Vibrotactile Signal Generation with GAN

Shotaro Agatsuma[1], Shin Takahashi[1], Satoshi Saga[2]

[1]University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, Japan
[2]Kumamoto University, 2-39-1 Kurokami, Chuo-ku, Kumamoto, Japan
Keywords: Vibrotactile information, Acceleration, GAN

ABSTRACT
To create valuable content for haptic display, we have been developing a method of generating alternative data from acquired one instead of collecting a vast data from real textures. We made a data generation model based on Generative Adversarial Network and held experiments to evaluate the performance of the model.

1 INTRODUCTION

Along with the development of virtual reality technology, various tactile displays have been developed. Tactile displays give tactile sensations to the user when s/he touches the device. Currently, many tactile displays employ mechanical vibrations to present tactile sensations. These displays are called as vibrotactile displays and there is a large amount of research about them. Many researchers are developing systems that employ recorded vibration as a tactile signal, and these systems present high-quality tactile sensations [1] [2]. The vibrations are generally collected with an acceleration sensor by the rubbing movement over real objects. Numerous research projects are carried out with regard to collecting the vibration from real objects [3] [4]. For example, Strese et al. [4] collected six types of data (acceleration, pressure, temperature, photo, sound, and power of the magnetic field) for 108 types of textures via a pen-typed sensing device. They rubbed the textures in one direction with the device and collected the data. They collected a lot of types of data under various kinds of conditions. However, their conditions of contact are fixed, for example, in rubbing direction, or in a contact angle of the device toward a texture.

Acts of touch are bidirectional phenomena. If the surface conditions, physical characteristics, or rubbing speed are different between the contactor and the contacted object, the induced phenomena are different between them. Therefore, it is unrealistic to collect all data under numerous combinations of conditions. To solve the problem, Ujitoko et al. [5] proposed the method to generate alternative data from known data instead of exhaustive data collection. The method can generate data that is similar to known data. However, the method cannot generate data that are different from known ones.

The goal of our research is to realize the system which generates unknown data from recorded one. The generated data should be alternative to data under different conditions from recorded vibration data. If the system is realized, we can reduce the cost of data collection, and obtain various unrecorded contents for vibrotactile displays from recorded one.

As a first step toward realizing the system, we propose a machine learning model using a Generative Adversarial Network (GAN) [6] to generate vibration data. GAN is one of the machine learning methods and mainly used for image generation. GAN can generate data that is similar to training data. GAN is composed of two models: generator and discriminator. The generator generates data, and the discriminator classifies the generated data and training data. The discriminator is trained to classify the two types of data accurately. The generator is trained to generate data that the discriminator cannot classify. After the repetitive training of the generator and the discriminator, the generator can generate data that is almost same to training data. By adjusting the configuration of GAN appropriately, GAN can generate data that is alternative to data under different conditions [7]. Also, in recent research, GAN can learn time-series data accurately. In the field of speech synthesis, several researchers have been using GAN, and they have successfully generated unrecorded speech data [8] [9].

We previously used a model based on Deep Convolutional GAN (DCGAN) [7] for the data generation [10]. In this paper, we use the model based on WaveGAN [9] for the data generation. WaveGAN is one of the GAN models for speech synthesis. Also, we held a data generation experiment to verify whether the generated model could generate vibration data effectively. In this experiment, we used 3-axis acceleration data as training data for the model. This data was collected by the experimenter with a 3-axis accelerometer on his finger. The GAN we have modeled generates data similar to the training data. In the rest of the paper, we explain the detail of the experiment, and discuss the result of the experiment.

2 ARCHITECTURE OF PROPOSED GAN MODEL

In this section, we explain the architecture of the proposed GAN model. In our research, the training data to make the GAN model is 3-axis acceleration. Currently, we are investigating what kind of setting is adequate to generate mechanical vibration data. As a first step, we are going to generate mechanical vibration data similar to the training data. We have composed a GAN model which is based on the WaveGAN [8]. This is a model for speech synthesis. The model is composed of DCGAN and WGAN-GP. The model was trained with a subset which consists of the spoken digits from “zero” to “nine,” and the model generates sounds that are similar to the subset. Generally, sound data is time-series data, and mechanical acceleration data is also time-series data. Therefore, WaveGAN is suitable for generating mechanical vibration data. In the preliminary research, the proposed GAN model is set to generate data that is similar to the training data. Under this setting, we first confirmed whether the model could learn the features of training data effectively in generating unknown data.

The architecture of the model has fewer convolutional layers than the original WaveGAN because the length of our training data is shorter than the sound data in the WaveGAN. Usually, the convolution layer reduces the size of the propagating data and summarizes the overall characteristics. If we use the same architecture with WaveGAN, our data will become too short because of the shorter length of our training data. Therefore, we reduce convolutional layers and suppress the shortage of data.

WaveGAN is based on DCGAN by Radford et al. [7]. However, unlike Radford’s model, it has one-dimensional convolution filters and strides. These settings increase the amount of convolution in the temporal direction and enable it to learn one-dimensional time-series data effectively. However, we use 3-axis time-series data of mechanical vibrotactile signals. Therefore, we made three models for each axis of the training data independently in order to apply the architecture of WaveGAN to the 3-axis data. Fig. 1 shows the architecture of the model. The detailed settings for each model are shown below.

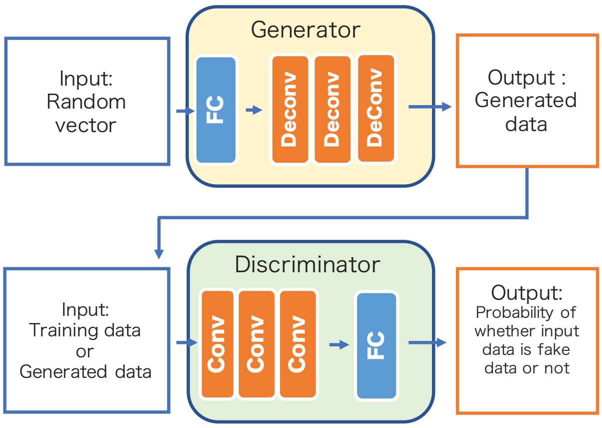

Fig. 1 Architecture of the GAN

FC is a fully connected layer. Conv is a convolution layer. Deconv is a deconvolution layer.

The input data of the generator is a 1×100 of noise vector based on a uniform distribution from -1 to 1. The output data is generated data that depends on training data. The size of the convolution filters is set to 1×25, and strides are set to 1×4. The activation function for the output layer is set to the hyperbolic tangent function, and that for the others is set to the ReLU function. The learning rate optimizer is set to Adam [13]. Next, the input data of the discriminator is the training data or the generated data by the generator. The training data is normalized between -1 and 1. The output data is the probability of whether input data is generated by data or not. The size of the convolution filters is set to 1×25, and strides are set to 1×4. For the output layer, the activation function is set to the sigmoid function, and that for the others is set to the Leaky ReLU function ($\alpha = 0.2$). The learning rate optimizer is set to Adam. In addition, we used WGAN-GP to suppress the vanishing gradient.

3 EXPERIMENT

In this section, we describe the data generation experiment with the proposed GAN model. We evaluated whether the model is appropriate for data generation from 3-axis acceleration data.

3.1 Settings

The training data was 3-axis acceleration data collected by an experimenter. We used a 100 mm x 100 mm sized artificial lawn texture for this experiment. Fig.2 shows the texture and how the data was collected. The accelerometer was placed on the nail of the finger. The experimenter rubbed the texture in one direction from left to right for 5 seconds at a speed of 5 cm/s and collected 3-axis acceleration data at a sampling rate of 1kHz. He performed this trial 90 times and collected 450,000 points of 3-axis acceleration data (the length is approximately 7 minutes and 30 seconds).

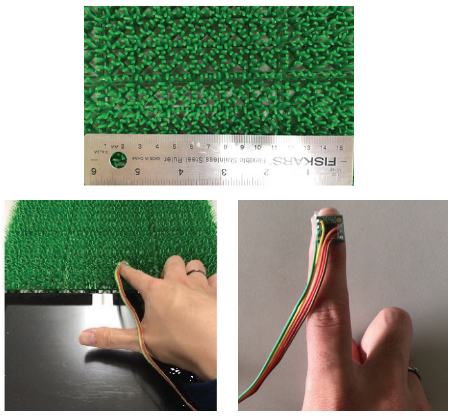

Fig. 2 Texture and data collection

Top: Texture used in the experiment. Bottom right: A fingertip with an accelerometer attached. The sensor axis is the X-axis in the horizontal direction, the Y-axis in the vertical direction, and the Z-axis in the depth direction. Bottom left: How the data was collected.

We trained the model using the collected data and generated data. We extracted 70,000 datasets of 1,024 sequential points randomly from the collected data and employed them as training data. During the training, the mini-batch size was set to 64, and the number of epochs was set to 200. After the training, we generated data using the trained model.

3.2 Results and Discussion

In this section, we describe the data generation result of the experiment. Fig. 3 shows three sets of graphs — one set (three graphs) for each axis. The graph at the left and the middle depict an example data sampled from the training data and the generated data, respectively. The graph at the right shows both of them. The vertical axis of the graph is the output value of the model, and the value is between -1 and 1. The horizontal axis is a temporal index. The data has a length of 1,024 points per data in time series.

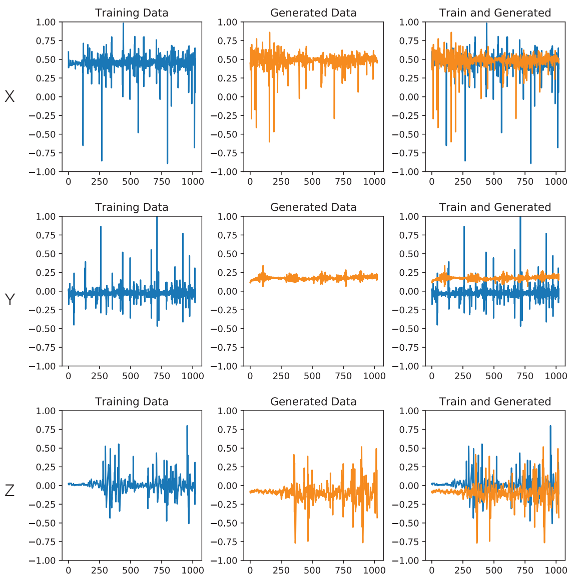

Fig. 3 Result of the generation with 200 epochs

From the results of the X-axis (the three graphs at the top of Fig. 3) and the Z-axis (the three graphs at the middle of Fig. 3), the characteristics of the training data are almost reproduced. The center values of the generated data are similar to training data. From these results, we found that the model is possible to generate data effectively for X and Z-axis.

From the results of the Y-axis (the three graphs at the bottom of Fig. 3), the generated data does not have characteristics of the training data. Contrary to the training data, data with a flat waveform was generated. It appears that learning was insufficient for the Y-axis. Therefore, we increased epochs to be 800 and generated data. Fig. 4 shows the result for 800 epochs. From the result, we can observe that the generated data has characteristics of the training data. Therefore, the model can generate data effectively through an appropriate learning amount.

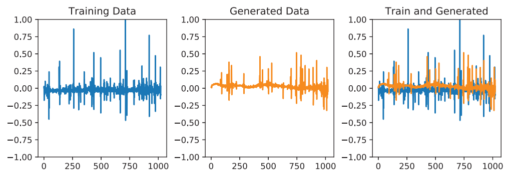

Fig. 4 Result of the generation with 800 epochs about Y-axis

4 CONCLUSION AND FUTURE WORK

We proposed to use the GAN to generate unknown vibrotactile data from recorded data. As the first step of realizing the proposed method, we made the model based on WaveGAN. We used a 3-axis acceleration as vibrotactile signals. Since the signals are 3-axis time-series data, we made the model for each axis independently. We held the data generation experiment using the proposed model and compared the generated data with the training data. As a result, the model generated the data similar to training data for each axis. Therefore, we found that there is a possibility that the model can generate data with vibration data.

In the future, we will improve the model to generate data that is not similar to the training data. Specifically, we will implement Conditional GAN [14]. The GAN can learn label data as well as training data at the same time if users use label data during data generation, it can generate the data corresponding to the label. If we operate the label appropriately, there is a possibility that the model generates the data which are different from known vibration data. In addition, we will evaluate the model based on T-SNE analysis in the same way as Ujitoko et al. [5] in order to evaluate the model more quantitatively. After that, we will present tactile sensations based on the generated data to human subjects in order to evaluate the generated data subjectively.

REFERENCES

[1] Youngjun Cho, Andrea Bianchi, Nicolai Marquardt and Nadia Bianchi-Berthouze. RealPen: Providing Realism in Handwriting Tasks on Touch Surfaces Using Auditory-Tactile Feedback, In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, pp. 195-205. ACM, 2016.

[2] Satoshi Saga and Koichiro Deguchi. Lateral-force-based 2.5-dimensional tactile display for touch screen, In Proceedings of Haptics Symposium 2012, pp. 15-22. IEEE, 2012.

[3] Arsen Abdulali and Seokhee Jeon. Data-Driven Modeling of Anisotropic Haptic Textures: Data Segmentation and Interpolation. In Haptics: Perception, Devices, Control, and Applications: 10th International Conference, pp. 228-239. Springer, 2016.

[4] Matti Strese, Yannik Boeck, and Eckehard Steinbach. Content-based Surface Material Retrieval. In Proceedings of 2017 IEEE World Haptics Conference (WHC), pp. 352-357. IEEE, 2017.
[5] Yusuke Ujitoko and Yuki Ban. Vibrotactile signal generation from texture images or attributes using generative adversarial network. In International Conference on Human Haptic Sensing and Touch Enabled Computer Applications, pp. 25-36. Springer, 2018.
[6] 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 27, pp. 2672-2680. Curran Associates, Inc., 2014.
[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.
[8] Olof Mogren. C-RNN-GAN: continuous recurrent neural networks with adversarial training. arXiv preprint arXiv: 1611.09904, 2016.
[9] Chris Donahue, Julian McAuley, and Miller Puckette. Synthesizing audio with generative adversarial networks. arXiv preprint arXiv:1802.04208, 2018.
[10] Shotaro Agatsuma, Jyunya Kurogi, Satoshi Saga, and Shin Takahashi. Signal generation for vibro-tactile display by generative adversarial network. In Proceedings of AsiaHaptics 2018, pp. 58-60. Springer, 2018.
[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems 30, pp. 5767-5777. Curran Associates, Inc., 2017.
[12] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:1804.03209, 2018.
[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.