Monocular Neural Image Based Rendering with Continuous View Control

Xu Chen* Jie Song* Otmar Hilliges
AIT Lab, ETH Zurich
{xuchen, jsong, otmar.hilliges}@inf.ethz.ch

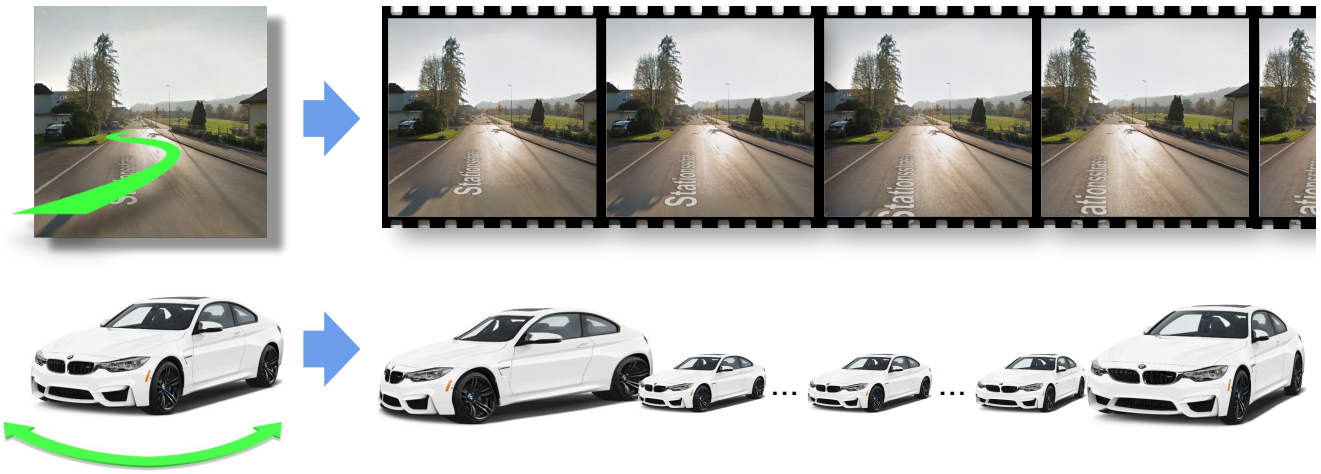

Figure 1: **Interactive novel view synthesis:** Given a single source view our approach can generate a continuous sequence of geometrically accurate novel views under fine-grained control. *Top*: Given a single street-view like input, a user may specify a continuous camera trajectory and our system generates the corresponding views in real-time. *Bottom*: An unseen hi-res internet image is used to synthesize novel views, while the camera is controlled interactively. *Please refer to our project homepage*[†].

Abstract

We propose a method to produce a continuous stream of novel views under fine-grained (e.g., 1° step-size) camera control at interactive rates. A novel learning pipeline determines the output pixels directly from the source color. Injecting geometric transformations, including perspective projection, 3D rotation and translation into the network forces implicit reasoning about the underlying geometry. The latent 3D geometry representation is compact and meaningful under 3D transformation, being able to produce geometrically accurate views for both single objects and natural scenes. Our experiments show that both proposed components, the transforming encoder-decoder and depth-guided appearance mapping, lead to significantly improved generalization beyond the training views and in consequence to more accurate view synthesis under continuous 6-DoF camera control. Finally, we show that our method outperforms state-of-the-art baseline methods on public datasets.

1. Introduction

3D immersive experiences can benefit many application scenarios. For example, in an online store one would often like to view products interactively in 3D rather than from discrete view angles. Likewise in map applications it is desirable to explore the vicinity of street-view like images beyond the position at which the photograph was taken. This is often not possible because either only 2D imagery exists, or because storing and rendering of full 3D information does not scale. To overcome this limitation we study the problem of interactive view synthesis with 6-DoF view control, taking only a single image as input. We propose a method that can produce a continuous stream of novel views under fine-grained (e.g., 1° step-size) camera control (see Fig. 1).

Producing a continuous stream of novel views in *real-time* is a challenging task. To be able to synthesize high-quality images one needs to reason about the underlying geometry. However, with only a monocular image as input the task of 3D reconstruction is severely ill-posed. Traditional image-based rendering techniques do not apply to the real-time monocular setting since they rely on multiple input views and also can be computationally expensive.

Recent work has demonstrated the potential of learning to predict novel views from monocular inputs by leveraging a training set of viewpoint pairs [52, 62, 40, 8]. This

*Equal contribution.

[†]https://ait.ethz.ch/projects/2019/cont-view-synth/

is achieved either by directly synthesizing the pixels in the target view [52, 8] or predicting flow maps to warp the input pixels to the output [62, 51]. However, we experimentally show that such approaches are prone to over-fitting to the training views and do not generalize well to free-from non-training viewpoints. If the camera is moved continuously in small increments, with such methods, the image quality quickly degrades. One possible solution is to incorporate much denser training pairs but this is not practical for many real applications. Explicit integration of geometry representations such as meshes [26, 34] or voxel grids [58, 6, 16, 54] could be leveraged for view synthesis. However, such representations would limit applicability to settings where the camera orbits a single object.

In this paper, we propose a novel learning pipeline that determines the output pixels directly from the source color but forces the network to implicitly reason about the underlying geometry. This is achieved by injecting geometric transformations, including perspective projection, 3D rotations and translations into an end-to-end trainable network. The latent 3D geometry representation is compact and memory efficient, is meaningful under explicit 3D transformation and can be used to produce geometrically accurate views for both single objects and natural scenes.

More specifically, we propose a geometry aware neural architecture consisting of a 3D transforming autoencoder (TAE) network [21] and subsequent depth-guided appearance warping. In contrast to existing work, that directly concatenate view point parameters with latent codes, we first encode the image into a latent representation which is explicitly rotated and translated in Euclidean space. We then decode the transformed latent code, which is assumed to implicitly represent the 3D geometry, into a depth map in target view. From the depth map we compute dense correspondences between pixels in the source and target view via perspective projection and subsequently the final output image via pixel warping. All operations involved are differentiable, allowing for end-to-end training.

Detailed experiments are performed on synthetic objects [3] and natural images [15]. We assess the image quality, granularity, precision of continuous viewpoint control and implicit recovery of scene geometry qualitatively and quantitatively. Our experiments demonstrate that both components, the TAE and depth-guided warping, drastically improve the robustness and accuracy for continuous view synthesis.

In conclusion, our main contributions are:

- We propose the task of continuous view synthesis from monocular inputs under fine-grained view control.
- This goal is achieved via a proposed novel architecture that integrates a transforming encoder-decoder network and depth-guided image mapping.
- Thorough experiments are conducted, demonstrating the efficacy of our method compared to prior art.

2. Related Work

View synthesis with multi-view images. The task of synthesizing new views given a sequence of images as input has been studied intensely in both the vision and graphics community. Strategies can be classified into those that explicitly compute a 3D representation of the scene [42, 28, 41, 7, 47, 46, 65, 4, 30], and those in which the 3D geometry is handled implicitly [12, 35, 36]. Others have deployed full 4D light fields [18, 31], albeit at the cost of complex hardware setups and increased computational cost. Recently, deep learning techniques have been applied in similar settings to fill holes and eliminate artifacts caused by the sampling gap, dis-occlusions, and inaccurate 3D reconstructions [14, 19, 61, 55, 49, 13, 37]. While improving results over traditional methods, such approaches rely on multi-view input and are hence limited to the same setting.

View synthesis with monocular input. Recent work leverages deep neural networks to learn a monocular image-to-image mapping between source and target view from data [29, 52, 8, 62, 40, 51, 59]. One line of work [29, 52, 8, 39] directly generates image pixels. Given the difficulty of the task, direct image-to-image translation approaches struggle with preservation of local details and often produce blurry images. Zhou et.al. [62] estimate flow maps in order to warp source view pixels to their location in the output. Others further refine the results by image completion [40] or by fusing multiple views [51].

Typically, the desired view is controlled by concatenating latent codes with a flattened viewpoint transform. However, the exact mapping between viewpoint parameters to images is difficult to learn due to sparse training pairs from the continuous viewpoint space. We show experimentally that this leads to a snapping to training views, with image quality quickly degrading under continuous view control. Recent works demonstrate the potential for fine-grained view synthesis, but either are limited to single instances of objects [48] or require additional supervision in the form of depth maps [63, 33], surface normals [33] and even light field images [50], which are cumbersome to acquire in real settings. In contrast, our method consists of a fully differentiable network, which is trained with image pairs and associated transformations as sole supervision.

3D from single image. Reasoning about the 3D shape can serve as an implicit step of free-from view synthesis. Given the severely under-constrained case of recovering 3D shapes from a single image, recent works have deployed neural networks for this task. They can be categorized by their output representation into mesh [26, 34], point cloud [11, 32, 23], voxel [58, 6, 16, 54, 44], or depth map based [9, 60, 53]. Mesh-based approaches are still not accurate enough due to the indirect learning process. Point clouds are often sparse and cannot be directly leveraged to project dense color information in the output image and voxel-based methods are

limited in resolution and number and type of objects due to memory constraints. Depth maps become sparse and incomplete when projected into other views due to the sampling gap and occlusions. Layered depth map representations [53] have been used to alleviate this problem. However, a large number of layers would be necessary which poses significant hurdles in terms of scalability and runtime efficiency. In contrast to explicit models, our latent 3D geometry representation is compact and memory efficient, is meaningful under explicit 3D transformation and can be used to render dense images.

Deep generative models. View synthesis can also be seen as an image generation process, which is related to the field of deep generative modelling of images [27, 17]. Recent models [2, 25] are able to generate high-fidelity images with diversity in many aspects including viewpoint, shape and appearance, but offer little to no exact control over the underlying parameters. Disentangling latent factors has been studied in [5, 20] to provide control over image attributes. In particular, recent work [64, 38] demonstrates inspiring results of viewpoint disentanglement by reasoning about the geometry. Although such methods can be used for view synthesis, the generated views lack consistency and moreover one cannot control which object to synthesize.

3. Method

Our main contribution is a novel geometry aware network design, shown in Fig. 2, that consists of four components: 3D transforming auto-encoder (TAE), self-supervised depth map prediction, depth map projection and appearance warping.

The source view is first encoded into a latent code ($z = E_{\theta_e}(I_s)$). This latent code z is encouraged by our learning scheme to be meaningful in 3D metric space. After encoding we apply the desired transformation between the source and target to the latent code. The transformed code ($z_T = T_{s\to t}(z)$) is decoded by a neural network to predict a depth map D_t as observed from the target viewpoint. D_t is projected back into the source view based on the known camera intrinsics K and extrinsics $T_{s\to t}$, yielding dense correspondences between the target and source views, encoded as dense backward flow map $C_{t\to s}$. This flow map is used to warp the source view pixel-by-pixel into the target view.

Note that attaining backward flow and hence predicting depth maps in the *target* view is a crucial difference to prior work. Forward mapping of pixel values into the target view I_t would incur discretization artifacts when moving between ray and pixel-space, visible as banding after re-projection of the (source view) depth map. The whole network is trained end-to-end with a simple per-pixel reconstruction loss as sole guidance. Overall, we want to learn a mapping $M : X \to Y$, which in our case can be decomposed as:

$$M(I_s) = B(P_{t\to s}(D_{\theta_d}(T_{s\to t}(E_{\theta_e}(I_s)))), I_s) = \hat{I}_t, \quad (1)$$

where B is the bi-linear sampling function, $P_{t\to s}$ is the perspective projection, and $E_{\theta_e}, D_{\theta_d}$ are the encoder and decoder networks respectively. This decomposition is an important contribution of our work. By asking the network to predict a depth map D_t in the target view, we implicitly encourage the TAE encoder E_{θ_e} to produce position predictions for features and the decoder D_{θ_d} learns to generate features at corresponding positions by rendering the transformed representation from the specified view-angle.

3.1. Transforming Auto-encoder

We take inspiration from recent work [45, 22, 57, 43] which itself builds upon earlier work by Hinton et al. [21], that uses encoder-decoder architectures to learn representations that are transformation equivariant, establishing a direct correspondence between image and feature spaces. We leverage such a latent space to model the relationship between viewpoint and implicit 3D shape.

To this end, we represent the latent code z_s as vectorized set of points $z_s \in \mathbb{R}^{n\times 3}$, where n is a hyper-parameter. This representation is then multiplied with the ground-truth transformation $T_{s\to t} = [R|t]_{s\to t}$ describing the viewpoint change between source view I_s and target view I_t to attain the rotated code z_t:

$$z_t = [R|t]_{s\to t} \cdot \tilde{z_s}, \quad (2)$$

where $\tilde{z}_s$ is the homogeneous representation of z_s. In this way the network is trained to encode position predictions for features which can then be decoded into images. All functions in the TAE module including encoding, vector reshaping, matrix multiplication and decoding are differentiable and hence amenable to training via backpropagation.

3.2. Depth Guided Appearance Mapping

We decode z_t into 3D shape in the target view, represented as a depth image D_t. From D_t we compute the dense correspondence field $C_{t\to s}$ deterministically via perspective projection $P_{t\to s}$. The dense correspondences are then used to warp the pixels of the texture (source view) I_s into the target view $\hat{I}_t$. This allows the network to warp the source view into the target view and makes the prediction of target view invariant to the texture of the input, resulting in sharp and detail-preserving outputs.

Establishing correspondences. The per-pixel correspondences $C_{t\to s}$ are attained from the depth image D_t in the target view by conversion from the depth map to 3D coordinates $[X, Y, Z]$ and perspective projection:

$$[X, Y, Z]^T = D_t(x_t, y_t)K^{-1}[x_t, y_t, 1]^T \quad (3)$$

$$\text{and} \quad [x_s, y_s, 1]^T \sim KT_{t\to s}[X, Y, Z, 1]^T \quad . \quad (4)$$

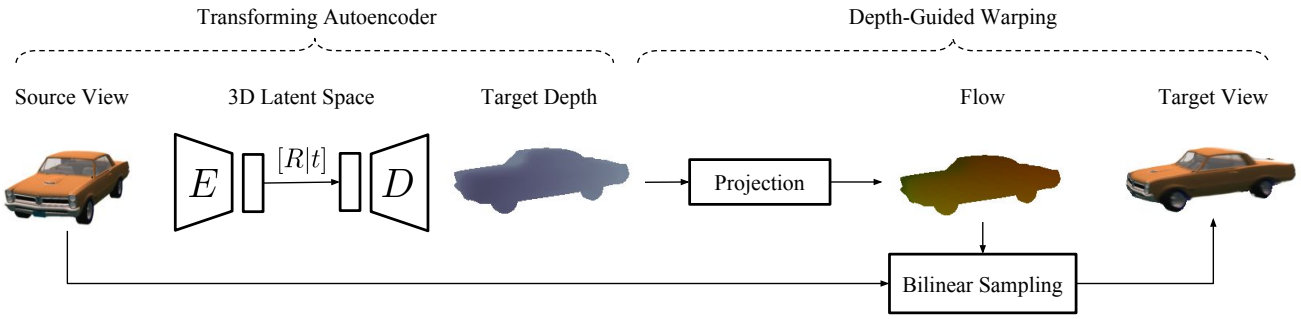

Figure 2: **Pipeline overview.** 2D source views are encoded and the latent code is explicitly rotated before a decoder network predicts the depth map in the target view. Dense correspondences are attained via perspective projection and used to warp pixels from source view to the target with bilinear sampling. All operations are differentiable and trained end-to-end without ground-truth depth or flow maps. The only supervision is a L_1 reconstruction loss between target view and ground truth image.

where each pixel (x_t, y_t) encodes the corresponding pixel position in the source view (x_s, y_s). Furthermore, K is the camera intrinsic matrix describing normalized focal length along both axes f_x, f_y and image center c_x, c_y. Note that only the focal length ratio f_x/f_y as well as image center affect view synthesis, while the absolute scale of the focal length is only important to predict geometry at correct scale.

Warping with correspondences. With the dense correspondences obtained, we are now able to warp the source view to the target view. This operation propagates texture and local details. Since the corresponding pixel positions that are derived from Eq. 4 are non-integer, this is done via differentiable bilinear sampling as proposed in [24]:

$$I_t(x_t, y_t) = \sum_{x_s} \sum_{y_s} I_s(x_s, y_s) \max(0, 1 - |x_s - C_x(x_t, y_t)|) \max(0, 1 - |y_s - C_y(x_t, y_t)|). \quad (5)$$

The use of backward flow $C_{t\to s}$, computed from the predicted depth map D_t, makes the approach amenable to gradient based optimization since the gradient of the per-pixel reconstruction loss provides meaningful information to correct erroneous correspondences. The gradients also flow back to provide useful information to the TAE network owing to the fact that the correspondences are computed deterministically from the predicted depth maps. While bearing similarity to [62], we introduce the intermediate step of predicting depth, instead of predicting the correspondences directly. This enforces the network to obey geometric constraints, resolving ambiguous correspondences.

3.3. Training

All steps in our network, namely 3D transforming autoencoder (TAE), self-supervised depth map prediction, depth map projection and appearance warping, are differentiable which enables end-to-end training. Among all modules, only the TAE module contains trainable parameters (θ_e, θ_d). To train the network only pairs of source and target views and their transformation are required. The network weights are optimized via minimization of the L_1 loss between the predicted target view $\hat{I}_t$ and the ground truth I_t.

$$\mathcal{L}_{recon} = \left\| I_t - \hat{I}_t \right\|_1 \quad (6)$$

Minimizing this reconstruction loss, the network learns to produce realistic novel views, to predict the necessary flow and depth maps and learn to form a geometrical latent space.

4. Experiments

We now evaluate our method quantitatively and qualitatively. We are especially interested in assessing image quality, granularity and precision of fine-grained viewpoint control. First, we conduct detailed experiments on synthetic objects, where ground-truth of continuous viewpoint is easy to obtain, to numerically assess the reconstruction quality. Notably, we vary the viewpoints in much smaller step-sizes than what is observed in the training data. Second, to evaluate generalizability, we test our system on natural city scenes. In this setting, given an image input, we specify the desired ground-truth camera trajectories along which the system generates novel views. Then we run an existing visual odometry system on these synthesized continuous views to recover the camera trajectory. By comparing the recovered trajectory with the ground-truth, we can evaluate the geometrical property of the synthesized images under the consideration of granularity and continuous view control. Finally, to better understand the mechanism of our proposed network, we further conduct studies on its two key components, namely depth-guided texture mapping and transforming auto-encoder. We evaluate the intermediate depth and flow, and qualitatively verify the meaningfulness of the latent space of the TAE.

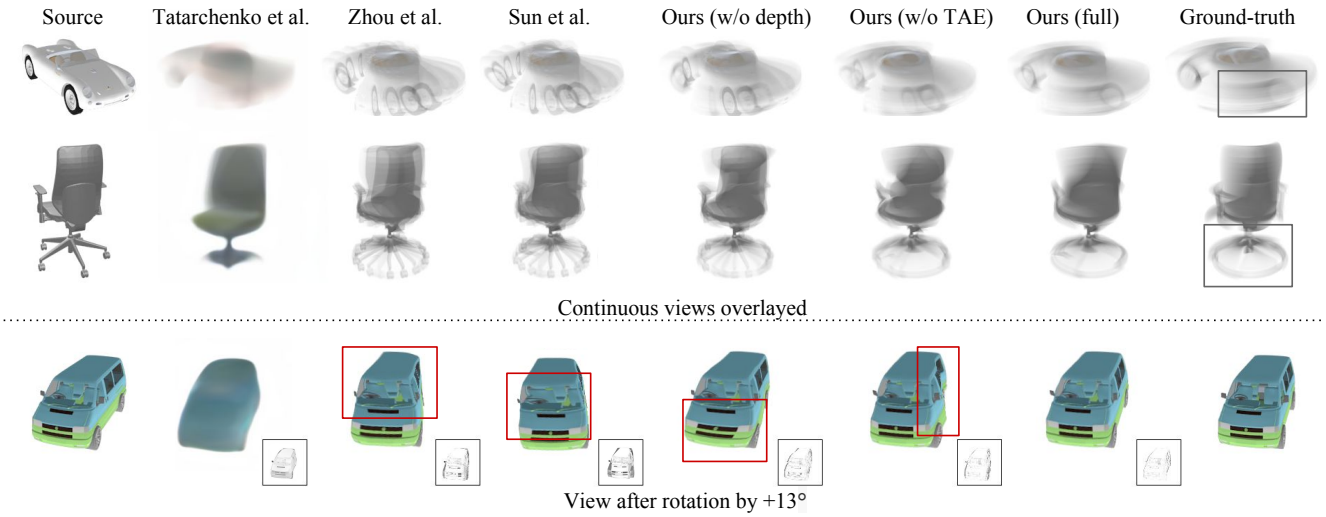

Figure 3: **Qualitative results for granularity and precision of viewpoint control on ShapeNet.** In the top two rows, we generate and overlay 80 continuous views with step size of 1° from a single input. Our method exhibits similar spin pattern as the ground truth, whereas other methods mostly converge to the fixed training views (see wheels of the car and chair indicated by the box). In the bottom row, a close look at specific views is given, which reveals that previous methods display distortions or converge to neighboring training views (Zhou et al.[62], Sun et al.[51]). The image generated by Tatarchenko et al.[52] is blurry. Corresponding error maps are also depicted. *Best viewed in color.*

4.1. Datasets

We conduct our experiments on two challenging datasets: synthetic objects [3] and real natural scenes [15].

ShapeNet [3] is a large collection of 3D synthetic objects from various categories. Similar to [62, 40, 51] we choose **car** and **chair** to evaluate our method. We use the same train test split as proposed in [62]. For training we render each models from 54 viewpoints with different azimuth and elevation. The azimuth goes from 0° to 360° with a step size of 20° and the elevation from 0° to 30° with a step size of 10°. Each training pair consists of two views of the same instance, with a difference in azimuth within ±40°.

KITTI [15] is a standard dataset for autonomous driving, containing complex city scenes in uncontrolled environments. We conduct experiments on the KITTI odometry subset which contains image sequences as well as the global camera poses of each frame. In total there are 18560 images for training and 4641 images for testing. We construct training pairs by randomly selecting target view among 10 nearest frames of source view. The relative transformation is obtained from the global camera poses.

4.2. Metrics

In our evaluations we report the following metrics:

Mean Absolute Error L_1 is used to measure per-pixel value differences between ground-truth and the predictions.

Structural SIMilarity (SSIM) Index[56] has values in [-1, 1] and measures the structural similarity between synthesized image and ground truth. We report SSIM in addition to the L_1 loss since it i) gives an indication of perceptual image quality and ii) serves as further metric that is not directly optimized during training.

Percentage of correctness under threshold δ (Acc). The predicted flow/depth $\hat{y_i}$ at pixel i, given ground truth y_i, is regarded as correct if $max(\frac{y_i}{\hat{y_i}}, \frac{\hat{y_i}}{y_i}) < \delta$ is satisfied. We count the portion of correctly predicted pixels. Here $\delta = 1.05$.

Rotation error and translation error are defined as:

$$RE = arccos(\frac{\mathbf{Tr}(\tilde{R} \cdot R^T) - 1}{2}), TE = arccos(\frac{\tilde{t} \cdot t^T}{\|\tilde{t}\|_2 \cdot \|t\|_2}) \quad (7)$$

where **Tr** represents the trace of the matrix.

4.3. Comparison with other methods

We compare with several representative state-of-the-art learning-based view synthesis methods. **Tatarchenko et al.** [52] treat the view synthesis as an image-to-image translation task and generate pixels directly. In their framework the viewpoint is directly concatenated with the latent code. **Zhou et al.** [62] generates flow instead of pixels. The view information is also directly concatenated. **Sun et al.** [51] combines both pixel generation [52] and image warping [62]. The original implementation in Zhou et al. [62] and Sun et al. [51] does not support continuous viewpoint input for objects. To allow for continuous input for comparison, we replace their encoded discrete one hot viewpoint represen-

tation with cosine and sine values of the view angles. The same encoder and decoder are used for all comparisons.

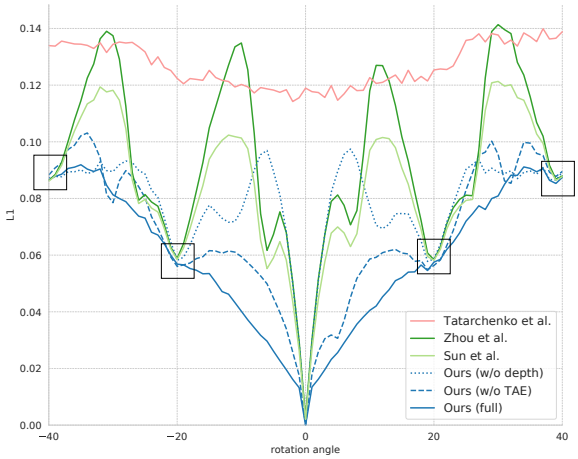

Figure 4: Comparison of L_1 **reconstruction error** as a function of view rotation on **car**. Ours outperforms other state-of-the-art baselines over the entire range and yields a smoother loss progression. Note that $0°$ here means no transformation applied to the source view. ($\pm 40°, \pm 20°$ are training views indicated by black boxes).

4.4. ShapeNet Evaluation

To test the granularity and precision of viewpoint control, for each test object, given a source view I_s, the network synthesizes 80 views around the source view with a step size of $1°$ which is much denser than the step size of $20°$ for training (and much denser than previously reported experiments). In total the test set contains 100,000 view pairs of objects.

To study the effectiveness of the transformation-aware latent space, we introduce **Ours (w/o TAE)** concatenating the viewpoint analogously to [52, 8, 62, 40, 62] while still keeping the depth-guided texture mapping process. To evaluate the depth-guided texture mapping process, we introduce **Ours (w/o depth)** which directly predicts flow without the depth guidance but does deploy the TAE.

Viewpoint dependent error. Fig. 4 plots the L1 reconstruction error between $[-40°, 40°]$ of all methods. Note that $0°$ here means no transformation applied to the source view. *Ours* consistently produces lower errors. More importantly it yields much lower variance between non-training and training views ($\pm 40°, \pm 20°$ are training views). While previous methods can achieve similar performance to ours at training views, their performance significantly decreases for non-training views. Notably, both of our designs (TAE and depth-based appearance) contribute to the final performance and the problem of snapping to training views persists with either of the two components discarded (**Ours (w/o TAE)** and **Ours (w/o depth)**). Tab. 1 summarizes the average L_1 error and SSIM for all generated views between $[-40°, 40°]$. Inline with Fig. 4, our method significantly outperforms previous methods on both car and chair. In addition, both of our ablative methods also perform better than previous methods, demonstrating the effectiveness of both modules.

	Car		Chair	
	L1	SSIM	L1	SSIM
Tatarchenko et al. [52]	0.084	0.919	0.110	0.917
Zhou et al. [62]	0.062	0.924	0.074	0.920
Sun et al. [51]	0.056	0.926	0.070	0.921
Ours (w/o depth)	0.052	0.932	0.066	0.926
Ours (w/o TAE)	0.045	0.943	0.065	0.930
Ours (full)	**0.039**	**0.949**	**0.056**	**0.938**

Table 1: **Quantitative analysis of fine-grained view control on ShapeNet.** Average L1 error and SSIM for all generated views between $[-40°, 40°]$ from the source view.

Qualitative results. The qualitative results in Fig. 3 confirm the quantitative findings. To demonstrate the capability of continuous viewpoint control, we generate and overlay 80 views with step size of $1°$ from a single input. Compared to previous approaches, our method exhibits similar spin pattern as the ground truth, whereas other methods mostly snap to the fixed training views (Zhou et al. [62], Sun et al. [51]). This suggests that overfitting occurs, limiting the granularity and precision of view control. A close look at specific views reveals that previous methods display distortions at non-training views, highlighted in red. The image generated by Tatarchenko et al. [52] is blurry.

4.5. KITTI Evaluation

We now evaluate our method in the more realistic setting of the KITTI dataset. Note that the dataset only contains fairly linear forward motion recorded from a car's dash. This setting is a good testbed for the envisioned application scenarios where one desires to extract 3D information retroactively.

Qualitative results In Fig. 5 we show qualitative results from novel views synthesized along a straight camera trajectory: Zhou et al. [62] and Sun et al. [51] both have difficulties to deal with viewpoints outside of the training setting and produce distorted images while ours are sharp and geometrically correct. Ours more faithfully reproduces the desired motion than [62] and [51] which remains stationary.

Complex trajectory recovery. To simulate real use cases, we introduce a new experimental setting. We specify arbitrary *desired* trajectories, specifically so that the camera moves away from the car's original motion. From this specification we generate a sequences of 100 images along the trajectories. Subsequently we run a state-of-the-art visual odometry [10] system to estimate the camera pose based on the *synthesized* views. If the view synthesis approach is

Figure 5: **Simple camera motion.** Setting: Given a source view we synthesize linear forward motion over 0.6m. Our method produce sharp and correct images while [62, 51] produces distorted images. Zhou et al. [62]'s motion is incorrect, while Sun et al. [51] stays stationary. Ours reflects a reasonable straight forward transition.

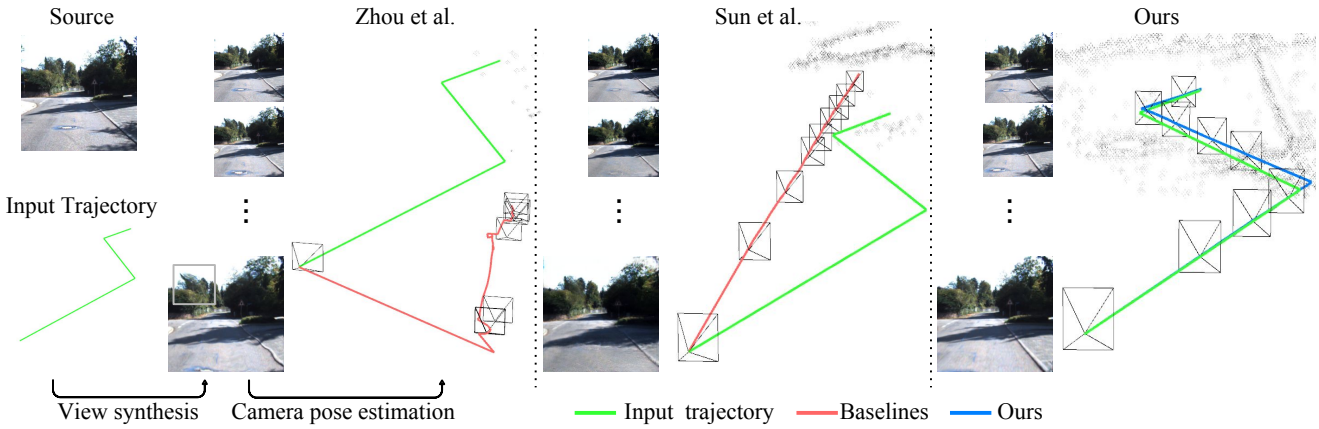

Figure 6: **Complex camera motion.** Setting: given a source view and an input trajectory, a continuous sequence of views is synthesized along the user defined trajectory (green). Trajectories are estimated via a state-of-the-art visual odometry system [10] and compared to the desired trajectory. The trajectory estimated from *Ours* align well with the ground-truth, while [62, 51] mostly produce straight forward or wrong motion regardless of the input.

geometrically accurate, the visual odometry system should recover the *desired* trajectory. Fig. 6 illustrates one such experiment. The estimated trajectory from ours aligns well with the ground-truth. In contrast, views from [62] result in a wrong trajectory and [51] mostly produce straight forward motion, possibly due to overfitting to training trajectories.

Quantitative results. To evaluate the geometrical properties quantitatively, we generate new views with randomly sampled transformation $T = [R|t]$. We then estimate the relative transformation between the input and the synthesized view $\tilde{T} = [\tilde{R}|\tilde{t}]$ and compare to the ground-truth T. This is done by first detecting and matching SURF features [1] in both views, and then computing and decomposing the essential matrix. We report the numerical error in Tab. 2. Our method produces drastically lower error in rotation and the translation, indicating accurate viewpoint control. Note that we had to remove [52] from this comparison since SURF feature detection fails due to the very blurry images.

	TE	RE
Zhou et al. [62]	0.557	0.086
Sun et al. [51]	0.435	0.080
Ours	**0.108**	**0.019**

Table 2: **Precision evaluation** of viewpoint control by camera pose estimation on KIITI.

4.6. Depth and Flow Evaluation

The quality of predicted depth map and warping flow is essential to produce geometrically correct views. We evaluate the accuracy of depth and flow prediction with two metrics (L_1 and Acc). Tab. 3 summarizes results for ShapeNet. Ours achieves the best accuracy in both flow and depth prediction, which directly benefits view synthesis (cf. Tab. 1). The relative ranking of the ablative baselines furthermore indicates that both the TAE and the depth-guided texture mapping help to improve the flow accuracy. The TAE furthermore guides the depth prediction. To illustrate that the reconstructed depth maps are indeed meaningful, we predict depth in different target views and visualize the extracted normal maps, as shown in Fig. 7.

Discussion Together these experiments indicate that the proposed self-supervision indeed forces the network to infer underlying 3D structure (yielding good depth which is necessary for accurate flow maps) and that it helps the final task without requiring additional labels.

	Flow		Depth	
	L1	Acc	L1	Acc
Zhou et al. [62]	0.035	69.1%	-	-
Ours (w/o depth)	0.029	76.3%	-	-
Ours (w/o TAE)	0.022	84.6%	0.134	89.0%
Ours (full)	**0.021**	**85.7%**	**0.132**	**91.1%**

Table 3: **Quantitative analysis of flow and depth prediction on car.** Average L_1 error and accuracy for all predicted flow and depth in target views between $[-40°, 40°]$ from the source view. Ours significantly outperforms the baselines.

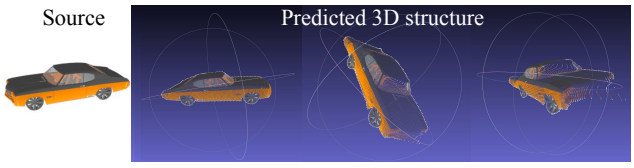

Figure 7: **Unsupervised depth prediction.** Depth map is predicted from the source view and visualized as point clouds depicted from different viewing angles.

4.7. Latent Space Analysis

To verify that the learned latent space is indeed interpretable and meaningful under geometrical transformation, we i) linearly interpolate between latent points of two objects and ii) rotate each interpolated latent point set. These point sets are then decoded into depth maps, visualized as normal maps in the global frame. Fig. 8 shows that interpolated samples exhibit a smooth shape transition while the viewpoint remains constant (i). Moreover, rotating the latent points only changes the viewpoint without affecting the shape (ii).

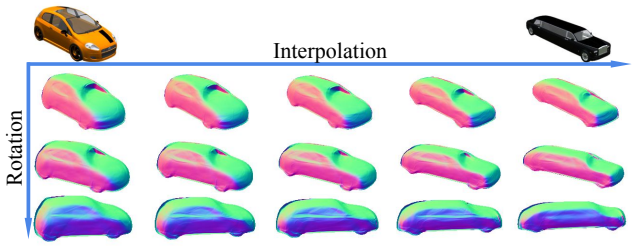

Figure 8: **Latent space analysis showing consistency of embeddings**. Left-to-right: latent space interpolation between *different* objects. Top-to-bottom: Rotation of *same* latent code. (Normals in global frame, extracted from depth).

4.8. Generalization to unseen data

We find that our model generalizes well to unseen data thanks to the usage of depth-based warping. Interestingly, our model trained on 256^2 images can be directly applied to high resolution (1024^2) images without additional training. The inference process takes 50ms per frame on a Titan X GPU, allowing for real time rendering of synthetized views. This enables many appealing application scenarios. For example, our model, trained on ShapeNet only, can be used in an app where downloaded 2D images are brought to life and a user may browse the depicted object in 3D. With a model trained on KITTI, a user may explore a 3D scene from a single image, via generation of free-viewpoint videos or AR/VR content (see Fig. 1).

5. Conclusion

We have presented a novel learning pipeline for continuous view synthesis. At its core lies a depth-based image prediction network that is forced to satisfy explicitly formulated geometric constraints. The latent representation is meaningful under explicit 3D transformation and can be used to produce geometrically accurate views for both single objects and natural scenes. We have conducted thorough experiments on synthetic and natural images and have demonstrated the efficacy of our approach.

Acknowledgement. We thank Nvidia for the donation of GPUs used in this work. We would like to express our gratitude to Olivier Saurer, Velko Vechev, Manuel Kaufmann, Adrian Spurr, Yinhao Huang, Xucong Zhang and David Lindlbauer for the insightful discussions, James Bern and Seonwook Park for the video voice-over.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006. 7

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 3

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 5

[4] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3):30, 2013. 2

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 3

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 2

[7] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Conference on Computer Graphics and Interactive Techniques*, pages 11–20. ACM, 1996. 2

[8] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2016. 1, 2, 6

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2

[10] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 6, 7

[11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[12] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005. 2

[13] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[14] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready autonomous driving? the kitti vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 5

[16] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 2

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3

[18] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Conference on Computer Graphics and Interactive Techniques*, volume 96, pages 43–54, 1996. 2

[19] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. In *SIGGRAPH Asia 2018 Technical Papers*, page 257. ACM, 2018. 2

[20] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017. 3

[21] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, 2011. 2, 3

[22] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 3

[23] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2

[24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 4

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[26] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2013. 3

[28] Johannes Kopf, Fabian Langguth, Daniel Scharstein, Richard Szeliski, and Michael Goesele. Image-based rendering in the gradient domain. *ACM Transactions on Graphics (TOG)*, 32(6):199, 2013. 2

[29] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics net-

work. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[30] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[31] Marc Levoy and Pat Hanrahan. Light field rendering. In *Conference on Computer Graphics and Interactive Techniques*, pages 31–42. ACM, 1996. 2

[32] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[33] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Geometry-aware deep network for single-image novel view synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[34] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *arXiv preprint arXiv:1901.05567*, 2019. 2

[35] Wojciech Matusik, Hanspeter Pfister, Addy Ngan, Paul Beardsley, Remo Ziegler, and Leonard McMillan. Image-based 3d photography using opacity hulls. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 427–437. ACM, 2002. 2

[36] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Conference on Computer Graphics and Interactive Techniques*, pages 39–46. Citeseer, 1995. 2

[37] 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 Transactions on Graphics (TOG)*, 2019. 2

[38] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. *arXiv preprint arXiv:1904.01326*, 2019. 3

[39] Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. Transformable bottleneck networks. *arXiv:1904.06458*, 2019. 2

[40] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5, 6

[41] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):235, 2017. 2

[42] 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(8):1576–1590, 2016. 2

[43] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 3

[44] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[45] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 3

[46] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528. IEEE, 2006. 2

[47] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM. 2

[48] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[49] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[50] Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4d rgbd light field from a single image. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[51] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 2, 5, 6, 7, 8

[52] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *Proc. of the European Conf. on Computer Vision (ECCV)*. Springer, 2016. 1, 2, 5, 6, 7

[53] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 2, 3

[54] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[55] Yunlong Wang, Fei Liu, Zilei Wang, Guangqi Hou, Zhenan Sun, and Tieniu Tan. End-to-end view synthesis for light field imaging with pseudo 4DCNN. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 333–348, 2018. 2

[56] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[57] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Interpretable transformations with encoder-decoder networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 3

[58] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[59] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[60] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[61] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *Conference on Computer Graphics and Interactive Techniques*, 2018. 2

[62] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 1, 2, 4, 5, 6, 7, 8

[63] Hao Zhu, Hao Su, Peng Wang, Xun Cao, and Ruigang Yang. View extrapolation of human body from a single image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[64] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: image generation with disentangled 3d representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 3

[65] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. In *ACM transactions on graphics (TOG)*, volume 23, pages 600–608. ACM, 2004. 2