Ada-Boost based Gesture Recognition using Time Interval Window

S.J. Hwang, J.P. Na, S.J. Park, J.H. Baek
Department of Information and Telecommunication Engineering
Korea Aerospace University
South Korea

Abstract **—The task of recognizing 3D gesture for controlling equipment is highly challenging due to the propagation of 3D smart TV recently. In this paper, the Ada-Boost algorithm is applied to 3D gesture recognition by using a Kinect sensor. We recognized time-invariant 3D gesture using global and local feature vectors that are normalized. The multi Ada-Boost algorithm is used to train and classify many 3D gesture types. Our experiment shows 95.17% of accuracy and 3.73% of error rate.**

Keywords-3D gesture recognition; machine learning; Ada-Boost

I. INTRODUCTION

As electronic devices evolve, many devices which called smart TV, smart refrigerator, smart phones and other products are being developed. Common features of these devices can be seen in changes in the way they are controlled. As camera and sensor technology evolved, many HCI (Human Computer Interaction) technology that appeared in multiple science fiction movies in the past have emerged. This paper proposes a method to control the behavior of recognized humans by a KINECT sensor. Related research in gesture recognition has been studied through the DTW (Dynamic Time Warping), and HMM (Hidden Markov Model) algorithm with temporal and spatial variations. In this study, we propose a set of the applicable features with the temporal and spatial variation in the Ada-Boost algorithm.

II. MACHINE LEARNING MODEL

A. Related Machine Learning Model

DTW, Neural networks, and the Hidden Markov Model are techniques to extract patterns with time and space variations. The DTW can include multiple candidates that calculate the distance between the input pattern and the reference pattern. In addition, DTW is available when including less training data because of the typical shape of the template to be used as a reference pattern. Neural networks can utilize only one candidate because they have to calculate the posterior probability of the input pattern. For the training of the Neural network, because it requires a huge amount of training data, it is not suitable for the application to obtain hard data. To evaluate the degree of similarity between the input pattern and the reference pattern, Hidden Markov models can be multiple candidates. In addition, there is no need to consider more about temporal and spatial changes in the process of matching the reference pattern, since temporal and spatial variations are expressed in the probability of each state and spread. Study of 3D gesture recognition using the HMM is increasing [1][2]

B. Ada-Boost Algorithm

After it sequentially generates classification rules, the Ada-Boost algorithm readjust the distribution of sample data from the observed values obtained by applying the previous classification rules. Weight of the sample data will start in the same state as early learning. As each round progresses, the misclassified data is given a high weighting from observations obtained by applying the previous classification rules. In contrast, distribution of the sample data is rebalanced in a way that gives low weights the correct classification data[3]. Show that the pseudocode of the Ada-Boost algorithm in Figure 1[4]. This Ada-Boost algorithm has been widely used to study pose recognition as posture data[5][6][7].

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

FIGURE I. PSEUDOCODE OF ADA-BOOST ALGORITHM.

III. 3D GESTURE ACQUISITION AND FEATURES VECTOR

A. KINECT Skeleton Data

Equipment to obtain the 3D Data is diverse[8]. In this paper, KINECT was used to acquire the human body coordinates. KINECT divides a user's body into a total of 20 joints with three-dimensional coordinates using infrared structured light. By using the KINECT, the required time is shortened for obtaining the coordinate values.

B. Set of Features Vector

A user's physical condition that uses gesture recognition cannot be the same for all. Feature normalizing was used to solve the problem of these physical differences. The hand feautre vector is defined by the vector from spine to hand. In addition, each hand feature vector is normalized by dividing the user's shoulder width. We improved recognition rate change according to the user's body by using the local and global features vector. Figure 2 shows proposed local and global feature vector.

$$f_1 = x_{12} - x_1,\ f_4 = x_{24} - x_{13},\ \cdots,\ f_{13} = x_{60} - x_{49}$$

$$f_2 = y_{12} - y_1,\ f_5 = y_{24} - y_{13},\ \cdots,\ f_{14} = y_{60} - y_{49}$$

$$f_3 = z_{12} - z_1,\ f_6 = z_{24} - z_{13},\ \cdots,\ f_{15} = z_{60} - z_{49}$$

$$f_{16} = \overline{p_s - p_e},\ f_{17} = x_s - x_e,\ f_{18} = y_s - y_e,\ f_{19} = z_s - z_e$$

$$f_{20}, f_{22}, f_{24} = \sum_{i=start}^{end} (\arccos((y_{i+1} - y_i)/(x_{i+1} - x_i))), (\theta > 0)$$

$$f_{21}, f_{23}, f_{25} = \sum_{i=start}^{end} (-\arccos((y_{i+1} - y_i)/(x_{i+1} - x_i))), (\theta < 0)$$

$$f_{26} = \sum_{i=start}^{end} \theta,\ f_{27} = \sum_{i=start}^{end} \theta^2, (\theta = \angle(p_{i+1} - p))$$

$$f_{28} = \overline{p_{max} - p_{min}}$$

$$f_{29} = (x_{max} - x_{min})/f_{28}, f_{30} = (y_{max} - y_{min})/f_{28}, f_{31} = (z_{max} - z_{min})/f_{28}$$

$$f_{32} = \sum_{i=start}^{end} \overline{p_{i+1} - p_i}$$

$$f_{33} = \max(\overline{p_{i+1} - p_i})$$

FIGURE II. LOCAL AND GLOBAL FEATURES VECTOR.

IV. TRAINING AND RECOGNITION

A. *Feature Vector Training*

In this study, we have designed a classifier whether input behavior is true or false based on a database which was constructed using the Ada-Boost algorithm. By applying the Ada-Boost algorithm for each respective operation, we have used the concept of multi Ada-Boost.

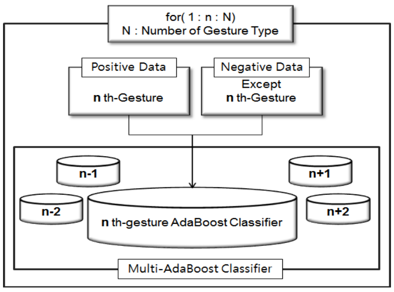


FIGURE III. METHOD OF BINARY CLASSIFICATION AND MULTI-ADA-BOOST CLASSIFIER.

Using the binary classification scheme of 1:N, all the behaviors except the desired one gets negative role. Therefore, as in Figure 3, when the gesture was classified into four kinds, Gesture 1 is the goal of this training, the rest of them being the negative training data.

B. *Time Interval Window*

In this study, the reference feature vector set is defined as a feature vector of 60 frames. Input feature set can be recognized in the slowest gestures and in the fastest gesture of 20 frames through the time interval window.

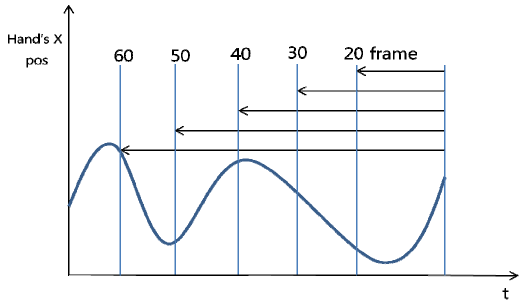


FIGURE IV. TIME INTERVAL WINDOW.

C. *Gesture Recognition*

Figure 5 shows a flowchart recognition that progress in every frame. In order to recognize the gestures in real time, in this paper, we entered the feature vector into the Ada-Boost categorizer. The categorizer performs the gesture recognition of only the number of every sampled frame. Number of recognized gestures are determined for each sample and the number of gestures per frame.

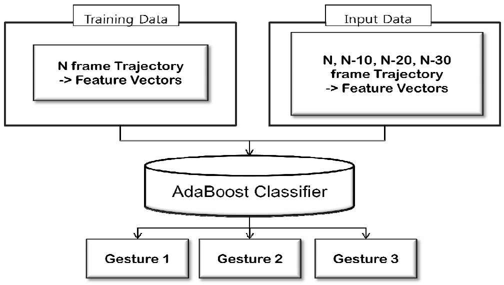
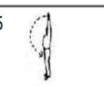
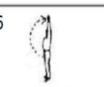
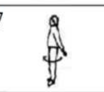
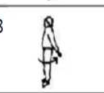


FIGURE V. FLOWCHART OF GESTURE RECOGNITION PER FRAME

V. EXPERIMENTS AND RESULT

Through two experiments, in this study, confirms the performance of the proposed feature. The first experiment compares the recognition rate of the proposed feature and Jia sheng's feature by a variety of gesture type[3]. In the second experiment, we performed comparative experiments of recognition rate due to changes in the speed of gesture of the proposed algorithm. The gestures used for the test are shown in Table 1.

TABLE I. GESTURE TYPE.

1 		5 	
2 		6 	
3 		7 	
4 		8 	

A. Compare with Gesture Type

In this experiment, we compare the change in recognition rate in the proposed algorithm with the algorithm of Jia Sheng.

TABLE II. RECOGNITION RATE ACCORDING TO THE GESTURE TYPE.

	Jia-Sheng Feature		Proposed Feature	
Gesture type	Accuracy (%)	Error (%)	Accuracy (%)	Error (%)
Gesture1	99.00	0.67	98.17	2.83
Gesture2	95.50	2.83	95.83	0.83
Gesture3	95.33	3.33	94.83	5.50
Gesture4	99.00	1.00	98.50	1.67
Gesture5	91.33	6.00	90.83	3.00
Gesture6	96.50	2.83	97.00	3.67
Gesture7	85.83	16.50	90.83	7.67
Gesture8	92.50	5.33	95.33	4.67
Mean	94.38	4.81	95.17	3.73

Table 2 shows a higher accuracy of the proposed algorithm than Jia Sheng's algorithm. In addition, error value decreased in the proposed algorithm. By using the normalized features vector with time interval window does not occur any reduction in recognition rate to changes in the gesture type. Therefore, we were able to know that the gesture recognition can be used for multiple gestures.

B. Compare with Gesture Speed Variation

TABLE III. RECOGNITION RATE ACCORDING TO THE GESTURE SPEED.

	Jia-Sheng Feature		Proposed Feature	
Speed (sec/gesture,	Accuracy	Error	Accuracy	Error
0.66 (20)	90.25	8.92	92.83	5.83
1.00 (30)	94.50	4.08	94.92	3.42
1.33 (40)	95.25	3.83	95.17	4.17
2.00 (60)	97.50	2.42	97.75	1.50
Mean	94.38	4.81	95.17	3.73

The most important feature of this proposed algorithm is robustness to changes in the speed. Thus, in this experiment, we have investigated the changes in the recognition rate when there was a change of speed. As shown in Table 3, recognition rate did not change significantly even after the speed of the gesture changed. In addition, the accuracy values have shown high from 92% to 97%. This means that proposed algorithm is robust to changes in the speed of the user's gesture.

VI. CONCLUSION

In this study, we propose an algorithm to react strongly to the operating speed of the user. The experimental results indicate that even if the gesture speed changed the decrease in the recognition rate did not occur, because we were classified based on features that applied time interval window.

We have experimented by controlling the PC-based media player using the proposed algorithm. In future research, we want to be able to conveniently control using gesture recognition in smart TV. As mentioned in the introduction, smart device's control scheme is developed around motion recognition conveniently. In this trend, we expect that this proposed algorithm is able to respond to the desires.

ACKNOWLEDGMENTS

This study was conducted with the assistance of the Korea Aerospace University Technical Research Center of the next generation broadcast media by the GRRC(Gyeonggi-do Regional Research Center) program.

REFERENCES

[1] Lee, Hyeon-Kyu, and Jin-Hyung Kim. "An HMM-based threshold model approach for gesture recognition."Pattern Analysis and Machine Intelligence, IEEE Transactions on 21.10, 961-973, 1999.

[2] Whitehead, Anthony, and Kaitlyn Fox. "Device agnostic 3D gesture recognition using hidden Markov models." Proceedings of the 2009 Conference on Future Play on@ GDC Canada. ACM, 2009.

[3] Sheng, Jia. A study of adaboost in 3d gesture recognition. Technical report, University of Toronto, 2005.

[4] Freund, Yoav, Robert Schapire, and N. Abe. "A short introduction to boosting."Journal-Japanese Society for Artificial Intelligence 14.771-780 , 1612, 1999.

[5] Wang, Chieh-Chih, and Ko-Chih Wang. "Hand Posture recognition using Adaboost with SIFT for human robot interaction." Recent progress in robotics: viable robotic service to human , 317-329, 2008.

[6] Yao, Yi, and Chang-Tsun Li. "Hand Posture Recognition Using SURF with Adaptive Boosting.", 2011.

[7] Changrampadi, Mohamed Hashim. "A fusion-based multiclass AdaBoost for classifying object poses using visual and IR images.", 2011.

[8] Hoffman, Michael, Paul Varcholik, and Joseph J. LaViola. "Breaking the status quo: Improving 3d gesture recognition with spatially convenient input devices", Virtual Reality Conference (VR), IEEE, 2010.