Novel View Synthesis by Geometric Transformation and Image Completion Neural Network

Miao Zhang, Xiaobai Ma (CS 236)
miaoz2, maxiaoba@stanford.edu

1 Introduction

The objective of synthesizing novel views is to build a network that can generate images of an object viewed from different angles, given only a single view of that object. It has a lot of practical applications. For example, in computer graphics, it allows better photo editing when it involves rotating an object in images. In virtual reality, it helps with reconstructing more realistic virtual environments. In robotics, it can potentially contributes to long term target tracking when the target comes back into sight in a previously occluded view. Synthesized views can provided a more comprehensive understanding of the appearance of the target. This task is generally challenging due to the ambiguity of 3D object shape given only one single view, especially inferring the unseen parts of the object. [1]

2 Related Work

Many work has been done in the field of synthesizing novel view including both geometry-based and learning-based approaches. Geometry-based methods try to first infer the underlying 3D structure of the object from the input image, and then apply geometric transformation to generate the new image. Such methods are able to achieve accurate view point transformation when the object structure is successfully inferred (1; 2; 3). However, inferring the accurate object structure from a single view is itself a difficult problem, especially when there are hidden parts.

Learning-based methods try to use a neural network to directly generate the new view image (4; 5). The lack of pixel-to-pixel correspondence information as well as the occlusion problem makes it difficult for such methods to generate high quality images.

Our project is mainly based on two papers. (6) introduces the idea of using the neural network to generate an appearance flow vector which "steals" pixels from the input image to generate the new image. Based on (6), (7) further uses an image completion network to improve the quality of the generated image which is conditioned on the output of the appearance flow network. The completion network is trained using GAN loss.

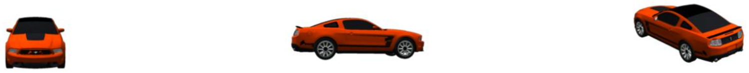
3 Dataset

The input to the system is a source image I_s and a one-hot vector θ specifying the change of the view point, the output is an image generated of the object viewed from the target view point $G(I_s)$. The true target view image is denoted as I_t.

We use the same dataset as in (6) and (7): ShapeNet 3D objects (8), with an open source image render engine (9) to render 2D views of the 3D object from different angle. The render engine also can generate the surface normal and object coordinates which are later used to generate visibility maps, which is an intermediate value in the network generating process. The dataset contains 7900 car

[1] The code of this work can be found at https://github.com/mollyzyy787/tvsn

Preprint. Work in progress.



(a) 0° azimuth and 10° elevation (b) 100° azimuth and 0° elevation (c) 220° azimuth and 20° elevation

Figure 1: Sample rendered view for a car model

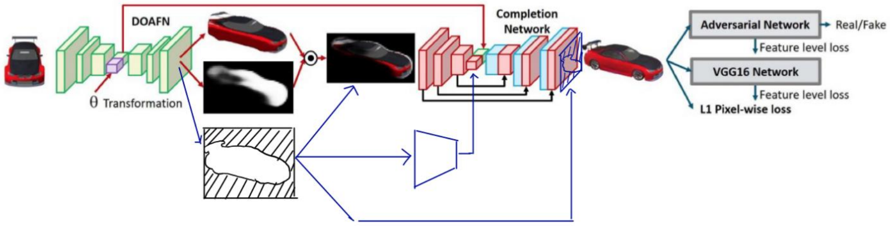


Figure 2: Generator Network Structure

objects, each contains 18×3 images sweeping through azimuth angles from 0 to 340° (20° interval), and elevation angles from 0 to 20° (10° interval).

4 Methods

In this section we briefly introduce the network structure and the training procedure, while highlighting the extension we made to (7). The full generation network composed of two parts. The first part is the appearance flow network noted as dis-occlusion-aware appearance flow network (DOAFN). The second part is the completion network noted as transformation-grounded view synthesis network (TVSN). The two networks are trained sequentially. Fig.2 gives an overview of the network structure as well as the training loss composition.

4.1 Appearance Flow Network

DOAFN takes the source image and the transformation matrix as input and outputs the a dense flow field as well as a visibility mask. The flow field maps the pixels in the target view, I_t, to the source image, I_s. The visibility map, M_{vis}, is a 2D mask specifying which part of the original image is still visible in the target view. In addition, we add a third output which is the predicted object contour M_c in the target view. The intuition is that in the training process, although only the visible parts of the original network could be directly seen in the target view, DOAFN is able to learn more about the object such as the predicted contour. We want to pass this additional information to the completion network to improve the final image quality.

DOAFN is trained in a supervised-learning fashion. The training loss is defined as:

$$L_{doafn} = L_1(I_t, I_{afn}) + BCE(M_{vis}, M_{vis,groundtruth}) + BCE(M_c, M_{c,groundtruth})$$

where I_{afn} is the image generated by applying the appearance flow on the source image (6). The first term is the L1 loss between the target image and the I_{afn}. The second and the third term are binary cross entropy loss of the corresponding output mask with respect to the ground truth.

4.2 Completion Network

The completion network (TVSN) takes $I_{doafn} = I_{afn} \odot M_{vis}$ and M_c as input, and outputs the final target image $G(I_s)$. It has an "hourglass" structure similar to (10) with a bottleneck-to-bottleneck identity mapping layer from DOAFN to TVSN.

The training loss for the generator from (7) is given by:

$$L_g = -logD(G(I_s)) + \alpha L_2(F_D(G(I_s)), F_D(I_t)) + \beta L_2(F_{vgg}(G(I_s)), F_{vgg}(I_t)) \\ + \gamma L_1(G(I_s), I_t) + \lambda L_{TV}(G(I_s)) \quad (1)$$

where I_s, $G(I_s)$, and I_t are the input, generated output and corresponding target image, respectively. $D(G(I_s))$ is the likelihood of $G(I_s)$ being a real image estimated by the discriminator similar as (11). F_D and F_{vgg} are the features extracted from the discriminator and VGG16 loss networks respectively. Finally, L_{TV} is a total variation regularization term used to refine and denoise the image. The discriminator loss is defined as:

$$L_D = -log(I_s) - log(1 - D(G(I_s)))$$

The training was conducted by the standard alternative optimization scheme, using Adam optimization algorithm.

We tested multiple ways of incorporating the additional input M_c into the completion network. First, we directly concatenate M_c with I_{doafn} as the input to the network (model noted as 'maskc'); Seconde, We also tried to let M_c first go through an encoder network and concatenate it with I_{doafn} in the encoded space ('maskp' since two inputs are encoded in parallel); We also used M_c as a mask on the output of the completion network. (The intuition is that the generation of the background is trivial, we wanted to focus the learning on the area where the object appears. The models combining the output mask and the first two extensions are referred to as 'maskco' or 'maskpo' respectively); Finally, noticing that in most cases, the visible parts of the object in both views take a large portion in the target image, we added I_{doafn} to the output of the completion network as the final output. In this way, the completion network is learning the residual between I_{doafn} and I_t. This extension, combined with the 'maskc', is referred to as 'maskcr'.

5 Results

5.1 Appearance Flow Network

With the contour mask of the target view as the additional output compared to the original DOAFN network in (7), the sample output is shown in Fig. 3. The content of each column is annotated in the following table.

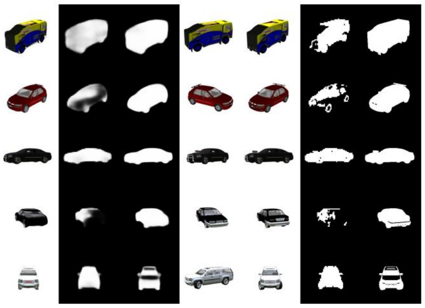


Figure 3: DOAFN output after 200 epoch of training

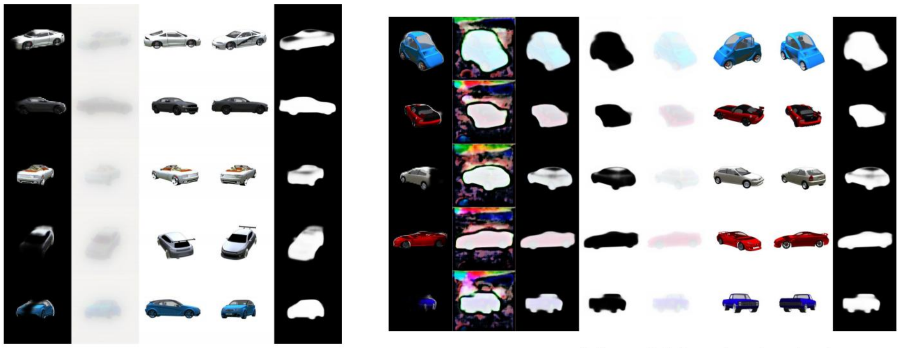
1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}
target view (generated)	visibility map (generated)	contour map (generated)	input view	target view	visibility map (true)	contour map (true)

Table 1: DOAFN output annotation

The training results for the our modified DOAFN network are satisfying as preliminary generation results and inputs to the completion network. Comparing the column 1, 2, 3 with their ground truth images in column 5, 6, 7 in Fig.3, we see that the generated image/masks are close to the ground truth with some blur.

5.2 Completion Network

We added the third column (generated contour map) as an additional input to the completion network (TVSN). There is no significant difference in performance between adding the addition input as an additional channel with the original input image (model 'maskc' and 'maskco') and adding it in parallel with the original input image (model 'maskp' and 'maskpo'), after 100 epoch of training. The reason is probably that the bottleneck-to-bottleneck identity mapping layer already gives the TVSN enough information to infer the object contour. However, we can see that applying the contour mask to the output of TVSN (model 'maskco' and 'maskpo') can result in a sharper separation between the object and the background. Some output samples of model 'maskc' and 'maskco' as well as the loss curves are shown in Fig.4 and 5.



(a) model 'maskc' output

(b) model 'maskco' output

Figure 4: TVSN result after 100 epoch of training

Table 2: TVSN model 'maskc' result annotation

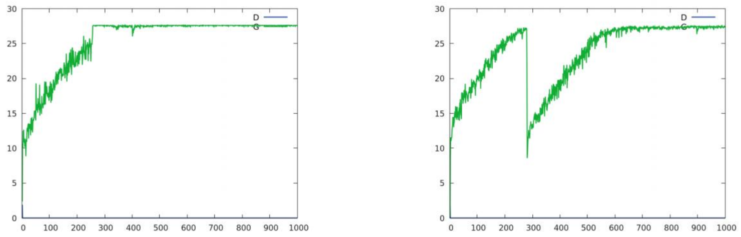
1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
DOAFN output	TVSN generated image	input view	target view	DOAFN generated contour map

Table 3: TVSN model 'maskco' result annotation

1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}
DOAFN output	TVSN output	TVSN output $\odot$ contour map	inverse contour map	generated image (3^{rd} col + 4^{th} col)	input view	target view	DOAFN generated contour map

It can be seen that the generated image after 100 epoch is still very blurry. From the DCGAN loss training curve in Fig.5, we can see that the network suffers from generator instability (shown from the sudden jump in Fig.5 (b)) and diminished gradient.

With the hypothesis that the discriminator is too strong to make the generator learn anything meaningful, we proposed a new model ('maskcr') aiming to strengthen the generator and weaken the discriminator, by adding I_{doadn} to generator output so that the generator is learning the residual between I_{doafn} and I_t, and by reducing the discriminator convolution layers from 6 to 3. In addition, we increased the weights of the VGG feature loss (β) from 0.001 to 0.1. Due to time limitation, we

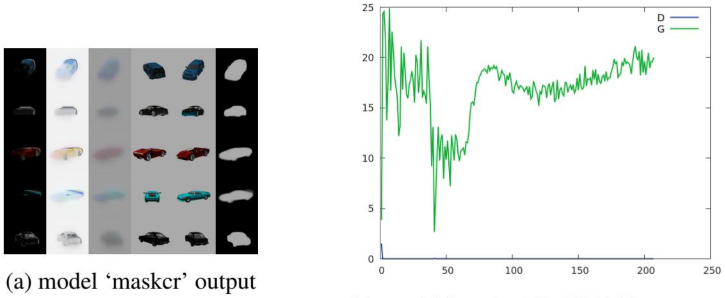


(a) model ‘maskc’ DCGAN loss curve

(b) model ‘maskco’ DCGAN loss curve

Figure 5: DCGAN loss curve
(Horizontal axis unit is $10\times$ # of epoch)

haven’s trained the new model as long as previous ones. Fig.6 shows the sampled results as well as loss curves up to the 21st epoch.



(a) model ‘maskcr’ output

(b) model ‘maskcr’ DCGAN loss curve
(Horizontal axis unit is $10\times$ # of epoch)

Figure 6: Model ‘maskcr’ result after 21 epoch of training

Table 4: TVSN model ‘maskcr’ result annotation

1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}
DOAFN output	TVSN output	generated image (1^{st} col + 2^{nd} col)	input view	target view	DOAFN generated contour map

If we take close look at Fig.6, we could find that the completion network is able to generate the unseen part in the source image. Although the loss for the discriminator still remains 0, the generator loss does exhibit a more interesting oscillation than previous models.

6 Conclusion

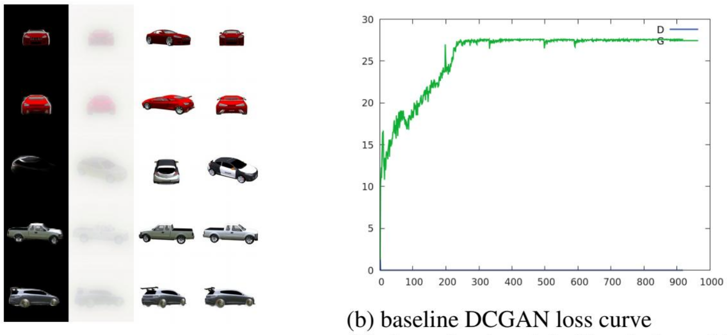
The idea of synthesizing novel views by combining appearance flow network (DOAFN) and completion network (TVSN) inherited from (7) is shown to be a promising direction, as the generated images started to resemble the targeted view after 100 epoch of training (the model was trained for 300 epoch in (7)). Our model, adding the DOAFN generated contour mask as an additional input to TVSN, shows comparable performance (similar visual effect and GAN loss curve) to the baseline model presented in the paper (shown in Fig. 7), but no significant improvement either; however, applying the generated contour mask to the final network output does show a more clearly defined object outline, shown in Fig.4 (b). The newly experimented model that learns the residual between I_{DOAFN} and I_t, with reduced discriminator complexity to combat the diminishing generator gradient issue, yields a more interesting generator loss curve, as shown in Fig.6 (although we still struggles to get the discriminator loss up). For future work, we will further carefully tune the weight for each term in the generator loss, and perhaps further reduce the discriminator complexity. With a satisfying first 100 epoch performance, we'll train the network for a full 300 epochs, and reset the discriminator parameters every 100 epochs.

7 Contribution

Miao Zhang: hyper parameter tuning, network architecture modification, loss function modification, web server management

Xiaobai Ma: data loading and rendering, network input and output modification, executing tests, debugging

Appendix



(a) baseline model output

(b) baseline DCGAN loss curve
(Horizontal axis unit is 10× # of epoch)

Figure 7: Baseline result after 100 epoch of training

Table 5: TVSN baseline model result annotation

1^{st}	2^{nd}	3^{rd}	4^{th}
DOAFN output	generated image (TVSN output)	input view	target view

References

[1] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 577–584.

[2] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh, "3d object manipulation in a single photograph using stock 3d models," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 127, 2014.

[3] K. Rematas, C. H. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars, "Novel views of objects from a single image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1576–1590, 2017.

[4] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *European Conference on Computer Vision*. Springer, 2016, pp. 322–337.

[5] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis," in *Advances in Neural Information Processing Systems*, 2015, pp. 1099–1107.

[6] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *European conference on computer vision*. Springer, 2016, pp. 286–301.

[7] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3d view synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 702–711.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[9] "Objrenderer," https://github.com/sunweilun/ObjRenderer.

[10] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.