Transformation-Grounded Image Generation Network for Novel 3D View Synthesis

Eunbyung Park[1] Jimei Yang[2] Ersin Yumer[2] Duygu Ceylan[2] Alexander C. Berg[1]
[1]University of North Carolina at Chapel Hill [2]Adobe Research
eunbyung@cs.unc.edu {jimyang,yumer,ceylan}@adobe.com aberg@cs.unc.edu

Abstract

We present a transformation-grounded image generation network for novel 3D view synthesis from a single image. Our approach first explicitly infers the parts of the geometry visible both in the input and novel views and then casts the remaining synthesis problem as image completion. Specifically, we both predict a flow to move the pixels from the input to the novel view along with a novel visibility map that helps deal with occulsion/disocculsion. Next, conditioned on those intermediate results, we hallucinate (infer) parts of the object invisible in the input image. In addition to the new network structure, training with a combination of adversarial and perceptual loss results in a reduction in common artifacts of novel view synthesis such as distortions and holes, while successfully generating high frequency details and preserving visual aspects of the input image. We evaluate our approach on a wide range of synthetic and real examples. Both qualitative and quantitative results show our method achieves significantly better results compared to existing methods.

1. Introduction

We consider the problem of novel 3D view synthesis—given a single view of an object in an arbitrary pose, the goal is to synthesize an image of the object after a specified transformation of viewpoint. It has a variety of practical applications in computer vision, graphics, and robotics. As an image-based rendering technique [21], it allows placing a virtual object on a background with a desired pose or manipulating virtual objects in the scene [22]. Also, multiple generated 2D views form an efficient representation for 3D reconstruction [37]. In robotics, synthesized novel views give the robot a better understanding of unseen parts of the object through 3D reconstruction, which will be helpful for grasp planning [41].

This problem is generally challenging due to unspecified input viewing angle and the ambiguities of 3D shape observed in only a single view. In particular inferring the appearances of unobserved parts of the object that are not visible in the input view is necessary for novel view synthesis. Our approach attacks all of these challenges, but our contributions focus on the later aspect, dealing with disoccluded appearance in novel views and outputting highly-detailed synthetic images.

Given the eventual approach we will take, using a carefully constructed deep network, we can consider related work on dense prediction with encoder-decoder methods to see what makes the structure of the novel 3D view synthesis problem different. In particular, there is a lack of pixel-to-pixel correspondences between the input and output view. This, combined with large chunks of missing data due to occlusion, makes novel view synthesis fundamentally different than other dense prediction or generation tasks that have shown promising results with deep networks [31, 7, 20]. Although the input and desired output views may have similar low-level image statistics, enforcing such constraints directly is difficult. For example, skip or residual connections, are not immediately applicable as the input and output have significantly different global shapes. Hence, previous 3D novel view synthesis approaches [49, 37] have not been able to match the visual quality of geometry-based methods that exploit strong correspondence.

The geometry-based methods are an alternative to pure generation, and have been demonstrated in [17, 22, 34]. Such approaches estimate the underlying 3D structure of the object and apply geometric transformation to pixels in the input (e.g. performing depth-estimation followed by 3D transformation of each pixel [13]). When successful, geometric transformation approaches can very accurately transfer original colors, textures, and local features to corresponding new locations in the target view. However, such approaches are fundamentally unable to hallucinate where new parts are revealed due to disocclusion. Furthermore, even for the visible geometry precisely estimating the 3D shape or equivalently the precise pixel-to-pixel correspondence between input and synthesized view is still challenging and failures can result in distorted output images.

Project page: http://www.cs.unc.edu/~eunbyung/tvsn

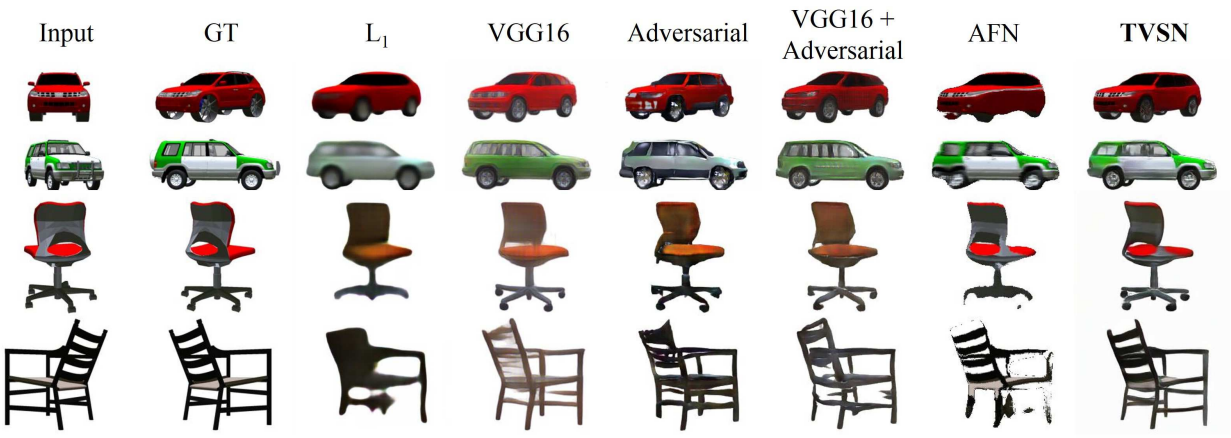

Figure 1. Results on test images from 3D ShapeNet dataset [4]. 1st-input, 2nd-ground truth. From 3rd to 6th are deep encoder-decoder networks with different losses. (3rd-L_1 norm [37], 4th-feature reconstruction loss with pretrained VGG16 network [20, 26, 38, 25], 5th-adversarial loss with feature matching [14, 33, 35, 6], 6th-the combined loss). 7th-appearance flow network (AFN) [51]. **8th-ours(TVSN)**.

In order to bring some of the power of explicit correspondence to deep-learning-based generation of novel views, the recent appearance flow network (AFN) [51] trains a convolutional encoder-decoder to learn how to move pixels without requiring explicit access to the underlying 3D geometry. Our work goes further in order to integrate more explicit reasoning about 3D transformation, hallucinate missing sections, and clean-up the final generated image producing significant improvements of realism, accuracy, and detail for synthesized views.

To achieve this we present *a holistic approach to novel view synthesis by grounding the generation process on viewpoint transformation*. Our approach first predicts the transformation of existing pixels from the input view to the view to be synthesized, as well as a visibility map, exploiting the learned view dependency. We use the transformation result matted with the predicted visibility map to condition the generation process. The image generator not only hallucinates the missing parts but also refines regions that suffer from distortion or unrealistic details due to the imperfect transformation prediction. This holistic pipeline alleviates some difficulties in novel view synthesis by explicitly using transformation for the parts where there are strong cues.

We propose an architecture composed of two consecutive convolutional encoder-decoder networks. First, we introduce a disocclusion aware appearance flow network (DOAFN) to predict the visibility map and the intermediate transformation result. Our second encoder-decoder network is an image completion network which takes the matted transformation as an input and completes and refines the novel view with a combined adversarial and feature-reconstruction loss. A wide range of experiments on synthetic and real images show that the proposed technique achieves significant improvement compared to existing methods. Our main contributions are:

- We propose a holistic image generation pipeline that explicitly predicts how pixels from the input will be transformed and *where there is disocclusion* in the output that needs to be filled, converting the remaining synthesis problem into one of image completion and repair.
- We design a disocclusion aware appearance flow network that relocates existing pixels in the input view along with predicting a visibility map.
- We show that using loss networks with a term considering how well recognition-style features are reconstructed, combined with L_1 loss on pixel values during training, improves synthesized image quality and detail.

2. Related Work

Geometry-based view synthesis. A large body of work benefits from implicit or explicit geometric reasoning to address the novel view synthesis problem. When multiple images are available, multi-view stereo algorithms [12] are applicable to explicitly reconstruct the 3D scene which can then be utilized to synthesize novel views. An alternative approach recently proposed by Flynn et al. [11] uses deep networks to learn to directly interpolate between neighboring views. Ji et al. [19] propose to rectify the two view images first with estimated homography by deep networks, and then synthesize middle view images with another deep networks. In case of single input view, Garg et al. [13] propose to first predict a depth map and then synthesize the novel view by transforming each reconstructed 3D point in the depth map. However, all these approaches only utilize the information available in the input views and thus fail in case of disocclusion. Our method, on the other hand, not

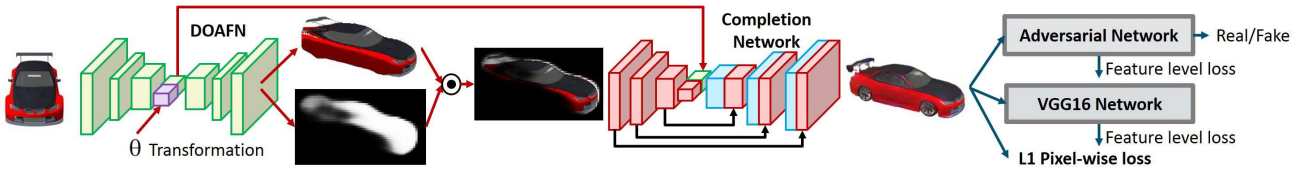

Figure 2. Transformation-grounded view synthesis network(TVSN). Given an input image and a target transformation (3.1), our disocclusion-aware appearance flow network (DOAFN) transforms the input view by relocating pixels that are visible both in the input and target view. The image completion network, then, performs hallucination and refinement on this intermediate result(3.2). For training, the final output is also fed into two different loss networks in order to measure similarity against ground truth target view (3.2).

only takes advantage of implicit geometry estimation but also infers the parts of disocclusion.

Another line of geometry-based methods utilize large internet collections of 3D models which are shown to cover wide variety for certain real world object categories [22, 34]. Given an input image, these methods first identify the most similar 3D model in a database and fit to the image either by 3D pose estimation [34] or manual interactive annotation [22]. The 3D information is then utilized to synthesize novel views. While such methods generate high quality results when sufficiently similar 3D models exist, they are often limited by the variation of 3D models found in the database. In contrast, our approach utilizes 3D models only for training generation networks that directly synthesize novel views from an image.

Image generation networks. One of the first convolutional networks capable of generating realistic images of objects is proposed in [8], but the network requires explicitly factored representations of object type, viewpoint and color, and thus is not able to generalize to unseen objects. The problem of generating novel views of an object from a single image is addressed in [49, 23, 37] using deep convolutional encoder-decoder networks. Due to the challenges of disentangling the factors from single-view and the use of globally smooth pixel-wise similarity measures (e.g. L_1 or L_2 norm), the generation results tend to be blurry and low in resolution.

An alternative to learning disentangled or invariant factors is the use of equivariant representations, i.e. transformations of input data which facilitate downstream decision making. Transforming auto-encoders are coined by Hinton et al. [16] to learn both 2D and 3D transformations of simple objects. Spatial transformer networks [18] further introduce differentiable image sampling techniques to enable in-network parameter-free transformations. In the 3D case, flow fields are learned to transform input 3D mesh to the target shape [50] or input view to the desired output view [51]. However, direct transformations are clearly upper-bounded by the input itself. To generate novel 3D views, our work grounds a generation network on the learned transformations to hallucinate disoccluded pixels.

Recently, a number of image generation methods introduce the idea of using pre-trained deep networks as loss function, referred as perceptual loss, to measure the feature similarities from multiple semantic levels [20, 26, 38, 25]. The generation results from these works well preserve the object structure but are often accompanied with artifacts such as aliasing. At the same time, generative adversarial networks [14, 33], introduce a discriminator network, which is adversarially trained with the generator network to tell apart the generated images from the real ones. The discriminator encapsulates natural image statistics of all orders in a real/fake label, but its min-max training often leads to local minimum, and thus local distortions or painting-stroke effects are commonly observed in their generated images. Our work uses a combined loss function that takes advantages of both the structure-preserving property of perceptual loss and the rich textures of adversarial loss (See Fig. 1).

Deep networks have also been explored for image completion purposes. Examples of proposed methods include image in-painting with deep networks [32] and sequential parts-by-parts generation for image completion [24]. Such methods assume the given partial input is correct and focus only on completion. In our case, however, we do not have access to a perfect intermediate result. Instead, we rely on the generation network both to hallucinate missing regions and also refine any distortions that occur due to inaccurate per-pixel transformation prediction.

3. Transformation-Grounded View Synthesis

Novel view synthesis could be seen as a combination of the following three scenarios: 1) pixels in the input view that remain visible in the target view are moved to their corresponding positions; 2) remaining pixels in the input view disappear due to occlusions; and 3) previously unseen pixels are revealed or disoccluded in the target view. We replicate this process via a neural network as shown in Figure 2. Specifically, we propose a disocclusion-aware appearance flow network (3.1) to transform the pixels of the input view that remain visible. A subsequent generative completion network (3.2) then hallucinates the unseen pixels of the target view given these transformed pixels.

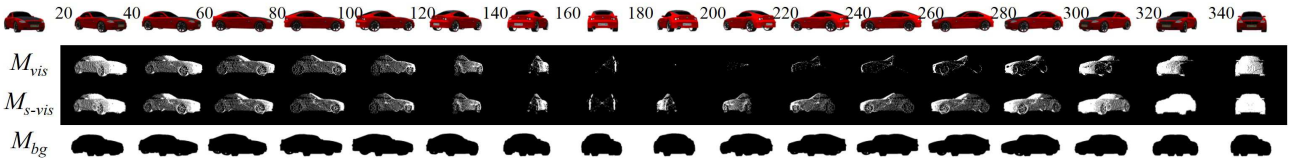

Figure 3. Visibility maps for different rotations: the first column in the first row is an input image. Remaining columns show output images and corresponding masks for rotations from 20 to 340 degrees in 20 degree intervals. The second, third and fourth rows show visibility maps M_{vis}, symmetry-aware visibility maps $M_{\text{s-vis}}$, and background masks M_{bg}, respectively. The input image is in the pose of 0 elevation and 20 azimuth. The visibility maps for the rotations from 160 to 340 show the largest difference between M_{vis} and $M_{\text{s-vis}}$. For example, $M_{\text{s-vis}}$ shows the opposite side of the car as visible and allows it to be filled in by the network based on the visible side.

3.1. Disocclusion-aware Appearance Flow Network

Recently proposed appearance flow network (AFN) [51] learns how to move pixels from an input to a target view. The key component of the AFN is a differentiable image sampling layer introduced in [18]. Precisely, the network first predicts a dense flow field that maps the pixels in the target view, I_{t}, to the source image, I_{s}. Then, sampling kernels are applied to get the pixel value for each spatial location in I_{t}. Using a bilinear sampling kernel, the output pixel value at spatial location $I_{\text{t}}^{i,j}$ equals to:

$$\sum_{(h,w)\in N} I_{\text{s}}^{h,w} \max(0, 1-|F_y^{i,j}-h|) \max(0, 1-|F_x^{i,j}-w|), \tag{1}$$

where F is the flow predicted by the deep convolutional encoder-decoder network (see the first half of Figure 2). $F_x^{i,j}$ and $F_y^{i,j}$ indicate the x and y coordinates of one target location. N denotes the 4-pixel neighborhood of $(F_y^{i,j}, F_x^{i,j})$.

The key difference between our disocclusion aware appearance flow network (DOAFN) and the AFN is the prediction of an additional visibility map which encodes the parts that need to be removed due to occlusion. The original AFN synthesizes the entire target view, including the disoccluded parts, with pixels of the input view, e.g. 1st row of AFN results in Figure 1. However, such disoccluded parts might get filled with wrong content, resulting in implausible results, especially for cases where a large portion of the output view is not seen in the input view. Such imperfect results would provide misleading information to a successive image generation network. Motivated by this observation, we propose to predict a visibility map that masks such problematic regions in the transformed image:

$$I_{\text{doafn}} = I_{\text{afn}} \odot M_{\text{vis}}, \tag{2}$$

where $M_{\text{vis}} \in [0,1]^{H\times W}$. To achieve this, we define the ground truth visibility maps according to the 3D object geometry as described next.

Visibility map. Let $M_{\text{vis}} \in \mathbb{R}^{H\times W}$ be the visibility map for the target view, given source image I_{s} and desired transformation parameter θ. The mapping value for a pixel in the target view corresponding to a spatial location (i,j) in I_{s} is defined as follows:

$$M_{\text{vis}}^{(PR(\theta)\mathbf{x}_{\text{s}}^{(i,j)})^h,(PR(\theta)\mathbf{x}_{\text{s}}^{(i,j)})^w} = \begin{cases} 1 & \mathbf{c}^\top R(\theta)\mathbf{n}_{\text{s}}^{(i,j)} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$\mathbf{x}_{\text{s}}^{(i,j)} \in \mathbb{R}^4$ is the 3D object coordinates and $\mathbf{n}_{\text{s}}^{(i,j)} \in \mathbb{R}^4$ is the surface normal corresponding to location (i,j) in I_{s}, both represented in homogeneous coordinates. Since we use synthetic renderings of 3D CAD models, we have access to ground truth object coordinates and surface normals. $R(\theta) \in \mathbb{R}^{3\times 4}$ is the rotation matrix given the transformation parameter θ and $P \in \mathbb{R}^{3\times 3}$ is the perspective projection matrix. The superscripts h and w denote the target image coordinates in y and x axis respectively after perspective projection. $\mathbf{c} \in \mathbb{R}^3$ is the 3D camera center. In order to compute the target image coordinates for each pixel in I_{s}, we first obtain the 3D object coordinates corresponding to this pixel and then apply the desired 3D transformation and perspective projection. The mapping value of the target image coordinate is 1 if and only if the dot product between the viewing vector and surface normal is positive, i.e. the corresponding 3D point is pointing towards the camera.

Symmetry-aware visibility map. Many common object categories exhibit reflectional symmetry, e.g. cars, chairs, tables etc. AFN implicitly exploits this characteristic to ease the synthesis of large viewpoint changes. To fully take advantage of symmetry in our DOAFN, we propose to use a symmetry-aware visibility map. Assuming that objects are symmetric with respect to the xy-plane, a symmetry-aware visibility map M_{sym} is computed by applying Equation 3 to the z-flipped object coordinates and surface normals. The final mapping for a pixel in the target view corresponding to spatial location (i,j) is then defined as:

$$M_{\text{s-vis}}^{i,j} = \mathbb{1}\left[M_{\text{sym}}^{i,j} + M_{\text{vis}}^{i,j} > 0\right] \tag{4}$$

Background mask. Explicit decoupling of the foreground object is necessary to deal with real images with natural background. In addition to parts of the object being

disoccluded in the target view, different views of the object occlude different portions of the background posing additional challenges. For example, transforming a side view to be frontal exposes parts of the background occluded by the two ends of the car. In our approach, we define the *foreground* as the region that covers pixels of the object in both input view and output view. The rest of the image belongs to the *background* and should remain unchanged in both views. We thus introduce a unified background mask,

$$M_{\text{bg}}^{i,j} = \mathbb{1}\left[B_{\text{s}}^{i,j} + B_{\text{t}}^{i,j} > 0\right], \tag{5}$$

where B_{s} and B_{t} are the background masks of the source and target images respectively. Ground truth background masks are easily obtained from 3D models. Examples of background masks are presented in Figure 3. When integrated with the (symmetry-aware) visibility map, the final output of DOAFN becomes:

$$I_{\text{doafn}} = I_{\text{s}} \odot M_{\text{bg}} + I_{\text{afn}} \odot M_{\text{s-vis}} \tag{6}$$

3.2. View Completion Network

Traditional image completion or hole filling methods often exploit local image information [9, 2, 45] and have shown impressive results for filling small holes or texture synthesis. In our setting, however, sometimes more than half of the content in the novel view is not visible in the input image, constituting a big challenge for local patch based methods. To address this challenge, we propose another encoder-decoder network, capable of utilizing both local and global context, to complete the transformed view inferred by DOAFN.

Our view completion network is composed of an "hourglass" architecture similar to [30], with a bottleneck-to-bottleneck identity mapping layer from DOAFN to the hourglass (see Figure 2). This network has three essential characteristics. First, being conditioned on the high-level features of DOFAN, it can generate content that have consistent attributes with the given input view, especially when large chunk of pixels are dis-occluded. Second, the output of DOAFN is already in the desired viewpoint with important low-level information, such as colors and local textures, preserved under transformation. Thus, it is possible to utilize skip connections to propagate this low-level information from the encoder directly to later layers of the decoder. Third, the view completion network not only hallucinates disoccluded regions but also fixes artifacts such as distortions or unrealistic details. The output quality of DOAFN heavily depends on the input viewpoint and desired transformation, resulting in imperfect flow in certain cases. The encoder-decoder nature of the image generation network is well-suited to fix such cases. Precisely, while the encoder is capable of recognizing undesired parts in the DOAFN output, the decoder refines these parts with realistic content.

Loss networks. The idea of using deep networks as a loss function for image generation has been proposed in [26, 38, 20, 6]. Precisely, an image generated by a network is passed as an input to an accompanied network which evaluates the discrepancy (the feature distance) between the generation result and ground truth. We use the *VGG16* network for calculating the feature reconstruction losses from a number of layers, which is referred as *perceptual loss*. We tried both a pre-trained loss network and a network with random weights as suggested in [15, 39]. However, we got perceptually poor results with random weights, concluding that the weights of the loss network indeed matter.

On the other hand, adversarial training [14] has been phenomenally successful for training the loss network at the same time of training the image generation network. We experimented with a similar adversarial loss network as in [33] while adopting the idea of feature matching presented in [35] to make the training process more stable.

We realized that the characteristics of generated images with these two kinds of loss networks, perceptual and adversarial, are complementary. Thus, we combined them together with the standard image reconstruction loss (L_1) to maximize performance. Finally, we added total variation regularization term [20], which was useful to refine the image:

$$\begin{aligned} &-\log D(G(I_{\text{s}})) + \alpha L_2(F_D(G(I_{\text{s}})), F_D(I_{\text{t}}))+ \\ &\beta L_2(F_{vgg}(G(I_{\text{s}})), F_{vgg}(I_{\text{t}})) + \gamma L_1(I_{\text{s}}, I_{\text{t}}) + \lambda L_{TV}(G(I_{\text{s}})) \end{aligned} \tag{7}$$

I_{s}, $G(I_{\text{s}})$ and I_{t} is the input, generated output and corresponding target image, respectively. $\log(D)$ is log likelihood of generated image $G(I_{\text{s}})$ being a real image, estimated by adversarially trained loss network, called discriminator D. In practice, minimizing $-\log D(G(I_{\text{s}}))$ has shown better gradient behaviour than minimizing $\log D(1-G(I_{\text{s}}))$.

F_D and F_{vgg} are the features extracted from the discriminator and *VGG16* loss networks respectively. We found that concatenated features from the first to the third convolutional layers are the most effective. L_1 and L_2 are ℓ_1 and ℓ_2 norms of two same size inputs divided by the size of the inputs. In sum, both generated images $G(I_{\text{s}})$ and ground truth image I_{t} are fed into D and *VGG16* loss networks, and we extract the features, and compute averaged euclidean distance between these two.

The discriminator D is simultaneously trained along with G via alternative optimization scheme proposed in [14]. The loss function for the discriminator is

$$-\log D(I_{\text{s}}) - \log(1 - D(G(I_{\text{s}}))) \tag{8}$$

We empirically found that $\alpha = 100$, $\beta = 0.001$, $\gamma = 1$, and $\lambda = 0.0001$ are good hyper-parameters and fixed them for the entire experiments.

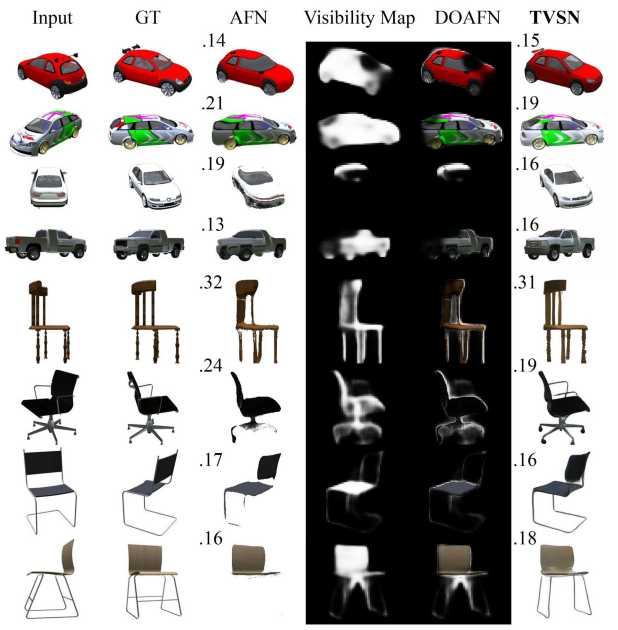

Figure 4. Results on synthetic data from ShapeNet. We show the input, ground truth output (GT), results for AFN and our method (TVSN) along with the L_1 error. We also provide the intermediate output (visibility map and output of DOAFN).

4. Experiments

4.1. Training Setup

We use rendered images of 3D models from ShapeNet [4] both for training and testing. We use the entire *car* category (7497 models) and a subset of the *chair* category (698 models) with sufficient texture. For each model, we render images from a total of 54 viewpoints corresponding to 3 different elevations (0, 10, and 20) and 18 azimuth angles (sampled in the range $[0, 340]$ with 20-degree increments). The desired transformation is encoded as a 17-D one-hot vector corresponding to one of the rotation angles between input and output views in the range $[20, 340]$. Note that we did not encode 0 degree as it is the identical mapping. For each category, 80% of 3D models are used for training, which leaves over 5 million training pairs (input view-desired transformation) for the car category and 0.5 million for the chair category. We randomly sample input viewpoints, desired transformations from the rest 20% of 3D models to generate a total of $20,000$ testing instances for each category. Both input and output images are of size $256 \times 256 \times 3$.

We first train DOAFN, and then the view completion network while DOAFN is fixed. After the completion network fully converges, we fine-tune both networks end-to-end. However, this last fine-tuning stage does not show notable improvements. We use mini-batches of size 25 and 15 for DOAFN and the completion network respectively. The learning rate is initialized as 10^{-4} and is reduced to 10^{-5} after 10^5 iterations. For adversarial training, we adjust the update schedule (two iterations for generator and one iteration for discriminator in one cycle) to balance the discriminator and the generator.

Table 1. We compare our method (*TVSN(DOAFN)*) to several baselines: (i) a single-stage encoder-decoder network trained with different loss functions: L_1 (L_1), feature reconstruction loss using VGG16 (*VGG16*), adversarial (*Adv*), and combination of the latter two (*VGG16+Adv*), (ii) a variant of our approach that does not use a visibility map (*TVSN(AFN)*).

	car		chair	
	L_1	SSIM	L_1	SSIM
L_1[37]	.168	.884	.248	**.895**
VGG	.228	.870	.283	**.895**
Adv	.208	.865	.241	.885
VGG+Adv	.194	.872	.242	.888
AFN[51]	.146	.906	.240	.891
TVSN(AFN)	**.132**	**.910**	**.229**	**.895**
TVSN(DOAFN)	**.133**	**.910**	**.230**	**.894**

4.2. Results

We discuss our main findings in the rest of this section and refer the reader to the supplementary material for more results. We utilize the standard L_1 mean pixel-wise error and the structural similarity index measure (SSIM) [44, 28] for evaluation. When computing the L_1 error, we normalize the pixel values resulting in errors in the range $[0, 1]$, lower numbers corresponding to better results. SSIM is in the range $[-1, 1]$ where higher values indicate more structural similarity.

Comparisons. We first evaluate our approach on synthetic data and compare to AFN. Figure 4 shows qualitative results.[1] We note that while our method completes the disoccluded parts consistently with the input view, AFN generates unrealistic content (front and rear parts of the cars in the 1st and 2nd rows). Our method also corrects geometric distortions induced by AFN (3rd and 4th rows) and better captures the lighting (2nd row). For the chair category, AFN often fails to generate thin structures such as legs due to the small number of pixels in these regions contributing to the loss function. On the other hand, both perceptual and adversarial loss help to complete the missing legs as they contribute significantly to the perception of the overall shape. In order to evaluate the importance of the visibility map, we compare against a variant of our approach which directly provides the output of AFN to the view completion network without masking. (For clarity, we will refer to our method

[1]The results from the original AFN [51] paper are not directly comparable due to the different image size. In addition, since the complete source code was not available at the time of paper submission, we re-implemented this method by consulting the authors.

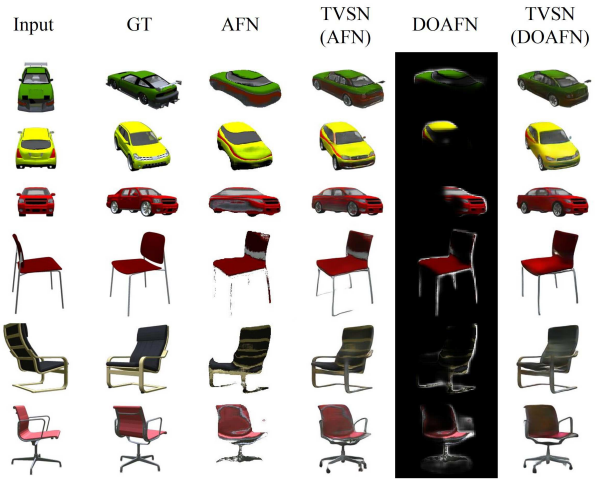

Figure 5. When a visibility map is not utilized (TVSN(AFN)), severe artifacts observed in the AFN output get integrated into the final results. By masking out such artifacts, our method (TVSN(DOAFN)) relies purely on the view completion network to generate plausible results.

as *TVSN(DOAFN)* and to this baseline as *TVSN(AFN)*.) Furthermore, we also implement a single-stage convolutional encoder-decoder network as proposed in [37] and train it with various loss functions: L_1 loss (L_1), feature reconstruction loss using VGG16 (VGG16), adversarial loss (Adv), and combination of the latter two (VGG16+Adv). We provide quantitative and visual results in Table 1 and Figure 1 respectively. We note that, although commonly used, L_1 and SSIM metrics are not fully correlated with human perception. While our method is clearly better than the L_1 baseline [37], both methods get comparable SSIM scores.

We observe that both TVSN(AFN) and TVSN(DOAFN) perform similarly with respect to L_1 and SSIM metrics demonstrating that the view completion network in general successfully refines the output of AFN. However, in certain cases severe artifacts observed in the AFN output, especially in the disoccluded parts, get smoothly integrated in the completion results as shown in Figure 5. In contrast, the visibility map masks out those artifacts and thus TVSN(DOAFN) relies completely on the view completion network to hallucinate these parts in a realistic and consistent manner.

Evaluation of the Loss Networks. We train our network utilizing the feature reconstruction loss of VGG16 and the adversarial loss. We evaluate the effect of each loss by training our network with each of them only and provide visual results in Figure 6. It is well-known that the adversarial loss is effective in generating realistic and sharp images as opposed to standard pixel-wise loss functions. However, some artifacts such as colors and details inconsistent with the input view are still observed. For the VGG16 loss, we experimented with different feature choices and empirically found that the combination of the features from the first three layers with total variation regularization is the most effective. Although the VGG16 perceptual loss is capable of generating high quality images for low-level tasks such as super-resolution, it has not yet been fully explored for pure image generation tasks as required for hallucinating disoccluded parts. Thus, this loss still suffers from the blurry output problem whereas combination of both VGG16 and adversarial losses results in the most effective configuration.

Figure 6. We evaluate the effect of using only parts of our system, VGG16 in TVSN(VGG16), and adversarial loss in TVSN(Adversarial), as opposed to our method, TVSN(VGG16+Adversarial) that uses both.

4.3. 360 degree rotations and 3D reconstruction

Inferring 3D geometry of an object from a single image is the holy-grail of computer vision research. Recent approaches using deep networks commonly use a voxelized 3D reconstruction as output [5, 46]. However, computational and spatial complexities of using such voxelized representations in standard encoder-decoder networks significantly limits the output resolution, e.g. 32^3 or 64^3.

Inspired by [37], we exploit the capability of our method in generating novel views for reconstruction purposes. Specifically, we generate multiple novel views from the input image to cover a full 360 rotation around the object sampled at 20-degree intervals. We then run a multi-view reconstruction algorithm [12] on these images using the ground truth relative camera poses to obtain a dense point cloud. We use the open source OpenMVS library [1] to reconstruct a textured mesh from this point cloud. Figure 7 shows multi-view images generated by AFN and our method whereas Figure 8 demonstrates examples of reconstructed 3D models from these images. By generating views consistent in terms of geometry and details, our method results in significantly better quality textured meshes.

4.4. 3D Object Rotations in Real Images

In order to generalize our approach to handle real images, we generate training data by compositing synthetic renderings with random backgrounds [36]. We pick 10,000 random images from the SUN397 dataset[36] and randomly

Figure 7. Results of 360 degree rotations

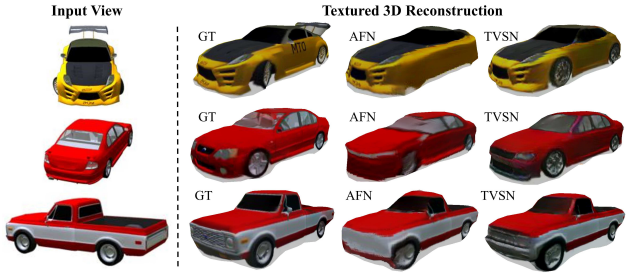

Figure 8. We run a multi-view stereo algorithm to generate textured 3D reconstructions from a set of images generated by AFN and our TVSN approach. We provide the reconstructions obtained from ground truth images (GT) for reference.

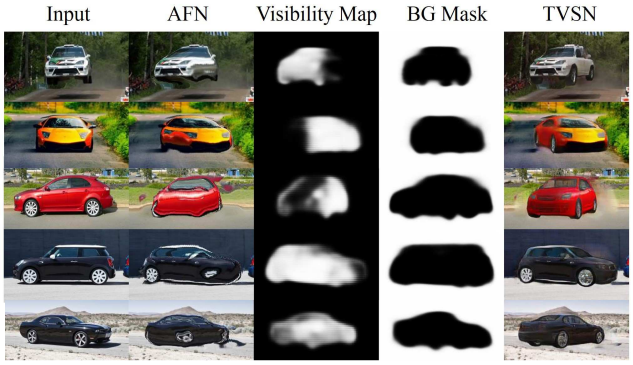

Figure 9. We show novel view synthesis results on real internet images along with the predicted visibility map and the background mask.

crop them to be of size $256 \times 256 \times 3$. Although this simple approach fails to generate realistic images, e.g. due to inconsistent lighting and viewpoint, it is effective in enabling the network to recognize the contours of the objects in complex background. In Figure 9, we show several novel view synthesis examples from real images obtained from the internet.

While our initial experiments show promising results, further investigation is necessary to improve performance. Most importantly, more advanced physically based rendering techniques are required to model complex light interactions in the real world (e.g. reflections from the environment onto the object surface). In addition, it is necessary to sample more viewpoints (both azimuth and elevation) to handle viewpoint variations in real data. Finally, to provide a seamless break from the original image, an object segmentation module is desirable so that the missing pixels in background can be separately filled in by alternative methods, such as patch-based inpainting methods [2] or pixel-wise autoregressive models [40].

5. Conclusion and Future Work

We present a novel transformation-grounded image generation network. Our method generates realistic images and outperforms existing techniques for novel 3D view synthesis on standard datasets of CG renderings where ground truth is known. Our synthesized images are even accurate enough to perform multi-view 3D reconstruction. We further show successful results for real photographs collected from the web, demonstrating that the technique is robust.

We observed that some structures in the generated novel views, such as headlights and wheels of cars, would consistently resemble common base shapes. This is more apparent if such structures are not observed in the input view. We believe the reason is the inherently deterministic nature of our encoder-decoder architecture, which can be alleviated by incorporating approaches like explicit diverse training [27] or probabilistic generative modeling [47, 48, 29, 43].

We hope that the proposed image generation pipeline might potentially help other applications, such as video prediction. Instead of pure generation demonstrated by recent approaches [28, 42], our approach can be applied such that each frame uses a transformed set of pixels from the previous frame[43, 3, 10] where missing pixels are completed and refined by a disocclusion aware completion network, where disocclusion can be learned from motion estimation [43, 10].

Acknowledgement

This work was started as an internship project at Adobe Research and continued at UNC. We would like to thank Weilin Sun, Guilin Liu, True Price, and Dinghuang Ji for helpful discussions. We thank NVIDIA for providing GPUs and acknowledge support from NSF 1452851, 1526367.

References

[1] openmvs: open multi-view stereo reconstruction library. https://github.com/cdcseacave/openMVS. Accessed: 2016-11-14. 7

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Computer Graphics (TOG)*, 2009. 5, 8

[3] B. D. Brabandere, X. Jia, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *NIPS*, 2016. 8

[4] A. X. Chang, T. Funkhouser, L. 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. *arXiv:1512.03012*, 2015. 2, 6

[5] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 7

[6] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 2, 5

[7] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, T. Brox, et al. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1

[8] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015. 3

[9] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. *28th Annual Conference on Computer Graphics and Interactive Techniques*, 2001. 5

[10] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016. 8

[11] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016. 2

[12] Y. Furukawa. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2015. 2, 7

[13] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 1, 2

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2, 3, 5

[15] K. He, Y. Wang, and J. Hopcroft. A powerful generative model using random weights for the deep image representation. In *NIPS*, 2016. 5

[16] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, 2011. 3

[17] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Computer Graphics (TOG)*, 2005. 1

[18] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 3, 4

[19] D. Ji, J. Kwon, M. McFarland, and S. Savarese. Deep view morphing. In *CVPR*, 2017. 2

[20] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1, 2, 3, 5

[21] S. B. Kang and H.-Y. Shum. A review of image-based rendering techniques. 2000. 1

[22] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Computer Graphics (TOG)*, 2014. 1, 3

[23] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015. 3

[24] H. Kwak and B.-T. Zhang. Generating images part by part with composite generative adversarial networks. *arXiv:1607.05387*, 2016. 3

[25] A. Lamb, V. Dumoulin, and A. Courville. Discriminative regularization for generative models. *arXiv:1602.03220*, 2016. 2, 3

[26] A. B. L. Larsen, S. K. Snderby, H. Larochelle, and OleWinther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. 2, 3, 5

[27] S. Lee, S. Purushwalkam, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *NIPS*, 2016. 8

[28] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 6, 8

[29] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2016. 8

[30] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 5

[31] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. 2015. 1

[32] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting deepak. In *CVPR*, 2016. 3

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 3, 5

[34] K. Rematas, C. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars. Novel views of objects from a single image. *arXiv:1602.00328*, 2016. 1, 3

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. chen. Improved techniques for training gans. In *NIPS*, 2016. 2, 5

[36] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2015. 7

[37] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2016. 1, 2, 3, 6, 7

[38] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 2016. 2, 3, 5

[39] I. Ustyuzhaninov, W. Brendel, L. Gatys, and M. Bethge. Texture synthesis using shallow convolutional networks with random filters. *arXiv:1606.00021*, 2016. 5

[40] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 8

[41] J. Varley, C. DeChant, A. Richardson, A. Nair, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. *arXiv:1609.08546*, 2016. 1

[42] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 8

[43] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 8

[44] 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. 6

[45] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *TPAMI*, 2007. 5

[46] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016. 7

[47] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 8

[48] X. Yan, J. Y. K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016. 8

[49] J. Yang, S. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NIPS*, 2015. 1, 3

[50] M. E. Yumer and N. J. Mitra. Learning semantic deformation flows with 3d convolutional networks. In *ECCV*, 2016. 3

[51] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *ECCV*, 2016. 2, 3, 4, 6