CPTNet: Cascade Pose Transform Network for Single Image Talking Head Animation

Jiale Zhang[1], Ke Xian[1], Chengxin Liu[1]*, Yinpeng Chen[1], Zhiguo Cao[1], and Weicai Zhong[2]

[1] Key Laboratory of Image Processing and Intelligent Control, Ministry of Education; School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Huawei CBG Consumer Cloud Service Search Product & Big Data Platform Dept
{jiale_zhang, cx_liu, zgcao}@hust.edu.cn

Abstract. We study the problem of talking head animation from a single image. Most of the existing methods focus on generating talking heads for human. However, little attention has been paid to the creation of talking head anime. In this paper, our goal is to synthesize vivid talking heads from a single anime image. To this end, we propose cascade pose transform network, termed CPTNet, that consists of a face pose transform network and a head pose transform network. Specifically, we introduce a mask generator to animate facial expression (*e.g.*, close eyes and open mouth) and a grid generator for head movement animation, followed by a fusion module to generate talking heads. In order to handle large motion and obtain more accurate results, we design a pose vector decomposition and cascaded refinement strategy. In addition, we create an anime talking head dataset, that includes various anime characters and poses, to train our model. Extensive experiments on our dataset demonstrate that our model outperforms other methods, generating more accurate and vivid talking heads from a single anime image.

1 Introduction

Talking head animation, as the name suggests, refers to the change of facial expression and head movement of anime characters. It is a very interesting task which has broad application scenarios, including game production, filmmaking, and virtual avatars. Recently, great progress has been made in human talking head generation with the introduction of Generative Adversarial Networks (GANs) [1]. For instance, some methods [2–4] are able to synthesize different expressions as well as change attributes of human face, such as hair color, skin and age. However, there is a big difference between human faces and anime faces: patterns of human faces are highly structured, while anime faces are rather diverse. Also, different anime style leads to significant different face patterns, such

* Chengxin Liu is the corresponding author.

as small mouth, no nose, and large eyes. Therefore, such methods trained on human datasets (*e.g.,* RaFD [5] and CelebA [6]) cannot be directly used for anime talking head generation.

In this paper, we aim at designing a model for generating anime talking heads, including the change of anime facial expression (*e.g.*, close eyes and open mouth) and the movement of anime head. Since there is no open source anime dataset available on the Internet, we create an anime talking head dataset consisting of 1842 anime IDs. Each anime ID has 150 face poses and 973 head poses with corresponding pose vectors.

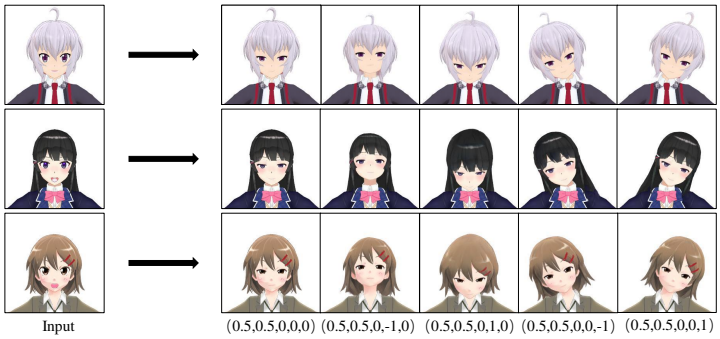

Fig. 1. Some qualitative results on our anime dataset. Given a single anime image (first column) and a target pose vector (last row), our method is able to generate vivid talking heads (from the second column to the sixth column). Specifically, we encode the change of facial expression and head movement into a pose vector (*i.e.,* left eye, right eye, mouth, top-down head, left-right head). Note that the values of eyes and mouth vary from 0 to 1, while head ranges from -1 to 1.

To generate vivid talking heads from a single anime image, we disentangle the problem into two stages: the change of anime facial expression and the movement of anime head. In particular, we propose a two-stage CPTNet, that includes a generator for facial expression, a generator for head movement, and a light-weight combiner, for single image talking head animation. The two generators require a straight anime image and a target pose vector as input, then generate the change of anime facial expression and the movement of anime head, respectively. The generator for facial expression change shares the same architecture and weights with the mask branch of head generator. The head generator has two branches: the mask branch [7] and the grid branch [8]. The mask branch is able to preserve static area via the mask and generate new pixels, but is prone to be blurred at uncertain area. The grid branch can generate clear predictions by bilinear sampling, but it can only copy pixels from the input image and cannot generate new pixels. In order to combine the advantages and make up for the disadvantages of these two branches, we utilize the light-weight combiner

to fuse the results of these two branches together. To handle large-scale pose transformations, we further propose a pose vector decomposition and cascaded refinement strategy for the mask branch. As shown in Fig. 1, given a single anime image and a target pose vector, our method is able to generate high-quality anime talking heads from a single image. Note that, by giving different target pose vectors, different poses of talking heads can be generated.

It is worth mentioning that our approach is inspired by Pramook [9] that builds a two-step system for talking head animation. However, our approach is different from his work in three main aspects. First, our approach has no constraints on input image format and anime characters, while Pramook requires an RGBA input and restricts anime head in the center of 128×128 regions of the input image. Second, we can handle larger-scale pose transformations via the pose vector decomposition and cascaded refinement strategy. Finally, weights sharing and light-weight fusion module ensure the efficiency of our model, which only takes up $40M$, while Pramook's model exceeds $300M$.

The contributions of this work can be summarized as follows:

• We present CPTNet, a novel two-stage pose transform network for talking head animation from a single anime image.

• We design a pose vector decomposition and cascaded refinement strategy to handle large-scale pose transformations.

• Extensive experiments on our anime dataset demonstrate the effectiveness of our approach, which outperforms other state-of-the-art methods.

2 Related Work

Generative Adversarial Networks. In recent years, GANs [1] have developed rapidly and have shown surprising results in computer vision, such as image translation [10–14], pose transform [15, 16], image generation [17], super-resolution imaging [18], and face expression editing [4, 3, 19]. A GAN-based architecture contains a generator and a discriminator, while the generator aims at generating realistic fake samples, the discriminator needs to be able to distinguish between the real samples and the fake samples generated by the generator. In training, this idea is completed by an adversarial loss.

Conditional GANs. Conditional GANs have also attracted the attention of researchers. For example, multi-domain transfer [14], human pose transform [15, 16], and facial expression editing [4, 3, 20]. Prior studies have tried to add some conditions to the basic GANs, such as class information [21] and text description [22], to generate images highly relevant to these conditions.

Image-to-Image Translation. Recent studies have shown surprising results in image-to-image translation [10–12, 23], *e.g.,* pix2pix [10] can complete image-to-image translation well with paired data. However, sometimes we are unable to obtain paired data in practice. As a result, some unsupervised methods are proposed [24, 11, 12, 23]. For instance, CycleGAN [24], using the strategy of cycle consistent loss to maintain the necessary attributes in input images, is able to complete high-quality image translation with unpaired data. CartoonGAN [12]

converts input images to cartoon style, and Art2Real [13] converts art images and real images to each other. Our task is more related to those works that use paired data due to our paired data.

Appearance and Pose Transform. Appearance and pose transform has attracted great attention in recent years. Most works [4, 15, 16] focus on human appearance and pose transform, such as changing color of hair, gender aging, and generating human talking head. GANimation [7], for instance, can change the appearance and expression of human. It requires a source human image and a target vector as input and then generates the changed image corresponding to the target vector. A few-shot learning method [15] has been proposed to generate human talking head. This method takes a few source human images and a target landmark as input, and generates the target human pose according to the target landmark. However, these works are all designed for human appearance and pose transform, they cannot handle anime characters well. Recently, Pramook [9] proposed a method that can perform anime pose transform through a single anime image and a target vector, which is most related to us. But his model is cumbersome and cannot handle large scale motion. In contrast, our method is efficient and is capable of handling large scale motion.

3 Method

Given an input straight image of anime character and a target pose vector, our goal is to design a network to transform the anime character to arbitrary target pose. Specifically, we disentangle this problem into two subtasks, including face pose transform (*e.g.*, close eyes and open mouth) and head pose transform. To address these two subtasks, we propose a cascade pose transform network. As shown in Fig. 2, our model consists of two stages: face pose transform stage and head pose transform stage. First, a mask generator is applied to transform the anime character to a target face pose (*e.g.*, close eyes and open mouth), then mask and grid branches generate two complementary head pose results according to the target head pose vector. Finally, a light-weight combiner is used to fuse these two results into our final result. Note that the mask generator in face pose transform stage has the same architecture and weights as the mask branch in head pose transform stage. More details will be introduced in the following sections.

3.1 Data Generation

Since there is no related anime dataset on the Internet, we create an anime pose dataset ourselves with a 3D software MikuMikuDance (MMD). Our dataset contains two parts, which are the face pose dataset of anime characters and the head pose dataset of anime characters. Below we will introduce them separately.

There are 1842 model IDs in our anime face pose dataset, and each model ID has 150 poses and corresponding pose vectors. The format of the face pose vector is: (left eye, right eye, mouth), and the value ranges of the three elements

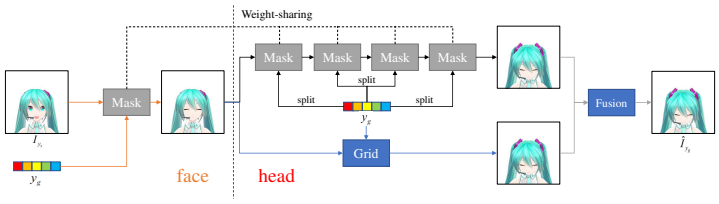

Fig. 2. Overview of our network architecture. Our model consists of two stages: face pose transform stage (*e.g.*, close eyes and open mouth); and head pose transform stage. In face pose transform stage, the mask generator transforms an anime character to a target face pose. In head pose transform stage, mask generator and grid generator are utilized to generate two complementary talking heads. Finally, the fusion module fuses these two results to generate the final result.

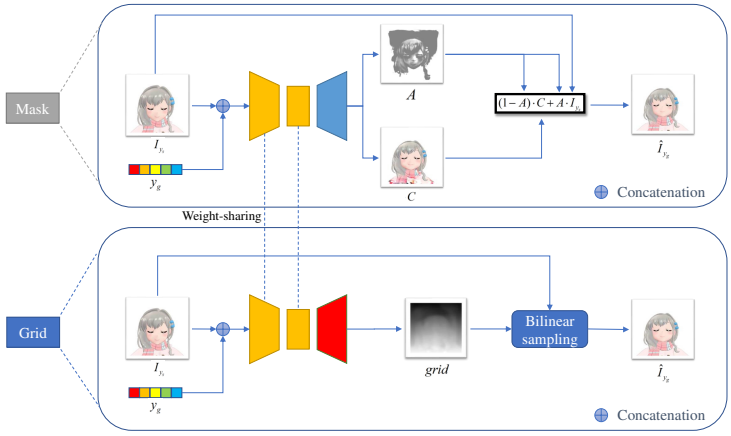

Fig. 3. Mask generator and Grid generator. Given a single anime image and a target pose vector, on one hand, the mask generator generates a single channel mask A and an RGB content image C. Then mask A linearly combines the source image and the content C to obtain the output. On the other hand, the grid generator generates a grid to get the output via bilinear sampling.

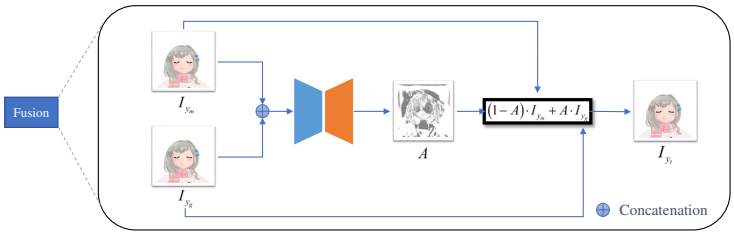

Fig. 4. Fusion. The output I_{y_m} of mask generator and the output I_{y_g} of grid generator are made a concatenation as the input of fusion. The output of fusion is an attention mask A, which guides two source images to fuse to obtain the final result.

are all $[0, 1]$, 0 means the eyes and mouth are fully closed, and 1 means the eyes and mouth are fully open.

Besides, there are 1868 model IDs in our anime head pose dataset, each model ID has 972 poses and corresponding pose vectors. The format of the head pose vector is: (top-down head, left-right head), and the value ranges of the two elements are all [-1,1], corresponding angel $[-20°, 20°]$. Note that our image resolution is 256×256 and the testing set for our task contains 260 anime models. Some examples are shown in Fig. 5.

When we train and test the model, the format of a pose vector that we use is (left eye, right eye, mouth, top-down head, left-right head), which means the combination of face pose vector and head pose vector.

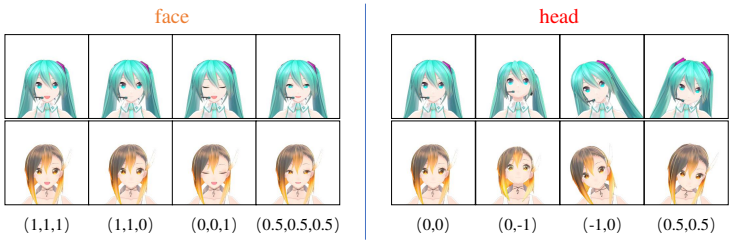

Fig. 5. Dataset. The left is an example of the anime face pose dataset, the right is an example of the anime head pose dataset, and the bottom row is the pose vector corresponding to each column of images.

3.2 Network Architecture

Mask Generator. Inspired by the recent success of generating human facial expression [7], we propose to use mask-guided generator to handle pose trans-

form. Fig. 3 illustrates the details of mask generator. Given a single image I_{y_s} and a target pose vector y_g, we first make a channel wise concatenation (I_{y_s},y_g) as input of mask generator. Then the mask generator generates a single channel mask A and an RGB content C, where mask A is designed to guide the network to focus on the transforming pose-related regions.

For instance, in the face stage, the only areas where the input image needs to be changed are the eyes and mouth, while the other areas are kept still, so the network only needs to generate the changed dynamic areas, the remaining static areas can be obtained directly from the input.

Finally, the output image can be obtained as:

$$\hat{I}_{y_g} = (1 - A) \cdot C + A \cdot I_{y_s}, \tag{1}$$

where $\cdot$ denotes element-wise multiplication. This strategy makes the learning process easier as the network only needs to focus on the dynamic areas, the static areas can be obtained from the input directly. However, mask generator is prone to be blurred, and it cannot maintain the color of input image well if given a large-scale pose vector.

Grid Generator. Since mask generator may produce blurry results, we adopt grid generator as a complementary component to synthesize new pose. The details of grid generator are shown in Fig. 3. For the input image and target pose vector, we use the same way as the mask generator. The main difference is that the output of grid generator is a two-channel grid vector, not mask and content. We use the grid vector to synthesize target image by bilinear sampling from the input image, which is similar to appearance flow [8].

The motivation of the grid generator is that bilinear sampling can make full use of the pixel information of the input image. The network only needs to generate a two-channel grid vectors to guide the sampling process from input image, rather than generating a complete RGB image, which significantly reduces the difficulty of learning process. In addition, since bilinear sampling directly copies pixels from the input image to the final image, the color identification of the input image will be well preserved, and the resulting image will not produce obvious blur.

However, the inherent limitation of this method is that it cannot generate new pixels. Therefore, some areas that do not contain relevant pixel information in the input image will not be generated well. Note that grid generator and mask generator share the same weights in encoder and residual blocks [25].

Fusion. The fusion architecture is shown in Fig. 4. After the mask generator and grid generator output the results I_{y_m} and I_{y_g}, we make a channel wise concatenation (I_{y_m},I_{y_g}) as the input of fusion. Similar to the mask generator, the fusion outputs an attention mask A to guide the two source images to fuse. Finally, the final image can be obtained as:

$$\hat{I}_{y_t} = (1 - A) \cdot I_{y_m} + A \cdot I_{y_g}, \tag{2}$$

The motivation of applying fusion module is straightforward, the mask generator can synthesize new pixels and regions that are not included in the input image,

but it may produce a certain degree of blur and the quality of the generated image is not stable, while the images generated by the grid generator are clearer and more stable, but it cannot generate new pixels. Therefore, a fusion module is adopted to combine the advantages of these two generators. It is noticeable that the size of our fusion model is only 5 MB.

3.3 Pose Vector Decomposition

In head pose transform stage, we observe that for large-scale pose vectors, the image generated by the mask branch is very blurry. We infer the reason is that the large-scale transform leads to few static regions, which significantly increases the difficulty of generating dynamic regions for mask generator. To solve this problem, we introduce the pose vector decomposition and cascaded refinement strategy as shown in Fig. 3. Specifically, we decompose large-scale pose vector of head movement into k small-scale pose vectors and pass them through k cascaded mask generators that share weights parameters. For instance, suppose the original large-scale pose vector is $(0, 0, 0, 0, 1)$, we first transform the anime character to a target pose $(0, 0, 0, 0, 1/k)$, then the pose $(0, 0, 0, 0, 2/k)$ is generated based on the result of $(0, 0, 0, 0, 1/k)$. This process repeats until $(0, 0, 0, 0, 1)$ is reached. In practice, we observed that $k = 4$ is sufficient to handle large-scale pose transforms. In this way, the mask generator can repeatedly perform small-scale transform to complete large-scale transform, which improves the final result.

3.4 Learning the Model

At the training stage, our loss function mainly consists of three terms: 1) the adversarial loss for improving the photo-realism of generated pose transformation images. 2) the content loss to improve consistency of generated images with ground truth 3) the perceptual loss to improve the clarity and details of the output.

Adversarial Loss. To make the generated images indistinguishable from real images, we adopt an adversarial loss as

$$L_{adv} = E_{\hat{I}_{y_g} \sim P_s} \left[D\left(\hat{I}_{y_g}\right) \right] - E_{I_{y_g} \sim P_{data}} \left[D\left(I_{y_g}\right) \right] \tag{3}$$

where $\hat{I}_{y_g}$ is the final generated image and I_{y_g} is the corresponding ground truth, P_s stands for the data distribution of synthesized images, P_{data} the distribution of real images. The generator needs to generate real fake images so it tries to maximize this objective, while the discriminator D needs to distinguish between real and fake images so it tries to minimize this objective.

Content Loss. Since our dataset contains pairs of samples, we use L1Loss to measure the distance between the generated image and the ground truth and then update the generator. The content loss is defined as

$$L_{pair} = E_{I_{y_g} \sim P_{data}} \left[\left\| I_{y_g} - \hat{I}_{y_g} \right\|_1 \right] \tag{4}$$

Perceptual Loss. Only a L1Loss constraint on the generated image and ground truth may cause the image to be blurred. So we adopt the perceptual loss [26] as another constraint. We let the generated image and its corresponding ground truth pass through the pre-trained VGG19 network [27], and extract the features of conv1_1, conv2_1, conv3_1, and conv4_2 layers for L1Loss, and finally weighted summation. So the perceptual loss can be obtained as

$$L_p = \sum_j E_{I_{y_g} \sim P_{data}} \left[\left\| \phi_j \left(I_{y_g} \right) - \phi_j \left(\hat{I}_{y_g} \right) \right\|_1 \right] \tag{5}$$

where $\phi_j(\cdot)$ represents the features of jth layer in VGG19, the j here specifically refers to the layers of conv1_1, conv2_1, conv3_1, and conv4_2. We find that this loss function can make the result smoother and clearer.

Full Loss. To generate the target image, we build a loss function L by linearly combining all previous partial losses:

$$L = L_{adv} + \lambda_1 L_{pair} + \lambda_2 L_p \tag{6}$$

where λ_1, λ_2 are the hyper-parameters that control the relative importance of every loss term.

4 Experiments

In this section, we evaluate our model on our own anime pose dataset. First we give the details about our experimental setting. Then we show the results of the face pose transform stage, head pose transform stage, and the final mixed pose separately to analyze the role of each module. Finally, we compare our model with some recent methods on our dataset.

4.1 Experimental Setting

Our generators and discriminator networks are build upon StarGAN [2], as it proved to achieve impressive results for image-to-image transform. For the mask generator, we made slight modification by adding a branch on the last convolutional layer so as to generate a single-channel mask A and an RGB content image C. For the grid generator, we change the last convolutional layer to output a two-channel grid to perform bilinear sampling on input image. As the discriminator, we remove the classification layer. Our fusion module contains only two downsampling layers and two upsampling layers, and output a single-channel mask.

The model is trained on our anime pose dataset mentioned above. We use Adam [28] with learning rate of 0.0001, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and batch size 8. We train for 600000 iterations and linearly decay the learning rate to zero over the last 100000 iterations. We perform one generator update after five discriminator updates. We set $\lambda_1 = 1000$, $\lambda_2 = 200$. It takes about three days to train the model with a single GeForce RTX 2080 Ti GPU.

Fig. 6. Face stage. The results of face stage. A represents mask, while C represents content. The last row means the target pose vectors, and every column is the generated mask A, content C, and output corresponding to the target pose vector.

4.2 Ablation Study

In this section, we performed an ablation study to evaluate the role of each of our modules: face stage only, head stage only, and the whole two-stage network. In head stage, we first evaluate the effectiveness of the mask generator, the grid generator and fusion module. Then we perform analysis on our pose vector decomposition and cascaded refinement strategy. We perform a qualitative and quantitative comparison for each of them in Fig. 6 and Table 1.

Face Stage. The face stage is trained to transform the input anime image to a target facial expression (*e.g.*, close eyes and open mouth) according to the target pose vector. Fig. 6 shows the mask A, the content C and the final generated result I_{y_g}. Note that the mask generator has learned to focus on the dynamic areas (darker areas) according to the target pose vector. Mask A can well locate the eyes and mouth in the input image, and cover them with gray pixels, which means that these areas need to be obtained from content C, and other white areas mean that they need to be obtained from the input image. In the content C, only the pixels related to the dynamic regions of the eyes and mouth will be carefully generated, the rest are only noise.

Head Stage. The task of the head stage is to complete the movement of the anime character head, including swinging up and down, swinging left and right. Compared with the change of face pose during the face stage, it is obviously more difficult to realize the change of head pose, because most of the regions in the input image are still for the change of face pose, only the eyes and mouth need to be changed, while the head movement requires large areas of change, especially for large-scale target pose vectors. Below we will evaluate the role of each module in the head stage. Some visual results are shown in Fig. 7. "Mask" means we only use the single-stage mask generator, "Mask + MS" means we use the mask generator and pose vector decomposition strategy, "Grid" means we use the grid generator only, "Mask + MS + Grid" means the complete model of head stage.

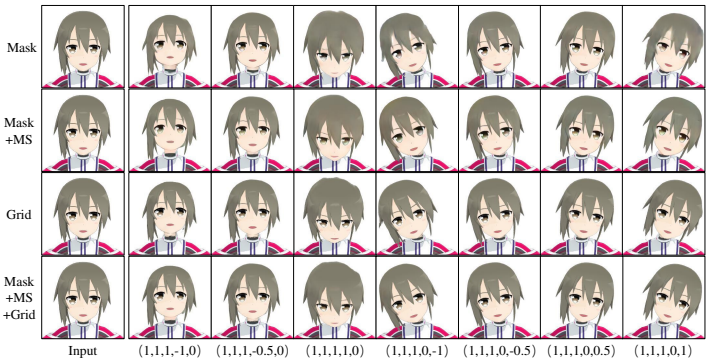

Fig. 7. Head stage. The ablation study of head stage. The first column is the input, while the rest columns are generated images corresponding to the pose vectors on the bottom row. Each row corresponds to a variant of our model. "Mask" means we only use the single-stage mask generator, "Mask + MS" means we use the mask generator and pose vector decomposition strategy, "Grid" means we use the grid generator only, "Mask + MS + Grid" means the complete model of head stage.

The results of "Mask" show the limitation of the single-stage mask generator. It can only handle small-scale head movements , for large-scale head movements, the generation results become very blurry, especially in the hair part. The reason is that large-scale head movement will cause the generated image cannot use the mask to obtain pixels from the corresponding position of the input image, the entire head can only rely on the generator to generate the corresponding content, so the advantage of the mask is basically useless here.

As the results of "Mask + MS" show, when we adopt pose vector decomposition and cascaded refinement strategy for the mask generator, significant improvement can be achieved. Due to the decomposition of the pose vector, the mask generator only needs to perform a small-scale transformation at a time, so that the mask can be used to obtain and copy pixels from the same position in the input image, reducing the blur and improving the quality of the generated image. Also, small-scale transformations reduce the difficulty of learning process. But it still cannot preserve the color identification of the input image.

From the results of "Grid", we can observe that it can perform well in the head movement of anime characters. The final output image is obtained by performing bilinear sampling on input image, so the generated image has several advantages: 1) It eases the blur caused by the direct output image of the deep convolution network [29]. 2) It can preserve the color identification of the input image. However, it can only copy pixels from the relevant areas in the input image, so for some areas that have no relevant information in the input image, such as the neck, messy pixels will appear.

The complete model "Mask + MS + Grid" means we adopt pose vector decomposition and cascaded refinement strategy, and fuse the results of mask

generator and grid generator. The visualization demonstrate that the fusion results can not only retain the advantages of each branch, but also alleviate the problems to a certain extent, which leads to more fine-grained results.

4.3 Qualitative Experimental Results

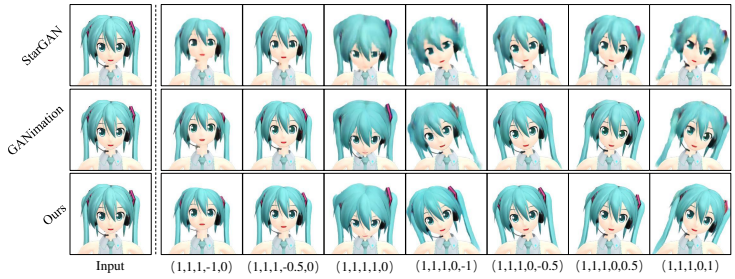

Fig. 8. Qualitative comparison with state-of-the-art. The results of StarGAN, GANimation, and ours. Each column represents a pose corresponding to the pose vector on the bottom row. Our method can generate much clearer results.

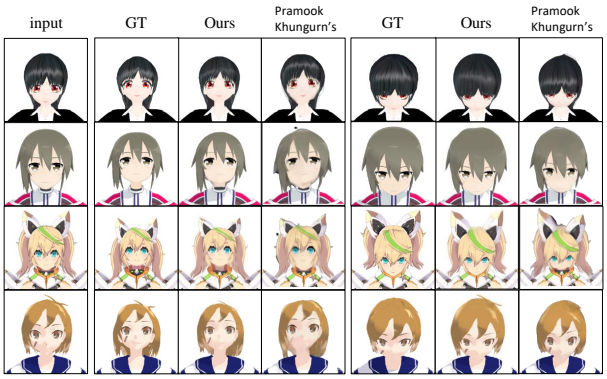

Fig. 9. Qualitative comparison with Pramook's. The results of ours and Pramook's. Each anime input corresponds to two generated poses. Our method can get better results.

Fig. 8 shows qualitative experimental results. The first column is the input image, while each remaining column corresponds to a target pose vector. We

compare our method with the state-of-the-art facial expression editing methods: StarGAN [2] and GANimation [7].

As shown in Fig. 8, StarGAN and GANimation are prone to generate blurs and artifacts, especially for the large-scale target pose vectors. Instead, our method generates more realistic images with much less blurs and artifacts. Even for the large-scale target pose vectors, our method can still perform well. This is contribute to the pose vector decomposition and cascaded refinement strategy, which is designed to performs anime-like progressive pose transformation rather than a single-step one.

Fig. 9 shows the comparison of our and Pramook's method [9]. Due to our pose vector decomposition and cascaded refinement strategy, we can get better results for the large-scale transform.

Table 1. Pose transformation comparison with other methods and our variants.

Method	L1 ↓	RMSE ↓	SSIM ↑
StarGAN [2]	0.0670	0.1974	0.7680
GANimation [7]	0.0611	0.1921	0.7878
Ours(mask)	0.0600	0.1885	0.7899
Ours(grid)	0.0545	0.1867	0.8069
Ours(mask+grid)	0.0541	0.1855	0.8072
Ours(mask+grid+ms)	**0.0525**	**0.1800**	**0.8101**

Table 2. Pose transformation comparison with Pramook's and our method.

Method	L1 ↓	RMSE ↓	SSIM ↑
Pramook's [9]	0.0664	0.2481	0.8022
Ours	**0.0401**	**0.1434**	**0.8417**

4.4 Quantitative Experimental Results

We use L1 norm, RMSE, and SSIM similarity [30] for quantitative evaluations. For the L1 norm and RMSE, the lower the score, the smaller the distance between the generated image and ground truth, in contrast, for SSIM, the higher the score, the greater the similarity between the generated image and ground truth.

As shown in Table 1, "Ours(mask)" means we only use the single stage mask generator, "Ours(grid)" means we use the grid generator only, "Ours(mask+grid)" means we use fusion to fuse the results of mask generator and grid generator, "Ours(mask+grid+ms)" means the complete model, including the pose vector decomposition and cascaded refinement strategy. Compared to other two methods, our method obtains higher SSIM score and lower L1, RMSE scores.

It is worth mention that Pramook's method requires RGBA format images, while images in our dataset are RGB format. So we chose about 100 models from our testing set and manually made them into RGBA format, then we use his open source weights and test code to obtain quantitative results on our dataset. As shown in Table 2, the calculated L1 and RMSE results are much higher than our method. For possible reasons, first we cannot refine the RGBA images to be consistent with his training dataset, because he has his own set of processing code and is not open released, while we can only make them manually. Besides, his method needs the character's head in the 128×128 position in the center of the image, while our dataset and method does not have this constraint. In addition, our method can handle larger motion of head than his.

4.5 Generalization and Yaw Rotation Results

Fig. 10. Generality and Yaw Rotation. The upper part is the generalization results of our model, where input anime images in the first column are downloaded from the Internet. The bottom is the yaw rotation results, our model can handle it as well.

The upper part of Fig. 10 demonstrates the generality of our model, where the input images are downloaded from the Internet, our model shows good generality. For yaw rotation, it is essentially the same as pitch and roll, our model can handle it as well, two examples are shown in the bottom of Fig. 10.

5 Conclusion

In this work, we present a novel cascade pose transform network for talking head animation. Different from previous methods, our approach is capable of generating vivid anime talking heads from a single anime image. Extensive experiments on our anime dataset validate the effectiveness of our approach. In the future, we plan to improve the generalization ability of the model and increase the resolution of the generated image.

6 Acknowledge

This work was funded by the DigiX Joint Innovation Center of Huawei-HUST.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the Advances in neural information processing systems. (2014) 2672–2680
2. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 8789–8797
3. Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., Wen, S.: Stgan: A unified selective transfer network for arbitrary image attribute editing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3673–3682
4. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing **28** (2019) 5464–5478
5. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H., Hawk, S.T., Van Knippenberg, A.: Presentation and validation of the radboud faces database. Cognition and emotion **24** (2010) 1377–1388
6. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. (2015) 3730–3738
7. Pumarola, A., Agudo, A., Martinez, A.M., Sanfeliu, A., Moreno-Noguer, F.: Ganimation: Anatomically-aware facial animation from a single image. In: Proceedings of the European Conference on Computer Vision. (2018) 818–833
8. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: Proceedings of the European conference on computer vision, Springer (2016) 286–301
9. Khungurn, P.: Talking head anime from a single image. https://pkhungurn.github.io/talking-head-anime/ (2019)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1125–1134
11. Zhao, Y., Wu, R., Dong, H.: Unpaired image-to-image translation using adversarial consistency loss. arXiv preprint arXiv:2003.04858 (2020)
12. Chen, Y., Lai, Y.K., Liu, Y.J.: Cartoongan: Generative adversarial networks for photo cartoonization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 9465–9474
13. Tomei, M., Cornia, M., Baraldi, L., Cucchiara, R.: Art2real: unfolding the reality of artworks via semantically-aware image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5849–5859
14. Shen, F., Yan, S., Zeng, G.: Neural style transfer via meta networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8061–8069
15. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9459–9468
16. Liu, W., Piao, Z., Min, J., Luo, W., Ma, L., Gao, S.: Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 5904–5913

17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 4401–4410
18. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4681–4690
19. Athar, S., Shu, Z., Samaras, D.: Self-supervised deformation modeling for facial expression editing. arXiv preprint arXiv:1911.00735 (2019)
20. Wu, R., Zhang, G., Lu, S., Chen, T.: Cascade ef-gan: Progressive facial expression editing with local focuses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2020) 5021–5030
21. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org (2017) 2642–2651
22. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 (2016)
23. Wu, R., Gu, X., Tao, X., Shen, X., Tai, Y.W., Jia, J.: Landmark assisted cyclegan for cartoon face generation. arXiv preprint arXiv:1907.01424 (2019)
24. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 2223–2232
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
26. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the European conference on computer vision, Springer (2016) 694–711
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13** (2004) 600–612