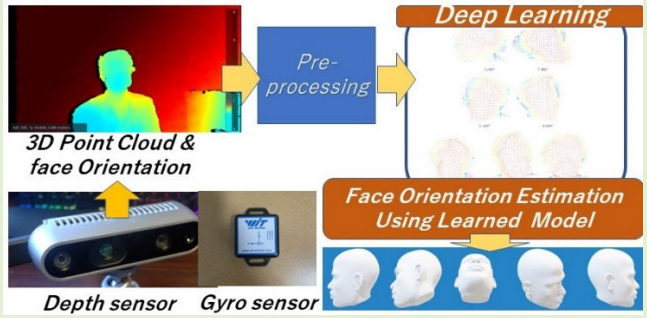
Depth-Gyro Sensor-Based Extended Face Orientation Estimation Using Deep Learning

Chinthaka Premachandra, *Senior Member, IEEE*, and Yuta Funahashi

***Abstract*—Conventional face direction estimation techniques detect the characteristic parts of the face, such as the nose, eyes, and mouth, and estimate the face orientation based on the movements of these features. However, these methods cannot accurately estimate the face direction when the characteristic parts of the face are hidden; for example, when the face is turned sideways or a mask is worn. Face detection using point cloud data has been investigated as a solution to these issues. Previous studies applied five classes of face direction estimation for the head using 3-D point cloud data. However, considering the practical use of driver assistance systems that verify the driver's status, these five classes are not sufficient for accurately detecting the face direction, and a more precise horizontal wide-range angle detection approach is necessary. In this study, we acquired 3-D point cloud data in k (where $k > 5$) classes while accurately measuring the horizontal angle of the face during the acquisition of the training data using gyroscopic sensors. The training data captured by this depth-gyro sensor integration generates accurate depth data for each direction. As a result, a low number of point cloud data samples for each face direction were sufficient for generating the directional classification model. Therefore, this depth-gyro sensor integrated data capturing significantly reduces the amount of required training data. Furthermore, we applied a weight reduction process for the point cloud data to reduce the training time and performed deep learning to estimate the face direction. The proposed method achieved high performance in face direction detection using deep learning, even with a comparatively small dataset.**



***Index Terms*— Deep learning, depth sensor, gyro sensor, point cloud data, point cloud data resolution reduction.**

I. Introduction

TECHNOLOGY for estimating the direction of the human face has various applications, such as driver assistance systems to prevent car and motorcycle drivers from distracted driving that includes looking away or falling asleep, systems to prevent cheating in examinations, and in marketing to analyze the preshelf behavior based on facial orientation (see Fig. 1). For example, Virtual Y has been applied extensively in recent years. YouTubers, i.e., active users of YouTube, use the technology for estimating the face direction to create virtual characters by linking the movements of users and 3-D avatars.

Manuscript received 14 May 2023; accepted 13 July 2023. Date of publication 21 July 2023; date of current version 31 August 2023. This work was supported in part by the Branding Research Fund of the Shibaura Institute of Technology. The associate editor coordinating the review of this article and approving it for publication was Dr. Hari P. Gupta. *(Corresponding author: Chinthaka Premachandra.)*

Chinthaka Premachandra is with the Graduate School of Engineering and Science, Shibaura Institute of Technology, Tokyo 135-8548, Japan (e-mail: chintaka@sic.shibaura-it.ac.jp).

Yuta Funahashi is with the Department of Electronics Engineering, School of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan.

This article has supplementary downloadable material available at https://doi.org/10.1109/JSEN.2023.3296531, provided by the authors.

Digital Object Identifier 10.1109/JSEN.2023.3296531

Conventional techniques for estimating the face direction detect the characteristic parts of the face, such as the nose, eyes, and mouth, and estimate the face orientation based on the movements of these parts [1], [2], [3], [4], [5]. However, when the face is turned to one side or a mask is worn, the characteristic parts of the face are hidden, and these methods cannot estimate the face direction. Face detection using point cloud data has been investigated as a solution to the aforementioned issues.

In a previous study, a method for estimating the face direction in five classes (directions) was developed using 3-D point cloud data [6]. This estimation model was based on the deep learning of 3-D point cloud data in five face directions: frontal, diagonal frontal, right, left, and horizontal, and the authors confirmed that their method was effective in estimating the five face direction classes.

However, the ambiguous five-class classification is not sufficient for accurate face direction detection when considering the practical use of driver assistance systems that verify the status of car drivers. Therefore, more accurate detections of the various horizontal angles of the face are necessary.

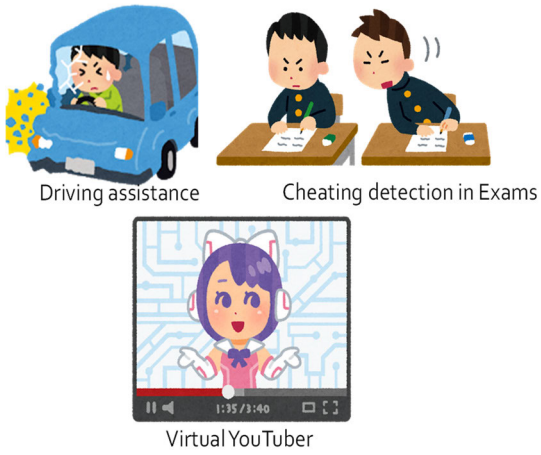


Fig. 1. Applications of face direction estimation.

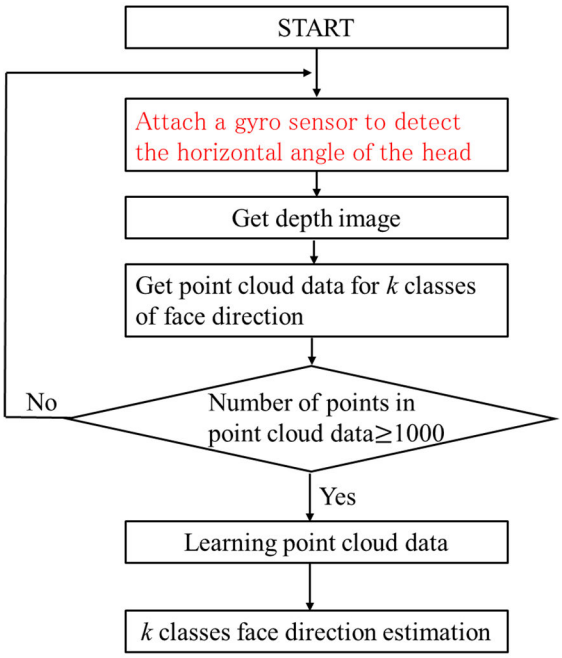


Fig. 2. System overview of the study.

In this study, we aim to expand face direction estimation by acquiring point cloud data for training over a wide range of faces. First, we obtain the point clouds for training when the subject faces the k-direction. Thereafter, we perform deep learning on the point clouds to achieve the k-class face orientation estimation.

An overview of the proposed system is presented in Fig. 2. We used a depth sensor and gyro sensor to conduct the study. Depth sensors can be applied to face direction detection in dark environments, such as nighttime, because each pixel of the depth camera contains distance information instead of color information. Furthermore, the use of data (point cloud data) from a depth camera protects the privacy of the user, thereby enabling practical applications [7], [8], [9]. As shown in Fig. 2, we used the gyro sensor to measure the face direction correctly while acquiring the original point cloud data for training and testing from a depth camera (as explained later). The training data captured by this depth-gyro sensor integration generate accurate depth data for each direction. As a result, a low number of point cloud data samples (100 samples) for each face direction were sufficient for generating the directional classification model. Therefore, this depth-gyro sensor integrated data capturing significantly reduces the amount of required training data. In the training stage, we first reduced the density of the point cloud data and downsampled them to a maximum of 1000 points to reduce the weight of the data, which reduced the training time. The same process was then applied to the test data during the verification experiment. Consequently, the proposed method exhibited rapid training and testing times. As per the validation experiments, high accuracy was achieved in the face direction estimation for the (k = 7, 9, 11, and 13) classes while using a smaller number of training data samples (100 samples) for each direction. Therefore, the application of point cloud data for face direction detection using the proposed method was validated.

Fig. 3. Depth camera used in this study.

II. Acquisition of Training Data

A. 3-D Point Cloud Data and Usefulness of Depth Camera

A depth camera was used to obtain the 3-D point cloud data. These data are generated using the distance image that is obtained from the depth camera, and the distance information from the depth camera is stored in each pixel of the obtained image. As the data are generated from distance information, they are not easily affected by the surrounding conditions and are effective even at night or under strong sunlight. Furthermore, the privacy of the subject can be maintained because only the point cloud data are used.

The depth camera that was used in this study was the RealSense D435 depth camera from Intel[1] [10], which was used in various studies [11], [12], [13], [14], [15], [16]. The equipment used in this study is shown in Fig. 3, and the specifications thereof are listed in Table I.

B. Application of Gyro Sensor

A gyro sensor was attached to the head to measure the angle of the face, and relevant training data for the face angle were

[1]Registered trademark.

TABLE I
SPECIFICATIONS OF THE DEPTH CAMERA

Item	Specification
Depth output resolution	Up to 1280 × 720
Depth frame rate	Up to 90 fps
Length × depth × height	90 mm × 25 mm × 25 mm
Use environment	Indoor/Outdoor
Ideal range	0.3 m to 3 m

Fig. 4. Gyro sensor used for measuring face direction during training data acquisition.

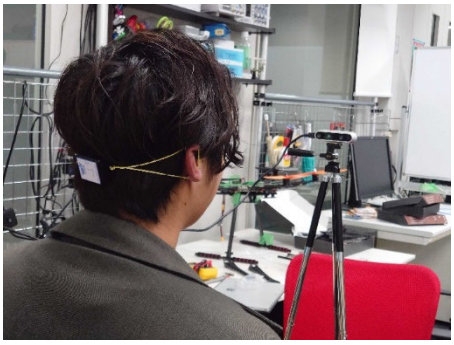


Fig. 5. Data acquisition environment.

acquired during the training stage. A gyro sensor, which is also known as an angular-rate sensor, detects changes in the rotation and orientation of an object as angular velocity and outputs it as an electrical signal. The angular velocity is the rotational speed per unit time.

In this study, we used the WT901BLECL gyro sensor manufactured by Witmotion Inc., [17]. The gyro sensor can measure angles through a wireless connection via Bluetooth and has a small size (51.3 × 36 × 15 mm), making it suitable for measuring the real face angle while acquiring the point cloud data of the head. The gyro sensor used in this study is shown in Fig. 4, and the point cloud data acquisition setting for the subject is shown in Fig. 5. The specifications of the gyro sensor are presented in Table II.

C. Head Extraction by Density-Based Spatial Clustering of Applications With Noise (DBSCAN)

When point cloud data are generated from the depth image that is obtained using a depth camera, only the necessary point cloud data of the head are extracted from the original data using a clustering method known as DBSCAN [18].

TABLE II
SPECIFICATIONS OF THE GYRO SENSOR

Item	Specification
Current	<40 mA
Working Temperature	−40℃ ~ +85℃
Battery life	about 10 hours
Data output frequency	0.2–200Hz
Interface	Type-C, Low-energy Bluetooth 5.0
Baud Rate	115200 (default, cannot be changed)
Voltage	3.3–5V
Output Content	3-axis Acceleration+Angle+Angular Velocity+Magnetic Field+Quaternion
Range	Acceleration (±16g), Gyroscope (±2000°/s), Magnet Field (±4900μT), Angle (X, Z-axis: ±180°, Y ±90°)
Resolution	Acceleration (0.005g), Gyroscope (0.61°/s), Magnet Field (16 bits)
Angle Accuracy(after calibrated)	X, Y-axis: 0.05° (Static) , X, Y-axis: 0.1° (Dynamic)

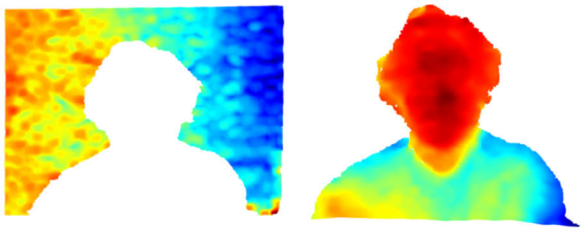


Fig. 6. Example of DBSCAN with background and people cropped.

DBSCAN is a point density-based clustering method. The process includes two parameters: *eps* and *min_points*. The *eps* parameter determines the clustering range for a single point. Smaller values enable finer clustering, whereas larger values result in more general clustering. In the experiment for this study, we set the *eps* parameter to 0.01.

The *min_points* parameter defines the clusters, provided that there exist more points than the *min_points* parameter within the radius that is determined by the *eps* parameter. When this parameter is increased, the points with higher density are considered as the same cluster; however, an extreme increase in the value will not result in any clustering [18]. In this experiment, we set the *min_points* parameter to 10. An example of the extraction of the human region, including the head, using DBSCAN is shown in Fig. 6.

D. Normalization of Point Cloud Data

Following the application of DBSCAN, we normalized the point cloud data to equalize the weights of the values as a preprocessing step prior to deep learning. The equation for the normalization is given as follows:

$$x_{\text{out}} = \frac{x_{\text{in}} - \sigma}{\max\{|x_{\text{in}} - \sigma|\}} \tag{1}$$

where x_{in} is the input, x_{out} is the output, and σ is the average of the point cloud data with respect to the x-, y-, and z-axes. The difference between each axis direction in 3-D was calculated for the point cloud data, and the value was divided by the

TABLE III
CLASSIFICATION OF FACE DIRECTION

Number	Face direction
1	Horizontal angle of face to camera: 90°
2	Horizontal angle of face to camera: 60°
3	Horizontal angle of face to camera: 30°
4	Horizontal angle of face to camera: 0°
5	Horizontal angle of face to camera: −30°
6	Horizontal angle of face to camera: −60°
7	Horizontal angle of face to camera: −90°

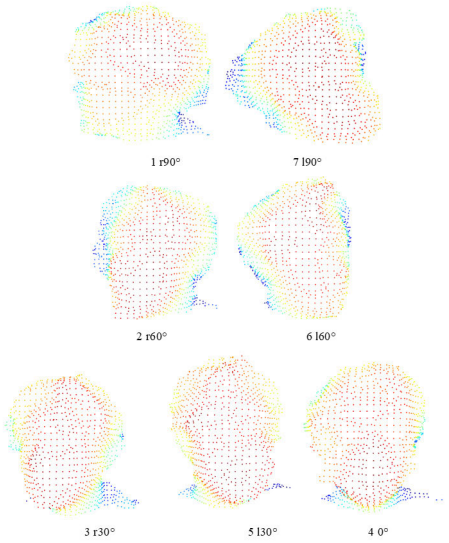


Fig. 7. Sample point cloud data for seven classes.

maximum value of the difference at each point to normalize the value between 0 and 1.

E. Classification of Face Direction

In the classification of the face direction, the state of the classification is described by seven classes ($k = 7$). However, the proposed method works for $k > 7$. Thus, in the experiments, we also tested the method for $k > 7$.

Table III summarizes the classification for the seven classes ($k = 7$), while Fig. 7 shows the samples of the point cloud data for the seven classes.

III. DEEP LEARNING OF HEAD POINT CLOUD DATA

In this study, we performed deep learning of the point cloud data of the head that were acquired using the depth camera and used the learned model for face orientation estimation.

The deep learning library and model that were used in this study are listed in Table IV.

A. Order Invariance and Movement Invariance

Two aspects must be considered to handle point cloud data with deep learning: order invariance and translation invariance [19]. Order invariance is a property whereby the output does not change, even if the order of the points is changed. For example, the intensity and color of the light, which are arranged in a regular order, are stored in each pixel in an RGB image. If the order is changed, the light intensity and color that are stored in the pixels also change, and subsequently, the output also changes. However, as the 3-D point cloud data that were used in this study were generated from the distance information obtained from the depth camera, the information in the point cloud data remained the same even if the order of the points was changed, and thus, the output remained the same.

TABLE IV
SOFTWARE USED FOR DEEP LEARNING

Deep learning library	PyTorch
Deep learning model	PointNet

Translation invariance is a property whereby the output does not change even if the input is moved in parallel or rotated. Equation (2) provides the expression for the translation invariance

$$f\left(x_1 + \boldsymbol{r}, x_2 + \boldsymbol{r}, + \cdots x_n + \boldsymbol{r}\right) = \boldsymbol{f}\left(x_1, x_2, \ldots, x_n\right) \quad (2)$$

where r is an arbitrary vector.

As can be observed from (2), the output cannot change even if the input is moved by an arbitrary vector r. This is referred to as translation invariance. The equation for rotational invariance is given as follows:

$$f\left(Rx_1, Rx_2, \ldots, Rx_n\right) = \boldsymbol{f}\left(x_1, x_2, \ldots, x_n\right) \quad (3)$$

where R is an arbitrary rotation matrix.

As can be observed from (3), the input by an arbitrary rotation matrix R does not change the output with rotational invariance.

B. Deep Learning Model: PointNet

We used PointNet by Charles et al. [20] as our deep learning model. Unlike images, where pixels are regularly arranged, point cloud data have an irregular structure. Therefore, they are often transformed into a 3-D voxel grid or image for deep learning. However, several problems arise with such data transformation. For example, the obtained data become unnecessarily large, and the training process would require substantial time. However, PointNet can perform deep learning using point cloud data as input.

PointNet considers the following two points. First, PointNet achieves order invariance using a target function known as max pooling, whereby the element with the largest value among the input elements is output. An example of max pooling is given as follows: if the input data are {1, 2, 3, 4, 5}, max pooling outputs the element with the largest value, which is 5. This is indicated in the following equation:

$$\max\{1, 2, 3, 4, 5\} = 5. \quad (4)$$

However, the output remains 5 even if the order of the input data is changed to {2, 4, 5, 1, 3}, as shown in the following equation:

$$\max\{2, 4, 5, 1, 3\} = 5. \quad (5)$$

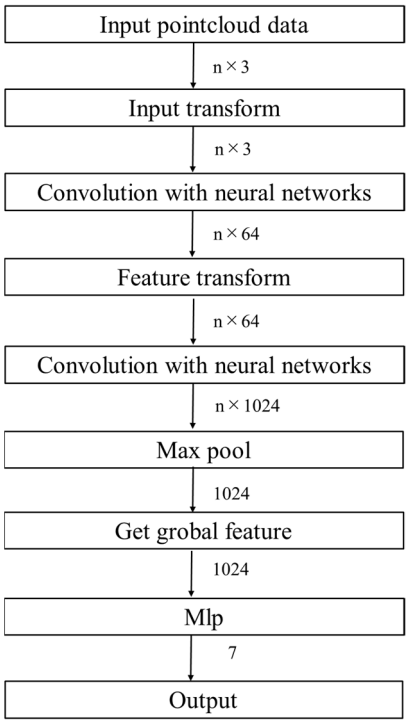


Fig. 8. Flow of deep learning using PointNet.

PointNet utilizes max pooling to achieve order invariance and an affine transformation matrix [21] to achieve translation invariance. An affine transformation is a combination of a linear transformation and a translation [22], [23], [24]. The affine transformation matrix is obtained through T-Net [25], a neural network that is order-invariant and uses max pooling. The input point cloud data are then shifted and rotated, and the affine transformation matrix is applied to the input 3-D point cloud data to approximate shift invariance. Consequently, PointNet establishes both translation invariance and order invariance.

C. Overall Flow of PointNet

Fig. 8 shows the overall flow of the deep learning process using PointNet, where n indicates the number of points. As the data used in this study were 3-D point cloud data, the input data volume was $n \times 3$. Movement invariance was obtained by applying the affine transformation matrix to the input data in the transform layer. Subsequently, the data were convolved using a neural network and passed through a multilayer perceptron (MLP). This was repeated and max pooling was performed to achieve order invariance. Thus, an overall feature set of 1024 dimensions was obtained. Thereafter, the obtained features were passed through the MLP to classify k classes.

IV. Verification Experiments

A. Experimental Environment

Depth images were captured using a depth camera and tripod, with the subject seated on a chair approximately 50–60 cm from the head and at the same height as the eyes of the subject. Data acquisition was performed with the cooperation of ten people (subjects).

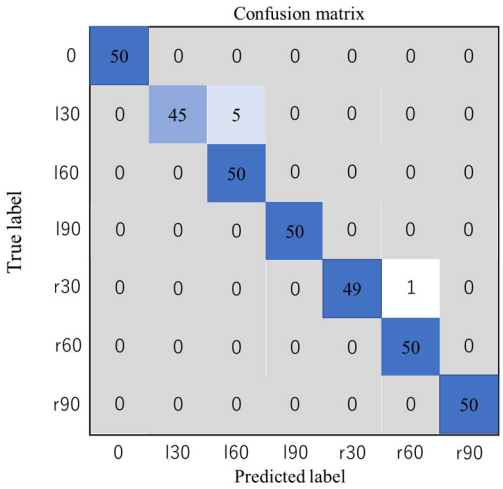


Fig. 9. Confusion matrix of validation experimental results.

For the seven-class ($k = 7$) classification, 700 point cloud datasets (100 for each of the seven classes of horizontal angles of the head) were prepared as the training datasets, whereas 350 datasets (50 for each of the seven classes) were prepared as the validation datasets to measure the accuracy. Similar datasets were prepared for $k = 9$, 11, and 13.

The experiments were conducted using a computer with the following configuration: Intel Core i5-7500 at 3.40 GHz, Intel HD Graphics 520, 16 GB.

B. Experimental Results

Fig. 9 presents the confusion matrix of the results when $k = 7$, and the number of training epochs was set to 15. The vertical and horizontal axes represent the true and estimated values, respectively. Fig. 9 shows that at all horizontal angles, which indicates that the estimation was highly accurate, where l and r indicate left and right, respectively, which were the directions in which the subject looked. For example, $l30$ indicates that the subject was looking 30° to the left with respect to the camera. The average learning time was 842.406 s and the average estimation time was 2.53 s.

We determined the accuracy, precision, and recall based on four parameters, namely, true positive (TP), true negative (TN), false negative (FN), and false positive (FP).

Accuracy is the percentage of total data that are correctly classified, where higher accuracy indicates better performance. The formula is given as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}. \tag{6}$$

Precision is the percentage of correct answers among those that are predicted as true by the trained model. Higher precision indicates fewer incorrect classifications. The precision is calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

Finally, recall is the percentage of correct answers among those that are true. It is a measure of the number of angles that can

TABLE V
SUMMARY OF ACCURACY, PRECISION, AND RECALL

Degree [°]	0	*l*30	*l*60	*l*90	*r*30	*r*60	*r*90	Overall
Accuracy [%]	100	90	100	100	98	100	100	98.3
Precision [%]	100	100	90.1	100	100	98	100	98.3
Recall [%]	100	90	100	100	98	100	100	98.3

TABLE VI
ACCURACY FOLLOWING FACE DIRECTION CLASS CHANGES

No of classes (k)	7	9	11	13
Accuracy (%)	98.3	95.2	91.8	89.2

be estimated without an error and is calculated as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

The experimental results of accuracy, precision, and recall when $k = 7$ are summarized in Table V. Table V demonstrates that the accuracy, precision, and recall reached over 98% in the experiments, where l and r indicate left and right, respectively, which were the directions in which the subject looked, as above described.

Fig. 10 presents a sample of the orientation estimation results that were obtained in this experiment (on the left) and the face regions obtained by DBSCAN for each orientation (on the right), where l and r denote left and right, respectively, GT is the labeled direction, and Pred is the predicted direction.

Furthermore, the face direction estimation accuracies for additional classes (when $k = 7$, 9, 11, and 13) are summarized in Table VI. It can be observed from the table that the point cloud-based face direction detection exhibited high accuracy. The accuracy was very high for directional classes 7 and 9, which means that the application possibility is high. It is confirmed that the proposed method exhibited high performance with a lower number of training data (100 samples for each direction) in deep learning.

Therefore, the depth-gyro sensor integrated data capturing significantly reduces the amount of required training data while achieving high classification performance.

In this study, we used a depth sensor to obtain the point cloud data for k ($k = 7$, 9, 11, and 13) classes while accurately measuring the horizontal angle of the head using a gyro sensor. The gyro sensor sent data to the computer through a Bluetooth wireless connection. However, the numerical angle value from the gyro sensor was not very stable in certain cases. We believe that this issue can be resolved by measuring the horizontal angle of the head using a microcomputer, such as an Arduino or Raspberry Pi, with a wired connection and a gyro sensor.

C. Comparisons With Conventional Methods

As mentioned in Section I, the proposed face orientation estimation method is an improvement upon a previous face direction orientation estimation method [6] that used a depth camera. In a direct comparison, the proposed method with depth-gyro sensor integration was able to improve the number of estimation directions with high accuracy.

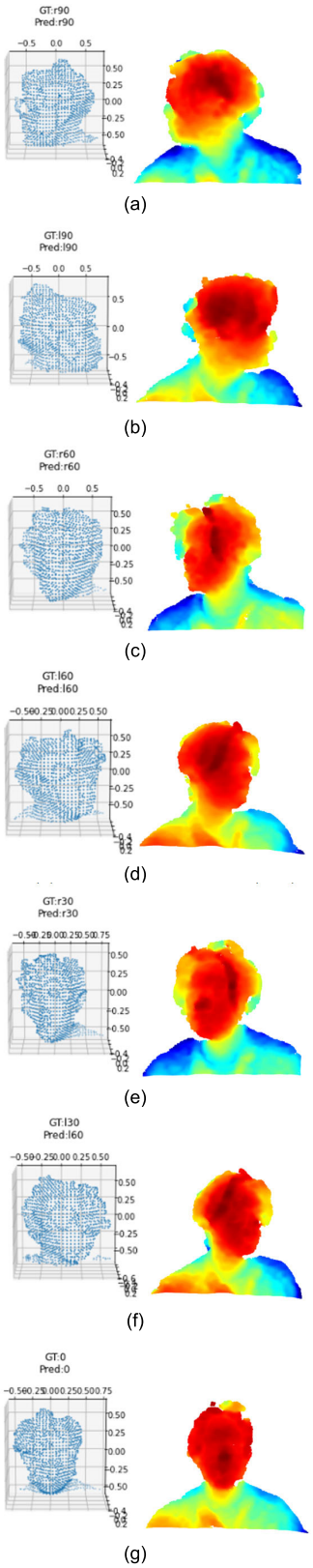


Fig. 10. Estimation results and sample distance images. (a) *r*90 is estimated as *r*90 (true). (b) *l*90 is estimated as *l*90 (true). (c) *r*60 is estimated as *r*60 (true). (d) *l*60 is estimated as *l*60 (true). (e) *r*30 is estimated as *r*30 (true). (f) *l*30 is estimated as *l*60 (false). (g) 0 is estimated as 0 (true).

TABLE VII
PERFORMANCE COMPARISON WITH CONVENTIONAL METHODS

Method	Accuracy (7 face directional classes estimation
Normal camera-based method 1 [26]	96.0%
Normal camera-based method 2 [1]	96.8%
Proposed method	**98.3**

In addition, we compared the proposed method with conventional face orientation estimation methods that use other devices, such as normal cameras. The comparison results are presented in Table VII, which pertains to the seven-directional class estimation mentioned in Table V. The accuracy of the proposed method showed better performance than conventional methods using 2-D cameras. Furthermore, the proposed method can be applied at night and in places where the user's privacy needs to be protected since it estimates the face orientation based solely on the point cloud data.

V. CONCLUSION

In this study, we estimated the horizontal direction of the head using the gyro sensor during the acquisition of point cloud data from the depth sensor for the learning stage of deep learning. The training data captured by this depth-gyro sensor integration generate accurate depth data for each direction. As a result, a low number of point cloud data samples for each face direction were sufficient for generating the directional classification model. Therefore, this depth-gyro sensor integrated data capturing significantly reduces the amount of required training data. We further applied a weight reduction process for the point cloud data to reduce the learning time. The proposed method achieved high performance in detecting the face direction using deep learning using a comparatively small dataset. The accuracy, precision, and recall reached over 98% in our experiments.

In the future, we plan to solve several errors resulting from the gyro sensor during the data acquisition process. Furthermore, real-time processing on small hardware should be considered for real applications.

REFERENCES

[1] J. Okubo, B. Sugandi, H. Kim, J. Kooi Tan, and S. Ishikawa, "Face direction estimation based on eigenspace technique," in *Proc. Int. Conf. Control, Autom. Syst.*, Oct. 2008, pp. 1264–1267.

[2] X. Wang, Q. Ruan, and Y. Ming, "3D face recognition using corresponding point direction measure and depth local features," in *Proc. IEEE 10th Int. Conf. SIGNAL Process.*, Oct. 2010, pp. 86–89.

[3] H. Tamaki et al., "Novel application of a range image sensor to eye gaze estimation by using the relationship between face and eye directions," in *Proc. 9th Int. Conf. Sens. Technol. (ICST)*, Dec. 2015, pp. 443–446.

[4] Y. Gu, L.-T. Hsu, L. Xie, and S. Kamijo, "Accurate estimation of pedestrian orientation from on-board camera and inertial sensors," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E99-A, no. 1, pp. 271–281, 2016.

[5] G. Yanlei and S. Kamijo, "Bicyclist recognition and orientation estimation from on-board vision system," *Int. J. Automot. Eng.*, vol. 6, no. 2, pp. 67–73, 2015.

[6] S. Sasaki and C. Premachandra, "Head posture estimation by deep learning using 3-D point cloud data from a depth sensor," *IEEE Sensors Lett.*, vol. 5, no. 7, pp. 1–4, Jul. 2021.

[7] Y. He, J. Yang, K. Xiao, C. Sun, and J. Chen, "Pose tracking of spacecraft based on point cloud DCA features," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5834–5843, Mar. 2022.

[8] P. Owerko and T. Owerko, "Novel approach to inspections of as-built reinforcement in incrementally launched bridges by means of computer vision-based point cloud data," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11822–11833, May 2021.

[9] R. C. Luo and S. Y. Chen, "Human pose estimation in 3-D space using adaptive control law with point-cloud-based limb regression approach," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 51–58, Feb. 2016.

[10] *Realsense HP*. (Nov. 25, 2021). [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/

[11] K. Huang, J. Tang, J. Yuan, and Z. Dong, "Weld line detection using RealSense depth camera based on depth map," in *Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, vol. 9, Dec. 2020, pp. 595–600.

[12] Z. Shareef and S. Reddy, "PrEduSense: Smart education kit for pre-primary classes using Intel® real sense," in *Proc. 1st India Int. Conf. Inf. Process. (IICIP)*, Aug. 2016, pp. 1–4.

[13] H. Matsumura and C. Premachandra, "Deep-learning-based stair detection using 3D point cloud data for preventing walking accidents of the visually impaired," *IEEE Access*, vol. 10, pp. 56249–56255, 2022.

[14] Y. Endo and C. Premachandra, "Development of a bathing accident monitoring system using a depth sensor," *IEEE Sensors Lett.*, vol. 6, no. 2, pp. 1–4, Feb. 2022.

[15] Y. Ito, C. Premachandra, S. Sumathipala, H. W. H. Premachandra, and B. S. Sudantha, "Tactile paving detection by dynamic thresholding based on HSV space analysis for developing a walking support system," *IEEE Access*, vol. 9, pp. 20358–20367, 2021.

[16] J. Hu, Y. Niu, and Z. Wang, "Obstacle avoidance methods for rotor UAVs using RealSense camera," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 7151–7155.

[17] *Wit Motion HP*. (Nov. 25, 2021). [Online]. Available: https://www.wit-motion.com/9-axis/wt901blecl-mpu9250-high-precision.html

[18] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A densitiy-based algorithm for discovering clusers," Inst. Comput. Sci., Univ. Munich Oettingen-straße, Miinchen, Germany, vol. 67, Tech. Rep. D-80538.

[19] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1–12.

[20] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[21] R. Turcinhodzic, "Graphical presentation of affine transformation crypt-analysis," in *Proc. 5th Int. Conf. Comput. Sci. Educ.*, Aug. 2010, pp. 1555–1559.

[22] A. Nikolaidis, "Affine transformation invariant image watermarking using moment normalization and radial symmetry transform," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 2729–2732.

[23] D. Stojanovic, I. Djurovi, and B. R. Vojcic, "Interference analysis of multicarrier systems based on affine Fourier transform," *IEEE Trans. Wireless Commun.*, Vol. 8, no. 6, pp. 2877–2880, Jun. 2009.

[24] P. Mondal and S. Banerjee, "Motion estimation in medical video sequences using affine transform," in *Proc. 25th IEEE Int. Symp. Computer-Based Med. Syst. (CBMS)*, Jun. 2012, pp. 1–4.

[25] J. Kossaifi, A. Bulat, G. Tzimiropoulos, and M. Pantic, "T-Net: Parametrizing fully convolutional nets with a single high-order tensor," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7814–7823.

[26] S. Abe, M. Morimoto, and K. Fujii, "Estimating face direction from wideview surveillance camera," in *Proc. World Autom. Congr.*, Sep. 2010, pp. 1–6.

Chinthaka Premachandra (Senior Member, IEEE) was born in Sri Lanka. He received the B.Sc. and M.Sc. degrees from Mie University, Tsu, Japan, in 2006 and 2008, respectively, and the Ph.D. degree from Nagoya University, Nagoya, Japan, in 2011.

From 2012 to 2015, he was an Assistant Professor with the Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science, Tokyo, Japan. From 2016 to 2017, he was an Assistant Professor with the Department of Electronic Engineering, School of Engineering, Shibaura Institute of Technology, Tokyo, where he was an Associate Professor from 2018 to 2022. In 2022, he was promoted to a Professor with the Department of Electronic Engineering, Graduate School of Engineering, Shibaura Institute of Technology, where he is currently the Manager of the Image Processing and Robotic Laboratory. His research interests include AI, UAV, image processing, audio processing, intelligent transport systems (ITS), and mobile robotics.

Dr. Premachandra is a member of IEICE, Japan; SICE, Japan; RSJ, Japan; and SOFT, Japan. He received the IEEE SENSORS LETTERS Best Paper Award from the IEEE Sensors Council in 2022 and the IEEE Japan Medal from the IEEE Tokyo Section in 2022. He has also received the FIT Best Paper Award and the FIT Young Researchers Award from IEICE and IPSJ, Japan, in 2009 and 2010, respectively. He has served as a steering committee member and an editor for many international conferences and journals. He is the Founding Chair of the International Conference on Image Processing and Robotics (ICIPRoB), which is technically co-sponsored by IEEE. He has been serving as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS (R-AL) and *IEICE Transactions on Information and Systems*.

Yuta Funahashi received the B.S. degree in electronic engineering from the Shibaura Institute of Technology, Tokyo, Japan, in 2022.

His research interests include depth data processing, image processing, and embedded systems.