

# A Hand Gesture Recognition Method using Inertial Sensor for Rapid Operation on Embedded Device

Sangyub Lee<sup>1,2</sup>, Jaekyu Lee<sup>1,2</sup> and Hyeonjoong Cho<sup>2,\*</sup>

<sup>1</sup>Embedded SW Research R&D Center, Korea Electronics Technology Institute  
Seongnam, South Korea

<sup>2</sup>Department of Computer and Information Science, Korea University  
Sejong, South Korea

[e-mail: syublee@keti.re.kr, jaekyu@keti.re.kr, raycho@korea.ac.kr]

\*Corresponding author: Hyeonjoong Cho

*Received September 2, 2019; revised October 24, 2019; accepted November 19, 2019;  
published February 29, 2020*

---

## Abstract

We propose a hand gesture recognition method that is compatible with a head-up display (HUD) including small processing resource. For fast link adaptation with HUD, it is necessary to rapidly process gesture recognition and send the minimum amount of driver hand gesture data from the wearable device. Therefore, we use a method that recognizes each hand gesture with an inertial measurement unit (IMU) sensor based on revised correlation matching. The method of gesture recognition is executed by calculating the correlation between every axis of the acquired data set. By classifying pre-defined gesture values and actions, the proposed method enables rapid recognition. Furthermore, we evaluate the performance of the algorithm, which can be implanted within wearable bands, requiring a minimal process load. The experimental results evaluated the feasibility and effectiveness of our decomposed correlation matching method. Furthermore, we tested the proposed algorithm to confirm the effectiveness of the system using pre-defined gestures of specific motions with a wearable platform device. The experimental results validated the feasibility and effectiveness of the proposed hand gesture recognition system. Despite being based on a very simple concept, the proposed algorithm showed good performance in recognition accuracy

---

**Keywords:** Pattern recognition, hand gesture recognition, motion recognition, inter-correlation matching, HUD, IMU sensor, wearable device.

---

A preliminary version of this paper was presented at APIC-IST 2019, and was selected as an outstanding paper. This research was supported by a research grant from the IT R&D program of MKE/KEIT, the Korean government [10062943, Development wearable device and services for industrial convergence that support intelligent drivers's ADAS system].

## 1. Introduction

Present-day automobiles typically include a navigation system on the dashboard; however, it is crucial that the driver's concentration remains on the road. Head-up displays (HUDs) can prevent the distractions caused by using mobile phones or onboard displays; nonetheless, drivers cannot easily control the graphic user interface (GUI) on an HUD [1]. To solve this problem, a driver's gestures can be recognized by a wearable device and sent directly to the HUD installed on the dashboard or the front of the instrument panel. Due to the increasing demand for convenient and useful services, the HUD and wearable device are typically connected wirelessly through Wi-Fi or Bluetooth.

The human-computer interaction technique [2] is a significant tool allowing many products to adopt human gesture recognition technology as a natural human interface with common machines [3-4]. The most intuitive and simple form of this technology employs hand gestures. Many gesture recognition algorithms have been developed and studied in recent years. The proposed applications of gesture recognition include intelligent wheelchairs [5], interactive presentation systems [6], automatic user state recognition for television control systems [7], and robot-assisted living [8]. There are two main types of hand gesture recognition: vision-based [9-12] and inertial sensor-based [13-15]. The former is used for 3D recognition in large systems such as image processing and the latter is commonly used to estimate real-time gait cycles and recognize hand gestures [16].

Most approaches require expensive systems and, if high-quality processors are applied, complex algorithms to recognize the gestures. Therefore, in this study, we propose an HUD that is of low cost and has only the minimum computing specifications and network interface. Moreover, gesture recognition systems require that gesture recognition operations are performed within a wearable device and only the results of the user's action are exchanged. Therefore, we propose a gesture recognition algorithm using only the sensor data within wearable devices, which can be applied to wearable devices with minimum computing power. This paper is organized as follows. Section 2 presents the system overview between the HUD and wearable device, including the features of the sensor data and gesture groups, and describes the gesture recognition method based on the sensor features and the recognition algorithm. Subsequently, Section 3 represents the result of experiments used to verify the proposed gesture recognition algorithm. The conclusions are presented in Section 4.

## 2. Material and Methods

### 2.1 System Configuration

The configuration of the HUD system to which the proposed gesture recognition method is applied is as follows. The HUD is connected to the driver's wearable device by Bluetooth communications and the driver's gesture information is transmitted to the HUD. Fig. 1 shows the system configuration and presents the function for changing the GUI of the HUD according to the driver's gesture. As shown in the figure, the HUD employs map information to indicate the direction of operation, current vehicle conditions, and safety operation assistance functions associated with the dash camera. Fast, real-time processing of data is required to express various functions in low-cost HUD products, enabling accurate information to be delivered to users through quick gesture recognition responses. Because of

the low computational power of HUD, separate processing for gesture recognition cannot be allocated; therefore, our method is designed to complete gesture recognition processing within wearable devices and pass the completed gesture values to the HUD. In other words, the algorithms designed in this study deliver minimal data to enable rapid gesture recognition.

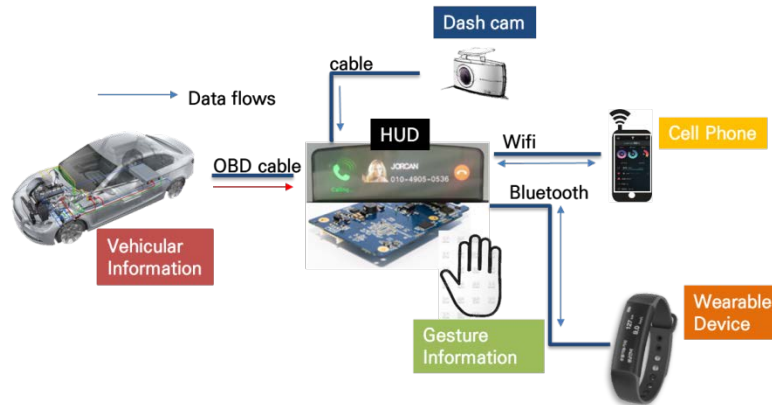
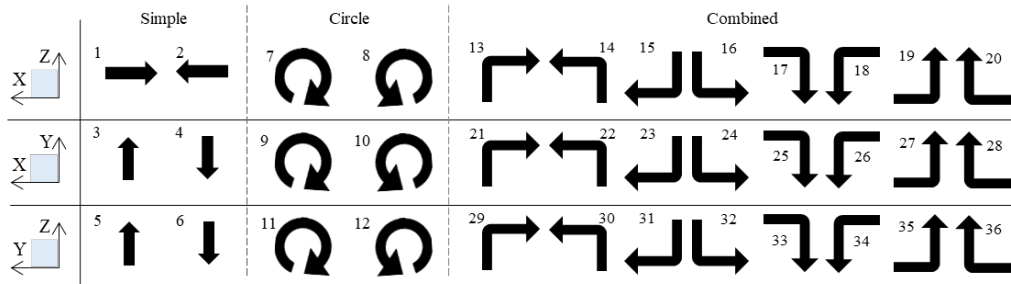


Fig. 1. HUD system configuration

## 2.2 Gesture Description

Because of the various advantages of accelerometers, such as their good mobility, low latency, and low cost, inertial sensor-based hand gesture recognition is used in wearable devices. Existing gesture recognition approaches of the inertial sensor-based system display template-matching [17], dictionary lookup [18], statistical matching [19], linguistic matching [20], and neural networks [21]. For sequential data such as the measurement of time series and acoustic features at successive time frames in speech recognition, the hidden Markov model is one of the most important models [22], as it effectively recognizes patterns exhibiting both spatial and temporal variation [23]. Some research has applied machine learning into gesture recognition using IMU sensors [24-28]. For example, the authors of [25] proposed a method of acceleration data sequence-based dynamic hand gesture recognition. They employed a long-short term memory recurrent neural network and focused on the energy consumption of a small wearable device. However, previous research that recognized gestures using IMU sensors involved limited gesture numbers; e.g., 8, 7, and 12 in [15], [7], and [14], respectively. Moreover, they did not classify similar gestures that have valid peak action at the same axes. Conversely, our proposed method can classify similar gestures through a comparison of pre-defined gestures; thus, it enables rapid recognition and simple processing of sensing data. For example, in Fig. 2, our method can differentiate between gestures #7 and #8, which have the same peak event on the X- and Z-axes. It can also distinguish gesture #7 from #13, #16, #18, and #19. In this study, we propose the sliding correlation, gap-deviation, and inter-correlation to recognize user gestures. These schemes operate according to three groups: simple, circle, and combined, which include 6, 6, and 24 gestures, respectively in Fig. 2.



**Fig. 2.** Gesture set scheme of operation according to three groups: simple, circle, and combined. Each row displays the gesture behavior of the three groups based on the effective plane shown on the left

To test the gesture recognition procedure, we implement the proposed scheme within a wearable device testing platform equipped with an IMU sensor. The generated accelerations and gyroscopes are sampled as 20 Hz. Therefore, the collectable maximum and minimum values are in the range of 32767 to -32768 as raw data, respectively. We used the maximum and minimum values during the pre-processing procedure for normalization. Once the motion is caught, the sensor data detected in one second is analyzed in order to recognize the gesture. Moreover, the sensing device is moved in the horizontal plane when performing gestures to ensure reliable recognition.

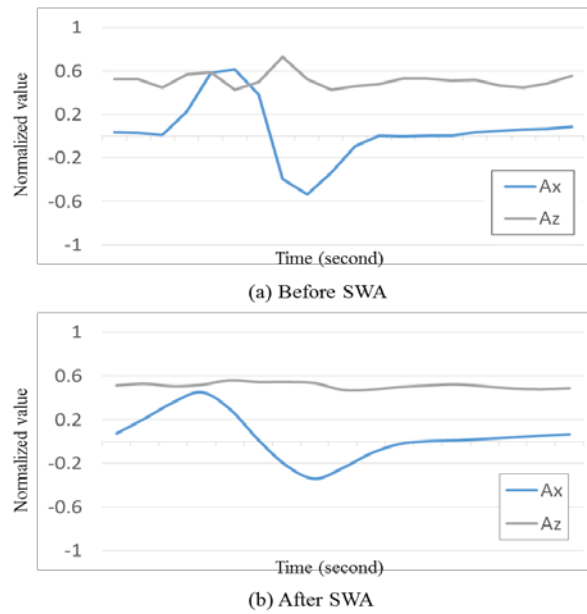
### 2.3 Gesture Configuration and Segmentation

To improve the usability of the gesture recognition procedure, we defined two rules. The first is that every defined gesture finishes within 1s and the second is that every single gesture is started in the normal state to ensure an effective start point for initializing gesture recognition. The procedure for holding the start point is as follows. The capture process observes the magnitude of the sensor data and chooses a sample that is larger than the effective threshold. After capturing the sensor data set, the samples are calculated using the correlation method. There are 36 gestures that can be expressed using this method. For pre-defined data, gestures can be categorized, allowing more gesture actions to be expressed than the gesture recognition method using conventional sensors; this minimizes the errors in similar actions. In particular, partial classification according to the relevant gap value of gesture actions provides users with the advantage of rapidly recognizing gestures. In our system, the gestures are divided into three groups, i.e., the simple gesture, the circle gesture, and the combined gesture. The six simple gestures, the six circle gestures, and the 24 combined gestures are shown in [Fig. 2](#). Through classification and gesture decomposition, the recognition algorithm is simplified. For example, gesture #13 (up-right) can be sequentially composed into #5 (up) and #1 (right). To classify two separate simple gestures and one combined gesture, the factors comprising the combined gesture are sequentially generated within 200 ms. Additionally, all gestures have to maintain a time gap of at least 1 s between themselves when the gestures are generated consecutively.

### 2.4 Data Acquisition and Pre-processing

In order to capture the start point of a gesture, a specific process is required. Because the IMU sensor continuously generates motion data, determining the start point of a motion is important from a resource use aspect. Due to their extremely minimal resources, small sized devices such

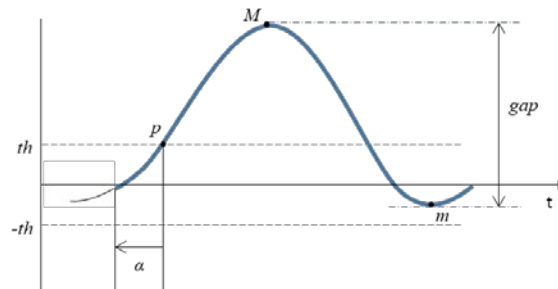
as wearable devices have to require a trigger mechanism. However, using raw data to detect the start point is not easy because the IMU sensor is sensitive to very small movements. Therefore, in the pre-processing stage, we propose the normalization of the raw data by its maximum value (32768). Using normalized data ensures that the system is robust against very small levels of noise and improves the functionality of the gesture recognition. Then, the normalized data are smoothed using a sliding window average (SWA) to cancel noise. **Fig. 3** represents the smoothing and noise-reducing effect of the SWA using an actual gesture data value, where **Fig. 3(a)** shows the normalized data set of gesture #1 and **Fig. 3(b)** shows the smoothed data set.



**Fig. 3.** Effect of the sliding window average filter on gesture #1: (a) before processing by the SWA and (b) after processing by the SWA

After SWA processing, the gesture recognition process is initiated. The start point of the appropriate data set is decided by a pre-defined threshold. We set the threshold of movement to 20% higher/lower than the normal freezing state. The system includes previous  $\alpha$  samples into the recognition data set in **Fig. 4**. To collect the entire data set, the system preserves sensor data constantly in the normal state. To exclude meaningless data, the system checks the gap between the maximum and minimum values. If the gap, represented by equation (1), is larger than threshold  $\beta$ , the gesture recognition module is continually operated. The concept of  $\alpha$  and threshold  $\beta$  is represented in **Fig. 4**, which illustrates that previous  $\alpha$  sample values are collected from the detected point,  $p$ . Therefore, the start point of the acquired data set is  $p-\alpha$  and the gap between the maximum point,  $M$ , and the minimum point,  $m$ , is represented. The gap comparison has the potential for error induced by instantaneous large noise; i.e., a false positive. However, as our system uses the SWA during pre-processing, the probability of a false positive is low

$$gap = \max_{i,j} (|y(i) - y(j)|), i \neq j \quad (1)$$



**Fig. 4.** Threshold detection where the y-axis represents the normalized value

## 2.5 Gesture Classification

After pre-processing, the proposed system compiles groups of expected gestures using the gap value. The acquired data set is assigned to the relevant group using the effective axis in [Table 1](#). For instance, if the system detects a valid value only at the x-axis, the acquired data set has a high probability of being gesture #1 or #2 of the simple group. Simply, the obtained data set is a simple gesture when a single axis acquires the relevant gap and a circle or combined gesture otherwise. However, when all axes have large gap values, the system selects the largest two axes among the relevant axes. The proposed system compares the pattern correlations between acquired data and the database. Because only the relevant axes are assigned during the comparison stage, we added weight to distinguish axes according to their pre-defined group. We set the weighted sum of the axes whose gaps are larger than  $\gamma$  to 90, and whose sum of axes is not equal to 10. To match specific gesture from aligned gesture group, we proposed sliding correlation and gap-deviation method that are described in this section. Dividing the gesture group, the gesture recognition process is relatively easy due to fewer candidates in belong group. If the system doesn't classify the group, it is takes long and hard to match and recognition with specific gesture. For example, when a valid signal is acquired at the AX and AY axis, the weighted result of X and Y is set to 45%. The remaining values of AZ, GX, GY, and GZ are then set to 2.5%, where AX is the acceleration of the x-axis and GX is the gyro value of the x-axis. When the three axes have a value that is greater than the threshold, the system selects the largest two axes among the relevant axes for classification.

**Table 1.** Group and relevant axes of all gestures analyzed in this study

Relevant axes	Group	Gesture #
X	simple	1, 2
Y		3, 4
Z		5, 6
X, Y	combined or circle	9,10,21~28
X, Z		7,8,13~20
Y, Z		11,12,29~36
X, Y, Z	combined or circle	7~36

## 2.6 Gesture Recognition

A flow chart of the proposed gesture recognition method is shown in Fig. 5. To match the gestures to target gestures from the aligned gesture group, we propose a sliding correlation and gap-deviation method. By dividing the gesture group, we ensure that the gesture recognition process is relatively simple due to fewer candidates in each group. If the system does not classify the group, it is time-consuming and difficult to match the gesture with the target gesture. For an acquired data set matched with stored gesture pattern data, the patterns must be synced with each other. To match the formulation, we adopted a sliding correlation method. The concept of sliding correlation is similar to that of SWA. The first several samples are shifted and removed to match with stored gesture data at the shifting stage. The system then determines the maximum correlation value produced in every sliding stage, which is compared with stored gesture data. The goal of this operation is similar to that of dynamic time warping (DTW) in [15].

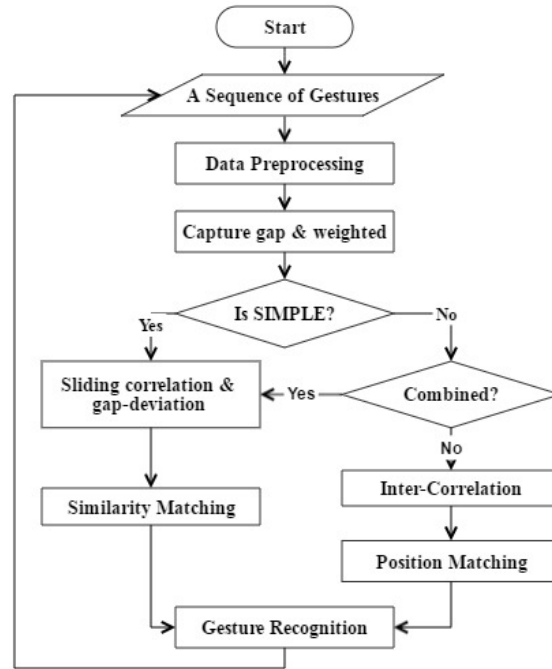


Fig. 5. Flow chart of the proposed method

The method compares the classified data set with stored gesture data that have been assorted into the same group using the sliding correlation. We assume that  $\mathcal{Y}$  is the pre-processed data set,  $\mathcal{Y} = \{y(1), y(2), \dots, y(N_S)\}$  and the  $\mathcal{Y}'$  subset of  $\mathcal{Y}$ ,  $\mathcal{Y}'_{a,b}$  means a subset of consecutively arranged data containing the first number,  $a$ , to the last number,  $b$ . For example,  $\mathcal{Y}'_{a,b}$  could be expressed as  $\mathcal{Y}'_{1,2} = \{y(1), y(2)\}$ ,  $\mathcal{Y}'_{1,3} = \{y(1), y(2), y(3)\}$ , ...,  $\mathcal{Y}'_{1,N_S} = \{y(1), y(2), y(3), \dots, y(N_S)\}$ . The sliding correlation set,  $\mathcal{SC}$ , or sync of formulation is represented in equation (2).

$$\mathcal{SC} = \{\rho(\mathcal{Y}'_{1,N_S-f_C}, \mathcal{DB}'_{f_C,N_S}), \rho(\mathcal{Y}'_{1,N_S-f_C+1}, \mathcal{DB}'_{f_C+1,N_S}), \dots, \rho(\mathcal{Y}'_{1,N_S}, \mathcal{DB}'_{1,N_S}), \dots, \rho(\mathcal{Y}'_{f_C,N_S}, \mathcal{DB}'_{1,N_S-f_C})\} \quad (2)$$

where  $\mathcal{DB}'$  is the subset of  $\mathcal{DB}$ . Its data are assorted into the same group,  $\mathcal{DB}'_{a,b}$ , which is the same role of  $\mathcal{Y}'_{a,b}$ .  $f_C$  is a correlation factor determining how many elements overlap and which indicate the count number of sliding.  $\rho(a, b)$  is expressed by  $\rho_{ab}$  in equation (3).

$$\rho(a, b) = \rho_{ab} = \frac{\sigma_{ab}}{\sigma_a \sigma_b} \quad (3)$$

There is the potential for abnormally high  $\mathcal{SC}$  due to a temporal match caused by user motion or the influence of prior motion. To ensure a match with the correct gesture, the typical formulation feature has to be compared with the required axes. Therefore, we apply a deviation concept between the two formulations. The result of  $\mathcal{SC}$  is divided by the deviation of the difference between the acquired data set in the pre-defined database.  $\mathcal{SC}_k$  denotes a  $k$ th element of  $\mathcal{SC}$ .

$$da = \frac{1}{N_S} \sum_{l=1}^{N_S} |\mathcal{Y}_l - \mathcal{DB}_l| \quad (4)$$

and,

$$dv = \sum_{l=1}^{N_S} |\mathcal{Y}_l - \mathcal{DB}_l - da| \quad (5)$$

Then, if  $\mathcal{SC}_k > 0$ ,

$$match = \arg \max_{\mathcal{DB}_k} \left( \frac{\mathcal{SC}_k^2}{dv} \right) \quad (6)$$

As a result, the acquired data is matched to the pre-defined gesture with the highest similarity and the lowest difference in the formulation trend.

### 3. Test Results

#### 3.1 Test Environment

In this section, we verify the effectiveness of the proposed method for hand gesture recognition using a wearable band. We also implement the transmit driver's gesture from the wearable band to the HUD. We applied the previously described system environment to our



proposed gesture recognition method. The wearable band operates a 32-MHz clock and is connected via Bluetooth in Fig. 6. The IMU sensor is a six-axis sensor unit from InvenSense. The six-axis data of ICM-20948 is stored as 16-bit data. The accelerometer measures the analog acceleration signals generated by a user's hand movements then converts the signals into digital ones via the internal 16-bit A/D converter. The acceleration signals from the accelerometer are processed through the I2C interface and the gesture recognition algorithm is run simultaneously. The recognized gestures are then transmitted to the HUD as GUI commands via Bluetooth.

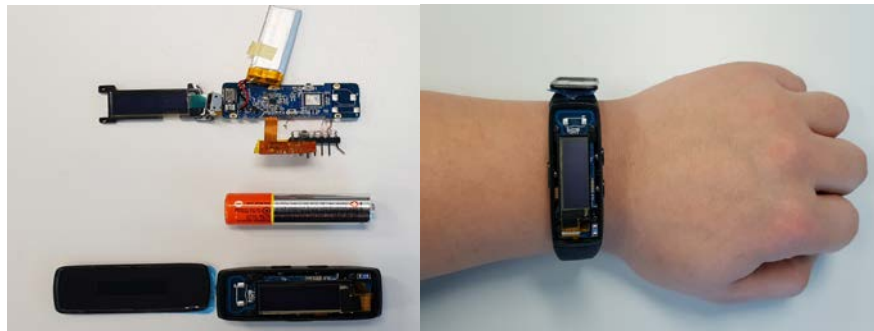


Fig. 6. Wearable band: (a) Components of the wearable band; (b) wearing condition

### 3.2 Experimental results

The implementation results are based on three subjects, with a total of 1,680 simple and circle gestures and 3,360 combined gestures collected in a week. Each test was executed 40–50 times for each target gesture where 10 men and 8 women executed the defined experiments. The subjects performed gestures with a horizontally equipped wearable band on their right wrist, as shown in Fig. 7(a) and (b). When a motion change occurred, the GUI of the HUD changed instantly according to the gesture recognition process. Table 2 summarizes the accuracy of group matching for each type of input gesture; i.e., if the input gestures of a specific group were classified in the same gesture group. For example, an incoming simple gesture was recognized as a simple gesture with an overall accuracy of 99.7%. The group matching accuracy of the circle and combined was 97.4% and 98.6%, respectively; thus, our classification processing procedure was validated. During the experiments, some input gestures were not detected at all. This was because the acceleration of the hand motions was too small to reach the threshold. The missing gestures were not considered when determining the recognition accuracy because they did not go through the recognition procedure. Table 3 shows a comparison of recognition performance between our proposed algorithm and three existing methods: Sign Sequence and Template Matching (SSTM) [3], DTW [15], Gesture Decomposition and Similarity Matching (SM) [4] and Machine Learning with Neuron memory [29]. Note that, although the recognition accuracy of our proposed algorithm was slightly lower than that of DTW, our system incorporates substantially more gestures; four times that of DTW. Moreover, some of the pre-defined gestures used in the recognition have similar features, so our proposed method can simply and rapidly classify gestures. Also, our proposed algorithm exhibited higher recognition accuracy than the similar SM method. Especially, we additionally have been implementing and testing the ML gesture recognition with neuron memory for comparison. The neuron memory can be stored before learned 12 gestures. Then we tested the gesture recognition 30 times per each gestures. As following the

presented algorithms in the left column, other columns of **Table 2** represent the number of gesture, average accuracy described in corresponding papers, and the overall number of test sample, respectively. Note that, although the recognition accuracy of our proposed algorithm is slightly higher than DTW. Also, some of pre-learned gestures using neuron memories which we have used to match the similar features is slightly higher but, our proposed method can be simple and low cost device as the view of implementation.

**Table 2.** Gesture group matching accuracy (%)

		Recognized gesture			Number of test samples
		simple	circle	combined	
Input gesture	simple	99.7	0.3	0.0	840
	circle	0.1	97.4	2.5	840
	combined	0.1	1.3	98.6	3,360

**Table 3.** Performance comparison between different gestures recognition methods

Algorithms	Number of gestures	Accuracy (%)	Number of test samples
DTW [13]	8	98.6	540
SSTM [3]	7	95.6	628
SM [4]	12	97.2	1,134
ML [29]	12	100	360
Proposed method	36	99.7	5,040



**Fig. 7.** Implementation and test environment for the gesture recognition method:  
(a) system set up; (b) gesture recognition test



**Fig. 8.** Real test environment for the gesture recognition method

## 4. Conclusions

In this study, we proposed a new hand gesture recognition method that is compatible with an HUD system for use in vehicle navigation. For fast link adaptation and reduced processing during gesture recognition, the required algorithm was designed to be simple and require minimal resources. Our simple yet effective method recognizes hand gestures with an IMU sensor based on the correlation matching technique. The system employed sliding correlation, gap-deviation, and inter-correlation for three gesture groups: simple, combined, and circle gestures. If the acquired data set was classified as simple or combined, the system operated sliding correlation and gap-deviation. For circle gestures, the system operated inter-correlation. To confirm the effectiveness of the system, we tested the proposed algorithm using pre-defined gestures of specific motions with a wearable platform device. The experimental

results validated the feasibility and effectiveness of the proposed hand gesture recognition system. Despite being based on a very simple concept, the proposed algorithm showed good performance in recognition accuracy. However, the algorithm is very sensitive to fluctuations in data trends; therefore, the correlation concept adjusted in the proposed algorithm is likely to produce false positives. For instance, various trends of user movement include velocity, direction, gradient, and device noise. Because the correlation between stored databases is key for the recognition, it would be difficult to employ the proposed algorithm into a commercial product for the public. Therefore, future research will further develop the optimizing recognition method and determine the most effective optimization factors for improving algorithm performance. Especially, circle and combined gesture recognition working in a driving environment is needed to be more rapid recognition since they are required to analyze exactly up, down, right and left.

## References

- [1] C. Yoon, K. Kim, S. B. and S. Y. Park, "Development of Augmented In-Vehicle Navigation System for Head-Up Display," in *Proc. of IEEE International conference ICT convergence*, pp.601-602, 2014. [Article\(CrossRefLink\)](#)
- [2] Min Yuan, Heng Yao, Chuan Qin and Ying Tiann, "A dynamic hand gesture recognition system incorporating orientation based linear extrapolation predictor and velocity assisted longest common subsequence algorithm," *KSII Transactions on internet and information systems*, vol. 11, no.9, pp.4491- 4509, 2017. [Article\(CrossRefLink\)](#)
- [3] R. Xu, S. Zhou and W. J. Li., "MEMS accelerometer based nonspecific user hand gesture recognition," *IEEE Sensors Journal*, vol. 12, no. 5, pp.1166-1173, 2012. [Article\(CrossRefLink\)](#)
- [4] R. Xie, X. Sun, X. Xia and J. Cao., "Matching-Based Extensible Hand Gesture Recognition," *IEEE Sensors Journal*, vol. 15, no. 6, pp.3475-3483, 2015. [Article\(CrossRefLink\)](#)
- [5] T. Lu., "A motion control method of intelligent wheelchair based on hand gesture recognition," in *Proc. of IEEE international conference industrial and electronics applications*, pp.957-962, 2013. [Article\(CrossRefLink\)](#)
- [6] B. Zeng, G. Wang and X. Lin., "A hand gesture based interactive presentation system utilizing heterogeneous cameras," *Tsinghua Science Technology*, vol. 17, no. 3, pp.329-336, 2012. [Article\(CrossRefLink\)](#)
- [7] S. Lian, W. Hu, K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system," *IEEE Transaction on consumer electronics*, vol. 60, no. 1, pp.107-115, 2014. [Article\(CrossRefLink\)](#)
- [8] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Transaction on System and Humans*, vol. 41, no. 3, pp.569-573, 2011. [Article\(CrossRefLink\)](#)
- [9] Yong-Suk Park, Se-Ho Park, Tae-Gon Kim and Jong-Moon Chung, "Implementation of Gesture Interface for Projected Surfaces," *KSII Transactions on internet and information systems*, vol. 9, no.1, pp.378- 290, 2015. [Article\(CrossRefLink\)](#)
- [10] Doyeob Lee, Dongkyoo Shin and Dognil Shin, "Real-Time Recognition Method of Counting Fingers for Natural User Interface," *KSII Transactions on internet and information systems*, vol. 10, no.5, pp.2363- 2373, 2016. [Article\(CrossRefLink\)](#)
- [11] D. Avola, L. Cinque, G. L. Foresti, and M. R. Marini, "An interactive and low-cost full body rehabilitation framework based on 3D immersive serious games," *Journal of Biomedical Informatics*, vol. 81, pp. 81-100, 2019. [Article\(CrossRefLink\)](#)
- [12] L. E. Sucar, R. Luis, R. Leder, J. Hernandez and I. Sanchez, "Gesture therapy: A vision-based system for upper extremity stroke rehabilitation," in *Proc. of IEEE International Conference Engineering in Medicine and Biology*, pp.3690-3693, 2010. [Article\(CrossRefLink\)](#)

- [13] D. Avola, L. Cinque, G. L. Foresti, M. R. Marini, D. Pannone, "VRheab: a fully immersive motor rehabilitation system based on recurrent neural network," *Multimedia Tools Applications*, vol. 77, pp. 24955-24982, 2018. [Article\(CrossRefLink\)](#)
- [14] J. Alon, V. Athitsos, Q. Yuan and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Transaction on pattern analysis machine intelligence*, vol. 31, no. 9, pp.1685-1699, 2009. [Article\(CrossRefLink\)](#)
- [15] J. K. Oh, "Inertial sensor based recognition of 3-D character gestures with an ensemble classifier," in *Proc. of 9th International Workshop Frontiers Handwriting Recognition*, pp.112-117, 2004. [Article\(CrossRefLink\)](#)
- [16] S. Zhou, Z. Dong, W. J. Li and C. P. Kwong, "Hand-written character recognition using MEMS motion sensing technology," in *Proc. of IEEE/ASME International Conference Intelligent mechatronics*, pp.1418-1423, 2008. [Article\(CrossRefLink\)](#)
- [17] A. Akl, C. Feng and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Transaction on signal Processing*, vol. 59, no. 12, pp.6197-6205, 2011. [Article\(CrossRefLink\)](#)
- [18] C. C. Yang, Y. L. Hsu, K. S. Shih and J. M. Lu, "Real-Time Gait Cycle Parameter Recognition Using a Wearable Accelerometry System," *IEEE Sensors Journal*, pp.7314-7326, 2011. [Article\(CrossRefLink\)](#)
- [19] J. S. Lipscomb, "A trainable gesture recognizer," *Pattern and Recognize*, vol. 24, no. 9, pp. 895-907, 1991. [Article\(CrossRefLink\)](#)
- [20] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*, McGraw-Hill: New York, 1979. [Article\(CrossRefLink\)](#)
- [21] D. H. Rubine, *The Automatic Recognition of Gesture*, Ph.D dissertation, Computer Science Department, Carnegie Mellon Univ., Pittsburgh, Dec. 1991. [Article\(CrossRefLink\)](#)
- [22] K. S. Fu, *Syntactic Recognition in Character Recognition*, Academic, New York, 1974. [Article\(CrossRefLink\)](#)
- [23] S. S. Fels and G. E. Hinton, "Glove-talk: A neural network interface between a data glove and a speech synthesizer," *IEEE Transaction on Neural Network*, vol. 4, no. 1, pp.2-8, 1993. [Article\(CrossRefLink\)](#)
- [24] C. M. Bishop, *Pattern Recognition and Machine Learning, 1st edition*, Springer, New York, 2006. [Article\(CrossRefLink\)](#)
- [25] T. Schlomer, B. Poppinga, N. Henze and S. Boll, "Gesture recognition with a Wii controller," in *Proc. of 2nd International Conference Tangible and Embedded Interaction*, pp.11-14, 2008. [Article\(CrossRefLink\)](#)
- [26] G. Costante, L. Porzi, O. Lanz, P. Valigi and E. Ricci, "Personalizing a smartwatch-based gesture interface with transfer learning," in *Proc. of Signal Processing Conference EUSIPCO*, pp.2530-2534, 2014. [Article\(CrossRefLink\)](#)
- [27] S. Shin and W. Sung, "Dynamic Hand Gesture Recognition for Wearable Devices with Low Complexity Recurrent Neural Networks," in *Proc. of IEEE International symposium on circuits and systems*, pp. 2274-2277, 2016. [Article\(CrossRefLink\)](#)
- [28] G. Devineau, F. Moutarde, W. Xi and J. Yang, "Deep Learning for Hand Gesture Recognition on Skeletal Data," in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition Proceedings*, pp.106-113, 2018. [Article\(CrossRefLink\)](#)
- [29] Niclas Gyllsdorff, *Distributed machine learning for embedded devices*, Ph. D dissertation, Teknisk, UPPSALA Univ., Sweden, 2018. [Article\(CrossRefLink\)](#)





**Sangyub Lee** received BS degree and the MS degree in electronic and electrical engineering from the Yonsei University, South Korea, in 2005. He received the PhD degree in Computer and Information Science from Korea University, South Korea, in 2019. He is a principal researcher at the Korea Electronics Technology Institute. He was with Samsung Electro-Mechanics as a senior engineer at the Central R&D center. His research focuses on automotive systems and various embedded platform systems.



**Jaekyu Lee** received BS degree and the MS degree in electronic and electrical engineering from the Chonbuk National University, South Korea, in 2012. He is a candidate of the PhD degree in Computer and Information Science from Korea University. He is a researcher at the Korea Electronics Technology Institute. His research focuses on automotive software and various embedded platform systems.



**Hyeonjoong Cho** received BS degree in electronic engineering from Kyungpook National University in 1996 and the MS degree in electronic and electrical engineering from the Pohang University of Science and Technology, South Korea, in 1988. He received the PhD degree in computer engineering from Virginia Polytechnic Institute and State University, Virginia Tech, in 2006. He is an associate professor in the Department of Computer and Information Science, Korea University. He is a member of the IEEE. His research focuses on real-time systems on various platforms including signal/multiprocessor, sensor networks, etc.