SETSCI Conference Proceedings, 4 (1): 282-284, 2019

ISAS 2019
3rd International Symposium on Innovative Approaches in Scientific Studies
Section: Engineering and Natural Sciences
April 19 – 21, 2019, Ankara, Turkey

# Gait Recognition using RGB-D Camera for Player Labeling

Yilmaz Atay[1*], Agusta Wicaksono[2] and Ahmet Arslan[3]

[1]*Department of Computer Engineering, Osmaniye Korkut Ata University, Osmaniye, Turkey*
[2]*Department of Computer Engineering, Selcuk University, Konya, Turkey*
[3]*Department of Computer Engineering, Konya Food and Agriculture University, Konya, Turkey*
[*]*Corresponding author and Speaker: yilmazatay@osmaniye.edu.tr*
*Presentation/Paper Type: Oral/Full Paper*

***Abstract –*** The identification of the 3-dimensional physical properties of humans has recently become a very important field of study. In this respect, today's applications and research use depth cameras such as Kinect as an input tool for identification of quantitative characteristics of people. Due to the completeness of data offered by conventional cameras, depth-cameras are the driving force for more up-to-date research. Introduction of human objects recognition offered by its library makes researchers easily take three-dimensional data from Kinect for skeletonization algorithms. The skeletonization is an algorithm that can facilitate the depth-camera's user to detect human objects and track the bone and each joint. In this way, a human can be tracked precisely by bone and joint motion. Moreover, the camera is able to do skeleton tracking over a human object. However, if the human object is actively moving in the field, the skeleton mapping may change. In this paper, we propose a new method for labeling players by using gait recognition and other features thus even the human is moving in/out of the camera field, the camera will recognize it as the same human based on its characteristic. This approach can be easily used in real-time recognition of people and is also performed to label the player. Experiments of this method were carried out in the real environment. Tests were performed successfully in experiments where people with different characteristics are involved. According to the obtained results, the proposed method has a good performance and can be applied in real-time easily.

*Keywords – Person identification, Skeleton mapping, Player labeling, Kinect, Gait recognition*

## I. Introduction

Nowadays lots of research or applications use depth camera as its input sensor. Due to the completeness of the data offered over the conventional cameras, depth-cameras are the driving force for more up-to-date research. One of the advantages of depth-camera is to have skeletonization features in the available libraries. Skeletonization is an algorithm that can facilitate the depth-camera's user to detect human objects and track the bone and each joint. In this way, a human being can be tracked precisely by bone and joint motion. This ability is not out of deprivation. Therefore, many researchers proposed skeletonization algorithm that has better results than the basic library. Several studies about skeletonization can be found in [1-4]. In those studies, some researchers offer better results in several different conditions.

Skeletonization algorithm can be used to track human's identical movement which so-called gait recognition. Gait is a rhythmic groove of a particular body part while doing something. A person's gait when walking can be different from the others. Identification of the gait is one of the biometric recognition of human objects. Some studies that utilizes skeletonization algorithm from depth-cameras to identify gait related to biometrics identification can be seen in [5-7]. The application in other fields such as [8, 9] are the use of gait recognition for security purpose. In sports and health respectively can be found in [10, 11] and [12-14]. In human interaction with machine, human movement are also used in order to recognize the body language [15] so that machines can understand the human's intention without using voice commands or keys.

The skeletonization algorithm which is included in the built-in library allows a depth-camera to skeletonize one or more human objects. This feature is based on segmentation of human objects that may be found more than one in a frame. Then skeletonization in each object is performed in detail. This makes the depth-camera's performance becomes very heavy because it requires a high processor to process multiple objects in a single frame. In this research depth-camera we use is Microsoft Kinect. The version we use on the other hand, is Kinect v1. Kinect v1 is different from Kinect v2 which can only detect skeleton up-to 2 objects only. Kinect v2 allows us to track the human objects up-to 6. However, Kinect v2 requires greater power capability than Kinect v1. Therefore, using Kinect v2 will maket his development unsuable in a low power computation such as single board computer.

This research aims to do dynamic player labeling in real time against one or more human objects that are immune to displacement of player's location even though the player goes in and out to Kinect's detection area. Kinect v1 and the SDK v1.8 libraries are used to get the skeleton to be mapped. In this research we use component gait in skeleton that are static and dynamic. In practice, however, dynamic features are less helpful if the human object performs abnormal movements such as jumping or dropping suddenly. Therefore, we added other features such as clothing color and skin color as a counterweight. We avoid facial recognition because of resource consuming. This paper is structured in several chapters that are important for further discussion. The first chapter is the introduction of the problem. The second chapter is a literature studies that discusses the sensors we use and some other techniques. The third chapter is a discussion of the proposed method. Chapter Four settings and implementation. The fifth chapter is the discussion of research results and the last is conclusion.

## II. Related Works

Some of the studies that became our concerns that have been done by some previous researchers discussed in detail. Those cover some method that will be considered either in implementation or in experimental result. And those previous works are:

### 1. *Distance Feature*

Aniruddha et al, perform gait recognition using Kinect by utilizing features that are dynamic, i.e. features that can change over time but have certain patterns [1]. The dynamic feature chosen by Aniruddha et al, is a feature of the distance between the wrists. Using Kinect v1, 20 joint points can be obtained and then calculated distance by Euclidean distance as below:

$$k = |X_{\text{LeftAngle}} - X_{\text{RightAngle}}|$$

Where *k* is 1 to *N*, and *N* is the total frame per second. Then another static distance feature is the distance between one joint to another joint that can be calculated with the vector spacing as follows:

$$d_{ij} = \sqrt{||X_i - X_j||^2 + ||Y_i - Y_j||^2 + ||Z_i - Z_j||^2}$$

Where *i* and *j* are joints to be calculated.

### 2. *Angle Feature*

In addition, dynamic features also use static features as characteristic of human objects. Static features can be represented by chest width, waist width, arm length, etc. Different from previous research that using distance features, in this research the researchers use the angle feature that is created when the object moves. This angle is formed between three joint points that can be clearly seen in Figure 2.

## III. The Proposed Approach

In order to identify players, we use the approach that we consider important. This approach will be discussed in depth. In our system, static and dynamic features are captured in real time. Therefore, we use the recording method where the trigger is the condition where the difference in distance between the two feet is significant with the difference in distance between the two feet on average. This indicates a significant change in a short period of time and indicates that humans are making a stepping move.

$$\| Stepdist_i - \overline{Stepdist} \| > t$$

Where "*Stepdist*" is the current distance between the both LeftAngle joints and RightAngle. Then $\overline{Stepdist}$ is the average value of the *Stepdist*. Whereas *t* is the specified threshold for trigger. For more explanation of triggering records on and off, the flow chart in Figure 1 below represents the details.

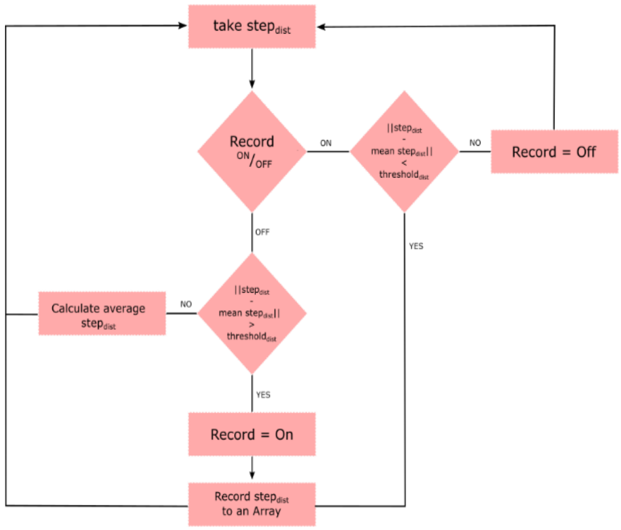
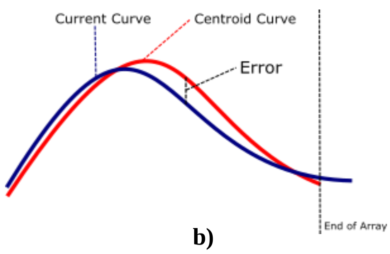



**Figure 1.** The flowchart diagram

The diagram in Figure 1 describes the mechanism of taking a dynamic feature in real time. When data retrieval takes place, the data is stored in an array that contains the value of the distance/angle per frame. In dynamic features we take steps and swings from each player for analysis. Each step and swing is measured by distance and angle. As we can see in the picture below, dynamic features are parsed from the player's body. Then the data array that is formed from each step is stored for later analysis in the next step.


**a)**




**b)**

**Figure 2. a**) Step and swing, **b**) Pattern comparison

if drawn, the data in the array will form a curve which contains angles and distances that represent a step and a human swing. This step and swing will be compared to centroid data. To find out how much the error is from the data we use RMSE in comparison.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(A_i - C_i)^2}{n}}$$

If the *RMSE* value of these two curves is too large then the threshold will break and the system considers it as data that has different characteristics. From all features used, 5 of them are static features and 4 of them are dynamic features. Dynamic features have a greater weight than static features so the threshold number will be 13 with the translation of;

$$Total\ T = 5Ts + 2 \times 4Td$$

The color comparison, on the other hand is performed by comparing the clothing's color around the human body. Instead of using RGB color space, the comparison is done in CIELab color space. This will make the comparison more accurate if there are two similar color given. Then a threshold is set depends on the Delta E of the two given color.

## IV. Experimental Results

The experimental results obtained regarding to the scenario are shown in Figure 3. In the first, we get the player to move around the area, staying there for a few minutes to let the camera performs the clustering. Then we get the player to do back and forth movement to the area to check the clustering result.

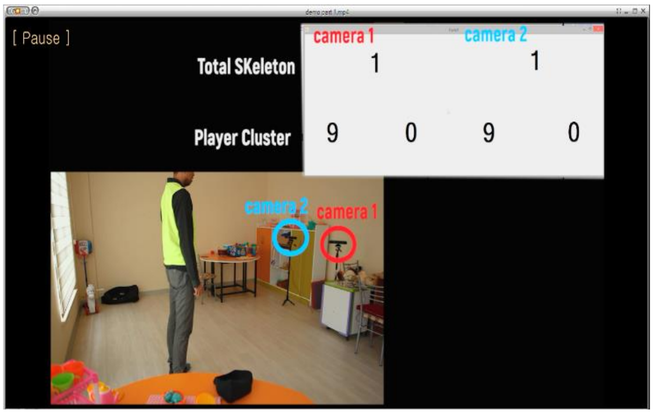



**Figure 3.** The sample output

## V. Conclusion

The clustering system in the multi-camera environment is possible to be done. The performance is affected by the tracking especially in the overlap area. Moreover, clustering performance may struggle due to random lighting source. In order to reach the highest performance, we may add more static and dynamic feature in the clustering model.

## References

[1] Kastaniotis, D., Theodorakopoulos, I., Theoharatos, C., Economou, G., & Fotopoulos, S. (2015). A framework for gait-based recognition using Kinect. *Pattern Recognition Letters*, *68*, 327-335.

[2] Lu, W., Zong, W., Xing, W., & Bao, E. (2014). Gait recognition based on joint distribution of motion angles. *Journal of Visual Languages & Computing*, *25*(6), 754-763.

[3] Chattopadhyay, P., Roy, A., Sural, S., & Mukhopadhyay, J. (2014). Pose Depth Volume extraction from RGB-D streams for frontal gait recognition. *Journal of Visual Communication and Image Representation*, *25*(1), 53-63.

[4] Yang, K., Dou, Y., Lv, S., Zhang, F., & Lv, Q. (2016). Relative distance features for gait recognition with Kinect. *Journal of Visual Communication and Image Representation*, *39*, 209-217.

[5] Sun, B., Zhang, Z., Liu, X., Hu, B., & Zhu, T. (2017). Self-esteem recognition based on gait pattern using Kinect. *Gait & posture*, *58*, 428-432.

[6] Gianaria, E., Grangetto, M., Lucenteforte, M., & Balossino, N. (2014, June). Human classification using gait features. In *International Workshop on Biometric Authentication* (pp. 16-27). Springer, Cham.

[7] Dikovski, B., Madjarov, G., & Gjorgjevikj, D. (2014, May). Evaluation of different feature sets for gait recognition using skeletal data from Kinect. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on* (pp. 1304-1308). IEEE.

[8] Yang, L., Ren, Y., & Zhang, W. (2016). 3D depth image analysis for indoor fall detection of elderly people. *Digital Communications and Networks*, *2*(1), 24-34.

[9] Nieto-Hidalgo, M., Ferrández-Pastor, F. J., Valdivieso-Sarabia, R. J., Mora-Pascual, J., & García-Chamizo, J. M. (2016). A vision based proposal for classification of normal and abnormal gait using RGB camera. *Journal of biomedical informatics*, *63*, 82-89.

[10] Arifoglu, D., & Bouchachia, A. (2017). Activity recognition and abnormal behaviour detection with recurrent neural networks. *Procedia Computer Science*, *110*, 86-93.

[11] Tran, T. H., Le, T. L., Hoang, V. N., & Vu, H. (2017). Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment. *Computer methods and programs in biomedicine*, *146*, 151-165.

[12] Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, *12*(2), 1437-1454.

[13] Yavşan, E., & Uçar, A. (2016). Gesture imitation and recognition using Kinect sensor and extreme learning machines. *Measurement*, *94*, 852-861.

[14] Panahi, L., & Ghods, V. (2018). Human fall detection using machine vision techniques on RGB–D images. *Biomedical Signal Processing and Control*, *44*, 146-153.

[15] Lei, J., Song, M., Li, Z. N., & Chen, C. (2015). Whole-body humanoid robot imitation with pose similarity evaluation. *Signal Processing*, *108*, 136-146.