vCapTouch: Interactive Touch Sensing Data Synthesis for Hand Gesture Recognition Based on Digital Twin

Chengshuo Xia, *Member, IEEE*, Qingyuan Peng, Zeyuan Fan, Tian Min, Daxing Zhang, and Congsi Wang, *Senior Member, IEEE*

***Abstract*—Touch sensing is a prominent pillar technique in various human–computer interactive scenarios, especially when touchscreen-based capacitive touch sensing has become a representative in user-end electronics. An intelligent touch-sensing system captures the capacitive touch-sensing images to recognize the objects via machine learning techniques. However, collecting the training dataset is usually laborious and time-consuming, requiring specific coding skills and knowledge. In this article, we introduced *vCapTouch*, a data generation method to synthesize the touch sensing data, which can be directly employed to train a machine learning model and recognize the real touching behavior, significantly lowering the need for real dataset collection. The presented method is primarily based on the idea of the digital twin. We implemented the method with Unity3D, a game engine that enables high interactivity, is easy to use, and has a low cost. We evaluated the proposed method on eight users with different touch screen devices and proved the feasibility of synthesizing the touch sensing data.**

***Index Terms*—Data synthesis, digital twin, hand gesture, touch sensing.**

I. Introduction

BY CAPTURING natural and fast interaction behaviors, touch sensing has become a fundamental interaction paradigm for building interaction systems [1]. As the gradual transition from resistive-based to capacitive-based, touch sensing has been widely used in end-user devices, including smartphones, tablets, and smartwatches, and has delved into hundreds of example applications on a daily basis [2]. As one of the core technologies for human–computer interaction (HCI), touch sensing-based research has been continuously extended to recognize daily objects, and body limbs have been applied in games and entertainment technologies [3]. The field has attracted extensive attention, and various degrees of technical improvements have been proposed for its core technologies, such as recognition accuracy and embedded development. Mainstream touch sensing data processing approaches typically integrate capacitive electrode data from the touchscreen into 2-D matrix or grayscale images for target object recognition [6]. It follows the typical machine-learning route by collecting the relevant capacitive electrode data, training the machine-learning model, and deploying it in the embedded system. This approach can effectively build intelligent interactive interfaces, enabling users to operate smart devices and computing systems using simple gestures or classify different objects for context-aware calculation.

Intelligent touch-sensing technology provides essential support for the further realization of pervasive interfaces. Existing research efforts constantly propose lowering the development threshold of touch sensing-based interactive systems to achieve a more technologically democratic user-based development approach [4]. For example, multitouch Kit [4] demonstrates a do-it-yourself (DIY) capacitive touch sensing toolkit with a commodity microcontroller. Using simple hardware devices and sample code on the Arduino, the nonprofessional user could easily make up their own capacitive touch-sensing system.

Most of the research driving the tooling of touch sensing systems has concentrated on improving signal acquisition and processing, as well as the electrode fabrication. However, little work has focused on the improvement of data processing and the integration of machine learning model development. As a data-driven system, collecting the training dataset is essential to ensure key prior knowledge for a touch sensing system. However, the dataset collection process usually is time-consuming. It would demonstrate the inconvenience and difficulty of adding or altering the recognized object for an already trained model. Thus, it forms a higher technical barrier to creating a flexible, customized, and interactive touch-sensing system.

The data synthesis technique gives a good opportunity to accelerate the development of touch-sensing systems. Relying on the generative model, it is capable of producing synthetic capacitive touch-sensing images in recognition system training, e.g., the generative adversarial network (GAN) utilized in the work of [5]. Though the generative model is able to provide the dataset to a certain extent, the model itself is still data-driven and requires real data samples to be deployed. The

Received 23 January 2025; revised 5 March 2025; accepted 18 March 2025. Date of publication 21 March 2025; date of current version 27 June 2025. This work was supported in part by the China Postdoctoral Science Foundation under Grant 2024M762548, and in part by the High-Level Innovation Institute Project of Guangdong Province under Grant 2021B0909050008. *(Corresponding author: Chengshuo Xia.)*

Chengshuo Xia, Qingyuan Peng, Daxing Zhang, and Congsi Wang are with the Guangzhou Institute of Technology, Xidian University, Xi'an 710126, China (e-mail: xiachengshuo@xidian.edu.cn).

Zeyuan Fan is with the Jiangsu University of Science and Technology, Zhenjiang 212003, China.

Tian Min is with the Graduate School of Science and Technology, Keio University, Shinjuku 160-8582, Japan.

Digital Object Identifier 10.1109/JIOT.2025.3553561

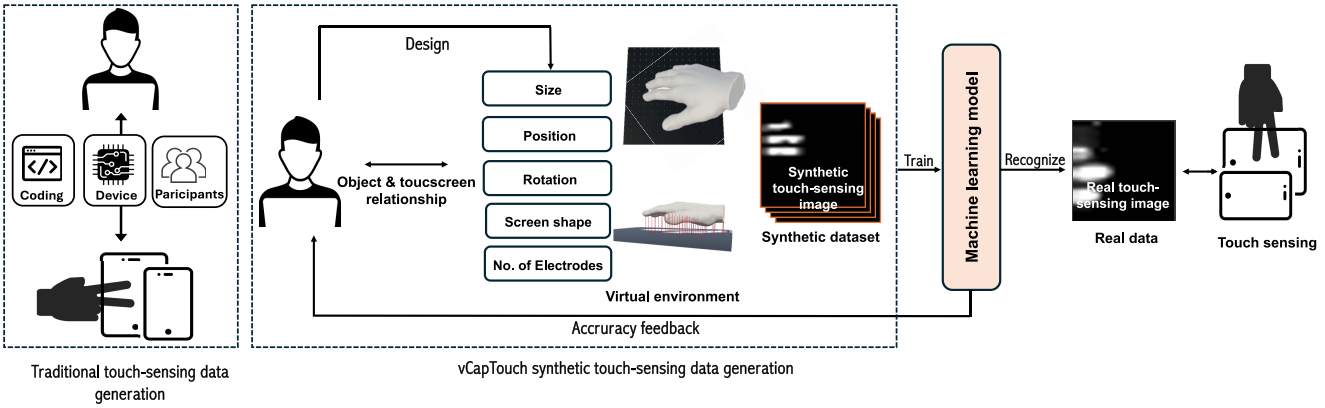


Fig. 1. Traditional touch sensing data generation requires enormous effort on code work, device, and user participation, which is costly and time-consuming and constructs higher requirements of the developer's skills. Our approach is built based on a virtual environment and produces synthetic images, which can be employed to train a machine learning model and recognize the real touching event. Following the digital twin paradigm, the developer could access the data synthesis process by setting different sizes, positions, and rotations of 3-D models and designs with different shapes and simulated electrodes/pixels of the touchscreen.

model's generative capacity is limited when facing the new object's capacitive touch-sensing image. Therefore, exploring a flexible and efficient capacitive touch-sensing image generation method would promote the application of capacitive touch sensing in a broader set of scenarios.

Thus, in this article, we introduced a zero-shot touch sensing data generation method, *vCapTouch* (Fig. 1). We employed the physics simulation in Unity3D to generate the touch-sensing images. Taking capacitive touch sensing as an instance, Ray-based collision detection is utilized to sense 3-D objects virtually, such as different hand gestures. The 3-D object's surface shape information can be sparsely detected in Unity3D, which simulates the sensing capability of sparse capacitive electrodes in a touchscreen. We also designed an augmentation scheme to further enhance the fidelity and usability of synthetic capacitive touch-sensing images. The generated synthetic capacitive touch-sensing images are subsequently used to train the machine-learning model. We evaluated our method by recognizing hand gesture images from 8 participants and comparing the result of the proposed method with other state-of-the-art approaches. The contributions of this article are as follows.

1) We present a novel touch sensing data synthesizing method. The ray-based collision detection and synthetic data augmentation are employed to obtain the synthetic dataset for touch sensing.
2) We validate the feasibility and superiority of the proposed method and the generalizability to different types of devices.

II. Related Work

A. Touch Sensing-Based Interactive IoT System

Utilizing touching detection is one of the most efficient interaction approaches between the user and the computing system, which has already been deployed in various end-user electronic devices [1]. Based on the most common technique, capacitive touch sensing has been attracting a lot of attention from researchers to investigate not only improving the sensing performance [6], but also promoting the practical and ubiquity of capacitive touch sensing [3], [4], [5]. To increase the commercial-off-the-shelf (COTS) device experience at the user end, researchers have explored the different operation approaches with force-based [7], nail-based [8] touch sensing and so on. In addition to the finger area, expanding the interaction area of capacitive touch sensing most concentrated on the human body skin, [9] as well as the artificial skin [10]. Moreover, improving the fabrication of capacitive electrodes and systems would facilitate a wider application of capacitive touch sensing, such as detecting everyday objects [11]. Multitouch Kit introduced a custom capacitive touch-sensing prototype with a commodity microcontroller [4]. Lower development costs have led to a gradual move toward lower development thresholds for capacitive touch sensing, and several toolkits have been created better to promote customized design in both hardware and software aspects. For example, Steuerlein and Mayer [5] presented the GAN-based deep learning toolkit to help design conductive fiducial tangible applications.

B. Data-Driven and Cross-Modal Data Synthesis

Data scarcity has been recognized as a critical issue in machine learning systems. Particularly in sensor-based machine learning systems, the shallow and small-scale dataset restricts the feasibility of mining the deep and generalized data representation. Data synthesis is an effective approach to expand the size of the collected dataset and enhance the performance of the trained model [12]. Classical approaches utilize the data-driven mechanism to learn the data distribution. The representative work is recognized as a GAN model with massive success in image generation [13]. To expand its application, the nonvision sensor-based GAN model could also generate various modal sensing data, including the inertial measurement unit (IMU) signal [14], WiFi signal [15], radar signal [16], capacitive touch-sensing image [5], and among others. Additionally, recent large fundamental models also explored an alternative way to produce the simulated data

with language input, such as the text-to-image and text-to-motion [17].

However, the involved data-driven approach still needs significant prior knowledge of the training dataset. The recent cross-modal approach provides a significant solution that transforms the extensive video data into other nonvision dataset generation [18], [19], [20]. It typically extracts the objects and humans from the video and calculates the related nonvision sensor signal. It allows the virtual elements to be manipulated as the real ones in a digital twin way. The relevant sensor signal could be simulated by accessing and calculating kinematic data, for example the IMU signal [20], [21], [22], radar signal [18], [19], infrared distance signal [21], lightness signal [23], and so on. The virtual environment-based data synthesis method would enable more explainable factors in data generation and allow an interactive process to create the sensor data, which is suitable for novice and learning for inexperienced users.

C. Assisting the Recognition System Development

Human motion and daily object recognition assist the computing system in understanding the context information from the user end and are capable of constructing the natural and pervasive interaction experience. Researchers have been studying the various approaches to facilitate the object recognition system development, including the computational design [24], [25], [26], hardware fabrication [27], machine-learning-based toolkit [28], [29], [30], [31], [32], data accessibility and visualization [33], [34], and so on. Building the specific user interface to recognize the user behavior requires cumbersome calculation steps, works thus are emerging in assisting the computational process for the developers, including the ergonomic constraints [24], augmented reality design [25] and so on. Facilitating the essential object's location selection and customized design draws intensive attention to such systems. For example, the FabHandWear [27] presented a hand wearables fabrication tool with hand model parameterization and components placement selection. To help the sensor usage in building the recognition interactive system, works also concentrated on the accessibility and visualization of the sensor's measurement, such as visualizing the data with interactive system prototyping in SensorViz [33] and utilizing the game engine to create the sensor simulation with BlenSor [34].

In addition, relying on the machine learning technique can build an intelligent recognition system for interaction. Interactive machine learning is important in enabling more users to engage in the machine-learning-based system development [35]. For example, the gesture-aware annotation for vision dataset building [38], augmented reality objects prototyping [39], and pose authoring in video [31]. A more generic development method of the toolkit has been proposed to facilitate machine-learning-based motion and object recognition. Many works on the toolkit have focused on integrating an end-to-end pipeline containing the necessary steps for system building with data collection, annotation, cleaning, dataset splitting, modeling training, and testing. This approach is feasible for multimodal sensing technologies, including the EMG [29], IMU [36], voice [37], and camera [28], [30], [31], [32]. These systems lower the development threshold and allow a DIY design for more customized scenarios.

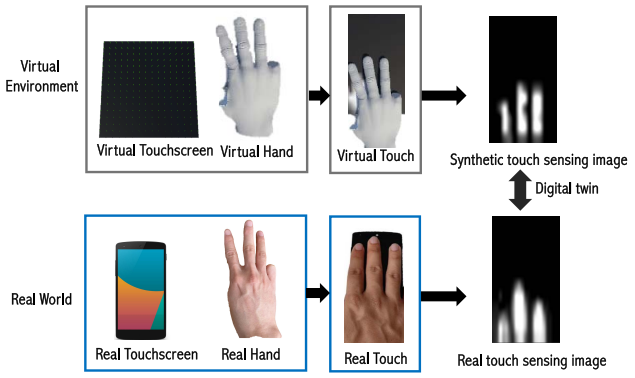


Fig. 2. Digital twin process of synthetic touch sensing data and real touch sensing data generation.

Therefore, currently, the synthetic touch sensing data technique is a data-driven method that uses real data as the core to generate relevant synthetic data. This method still requires expensive resources and time to obtain real datasets and their data distribution. Nevertheless, our method uses digital twin technology to create corresponding electrode points and touch gestures in a virtual environment to obtain corresponding synthetic touch sensing data as a training set. As far as we know, this is the first work to use the digital-twin method to obtain a synthetic touch-sensing dataset. Since the entire process is completed in the virtual environment, it has the advantages of high interactivity, low cost, and ease of use.

III. Virtual Touch Sensing Data Generation

A. vCapTouch: Digital Twin-Based Touch Sensing Data Synthesis

Generally, the digital twin is a virtual representation that reflects the real physical object and has been widely applied to the manufacturing field. *vCapTouch* applies this idea to produce the synthetic touch sensing data, allowing the whole data generation process to be more intuitive and interactive. Fig. 2 illustrates the process of a digital twin for generating synthetic touch sensing data. We reconstructed the virtual hand and virtual touchscreen elements. Subsequently, the virtual hand could be placed above the virtual touchscreen to simulate the real hand contacting the touchscreen, which is able to generate synthetic touch sensing data.

B. Sensing Pipeline of vCapTouch

As the capacitive touchscreen has become the mainstream in user-end electronics, we mainly adopted the capacitive touch-sensing process as the simulated object. In the practical touchscreen, the human skin, as a large conductor, can build a virtual ground and absorb the electric charge. When the human finger contacts the touchscreen's capacitive electrodes, it will create an additional capacitor and alter the original electric field distribution to increase the capacitance values. Access

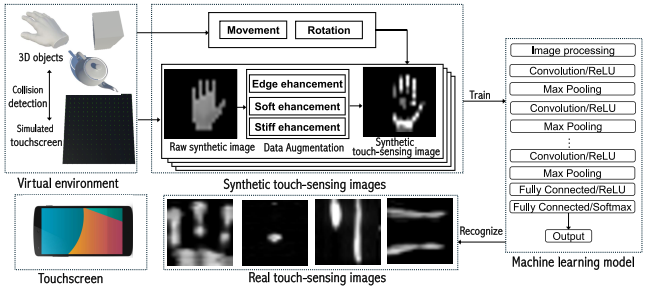


Fig. 3. Whole pipeline of *vCapTouch*. The simulation of touch sensing is performed in the virtual environment, which maintains high interactivity and intuition.

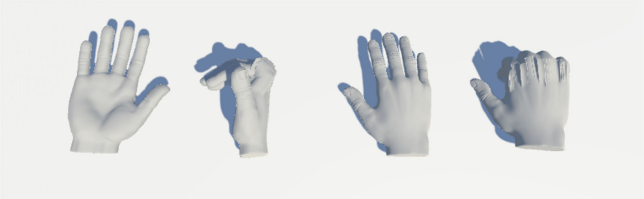


Fig. 4. 3-D hand model with different gestures in Unity3D. And each hand model contains the mesh and collider elements.

to the touchscreen's capacitive value allows the creation of the capacitive matrix and the relevant grayscale image by normalization into 0 to 255. Different contact area sizes of the human body would lead to various capacitance values.

We adopted this idea and realized the touch sensing data synthesis in the virtual environment, Unity3D, to form a digital twin-based simulation. Fig. 3 shows the pipeline of *vCapTouch* synthesizing the touch sensing data. By referring to the real situation, we employed the virtual object and virtual cubes simulating the electrodes to generate the sensing matrix, which can be converted to the capacitive touch-sensing image. Unlike other mechanism-based simulations (e.g., simulating the electric field variation) and data-driven simulation (e.g., GAN-based), the main advantage of *vCapTouch* is its high interactivity, flexibility, and intuition characteristics. By employing the virtual objects, users may reference the real touching behavior to obtain the synthetic touch sensing data without specific coding skills, and prior knowledge could determine the screen sizes, electrode numbers, different touching patterns, and recognized categories.

C. 3-D Objects in Virtual Environment

The import of 3-D objects plays a significant role in synthesizing touch sensing data. To create the entity of touching behavior, we utilized 3-D objects with mesh colliders in Unity3D (Fig. 4). Since the representative touching entity is the human hand, we mainly used the 3-D hand model to introduce our method in the following sections. By accessing and controlling the different hand joints, it is able to perform various hand gestures, such as a fist or a peace sign.

D. Ray-Based Simulation to Generate the Raw Touch-Sensing Image

In addition to the hand model object conducting the touching behavior, the other critical element to be simulated is the touchscreen. We utilize the collision detection based on *Raycast* functionality in Unity3D to form a detection matrix. Specifically, we create cubes to simulate each electrode as the detection point and arrange the cubes into the corresponding matrix (e.g., 25 * 17). To entitle the value of each cube, the *Raycast* functionality is performed by each cube. When the 3-D object is located on top of the cube matrix, each cube will return a distance information between the cube and the 3-D object's surface. Since the 3-D object's surface has different geometries, a distance matrix with different values is subsequently formed as the simulated capacitive matrix. In a real situation, capacitive touch sensing is able to perceive the information about the touched object because the touched area would cause different capacitance values. With different hand gestures, the touchscreen electrode's capacitance values could reflect the geometry shape information from the touched object. Therefore, we applied this way to virtually sense the contacting object's geometry shape information. In the virtual environment, the distance information could also reflect the 3-D object surface geometry shape information according to various distance values. For example, the place without the object will not generate distance values related to the nontouched area and will not have a capacitance value that varies.

After obtaining the simulated capacitive matrix, via normalization, the capacitive matrix could be converted into 0 to 255, which could be a raw synthetic capacitive touch-sensing image. Fig. 5 illustrates ray-based collision detection in Unity to simulate the capacitive touch-sensing image. The detailed evaluation of synthetic image quality is performed in Sections IV, IV-C, and IV-D. The generated capacitive touch-sensing image contains the rough outline of the imported 3-D object's surface, e.g., the palm and fingers for a 3-D hand model.

During this process, the generated touch-sensing images can be labeled and annotated by indicating various hand gestures. The placement of the virtual hand model could be flexibly adjusted to form different hand gesture classes for further training and recognition.

E. Augment the Raw Collision-Based Synthetic Image

In the virtual environment, collision-based detection is capable of obtaining the 3-D hand's rough outline to form a raw touch-sensing image. Compared with the real situation, since the human hand's skin is soft and the joint bone/skeletal point is stiff, the produced real capacitive touch-sensing image typically has a lighter point with a skeletal point and a smaller value of soft tissue. Additionally, based on the sensing principle, the stiff touching point will cause a higher value and affect the surrounding capacitance variation because of capacitor coupling. Therefore, to enable the raw synthetic capacitive touch-sensing image to be close to the real capacitive touch-sensing image, we developed an augmentation scheme to enhance the fidelity of synthetic touch sensing data. The augmentation scheme could be divided into three parts:*stiff*, *soft enhancement*, and *edge*. (Figs. 6 and 7).

Soft Enhancement: Due to the soft tissue and stiff skeletal point is prone to form a concave area in the soft tissue

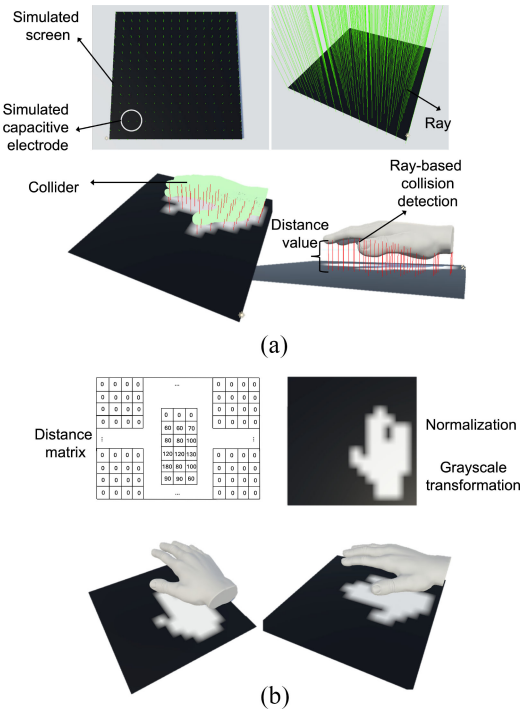


Fig. 5. Illustration of ray-based collision detection in Unity to generate the raw synthetic capacitive touch-sensing image. (a) shows the basic detection principle. The simulated screen contains several cubes (16 * 16 as an example), and the *Raycast* function was applied to the cube to recognize a simulated capacitive electrode. The shoot ray could detect the collision between the cube and the 3-D hand and return the relevant distance value. Thus, each cube would have a unique distance value and form a distance matrix in (b). Following the normalization process, the matrix is converted into a grayscale image, which is the raw synthetic capacitive touch-sensing image in *vCapTouch*.

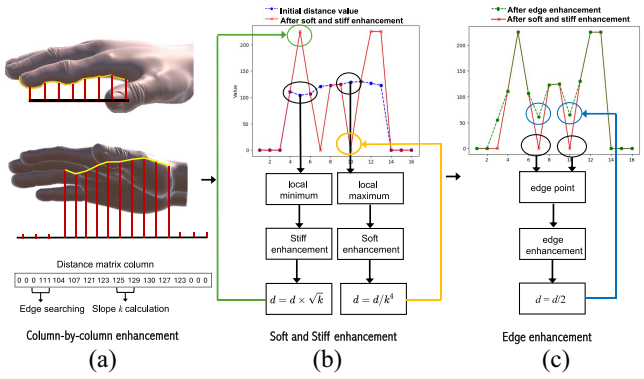


Fig. 6. Illustration of the augmentation scheme for the raw synthetic image. The primary parts including the soft, still, and edge enhancement. (a) shows one column of the distance matrix. The slope is calculated to find the edge as well as the local minimum and maximum point. (b) Local minimum point of the slope is recognized as the still area and is applied by stiff transformation to increase the distance value. And the local maximum point's value would be decreased. (c) Edge enhancement introduction.

(e.g., the area between two finger joints) while contacting the touchscreen. Thus, the area related to the soft tissue would generate a smaller capacitive value. In our scheme, we first check the raw synthetic capacitive touch-sensing image column-by-column and derive the slope information based on the difference between the two nonzero values. When the slope is larger than zero, it is recognized that this point is

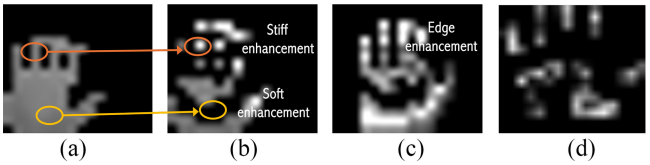


Fig. 7. Different synthetic images in the augmentation process. (a) Is the raw synthetic capacitive touch-sensing image after ray-based collision detection. (b) Is the image after soft and stiff enhancement? It is noted that some skeletal areas have been enhanced with higher value, and the partial palm region is processed with lower value. (c) Is the edge enhancement effect. (d) Is the real capacitive touch-sensing image.

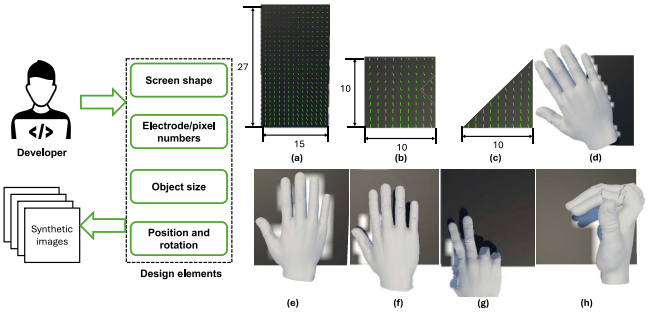


Fig. 8. Interactive design process between the developer and the synthetic capacitive touch-sensing image. *vCapTouch* allows several design factors to be accessed during the data synthetic process. (a)–(c) show different electrode/pixel numbers and shapes of the simulated touchscreen. (d) shows a relationship between a bigger hand model and the touchscreen. (e) and (f) present the hand model with different position and rotation statuses. (g) and (h) are different hand gestures.

in a concave area, and therefore, the raw synthetic capacitive value needs to be decreased. To form a smoother enhancement, we employed a nonlinear transformation that divided the raw synthetic capacitive value by the slope to the fourth to reduce the value.

Stiff Enhancement: The skeletal point is able to form a protruding characteristic and leads to a higher capacitive value when touching the screen. Following the same process, this point should protrudeprotrude when the slope value from the raw synthetic capacitive touch-sensing image is less than zero. A nonlinear transformation is then applied to further decrease the point's value. The raw synthetic value is updated by multiplying the square root of the slope.

Edge Enhancement: The *edge enhancement* aims to simulate the capacitor coupling situation of touching an object's edge. We detect the edge of the obtained outline and apply a linear transformation to expand the simulated capacitive value from the edge to the outer. We compared synthetic capacitive touch-sensing images before and after the *edge enhancement* [Fig. 7(c)].

F. Interactive Design for Automation Synthetic Dataset Building

Unlike the mechanism and data-driven simulation, our method is developed from the sensing result and inversely designs a way to simulate the result. Traditional data synthesis techniques have fewer controllable and accessible factors, developers cannot independently design the synthesis

Algorithm 1 Proposed Algorithm of Automation Synthetic Dataset Building

Input: Original hand model position: $P_o(x,y,z)$ and Euler angle of hand model attitude $E_o(x,y,z)$; Target hand model position and attitude: $P_t(x,y,z)$ and $E_t(x,y,z)$; Sampling rate: f; Size of dataset: S;

Output: Synthetic touch-sensing image $\{C_i\} = \{c_1, c_2, ..., c_n, \}$;

$time \leftarrow 0$
$size \leftarrow 0$
while $size < S$ **do**
  if $time = time + 1/f$ **then**
    $\lambda \in \mathcal{N}(0, 1)$
    $\beta \in \mathcal{N}(0, 1)$
    $P = P_o + \lambda * (P_t - P_o)$
    $E = E_o + \beta * (E_t - E_o)$
    $c = vCapTouch$
  end if
  $time++$
end while
$\{C_i\} \leftarrow c$

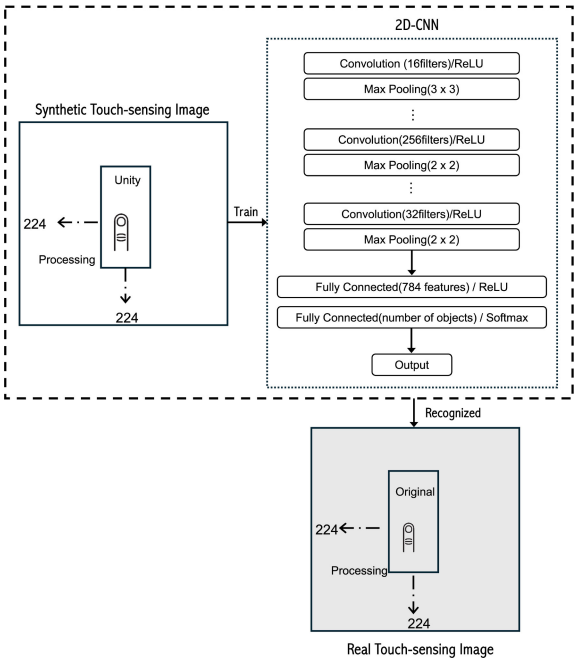


Fig. 9. Employed machine learning models introduction.

process, and the trial-and-error cost is high. Conversely, proposed *vCapTouch* follows a digital twin-based development paradigm, constructing the data synthesis process similar to the real process. Developers would be provided with more accessible factors during the synthesis. Fig. 8 presents the potential factors to develop different synthetic capacitive touch-sensing images, including the touchscreen shape, electrode/pixel numbers, and the spatial relationship between the hand model and the touchscreen. It is only necessary to perform the relevant operations in the virtual environment, making the data synthesis process easier and more intuitive.

Therefore, we developed an automation algorithm to generate the synthetic touch-sensing images to assist the synthetic dataset building (Algorithm 1). In a real situation, the position and angle of each user's touchscreen contact are various. Since the main process of capacitive touch-sensing image synthesis is completed by constructing the 3-D hand model and touchscreen, the key idea of the dataset automation scheme is to produce various rotation and position relationships between the 3-D hand model and touchscreen in the virtual environment. Therefore, to enrich the distribution of the synthetic touch-sensing dataset, we configure the original position and rotation, as well as the target position and rotation value of the 3-D hand model in the dataset generation algorithm. Altering the 3-D hand model's position and rotation characteristics could produce different touch positions and angles through random factors with the set sampling frequency and form a rich synthetic touch-sensing dataset distribution.

For each hand gesture, the hand model is designed with the recognized gesture class at first correspondingly. Then, the initial hand model position P_o and angle E_o, and the target position P_t and angle E_t are indicated and input into the algorithm. This sets the basic range of possible touching areas. In this area, the hand model's position and angle would be updated by different time slots by using random factors of λ and β. Therefore, it can obtain the distributions of synthetic touch-sensing images regarding the various touched positions and angles under any given hand gesture.

G. Recognition Model

To recognize the user's touching behavior, the machine learning model is adopted to learn the deep features of capacitive touch-sensing images. It aims to classify different touching hand gestures on the touchscreen. One of the significant parts of ensuring a high-performance machine learning model is to gain the distribution of recognized data that is as similar to the distribution of training data as possible. In our method, the prior knowledge of training data is created from the virtual environment, and the recognized data is based on the real capacitive touch sensing data. The incoherent difference between the two source domains could possibly lead to a potential difference in data distribution. Therefore, the feature mining capability to construct a deep representation space to guarantee a similar data distribution between the synthetic and real capacitive touch-sensing images is significant.

In order to better capture the local and global features from the synthetic capacitive touch-sensing image, we deploy a deep CNN model to be trained purely by the synthetic capacitive touch-sensing image. Fig. 9 presents the employed deep CNN model. A total of nine convolutional layers are deployed in the recognition model. Several different max-pooling operations are used in different convolutional layers to effectively reduce the spatial dimensions of the feature map while retaining the important feature information. The final output through the fully connected layers is the length of the recognized object category number.

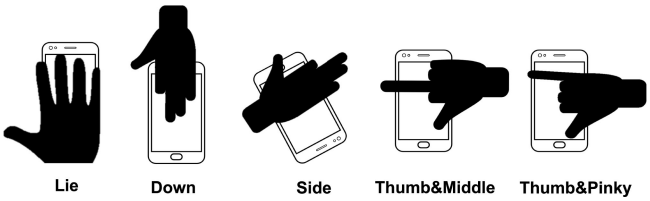


Fig. 10. Evaluated touching hand gestures in experiment. It shows five types of static finger gestures, including the hand lying on the screen, vertically put on the screen, hand side touching the screen, thumb and middle fingers, and thumb and pinky on the screen.

IV. Evaluation

The experiments that were conducted assessed the quality of the generated synthetic capacitive touch-sensing image compared with the real capacitive touch-sensing image and evaluated the recognition ability of synthetic data compared to real data. To validate the generalizability of the proposed method, we also tested the synthetic touch-sensing image method applied to resistive touchscreen devices, i.e., the piezoresistive sensor array.

A. Configuration

Fig. 10 introduced the tested hand gestures. We designed five types of static finger gestures, including the *lie, down, side, thumb and middle, thumb and pinky*. A total of eight participants were recruited to contribute to the real capacitive touch-sensing image dataset. Following the instruction, the user puts the fingers on the screen with designated gestures (Fig. 11). The process was repeated 10 times, and each time, the users followed their natural pattern of placing the finger to capture 10 real touch-sensing images by the touchscreen. We then collected 8 participants * 10 times * 10 images * 5 gestures = 4000 real touch-sensing images dataset.

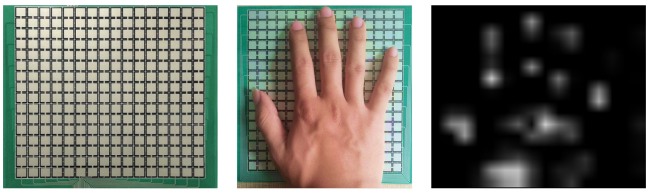
1) Capacitive Touchscreen: A capacitive touchscreen-based smartphone (LG Nexus 5) was utilized to capture the real capacitive touch-sensing image. We followed the common approach to access the controller with a customized kernel and obtained the capacitive touch-sensing image of the touchscreen [43]. The touchscreen size is 4.95 inches with 27*15 capacitive detection electrodes, which specify the size of the generated real capacitive touch-sensing image.

2) Resistive Touch Device: Additionally, we also employed a resistive touch-sensing device to validate the effectiveness of the proposed method. It utilized piezoresistor pressure sensing to perceive the touched object. It consists of several pressure sensors, each with a different force applied. So, it can also derive a grayscale image according to the normalization. We therefore tested on the resistive touch device to assess the cross-device generalizability of the proposed method. Our experiment involved a thin-film resistive pressure sensor of model M1616 (Fig. 12).

B. Synthetic Dataset

We configured various 3-D hand model positions and rotations in the virtual environment to construct the synthetic capacitive touch-sensing image dataset for machine learning model training to generate 400 synthetic images for each hand posture. Thus, the total synthetic capacitive touch-sensing image dataset would be 400 * 5 = 2000 images. Since the *edge enhancement* in Section IV-D was designed for simulating the capacitor coupling situation, the synthesis for resistive touch sensing data did not involve the *edge enhancement* part.

Fig. 11. Real dataset collection illustration with capacitive touchcreen.



(a) (b) (c)

Fig. 12. Piezoresistive sensors-based touch sensing device and related grayscale image.

C. Quality of Synthetic Data

We first evaluate the similarity between the synthetic data and real data to test the quality of the synthetic image from *vCapTouch*. The mean squared error (MSE) and structural similarity index measure (SSIM) were calculated between the synthetic images and the real images. To compare the proposed method with baseline methods, we also employed other images synthesis methods including the GAN [5], conditional GAN (CGAN) [40], conditional variational autoencoder (CVAE) [41], and conditional diffusion model (CDM) [42]. Since the baseline methods are data-driven, we used 80% of the real dataset for model training and generated the corresponding synthetic images (400*5) for testing with the remaining real images. Similarly, the synthetic images from the *vCapTouch* maintain the same size as the synthetic images dataset to calculate the MSE pixel by pixel. We also compared the result between the original synthetic capacitive touch-sensing images (i.e., w/o the augmentation scheme) and the ultimate synthetic capacitive touch-sensing images (i.e., with the augmentation scheme). Each synthetic image from a different method was compared with each real image to obtain the MSE and SSIM metrics. We tested five classes of gestures and calculated the average results. The result is shown in Tables I and II.

From the result, it is clear that the synthetic images from the proposed methods have the lowest MSE value and the best similarity with real images. All the baseline methods are capable of learning a data distribution from the trained real data. Moreover, SSIM considers more structural information between two images. The result from SSIM also shows that the synthetic images from the proposed methods keep the highest similarity to the real images. However, due to the characteristics of capacitive touch-sensing images, the grayscale image

TABLE I
MSE RESULTS OF COMPARING THE SYNTHETIC IMAGES WITH REAL IMAGES. SEVERAL BASELINE METHODS ARE INVOLVED IN THE EVALUATION

Method	MSE	
	Capacitive Screen	Resistive Device
GAN [5]	60.38 ± 29.82	54.23 ± 28.87
CGAN [40]	54.62 ± 27.03	42.69 ± 23.08
CVAE [41]	55.25 ± 29.64	44.75 ± 25.36
CDM [42]	52.07 ± 28.76	38.62 ± 22.57
w/o Augmentation	46.58 ± 27.13	38.18 ± 22.45
vCapTouch	**46.09 ± 25.97**	**34.74 ± 19.63**

TABLE II
SSIM RESULTS OF COMPARING THE SYNTHETIC IMAGES WITH REAL IMAGES. SEVERAL BASELINE METHODS ARE INVOLVED IN THE EVALUATION

Method	SSIM	
	Capacitor Screen	Resistance Screen
GAN	0.41 ± 0.12	0.38 ± 0.17
CGAN	0.46 ± 0.12	0.51 ± 0.17
CVAE	**0.48 ± 0.14**	0.52 ± 0.20
CMD	0.45 ± 0.16	0.56 ± 0.16
w/o Augmentation	0.47 ± 0.21	0.62 ± 0.21
vCapTouch	**0.48 ± 0.22**	**0.64 ± 0.19**

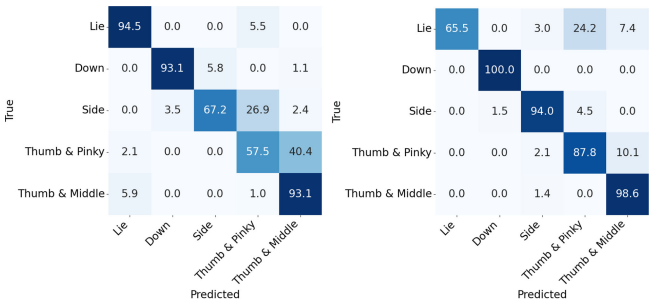


Fig. 13. Results of recognizing the real touch data using the model purely trained by synthetic data. (a) Capacitive. (b) Resistive.

usually has a relatively simple distribution and fails to mine the deep features of touch-sensing images. Compared with other approaches, the data-driven methods are not able to offer a good performance. Conversely, the *vCapTouch* utilized the digital-twin simulation to produce more fine-grained synthetic images. Moreover, we compared the method with and w/o the designed augmentation scheme in Section III-E. It is also notable that the augmentation scheme could further increase the fidelity of synthetic images.

D. Hand Gesture Recognition Performance

One of the biggest advantages of the proposed method is its zero-cost data synthesis process for touch sensing. Compared with other generative data synthesis methods (e.g., GAN), the data-driven generative method still requires a certain real dataset as the premise to gain the real data's prior knowledge and distribution. Thus, dataset collection volunteering and device usage require a high human resource cost. So, we only evaluated the recognition performance of *vCapTouch*, which utilizes the synthetic data for the machine learning model

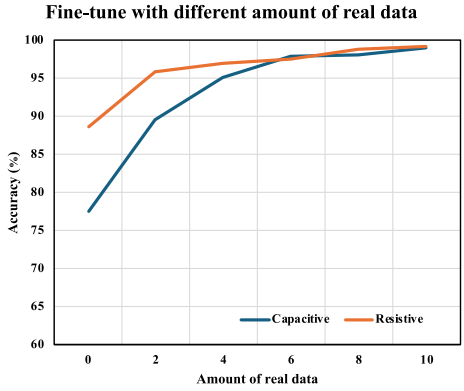


Fig. 14. Results of model performance with different amount of real data for fine-tuning.

(introduced as in Section III-G) training and real data for recognition to assess the effectiveness of the proposed method.

All of the synthetic images were employed for model training. All the real images were used as the testing dataset. The trained model was tested on each user's real data, and the average result was calculated. We repeated the experiment three times to get the average accuracy. The evaluation employed the PyTorch deep learning framework for building the machine learning model. The laptop with Intel Core i5-12500H CPU and Nvidia GeForce RTX 2050 GPU was utilized. The Adam optimizer was employed during the training process, and the batch size was 5. The training epoch was 200.

Fig. 13 presents the confusion matrix of the recognition model. Synthetic data purely train the model and recognize the real data could reach the accuracy of 77.56% and 88.57% for capacitive and resistive touch-sensing, respectively. Basically, the recognition performance of resistive touch-sensing could outperform the capacitive touch-sensing. Both recognition capabilities did not show an excellent performance compared with using real data for model training [4], [5], [6]. This is because there is an inherent difference between the source data (i.e., synthetic images) and target data (i.e., real images), which results in different data distribution and enables the trained model to present limited generalizability on domain transfer-based recognition.

E. Domain Transfer and Generalizability

1) Cross-Domain From Synthetic Data Training to Real Data Recognition: To better facilitate the usage of synthetic images training the recognition model, we evaluated the pretrain and a fine-tuning process to improve the performance of the trained model purely by synthetic data. Thus, the model was pretrained by synthetic images from *vCapTouch*. Then, the weights of convolutional layers were frozen, and a small number of real images fine-tuned the fully connected layers to realize the domain transfer in a supervised way [21]. During the experiment, a certain number of real images were used for fine-tuning, and the remaining real images were used for the test. Fig. 14 shows the performance results.

We tested 2, 4, 6, 8, and 10 real images extracted from the real dataset of each type of gesture for each user separately.

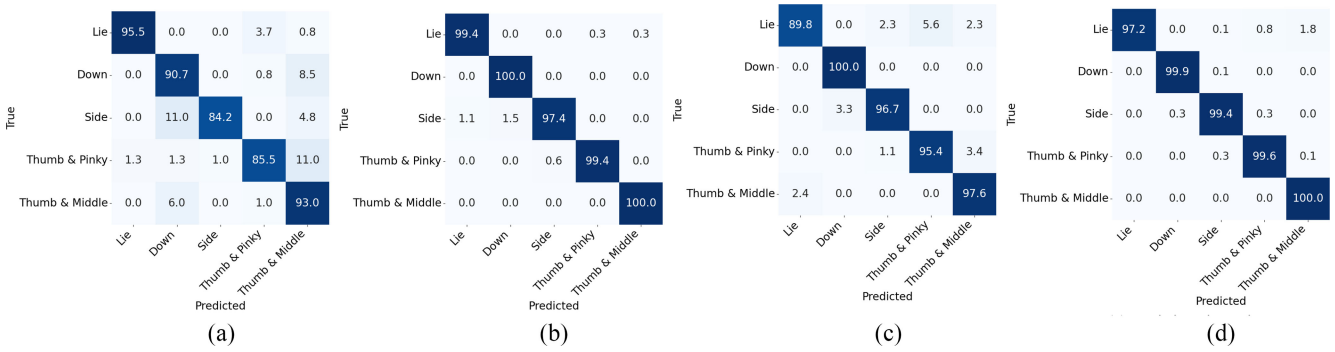


Fig. 15. Confusion matrix of model performance with different amount of real data for fine-tuning. (a) Capacitive with 2 images each user for fine-tuning. (b) Capacitive with 10 images each user for fine-tuning. (c) Resistive with 2 images each user for fine-tuning. (d) Resistive with 10 images each user for fine-tuning.

Therefore, there are 5*2*8=80, 5*4*8=160, 5*6*8=240, 5*8*8=480, and 5*10*8=400 real images used for the practical fine-tuning process. Fig. 15 gives the various confusion matrices with different amounts of real images used. From the result, it is evident that with the amount of real data utilized, the model is able to increase its recognition capability and complete the domain transfer of different data source domains. Generally, both the capacitive and resistive-based touch sensing would improve more, while only two images from each user's gesture were added. For example, the recognition model accuracy of capacitive touch-sensing is increased from 77.52% to 89.50%. With four more real images used, the model can reach over 95%. The same situation exists in resistive touch sensing. With two images from each user's gesture applied for fine-tuning, the accuracy is raised from 88.57% to 95.79%. Thus, the results demonstrate the model's fast domain transfer ability, which is purely trained by synthetic data with only a few real data sampled for fine-tuning. From the result, though using pure synthetic data to train the model did not produce an excellent result, only a few real data are needed to improve the model performance to an excellent result via domain transfer. Thus, the proposed data synthesis method reduces the dataset collection cost and time and improves efficiency compared to sampling and processing the traditional large real dataset.

2) Cross-Domain and Cross-Gesture From Synthetic Data Training to Real Data Recognition: In addition, since the synthetic data from *vCapTouch* is able to reduce the cost of training dataset collection and maintain the high flexibility to design various hand gestures, it is also significant to test the generalizability not only cross the domain but also the target gestures. We pretrained the model with two hand gestures classification (i.e., the lie and down) to ensure a feature-extraction capability of the model and recognize another three target hand gestures (i.e., side, thumb&middle, and thumb&pinky) as the downstream task. Similarly, a small amount of real data of three target hand gestures was employed to fine-tune the pretrained model and tested on the remaining real data. The whole evaluation process is the same as the one introduced above.

Figs. 16 and 17 presented the results of the fine-tuning test. From the result, the model pretrained by synthetic data is capable of maintaining a good feature extraction and is

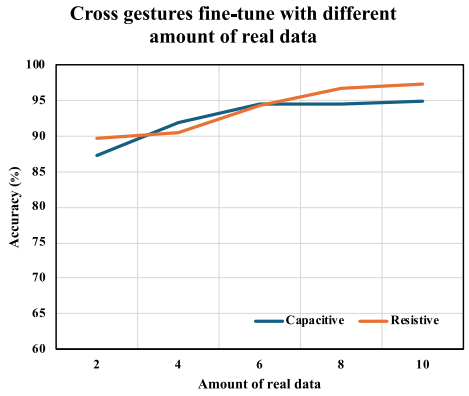


Fig. 16. Cross-gesture results of model performance with different amount of real data for fine-tuning.

prone to be transferred into target data distribution, even if the corresponding synthetic data is not involved during the training process. With only 2 to 4 images from each user's gesture acquisition as the training dataset, the pretrained model could reach over 90% accuracy on target gesture recognition with capacitive and resistive touch-sensing devices simultaneously.

Therefore, the domain-transfer test on eight participants proved our approach keeps a practical application potential that allows the developer to use the digital twin-based touch sensing data synthesis for model pretraining and then capture a tiny amount of use's real touch data for fine-tuning. Since only a small amount of real data is required, it omits the traditional large dataset collection process. It could be performed at the user-end electronics very quickly (e.g., a calibration process in a smartphone).

V. Discussion

A. Spatial Relationship Between 3-D Objects and Touchscreen

As the cost of touch-sensing devices becomes lower, we are able to obtain more customized capacitive touchscreens [9] of different sizes, different electrode sparsity, and number of electrodes in the real world. Our method primarily considered intuitively building the position relationship as the 3-D object can be manipulated with different gestures, perspectives, and

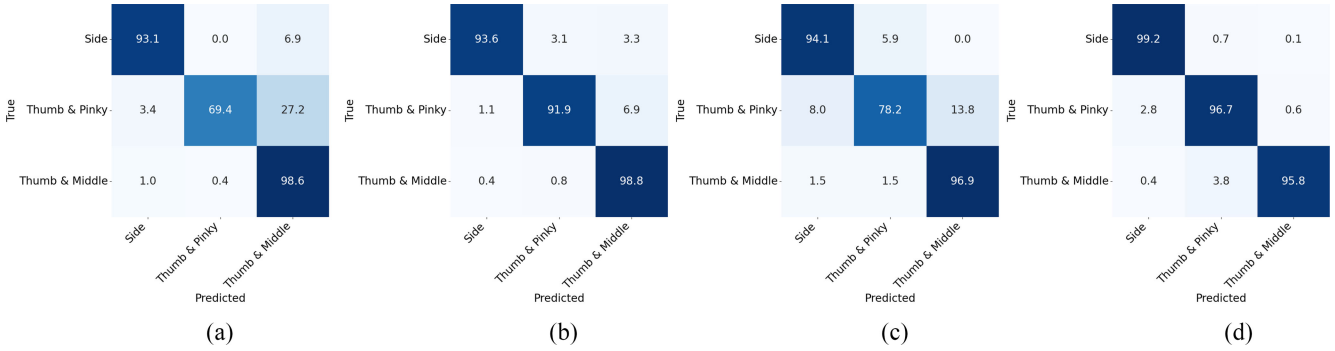


Fig. 17. Confusion matrix of model performance with different amount of real data for cross-gesture fine-tuning. (a) Capacitive with 2 images each user for fine-tuning. (b) Capacitive with 10 images each user for fine-tuning. (c) Resistive with 2 images each user for fine-tuning. (d) Resistive with 10 images each user for fine-tuning.

locations. Since *vCapTouch* mainly adopts a digital twin-like simulation, the spatial relationship between the 3-D object and touchscreen requires careful design, including the 3-D object position related to the touchscreen and the size of the 3-D objects and touchscreen. For example, a bigger touchscreen with a relatively small hand or a bigger hand with a relatively tiny touchscreen. This requires the synthesis of the capacitive touch-sensing image, which considers not only the touching relationship but also the practical device status. Our method relies on the interactive way, which is prone to modify the size of the touchscreen and 3-D object, and also easy to alter the electrode numbers and sparsity, quickly forming a simulation close to the real situation.

B. Reconstruction of 3-D Objects

There are diverse approaches to obtaining and importing 3-D objects, such as CAD design, scanning-based methods, 3-D point cloud conversion, and so on. However, this method mainly focused on the texture and mesh building to create a realistic Reconstruction. The core idea used in *vCapTouch* is based on the collision detection via *Raycast* functionality in Unity. Since the *Raycast* mainly simulated the detection principle of the infrared sensor, the transmitted ray would be impeded by the obstacle. A physical engine would simulate this phenomenon by constructing the collider of the 3-D object as the physical entity. Therefore, one of the most significant premises for using the functionality is to ensure a physical entity of detected 3-D objects.

In addition to the physical characteristics, the surface fineness of the 3-D object is also necessary for capacitive touch-sensing image synthesis. A simulated electrode with a ray transformed the high-dimension geometric information into low-dimension distance information. Thus, it proposes a higher requirement for the reconstruction of 3-D objects. A high-precision 3-D scanner or careful manual design is required.

C. Extending the Data Synthesis to More Types of Object Recognition

In this article, we primarily assessed the touchscreen's human hand gesture recognition. Nevertheless, the research of touch sensing has gradually extended to more daily object recognition. For example, Capacitivo [11] extended capacitive touch sensing to nonmetallic object recognition. Fabricated customized electrodes can recognize more common daily objects such as glass, fruit, and food. Other contact-based object (e.g., piezoresistance sensor arrays) methods also make object recognition more ubiquitous and pervasive. Our method is not limited to detecting conductors and metallic object recognition, and we believe that more daily object recognition system development could benefit from our data synthesis method. For different sensing principles, as long as the sparse electrodes are utilized, we could still rely on collision-based detection to obtain the basic low-dimensional outline information and design the specific augmentation scheme to complete the relevant data synthesis.

As introduced, ray-based detection lays the core part of touch-sensing image synthesis. The main employed information is its returned distance value. We may argue that this type of detection could have more prominent space for data synthesis in object or motion recognition. Xia et al. [21] realized a distance-based hand motion recognition data synthesis, utilizing various ray-based collision detection to form a distance variation matrix. Such an idea is also feasible for detecting human body motion with a distance sensor placed on the body joint to detect the distance variation between body limbs.

D. Potential Application Insights

vCapTouch introduced a novel data synthesis method to generate the capacities image, focusing on interactivity and intuition characteristics. We also envision various directions for applying this technique to the various communities. We also open-sourced the developed *vCapTouch* as a Unity package in Github for further interested utilization.[1]

1) Education With VR/AR: VR/AR systems are known for their immersive, interactive, and realistic characteristics, recognized as mainstream technical means in skill learning and education [44]. However, most VR/AR systems concentrate on communication and synchronization in teaching and learning. We should have figured out more work focusing on the sensor rationale with VR/AR systems for education. On the

[1]GitHub Link: xxxxxxx will be uploaded later.

contrary, *vCapTouch* is a great tool for novices to study touch sensing-related knowledge. Since the main work is completed in the virtual environment, it is open to be combined with a VR/AR system to construct an immersive and interactive study environment. By manipulating the 3-D object in the virtual space, the novice is able to understand what kind of information will be acquired by touch-sensing. Although the *vCapTouch* did not follow a capacitive or resistive sensing mechanism, namely, simulate the capacitance variation with human skin, it still provides a good opportunity to understand the basic sensing format and outputs.

2) Plugin and Toolkit for Interactive Machine Learning: Interactive machine learning allows a human-in-the-loop process in machine learning development and emphasizes the experiences of users [38]. The *vCapTouch* could also be utilized as a plugin and toolkit in sensor-based machine learning system development. Since the method represents its interactive data generation process, it would introduce more user experiences during the dataset collection period. Based on the *vCapTouch*, we can create the plugin with a concise interface to allow the users to access the data synthetic process better and utilize the synthetic dataset to develop the machine learning system with lower cost and higher efficiency. Besides, the *vCapTouch* could also be combined with other sensor simulation approaches, for example, the IMU [20], radar [18], and light [23], build a more comprehensive sensor simulated toolkit facilitating the sensor-based machine learning system development.

E. Novel Interactive Data Synthesis Direction

As mentioned before, the data scarcity issue has already been recognized as a severe bottleneck that prevents the generalizability of the machine learning model. The typical method utilized the idea of machine learning that employed the known dataset to learn the prior data distribution and generate the synthetic data from the learned knowledge space. This approach places a greater emphasis on completeness of knowledge and thus requires the dataset of prior knowledge to be diverse and large-scale, which still faces enormous challenges currently. Our method presents an alternative way of generating the data through the physical engine. It conducted little concern about the sensing principle but focused on the sensing result and simulated the data accordingly. Relying on the virtual environment, virtual avatar, and virtual objects, the data synthesis process would be intuitive and more suitable for developers with little skill. We are convinced that this method supports broader applications in machine learning, meta-verse, prototyping, and other fields.

F. Lack of More Detailed and Depth Evaluation

This article has primarily focused on the use of capacitive touch-sensing images as the training dataset for machine-learning-based object recognition. The evaluation has concentrated on accuracy performance and generalizability validation, but a more in-depth evaluation at the signal level, such as exploring the boundary limitation on the smallest pixel size of synthetic capacitive touch-sensing images, is needed. Some works have utilized QR-code format fiducial tags for tangible recognition, which clearly require a more refined perception capability of the touchscreen. Therefore, it is crucial to explore the boundary conditions related to various touchscreen devices, as this could significantly enhance the perception capability of the touchscreen and thereby improve the object recognition process.

VI. Conclusion

This article presented an alternative touch sensing data synthesis method, *vCapTouch*. Following a digital twin-based approach, the virtual touchscreen and 3-D objects are constructed in the virtual environment, Unity3D. We can convert the surface geometry information into a distance matrix by employing ray-based collision detection. A designed augmentation scheme can transform and enhance The synthetic touch-sensing image. The generated images could train a machine learning model and recognize the real touching behavior without large real dataset collection. *vCapTouch* presents a novel, low-cost, highly accessible, interactive touch sensing data synthesis method.

References

[1] T. Grosse-Puppendahl et al., "Finding common ground: A survey of capacitive sensing in human-computer interaction," in *Proc. CHI Conf. Human Factors Comput. Syst.*, New York, NY, USA, 2017, pp. 3293–3315.

[2] J. Qin et al., "Flexible and stretchable capacitive sensors with different microstructures," *Adv. Mater.*, vol. 33, no. 34, 2021, Art. no. 2008267.

[3] B. Parilusyan, M. Teyssier, V. Martinez-Missir, C. Duhart, and M. Serrano, "SenSurfaces: A novel approach for embedded touch sensing on everyday surfaces," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 1–19, 2022.

[4] N. Pourjafarian, A. Withana, J. A. Paradiso, and J. Steimle, "Multi-touch kit: A do-it-yourself technique for capacitive multi-touch sensing using a commodity microcontroller," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, New York, NY, USA, 2019, pp. 1071–1083.

[5] B. Steuerlein and S. Mayer, "Conductive fiducial tangibles for everyone: A data simulation-based toolkit using deep learning," *Proc. ACM Human-Comput. Interact.*, vol. 6, pp. 1–22, Sep. 2022.

[6] S. Mayer, X. Xu, and C. Harrison, "Super-resolution capacitive touchscreens," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2021, pp. 1–10.

[7] K. Ikematsu, M. Fukumoto, and I. Siio, "Ohmic-sticker: Force-to-motion type input device that extends capacitive touch surface," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, 2019, pp. 1021–1030.

[8] K. Ikematsu and S. Yamanaka, "ScraTouch: Extending interaction technique using fingernail on unmodified capacitive touch surfaces," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 3, pp. 1–19, 2020.

[9] A. S. Nittala, A. Withana, N. Pourjafarian, and J. Steimle, "Multi-touch skin: A thin and flexible multi-touch sensor for on-skin input," in *Proc. CHI Conf. Human Factors Comput. Syst.*, New York, NY, USA, 2018, pp. 1–12.

[10] M. Teyssier, G. Bailly, C. Pelachaud, E. Lecolinet, A. Conn, and A. Roudaut, "Skin-on interfaces: A bio-driven approach for artificial skin design to cover interactive devices," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, New York, NY, USA, 2019, pp. 307–322.

[11] T.-Y. Wu, L. Tan, Y. Zhang, T. Seyed, and X.-D. Yang, "Capacitivo: Contact-based object recognition on interactive fabrics using capacitive sensing," in *Proc. 33rd Annu. ACM Symp. User Interface Softw. Techno.*, New York, NY, USA, 2020, pp. 649–661.

[12] Q. Wen et al., "Time series data augmentation for deep learning: A survey," 2020 *arXiv:2002.12478*.

[13] J. J. Bird, C. M. Barnes, L. J. Manso, A. Ekárt, and D. R. Faria, "Fruit quality and defect image classification with conditional GAN data augmentation," *Scientia Horticulturae*, vol. 293, Feb. 2022, Art. no. 110684.

[14] Z. Yang, Y. Li, and G. Zhou, "TS-GAN: Time-series GAN for sensor-based health data augmentation," *ACM Trans. Comput. Healthcare*, vol. 4, no. 2, pp. 1–21, 2023.

[15] J. Zhang et al., "Data augmentation and dense-LSTM for human activity recognition using WiFi signal," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4628–4641, Mar. 2021.
[16] M. M. Rahman, S. Z. Gurbuz, and M. G. Amin, "Physics-aware generative adversarial networks for radar-based human activity recognition," *IEEE Trans. Aerosp. Electronic Syst.*, vol. 59, no. 3, pp. 2994–3008, Jun. 2022.
[17] M. Zhang et al., "Motiondiffuse: Text-driven human motion generation with diffusion model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 6, pp. 4115–4128, Jun. 2024.
[18] K. Ahuja, Y. Jiang, M. Goel, and C. Harrison, "Vid2doppler: Synthesizing doppler radar data from videos for training privacy-preserving activity recognition," in *Proc. CHI Conf. Human Fact. Comput. Syst.*, New York, NY, USA, 2021, pp. 1–10.
[19] Z. Cui, L. Mei, S. Pei, B. Li, and X. Zhou, "Privacy-preserving human activity recognition via video-based range-doppler synthesis," in *Proc. 27th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, 2024, pp. 649–654.
[20] H. Kwon et al., "IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 3, pp. 1–29, 2020.
[21] C. Xia, A. Saito, and Y. Sugiura, "Using the virtual data-driven measurement to support the prototyping of hand gesture recognition interface with distance sensor," *Sensors Actuators A, Phys.*, vol. 338, May 2022, Art. no. 113463.
[22] C. Xia and Y. Sugiura, "Virtual IMU data augmentation by spring-joint model for motion exercises recognition without using real data," in *Proc. ACM Int. Symp. Wearable Comput.*, 2022, pp. 79–83.
[23] K. Matsuo, C. Xia, and Y. Sugiura, "Virsen1.0: Toward sensor configuration recommendation in an interactive optical sensor simulator for human gesture recognition," *Int. J. Digit. Human*, vol. 2, no. 3, pp. 223–241, 2023.
[24] J. M. E. Belo, A. M. Feit, T. Feuchtner, and K. Grønbæk, "XRgonomics: Facilitating the creation of ergonomic 3D interfaces," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2021, pp. 1–11.
[25] M. Tatzgern, V. Orso, D. Kalkofen, G. Jacucci, L. Gamberini, and D. Schmalstieg, "Adaptive information density for augmented reality displays," in *Proc. IEEE Virtual Real. (VR)*, 2016, pp. 83–92.
[26] C. Xia, X. Fang, R. Arakawa, and Y. Sugiura, "VoLearn: A cross-modal operable motion-learning system combined with virtual avatar and auditory feedback," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 1–26, 2022.
[27] L. Paredes et al., "Fabhandwear: An end-to-end pipeline from design to fabrication of customized functional hand wearables," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–22, 2021.
[28] T. C. Alexander, H. S. Ahmed, and G. C. Anagnostopoulos, "An open source framework for real-time, incremental, static and dynamic hand gesture learning and recognition," in *Proc. 13th Int. Conf. Human-Comput. Interact. Novel Methods Techn.*, San Diego, CA, USA, 2009, pp. 123–130.
[29] J. Karolus et al., "Embody: A data-centric toolkit for EMG-based interface prototyping and experimentation," *Proc. ACM Human-Comput. Interact.*, vol. 5, pp. 1–29, May 2021.
[30] J. Liao, K. Van, Z. Xia, and R. Suzuki, "RealityEffects: Augmenting 3D volumetric videos with object-centric annotation and dynamic visual effects," in *Proc. ACM Design. Interact. Syst. Conf.*, 2024, pp. 1248–1261.
[31] Y. Zhang, C. Nguyen, R. H. Kazi, and L.-F. Yu, "PoseVEC: Authoring adaptive pose-aware effects using visual programming and demonstrations," in *Proc. 36th Annu. ACM Symp. User Interface Softw. Technol.*, 2023, pp. 1–15.
[32] D. Kohlsdorf, T. Starner, and D. Ashbrook, "Magic 2.0: A Web tool for false positive prediction and prevention for gesture recognition systems," in *Proc. IEEE Int. Conf. Autom. Face & Gesture Recognit. (FG)*, 2011, pp. 1–6.
[33] Y. Kim et al., "SensorViz: Visualizing sensor data across different stages of prototyping interactive objects," in *Proc. ACM Design. Interact. Syst. Conf.*, 2022, pp. 987–1001.
[34] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Proc. 7th Int. Symp. Vis. Comput. (ISVC)*, Las Vegas, NV, USA, 2011, pp. 199–208.
[35] R. Williams, S. P. Kaputsos, and C. Breazeal, "Teacher perspectives on how to train your robot: A middle school AI and ethics curriculum," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 15678–15686.
[36] J. Haladjian, "The wearables development toolkit: An integrated development environment for activity recognition applications," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 1–26, 2019.
[37] K. Sabir, C. Stolte, B. Tabor, and S. I. O'Donoghue, "The molecular control toolkit: Controlling 3D molecular graphics via gesture and voice," in *Proc. IEEE Symp. Biol. Data Vis. (BioVis)*, 2013, pp. 49–56.
[38] Z. Zhou and K. Yatani, "Gesture-aware interactive machine teaching with in-situ object annotations," in *Proc. 35th Annu. ACM Symp. User Interface Softw. Technol.*, 2022, pp. 1–14.
[39] K. Monteiro, R. Vatsal, N. Chulpongsatorn, A. Parnami, and R. Suzuki, "Teachable reality: Prototyping tangible augmented reality with everyday objects by leveraging interactive machine teaching," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2023, pp. 1–15.
[40] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional GAN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2970–2979.
[41] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," 2017, *arXiv:1703.10960*.
[42] R. Huang et al., "FastDiff: A fast conditional diffusion model for high-quality speech synthesis," 2022, *arXiv:2204.09934*.
[43] A. Guo, R. Xiao, and C. Harrison, "CapAuth: Identifying and differentiating user handprints on commodity capacitive touchscreens," in *Proc. Int. Conf. Interact. Tabletops & Surfaces*, 2015, pp. 59–62.
[44] W. Alhalabi, "Virtual reality systems enhance students' achievements in engineering education," *Behav. Inf. Technol.*, vol. 35, no. 11, pp. 919–925, 2016.

Chengshuo Xia (Member, IEEE) received the Ph.D. degree from Keio University, Shinjuku, Japan, in 2022.

He is a Lecturer with Guangzhou Institute of Technology, Xidian University, Xi'an, China. He was a Visiting Researcher with the University of California at Los Angeles, Los Angeles, CA, USA, in 2022. He was a Postdoctoral Researcher with Keio University. His research interests includes ubiquitous computing, human–computer interaction, and energy harvesting.

Qingyuan Peng is currently pursuing the master's degree with Guangzhou Institute of Technology, Xidian University, Xi'an, China.

His research interests include the human activity recognition, human–computer interaction, and sensing.

Zeyuan Fan is currently pursuing the undergraduate degree with Jiangsu University of Science and Technology, Zhenjiang, China.

His research interests include the VR-based system development and interactive design.

Tian Min received the bachelor's degree from CUHK-Shenzhen, Shenzhen, China, in 2020, and the master's degree from Keio university, Shinjuku, Japan, in 2024, where he is currently pursuing the Ph.D. degree.

His research interests include the human–computer interation and ubiqitous computing.

Daxing Zhang received the PhD. degree from Xidian University, Xi'an, China, in 2008.

He is an Associate Professor with Guangzhou Institute of Technology, Xidian University. His research interests includes industrial intelligence.

Congsi Wang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electromechanical engineering from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively.

He served as a Visiting Scholar with the University of California at Berkeley, Berkeley, CA, USA, from December 2012 to December 2013, and a Visiting Fellow with the University of New South Wales, Sydney, NSW, Australia, from November 2017 to October 2018. He is currently a Professor and the Deputy Dean with the School of MechanoElectronic Engineering, Xidian University. His research interests include electromechanical coupling of electronic equipment with an emphasis on the modeling, influencing mechanism, and design and application of structuralelectromagnetic-thermal coupling.