International Journal on Advances in Networks and Services, vol 13 no 3 & 4, year 2020, http://www.iariajournals.org/networks_and_services/

Machine Learning-based Classification and Generation of Vibrotactile Information

Satoshi Saga*, Shotaro Agatsuma†, Simona Vasilache†, and Shin Takahashi†
* Kumamoto University, Kumamoto, JAPAN. 8608555. Email: saga@saga-lab.org
† University of Tsukuba, Tsukuba, JAPAN. 3058573. Email: {agatsuma@iplab, simona@, shin@}cs.tsukuba.ac.jp

***Abstract*—In the field of tactile displays, many researchers are developing systems that employ recorded tactile information as an input signal for tactile display. Various tactile information has been recorded from real textures and presents high-quality tactile sensations via the displays. However, collecting, classifying, and generating large amounts of tactile information data under many different conditions with complicated sensors are difficult to realize. Thus, we developed a method of collecting accelerations in haptic behaviors using wireless microcomputers and implemented a Convolutional Neural Network-based classification method of tactile information. We had succeeded in classifying 30 types of data with an accuracy of 88.9%. Furthermore, we proposed to generate unrecorded data under various conditions from recorded data. We construct a data generation model using a Generative Adversarial Network. The model generates unrecorded three-axis, time-series acceleration data from recorded acceleration data obtained by stroking real objects. To evaluate the quality of the data generated, we presented generated vibrotactile information to users via a tactile display. We revealed that the generated data were indistinguishable from real data. Besides mixing and generating data of two or more classes, we generated new, unrecorded data with mixed features of the original classes.**

Keywords–Tactile Information; Machine Learning; Convolutional Neural Network; Generative Adversarial Network.

I. Introduction

Today, many researchers are developing tactile display systems that employ recorded tactile information as an input signal, and these systems present high-quality tactile sensations. To enhance this kind of displaying method, it is necessary to collect and classify various recorded vibration types. Our research proposed a solution to collecting and generating haptic information without complicated devices [1]. In this approach, we collect, classify, and generate only acceleration as tactile information. By using only acceleration data, we collect the information easier than conventional research. Furthermore, employing machine learning-based classification and generation methods, we propose a consistent handling approach of the information for tactile displays.

Many kinds of researches on the collection and classification of recorded tactile information have been performed. To ensure high-quality tactile display, it is necessary to collect and analyze data under various conditions. However, multiple conditions were not addressed in the following works [2], [3]. For example, Strese et al. [3] collected six types of physical data (accelerations, pressures, temperatures, images, sounds, and magnetic field powers) for 108 textures, using a pen-type device. However, there are many more than 108 textures in the real world, and not all conditions, such as stroking directions, contact angles, or pressure force, were explored. However, most of these researches collect tactile information under limited experimental environments using a device that has many sensors. Therefore, it is difficult to collect haptic information outside of the experiment environment, for example, in daily behavior.

On the other hand, we collect, classify, and generate only acceleration as tactile information. Using a wireless microcomputer with an accelerometer, we collect the information easier than conventional research. To classify haptic information, we implemented a classifier using machine learning. The classifier classifies the collected tactile information. It is also used as a search engine for surface material retrieval of the newly collected information to expand the database. We attached a ZigBee-based wireless microcomputer with an accelerometer to the experimenter's finger or pen and performed stroking of various objects. We collected 30 types of accelerations in stroking haptic behaviors. As a machine learning method, we used the Convolutional Neural Network (CNN) to classify the haptic information with high precision and we succeeded in classifying 30 types of data with an accuracy of 88.9%.

Furthermore, realistic surface reproduction is challenging because touching is bidirectional. If the object surface, physical characteristics, or stroking speed differ between the contactor and the contacted object, the induced phenomena will vary. By recording real texture data in many conditions, such as stroking directions, contact angles, or pressure force, you may prepare several real object data. However, an ideal complete dataset would be unimaginably large. Several researches have used GANs (Generative Adversarial Networks) [4] to generate data for tactile displays to solve this problem. Ujitoko et al. [5] employed a GAN for generating time-series data equivalent to real texture. The model consists of an encoder and a generator, and the encoder transformed texture images into labeled vectors. Then the generator generated spectrograms by 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 for simple (i.e., pen-type) vibrotactile displays. However, their proposed system requires high computational cost because the size of the model is large.

Our model is more straightforward than the above model. We generate three-axis acceleration data available for more types of situations (e.g., displaying, analyzing, and recognizing the vibrotactile signals) than one-axis data. Our method eliminates the need to collect vast vibrotactile signal data from various real objects. Instead of the data collection, unrecorded vibrotactile stimulations are created by employing existing recorded data of real textures. This method reduces the collection cost of real data and greatly expands the utility of limited recorded data for vibrotactile displays. We generated new data with the aid of a GAN. GANs generate images that find many applications in super-resolution [6] and audio synthesis; some sounds are very similar to the human voice [7]. GANs can generate high-quality time-series data. Our data generation model is based on WaveGAN [7], which was developed for audio synthesis. We generated nine types of time-series data based on real textures. By using the tablet-mounted vibrotactile display developed by Saga et al. [8], we performed a user study to evaluate whether the generated stimuli were realistic. Besides, we explored whether it was possible to mix the characteristics of two textures by combining two label data types in the input.

Our principal contribution is that we proposed a solution to collecting and classifying haptic information without complicated devices. In this approach, we collected only acceleration as haptic information. Using a wireless microcomputer that has an accelerometer, we collected haptic information easier than conventional research. Besides, by employing a CNN-based machine learning approach, we realized an accurate classifier for 30 types of textures. Furthermore, we generated unrecorded time-series data using a GAN. Our model has a simpler architecture than an earlier model [5] and requires fewer computational resources. We generate three-axis time-series data to display, analyze, and recognize the vibrotactile signals.

The structure of this paper is as follows. This section describes the purpose of our research and our approach. In Section II, we propose a solution to collecting and classifying haptic information without complicated devices. We propose a generation method of unrecorded time-series data using a GAN and describe our proposed GAN model's system architecture and generated data in Section III. Section IV deals with the user study. 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 work.

II. Collection and Classification of Vibrotactile Information

In recent years, several methods for classifying tactile information with several sensor inputs are realized by using machine learning technology. We also considered that it is possible to achieve similar classification with fewer sensors while incorporating such human tactile movements. Then we have proposed a tactile information classification system using only acceleration sensors [9]. Usually, texture is evoked by active human tactile movement and friction between texture and finger. In other words, the acceleration sensor attached near the texture captures information, including both his active motion and the vibration generated from the texture. Therefore, we focused on this acceleration and proposed gathering and classifying the acceleration information collected by the wireless sensors attached near the texture.

A. *Collection of acceleration by a wireless sensor network*

We focus on the acceleration between the user and the texture during the stroking operation, construct a wireless sensor network attached to the texture, and collect and classify the induced acceleration information between them. The system consists of the sensor based on a power-saving compact microcomputer with an accelerometer (Figure 1), and a computer enabling signal identification by machine learning.

Figure 1. Overview of ZigBee microcomputer. Left: TWE-Lite-2525A [10]. ZigBee microcomputer: it includes a 3-axes accelerometer, a battery cell, and a communication module. Right: Overview of data collection.

Figure 2. Packet structure: Δt shows a time, and x_i, y_i, z_i shows consecutive measured values.

The collected tactile information is transmitted to the computer by ZigBee wireless communication and then classified. ZigBee available in Japan uses the 2.4 GHz band, and at this frequency, a maximum of 250 Kbps, the stable transmission is possible. For this reason, the transfer rate is 144 Kbps, so it is difficult to transmit at a high cycle due to the restrictions of ZigBee communication. Therefore, although the maximum measurement cycle in the initial firmware is 33 Hz, we adopted a method in which ten consecutive measurement values are stored in one packet and transmitted (Figure 2). As a result, we achieved the transmission of the 3-axis acceleration sensor's measured values at 330 Hz on ZigBee communication. By attaching these sensors to a finger or a pen, we record the human's active movement and the vibration caused by friction between the texture and the finger.

B. *Design of classifier by machine learning*

In this section, we introduce a method of classifying the acceleration information collected by machine learning. We used a CNN (Convolutional Neural Network), which is widely used in machine learning for images, and constructed a 13-layered network (Figure 3). As an input to this network, we used a 200-point continuous time series with information on three axes of x, y, and z. These sequences were randomly extracted from the collected acceleration information. This division extraction can increase the learning pattern and improve the generalization performance of the model. The output of this model represents the probability that the input data belongs to each class. We used Tensorflow [11] to build this CNN model. Tensorflow is a machine learning library developed by Google. Besides, the model is based on VGG [12], which is a typical configuration of CNN, and the number of convolutions in the convolution layer is reduced to match our input information. To improve accuracy, we increased the number of convolutional layers. A convolution filter with a size of 1×5 was used. The number of filters was 64 for the first and second convolution layers, 128 for the third and fourth layers, and 256 for the fifth and sixth layers. Also, the ReLU function [13] was used as the activation function.

Moreover, generalization performance was improved by doubling the number of filters in the pooling layer. In the pooling layer, max-pooling of 1×2 was used to absorb small data errors. The cross-entropy error was used for the loss function. Adam [14] was used as the weight optimization algorithm. To suppress over-learning, Batch Normalization [15] was applied after calculating each layer's activation function. Also, we used Dropout [16] between the fully connected second layer and the output layer to suppress over-fitting.

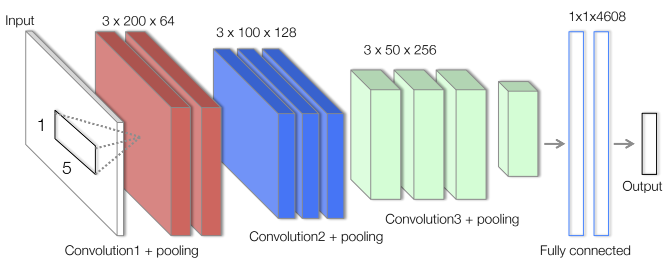

Figure 3. Composition of our CNN model.

C. The evaluation result of tactile information classification accuracy

Here, we show the results of verifying the classification accuracy with the collected tactile information. The ZigBee microcomputer was attached to a finger or pen, and the experimenter collected tactile information by stroking the surface of objects made of various materials. The experiment conditions are shown in Figure 1. The object used in this experiment is shown in Figure 4.

Each texture to be examined is a planar object about 100 mm in length and width. "Carpet1", "Carpet2", and "Carpet3" are part of carpets made from different materials. "Sponge-g" and "Sponge-y" are the front and the back of a household sponge. "Sponge-b" is made of styrofoam. "Stonetile1", "Stonetile2", and "Stonetile3" are stone tiles made from different materials. "Whitetile1", "Whitetile2", and "Whitetile3" are white tiles with different textures. "Woodtile1", "Woodtile2", and "Woodtile3" are wood plates of different textures. Haptic information is collected by stroking these objects with the finger/pen at an almost constant speed. At the surface of each object, acceleration was collected during the stroking movement of going back and forth for three minutes. By performing this operation three times per object, acceleration data for nine minutes per object was collected. Thus, there are 30 types of combinations between the objects and contactors in the experiment. Moreover, the stroking speed also matters. For each combination, the experimenter stroked at velocities of 100 mm/s, 200 mm/s, and 400 mm/s.

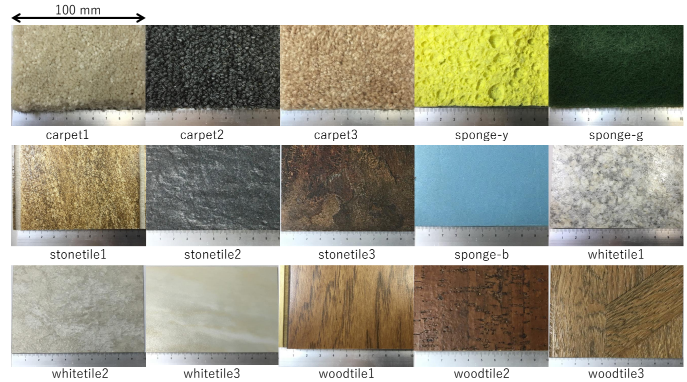

Figure 4. 15 textures. These are plate-shaped objects with 70–100 mm length and 100–130 mm width.

The collected data is classified and evaluated by machine learning using the CNN described above. The input of CNN is 3×200 acceleration data, and the output is 1×30, which is a probability vector representing the class to which the input data belongs. At the time of classification, to confirm the generalization performance of the model created by this CNN, all data was divided into ten parts and 10-fold cross-validation was performed. Table I shows the confusion matrix at 400 mm/s, and Table II shows the classification accuracy for each stroking speed. As can be seen from the tables, each class diagonal component shows a value close to 1, indicating that 30 types of texture information can be classified with high accuracy. The average of all classes was 93.2%. Besides evaluating the effect of different stroke conditions (stroking speed: 100, 200, and 400 mm/s) on the same texture, we combined them in one class and classified the 30 types of textures. The result shows 88.9% of classification accuracy. From these results, we concluded that there is a possibility that the textures can be classified regardless of the stroking speed.

III. Generation of Unrecorded Tactile Information

As we have introduced so far, some researchers have used machine learning based on collected data to classify the textures [2], [3], [9]. However, there are enormous amounts of conditions of the combination between textures and stroking motions, and these studies cannot cover all of these combinations. For example, Strese et al. [3] collected data on various conditions with a pen-type device, although the acute angle against the object is fixed. Thus the data that can be

TABLE I. Confusion matrix of 30 kinds of data classification under 400 mm/s movement.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
A: carpet1-pen	0.94	0	0	0	0.02	0	0.03	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B: carpet1	0	0.92	0	0	0	0.03	0	0	0	0	0	0.01	0	0	0	0.01	0	0	0	0	0.01	0	0	0.01	0	0	0	0	0	0
C: carpet2-pen	0	0	0.92	0	0.05	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0
D: carpet2	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0	0	0.01
E: carpet3-pen	0.02	0	0.02	0	0.95	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F: carpet3	0	0.04	0	0.03	0	0.9	0	0.01	0	0.01	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G: sponge-b-pen	0	0	0	0	0	0	0.96	0	0.01	0	0.01	0	0.01	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H: sponge-b	0	0	0	0	0	0	0	0.92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0.02	0	0.01
I: sponge-g-pen	0	0	0	0	0	0	0	0	0.98	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J: sponge-g	0	0	0	0	0	0	0	0	0	0.94	0	0.03	0	0	0	0.01	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0
K: sponge-y-pen	0	0	0.01	0	0	0	0	0	0.03	0	0.95	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L: sponge-y	0	0.04	0	0.03	0	0	0	0	0	0	0	0.91	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0
M: stonetile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.98	0.01	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
N: stonetile1	0	0	0	0	0	0.01	0	0	0	0.01	0	0.01	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.01
O: stontile2-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.86	0.01	0.02	0	0	0.02	0	0.01	0	0	0	0	0	0	0.08	0
P: stonetile2	0	0	0	0.01	0	0.01	0	0	0	0	0	0	0	0	0	0.92	0	0.03	0	0	0	0	0	0.01	0	0	0.02	0	0	0
Q: stonetile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0.9	0	0.01	0.01	0	0	0	0	0	0	0	0	0.07	0
R: stonetile3	0	0	0	0	0	0	0	0	0	0.02	0	0.01	0	0	0	0.01	0	0.94	0	0	0	0	0	0	0.01	0	0.02	0	0	0
S: whitetile1-pen	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0.02	0	0.87	0.01	0	0.02	0	0	0.03	0.04	0	0	0.01	0
T: whitetile1	0	0	0	0	0	0	0.01	0	0	0	0	0	0.01	0	0.03	0	0.01	0	0	0.85	0	0.01	0	0	0	0.03	0	0	0.06	0
U: whitetile2-pen	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.01	0	0	0.97	0	0	0	0	0	0	0	0	0
V: whitetile2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0.95	0	0	0	0.01	0	0	0.01	0
W: whitetile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0.01
X: whitetile3	0	0	0	0	0	0	0	0.01	0	0.02	0	0	0	0.04	0	0	0	0	0	0	0	0	0	0.89	0.01	0.02	0.01	0	0	0.01
Y: woodtile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0.01	0	0	0	0	0	0.97	0	0	0	0.01	0
Z: woodtile1	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.01	0.01	0	0	0.01	0	0	0	0	0	0.94	0	0	0.01	0
AA: woodtile2-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0.01	0	0	0	0	0	0	0	0	0.94	0	0	0
AB: woodtile2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0.99	0	0
AC: woodtile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.08	0.01	0.02	0	0	0.03	0	0.01	0	0	0	0	0	0	0.86	0
AD: woodtile3	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.03	0	0	0	0	0	0.94

TABLE II. Result summary. "Mixed" is the result of classifying 30 kinds of texture, each of which include data under three different speed.

Stroking speed (mm/s)	100	200	400	Mixed
Classification accuracy (%)	89.0	88.0	93.2	88.9

obtained may change if the angle is changed. In this way, it is unrealistic to comprehensively collect data from an actual object because of the significant number of conditions to consider. The existence of data of conditions that cannot be collected also means that tactile information of objects under those conditions cannot be classified. Therefore, we consider this problem and propose a new method to replace the method of collecting data directly from an object. Instead of an exhaustive direct data collection method, machine learning is used to generate alternative data from the minimum amount of collected data [1]. As a result, the data collection cost can be minimized. Besides, by adjusting the machine learning model, it is possible to generate data that will replace the data that has not been collected. As a first step in realizing the proposed method, we implemented data generation based on machine learning that focuses on the stroking movement's acceleration data.

A. *Tactile data generation by GAN*

We used the GAN [4] as a data generation method. GAN is a machine learning method mainly for generating unknown images, but it is being applied to various fields such as image resolution enhancement, image property synthesis, and voice synthesis. In the field of speech synthesis, previous research has succeeded in synthesizing speech that is almost the same as that produced by a human. On the other hand, the number of studies on processing tactile data using GAN is still small. Ujitoko et al. [5] proposed a GAN model that generates vibration data corresponding to a texture image This model consists of an "Encoder" network and a "Generator" network. The Encoder network converts the texture image into label data. In the Generator network, data is generated using GAN, which is trained with label data and acceleration data collected in advance. Using this method, Ujitoko et al. generated data for nine classes of textures.

In order to generate more effective data, we constructed a machine learning model for vibration data generation based on WaveGAN [7], which is a GAN for speech synthesis. However, WaveGAN does not support multi-class generation and is not suitable for generating various types of data. To deal with this problem, we combined the method of Conditional GAN [17]. In Conditional GAN, by adding label data to the training data and making it learn, the data corresponding to the label data can be specified and generated at the time of data generation. We introduced multi-class generation by introducing Conditional GAN into our GAN. As the label data, we used one-hot vectors, a type of vector having the same length as the number of classes to be trained and having only 0 or 1 as elements. Besides, we processed acceleration data on three axes in consideration of expandability. To prevent the axes' features from being convolved during learning, each axis was set to be convolved independently in the time direction during learning.

The structure of the data generation model is shown in Table III. "Generator" and "Discriminator" in the table indicate the neural network layer structure that constitutes the model. "Input" indicates the input layer, and "Output" indicates the output layer. Each layer between the input layer and the output layer is a hidden layer. In the hidden layer shown here, values propagate from the input layer side to the output layer side. "Kernel Size" shows the shape of the kernel for each convolutional layer, and "Output Shape" shows the shape of the output data for each layer.

B. *Evaluation of tactile information generation model*

Using the constructed model, data generation, and reproducibility verification experiment of the generated data was performed. The wired accelerometer collected the acceleration data to record higher frequency data (1 kHz) during the stroking movement on each texture instead of using the wireless microcontroller. For training data, we used triaxial

TABLE III. Proposed GAN structure. The left shows the configuration of "Generator" and the right shows the configuration of "Discriminator".

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)

acceleration data. An experimenter attached the acceleration sensor to his finger and rubbed the texture in one direction. For recording texture information, we used nine types of textures to collect the data. The texture used for the collection is shown in Figure 5. "Artificial Grass" is the texture of artificial grass with many protrusions, and "Cloth" is the texture of smooth cloth. "Carpet" is the texture of a hard carpet, and "Cork Sheet" is a plate-shaped cork. "Punched Plastic Sheet" is a texture with a lot of punch holes on a smooth board, and "Tile" is a texture of tiles arranged regularly. "Luncheon Mat 01", "Luncheon Mat 02", and "Luncheon Mat 03" are luncheon mats with different surface materials. The overview of the data acquisition is shown in Figure 6.

The hyperparameters used for learning are shown in Table IV. In case the length of the collected data was less than 16,384 points during learning, data was created by repeating each collected data 10 times, and 16,384 points were randomly extracted from the data around 40,000 points and learned. The learning amount was 40 epochs. The generated data is 16,384 points of 3-axis time series data.

TABLE IV. The hyperparameters in our model.

Name	Value
Batch size	64
Phase Shuffle	2
Loss	WGAN-GP
WGAN-GP λ	10
Generator updates per discriminator	2
Optimizer	Adam
	$(\alpha = 1 \times 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.9)$

Figure 5. Textures used in the experiment.

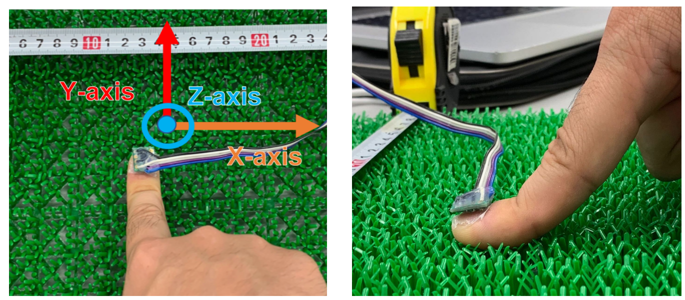

Figure 6. Overview of data acquisition.

The created model can generate nine classes of texture data. To evaluate the reproducibility of the generated data, we took the spectrogram of the generated data and visualized the data. Short-time Fourier transform (STFT) was performed on the generated data and the training data for spectrogram conversion. At this time, a Hamming window (N=256) was used, and the hop size was set to 128. The spectrogram values are normalized from 0 to 1.

An example of the spectrogram of the data generation result is shown in Figure 7. The settings for the spectrogram generation are shown in Table IV. By comparing the spectrograms, you can see the similarity between them. It was difficult to distinguish between training data and generated data for the three textures given in the example. Thus, the data generation that captures the characteristics of the training data was successful. Figures 8 and 9 show the spectrograms of generated data, including the other 6 classes.

IV. TACTILE DISPLAY EXPERIMENT USING GENERATED DATA

In this section, we describe a tactile presentation experiment using generated data. To evaluate the data generated by our model in more detail, we conducted an experiment to present a tactile sensation to the user using the generated data. In this experiment, data were generated using the WaveGAN-based model described in the previous section. The nine classes of collected data described in the previous section were used as training data for generating the data.

In this experiment, two items are investigated. One is to distinguish between tactile presentation using training data and generated ones. If the participant cannot distinguish the two

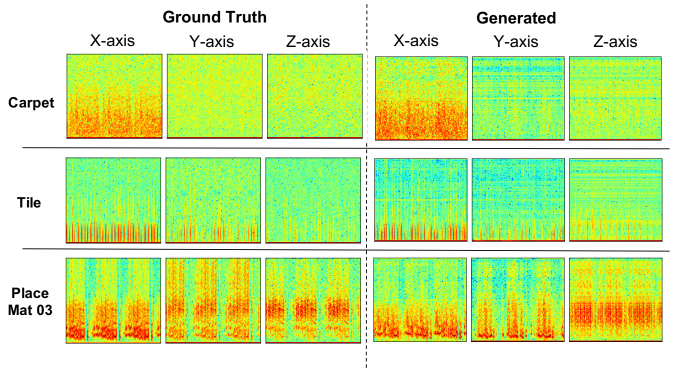

Figure 7. Spectrograms for each labeled class of collected data: Left shows learning data and right shows generated data.

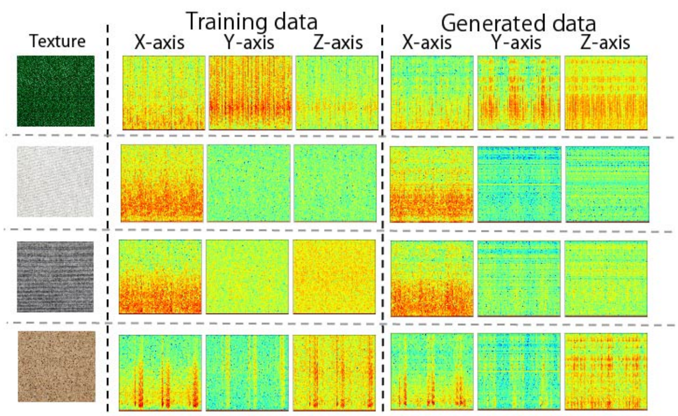

Figure 8. Spectrogram of generated data based on collected data (1). The texture here is the same as the textures shown in Figure 5

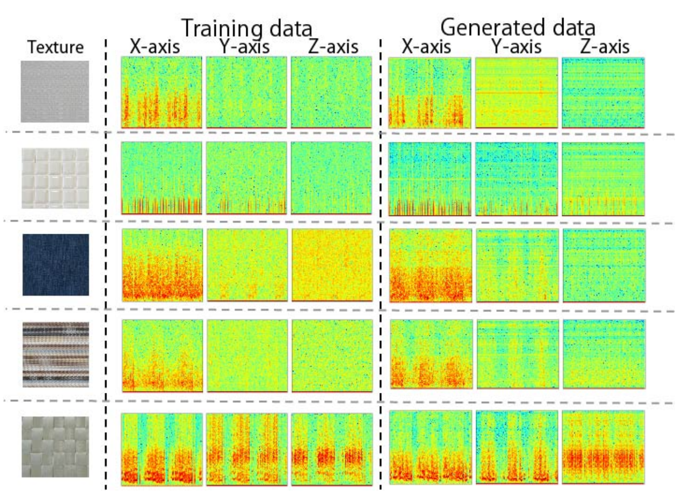

Figure 9. Spectrogram of generated data based on collected data (2). The texture here is the same as the textures shown in Figure 5

data, it indicated that valid data has been generated. The other is a comparison of reality between them. After stroking the actual texture and virtual texture of the two data, we ask the participants to evaluate the reality of the tactile presentation of each data. If the effective data is generated, the evaluation values between the training data and the generated one will show similar values. Based on the results of the experiment, we evaluate whether the model can generate effective data generation or not. For the specific procedure of the evaluation experiment, we decided to use the method of Ujitoko et al. [5]. The tablet-mounted vibrotactile display developed by Saga et al. [8] was used for this tactile presentation.

The participants were ten university students (8 males, 2 females, all in their 20s) in the experiment. The Ethics Review Committee has approved this experiment of the University of Tsukuba (Review approval number, 2019R299). At the beginning of the experiment, the participants filled out a consent form.

A. Experiment procedure

Tactile stimulation is presented in the two rectangle areas, A and B, on the tactile display, and the participant is asked to stroke the tactile display along the area. A moving target was displayed on the touchscreen and showed a 500 mm/s of stroking movement during the experiment. The participants were asked to adjust their stroking movement to follow the target. Either tactile sensation derived from training data or the generated data is presented in A or B. We ask the participants to answer which stimulation is the tactile presentation derived from the generated data. Figure 10 shows an overview of the experiment. Figure 11 shows the displaying interface in this experiment.

Figure 10. Overview of tactile presentation experiment.

The area the generated data is displayed in was randomly determined for each trial. After the experiment is completed, an evaluation is performed based on the value of a 100 mm long Visual Analog Scale (VAS) [18]. Using the VAS, we collect

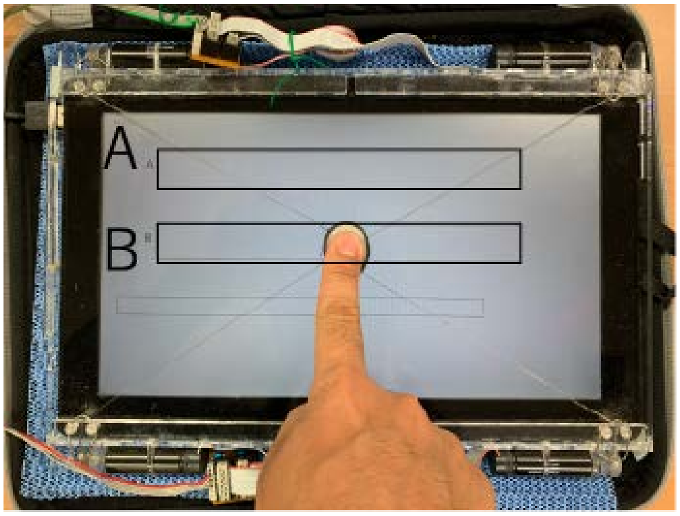

Figure 11. Tactile display during the experiment.

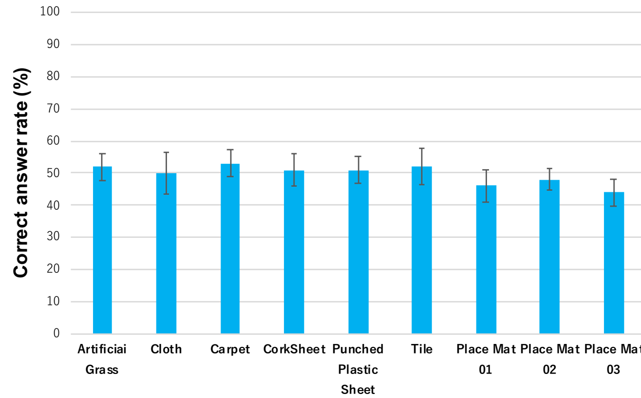

Figure 12. Correct answer rate of distinguishing task between generated data and training data for each texture.

the evaluated answer of each stimulus from the participants. Ten trials were performed for each texture, using these two surveys as one trial. A questionnaire survey was conducted to obtain the participants' opinions about the experiment after completing all trials. It took about one hour to complete the experiment.

B. Results and discussion

In the following, we describe the experimental results. Figure 12 shows the correct answer rate of the experiment that distinguishes the tactile presentation of the generated data from the training one. These values are the average of the results of all participants. If the result of the correct answer rate is close to 50%, it shows that the participants cannot distinguish between the generated data and the training data. That is, it shows the realization of effective data generation that reflects the characteristics of the training data.

As you can see in Figure 12, the value of any texture is almost 50%. Therefore, the participants could not distinguish between the presentations in the training and the generated data. In the post-experimental questionnaire, almost all par官ticpants answered that they could not distinguish between the presentation by the training data and the generated data. From these results, we confirmed the possibility that our proposed model can generate data close to the actual acceleration data. From the detailed result of each texture, most participants showed a correct answer rate of 40% to 60%. Especially for the results of the "Carpet" texture, 7 out of 10 people showed a correct answer rate of 50%. This result indicates that our model may be especially useful for data generation based on rough texture data such as "Carpet" .

Next, Figure 13 shows the results for the presentation's reality values by the generated data and the training data. These values are the average of the results of all participants. The closeness of the evaluation results between reality values of the generated data and the training data shows effective data generation achievement. Figure 13 shows that the presentation's reality values by the generated data and the training data are almost the same for all textures. To verify whether there is a significant difference between the results of the generated data and the training data, Student's t-test was performed on the pair of data for each texture, and no significant difference was found for all textures ($p > 0.05$). From this result, by generated data, we succeeded in presenting tactile sensation with the same degree of realism as that by training data. In the previous study of Ujitoko et al. [5], they found a significant difference in some textures. Thus, our method had generated higher quality data than the previous study.

Looking at Figure 13, we obtained values between 50% and 60% for textures other than "Cloth" and "Tile". According to an experiment conducted by Saga et al. [8], the reality values were between 50% and 70% by using the recorded vibration.

Our results agreed with the results of Saga et al. [8], and we found that the generated data could reproduce the performance of the recorded data sufficiently. Here we consider two textures with low reality values. The display we used this time effectively reproduces a rough tactile sensation because it presents vibration to the fingertips but is not suitable for reproducing a smooth tactile sensation like "Cloth". Since "Cloth" has the texture of cloth, the tactile display used this time had difficulty in presenting it. In the future, it is necessary to investigate using a tactile display that excels in smooth tactile sensation.

Regarding the "Tile" results, because the change in acceleration was small, the participants felt little vibration. The reason for the slight difference in acceleration is the shallow unevenness of the Tile. Although the reality value was low in the result of "Tile" this time, we considered improving this value by using more precise tactile displays. Since there was almost no difference in reality between the generated data and the training data, and data close to the training data can be generated for "Tile" (Figure 7), we confirmed that the data generation was successful.

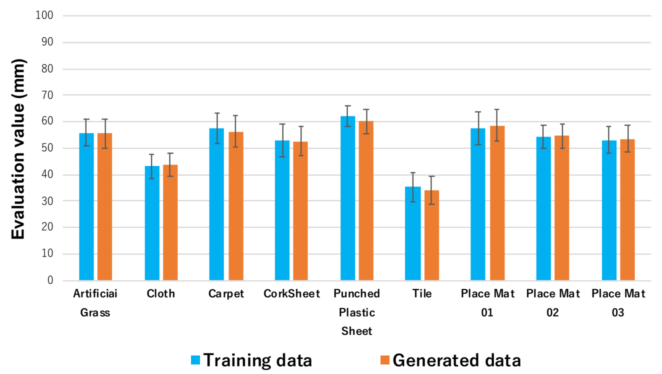

Figure 13. Reality evaluation value for each texture in tactile presentation experiment. The blue graph results from the tactile presentation using the training data, and the orange graph results from the tactile presentation using the generated data.

V. DATA GENERATION BY MERGING TWO CLASSES

This section confirm the data generation ability of an unknown class by the created 9-class generation model. The unrecorded data was generated by merging two classes and specifying two input labels instead of specifying one class for the created model. Figure 14 shows the schematic diagram of the label synthesis of tactile information. We will look at the result of merging the classes of “TIle” and “Place Mat 03” and specifying them. When “Tile” is specified in the generation model, the elements of “Tile” are specified as 1 in the input vector. For an unrecorded class generation, we set the element of “Tile” between 0.0 and 1.0, and “Place Mat 03” to 1. Figure 15 shows the spectrogram of each generated data. The spectrograms of “Tile” and “Place Mat 03” are shown on the left side of Figure 15, and the spectrogram of generated data with a merged element is shown on the right. It can be seen that as the element is increased from 0.0 to 1.0, the features are highly mixed, especially on the X-axis. In other words, it can be seen that unknown data can be generated by changing the input elements.

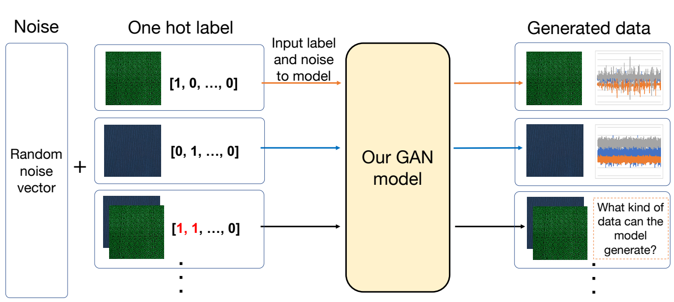

Figure 14. The schematic diagram of the label synthesis of tactile information

In addition, to investigate whether the difference of the texture used for data generation affects the result, we generated the 3-axis acceleration data obtained from the nine classes of texture shown in Figure 5. This time the scale of the input vector is fixed to 1.0. Thus the element of “Tile” was set in the range between 0.0 to 1.0, and the other element was set in the range between 1.0 to 0.0. The following Figures 16, 17, 18 show the results. Figure 16 shows the results for combinations of “Tile” and “ArtGrass”, “Tile” and “Cloth”, and “Tile” and “Carpet”. Figure 17 shows the results for “Tile” and “Cork”, “Tile” and “Punched Plastic Sheet”, and “Tile” and “Place Mat 01”. Figure 18 shows the results for “Tile” and “Place Mat 02”, and “Tile” and “Place Mat 03”.

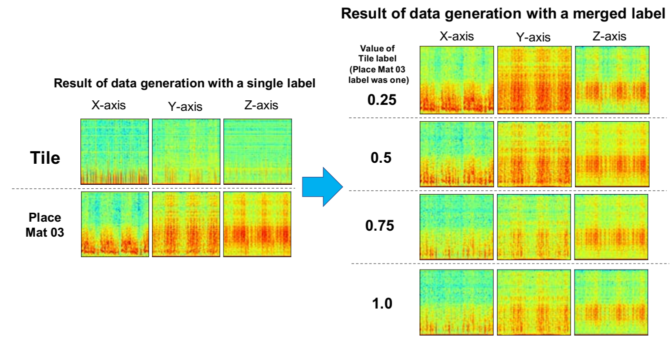

Figure 15. Spectrogram of each generated data. Left shows single class, right shows mixed class spectrograms.

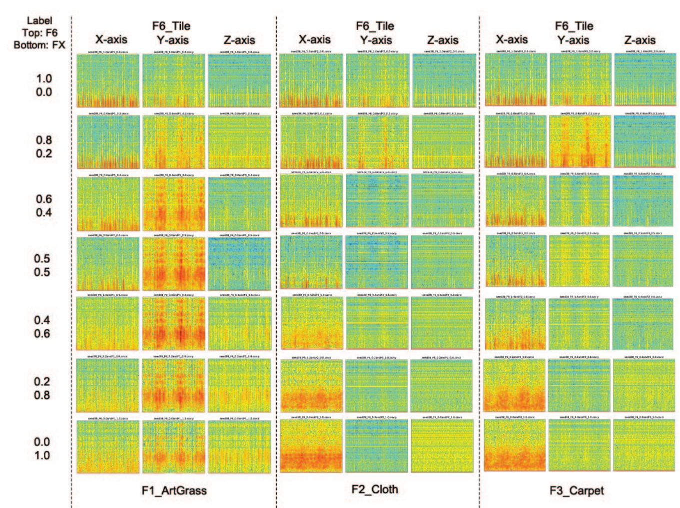

Figure 16. The generated data’s spectrograms combined the “Tile” label and the label of another texture (1). This figure shows the results for combinations of “Tile” and “ArtGrass”, “Tile” and “Cloth”, “Tile” and “Carpet”.

In Figures 16, 17, and 18, the upper number shows the element value of Tile, and the number below is the element value of the other texture. From the results, it can be seen that the data that strongly reflects the feature of the texture with the higher label value is generated in any combination. Similar to the result in Figure 15, this tendency is remarkable,

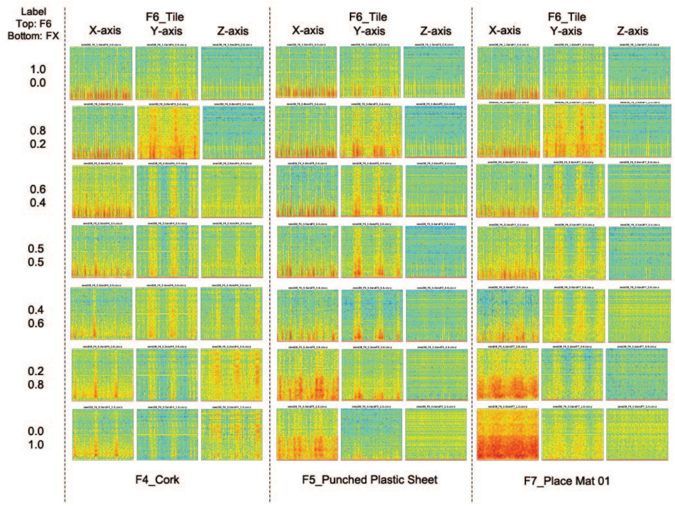

Figure 17. The generated data's spectrograms combined the "Tile" label and the label of another texture (2). This figure shows the results for combinations of "Tile" and "Cork", "Tile" and "Punched Plastic Sheet", "Tile" and "Place Mat 01".

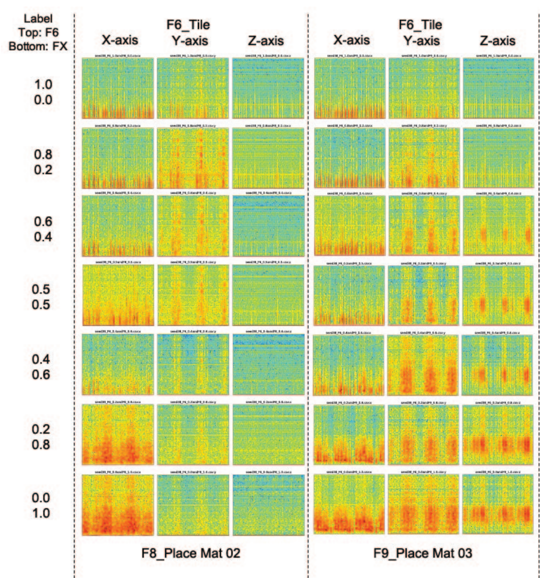

Figure 18. The generated data's spectrograms combined the "Tile" label and the label of another texture (3). This figure shows the results for combinations of "Tile" and "Place Mat 02", "Tile" and "Place Mat 03".

especially on the X-axis. From the result, we found that data synthesis is possible even when the texture is other than Place Mat 03. When the two textures' label values are the same, the "Tile" texture characteristics are often strongly represented in the generated data. The "Tile" texture feature is that the strong and weak regions of the spectrum are finely repeated, but this feature appears in the generated data even when the "Tile" label's value is small. The "Tile" feature appears in the generated data from the case where the "Tile" label element is over 0.4 in all the results except the "Cloth" and "Carpet" results in Figure 16. From this, we found a feature that easily influences the generated data when the labels are combined.

From the result of the generation experiment described in this section, we found that new data was generated with mixed features of two textures by manipulating the label input to the GAN model. We also found that changing the rate of each element can control the interpolated characteristics of each texture. By applying the method, it will be possible to design and generate unrecorded information. For example, by inputting the numerical value of the speed or the pressure, the model can generate the data accordingly. In the future, we plan to conduct a detailed investigation of how the product will be affected when data is generated with variable numerical labels. Based on the results, we will consider constructing a new GAN model that assumes variable numerical labels.

VI. Conclusion and Future Work

This paper proposed a solution for collecting and generating haptic information without complicated devices [1]. In this approach, we collect, classify, and generate only acceleration as tactile information. By using only acceleration data, we collect the information easier than in conventional research. Furthermore, employing machine learning-based classification and generation methods, we propose a consistent handling approach of the information for tactile displays. By using the ZigBee-based microcomputers and implementing a CNN-based classification method of haptic information, we succeeded in classifying 30 types of data with an accuracy of about 88.9%.

Furthermore, we developed a method to generate unrecorded data under conditions differing from those at the initial recording time. We constructed a data generation model using a GAN. The model makes simple calculations and generates unknown data from recorded acceleration data obtained by stroking 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.

Moreover, using GAN, which is based on the method of voice generation, as the method of generating tactile information, we realized effective data generation with a simpler machine learning configuration than previous studies. By creating a vibration data generation model using GAN and generating 3-axis data, we succeeded in generating information close to the actual acceleration sensor's information. To evaluate the quality of the generated data, we devised a string-based tactile display and presented generated vibrotactile information to participants. The participants reported that the generated data were indistinguishable from real data. Besides mixing and generating data of two or more classes, we generated unrecorded data with mixed features of the original classes.

In the future, we aim to construct a system that enables a lot of tactile information processing by making these collection, classification, and generative models more versatile.

ACKNOWLEDGMENTS

This work was partly supported by JSPS KAKENHI Grant Number 19K22879 (Grant-in-Aid for Challenging Exploratory Research) and 18H04104 (Grant-in-Aid for Scientific Research (A)).

REFERENCES

[1] Shotaro Agatsuma, Junya Kurogi, Satoshi Saga, Simona Vasilache, and Shin Takahashi. Simple Generative Adversarial Networkto Generate Three-axis Time-series Data for Vibrotactile Displays. In *Proceedings of International Conference on Advances in Computer-Human Interactions, ACHI 2020*, pp. 19–24, March 2020.

[2] 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, EuroHaptics 2016, London, UK*, pp. 228–239. Springer International Publishing, 2016.

[3] Matti Strese, Yannik Boeck, and Eckehard Steinbach. Content-based surface material retrieval. In *World Haptics Conference (WHC), 2017 IEEE*, pp. 352–357. IEEE, 2017.

[4] 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*, pp. 2672–2680, 2014.

[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] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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*, pp. 4681–4690, 2017.

[7] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. GANSynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations*, 2019.

[8] Satoshi Saga and Ramesh Raskar. Simultaneous geometry and texture display based on lateral force for touchscreen. In *Proceedings of IEEE World Haptics 2013*, pp. 437–442, Apr. 2013.

[9] Shotaro Agatsuma, Shinji Nakagawa, Tomoyoshi Ono, Satoshi Saga, Simona Vasilache, and Shin Takahashi. Classification method of rubbing haptic information using convolutional neural network. In *Proceedings of International Conference, HCI International 2018*, pp. 159–167, July 2018.

[10] Mono Wireless Inc. TWE-Lite-2525A. https://mono-wireless.com/jp/products/TWE-Lite-2525A/index.html (accessed on 2020.11.20).

[11] Google Inc. Tensorflow. https://tensorflow.org/ (accessed on 2020.11.20).

[12] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323, 2011.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[15] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

[16] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

[17] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[18] Kathryn A Lee, Gregory Hicks, and German Nino-Murcia. Validity and reliability of a scale to assess fatigue. *Psychiatry Research*, Vol. 36, No. 3, pp. 291–298, 1991.