VIRTUAL REALITY GAME CONTROLLED WITH USER'S HEAD AND BODY MOVEMENT DETECTION USING SMARTPHONE SENSORS

Herman Tolle[1], Aryo Pinandito[2], Eriq Muhammad Adams J. [3] and Kohei Arai[4]
[1,2,3]Multimedia, Game & Mobile Research Group, Informatics Department of Faculty of Computer Science, Universitas Brawijaya, Jalan Veteran Malang, Indonesia
[4]Information Science Department, Saga University, Saga, Japan
E-Mail: emang@ub.ac.id

ABSTRACT

This paper proposes methods for detecting user's body and head movement using accelerometer and gyroscope to control the natural visualization of the 3D game object in smartphone platforms. A virtual reality (VR) game has implemented on smartphone to demonstrate the new controlling system. The user can control the visualization of the 3D VR displayed in a new head-mounted displayed (HMD) like Google Cardboard by moving user's head, walking and rotating. The pattern of real-time data gathering from the accelerometer and gyroscope inside a smartphone is recognized to determine user's movement for controlling the visualization of VR game object. Our method success to implement on a virtual reality labyrinth game while the user had to find the way on the labyrinth game by walking, rotating and moving the head to control the game visualization.

Keywords: accelerometer, gyroscope, Google cardboard, virtual reality game.

INTRODUCTION

The rapid development of mobile devices has big impact in people's lives. Nowadays mobile devices such as smartphones have become part of everyday life of modern human living as a phone cell, messaging media, social media access, gaming and entertainment, and into personal assisting devices.

The Operating System (OS) on mobile devices also varied while mobile device manufacturers vying to design and develop their own operating system and application programs to provide convenience for users. Thus, this kind of competition emerges many new ideas, innovations, and creativities among them. OSes in the mobile device platform that are very popular for its use in entertainment are iOS and Android operating system. The iOS is an operating system that used in Apple products such as iPhone and iPad. Allan (2011) and Mark (2012) describes the iOS operating system along with its basic sensor in detail while Android is the common operating system found in mobile devices and smartphones that run mainly on ARM processors.

Virtual Reality (VR) and Augmented Reality (AR) is a technology in which both of them used in modeling the real world into a computer system to assist and support human activities. While using this technology, users are in a virtual space while still having a feel in real life. It is common in the implementation of VR and AR technology combined with a 3D image display system. Rideout (2010) and Foino (2012) explain 3D and graphics programming in mobile devices using OpenGL respectively.

High-end mobile devices are currently equipped with accelerometer sensor in providing instantaneous acceleration and gyroscope sensor that sense the angular velocity of the device. These sensors are mainly used in motion intensive mobile gaming application. iOS has various tools and libraries that is used to develop video-based games and applications that utilizes accelerometer and gyroscope sensors (Developer on iTunes, 2014) while Android devices able to detect device movements with a sensor fusion (Google Tech Talks, 2014). Therefore, a 3D VR system that utilizes movement of user body as a 3D display controller can be implemented on mobile devices.

In previous research conducted by Arai and Tolle (2013), an application has been developed to help visualization of the interior of an architect-designed building where users allowed to see and imagine visualized image in 3D while moving and holding the device. User's movements handled in the research are user's rotation, vertical tilting, and forward-backward horizontal movements. Kwapisz (2010) uses Android-based smartphone's accelerometer sensor to recognize human activities by keeping it the pocket using decision trees (J48), logistic regression, and multilayer neural networks classification techniques. Shoaib *et al.* (2014) and Wu *et al.* (2012) also introduce evaluation on classification method of using an internal sensor of a mobile device for recognizing user physical activities. Most of these research are able to recognize correctly in over 90% accuracy rate using only 10 seconds worth of data. These research shows that mobile devices can be effective supports human movement detection by utilizing sensors that located on a mobile devices, but most of these previous work using a combination of sensors data and complex classification method.

This research is trying to propose a new method on detecting user's body and head movement and orientation utilizing accelerometer and gyroscope sensor inside a smartphone by putting the device in user's head using simple Head-mounted Display (HMD) like Google cardboard (called as dummy HMD) in efficient and effective way. Utilizing smartphones on Google Cardboard gives a new oppurtunism of using the smartphone as the controller of a VR visualization like implemented in our Labyrinth 3D virtual reality game.

USER'S MOVEMENT DETECTION METHOD

Overview of user movement detection

This system working with a dummy HMD using a smartphone with internal sensors (accelerometer, gyroscope, and magnetometer). User wears an HMD with a smartphone as shown in Figure-1 while system display the virtual reality game application developed on the smartphone. The user can control the view of 3D virtual reality object while moving the head or walking and rotating his/her body. User's action and movement is detected through real-time data gathering from accelerometer and gyroscope or magnetometer.

Figure-1. Google cardboard with smartphone and how to use cardboard as HMD (Google cardboard, 2014).

The method for detecting user's movement is based on the pattern of data gathering through internal sensors. Type of movements are as follows: 1) body walking, 2) body rotating, 3) head looking to left, 4) head looking to the right, 5) head looking up, and 6) head looking down. The movement is related with the direction of view of the user. Detecting head movement is possible using gyroscope but for body movement should use the combination of sensors. We develop a new method for detecting user's head and body movement and implement it on a virtual reality game to prove the succesfull of the method.

Detecting head movement

Implementation of this system as shown in Figure 2 using a smartphone in horizontally (landscape) position. Type of head movement is look-to-the-left, look-to-the-right, looking-up, and looking-down. For looking-left and looking-right movement, the gyroscope is possible to detect the attitude. Otherwise for looking-up and looking-down head movement, degree of head (phi degree) is calculated from accelerometer using equation (1) (Rideout, 2010). The phi result is using for controlling the view of 3D virtual reality game.

$$phi = arc\tan\left(\frac{a_z}{-a_x}\right) \qquad (1)$$

phi = the degree of face looking up or down,
a_x = accelerometer x,
a_z = accelerometer z

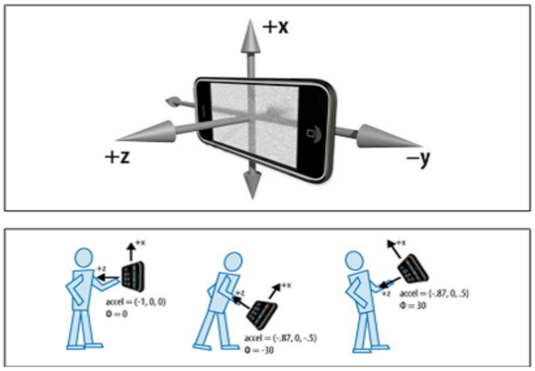


Figure-2. Pitching motions of users' attitude changes (Rideout, 2010).

Detecting body movement

There are 3 approaches for detecting user walking step, known as the pedometer, as follows (Arai, 2013): 1) Double integration of the acceleration to measure speed and then the distance. 2) The combination of gyro, magnetometer and accelerometer to find Linear Movement 3) Using CoreMotion Data.

In the previous research work (Arai, 2013) implementation of 3D VR viewer using 3rd approach. In this research work, an improvement is proposed by the combination of two methods on detecting user's walking. The first method is using CoreMotion data and the second method using only accelerometer. Since CoreMotion only exists on iOS based devices, the implementation of the method is for iOS based devices only like iPhone and iPad.

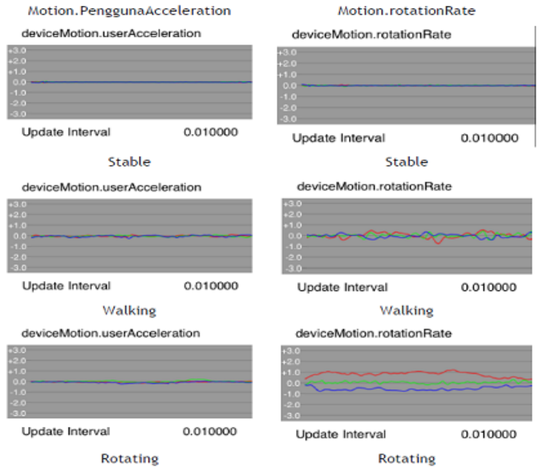


Figure-3. Signal pattern of coremotion while user walking and rotating (Arai, 2013).

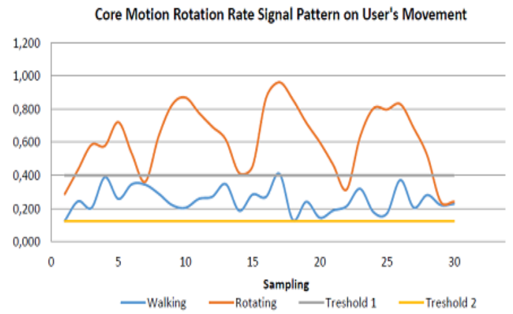


Figure-4. Plotting of core motion rotation rate signal pattern on user movement.

The new experiment is done to prove that rotation rate data on Coremotion in iOS devices is possible to detect movement of walking and rotating as shown in Figure-3. Figure-4 show the magnitude of accumulating data of rotation rate in proper sample number has different level of magnitude on walking and rotating. Based on this data, two threshold number is applied to segmented the interpretation of walking and rotating movement. When system get the accumulation magnitude is lower than threshold1 it means that user is not moving, if the magnitude value in between threshold1 and threshold2 means user is walking, and above the treshold2 means that user is rotating. This method has 84.5% accuracy of detecting user walking, some error happen when user walking too slow or too fast.

Another approach to detecting user walking by finding linear movement using a comparison of current and previous accelerometer data (Wu, 2014). This method tries to compare previous value and the current value of accelerometer values as described in equation (2) until equation (5).

$$oldValue = (a_x * a_{x_{old}}) + (a_y * a_{y_{old}}) + (a_z * a_{z_{old}}) \qquad (2)$$

$$R_{old} = \sqrt{(a_{x_{old}})^2 + (a_{y_{old}})^2 + (a_{z_{old}})^2} \qquad (3)$$

$$R_{new} = \sqrt{(a_x)^2 + (a_y)^2 + (a_z)^2} \qquad (4)$$

$$diff = \frac{oldValue}{R_{old} * R_{new}} \qquad (5)$$

$$f(diff) = \begin{cases} m = 1 \ IF \ 0{,}9 < diff < 0{,}994 \\ m = 0 \ IF \ diff > 0{,}994 \cup diff < 0{,}9 \end{cases}$$

a_x = $accelerometer\ x\ value$
a_y = $accelerometer\ y\ value$
a_z = $accelerometer\ z\ value$
a_{xold} = $previous\ accelerometer\ x\ value$
a_{yold} = $previous\ accelerometer\ y\ value$
a_{zold} = $previous\ accelerometer\ z\ value$

Figure-5. Labyrinth game screen flow.

Results of the diff then use to determined the walking movement. If the value of diff in between 0,9 and 0,994 then user is determine as walking, and other values for not walking. Since both approaches has good accuration on detecting user's walking and rotating, as reported on Table-2 in the implementation section, implement a combination of these 2 approach gain better accuration.

DESIGN OF LABYRINTH 3D VR GAME

The virtual reality labyrinth game is designed to give the user more immersive experiences as they play on real labyrinth game by using a natural interaction method and 3D stereoscopic rendering. In order to be able viewed through Google Cardboard, virtual reality labyrinth game should be rendered using 3D stereoscopic rendering technique. Thus, the game has two viewports, left viewport for left eye and right viewport for right eye as shown in Figure 9. Natural interaction method is used to simulate user's body and head movement. User has to find the way out to the goal point through a maze by control virtual character movement using their body movement and virtual character direction in a first person view using their head movement.

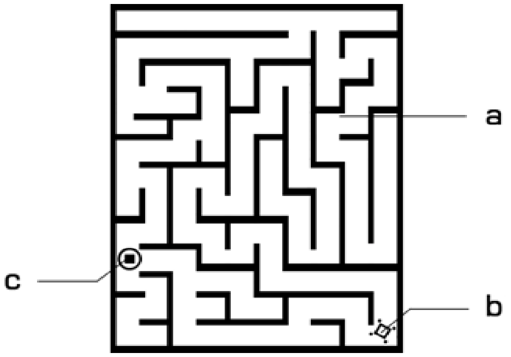


Figure-6. Design of labyrinth maze with the information of user position (b) and goal position (c).

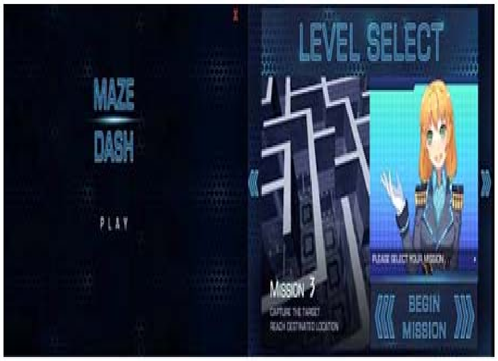


Figure-7. The VR Labyrinth game screen.

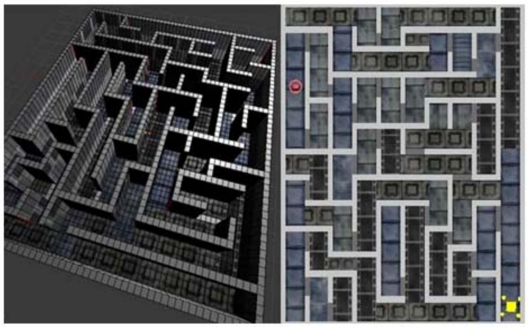


Figure-8. Maze level in the VR Labyrinth game.

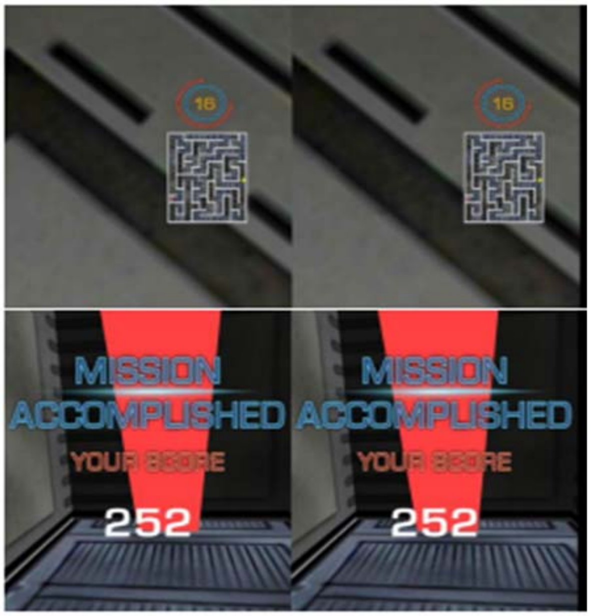


Figure-9. 3D stereoscopic rendering in the Labyrinth game.

Game design

Labyrinth game screen flow is shown in Figure-5 contain 4 processes as follows: 1) Main Menu, 2) Level Select, 3) Playing the game, and 4) Mission results. When playing the game, user sees a 3D object of labyrinth maze together some information to help the player knowing user position (b) and goal position (c) as shown in Figure 6.

A user playing the game by move his/her head, or body like rotating and step walking. User is challenged to find the goal within the limit of time. If the user completes one level, then automatically continuing to the next level with more difficult maze area. The labyrinth game is designed for several difficulties level of maze area. Some of maze area level is shown in Figure-7.

Labyrinth 3D VR game implementation

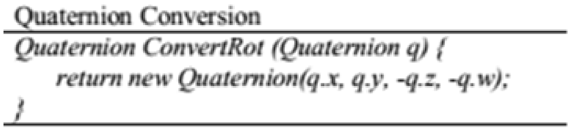
Unity3D game engine and smartphone with Android OS as well as iOS platform are used to implement virtual reality labyrinth game. The implementation on Labyrinth 3D VR game as shown in Figure-7 until Figure-9 show some screenshot of the game. The user can play several level of difficulties of maze layout.

Implementing VR on an HMD should use 2 viewports, the left viewport for the left eye and the right viewport for the right eye as shown in Figure 9. Left and right viewport showing the same screen but shifted 0.03 unit from the x-axis. Left viewport shifted 0.03 to the left and right viewport shifted 0.03 to the right of the x-axis.

In the implementation of head movement detection using gyroscope and accelerometer sensor in Unity3D, these processes has to be considered as follows: 1) Quartenation conversion, 2) Screen orientation adaptation, 3) Quaternion interpolation, and 4) User walking detection.

Quaternion conversion

Gyroscope's attitude value which is defined as quaternion value in Unity3D needs to be converted since Unity3D using a left-handed coordinate system and smartphone using a right-handed coordinate system. Figure-10 show implementation of quaternion value conversion between smartphone and Unity3D.

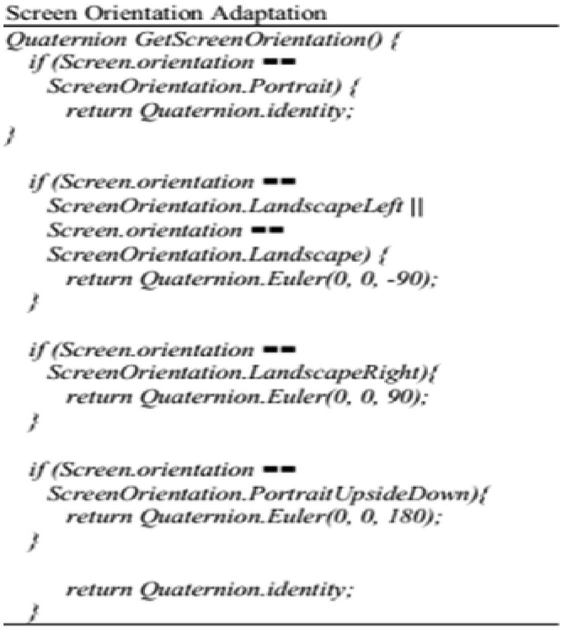


```
Quaternion Conversion
Quaternion ConvertRot (Quaternion q) {
      return new Quaternion(q.x, q.y, -q.z, -q.w);
}
```

Figure-10. Convert quaternion value.

Screen orientation adaptation

To adapt with smartphone screen's orientation such landscape orientation and portrait orientation, quaternion value must be rotated using Euler angles as explained in Figure-11.

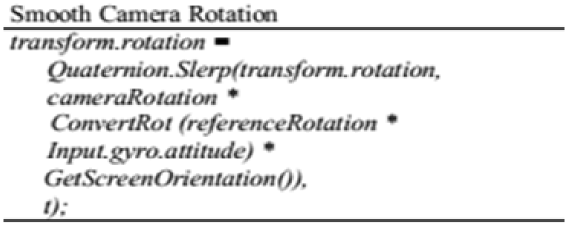


```
Screen Orientation Adaptation
Quaternion GetScreenOrientation() {
  if (Screen.orientation ==
    ScreenOrientation.Portrait) {
      return Quaternion.identity;
}

  if (Screen.orientation ==
    ScreenOrientation.LandscapeLeft ||
    Screen.orientation ==
    ScreenOrientation.Landscape) {
      return Quaternion.Euler(0, 0, -90);
  }

  if (Screen.orientation ==
    ScreenOrientation.LandscapeRight){
      return Quaternion.Euler(0, 0, 90);
  }

  if (Screen.orientation ==
    ScreenOrientation.PortraitUpsideDown){
      return Quaternion.Euler(0, 0, 180);
  }

      return Quaternion.identity;
  }
```

Figure-11. Screen orientation adaptation.

Quaternion interpolation

Quaternion values must be interpolated spherically to give smooth first person camera visualization when user do a head movement. Figure-12 describe the implementation of smooth first person camera visualization by using spherical interpolation between two quaternion values by *t*.

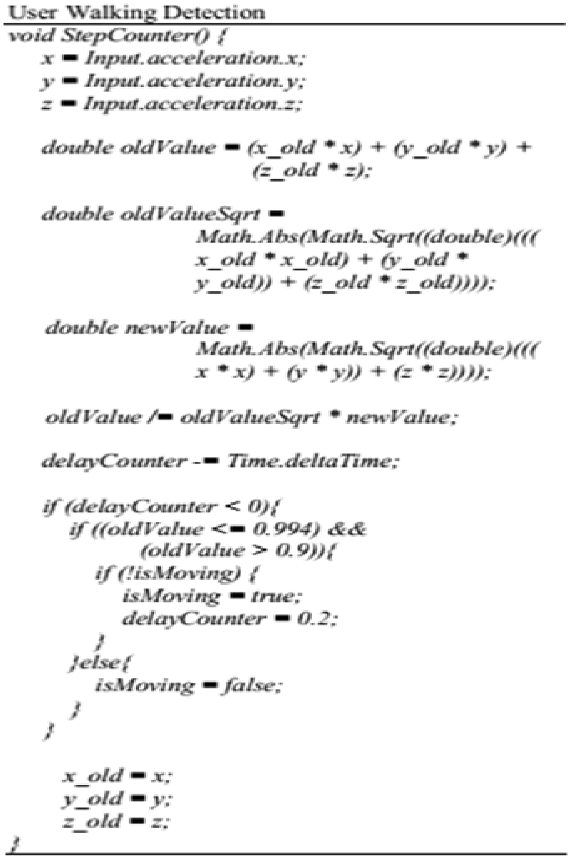


```
Smooth Camera Rotation
transform.rotation =
    Quaternion.Slerp(transform.rotation,
    cameraRotation *
     ConvertRot (referenceRotation *
    Input.gyro.attitude) *
    GetScreenOrientation()),
    t);
```

Figure-12. Quaternion interpolation.

User walking detection

User walking detection is implemented by comparing between accelerometer values in previous update frame and current update frame as described in Figure-13.



```
User Walking Detection
void StepCounter() {
    x = Input.acceleration.x;
    y = Input.acceleration.y;
    z = Input.acceleration.z;

    double oldValue = (x_old * x) + (y_old * y) +
                      (z_old * z);

    double oldValueSqrt =
                  Math.Abs(Math.Sqrt((double)(((
                  x_old * x_old) + (y_old *
                  y_old)) + (z_old * z_old))));

    double newValue =
                  Math.Abs(Math.Sqrt((double)(((
                  x * x) + (y * y)) + (z * z))));

    oldValue /= oldValueSqrt * newValue;

    delayCounter -= Time.deltaTime;

    if (delayCounter < 0){
      if ((oldValue <= 0.994) &&
            (oldValue > 0.9)){
        if (!isMoving) {
          isMoving = true;
          delayCounter = 0.2;
        }
      }else{
        isMoving = false;
      }
    }

    x_old = x;
    y_old = y;
    z_old = z;
}
```

Figure-13. Implementation of user walking detection.

RESULTS & DISCUSSION

We developed a VR 3D game to implement proposed method on detecting user's head and body movement for utilizing using smartphone sensors based controller system. The game is developed using Unity3D for Android and iOS based smartphone platform for playing with user's head and body movement using dummy HMD like Google Cardboard. Evaluation of the proposed method is conducted in 3 ways as follows: accuracy evaluation, movement speed, and user usability test.

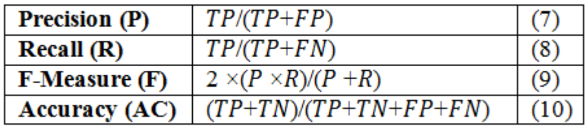
Accuracy evaluation is conducted to measure the precision and recall of detecting user's movement, and movement speed is conducted to measure the eligible speed for each movement. Usability test is to check the satisfaction of the user while playing the game.

Accuracy evaluation

Accuracy, precision and recall parameter is used to evaluate the performance of proposed method on detecting user movement. We only measure the accuration of body movement since head movement already has perfect accuration. Accuracy is the overall success rate, it is a measure of the global performance of the algorithm in what concerns correct decisions (Mendonça, 2009). Precision is the fraction of detections that are relevant

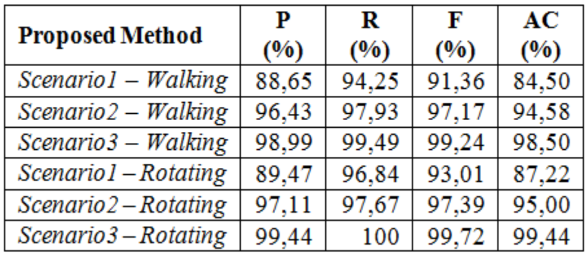
(Mendonça, 2009). Recall "refers how often the algorithm reports that an abnormality exists in the instances where it actually exists" (Mendonça, 2009). F-measure combines precision and recall by their harmonic mean (Mendonça, 2009). All of the calculation of parameters is based on the results of user movement interpretation in four categories: true-positive (TP), true-negative (TN), false-positive (FP) and false-negative (FN), as shown in Table 5. The results of proposed method in 3 scenarios as shown in Table 2, showing the best results at scenario3 when using a combination of 2 algorithms.

Table-1. Parameters of accuracy evaluation.



Precision (P)	*TP*/(*TP*+*FP*)	(7)
Recall (R)	*TP*/(*TP*+*FN*)	(8)
F-Measure (F)	2 ×(*P* ×*R*)/(*P* +*R*)	(9)
Accuracy (AC)	(*TP*+*TN*)/(*TP*+*TN*+*FP*+*FN*)	(10)

Table-2. Accuracy evaluation of proposed method.



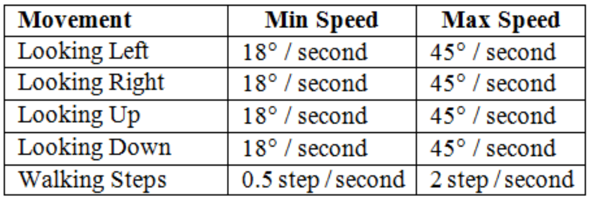
Proposed Method	P (%)	R (%)	F (%)	AC (%)
Scenario1 – Walking	88,65	94,25	91,36	84,50
Scenario2 – Walking	96,43	97,93	97,17	94,58
Scenario3 – Walking	98,99	99,49	99,24	98,50
Scenario1 – Rotating	89,47	96,84	93,01	87,22
Scenario2 – Rotating	97,11	97,67	97,39	95,00
Scenario3 – Rotating	99,44	100	99,72	99,44

Scenario1 is using accelerometer only, scenario2 is using coremotion rotation rate and scenario3 using the combination of both accelerometer linear movement and rotation rate approach. The results show that the combination method on scenario3 gives better results on all of the parameters.

Movement speed evaluation

Each movement should evaluated to determine the normal speed eligible in the systems. The speed of rotating and walking movements are evaluated to get the minimum speed and maximum speed while the application can detect movement correctly. The results of each movement as shown in Table 3 shows that rotating has appropriate value between 18°-45° per second, and walking speed is between 0.5 step – 2 step per second.

Table-3. Movement speed.



Movement	Min Speed	Max Speed
Looking Left	18° / second	45° / second
Looking Right	18° / second	45° / second
Looking Up	18° / second	45° / second
Looking Down	18° / second	45° / second
Walking Steps	0.5 step / second	2 step / second

Usability test

Usability test is conducted to measure the satisfaction of the user while playing the game. Experiments on 30 users playing the game show that all user can enjoy playing the game using their head and body movement.

The proposed method is more efficient and effective comparing to other researchers describe in introduction section because our proposed method using sensors from single smartphone put in user's head only. The efficient of proposed method shown in the simple but accurate algorithm and segmentation method, while effectiveness is observable in using combination of 2 approach on detecting user's walking that gain the accuration compared to using only single approach.

However, this kind of game controlling system using user's body movement require a spacious area for playing the game.

CONCLUSIONS

Detecting user's body and head movement are possible through a smatrphone internal sensors put in user's head with a dummy HMD like Google cardboard. The proposed method is succeeded to detect user's head movement and user's body movement. The proposed method is more efficient and effective comparing to others. Implementation of head and body movement controller system on a virtual reality 3D game give the user more immersive experiences on playing labyrinth game.

The work described in this paper is part of a larger effort to applying the internal sensors data on smartphone as a full head controller system using smartphone and carboard. In the near future, the researcher could improve the method to implement in other virtual reality application or augmented reality and for specific function of the head controller system. We believe that mobile sensor data provides tremendous opportunities for data mining and we intend to leverage our smartphone head controller platform to the fullest extent possible.

ACKNOWLEDGEMENTS

The authors would like to thank to Alfa Dhio for creating the Labyrinth game and Imam Syafii for doing the experiment on user walking detection. Boths are students at Informatics Department of Brawijaya University. Part of this research work is funded by DIPA PTIIK UB 2014.

REFERENCES

[1] Allan A. 2011. Basic Sensors in iOS. O'Reilly.

[2] Arai K., Tolle H., Serita A. 2013. Mobile Devices Based 3D Image Display Depending on User's Actions and Movements. International Journal of Advanced Research in Artificial Intelligence (IJARAI), Vol. 2, No. 6.

[3] Foino R. M. 2012. Game and Graphics Programming for iOS and Android with OpenGL ES 2.0. John Wiley & Sons, Inc.

[4] Google Cardboard SDK, http://developers.google.com/ cardboard. Accessed in January 23rd, 2015.

[5] Google Tech Talks. 2014. Sensor Fusion on Android Devices: A Revolution in Motion Processing. http://www.youtube.com/watch?v=C7JQ7Rpwn2k. Accessed in: March 10th, 2014.

[6] Jennifer R. Kwapisz, Gary M. Weiss , Samuel A. Moore, 2010. Activity recognition using cell phone accelerometers. ACM SIGKDD Explorations Newsletter, Vol. 12, No. 2.

[7] Mark D. *et al.* 2012. Beginning iOS 6 Development Exploring the iOS SDK. Apress.

[8] Rideout P. 2010. iPhone 3D Programming. O'Reilly.

[9] Shoaib M., Bosch S., Incel O. D., Scholten H. and Havinga P. J. 2014. Fusion of smartphone motion sensors for physical activity recognition. Sensors. 14, pp. 10146–10176.

[10] Wu W., Dasgupta S., Ramirez E. E., Peterson C. and Norman G. J. 2012. Classification accuracies of physical activities using smartphone motion sensors. J. Med. Internet Res. 14, e130.