HoloGAN: Unsupervised Learning of 3D Representations From Natural Images

Thu Nguyen-Phuoc [1] Chuan Li [2] Lucas Theis [3] Christian Richardt[1] Yong-Liang Yang [1]

[1] University of Bath [2] Lambda Labs [3] Twitter

Abstract

We propose a novel generative adversarial network (GAN) for the task of unsupervised learning of 3D representations from natural images. Most generative models rely on 2D kernels to generate images and make few assumptions about the 3D world. These models therefore tend to create blurry images or artefacts in tasks that require a strong 3D understanding, such as novel-view synthesis. HoloGAN instead learns a 3D representation of the world, and to render this representation in a realistic manner. Unlike other GANs, HoloGAN provides explicit control over the pose of generated objects through rigid-body transformations of the learnt 3D features. Our experiments show that using explicit 3D features enables HoloGAN to disentangle 3D pose and identity, which is further decomposed into shape and appearance, while still being able to generate images with similar or higher visual quality than other generative models. HoloGAN can be trained end-to-end from unlabelled 2D images only. Particularly, we do not require pose labels, 3D shapes, or multiple views of the same objects. This shows that HoloGAN is the first generative model that learns 3D representations from natural images in an entirely unsupervised manner.

1. Introduction

Learning to understand the relationship between 3D objects and 2D images is an important topic in computer vision and computer graphics. In computer vision, it has applications in fields such as robotics, autonomous vehicles or security. In computer graphics, it benefits applications in both content generation and manipulation. This ranges from photorealistic rendering of 3D scenes or sketch-based 3D modelling, to novel-view synthesis or relighting.

Recent generative image models, in particular, generative adversarial networks (GANs), have achieved impressive results in generating images of high resolution and visual quality [3, 11, 12] while their conditional versions have achieved great progress in image-to-image translation [9, 22] or image editing [5, 26]. However, GANs are still fairly limited in their applications, since they do not allow explicit control over attributes in the generated images, while conditional GANs need labels during training (Figure 1 left), which are not always available.

We propose HoloGAN, an unsupervised generative image model that learns representations of 3D objects that are not only explicit in 3D but also semantically expressive. In this work, we focus on designing a novel architecture that allows unsupervised learning of 3D representations from images, enabling direct manipulation of view, shape, and appearance in generative image models. Such representations can be learnt directly from unlabelled natural images, without any supervision of poses, 3D shapes, multiple views of objects, or geometry priors such as symmetry and smoothness over the 3D representation that are common in this line of work [1, 10]. Unlike other GAN models, HoloGAN employs both 3D and 2D features for generating images. HoloGAN first learns a 3D representation, which is then transformed to a target pose, projected to 2D features, and rendered to generate the final images (Figure 1 right). Different from recent work that employs hand-crafted differentiable renderers [13, 15, 17], HoloGAN learns perspective projection and rendering of 3D features from scratch using a projection unit [20]. This novel architecture enables HoloGAN to learn 3D representations directly from natural images for which there are no good hand-crafted differentiable renderers. To generate new views of the same scene, we directly apply 3D rigid-body transformations to the learnt 3D features, and visualise the results using the neural renderer that is jointly trained. This has been shown to produce sharper results than performing 3D transformations in high-dimensional latent vector space [20]. In summary, our main technical contributions are: **1)** A novel architecture that combines a strong inductive bias about the 3D world with deep generative models to learn disentangled representations (pose, shape, and appearance) of 3D objects from images. The representation is explicit in 3D and expressive in semantics. **2)** An unconditional GAN that, for the first time, allows native support for view manipulation **3)** An unsupervised training approach that enables disentangled representation learning without using labels.

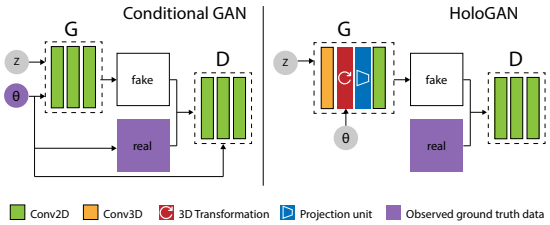

Figure 1. Comparison of generative image models. Data given to the discriminator are coloured purple. **Left:** In conditional GANs, the pose is observed and the discriminator is given access to this information. **Right:** HoloGAN does not require pose labels and the discriminator is not given access to pose information.

2. Method

To learn 3D representations from 2D images without labels, HoloGAN extends traditional unconditional GANs by introducing a strong inductive bias about the 3D world into the generator network. Specifically, HoloGAN generates images by learning a 3D representation of the world and to render it realistically such that it fools discriminator. View manipulation therefore can be achieved by directly applying 3D rigid-body transformations to the learnt 3D features. In other words, the images created by the generator are a *view-dependent* mapping from a learnt 3D representation to the 2D image space. This is different from other GANs which learn to map a noise vector $\mathbf{z}$ directly to 2D features to generate images.

Figure 2 illustrates the generator architecture of HoloGAN: HoloGAN first learns a 3D representation (assumed to be in a canonical pose) using 3D convolutions (Section 2.1), transforms this representation to a certain pose, projects and computes visibility using the projection unit (Section 2.2), and computes shaded colour values for each pixel in the final images with 2D convolutions. HoloGAN shares many rendering insights with RenderNet [20], but works with natural images, and needs neither pre-training of the neural renderer nor paired 3D shape–2D image training data.

2.1. Learning 3D representations

HoloGAN generates 3D representations from a learnt constant tensor (see Figure 2). The random noise vector $\mathbf{z}$ instead is treated as a "style" controller, and mapped to affine parameters for adaptive instance normalization (AdaIN) [8] after each convolution using a multilayer perceptron (MLP) $f:\mathbf{z}\rightarrow\gamma(\mathbf{z}),\sigma(\mathbf{z})$.

Given some features $\mathbf{\Phi}_l$ at layer l of an image $\mathbf{x}$ and the noise "style" vector $\mathbf{z}$, AdaIN is defined as:

$$\text{AdaIN}(\mathbf{\Phi}_l(\mathbf{x}),\mathbf{z})=\sigma(\mathbf{z})\left(\frac{\mathbf{\Phi}_l(\mathbf{x})-\mu(\mathbf{\Phi}_l(\mathbf{x}))}{\sigma(\mathbf{\Phi}_l(\mathbf{x}))}\right)+\gamma(\mathbf{z}). \quad (1)$$

This can be viewed as generating images by transforming a template (the learnt constant tensor) using AdaIN to match the mean and standard deviation of the features at different levels l (which are believed to describe the image "style") of the training images. Empirically, we find this network architecture can disentangle pose and identity much better than those that feed the noise vector $\mathbf{z}$ directly to the first layer of the generator.

HoloGAN inherits this style-based strategy from StyleGAN [12] but is different in two important aspects. Firstly, HoloGAN learns 3D features from a learnt 4D constant tensor (size $4\times4\times4\times512$, where the last dimension is the feature channel) before projecting them to 2D features to generate images, while StyleGAN only learns 2D features. Secondly, HoloGAN learns a disentangled representation by combining 3D features with rigid-body transformations during training, while StyleGAN injects independent random noise into each convolution. StyleGAN, as a result, learns to separate 2D features into different levels of detail, depending on the feature resolution, from coarse (e.g., pose, identity) to more fine-grained details (e.g., hair, freckles). We observe a similar separation in HoloGAN. However, HoloGAN further separates pose (controlled by the 3D transformation), shape (controlled by 3D features), and appearance (controlled by 2D features).

2.2. Learning with view-dependent mappings

In addition to adopting 3D convolutions to learn 3D features, during training, we introduce more bias about the 3D world by transforming these learnt features to random poses before projecting them to 2D images. This random pose transformation is crucial to guarantee that HoloGAN learns a 3D representation that is disentangled and can be rendered from all possible views as also observed by Tran et al. [23] in DR-GAN.

Rigid-body transformation We assume a virtual pin-hole camera that is in the canonical pose (axis-aligned and placed along the negative z-axis) relative to the 3D features being rendered. We parameterise the rigid-body transformation by 3D rotation, scaling followed by trilinear resampling. Assuming the up-vector of the object coordinate system is the global y-axis, rotation comprises rotation around the y-axis (azimuth) and the x-axis (elevation).

Projection unit In order to learn meaningful 3D representations from just 2D images, HoloGAN learns a differentiable projection unit [20] that reasons over occlusion. The projection unit is composed of a reshaping layer that concatenates the channel dimension with the depth dimension, thus reducing the tensor dimension from 4D ($W\times H\times D\times C$) to 3D ($W\times H\times(D\cdot C)$), and an MLP with a non-linear activation function to learn occlusion.

2.3. Loss functions

Identity regulariser To generate images at higher resolution (128×128 pixels), we find it beneficial to add an identity regulariser L_{identity} that ensures a vector reconstructed from a generated image matches the latent vector $\mathbf{z}$ used in the generator G. We find that this encourages HoloGAN to only use $\mathbf{z}$ for the identity to maintain the object's identity when poses are varied, helping the model learn the full variation of poses in the dataset. We introduce an encoder network F that shares the majority of the convolution layers of the discriminator, but uses an additional fully-connected layer to predict the reconstructed latent vector. The identity loss is:

$$L_{\text{identity}}(\text{G})=\mathbb{E}_{\mathbf{z}}\|\mathbf{z}-\text{F}(\text{G}(\mathbf{z}))\|^2. \quad (2)$$

Style discriminator Our generator is designed to match the "style" of the training images at different levels, which effectively controls image attributes at different scales. Therefore, in addition to the image discriminator that classifies images as real or fake, we propose multi-scale *style discriminators* that perform the same task but at the feature level. In particular, the style discriminator tries to classify the mean $\mu(\mathbf{\Phi}_l)$ and standard deviation $\sigma(\mathbf{\Phi}_l)$, which describe the image "style" [8]. Empirically, the multi-scale style discriminator helps prevent mode collapse and enables longer training. Given a style discriminator $\text{D}_l(\mathbf{x})=\widetilde{\text{D}}_l(\mu(\mathbf{\Phi}_l(\mathbf{x})),\sigma(\mathbf{\Phi}_l(\mathbf{x}))$ for layer l, the style loss is defined as:

$$L^l_{\text{style}}(\text{G})=\mathbb{E}_{\mathbf{z}}[-\log\text{D}_l(\text{G}(\mathbf{z}))]. \quad (3)$$

The total loss can be written as:

$$L_{\text{total}}(\text{G})=L_{\text{GAN}}(\text{G})+\lambda_{\text{i}}\cdot L_{\text{identity}}(\text{G})+\lambda_{\text{s}}\cdot\sum_l L^l_{\text{style}}(\text{G}). \quad (4)$$

We use $\lambda_{\text{i}}=\lambda_{\text{s}}=1.0$ for all experiments. We use the GAN loss from DC-GAN [21] for L_{GAN}.

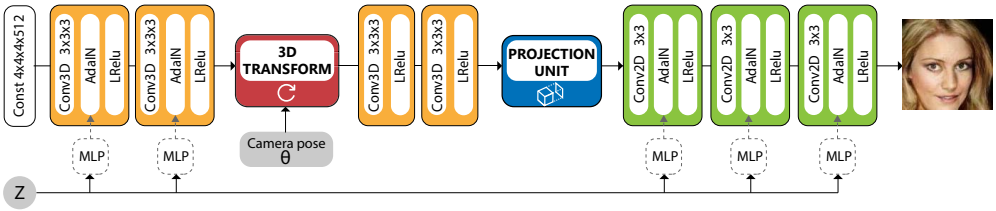

Figure 2. HoloGAN's generator network: we employ 3D convolutions, 3D rigid-body transformations, the projection unit and 2D convolutions. We also remove the traditional input layer from $\mathbf{z}$, and start from a learnt constant 4D tensor. The latent vector $\mathbf{z}$ is instead fed through MLPs to map to the affine transformation parameters for adaptive instance normalisation (AdaIN). Inputs are coloured gray. Best viewed in colour.

3. Results

Data We train HoloGAN using a variety of datasets: CelebA [16], Cats [27], Chairs [4], Cars [24], and LSUN bedroom [25]. We train HoloGAN on resolutions of 64×64 pixels for Cats and Chairs, and 128×128 pixels for CelebA, Cars and Bedroom.

Implementation details We use adaptive instance normalization [8] for the generator, and spectral normalization [19] for the discriminator. We train HoloGAN from scratch using the Adam solver [14]. During training, we sample $\mathbf{z} \sim \mathcal{U}(-1,1)$, and also sample random poses from a uniform distribution We use $|\mathbf{z}| = 200$ for Cars at 128×128, and $|\mathbf{z}| = 200$ for the rest.

Qualitative evaluation Figures 3 and 4 show that HoloGAN can smoothly vary the pose along azimuth and elevation while keeping the same identities for multiple different datasets. Note that the LSUN dataset contains a variety of complex layouts of multiple objects. This makes it a very challenging dataset for learning to disentangle pose from object identity.

Quantitative results To evaluate the visual fidelity of generated images, we use the Kernel Inception Distance (KID) by Bińkowski et al. [2]. The lower the KID score, the better the visual quality of generated images. We compare HoloGAN with other recent GAN models: DCGAN [21], LSGAN [18], and WGAN-GP [6], on 3 datasets in Table 1. Note that KID does not take into account feature disentanglement, which is one of the main contributions of HoloGAN. We use a publicly available implementation[1] and use the same hyper-parameters (that were tuned for CelebA) provided with this implementation for all three datasets. Similarly, for HoloGAN, we use the same network architecture and hyper-parameters [2] for all three datasets. We sample 20,000 images from each model to calculate the KID scores shown below.

Table 1 shows that HoloGAN can generate images with competitive (for CelebA) or even better KID scores on more challenging datasets: Chairs, which has high intra-class variability, and Cars, which has complex backgrounds and lighting conditions. This also shows that HoloGAN architecture is more robust and can consistently produce images with high visual fidelity across different datasets with the same set of hyper-parameters (except for azimuth ranges). More importantly, HoloGAN learns a disentangled representation that allows manipulations of the generated images.

[1] https://github.com/LynnHo/DCGAN-LSGAN-WGAN-WGAN-GP-Tensorflow

[2] Except for ranges for sampling the azimuth: 100° for CelebA since face images are only taken from frontal views, and 360° for Chairs and Cars

Method	CelebA 64×64	Chairs 64×64	Cars 64×64
DCGAN [21]	1.81 ± 0.09	6.36 ± 0.16	4.78 ± 0.11
LSGAN [18]	1.77 ± 0.06	6.72 ± 0.19	4.99 ± 0.13
WGAN-GP [6]	**1.63 ± 0.09**	9.43 ± 0.24	15.57 ± 0.29
HoloGAN (ours)	2.87 ± 0.09	**1.54 ± 0.07**	**2.16 ± 0.09**

Table 1. KID [7] between real images and images generated by HoloGAN and other 2D-based GANs (lower is better). We report KID mean×100 ± std.×100.

Disentangling shape and appearance We show that apart from pose, HoloGAN also learns to further divide identity into shape and appearance. We sample two latent codes, $\mathbf{z}_1$ and $\mathbf{z}_2$, and feed them through HoloGAN. While $\mathbf{z}_1$ controls the 3D features (before perspective morphing and projection), $\mathbf{z}_2$ controls the 2D features (after projection). Figure 5 shows the generated images with the same pose, same $\mathbf{z}_1$, but with a different $\mathbf{z}_2$ at each row. As can be seen, while the 3D features control objects' shapes, the 2D features control appearance (colour and lighting). This shows HoloGAN learns to separate shape from appearance directly from unlabelled images, allowing manipulation of these factors.

4. Discussion

HoloGAN's ability to separate pose from identity depends on the variety and distribution of poses included in the training dataset. Currently, during training, we sample random poses from a uniform distribution. Future work therefore can explore learning the distribution of poses from the training data in an unsupervised manner to account for uneven pose distributions. Finally, it will be interesting to explore further disentanglement of objects' appearances, such as texture and illumination.

References

[1] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, 2015. 1

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 1

[4] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. arXiv:1512.03012, 2015. 3

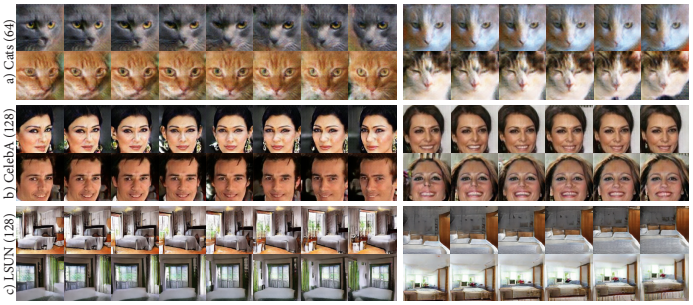

Figure 3. HoloGAN supports changes in both azimuth (range: 100°) and elevation (range: 35°). However, the available range depends on the dataset. For CelebA, for example, few photos in the dataset were taken from above or below.

Figure 4. For the car dataset, HoloGAN fully captures the full 360° azimuth and elevation (range: 35°).

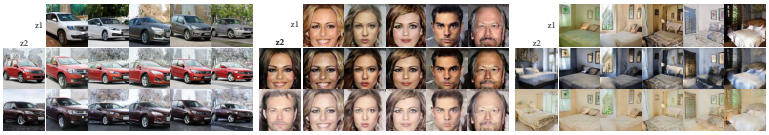

Figure 5. Combinations of different latent vectors $\mathbf{z}_1$ (for 3D features) and $\mathbf{z}_2$ (for 2D features). While $\mathbf{z}_1$ influences objects' shapes, $\mathbf{z}_2$ determines appearance (texture and lighting). Best viewed in colour.

[5] Garoe Dorta, Sara Vicente, Neill D. F. Campbell, and Ivor Simpson. The GAN that warped: Semantic attribute editing with unpaired data. arXiv:1811.12784, 2018. 1

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of Wasserstein GANs. In *NIPS*, pages 5767–5777, 2017. 3

[7] 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*, pages 6626–6637, 2017. 3

[8] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2, 3

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017. 1

[10] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 1

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 1

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2

[13] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, 2018. 1

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[15] Tzu-Mao Li, Miika Aittala, Fredo Duran, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphic*, 37(6):162:1–11, 2018. 1

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 3

[17] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *ECCV*, pages 154–169, 2014. 1

[18] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 3

[19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 3

[20] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes. In *NeurIPS*, pages 7902–7912, 2018. 1, 2

[21] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 3

[22] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017. 1

[23] Luan Tran, Xi Yin, and Xiaoming Liu. Representation learning by rotating your faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[24] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981, 2015. 3

[25] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv:1506.03365, 2015. 3

[26] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *ECCV*, 2018. 1

[27] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection – how to effectively exploit shape and texture features. In *ECCV*, 2008. 3