International Journal of the Digital Human

ISSN online: 2046-3383 - ISSN print: 2046-3375
https://www.inderscience.com/ijdh

VirSen1.0: toward sensor configuration recommendation in an interactive optical sensor simulator for human gesture recognition

Kana Matsuo, Chengshuo Xia, Yuta Sugiura

Article History:

Received:	03 February 2023
Last revised:	12 May 2023
Accepted:	17 May 2023
Published online:	25 August 2023

VirSen1.0: toward sensor configuration recommendation in an interactive optical sensor simulator for human gesture recognition

Kana Matsuo*, Chengshuo Xia and Yuta Sugiura

Department of Information and Computer Science,
Faculty of Science and Technology,
Keio University 3-14-1,
Hiyoshi, Kohoku-ku, Yokohama,
223-8522, Japan
Email: tsuokana@keio.jp
Email: csxia@keio.jp
Email: sugiura@keio.jp
*Corresponding author

Abstract: Research is underway on the use of sensor simulation in generating sensor data to design a real-world human gesture recognition system. The overall development process suffers from poor interactive performance, because developers lack an efficient tool to support the sensor configuration, result checking, and trial-and-error that arise when designing a machine learning system. Hence, we have developed *VirSen*1.0, a virtual environment with a user interface to support the process of designing a sensor-based human gesture recognition system. In this environment, a simulator produces lightness data and combines it with an avatar's motion to train a classifier. Then, the interface visualises the importance of the features used for the model, via the permutation feature importance, and it provides feedback on the effect of each sensor to the classifier. This paper proposes a complete development process, from acquisition of learning data to creation of a learning model, using a single software tool. Additionally, a user study confirmed that by visualising the importance of the features used in the model, users can create learning models that achieve a certain level of accuracy.

Keywords: sensor simulator; interactive system; optical sensor; machine learning; graphical user interface.

Reference to this paper should be made as follows: Matsuo, K., Xia, C. and Sugiura, Y. (2023) 'VirSen1.0: toward sensor configuration recommendation in an interactive optical sensor simulator for human gesture recognition', *Int. J. Digital Human*, Vol. 2, No. 3, pp.223–241.

Biographical notes: Kana Matsuo received her Master's degree from Keio University. She is a researcher at the Department of Information and Computer Science, Keio University. Her research interests include human-computer interaction and virtual reality.

Chengshuo Xia is a Lecturer/Assistant Professor at Xidian University (Guangzhou), China. He is also a Guest Lecturer at Keio University, Japan. His research interests include human-computer interaction, wearable computing and energy harvesting.

Yuta Sugiura is an Associate Professor of the Department of Information and Computer Science at Keio University. Before joining the department, he worked as a post-doctoral researcher at National Institute of Advanced Industrial Science and Technology (2015–2016). He received his PhD from the Graduate School of Media Design at Keio University in 2013. His research interests are in user interfaces and ubiquitous computing. He has received several awards including the IPSJ Prize, UIST Best Talk Award, and the Good Design Award. He served as a program committee member for various international conferences including ACM UIST, TEI, and SIGGRAPH ASIA E-Tech.

1 Introduction

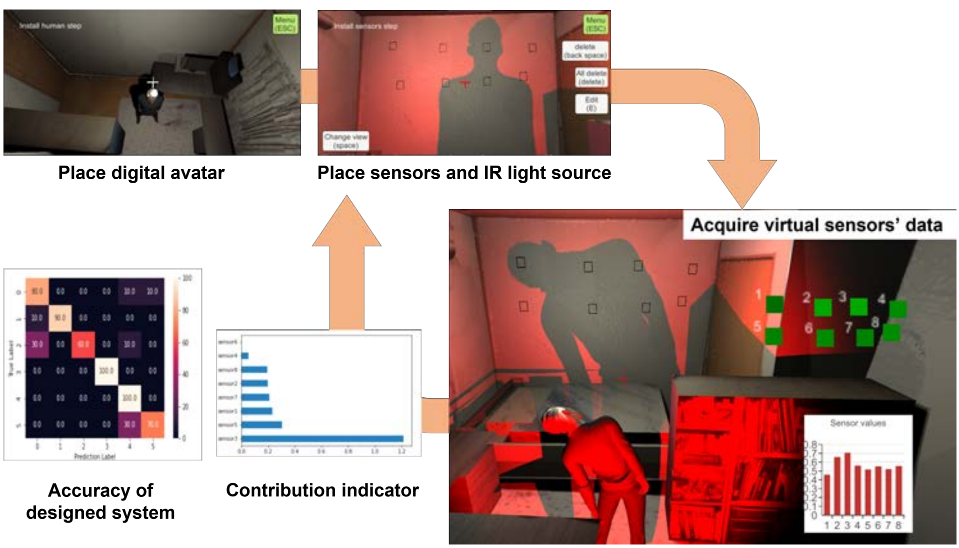
Methods for estimation of human gestures by utilising real-world sensors and machine learning have been studied extensively (Xia and Sugiura, 2021; Xia et al., 2022). Gesture recognition is now being applied in health surveillance, sports, entertainment, medicine, efficient human interfaces, parenting, caregiving, and many other fields (Fu et al., 2020). When sensors identify human motions and gestures, the system performance depends on the number of sensors, the sensor placements, and the motions to be recognised. In the real world, the biggest challenge is that it is time-consuming and expensive to create and experiment with multiple devices while considering all possible combinations of sensor numbers and placements. On the other hand, if sensors could be simulated, then different sensor placements could be tested without needing real objects, which would significantly improve the efficiency of developing sensor-based applications. Moreover, the search for sensor placements could be automated, which would eliminate the need to physically install sensors and thus save even more time.

To this end, researchers have simulated real-world sensors on a computer (Rossmann et al., 2012; Park et al., 2014; Xia and Sugiura, 2021). While that approach can potentially decrease the costs involved in dataset collection and augment existing datasets to improve the performance of a machine learning classifier, the existing research has focused on the level of data generation. Few studies have explored the notion of having users interactively and intuitively develop a gesture recognition system with techniques such as dynamically changing sensor placements or obtaining feedback about the classifier's performance. Interactive simulator operation through a user interface (UI) would enable the user to become familiar with the simulator more quickly. Such a UI would also have the potential to popularise sensor simulators among people without a background in information engineering, thereby creating a future in which internet of things (IoT) devices could be specified by ordinary users according to their daily lifestyle needs.

Optical sensors are real-world sensors that measure human motions (Manabe, 2013; Kawashima et al., 2017; Murakami et al., 2019; Fu et al., 2020). In contrast to standard vision sensors, optical sensors that sense infrared light can acquire motion data in the dark and capture differences in light intensity generated by reflections due to human motions, thus preserving the measurement target's privacy. In addition, optical sensors for infrared sensing are frequently implemented in home environments for systems that are used to track and evaluate daily activities (Fu et al., 2020).

Given the above background, we have developed a UI, called *VirSen*1.0, that enables users to interactively arrange optical sensors in a virtual space to design a gesture estimation system. Figure 1 shows an overview of this simulator. The UI can be used to measure a human avatar in a three-dimensional space, and the light sources and sensors necessary for measurement can be virtually installed and later moved or deleted. The training data for machine learning is acquired by having the avatar perform a gesture that the user wants to identify. After data acquisition, the results of recognition by a support vector machine (SVM) classifier are displayed, and a permutation feature importance (PFI) contribution indicator supports the search for sensor placements with a high recognition rate. In this paper, we report the results of a user study in which 16 participants used our prototype UI and evaluated its functionality.

Figure 1 Overview of the proposed simulator (see online version for colours)



Our main contributions in this paper are as follows.

1 we have developed an interactive sensor simulator that enables flexible placement of virtual optical sensors and includes a PFI contribution indicator that shows each sensor's contribution

2 we have demonstrated through a user study that the sensor PFI contribution indicator can contribute to the development of sensor systems with high recognition rates.

2 Related work

2.1 Simulators for sensor measurement

Buchmayr et al. (2011) developed a UI that reproduces behaviour data from sensors (e.g., those embedded in IoT devices) on a computer so that the data can be obtained quickly. The UI provides a floor plan visualised as a 2D map on which users can freely place sensors, and sensor behaviour data can be obtained by touching or clicking the sensors on the screen with a mouse. Rossmann et al. (2012) developed a simulation method and a sensor simulation framework for optical sensors such as cameras, time-of-flight cameras, and laser range scanners. The surface of a planet is depicted in a VR simulator, thus allowing optical sensors to be reproduced. Park et al. (2014) used Kinect and 3D avatars to reproduce real-world human motions on a computer in a more realistic manner than before, and they tested a generic human detection algorithm that could detect 3D avatars as humans. Then, they tested the algorithm's parameters in the real world and found that it could detect real-world humans, as well. To reduce the cost of generating new training data whenever sensor locations are moved, **?** developed a system to assist in sensor placement and training model generation by using virtual sensor data for HAR system development.

As mentioned above, various studies have focused on simulation of sensor measurements. However, our research stands out in its interactive approach of using a single software tool to simulate sensors, acquire training data, and create training models. This streamlined process enables multiple trial-and-error processes. Another unique aspect is the use of the PFI contribution indicator during those processes, which enables efficient exploration of sensor placements and ultimately leads to higher recognition rates.

2.2 Measurement of human body movements with optical sensors

There are several prior studies on ways to measure, estimate, and identify human body movements via optical sensors. Kawashima et al. (2017) used infrared sensor arrays attached to the ceiling to identify daily activities such as walking, sitting, and standing. A shallow convolutional neural network (CNN) comprising three layers was used to identify data at a frame rate of 10 fps with 85.75% accuracy. Manabe (2013) used a single photo reflector to identify multi-touch gestures and demonstrated that even a single photo reflector can discriminate between touch and gestures using multiple fingers. AffectiveHMD (Murakami et al., 2019) focuses on the fact that the unevenness of the facial skin surface varies with human facial expressions. It uses a photo reflector inside a head-mounted display (HMD) to measure the distance between the facial skin surface and the sensor.

While previous studies have used optical sensor data to estimate human body movements, no existing research specifically focuses on simulating the sensors. In our study, the target sensor for simulation is the phototransistor of a photo reflector, which is a type of optical sensor. Motion detection and recognition are based on the optical sensor's output value, which changes when a physical action is performed in front of the sensor.

2.3 Applications of permutation feature importance

In this study, we use the PFI (Fisher et al., 2018) to visualise the importance of the features used for learning and provide the user with feedback about the model. Various prior studies also calculated the importance of features and used the PFI for improvement in the recognition accuracy of learning models and for feature analysis.

Huang et al. (2016) applied the permutation importance (PI) to power-load forecasting in smart grids and calculated the PI of 243 features used in a short-term load forecast (STLF), after which features with high impact on the learning model were extracted. This technique improved the load forecast's accuracy and is simpler than the conventional method because it requires less computation time. Kaushik and Birok (2021) applied the PI to predict heart failure via the XGBoost algorithm, and they found they could improve the prediction accuracy by using only the features that were important for learning. Gazi et al. (2021) applied the PI to track a patient's status in virtual reality exposure therapy via machine learning. Several features were used to detect patient anxiety during the therapy, and the PI enabled identification of the features that were important for anxiety detection. Schelthoff et al. (2022) used the PFI to estimate wait times for operations at a semiconductor manufacturing plant via machine learning. The features in the plant that affected the waiting time were inferred by treating each step of the semiconductor fabrication process as a feature.

In the above studies, the PFI and PI were applied to create learning models that gave high recognition rates, or they were used to analyse important features. In this study, each instance of data collected from the simulator's sensors is treated as a feature. This approach differs from previous approaches in that it performs PFI analysis of the sensor data and uses the results to derive a sensor arrangement that yields a recognition rate close to 100% for the simulator user.

3 Methodology

3.1 Overview

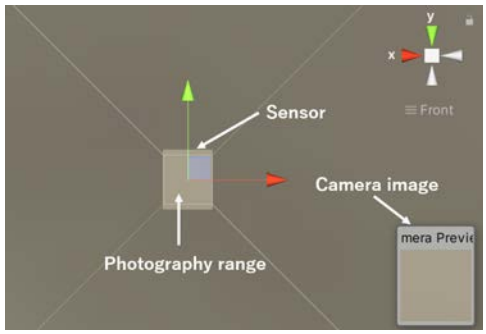
This section describes the method used in this study to simulate sensor data and create a gesture recognition model. The virtual sensor comprises a camera and a rectangular object that is the same size as a physical sensor and emulates its behaviour. An SVM is used to generate the gesture recognition model because it is impractical to collect a significant amount of training data. The PFI method is applied to assess the usefulness of features in the model by measuring their impact on the model's prediction error. This approach enables effective exploration of sensor placement during the trial-and-error process.

3.2 Simulated optical sensor

Sensors placed in a virtual space assume the light intensity to be the actual optical sensor's output value. Thus, as illustrated in Figure 2, the virtual sensor here comprises the virtual camera and rectangular object mentioned above. The camera observes the object's surface, and the camera image's intensity value is output every frame, thus giving the same output behaviour as an actual sensor. The intensity is given by equation (2), where I denotes the intensity value, and R, G, and B respectively denote the red, green, and blue RGB values. Each RGB value has a range of 0–1, and the intensity is output as a decimal value between 0–1.

$$I = 0.229 \times R + 0.587 \times G + 0.114 \times B, \tag{1}$$

Figure 2 Structure of the optical sensor on the simulator (see online version for colours)



3.3 Gesture recognition model

To develop the gesture recognition model, an SVM classifier is generated using sensor data from the simulator. The SVM algorithm uses a radial basis function kernel. In our method, training data acquisition requires a subject to perform a gesture multiple times. Furthermore, generation of a large amount of training data leads to longer simulation times. We thus use the SVM because it is impractical to prepare a significant amount of training data.

3.3.1 PFI theory

As stated above, the PFI is typically applied to measure the usefulness of features in machine learning models. The theory of the PFI is as follows. First, the importance of features is determined by calculating their impact on the model's prediction error after reordering them. If swapping certain features increases the model error, then the model's predictions depend on those features, which mean that they are important. In contrast, if the model error does not change when a feature is swapped, then that feature is considered unimportant. The importance of each feature is expressed by equation (2), with the fitted predictive model m and a tabular dataset D as inputs:

$$i_i = s - \frac{1}{K}\sum_{k=1}^{k} s_{kj}. \tag{2}$$

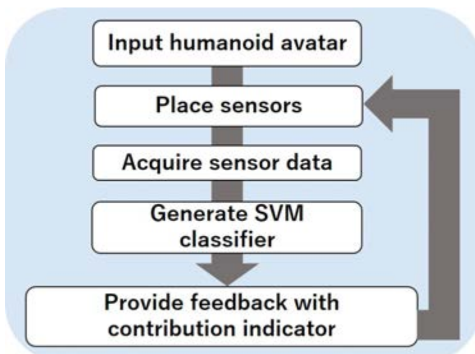
Here, s is a reference score, in terms of accuracy, when D is input to m. For each feature j, k is a number from 1–K, and D_{kj} is a new dataset in which the position of feature j in D is randomly shuffled. Then, s_{kj} is the score obtained by inputting $\tilde{D}_{kj}$ to model m. Lastly, i_i is the importance of feature j.

4 Implementation

4.1 Overview

The process flow of *VirSen*1.0, which is built on Unity, is shown in Figure 3. A start menu appears when the simulator is initiated. The user then selects the objects to use (human avatars, light sources, and sensors) and places them in the measurement environment by using a mouse. Training data is obtained from the simulator, and the recognition accuracy is calculated by an SVM. In addition, a PFI-based contribution indicator provides feedback to the user about which sensor locations have the greatest impact on the learning model. This helps the user determine a sensor placement with a recognition accuracy close to 100%.

Figure 3 Process flow of the proposed simulator (see online version for colours)



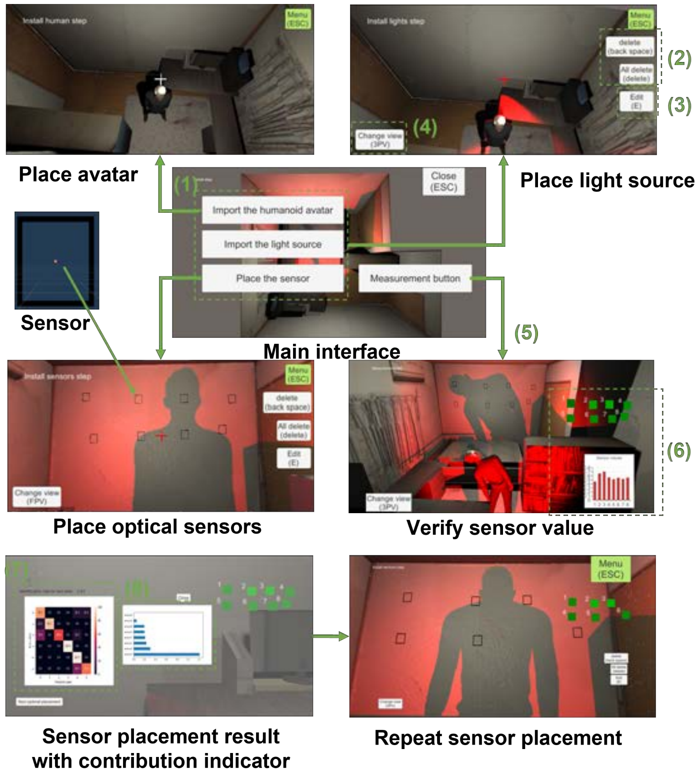
4.2 Simulated optical sensor

The optical sensor utilised in this study comprises an infrared LED and a transistor as a photodetector. The photocurrent generated in proportion to the light received by the photodetector is amplified and output. The start of data acquisition is defined as when the value of one of the installed sensors exceeds a specific threshold value as compared to the value in the previous frame. We adopt a new infrared LED with a light distribution angle of 104 degrees to enable measurement farther from the sensor, thereby expanding the measurement range to approximately 3 m. As shown in Figure 5, body movements occur between the sensor and the light source, and the value of each sensor changes when the light is blocked.

4.3 3D human gestures and environment

We use Xsens[1], a motion capture tool that uses inertial sensors, to reproduce body movements, which are then recorded as Filmbox format files. The six different gestures shown in Figure 6 were examined in this study: jump, squat, lean upper body to the right, lean left, raise right hand, and raise left hand.

Figure 4 *VirSen*1.0 (see online version for colours)



Note: 1 – place objects (human avatars, light sources, sensors); 2 – delete objects; 3 – move objects after placement; 4 – switch viewpoints during placement (third-person/first-person); 5 – obtain training data; 6 – visualise sensor values; 7 – calculate recognition accuracy; 8 – view result with PFI contribution indicator.

Figure 5 Measurement principle (see online version for colours)

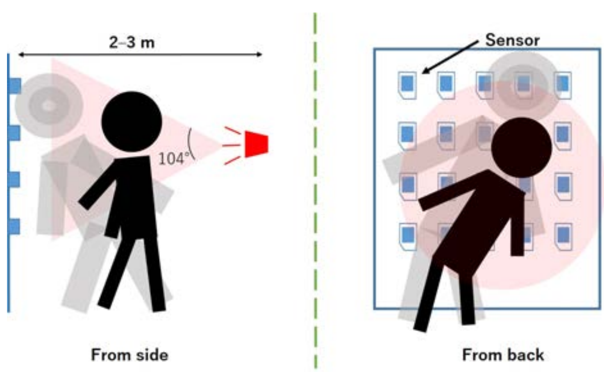


Figure 6 Recognised gesture types

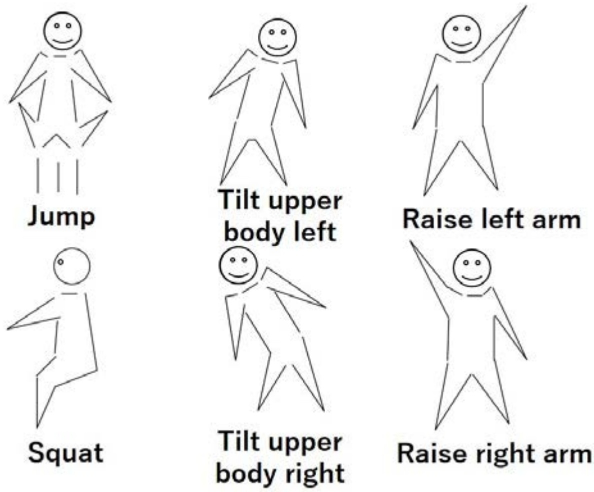


Figure 7 Measurement environment on the simulator (see online version for colours)

The real-world environment where the sensors are placed are reproduced on a computer by using Azure Kinect DK and simultaneous localisation and mapping technology (SLAM). The Kinect DK is a device comprising an RGB camera, infrared camera, and depth camera. The combination of this device and SLAM technology reproduces the shape of the space captured by the camera in a 3D point cloud (Matsuo et al., 2020). The 3D point cloud data is meshed using Blender to fill in any missing areas. The measured environment and movements are reconstructed on the simulator by importing the Filmbox file containing the recorded 3D model of the measurement environment and the movements. Figure 7 shows the measurement environment on the simulator.

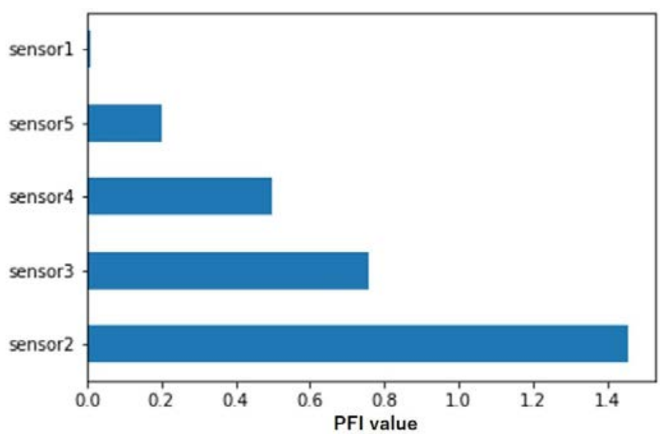
4.4 Gesture recognition model

The SVM is implemented by using the SVC class of scikit-learn, a Python machine learning library. To develop the gesture recognition model, a human body moves in front of the sensors placed on the simulator, and sensor data is acquired to calculate the recognition accuracy. The sensor values for a frame in which the sensor responds are recorded in a CSV file and labelled with the gesture type. When the simulator finishes generating sensor data, it uses UDP socket communication to signal the Python environment running in the background. Once the signal is sent, the sensor data is used to generate the SVM classifier.

4.5 PFI contribution indicator

To support placement search, we use the PFI, which measures the degree to which the features used in learning influenced a model when it was generated. The UI for the 'PFI contribution indicator' is shown in Figure 8. This information is feedback to the user as a bar graph in order of decreasing influence of each sensor on the recognition rate. We use the Python machine learning library scikit-learn to implement the PFI. Learning is performed on data sent from Unity to calculate the importance of each feature, and the importance results are sent back to Unity via socket communication. The simulator screen displays the PFI calculation results in a graph.

Figure 8 UI of the PFI contribution indicator (see online version for colours)



5 User study

5.1 Overview and purpose

As discussed in the previous section, *VirSen*1.0 is an interactive sensor simulation with a sensor placement recommendation function based on machine learning PFI calculations. Users can freely position sensors with a mouse and view the status of training data acquisition and the sensor values. Then, a user can examine the recognition accuracy and confusion matrix for the sensor placement, and the importance of each sensor to the training model.

The uniqueness of our approach is that we visualise the importance of sensor placement by using the PFI. While it has previously been used to build learning models with high recognition accuracy or to analyse important features, few studies have explored PFI visualisation with a UI built into a developer tool. Here, we hypothesise that the PFI contribution indicator improves the recognition rate when users search for sensor placements.

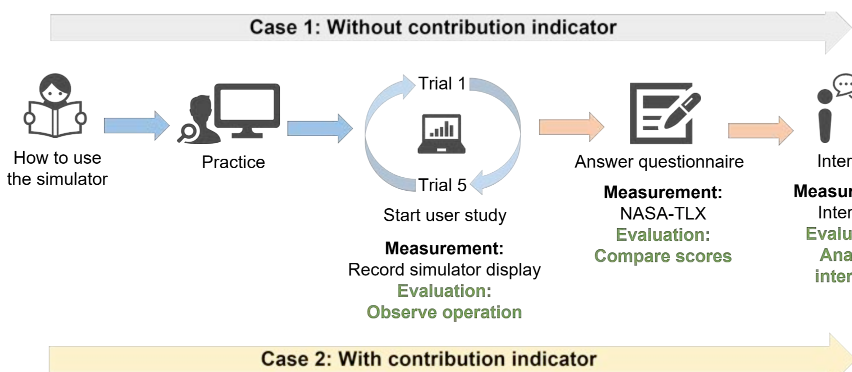
If our hypothesis is verified, we can consider the PFI contribution indicator to support a user's sensor placement search. We believe this will help lower the bar for the development of data-driven, real-world sensing systems. With this motivation, we evaluated our hypothesis by conducting a user study.

5.2 *Design and task*

5.2.1 *Design*

In this user study, the simulator was actually used by several users. The experimental design is shown in Figure 9. We divided the participants into two groups: one operating the simulator without the PFI contribution indicator (case 1), and one operating the simulator with it (case 2). The participants were asked to perform five trials of simulating the sensor placement.

Figure 9 Overview of the user study design (see online version for colours)



Note: Each participant was randomly assigned to one of two cases.

5.2.2 *Task*

The participants were given a task of using the sensor simulator to search for a five-sensor arrangement with the highest possible recognition accuracy for six gestures (Figure 6).

5.3 *Evaluations*

For indicators corresponding to our hypothesis, we performed three evaluations: comparison of the recognition rates in cases 1 and 2; measurement of the physical, mental, and other demands during simulator operation; and subjective comments from interviews with the participants after they had used the simulator. In addition, the simulator screen was recorded during the sensor placement search.

5.3.1 NASA-TLX

The questionnaire was based on the NASA Task Load Index (NASA-TLX) (Hart and Lowell, 1988), which is a commonly used subjective mental workload assessment method comprising six items. The mental workload here refers to an index for examining what demands or ecological burdens the load from a particular task imposes on the subject. If the PFI contribution indicator reduced the user's concerns about how many sensors to move and where to move them, then it would mean that the mental demands during simulator operation were reduced. Thus, we measured the cognitive workload with the NASA-TLX questionnaire (Byers et al., 1989), as in related works. The NASA-TLX items are scored on a scale of 0–100, and the details are listed in Table 1.

Table 1 NASA-TLX questions and rating scale

Workload	*Descriptive question*	*Endpoints*
Mental demand (MD)	How much mental and perceptual activity was required? Was the task easy or demanding, simple or complex?	0 to 100
Physical demand (PD)	How much physical activity was required? Was the task easy or demanding, slack or strenuous?	0 to 100
Temporal demand (TD)	How much time pressure did you feel due to the pace at which the tasks or task elements occurred? Was the pace slow or rapid?	0 to 100
Performance (PF)	How successful were you in performing the task? How satisfied were you with your performance?	0 to 100
Effort (EF)	How hard did you have to work (mentally and physically) to accomplish your level of performance?	0 to 100
Frustration (FR)	How irritated, stressed, or annoyed versus content and complacent did you feel during the task?	0 to 100

5.3.2 Interviews

Semi-structured interviews were conducted after the NASA-TLX responses to ascertain the simulator's usability and the need for its functions, including the ability to display the PFI contribution indicator.

5.4 Participants

The user study was conducted with 16 participants (4 women and 16 men, aged 21 to 28 years old, $SD = 1.68$). Each participant was paid the equivalent of 11 US dollars, and none of them had used the sensor simulator before.

5.5 Procedure

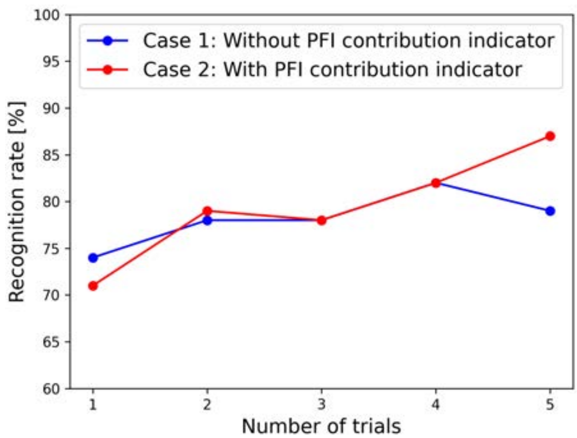
We conducted five trials over a total of 45 minutes. The participants watched a three-minute instructional video on how to use the system and then were given three minutes to practice. Each trial began with installation of the sensors, followed by training data acquisition, and it finished with calculation of the recognition accuracy results. The study participants were divided into two groups of eight for cases 1 and 2.

6 Results

6.1 Recognition accuracy variation

Figure 10 shows the recognition accuracy results over five trials. The recognition accuracy was initially higher for case 1, but by the fifth trial, case 2 produced the higher recognition accuracy. In addition, the participants in case 2 had the highest recognition accuracy over all five trials.

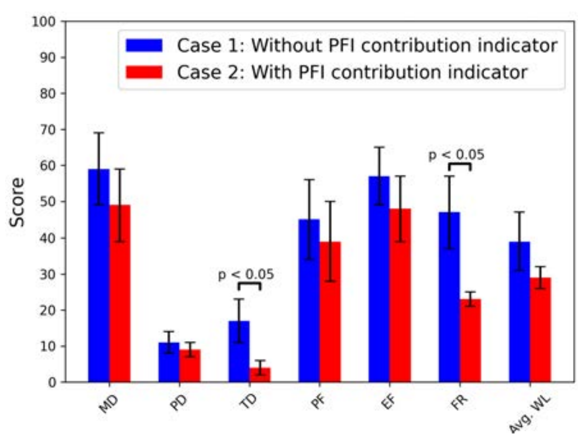
Figure 10 Changes in recognition accuracy over five trials (see online version for colours)



6.1.1 NASA-TLX

Figure 11 shows the NASA-TLX results in terms of the average scores over all participants. The results of a t-test at the 5% significance level ($p < 0.05$) showed a significant difference between TD and FR.

Figure 11 NASA-TLX results (see online version for colours)

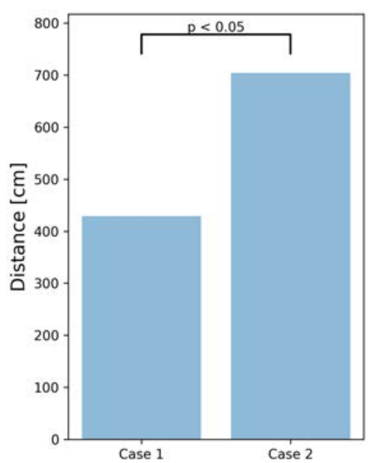


7 Discussion and limitations

7.1 Improvement in recognition rate

As shown in Figure 10, the recognition accuracy in case 2 was similar to that in case 1 until the fourth trial, but case 2 achieved a higher recognition rate in the fifth trial. The reason for the difference in recognition rates is that in case 2, with the PFI contribution indicator, the user could learn the effect of sensor placement on the recognition rate during simulator operation and then choose a final placement that would produce a high recognition rate. Observation of the sensor placements showed that, in both cases 1 and 2, the participants initially tended to place the sensors on the edges of shadows, i.e., the body's shadows projected on the wall where the sensors were placed. This type of placement is not conducive to improving the recognition rate. In general, it is better to place the sensors in a balanced arrangement on the wall, outside the shadow edges, or completely scattered in the shadows (in a rectangular arrangement or close to a well-balanced pentagonal arrangement). Figure 12 shows the total distance that the participants in each case moved the sensors during the user study. A t-test at the 5% significance level ($p < 0.05$) confirmed that there was a significant difference between cases 1 and 2. In particular, the participants in case 2 moved the sensors farther than the participants in case 1 did. In addition, the tendency to place sensors on the edge of the shadows was observed to improve more often in case 2 as trials were repeated.

Figure 12 Total distance that sensors were moved (see online version for colours)



When the participants were asked whether the simulator itself was effective for finding sensor placements, many responded affirmatively. When we asked what they specifically appreciated, the responses included the following: "if a sensor's contribution was low, I could determine that the placement was inappropriate"; and "it was easier to understand which sensors needed to be moved. The results were slightly better after moving the sensor according to what I saw on the actual PFI contribution indicator". In looking for a sensor placement with a high recognition rate, there are two aspects: which sensor to move, and which position to move that sensor to. Case 1 required the participants to consider which sensor to move in each trial, whereas in case 2, by checking the PFI contribution indicator, they could immediately identify which sensors did not contribute to improving the recognition rate. Thus, as confirmed by the interview results, the participants could focus on where to move the sensors. The significant difference between Temporal Demand and Frustration for the NASA-TLX also supports this view.

Overall, these results indicate that the PFI contribution indicator improves the recognition rate in searching for sensor placements. It has been suggested that this is because it reduces the need to think about which sensor to move in each trial. However, it is difficult to decide where to move a sensor according to the PFI contribution indicator alone.

7.2 Trial-and-error

There were several positive responses from the participants regarding the ease of trial-and-error in the user study. These included comments like the following: “I thought it would be good to have a simulator because it is time-consuming to search for the best sensor placement in the real world” and “I think the simulator is very effective because I can repeat the placement many times and use trial-and-error”. When asked if the simulator made trial-and-error easier than in the real world, one participant replied, “in the real world, it’s hard to move sensors and get training data, but the simulator doesn’t have that”. These results suggest that trial-and-error is easier in this simulator than in the real world.

Our study utilised an interactive sensor simulator where the user tries out sensor placements. Other simulators can suggest sensor placements with high recognition accuracy by using computational techniques such as combinatorial optimisation algorithms (Jourdan and de Weck, 2004; Mallardo et al., 2013; Song et al., 2017; Krzakala et al., 2021). However, those approaches can lead to local solutions, and the more sensors there are, the longer the computation time becomes. In addition, they may suggest a complex sensor placement that is not feasible in practice. In contrast, our simulator avoids these problems because it can be operated interactively, and the user can think about the placement to reduce the computation time. We also achieved 95% accuracy for the users’ arbitrary sensor placements over 30 minutes across five trials. Accordingly, we believe that allowing users to interactively perform their own sensor placements through trial-and-error on the simulator is very feasible and will enable quick discovery of a sensor placement with high recognition accuracy.

7.3 Usability

Next, when we asked whether it was difficult to use the keyboard and mouse, all the participants said that it was not difficult. A few participants also gave positive responses such as “it was intuitive and easy to navigate”. Regarding the training data acquisition, there were positive comments such as “I thought it was effective in that I could see the process of acquiring training data”. The simulator also allows placement of sensors in a 3D environment with a mouse click. Moreover, the simulator itself was rated positively, with comments such as “it is more intuitive than setting the coordinates yourself”. Overall, the simulator’s interactive nature was effective in reducing the amount of work required to install sensors and collect training data, but making the simulator more interactive could further improve its usability.

7.4 Roadmap for future simulator UI design

The sensor simulator in this study was designed for ordinary people without specialised knowledge. While we obtained useful feedback during the interviews in terms of how to further improve the sensor search, our focus is how to better support novice developers in sensor-based machine learning system development; as such, we view the current study as a first step. In terms of the next design steps, we intend to adhere to the following roadmap.

Comments common to all the participants included the following: "I would like to be able to see past placements and results"; and "I would like the ability to revert to previous placements with high recognition rates". In addition, there were comments that "while searching, I lost track of past placements and results". Hence, we should implement functionality to check the results of past placements and revert to them. People presumably want this functionality so that they can refer to sensor locations with low contributions to the recognition rate, especially when the PFI contribution indicator is present. This situation occurs when users perform trial-and-error in location configuration, and effective information management can help them understand how to find desired sensor locations.

Regarding the PFI contribution indicator itself, the users commented that "if the contribution level is known, it would be good to have a placement suggestion function from the simulator side to suggest a sensor placement". When the participants in case 2 were asked whether the sensor simulator was effective, one negative comment was "the PFI contribution indicator shows which sensors to move, but in the end it does not tell us where to move them, so moving them may result in a lower recognition rate". We consider this a critical issue, because it indicates the difficulty for both amateur users and machine learning experts to determine why a machine learning model produced the results that it did (Fisher et al., 2018). However, as discussed in Section 7.2, we still believe that trial-and-error is important. Although we have not come up with a specific solution yet, we would like to prioritise this issue in the future.

Currently, our simulator implements two visualisations: a green brightness visualisation of the sensor surface values, and a bar graph of the sensor values. One possible new interactive element would be a display of the sensor values near the sensor when the cursor hovers over it. Also, one participant asked, "is it possible to freely change the viewpoint while acquiring training data?". Currently, the viewpoint can be changed, but it is limited to two types. Hence, we plan to add interactions that will allow the user to freely change viewpoints while acquiring training data. In addition, because we observed that users lost track of which motion data they were capturing, we will consider adding a progress bar or other UI element to indicate which motion training data is currently being captured and how soon the training will be completed. According to a report by Ohtsubo and Yoshida (2014), on the effect of a progress bar's shape on time evaluation, a ring-shaped progress bar with a central angle of 90 degrees yields the most time reduction. Thus, the addition of a progress bar would have the effect of reducing the perception that an operation is taking too long.

7.5 Limitations

7.5.1 Clothing influence

Because the clothing worn by the human avatars in our simulator was not changeable, we did not examine the effect of clothing on the recognition accuracy, nor the effect on the optical sensor of infrared light reflection due to the colour of clothing. We thus intend to measure the effect of clothing colour on the optical sensor in the real world and see if it has any effect on the simulator.

7.5.2 Sensor noise errors

The optical sensor simulator in this study did not consider sensor noise and other errors. If the simulator takes these factors into account, it will be possible to obtain results that are more in line with real-world conditions.

7.5.3 Processing speed limitations

Our simulator reduces the frequency of acquiring sensor values from the optical sensor to match the processing speed on the software side. Accordingly, we believe that by increasing the processing speed of Unity, it will be possible to bring the simulator more in line with real-world conditions. In addition, because the frequency of sensor value acquisition was reduced to match Unity's processing speed, the low fps accuracy and screen choppiness caused problems that made it difficult to finetune the sensor placement. User study participants also pointed out this issue during the interviews. Because difficulty in sensor placement is a critical issue, we will explore ways to implement a lightweight simulator in the future.

7.5.4 Restrictions on recognition targets

In this user study, the sensors were placed on only one wall, and the participants were asked to identify large body movements. As we expect that sensors will also be able to recognise and identify larger movements, our future work will explore the placement of sensors on two or more walls to recognise even more complex movements.

8 Conclusions

This paper reported our development of an optical sensor simulator to determine the necessary features and UI for a real-world simulator. Through a user study including interviews and surveys, we confirmed that the PFI contribution indicator could assist users in exploring sensor placements and generating data with high recognition rates. In particular, the training model for case 1 with the PFI contribution indicator achieved a high recognition rate. Additionally, we found that the simulator was easy to use. We expect that this simulator, which facilitates the entire development process from acquisition of learning data to creation of a learning model via a single tool, will reduce user effort in sensor placement search and make the development of real-world human gesture recognition systems more efficient.

In addition, we have obtained ideas for new features to be implemented from the user's perspective, as described in Section 7.4. Our future works will include improving the simulator to better support user sensor placement searches based on these ideas obtained from the user study. The PFI contribution indicator in this study only shows which sensor to move, which leaves the user to apply trial-and-error to determine a sensor movement direction to improve the recognition accuracy. Hence, we aim to add feedback on sensor placement locations that will improve the learning accuracy, through analysis of past sensor configurations via reinforcement learning, among other methods, in parallel with trial-and-error by the user. Additionally, for greater versatility, we aim to simulate not only optical sensors but also inertial measurement unit sensors.

Acknowledgements

This work was partly supported by JST PRESTO Grant Number JPMJPR2134.

References

Buchmayr, M., Kurschl, W. and Küng, J. (2011) 'A simulator for generating and visualizing sensor data for ambient intelligence environments', *Procedia Computer Science*, Vol. 5, pp.90–97.

Byers, J.C., Bittner, A.C. and Hill, S.G. (1989) 'Traditional and raw task load index (TLX) correlations: are paired comparisons necessary?', *Advances in Industrial Ergonomics and Safety*, Vol. 1.

Fisher, A., Rudin, C. and Dominici, F. (2018) *All Models are Wrong, But Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously*, arXiv.

Gazi, A.H., Lis, P., Mohseni, A., Ompi, C., Giuste, F.O., Shi, W., Inan, O.T. and Wang, M.D. (2021) 'Respiratory markers significantly enhance anxiety detection using multimodal physiological sensing', *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp.1–4.

Hart, S.G. and Lowell, E.S. (1988) 'Development of NASA-TLX (task load index): results of empirical and theoretical research', *Advances in Psychology*, Vol. 52, pp.139–183, North-Holland.

Huang, N., Lu, G., and Xu, D. (2016) 'A permutation importance-based feature selection method for short-term electricity load forecasting using random forest', *Energies*, Vol. 9, No. 10, p.767.

Jourdan, D.B. and de Weck, O.L. (2004) 'Layout optimization for a wireless sensor network using a multi-objective genetic algorithm', *IEEE 59th Vehicular Technology Conference, VTC 2004-Spring (IEEE Cat. No. 04CH37514)*, Vol. 5, pp.2466–2470.

Kaushik, S. and Birok, R. (2021) 'Heart failure prediction using XGBoost algorithm and feature selection using feature permutation', *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp.1–6.

Kawashima, T., Kawanishi, Y., Ide, I. and Murase, H. (2017) 'Action recognition from extremely low-resolution thermal image sequence', *14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp.1–6.

Krzakala, P., Assouel, A., Dunitz, M., Anterrieu, E., Cabot, F., Khazaal, A., Rodriguez-Fernandez, N., Rougé, B., Morel, J.M., Kerr, Y.H. and Colom, M. (2021) 'Irregular layout for a satellite's interferometric array', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 14, pp.9408–9423.

Mallardo, V., Aliabadi, M.H. and Khodaei, Z.S. (2013) 'Optimal sensor positioning for impact localization in smart composite panels', *Journal of Intelligent Material Systems and Structures*, Vol. 24, No. 5, pp.559–573.

Manabe, H. (2013) 'Multi-touch gesture recognition by single photoreflector', *Adjunct Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13 Adjunct)*, Association for Computing Machinery, New York, NY, USA, pp.15–16.

Matsuo, K., Hachiuma, R., Saito, H. and Sugiura, Y. (2020) 'Interactive defect complement system for 3D scan data of the room', *Correspondences on Human Interface*, CD-ROM.

Murakami, M., Kikui, K., Suzuki, K., Nakamura, F., Fukuoka, M., Masai, K., Sugiura, Y. and Sugimoto, M. (2019) 'AffectiveHMD: facial expression recognition in head mounted display using embedded photo reflective sensors', *ACM SIGGRAPH 2019 Emerging Technologies (SIGGRAPH '19)*, Association for Computing Machinery, New York, NY, USA, Article 7, pp.1–2.

Ohtsubo, M. and Yoshida, K. (2014) 'How does shape of progress bar effect on time evaluation', *International Conference on Intelligent Networking and Collaborative Systems*, pp.316–319.

Park, J., An, K. and Choi, J. (2014) 'Realistic 3D simulation of multiple human recognition over perception sensor network', *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp.507–512.

Rossmann, J., Hempe, N., Emde M. and Steil T. (2012) 'A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications', *ROBOTIK; 7th German Conference on Robotics*, pp.1–6.

Schelthoff, K., Jacobi, C., Schlosser, E., Plohmann, D., Janus, M. and Furmans, K. (2022) 'Feature selection for waiting time predictions in semiconductor wafer fabs', *IEEE Transactions on Semiconductor Manufacturing*, Vol. 35, No. 3, pp.546–555.

Song, S., Qiu, X., Wang, J., and Meng, M.Q.H. (2017) 'Design and optimization strategy of sensor array layout for magnetic localization system', *IEEE Sensors Journal*, Vol. 17, No. 6, pp.1849–1857.

Xia, C. and Sugiura, Y. (2021) 'Optimizing sensor position with virtual sensors in human activity recognition system design', *Sensors*, Vol. 21, No. 20, p.6893.

Xia, C., Saito, A. and Sugiura, Y. (2022) 'Using the virtual data-driven measurement to support the prototyping of hand gesture recognition interface with distance sensor', *Sensors and Actuators A: Physical*, Vol. 338, p.113463.

Notes

1 https://www.0c7.co.jp/products/sensing/xsens/product-lineup/mvn-animate.html.