\tilde{D}_a$.

$$p_b \sim K T_{a\to b} D_a K^{-1} p_a \tag{2}$$

Accordingly, we can either apply inverse (backward) or direct (forward) warping depending on the features we want

to warp, i.e., respectively from p_b to p_a or from p_a to p_b. While the former is often implemented in the form of spatial transformer networks [22] for tasks such as self-supervised depth estimation [15], the latter is used less frequently, because of the collisions and holes it produces in the warped feature maps. As proposed in [20], a single DepthDecoder – i.e., the second in Figure 2 – would be sufficient to obtain depth map $\tilde{D}_t$, which can then be used to run inverse warping for f_s and provide features $\hat{f}_t^{inv}$ to the NVSDecoder. However, the DepthDecoder itself cannot benefit from the encoder features, since they are still aligned with the source viewpoint. This produces sub-optimal depth predictions and, as a consequence, hinders the accuracy of the NVSDecoder.

For this purpose, we deploy a second DepthDecoder – i.e., the leftmost in Figure 2 – sharing the weight with the first one, to estimate a depth map $\tilde{D}_s$ from features f_s and the embedding z_s. This first depth map, aligned with the source view, allows for running direct warping of the features f_s and aligning them to the target viewpoint, obtaining $\hat{f}_t^{dir}$. The initial decoder benefiting of skip connections produces a more accurate depth map $\tilde{D}_t$, which in turn allows for a better warping of features f_s into $\hat{f}_t^{inv}$ for the NVSDecoder and, finally, increases the quality of the generated, novel image. Since the features f_s are extracted at different resolutions, intermediate depth maps estimated at the same resolutions are used both in direct and inverse warping operations.

D. Loss function

The whole model is trained in an end-to-end fashion, tightly linking NVS and depth estimation. The overall loss function is made of four terms detailed in the remainder.

Image reconstruction loss ($\mathcal{L}_{recon}$). We supervise the NVSDecoder by upsampling each intermediate output $\tilde{I}_t^i$ from scale i up to the original full resolution ($\tilde{I}_t^{i\uparrow}$) and by computing its L1 distance from ground truth image I_t.

$$\mathcal{L}_{recon} = \sum_{i=0}^{\mathcal{S}} |\tilde{I}_t^{i\uparrow} - I_t| \tag{3}$$

Photometric reprojection loss ($\mathcal{L}_{photo}$). In addition to features warping, the second DepthDecoder also uses depth maps $\tilde{D}_t$ to warp source images I_s and generate reprojected images $\hat{I}_t$, which provide self-supervision for the depth estimation task. Dissimilarity between the reprojected images $\hat{I}_t^{\,i}$ and the real images I_t is measured at each intermediate depth map scale i, with the real images being downsampled accordingly.

$$\mathcal{L}_{photo} = \sum_{i=0}^{\mathcal{S}} |\hat{I}_t^i - I_t^{i\downarrow}| \tag{4}$$

VGG perceptual loss ($\mathcal{L}_{VGG}$) As already demonstrated by [20], the adoption of a VGG perceptual loss allows for enhancing the realism of generated results by increasing the sharpness of shapes. This is done by applying the feature extractor Θ of a pre-trained VGG16 network on the generated and the ground truth image individually, and then computing the L1 distance between such features.

$$\mathcal{L}_{VGG} = |\Theta(\tilde{I}_t) - \Theta(I_t)| \tag{5}$$

Edge-aware smoothness loss ($\mathcal{L}_{smooth}$). We further promote the smoothness of predicted depth maps by means of an edge-aware term, as proposed in [15], [16].

$$\mathcal{L}_{smooth}^s = |\partial_x D_s| e^{\partial_x I_s} + |\partial_y D_s| e^{\partial_y I_s} \tag{6}$$

This encourages smooth predictions for textureless regions of the images while preserving depth discontinuities. Similarly, $\mathcal{L}_{smooth}^t$ can be computed from $\tilde{D}_t$ and I_t.

Depth consistency loss ($\mathcal{L}_{skip}$). In order to improve the results by the DepthDecoders, even when dealing with altered or incomplete features as a consequence of either latent space transformation or direct warping, an L1 loss is introduced. We compare the last depth map of the DepthDecoder processing z_t and f_t, and the depth map from $\hat{z}_t$ and $\hat{f}_t^{dir}$ input. The first two terms are respectively embedding and features obtained by encoding the target view, and the last two terms by encoding the source view and applying the transformation introduced in Sec. III-B and III-C (i.e., step 1 in Figure 2).

$$\mathcal{L}_{skip} = |D(z_t, f_t) - D(\hat{z}_t, \hat{f}_t^{dir})| \tag{7}$$

The overall loss function is defined as

$$\begin{aligned} \mathcal{L}_{tot} = &\alpha\mathcal{L}_{recon} + \beta\mathcal{L}_{photo} + \gamma\mathcal{L}_{VGG} + \\ &\delta(\mathcal{L}_{smooth}^s + \mathcal{L}_{smooth}^t) + \omega\mathcal{L}_{skip} \end{aligned} \tag{8}$$

with α, β, γ, δ, ω being weighting terms.

IV. Experiments

We assess the effectiveness of our framework on both synthetic and real images and compare it with existing works built over the same principles, by using the pre-trained models released by the authors. We use the Adam optimizer with β_1 = 0.9, β_2 = 0.999, set the initial learning rate to $6e^{-5}$ with a linear decay program, and train our model for a total of 100k steps with a batch size of 8. The following sections introduce the datasets and metrics used for our evaluation, discuss the results in comparison with existing approaches, and present an ablation study to validate our design choices.

A. Dataset

Following [20], [8], we evaluate our framework on ShapeNet[5] and KITTI[14], respectively representative of synthetic and real images.

ShapeNet. It provides a collection of 3D synthetic objects with thousand of models for each category. We select two categories, in order to benchmark our framework on images with different topologies: chair models present complex designs, often with wiry and elongated features that are harder to be synthesized. On the opposite, car models are more consistent in the shape but the main body presents small details and rich textures that cannot be found in chairs. The dataset used is composed of 6777 chair models and 3514 car models, each rendered in 72 poses (both elevation

TABLE I: **Experimental results for NVS on ShapeNet.** We compare the synthesis quality achieved by Chen et al. [8], Hou et al. [20] and our method. Best results in **bold**.

		L1↓	SSIM↑
Chairs	Chen et al. [8]	0.097	0.907
	Hou et al. [20]	0.073	0.923
	Ours	**0.026**	**0.943**
Cars	Chen et al. [8]	0.044	0.946
	Hou et al. [20]	0.049	0.946
	Ours	**0.026**	**0.954**

and azimuth are taken with a sampling distance of 10°, respectively in [0°,40°] and [0°,360°] ranges) at a resolution of 512×512. For our model, those are downscaled to the proper input size and pairs of images are constructed in a range of [-40°,+40°] azimuth and random elevation. The train test split used is equivalent to the one used by [20].

KITTI. This dataset has been collected during driving sessions, using a car equipped with standard cameras and Velodyne LiDAR sensors. It represents a standard real-world benchmark, used for many computer vision tasks. From KITTI we select 80k samples, obtained by excluding static scenes. The images are center-cropped to match the input size and pairs are constructed by taking the target view randomly within a distance of 7 frames from the source. We use the standard Eigen split[12] for training, while we use the images from the Eigen test split as source views and sample corresponding target views from the sequences containing them. Poses are extracted from oxts files provided with the datasets.

B. Metrics

To quantitatively evaluate the results achieved by our framework, we adopt standard metrics for both tasks. Performances relative to NVS are expressed by the L1 norm, measuring the per-pixel difference between generated and ground truth images and the structural similarity index (SSIM) [53], measuring the perceptual quality of the image. On the KITTI dataset, we also measure the Peak Signal-to-Noise Ratio (PSNR), the higher the better, as well as the Learned Perceptual Image Patch Similarity (LPIPS) [57]. For the depth estimation task, we measure the accuracy of the predictions by the second DepthDecoder – i.e., producing the depth for the target view, crucial for performing the target image synthesis – with respect to ground truth depth, adopting a standard set of metrics from [12] including square-root of scale-invariant logarithmic error (SILog), absolute relative error (Abs. Rel.), squared relative error (Sq. Rel.), root mean squared error (RMSE), root mean squared error between log depths (Log. RMSE) and threshold accuracy ($\delta < 1.25^i$, $i \in [1, 2, 3]$, with δ being the maximum between the prediction over ground truth ratio and its inverse).

C. ShapeNet evaluation

Table I shows results for NVS on the ShapeNet dataset, on chairs and cars. We can notice how our framework consistently achieves a higher quality synthesis with respect to existing works [20], [8]. This can be further appreciated qualitatively in Figure 3. On the one hand, we can notice how [8] often results in distorted shapes due to propagation of errors originated from a poor estimation of the target image depth, obtained exclusively by transformative projection in the synthesis step, while we can observe how [20] sometimes produces abnormally elongated shapes, mostly visible on cars. On the other hand, our improved encoding and decoding process allows for obtaining novel views geometrically consistent with the original one, preserving fine structures (e.g., chair legs).

Fig. 3: **Qualitative results on ShapeNet.** From left to right: (a) source and (b) target images, novel views by (c) Chen et al. [8], (d) Hou et al. [20] and (e) our framework.

D. KITTI evaluation

We further evaluate the effectiveness of our framework on KITTI, a real and more challenging dataset. Specifically, thanks to the availability of depth data collected by the Velodyne LiDAR sensor, we both evaluate the quality of the generated novel views, as well as the accuracy of estimated depth maps for such images. We point out how this depth estimation task differs from standard single-image depth estimation [15], [16], since depth is estimated according to a viewpoint different from the one of the source image, and thus is not directly comparable with this field of the literature.

TABLE II: **Experimental results for NVS and depth estimation on KITTI.** We compare the synthesis quality (left) and depth accuracy (right) achieved by Chen et al., [8], Hou et al. [20] and our method. Best results in **bold**.

	L1↓	SSIM↑	PSNR↑	LPIPS↓	SILog↓	Abs.Rel.↓	Sq.Rel.↓	RMSE↓	RMSE_{log} ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Chen et al. [8]	0.200	0.657	16.012	0.350	25.947	0.191	3.711	10.013	0.270	0.755	0.889	0.944
Hou et al. [20]	0.231	0.649	15.068	0.451	28.852	0.233	2.865	9.037	0.304	0.633	0.848	0.939
Ours	**0.178**	**0.699**	**17.248**	**0.339**	**15.858**	**0.116**	**1.189**	**6.089**	**0.167**	**0.863**	**0.960**	**0.988**

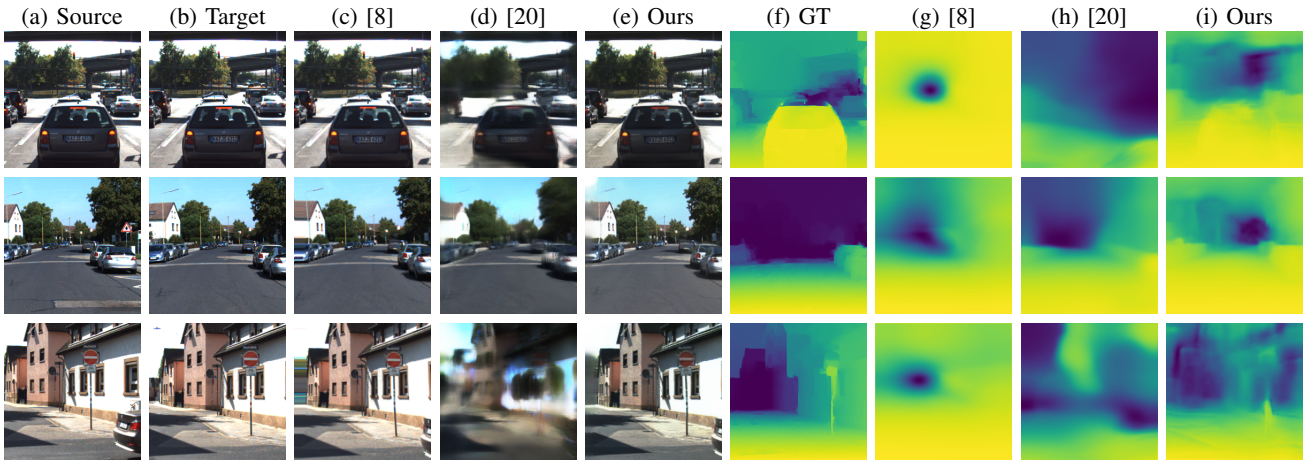

Fig. 4: **Qualitative results on KITTI.** From left to right: (a) source and (b) target images, novel views by (c) Chen et al. [8], (d) Hou et al. [20] and (e) our framework, (f) ground truth depth, estimated depth maps by (g) Chen et al. [8], (h) Hou et al. [20] and (i) our framework.

Table II collects quantitative comparisons between existing approaches [8], [20] and our pipeline, reporting scores concerning both NVS and depth estimation tasks. We can notice how our proposal outperforms both on any metric. Figure 4 shows qualitative results concerning this experiment. We can notice how our framework, by jointly learning how to deal with both NVS and depth estimation, produces more detailed images and depth maps with respect to existing approaches.

Finally, we also highlight a limitation common to any of the considered approaches, made evident by the KITTI dataset itself. By their formulation, these frameworks cannot properly deal with moving objects in the scene, often occurring in real environments such as those featured in the KITTI dataset. As a consequence, rendering a target view in which we witness both a change of the camera viewpoint and the motion of some objects in the scene will produce a novel view in which the objects themselves did not move. Figure 4 shows an example on the third row. The rightmost car in the source image moves and, in addition to the small movement of the camera, disappears from the target view. However, each of the three methods generate images in which the car is still visible as if it was static. We leave explicitly modelling of the objects' movements in the scene as independent motions to future work.

E. Ablation study

We conclude our evaluation by measuring the impact of each component in our pipeline through an ablation study carried out on the KITTI dataset. Table III reports the outcome of this experiment, both in terms of NVS and depth estimation. Best results are reported in bold, while second bests are colored in blue.

On top, we recall the results achieved by our full pipeline. Then, we first study the impact of architectural choices, reporting the results achieved by (I) forwarding direct warped features of the encoder to the NVSDecoder by means of source depth map – i.e., by removing the second DepthDecoder – as well as by (II) removing skip connections forwarded as input to the NVSDecoder or (III) the DepthDecoder. We can notice how dropping any of these design choices leads to a sensible decline in both NVS quality and predicted depth accuracy. Figure 5 confirms this qualitatively. Specifically, configuration (III) corresponds to [20], indeed presenting the same blurring and distortion effects as it. Configuration (II) underline the clear effectiveness of the skip connections in the RGB space: here only a vague shape can be synthesized from the encoded information, in contrast with the highly detailed images obtained by our complete pipeline. Finally, removing skip connections directed to the DepthDecoders (III) affects the NVS process in a minor manner compared to what we observed for the NVSDecoder. However, the removal of such connections yields blurred depth maps, consequently causing the inverse warping to generate less effective reprojection of the encoder features to match the target viewpoint.

At the bottom of the table, we also measure the impact of the different terms building up our loss function. We can notice how, by selectively turning off single terms, we can prioritize one task over the other. For instance, by neglecting the multiscale reconstruction loss $\mathcal{L}_{recon}$ (IV) we

TABLE III: **Ablation study on KITTI – NVS and depth estimation.** We compare the synthesis quality (left) and depth accuracy (right) achieved by our full pipeline and ablated variants (I-VIII). Best results in **bold**, second best in blue.

		L1↓	SSIM↑	PSNR↑	LPIPS↓	SILog↓	Abs.Rel.↓	Sq.Rel.↓	RMSE↓	RMSE_{log} ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
	Ours (full)	0.178	0.699	17.248	0.339	15.858	0.116	1.189	6.089	0.167	0.863	**0.960**	**0.988**
(I)	No inverse warping	0.190	0.688	16.594	0.404	26.812	0.217	2.665	8.498	0.281	0.672	0.878	0.950
(II)	NVSDec w/o skips	0.309	0.620	12.843	0.702	21.497	0.167	1.898	7.480	0.224	0.776	0.927	0.974
(III)	DepthDec w/o skips	0.185	0.690	16.974	0.352	19.485	0.151	1.669	7.112	0.204	0.810	0.941	0.981
(IV)	w/o $\mathcal{L}_{recon}$	0.189	0.693	17.015	0.337	**15.690**	**0.112**	**1.163**	**6.057**	**0.165**	**0.868**	**0.960**	**0.988**
(V)	w/o $\mathcal{L}_{VGG}$	**0.172**	**0.709**	**17.364**	0.427	18.756	0.136	1.480	6.833	0.196	0.824	0.944	0.982
(VI)	w/o $\mathcal{L}_{photo}$	0.180	0.698	17.177	0.338	16.204	0.117	1.361	6.326	0.170	0.860	0.958	0.985
(VII)	w/o $\mathcal{L}_{smooth}$	0.183	0.694	17.064	0.341	16.189	0.117	1.314	6.266	0.170	0.861	0.958	0.986
(VIII)	w/o $\mathcal{L}_{consistency}$	0.179	0.701	17.160	**0.334**	15.943	0.117	1.260	6.148	0.167	0.863	0.959	0.987

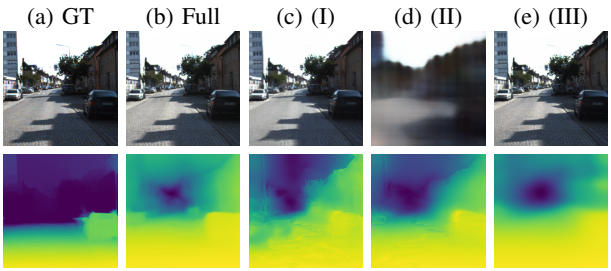

Fig. 5: **Qualitative results on KITTI – ablation study.** From left to right: (a) ground truth target image (top) and ground truth depth (bottom), followed by (b-e) results achieved by configurations (I), (II) and (III) from Table III.

reduce the supervision given to NVS. Consequently, depth metrics improve at the expense of visual fidelity due to discoloration and artifacts. Removing the contribution of $\mathcal{L}_{VGG}$ (V), the network no longer explicitly optimizes for perceptual similarity in generated images, resulting in fuzzier outputs. However, surprisingly, this leads to an improvement in structural similarity, possibly due to the spreading of otherwise small artifacts in the sky of KITTI images. To conclude, we can notice how removing each one of the remaining terms independently (VI,VII,VIII) has minor consequences compared to the removal of $\mathcal{L}_{recon}$ and $\mathcal{L}_{VGG}$.

V. Conclusions

In this paper, we proposed a new pipeline for source-to-target novel view synthesis. Given a source image and a target viewpoint, our framework can generate a novel frame from this latter. Being our model explicitly trained to learn for both the image synthesis and depth estimation tasks, it achieves superior results both in terms of novel generated images as well as predicting their corresponding depth maps with respect to existing approaches for source-to-target view synthesis. Future research directions will aim at handling independently moving objects between the source and target images, making our approach suitable for novel view synthesis on unconstrained video sequences as well.

References

[1] Alex Yu and Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks, 2021.

[2] Kara-Ali Aliev, Dmitry Ulyanov, and Victor S. Lempitsky. Neural point-based graphics. *ArXiv*, abs/1906.08240, 2020.

[3] Zuria Bauer, Zuoyue Li, Sergio Orts-Escolano, Miguel Cazorla, Marc Pollefeys, and Martin R. Oswald. NVS-MonoDepth: Improving monocular depth prediction with novel view synthesis. In *2021 International Conference on 3D Vision (3DV)*. IEEE, dec 2021.

[4] Ang Cao, C. Rockwell, and Justin Johnson. Fwd: Real-time novel view synthesis with forward warping and depth. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15692–15703, 2022.

[5] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, L. Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

[6] Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph.*, 32:30:1–30:12, 2013.

[7] 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. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14104–14113, 2021.

[8] Xu Chen, Jie Song, and Otmar Hilliges. Monocular neural image based rendering with continuous view control. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4089–4099, 2019.

[9] Inchang Choi, Orazio Gallo, Alejandro J. Troccoli, Min H. Kim, and Jan Kautz. Extreme view synthesis. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7780–7789, 2019.

[10] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.

[11] Paul E. Debevec, Camillo Jose Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[13] Andrew W. Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63:141–151, 2003.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017.

[16] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3827–3837, 2019.

[17] Pengsheng Guo, Miguel Ángel Bautista, Alex Colburn, Liang Yang, Daniel Ulbricht, Joshua M. Susskind, and Qi Shan. Fast and explicit neural view synthesis. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 11–20, 2022.

[18] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37:1 – 15, 2018.

[19] Geoffrey E. Hinton, Alex Krizhevsky, and Sida I. Wang. Transforming auto-encoders. In *ICANN*, 2011.

[20] Yuxin Hou, Arno Solin, and Juho Kannala. Novel view synthesis via depth-guided skip connections. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3119–3128, January 2021.

[21] Ronghang Hu and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12508–12517, 2021.

[22] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[23] Shubhendu Jena, F. Multon, and Adnane Boukhayma. Neural mesh-based graphics. *ArXiv*, abs/2208.05785, 2022.

[24] Johannes Kopf, Fabian Langguth, Daniel Scharstein, Richard Szeliski, and Michael Goesele. Image-based rendering in the gradient domain. *ACM Transactions on Graphics (TOG)*, 32:1 – 9, 2013.

[25] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014.

[26] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Geometry-aware deep network for single-image novel view synthesis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4616–4624, 2018.

[27] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien P. C. Valentin, S. Khamis, Philip L. Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B. Goldman, Cem Keskin, Steven M. Seitz, Shahram Izadi, and S. Fanello. Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37:255, 2018.

[28] Moustafa Meshry, Dan B. Goldman, S. Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6871–6880, 2019.

[29] 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 *ECCV*, 2020.

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv:2201.05989*, January 2022.

[31] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (TOG)*, 38:1 – 15, 2019.

[32] David Novotný, Benjamin Graham, and Jeremy Reizenstein. Perspectivenet: A scene-consistent image generator for new view synthesis in real indoor environments. In *NeurIPS*, 2019.

[33] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 702–711, 2017.

[34] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36:1 – 11, 2017.

[35] Konstantinos Rematas, Chuong H. Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Novel views of objects from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1576–1590, 2017.

[36] Gernot Riegler and Vladlen Koltun. Free view synthesis. *ArXiv*, abs/2008.05511, 2020.

[37] Gernot Riegler and Vladlen Koltun. Stable view synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12211–12220, 2021.

[38] C. Rockwell, David F. Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14084–14093, 2021.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[40] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[41] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1:519–528, 2006.

[42] Jonathan Shade, Steven J. Gortler, Li wei He, and Richard Szeliski. Layered depth images. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998.

[43] Yujiao Shi, Hongdong Li, and Xin Yu. Self-supervised visibility learning for novel view synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9670–9679, 2021.

[44] Daeyun Shin, Zhile Ren, Erik B. Sudderth, and Charless C. Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2172–2182, 2019.

[45] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2441, 2019.

[46] Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4d rgbd light field from a single image. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2262–2270, 2017.

[47] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021.

[48] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *ECCV*, 2018.

[49] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15162–15172, 2021.

[50] Satoshi Tsutsui, Weijia Mao, Sijing Lin, Yunyi Zhu, Murong Ma, and Mike Zheng Shou. Novel view synthesis for high-fidelity headshot scenes. *ArXiv*, abs/2205.15595, 2022.

[51] Shubham Tulsiani, Saurabh Gupta, David F. Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 302–310, 2018.

[52] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4688–4697, 2021.

[53] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

[54] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475, 2020.

[55] Xin Yang, Jingyu Chen, Yuanjie Dang, Hongcheng Luo, Yuesheng Tang, Chunyuan Liao, Peng Chen, and Kwang-Ting Cheng. Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):156–167, 2021.

[56] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4585, 2021.

[57] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[58] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ArXiv*, abs/1805.09817, 2018.

[59] Hao Zhu, Hao Su, Peng Wang, Xun Cao, and Ruigang Yang. View extrapolation of human body from a single image. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2018.