3D Face Cartoonizer: Generating Personalized 3D Cartoon Faces from 2D Real Photos with a Hybrid Dataset

Ming Guo[1], Shunfei Wang[2], Zhibo Wang[1], Ming Lu[3], Xiufen Cui[2], Xiao Ling[2], and Feng Xu[1]

[1] BNRist and school of software, Tsinghua University
[2] OPPO, China
[3] Intel Labs China

Abstract. Cartoon face is a prevalent kind of stylized face, which is widely used in movies, TVs and advertisements. Although plenty of methods have been proposed to generate 2D cartoon faces, it is still challenging to learn personalized 3D cartoon faces directly from 2D real photos. To solve this problem, we contribute the first 3D cartoon face hybrid dataset with both large amounts of low-quality and a small number of high-quality face triplets. Each triplet contains a 2D real face, as well as its corresponding 2D and 3D cartoon faces. To leverage the hybrid dataset, we propose *Recon2AGen* which first pretrains our network with low-quality triplets in a reconstruction-then-generation manner and then finetunes it with high-quality triplets in an adversarial manner. In this way, we solve the 2D-to-3D ambiguity and the real-to-cartoon transformation by disentangling the task into three progressively learned subtasks. And the hybrid dataset is fully explored to achieve generalizable and high accuracy results. Extensive experiments show that our generated 3D cartoon faces are of high quality and can be easily edited and animated, enabling extensive practical applications. Code and dataset will be released.

Keywords: 3D face generation · Cartoon face · Dataset.

1 Introduction

As a popular kind of stylized face, cartoon faces have rich application scenarios. Generating cartoon faces directly and automatically from real faces can largely extend the ability of digital content creation, and is widely demanded in applications such as movies, advertisements, games and virtual reality.

Although many previous works can generate high-quality 2D cartoon faces [9, 5, 21, 12, 18, 22], the generation of 3D cartoon faces still mainly relies on tedious manual works by artists with professional 3D modeling software. Therefore, high-quality 3D cartoon faces are usually only used in high-end fields such as game and film production. As for ordinary users, it is challenging to customize 3D cartoon faces. Although there are some 3D stylized face generation methods,

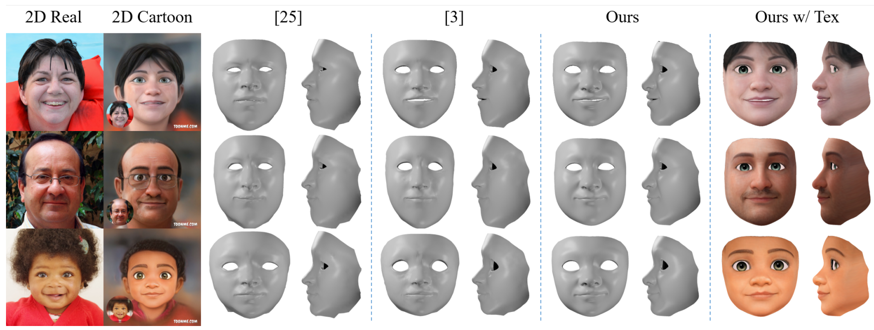


Fig. 1. We propose 3D Face Cartoonizer to generate personalized 3D cartoon faces directly from 2D real face photos. From left to right: input real faces, 2D cartoon faces, 3D cartoon face results of [25] and [3], and our results without and with texture. Notice that neither [25] nor [3] can directly obtain the results from real photo inputs as we do. They relay on the 2D cartoon faces.

most of them focus on caricatures, which are quite different from cartoon faces. Besides, they usually require sketches as additional input [10, 11] or need to know the 3D model of the input real face [24].

In this paper, we introduce a learning-based method to generate both the shape and texture of 3D cartoon faces directly from 2D real face images, as shown in Figure 1. To achieve this, we built a hybrid dataset, consisting of 6,842 low-quality and 130 high-quality face triplets. Each triplet contains a 2D real face image, its corresponding 2D cartoon face and textured 3D cartoon face. The 2D cartoon face is generated using a popular web application called ToonMe[4]. For the low-quality triplets, the 3D cartoon faces are generated by a landmark-guided deformation method inspired by [25], which only fits "coarse shapes". We get a large number of low-quality triplets as the fitting is automatic given the landmark annotations. The high-quality triplets contain 3D cartoon faces of much higher quality, but obtaining them is expensive and time-consuming as the creation heavily relays on professional artists. So, we just collect a relatively small number of high-quality triplets in the hybrid dataset. In general, our hybrid dataset strikes a balance between high quality, large quantity, and low costs.

Given the hybrid dataset, generating 3D cartoon faces from 2D face images is still challenging. For geometry generation, learning the relationship between 3D cartoon faces and 2D real faces suffers not only the 2D-to-3D ambiguity but also the real-to-cartoon transformation. We address this by a novel training strategy, called *Recon2Gen*, that learns reconstruction before generation by just using the low-quality dataset. This strategy is effective as it utilizes an easier task (reconstruction) to pretrain the network, making the final task (generation) can be achieved even with low-quality training data. Next, to make our results of "higher quality", we further finetune our geometry synthesis module in an adversarial manner with the high-quality data. Given the model already trained by the low-quality data, only 90 high-quality triplets for training are enough to gen-

[4] https://toonme.com

erate vivid results during testing. This three-stage progressive learning strategy, named *Recon2AGen* (adding an "*A*" to *Recon2Gen* standing for the adversarial training), leverages the characteristic of our hybrid dataset and achieves high-quality results for 3D carton face generation.

To further synthesize a fully textured 3D cartoon face, we integrate style transfer and texture generation into a single geometry-aware UV-space synthesis approach, which gives personal-stylized cartoon texture for our generated cartoon meshes. In summary, the contributions can be concluded as follows:

(i) To the best of our knowledge, 3D Face Cartoonizer is the first method that generates 3D cartoon faces directly from 2D facial images with high-quality geometry and texture.

(ii) We contribute the first hybrid dataset for 3D cartoon face generation, which contains 2D facial images and their corresponding 2D and 3D cartoon faces (with different qualities for the 3D cartoon faces), and will be released for future research.

(iii) We propose a novel *Recon2AGen* method which fully explores our hybrid dataset and solves both the 2D-to-3D ambiguity and the real-to-cartoon transformation.

2 Related Work

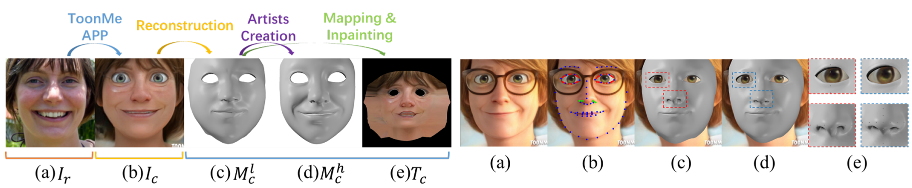
3D Stylized Face Reconstruction Thanks to the rapid development of 3D human face reconstruction, several stylized face reconstruction works have emerged recently. However, almost all of them are focused on caricatures. Due to the considerable geometrical difference between caricatures and real faces, previous work [25] has demonstrated that directly using normal face models such as 3DMM [27], FaceWareHouse [4], FaceScape [26] and FLAME [16] cannot fit them correctly. Approaches that aim to address this limitation can be divided into two categories. In the first category, the problem is solved by manually making 3D meshes for caricatures and using the results to build a specific parametric model, such as [19]. However, it is time-consuming and costly for artists to make a great number of 3D caricatures. In the second category, the problem is solved by designing parametric models beyond the scope of normal face models, such as [25] and [3]. However, they both only consider sparse constraints of landmarks, so the results are not so satisfying and lack details. While plenty of methods are proposed for 3D caricature faces, as one popular style type, 3D cartoon faces are never studied by existing methods to the best of our knowledge.

3D Stylized Face Generation Similar to 3D stylized face reconstruction, existing works on 3D stylized face generation primarily focus on caricature face generation. Some of them are based on the 3D models of real faces and generate caricatures by exaggerating the difference between the input face and the mean face [15, 24]. Other methods require additional input like sketches drawn by users [10, 11]. Although they can obtain 3D caricatures, the above methods are highly dependent on the quality of the input sketch and are not totally automatic. [17]

uses 2D real faces as input, while the generated results are limited in a predefined PCA space.

Face Texture Generation The texture is essential for 3D faces generation. However, as the texture of cartoon faces is quite different from real faces, existing statistical texture models [2, 16] built on real faces cannot be applied to 3D cartoon faces. To extend the space of the predefined linear model, GAN is used to generate real face texture in high fidelity [8, 6]. [5] designs a GAN-based cycle-consistency network to transfer the color style from real faces to caricatures for 2D caricatures generation. Other methods [20, 23, 1] try to solve face style translation tasks by designing an encoder for StyleGAN [14]. These style transfer methods focus only on 2D cartoon portrait generation but not complete UV texture.

3 Hybrid Cartoon Dataset



(a) The pipeline for building the triplets in our hybrid dataset.

(b) The 106-point landmarks with the fitting results.

Fig. 2. Preparation process of our hybrid cartoon dataset.

To facilitate the learning of 3D cartoon face generation, we construct a hybrid cartoon dataset with both low and high-quality data. It connects real and cartoon face domains, providing both 2D and 3D information with different quality as Figure 2a. Specifically, our dataset contains 6,972 data triplets, each of which includes a real facial image I_r, its corresponding 2D cartoon image I_c generated by ToonMe, and its 3D cartoon face mesh M_c with texture T_c. For the 6,842 low-quality data triplets, the 3D cartoon face meshes M_c^l are generated by a denser-landmark guided face fitting algorithm. To provide more explicit 3D cartoon guidance, our dataset also contains 130 high-quality 3D cartoon faces M_c^h created by expert artists based on M_c^l, forming the high-quality triplets. Details are described as follows.

The real face photos I_r are selected from FFHQ dataset [14] and we manually filter out samples whose generated 2D cartoon faces have obvious artifacts. From the 2D cartoon images I_c, we construct the 3D cartoon faces of two quality levels (M_c^l and M_c^h) with cartoon textures T_c.

Low-Quality Data To get a large amount of low-quality faces M_c^l from given 2D cartoon images automatically, we refine the landmark-guided fitting-based stylized face reconstruction method [25] with extra style-related landmarks. Notice that the standard 68-point landmark-setting (blue points in Figure 2b(b)) used in [25] fails to reconstruct accurate eyelid and nose shapes of a cartoon face as shown in Figure 2b(c). Therefore, we introduce extra landmarks around the eyelids and nostrils, annotated as red and green points in Figure 2b(b) respectively to get the final 106-point landmarks setting. These extra landmarks can be automatically located by extracting the eyelid and nostril edges via Sobel operator [7] with the help of the pre-mentioned 68-point landmarks. As shown in Figure 2b(d-e), this helps the fitting algorithm reconstruct better 3D cartoon faces with smoother and more natural eyelid contours and the characteristic chubby nose shapes in this cartoon style.

High-Quality Data The cartoon face reconstruction method above only relies on landmarks and fails to capture the detailed stylized shapes of the cartoon faces, *e.g.*, deep nasolabial folds. Therefore, to enhance the quality of the 3D cartoon faces, we select 130 representative faces from the low-quality 3D cartoon data and ask three expert artists to refine these 3D models according to the corresponding 2D cartoon images with the 2D real photos as extra identity reference. These are referred to as high-quality data in our dataset.

Texture As textures are one of the key components in reflecting cartoon style, we reconstruct cartoon textures based on the 2D cartoon images and the 3D cartoon faces. We first obtain a coarse cartoon texture map by mapping the 2D cartoon face image onto the 3D cartoon mesh. Then, for the invisible facial regions, we set their texture color as their symmetrical counterparts on the visible areas and finally apply Poisson inpainting to achieve complete cartoon texture maps as Figure 2a(e).

4 Method

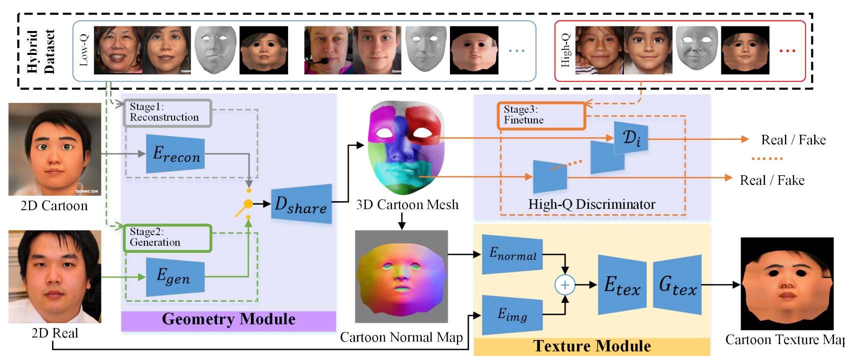


Fig. 3. Overview of our 3D Face Cartoonizer framework. Our framework contains two modules for geometry generation and texture generation. We design *Recon2AGen*, a progressive strategy that learns reconstruction before generation with the low-quality data and then learns to synthesize fine details with the high-quality data in an adversarial manner enabled with the High-Q Discriminator. Notice that we only need 2D cartoon faces for training, which are not required for real usage.

The architecture of the proposed 3D Face Cartoonizer is shown in Figure 3. Our method consists of a geometry module and a texture module which are trained on our hybrid cartoon dataset. We will introduce the two modules separately below.

4.1 Geometry Module

The geometry module is the most critical component of our method. It is trained to output a 3D cartoon facial geometry from a single real facial image. In order to generate high-quality outputs, we carefully design its network structures and train it using a novel training strategy *Recon2AGen*.

Network Structure We use a special encoder-decoder architecture for the geometry module. It contains two encoders E_{recon} and E_{gen} which can regress the feature vectors from 2D cartoon images and 2D real images respectively. The feature vectors can be divided into a geometry feature vector which only encodes the 3D cartoon facial geometry, and a 3D pose $P \in se(3)$. The geometry feature, either output by E_{recon} or E_{gen}, will then be fed into a shared decoder D_{share}, which has the same network architecture as [3]. The decoder D_{share} predicts deformation representation in [25] instead of directly outputting the vertex positions of the 3D cartoon face. By applying the estimated deformation gradient to the mean 3D cartoon face, the geometry module can generate the final shape of the 3D cartoon face.

Recon2AGen Training Strategy Generating a high-quality 3D cartoon face from a single real facial image is not easy as there are three major difficulties in learning this task: *1)* Recovering a 3D face from one 2D image is an ill-posed problem. *2)* Converting real faces to cartoon style needs to preserve the user-specific identity information. *3)* Learning to generate high-quality 3D cartoon faces is not easy when most of the training data is low-quality. Training the geometry module to solve the three problems in one stage will make the task even harder as all these difficulties will be coupled together. Therefore, we propose *Recon2AGen*, a progressive training strategy, in which the geometry module learns to solve these three problems respectively in three different stages. In *Stage 1*, the geometry module learns reconstruction using E_{recon} and D_{share}. It is trained to reconstruct the 3D cartoon face from a given 2D cartoon image, supervised by a large quantity of low-quality training data to overcome the ill-posed problem. In *Stage 2*, we fix the decoder D_{share} and train a new encoder E_{gen} from scratch to transfer the input 2D real image to 3D cartoon domain. In *Stage 3*, to enhance the quality of the 3D cartoon face generation, we finetune the geometry module on the artist-made high-quality data in a region-based adversarial manner. Specifically, we train 8 independent discriminators $\mathcal{D}_i$, $i \in \{1, \cdots, 8\}$, to distinguish the generated 3D face shapes in different local regions from the artist-made data while the geometry module is finetuned to fool these discriminators.

Training Losses To cooperate with *Recon2AGen* training strategy, we implement well-designed training losses L to train the geometry module, expressed as,

$$L = w_{geo}L_{geo} + w_{lmk}L_{lmk} + w_{sm}L_{sm} + w_{adv}L_{adv}, \tag{1}$$

where L_{geo} is a geometry loss, L_{lmk} is a landmark loss, L_{sm} is a smoothing loss, and L_{adv} is an adversarial loss. w_{geo}, w_{lmk}, w_{sm} and w_{adv} are the corresponding weights for different loss functions.

The geometry loss L_{geo} measures the difference between the generated 3D cartoon face $\hat{M}_c$ and the ground truth M_c, written as,

$$L_{geo} = \|\hat{M}_c - M_c\|_2^2. \tag{2}$$

The landmark loss is used to maintain the consistence between the ground truth 2D landmarks and the corresponding 3D vertices, expressed as,

$$L_{lmk} = \sum_{i \in \mathcal{K}_{lmk}} \|\boldsymbol{\Pi} \boldsymbol{P} \hat{\boldsymbol{v}}_i - \boldsymbol{k}_i\|_2^2 \tag{3}$$

where $\boldsymbol{k}_i$ is the detected 2D landmark and $\hat{v}_i$ is its corresponding vertex on the estimated 3D cartoon face. $\mathcal{K}_{lmk}$ is the set of the 3D landmark indices on the mesh. $\boldsymbol{\Pi}$ is the camera projection matrix.

Inspired by the Laplacian smoothing algorithm, we use a smoothing loss L_{sm} to alleviate artifacts such as folding surfaces and self-intersections by constraining the Laplacian coordinates of the estimated cartoon faces to be similar to those of the ground truth,

$$L_{sm} = \|\mathcal{L}\hat{M}_c - \mathcal{L}M_c\|_2^2, \tag{4}$$

where $\mathcal{L}$ denotes the Laplacian operator.

To extract the characteristics of the artist-made high-quality data, we use an adversarial loss which is formulated with the 8 discriminators. The discriminators are trained to distinguish the generated cartoon faces from the ground truth. The geometry module tries to fool the discriminators while keeping fitting the ground truth meshes. The L_{adv} is written as,

$$L_{adv} = \sum_{i=1}^{8} \mathbb{E}[\log \mathcal{D}_i(M_c)] + \mathbb{E}[\log(1 - \mathcal{D}_i(\hat{M}_c))]. \tag{5}$$

The geometry module is trained progressively in three stages. In the first two training stages, we set the loss weight w_{geo} to 10, w_{lmk} to 1×10^{-4}, w_{sm} to 1×10^3 and w_{adv} to 0. In *Stage 3*, we set w_{adv} to 5×10^{-4} while keeping the other loss weights unchanged. In this stage, the geometry module and the discriminators are trained in an adversarial manner, formulated as,

$$(E_{gen}, D_{share})* = \underset{E_{gen}, D_{share}}{\operatorname{argmin}} \max_{\mathcal{D}_i} L \tag{6}$$

4.2 Geometry-aware Texture Synthesis

Based on the proposed hybrid dataset, we train a geometry-aware GAN-structured network to synthesize complete cartoon texture from an input real facial image.

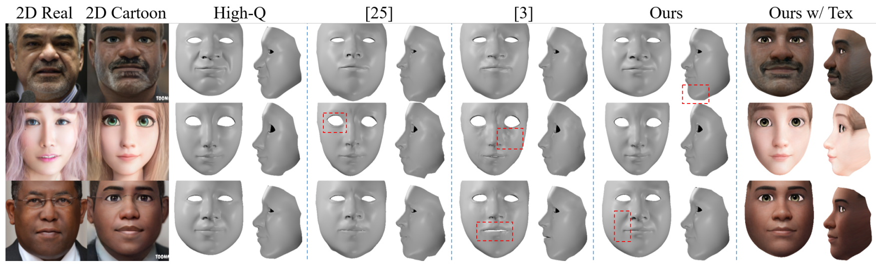


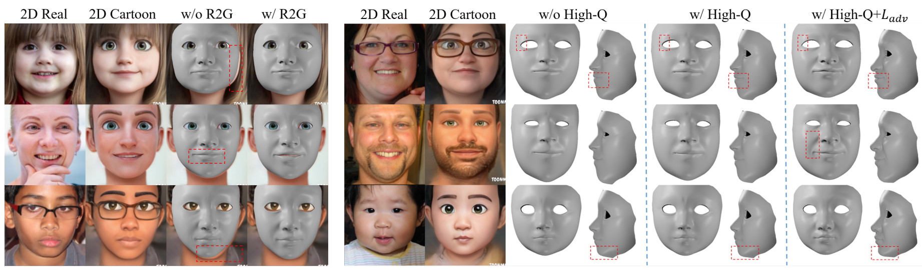
Fig. 4. Qualitative comparisons of our 3D cartoon face generation with other 3D cartoon face reconstruction methods ([25] and [3]).

As textures are strongly correlated to facial geometry, the texture module is guided by the geometric information of the 3D cartoon face predicted by the geometry module, as shown in Figure 3. Our geometry-aware GAN does not directly concatenate the input image with the geometry guidance. Instead, we first use two shallow encoders, noted as E_{img} and E_{normal}, to transfer the input image and the normal map in UV-space predicted by the geometry module into two feature maps to combine these two pieces of information in feature space. They will be added and then injected into pSp [20] which is the state-of-the-art encoder for StyleGAN (noted as E_{tex}). Finally, the pretrained StyleGAN using our texture dataset will generate texture maps in the UV-space with the input feature map output by E_{tex}. Details of the network structure and loss setting of our texture module can be found in our supplementary material.

5 Experiments

In this section, we evaluate the proposed hybrid dataset and 3D Face Cartoonizer with thorough qualitative and quantitative experiments. More results and further applications such as style editing and animation are in our supplementary material.

Experimental Settings. We train and evaluate both the geometry module and the texture generation module on our hybrid cartoon dataset. For the geometry module, we randomly choose 6140 triplets for training and the rest 702 triplets for testing from the low-quality data. In the high-quality data, there are 90 triplets for training and 40 triplets for testing. When training the texture generation module, we remove all triplets with eyeglasses to achieve clean facial textures. Finally, 4889 triplets are used for training leaving 556 triplets for testing. Both the real and cartoon images are aligned and resized to 224×224 before being fed into our network. The total training process costs about 6 days on a single GTX 2080 GPU. The average time to process an image in the inference stage is 0.02 ms for the geometry module and 0.08 s for the texture module.



(a) Evaluation of the *Recon2Gen* Strategy (b) Evaluation of the *AGen* Strategy

Fig. 5. Qualitative evaluations of the key components in our technique.

5.1 Comparisons

To the best of our knowledge, there is no previous method that can directly convert a real portrait to a 3D cartoon face. Therefore, prior to our work, a naive way to automatically generate a 3D cartoon face from a real image is to concatenate 2D real-to-cartoon translation and 2D stylized face reconstruction techniques. We combine ToonMe App with two state-of-the-art stylized face reconstruction methods separately, including a fitting-based method proposed by [25] and a learning-based method proposed by [3]. Note that [3] is retrained on our proposed dataset. We compare our methods with these two naive methods both qualitatively and quantitatively.

Qualitative Comparisons The qualitative comparisons among our 3D Face Cartoonizer and these two indirect solutions are shown in Figure 4. Compared with the two baselines, our geometry generation performs better in the following aspects. (1) On the "identity similarity", our results have more personalized overall face shapes (the 1st and 3rd row). (2) On the "style similarity", our results better represent the cartoon style, such as the chubby noses (the 1st row), the sharp curvatures of lips (the 2nd row), and plump shapes around cheekbones (the 1st and 3th row). (3) Our results also have richer "details" such as nasolabial folds shown in all rows. Besides, [3] suffers from unpleasant wrinkled surfaces(the 2nd row). Again, it should be noted that these naive solutions require ToonMe App to convert the real facial image to cartoon style and this will lead to more computation time. [25] also requires extra landmark annotations. On the contrary, our geometry module can synthesize high-quality 3D cartoon faces directly from an input real facial image and greatly enhance the convenience of 3D cartoon face generation.

We also demonstrate the effect of our texture generation module in the last column in Figure 4 by rendering the 3D cartoon face results with the generated textures. To make our results more vivid, we add 3D eye models by calculating the size and the location of the eyeballs according to the eyelids of the generated 3D cartoon face and selecting a suitable iris color based on the estimated race of the input photo using [13]. The generated textures dramatically enhance aesthetics and the identity similarity with the input face image.

Table 1. Perceptual study of different 3D cartoon face generation methods.

Metrics	[25]	[3]	Ours	High-Q
Artistry	2.67	2.88	**4.39**	4.27
Identity Similarity	2.56	2.75	3.72	**4.01**
Style Similarity	2.84	3.31	4.23	**4.28**

Quantitatively Comparisons We also quantitatively compare our method with [25] and [3]. As there is no "accurate" ground truth for 3D cartoon face generation, we conduct a perceptual study to demonstrate the visual quality of these methods. We invite 30 volunteers and each volunteer scores the results of the same input achieved by all approaches simultaneously. To get a thorough and quantitative evaluation of each method, we design 3 scoring dimensions (artistry, identity similarity and style similarity) according to previous works on similar tasks [17, 24]. Table 1 presents the average scores from the 30 volunteers. Our method achieves the most favorite results in all the aspects among all automatic methods and even outperforms the high-quality data in the artistry dimension.

5.2 Ablation Study

In this subsection, we evaluate the effect of the proposed *Recon2AGen* training strategy and *AGen* finetuning in 3D Face Cartoonizer.

***Recon2Gen* Training**. We measure the effectiveness of our *Recon2Gen* training strategy in improving the quality of cartoon face generation in Figure 5a. Without the *Recon2Gen* training strategy (noted as "w/o R2G", the geometry module fails to generate accurate face contours and sometimes leads to inaccurate face orientations. By pre-training on the task of cartoon face reconstruction (noted as "w/ R2G"), the geometry module generates better facial poses (the 1st row), expressions (the 2nd row), and more accurate facial contours (the last row).

***AGen* Finetuning**. To demonstrate the effect of *AGen* finetuning, We compare our method with two extra ablations, including one geometry module only trained with the low-quality data and another finetuned with the high-quality data without using the adversarial loss. As shown in Figure 5b, directly introducing high-quality data in the training brings minor improvement to the generated cartoon faces. By using an adversarial training loss, the geometry module is able to learn the style of the artist-made cartoon faces, such as shaper eye corner contours (the 1st row), a more vivid smile (side view of the 1st row), more pronounced nasolabial folds (the 2nd row), and a chubby and protruding chin (the 3rd row).

6 Conclusion

We proposed 3D Face Cartoonizer, the first automatic solution that directly generate a high-quality 3D cartoon face from a single facial image. In our solution, the method *Recon2AGen* solves the 2D-to-3D ambiguity and the real-to-cartoon transformation in a progressive manner with limited high-quality data, avoiding the huge cost of building a large scale artists-made 3D cartoon face dataset. For cartoon face generation, we presented a hybrid dataset consisting of triplets of real face images and their corresponding 2D and 3D cartoon faces, which may enable more research in this topic. Furthermore, the idea of leveraging a hybrid dataset is a good trade-off between high quality and low cost, and may be extended to other kinds related tasks in the future.

References

1. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: A residual-based stylegan encoder via iterative refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6711–6720 (2021)
2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 187–194 (1999)
3. Cai, H., Guo, Y., Peng, Z., Zhang, J.: Landmark detection and 3d face reconstruction for caricature using a nonlinear parametric model. Graphical Models **115**, 101103 (2021)
4. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics **20**(3), 413–425 (2013)
5. Cao, K., Liao, J., Yuan, L.: Carigans: Unpaired photo-to-caricature translation. arXiv preprint arXiv:1811.00222 (2018)
6. Deng, J., Cheng, S., Xue, N., Zhou, Y., Zafeiriou, S.: Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7093–7102 (2018)
7. Duda, R.O., Hart, P.E., et al.: Pattern classification and scene analysis, vol. 3. Wiley New York (1973)
8. Gecer, B., Ploumpis, S., Kotsia, I., Zafeiriou, S.: Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
9. Gong, J., Hold-Geoffroy, Y., Lu, J.: Autotoon: Automatic geometric warping for face cartoon generation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 360–369 (2020)
10. Han, X., Gao, C., Yu, Y.: Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. ACM Transactions on graphics (TOG) **36**(4), 1–12 (2017)
11. Han, X., Hou, K., Du, D., Qiu, Y., Cui, S., Zhou, K., Yu, Y.: Caricatureshop: Personalized and photorealistic caricature sketching. IEEE transactions on visualization and computer graphics **26**(7), 2349–2361 (2018)
12. Jang, W., Ju, G., Jung, Y., Yang, J., Tong, X., Lee, S.: Stylecarigan: caricature generation via stylegan feature map modulation. ACM Transactions on Graphics (TOG) **40**(4), 1–16 (2021)
13. Karkkainen, K., Joo, J.: Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1548–1558 (2021)

14. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
15. Lewiner, T., Vieira, T., Martínez, D., Peixoto, A., Mello, V., Velho, L.: Interactive 3d caricature from harmonic exaggeration. Computers & Graphics **35**(3), 586–595 (2011)
16. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4d scans. ACM Trans. Graph. **36**(6), 194–1 (2017)
17. Liu, J., Chen, Y., Miao, C., Xie, J., Ling, C.X., Gao, X., Gao, W.: Semi-supervised learning in reconstructed manifold space for 3d caricature generation. In: Computer Graphics Forum. vol. 28, pp. 2104–2116. Wiley Online Library (2009)
18. Pinkney, J.N., Adler, D.: Resolution dependent gan interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334 (2020)
19. Qiu, Y., Xu, X., Qiu, L., Pan, Y., Wu, Y., Chen, W., Han, X.: 3dcaricshop: A dataset and a baseline method for single-view 3d caricature face reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10236–10245 (2021)
20. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2287–2296 (2021)
21. Shi, Y., Deb, D., Jain, A.K.: Warpgan: Automatic caricature generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10762–10771 (2019)
22. Song, G., Luo, L., Liu, J., Ma, W.C., Lai, C., Zheng, C., Cham, T.J.: Agilegan: stylizing portraits by inversion-consistent transfer learning. ACM Transactions on Graphics (TOG) **40**(4), 1–13 (2021)
23. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. arXiv preprint arXiv:2102.02766 (2021)
24. Vieira, R.C.C., Vidal, C.A., Cavalcante-Neto, J.B.: Three-dimensional face caricaturing by anthropometric distortions. In: 2013 XXVI Conference on Graphics, Patterns and Images. pp. 163–170. IEEE (2013)
25. Wu, Q., Zhang, J., Lai, Y.K., Zheng, J., Cai, J.: Alive caricature from 2d to 3d. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7336–7345 (2018)
26. Yang, H., Zhu, H., Wang, Y., Huang, M., Shen, Q., Yang, R., Cao, X.: Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 601–610 (2020)
27. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S.Z.: High-fidelity pose and expression normalization for face recognition in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 787–796 (2015)