

Novel View Synthesis for Large-scale Scene using Adversarial Loss

Xiaochuan Yin, Henglai Wei, Penghong Lin, Xiangwei Wang, Qijun Chen

No Institute Given

Abstract. Novel view synthesis aims to synthesize new images from different viewpoints of given images. Most of previous works focus on generating novel views of certain objects with a fixed background. However, for some applications, such as virtual reality or robotic manipulations, large changes in background may occur due to the egomotion of the camera. Generated images of a large-scale environment from novel views may be distorted if the structure of the environment is not considered. In this work, we propose a novel fully convolutional network, that can take advantage of the structural information explicitly by incorporating the inverse depth features. The inverse depth features are obtained from CNNs trained with sparse labeled depth values. This framework can easily fuse multiple images from different viewpoints. To fill the missing textures in the generated image, adversarial loss is applied, which can also improve the overall image quality. Our method is evaluated on the KITTI dataset. The results show that our method can generate novel views of large-scale scene without distortion. The effectiveness of our approach is demonstrated through qualitative and quantitative evaluation.

...

Keywords: View synthesis.

For geometry-based methods, advance knowledge of the 3D geometry of the original image is required. It is difficult to collect the dense point clouds necessary for 3D construction. Moreover, structural estimation from a single image outdoors is also a difficult research direction. The regions that are unseen in the input views due to self-occlusions must be deduced, otherwise, there will be holes in the generated image. Consequently, post-processing using an additional hole-filling algorithm is necessary [1]. Therefore, this kind of algorithm may not always be feasible.

In learning-based approaches, the attributes of a certain objects are learned, such as symmetry in shapes. Thus, an image of this object from a novel view can be obtained based on prior knowledge. Methods of this kind are restricted by the availability of prior knowledge related to the relevant object category, such as cars or chairs. For large-scale environments, such as outdoor environments, the generated images from novel views may be distorted because these methods do not explicitly consider structural information. The reason for including structural information is that position and orientation of each object will incur different pixel allocations during egomotion of the camera.

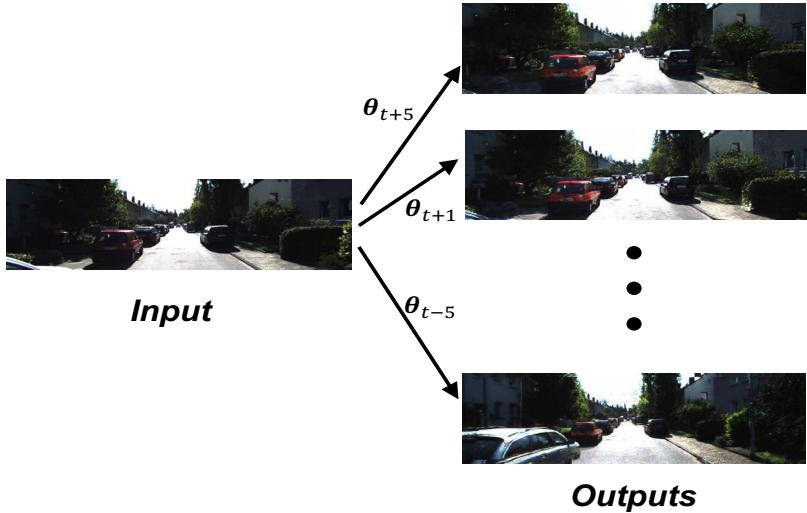


Fig. 1. Illustration of our purpose. Our method aims to synthesize high-quality target images with the corresponding control variables θ_{t+i} . The output images are the synthesized results from our method.

In this work, we combine the strengths of geometry-based and learning-based approaches. We introduce a new parametric image generation method for novel image synthesis based on corresponding camera transformation. The structural information of environments is explicitly considered in our framework. We incorporate the geometric information into the feature maps of our networks instead of constructing the 3D structure of the environments as an intermediate stage. The mean squared error is normally adopted as the loss function for view synthesis problem. Blurry image will be obtained because of the random filling of missing (unseen) regions. To fill in the missing parts and textures and improve the quality of the generated images, generative adversarial networks are applied as a loss function learning mechanism. The transformation constraint is treated as an auxiliary term in our adversarial losses.

The main contributions of our paper are summarized as follows.

1. We propose a novel view synthesis method for large-scale scenes by incorporating inverse depth features. The inverse depth features are obtained by networks trained with sparse labeled depth values. Structural information is explicitly considered in our framework.
2. We extend the proposed framework to fuse arbitrary number of input images with corresponding viewpoints.
3. The proposed framework can be trained in an end-to-end way. Images of arbitrary size can be used as inputs because our networks are fully convolutional.

4. We apply GAN to improve the quality of generated images, and transformation constraint is used as auxiliary term in our discriminator.

The structure of this paper is organized as follows. In section 2, we introduce the related works about view synthesis problem and generative adversarial networks. In section 3, we introduce our framework for view synthesis with single and multiple inputs. Experiments of novel view synthesis using single and multiple images are reported and analyzed in section 4. We conclude our paper in the last section.

1 Related Works

The ability to predict the future movements of a robots' surroundings is one of the crucial requirements for decision-making and planning in robotics. Most related works anticipate the movement of certain objects in the scene under the assumption that the background is fixed [2]. However, in the real world, robot's view constantly changing as it moves, which introduces additional challenges when solving this problem.

Deep neural networks have achieved great success in recent years because of their representation power. Research on novel view synthesis using convolutional neural networks can be categorized in two main directions.

Learning depth values for view synthesis For geometry-based view estimation method, reprojection is the most commonly applied for image synthesis. The quality of the generated image depends on the precision of estimated depth values. To generate entire image, a dense depth map is required. Currently, inferring depth values from images captured outside is still a challenging problem [3,4]. Furthermore, this kind of method cannot infer information about the regions that are missing in the input images. Consequently, holes will appear in the output image [5]. The geometry inferred from a single image is uncertain, and as a result, unrealistic and jarring rendering artifacts will occur in the generated view.

Disentangling pose features Auto-encoders are mainly applied to incorporate control variables [6]. Variables that control the view changes associated with hidden nodes of objects are decoded for image generation. In previous researches, objects such as cars and chairs have been applied for view synthesis. Gated Boltzmann machines have also been proposed to predict the transformation controlled by the multiplication of certain latent variables [7]. Synthetic images of chairs can be generated from given poses [8], where the pose parameters are treated as the attribute of the model. However, framework of this kind cannot be applied for large-scale image generation, because it does not consider the structural information. Depth regression as structural constraints does not help.

Image rendering The goal of image-based rendering is to synthesize a new view of scene via warping from nearby input images. Learning and generation of selection mask in image space is an effective strategy. Interpolation between views is achieved by learning the weight or mask of pixels in different views

[9,?]. The appearance flow method is based on calculating the allocation of the pixels in the output view [10,?]. This kind of methods cannot infer the image information from regions that are missing in the original images. In addition, the structure of the generated images cannot be preserved, meaning that distortion will inevitably occur.

Generative adversarial networks Generative adversarial networks (GAN) are generative models via an adversarial process, which have achieved impressive performances in various tasks [11,?]. GAN contains two models: generator G and discriminator D. It corresponds to a minimax two-player game. The generator tries to capture the data distribution and generate realistic data. On the other hand, the discriminator tries to distinguish whether data is generated or real. It is believed the GAN framework can be used to learn the loss function via the training process [12]. Many methods for improving the quality of generated images have been proposed, such as Least Squares GAN, WGAN [13] and etc. The conditional GAN approach has been proposed to obtain images based on conditional attributes or images [14]. Based on this structure, several applications and modifications are introduced, such as image translation framework [12]. Auxiliary classifier GAN (AC-GAN) [15] uses the conditional variables by designing additional classifier in discriminator.

2 Method

In this section, we present our framework for novel view synthesis. Our goal is to synthesize high-quality images from novel views based on single or multiple input images and specified transformation variables. For the generation of images of large-scale scenes, we propose a novel framework for calculating the location of features by incorporating inverse depth features as geometric information. We adopt this approach because according to our daily observations, the displacements of pixels are related to the inverse depths of corresponding objects. In comparison with far objects, the displacements of pixels representing near objects are larger related to the egomotion of camera. This is well-studied and described as instantaneous motion model in image space.

2.1 Camera motion and feature allocation

The instantaneous motion model describes the optical flow field as a function of the egomotion of camera and inverse depth value [16,?]. The vector describing the camera’s egomotion, $\theta_t = (\Omega; \mathbf{t})$, consists of rotation and translation vectors. The rotation vector is denoted by $\Omega = (\Omega_x, \Omega_y, \Omega_z)^T$, and the translation vector is denoted by $\mathbf{t} = (t_x, t_y, t_z)^T$. The displacement of a pixel u can be expressed as a function of its inverse depth value d and motion parameters.

$$\mathbf{u} = \mathbf{Q}_\Omega \Omega + d \mathbf{Q}_t \mathbf{t}$$

where

$$\mathbf{Q}_\Omega = \begin{pmatrix} \frac{\tilde{x}\tilde{y}}{f} & -f - \frac{\tilde{x}^2}{f} & \tilde{y} \\ f + \frac{\tilde{y}^2}{f} & -\frac{\tilde{x}\tilde{y}}{y} & -\tilde{x} \end{pmatrix}$$

$$\mathbf{Q}_t = \begin{pmatrix} -f & 0 & \tilde{x} \\ 0 & -f & \tilde{y} \end{pmatrix}$$

Here, $\tilde{x} = x - x_0$ and $\tilde{y} = y - y_0$ are the centered image coordinates, f denotes the focal length, and (x_0, y_0) represents the coordinates of the principal point. The pixel skew and aspect ratio in the camera calibration matrix is set to 0 and 1, respectively.

The displacement of a pixel consists of two components. The first component depends on the rotation of the camera and the second component depends on the inverse depth values and the translation of the viewpoint. This description is consistent with our observation that for an object at a large distance, the translation of the camera will result in a small displacement, and vice versa.

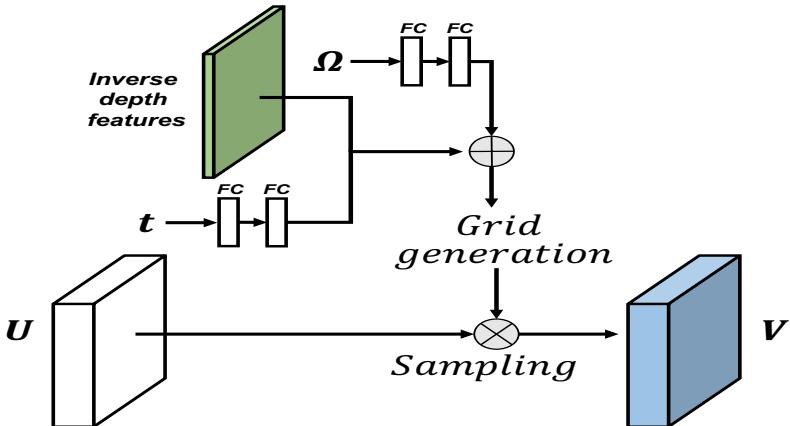


Fig. 2. The basic structure of sampling grid generation framework. The sampling grid is generated using inverse depth features and control variables (Euler angles and translations along the x, y and z axes). \mathbf{U} and \mathbf{V} denote the input and output feature maps, respectively.

We sample the output features $\mathbf{V} \in \mathbb{R}^{H \times W \times C}$ from the input feature map $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$ using parameterized sampling grid $\mathbf{G} \in \mathbb{R}^{HW \times 2}$, $G_i = (x_i^s, y_i^s)^T$, where H and W are the height and width of the grid, respectively, and C is the number of channels. Each node in the output feature map is obtained through this process.

The pointwise transformation is defined as follows:

$$\begin{pmatrix} x_i^t \\ y_i^t \\ x_i^s \\ y_i^s \end{pmatrix} = g(\Omega) \begin{pmatrix} x_i^t \\ y_i^t \\ x_i^{t^2} \\ x_i y_i \\ y_i^{t^2} \\ 1 \end{pmatrix} + d(x_i^s, y_i^s) h(\mathbf{t}) \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \quad (1)$$

where (x_i^t, y_i^t) and (x_i^s, y_i^s) denote the target and source coordinates in the output and input feature maps, respectively, and $d(x_i^s, y_i^s) \in [0, 1]$ denotes the inverse depth feature corresponding to (x_i^s, y_i^s) in the feature space. $d(x_i^s, y_i^s)$ is initialized with $d(x_i^t, y_i^t)$, and updated to $d(x_i^s, y_i^s)$ after one iteration of Equation 1. The coordinates x_i^s, y_i^s, x_i^t , and y_i^t are normalized to lie in the range $[-1, 1]$. $g(\Omega) \in \mathbb{R}^{2 \times 6}$ is a parametric transformation in terms of the rotation variables Ω . $h(\mathbf{t}) \in \mathbb{R}^{2 \times 3}$ is a parametric transformation in terms of the translation variables \mathbf{t} . The transformation matrices are generated via fully connected networks. The basic structure of the sampling grid generation framework is shown in Figure 2.

We apply the bilinear sampling kernel for feature sampling from the input feature map [17], which is defined as:

$$V_i^c = \sum_h^H \sum_w^W U_{hw}^c \max(0, 1 - |x_i^s - h|) \max(0, 1 - |y_i^s - w|)$$

The gradients flow back to the input feature map and the sampling grid by virtue of the differentiable sampling mechanism.

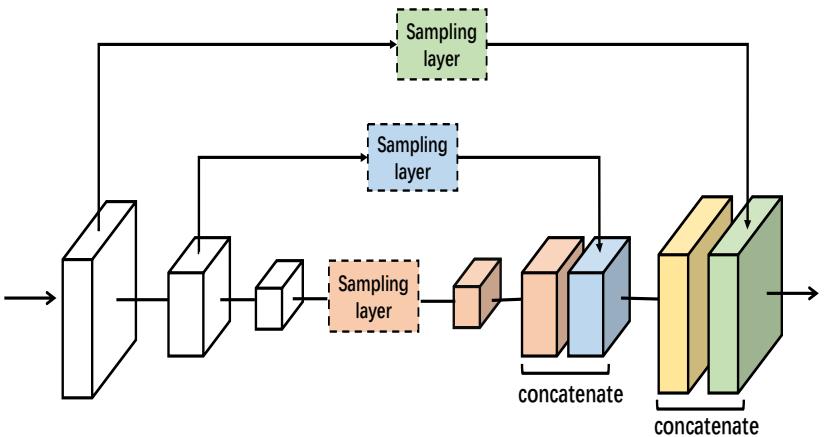


Fig. 3. Structure of our image generation network. We concatenate the features at different representation levels.

Our network structure is shown in Figure 3. We stack transformations at different representation levels. The networks can be trained with L_1 loss, supervised by the corresponding output image. The generated images with L_1 loss function may incur blur for uncertain parts. Because view synthesis trained with the L_1 loss cannot infer the unseen parts and textures, the image generated in this way may suffer from blur in certain regions. Hence, generative adversarial networks are introduced to fill in the textures and unseen parts in the generated images.

2.2 Generative adversarial networks

The L_1 loss function captures the main structure of the target images. Unseen regions and textures will cause blurring because all the possibilities for the unknown pixels will be mixed. To synthesize realistic images, generative adversarial networks are introduced to learn the loss function to fill in missing textures and occluded objects. Furthermore, in addition to improving the quality of images, auxiliary loss is introduced to ensure that the generated images satisfy the specified transformation constraint. Based on these requirements, the Auxiliary Classifier GANs (AC-GAN) framework is applied to construct our discriminator. Least squares loss function (LSGAN) [18] is used as part of our AC-GAN to learn the image distribution, which is used to judge whether an input image is real or not. Compared with the regular GAN, LSGAN can generate more realistic images.

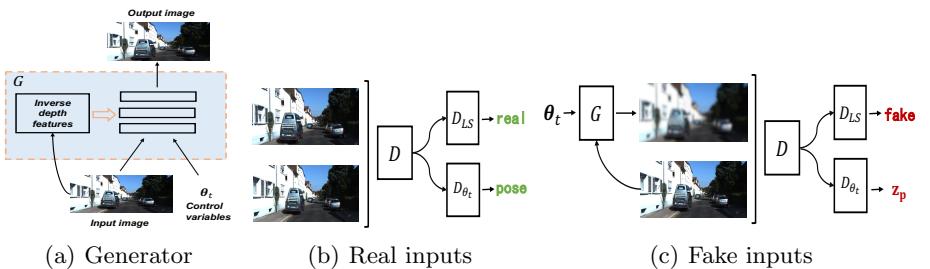


Fig. 4. Structure of our generator networks. Image is generated from a single input image and the control variable vector θ_t . The inverse depth features of input image are provided in the branch networks. Adversarial loss with a real input image classifier and the real pose θ_t regression. Adversarial loss with a generated image classifier and the fake random pose variables \mathbf{z}_p regression.

The generator G and discriminator D are obtained by minimizing the objective function of generative adversarial networks. The objective function of discriminator consists of two terms: least squares term and auxiliary term. The least squares term models the distribution of image, and it determines whether

the generated images are real or fake. The objective function of the least squares term is defined as follows:

$$\begin{aligned}\mathcal{L}_{LS} = & \frac{1}{2} \mathbb{E}_{Y \sim p_{data}(Y)} [(D_{LS}(Y|X) - 1)^2] + \\ & \frac{1}{2} \mathbb{E}_{z \sim p_{z(z)}} [(D_{LS}(G(X, \theta_t, z)|X) + 1)^2]\end{aligned}\quad (2)$$

where $z \sim p_{z(z)}$ denotes random values, which bring randomness to fill the unseen regions and textures. In our work, dropout layers are applied to introduce randomness, it is employed after the sampling layer in training and test phases.

The auxiliary term works as regularization term. It is applied to ensure the two input images satisfy the specific requirements of transformation. The objective function of auxiliary term is expressed below:

$$\begin{aligned}\mathcal{L}_{\theta_t} = & \frac{1}{2} \mathbb{E}_{Y \sim p_{data}(Y)} [(D_{\theta_t}(Y|X) - \theta_t)^2] + \\ & \frac{1}{2} \mathbb{E}_{z_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})} [(D_{\theta_t}(G(X, \theta_t, z)|X) - z_t)^2]\end{aligned}\quad (3)$$

The final objective function of discriminator is:

$$\min_{D_{LS}, D_{\theta_t}} \mathcal{L}_{AC-GAN} = \mathcal{L}_{LS} + \lambda \mathcal{L}_{\theta_t} \quad (4)$$

where λ is the weight of auxiliary term, and λ is set to 0.1.

The final objective function of generator is defined as follows:

$$\begin{aligned}\min_G \mathcal{L}_{AC-GAN} = & \frac{1}{2} \mathbb{E}_z [(D_{LS}(G(X, \theta_t, z)|X))^2] + \\ & \lambda \mathbb{E} [(D_{\theta_t}(G(X, \theta_t, z)|X) - \theta_t)^2] + \\ & \varphi \mathbb{E} [|G(X, \theta_t, z) - Y|]\end{aligned}\quad (5)$$

where φ is the weight of L_1 loss term.

The structure of generator networks is shown in Figure ???. A single image and a vector of control variables are taken as the inputs. The discriminator's structure of our framework is illustrated in Figure ?? and Figure ???. Both the generated and original input images are taken as the inputs for discrimination. For our task, in addition to generate high-quality images, the specified transformation should also be satisfied. Hence, there are two branches for our discriminator's network. The purpose of one branch is to ensure the generation of realistic image, the purpose of the other branch is to ensure the generated image meets the transformation requirements. For a synthesized input image, random values \mathbf{z}_p following a normal distribution are used as the supervised signals.

2.3 Fusion of input views

Image from a single view may not contain adequate information for view inference. Information obtained from different views can provide supplementary

details for novel view synthesis. Therefore, we extend our method as described above to the fusion of multiple input images from different viewpoints.

Max selection is introduced in our framework to fuse features from multiple feature maps. The objective function for max selection is expressed as follows:

$$V_{out}(p) = \max(V_1(p), V_2(p), \dots, V_N(p)), \forall p \in S \quad (6)$$

where V is the feature map after performing grid sampling, $V : S \subset \mathbb{R}^3 \mapsto \mathbb{R}$, $p = (x, y, c)$ denotes spatial location (x, y) in channel c , and N is the number of feature maps to be fused.

The network structure for image fusion is shown in Figure 5. Our method does not restrict the number of inputs. We apply the max selection after the sampling layer. The maximum element at each position is selected as the output.

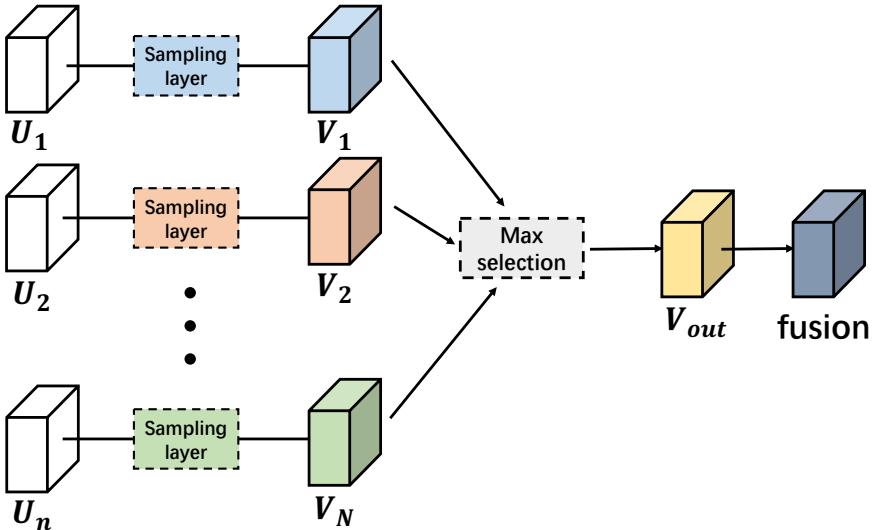


Fig. 5. Feature fusion network for the fusion of multiple images from different corresponding viewpoints.

2.4 Network architecture

We employ DenseNet-201 as the backbone model without pooling layers [19]. This architecture connects each layer to every other layer in a feed-forward fashion. By virtue of this feature, DenseNets alleviate the vanishing-gradient problem, strengthen feature propagation and reuse features of lower layers, and substantially reduce the number of parameters.

We sample and transform the features in transition layers of DenseNet. In order to include more details of input images to the synthesized images, features

of different levels in transition layers are sampled and concatenated. Otherwise, words on traffic sign and some small objects in the input images will not appear on the synthesized images. Both inverse depth estimation networks and image generation networks are based on DenseNet-201. Parameters of inverse depth estimation networks are fixed during the training and testing phases of image synthesis networks, it is used to provide the inverse depth feature maps at different levels.

3 Experiments

In this section, we conduct experiments to verify the effectiveness of our method on the KITTI dataset [20], which is collected in urban environments. The KITTI dataset consists of image sequences from a driving vehicle with depth captured by a LiDAR and ground truth position by a GPS localization system. The depth values obtained from LiDAR are sparse, and only 5% of pixels contain valid values. Our method can predict the picture of novel view up to move ± 7 meters along the z axis and rotate around the y axis by ± 22 degrees.

Our neural networks are trained and deployed on a NVIDIA TITAN X GPU with 12 GB memory. We use PyTorch to train our model. Our network is modified based on DenseNet-201 downloaded from model zoo. We apply the images in sequence 00 and 02 as training set (91960 image pairs). Images in sequence 08, 09, 10 are taken as test set (68540 image pairs). Images of different sequences are collected in different urban environments. Therefore, environments in the test set will not appear in the training set. All the image are down-sampled by a factor of 2 and resized to 176x608. The Euler angles and translations along the x,y and z axes are applied as inputs to our networks.

The inverse depth values are preprocessed by normalizing the values to $\frac{2}{x-1.5} - 1$, where x is the depth value collected in LiDAR. All layers are trained using SGD and ADAM optimizer [21] ($\beta_1 = 0.5, \beta_2 = 0.9$), and learning rate is set to 0.0001. Our networks are trained in an end-to-end way after 200,000 iterations.

In order to evaluate the effectiveness of our method, we designed the experiments for novel view synthesis for large-scale scene. We testify our framework for generating novel view image from a single input image, multiple input images. Our method is evaluated quantitatively using mean pixel L_1 error and inception score¹. These two metrics are widely applied to evaluate the synthesized images.

3.1 View synthesis from a single input

Our method can predict the image from a single input image with the given viewpoint. We generate the results from ten adjacent time steps (five time steps forward and five time steps backward). + and – denote forward and backward

¹ The code and model for inception score evaluation are available at <https://github.com/openai/improved-gan>

movement respectively. Generated images of \pm one time step, \pm three time steps and \pm five time steps are shown in Figure 6.

We evaluate our method with two parameterized methods: Multi-view 3D Models (MV3D) [6]² and appearance flow method [10]³. We also train our networks with L_1 Loss function. In order to maintain the structural information explicitly, MV3D preserves the structure of environments by adding the depth regression as additional output channel. However, this method is not effective for large-scale scenes, even for image synthesis with slight egomotion. The appearance flow method can keep the structural information and generate high-quality images for slight movements. However, images will become distorted for large movement. The generated images using MV3D and appearance flow methods on the KITTI dataset are consistent with the results demonstrated in [10]. Figure 6(d) shows the generated images with L_1 loss function, which demonstrate that our framework is an effective method to incorporate the structural information. For missing parts in the original images, it will become blurry in the corresponding parts in the output regions. Figure 6(e) demonstrates that GAN loss can improve the quality of the generated images by filling the blurry parts with textures and details. Our method can also conjecture objects to fill the unseen parts.

Table 1 shows the mean pixel L_1 errors of different time intervals of three methods. L_1 error measures the low frequency signals of images. Therefore, lower L_1 error means the method can capture more structural information of target images. Our framework trained with L_1 loss has achieved the lowest L_1 error. The final objective function of our method reaches a compromise between maintaining structural information and filling uncertain details by involving GAN loss and L_1 loss. By doing this, L_1 error of our generated images will increase. For all the methods, L_1 errors will increase as the time interval increase. The inception score of our method does not change.

Method	\pm One time step	\pm Two time steps	\pm Three time steps	\pm Four time steps	\pm Five time steps	Mean
MV3D [6]	0.1933	0.2057	0.2224	0.2411	0.2593	0.2244
Appearance Flow [10]	0.2232	0.2850	0.3234	0.3508	0.3720	0.3109
Our networks (L_1)	0.1458	0.17548	0.1991	0.2203	0.2394	0.1959
Our networks (L_1 +GAN)	0.2057	0.2360	0.2630	0.2885	0.3056	0.2598

Table 1. Mean pixel L_1 error on the KITTI dataset for novel view synthesis.

Inception scores of different time intervals are shown in Table 2. Distortion of the generated images does not affect the inception score. Therefore, both mean

² The code and model are available at <https://github.com/lmb-freiburg/mv3d>

³ The code and model are available at <https://github.com/tinghuiz/appearance-flow>

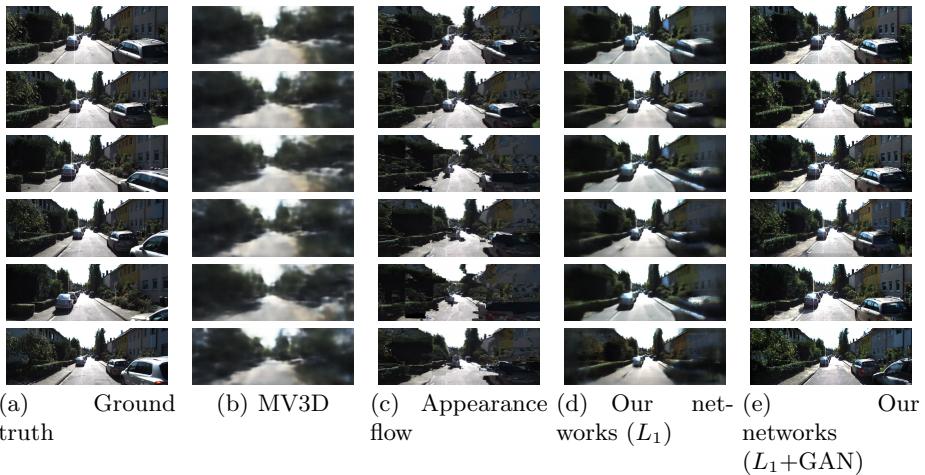


Fig. 6. Examples of novel view synthesis results with single input image from the KITTI dataset. For each frame, we show (a) ground truth images, (b) results generated with MV3D method, (c) results generated with appearance flow method (d) results produced by our networks trained with L_1 loss, and (e) synthesized images by our networks with L_1 and GAN loss. Different rows shows the results of different time intervals. Images from the first row to the bottom row are one time step forward, one time step backward, three time steps forward, three time steps backward, five time steps forward and five time steps backward.

pixel L_1 error and inception score need to be taken as evaluation metrics for view synthesis methods.

Method	\pm One time step	\pm Two time steps	\pm Three time steps	\pm Four time steps	\pm Five time steps	Mean
MV3D [6]	1.705 \pm 0.018	1.700 \pm 0.019	1.702 \pm 0.014	1.720 \pm 0.017	1.716 \pm 0.024	1.709
Appearance Flow [10]	3.196\pm0.038	3.084 \pm 0.023	2.951 \pm 0.034	2.855 \pm 0.048	2.768 \pm 0.035	2.971
Our networks (L_1)	2.819 \pm 0.034	2.836 \pm 0.021	2.874 \pm 0.026	2.872 \pm 0.040	2.860 \pm 0.036	2.852
Our networks (L_1+GAN)	3.196\pm0.054	3.203\pm0.053	3.196\pm0.038	3.173\pm0.036	3.196\pm0.040	3.193

Table 2. Mean and standard deviation (std) of inception score on the KITTI dataset for novel view synthesis. Higher is better. For the real images on the KITTI dataset, the mean inception score is 2.9656.

3.2 View synthesis from multiple inputs

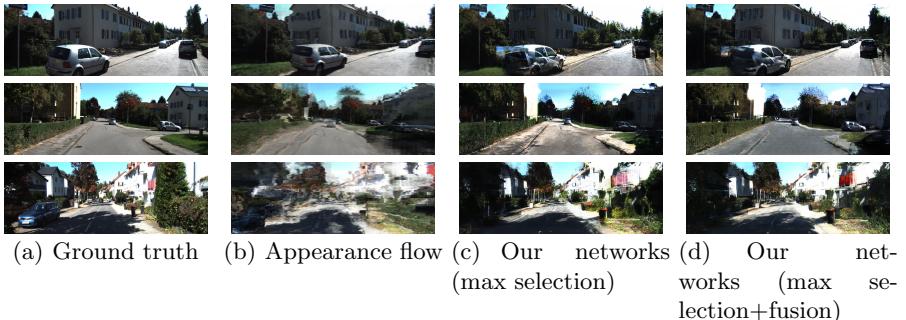


Fig. 7. Examples of novel view synthesis results with two input images from the KITTI dataset. For each frame, we show (a) ground truth images, (b) multiple image synthesis results generated with appearance flow (c) results produced by max selection in the feature maps of networks, whose parameters are obtained from view synthesis from a single input, and (d) results generated by adding fusion layer after the max selection operator.

A single image may not provide enough information to synthesize the image from a novel viewpoint. Bunch of input images from different viewpoints will improve the quality and deterministic of generated image apparently.

We extend our method by adding max selection layer after the sampling layers. The maximum value in the feature map of corresponding position is selected for further layer. From our observation, even without fine-tuning process,

we can obtain the fused images. However, the fused images from multiple viewpoints will cause double edges of certain objects. This effect may be caused by the bilinear sampling layers in the down-sampled feature maps. Therefore, another convolutional layer is applied as fusion layer to eliminate this effect.

Both our method and the appearance flow method can take multiple images as inputs. Results of view synthesis from two input images are shown in Figure 7. Two random images from ten adjacent images are selected as inputs to our networks. Firstly, we directly select maximum values in two feature maps as inputs to the following layers. Parameters of this networks are obtained for view synthesis from a single input without fine-tuning process. The synthesized images are shown in Figure 7(c). Experiments show that the generated images can keep the main structure of target images. The fusion layer (Conv+BatchNorm+Relu) is introduced as post-processing stage after each max selection operator. After fine-tuning the whole networks, the fusion layer can improve the image quality by enhancing the details and removing the noise pixels. Comparing with our method, images generated with appearance flow method will become distorted for large movements. The appearance flow method is not sensitive to small ego-motion. For example, comparing with the input image, there are small changes in the output image (first row in Figure 7(b)). Car disappeared in our results shown in bottom row of Figure 7(c) and 7(d), because it is not shown in one of the input images.

The quantitative comparison between appearance flow method and ours' for multiple inputs are listed in Table 3. From the L_1 error and inception score shown in the table, images generated by our method (without and with fusion layer after max selection) can preserve the structure of environments.

Method	L_1	Inception Score
Appearance Flow [10]	0.2698	3.0959 ± 0.0788
Our networks (max selection)	0.2481	3.1409 ± 0.0621
Our networks (max selection+fusion)	0.2318	3.0945 ± 0.1060

Table 3. Mean pixel L_1 error and inception score on the KITTI dataset for novel view synthesis with two input images. The two inputs are randomly selected from ten adjacent frames.

4 Conclusions

In this paper, we propose a novel end-to-end framework to generate a image from novel viewpoint. Images are synthesized by convolutional neural networks with generative adversarial loss. Structural information is explicitly incorporated into the feature maps in our networks. Therefore, we can synthesize high-quality

images by providing the input image and corresponding viewpoint without distortion for large-scale scenes. Our framework can be easily extended to the fusion of multiple input images.

In future work, we would like to incorporate our model with other object moving prediction methods. Stacked generative adversarial networks may be applied to generate high resolution images.

References

1. Zhu, C., Li, S.: Depth image based view synthesis: New insights and perspectives on hole generation and filling. *IEEE Transactions on Broadcasting* **62**(1) (2016) 82–93
2. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: *Advances in Neural Information Processing Systems*. (2016) 64–72
3. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: *Proceedings of Advances in Neural Information Processing Systems*. (2014) 2366–2374
4. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(10) (Oct 2016) 2024–2039
5. Mahjourian, R., Wicke, M., Angelova, A.: Geometry-based next frame prediction from monocular video. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. (2017) 1700–1707
6. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3d models from single images with a convolutional network. In: *European Conference on Computer Vision (ECCV)*. (2016)
7. Memisevic, R., Hinton, G.: Unsupervised learning of image transformations. In: *Computer Vision and Pattern Recognition*. (2007) 1–8
8. Dosovitskiy, A., Springenberg, J.T., Brox, T.: Learning to generate chairs with convolutional neural networks. In: *Computer Vision and Pattern Recognition*. (2015) 1538–1546
9. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deep stereo: Learning to predict new views from the world’s imagery. In: *Computer Vision and Pattern Recognition*. (2016) 5515–5524
10. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: *European Conference on Computer Vision*. (2016) 286–301
11. Goodfellow, I.J., Pougetabadi, J., Mirza, M., Xu, B., Wardefarley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. *Advances in Neural Information Processing Systems* **3** (2014) 2672–2680
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *CVPR*. (2017)
13. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. (2017)
14. Mirza, M., Osindero, S.: Conditional generative adversarial nets. *Computer Science* (2014) 2672–2680
15. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: *Proceedings of the 34th International Conference on Machine Learning*. Volume 70. (2017) 2642–2651

16. Heeger, D.J., Jepson, A.D.: Subspace methods for recovering rigid motion i: Algorithm and implementation. *International Journal of Computer Vision* **7**(2) (1992) 95–117
17. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: *Advances in Neural Information Processing Systems*. (2015) 2017–2025
18. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: *International Conference on Computer Vision*. (2017)
19. Huang, G., Liu, Z., Weinberger, K.Q., Laurens, V.D.M.: Densely connected convolutional networks. (2017)
20. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. (2012)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: *In International Conference on Learning Representations*. (2015)