NVS Machines: Learning Novel View Synthesis with Fine-grained View Control

Xu Chen*  Jie Song*  Otmar Hilliges
AIT Lab, ETH Zurich

Abstract

We present an approach that learns to synthesize high-quality, novel views of 3D objects or scenes, while providing fine-grained and precise control over the 6-DOF viewpoint. The approach is self-supervised and only requires 2D images and associated view transforms for training. Our main contribution is a network architecture that leverages a transforming auto-encoder in combination with a depth-guided warping procedure to predict geometrically accurate unseen views. Leveraging geometric constraints renders direct supervision via depth or flow maps unnecessary. If large parts of the object are occluded in the source view, a purely learning based prior is used to predict the values for dis-occluded pixels. Our network furthermore predicts a per-pixel mask, used to fuse depth-guided and pixel-based predictions. The resulting images reflect the desired 6-DOF transformation and details are preserved. We thoroughly evaluate our architecture on synthetic and real scenes and under fine-grained and fixed-view settings. Finally, we demonstrate that the approach generalizes to entirely unseen images such as product images downloaded from the internet.

1. Introduction

In this paper we study the task of free-form novel view synthesis: given a 2D image we synthesize novel, unseen views of a 3D object or scene with fine-granular control over the camera position and orientation (see Fig. 1). While this is possible but cumbersome with a graphics pipeline, easy-to-use and computationally cheap learning based approaches would enable a number of compelling applications in graphics and computer vision such as free viewpoint image-based rendering, content creation for AR/VR, synthesis of training data for estimation tasks, scene reconstruction and more.

Existing deep learning approaches to predict novel views typically leverage auto-encoder architectures to learn a mapping between paired views to predict target views at inference time [14, 24, 32, 40, 41]. While demonstrating the feasibility of the task, many methods provide no control over the exact viewing angle, only allow for a fixed, small set of views [40, 32], can not always recover scene depth, and sometimes require extensive ground truth annotations [41].

In this paper we propose a simple but effective neural network architecture that learns to synthesize high quality, novel views of objects or scenes under fine-grained viewpoint control in a self-supervised manner. Our work is guided by two observations: 1) to produce high quality renderings of the object close to the original view, it is important to be able to reason about the geometry of the scene or at least about approximate depth (cf. [28]). 2) However, once some parts of the scene become dis-occluded, learning based priors are a powerful option to predict the correct pixel values for previously occluded parts of the scene.

Figure 1: NVS Machines: Given a single source view our approach can generate accurate novel views with fine-grained control. Top: Synthetic views are generated given a single source input and a user specified 3D trajectory (illustration in inset). We feed the generated views to a SLAM system. In green: GT trajectory; in blue trajectory recovered from synthesized views (length 5m). Bottom: A hi-res image from the internet is used to synthesize novel fine-grained unseen views; model trained on synthetic data only.
We propose an architecture that leverages a transforming auto-encoder (T-AE), inspired by the concept of capsules in Hinton et al. [13], where latent codes are explicitly transformed in Euclidean space. This forces the network to implicitly learn a latent representation that is meaningful under 3D transformation. The transformed code is decoded into a depth-map of the target view. From these depth maps we compute dense correspondences between pixels in source and target view via perspective projection. Subsequently the final output image is created via a bi-linear sampling function. Importantly, these steps are learned in an unsupervised fashion and hence do not require access to ground truth depth or flow maps. The functions mapping from depth to flow and to the final view are all differentiable (although they do not have trainable parameters) and hence the entire network can be trained end-to-end. Furthermore, the final reconstruction loss is a meaningful signal for both the establishment of correspondences and the task of depth prediction: for the final per-pixel prediction to be correct, the flow must be correct and for the flow to be correct an accurate depth representation is required. This approach can synthesize views with small step-sizes and allows for explicit view control.

However, the depth based approach alone can not accurately handle cases with large dis-occlusions. Therefore, we incorporate two additional branches in our architecture that predict a) an image in the target view directly and b) a mask that allows to fuse the outputs from the depth and the image branch such that the overall result yields high-quality images of both intermediate and extrapolated views, even if relatively far away from the source view.

In summary this paper makes the following contributions: (i) We propose a neural network architecture that is structured to reason geometrically, enabling fine-grained view synthesis of 3D objects or scenes taking only a single image as input. (ii) The architecture leverages a 3D transforming auto-encoder in combination with differentiable geometric transformations. Trained end-to-end this approach enforces the network to implicitly learn the underlying 3D structure of the environment via unsupervised learning of depth and flow map predictions. (iii) We provide thorough experimental evidence both qualitatively and quantitatively demonstrating the effectiveness of our method for fine-grained view synthesis. We illustrate this in a variety of contexts, including rigid objects such as cars and chairs, and natural scenes taken from the KITTI dataset [9]. (iv) We show that the proposed architecture outperforms the state-of-the art in terms of view control and image reconstruction quality, even in the fixed view synthesis setting, proposed in previous work which casts this as image-to-image translation task (e.g., [40, 24, 31, 32]). (v) Finally, our method generalizes to entirely unseen imagery of real objects and enables a number of interesting application scenarios.

2. Related Work

Computer graphics approaches create high-quality 2D imagery via manipulation of the scene and camera parameters via perspective projection, given a 3D model of an object or scene. Image-based rendering systems require dense imagery for interpolation of rays [11, 21] or detailed 3D geometry reconstructed from several input views [3, 12, 27, 36, 42] and may be computationally expensive.

Learning in Euclidean space. Given the severely under-constrained case of reconstructing 3D environments from a single image, recent work has attempted to learn geometry information based on deep neural networks [2, 7, 8, 10, 18, 19, 30, 38]. For example, Eigen et al. [5] directly predict depth and [7, 22] reconstruct point sets from single images. [2, 10, 26] produce voxel occupancy maps and Sinha et al. [30] study suitable intrinsic shape representations for learning of deformation invariant descriptors. Kar et al. [18] propose a system for learning multi-view stereo.

Novel view synthesis. Deep learning techniques have been used to predict novel views from a single image input [41, 14, 40, 24, 32, 34]. Many of these approaches [32, 37, 25, 14] leverage autoencoder architectures to treat novel view synthesis as an image-to-image translation [15] problem. Flow-based approaches have been proposed in order to allow for preservation of local details in the target view [40, 24] via establishment of dense correspondences.

While demonstrating the feasibility of the task, these approaches provide no control over the exact viewing angle [14] or only allow for a fixed, small set of viewing angles [24, 40].

RNNs that sequentially rotate the object have been proposed to overcome this issue [35]. Another option is to learn disentangled latent representations [20] via variational autoencoders. However, full disentanglement remains an unsolved issue, and even if solved does not provide exact control over the view in metric space. Recent work [4, 39, 41] demonstrate the potential for fine-grained view synthesis, but require either additional supervision in the form of depth maps, which are cumbersome to acquire in real settings, or require dense image pairs for training, or need stereo input images. Our work also learns the task of novel view synthesis but contributes a learning scheme that produces high quality synthetic images with fine-grained control and self-supervised learning of intermediate depth and flow maps.

Transforming auto-encoders. Explicit rotation of latent codes was first proposed by Hinton et al. [13] with the goal of translation invariance in classification. Novel view synthesis approaches often augment latent codes with flattened viewpoint transforms before decoding into target views or flow maps [32, 20, 40]. However, we experimentally show that directly decoding to flow maps can inhibit regularization of the latent code to be aware of smooth view transformation.
3. Method

The goal of this work is to generate novel views of a 3D scene with high accuracy and under fine-grained view control, allowing for example to smoothly rotate a camera around an object, where previous work only allowed for prediction of fixed views. Moreover, it produces highly accurate predictions especially for cases with only mild disocclusions. However, for cases in which many pixels were originally hidden, learning based priors of image appearance are effective. To handle both cases we propose additional branches in the network that decode the transformed latent representation directly into the target view (image branch) and into a per-pixel mask. The final prediction is attained via a weighted average between the depth and image based predictions, where the mask provides the weights (see Fig. 3).

3.2. Depth-aware View Synthesis

Our main contribution is a depth-based approach to view synthesis (Fig. 2). 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 \( z_T = T_{s \rightarrow t}(z) \) is then decoded by a neural network to predict a depth map \( D_t \) as observed from the target viewpoint. \( D_t \) is then projected back into the source view based on the known camera intrinsics \( K \) and extrinsics \( T_{s \rightarrow t} \), yielding dense correspondences between the target and source views, encoded as dense backward flow map \( C_{t \rightarrow s} \). This flow map is then used to warp the source view pixel-by-pixel into the target view. Here attaining backward flow and hence predicting depth maps in source view is crucial. 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 \( M : X \rightarrow Y \), which in our case can be decomposed as:

\[
M(I_s) = B(P_{t \rightarrow s}(D_{\theta_d}(T_{s \rightarrow t}(E_{\theta_e}(I_s)))), I_s) = \hat{I}_t, \quad (1)
\]
where $B$ is the bi-linear sampling function and $E_{\theta_a}, D_{\theta_d}$ 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 T-AE encoder $E_{\theta_a}$ to produce position predictions for features and the decoder $D_{\theta_d}$ learns to generate features at corresponding positions by rendering the transformed representation from the specified view-angle [13].

Moreover, the use of backward flow $C_{t\rightarrow s}$, 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 T-AE network owing to the fact that the correspondences are computed deterministically from the predicted depth maps.

**Transforming auto-encoder.** We deploy a T-AE network [13], applying the desired 3D transform to the latent codes ($z_{T} = T_{s\rightarrow t}(z)$). The encoder and decoder are implemented with fully convolutional layers. The latent code $z$ is represented as vectorized set of points $z \in \mathbb{R}^{n \times 3}$, where $n$ is a hyper-parameter. This representation is then multiplied with the ground-truth transformation $T_{s\rightarrow t}$ describing the camera motion between source view $I_{s}$ and target view $I_{t}$ to attain the rotated code $z_{T}$. All functions in the T-AE module including encoding, vector reshaping, matrix multiplication and decoding are differentiable and hence amenable to training via backpropagation.

**Depth-based prediction.** We decode $z_{T}$ into a depth image $D_{t}$ in the target view. Note that during training no supervision via depth images is provided. From $D_{t}$ we compute the dense correspondence field $C_{t\rightarrow s}$ deterministically via perspective projection $P_{t\rightarrow s}$. The dense correspondences are then used to warp the pixels of the source view $I_{s}$ into the target view $\hat{I}_{t}$. This allows the network to copy information from the source view and makes the prediction texture invariant, resulting in sharp and detail-preserving outputs.

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

$$[X, Y, Z]^{T} = D_{t}(x_{t}, y_{t})K^{-1}[x_{t}, y_{t}, 1]^{T}$$

and $[x_{s}, y_{s}, 1]^{T} = K T_{s\rightarrow t} [X, Y, Z, 1]^{T}$,

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 intrinsics matrix.

**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. 3 are non-integer, this is done via differentiable bilinear sampling as proposed in [16]:

$$I_{t,d}(x_{t}, y_{t}) = \sum_{x_{s}} \sum_{y_{s}} I_{s}(x_{s}, y_{s}) \text{max}(0, 1 - |x_{s} - C_{x}(x_{t}, y_{t}))$$

$$\text{max}(0, 1 - |y_{s} - C_{y}(x_{t}, y_{t}))) \tag{4}$$

We note that this approach bears similarity to the approach in [40]. However, instead of predicting the dense correspondence field directly, we introduce the intermediate step of predicting depth. This guides the network to more explicitly reason about geometric constraints and avoids ambiguities in the pixel locations from source to target view.

**3.3. Fusion**

When parts of the scene are occluded in the source view, the depth based approach alone can not fully reconstruct the target view. For such cases we leverage two additional branches in our architecture that predict a) an image $I_{t,p}$ in the target view directly and b) a mask $M$ for the fusion of depth and image branch predictions such that the overall result yields high-quality images. As shown in Fig. 3, both of these predictions are directly decoded from the transformed latent code. The final output is then given by:

$$I_{t}(x_{t}, y_{t}) = M \odot I_{t,d} + (1 - M) \odot I_{t,p}, \tag{5}$$

where $\odot$ denotes the element-wise multiplication.

**3.4. Training**

The depth branch consists of three main steps: prediction of the depth map in the target view, projection of the depth values to obtain dense correspondences and finally warping of the source image into the target view. All three steps are differentiable which enables end-to-end training. However, only the T-AE module contains trainable parameters ($\theta_a, \theta_d$). 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 $I_{t}$. All three branches are trained via a reconstruction loss, applied on intermediate outputs and the final fused images.

$$L_{\text{recon}} = \|I_{t} - \hat{I}_{t,p}\|_1 + \|I_{t} - \hat{I}_{t,d}\|_1 + \|I_{t} - \hat{I}_{t}\|_1 \tag{6}$$

In addition, to improve realism of the pixel branch, we apply a least squares adversarial loss [25] and a perceptual loss [17]:

$$L_{\text{adv}} = (1 - \text{Dis}(I_{t,p}))^2 \text{ and } L_{\text{vgg}} = \|F(I_{t,p}) - F(\hat{I}_{t,p})\|_2 \tag{7}$$

where $\text{Dis}$ is a discriminator and $F$ is a pre-trained VGG [29] based feature extractor.
4. Experiments

We now evaluate our method quantitatively and qualitatively. We are especially interested in assessing image quality, granularity and precision of viewpoint control. First, we conduct detailed ablation experiments on synthetic objects, where fine-grained groundtruth is easy to attain, to numerically assess the reconstruction quality. Notably, we vary the viewpoints in much smaller step-sizes than what is observed in the training data (1° vs 20°). A core contribution of our work is the implicit learning of 3D structure from monocular views. To assess this aspect we also evaluate the intermediate depth and flow maps. Second, to evaluate generalizability, we test our system on natural city scenes. In this setting, we specify ground-truth camera trajectories for which we generate novel views and run an existing SLAM system on these to compare trajectories. Furthermore, we directly compare with prior work \[24, 31, 32, 40\] in their original setting, where a fixed set of seen views (of unseen objects or scenes) is generated.

4.1. Dataset

We conduct our experiments on two challenging datasets. A synthetic [1] and a real dataset [9].

**ShapeNet** We take models such as cars and chairs from the ShapeNet dataset [1]. In total we use 6555 instances of objects for training and 1640 for testing as suggested in [31].

**KITTI** KITTI [9] is a standard dataset for autonomous driving and 3D reconstruction, containing complex city scenes in uncontrolled environments. We still follow [31] using 18560 images to train our model, and 4641 images for testing.

4.2. Metrics

In our evaluations we report the following metrics:

**Mean Absolute Error** \(L_1\), is used to measure per-pixel color differences between synthesized images and ground truth images or depth and flow maps.

**Structural SIMilarity (SSIM) Index** [33], which has value in \([-1, 1]\), measuring the structural similarity between synthesized image and ground truth. The SSIM gives an indication of perceptual image quality.

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

4.3. Implementation Details

The networks are trained with Adam (\(lr = 0.00006, \beta_1 = 0.5, \beta_2 = 0.999\)). The encoder consists of 7 convolution layers with a stride of 2, kernel size of 4 and padding size of 1. More details please refer to the supplementary.
4.4. Evaluation on Synthetic Objects

Ablation study: fine-grained view control. We first evaluate the performance of our model under fine-grained viewpoint manipulations. In particular, we are interested in the generalization capabilities of the proposed architecture, hypothesized to be induced by the 3D-transformation aware latent space and the implicit prediction of depth and flow maps. In this experiment we demonstrate continuous synthesis of unseen target views from unseen source views. Here unseen means the view is not in the training set. For training, we render 18 views with varying azimuth between 0° and 360° with a step size of 20° for each 3D model. During testing, we randomly sample 125 unseen source views from each test object. Given a source view $I_s$, the network synthesizes 80 views with a step size of 1°.

We study two key aspects of our method: feature transformation and depth-guided view synthesis. We contrast T-AE based feature transformation (FT) with direct concatenation of latent code to the view transform (CAT). We also compare depth-guided view synthesis (Depth) with the flow only alternatives (Flow). This results in 4 combinations. Depth+FT represents ours, while Flow+CAT is similar to [24, 31, 40] but is extended for continuous view control. To ensure fair comparison, we use the same encoder and decoder architecture for all models.

Fig. 5 plots the reconstruction error between $[-40°, 40°]$ across these four alternatives. Ours consistently produces lower errors and yields much lower variance in between non-

![Figure 5: Comparison of $L_1$ reconstruction error as function of view rotation. Ours outperforms the baselines over the range and yields a smoother loss progression.](image)

![Figure 6: Flow comparison between Flow+FT and ours. With the guidance from depth, our method (Depth+FT) predicts a more accurate flow, resulting in a more accurate image.](image)

![Figure 7: Channel fusion examples for car and scene. Image from The pixel branch are in the correct shape but lacks detail. The visibility branch correctly predicts the invisible part. By fusion we yield realistic, detail preserving and complete outputs.](image)

![Figure 8: Unsupervised depth prediction for objects. (a): depth prediction; (b) reconstruction. The depth prediction and reconstruction appears similar to the ground-truth.](image)

![Figure 9: Unsupervised depth prediction for scene. Our predicted depth images in target view are compared with the ground-truth Lidar scans. The GT is only available at the bottom part of the image due to the limited working range of Lidar. For better visualization inverse depth is shown and the sparse Lidar points are upsampled.](image)
training training views. Notably, both depth-based models consistently outperform the flow-based counterparts. Feature transformation also performs better than viewpoint concatenation, demonstrating the effect of both design choices.

Fig. 4 shows qualitative results. To demonstrate the attained smoothness, we generate and overlay 80 views with step size of 1 degree from a single input. Compared to the baselines, our method exhibits similar spin pattern as the ground truth, whereas other methods mostly converge to the fixed training views. This suggests that over-fitting occurs, limiting view control. A close look at a specific view reveals that the baselines display distortions or converge to training views. Tab. 1 summarizes the average $L_1$ error and SSIM. Ours again outperforms all baselines on both metrics.

**Depth and optical flow.** Self-supervised depth map prediction is one of our core contributions. To illustrate that the reconstructed depths is indeed meaningful, we synthesize depth maps in the target view, given a single source view (see Fig. 8). Furthermore, our network can predict depth for multiple views and these can be fused into a 3D model of the underlying object. Fig. 8 (b) provides examples of such reconstructions attained via InfiniTAM. To quantitatively assess the depth prediction we calculate the $L_1$ and Accuracy between predicted and ground truth depth images (Tab. 1). Accurate flow is essential to synthesize correct and realistic images. We find that with the guidance from depth, the flow becomes much more accurate.

Fig. 6 shows that improved accuracy of flow contributes to reconstructing images. 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.

**Channel fusion.** Although the depth-based branch can produce sharp predictions, geometry alone can not compensate for missing information. Therefore we introduce an additional pixel generation branch and a visibility prediction branch. As shown in Fig. 7, the pixel branch can produce plausible images in the correct shape, however the image lacks detail. The visibility branch correctly predicts the invisible part, such as the tail of car in the first row or the unseen part of the scene when moving backward. By fusing depth and pixel branches via the visibility mask, we yield realistic, detail preserving and complete outputs. More detailed analysis can be found in the supplementary.

**Fixed view comparison: synthetic.** We now directly compare with state-of-the-art methods in the setting test contains only a fixed set of views. Tab. 2, shows that ours consistently outperforms all baselines (more details in supplementary).

### Table 1: Quantitative analysis of fine-grained view control on generated images, flow and depth maps.

| Methods    | Image | Flow | Depth |
|------------|-------|------|-------|
|            | L1 SSIM | L1 Acc | L1 Acc |
| Flow+CAT   | .062  | .924  | .035  | .691 |
| Flow+FT    | .052  | .932  | .029  | .763 |
| Depth+CAT  | .045  | .943  | .022  | .846 |
| Depth+FT   | .039  | .949  | .021  | .857 |
| Ours       | .066  | .932  | .141  | .898 |

### Table 2: Quantitative comparison on fixed views.

| Methods    | Car | Chair | KITTI |
|------------|-----|-------|-------|
|            | L1 SSIM | L1 SSIM | L1 SSIM |
| L1[32]     | .139  | .223  | .295  |
| AFN[40]    | .148  | .229  | .418  |
| TVSN[24]   | .119  | .202  | –     |
| MV.p[31]   | .111  | .187  | .259  |
| MV.f[31]   | .119  | .208  | .397  |
| MV[31]     | .098  | .181  | .203  |
| Ours.p     | .078  | .161  | .254  |
| Ours.d     | .083  | .159  | .384  |
| Ours       | .066  | .141  | .191  |

4.5. Evaluation on Natural Scenes

We now demonstrate that our method works not only for synthetic objects but also for natural images.

**Trajectory recovery.** To provide a realistic and challenging evaluation we propose the following setting: we first specify arbitrary GT trajectories (away from the car’s motion in KITTI) and later run a state-of-the-art SLAM [6] system to estimate the camera pose based on the synthesized views. We report deviation from the input trajectory (up-to-scale). Fig. 11 illustrates that the estimated trajectory and camera pose align well with the ground-truth. In contrast, the state-of-the-art in fixed view synthesis [31], mostly produces straight forward motion regardless of the input, possibly due to overfitting to training trajectories. In Fig. 12, [31] produces distorted images while ours are sharp and correct.

**Stereo evaluation.** To further evaluate generalization to unseen training views we synthesize the right stereo view

![Figure 10: Stereo prediction.](image-url)

From left to right, ground truth stereo pairs, prediction by MV[31], prediction by ours. Our method produces reasonable results, while [31] mostly copies the input view, leading to high error.
Figure 11: **Estimated camera pose from synthesized views.** Left: ours. Right: state-of-the-art in fixed view synthesis [31]. In each block the first image is the source view and the following three images are synthesized views along the user defined trajectory (green). The inset indicates the frame index. Trajectories are estimated via a state-of-the-art SLAM system [6] and compared to the input trajectory. Trajectory estimated from our synthesized views align well with the ground-truth, while [31] mostly produces straight forward motion regardless of the input.

Figure 12: **Direct comparison of synthesized views on a trajectory.** Top: ours. Bottom: [31]. Our method produce sharp and correct images while [31] produces distorted images.

from the left, given a desired baseline. Fig. 10 shows that ours produces reasonable results, while [31] mostly copies the input view, leading to high error.

**Fix view comparison: real.** Previous approaches [40, 31] evaluate the performance by directly measuring the image reconstruction error on test images. We perform the same evaluation (Tab. 1) and show lower $L_1$ errors. However, we notice that this test setting does not reflect the goal of fine-grained view control since all views form discrete pairs (details in supplementary).

**Depth evaluation** Finally we evaluate the depth prediction. The average absolute relative difference is 0.186. Qualitative results are shown in Fig. 9.

### 4.6. Applications

We find that our model generalizes well to unseen data thanks to the usage of depth-based warping instead of direct pixel generation. Interestingly, our depth-based model, trained on 256x256 images, can be applied to high resu-
tion (1024x1024) images with minimal modification. We first down-sample the input image to 256x256 and predict the target depth map. We then upsample the depth map to original resolution and apply the warping. The whole 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 illustration we have implemented some proof-of-concept demonstrators.

First, 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. 13 and accompanying video).

5. Conclusion

We have presented a novel self-supervised approach to learning to generate novel views with fine-grained viewpoint control. At the core of our method lies a depth-based image prediction network that is forced to satisfy explicit geometric constraints, given by the camera transform between input and target view. We have shown experimentally that our approach outperforms the state-of-the-art in terms of reconstruction quality and view control on synthetic data. Furthermore, the method also predicts geometrically meaningful images when trained on the KITTI dataset, containing complex real-world imagery. Finally, we have shown that the method generalizes to entirely unseen data such as high-resolution imagery of real objects and scenes, downloaded from the internet and hence enables compelling application scenarios.

References

[1] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3D model repository. CoRR, abs/1512.03012, 2015.

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3dr2n2: A unified approach for single and multi-view 3D object reconstruction. In European Conference on Computer Vision, pages 628–644. Springer, 2016.

[3] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 11–20. ACM, 1996.

[4] A. Dosovitskiy, J. T. Springenberg, M. Tatschennko, and T. Brox. Learning to generate chairs, tables and cars with convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 39(4):692–705, 2017.

[5] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems, pages 2366–2374, 2014.

[6] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. Mar. 2018.

[7] H. Fan, H. Su, and L. Guibas. A point set generation network for 3D object reconstruction from a single image. In Conference on Computer Vision and Pattern Recognition (CVPR), volume 38, 2017.

[8] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5515–5524, 2016.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[10] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In European Conference on Computer Vision, pages 484–499. Springer, 2016.

[11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 43–54. ACM, 1996.

[12] L. He, J. Shade, S. Gortler, and R. Szeliski. Layered depth images. 1998.

[13] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I, ICANN'11, pages 44–51, Berlin, Heidelberg, 2011. Springer-Verlag.

[14] R. Huang, S. Zhang, T. Li, and R. He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. arxiv, 2016.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015.

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, pages 694–711. Springer, 2016.

[18] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. In Advances in Neural Information Processing Systems, pages 364–375, 2017.

[19] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. CoRR, vol. abs/1703.04309, 2017.

[20] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. arXiv preprint arXiv:1503.03167, 2015.

[21] M. Levoy and P. Hanrahan. Light field rendering. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 31–42. ACM, 1996.
[22] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In AAAI Conference on Artificial Intelligence (AAAI), 2018. 2

[23] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang. Multi-class generative adversarial networks with the L2 loss function. CoRR, abs/1611.04076, 2016. 4

[24] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 1, 2, 5, 6, 7

[25] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised geometry-aware representation for 3d human pose estimation. CoRR, abs/1804.01110, 2018. 2, 3

[26] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 3, 2017. 2

[27] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. 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

[28] J. Shade, S. Gortler, L.-w. He, and R. Szeliski. Layered depth images. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’98, pages 231–242, New York, NY, USA, 1998. ACM. 1

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 4

[30] A. Sinha, J. Bai, and K. Ramani. Deep learning 3d shape surfaces using geometry images. In European Conference on Computer Vision, pages 223–240. Springer, 2016. 2

[31] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In European Conference on Computer Vision (ECCV), 2018. 2, 5, 6, 7, 8

[32] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Single-view to multi-view: Reconstructing unseen views with a convolutional network. CoRR, abs/1511.06702, 2015. 1, 2, 5, 7

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004. 5

[34] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Interpretable transformations with encoder-decoder networks. 2017 IEEE International Conference on Computer Vision (ICCV), pages 5737–5746, 2017. 2

[35] J. Yang, S. Reed, M. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. Neoplasia, 2015-January:1099–1107, 2015. 2

[36] F. Yu and D. Gallup. 3d reconstruction from accidental motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3986–3993, 2014. 2

[37] B. Zhao, X. Wu, Z. Cheng, H. Liu, and J. Feng. Multi-view image generation from a single-view. CoRR, abs/1704.04886, 2017. 2

[38] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In CVPR, 2017. 2

[39] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817, 2018. 2

[40] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In European Conference on Computer Vision, 2016. 1, 2, 4, 5, 6, 7, 8

[41] H. Zhu, H. Su, P. Wang, X. Cao, and R. Yang. View extrapolation of human body from a single image. CoRR, abs/1804.04213, 2018. 1, 2

[42] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In ACM transactions on graphics (TOG), volume 23, pages 600–608. ACM, 2004. 2