Simple Generative Adversarial Network to Generate Three-axis Time-series Data for Vibrotactile Displays

Shotaro Agatsuma
Graduate School of Systems and Information Engineering,
University of Tsukuba, Japan
e-mail: agatsuma@saga-lab.org

Junya Kurogi
Faculty of Engineering,
Kumamoto University, Japan
e-mail: kurogi@saga-lab.org

Satoshi Saga
Faculty of Advanced Science and Technology,
Kumamoto University, Japan
e-mail: saga@saga-lab.org

Simona Vasilache
Faculty of Engineering, Information and Systems,
University of Tsukuba, Japan
e-mail: simona@cs.tsukuba.ac.jp

Shin Takahashi
Faculty of Engineering, Information and Systems,
University of Tsukuba, Japan
e-mail: shin@cs.tsukuba.ac.jp

***Abstract*—Various kinds of vibrotactile information have been recorded from real textures and used to present high-quality tactile sensations via tactile displays. However, it is unrealistic to collect large amounts of vibrotactile data under many different conditions. Thus, we develop a method whereby recorded data can be changed to represent conditions differing from those at the time of initial recording. In the first step, we construct a data generation model using a Generative Adversarial Network (GAN). The model makes simple calculations and generates unknown data from recorded acceleration data obtained by rubbing real objects. The model can generate three-axis, time-series data. To evaluate the quality of the data generated, we devised a string-based tactile display and presented generated vibrotactile information to users. Users reported that the generated data were indistinguishable from real data.**

Keywords–Acceleration; Generative Adversarial Networks; Vibrotactile Display.

I. INTRODUCTION

Currently, various tactile displays have been developed, and a lot of applications that enable users to touch virtual objects are released. The quality of such applications is measured by the extent of realism felt when the virtual objects are touched. It is difficult to create realistic tactile sensations. In particular, realistic surface reproduction is challenging because touching is bidirectional, thus affected by object condition. If the object surface, physical characteristics, or rubbing speed differ between the contactor and the contacted object, the induced phenomena differ. To ensure high-quality tactile sensation, it is necessary to collect and analyze data under various conditions [1][2]. However, many conditions were not addressed in the cited works. For example, Strese et al. [2] collected six types of data (accelerations, pressures, temperatures, images, sounds, and magnetic field powers) for 108 textures, under various conditions, using a pen-type device. However, there are many more than 108 textures, and not all rubbing directions or contact angles were explored.

To solve this problem, our method eliminates the need for vibrotactile signal data from real objects; vibrotactile stimulation is created employing existing recorded data on real textures. We do not collect data from real objects; we generate alternative data under conditions different from those at the times of the original recordings. This reduces the cost of data collection and greatly expands the utility of vibrotactile displays.

In the first step, we generate vibrotactile acceleration data by acceleration data collected from rubbing real objects. The generated data can be used as output signals for vibrotactile displays [3][4]. Today, accelerometers are both small and inexpensive; a collection of acceleration data is simple. Thus, we use the data to generate new data with the aid of a Generative Adversarial Network (GAN) [5]. GANs generate images that find many applications in super-resolution [6] and audio synthesis; some sounds are very similar to the human voice [7][8]. GANs can generate high-quality time-series data. Our data generation model is based on WaveGAN [7], which was developed for audio synthesis. We generate nine types of time-series data based on real textures. We create data spectrograms to evaluate realism. We perform a user study employing a vibrotactile display to evaluate whether the vibrotactile stimuli were realistic. We explored whether it was possible to mix the characteristics of two textures by combining two types of label data in the input.

Our principal contribution is that we generate time-series data using a GAN originally developed for audio synthesis. The training data of the model are accelerations recorded by rubbing real objects. Our model has a simpler architecture than an earlier model [9], and thus requires fewer computational resources. We generate three-axis time-series data for vibrotactile displays that require more than two datasets [3]. The three-axis data facilitate the analysis and recognition of tactile signals.

The structure of this paper is as follows. This section describes the purpose of our research and our approach. In Section II below, we review related work. Section III describes our model architecture; Section IV deals with data generation. Section V describes the user study. Section VI presents a preliminary experiment on multi-label (merged) data generation. Section VII draws conclusions and describes our future plans.

II. RELATED WORK

Vibrotactile displays reproduce real textures, including the mechanical vibrations of actuators [10], electrostatic forces [11], and so on. Some real-object data are available,

 ISBN: 978-1-61208-761-0

but a complete dataset would be unimaginably large. We initially use a GAN to generate data based on the three-axis accelerations of real textures. GANs are machine-learning models generating images that may be simple or complex; the latter include super-high-resolution images [6] and images translated from other images [12]. A GAN features a generator and a discriminator that, respectively, generate and classify training and test data. The discriminator accurately classifies the two types of data. The generator creates data that the discriminator cannot initially classify. After repetitive training of the generator and the discriminator, the generator generates data that are almost the same as the training data.

A few scholars have used GANs to generate data for tactile displays. Ujitoko et al. [9] employed a GAN generating time-series data equivalent to texture images. The model featured an encoder and a generator that, respectively, transformed texture images into labeled vectors and generated spectrograms using the recoded accelerations and the labels. The spectrograms were transformed into tactile signals for pen-type vibrotactile displays. The model generated nine types of high-quality, one-axis time-series data that only found applications in simple (i.e., pen-type) vibrotactile displays. It appeared that the computational demand was high; the model featured many neural networks. Our model is simpler than the model, and we generate three-axis acceleration data that are available for more types of situations (e.g., displaying, analyzing, and recognizing the vibrotactile signals) than one-axis data. We employ a GAN originally developed for audio synthesis; some such GANs generate high-quality sounds [7][8]. Acceleration data, like sounds, are time-series data. Specifically, we employed the WaveGAN of Donahue et al [7]. The model architecture is simple. However, Donahue et al. were concerned that spectrograms served as both inputs and outputs; it was thought that spectrogram inversion might compromise quality. Thus, we did not use spectrograms.

III. The architecture of our GAN

Table I shows the architecture of our GAN. "C" refers to the several classes of training data. "n" refers to batch size. The table shows the architecture of the generator and the discriminator, and the input and output layers; the intermediate layers are hidden layers. The input data propagate to the output layer. The kernel shapes of each convolutional layer are shown, as are the output data shapes of all layers.

As mentioned above, we employed WaveGAN. However, WaveGAN generates only a single data type. We thus additionally implemented a conditional GAN [13] that generates class-specified data by attaching class labels c to the training data. In this GAN, all data are associated with a class label c and a noise z. This allowed us to generate many types of data using one-hot vectors as labels. The vectors have values of either zero or one, and their lengths are the same as the number of classes. Each class vector has the value of one and all others have values of zero The model applies convolution to each axis, and the convolution operates three-acceleration data but only in the time direction.

We describe the details of the generator and the discriminator. The inputs of the generator are random noise vectors based on uniform -1 to 1 distributions. The vector length is 1×100 and is combined with a label vector when input. The output depends on the training data. The discriminator inputs

TABLE I. THE ARCHITECTURE OF OUR GAN.

Generator	Kernel Size	Output Shape
Input : Uniform(-1,1)+C		(n, 100+C)
Dense	(100+C, 49152)	(n, 49152)
Reshape		(n, 3, 16, 1024)
LeakyReLU (α = 0.2)		(n, 3, 16, 1024)
Trans Conv2D (Stride = (1, 4))	(1, 25, 512, 1024)	(n, 3, 64, 512)
LeakyReLU (α = 0.2)		(n, 3, 64, 512)
Trans Conv2D (Stride = (1, 4))	(1, 25, 256, 512)	(n, 3, 256, 256)
LeakyReLU (α = 0.2)		(n, 3, 256, 256)
Trans Conv2D (Stride = (1, 4))	(1, 25, 128, 256)	(n, 3, 1024, 128)
LeakyReLU (α = 0.2)		(n, 3, 1024, 128)
Trans Conv2D (Stride = (1, 4))	(1, 25, 64, 128)	(n, 3, 4096, 64)
LeakyReLU (α = 0.2)		(n, 3, 4096, 64)
Trans Conv2D (Stride = (1, 4))	(1, 25, 1, 64)	(n, 3, 16384, 1)
Output : Tanh		(n, 3, 16384, 1)

Discriminator	Kernel Size	Output Shape
Input : Training data or Generated data		(n, 3, 16384, 1+C)
Conv2D (Stride = (1, 4))	(1, 25, 1+C, 64)	(n, 64, 4096, 64)
LeakyReLU (α = 0.2)		(n, 64, 4096, 64)
Phase Shuffle		(n, 64, 4096, 64)
Conv2D (Stride = (1, 4))	(1, 25, 64, 128)	(n, 64, 1024, 128)
LeakyReLU (α = 0.2)		(n, 64, 1024, 128)
Phase Shuffle		(n, 64, 1024, 128)
Conv2D (Stride = (1, 4))	(1, 25, 128, 256)	(n, 64, 256, 256)
Phase Shuffle		(n, 64, 256, 256)
LeakyReLU (α = 0.2)		(n, 64, 256, 256)
Conv2D (Stride = (1, 4))	(1, 25, 256, 512)	(n, 64, 64, 512)
LeakyReLU (α = 0.2)		(n, 64, 64, 512)
Phase Shuffle		(n, 64, 64, 512)
Conv2D (Stride = (1, 4))	(1, 25, 512, 1024)	(n, 3, 16, 1024)
LeakyReLU (α = 0.2)		(n, 3, 16, 1024)
Reshape		(n, 49152)
Output : Dense	(49152, 1)	(n, 1)

are either training or generated data. The outputs are data that have been manipulated by the discriminator layers. We use the WGAN-GP [14] as a loss function; the discriminator outputs are used to calculate losses. We employed PhaseShuffle (Donahue et al. [7]) to generate data effectively. The phases of the layer activations are perturbed using $-n$ to n samples before being input to the next layer. We used the weight initialization method of He et al. [15] to each convolution layer in both models.

IV. Data Generation

We generated data using the model described above and confirmed that the data exhibited the characteristics of training data. We first used an earlier dataset to explore whether the model could generate similar data. Second, we used acceleration data collected by rubbing real textures with an index finger. We explored whether the model was valid when the methods used to collect training data differed.

A. Data Generation Using Lehrstuhl Für Medientechnik Haptics Texture Database

1) Training Settings: We used nine textural, three-axis acceleration datasets (Figure 1) from the Lehrstuhl Für Medientechnik (LMT) Haptic Texture Database [16] as training data; these were the data employed by Ujitoko et al. [9]. The data were collected by rubbing various textures in one direction using a pen-type device; the sampling rate was 10 kHz.

Table II shows the hyperparameters used to train the model. The discriminator input was normalized to a value between -1 to 1. We extracted 6,000 random datasets, each containing 16,384 sequential points, for each texture, and employed these for training. We generated a three-axis time-series dataset featuring 16,384 sequential points. We trained the model for

 ISBN: 978-1-61208-761-0

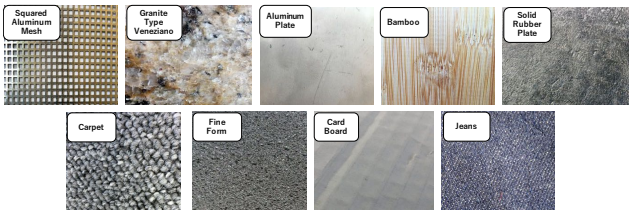

Figure 1. Textures that were chosen from the LMT Haptic Texture Database for this experiment.

40 epochs using a Windows PC with two GPUs (NVIDIA GTX1080 Ti); training required about 47 hours. We found that we succeeded in training the model quickly using the general-purpose GPU and a PC.

TABLE II. THE HYPERPARAMETERS USED.

Name	Value
Batch size	64
Phase Shuffle	2
Loss	WGAN-GP
WGAN-GP λ	10
Generator updates per discriminator	2
Optimizer	Adam (α = 1e-4, β 1 = 0.5, β 2 = 0.9)

2) Results: We drew spectrograms of the training and generated data (Figure 2) to determine whether they were similar. We extracted three classes. The three spectrograms on the left show training data (Ground Truths); the three on the right display the generated data. We computed the spectrograms in a wave format using a 256-point short-time Fourier transform (STFT) with a Hamming window of 256 and a hop size of 128. All values were normalized to between 0 and 1. The spectrograms show that the generated data exhibited the characteristics of training data. In particular, the generated "Bamboo" data were indistinguishable from the training data. Therefore, the model well-learned the characteristics of the training data. However, the generated data did not reproduce the characteristics of "Granite Type Veneziano"; the generated data differed from the training data.

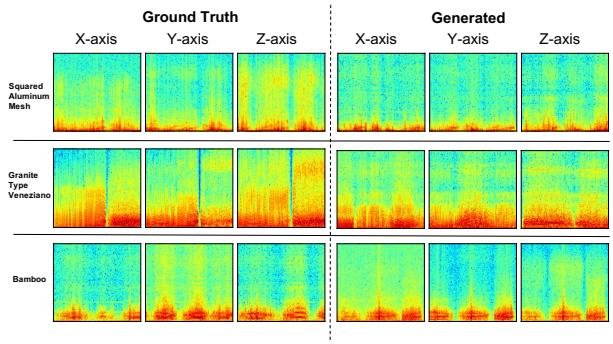

Figure 2. Spectrograms of each labeled class in the LMT Haptic Texture Database.

B. Data Generation Using Real Texture Data

1) Training Settings: We obtained three-axis acceleration data by rubbing nine textures (Figure 3) with an index finger bearing a three-axis accelerometer. "Artificial Grass" was a spiky artificial grass. "Cloth" was a silky cloth. "Carpet" was a hard carpet. "Cork Sheet" was a plate-like cork. "Punched Plastic Sheet" was a smooth punched plastic plate. "Tile" was a patterned tile. "Place Mat 01, 02, and 03" were placemats made from different materials. Figure 4 shows an overview of the data collection. The collector was one of the authors (male, 24 years of age). All textures were traced from left to right for 6 seconds at about 5 cm/s. The sampling rate was about 1 kHz. A metronome was used to ensure that the speed was approximately constant. The angle between the finger and each texture was about 45°. Each texture was sampled 80 times. We removed the first and last 1,000 points of sequential data.

We used the hyperparameters employed above (Table II). We created about 40,000 data points from 10 repeats of each collected data because the collected data lengths were shorter than 16,384 points. We extracted 6,000 random datasets each of 16,384 three-axis, time-series sequential points from the data on each texture; these served as training data. We trained the model for 40 epochs using the PC described above; training required about 46 hours, and was thus relatively fast even though the data differed from those in the LMT Haptics Texture Database.

Figure 3. The Sampled Textures.

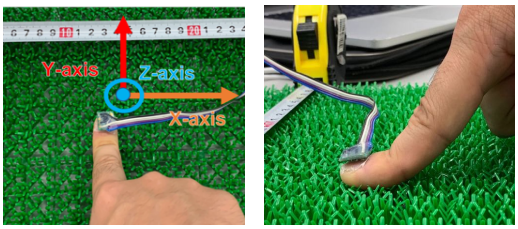

Figure 4. Overview of data collection.

2) Results: Figure 5 shows sample spectrograms prepared in a manner similar to dataset generation. The data exhibit the characteristics of training data; all generated and training data were identical. Therefore, we found that the model can generate data effectively, even using the training data that is different from the LMT Haptics Texture Database.

V. The User Study

To evaluate the quality of data generated by our model, we presented vibrotactile stimuli to users. We employed the collected data described above as training data. Ten participants (eight males and two females, age 22-24 years) were enrolled. The work was approved by the Ethics Committee of the University of Tsukuba (authorization number 2019R299) and written informed consent was obtained from all participants.

We performed two user studies. First, we explored whether vibrotactile stimuli based on generated data could be distinguished from those based on training data. The more difficult this was, the more effectively our model learned the data

 ISBN: 978-1-61208-761-0

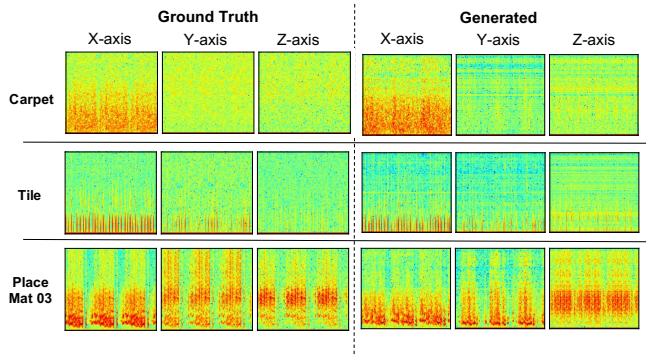

Figure 5. Spectrograms for each labeled class of collected data. The spectrogram settings are the same as those of Figure 2.

characteristics. Second, we explored the realism of vibrotactile stimuli based on training and generated data. We used the task design of Ujitoko et al. [9] and the vibrotactile display proposed by Saga et al. [3] (Figure 6 left). A finger pad was connected via threads to four motors on the four corners of the tablet. The strings were wound to deliver vibrotactile stimuli; the X-axis and Y-axis vibrations were independently controlled. This was appropriate because our model generated three-axis time-series data. The generated data is not only applied for 1-axis vibrotactile displays but also used for vibrotactile displays that need more types of data like it. We used the first 4,000 training and generated data points to present vibrotactile sensations; we were careful to ensure that data repetition did not affect sensation.

A. *Procedure of the User Studies*

Figure 6 shows an overview of the user studies and the vibrotactile display employed. Each participant placed an index finger on the pad and moved the finger from left to right on the surface of the display over two different predefined paths; s/he received vibrotactile stimuli created by test or generated data and was asked to identify the path that employed generated data. S/he then rubbed the real texture and scored realism using a Visual Analog Scale [17]. To control movement speed, we used a guide bar (on a screen) to indicate where to move. Each participant followed the movement of the bar; the finger moved at approximately 5 cm/s. The display order of training and generated trial data were randomized. We performed 10 repeat experiments for each texture; thus, each participant performed 90 tests. We explored participant views via a questionnaire. All experiments were concluded in approximately 1 hour.

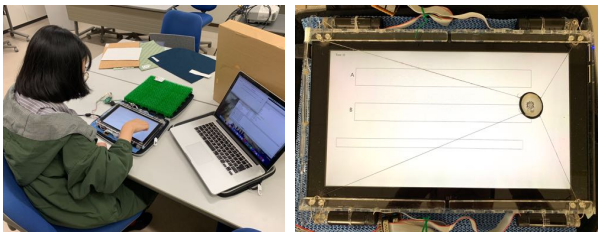

Figure 6. Left: An overview of the experiments. Right: The string-based vibrotactile display.

B. *Result and Discussion*

The top panel of Figure 7 shows the correct identification frequencies (“Correct answer rates”) of stimuli created using generated data. A value close to 50% indicated that a participant failed to distinguish training from generated data. Thus, the closer the value to 50%, the more effective the model. All values were about 50%. It was not possible to distinguish the training from the generated data. When completing the questionnaires, most participants indicated that they could not distinguish the data. Thus, the model generated data very similar to real acceleration data. The correct answer rates of most participants were 40-60% for each texture. Notably, seven participants exhibited 50% correct answer rates for “Carpet” (a rough texture).

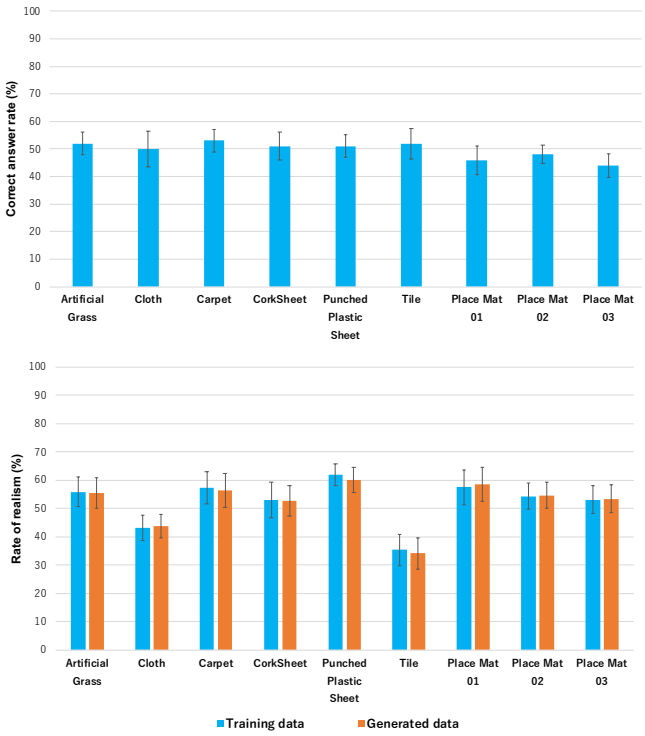

Figure 7. Top: The correct answer rate for each texture. Bottom: The realism of each texture.

The bottom panel of Figure 7 deals with realism; the values are the averages of all answers. If the values for generated data are close to those for training data (as was indeed the case for all textures), the two types of data were similar. The paired Student’s t-test revealed no significant difference between the training and generated data for any texture; vibrotactile stimuli created using generated data were as realistic as those prepared to employ training data. In contrast, significant differences between training and generated data were evident for some textures in the work of Ujitoko et al. [9]. Our model may generate higher-quality data.

The realism scores were 50-70% for all textures except “Cloth” and “Tile.” Saga et al. [3] reported realism scores of 50-70% using the vibrotactile display that we employed to present real textures. Thus, our vibrotactile display performed well. Turning to the two textures with lower values: “Cloth” scored poorly because the vibrotactile display did not repro-

 ISBN: 978-1-61208-761-0

duce the stimuli well. The display preferentially reproduces rough textures (the fingertip vibrations are large) and, thus, not silky textures such as “Cloth”. In the future, we will use a different display. The “Tile” value was low because the stimuli were weak, explained by the fact that the accelerations were small. The “Tile” featured a gutter (Figure 3) that affected changes in acceleration; these were small because the gutter was shallow and fingertip vibration thus very low. This will be improved by changing the data collection method and the display. The bottom panel of Figure 7 reveals almost no difference between the realism of generated and training data, even for “Tile” (Figure 5). Therefore, it appears that the model succeeded in generating data effectively.

VI. Data generation with the merged label

We explored whether the model generated unknown data when we varied the input label; we performed a preliminary experiment. We merged two input labels and generated data. Before we generated data for “Place Mat 03”, we merged the label for data generation based on “Tile” with the “Place Mat 03” input label. In the “Tile” label, the index for “Tile” ranged from 0 to 1. The “Place Mat 03” index was 1.

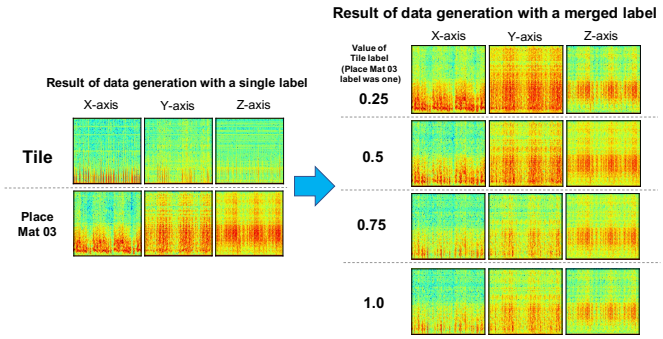

Figure 8. Spectrograms of all labeled generated signals.

Figure 8 shows the results. The two images on the left show the data generated using the standard single labels. The four images on the right show the data generated using multiple labels. The spectrograms change as the label values vary. The greater the value of the “Tile” label, the more mixed the data characteristics become, especially on the X-axis. Thus, the model is likely to generate unknown data if we manipulate the input label. We will determine what types of data the model generates under various conditions.

VII. Conclusions and future work

We used GANs to generate vibrotactile signals. Our GAN is based on WaveGAN [7] and a conditional GAN [13]. We generated three-axis time-series data; earlier work created only one-axis data. The model is smaller than the earlier model. The training was complete in about 46 hours using a general-purpose GPU and PC. Three-axis data can be used for vibrotactile displays that are more elaborate than one-axis pen-type displays. In the user study, we found that vibrotactile stimuli based on generated data were as realistic as stimuli based on training data. In the future, we will deliver real textures using higher-quality vibrotactile displays than the ones used by Saga et al. [3]. We will also explore whether the model can generate unknown data when we manipulate the input label; our preliminary experiment suggests that this is likely. We will examine the data generated when we merge three or more labels.

Acknowledgment

This work was partly supported by JSPS KAKENHI 18H04104G1 (Grant-in-Aid for Scientific Research (A)) and 19K2287900 (Grant-in-Aid for challenging Exploratory Research).

References

[1] A. Abdulali and S. Jeon, “Data-Driven Modeling of Anisotropic Haptic Textures: Data Segmentation and Interpolation,” in Haptics: Perception, Devices, Control, and Applications: 10th International Conference. Springer International Publishing, 2016, pp. 228–239.

[2] M. Strese, Y. Boeck, and E. Steinbach, “Content-based Surface Material Retrieval,” in 2017 IEEE World Haptics Conference (WHC). IEEE, 2017, pp. 352–357.

[3] S. Saga and K. Deguchi, “Lateral-force-based 2.5-dimensional tactile display for touch screen,” in Haptics Symposium 2012. IEEE, 2012, pp. 15–22.

[4] Y. Cho, A. Bianchi, N. Marquardt, and N. 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, ser. UIST ’16. ACM, 2016, pp. 195–205.

[5] I. Goodfellow et al., “Generative Adversarial Nets,” in Advances in neural information processing systems. Curran Associates, Inc., 2014, pp. 2672–2680, and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and.

[6] C. Ledig 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. IEEE, 2017, pp. 4681–4690.

[7] C. Donahue, J. McAuley, and M. Puckette, “Adversarial Audio Synthesis,” in International Conference on Learning Representations, 2019. [Online]. Available: https://openreview.net/forum?id=ByMVTsR5KQ [accessed: 2020-02-29]

[8] J. Engel et al., “GANSynth: Adversarial Neural Audio Synthesis,” in International Conference on Learning Representations, 2019. [Online]. Available: https://openreview.net/forum?id=H1xQVn09FX [accessed: 2020-02-29]

[9] Y. Ujitoko and Y. 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. Springer, 2018, pp. 25–36.

[10] K. Minamizawa, Y. Kakehi, M. Nakatani, S. Mihara, and S. Tachi, “TECHTILE toolkit: A prototyping tool for designing haptic media,” in Proceedings of the 2012 Virtual Reality International Conference, ser. VRIC ’12. ACM, 2012, p. 26.

[11] H. Tomita, S. Saga, H. Kajimoto, S. Vasilache, and S. Takahashi, “A Study of Tactile Sensation and Magnitude on Electrostatic Tactile Display,” in 2018 IEEE Haptics Symposium (HAPTICS). IEEE, 2018, pp. 158–162.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, 2017, pp. 1125–1134.

[13] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” arXiv preprint arXiv:1411.1784, 2014.

[14] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved Training of Wasserstein GANs,” in Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 2017, pp. 5767–5777.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification,” in Proceedings of the IEEE international conference on computer vision. IEEE, 2015, pp. 1026–1034.

 ISBN: 978-1-61208-761-0

[16] M. Strese, C. Schuwerk, A. Iepure, and E. Steinbach, “Multimodal Feature-Based Surface Material Classification,” IEEE transactions on haptics, vol. 10, no. 2, IEEE, 2016, pp. 226–239.

[17] K. A. Lee, G. Hicks, and G. Nino-Murcia, “Validity and reliability of a scale to assess fatigue,” Psychiatry research, vol. 36, no. 3, Elsevier, 1991, pp. 291–298.

 ISBN: 978-1-61208-761-0