

# Novel View Synthesis via Depth-guided Skip Connections

Yuxin Hou      Arno Solin      Juho Kannala  
 Department of Computer Science, Aalto University, Espoo, Finland  
 firstname.lastname@aalto.fi

## Abstract

We introduce a principled approach for synthesizing new views of a scene given a single source image. Previous methods for novel view synthesis can be divided into image-based rendering methods (e.g., flow prediction) or pixel generation methods. Flow predictions enable the target view to re-use pixels directly, but can easily lead to distorted results. Directly regressing pixels can produce structurally consistent results but generally suffer from the lack of low-level details. In this paper, we utilize an encoder-decoder architecture to regress pixels of a target view. In order to maintain details, we couple the decoder aligned feature maps with skip connections, where the alignment is guided by predicted depth map of the target view. Our experimental results show that our method does not suffer from distortions and successfully preserves texture details with aligned skip connections.

## 1. Introduction

Novel view synthesis (NVS) is the task of generating new images of a scene given single or multiple inputs of the same scene (see, e.g., in Fig. 1: given one image of the object, we generate a new image of the object from a novel viewpoint). NVS has various applications. For example, it can be used in virtual reality applications, where capturing all possible viewpoints of real-world scenes is impractical. With NVS one can just capture few images to offer a seamless experience. Moreover, NVS enables users to edit images more freely (e.g., rotating products interactively in 3D for online shopping).

Generally, to solve the NVS task, comprehensive 3D understanding of the scene by the model is important. Given the 3D geometry, we can render target views with 3D model-based rendering techniques. In that case, some methods estimate the underlying geometry of the scene with 3D representations like voxels [7], and mesh [16], but these methods can be computationally expensive. Unlike traditional 3D model-based rendering, image-based rendering (IBR) methods render novel views directly from input im-

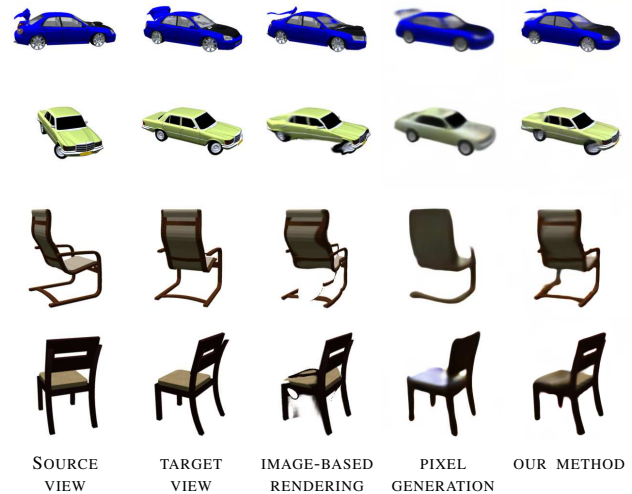


Figure 1: Results of image-based rendering methods suffer from distortion, while the results of direct pixel generation methods lack detailed features. Our method has the benefits from both warping methods and pixel generation methods.

ages. Some IBR methods predict the appearance flow directly without geometry [14, 37]. Some IBR methods render with explicit geometry, such as 3D warping with depth maps, which use a geometric transformation to obtain pixel-to-pixel correspondences [4]. Since the pixels from input views can be re-projected to the target view directly, original low-level details of the scene like colours and textures are well-preserved. However, estimating the accurate correspondences can be challenging, especially for single input views in texture-less regions and occlusion regions, and the failures can easily lead to distorted synthesized results. On the other hand, some methods attempt to regress pixels directly [28, 34]. These pixel generation methods can generate structurally consistent geometric shapes, but the visual quality of generated results are worse than methods that exploit correspondence since the lack of detailed features. Fig. 1 shows the limited performance of pixel generation methods and image-based rendering methods.

In this paper, we aim to combine the advantages of both

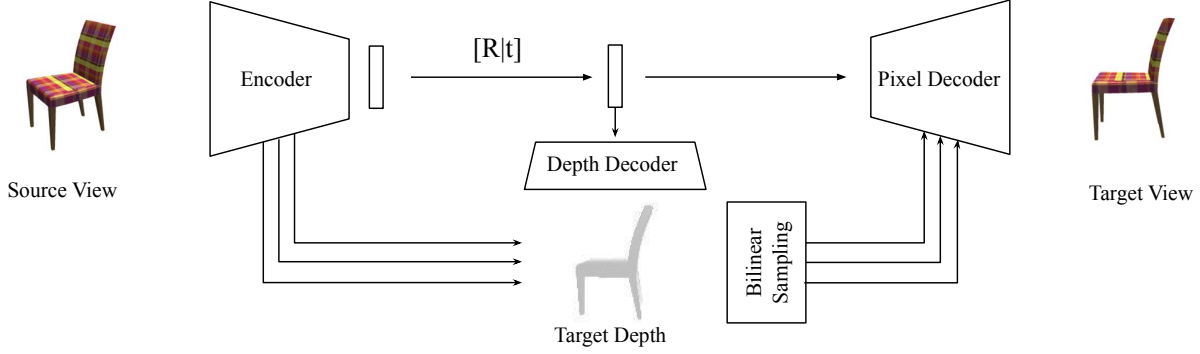


Figure 2: Overview of our architecture. There are three main components: the encoder, the depth decoder, and the pixel decoder. We apply the given geometric transformation on the latent code from the source view directly to obtain the latent code for the target view. Given the transformed latent code, we predict the depth map of the target view, and use the predicted depth to warp feature maps of source view to the target with bilinear sampling. The warped feature maps are then passed to the decoder as skip connections to assist pixel generation.

image-based rendering methods and direct pixel generation methods. The main benefit of image-based rendering methods is that they can exploit the correspondence pixels of target views in the source view and re-use pixels, so the predictions will not lose detailed features. Similarly, we decide to utilize the skip connections between the encoder and the decoder to transfer low-level features. Skip connections with U-net architectures [23] have proved to be useful for passing feature information to lower layers in vision tasks like semantic segmentation, where the output and the input are well-aligned spatially. However, for novel view synthesis, the skip connections cannot be applied immediately because of the different shapes of the input view and the target view. To address the problem, we predict the depth map of the target map, warping multi-level feature maps before passing them to the generation decoder. Compared to warp image pixels directly, the generation decoder can exploit learned prior knowledge to ‘correct’ distortion.

In conclusion, we propose an image generation pipeline with transforming auto-encoders which utilize warped skip connections to transfer low-level features. We compare our model against both state-of-the-art pixel generation methods and image-based rendering methods, demonstrating that our method can alleviate common issues like distortion and lack of details.

## 2. Related Work

**Geometry-based view synthesis** A large body of research attempts to solve the novel view synthesis problem via explicitly modelling the underlying 3D geometry. To represent the 3D structure, common 3D representations like voxel [7, 19, 30, 21, 15], point cloud [33], mesh [16], and layered representations [24, 29, 36] are widely applied. Meshes and point-clouds can suffer from sparsity. Voxel-

based methods are limited with the type of scenes and resolutions because of the memory constraints. Layered representations rely on a large number of layers to reach good quality. Apart from discrete representations, recent interests in continuous 3D-structure-aware representation also show promising results [20, 26]. In this paper, though we do not use explicit 3D representation, we predict the depth map from single view (2.5D) to help understand the 3D geometry of the input view and the depth map will be used to guide aligned skip connections.

### Image generation with disentangled representation

Many deep generative networks are capable of generating photorealistic images, but to solve the novel view synthesis task, networks generally required explicitly disentangled representations, like decoupling the pose and identity features. Cheung *et al.* [5] utilize an auto-encoder to learn disentangled latent units representing various factors of variations (poses, illumination conditions, *etc.*) and generate novel manipulations of images. Tatarchenko *et al.* [28] use separate convolutional layers and fully connected layers to process the image and the angle independently, then the separate latent representations are merged before up-convolutional layers. Casale *et al.* [2] divide the feature vector into an object feature and a view feature, and utilize GP priors to model correlations between views. Inspired by traditional graphics rendering pipelines, Zhu *et al.* [38] build the 3D representation from three independent factors: shape, viewpoint, and texture. Besides factorizing latent representations, some methods use equivariant representations to handle transformations. Hinton *et al.* [13] proposed Transforming auto-encoders (TAE) to model both 2D and 3D transformations of simple objects. Generally, directly generated images may suffer from blurriness, lack of texture details, or inconsistency of identity.

**Image-based Rendering** Image-based rendering re-uses the pixels from source images to generate target views. Previous image-based rendering methods can be classified into three categories according to how much geometric information is used: rendering without geometry, rendering with implicit geometry, and rendering with explicit geometry (approximate or accurate geometry) [25]. Some traditional methods construct a continuous representation of the plenoptic function from observed discrete samples with unknown scene geometry, like building lightfield [18] or lumigraph [12, 1] with dense input views. Recently, some learning-based methods predict correspondence without geometry directly. Zhou *et al.* [37] use separate encoders for input images and viewpoint transformation and predict appearance flow field directly. Ji *et al.* [14] used a rectification network before generating dense correspondences between two input views, and the view morphing network finally synthesizes the target middle view via sampling and blending. When the depth information is available, 3D warping can be used to render nearby viewpoints. Some methods estimate depth maps from multi-view inputs [8, 9]. Choi *et al.* [6] estimate a depth probability volume that accumulated from multi inputs rather than a single depth map of the novel view. Chen *et al.* [4] use the TAE to predict the depth map of the target views directly. Our method does not re-use the pixels from inputs directly; instead, we re-use the feature maps extracted from the input view.

To combine the advantages of image-based rendering and image generation, Park *et al.* [22] used two consecutive encoder-decoder networks, first predicting a disocclusion-aware flow and then refining the transformed image with a completion network. Sun *et al.* [27] proposed a framework to aggregate flow-based predictions from multiple input views and the pixel generation prediction via confidence maps. In this paper, we present a different way that can bring the power of explicit correspondence to image generation via skip connections.

### 3. Methods

Our architecture consists of three main parts: the encoder  $\phi$ , the depth prediction decoder  $\psi_d$ , and the pixel generation decoder  $\psi_p$ . Fig. 2 shows the overview of our pipeline. The encoder extracts feature maps and latent representation for the source view firstly. To exploit geometric transformation explicitly, we apply the given transformation matrix on the latent representation from the source view to obtain the latent representation of the target view, which will be passed to the depth decoder and the pixel decoder. To take advantage of correspondence pixels in the source view, we predict the depth map for the target view given the transformed latent code. Then we can use the estimated depth map to find dense correspondences between target and source views. Instead of warping the source image into the target view,

we warp the multi-level feature maps extracted from the encoder via bilinear sampling and then pass them to the decoder as skip connections, transferring low-level details to assist final pixel regression.

#### 3.1. Transformable Latent Code

Inspired by [4] that applying the transformation matrix on latent code directly to predict depth map for target view, we also adopt the idea of using a TAE to learn a compact latent representation that are transformation equivariant. Given the source image  $I_s$ , the learnt latent code  $z_s = \phi(I_s)$  can be regarded as a set of points  $z_s \in \mathbb{R}^{n \times 3}$  extracted by encoder  $\phi$ . Then the representation is multiplied with the given transformation  $T_{s \rightarrow t} = [R | t]_{s \rightarrow t}$  to get the transformed latent code for the target view:

$$\tilde{z}_t = T_{s \rightarrow t} \cdot \dot{z}_s, \quad (1)$$

where  $\dot{z}_s$  is the homogeneous representation of  $z_s$ . Intuitively, training in this way will encourage the latent code to encode 3D position information for features.

#### 3.2. Depth-guided Skip Connections

Since [4] is an image-based rendering method, the quality of prediction results relies on the accuracy of estimated depth maps. However, the monocular depth estimation with the TAE architecture (w/o skip connections) is challenging. In that case, pure image-based rendering methods can lead to distortion easily because of the unstable depth prediction. Also with a monocular input, image-based rendering cannot inpainting the missing parts since they do not have correspondences in source views. In this work, to alleviate the mentioned limitations, we decide to synthesis the target view with a pixel generation pipeline, regressing the pixel value with the pixel decoder  $\psi_p$ .

Rethinking about the TAE architecture used in [4] and the design of the equivariant latent code  $z$ , though  $z \in \mathbb{R}^{n \times 3}$  might be sufficient for encoding position predictions for features and then be mapped into depth maps, regressing the pixels directly can be difficult. It is mainly because the downsample and upsample process can lose much detailed information, especially for the view that includes many small objects or rich textures. Generally, the skip connections have proved effective in recovering fine-grained details, but it cannot be used directly with the TAE architecture since the shape of output changed. In that case, there are two decoders in our framework, one for depth prediction  $\psi_d$  and one for pixel generation  $\psi_p$ . After getting the depth estimation for target view  $\hat{D}_t = \psi_d(\tilde{z})$ , we use the depth map to warp the feature maps  $F^i$  at different level  $i$ . In order to maintain texture details, we also have a feature map that maintains image resolution, which we call *conv0*. Given the camera intrinsic matrix  $K$ , the relative pose  $T_{t \rightarrow s}$  and the predicted depth map of the target view  $\hat{D}_t$ , we can

find the correspondences in the source view in the following way [35]:

$$p_s \sim K T_{t \rightarrow s} \tilde{D}_t(p_t) K^{-1} p_t, \quad (2)$$

where  $p_t$  and  $p_s$  denote the homogeneous coordinates of a pixel in the target view and source view respectively. Since the obtained correspondences  $p_s$  are continuous values, we use differentiable bilinear sampling that interpolates the values of the 4-pixel neighbours of  $p_s$  to approximate  $F_i(p_s)$ . The warped feature maps can be represented as  $\tilde{F}_t^i$ , which will be passed to the pixel decoder  $\psi_p$  for concatenation.

Guided by predicted depth maps, the skip connections of warped feature maps enable the method to benefit from establishing explicit correspondences and maintain the low-level details. Also compared to image-based rendering that warp pixels directly, using multi-level skip-connections of warped feature maps helps to exploit learned prior information and avoid the loss of information.

### 3.3. Training Loss Functions

The whole framework can be trained in an end-to-end manner since all modules in our pipeline are differentiable. For each input sample, only a single source image and the target image and their relative transformation are given. We optimize both the encoder, the depth decoder and the pixel decoder jointly. For pixel regression, we use multi-scale L1 reconstruction loss and VGG perceptual loss to encourage generating realistic images. To train the depth decoder in an unsupervised manner, we use the edge-aware smoothness loss and introduce a depth consistent loss to make more stable predictions.

**Multi-scale Reconstruction Loss** To integrate learned prior knowledge and alleviate the negative impacts introduced by wrong depth prediction (*e.g.*, distortion), we make multi-scale novel view predictions, finalizing the final prediction from coarse to fine-grained. The total reconstruction loss  $\mathcal{L}_{\text{reco}}$  is the weighted combination of the individual losses at different scales in the pixel decoder:

$$\mathcal{L}_{\text{reco}} = \sum_i w_i |\tilde{I}_t^i - I_t|, \quad (3)$$

where  $\tilde{I}_t^i$  is the upsampled predicted target images and  $w_i$  is the weight for results at different scale  $i$ . The weights decrease according to the resolution of prediction. Intuitively, compared to only considering the final prediction, the multi-scale reconstruction loss should help since it will produce gradients from larger receptive fields rather than small neighbourhoods. Also, as it needs to predict correct results at coarse levels, it can encourage the latent code to understand the 3D scene without the skip connections and alleviate the dependence of the skip connections, avoiding the distortion with inaccurate depth estimation.

**VGG Perceptual Loss** Similar to [22], besides L1 reconstruction loss, we adopt the VGG perceptual loss to get sharper synthesis results. A pretrained VGG16 network is used for extracted features from generation results and ground-truth images, and the perceptual loss is the sum of feature distances (we use L1 distance) computed from a number of layers.

**Depth Consistent Loss** To regularize the latent code and its depth prediction without supervision, we introduce a depth consistent loss. Intuitively, the depth decoder project the latent code into depth maps, so it should work for both extracted latent code  $z$  and transformed latent code  $\tilde{z}$ . During the training, we also extract the latent code  $z_t$  from the target view via the encoder. Instead of encouraging the transformed latent code  $\tilde{z}_t$  to be same with the latent code extracted target view  $z_t$ , we encourage the distance of depth predictions to be small:

$$\mathcal{L}_{\text{depth}} = |\psi_d(z_t) - \psi_d(\tilde{z}_t)|. \quad (4)$$

**Edge-aware Smoothness Loss** Similar to [11], we use an edge-aware smoothness loss that should encourage predicted depth maps to be locally smooth. The loss is weighted by an edge-aware term since the depth discontinuities often occur at image edges:

$$\mathcal{L}_{\text{edge}} = \frac{1}{N} \sum_{i,j} |\partial_x \tilde{D}_t^{ij}| e^{-\|\partial_x I_t^{ij}\|} + |\partial_y \tilde{D}_t^{ij}| e^{-\|\partial_y I_t^{ij}\|}, \quad (5)$$

where  $\tilde{D}_t$  is the predicted depth map of the target view and  $I_t$  is the ground-truth target view.

In conclusion, the final loss function for training the framework jointly will be

$$\mathcal{L} = \lambda_m \mathcal{L}_{\text{reco}} + \lambda_v \mathcal{L}_{\text{vgg}} + \lambda_d \mathcal{L}_{\text{depth}} + \lambda_e \mathcal{L}_{\text{edge}}, \quad (6)$$

where the  $\lambda_m$ ,  $\lambda_v$ ,  $\lambda_d$ , and  $\lambda_e$  are weights for different loss functions.

## 4. Experiments

To show that our method can combine the advantages of both image generation and 3D warping methods effectively, we compared our method with three state-of-the-art methods: one typical image generation method proposed by Tatarchenko *et al.* [28], one image-based rendering method proposed by Chen *et al.* [4], which also share the similar TAE architecture with ours; and an explicit aggregation scheme proposed by Sun *et al.* [27] that predict confidence maps for both pixel generation results and flow prediction results. We replace the discrete one-hot viewpoint representation in [28] with cosine and sine values of the view angles. We jointly train the encoder, the depth decoder, and the pixel decoder using the Adam [17] solver with  $\beta_1 = 0.9$



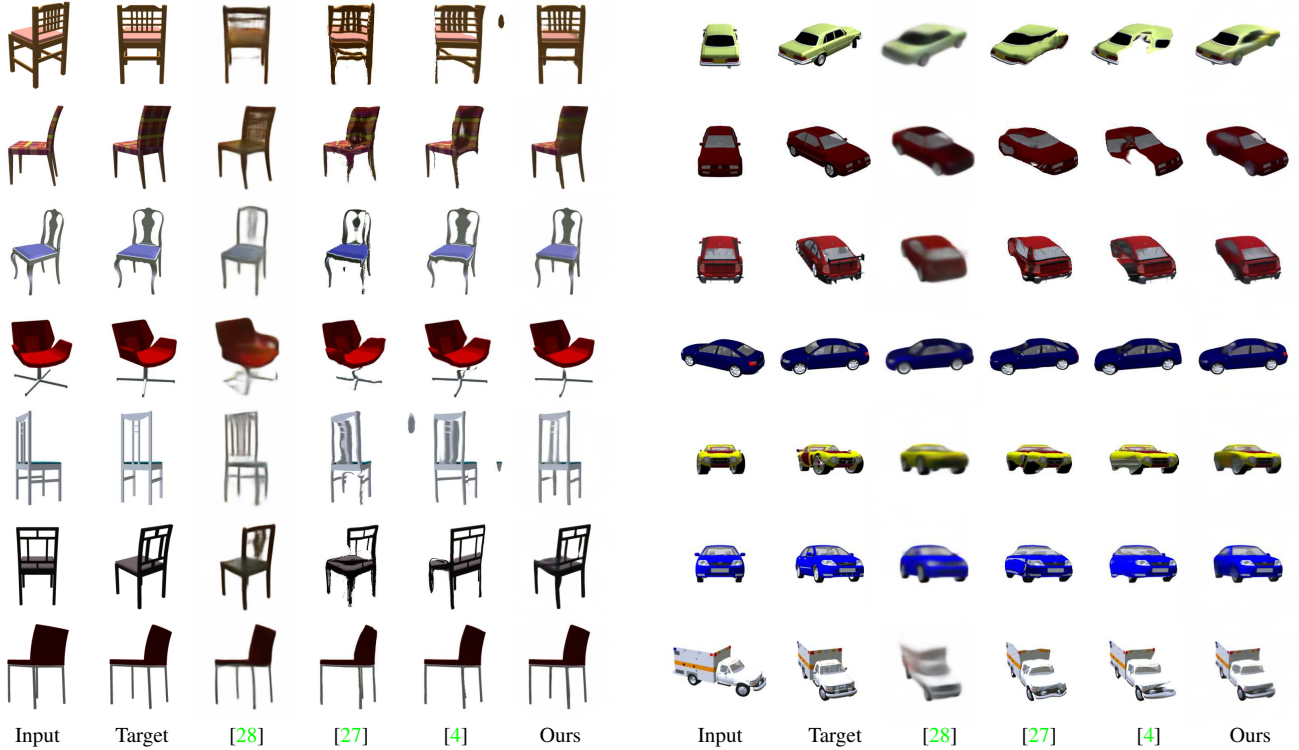


Figure 3: Results on ShapeNet objects. Our methods generate as structure-consistent predictions as pixel generation methods (for example, it can generate the missing chair legs compared to [4]); on the other hand, our generated images are not blurry and include rich low-level details as image-based rendering methods (zoom in for better visualization).

and  $\beta_2 = 0.999$ , and a learning rate of  $10^{-4}$ . For our methods, we add skip connection with feature maps *conv0*, *conv2*, *conv3*, *conv4* from the encoder because of the best performance. To evaluate the predictions, we report numbers of mean absolute error L1 that measures per-pixel difference and the structural similarity (SSIM) index [32] that indicates perceptual image quality. For L1 metric, smaller is better; for the SSIM metric, larger is better.

#### 4.1. Datasets

We conduct our experiments on two different types of datasets: for objects we use ShapeNet synthetic dataset [3] and for real-world scenes we use KITTI Visual Odometry [10] dataset. More specifically, we select cars and chairs in the ShapeNet dataset. Generally, datasets with complicated structures and camera transformation will challenge the 3D understanding (e.g., depth estimation in our method), while datasets with rich textures will show if the methods can preserve the fine-grained details well. Among our selected datasets, the chairs have more complicated shapes and structures, but for each chair the texture is more simple. Reversely, the cars have much simpler shapes but there will be more colorful patterns on each car. For KITTI, the

scene includes more objects, and translations are the main transformation between frames, unlike ShapeNet where rotation is the key transformation. In that case, the accurate depth estimation is less necessary ([31] shows that even approximating the scene as a single plane can give reasonable results), while the ability to recover the low-level details is more important for performance.

**ShapeNet** ShapeNet is a large-scale synthetic dataset of clean 3D models [3]. For ShapeNet objects, we use rendered images with the dimension of  $256 \times 256$  from 54 viewpoints (the azimuth from  $0^\circ$  to  $360^\circ$  with  $20^\circ$  increments, and the elevation of  $0^\circ$ ,  $10^\circ$ , and  $20^\circ$ ) for each object. The training and test pairs are two views with the azimuth difference within the range  $[-40^\circ, 40^\circ]$ . For ShapeNet chairs, there are 558 chair objects in the training set and 140 chair objects in the test set; For ShapeNet cars, there are 5,997 car objects in the training set and 1,500 car objects in the test set.

**KITTI** We use the KITTI odometry datasets since the ground-truth camera poses are provided. There are 11 sequences and each sequence contains around 2,000 frames on average. We restrict the training pairs to be separated by at most 7 frames.

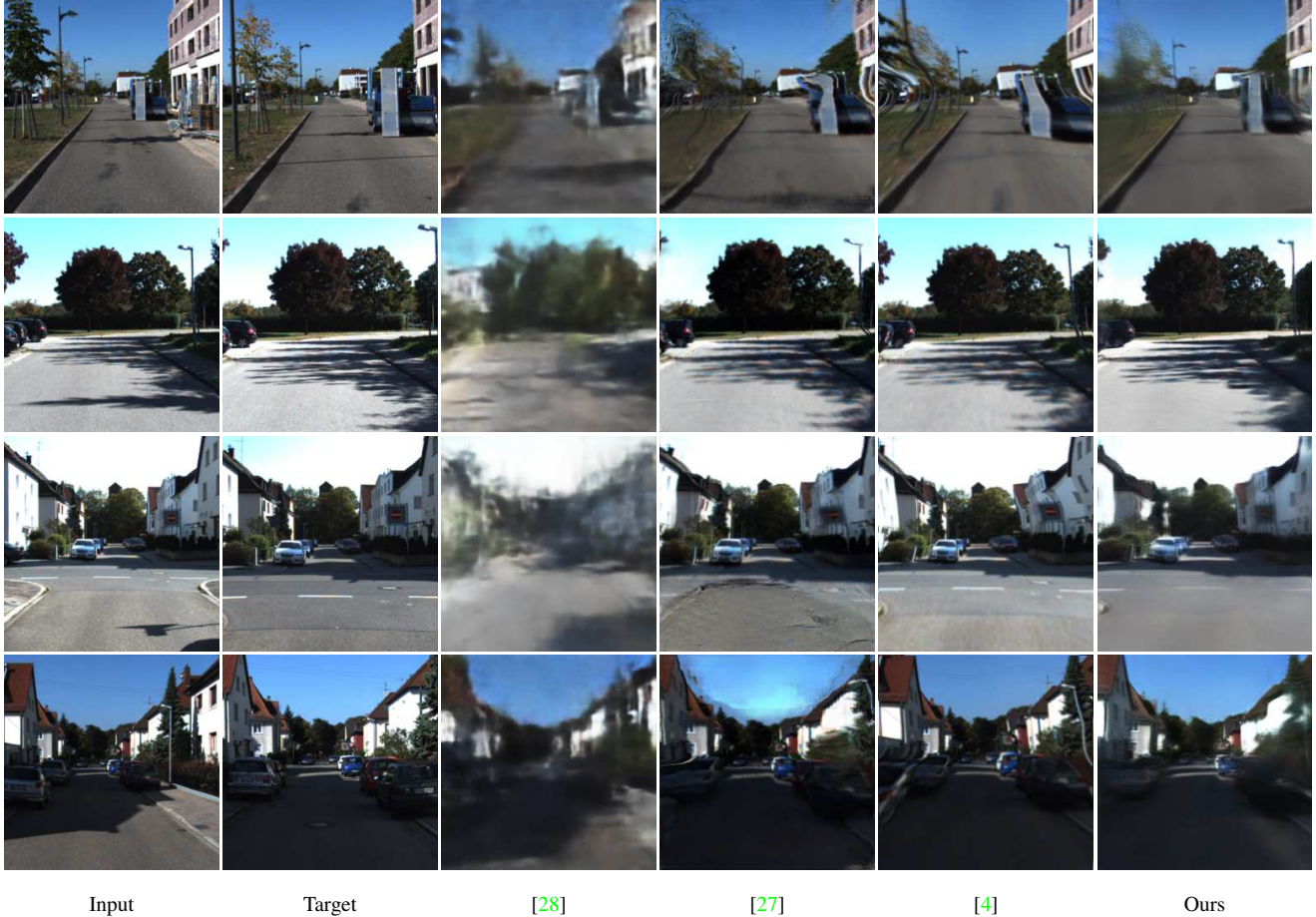


Figure 4: Qualitative results on KITTI. Our method generate clear and structure consistent predictions, while pixel generation methods [28] struggle with blurry and image-based rendering methods [4] suffer from distortion (see the house and truck on row 1, the street light pole on row 2, and the houses on row 3 and row 4).

For all selected datasets, we use the same train/test split as [27]. Since the evaluation samples in [27] are created for multi-view inputs, we randomly select the source views as our input view. In total, for ShapeNet chairs there are 42,834 pairs for testing; for ShapeNet cars there are 42,780 pairs for testing, and for KITTI there are 10,000 pairs for testing.

## 4.2. ShapeNet Evaluation

Table 1 shows the comparison results on ShapeNet objects. Our methods perform the best for both chair and car objects, showing that it can deal with complicated 3D structures of chairs as well as rich textures of cars at the same time. Fig. 3 shows the qualitative results for all methods. The pixel generation method [28] produces blurry results and the object identity failed to be preserved because of the lack of low-level details. Image-based rendering methods [4] suffer from the distortion. Our method combine the advantages of the two type of approaches. On the one hand,

our method exploits learned prior knowledge and generate structure consistent predictions as [28] (e.g., our method can generate the missing chair leg in the last row and the missing tires of [4] in row 5 and row 6); on the other hand, our generated images are not blurry and includes rich low-level details as [4]. The aggregation method [27] is mainly designed for multi-view inputs, so their modules cannot benefit from the recurrent neural network architecture when the input is a single source image.

## 4.3. KITTI Evaluation

We also evaluate all methods on KITTI to show that our model can capture low-level details well with the aid of skip connections. Table 2 shows the quantitative results on the KITTI dataset. We achieve the best SSIM results and get comparable L1 performance as the aggregation method [27]. Since [27] uses adversarial loss for their pixel generation module, it can inpaint missing regions better than other methods. As a pixel generation method, our method



Figure 5: Ablation study results. We compare the performance of our full model with its variants. Results show that the multi-scale loss help to generate thin structures (like the chair leg in the 1st and 3rd row of the 4th column). The  $\mathcal{L}_{VGG}$  makes the results sharper. The skip connection from *conv0* maintains the detailed features (like the pattern for 2nd row). The  $\mathcal{L}_{depth}$  leads to more stable results.

Table 1: Results on ShapeNet objects. Our methods perform the best for both chair and car objects, showing that it can deal with complicated 3D structures of chairs as well as rich textures of cars.

METHODS	CHAIR		CAR	
	L1	SSIM	L1	SSIM
Tatarchenko [28]	0.1043	0.8851	0.0491	0.9226
Sun [27]	0.0810	0.8993	0.0444	0.9282
Chen [4]	0.0769	0.9099	0.0396	0.9395
Ours	<b>0.0584</b>	<b>0.9256</b>	<b>0.0286</b>	<b>0.9493</b>

Table 2: Results on KITTI. We achieve the best SSIM results, and the L1 performance is better than both the pixel generation method [28] and the image-based rendering method [4].

METHODS	KITTI	
	L1	SSIM
Tatarchenko [28]	0.3119	0.6191
Sun [27]	<b>0.1868</b>	0.6582
Chen [4]	0.2354	0.6461
Ours	0.1985	<b>0.7043</b>

is obviously better than [28] in terms of the fine-grained textures, which shows the effectiveness of aligned skip connections. Compared to image-based rendering method [4], we still perform better on both L1 error and SSIM. In Fig. 4, the qualitative results show the same finding. Generally, our method generates clear predictions, and preserves the structure better (check the house and truck in row 1, the street light pole in row 2, the houses in row 3 and row 4).

#### 4.4. Ablation Studies

To understand how different blocks of the framework play their roles, we conduct ablation studies on the ShapeNet chairs, since it is the most challenging selected dataset for 3D structures. Table 3 and Fig. 5 show the per-

formance of different variants. Firstly, we compare the performance of our baseline architecture and the image-based rendering method [4]. In the baseline architecture, we use our two-decoders architecture and optimize with the simple L1 reconstruction loss only, while [4] uses one decoder for depth estimation. Since our baseline architecture achieves better performance compared to [4], it shows the effectiveness of our framework on combining the warping methods and pixel generation methods. Moreover, we observe that without the skip connections, our method can be regarded as a typical pixel generation methods that still suffer the same issue as [28], which also proves our assumption that the compact equivariant latent code cannot encode sufficient information for pixel regression and we need the assistance

Table 3: Results for ablation studies. All designed modules and loss functions are both useful for boosting performance. The baseline arch means using our architecture with L1 reconstruction at the final resolution only, which shows that the two-decoders architecture helps already.

METHODS	L1	SSIM
Chen [4]	0.0769	0.9099
Our baseline architecture	0.0611	0.9228
Our final arch	<b>0.0584</b>	<b>0.9256</b>
w/o skip connections	0.0949	0.8971
w/o $conv0$	0.0654	0.9166
w/o multi-scale loss	0.0602	0.9231
w/o $\mathcal{L}_{VGG}$	0.0603	0.9236
w/o $\mathcal{L}_{depth}$	0.1109	0.8862
w/o $\mathcal{L}_{edge}$	0.0618	0.9212

from skip connections. Other numbers and qualitative results show the selected loss functions are both useful for boosting performance. The multi-scale reconstruction loss help to generate thin structures better (like the chair leg in the 1st and 3rd row of the 4th column). The  $\mathcal{L}_{VGG}$  makes the results sharper. The skip connection from  $conv0$  maintains the detailed features (like the patterns for 2nd row). Both  $\mathcal{L}_{edge}$  and  $\mathcal{L}_{depth}$  regularize the depth prediction, especially the depth consistent loss  $\mathcal{L}_{depth}$ . The qualitative results show that without  $\mathcal{L}_{depth}$  the synthesis target images will suffer from distortion because of the unstable quality of predicted depth maps.

#### 4.5. Depth estimation

For the selected datasets we used for evaluation, since the accuracy of the predicted depth maps affect the most on ShapeNet chairs, we evaluate four depth metrics on ShapeNet chairs. L1-all compute the mean absolute difference, L1-rel compute the mean absolute relative difference  $L1-rel = \frac{1}{n} \sum_i |gt_i - pred_i|/gt_i$ , and L1-inv metric is mean absolute difference in inverse depth  $L1-inv = \frac{1}{n} \sum_i |gt_i^{-1} - pred_i^{-1}|$ . Except L1 metrics, we also utilize sc-inv  $= (\frac{1}{n} \sum z_i^2 - \frac{1}{n^2} (\sum z_i)^2)^{\frac{1}{2}}$ , where  $z_i = \log(pred_i) - \log(gt_i)$ . L1-rel normalizes the error, L1-inv puts more importance to close-range depth values, and sc-inv metric is scale-invariant. Table 4 shows that our predicted depth is more accurate compared to [4], which also explains why we can achieve better results than their method. Fig. 6 also visualized the predicted map as point clouds from different viewing angles, which shows that our predicted depth map is less distorted.

Table 4: Depth estimation results on ShapeNet chairs. L1-all compute the mean absolute difference, L1-rel normalizes the error, L1-inv puts more importance to close-range depth values, and sc-inv metric is scale-invariant.

	L1-ALL	L1-REL	L1-INV	SC-INV
Chen [4]	0.0707	0.0360	0.0189	0.0583
Ours	<b>0.0610</b>	<b>0.0305</b>	<b>0.0161</b>	<b>0.0523</b>

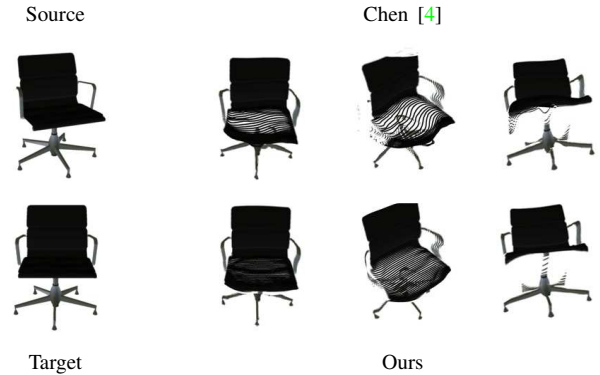


Figure 6: Unsupervised depth prediction results that are visualized as point clouds depicted from different viewing angles. The left col shows the source input and the target view. The right cols show the comparison of the predicted depth map of the target view (The upper row shows results from [4] and the bottom row shows our results).

## 5. Conclusion and Discussion

In this paper, we propose an image generation pipeline that can take advantage of explicit correspondences. We predict the depth map of the target view from a single input view and warp the feature maps of the input view. The warping enables the skip connections to transfer low-level details, so our method can produce clear predictions. Experiment results show that our methods perform better than warping methods and pixel generation methods, alleviating distortion and blurry issues.

Currently for depth prediction, the TAE architecture can only provide a coarse depth map without skip connections, which cannot get correct predictions for thin structures like the arm of chairs. Investigating how to obtain accurate estimation for thin structures can be the future work to further improve the performance.

The code for the experiments can be found at <https://github.com/AaltoVision/warped-skipconnection-nvs>.

**Acknowledgements** We acknowledge the computational resources provided by the Aalto Science-IT project. This research was supported by the Academy of Finland grants 324345 and 309902.



## References

- [1] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001.
- [2] Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems (NIPS)*, pages 10369–10380, 2018.
- [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.
- [4] Xu Chen, Jie Song, and Otmar Hilliges. Monocular neural image based rendering with continuous view control. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4090–4100, 2019.
- [5] Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014.
- [6] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7781–7790, 2019.
- [7] 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 *European Conference on Computer Vision (ECCV)*, pages 628–644. Springer, 2016.
- [8] Paul E Debevec, Camillo J Taylor, and Jitendra 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, 1996.
- [9] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5515–5524, 2016.
- [10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [11] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017.
- [12] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996.
- [13] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.
- [14] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2155–2163, 2017.
- [15] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems (NIPS)*, pages 365–376, 2017.
- [16] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3907–3916, 2018.
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [18] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.
- [19] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7588–7597, 2019.
- [20] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4531–4540, 2019.
- [21] Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. Transformable bottleneck networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7648–7657, 2019.
- [22] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, pages 3500–3509, 2017.
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [24] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998.
- [25] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. International Society for Optics and Photonics, 2000.
- [26] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1121–1132, 2019.
- [27] 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 *European Conference on Computer Vision (ECCV)*, pages 155–171, 2018.

- [28] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision (ECCV)*, pages 322–337. Springer, 2016.
- [29] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 551–560, 2020.
- [30] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2634, 2017.
- [31] Peter Vangorp, Christian Richardt, Emily A Cooper, Gaurav Chaurasia, Martin S Banks, and George Drettakis. Perception of perspective distortions in image-based rendering. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.
- [32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [33] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7467–7477, 2020.
- [34] 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*, pages 1099–1107, 2015.
- [35] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017.
- [36] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, pages 1–12, 2018.
- [37] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European Conference on Computer Vision (ECCV)*, pages 286–301. Springer, 2016.
- [38] 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)*, pages 118–129, 2018.