# 3D face model reconstruction combine with cycle-GANl

*Shupei Zhang, Tianyu Lang, Chenge Liu*

## I. Introduction

3D face reconstruction from a single image or a series of images of a video can be very challenging due to the variance in lighting conditions and camera angles. The goal is to reconstruct a detailed 3D face model with textures from limited 2D data. Despite its difficulty, it can be very useful in video compression, animation, game development, augmented reality and motion capture.

Previous works try to solve this problem by doing parameter regression on a parametric face model (3DMM, 3D Morphable Model)[1], but the limited 3D face data used to generate the 3DMM restricts those methods to generalize very well. Face models produced by those methods might lack some facial details, especially when there is a lot of facial hairs in the 2D image. Some researches try to tackle this problem by segmenting different facial regions and conduct optimization separately. But still they are restricted to the pre-defined sub-space. Another problem with this approach is that they require huge amount of training data to perform 3DMM parameter regression.

More recent works are utilizing weak-supervision or self-supervision techniques [13][12] [15]. Data augmentation and generative models reduce the amount of data needed to train the models. Some researchers develop unique data path in their models to use data from other domains like 2D face landmark data. Some other works extend 3DMM model to improve the ability to generalize.

We combine the Cycle-GAN method[16]with uv-map representation. Cycle-GAN is a successful neural network which has a good performance on image editing and image style transfer problems. But to our best knowledge, no researcher has tried this method on 3D face model reconstruction. So we want to see what will happen if we combine those two methods.

We propose to develop a generative model that doesn't need huge amount of data to train. It will be made up of two parts: 2D image to 3D model network and 3D model to 2D image network. We will use generative adversarial mechanism. The first part will generate realistic 3D face model from 2D images, and the second part will make sure that the 3D face model is a good reconstruction.

## II. Review of Existing Work

Our literature review has two parts. The first part is summarizing the existing algorithm for the reconstruction of 3D face models from images and videos. The second part is a review for cycle GAN. During our research, we did not see any existing method by use cycle GAN to reconstruct 3D face models from single images. Therefore, we want to see what will happen if we combine cycle GAN and 3D face model reconstruction. We accept this method will have more accuracy and good average performance on different data sets.

### Review of 3D face models reconstruction

During our research, we discovered many methods for 3D face reconstruction. All those algorithms have a common target which is reconstructing the 3D face model based on the input information, such as input image, video and head position. And we can use a point cloud to represent this model. And by extracting different features from the input image, we can construct a 3D face model to show the shape, depth, and pose of the face in the original image. Therefore, how to predict and show the shape, depth and pose is the most important part to solve in the reconstruct 3D face model. There are three kinds of methods for solving this problem. The traditional method, 3D face reconstruction method based on the model and end-to-end 3D face reconstruction method. For the traditional method, most of them are based on image information, such as 3D face reconstruction based on one or more information modelling techniques such as image brightness, edge information, linear perspective, colour, relative height, parallax, etc. However, the overall performance of those traditional methods is not as good as the other two kinds of methods. Therefore, we will pay more attention to the model-based method and end-to-end method in the literature review.

#### *A. Traditional face model reconstruction algorithm*

To solve this problem, at the beginning we find people trying to create a model and define some fiducial points in this model based on the input view. And change the face model by those fiducial points to have an animation of facial expressions. In 1997, Yin and Basu [9] came up with a new method to do this process automatically which called the layered force spreading method (LFSM). They use two views of the human's face to extract fiducial points, those points include eyes, nose, mouth, face shape etc. Then generate a generic 3D face model by two 2D face model of the front view and side view. For the animation part, they use connected multiple layers to control those facial points in this 3D face model. The first layer connects to those facial points directly and the second layer connects to the first layer and so on. Those layers are assigned a different weight of force so the force of controlling animation can express layer by layer. The LFSM method provides a more accurate and realistic performance of

3D face animation. And this method also can use as a way to compression remote meeting videos by sending the motion vectors or matrix. However, because of the computational ability limitation of that time, it is difficult to set a large number of fiducial points and represent those points in the generic face model. The less number of fiducial points will affect the final performance of the animation. This method still provides a basement of many current algorithms nowadays.

Even though the method using fiducial points and change them in a generic model has a satisfying result but this kind of method may cause a reconstruction face model that looks like the generic model rather than the specific face needed for reconstruction. Therefore, in 2002, Roy-Chowdhury et al. [4] came up with a 3D face SFM ( structure from motion ) model combined with a generic face model. At the first, they use the SFM algorithm to obtain an estimated model, because this method reconstructs the model from video data, the quality of the model is not good and there will be many errors in the local region. Then they put those regions into a generic model as help to reconstruct the generic model by using an optimization function. This function is defined by the differences between the 3D estimation and generic model. The final target is to reduce this function which will fix the errors in the original SFM estimation model. And this method helps to solve the problem of the generic model which is too general for a specific person's facial reconstruction. However, the best way to solve this problem is to reconstruct a model of this person from the beginning to the end. Since the dataset is not really large at that time, they can not find enough data for one specific person to reconstruct. We will introduce more in the end-to-end algorithm part.

### *B. 3D face reconstruction method based on the model*

In recent years, deep learning and neural network have had high improvements, people find its good probability to find the parameters and relationship in the fiducial points of the input image and output 3D face model. Therefore many algorithms based on deep learning have occurred. And those algorithms show a very good performance in control features of a generic model. And many accurate generic models have been produced due to the huge amount of training dataset.

First of all, many model-based rely on a generic model called 3DMM (3D Morphable Model), and they use a neural network to calculate the perimeter of this model to represent a specific human's face model. The original 3DMM was come up in 2002 by Volker Blanz and Thomas Vetter[2]. The 3D deformable model is based on the 3D face database, and takes face shape and face texture statistics as constraints, and takes the influence of face attitude and lighting factors into consideration, so the generated 3D face model has high accuracy. 3DMM is a linear combination of the data object model database, on the basis of above 3D face said, suppose we establish the deformation of 3D facial models is composed of m face models, we can output the final 3DMM by using the average of those model. And we can change the weight of different face models to change the final output performance.

In 2017, Xiangyu Zhu etc.[17] came up with a face alignment method for large poses. This method hires CNN to solve this problem. They use a model-view feature called Pose Adaptive Feature (PAF) and an image-view feature called Projected Normalized Coordinate Code (PNCC). And they combine those two features to achieve the optimal solution. For the network, they use an initial parameter p, they construct PNCC and PAF with p and train a two-stream CNN Netk to conduct fitting. The outputs of the two streams are merged with an additional fully connected layer to predict. In the PNCC stream, the input is the $200 \times 200 \times 3$ colour image stacked by the $200 \times 200 \times 3$ PNCC. In the PAF stream, the input is the $200 \times 200 \times 3$ colour image and $64 \times 64$ feature anchors. And they also combine several loss functions to fit the 3DMM to the input image which is called Optimized Weighted Parameter Distance Cost (OWPDC). This loss function is used to reduce the difference in the controlling parameters of the input image and 3DMM. Because of the calculation ability of the neural network and huge training dataset, this method has a good performance for both small pose and large pose 3D face alignment.

Because the 3DMM is used as a liner base representation, it will limit the performance when there are some nonlinear features in the face model. Therefore, in 2018, Luan Tran and Xiaoming Liu [14]came up with a nonlinear 3DMM. The work is based on the encoder-decoder model. The encoder learns the projection parameters, shape and texture parameters by input image. The decoder can learn 3D shape and texture directly by shape and texture parameters, so it can be regarded as a nonlinear deformable model of a 3D face. Then, based on the Z-Buffer algorithm, the rendering layer uses projection parameters and 3D shapes and textures to render the 3D model into a 2D image. The goal of the model is to minimize the pixel-level difference between the 2D projection of the 3D face and the input image. In order to make the generated face more real, the author introduced patchGAN to learn high-quality texture and local features and used the loss function related to the alignment of feature signs to adjust the encoder. The experimental results show that the decoder, as a nonlinear deformation model, has a more powerful representation ability and can reconstruct more face details.

After two years, in 2020, Jianzhu Guo's team [6] came up with a new light-weighted CNN to achieve fast, accurate and stable 3D dense face alignment. They use a lightweight backbone like MobileNet for predicting 3DMM parameters at first. This is a small set of 3DMM parameters in order to reduce the training and process time. They also hire Vertex Distance Cost (VDC) and Weighted Parameter Distance Cost (WPDC) to meta joint optimization. After that, they treat Landmark-regression Regularization as a task-level regularization to add an additional landmark-regression task on the global pooling layer, trained by L2 loss. For the training process, they use a batch-level 3D aided short-video synthesis strategy, which expands one still image to several adjacent frames, forming a short synthetic video in a mini-batch. They put different kinds of noise and rotation to a single input image to make several frames. Then put them together to make up a short video. By this training method, it increases the performance on consistency. The structure is shown in figure 1.

### *C. End-to-end 3D face reconstruction method*

Although 3DMM has a really good performance, it is accurate enough and easy to use. There's a problem with that. Because there is so much variation in the world's faces, 3DMM needs to integrate a lot of face information to store all deviations from the average face. As a result, this model will be too average to represent humans' faces in different races and ages. To solve this problem, a new method occurs. End-to-end 3D face reconstruction is a new method in recent years. They bypass face models like 3DMM, design their own 3D face representation method, and use CNN structure for direct regression to reconstruct 3D face end-to-end.

The acquisition cost of real data sets in 3D face reconstruction is very high, so researchers often do research based on a small amount of data or simulation data, and the generalization ability of the trained model will be limited. Self-supervised learning is an important idea to solve this problem. And our method also tries to do self-supervised learning to reduce the amount of training data we need for our neural network. There are some self-supervised learning methods used to reconstruct the 3D face model. In 2017, Ayush Tewari etc. [10] came up with a new network called MoFA (Model-based Deep Convolutional Face Autoencoder). This network will construct a 3D face model and then use a decoder to get a 2D image based on this 3D face model. The input is firstly extracted with a deep encoder to obtain the semantically related coefficients, which include face attitude, shape, expression, skin, scene illumination and other information. Then, this coefficient is input into a model-based decoder to achieve the projection of the 3D model to a 2D image. The model can use the 3DMM model. The final loss is based on the pixel loss of the reconstructed image and the input image. It can also add key point losses, with coefficient regularization losses as constraints. And because the input and the output are all images, it does not need a corresponding 3D face model to the input image. By using this method, we can reduce the need for training data.

In 2018, Yao's team[5] came up with a new method for reconstructing 3D face modelling by using a Position Map Regression Network (PRN). Rather than use a model-based method or really deep neural network, they use a model-free and light-weighted network to achieve this goal. Firstly, they use a UV position map as the presentation of a full 3D facial structure, this map matches the 3D ground truth model into a 2D input image. This representation is able to simultaneously obtain the 3D facial structure and dense alignment result by using a CNN to regress the position map directly from unconstrained 2D images. So it makes CNN have fewer layers and easy to train. And for the CNN part, they use an encoder-decoder structure to learn the transfer function. The encoder part reduces the $256 \times 256 \times 3$ input image into $8 \times 8 \times 512$ feature maps, and the decoder part generates the predicted $256 \times 256 \times 3$ position map. For the loss function, they choose to use Mean square error (MSE). And they also apply a weight mask for their loss function since the central part of the face is more important than other parts. It will concentrate the training process on the eye, nose and mouth. They use some datasets containing both 2D face images and their corresponding 3D point clouds with semantic meaning to train their neural network. During the training, they apply some rotation and scale colour channels to make this network can work on some extreme conditions input image. This method can reconstruct a 3D face modelling form a 2D image, the output of this network is a point cloud. And because this network can both finish reconstruction and alignment work, so the time of running this network is more quickly than the other same functional network.

In 2019, Ayush Tewari's team [11]came up with a face model that is learned from the video. Those videos are downloaded from Youtube and those are all related to interviews or speeches of celebrities. This dataset contains over 140k videos of over 6000 celebrities crawled from Youtube. (VoxCeleb2). For now, much research chooses to use 3DMM to reconstruct a 3D face model, but it brings a problem which is 3DMM is training by a specific dataset, it will make the final result too average and it is hard to fit for people in a different race. However, if we train our model from the beginning, it needs lots of data. This paper shows if we use much low-quality data we can also reach a state of the art performance. Those videos data easily get and

we can find almost unlimited data from the Internet to train our network. For the design of the network, first, their identity model is represented by a deformation graph. The whole network is consistent with M same networks. The inputs of those networks are the same person in a different position or expression. They use a convolutional network to extract the low-level features of the input image. Then they use a shared identity network to estimate the common parameters of M frames. They also apply two additional convolution layers to extract medium-level features. The resulting M medium-level feature maps are fused into a single multi-frame feature map via average pooling. It ensures the accuracy of the face model in different positions and expressions. In summary, they use a differentiable mesh deformation layer in combination with a differentiable face renderer to implement a model-based face autoencoder.

### REVIEWS FOR CYCLE GAN

After doing much research on 3D face modelling reconstruction, we find there is no one to use cycle GAN in this part. And we already see this good performance of cycle GAN in changing the style of image and create a high-quality image. We are trying to use cycle GAN in this area to see if it can also have a good performance.

In 2017, Jun-Yan Zhu's team [16] came up with a new method for doing GAN ( Generative adversarial networks), called Cycle-Consistent Adversarial Networks. General GAN is oriented towards the data of one domain. The generator G tries to generate the data close to the real data as far as possible, while the discriminator D tries to distinguish the real data from the generated data of the domain as far as possible. The two have been playing games, in which the generator gradually gained the upper hand, and finally, the generated data is no different from the data of the domain. And for cycle GAN, it can be regarded as the fusion of two GAN. One GAN is composed of generator G and discriminator DY, which can realize image generation and discrimination from the X domain to the Y domain. The other GAN is composed of generator F and discriminator DX, which can realize image generation and discrimination from the Y domain to the X domain. The two networks constitute the process of the cycle. The detailed structure is shown in figure 2.

In addition to the classic basic GAN network against Loss, the part of Loss also puts forward a cycle-loss. Because the network needs to ensure that the generated image retains the characteristics of the original image, if we use the generator Genratora-B to generate A false image, we need to be able to use another generator, Generator-a, to try to restore the original image. This process must satisfy cyclic consistency.

The overall optimization goal of CycleGAN is shown in the following equation.

$$\begin{aligned} L(G, F, DX, DY) &= L_{GAN}(G, DY, X, Y) \\ &+ L_{GAN}(F, DX, Y, X) + \lambda L_{cyc}(G, F) \end{aligned} \tag{1}$$

This equation consists of three parts. The first part, LGAN(G, DY, X, Y), represents the optimization goal of generator G and discriminator DY.In the second part, LGAN(F, DX, Y, X) represents the optimization objectives of generator F and discriminator DX, and these two parts are the optimization objectives of the GAN algorithm itself. In the third part, Lcyc(G, F) represents cycle consistency loss, which is used to constrain the consistency of the image when it is transformed from the transform domain to the original domain.

So we are thinking about using a single image as the input for the first generator to generate a 3D face model. We will use some pre-trained technology in this part to reduce the training time. And then using the first discriminator to compare the output model with the ground truth model. After that, the second generator will project the 3D face model into 2D space as an image, send this result to the second discriminatory to compare it with the input image. After training, the accuracy of generators and discriminators will be improved and the first generator will give a perfect result at the end.

## III. OUR PLAN

We plan to approach this using end-to-end GAN. At first, we want to reduce the need for training data. According to what I have introduced before, the collection of data of face photo and its corresponding model is really expensive. Therefore we want to develop a method which only needs little data to start training and then it can only rely on a single image to training by itself. We are trying to reconstruct an existing end-to-end method to generate a 3D face model for the first part of our cycle GAN. And we can use a traditional or neural network method to project the 3D face model to a 2D image as our second generator. We will design a new feature representation and loss function to reduce the training time and come up with a light-weighted weak supervised network. After doing this, if we have more time, we are thinking to do some video compression based on the 3D face model reconstruction. For example, we can use the reconstruction 3D face model to replace the person's face in the video. And we only need to send some motion vectors of this model to show video content.

## IV. METHODOLOGY

In this part, I will have three parts to introduce our method. In the first part, I will introduce how do we process the training data and use it for the training process. In the second part, I will show you the structure of our

neural network and explain each layers' relationship. In the last part, I will show the loss function of our network, and explain how does this function helps us to train the network. For now, we are almost done with the structure and training part of our network, then we will do some tests and evaluations for our method.

### A. Process the training dataset

Most of the data obtained from the Internet is not available out of the box, which means that we need to do some preliminary collation of the data, such as counting the amount of data, removing unnecessary files, necessary format conversion, and so on. For the training data, we hire a database called 300WLP. 300W Released in 2013, it contains 300 indoor pictures and 300 outdoor pictures, among which the expressions, lighting conditions, posture, occlusion and face size of the data set vary greatly, and it is collected through Google search for party, conference and other difficult scenes. The data set is marked with 68 key points, And the 300WLP dataset is the synthesized large-pose face images from 300W. Because we want to use this dataset to both trains the accuracy and robustness of our network. The large-pose 3D face model reconstruction is always the most difficult problem in this area because of the lack of information for the whole face. We hope our method can have a good enough performance when face to the large-pose face image. Through the BFM shape model and expression model, the image coordinates of the final 3D point cloud (53,215 in total) can be obtained. Each point has three coordinates of X, Y and Z, with a total of 53,215X3 values. The 68 points x and Y coordinates of these point clouds are commonly used key points of 68 human faces, and the x and Y coordinates of about 40k points are key points of dense faces. Those 53215x3 values are represented by a 3-channel 256x256 image which is a UV position map. We will generate the ground truth UV position map of images in the 300WLP dataset. First of all, load the image and fitted parameters to the 3DMM, after that we will generate a mesh and transform it into the right position according to the original image. In this step, we use 3DFFA to help us finish the transformation part. Then we will crop the image by key points, because the original image may include noisy background or multiple human faces. So we crop the image based on the ground truth key points to ensure we are concentrating on the face which has been labelled. After that, we can generate a UV position map by the ground truth landmark. Therefore, after processing the dataset, we can get a new dataset that has a cropped image and a corresponding ground truth UV position map. Then I will introduce how do we design this network to have a good fit with this training data set.

### B. The structure of network

Now, I will introduce the main structure of our neural network. We combine the cycle-GAN method with RPnet. Cycle-GAN is a successful neural network which has a good performance on image edit and image style change problem, but no one tries this method on 3D face model reconstruction area, So we want to see what will happen if we combine those two methods. Our first generator is the reconstruction of PRnet[5], this net is an encoder-decoder structure. It consists of a convolutional layer, 10 block residual blocks and 17 decoder layers. Among them, the convolution of each middle layer and the convolutional convolution is followed by Batchnorm and Relu. The activation of the last layer of the convolutional convolution adopts Sigmoid. Kernel of each layer in the encoder is 4, Out Channel is from 16 to 512, the output size is from 256 to 8(the size of every two blocks is halved), the decoder is vice versa. The input and output are both 256x256 RGB images, where the input is face image and the output is 3D point cloud coordinates (65,536 in total). Finally, 2d and 3D coordinates of corresponding points are lifted respectively to carry out 3D reconstruction or face key points. And then after getting the UV map result of the first generator, we will put it into our first discriminator. We use Resnet 18 [7] as our discriminator, it consists of one Ordinary convolutional layer then follows by Four layers of ResNet, each layer contains two blocks, each Block has two convolutional layers. Then we put a fully connected layer follows by a Pooling Layer in the end. This network will accept a UV map as input and output a percentage. This percentage will show the similarity between our generate UV map and the ground truth UV map.

Then for the second part of our network, we are working on the generator as a method to extract the 3D facial landmark from the input image. The 3D facial landmark is 68 key points in the human face based on the face alignment. [3] And the second discriminator will judge how different between the facial landmark we generated and the ground truth facial landmark of the input image. This structure will help our network learn about the texture and shape of the 3D face model.

### C. The loss function

At first, we will do some pre-trained process of our network. We use the original loss function of the PRnet to do the pre-trained process. The starting point of this loss function design is to compare the output of the computational network with the ground-truth position map.MSE(Mean Square Error) is a commonly used loss function, but MSE treats all points equally. It is generally believed that the middle part of the face has more distinct

features compared with other regions. Weight Mask is introduced to add to the calculation of the loss function.

$$Loss = \sum |p(x,y) - p_g(x,y)| * W(x,y) \quad (2)$$

P (x,y) represents the position map output by the network, the second p(x,y) represents the ground truth of position map and W(x,y) stands for weight mask and is set to 16:4:3:0.

After doing some pre-training process, we will use our discriminator to help us train. For the loss of the generator, we will calculate the difference between 1 and the percentage given by the discriminator. Then we will forward this loss to our generator to help it train better. And for the discriminator part, we will calculate the difference between 1 and the percentage given by input the original image. Then will forward the loss to the discriminator to help it train. And for the second part, we will calculate the difference between the original image and the 2d image of our model. The main purpose of the training is to make the generator can output a good enough UV map which can pass the discriminator's check. At the same time, the discriminator also improves its ability to check the reality of the output UV map from the generator.

## V. Labs

- Lab 1: Replicate an end-to-end 3D face model reconstructionmethod.
- Lab 2: Project 3D face model into a 2D image.
- Lab 3: Data pre-processing and augmentation.
- Lab 4: Model design and implementation.
- Lab 5: Model tuning.

In lab 1, we will find an open-source method to reconstruct a 3D face model, based on this model we will design a new representation method and loss function to try to reduce the training time of the network. In lab 2, we will come up with a method to extract the feature of the 3D face model and project it to a 2D image. In lab 3, we will process our training data, because we want to use in-the-wild images to train our network, so we need to do some optimization of those low-quality images. In lab 4 and 5, we will train our network and do some experiments to see the results of our method. Most labs will hold in November, and the experiments part will start in December.

## VI. Implementation

During the implementation, we have tried several methods in a different part. After doing some experiments we finally choose the method which has the best performance during the experiment. We have two generators and two discriminators in our network, you can the main structure of our network in fig.3. At first, we implement a tool to automatically process the database into the format we can use for training. We used the ground truth 3DMM parameters to generate 3D face models. And then convert the 3D coordinates to UV-maps. Each pixel in the original image will have a corresponding pixel in the UV-map at the same location, with the RGB channel being the XYZ coordinates. After processing, we will have an input image and its corresponding ground-truth UV position map. Then we implemented the first generator of our network. For this part, We have tried 2 possible approaches for this part. The first one uses UV map representation of the 3D face model and the second one will use the voxel representation of the 3D face model. Based on the performance of those two methods, we decide to use the first method. We reconstruct the PRnet in a PyTorch version and design a new loss function and training process for it. The loss function is the MSE function which I mentioned before. Before we train the whole network, we also write a pre-trained code because we find that although PRnet has a simple structure, it is difficult to train. If we train the whole network without doing any pre-train process, the performance of training is terrible. Therefore, we use a part of our processed database to do this pre-train process. We also use a weighted mask to assign different weights to different parts of the human face. For example, we will pay more attention to the area around the eye and nose compared to the area surrounding the neck. The experiment shows that this structure and loss function can have a good result. For the second generator of our network, We also have tried two methods in this part. In the first method, we want to generate a 2D image using the 3D model we have, then use a pixel to pixel loss to train our network. The second method is we will get the ground truth 3D landmarks from the input image and compare it with the 3D landmarks generated from our UV-map. For the first method, We want to compare the original 2D image with a 2D image from the generated 3D mesh. So we implement a function that can convert obj file to 2D image without any third-party tool during training. However, there is a problem that We try to make the loss of the 2d images be differentiable for backpropagation but we still not solve this problem. So the result of the first method is not as good as expecting. After this failure, we choose the second method. In the second generator, we use the 3D facial landmark regressed from the UV position map generated by our first generator. We can simply extract it because 68 key points of the 3D landmark in the human face should be in the same location as the UV position map. After that, we can also get the ground-truth 3D facial landmark from the ground-truth UV position map. Then we send those two sets of 3D facial landmarks to our second generator to compare the difference. The result shows after applying

this method, we can have a better shape of the 3D face model and head position estimation. And also, it should be helpful for getting and rendering the texture of our 3D face model because the landmark also provides reliable face alignment information. For two discriminators, we have tried many different network structures to have a good performance. In the end, we decide to use Resnet to handle this job which I mentioned in the former part.

## VII. Result

According to what I have introduced before, our network is light-weighted and easy to train. Based on this feature, we also used a face detector [8] before reconstructing the face model. This allows us to deal with arbitrary numbers of faces in input images. Multiple face models can be reconstructed at the same time. This method allows a stable and quick reconstruction of multiple human faces in a single image. It will automatically get the positions of different persons in the picture and save this information for rendering uses. For the 3D face model reconstruction part, we can get a detailed 3D face model that has the same head position as the input image. You can see the result in fig.4. And for the side view problem, our method also shows a good robust, even though we do not have a whole face image, our network can also output a 3D model for the whole face. The result can see in fig.5. The other interesting result is in fig.6. Our network can also out a face model for a cartoon character, it is quite interesting because this result can be used in the animation industry to have a quick change from a 2D cartoon to a 3D movie. Unfortunately, we do not have more time to do the experiment on another testing database to evaluate our method, but I believe that this method should get a good enough performance in different databases.

## VIII. Conclution

In conclusion, we come up with a new method of 3D face model reconstruction by combining the cycle-GAN idea. And the result of our network looks good enough. And the most important point is our method reduces the amount of training data needed by using an unsupervised learning algorithm. The data of the ground-truth 3D face model is really hard to collect, so it is meaningful to use fewer data to train a good performance 3D face model reconstruction network. Based on this research, we also find the potential ability of the cycle-GAN idea to process the 3D face reconstruction problem. It can improve the shape of the reconstruction 3D model and head position estimation. And also, it should be helpful for the texture extract process. The 3D facial landmark will save the space information of the 3D model rather than only use the 2D landmark to represent the position information of a face image. In addition, our network should have a better performance with training on a larger database. During the experiment, we found that the result of the pre-trained process decides the overall performance of our method. Therefore, if we can have a better pre-trained model, we should have a state-of-art result. I think the more important problem in this area is how to use the reconstructed 3D face model to produce an animation. And we can use this model and animation to compress the video by rendering it to the original video. I am new to this area and have met many difficulties at the beginning so I do not have enough time to do more research on the animation part. I will do it in the future to see how can this method help people's real life.

## IX. Additional image

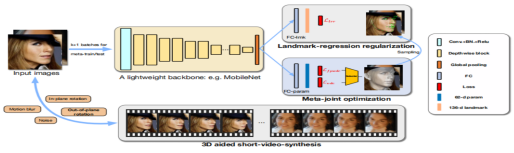


Fig. 1. The architecture of the deep neural network used in Jianzhu' paper and the process of training network.

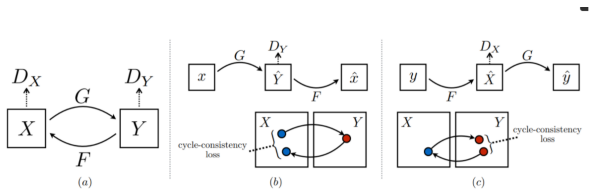


Fig. 2. The architecture of cycle GAN

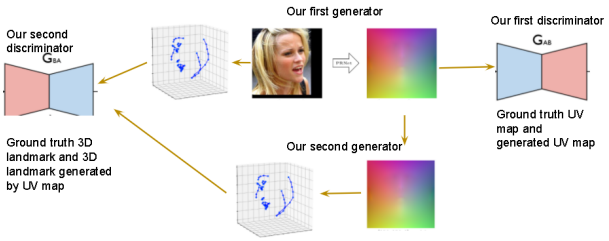


Fig. 3. The main architecture of ournetwork

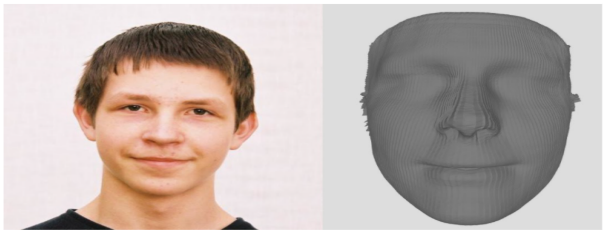

Fig. 4. The result of a front-view human face

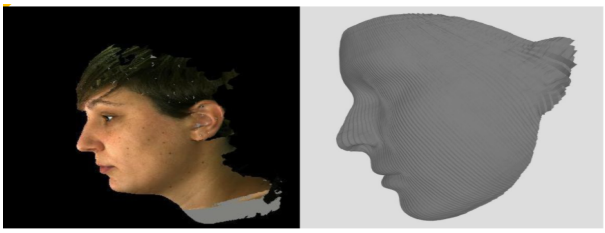

Fig. 5. The result of a side-view human face

Fig. 6. Theresult of a cartoon character

## REFERENCES


[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '99*, pages 187–194, Not Known, 1999. ACM Press.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. *SIGGRAPH'99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 09 2002.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[4] A. R. Chowdhury, R. Chellappa, S. Krishnamurthy, and T. Vo. 3d face reconstruction from video using a generic model. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 449–452 vol.1, Aug 2002.

[5] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 557–574, Cham, 2018. Springer International Publishing.

[6] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z. Li. Towards Fast, Accurate and Stable 3D Dense Face Alignment. *arXiv e-prints*, page arXiv:2009.09960, September 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[9] Lijun Yin and A. Basu. Mpeg4 face modeling using fiducial points. In *Proceedings of International Conference on Image Processing*, volume 1, pages 109–112 vol.1, Oct 1997.

[10] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3735–3744, 2017.

[11] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Fml: Face model learning from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10812–10822, 2019.

[12] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-Supervised Multi-Level Face Model Learning for Monocular Reconstruction at Over 250 Hz. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2549–2559, 2018.

[13] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. MoFA: Model-Based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017.

[14] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Xiaoguang Tu, Jian Zhao, Mei Xie, Zihang Jiang, Akshaya Balamurugan, Yao Luo, Yang Zhao, Lingxiao He, Zheng Ma, and Jiashi Feng. 3D Face Reconstruction from A Single Image Assisted by 2D Face Images in the Wild. *IEEE Transactions on Multimedia*, pages 1–1, 2020.

[16] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[17] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face Alignment Across Large Poses: A 3D Solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):78–92, January 2019.