The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Hash based Early Recognition of Gesture Patterns

Yoshiyasu Ko*, Atsushi Shimada*, Hajime Nagahara*, Rin-ichiro Taniguchi*

* Kyushu University, Fukuoka, Japan
E-mail: {ko,atsushi,nagahara,rin}@limu.ait.kyushu-u.ac.jp

Abstract: We propose an accelerated approach of gesture recognition, "Early Recognition". Early recognition is a method to make a decision of human gesture at their beginning part of it. A gesture consists of a sequential postures. In the training phase, a lot of postures which make the gestures are trained by Self-Organizing Map (SOM). Then, representative postures on the SOM are investigated their contribution ratios for each gesture. Besides, the representative postures are registered into a hash table to realize quick search of a posture in the test phase. When an input gesture is given, a posture in each frame is used as a query and the most similar posture is found from the training samples. Then, if the contribution ratio for a certain gesture exceeds a predefined threshold, the system outputs the gesture label as the recognition result. Through our experiments, we found out that the proposed method outperformed the traditional method in terms of the computational cost and the response speed until the result was determined. Besides, the recognition accuracy of the proposed method was almost the same as the traditional one.

Keywords: Early Recognition, Locality-Sensitive Hashing, Self-Organizing Map

1 INTRODUCTION

Recognizing human gestures plays an important role to realize a man-machine interactive system. We tackle the problem of realizing a seamless interaction required video action game consoles, robotics, remote controller, etc. Generally, a system recognizes a human gesture after the gesture has completely finished. Therefore, if the length of the gesture is long, a user has to wait for the response until the recognition process finishes. That's why general system occurs a large time lag, and loses the smoothness.

In recent years, a new methodology of gesture recognition called "early recognition" has been proposed[1][2]. Early recognition is the method that system determines the recognition result before user's gesture has finished. It gives us very useful solution for the problem of delay of interaction. In the previous approach[3], Self-Organizing Map (SOM)[4][5] is utilized for making a codebook of representative postures from training samples, then contribution ratio of each representative posture is estimated for each gesture class. In recognition process, the best matched representative posture is found from the codebook by searching all nodes of SOM. Therefore, the larger the number of nodes becomes, the longer time search phase takes. Furthermore, if searching takes longer time than a frame rate of input, a system will drop several frames, that may result in lower performance of recognition speed. Generally, the performance depends on the number of nodes, but it requires a large computational cost to find a best matched one.

In this paper, we propose the accelerated approach for searching the best matched node by using Locality-Sensitive Hashing (LSH)[6], in which the computational time does not depend on the size of code book. In the following this paper, we will give a detailed explanation about the proposed method. Besides, the effectiveness of the proposed method becomes clear through the experimental results.

2 PROCESSING FLOW OF PROPOSED METHOD

Proposed method is divided into two processes; training process and recognition process.

2.1 Training Phase

Training process is divided into three steps. Step1 and Step2 are inspired from the previous work by Kawashima *et al*[3]. Flow of this process is explained as follows,

Step 1 Training of Representative Postures

Step 2 Calculation of Contribution Ratio

Step 3 Registration to Hash Table

First, the training sample is clustered by SOM and center vector of each cluster, which is called "representative posture" is output to each neuron of SOM. Secondly, "contribution ratio", which shows how often each representative posture is used for each gesture class is calculated. Finally, sets of representative postures and contribution ratio are registered to hash table. In following subsection, we explain each step in detail.

Step 1 *Training of Representative Postures*

The posture data, which are the skeleton data (Fig.1) sequences of human posture from Microsoft Kinect SDK, are collected as training samples. Each skeleton data has 20 observation points and each observation point

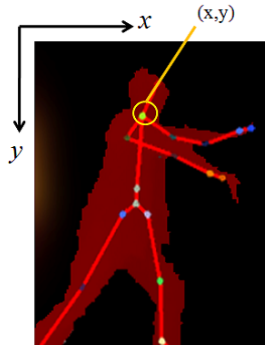

Fig. 1. Skeleton data

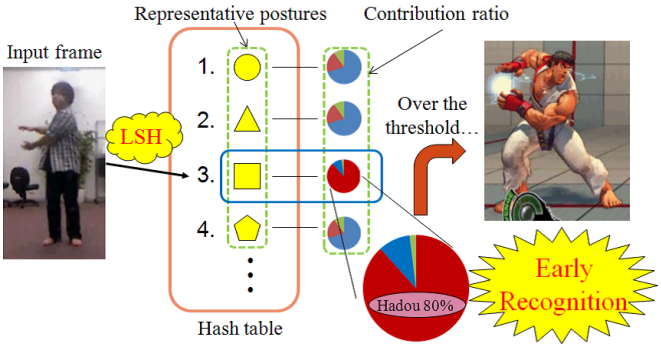

Fig. 2. Flow of Recognition Phase

has 2 dims (x, y) information, so each frame data has 40 dims vectors as information of posture (1 frame = 1/30 sec). Then, that samples of motion's template postures are clustered by SOM and a output center vector of each cluster, which called "representative posture", are registered to each neuron of SOM.

Step 2 *Calculation of Contribution Ratio*

"Contribution Ratio" denotes how often a representative posture is used for each gesture class. The template postures of each motion are input to SOM again and we count the number of times which shows how often the neurons of SOM are used for each gesture class. Then, the ratio of that number of one motion to that of all motions is treated as contribution ratio. In short, The higher value it becomes in only one motion, the more peculiar the posture is to the motion. That ratio is also registered to SOM with each representative posture. Then, the representative posture and the contribution ratio of that are linked by a label number.

Step 3 *Registration to Hash Table*

The representative postures registered to neurons of SOM are registered to hash table. The parameter l and k (Sec.3) need to be selected the proper values because this values are directly relative to the recognition accuracy.

2.2 Recognition Phase

Flow of the recognition phase is shown by Fig.2. The recognition phase is performed frame by frame. In each frame, a best matched posture is searched from all registered representative postures in hash table. If a matched posture is found, its corresponding contribution ratio for each gesture is examined. If a contribution ratio of a certain gesture is higher than a predefined threshold, the system makes final decision of gesture recognition. In other words, the gesture label is the recognition result.

The previous method requires $O(n)$ (n is the number of representative postures) operations to find the best matched posture because of a greedy algorithm. When the n increases, the calculation cost also increases explosively. It is not desirable from the viewpoint of early recognition. Meanwhile, the operation cost of a hash-based approach is not affected by the number of samples. LSH requires $O(kl)$ operations. The values of the k and l are very smaller than n. Moreover, it guarantees a constant operation cost even if the number of postures increases. The search method of LSH is explained in Sec.3.

3 LOCALITY-SENSITIVE HASHING (LSH)

Locality-Sensitive Hashing (LSH) is a kind of approximate nearest neighbor search. Nearest neighbor search needs to calculating distance from query data point to each data point and it takes an enormous amount of time. However, "approximate" nearest neighbor search is the way of finding slate points which are near from query data point. This way can reduce much calculating time because of not requiring to calculate distance between each data points and query data point.

This search method uses "locality-sensitive" hash function. It is the function that the nearer the distance from query data to a data point, the higher value the probability that hash values of both data are matched takes. This function is defined by,

$$h(\boldsymbol{p}) = \left\lfloor \frac{\boldsymbol{a} \cdot \boldsymbol{p} + b}{w} \right\rfloor \quad (1)$$

$\boldsymbol{a}$ is the vector whose value of each element is obtained independently by Gaussian distribution. $\boldsymbol{p}$ is query data, and w is width of hash, b is the real number which is selected in an interval $[0, w]$.

In making search table phase, the k function $g =< h_1, \cdots, h_k >$ is applied for all data points, and the product set of each function's result is obtained. Furthermore, The l group of function $\{g_1, \cdots, g_l\}$ is also applied, and the search table is constructed.

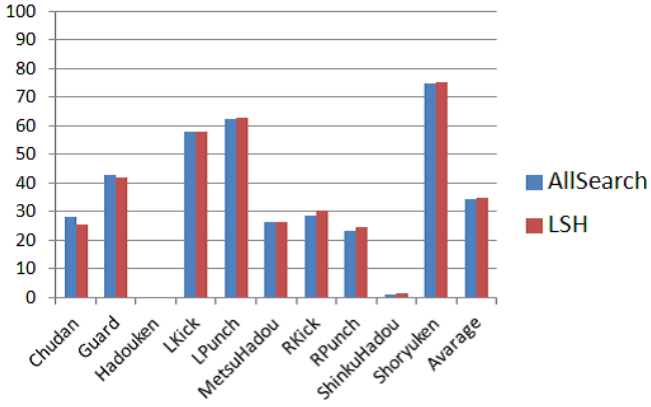

Fig. 3. Recognition Accuracy of 10×10 SOM[%]

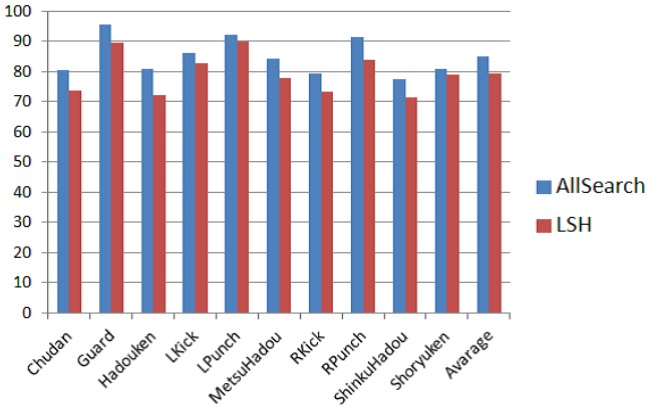

Fig. 4. Recognition Accuracy of 100×100 SOM[%]

In searching phase, when the query vector p is input, the hash value of this vector is calculated and matched product set is searched on each table. Then, the sum of sets of each search result is output as nearest neighbor slate points.

4 EXPERIMENTS

4.1 Preparation

We used the video game software "STREET FIGHTER 4" as the theme of early recognition. Ten kinds of gestures were selected for the experiment. The Kinect sensor captured each gesture at 30fps, which consists of 40-dimensional feature vector (2-dimensional position × 20 points) in each frame. Eight examinees played each gesture 30 times. Therefore, totally 2,400 (30×10×8) samples were collected. These samples were used for 6-fold cross validation in the following evaluation.

The parameters of LSH were set to k =64 and l =5. The size of SOM was changed as 10×10 and 100×100. The threshold to determine the recognition result was set to be 0.6. In other words, if the contribution ratio of the best matched posture exceeded the threshold, the system output

Table 1. Computational Time (sec)

	LSH	All Search
10×10	0.000043	0.00121
100×100	0.000059	0.1031

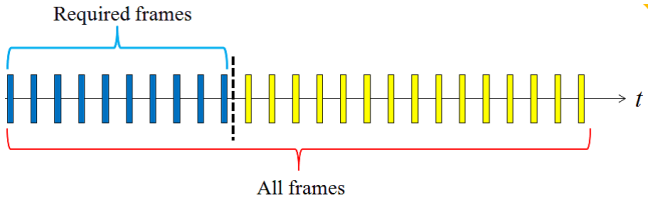

Fig. 5. Definition of Frame Ratio for Recognition (FRR)

the recognition result. In the test phase, if the computational time of each frame exceeded the video rate (i.e. 0.033sec), several frames were skipped (not processed) according to the time. For example, if it took about 0.066sec for each frame, every two frames were used for the evaluation.

4.2 Results

4.2.1 Computational Cost

Table 1 shows the average computational time for finding the best matched posture. In the case of greedy search (all search in the table), the computational time is proportional to the number of neurons. Therefore, the 100×100 size of SOM took about 100 times longer than the 10×10 size of SOM. If we use the 100×100 size of SOM, it will take about 0.1 sec (work at about 10fps). Therefore, every three frames will be used in the online recognition process.

On the other hand, LSH took about 0.00004∼0.00006 sec for each frame, and the computational time was almost constant even if the number of neurons increased. Therefore, we need not skip any frame if the LSH is used for searching the best matched posture.

4.2.2 Accuracy

The recognition accuracy is shown in Fig. 3 and Fig. 4. The horizontal axis shows the label of each gesture and most right is the average of all gestures. The vertical axis means the recognition ratio. We compared two approaches; a greedy search and a LSH-based search. In Fig. 3, the 10×10 size of SOM was used. As discussed above, the computational time was less than 0.033 sec so that any frame was not skipped, but the recognition accuracy was very low for several gesture classes. It makes no sense to reduce the computational time since it results in low accuracy of recognition result.

On the other hand, when the SOM had 100×100 neurons, every three frames were used in the greedy search. The ac-

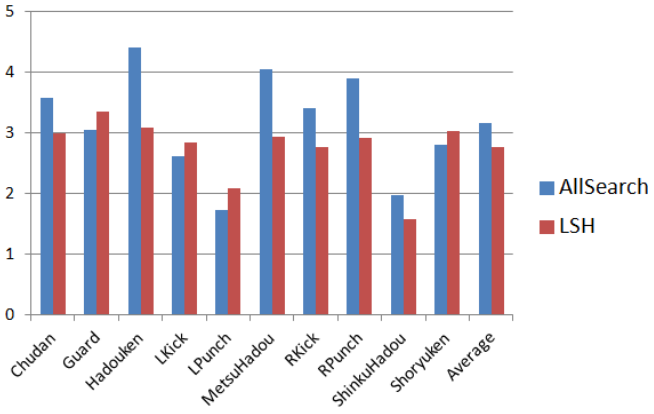

Fig. 6. Frame Ratio for Recognition [%]

curacy of both methods was almost the same. Strictly speaking, however, the accuracy of the greedy search was slightly more than the one of LSH even though the greedy search skipped several frames. We guess the reason as follows. The LSH-based search provides an approximate nearest neighbor search result. Therefore, the found posture is not always the best matched one, and if such a posture unfortunately has a high contribution ratio for a wrong gesture, a wrong label will be output from the system.

4.2.3 Response Speed

One of the most important evaluations is to measure the response speed of the system. In other words, how early the system could output the correct recognition result is evaluated. To measure the response speed, we define the frame ratio for recognition (FRR) as follows.

$$FRR = \frac{\text{\# of required frames}}{\text{\# of all frames}} \times 100 \qquad (2)$$

Fig. 5 supports the explanation of FRR. The FRR evaluates the percentage of the number of frames which are used for recognition to the all frames. Therefore, the smaller value of the FRR is desirable. The evaluation was performed when the system output the correct label only.

Fig. 6 shows the evaluation result of FRR. The FRR of LSH-based method was totally lower than the one of the greedy method. The greedy method skipped several frames in the online recognition process because of its computational time. That's why the FRR becomes higher than LSH-based method, which could investigate all of the input frames.

4.3 Discussion

Through the above experimental results, we found out some advantages of the proposed method. Firstly, the recognition accuracy of the proposed method did not become lower even it reduced the computational cost. Secondly, the response speed was faster than the traditional approach. Therefore, the proposed method realizes two important issues; high accuracy and quick response for input gesture patterns. Finally, the proposed method will work well even if the number of gesture classes increases since the computational cost is independent of the samples.

5 CONCLUSION

We have proposed an accelerated approach of early recognition of human gestures. Self-Organizing Map (SOM) is utilized for learning human gestures and obtaining the representative postures. When we investigate the postures which is the most similar to input posture, if searching all nodes of SOM in previous method is conducted, the computational cost becomes larger in proportion to the number of the representative postures. Moreover, if this phase takes longer time than a frame rate of input, the searching will drop the several frames and give bad effect on the result of recognition. Therefore, we used the Locality-Sensitive Hashing (LSH), which is a kind of approximate nearest neighbor searching, instead of previous method. Through the experiments, we could prove that the accuracy could be kept with much lower computational cost than the previous method. We are now researching the recognition which used the weighting to postures data to improve the recognition accuracy.

REFERENCES

[1] A. Mori, S. Uchida, R. Kurazume, R. Taniguchi, T. Hasegawa and H. Sakoe. "Early recognition and prediction of gestures". Proc. of International Conference on Pattern Recognition, vol. 3 of 4, pp.560-563, 2006.

[2] S. Uchida, K. Amamoto. "Early Recognition of Sequential Patterns by Classifier Combination". CD-ROM Proc. of International Conference on Pattern Recognition, 2008.

[3] Manabu Kawashima, Atsushi Shimada, Rin-ichiro Taniguchi : "Early Recognition of Gesture Patterns using Self-Organizing Map", The Fifth Joint Workshop on Machine Perception and Robotics (MPR2009)

[4] T. Kohonen. "Self-Organization and Associative Memory". Springer-Verlag, 1989.

[5] Teuvo Kohonen. "Self-Organizing Maps". Springer Series in Information Science, 1995.

[6] Indyk, Motwani. "Approximate nearest neighbor: towards removing the curse of dimensionality". STOC'98, section 4.2