Computers & Graphics 82 (2019) 275–285

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on SMI 2019

Learning multi-view manifold for single image based modeling

Jiahao Cui [a,*], Shuai Li [a,b], Qing Xia [a], Aimin Hao [a], Hong Qin [c,*]

[a] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100083, China
[b] Qingdao Research Institute, Beihang University, Qingdao 266000, China
[c] Department of Computer Science, Stony Brook University (SUNY Stony Brook), Stony Brook, NY 11794-2424, USA

ARTICLE INFO

Article history:
Received 20 March 2019
Revised 27 May 2019
Accepted 28 May 2019
Available online 3 June 2019

Keywords:
Multi-view
Generative network
Manifold learning
3D generation

ABSTRACT

Image based modeling has an inherent problem that the complete geometry and appearance of a 3D object cannot be directly acquired from limited 2D images, namely reconstruction of a 3D object when only sporadic views are available is challenging due to occlusions and ambiguities within limited views. In this paper, we present a generative network architecture to address the problem of single image based modeling by learning multi-view manifold of 3D objects, which we call Multi-view GAN. Penalties for shape identity consistency and view diversity are introduced to guide the learning process, and Multi-view GAN can provide a powerful representation which consists of 3D descriptors both for shape and view. This disentangled and oriented representation affords us to explore the manifold of views, thus one can detail a 3D object without "blind spot" even if only single view is available. We have evaluated our method on multi-view and 3D shape generation with a wide range of examples, and both qualitative and quantitative results demonstrate that our Multi-view GAN significantly outperforms state-of-the-art methods.

1. Introduction

Image based modeling is a highly efficient and WYSIWYG modeling method for 3D objects and even for large-scale scenes, therefore it is one of the most attractive topics in computer graphics and computer vision. Over the past decades, researchers have made impressive progresses on image based modeling [1–4]. However, most of these methods are limited to certain undesired restrictions. For instance, a dense photo gallery from various views of an object is usually needed and these views should also be calibrated using relative baseline as that in [2], so it is impractical when users want to model the object from sporadic views or even one single view. In the case of modeling from one input image, depth information and neural networks are commonly needed to reconstruct the structure of the object, infer unseen parts and complete the shape by referencing existing database [5]. These restrictions stem from an inherent shortcoming of image based modeling that the complete geometry and appearance of a 3D object cannot be acquired from limited 2D images.

To overcome these challenges, a wide variety of approaches have been proposed, which can be grouped into two categories. Some works focus on view synthesis and build explicit or implicit correspondences between views with neural networks [6,7]. Based on this, new views are generated from existing pixels or pieced codes that encoded from source views and modified with target view parameters. The synthesized results of these methods are desirable in similar views and image resolutions, but they may fail when there is a large difference between the viewpoint of the source image and the target image. Other works concentrate on learning the disentangled representations [8–10], where supervised learning is applicable in embedding intrinsic features and viewpoints into designed descriptors. However, the learned descriptors tend to be more descriptive or discriminative than generative, which leads to limited content variation and image resolution in generated views.

We re-consider the problem of single image based modeling as multi-view generation, which has significant and practical applications in computer graphics. Given a single view of a 3D object, our goal is to generate images of the object viewed from plentiful viewpoints. Multiple generated 2D views constitute an efficient representation for 3D object modeling. This task is extremely challenging due to the lack of 3D knowledge of the object when only observed in a single view. Additionally, the ambiguities of geometry due to occlusions also make the problem intricate. Our contributions focus on learning multi-view manifold of 3D objects and generating new views with a learned representation. As shown in Fig. 1, we present a multi-view manifold learning framework, we call Multi-view GAN, for single image based modeling. Multi-view

* Corresponding authors.
E-mail addresses: cuijh@buaa.edu.cn (J. Cui), qin@cs.stonybrook.edu (H. Qin).

https://doi.org/10.1016/j.cag.2019.05.030
0097-8493/

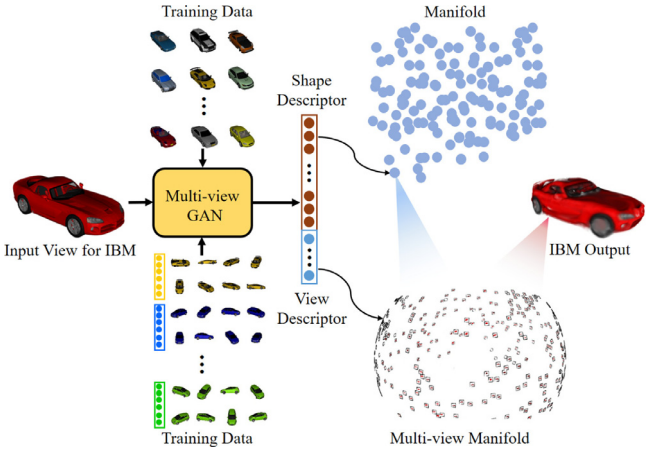

Fig. 1. Given a single view image as input, Multi-view GAN produces a unified representation which consists of a 3D shape descriptor and a view descriptor. The complete representation is learned from the manifold of training data, especially from the multi-view manifold. For single image based modeling, the representation can be used to synthesize identity preserved images at plentiful viewpoints specified by the view descriptors in embedding space of views.

GAN provides a complete representation of 3D objects which can interlink the shape manifold and the view manifold in a reduced latent space.

Generative Adversarial Networks (GANs) [11] can generate samples following a data distribution through a two-player game between a generator G and a discriminator D. Despite many recent promising developments, image synthesis remains to be the main objective of GANs [12,13]. Different from the discriminator in conventional GANs, our D not only does the task of real/fake image classification but also suggests generator to synthesize a group of identity preserved images with different viewpoints of the same object. We conduct G with an encoder-decoder structure to operate a two-stage training process. Multi-view GAN first captures the data distribution in the latent space of shape. Afterwards, multi-view images are grouped as input and the encoder G_{enc} outputs their shape descriptors which are then modified with noise vectors for view manifold learning. Identity consistency penalty, which is computed with feature matching results among views, is attached to the objective function in the second training stage. At the same time, in order to avoid mode collapse in view latent space, view diversity is encouraged by the objective function developed from the standard deviation of global feature maps which are outputs of the later convolution layers in D. By this means, Multi-view GAN is capable of capturing the data distribution both in the latent space of shape and view.

The learned representation is generative, disentangled and oriented. The input to encoder G_{enc} is an image of a 3D object from a general viewpoint, the outputs of decoder G_{dec} are synthesized images at numerous target viewpoints sampled in view manifold. In other words, the learned representation bridges G_{enc} and G_{dec}, and also enable us to detail 3D objects without "blind spot" by navigating the manifold of view subspace and generate new shapes with the manifold of shape. A wide range of experiments in shape morphing and interpolation of views show that the proposed technique achieves significant improvement compared to existing methods. Our salient contributions can be summarized as follows:

1. We propose Multi-view GAN, a holistic learning framework that can capture data distribution in multi-view manifold space and the whole latent space of 3D objects.
2. We design a novel discriminator which is aware of geometry consistency and view diversity across a group of images. Meanwhile, it certainly holds the ability to evaluate the probability that an image came from the training data rather than the generator.
3. We show the advantages of the learned representations by navigating in the manifold of views and shapes to detail a 3D object from more views and generating new 3D shapes.

2. Related work

2.1. Multi-view synthesis

Generating new views of one object based on a few input views is a longstanding problem in computer graphics and vision. A large body of works benefit from explicit geometric reasoning to address this problem. Traditional methods for view synthesis directly reuse the pixels from available images. Debevec et al. [14] use the potential geometry to synthesize multiple views by rendering new views. Gortler et al. [15] capture the complete appearance of objects and use a subset of the plenoptic function to describe the light field. The description is then used to render images of the object from new camera positions. Seitz and Dyer [16] synthesize image changes of viewpoint by prewarping two images prior to computing a morph and then postwarping the interpolated images. A number of recent works in this area have used a unified framework for these techniques [17,18]. These hand-built methods do not leverage training data and can therefore only generate already seen content. In cases when multiple images are available, modern multi-view stereo algorithms [19] have demonstrated their ability to generate impressive quality results. An alternative approach proposed by Flynn et al. [20] perform compositing through learned geometric reasoning using a Convolutional Neural Networks (CNNs), and can generate intermediate views of a scene by interpolating from a set of surrounding views. Ji et al. [21] propose to rectify the two view images first with estimated homography by deep networks, and then synthesize intermediate view images with other deep networks. These methods fundamentally rely on finding visual correspondences and composting the corresponding input image rays for each output pixel. Therefore, they can only generate already seen content or break down when there are only a couple of views from very different viewpoints.

Other multi-view generation methods utilize CNNs to function as image decoders [8]. Dosovitskiy et al. [6] train a CNN that is capable of functioning as a renderer, but the network requires explicitly factored representations of object identity, pose and color. Tatarchenko et al. [22] and Yang et al. [23] build on this work utilizing the insight that the graphics code, instead of being presented explicitly, can be implicitly captured by an source image along with the desired transformation. A common module in these methods is a decoder CNN to generate pixels corresponding to the transformed view from an implicit/explicit graphics code. Due to the challenges of disentangling the factors from single-view and the use of globally smooth pixel-wise similarity measures, the generation results tend to be blurry and in low resolution. More recently, Zhou et al. [24] trained a deep generative convolutional encoder-decoder model to generate an appearance flow vector indicating the corresponding pixel in the input view to copy from. However, direct transformations are clearly upper-bounded by the input. Park et al. [7] first explicitly infer the parts of the geometry visible both in the input and novel views and then re-cast the remaining generation problem as image completion.

2.2. Representation learning

Synthesizing new views of objects can be thought as decoupling viewpoint and identity, and has long been studied as part of representation learning and view-invariant recognition. Hinton et al.

[25] learn a hierarchy of computational units, which locally transform their input, for generating rotations to an input stereo pair, and argue for the use of similar computational units for recognition. Cheung et al. [26] propose an auto-encoder with decoupled semantic units representing identity, pose and other factors of variation and demonstrate that the proposed method is capable of synthesizing new views of faces. Jaderberg et al. [27] argue for the use of computational layers that perform spatial transformation over their input features as effective modules for recognition tasks. Jayaraman and Grauman [28] study the task of synthesizing features transformed by ego-motion and demonstrate its utility as an additional task for learning semantic representation space. Kulkarni et al. [8] propose a similarly motivated variational method for decoupling and manipulating the factors of variation for images of faces. While the representation learning approaches convincingly demonstrate the ability to disentangle factors of variation, the view manipulations demonstrated are typically restricted to small rotations or categories with limited shape variance like digits and faces.

Prior works also explore joint representation learning and face rotation for Pose-Invariant Face Recognition. In [29], Multi-View Perceptron is used to untangle the identity and view representations by processing them with different neurons and maximizing the data log-likelihood. Yim et al. [30] use a multi-task CNN to rotate a face with any pose and illumination to a target pose, and the L2 loss-based reconstruction of the input is the second task. Both works focus on image synthesis and the identity representation is a by-product during the network learning. In contrast, DR-GAN [10] focuses on representation learning, of which face rotation is both a facilitator and a by-product. The discriminator in DR-GAN is a multi-task CNN which deal with real/fake image classification along with identity and pose classification. On account of employing labeled dataset in supervised learning, the learned representation is discrete in view space and lack of generative capacity. Similarly, Tian et al. [9] introduce CR-GAN, which include a generation sideway to maintain the completeness of the learned embedding space. Compared with DR-GAN, CR-GAN yields better quality multi-view image generations from unseen data in wild conditions. However, the identity of the object is not well retained in a series of generations.

3. Method

3.1. Problem statement and overview

Given a set of views $\mathbf{x} = \{V_{ij} | i = 1, \ldots, n, j = 1, \ldots, m\}$ belonging to several 3D objects of the same class, where n is the number of objects in this class and m is the number of views of each object in this class, the goal of our generative network is to learn the view subspace as well as the whole latent space of these 3D objects. Furthermore, we train a decoder in the generator part to embed views of one object into the latent space in order to navigate in the manifold of views and get the complete representation of this object.

As proposed in [11], the Generative Adversarial Network (GAN), which consists of a generator G and a discriminator D, is capable of capturing the data distribution in the latent space. The generator attempts to confuse the discriminator by keeping trying to synthesize realistic-looking images from a random noise vector $\mathbf{z}$. The discriminator tries to distinguish between real images $\mathbf{x}$ and synthesized ones $G(\mathbf{z})$. In practice, D and G play a game with loss function:

$$\max_D \mathcal{L}^D_{gan} = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))], \quad (1)$$

$$\max_G \mathcal{L}^G_{gan} = \mathbb{E}_{z\sim p_z(z)}[\log(D(G(z))]. \quad (2)$$

This adversarial game has a global optimum when the distribution p_z of the synthesized samples and the distribution p_{data} of the real samples are the same. Compared to prior GANs, our Multi-view GAN also needs to capture the distribution p_{view} of views in a subspace for each object. To this end, different from the discriminator in conventional GAN, our D not only does the task of real/fake image classification but also suggests G to generate a group of images from different viewpoints of the same object. Therefore, two additional estimations are included in our objectives. Given a group of real images $\mathbf{x}$, D aims to classify it as the real group of views belong to the same object. While given a group of synthesized views from the generator $\hat{\mathbf{x}} = \{G(z_i) | i = 1, \ldots, n\}$, D attempts to classify $\hat{\mathbf{x}}$ as fakes, using the following objectives:

$$\mathcal{L}^D_{gan} = \sum_{i=1}^{n} \mathbb{E}_{x_i\sim p_d(x_i)}[\log(D(x_i)] + \sum_{i=1}^{n} \mathbb{E}_{z_i\sim p_z(z_i)}[1 - \log(D(G(z_i))], \quad (3)$$

$$\mathcal{L}^D_{id} = f_{id}(F^g(\mathbf{x})) - f_{id}(F^g(\hat{\mathbf{x}})), \quad (4)$$

$$\mathcal{L}^D_{view} = f_{view}(F^l(\mathbf{x})) - f_{view}(F^l(\hat{\mathbf{x}})), \quad (5)$$

where $\mathcal{L}^D_{id}$ is the penalty for object identity consistency across different views and $\mathcal{L}^D_{view}$ is the penalty for views diversity. These two penalties are computed by two functions f_{id} and f_{view} respectively using global features F^g and local features F^l from discriminator D. More details about features and functions will describe in Sections 3.2 and 3.3 The final objective for training D is the weighted average of all objectives:

$$\max_D \mathcal{L}^D = \lambda_g \mathcal{L}^D_{gan} + \lambda_{id} \mathcal{L}^D_{id} + \lambda_{view} \mathcal{L}^D_{view}. \quad (6)$$

Meanwhile, G aims to learn a mapping from a group of correlative and assembled codes to synthesized images $\hat{\mathbf{x}} = \{G(z_i) | i = 1, \ldots, n\}$. Each of these codes consists of two parts: identity code and view code. To generate a group of synthesized images which seem like multi-views of the same object, identity codes in a code group are the same to each other, meanwhile view codes are sampled from a uniform distribution. G attempts to puzzle D with its generations using the following objectives:

$$\mathcal{L}^G_{gan} = \sum_{i=1}^{n} \mathbb{E}_{z_i\sim p_z(z_i)}[\log(D(G(z_i))], \quad (7)$$

$$\mathcal{L}^G_{id} = f_{id}(F^g(\hat{\mathbf{x}})), \quad (8)$$

$$\mathcal{L}^G_{view} = f_{view}(F^l(\hat{\mathbf{x}})), \quad (9)$$

$$\max_G \mathcal{L}^G = \lambda_g \mathcal{L}^G_{gan} + \lambda_{id} \mathcal{L}^G_{id} + \lambda_{view} \mathcal{L}^G_{view}. \quad (10)$$

On one hand, after training D and G alternately, D becomes more alertness with synthesized images and the additional penalties force G to generate images satisfying the criterion of identity consistency and views diversity. On the other hand, with the assembled codes and grouped views as inputs, views belonging to the same object are embedded to neighboring positions in the latent space, which consequently forms view subspace separated by identity codes. In other words, the learned representation is explicitly disentangled.

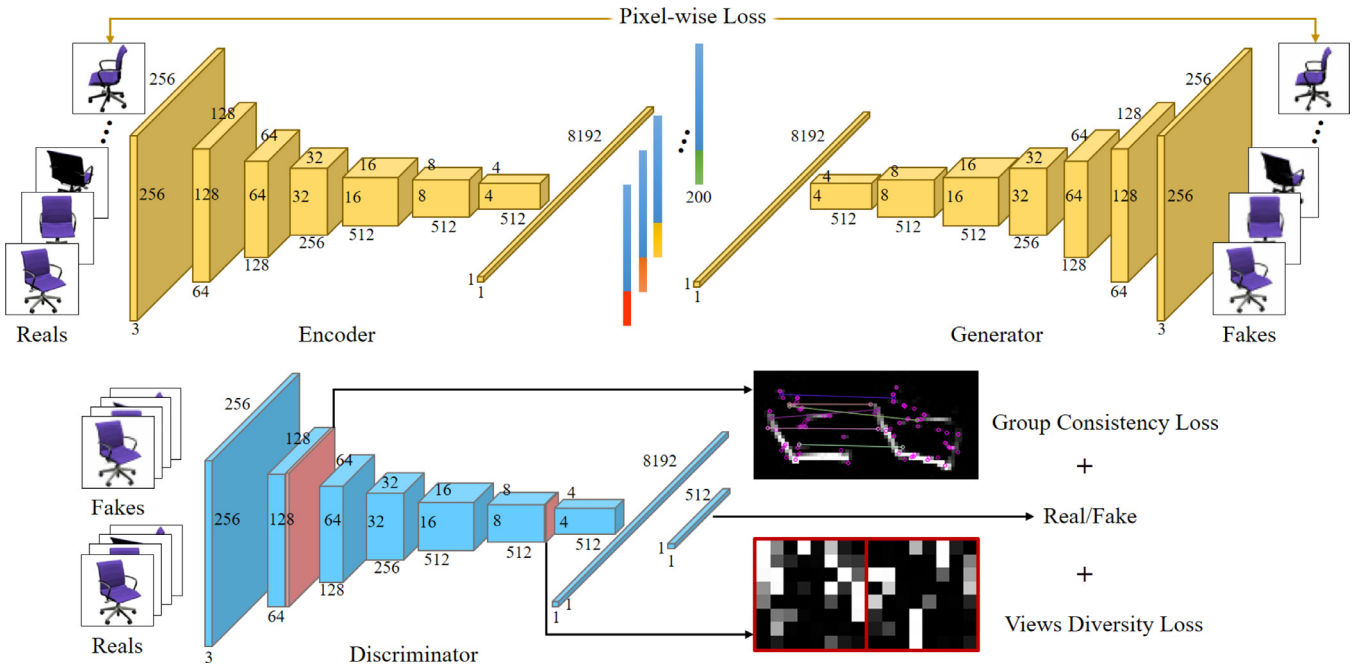

Fig. 2. The network structure of our Multi-view GAN. The encoder-generator-discriminator structure is required for a two-stage training process. The network first captures the data distribution in the latent space of shape. Besides, the encoder is trained to produce descriptors that can be decoded into original images. Modified noise vectors and losses of group consistency and view diversity are used in the second training stage for view subspace learning. Feature maps from the earlier and the later convolution layers participate in the computation of losses respectively.

3.2. Network architecture

GANs have the ability to produce appealing samples, but the training processes of these networks are not always stable, especially in the case of generating high-resolution images. In addition, GANs have a tendency to capture only a subset of the variation found in training data (mode collapse). These two weaknesses of GANs bring greater challenges to our task. One reason is that higher resolution means more details for our discriminator to estimate identity consistency through local features. Another reason is that the sample distribution of multi-views is concentrated naturally which may induce mode switch or mode collapse. In order to overcome these problems, we adopt WGAN-GP loss [31] in the task of real/fake image classification and add layers progressively during training as those in [32]. The network structure of Multi-view GAN is shown in Fig. 2.

In our Multi-view GAN, the generator G maps a 200-dimension latent vector $\mathbf{z}$ to a 256×256 image with upsampling and convolution. For the latent vector, identity code is the first 164 dimensions and view code is the remaining 36 dimensions. We randomly sample each dimension independently from a uniform distribution in the range of $[-1, 1]$. The discriminator D basically mirrors the generator except that it uses average pooling to achieve downsampling. During view manifold learning, feature maps of the first two layers in D also take part in the computation of penalties in Eqs. (4) and (8). Meanwhile, feature maps of the last but one layer are used to compute penalties for views diversity in Eqs. (5) and (9). Leaky ReLU [33] layers are used after convolution layers in both G and D. The encoder E has the same structure with D but without the fully connected layer for real/fake classification.

The encoder-generator-discriminator structure is required for a two-stage training process. We firstly feed the network with images of different objects. The inputs are not grouped but they have same viewpoint. Simultaneously the input vectors for G are completely irrelevant neither in identity code nor in view code across a training batch. Identity consistency and view diversity penalties are not involved in this pre-training stage and our purpose here is only to obtain a coarse distribution of the training dataset. The encoder E is trained when G and D are fixed after the alternating training. Given an input image, E produces an output in the same form of latent vector. G reconstructs this image with the output code and the generation is compared with the target image. We adopt pixel-wise loss to enforce E to be trained as an inverse of G and be able to yield representations of training data.

In the second training stage, for each image used in pre-training, E provides a 200-dimensional vector to locate the corresponding object in the latent space. We truncate the vector at 165th dimension and concatenate it with 36-dimensional noise vectors. This time we feed the network with grouped views and grouped latent vectors which have the same identity code. Identity consistency penalty constrains the mapping results of similar vectors to indicate the same target. Variation of views is encouraged by view diversity penalty. Based on the learned coarse data distribution, the network is further trained to learn the view subspace as well as to disentangle identity and view information in the final representation.

3.3. Identity consistency and views diversity

The modified latent vectors in a group are the same in the first 164 dimensions, which means they can represent neighbor parts in the latent space. However, these neighbor parts may indicate images that are different in shapes, colors or other aspects after the network learning the embedding. That is, the differences may not be exactly reflected in viewpoints and may disturb shapes described by fixed dimensions. Therefore, we introduce penalty functions of identity consistency and views diversity to constrain the generator to produce multi-views with the modified latent vectors. Afterwards, the network can learn the view subspace and disentangle identity and view information in the representation. The penalties enable discriminator to recognize a group of images that satisfy the establishing conditions of multi-views and provide guidance for the training of generator.

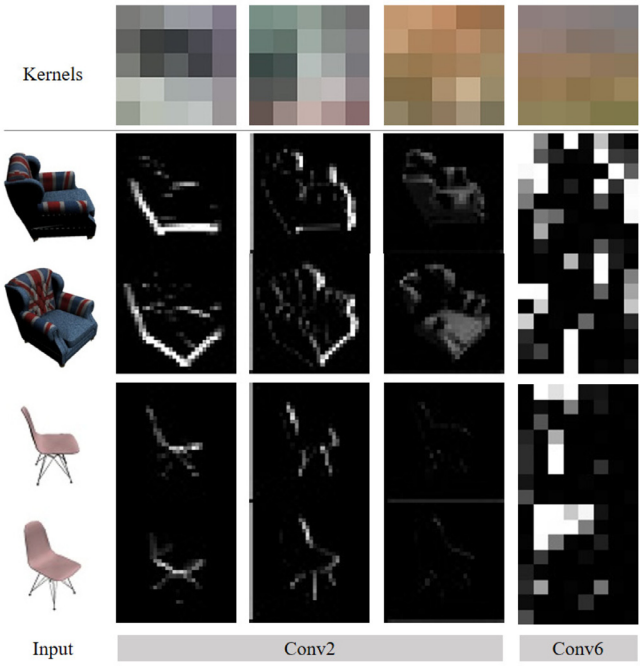

Fig. 3. Visualization and comparison of feature maps and convolution kernels from discriminator after the first training stage. Feature maps produced by the earlier layers present legible outlines or shapes. While feature maps produced by the later layers are more complicated to interpret, there is significant diversity among feature maps which result from the same kernel but correspond to different views.

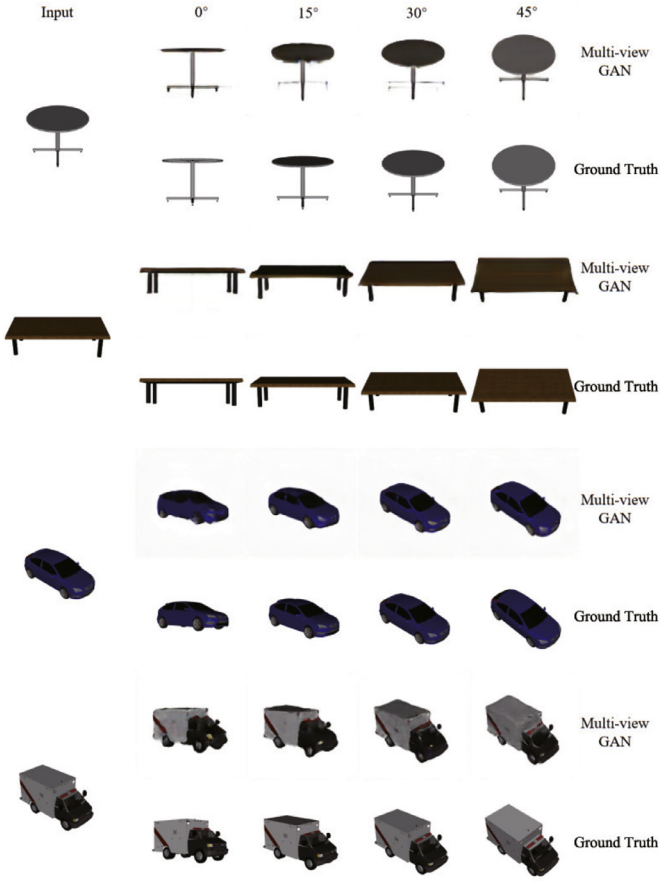

Fig. 4. Examples of views generation (elevation angle). In each pair of rows, images in the top row are the generation results at four target viewpoints and images in the second row are the corresponding ground truth images.

In systems for 3D reconstruction from collections of images, feature detection and matching are used to solve spatial positions of pixels. Our task can be seen as an inverse project of multi-view 3D reconstruction. We have no need to estimate viewpoint parameters or compute spatial positions because we just take advantage of feature matching to make up relations between images that indicate a same object. We adopt the scale-invariant feature transform (SIFT) algorithm [34] to extract features and use the k-nearest neighbors algorithm (KNN) to deal with feature matching. Matching results are then used to evaluate relations between images.

While on aspect of increasing views diversity, we compute differences of global features between images which are fed into discriminator and use penalty function to stimulate these differences, which is similar to those projects that aim to increase variation of the generation of GANs. Salimans and Kingma [35] suggest minibatch discrimination as a solution. They compute feature statistics not only from individual images but also across minibatches, thus encouraging the minibatches of generated and training images to have similar statistics. This is implemented by adding a minibatch layer towards the end of the discriminator, where the layer learns a large tensor that projects the input activation to an array of statistics. A separate set of statistics is produced for each example in a minibatch and it is concatenated to the layer's output so that the discriminator can use the statistics internally. Karras et al. [32] simplify this approach by computing the standard deviation for each feature in each spatial location over the minibatch while also improve the variation. We test this method in our learning framework but there is no obvious improvement in variation of viewpoints. The reason may be that it is a vague-objective problem for discriminator to find differences among statistical data. Hence, we append the standard deviation result to our objectives directly.

We use feature maps which are outputs of convolution layers of our discriminator to perform penalty function tasks mentioned above. This idea is inspired by studies on the interpretability of convolution neural networks [36,37]. We visualize the feature maps and convolution kernels after the first training stage and the results are shown in Fig. 3. Similar to convolution neural networks which are trained for recognition and classification tasks, kernels for detecting edges, colors and textures appear in the earlier layers. Distinctly, feature maps produced by the earlier layers are results of those kernels and present legible outlines or shapes. Feature maps contain less information than original images universally. However, on account of the low quality outputs of GANs, feature matching between blurry images leads to inaccurate results. Legible outlines and shapes provide equilibrium between original images and synthesized images in feature matching.

For kernel $k_i (i = 1, \ldots, 64)$ in the first convolution layer of D, it produces m (size of input images) feature maps $F^g{}_j (j = 1, \ldots, m)$. After feature matching among feature maps, we sort top 5 matched feature quantities for every feature map. The sum of the top matched feature quantities is transferred into activation function before computing the average matching score f_{id} across all kernels in the layer. While feature maps produced by the later layers are more complicated to interpret, there is significant diversity among feature maps which result from the same kernel but correspond to different views. Penalty for view diversity is computed as the standard deviation across feature maps of the same kernel for a group of input images.

4. Experiments

Our Multi-view GAN aims to learn the view subspace as well as the whole latent space of 3D objects and acquire complete

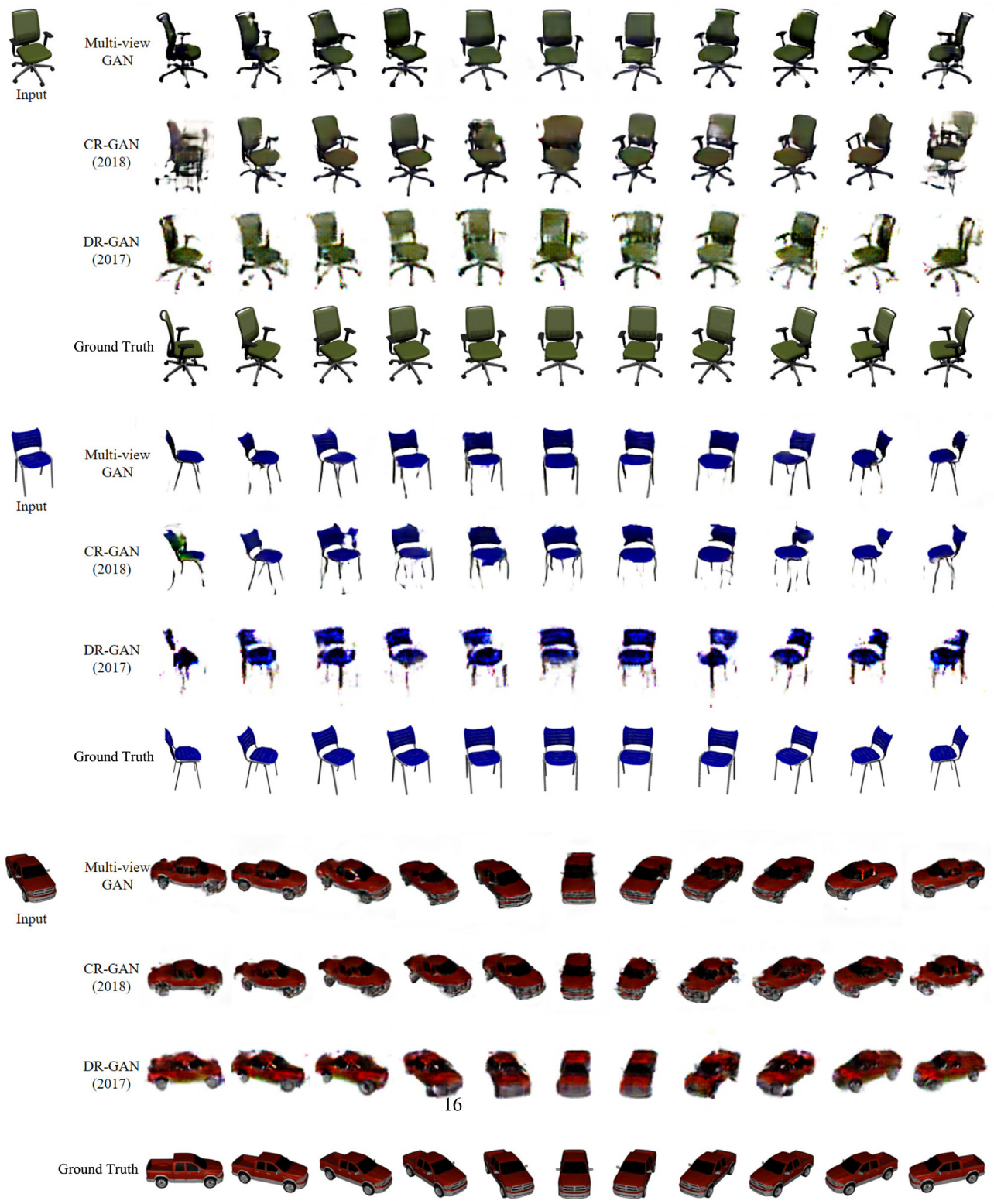

Fig. 5. Results of 180° rotations. We generate multiple new views based on the input image, then sample viewpoints of the same elevation and interpolate in rotation between viewpoints. Compared with CR-GAN and DR-GAN, our Multi-view GAN can generate far more realistic chairs and cars that are similar to the ground truth in all views.

representations sequentially. The learned representations are disentangled and can be used to locate objects in the latent space for generation of other views or even shapes. We achieve this by training an encoder-generator-discriminator network in two stages successively. We have evaluated our Multi-view GAN quantitatively for feature matching among grouped images and qualitatively for shape morphing and interpolation of views. In order to verify the ability of our method to generate 3D shapes, we have also conducted experiments to reconstruct 3D objects with one single view as input.

Fig. 6. Examples of morphing different chairs, cars and tables. Images in red frames are source images and the transformation is manipulated from origin images to terminal images through linear interpolation on shape descriptors. Cars and tables have less differences among objects in shape than chairs. In order to demonstrate the consecutiveness of the learned latent space, images in the yellow frame as well as images on both sides are non-transformed.

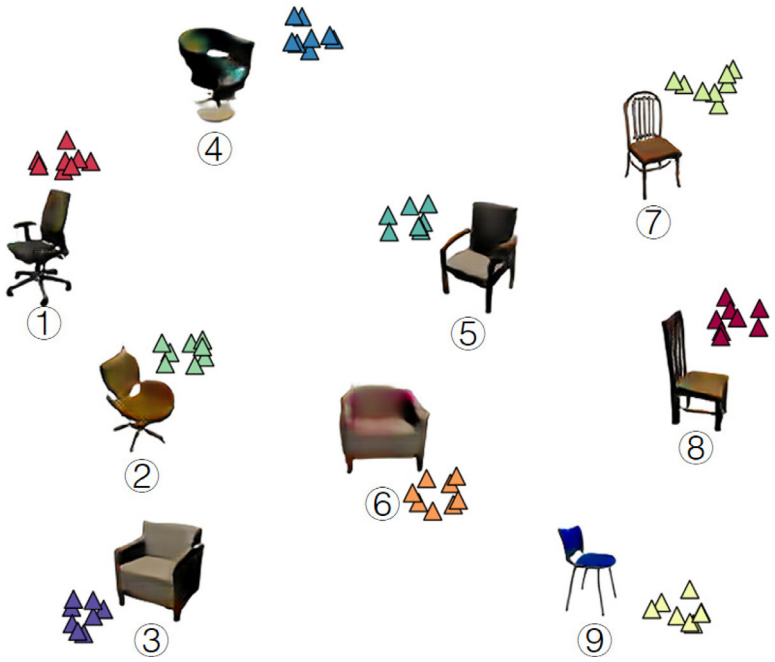

Fig. 7. Visualization for the learned embedding space of Multi-view GAN. Markers with the same color indicate the same object. Aggregations of same color markers mean multi-views of the same object are embedded close to each other in the latent space. View subspace of chair No.9 is shown in Fig. 8.

4.1. Experimental settings

Datasets. We evaluate our Multi-view GAN on datasets of different object classes. We use renderings of 3D models of chairs, made public by Aubry et al. [38] as well as tables and cars from the ShapeNet [39] dataset. For each 3D object class, 90% of models in this class are used for training and the rest 10% are testing instances.

Aubry et al. [38] provide renderings of 1393 chair models, each rendered from 62 viewpoints: 31 azimuth angles (with a step of 11°) and 2 elevation angles (20° and 30°), with a fixed distance to the chair. We find that the dataset includes some approximately duplicate models, which differ only in color, and some low-quality models. After removing these we end up with a reduced dataset of 737 models, which we use in our experiments. We crop the renders to have a small border around the chair and resize them to a common resolution of 256×256 pixels.

We take models of tables and cars from ShapeNet [39], a dataset containing tens of thousands of consistently aligned 3D models of multiple classes. We render a turntable of each model using 16 azimuth angles (from 0° to 350°) and 4 elevation angles (from 0° to 45°), which results in 64 images per model. Positions

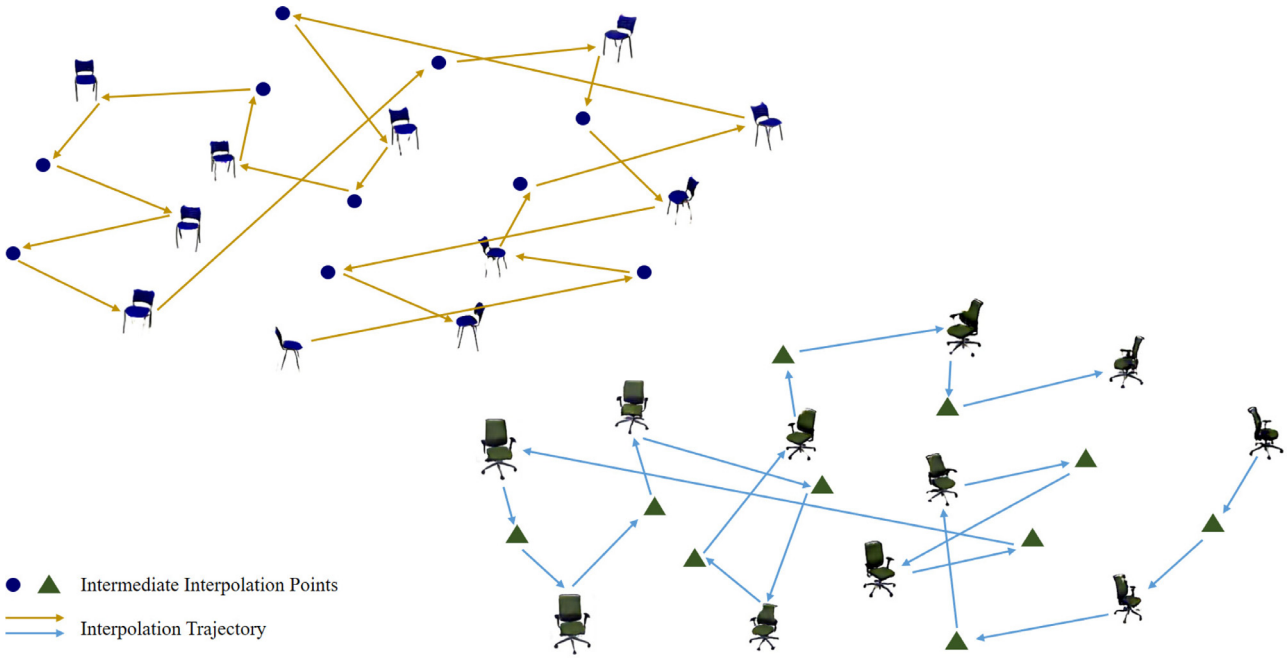

Fig. 8. Visualization for the multi-view manifold and the navigation trajectory. In order to display the interpolation trajectory and navigation directions clearly, we add some intermediate interpolation points based on views in Fig. 5. Images of similar viewpoints are embedded close to each other in the latent space.

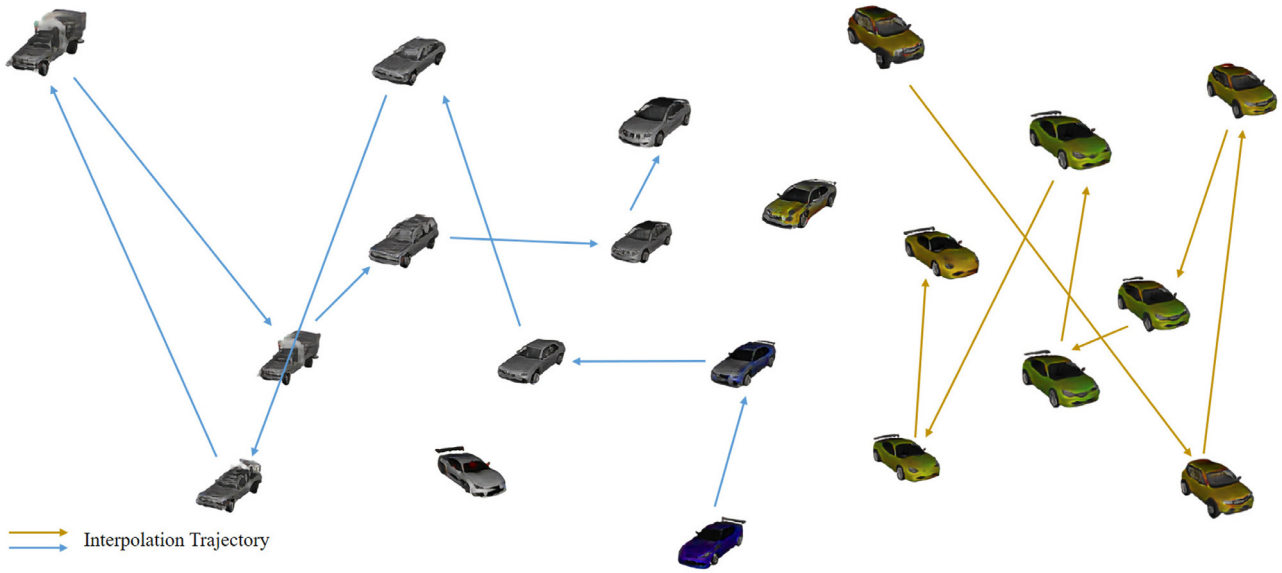

Fig. 9. Visualization for the interpolation trajectory of shape morphing.

of the camera and the light source are fixed during rendering. For experiments in this paper we use renderings of 1328 car models and 1388 table models. All renderings are of 256×256 pixels.

Implementation details. Our implementation is extensively modified from a publicly available implementation of PG-GAN [32]. The batch size is set to be 128 at initial resolution of 4×4 and reduce to 8 at resolution of 256×256. All weights are initialized from a zero-centered normal distribution with a standard deviation of 0.02. We train the networks using Adam optimizer [40]. We do not use any learning rate decay or ramp down, but for visualizing generator output at any given point during the training, we use an exponential running average for the weights of the generator with decay 0.999. Moreover, the networks are trained after 12,000 thousands of images for each dataset.

Evaluation metrics. We utilize the standard L_1 mean pixel-wise error and the structural similarity index measure (SSIM) for evaluation [41,42]. When computing the L_1 error, we normalize the L_1 distance results into the range of [0, 1], lower numbers corresponding to better results. Then we calculate the mean value of these normalized results as the final L_1 error. SSIM is in the range of $[-1, 1]$ and the closer to 1 indicates more structural similarity.

4.2. Multi-view generation

In this section we show that the network is able to generate multi-views of a 3D object with a single view as input. We compare our method with DR-GAN and CR-GAN by computing the L_1 error and SSIM between generation results and ground truth images. In Fig. 4 we show some examples of elevation transfer. For 4 views of each object the effect of view subspace learning is already visible. Synthesized images at target viewpoints are distinct and fine details preserved.

Fig. 10. Failure cases of our method. When dealing with complex structures or 3D objects with rich local details, the embedding is sometimes not accurate. It may be caused by partial learning of the latent space which means the identity codes can not cover the features of shape. As a result the adding of view codes may give rise to a few changes in shape as well as changes in viewpoints (see images in the first row). In other cases, the ambiguity may lead to incomplete generations and/or generations without any meaning (see images in the second row).

Fig. 11. We run a multi-view reconstruction algorithm to obtain a textured mesh with generated views from single view as input. Our method results in better meshes where views generated by CR-GAN failed.

Fig. 5 shows some representative examples of viewpoint rotation. Specifically, we generate multiple novel views from the input image, then sample viewpoints of the same elevation and interpolate in rotation between viewpoints and finally cover an 180° rotation around the object. Compared with CR-GAN and DR-GAN, our Multi-view GAN can generate far more realistic chairs and cars that are similar to the ground truth in all views. Meanwhile, images synthesized by DR-GAN or CR-GAN cannot maintain high frequency components and are blurry. Besides, in spite of the difficulties in generating images from a viewpoint that has wide gap with the input view, our Multi-view GAN can synthesize reasonable images in the condition of large viewpoint transformation where DR-GAN and CR-GAN fail in sharp contrast. In terms of neighbouring views, our Multi-view GAN produces favorable images with smoother transition and better identity preservation. We provide quantitative comparison to these two methods in Table 1. We note that, although commonly used, L_1 and SSIM metrics are not fully correlated with human perception. While our method is clearly better than others in the L_1 baseline, and all methods get comparable SSIM scores.

Table 1
We compare our method to DR-GAN and CR-GAN by computing the L_1 error and SSIM score. For each class of objects, statistics is the average of results from all generated views.

Method	Tables		Cars		Chairs	
	L1	SSIM	L1	SSIM	L1	SSIM
MV-GAN	**0.133**	**0.908**	**0.142**	0.89	**0.236**	**0.885**
CR-GAN	0.168	0.879	0.235	0.892	0.267	0.865
DR-GAN	0.206	0.892	0.261	0.887	0.264	0.87

4.3. Modeling transformations

Remarkably, the generative network can not only imagine previously unseen views of a given object, but also invent new objects by interpolating between given ones. It is the foundation of shape manifold navigation. We show how Multi-view GAN is able to generate chairs, cars and tables that are significantly transformed relative to the original images. in Fig. 6, each row shows a different type of transformation. In the examples of morphing chairs, images on both sides are source images. Even in the presence of large transformations, the quality of the generated images is basically as good as without transformation. The image quality typically degrades a little in case of sharp variation of chair shapes (such as transformation from rotating office chairs pedestal to separated legs). Cars and tables, by contrast, have less differences among objects in shape. In order to demonstrate the consecutiveness of the learned latent space, images in the central column as well as on both sides are non-transformed. The network easily deals with extreme color-related transformations. Also when handling changes in structures, the network is able to find reasonable transition (see

the transformation of numbers of table legs and frames in the second to the last row) or use correlative shapes in the latent space. To obtain interpolation in shapes, we simply linearly change the identity code parts in the latent vector from one object to another. The interpolation result is used to locate newly produced shapes in the latent space and then participate in sampling views that have same viewpoints with source images.

4.4. Latent space learning

Figs. 7 and 8 show the embedding space learned by our Multi-view GAN. We use T-distributed Stochastic Neighbor Embedding (t-SNE) algorithm [43] to visualize the latent vectors of 9 different chairs. Markers with the same color indicate the same object. Multi-view images of the same chair are embedded close to each other in the latent space of shape, which means the identities are well preserved by shape descriptors in the latent vectors. The embedding of shape descriptors reflects the distribution of shapes and other appearance attributes. Chairs with hollow back are embedded neighbouring to each other (see chair No.7 and No.8). Situation is the same for sofas (see chair No.3 and No.6). Transition on appearance can also be observed from chains of neighbouring embedded chairs (Moving along the path from No.6, No.2, No.1 and to No.4 we can see legibly gradual varying of colors and shapes). View subspace of chair No.9 is shown in Fig. 8. Images of similar viewpoints are embedded close to each other in the view subspace.

4.5. Discussion

Visualization for the multi-view manifold and the navigation trajectory are shown in Fig. 8. In order to display the interpolation trajectory and navigation directions clearly, we add some intermediate interpolation points based on views in Fig. 5. Images of similar viewpoints are embedded close to each other in the latent space. Visualization for the interpolation trajectory of shape morphing is shown in Fig. 9. Fig. 10 shows failure cases of our method. When dealing with complex structures or 3D objects with rich local details, the embedding is sometimes not accurate. It may be caused by partial learning of the latent space which means the identity codes can not cover the features of shape. As a result the adding of view codes may give rise to a few changes in shape as well as changes in viewpoints (see images in the first row). In other cases, the ambiguity may lead to incomplete generations and/or generations without any meaning (see images in the second row).

4.6. Application

Constructing 3D geometry of an object from a single image is an attractive problem of computer vision research. Recent methods using deep networks generally use a voxelized 3D reconstruction as output [5,44]. However, computational and spatial complexities of using such voxelized representations in encoder-decoder networks restricts the output resolution considerably. We develop the performance of our learning framework in generating novel views for reconstruction purposes. We generate multiple novel views from the input image to cover a full 360° rotation around the object sampled at 15° intervals as well as 45° elevation sampled at 5° intervals. Afterwards, we run a multi-view reconstruction algorithm [45] on these images to obtain a textured mesh from these views. Fig. 11 demonstrates examples of reconstructed 3D models. By generating views consistent in terms of geometry and details, our method results in better meshes.

5. Conclusions

We propose Multi-view GAN, a holistic learning framework that can capture data distribution in multi-view manifold. We design a novel discriminator which is aware of consistency in geometry and diversity in view across a group of images. Meanwhile, it certainly holds the ability of real/fake image classification. We show the prominence of learned representations in navigating the manifold of views to detail a 3D object as well as in generating new shapes. Our method generates realistic images and outperforms state-of-the-art techniques for novel view synthesis on datasets of renderings where ground truth is known. Our synthesized images are even accurate enough to perform multi-view 3D model generation.

We hope that the proposed image generation pipeline might be also suitable to real photographs. For the future work, we can improve the network to reply to views with complex background from realistic environment. On the other hand, the learned view descriptors are not corresponding to specific view parameters for different objects. The reason is that there is no uniform original point or coordinate axis for heterogeneous 3D shapes. We should also concordant descriptors in different subspace by consistency analysis on images of the same viewpoints.

Acknowledgments

This research is supported in part by the National Key R&D Program of China under Grant No. 2017YFF0106407, National Natural Science Foundation of China under Grant Nos. 61672077 and 61532002, the Applied Basic Research Program of Qingdao under Grant No. 161013xx, National Science Foundation of USA under Grant Nos. IIS-1715985 and IIS-1812606, the Capital Health Research and Development of Special under Grant No. 2016-1-4011, Fundamental Research Funds for the Central Universities, and Beijing Natural Science Foundation-Haidian Primitive Innovation Joint Fund under Grant No. L182016.

References

[1] Xiao J, Fang T, Zhao P, Lhuillier M, Quan L. Image-based street-side city modeling. In: Proceedings of the ACM SIGGRAPH Asia 2009. New York, NY, USA: ACM; 2009. p. 114:1–114:12. ISBN: 978-1-60558-858-2
[2] Furukawa Y, Ponce J. Carved visual hulls for image-based modeling. Int J Comput Vis 2009;81(1):53–67.
[3] Wu J, Xue T, Lim JJ, Tian Y, Tenenbaum JB, Torralba A, et al. Single image 3d interpreter network. In: Leibe B, Matas J, Sebe N, Welling M, editors. Proceedings of the european conference on computer vision – ECCV 2016. Cham: Springer International Publishing; 2016. p. 365–82. ISBN: 978-3-319-46466-4
[4] Fan H, Su H, Guibas L. A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR), 00; 2017. p. 2463–71.
[5] Choy CB, Xu D, Gwak J, Chen K, Savarese S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: Proceedings of the European conference on computer vision – ECCV 2016; 2016. p. 628–44.
[6] Dosovitskiy A, Springenberg JT, Tatarchenko M, Brox T. Learning to generate chairs, tables and cars with convolutional networks. IEEE Trans Pattern Anal Mach Intell 2017;39(4):692–705.
[7] Park E, Yang J, Yumer E, Ceylan D, Berg AC. Transformation-grounded image generation network for novel 3d view synthesis. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2017. p. 702–11.
[8] Kulkarni TD, Whitney WF, Kohli P, Tenenbaum J. Deep convolutional inverse graphics network. In: Proceedings of the advances in neural information processing systems; 2015. p. 2539–47.
[9] Tian Y, Peng X, Zhao L, Zhang S, Metaxas DN. Cr-gan: learning complete representations for multi-view generation. arXiv preprint arXiv:1806.11191 2018.
[10] Tran LQ, Yin X, Liu X. Representation learning by rotating your faces. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2018 (Early Access). doi:10.1109/TPAMI.2018.2868350.
[11] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Proceedings of the advances in neural information processing systems 27. Curran Associates, Inc.; 2014. p. 2672–80.

[12] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 2015.
[13] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of the advances in neural information processing systems; 2016. p. 2172–80.
[14] Debevec PE, Taylor CJ, Malik J. Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach. In: Proceedings of the SIGGRAPH, 96; 1996. p. 11–20.
[15] Gortler SJ, Grzeszczuk R, Szeliski R, Cohen MF. The Lumigraph. In: Proceedings of the SIGGRAPH, 96; 1996. p. 43–54.
[16] Seitz SM, Dyer CR. View morphing. In: Proceedings of the twenty-third annual conference on computer graphics and interactive techniques. ACM; 1996. p. 21–30.
[17] Buehler C, Bosse M, McMillan L, Gortler S, Cohen M. Unstructured lumigraph rendering. In: Proceedings of the twenty-eighth annual conference on Computer graphics and interactive techniques. ACM; 2001. p. 425–32.
[18] Chaurasia G, Sorkine O, Drettakis G. Silhouette-aware warping for image-based rendering. In: Proceedings of the computer graphics forum, 30. Wiley Online Library; 2011. p. 1223–32.
[19] Furukawa Y, Hernández C. Multi-view stereo: a tutorial. Found Trends Comput Graph Vis 2015;9(1–2):1–148.
[20] Flynn J, Neulander I, Philbin J, Snavely N. Deepstereo: Learning to predict new views from the world's imagery. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 5515–24.
[21] Ji D, Kwon J, McFarland M, Savarese S. Deep view morphing. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2017. p. 7092–100.
[22] Tatarchenko M, Dosovitskiy A, Brox T. Single-view to multi-view: Reconstructing unseen views with a convolutional network. CoRR 2015;1(2):2. abs/1511.06702.
[23] Yang J, Reed SE, Yang M-H, Lee H. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In: Proceedings of the advances in neural information processing systems; 2015. p. 1099–107.
[24] Zhou T, Tulsiani S, Sun W, Malik J, Efros AA. View synthesis by appearance flow. In: Proceedings of the European conference on computer vision. Springer; 2016. p. 286–301.
[25] Hinton GE, Krizhevsky A, Wang SD. Transforming auto-encoders. In: Proceedings of the international conference on artificial neural networks. Springer; 2011. p. 44–51.
[26] Cheung B, Livezey JA, Bansal AK, Olshausen BA. Discovering hidden factors of variation in deep networks. arXiv preprint arXiv:1412.6583 2014.
[27] Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. In: Proceedings of the advances in neural information processing systems; 2015. p. 2017–25.
[28] Jayaraman D, Grauman K. Learning image representations tied to ego-motion. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1413–21.
[29] Zhu Z, Luo P, Wang X, Tang X. Multi-view perceptron: a deep model for learning face identity and view representations. In: Proceedings of the advances in neural information processing systems; 2014. p. 217–25.
[30] Yim J, Jung H, Yoo B, Choi C, Park D, Kim J. Rotating your face using multi–task deep neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 676–84.
[31] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved training of wasserstein gans. In: Proceedings of the advances in neural information processing systems; 2017. p. 5767–77.
[32] Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 2017.
[33] Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the international conference on machine learning, ICML, 30; 2013. p. 3.
[34] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 2004;60(2):91–110.
[35] Salimans T, Kingma DP. Weight normalization: a simple reparameterization to accelerate training of deep neural networks. In: Proceedings of the advances in neural information processing systems; 2016. p. 901–9.
[36] Bau D, Zhou B, Khosla A, Oliva A, Torralba A. Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 6541–9.
[37] Fong R, Vedaldi A. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 8730–8.
[38] Aubry M, Maturana D, Efros AA, Russell BC, Sivic J. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 3762–9.
[39] Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, et al. Shapenet: an information-rich 3d model repository. arXiv preprint arXiv:1512.03012 2015.
[40] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 2014.
[41] Mathieu M, Couprie C, LeCun Y. Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 2015.
[42] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP, et al. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.
[43] Maaten Lv d, Hinton G. Visualizing data using t-sne. J Mach Learn Res 2008;9(Nov):2579–605.
[44] Wu J, Zhang C, Xue T, Freeman B, Tenenbaum J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Proceedings of the advances in neural information processing systems; 2016. p. 82–90.
[45] Fuhrmann S, Langguth F, Goesele M. MVE – a multi-view reconstruction environment. In: Klein R, Santos P, editors. Proceedings of the Eurographics workshop on graphics and cultural heritage. The Eurographics Association; 2014. ISBN: 978-3-905674-63-7