View Independent Generative Adversarial Network for Novel View Synthesis

Xiaogang Xu[1] Ying-Cong Chen[1] Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong [2]Tencent YouTu Lab
{xgxu, ycchen, leojia}@cse.cuhk.edu.hk

Abstract

Synthesizing novel views from a 2D image requires to infer 3D structure and project it back to 2D from a new viewpoint. In this paper, we propose an encoder-decoder based generative adversarial network VI-GAN to tackle this problem. Our method is to let the network, after seeing many images of objects belonging to the same category in different views, obtain essential knowledge of intrinsic properties of the objects. To this end, an encoder is designed to extract view-independent feature that characterizes intrinsic properties of the input image, which includes 3D structure, color, texture etc. We also make the decoder hallucinate the image of a novel view based on the extracted feature and an arbitrary user-specific camera pose. Extensive experiments demonstrate that our model can synthesize high-quality images in different views with continuous camera poses, and is general for various applications.

1. Introduction

We tackle the problem of novel view synthesis – given a single 2D image of objects, we aim to synthesize a new one captured from an arbitrary viewpoint. This potentially benefits a large variety of applications in computer vision and robotics. For instance, multiple synthesized 2D views form an efficient 3D representation as a collection of images [4]. In robotics, able to see objects in various viewpoints is helpful for planing [19].

Existing methods for novel view synthesis fall into two categories – geometry- and learning-based ones. Given a 2D image, geometry-based methods [16, 27] first estimate its 3D representation, and project it back to 2D space based on the target view. By directly building 3D models, these methods allow synthesizing 2D new images from arbitrary viewpoints. However, estimating the 3D structure from a single 2D image is intrinsically ill-posed. If not restricted to specific scenarios, e.g., faces [1], 3D models cannot be accurately generated.

On the other hand, with powerful convolutional neural networks (CNN), learning-based methods [28, 34, 29, 30] directly produce the final image in the target view, without explicitly estimating its 3D structure. View synthesis is thus implemented with a mapping function between the source and target views associated with their camera poses [34]. As estimation of 3D models is not needed, it is applicable to a wider range of scenarios. The limitation is that directly producing a 2D image without considering 3D structure does not generalize well. To address this issue, method of [24] incorporates extra 3D information, which is however not considered in this paper since we do not think 3D information is always accessible in practice and instead propose a more general solution for novel view synthesis.

Recently, generative adversarial networks (GAN) [8, 26] were applied to multi-view synthesis. Current GAN-based approaches usually discretize camera parameters into a fixed-length vector to improve performance [29, 30]. Nevertheless, 3D-related information contained in camera poses is inevitably damaged with such setting.

In this paper, we propose a method to benefit from both learning- and geometry-based methods while ameliorating their drawback. Our method is essentially learned-based, and yet still infers 3D knowledge implicitly. The key idea is based on the fact that any 2D image is a projection of the 3D world. If a certain feature is invariant with viewpoints, it depicts important intrinsic property of the 3D world. By specifying the camera pose, we reconstruct the 2D image according to the view independent "3D" feature.

It follows a virtual camera system – that is, all intrinsic information like shape, color, texture, and illumination, is first estimated. Then a 2D image is rendered based on the 3D information as well as the visual camera. Our system, which is called *View Independent Generative Adversarial Network (VI-GAN)*, simulates such a pipeline. We first infer knowledge that characterizes the 3D world based on a 2D image. Then with user specified camera pose, we project this 3D world back to image space to synthesize a new view. This is the first attempt to take this strategy. It is noteworthy that this setting naturally leads to an encoder-decoder architecture where the encoder embeds the 2D image to a latent 3D feature, and the decoder renders a novel image based on the learned 3D feature and target camera pose.

Inferring the 3D information from a single 2D image is inherently ill-posed since there exist an infinite number of solutions that produce the same 2D image. To constrain the problem, we additionally incorporate camera pose and location information of the 2D image where camera pose can be inferred from a single RGB image accurately [15].

Further, two discriminators are introduced to promote the realism and pose accuracy of generated results. These discriminators are trained with two objectives separately, i.e., classifying the real and generated images and predicting the pose of input images. By adapting these discriminators, our model generates realistic results of a given camera pose. Our total contribution lies in the following ways.

- We propose a novel view synthesis framework to synthesize new images in *arbitrary* views with weakly supervised 2D training data.
- Our model extracts view-independent features to implicitly describe the properties of 3D world, making our model generalize well for unseen viewpoints.
- Extensive experiments demonstrate that our model generates high quality images and can be used in a wide range of tasks.

2. Related Work

Existing methods of novel view synthesis can be divided into two categories of geometry- and learning-based ones. Geometry-based approaches explicitly learn the 3D structure from the input 2D image. This allows synthesizing images of an arbitrary novel view. Learning-based approaches directly map the image of a certain view to another, without inferring 3D information.

Geometry-based Methods Lin et al. [20] and Pontes et al. [25] explicitly estimated the point cloud and mesh based on the input image. However, the estimated structures are often not dense enough, especially when handling complicated texture [25]. Garg et al. [7] and Xie et al. [33] estimated the depth instead, which is also related to 3D structure. Nevertheless, they are designed for binocular situations only.

Rematas et al. [27] and Kholgade et al. [16] proposed exemplar-based models that utilize large-scale collections of 3D models. Given an input image, these approaches first identify the most similar 3D model in the database and fit it to the input. The 3D information is then utilized to synthesize novel views. It is clear that the accuracy of these methods depends on variation and complexity of 3D models. 3D Morphable Model (3DMM) [1] and its variant [6] allow generating a high quality 3D model by fitting the 3D structure and texture map from precomputed results learned from accurate 3D models. They are only applicable to faces.

Learning-based Methods Recently, convolutional neural networks are introduced for novel view synthesis [5, 28, 34, 24, 34, 24, 34, 28, 29, 30, 2]. Early methods [5, 28] directly mapped input images to another view with an encoder-decoder architecture. Note that these solutions are hard to disentangle pose-invariant factors from a single view. To improve result quality, Zhou et al. [34] predicted appearance flow instead of synthesizing pixels from scratch. It does not deal with areas whose pixels are not contained in input [24]. Park et al. [24] concatenated another generator behind such a network for enhancement. It needs 3D annotation for training, which is however not considered in our setting.

To improve the realism of synthesized images, in [29, 30, 2], generative adversarial networks (GAN) [8] are used. GAN-based methods have a discriminator to distinguish between real and generated images. With a generator to test the discriminator, missing pixels are hallucinated and the output becomes realistic. We note all these methods essentially learn mapping between images of different camera poses without inferring the 3D structure. This impedes the generalization capacity for unseen viewpoints. As a result, these methods can only synthesize decent results in several preset views. In contrast, our method can synthesize novel viewpoints even if they never appear in the training set. It is a learning based approach and implicitly infers the 3D structure in the latent space.

3. Proposed Method

In this section, we explain the motivation, as well as each component in our network. The overall framework is shown as Fig. 1, which is trained with weakly-supervised 2D image pairs. A virtual camera system aims at controlling the camera to display the view of a 3D object. Since the structure is supposedly independent of camera poses, it can be represented by features that are only related to intrinsic properties of the object. Thus the key to novel view synthesis is to separate the *intrinsic feature* of objects from the *camera pose*. By achieving this, we re-render the object by combining the intrinsic features with a new camera pose. In our model, we use an encoder for the disentangling task and a decoder for the rendering task.

3.1. Network Architecture

Encoder Given a 2D image I_A, the encoder E is responsible for extracting view-independent features F_A. Ideally, such features should include all intrinsic properties of the objects presented in I_A, and be also irrelevant to the camera pose, of which I_A is taken. This seems impossible at the first glance, since some parts are invisible in I_A. Analogy to human ability to accomplish this task by searching similar scenes in memory, we train the encoder with data from different viewpoints.

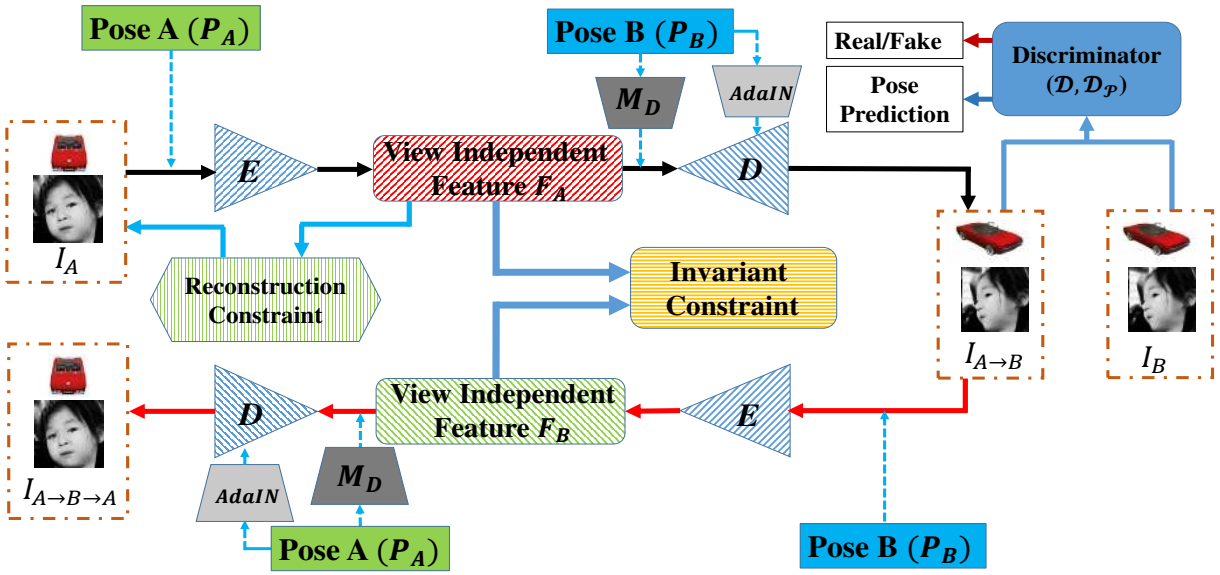


Figure 1. Overall structure of VI-GAN. The encoder extracts the view-independent feature, which is the implicit representation of the 3D world. The decoder utilizes the extracted features and new camera pose parameters to synthesize novel views. Two discriminators are set to predict the realism of input and pose information respectively. Our system is trained *without* 3D supervised data.

To reduce the difficulty of training, the camera pose is also incorporated as extra information into the encoder. This operation is practical since directly computing precise camera pose from a single RGB view is realizable [15]. Specifically, the camera pose can be parameterized by a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a shifting vector $T \in \mathbb{R}^{3\times1}$. We reshape R to 9×1 and concatenate it with T, leading to a 12×1 camera pose feature P_A. So F_A is produced by

$$F_A = E(I_A \oplus P_A), \tag{1}$$

where P_A is employed as a global feature to be concatenated with I_A. $\oplus$ is the concatenation operation. This concatenation operation provides the camera pose for each pixel to help 3D inference since 3D coordinates can be computed by combining the camera pose and the corresponding location on the image plane.

To infer 3D related knowledge based on 2D images, the encoder E needs to implicitly register the 2D image to the latent 3D model. In this regard, the location information of the 2D image is fatal. However, CNN only perceives local regions without considering location due to the *spatial invariance* property. The work of [22] remedies this issues by concatenating the pixel location as two extra channels to the feature maps, known as the *CoordConv* operation. In our model all standard convolutional layers in the image generator are substituted with CoordConv.

Decoder With the extracted pose-independent features F_A and a target pose P_B, the decoder functions as a renderer to convert them back to the image space. More specifically, we use an embedding network M_D to accommodate the channel numbers of P_B and F_A, then concatenate F_A and $M_D(P_B)$ to form the input to the decoder. This is formulated as

$$I_{A\to B} = D(F_A \oplus M_D(P_B)), \tag{2}$$

where D denotes the decoder.

In principle, the architecture of the decoder reverses encoder. Yet we also discover that replacing instance normalization with Adaptive Instance Normalization (AdaIN) [12] in the residual blocks of the decoder boosts performance. Specifically, the mean $\mu(x)$ and variance $\sigma(x)$ of the instance normalization layer are inferred by the target pose P_B instead of the feature map itself. This makes objects with similar pose share feature statistics, easier for the decoder to render results of the target pose. In addition, both M_D and the computation of AdaIN parameters are implemented by simple multilayer perceptron networks that adapt the pose information as input.

3.2. Learning 3D Related Knowledge

In general, training of our model includes a view-independent loss term, a series of image-reconstruction loss terms, a GAN loss term and a pose prediction loss term.

View-independent Loss The view-independent loss aims to impose the pose-independent property for the latent feature. Let F_A, derived from equation (1), be the latent feature of I_A captured with camera pose P_A. We first randomly choose another pose P_B and render the target view $I_{A\to B}$ by Eq. (2). Then another latent feature corresponding to

$I_{A\to B}$ is obtained by $F_B = E(I_{A\to B} \oplus P_B)$. If F_A is identical to F_B, they can be regarded as *view-independent*. Therefore, this loss is formulated as

$$\mathcal{L}_{VI} = \mathbb{E}(\|F_A - F_B\|), \tag{3}$$

where $\mathbb{E}$ refers to computing the expectation value.

Image-reconstruction Loss The first term of the image-reconstruction loss derives from generation of target view I_B. To guarantee precision of the synthesized view, we use pixel-level and perceptive loss jointly to facilitate mapping latent features back to the image space. As shown in [14, 18, 31], jointly using both of them could result in high-quality synthesis. This is shown as

$$\begin{aligned} \mathcal{L}_{pixel} &= \mathbb{E}(\|I_{A\to B} - I_B\|), \\ \mathcal{L}_{per} &= \mathbb{E}((\mathcal{V}(I_{A\to B}) - \mathcal{V}(I_B))^2), \end{aligned} \tag{4}$$

where $\mathcal{L}_{pixel}$ is the pixel-level loss and $\mathcal{L}_{per}$ is the perceptive loss. $\mathcal{V}$ includes features extracted from the VGG16 network. Meanwhile, the decoder should also have the ability of reconstructing the input view upon feeding its camera parameters. To this end, we set an input reconstruction loss term as

$$\mathcal{L}_{rec} = \mathbb{E}(\|I_A - I_{A\to A}\|), \tag{5}$$

where $I_{A\to A} = D(F_A \oplus M_D(P_A))$ is the reconstruction of input view I_A.

To further promote the precision and realism of synthesized images, cycle restriction is also adopted, which makes generated images transform back to the original view [21, 13, 35]. This cycle loss term is explicitly computed at pixel and perceptive levels as

$$\begin{aligned} I_{A\to B\to A} &= D(E(I_{A\to B} \oplus P_B) \oplus M_D(P_A)), \\ \mathcal{L}_{cyc} &= \mathbb{E}(\|I_{A\to B\to A} - I_A\|), \\ \mathcal{L}_{cyc_{per}} &= \mathbb{E}((\mathcal{V}(I_{A\to B\to A}) - \mathcal{V}(I_A))^2). \end{aligned} \tag{6}$$

GAN Loss The inferred view-independent representation contains features of other views. The decoder is also required to hallucinate possibly missing parts, especially the area of occlusion that is not viewed in input. We use an auxiliary network as a discriminator to achieve this goal.

With objective to improve the realism of synthesized images, the discriminator aims to identify realistic characteristic of input. This loss term is implemented by Wasserstein GAN-Gradient Penalty (WGAN-GP) [10]. We train an essential $\mathcal{D}$ to maximize the Wasserstein distance between real samples and synthesized ones. If we denote distributions of real images as $\mathbb{P}_r$ and synthesized images as $\mathbb{P}_f$, the loss term for $\mathcal{D}$ and generator $\mathcal{G}$, which includes E and D, is written as

$$\begin{aligned} \mathcal{L}_{\mathcal{GAN}_\mathcal{D}} &= \mathbb{E}_{\mathbb{P}_f}[\mathcal{D}(I_{A\to B})] - \mathbb{E}_{\mathbb{P}_r}[\mathcal{D}(I_B)] + \lambda_{gp}\mathcal{L}_{gp}, \\ \mathcal{L}_{\mathcal{GAN}_\mathcal{G}} &= \mathbb{E}_{\mathbb{P}_r}[\mathcal{D}(I_B)] - \mathbb{E}_{\mathbb{P}_f}[\mathcal{D}(I_{A\to B})], \end{aligned} \tag{7}$$

where $\mathcal{D}(\mathcal{X})$ is the output of $\mathcal{D}$ with input $\mathcal{X}$, $\mathcal{L}_{gp}$ is the gradient penalty term defined in [10]. λ_{gp} is the weight set to 10 during training.

Pose Prediction Loss To boost accuracy of synthesis in term of camera poses, another discriminator, denoted as $\mathcal{D}_\mathcal{P}$, is employed. A pose prediction loss is adopted in this network to ensure the generated images to be consistent with their target poses. Specifically, instead of distinguishing between real and fake samples, this discriminator is trained to predict the camera pose of a given image. Our generator, on the other hand, pushes the discriminator to output the target pose for the synthesized sample. This loss term is formulated as

$$\begin{aligned} \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{P}}} &= \mathbb{E}_{\mathbb{P}_r}((\mathcal{D}_\mathcal{P}(I_B) - P_B)^2), \\ \mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{P}}} &= \mathbb{E}_{\mathbb{P}_f}((\mathcal{D}_\mathcal{P}(I_{A\to B}) - P_B)^2), \end{aligned} \tag{8}$$

where $\mathcal{D}_\mathcal{P}(\mathcal{X})$ is the output of $\mathcal{D}_\mathcal{P}$ with input $\mathcal{X}$. $\mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{P}}}$ and $\mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{P}}}$ are the loss terms for discriminator and generator respectively. Further, $\mathcal{D}_\mathcal{P}$ enables our system to handle the situation where input camera pose is not accessible, since users can use $\mathcal{D}_\mathcal{P}$ to estimate the corresponding parameters of a given real image.

In summary, the overall loss terms for the encoder, decoder and discriminators in VI-GAN are defined as

$$\begin{aligned} \mathcal{L}_{E,D} &= \lambda_1\mathcal{L}_{VI} + \lambda_2\mathcal{L}_{pixel} + \lambda_3\mathcal{L}_{per} + \lambda_4\mathcal{L}_{rec} + \\ &\lambda_5\mathcal{L}_{cyc} + \lambda_6\mathcal{L}_{cyc_{per}} + \lambda_7\mathcal{L}_{\mathcal{GAN}_\mathcal{G}} + \lambda_8\mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{P}}}, \end{aligned} \tag{9}$$

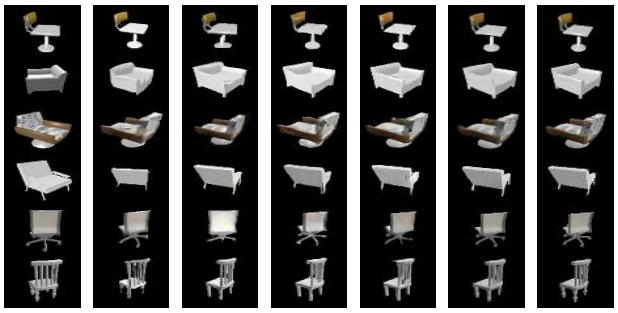
$$\mathcal{L}_\mathcal{D} = \lambda_9\mathcal{L}_{\mathcal{GAN}_\mathcal{D}} + \lambda_{10}\mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{P}}}. \tag{10}$$

In our experiments, the values from λ_1 to λ_{10} are set to make loss not far away from each other. The detailed structure of VI-GAN is given in the supplementary material.

4. Experiments

We evaluate VI-GAN on a wide range of datasets including ShapeNet [3], Multi-PIE [9] and 300W-LP [36]. ShapeNet [3] contains a large number of 3D models belonging to various categories. Images rendered by [4] from this dataset are employed, whose camera poses are continuous. We utilize this dataset to analyze the function of each component in our method and evaluate the applicability of VI-GAN on general objects. Especially, for each category, we use 80% models for training and 20% for testing.

Multi-PIE [9] is a dataset, which contains images of persons under 13 camera-poses with 15° intervals at head height. We use 250 subjects from the first session of this



(a) (b) (c) (d) (e) (f) (g)

Figure 2. Results of ablation experiments. (a) is the input and (g) is the ground truth. (b)-(f) are synthesized by VI-GAN (w/o VI), VI-GAN (w/o Pose), VI-GAN (w/o Coord), VI-GAN (w/o AdaIN) and VI-GAN respectively. The images are with size 128×128. Please zoom in to see details.

Setting	L_1	SSIM
VI-GAN(w/o VI)	16.43	0.82
VI-GAN(w/o Pose)	16.81	0.80
VI-GAN(w/o Coord)	14.35	0.85
VI-GAN(w/o AdaIN)	14.02	0.84
VI-GAN	**12.56**	**0.87**

Table 1. Mean pixel-wise L_1 error (lower is better) and SSIM (higher is better) between ground truth and predictions generated by different settings in ablation experiments. When computing the L_1 error, pixel values are in range of $[0, 255]$.

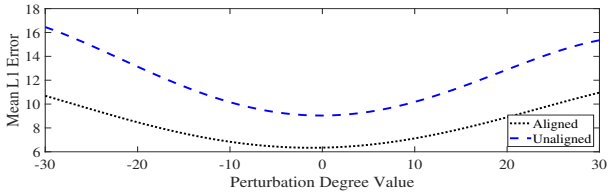
dataset where the first 200 subjects are for training and the rest 50 are for testing. This dataset is employed to analyze the sensitivity of camera poses and is utilized in comparison with existing GAN-based methods, since the camera poses of these images are discrete.

300W-LP [36] has various face images with continuous camera poses and 3DMM parameters. We use 80% identities for training and 20% for testing.

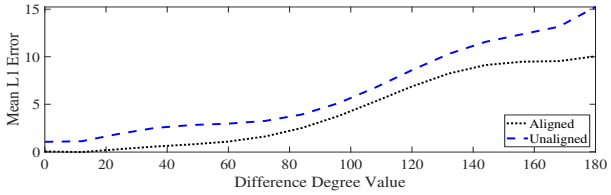
4.1. Effectiveness of Each Part

The view-independent loss, pose prediction loss, CoordConv operation, and AdaIN contribute to the quality of final synthesis. In this section, we disable each of them separately to show their respective necessity. Moreover, the experiments are conducted on the "chair" category from the ShapeNet dataset [3]. During testing, the mean pixel-wise L_1 error and the structural similarity index measure (SSIM) [32, 23] are computed between synthesized results and the ground truth.

Contribution of View-independent Loss Fig. 2(b) shows samples output by the model without the loss defined in Eq. (3), which is called "VI-GAN (w/o VI)". It is distinct since the results are either vague or lacking pose accuracy. Its L_1 error increases largely while SSIM score reduces a lot. This is because the model cannot infer accurate 3D information without this basic constraint.



(a) Pose Sensibility Analysis of Encoder



(b) Pose Sensibility Analysis of Decoder

Figure 3. Pose sensibility analysis for the encoder and decoder.

Contribution of Pose Prediction Loss Fig. 2(c) shows several results without the pose prediction loss define in Eq. (8) and this model is named "VI-GAN (w/o Pose)". As shown in this figure, without this loss, the model does not ensure an accurate pose. Also, the generated image tends to be more blurry. As shown in Table 1, without this term, the L_1 error increases a lot, while the SSIM drops from 0.87 to 0.80. Such degeneration is caused by the fact that this loss enables the generator to be aware of relation between the camera poses and view-independent features.

Contribution of CoordConv We demonstrate the function of CoordConv by setting the generator of VI-GAN as traditional convolution that is named "VI-GAN (w/o Coord)". We observe from Fig. 2(d) and (f) that the quality and pose accuracy of generated images are heavily damaged without the CoordConv. Results in Table 1 also confirm this conclusion. It is because coordinate information is crucial for 3D feature learning. Therefore, the CoordConv in generator is conducive for novel view synthesis.

Contribution of AdaIN We explain the role of AdaIN by setting another model, which is called "VI-GAN (w/o AdaIN)", with instance normalization in the residual blocks of the decoder instead. The comparison with VI-GAN is given in Fig. 2(e)-(f) where the artifacts are observed. The quantitative errors in Table 1 indicate that AdaIN can refine the output.

4.2. Sensitive Analysis of Camera Pose

We also provide analysis on conditioned pose information. Our experiments are conducted on Multi-PIE [9], since the camera movement in this dataset only has one degree of freedom.

Sensitivity of Pose A Note that the input camera pose P_A provides additional information for the encoder, which can be obtained by the method of [15]. We analyze how quality of the results changes with varying pose accuracy. As shown in Fig. 3(a), the input camera pose of encoder is influential in synthesis while its effect is stable within a certain perturbation range. Besides, this stable range surpasses the deviation margin of modern camera pose evaluation methods [15], which is roughly $\pm 10^{\circ}$. Thus normal pose perturbation does not impede our method in practice.

Sensitivity of Pose B The target camera pose P_B is determined by users. Note that the distance between P_A and P_B also influences the synthesis quality. Fig. 3(b) shows the mean L_1 error versus the difference between P_A and P_B. The error remains small within 90 degrees, but rises over 90 degrees. This confirms our intuition – if P_B is very different with P_A, it is hard to synthesize the target because more information is missing.

4.3. Latent Feature Encodes 3D Information?

It is noted that by imposing view-independent constraint, our encoder implicitly captures 3D structure of objects. To demonstrate this, we show that a learned encoder can help learning of 3D tasks. Two schemes for 3D face landmark estimation are adopted with the same network. This network consists of two parts where the encoder is identical with the encoder in VI-GAN and Multilayer Perceptron (MLP) is with 2-layers for estimating the coordinate of landmarks based on features extracted by the encoder.

These two schemes are designed with the following procedures: (1) The overall network is trained from scratch to learn 3D features directly. (2) The encoder is pre-trained by VI-GAN with the view-independent constraint. 3D supervised data is then used to train the overall network.

We use 300W-LP [36] as training data whose 3D landmarks are obtained by utilizing their 3DMM parameters. Besides, the mean Normalized Mean Error (NME) [36] is employed for evaluation. The testing data includes 2,000 images from AFLW2000-3D [17], and each image contains 68 landmarks. When the train loss of both settings no longer changes, we report their results where the mean NMEs of setting (1) and (2) are 12.7% and 6.8% respectively. This demonstrates that the feature learned by the encoder of VI-GAN is 3D-related. It gives a good initialization for the 3D learning tasks. In the future, we plan to explore more 3D tasks with VI-GAN.

5. Applications

As a general framework, our model does not need much task-specific knowledge, and thus is applicable to various applications. In the following, we take face and object rotation as applications to demonstrate the effectiveness of our approach. All the models of VI-GAN in experiments are trained with Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is 10^{-4}. The batch size is set as 24. In each training epoch, we train one step for the generator and one step for the discriminators. The image size is 128×128 for each dataset. Experiments are all conducted on one TITAN V GPU.

5.1. Face Rotation

Discrete Face Rotation Face rotation aims to synthesize a human face of another view. As indicated in Section 4, Multi-PIE [9] contains 13 view points at head height, and thus is suitable for this task. We evaluate our approach in the aligned and unaligned settings. For the aligned setting, all faces are aligned and only the face region is used for both training and testing. This reduces variation of images, and makes the method focus on the face part. The unaligned setting means all images are not cropped, which is more challenging.

We compare our approach with CR-GAN [29] and DR-GAN [30]. CR-GAN utilizes two learning pathways in GAN to improve synthesis; DR-GAN disentangles identity representation from other face variations to synthesize identity-preserving faces at target poses. Results of these settings are shown in Figs. 4 and 5.

Our method outperforms CR-GAN and DR-GAN in both aligned and unaligned settings. DR-GAN may generate images without correct lighting. Although CR-GAN generates better results, the synthesized images could be less natural as shown in the red border of Fig. 5.

The Frechet Inception Distance (FID) [11] is commonly employed to measure the quality of generated images. The lower FID is, the closer distance the domains of real and generated images have. The FID of aligned and unaligned settings are shown in Table 2, which manifest decent performance of our system. Besides, the L_1 error and SSIM are calculated, which also support this conclusion.

Continuous Face Rotation Note that both CR-GAN and DR-GAN can only synthesize face images of discrete viewpoints. To evaluate our approach in a continuous setting, we additionally conduct experiments on 300W-LP [36] dataset, whose images contain continuous camera poses. In this setting, PRNet [6] is used for comparison. PRNet [6] uses the UV position map in 3DMM to record 3D coordinates and trains CNN to regress them from single views.

Fig. 6 qualitatively compares our method with PRNet [6]. The PRNet model is also trained on 300W-LP with publicly available implementation. As illustrated, PRNet [6] may introduce artifacts when information of certain regions is missing. This issue is severe when turning a profile into a frontal face. In contrast, our model produces more realistic images from different viewpoints.

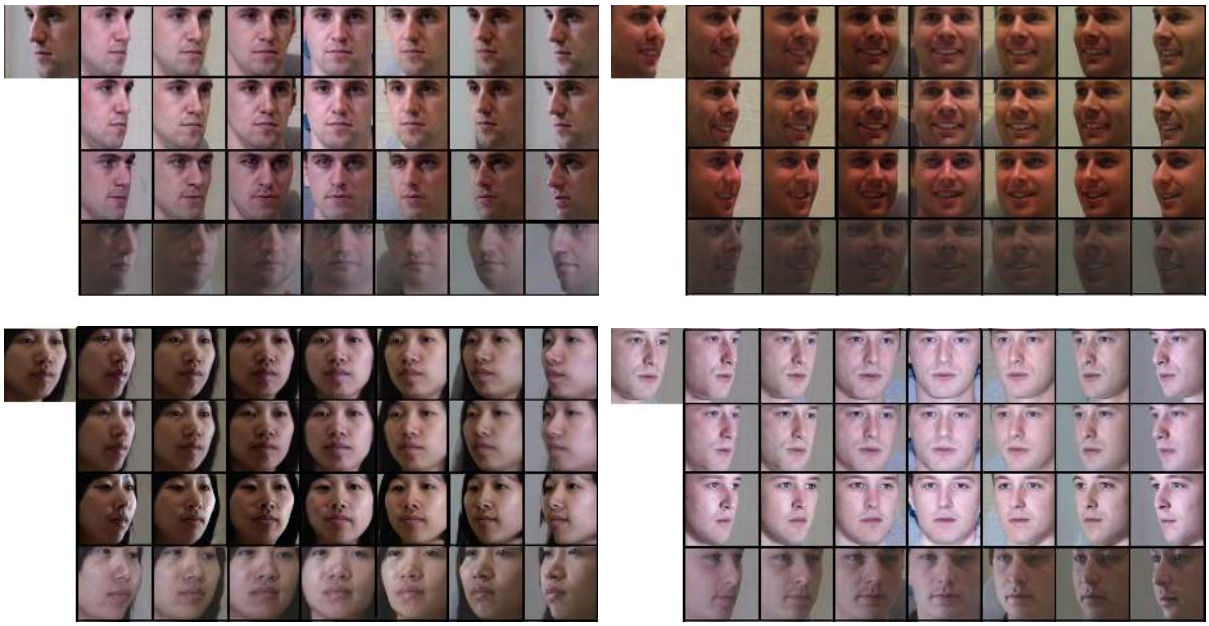


Figure 4. Comparison on aligned Multi-PIE. For each image, the top row is the ground truth while the second row is generated by VI-GAN. The third and fourth rows are the output of CR-GAN [29] and DR-GAN [30] respectively. Obviously, DR-GAN cannot handle the pose-irrelevant factors, such as lighting.

Figure 5. Comparison on unaligned Multi-PIE. For each image, the top row is the ground truth. The other rows are synthesized by VI-GAN, CR-GAN [29] and DR-GAN [30] from top to bottom. The images in red box on the right are obtained by zooming into the front face in red box on the left. Though CR-GAN can generate multiple views, the synthesized faces are blurry.

Method	Aligned			Unaligned		
	FID	L_1	SSIM	FID	L_1	SSIM
CR-GAN	8.76	10.17	0.76	13.92	15.45	0.68
DR-GAN	107.5	31.92	0.36	151.1	43.11	0.23
VI-GAN	**6.51**	**5.86**	**0.88**	**9.05**	**9.73**	**0.80**

Table 2. FID, mean pixel-wise L_1 error, and SSIM of different methods with respect to aligned and unaligned situations. For FID and L_1 error, the lower the better; for SSIM, the higher the better.

Method	Chair		Sofa		Bench	
	L_1	SSIM	L_1	SSIM	L_1	SSIM
MV3D	24.25	0.76	20.24	0.75	17.52	0.73
AF	18.44	0.82	14.42	0.85	13.26	0.77
VI-GAN	**12.56**	**0.87**	**11.52**	**0.88**	**10.13**	**0.83**

Table 3. Mean pixel-wise L_1 error and SSIM between the ground truth and predictions given by different methods.

We also build a quantitative evaluation scheme when turning into frontal faces. Given a synthesized frontal image, it is aligned to its ground truth followed by cropping into facial area. Its ground truth is also cropped in the same fashion. L_1 error and SSIM are calculated between two facial areas. For PRNet, the L_1 error is 22.65 and SSIM is 0.65; For VI-GAN, the L_1 error is 15.32 and SSIM is 0.73. Hence, VI-GAN yields higher precision.

5.2. Object Rotation

Object rotation aims at synthesizing novel views for certain objects. Compared with faces, rotation of general objects is more challenging, as different objects may have diverse structure and appearance. To show the capacity of our model, we evaluate our model on the ShapeNet [3] dataset using samples of "chair", "bench" and "sofa". Results are illustrated in Figs. 7, 8 and 9 respectively. Results on more categories are included in supplementary material.

Figure 6. Comparison between VI-GAN and PRNet [6]. For each image, the top, second and the third rows are images of ground truth, VI-GAN and PRNet respectively. PRNet does not handle profile cases well while our output matches ground truth better.

(a) Input (b) MV3D (c) AF (d) Ours (e) GT

Figure 7. Results in "chair" category. (a) is the 2D input view. (b) and (c) are generated by MV3D [28] and AF [34] respectively. (d) is the result synthesized by our system while (e) is the ground truth. The images are with size of 128×128. It is clear that VI-GAN outperforms both MV3D and AF.

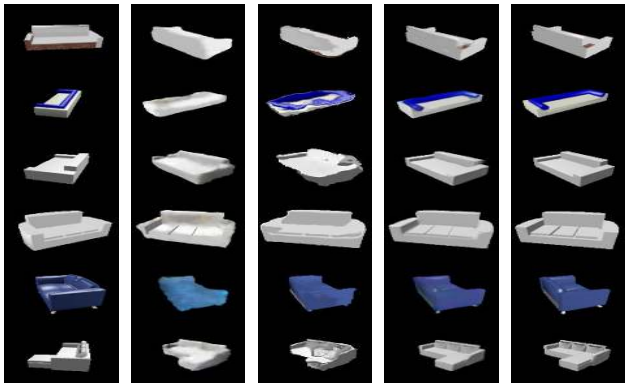
MV3D [28] and Appearance-Flow (AF) [34] are two methods that perform well on this dataset. They deal with continuous camera poses by taking the difference between the 3×4 transformation matrices of the input and target views as the pose vector. We compare our model with them both qualitatively and quantitatively. As shown in Figs. 7, 8 and 9, MV3D [28] and AF [34] miss small parts, while our results are closer to the ground truth. Table 3 shows that our model achieves the lowest L_1 error and the highest SSIM.

6. Conclusion

We have proposed a novel 3D view synthesis network, called VI-GAN, which can generate target views from a singe RGB image with continuous camera parameters. Our system combines benefit of current learning and geometry-based methods by inferring view-independent latent representation. Our experiments demonstrate that our method outperforms existing techniques on a wide range of datasets. VI-GAN is trained with weakly supervised 2D data, while learned features are beneficial to 3D-related learning tasks.

(a) Input (b) MV3D (c) AF (d) Ours (e) GT

Figure 8. Results in "bench" category. Order of each column is the same as that of Fig. 7.



(a) Input (b) MV3D (c) AF (d) Ours (e) GT

Figure 9. Results in "sofa" category. Order of each column is the same as that of Fig. 7.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999. 1, 2

[2] Jie Cao, Yibo Hu, Bing Yu, Ran He, and Zhenan Sun. Load balanced gans for multi-view face image synthesis. *arXiv e-prints*, 2018. 2

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv e-prints*, 2015. 4, 5, 7

[4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1, 4

[5] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015. 2

[6] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. *arXiv e-prints*, 2018. 2, 6, 8

[7] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2

[9] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 2010. 4, 5, 6

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 4

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 6

[12] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3

[13] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *arXiv e-prints*, 2018. 4

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 4

[15] Alex Kendall, Roberto Cipolla, et al. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 2, 3, 6

[16] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Graphics*, 2014. 1, 2

[17] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCVW*, 2011. 6

[18] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv e-prints*, 2015. 4

[19] Jangwon Lee and Michael S Ryoo. Learning robot activities from first-person human videos using convolutional future regression. In *CVPRW*, 2017. 1

[20] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. *arXiv e-prints*, 2017. 2

[21] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. 4

[22] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv e-prints*, 2018. 3

[23] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv e-prints*, 2015. 5

[24] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, 2017. 1, 2

[25] Jhony K Pontes, Chen Kong, Sridha Sridharan, Simon Lucey, Anders Eriksson, and Clinton Fookes. Image2mesh: A learning framework for single image 3d reconstruction. *arXiv e-prints*, 2017. 2

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv e-prints*, 2015. 1

[27] Konstantinos Rematas, Chuong H Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Novel views of objects from a single image. *IEEE TPAMI*, 2017. 1, 2

[28] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2016. 1, 2, 8

[29] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. Cr-gan: learning complete representations for multi-view generation. *arXiv e-prints*, 2018. 1, 2, 6, 7

[30] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017. 1, 2, 6, 7

[31] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 2016. 4

[32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 5

[33] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV*, 2016. 2

[34] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 1, 2, 8

[35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv e-prints*, 2017. 4

[36] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016. 4, 5, 6