The Use of Progressive Generative Adversarial Network Structures for 3D Face Generation

Sebastian Oliver
University of Cape Town
Cape Town, South Africa
olvseb001@myuct.ac.za

ABSTRACT

Generative adversarial networks(GANs) have been used with great success in 2D face generation and are starting to creep into 3D face generation more and more as more 3D datasets become available to train the neural networks(NNs). This literature review looks into the techniques used by researchers when addressing the problem of face generation using GANs. The uniqueness of each individual face makes it an interesting problem that GANs can solve. Face generation splits into 2D and 3D, with the interest in this review being in 3D. The types of GANs used to generate models include: conditional, controllable and progressive. In this review the focus is on progressive GANs (ProGANs), what makes them different is that they learn from the bottom-up. An implementation for a 3D ProGAN for 3D face generation does not exist and this poses an interesting opportunity in an ever-growing field that is machine learning.

1 INTRODUCTION

Generative Adversarial Networks consist of two neural networks which compete against each other. One of the NNs is known as the generator and the other is the discriminator. First proposed by Goodfellow et al.[11], the aim of the generator is to trick the discriminator into thinking that its generated models are real. The discriminator provides feedback that the generator uses to improve. This relationship between the two players is best described by the counterfeiting example, where the generator is attempting to pass its own fake money as real money to the discriminator. They reach equilibrium once the discriminator can no longer tell which money is fake and which is real [10, 11]. Progressive GANs involve learning how to do a concept using a bottom-up approach of getting the basics dialled first.

Face generation involves generating new faces from existing data. The quality of the generated faces depends on both the type of dataset used, the method used for face generation and the hardware available. Applications in which the results of face generation techniques can be used are vast, ranging from virtual reality and augmented reality to movies and computer games.

GANs have been used for face generation as they are able to generate output from random noise. They are able to generate both 2D and 3D faces but the progress being made by GANs in this area is very much limited to the datasets available. It is hard to predict just how good GANs would be able to get when larger datasets do become available as they have already produced pretty remarkable results as in StyleGAN3 by Karras et al. [18].

This literature review will take a look at how far recent researches have been able to take face generation using GANs. An emphasis will be on using ProGANs and 3D face generation.

2 GENERATIVE ADVERSARIAL NETWORKS

The two networks compete against each other until they reach a condition of Nash equilibrium, which results in the GAN converging. When this convergence condition is met it means that the discriminator is unable to tell whether the data being input to it is real or fake (from the generator). Meaning the generator is able to generate the targeted concept to a realistic degree.

2.1 Basic Structure

As mentioned earlier, it consists of two competing NNs, both are trained simultaneously on the same data but have different functions. A simplified overview of a structure of a GAN can be seen in figure 1.

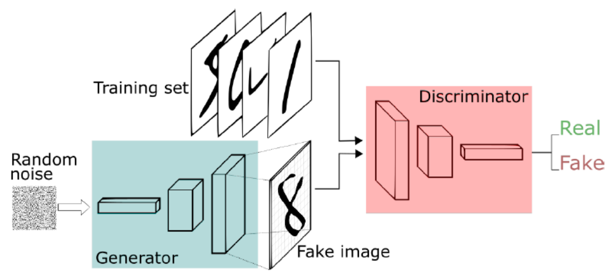

Figure 1: Overview of a GAN [3].

Generally, the discriminator is of main interest as the discriminator is usually discarded once the generator is trained. GANs learn through the use of a minimax value function, where V is the value function whose inputs are G, the generator, and D the discriminator [10, 11]. The equation for this loss function can be seen in equation 1.

$$L_{adv} = \min_{G} \max_{D} \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \tag{1}$$

GANs don't have to exclusively learn through this minimax value function. These types of functions are known as loss functions and there are quite a few to choose from, depending on the application. A popular improvement made to the original GAN was the introduction of the Wasserstein GAN by Arjovsky et al. [2], where they cured the common training problems that the original GAN has. Such as more stable training, getting rid of mode collapse and additional debugging functionality. This was mainly done through the use of an alternative loss function deemed the WGAN loss function, which can be seen in equation 2.

$$L_{WGAN} = \inf_{\gamma \epsilon \Pi_{(P_{(r)}, P_{(g)})}} \mathbb{E}_{(x_r, x_g) \sim \gamma}[x_r - x_g] \tag{2}$$

Other loss functions include WGAN-GP, cGAN, pixel-wise, feature, total variation, patch-wise, symmetry, identity preservation, and cycle consistency. Out of all those, the one most prevalent to this paper will be the WGAN-GP which can be seen in equation 3, this is due to it being used by Karras et al. [17].

$$L_{WGAN-GP} = \mathbb{E}_{x_g \sim \mathbb{P}_g}[D(x_g)] - \mathbb{E}_{x_r \sim \mathbb{P}_r}[D(x_r)] + \lambda \mathbb{E}_{x_g \sim \mathbb{P}_{xg}}[(\| \nabla_{x_g} D(x_g) \|_2 \tag{3}$$

2.2 Types

There are three main types that GANs are typically categorised into and they are: conditional, controllable and progressive.

2.2.1 Conditional GANs. Involves adding an additional label to the noisy input data for the generator to aid training, as well as adding this label as input to the discriminator.

2.2.2 Controllable GANs. Involves tweaking the noisy input data, which has a significant impact on the output. A trained classifier is used to identify certain features in the generated image. Results in defining noise vectors for the different features.

2.2.3 Progressive GANs. Involves the generation learning continuously. It starts off with a low-resolution image and as it continues to learn, the image becomes of a higher resolution and the details become finer.

3 FACE GENERATION USING GANS

Out of all the different facial application techniques that GANs have been applied to, such as face inpainting, face deblurring and face generation, face generation has been the main focus. The past trend of which types of GANs have been used for face generation are as follows. Between 2018 and 2019 conditional GANs were commonly researched, while from 2019 up to the present controllable GANs are the more popular choice. The reasons for choosing recently controllable GANs have been due to the better quality output it produces as well as better convergence [14]. Facial GANs have evolved over time from uncontrollable grey-scale images to photo-realistic, high-resolution images with style control. The main improvement found in facial GANs has been the changes in the type of loss functions that are used. Below we will go through each of the different modelling techniques used for representing faces followed by a quick overview of both 2D and 3D face generation, with an emphasis on 3D face generation.

3.1 Modelling Techniques

There are various ways in which you are able to represent a 3D model. Below is an explanation of each of the types of the various models.

3.1.1 3D morphable models. A 3D morphable model(3DMM) was first introduced as a principal component analysis technique by Blanz and Vetter [5] in 1999 making it quite an old form of face representation. These days it is represented as a generated model that is based on two ideas. The first is that all faces are in the form of point to point correspondence, usually based on some example faces in a processing procedure and then kept for any further processing steps. The second is to separate facial shape and colour, as well as remove these from external factors such as lighting and camera parameters. An overview of how a 3DMM works can be seen in figure 2.

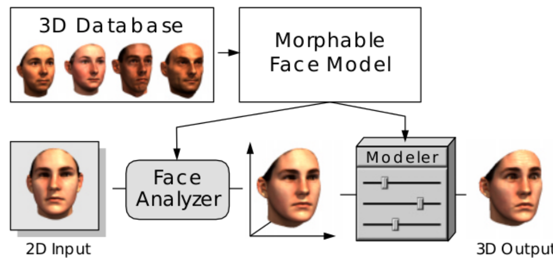

Figure 2: Overview of a 3DMM [5].

3.1.2 Meshes. A mesh involves dividing up an image into connecting polygons. These interconnected polygons form a 3D model of an object. The most commonly used mesh is the 3D triangle mesh.

3.1.3 Point clouds. Are an unordered set of 3D points, which may have additional information such as the RGB of each point [23]. Each point will at least have an (x, y, z) value representing its position in 3D Cartesian space.

3.1.4 Voxels. Voxels are pretty much a 3D version of a pixel [15]. A collection of voxels connect together to form a 3D model. An easy way to think of a voxel is a lot of connected cubes.

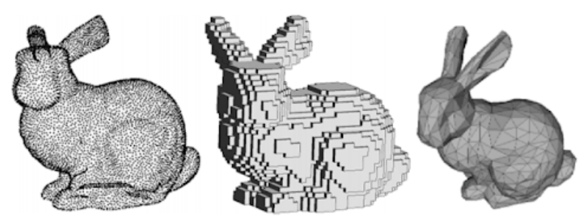

Figure 3: 3D data forms of the Stanford bunny model: Point clouds(on the left), voxel(in the middle) and 3D triangle mesh(on the right)[13].

3.2 2D Generation

The more explored area compared to its 3D counterpart. 2D face generation falls into two categories depending on the type of data set you to use. They are static and dynamic, static being if your data set involves images and dynamic if your data set is a motion picture/video. The most successful GAN in the field of 2D face generation has to be StyleGAN3 by Karras et al [18]. In the following subsections, we will see the different kinds of techniques that are used in 2D face generation.

3.2.1 *Image super-resolution.*
Involves generating face images that are of a high resolution, and uses a static data set. Progressive GANs are a popular choice for this application due to being able to increase the resolution.

3.2.2 *Image-to-image translation.*
Involves translating images from one style to another, depending on the style specified, using a static data set. Conditional GANs would be a good choice for the application due to the additional input specifier.

3.2.3 *Image matting.*
Determines foreground object(s) in images and videos, and uses a static data set. Controllable GANs would apply here as it is relatively easy to change the output as you would alter its latent space.

3.2.4 *Face completion.*
Filling in or repairing the missing parts of a face, uses a static data set.

3.2.5 *Face reenactment.*
Involves altering the source face shape to a target face shape while preserving the identity and appearance of the target face, using a dynamic data set.

3.3 3D Generation

The more challenging type of generation has two main categories under which pieces of research fall, depending on the kind of input they are using. They are 3D face generation from 2D images and 3D face generation from 3D models. The latter is the less explored option of the two, due to the higher computation and data set constraints. Below is a summary of a few pieces of research that used different kinds of input(2D or 3D).

3.3.1 *3D face generation from 2D images.*
One of the main deal-breakers of using 2D data sets as input is that they miss out on some of the detail that a 3D data set would have, leading to unrealistic parts of the created 3D facial model.
HoloGAN by Nguyen-Phouc et al. [21] was the first generative model to learn 3D representations from natural images using unsupervised learning. This study was not exclusive to face generation, all kinds of objects were used and tested. The success of HoloGAN depends on the variability of the types of poses in the used dataset.
GANFit by Gecer et al. [9] uses GANs and deep convolutional NNs to generate the facial texture and shape from a single image. This is done by converting the single image to a UV map which is then fitted to a 3DMM through a novel technique based on GANs and a differentiable renderer [9]. According to this paper, [9] it is the first time a GAN has been used for model fitting and they produced great results.
AvatarMe by Lattas et al. [19] revolved around being able to produce a high-resolution photorealistic 3D face from an "in-the-wild" image. They implement this by building upon existing techniques, plus training image translation networks that are able to perform estimation of diffuse and specular albedo, as well as diffuse and specular normal. This paper proved that it is possible to produce rendering-ready faces from arbitrary face images [19].
TBGAN by Gecer et al. [8] involves multi-branch GAN that is able to preserve the correlation between three different modalities, shape, texture and normal. They did this through non-linear generative networks.

3.3.2 *3D face generation from 3D models.*
Using 3D models as input helps to generate more accurate results, although the computation required to process this input does lead to longer training times.
Decoupled 3D Face by Abrevya et al. [1] shows that using GANs contribute to better facial modelling performances, to a great degree in decoupling natural factors such as expression and identity. They do this by using a unique 3D-2D architecture, where the generator is 3D and the discriminator is 2D, which allows for generating 3D meshes as well as taking advantage of the discriminative power of convolutional NNs [1].
MeshGAN by Cheng et al. [6] is the first GAN that is able to generate 3D facial images of different identities and expressions. They do this by implementing the first GAN architecture to generate 3D meshes using convolutions directly on meshes [6]. The results indicate that MeshGAN is able to generate models with better details than GAN and auto-encoder architectures.
3DFaceGAN by Moschoglou et al. [20] proposed the first GAN tailored towards modelling the distribution of 3D facial surfaces. An important part of this was to keep all the high-level details as well. They did this through an auto-encoder like network structure for the GAN, which is able to produce realistic 3D facial data, while retaining all the high-level details [20].

4 PROGANS FOR FACE GENERATION

When it comes to ProGANs and face generation, there have not been many intersections into the third dimension. All of the recent literature I was able to find was on 2D face generation. This is both a good and a bad finding. Good as this mean the problem is unexplored but bad as it could not be explored for a reason.

4.1 The First ProGAN

As mentioned earlier, progressive GANs(ProGANs) learn by adding more and more layers to their architecture in order to produce an image with a higher resolution than before. A good way to think about how they learn is how we learn subjects like maths. We start out with basics such as addition and subtraction and slowly start building on these concepts to understand more difficult ones, such as the minimax function. ProGANs do the same as they start out by understanding how to generate the target output in a low resolution before progressing to a higher resolution. In order for this to happen successfully, the generator and discriminator have to be developed at the same rate, otherwise, the minimax game will become one-sided. They are mirror images of each other.
Karras et al. in [17] proposed the first ProGAN as we know it in 2017. In this paper, it was apparent that ProGANs are the way forward if you are wanting to generate output in a high resolution. Not only are they able to produce high-resolution results but their training time is a lot shorter than the other types of GANs(conditional and controllable) as most of their training iterations are done at lower resolutions, ending up with a 2-6 times reduction in training time depending on the resolution of the output [17]. The resolution of both the generator and discriminator are increased just before the layers converge. When the resolution is increased, there is only a little bit of new information to be learned by the networks. This progressive probing ensures that mode collapse does not occur. When adding in additional layers to increase the resolution, Karras

et al. [17] did this by slowly fading in the new line using the nearest neighbour technique.
Training tricks include minibatch standard deviation which allows the discriminator to look at the distribution of training images at the same time, not just a single image at a time [17]. Meaning it is able to see the variation in the images so it can penalize the generator if the generator's variation is not correct. Next is the equalized learning rate is exactly what it sounds like, it ensures that both networks learn at the same rate. Both networks are therefore updated at the same speed. Lastly, pixelwise feature vector normalization discourages the generator from generating images that are clearly broken. This in turn helps avoid training that spirals out of control, leading to mode collapse. Lastly, the primary loss function that they used was the improved Wasserstein's loss. Using both the CelebA-HQ dataset and the LSUN dataset they managed impressive results. They managed to output high-resolution images of people and objects that do not actually exist. They were slowly generated by their ProGAN.

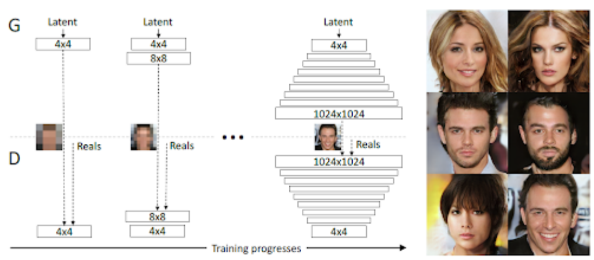

Figure 4: Overview of the training process of a ProGAN [17]

4.2 Latest in ProGANs

Since the first ProGAN was proposed not much progress has been made in terms of face generation, especially in 3D. Most of the research was about some aspects of face generation that were not specifically progressive but more seen as an alternative to a ProGAN.
MSG-GAN by Karnewar et al. [16], standing for multi-scale gradients for GANs, was proposed to eliminate the difficulty that GANs face when adapting to different datasets. A reason for this issue is due to the gradients shared between the generator and discriminator become uninformative. Their solution involves sharing the gradients between the generator and discriminator at multiple scales. The results prove that their technique outperforms the state of the art techniques in most cases, but not all.
Although no papers seemed to tackle the problem of a 3D ProGAN for face generation, EKlund et al. [7] did implement a 3D ProGAN for synthesizing brain volumes. This is how they implemented their 3D ProGAN. It was based on the original ProGAN by Karras et al . [17], although this is a 2D GAN all that needed to be done to convert it to a 3D were some tweaks here and there. Such as changing all the 2D convolutions with 3D convolutions and an added extra dimension to each of the calls to TensorFlow [7]. They did also lower the number of filters to 128 and had a lower learning rate than the original ProGAN. The results produced synthetic brains with a resolution of 64x64x64 voxels [7].

4.3 Implementation

From looking into what kinds of programming languages and libraries researchers used for their implementation. For the programming language, it was unanimous for Python, but for the library used to build their GANs, it was not so unanimous. The popular options include Keras, PyTorch, Theano and TensorFlow. Below is a brief description of each.

4.3.1 Keras. An open-source library for developing and evaluating NNs, which is able to run on top of existing frameworks like TensorFlow. The library is flexible, modular, user-friendly and portable. It is able to work on both CPUs and GPUs. With regards to NNs has objectives, layers, activation functions and optimizers within it. It does also have a vast array of data types available.

4.3.2 PyTorch. An open-source library based on the C framework Torch. Commonly used in machine learning and deep learning applications like computer vision and natural language processing. Advantages include high execution speeds (including for heavy loads), and the ability to run on simple processors in addition to CPUs and GPUs. It does also come with powerful APIs that enable a lot of extension-ability. As well as it is compatible with Python's IDE tools which make for easy debugging.

4.3.3 Theano. A numerical computation library made specifically for machine learning. Advantages include efficient definition, optimization and evaluation of mathematical expressions and matrix calculations. Exclusively used for machine learning and deep learning developers. Is able to integrate with NumPy. When run with a GPU it can perform data-intensive computations 140 times faster, plus it has built-in testing functionality.

4.3.4 TensorFlow. An open-source library that specializes in differentiable programming. Has flexible architecture and framework making it able to run on both CPUs and GPUs, however it does perform best when operating on a tensor processing unit. It is beginner-friendly and is not limited to working on desktops, it is able to work on smartphones too.
Weighing up all of the different positives and negatives of each of the machine learning libraries, the decision has been made to use PyTorch as high execution speeds are of utmost importance, especially when working with three-dimensional data.
Both the codebases for ProGAN implemented by Karras et al. [17], as well as the 3D-ProGAN implemented by Eklund et al. [7] are publicly available. This will help greatly in guiding the implementation of a 3D-ProGAN for face generation.

5 DATASETS

Choosing a dataset is quite a crucial part of doing a research study as it is what your model will use to learn from, making this decision an important one. 3D facial datasets have nowhere close to the number of images that are normally present in a typically 2D facial dataset. This makes it both a bit more challenging to choose a dataset as well as exciting as nobody quite knows what kind of results to expect when such datasets do become available. Most datasets require you to request permission as facial data is very personal. Most of the datasets were chosen due to their diversity and the number of samples they had, as the more samples you have

Table 1: Comparison of datasets

Dataset	Number of images	Number of subjects	Type of data	Download access
Binghampton University 3D Facial Expression (BU3DFE)	2500	100	3D expression model	Request permission
Chinese Academy of Sciences Institute of Automation (CASIA-3D)	4624	123	2D colour image & 3D facial triangulated surface	Create an account
FaceScape	18760	938	3D face models, parametric models multi-view images	Request permission
Facewarehouse	-	150	RGBD images, feature points fitted meshes	Request permission

the better the generator is able to get. A comparison of datasets can be found in table 1.

6 EVALUATION OF FACIAL GANS

This is a challenging part of GAN research as there are no commonly agreed-upon metrics that are used. All the different researches seem to use a bit of this and a bit of that. Following you will find two types of evaluation, namely qualitative and quantitative.

6.1 Qualitative Evaluation

This type of evaluation involves checking the quality of the generated faces, such as the resolution.

6.1.1 Frechet Inception Distance(FID). Looks at the difference between the mean and covariance of the real and fake distributions [12]. A lower FID indicates that the real and fake distributions are closer together, they have similar features.

6.1.2 Kernel Inception Distance(KID). Involves looking at the dissimilarity through the squared maximum mean discrepancy. A lower KID means a better-generated sample.

6.1.3 Perceptual Path Length(PPL). Estimates the quality of the latent space by measuring the distance from the latent space to the output [4]. A Smaller PPL indicates higher image quality.

6.1.4 Structural Similarity Index(SSIM). Measures the low-level similarity to the real image. The higher the SSIM, the more similar the generated image is to the real one.

6.1.5 Multi-Scale Structural Similarity Index(MS-SSIM). An improved version of SSIM as the similarity is conducted over multiple scales instead of just the one. As with SSIM, a higher MS-SSIM indicates a higher similarity between the two images.

6.1.6 Inception Score(IS). Looks at two properties of images, that is quality and diversity [22]. A high IS indicates that conditional probability is low (finding relevant objects and features), while the marginal probability is high when there is a diverse set of features.

6.2 Quantitative Evaluation

Quantitative measurements are all about the accuracy of how well GANs are able to generate objects, in this case, faces. It is common to report back on the different accuracies achieved for each of the different emotions(happy, sad, angry, etc.), the different features of the face(eyes, nose, mouth, etc.) and the age.

7 ISSUES

Most issues related to GANs are based on how the generator and discriminator interact with each other. If there is unstable learning, which leads to either the generator or discriminator being learning faster than the other, issues arise such as the ones that follow. Not all issues discussed below are entirely about the GANs themselves, issues also arise due to the datasets that are available to use.

7.1 Mode Collapse

When one of the NNs gets stuck in the local minima of gradient descent. This occurs as GANs tend to not learn the whole distribution of the dataset. The generator focuses on producing what it thinks the discriminator is most likely to accept, but this leads the discriminator to accept only the best samples produced by the generator. An equilibrium isn't achieved between them.

7.2 Vanishing Gradients

Mostly due to the type of loss function that is being used, it tends to occur when the discriminator becomes too much of a perfectionist and gives no helpful feedback to the generator. This leaves the generator with no direction on how to improve its generated samples, it gets stuck.

7.3 Dataset Access

As faces are very personal, nearly all the 3D face datasets require you to request permission to be able to download and use them (and if you are a student you will need your supervisor to request permission as most of the datasets state that this is how students can gain access). This leads to an issue of not acquiring the best dataset available.

8 CONCLUSION

The recent techniques used for applying GANs to face generation. It was clear to see that the more explored area of face generation lies in 2D, as the datasets available are mostly 2D datasets. Using 2D datasets for 3D applications produces satisfactory results but not quite at the same level as when you use a 2D dataset for a 2D application. There were a few researches that did use a 3D dataset for a 3D application and these were seen to outperform the other state of the art techniques that used a 2D dataset for a 3D application.

As 3D datasets are not as available as 2D datasets, the expected performance of a 3D GAN with a 3D dataset is unknown in terms of how good they are able to generate. But due to more and more 3D datasets becoming available to use, under certain conditions, this starts to ask the question of how well a GAN is able to do using a 3D dataset for a 3D application. From the look at some of the latest researches released regarding GANs and 3D face generation, there was no sign of a 3D ProGAN used for 3D face generation. Meaning either it has been attempted before behind closed doors and failed or it is an empty area waiting to be explored. 3D ProGANs do exist just not for 3D face generation, meaning they are not impossible to create and that would be the next step this project is looking to take.

REFERENCES

[1] Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. 2019. A decoupled 3d facial shape model by adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9419–9428.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[3] Alan Berman. 2020. *Generative adversarial networks for fine art generation*. Master's thesis. University of Cape Town.

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401* (2018).

[5] Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 187–194.

[6] Shiyang Cheng, Michael Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. 2019. Meshgan: Non-linear 3d morphable models of faces. *arXiv preprint arXiv:1903.10384* (2019).

[7] Anders Eklund. 2019. Feeding the zombies: Synthesizing brain volumes using a 3D progressive growing GAN. *arXiv preprint arXiv:1912.05357* (2019).

[8] Baris Gecer, Alexandros Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. 2020. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *European conference on computer vision*. Springer, 415–433.

[9] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. 2019. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1155–1164.

[10] Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* (2016).

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

[13] Long Hoang, Suk-Hwan Lee, Oh-Heum Kwon, and Ki-Ryong Kwon. 2019. A deep learning method for 3D object classification using the wave kernel signature and a center point of the 3D-triangle mesh. *Electronics* 8, 10 (2019), 1196.

[14] Amina Kammoun, Rim Slama, Hedi Tabia, Tarek Ouni, and Mohmed Abid. 2022. Generative Adversarial Networks for face generation: A survey. *ACM Computing Surveys (CSUR)* (2022).

[15] Nilanjana Karmakar, Arindam Biswas, Partha Bhowmick, and Bhargab B Bhattacharya. 2013. A combinatorial algorithm to construct 3D isothetic covers. *International Journal of Computer Mathematics* 90, 8 (2013), 1571–1606.

[16] Animesh Karnewar and Oliver Wang. 2020. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7799–7808.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Proc. NeurIPS*.

[19] Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. 2020. AvatarMe: Realistically Renderable 3D Facial Reconstruction" in-the-wild". In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 760–769.

[20] Stylianos Moschoglou, Stylianos Ploumpis, Mihalis A Nicolaou, Athanasios Papaioannou, and Stefanos Zafeiriou. 2020. 3dfacegan: Adversarial nets for 3d face representation, generation, and translation. *International Journal of Computer Vision* 128, 10 (2020), 2534–2551.

[21] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7588–7597.

[22] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems* 29 (2016).

[23] Wenxuan Wu, Zhongang Qi, and Li Fuxin. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9621–9630.