2012 Third International Conference on Emerging Applications of Information Technology (EAIT)

Gesture Classification with Machine Learning using Kinect Sensor Data

Sambit Bhattacharya, Bogdan Czejdo and Nicolas Perez
Department of Mathematics and Computer Science
Fayetteville State University
Fayetteville, NC, USA
{sbhattac, bczejdo, nperez3}@uncfsu.edu

Abstract— **We present approaches for gesture classification and gesture segmentation by using machine learning on the Kinect sensor's data stream. Our work involved three phases. Firstly we developed gesture classification from a known vocabulary of gestures in an edited data stream. Secondly we extended those techniques to detect and classify a gesture in an unedited stream which also captures random movements. Thirdly, we apply rules to filter out movements that were not intentional gestures and yet resembled certain gestures in our vocabulary.**

***Keywords-** human gesture; machine learning; gesture classification; gesture recognition; gesture segmentation; Kinect sensor*

I. Introduction

Recognition and classification of human gestures is an active area of research since it is an important component of human-machine interaction. Most existing approaches use low level image processing operators to extract salient features from the moving human body in video data, which are then used to train machine learning algorithms [1,2]. Such feature extraction techniques pose a significant challenge and work under certain restrictive assumptions such as lighting conditions and even the hardware design of the video capturing device [1]. Gesture segmentation also remains a problem with multiple approaches described in the literature [2]. The Microsoft Kinect offers a hardware and software platform which solves some key computer vision problems with high levels of accuracy and well understood error characteristics and limitations [3]. In this paper we present our approach in solving the problem of gesture recognition by using machine learning on the Kinect's data stream of locations of joints on a human body, in other words skeletal data. Our approach consists of three parts. Firstly we develop gesture classification from a known vocabulary of gestures in an edited data stream. Secondly we extend those techniques to detect and classify a deliberate gesture in an unedited stream which also captures random movements. Thirdly, we apply rules to filter out movements that were not intentional gestures and yet resembled certain gestures in our vocabulary; in other words we developed techniques for both the gesture segmentation and classification problem in an unedited stream.

II. The Kinect Joint Data Stream and Gesture Vocabulary

A. The Kinect Sensor and its Data Streams

Here we describe in brief the main functions of Kinect sensor. We next describe how the Kinect data stream is read and how we modify the data organization for processing purposes.

The Kinect sensor can measure depth data (estimates of distance from Kinect to pixels in scenery) and it can identify and locate human skeleton. This skeleton is identified through positions of skeletal joints computed by Kinect from depth data. The Kinect processes and sends the skeletal data along with the depth data at the rate of 30 frames a second. It provides data streams for skeletal joints of up to two humans.

The precision of Kinect measurements and distortions was extensively studied in [3]. This experimental study reports that the random error of depth measurements increases quadratically with increasing distance from the sensor and the depth resolution also decreases quadratically with increasing distance from the sensor. It also measured the error bounds and resolutions limits and recommends that for mapping applications the data should be acquired within 1–3 m distance to the sensor. Our experiments were performed with sufficient variations within these limitations.

The Kinect sensor sends skeletal data to the computer as a temporal sequence of X, Y, Z coordinates of all 20 tracked joints. The data is grouped into time-stamped frames and the ith frame can be represented as $C_i^{1x}\ C_i^{1y}\ C_i^{1z} \ldots C_i^{20x}\ C_i^{20y}\ C_i^{20z}$, where C is a single coordinate value and the super-script has joint number followed by axis letter. When recording a data stream that starts at the 0th frame for which n+1 frames have been seen so far, we organize the values as $C_0^{1x}\ C_1^{1x}\ C_2^{1x} \ldots C_n^{1y}$ $C_0^{1y}\ C_1^{1y}\ C_2^{1y} \ldots C_n^{1y}\ C_0^{1z}\ C_1^{1z}\ C_2^{1z} \ldots C_n^{1z}\ C_0^{2x}\ C_1^{2x}\ C_2^{2x} \ldots C_n^{2x} \ldots$ *and so on* … We can further abbreviate this notation to $C_{0,n}^{1x}$ $C_{0,n}^{1y}\ C_{0,n}^{1z}\ C_{0,n}^{2x}\ C_{0,n}^{2y}\ C_{0,n}^{2z} \ldots\ C_{0,n}^{20x}\ C_{0,n}^{20y}\ C_{0,n}^{20z}$ where the subscript *0, n* means the sequence starting at *0* and ending at *n*. We additionally perform a data scaling step as recommended in [ref] for each C along the X, Y and Z directions for every joint as $C_{scaled} = (C_{original} - min) / (max - min)$ where max and min are the maximum and minimum values of that particular feature. For running the machine learning programs described next, we consider this sequence as a single feature vector where

This work was partially supported by National Science Foundation (award id: 0959958) and an internal grant award of Fayetteville State University's Integrated STEM Academic Success (ISAS) program.

n is appropriately chosen and some joints maybe deleted from the stream based upon necessity.

B. The Gesture Vocabulary

The aircraft gestures are based on aircraft marshaling gestures used in the military air force [4]. Let us start from the *liftoff* gesture. Both arms start down against the body in a rest position as shown in Figure 1A. The arms are swept up on their respective sides until they are parallel to the ground and stay like that until the corresponding maneuver is executed (Figure 1B). The *land* gesture is done in the reverse. Both arms start out to their respective sides of the body (Figure 1C), parallel to the ground. Then, the arms sweep downward and stop when they reach the side of the body.

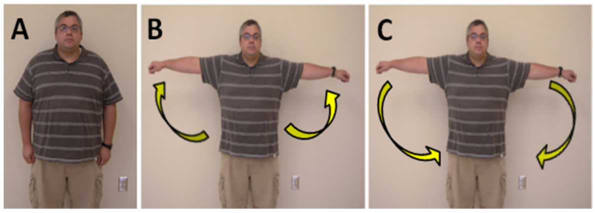

Figure 1. Start of liftoff (A). Ends (B, C) of liftoff and land.

In the *forward* gesture both arms start out in front of chest parallel to the floor, hands are flat. The hands are brought back towards chest while forming a triangle with finger tips forming the top vertex. The motion stops when hands contact chest. In the *left* gesture both arms start out to their respective sides, parallel to the ground. The right arm bends at the elbow and moves pointing to the left hand side of the body while keeping parallel to the ground. For the *stop* gesture both arms start down against the body in a rest position (Figure 1A). They are swept out to their respective sides, ending over the head where the hands meet (much higher than arms in Figure 1B). In the *tilt left* gesture both arms start out to their respective sides, parallel to the ground. The waist is bent, causing the upper body to tilt to its left side while holding the upper body position still. The remaining gestures *right* and *tilt right* can be described similarly.

It can be seen from the above description that recognizing sequence of gestures in real life situation can poses serious challenges for machine learning algorithms. The preparatory movement for the gesture can be mistaken and interpreted as an additional gesture that was actually not intended. For example, almost any gesture in our vocabulary requires the raising of hands to shoulder level, which can be interpreted as liftoff even though the intended gesture maybe different.

Let us first, therefore, solve the problem of a single gesture classification. It should also be noted that we use the following abbreviations of gesture names in the tables of this paper – FW for *forward*, LN for *land*, LF for *left*, LO for *liftoff*, RT for *right*, ST for *stop*, TL for *tilt left* and TR for *tilt right*.

III. Experimental Evaluation of Machine Learning on Edited Joint Data Stream

In this section we describe our experimental evaluation of three different algorithms for classification of human bodily motion into one of the classes in our gesture vocabulary. For the data presented in this section, the data stream was edited in two ways. Firstly, 7 upper body joints out of the total 20 which move during these specific set of gestures were kept. Secondly, the starting and ending frames of each gesture were marked by a human observer. For this reason, we call this the edited joint data stream. Thus in these group of experiments we are testing the accuracy of the machine learning algorithms where the input data for both training and testing are guaranteed to contain one of the known gestures from the vocabulary. In the next section we will describe our approach of solving the problem for unedited data.

A. Experimental results on machine learning

We chose a more contemporary and popular algorithm namely Support Vector Machine (SVM) and an earlier one, the Decision Tree (DT) as the two machine learning techniques to apply and compare. We observed excellent accuracies for both SVM and DT classification, with SVM performing better as shown in the confusion matrices of Table I and Table II. We also investigated the choice of kernel function for SVM by comparing the linear kernel with the radial basis function (RBF).

TABLE I. Confusion Matrix for 10-fold Crossvalidation, SVM With Linear Kernel, total accuracy = 99.97%

	FW	LN	LF	LO	RT	ST	TL	TR
FW	480	0	0	0	0	0	0	0
LN	0	460	0	0	0	0	0	0
LF	0	0	480	0	0	0	0	0
LO	0	0	0	480	0	0	0	0
RT	0	0	0	0	440	0	0	0
ST	0	0	0	1	0	399	0	0
TL	0	0	0	0	0	0	400	0
TR	0	0	0	0	0	0	0	380

TABLE II. Confusion Matrix for 10-fold Crossvalidation, Decision Tree, total accuracy = 99.32%

	FW	LN	LF	LO	RT	ST	TL	TR
FW	479	0	1	0	0	0	0	0
LN	0	459	0	0	0	0	1	0
LF	0	0	477	0	1	0	2	0
LO	0	0	2	475	0	3	0	0
RT	0	0	1	0	437	0	0	2
ST	0	0	0	2	0	398	0	0
TL	1	1	3	1	0	0	394	0
TR	0	0	1	0	2	0	0	377

For RBF we performed a grid search over the cost and gamma parameters to determine the best pair of values that would give the highest accuracy on crossvalidation [5], while in the case of the linear kernel the search for the best choice of only the cost parameter was executed much faster. After comparison, SVM-linear had the best accuracy followed by SVM-RBF and then DT. We thus choose SVM-linear over SVM-RBF due to two reasons – complexity of finding the best parameters and also due to general guidelines which state that the accuracy of a linear kernel is better than RBF when the data dimension is much higher than the sample size [7], as is true in our case.

B. Comparison with subject-wise crossvalidation

The choice of human subject generating data can determine performance of machine classifier. There is usually the need for analysis to understand the impact of different human subjects on machine learning outcomes. The results are best captured by subject-wise cross-validation accuracy that for our data is shown in Table III for various machine learning techniques for three different human subjects. The human subjects are numbered 1, 2, and 3. The table lists in each row two human subjects each producing a set of gestures taken as a training data set and the remaining human subject producing set of gestures taken as a testing data set. Columns two and three list various machine learning techniques accuracy for such selected learning and training data set. Let us consider the first row where human subjects 1 and 2 each are producing training data set and human subject 3 producing testing data set. Let us first consider comparison of machine learning techniques for classifiers based on SVM (linear kernel) and Decision Trees. We see clearly that SVM (linear kernel) is practically subject independent whereas Decision Tree was very sensitive to the choice of the subjects. The accuracy for SVM (linear kernel) was near perfect even though the training and testing was done on different subjects. The accuracy for Decision Trees decreased dramatically to 36% in the worst case, while being in the high 70% for the other cases, when the training and testing was done on different subjects. This row simple tells us that learning from both 1 and 3 subjects creates very specific and accidental learning process where the classifier based on Decision Trees tested using subject 3 data performed better.

TABLE III. ACCURACIES FOR SUBJECT-WISE CROSSVALIDATION

	SVM	DT
train = 1,2 test = 3	97%	36%
train = 1,3 test = 2	99%	77%
train = 3,2 test = 1	99%	79%

IV. EXPERIMENTAL EVALUATION OF MACHINE LEARNING ON UNEDITED JOINT DATA STREAM

We first describe what is meant by an unedited joint data stream. Given a stream $C_{0,n}^{1x}$ $C_{0,n}^{1y}$ $C_{0,n}^{1z}$ $C_{0,n}^{2x}$ $C_{0,n}^{2y}$ $C_{0,n}^{2z}\ldots$ $C_{0,n}^{20x}$ $C_{0,n}^{20y}$ $C_{0,n}^{20z}$ we do not have knowledge of where a gesture starts and ends within this stream. We only know that there is some *i, j* where $0 < i$ and $j < n$ for which a gesture found in our training vocabulary starts at *i* and ends at *j*. We present our approach for detecting this gesture embedded in the data stream and also classifying by gesture type. It should be noted that the recognition is done offline and for practical considerations like storage and processing time the stream is cropped to sequences. These sequences are still much larger than the length of the gestures and for each instance the person performing the gesture was instructed to make random bodily movements before the start and after the end of the gesture. As a result the data poses an inherent challenge as evidenced by work such as [9] which tries to recognize gestures in unsegmented video streams.

A. Cumulative sum of SVM probabilities

Support Vector Machines were extended to return the probability estimates along with classification by [8] which is implemented in the LIBSVM library [6]. This work extended the pairwise coupling method of multi-class classification that combines all comparisons for each pair of classes to generate class probabilities. These probability values are an improvement over other similar methods like voting. We next describe the use of these probability estimates in detecting a gesture in an unedited data stream. We created a sliding window based on an estimate of time length of gesture and passed it over the unedited data stream. This process generates one probability estimate per gesture for each position of the sliding window. In our case with eight gestures, eight probability values are generated at each position. Next we compute the cumulative sum of these probability values, and decide the predicted gesture as the gesture with maximum cumulative sum.

We executed this procedure on variety of data streams. The first group of data streams contained a gesture or a gesture sequence with minimum accidental movement of the hands. The second group of data streams contained in addition to intended gesture or a lot of accidental movement of the hands making the data intentionally very “noisy”. The first group generally was recognized with very highly accuracy. Let us concentrate in this paper on very “noisy” data streams examples of which are shown in Figures 2 and 3. The intended gestures in very “noisy” data streams was recognized with much lower accuracy of 46% as shown in Table IV.

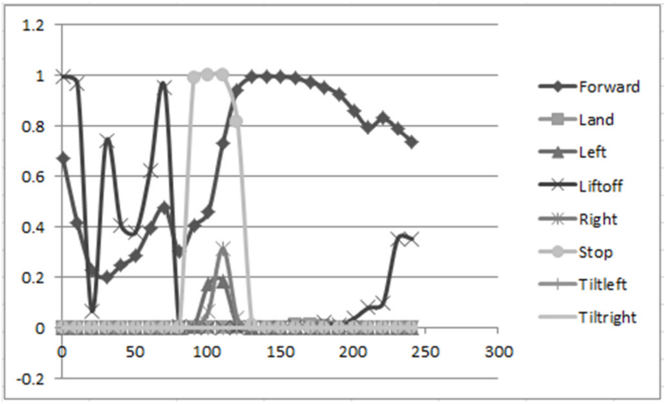

Figure 2. Probabilities for unedited stream, example 1

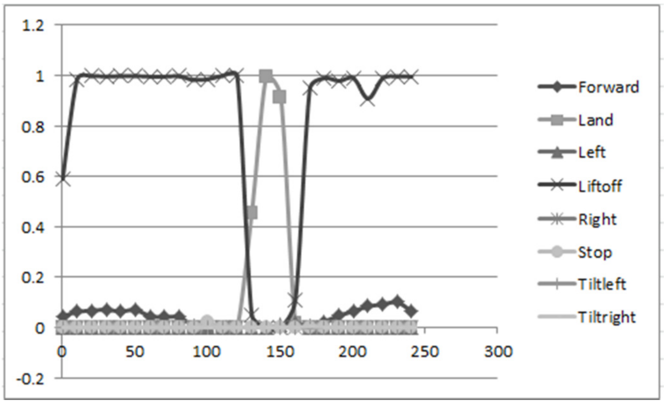

Figure 3. Probability for unedited stream, example 2

TABLE IV. CONFUSION MATRIX FOR UNEDITED DATA STREAM, SVM, NO RULE APPLIED, TOTAL ACCURACY = 46%

	FW	LN	LF	LO	RT	ST	TL	TR
FW	11	0	0	2	0	2	0	0
LN	1	2	0	12	0	0	0	0
LF	0	0	5	10	0	0	0	0
LO	0	0	0	15	0	0	0	0
RT	0	0	0	10	5	0	0	0
ST	0	0	0	11	0	4	0	0
TL	0	0	0	6	0	0	4	0
TR	0	0	0	5	0	0	0	5

B. Improvement of Gesture Recognition by Applying Rules

As we discussed before, the main challenge for proper gesture recognition by machine is an unintended gesture in the data stream especially in very "noisy" data streams. This may happen because the person, in order to take the initial pose, may have to pass through motions that are similar to a gesture existing in the vocabulary (see *liftoff* gesture discussed in Section 2).

We have developed, therefore, the set of rules to improve the automatic recognition of gestures in a data stream. One of the rules was that if the *liftoff* gesture (recognized in the stream of data) is followed by any other gesture with comparably high cumulative probability value then only the gesture that followed the *liftoff* would be returned as intended. Obviously, both the *liftoff* gesture and the following gesture would need to have a high cumulative sum of SVM probabilities. The proper threshold needed to be chosen correctly through data analysis to minimize the confusion values in the confusion matrix. The effect of the rule discussed above is shown in Table V.

CONCLUSIONS

In this paper we described techniques of machine learning for gesture classification as applied to aircraft marshaling. The characteristic features of our research are as follows. We have used the Kinect sensor's joint coordinates data stream as the feature describing the moving human body in video data. We used machine learning methods and chose one (SVM, linear kernel) that is best in accuracy. Our requirements were not only the high classification accuracy, but also high subject-wise cross-validation and availability of the probability function providing information about how likely the data sub-stream can be interpreted as a given gesture. We have also studied various techniques to improve gesture sequence analysis to filter out unintentional gestures.

TABLE V. CONFUSION MATRIX FOR UNEDITED DATA STREAM, SVM, WITH APPLICATION OF RULE, TOTAL ACCURACY = 83.33%

	FW	LN	LF	LO	RT	ST	TL	TR
FW	16	0	0	2	0	2	0	0
LN	1	19	0	0	0	0	0	0
LF	0	0	14	6	0	0	0	0
LO	0	0	0	20	0	0	0	0
RT	0	0	0	5	15	0	0	0
ST	0	0	0	9	0	11	0	0
TL	0	0	0	0	0	0	15	0
TR	0	0	0	0	0	0	0	15

The described advancements in recognition and classification of human gestures will contribute to building a model for systems supporting human-machine interactions based on human gestures and body movement.

REFERENCES

[1] Mitra, S. and T. Acharya (2007). "Gesture recognition: A survey." IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews 37(3): 311-324.

[2] Weinland, D., R. Ronfard, et al. (2011). "A survey of vision-based methods for action representation, segmentation and recognition." Computer Vision and Image Understanding 115(2): 224-241.

[3] Khoshelham, K. and S. J. Oude Elberink (2012). "Accuracy and resolution of Kinect depth data for indoor mapping applications." Sensors 12(2): 1437-1454.

[4] Wikipedia article. Retrieved June 29, 2012, from http://en.wikipedia.org/wiki/Aircraft_marshalling.

[5] Chih-Wei Hsu, C.-C. C., and Chih-Jen Lin. "A Practical Guide to Support Vector Classification." Retrieved June 29, 2012, from http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[6] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[7] Ben-Hur, A. and J. Weston (2010). "A User's Guide to Support Vector Machines Data Mining Techniques for the Life Sciences." **609**: 223-239.

[8] T. F. Wu, C. J. L., and R. C. Weng. (2004). "Probability estimates for multi-class classification by pairwise coupling." Journal of Machine Learning Research 5: 975-1005.

[9] Morency, L. P., A. Quattoni, et al. (2007). Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on.