3DGEN: A GAN-based approach for generating novel 3D models from image data

Antoine Schnepf
a.schnepf@criteo.com
Criteo AI Lab

Flavian Vasile
f.vasile@criteo.com
Criteo AI Lab

Ugo Tanielian
u.tanielian@criteo.com
Criteo AI Lab

Abstract

The recent advances in text and image synthesis show a great promise for the future of generative models in creative fields. However, a less explored area is the one of 3D model generation, with a lot of potential applications to game design, video production, and physical product design. In our paper, we present 3DGEN, a model that leverages the recent work on both Neural Radiance Fields for object reconstruction and GAN-based image generation. We show that the proposed architecture can generate plausible meshes for objects of the same category as the training images and compare the resulting meshes with the state-of-the-art baselines, leading to visible uplifts in generation quality.

1 Introduction

Generative models such as StableDiffusion (Rombach et al., 2022) or DALLE2 (Ramesh et al., 2022) are rapidly changing the boundaries of machine-assisted creativity, especially in the case of image synthesis. Researchers and practitioners are inventing new ways to create and remix art, either by text-conditioned image generation, image inpainting or outpainting, full video generation. In the same time, the class of Neural Radiance Fields models (NeRF, Mildenhall et al., 2021) are making rapid advances in photorealistic 3D model/scene reconstruction from partial views. NeRF uses an implicit volumetric representation to represent 3D scenes, making it possible to render them at an arbitrary resolution with low memory costs.

Some of the existing work, such as Generative Radiance Fields (GRAF, Schwarz et al., 2020), has been starting to bridge the gap between reconstruction and generation. GRAF can generate new volumetric models from a set of views of similar objects. However, a major limitation of the GRAF model is that this volumetric representation is not adapted to produce plausible object meshes, and is therefore not a good match for 3D-native creative environments such as game design, virtual-reality (VR) world design, animation. On the other hand, UNISURF (Oechsle et al., 2021) showed that radiance fields and implicit surface representations can be unified, and proposed a joint optimization task that both improves NERF and allows to extract 3D meshes.

In this paper, we propose a potential solution for the shortcomings of GRAF, which we name 3DGEN. This solution builds on both GRAF (Schwarz et al., 2020) and UNISURF (Oechsle et al., 2021), and can generate volumetric objects with a corresponding implicit surface, hence making them easily exportable to 3D meshes. In Figure 2, we showcase one potential way to control the object generation: the interpolation in the latent space between two existing object meshes leads to a set of plausible object meshes of the same type (in this case cars and chairs).

Preprint.

Figure 1: Left: 3DGEN with cars and chairs. Right: disentanglement of shape and appearance.

Figure 2: Latent space interpolation and the corresponding mesh extraction.

2 Our approach

We begin by introducing a conditional version of UNISURF g_θ which encodes an object conditionally to a shape code and an appearance code, in a disentangled manner. Similarly to GRAF (Schwarz et al., 2020), we construct a generator G_θ (sharing parameters with g_θ) by stacking (i) a module that casts rays, (ii) g_θ that conditionally computes the emitted radiance and occupancy probabilities along the casted rays, and (iii) a differentiable volumetric renderer to produce the output images (Mildenhall et al., 2021). We introduce the discriminator D_ϕ, a convolutionnal neural network.

We train this setup with the non-saturating GAN objective (Goodfellow et al., 2014) with R1-regularization (Mescheder et al., 2018), to which a smoothing term for the implicit surface is added (Oechsle et al., 2021) (more details on each term can be found in Appendix):

$$\min_\theta \max_\phi \Big(\mathcal{L}_{\text{adv}}(\theta, \phi) - \lambda \mathcal{R}_1(\phi) + \gamma \mathcal{L}_{\text{smooth}}(\theta) \Big) \tag{1}$$

Implementation details and surface extraction. Our model is initialized such that the initial implicit surfaces are spheres (Oechsle et al. (2021), Gropp et al. (2020)). As the training progresses, the points are sampled along rays within a narrowing interval centered around the first intersection with the implicit surface (Oechsle et al., 2021). By doing so, in the early stages of the training, the formulation is similar to GRAF, while in the later stages of the training, points are sampled close to the implicit surface. It is therefore possible to extract a well defined surface with the Marching Cube algorithm (Lorensen and Cline, 1987).

Experiments. We test our model on (i) a dataset of cars rendered from The Carla Driving simulator (Dosovitskiy et al., 2017) and (ii) a dataset of chairs, rendered from Photoshapes (Park et al., 2018). Figure 1 shows the generated cars and chairs, as well as disentanglement of shape and appearance. Figure 2 shows latent space interpolation and the exportation to meshes. In the Appendix, 3D-GEN is compared to the baseline GRAF for camera poses interpolations in Figure 3 and meshes extraction in Figure 4. To evaluate both methods, we report the Frechet Inception Distance (FID, Heusel et al., 2017). GRAF / Ours: $71/97$ (Cars); $48/126$ (Chairs).

3 Conclusion and future work

This work present 3D-GEN, a generative model that unifies radiance fields and implicit surfaces. The model can learn an underlying distribution of radiance fields and surfaces from a dataset composed only of 2D images of objects of a similar class, and therefore *generate new objects* from this class. During inference, the model can both render views from any angle and easily export to a mesh based representation, which makes it applicable to 3D content creation. In our future work we intend to further improve the quality of the generated objects by reducing artefacts in the shapes and producing more diverse outputs and to compare it with the recent GET3D model proposed in Gao et al. (2022).

References

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. URL https://arxiv.org/abs/1711.03938.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes, 2020. URL https://arxiv.org/abs/2002.10099.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? 2018. doi: 10.48550/ARXIV.1801.04406. URL https://arxiv.org/abs/1801.04406.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021.

Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. PhotoShape. *ACM Transactions on Graphics*, 37(6):1–12, dec 2018. doi: 10.1145/3272127.3275066. URL https://doi.org/10.1145%2F3272127.3275066.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33: 20154–20166, 2020.

4 Appendix

4.1 Background

This section summarizes the mathematical formalism required for our proposed model 3DGEN.

Neural Radiance Field (Mildenhall et al., 2021). Neural radiance fields are neural networks parameterized by $\theta \in \Theta$ mapping a spatial location $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d} \in \mathbb{R}^3$ to a view-dependant radiance $c_\theta(\mathbf{x}, \mathbf{d})$ and a volumetric opacity $\sigma_\theta(\mathbf{x})$:

$$f_\theta : \mathbf{x}, \mathbf{d} \longrightarrow c_\theta(\mathbf{x}, \mathbf{d}), \sigma_\theta(\mathbf{x}) \tag{2}$$

Differentiable volume rendering. Given the camera origin $\mathbf{o} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d}$, we can shot a ray $\mathbf{r} = \{\mathbf{o} + t\mathbf{d} | t \in \mathbb{R}_+\}$ through the radiance field. Given N points $\{\mathbf{x}_i\}$ sampled along this ray, differentiable volume rendering approximates the perceived color of the ray as follows:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i c_\theta(\mathbf{x}_i, \mathbf{d}) \tag{3}$$

$$\alpha_i = 1 - \exp(-\sigma_\theta(\mathbf{x}_i)\delta_i)) \tag{4}$$

$$T_i = \prod_{j<i} (1 - \alpha_i) \tag{5}$$

where T_i is the accumulated transmittance along the ray $\mathbf{r}$ and $\delta_i = ||\mathbf{x}_{i+1} - \mathbf{x}_i||_2$ is the distance between two adjacent points.

UNISURF (Oechsle et al., 2021). Assuming solid non-transparent objects, i.e. $\sigma_\theta(\mathbf{x}_i) \in \{0, +\infty\}$, α_i can be reinterpreted as the occupancy probability at position $\mathbf{x}_i$ from equation 4.

We can therefore derive an implicit surface $\mathcal{S}_\theta$:

$$\mathcal{S}_\theta = \{\mathbf{x} \in \mathbb{R}^3 | \alpha_\theta(\mathbf{x}) = 0.5\} \tag{6}$$

A regularization loss $\mathcal{L}_{\text{smooth}}$ on the implicit surface is introduced:

$$\mathcal{L}_{\text{smooth}}(\theta) = \sum_{\mathbf{x}_s \in \mathcal{S}_\theta} \left\| \mathbf{n}_\theta\left(\mathbf{x}_s\right) - \mathbf{n}_\theta\left(\mathbf{x}_s + \boldsymbol{\epsilon}\right) \right\|_2 \tag{7}$$

Here $\boldsymbol{\epsilon}$ is a small perturbation and $\mathbf{n}(\mathbf{x}_s)$ denotes the surface normal at position $\mathbf{x}_s$. The surface normal can be computed using the formula:

$$\mathbf{n}_\theta(\mathbf{x}_s) = \frac{\nabla \alpha_\theta(\mathbf{x}_s)}{\|\nabla \alpha_\theta(\mathbf{x}_s)\|_2} \tag{8}$$

GRAF (Schwarz et al., 2020). Built upon the original NeRF formulation, a conditional neural radiance field (cNeRF) generates radiance fields conditionally to an appearance code z_a and a shape code z_s:

$$g_\theta : \mathbf{x}, \mathbf{d}, \mathbf{z_s}, \mathbf{z_a} \longrightarrow c_\theta(\mathbf{x}, \mathbf{d}, \mathbf{z_s}, \mathbf{z_a}), \sigma_\theta(\mathbf{x}, \mathbf{z_s}) \tag{9}$$

The generator G_θ is composed of three modules:

(i) a module that sample the camera parameters $\boldsymbol{\xi}$ and subsequently cast $K \times K$ rays

(ii) a cNeRF g_θ to conditionally compute radiances and volumetric opacities along the sampled rays

(iii) a volume renderer to produce the output image

To train the discriminator D_ϕ and the generator G_θ, the adversarial loss $\mathcal{L}_{\text{adv}}(\theta, \phi)$ and the regularization term $\mathcal{R}_1(\phi)$ are introduced:

$$\mathcal{L}_{\text{adv}}(\theta, \phi) = \mathbb{E}_{\mathbf{I} \sim p_{\text{data}}, \boldsymbol{\nu} \sim p_{\text{patch}}} \Big[f\left(-D_\phi(\Gamma(\mathbf{I}, \boldsymbol{\nu}))\right) \Big] + \mathbb{E}_{\mathbf{z_s}, \mathbf{z_a} \sim p_{\text{latent}}, \boldsymbol{\xi} \sim p_{\text{cam}}, \boldsymbol{\nu} \sim p_{\text{patch}}} \Big[f\left(D_\phi(G_\theta(\mathbf{z_s}, \mathbf{z_a}, \boldsymbol{\xi}, \boldsymbol{\nu}))\right) \Big] \quad (10)$$

$$\mathcal{R}_1(\phi) = \mathbb{E}_{\mathbf{I} \sim p_{\text{data}}, \boldsymbol{\nu} \sim p_{\text{patch}}} \Big[\big|\big| \nabla D_\phi\left(\Gamma(\mathbf{I}, \boldsymbol{\nu})\right) \big|\big|_2^2 \Big] \quad (11)$$

with $f(x) = -\log(1 + \exp(-x))$. Here, $\boldsymbol{\nu}$ denotes denotes the patching strategy and Γ the patching operator to transform the training images into $K \times K$ patches. Note that during inference, the generator G_θ can be used to generate images at any resolution. The patch constraint only holds during training, to make this setup trainable in practice. For more details, we refer the reader to Schwarz et al. (2020).

4.2 Experimental comparison between 3DGEN and GRAF

Figure 3: Camera poses interpolations on both Cars and Chairs: varying rotation (up), varying elevation (down). 3D-GEN (left) and GRAF (right).

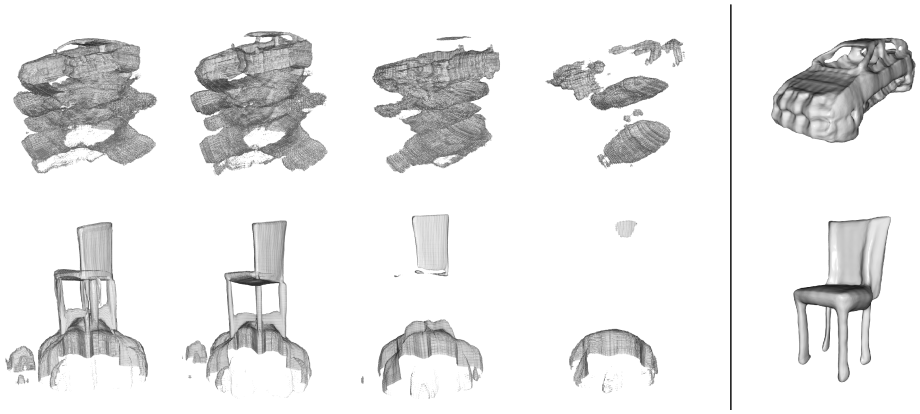

Figure 4: Extracting surface meshes from GRAF at level set $\sigma \in \{1, 10, 50, 100\}$ (left) and from 3D-GEN (right).