Generative adversarial networks for generating RGB-D videos

Yuki Nakahira* and Kazuhiko Kawamoto*
* Chiba University, Chiba, Japan
E-mail: nakahira@chiba-u.jp, kawa@faculty.chiba-u.jp

***Abstract*—Generative adversarial networks(GANs) have been successfully applied for generating high quality natural images and have been extended to the generation of RGB videos and 3D volume data. In this paper we consider the task of generating RGB-D videos, which is less extensively studied and still challenging. We explore deep GAN architectures suitable for the task, and develop 4 GAN architectures based on existing video-based GANs. With a facial expression database, we experimentally find that an extended version of the motion and content decomposed GANs, known as MoCoGAN, provides the highest quality RGB-D videos. We discuss several applications of our GAN to content creation and data augmentation, and also discuss its potential applications in behavioral experiments.**

I. Introduction

The use of color videos with depth (RGB-D) has been spread to various applications in computer vision such as people tracking[1], object recognition[2], pose estimation[3][4] and human activity recognition[5][6]. For such applications, deep learning is thought to be useful but preparation of large RGB-D training dataset is time-consuming and remains to be an issue. One method of tackling this issue are generative models, which can possibly be used for data augmentation.

Generative adversarial networks (GANs) [7] are generative models and are able to generate novel dataset based on learned dataset. Over the past few years, many GANs for image generation have been proposed, such as image colorization [8], domain translation [9], image super-resolution [10], high resolution image generation [11]. These previous studies demonstrate that GANs have the ability of generating high quality natural images. In recent years GANs have been applied to generation of videos and 3D volume data. The first attempt to generate videos using a GAN is VideoGAN [12]. VideoGAN decomposes a given video into static background and moving foreground to effectively describe the dynamics of the video. The motion and content decomposed GAN (MoCoGAN) [13] is a state-of-the-art method for video generation. MoCoGAN decomposes a given video into motion and content, and represents the dynamics of the video in the latent motion subspace.

In this paper we consider the task of generating RGB-D videos, which is less extensively studied and still challenging. We explore deep GAN architectures suitable for the task, and develop 4 GAN architectures based on the existing video-based GANs. The architectures are: (1) a model extended the simple 2D GAN to RGB-D video generation using 3D CNN, (2) a model extended from VideoGAN which decompose video into foreground and background, (3) a model extended from MoCoGAN which decompose video into motion and content, and embed them in latent space, and model with RNN, (4) a model which generate RGB and depth independently based on MoCoGAN.

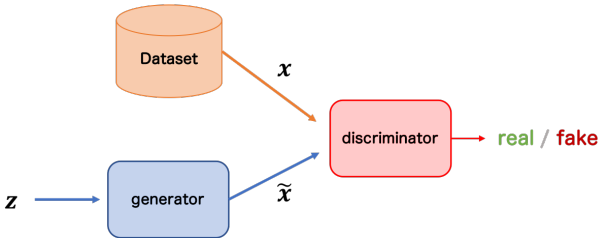


Fig. 1: Basic architecture of GANs

II. Related works

We review the basic GAN and its extension to video generation: VideoGAN and MoCoGAN.

A. Generative Adversarial Networks

GANs train generative models via an adversarial process. The GANs architecture consists of two networks: a generator network and a discriminator network. Fig.1 shows the architecture. The generator network G produces a sample from a latent code $\boldsymbol{z}$,

$$\tilde{\boldsymbol{x}} = G(\boldsymbol{z}). \tag{1}$$

The discriminator network D estimates the probability that a sample came from the training dataset rather than G.

We train G such that it generates a sample similar to the dataset, and train D to discriminate samples from the dataset to ones that are generated by G. Training of G and D is achieved via solving a minimax problem given by the objective

$$\min_G \max_D V(D,G), \tag{2}$$

where the $V(D,G)$ is

$$V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\tilde{\boldsymbol{x}}\sim p_g(\boldsymbol{x})}[\log(1 - D(\tilde{\boldsymbol{x}}))], \tag{3}$$

where p_g and p_{data} denote the generator distribution and the data distribution, respectively. In practice, (2) is optimized with a gradient based method.

B. VideoGAN

The VideoGAN [12] has following two assumptions: 1) videos are captured with a stationary camera, and 2) only objects move in videos. Based on these assumptions, Vondrick et al. argued that a scene can be decomposed into a moving foreground and a static background and the decomposition is useful for modeling scene dynamics.

Specifically, the architecture of the generator network is a two-stream model that explicitly models the foreground with 3D convolutional neural network (3DCNN) and the background with 2DCNN, respectively. The generator produces a video from a single latent code $\boldsymbol{z}$, so a point in the latent space corresponds to a video. In the training, the network to train weights to add up the foreground and the background are also integrated. The discriminator network is the same as a usual GAN, which distinguish the generated samples from the actual dataset samples.

VideoGAN tends to generate clear background and can attach appropriate motion to appropriate object in the scene. On the other hand, the generation of foreground needs improvement because the resolution of generated objects with motion is low.

C. MoCoGAN

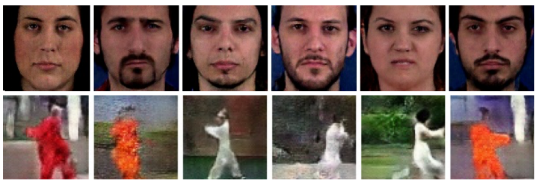
The MoCoGAN[13] producing a video from latent code consists of content and motion part. Separating the two enables easier control over the content as well as the movements within the generated video.

Furthermore, Tulyakov et al. pointed out that previous approaches have little consideration of the temporal attributes of the video. Videos recording the same actions with difference in speed will be represented by different points in the latent space. To solve this issue, they has sampled a single point from the motion latent space representing the initial state of the RNN. This is then fed to the RNN to produce motion vectors that have a length of the video frames. In other words, the RNN is capable of learning trajectories in the motion latent space corresponding to every movement, and its speed can be defined by the sampling interval of the motion vectors. This architecture enables modeling the dynamics of the video with ease.

Fig.2 shows examples of generated samples with MoCoGAN and VideoGAN. These examples demonstrate that MoCoGAN provides more natural images than VideoGAN. The result is also supported in subjective evaluations using crowdsourcing [13].

III. Proposed models

In this section, we propose 4 GAN architectures for RGB-D videos based on simple GAN, VideoGAN and MoCoGAN. Our basic idea is to extend the GAN architectures for image generation to those for RGB-D video generation by replacing 2D convolution with 3D convolution (spatio-temporal convolution).



(a) generation result by MoCoGAN

(b) generation result by VideoGAN

Fig. 2: comparison generated samples between MoCoGAN and VideoGAN, reprinted from [13]

A. RGBD-GAN

The simplest GAN architecture for image generation consists of a single generator and a single discriminator which are built based on 2DCNN. We extend the 2DCNN based simplest architecture to the 3DCNN based architecture for RGB-D videos. We call this architecture RGBD-GAN. Fig.3a shows the RGBD-GAN architecture. In the figure, "3D" represents that the network has 3 dimensional (spatio-temporal) convolution.

The RGBD-GAN generates samples from a latent code using (1). Learning RGBD-GAN is made by solving the minimax problem of (2).

B. RGBD-VideoGAN

Second we develop a VideoGAN based architecture for RGB-D video generation by adding depth channel. We call this architecture RGBD-VideoGAN. Fig.3b shows the architecture.

The RGBD-VideoGAN generates samples from a latent code,

$$\tilde{\boldsymbol{x}} = m(\boldsymbol{z}) \odot G_f(\boldsymbol{z}) + (1 - m(\boldsymbol{z})) \odot G_b(\boldsymbol{z}), \tag{4}$$

where G_f and G_b represent the generator for foreground and background, respectively, and m represents the network that estimates the ratio of the foreground video and background image. Note that $\odot$ is element-wise multiplication. The optimization problem is the same as (2).

C. RGBD-MoCoGAN

Third we develop a MoCoGAN based architecture for RGB-D video generation by adding depth channel. We call this architecture RGBD-MoCoGAN. Fig.3c shows the architecture.

The RGBD-MoCoGAN generates samples from two latent codes,

$$\tilde{\boldsymbol{x}} = \left[G_I\left(\begin{bmatrix} \boldsymbol{z_C} \\ \boldsymbol{z_M^{(1)}} \end{bmatrix}\right), ..., G_I\left(\begin{bmatrix} \boldsymbol{z_C} \\ \boldsymbol{z_M^{(T)}} \end{bmatrix}\right)\right], \tag{5}$$

where G_I represents the generator, T represents the video length, $\boldsymbol{z_C}$ and $\boldsymbol{z_M}$ are latent vectors corresponding to content and motion part of the video, respectively. Similar to MoCoGAN [13], $\boldsymbol{z_C}$ is sampled from the Gaussian distribution with mean 0 and covariance matrix I_{d_C}, where I_{d_C} is the $d_c \times d_c$ identity matrix. Also $\boldsymbol{z_M}$ is recursively generated from

$$\boldsymbol{z_M^{(t)}} = R_M(\boldsymbol{z_M^{(t-1)}}, \boldsymbol{\epsilon^{(t)}}),\ t = 1, 2, \ldots, T. \tag{6}$$

where R_M represents the recurrent neural network and $\epsilon^{(t)}$ is a normal Gaussian random variable with covariance matrix I_{d_M}. For a video, the content vector $\boldsymbol{z_C}$ is sampled once and fixed, a series of motion vectors $[\boldsymbol{z_M^{(1)}}, ..., \boldsymbol{z_M^{(T)}}]$ is produced by R_M. The objective function is

$$\min_{G_I, R_M} \max_{D_I, D_V} V(G_I, R_M, D_I, D_V), \tag{7}$$

where

$$\begin{aligned} &V(G_I, R_M, D_I, D_V) \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_I(\boldsymbol{x}^{(t)})] + \mathbb{E}_{\tilde{\boldsymbol{x}} \sim p_g}[\log(1 - D_I(\tilde{\boldsymbol{x}}^{(t)}))] \\ &\quad + \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_V(\boldsymbol{x})] + \mathbb{E}_{\tilde{\boldsymbol{x}} \sim p_g}[\log(1 - D_V(\tilde{\boldsymbol{x}}))]. \end{aligned} \tag{8}$$

where D_I represents image discriminator, D_V represents video discriminator, p_g represents the distribution of generator, p_{data} represents the distribution of the training data.

D. RGB+D-MoCoGAN

Each of the above three GANs includes a signal generator which produces both of the RGB and depth signals. The use of only the single generator implicitly assumes that these two channels depend on each other. However RGB signals are essentially different from depth signals because RGB is color but depth is geometric distance. Hence we develop a MoCoGAN based architecture by introducing two generators; the first generator G_{rgb} produces RGB videos, and the second G_{depth} produces depth videos. Fig.3d shows the architecture in which the two generators G_{rgb} and G_{depth} are placed in the network. We call this architecture RGB+D-MoCoGAN.

The RGB+D-MoCoGAN generates a sample from a latent code,

$$\begin{aligned} \tilde{\boldsymbol{x}} &= \left[\tilde{\boldsymbol{x}}_{\text{rgb}}^{(1)} \oplus \tilde{\boldsymbol{x}}_{\text{depth}}^{(1)}, ..., \tilde{\boldsymbol{x}}_{\text{rgb}}^{(T)} \oplus \tilde{\boldsymbol{x}}_{\text{depth}}^{(T)}\right], \\ \tilde{\boldsymbol{x}}_{\text{rgb}} &= \left[G_{\text{rgb}}(\begin{bmatrix} \boldsymbol{z_C} \\ \boldsymbol{z_M^{(1)}} \end{bmatrix}), ..., G_{\text{rgb}}(\begin{bmatrix} \boldsymbol{z_C} \\ \boldsymbol{z_M^{(T)}} \end{bmatrix})\right] \\ \tilde{\boldsymbol{x}}_{\text{depth}} &= \left[G_{\text{depth}}(\begin{bmatrix} \boldsymbol{z_C} \\ \boldsymbol{z_M^{(1)}} \end{bmatrix}), ..., G_{\text{depth}}(\begin{bmatrix} \boldsymbol{z_C} \\ \boldsymbol{z_M^{(T)}} \end{bmatrix})\right] \\ \boldsymbol{z_M^{(t)}} &= R_M(\boldsymbol{z_M^{(t-1)}}, \boldsymbol{\epsilon^{(t)}}),\ t = 1, 2, \ldots, T \end{aligned} \tag{9}$$

where R_M represents the network which generate $\boldsymbol{z_M}$, G_{rgb} represents the generator to produce rgb video frame, G_{depth} represents the generator to produce depth video frame, $\oplus$ represents channel-wise concatenation T represents the video length. The objective function is

$$\min_{G_{\text{rgb}}, G_{\text{depth}}, R_M} \max_{D_I, D_V} V(G_{\text{rgb}}, G_{\text{depth}}, R_M, D_I, D_V), \tag{10}$$

$$\begin{aligned} &V(G_{\text{rgb}}, G_{\text{depth}}, R_M, D_I, D_V) \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_I(\boldsymbol{x}^{(t)})] + \mathbb{E}_{\tilde{\boldsymbol{x}} \sim p_g}[\log(1 - D_I(\tilde{\boldsymbol{x}}^{(t)}))] \\ &\quad + \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_V(\boldsymbol{x})] + \mathbb{E}_{\tilde{\boldsymbol{x}} \sim p_g}[\log(1 - D_V(\tilde{\boldsymbol{x}}))], \end{aligned} \tag{11}$$

D_I represents image discriminator, D_V represents video discriminator, p_g represents the distribution of generator, p_{data} represents the distribution of the training data.

IV. Experiment

We compare the four GAN architectures with a facial expression video dataset and evaluate them from qualitative and quantitative point of views.

A. Dataset

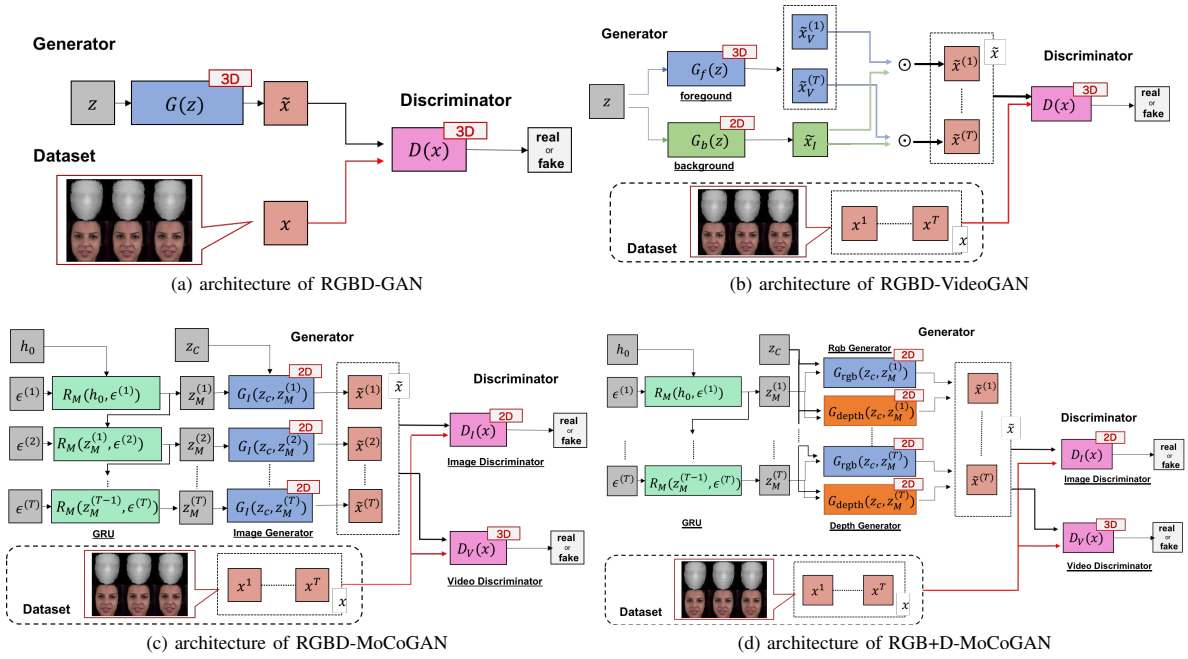
In the experiment, we made a new RGB-D video dataset from an existing RGB video dataset by predicting depth channel. The base dataset is the MUG Facial Expression Database [14], which records facial expressions of 86 people. The size of video image is 896×896, video length is 50 to 160 frames, the class label has six types of facial expressions: anger, disgust, fear, happy, sad, and surprise. We perform the following preprocessing for each image frame in a video to obtain depth information:

1) Crop face region;
2) Resize 192×192;
3) Apply 3-dimensional facial reconstruction [15] to obtain a voxel data like $(height, width, depth) = (192, 192, 200)$;
4) Treat the maximum value on z-axis of the voxel as depth;
5) Resize the rgb image obtaned by step 1) and the depth image obtained by step 3) to 64×64.

Using the above preprocessing, we prepare the RGB-D videos dataset of the size of 64×64. The length of the generated video by all of models is fixed at 16 frames, so we extract multiple subsequences from each video.

B. Condition

We build the generators and discriminators with five convolutional layers. We use rectified linear unit (ReLU) for the generators and leaky-ReLU for the discriminators. In addition we adopt batch normalization to train them. To stabilize the training process, Gaussian noises with $\mu = 0, \sigma = 0.2$ are added to input of each layer of discriminator [16]. According to the papers of VideoGAN and MoCoGAN, we use Gaussian distribution with $\mu = 0, \sigma = 0.33$ as the distribution of the latent code. The dimension of the input noise is 100 for RGBD-GAN and RGBD-VideoGAN. d_C and d_M for RGBD-MoCoGAN and RGB+D-MoCoGAN are 50 and 10, respectively. We used GRU[17] as the recurrent neural network R_M. Table I shows other training configurations.



(a) architecture of RGBD-GAN

(b) architecture of RGBD-VideoGAN

(c) architecture of RGBD-MoCoGAN

(d) architecture of RGB+D-MoCoGAN

Fig. 3: Proposed architectures

TABLE I: Training configuration

dataset	The MUG Facial Expression Database type: RGB-D video the numbers of samples: 3219 (channel, frame, height, width) : (4, 16, 64, 64)
batchsize	35
training epoch	200
optimizer	Adam($\alpha = 0.0002, \beta_1 = 0.5, \beta_2 = 0.999$)

C. Qualitative evaluation

We trained our proposed models with the dataset and compared visually the quality of the results. Fig. 4 shows three example sequances (4, 8, 12, and 16th frames) of the RGB-D videos generated by the proposed four models. In Fig. 4, we separately show the RGB and depth signals.

We visually and subjectively compare the results based on the three points:

1) RGB frame quality,
2) depth frame quality,
3) dynamics size.

Table II summarizes the qualitative evaluation results. As shown in Table II, the best model is RGBD-MoCoGAN. RGBD-GAN and RGBD-VideoGAN provide the poor quality RGB frames. RGBD-MoCoGAN and RGB+D MoCoGAN provide the good quality RGB frames but the latter provides the poor quality depth frames.

D. Quantitative evaluation

For quantitative evaluation, we use inception-score [18] which is thought to be correlated very well with human judgment. The inception-score is a Kullback-Leibler divergence between two probability distributions and defined by

$$\exp(\mathbb{E}_{\boldsymbol{x}\sim p_g(\boldsymbol{x})}[\mathrm{KL}(p(y|\boldsymbol{x})||p(y)]), \tag{12}$$

where $\boldsymbol{x}$ and y represent a generated sample and the label (e.g. happiness, anger). The probability distributions $p(y)$ and $p(y|\boldsymbol{x})$ are estimated by the pre-trained classifier model. We use C3D [19] as the classifier model which trained with the same dataset in this experiment.

The metric based on the fact that good samples are expected to yield:

1) low entropy $p(y|x)$, i.e. high prediction confidence;
2) high entropy $p(y)$, i.e. highly varied predictions.

We calculate the average and the variance of inception-score from 10000 generated samples. As shown table III, the best model is RGBD-MoCoGAN.

E. Discussion

From the qualitative and quantitative point of view, the RGBD-MoCoGAN model provides the best quality RGB-D videos. This result indicates that the MoCoGAN architecture can properly model video dynamics. We thought that RGB+D-MoCoGAN was the best model because the separation of RGB and depth channels in the generation process was intuitively appropriate. However this is not the case with the facial expression dataset. This result indicates that RGB and depth channels

TABLE II: Qualitative evaluation

	RGBD-GAN	RGBD-VideoGAN	RGBD-MoCoGAN	RGB+D-MoCoGAN
quality of RGB frame	Poor	Poor	Good	Good
quality of depth frame	Good	Poor	Good	Poor
size of dynamics	Average	Poor	Good	Good

TABLE III: Evaluation with inception score

Model	Inception Score	Rank
RGBD-GAN	4.73 ± 0.0149	2
RGBD-VideoGAN	4.55 ± 0.0157	4
RGBD-MoCoGAN	4.77 ± 0.00914	1
RGB+D-MoCoGAN	4.67 ± 0.0184	3
Dataset	5.17 ± 0.0351	-

have a strong correlation and these individual modeling makes training hard.

V. Conclusion

RGB-D video generation will have a considerable impact on a wide range of fields in computer vision. To the best of our knowledge, GANs for RGB-D video generation are less extensively studied. In this paper, we propose the four GAN architectures based on GANs for video generation. With the facial expression dataset, we reveal that RGBD-MoCoGAN generates highest quality RGB-D videos. Currently we do not experiment extensively using other RGB-D videos such as natural scenes and gestures. With such dataset, we further develop better GAN architectures for RGB-D video generation.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP16K00231.

References

[1] Shuran Song and Jianxiong Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 233–240. IEEE, 2013.

[2] Mohammad Muntasir Rahman, Yanhao Tan, Jian Xue, and Ke Lu. Rgb-d object recognition with multimodal deep convolutional neural networks. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 991–996. IEEE, 2017.

[3] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 3d human pose estimation in RGBD images for robotic task learning. *CoRR*, abs/1803.02622, 2018.

[4] Ruotao He, Juan Rojas, and Yisheng Guan. A 3d object detection and pose estimation pipeline using RGB-D images. *CoRR*, abs/1703.03940, 2017.

[5] Yue Ming, Qiuqi Ruan, and Alexander G Hauptmann. Activity recognition from rgb-d camera with 3d local spatio-temporal features. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 344–349. IEEE, 2012.

[6] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. June 2016.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. July 2017.

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017.

[10] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 105–114, 2017.

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018.

[12] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[13] Xiaodong Yang Sergey Tulyakov, Ming-Yu Liu and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The mug facial expression database. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.

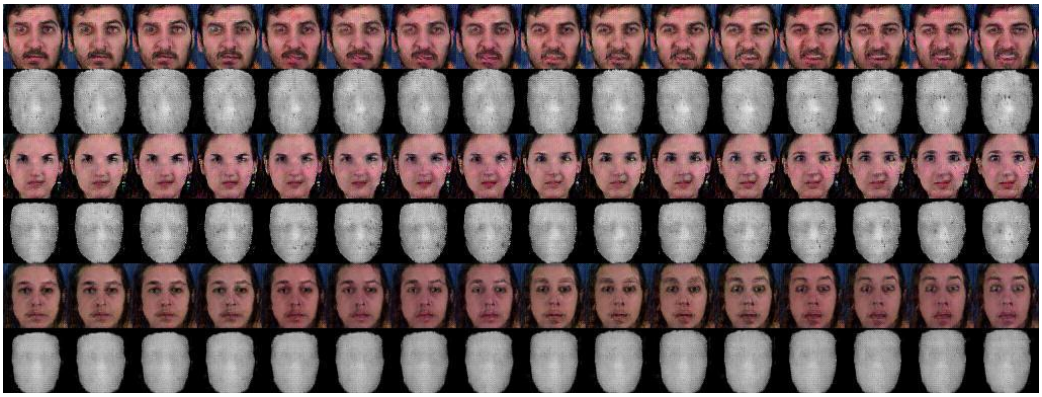
[15] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1031–1039. IEEE, 2017.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[17] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Deep Learning Workshop*, 2014.

[18] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

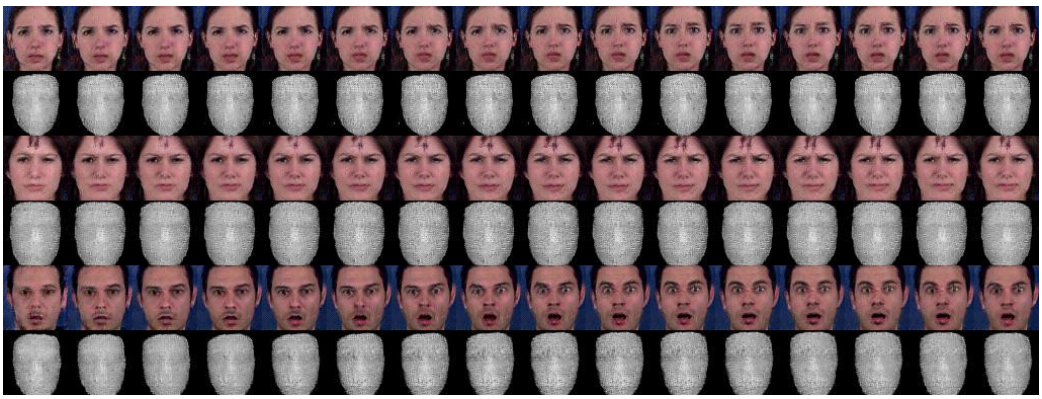
[19] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.



(a) RGBD-GAN

(b) RGBD-VideoGAN

(c) RGBD-MoCoGAN



(d) RGB+D-MoCoGAN

Fig. 4: Generated samples（upper is rgb images, lower is depth images）