Int. J. Advance Soft Compu. Appl, Vol. 9, No. 1, March 2017
ISSN 2074-8523

Real Time Advanced Head Movement Recognition for Application Controller Based On Android Internal Gyroscope Sensor

Nurizal Dwi Priandani[1], Herman Tolle[2], and Fitri Utaminingrum[3]

[1,2,3] Department of Computer Science,
Faculty of Computer Science – Brawijaya University
e-mail: priandani@ub.ac.id, emang@ub.ac.id, f3_ningrum@ub.ac.id

Abstract

This research proposes a head movement recognition method based on Android internal Gyroscope sensor. The contribution of this research is to proposes head movement as a new way of Human-Computer Interaction approach on Android smartphone. The head movement which has been used was an advanced movement from the single movement to the combination movement. Another challenge was head identification process based on built-in Android gyroscope sensor, which the sensor in the Android produced data noisier than the sensor in the iOS. Based on that, a preprocessing method has been proposed to enhance the head movement recognition. After the preprocessing phase, the head movement recognition phase conducted by the Nearest-Neighbour Classification Algorithm. Based on the proposed method, a controller application prototype has been made and deployed into the smartphone which combined with the dummy HMD to be used on the head. The experiment has been conducted on different hardware environments to understand the robustness of the proposed method. Based on the accuracy experiment, the success rate of our proposed method was about 95%. Based on the performance experiment, the average of the time amounts to execute the identification process was about 2.5 ms, which less than 100ms. Based on the experiment result, our proposed method is appropriate for a new experience of real-time human-computer interaction with head movement control only, although in a different hardware environment.

Keywords: *Human-Computer Interaction, Head Movement, Controller, Pattern Recognition, Gyroscope Sensor, Android Mobile.*

1 Introduction

Topics of Human-Computer Interaction (HCI) has been a hot issue for research on a few years back [1]. Currently, the most common approach to HCI is based on a mouse, keyboard and touch screen. Some studies also suggested a new approach based on HCI hand gestures. HCI by using the hand as a base input is not effective if the user has a disorder or problem with his hand. For example, a patient who suffered a broken hand, has a stroke disease (Cerebrovascular Accident) or disability person will be difficult to use hands to operates the device, especially mobile device. So, we need another scheme that can deal with other matters that use other body parts, such as the head, as the HCI approach.

There are several HCI based on activity recognition approaches currently developed and researched, vision-based and sensor-based approach [2]. A vision-based approach is an approach which using the image and then processed in a way which has meaning. For examples of the use of HCI with the vision-based approach is the Smart Room [3]. A sensor-based approach is an approach which uses sensors, such as an accelerometer and a gyroscope as the input. Sensor-based HCI approach has been applied on the Nintendo Wii Remote [4].

Currently, the smartphone was equipped with innovative sensors. The smartphone feature will be more varied when equipped with the sensors. Most Android devices have built-in sensors which measure motion, orientation, and various environmental conditions [5]. Some examples of sensors which built-in to the Android device are proximity, gyroscope, and accelerometer sensor.

In this paper, we propose a method of advanced head movement recognition as smartphone application controller for HCI purpose based on Android gyroscope sensor. The contribution of this study is the use of head movement as a new HCI methods on Android smartphones. Other contributions are the utilization of head movement as an advance movement, from the single movement to combination movement. The next challenge is the head movement recognition using gyroscope sensors that built-in to the Android-based devices. As described in [6] and [8] that the recognition of a movement using a sensor on the Android yield results is not better than sensors on iOS. Based on this discourse, we propose a new recognition method at preprocessing and feature extraction stage. The new method is expected to be able to produce a direct recognition process to Android smartphone device, which reliable, light, and fast, but with minimum training data.

2 Related Work

Many researches have been done on the topic of an activity or head movement recognition as the Human-Computer Interaction. Some previous research has been proposed methods of head movement recognition based on accelerometer sensor, that are [9], [10] and [11]. The research was limited to head movement recognition method only, and there was still no application. The research [12], has discussed the use of head movement as a mouse replacement using the stand-alone sensor. However, research [6] proposed the combination of the user's head and body movement as a controller for virtual reality labyrinth game. Control Mobile Application through Head Pose Movement Detection with CoreMotion feature in the iOS has been conducted on [7]. This paper presents single movement as the only recognized / detected head movement.

Researchers have proposed some studies related to gesture or activity recognition. The study [13], discussed the inertia-based gesture recognition. The accelerometer and gyroscope sensor was used. This study shows that the gyroscope sensor is suitable for rotational movement. The research [14], discussed the hand gesture recognition which the proposed method was a high performance and training-free method. This study revealed a problem about the number of training data which affected to the computation time and memory consumption so that it took the method to clear the data as a preprocessing phase. The research [15] examined the gesture recognition based on accelerometer using distance metric learning classification algorithm, Nearest-Neighbour. The distance metric used was Mahalanobis distance. This algorithm is known for its light but requires a large number of training data to generate a high accuracy.

Based on research [13], [14], and [15] which has been describing above, it can be concluded that the process of pattern recognition using a classification approach depends on the amount of training data. Larger training data is better to improve the accuracy but takes higher cost and high memory processes. On the other hand, the mobile application development requires light data processing and less memory consumption. The preprocessing phase is crucial. Data cleaning and identifying important features which used can be a solution for a simple classification method by using a small number of training data. Although the data is inconsiderable, if the data is clean then the application will be a real-time and lightweight.

3 Head Movement

The human neck can perform six movements. Neck movements are called as the head movements because the neck is an activator of the head, and the head is the object moved by the neck. The basic movements of the human head are [16]:

1. Flexion (*downward*)

   This movement is commonly called nodding head motion downwards with the chin closer to the chest.

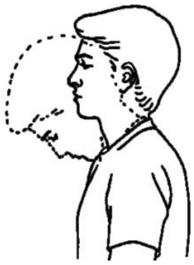

Fig. 1: Flexion Head Movement [16]

2. Extension (upward)

   This head movement is commonly referred look up or seeing an object which positioned up above.

Fig. 2: Extension Head Movement [16]

3. Lateral Rotation to the left (leftward)

   This head movement is commonly referred as look left which means head is turning to the left.

4. Lateral Rotation to the right (rightward)

   This head movement is commonly referred as look right which means head is turning to the right.

Fig. 3: Lateral Rotation to the right [16]

5. Lateral Flexion Left (Tilt Left)

   Head movement is commonly called a tilt left that the head movement with the left ear closer to the shoulder.

6. Lateral Flexion Right (Tilt Right)

This head movement is commonly known as a tilt right that the head movement with the ear closer to the right shoulder.

Fig. 4: Lateral Flexion Right and Left [16]

4 Pattern Recognition

Pattern recognition can be interpreted as a data classification based on knowledge which had obtained or on statistical information taken from patterns and/or that representation [17]. On the pattern recognition application, it is a necessity to process the raw data and convert it into a form which can be accepted by the machine for the next practice [17]. The process of pattern recognition can be classified into two forms, the classification, and clustering [18]. In the classification, the label of the pattern which specified into classes based on abstractions which produced by using a set of training patterns or domain of knowledge [18]. Clustering produced data partitions which help in taking a decision [18].

Several paradigms used to solve pattern recognition problems. Two of them are [17]:

1. Statistical pattern recognition
2. Syntactic pattern recognition

The statistical pattern recognition approach is more popular than syntactic pattern recognition and has more attention in the literature. It because most of the practical problems are noisy data and uncertainty. Statistics and probability approach are very powerful. On the other hand, the used of formal language theory for pattern recognition based on linguistic tools such as syntactic is not suitable if used in the noisy data environment [17].

In statistical pattern recognition approach, vector spaces can be used to represent patterns and classes. The abstraction, normally deal with the probability or point distribution on the multi-dimensional of spaces. The representation of vector space can be related to the sub-spaces or projections and similarities between the point. In this case, calculation of the distance [17].

4.1 Nearest-Neighbour Classification Algorithm

To classify a pattern, Nearest-Neighbor Classification is a method with Similarity Based Classification approach. Nearest-Neighbor Classification works by comparing the input pattern to the standard pattern which is given to each other. The classification approach sorts the proximity distance of input pattern which classified by the pattern in the training set which has been determined. An input pattern is classified into a pattern which has the shortest distance to the training set. The easiest way to calculate the similarity between two data in the dataset is to measure the distance using the distance metric [17].

Distance measurement is used to find the difference inter-representation of the pattern. A similar pattern has to be adjacent. To use distance measurements must be ensured that all the features have the same range of values. The examples of distance measurement methods are Manhattan Distance, Euclidean Distance and Minkowski Distance [17].

4.1.1 Manhattan Distance

A distance measurement is used to find the dissimilarities inter-representation of the pattern [17]. The pattern which more similar will have a shorter distance. Distance metric which popularly used is Minkowski metric in equation below [17]:

$$d^m(X,Y) = \left(\sum_{k=1}^{d} |x_k - y_k|^m\right)^{\frac{1}{m}}$$

When *m* is 1, that equation will be Manhattan metric in the form which explained below:

$$d(X,Y) = \sum_{k=1}^{d} |x_k - y_k|$$

The advantage of *Manhattan distance* compared to *Euclidean Distance* is a faster computation time [19].

5 Proposed Method

In this section, the main design and task of head movement recognition based on Android Gyroscope Sensor are explained. The major tasks of the head movement recognition method are; application deployment designing, the gyroscope sensor signal pattern during the head movement analyzing, and designing of the recognition method.

5.1 Application Deployment Design

The proposed method works with a dummy HMD complementary with an Android smartphone which has internal gyroscope sensor. The user wears an HMD with an Android smartphone as shown in Figure 5 while system displays the application that was developed. Dummy HMD was used for integrating human head and eyes through the smartphone's display. Through this scheme, the user could easily move his head to recognized later.

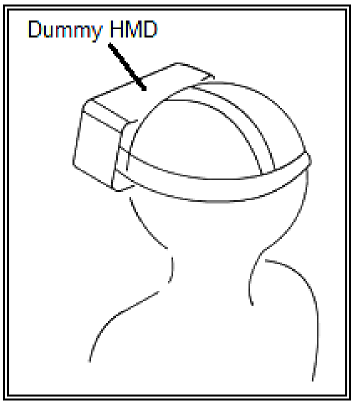

Fig. 5: Application Deployment Design

5.2 Head Movement Gyroscope Sensor Signal Analysis

Before designing a recognition method, analysis of the Android gyroscope sensor signal which produced the head movement has been done. These activities were conducted to determine the form and the signal pattern characteristic which has been produced. In Figure 6 and 7 respectively indicated the data signal pattern or change rate of the point which has been obtained from Gyro-sensors based on *single head movement* and *double head movement*.

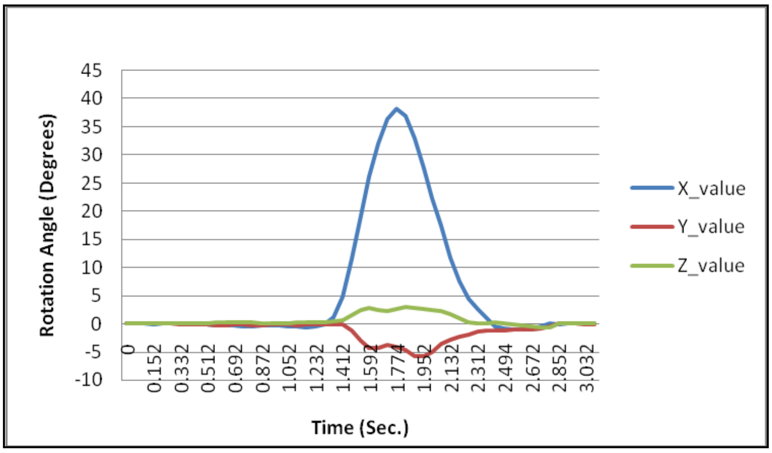

Fig. 6: Gyroscope Signal Pattern During Leftward Head Movement

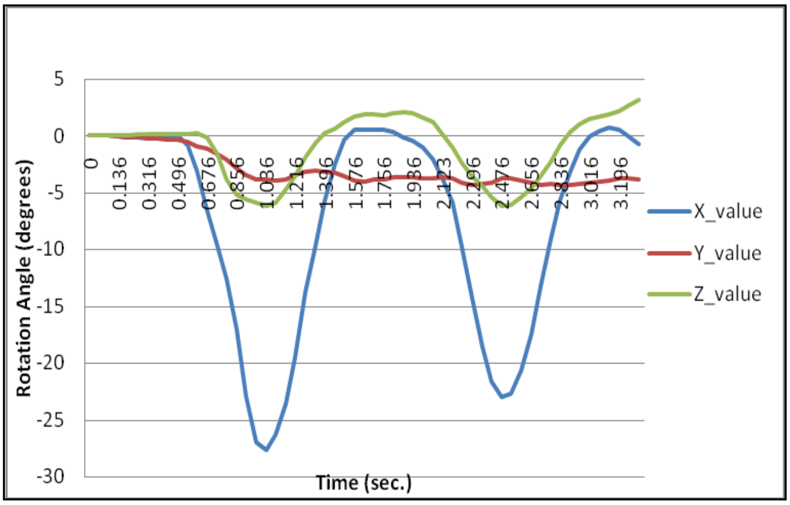

Fig. 7: Gyroscope Signal Pattern During Rightward-Rightward Head Movement

The Figure 6 and 7 shows the head movement signal pattern based on Gyroscope sensor in the Android smartphones. In a single movement, there was an affected certain axis. Meanwhile, in double movement, there was two waveforms which formed during the movement of the head. Every waveform had a time interval which affected by the speed of the user's head movement. There was the time interval between the first movement and the second movement in the waveform which formed on the double movement. There was noise which formed during the head movement in all movements. The noise was a value which appears on unrelated axes.

As illustrated in Figure 6 and 7, the x-axis was the main factor in the rightward and leftward movement. However, there were noises, the y-axis, and z-axis, which consisted a fluctuative value. To ensure the reliability of the application that associated with the recognition accuracy and performance, a scheme was needed to clean up the data from noises.

5.3 Recognition Method

The method in the head movement recognition application based on four repeatable process steps as follows: 1) read Gyroscope sensors data, 2) clean the data, 3) recognize the data/signal pattern, and 4) determine the head movement. The head movement recognition task will be divided into two processes, training, and testing / recognition process. Figure 8 shows the general process and the gyro signal pattern recognition phase of the head movement.

The training process is a process to obtain training data which the process was carried out two times for each movement. Training data is data that directly labeled or classified by users. The labeling was guided by the application by providing an instruction of head movements for the user. Training data serves as a calibration time of the head movement, and to get an angle value point of the head movement alteration. Training phase consists of several processes, such as data acquisition, preprocessing, feature value and calibration time. The outcome of this phase is saved calibration time and features value as a data set.

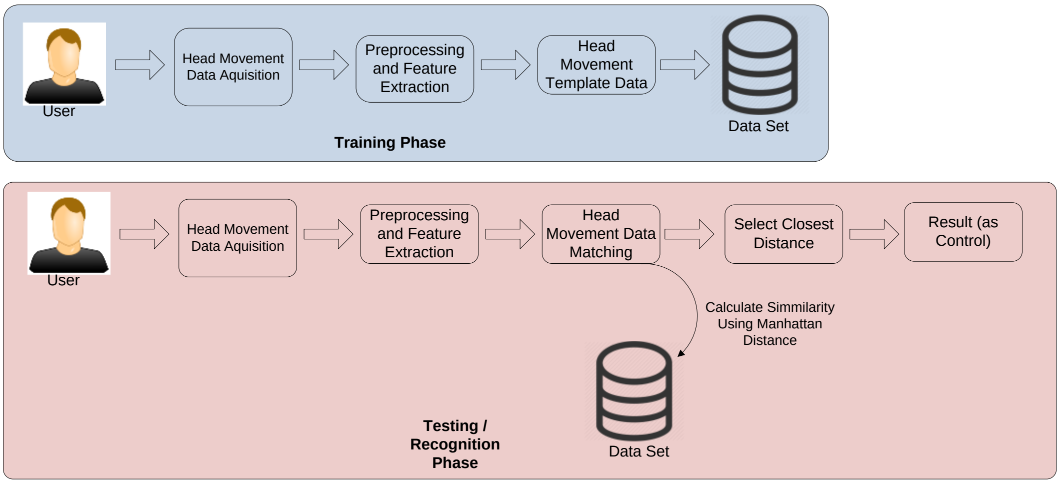

Fig. 8: Head Movement Recognition Design Steps

The testing or recognition phase is a process of movement recognition based on the measurement of the training data. There were several similar phases, the data acquisition, and data preprocessing phase to get features value

5.3.1 Data Acquisition

Data acquisition process is a process of angle value change data recording which obtained from Gyro-sensor by head movement. Technically, data acquisition process worked simultaneously or in parallel with preprocessing and feature extraction which described in the next section.

5.3.2 Preprocessing

Preprocessing was responsible to clean the data from noise. The definition of Noise has been described in the previous section.

5.3.2.1 *Rotation Angle Thresholding*

To eliminate noise, the threshold was given on each axis where it was assumed that the threshold value was the user minimum angle limits to perform a rotational head movement. Threshold value were +10 and -10. After the threshold was given, the first axis which passed the threshold became a major concern. For example, Figure 9 (a) and (b) shows that x-axis passed the threshold value for the first time. After knowing the major axis, the value of the other axis was set to 0. The waveform which has been cleaned from noise was acquired which displayed in Figure 10 (a) and (b).

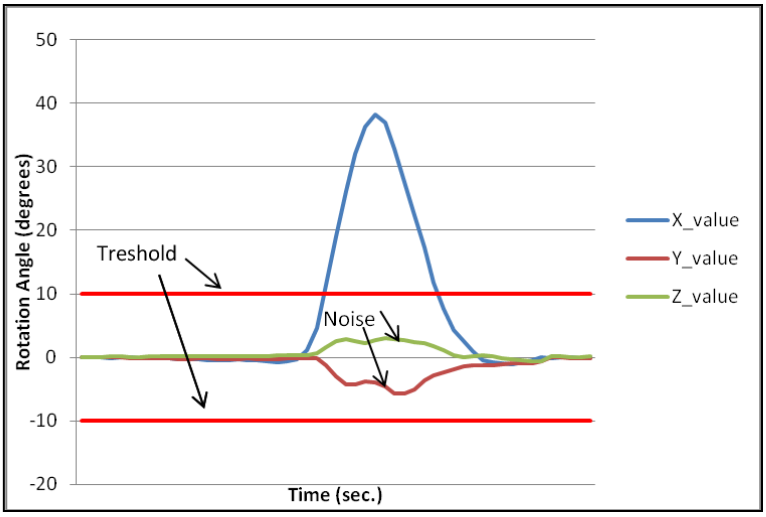

Fig. 9 (a): Giving Threshold on Raw Data of Single Head Movement

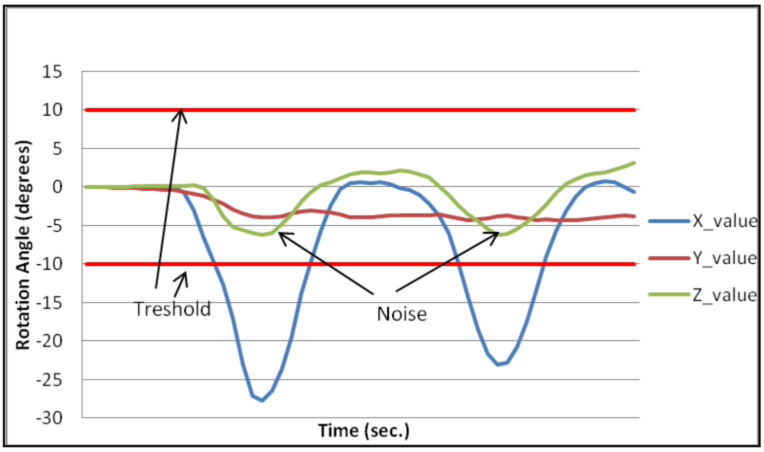

Fig. 9 (b): Giving Threshold on Raw Data of Combination Head Movement

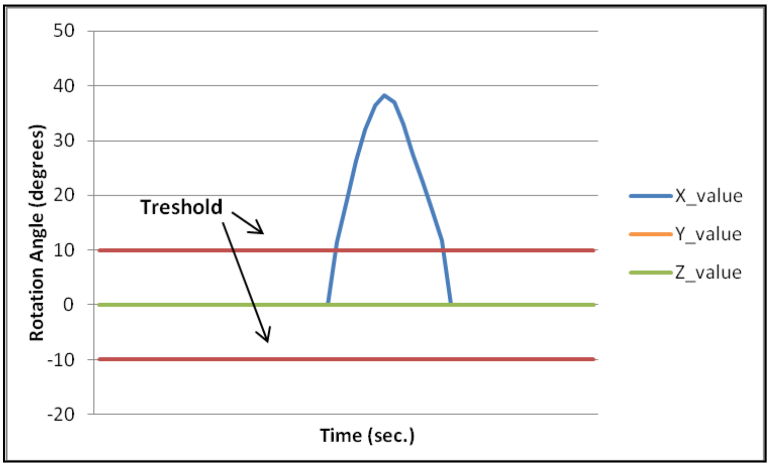

Fig. 10 (a): Clean Waveform of Single Head Movement

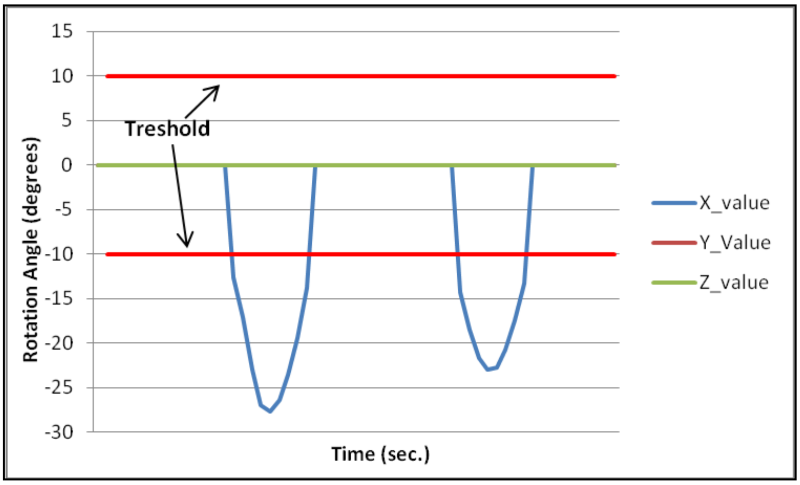

Fig. 10 (b): Clean Waveform of Combination Head Movement

5.3.2.1 *Head Movement Time Calibration*

Time calibration is a period that user takes to perform the head movement according to his body conditions. The parameter which used for movement time calibration were a time interval of user's head movement and the time interval between head movement. Figure 11 shows the illustration about taking time interval as a step for movement calibration.

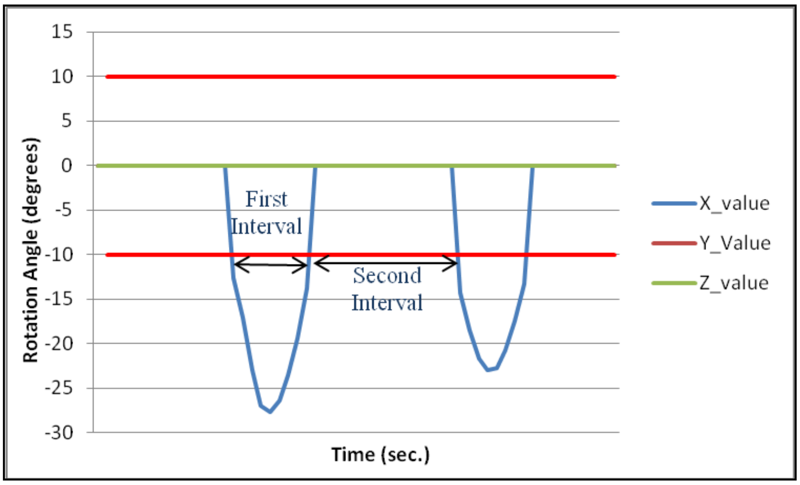

Fig. 11: Time Interval of user's head movement and between head movement

Every head movement time was obtained from training data which given from user. Based on three training data, minimum time interval (denoted as T_{min}) and maximum time interval (denoted as T_{max}) was obtained. The time which allowed as control by user (denoted as T_{test}) was:

$$T_{min} \leq T_{test} \leq T_{max}$$

If $T_{test} \leq T_{min}$, the movement would not be recognized because it was too fast. A head movement that too fast could occurred when head movements were irregular. If $T_{max} \leq T_{test}$, the movement would not be recognized because it was too slow. A head movement that too slow could occurred when body movement orientation changed or head movement that looked away not for control.

5.3.3 Feature Extraction

The feature which used was maximum or minimum head movement angle values. The feature values were obtained from the peak point of the axis and other axis value in the same point of time. The point of time was the peak point of the axis which became a major concern. To handle the double movement issue, each axis got two peak point values as a feature with details as follows:

- X_1 = Maximum or minimum point *X*-axis values on the first movement.
- X_2 = Maximum or minimum point *X*-axis values on the second movement.
- Y_1 = Maximum or minimum point *Y*-axis values on the first movement.
- Y_2 = Maximum or minimum point *Y*-axis values on the second movement.
- Z_1 = Maximum or minimum point *Z*-axis values on the first movement.
- Z_2 = Maximum or minimum point *Z*-axis values on the second movement.

If the user made a single movement, then feature value for the second movement on each axis automatically set as 0.

In the training phase, The feature values were obtained without checking head movement time interval. The feature values were stored as movement data template. In the testing or recognition phase, feature values obtained by checking the head movement time first. If head movement time has qualified for recognition process, feature values will be returned and used in the classification or data matching processes.

5.3.3 Head Movement Data Classification

After preprocessing and feature extraction stage, the next process was data matching to training data. This process used Nearest-Neighbor Classification algorithm with Manhattan distance metric. In the end of matching stage, the most similar class or label for the given test data was obtained. The utilization of Nearest-Neighbor Classification algorithm was intended to simplify computation where clean data have been obtained before. The Manhattan distance metric utilization on the distance measurement of Nearest-Neighbor Classification algorithm was expected to accelerate and simplify the calculations. The combination was solely intended to get a real-time head movement recognition.

6 Experimentation

In this section, we present the setup and results from the experiment. The main purpose of the experiment was to determine the robustness related to recognition success rate and performance of our proposed method in various device types. For the experiment purpose, an Android-based real-time head movement recognizer application prototype was developed which shown in Figure 12.

Fig. 12: Application Prototype

6.1 Experimental Setup

In this experiment, the head movement which recognized was six kinds of single head movement and three kinds of double head movement. The six kinds single head movement are flexion (downward), extension (upward), lateral rotation to the left (leftward), lateral rotation to the right (rightward), lateral flexion left (tilt left), and lateral flexion right (tilt right). However, the three kinds of combination head movement are leftward-leftward, rightward-rightward, and downward-downward. The number of testing and training data that have been used for each movement were 20 and 2 samples, respectively.

The examination which has been conducted was an accuracy and performance test. Accuracy experiment purpose was to determine head movement recognition success rate by using the proposed method. Performance experiment was performed to determined how fast the proposed method in the recognition process. Performance experiment purpose was to measure whether the developed method considers being real-time. Response-time limits on a real-time control were 100 milliseconds [20].

Accuracy and performance experiment have been conducted on different hardware environment with various types of gyroscope sensor. The device which selected in the experiment was based on its price. The utilization of different devices was intended to measure the robustness of the proposed method design to different types of gyroscope sensor. The first devices used were LG Nexus 5 and Nexian Journey One for the second device. LG Nexus 5 features an InvenSense

MPU-6515 MEMS built-in Gyroscope sensor with 450 MHz Adreno 330 GPU and Quad-core 2.26 GHz Snapdragon 800 processor. Nexian Journey One features a Bosch BMG160 built-in Gyroscope sensor with CPU Quad-core 1.3GHz. We used gyro-sensor raw data with no filter. Record or capture rate which we used was 20 cycles per second.

To ensure free-hand testing, we use Cardboard to help user keeping their phone at their head. Figure 13 shows the experimental environment setup, whose objective was to simulate a scenario which a user performed head movement using Cardboard.

Fig. 13: User Experimental Setup

6.2 Accuracy Experimental Result

The accuracy experiment has been conducted by the head movement on T_{min} and T_{max} interval; 400ms and 1000ms. The accuracy experiment result on the first device is shown in Table 1. However, the accuracy experiment using the second device is provided in table 2. The average of the accuracy of the first and second device were 95% and 93,89%.

Table 1: Accuracy rate of each head movement using the first device

Movement	Total data	True	False	Accuracy Rate (%)
Downward	20	20	0	100
Upward	20	20	0	100
Leftward	20	20	0	100
Rightward	20	20	0	100
Tilt Left	20	20	0	100
Tilt Right	20	20	0	100
Leftward - Leftward	20	16	4	80
Rightward – Rightward	20	18	2	90
Downward - Downward	20	17	3	85
Accuracy average				**95**

Table 2: Accuracy rate of each head movement using the second device

Movement	Total data	True	False	Accuracy Rate (%)
Downward	20	20	0	100
Upward	20	20	0	100
Leftward	20	20	0	100
Rightward	20	20	0	100
Tilt Left	20	20	0	100
Tilt Right	20	20	0	100
Leftward - Leftward	20	15	5	75
Rightward – Rightward	20	18	2	90
Downward - Downward	20	16	4	80
Accuracy average				**93.89**

Based on Table 1 and 2, by comparing both tables, there was the insignificant effect on the accuracy of our proposed method to the hardware difference. There were some errors on the particular head movement, especially on the combination head movement. To do the combination movement, the user, in some case, does an inconsistent movement. For example, in the case when the user does the combination head movement leftward-leftward. Before the user does the second leftward, the user does not return to the initial head position (between 10 and -10 degrees) after the first leftward movement. It made the movement unrecognizable as the head movement and considered as an alteration of the user body orientation position.

6.3 Performance Experimental Result

The performance experiment result on the first and second device are shown in Table 3.

Table 3: Performance experiment result

Movement	Average Processing Time (ms)	
	Device 1	Device 2
Downward	2	3
Upward	3	2
Leftward	2	2
Rightward	2	2
Tilt Left	3	3
Tilt Right	3	2
Leftward - Leftward	2	2
Rightward – Rightward	3	3
Downward - Downward	2	3
Average	**2.44**	**2.44**

The average of the recognition process was about 3 ms for all movements. The average of the time amount was less than 100 ms. 100ms was the limit of the response time on a real-time application [20]. It can be stated that our proposed method was light and real-time.

7 Discussion

Based on the experiment which has been conducted, there were some summarizes. First, based on the accuracy experiment, the success rate of our proposed method was about 95%. This suggests that our proposed method performs very well to recognizes the head movement using the gyroscope sensor which built in the Android Smartphone by combining it with the dummy HMD, like Google Cardboard. Consequently, we can conclude that our proposed method accuracy is comparable to results of other respectable related works. Second, based on the performance experiment, the average of the time amounts to execute the recognition process was 2.44 ms, which is less than 100 ms. This concludes that our proposed method works efficiently to recognizes the head movement on the smartphone. It is caused by the Nearest-Neighbour Classifier and a a small number of feature choices which included important things. The amount of data training affects the head movement recognition performance. We believe that too many data training will not improve the accuracy significantly but will reduce the performance speed. Based on the conclusions, it can be stated that our proposed method able to recognize the head movement through Android internal Gyroscope sensor by combining it with the dummy HMD accurately, light, and real-time.

There are many potential applications can be developed by using our proposed method. This application will be focus on novelty of Human-Computer Interaction method by using human head movement as the controller. Our proposed method can be implemented as an application to help patients in the hospital which unable to use their hands normally. The application will be put on the patient's head. The patient's head movement as application controller will do many things. For example, the patient's head movement, based on the setting, can call the doctor or the nurse. Our proposed method can be implemented as an application which related to learning activities in education institution for disability students. For example, the student's head movement will be able to write the answers in examination or test, rather than use their foot which happen nowadays. We believe that our proposed method will bring another positive impact to the society by implementing our proposed method to become a new method HCI.

8 Conclusions and Future Works

In this paper, we have presented a new methodology to recognize the head movement based on the Android internal Gyroscope sensor. By using the dummy HMD to keep the device on user head, the head movement was able to recognized. The recognized head movement were single and combination movement. The core of our proposed method includes Gyroscope sensors data acquisition; cleaning the data; get feature value; and recognize the data/signal pattern using Nearest-Neighbour Classifier algorithm with Manhattan metric distance measuring. The accuracy and performance test result showed that our proposed method had a high accuracy by using only 2 data training on each movement and a very fast performance, which below than 100ms. Our proposed method is a major step toward building a new HCI method by using the head movement on the Android mobile device by combining with dummy HMD. In the future, It is a necessity to conduct the usability survey which consist the easy to use, effectiveness, and satisfaction variable. However, the User Interface and system feedback design are needed to improve the usability aspect for the user.

ACKNOWLEDGEMENTS
This research publication was supported by the Faculty of Computer Science – Brawijaya University. The authors would like to thank Yonas Asmara for their effort to conduct the experiments.

References

[1] Just A. 2006. Two-handed gestures for Human–Computer Interaction. Ph.D. thesis

[2] Guan, D., Yuan, W., Jehad Sarkar, A., Ma, T., & Lee, Y. K. 2011. Review of Sensor-based Activity Recognition Systems. *IETE Technical Review*, *28*(5), 418.

[3] Pentland, A., 1998. Smart rooms, smart clothes. In *International Conference on Pattern Recognition*, vol. 2, pp. 949-53.

[4] Huang, L., Zhou, Z., & Liu, R. 2014. Research on interaction-oriented gesture recognition. *Sensors and Transducers*, *163*(1), 316–322.

[5] Android Developer, 2010. *Sensors Overview*. Available at http://developer.android.com/guide/topics/sensors/sensors_overview.html [Accessed April 5, 2016].

[6] Tolle, H., Pinandito, A., Jonemaro, E. M. Adams, Arai, K., 2015. Virtual Reality Game Controlled With User's Head And Body Movement Detection Using Smartphone Sensors. *ARPN Journal of Engineering and Applied Sciences. Vol 10, no 20, pp 9776-9782*.

[7] Tolle, H., Arai, K., 2016. Design of Head Movement Controller System (HEMOCS) for Control Mobile Application through Head Pose Movement Detection. *International Journal of Interactive Mobile Technologies (iJIM). Vol 10(2).*

[8] Shoaib M., Bosch S., Incel O.D., Scholten H., Havinga P. J. 2014. Fusion Of Smartphone Motion Sensors For Physical Activity Recognition. *Sensors. 14:10146–10176.*

[9] King, L.M., Nguyen, H.T., and Taylor, P.B., 2005. Hands-Free Head movement Gesture Recognition Using Artificial Neural Networks And The Magnified Gradient Function, *in Proc. 27th Annu.Conf.Eng.Med. Biol., pp. 2063-2066*.

[10]Nguyen, S. T., Nguyen, H. T., Taylor, P. B., and Middleton, J., 2006. Improved Head Direction Command Classification Using An Optimized Bayesian Neural Network. *In Proc. 28th Annu. Int. Conf. EMBS, pp. 5679-5682*.

[11]Manogna, S., Vaishnavi, S., and Geethanjali, B. 2011. Head Movement Based Assist System For Physically Challenged. *in Proc. 4th ICBBE,pp 1-4*.

[12]Kim, S., Park, M., Anumas, S., and Yoo, J., 2010. Head Mouse System Based On Gyro And Opto-Sensors. *In Proc. 3rd Int. Conf. BMEI, vol. 4. pp. 1503-1506*.

[13]Jung, Y., & Cha, B. 2010. Gesture Recognition Based On Motion Inertial Sensors For Ubiquitous Interactive Game Contents. *IETE Technical Review*, *27*(2), 158–166.

[14]Yin, L., Dong, M., Duan, Y., Deng, W., Zhao, K., & Guo, J., 2014. A high-performance training-free approach for hand gesture recognition with accelerometer. *Multimedia Tools and Applications*, *72*(1), 843–864.

[15]Marasovic, T., & Papic, V., 2012. Accelerometer Based Gesture Recognition System Using Distance Metric Learning For Nearest Neighbour Classification. In *IEEE International Workshop on Machine Learning for Signal Processing*.

[16]Braine Spine, 2013. *Six Movement of The Neck*. Available: http://www.brain-spine.com/cervical_activities/six_movements_of_the_neck.html [Accessed November 10, 2015].

[17]Murty, M.N. & Devi, V.S., 2011. *Pattern recognition: An algorithmic approach*, Springer.

[18]Dougherty, G., 2012. *Pattern Recognition and Classification: An Introduction*, Springer.

[19]Jain, A.K., Dubes, R.C. 1988. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ

[20]Nielsen, Jakob. 2009. *Powers of 10: Time Scales in User Experience*. Available: http://www.nngroup.com/articles/powers-of-10-time-scales-in-ux/ [Accessed April 4, 2016].