RECOGNIZING USER INTERFACE CONTROL GESTURES FROM ACCELERATION DATA USING TIME SERIES TEMPLATES

Pekka Siirtola, Perttu Laurinen, Heli Koskimäki and Juha Röning
Intelligent Systems Group, P.O. BOX 4500, FI-90014, University of Oulu, Finland

Keywords: Gesture recognition, Accelerometer, Template-based matching.

Abstract: This study presents a method for recognizing six predefined gestures using data collected with a wrist-worn triaxial accelerometer. The aim of the study is to design a gesture recognition-based control system for a simple user interface. The recognition is done by matching the shapes that user's movements cause to acceleration signals to predefined time series templates describing gestures. In this study matching is done by using three different trajectory distance measures, the results show that the weighted double fold gives the best results. The superiority of this distance measure was shown using a statistical significance test. A user-dependent version of the method recognizes gestures with accuracy of 94.3% and a recognition rate of the user-independent version is 85.5%. This work was supported by the EU 6th Framework Program Project XPRESS.

1 INTRODUCTION AND RELATED WORK

In some situations gesture recognition is a good option for handling human-computer interaction because it enables natural interaction and no input devices, such as a keyboard and a mouse, are needed. In fact, in recent years gesture recognition systems have become more widely known among the public as new products controlled by gestures have become available. For instance gesture-controlled game consoles have recently appeared in stores.

This work studies the recognition of six gestures: *punch - pull*, *pull - punch*, *left - right*, *right - left*, *up - down* and *down - up*. These gestures were selected for this study because the future purpose of the gesture recognition system is to control a simple user interface. The interface view is a table and each cell of the table is a button. Using gestures, the user can decide which button to push. All the gestures selected for this study include two phases, action and counter-action, because it is natural for a human to return the hand to the original position after each performed gesture. Moreover, the gestures of this study were selected so that they can be performed by moving hand along one out of three coordinate axis so gestures contain movement mainly in one dimension, though the data is tridimensional. Therefore, for each gesture, two out of three acceleration channels are considered useless and are removed in order to improve the recognition rates.

Mainly two different types of methods have been used to recognize gestures: template-based methods and HMM methods. However, in (Ko et al., 2008) it is shown that gestures can be recognized more accurately using templates than by using HMM. Several template-based gesture recognition systems are proposed in the literature. In (Corradini, 2001) dynamic time warping (DTW) was used to recognize a small gesture vocabulary from offline data. The study did not use body-worn sensors, instead the system was trained with video sequences of gestures. A recognition accuracy of 92% was attained when five gestures such as stopping and waving were recognized.

In (Stiefmeier and Roggen, 2007) gesture signals were transformed into strings to make similarity calculations faster and real-time. The study used several inertial sensors: the sensors were attached to the lower arms, upper arms and the torso of the body. Human motion was presented by strings of symbols, and by combining the data provided by different sensors, the relative position of the arms with respect to the torso was computed. The method was demonstrated by spotting five predefined gestures from a bicycle maintenance task. An average classification rate of 82.7% was achieved when the method was tested with three persons.

Methods similar to those in our study were used in (Ko et al., 2008). The study used two wrist-worn

Siirtola P., Laurinen P., Koskimäki H. and Röning J. (2010).
RECOGNIZING USER INTERFACE CONTROL GESTURES FROM ACCELERATION DATA USING TIME SERIES TEMPLATES.
In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 176-182
DOI: 10.5220/0002899601760182

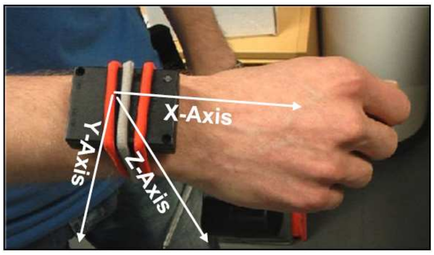

Figure 1: Accelerometer attached to the user's active wrist.

accelerometers, one on each wrist, and DTW as a distance measure. The frequency of the accelerometers was 150 Hz. Because DTW was used, the parameters for endpoint detection had to be defined by hand or by using a complex automated way. This made the DTW approach less generic. To make recognition faster, Ko *et al.* transformed the signal into a more compact representation by sliding a window of 50 samples with a 30-sample overlap through the signal. This way the number of points was reduced, making the system faster but at the same time making the system less sensitive to fast changes in the signal. Therefore, if fast movement is an important part of the gesture, this can cause problems. In the study 12 gestures of a cricket umpire, performed by four actors, were recognized. The system was tested using many different settings, for example offline and online. Each actor performed each gesture only once, and these data were used for testing, so the total number of gestures in the test data was only 48. The accuracy of the system was 93.75% when recognition was done using one template per gesture, as was done in our study, also.

The paper is organized as follows: Section 2 describes sensors and data sets. Section 3 introduces the techniques and gestures used in this study. Section 4 evaluates the performance and accuracy of the proposed method with the data sets presented in Section 2. Finally, conclusions are discussed in Section 5.

2 DATA SET

The data were collected using a mobile device equipped with a 3D accelerometer, 3D gyroscope, 3D magnetometer and two proximity sensors. In this study only accelerometers were used and the measuring device was attached to the active wrist of the user, see Figure 1. The sampling frequency of the accelerometer was 100Hz.

The data were collected from seven persons. Two separate *gesture data* sets were collected from each person: a training data set that included five repetitions each of six gestures and a test set that included ten repetitions of each gesture. These data sets were used to test how well the presented method detects the performed gestures from continuous data streams.

In addition, a *performance data* set around 30 minutes long that does not include any gestures was also collected from each person. This data set included other activities such as walking and working. This data set was used to test the speed and accuracy of the gesture recognition method. Accuracy was tested with this data set by testing how many false positive results the system found from a signal that did not include any predefined gestures.

3 METHODS

The purpose of the proposed method is to find predefined gestures from continuous accelerometer data streams. Basically, the system compares the shapes of studied signals with the shapes of template patterns describing gestures the system is trained to recognize. If the shape of the studied gesture is similar to the shape of some template, we know which gesture is performed. The quality of the proposed method depends mostly on four things: the quality of the templates, the accuracy of the similarity measure, selection of a proper similarity limit and the goodness of the sliding method. Of course, pre-processing also has its own important role.

3.1 Data Pre-processing

The raw acceleration data were pre-processed by first smoothing and then compressing them.

Smoothing was done using moving average (MA) filter and same weight were given to each point.This way the number of disturbances could be reduced and the signal became smoother and easier to handle.

After the smoothing, the signals were compressed in order to speed up calculations. The data were compressed so that they contained points of the original data where the derivative is equal to zero. Nevertheless no more than m sequential points were allowed to be removed from the original data. Therefore, if the number of points between two sequential derivative points was r and $r > m$, $r, m \in \mathbb{Z}_+$ then $\lfloor r/m \rfloor$ points, located at equidistant intervals, were also included in the compressed signal, see Figure 2.

3.2 Choosing Time Series Templates

The gestures of the study includes two phases, action and counter action. The use of gestures consisting of only one phase seemed to confuse users and the recognition system, because users tend to move their

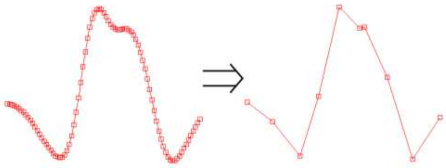

Figure 2: Template and a compressed version of it.

hand back to its original position. So, if both the action and counter-action are predefined as gestures that the system is trying to recognize, users easily accidentally perform two gestures instead of one. Selecting the gestures so that they contain an action and a counter-action solves this problem.

All six gestures of the study were selected so that movement is performed along only one out of three coordinate axis and thus only the data of this acceleration channels is needed in recognition, see Figure 3. So, the data given by two other channels is not important and it can be considered that it mostly consist disturbances, white noise and other non-valid information and therefore these channels are not used in recognition. Thereby the gestures and gesture templates are one-dimensional but the data are tri-dimensional and therefore templates are only needed to slid through one acceleration channel. The sensor was attached to the wrist so that the templates of the gestures *punch* - *pull* and *pull* - *punch* are slid through the x-axis accelerometer data, because these gestures cause mainly x-axis movement, see Figure 1. Correspondingly, *left* - *right* and *right* - *left* are slid through the y-axis data and *up* - *down* and *down* - *up* through the z-axis data. Elimination of two acceleration channels makes gesture recognition not only more accurate but also faster, because similarity calculation is faster from the one-dimensional acceleration signal than from the tri-dimensional signal.

3.2.1 User-dependent Case

In the user-dependent case, a *class template*, which is a template that is used to recognize a certain gesture, was selected for each gesture using a training data set. The class templates for each gesture were labeled from the training data set and they were used as training templates. Among these training templates, one at a time was selected as a candidate class template and used to recognize other training templates. As a class template describing gesture A was selected candidate class template $P_{A,i}$ which minimizes the sum

$$\sum_{j=1}^{n} d(P_{A,i}, P_{A,j}), \qquad (1)$$

when $1 \leq i \leq n$ and n is the total number of training templates of class A, $P_{A,j}$ is a training template of gesture A and $d(\cdot, \circ)$ is some similarity measure.

3.2.2 User-independent Case

A user-independent version of the presented gesture recognition system was tested using gesture templates selected in three different ways. The first two were suggested by (Ko et al., 2008).

Minimum Selection. In the case of minimum selection a class template describing gesture A was selected using Equation 1. In the user-dependent case the training and test data sets were performed by the same person, but in the user-independent case Equation 1 was applied to the training template set extracted from six persons. One person was left out as a test person.

Average Selection. Average selection was also done using Equation 1. Now the data of six persons were also used for training and the data of one person were left out for testing. Equation 1 was performed separately for each of the six training data sets to find six templates that have minimum inter-class distances, and the resultant six class templates were combined as one average template using the method presented in (Gupta et al., 1996). The method was used, though in (Niennattrakul and Ratanamahatana, 2007) it is claimed that the method does not produce the real average of two templates. Still, this DTW-based method works really well, giving a good estimation of the average template of two templates, and no better averaging methods seem to be available.

Evolutionary Selection. Evolutionary selection of a class template was done using a slightly modified version of the algorithm presented in (Siirtola et al., 2009). This evolutionary algorithm produces an optimal template describing some periodic time series. In this case the training data sets of six persons were fused so that the training gestures of each gesture A were combined as a periodic time series. This time series was given as an input to the algorithm presented in (Siirtola et al., 2009), and using it an optimal template describing the periods was found. The purpose of the algorithm is to find a template P that maximizes the fitness function

$$f(P) = \frac{\text{Number of found gestures using } P}{\text{Correct number of gestures}}. \qquad (2)$$

Template P which maximizes this function was selected as the class template.

3.3 Sliding and Decision Making

The purpose of sliding is to find every shape of time series T that is similar to class template P. In the case

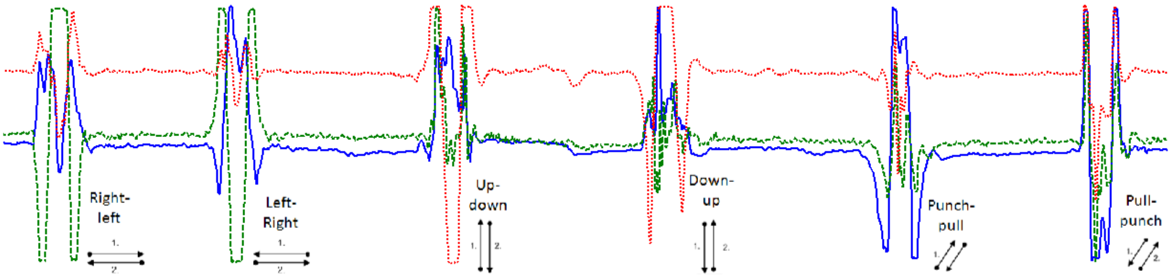

Figure 3: Gestures and corresponding tri-axial acceleration.

of online recognition, the functioning of the sliding method is in an important role because the starting point and ending point of the performed gesture is not known in advance. This means that if sliding method cannot find these points, it is not possible to recognize the gestures, either. In this study sliding method presented in (Siirtola et al., 2009) was used.

If more than one template P_i is found similar to some subsignal S, then the class of template P_i for which the ratio $\frac{d(S_k,P_i)}{\delta_i}$ is the smallest, where δ_i is predefined similarity limit for template P_i, is considered as a class of subsignal S_k. Note that different templates can have different similarity limits because some gestures are more difficult to perform and recognize than others.

4 EXPERIMENTS

4.1 Gesture Data

The gesture data presented in Section 2 were tested in two cases: a user-dependent case where the gesture recognition method was trained and tested with the same person's data, and a user-independent case where the data of the test person were not used in training.

4.1.1 User-dependent Case

User-dependent version of the method was tested using three different distance measures: *weighted double fold* (WDF) distance measure (Siirtola et al., 2008), *double fold* (DF) (Laurinen et al., 2006) and DTW. Also two different point-to-point distance measures were tested, Euclidean distance (ED) and Chebychev distance (CD).

The results (see Table 1) show that the combination of WDF and ED produces the highest total recognition accuracy; on average 94.3% of the gestures were recognized correctly. In fact, this combination gave the best recognition rates for six out of seven test persons. It seems that user-dependent version is very reliable because the gestures of every person can be recognized with an accuracy of at least 90%. When DTW and ED are used, the total recognition rate is 4.3 percentage units smaller. According to paired t-test with 6-degrees of freedom and $p = 0.95$ this improvement is statistically significant.

Note that the recognition rates drop when CD is used instead of ED as a point-to-point distance measure. The results show that using Chebychev distance and DTW or WDF, the gestures of some persons can be recognized with very high accuracy but the gestures of other persons seem to be difficult to recognize. For instance, using WDF the difference between the highest and lowest rates is almost 40 percentage units. CD considers only one dimension relevant, but the results show that by considering both dimensions relevant, as is done in the case of ED, better recognition rates are gained.

The good results using WDF came as no surprise since WDF is specially designed to measure the similarity of sparse signals, where the data points of the signals are not distributed at equal-length intervals (Siirtola et al., 2008). Compression presented in Section 3.1 produces such sparse signals.

4.1.2 User-independent Case

In the user-independent case a combination of WDF and ED was used as a distance measure because the results of Table 1 show that this combination gives the highest recognition rates.

Three different ways of choosing class templates for user-independent gesture recognition were introduced in Section 3.2.2. These methods were compared and the results are given in Table 2.

The highest recognition accuracy of 85.5% was achieved by using evolutionary selection. This template choosing method produced the best recognition results for five out of seven test persons. Based on these results it can be seen that the proposed method can be used for reliable user-independent gesture recognition. The other two methods seem to be almost equally accurate between themselves by recognizing gestures with an accuracy around 82%.

Table 1: Recognition accuracy in a user-dependent case. Comparison of local distance measures and similarity measures.

Measure / Test person	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Total
DTW + ED	95.0%	95.0%	93.3%	83.3%	88.3%	93.3%	81.7%	90.0%
DTW + CD	90.0%	91.7%	95.0%	86.7%	60.0%	91.3%	90.0%	86.4%
WDF + ED	95.0%	96.7%	98.3%	90.0%	90.0%	98.3%	91.7%	**94.3%**
WDF + CD	96.7%	65.0%	91.7%	71.7%	58.3%	96.6%	81.7%	88.6%
DF + ED	95.0%	91.7%	96.7%	78.3%	88.3%	85.0%	81.7%	88.1%
DF + CD	75.0%	88.3%	86.7%	60.0%	66.7%	70.0%	78.3%	74.6%

Table 2: Recognition accuracy in a user-independent case using different template choosing methods. MS = Minimum selection, AS = Average selection, ES = Evolutionary selection.

Method / Test person	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Total
MS	91.7%	81.7%	91.7%	85.0%	85.0%	58.3%	81.7%	82.1%
AS	100.0%	81.7%	91.7%	85.0%	85.0%	60.0%	76.7%	82.9%
ES	100.0%	83.3%	90.0%	80.0%	90.0%	68.3%	86.7%	**85.5%**

Table 3: User-independent recognition results using the evolutionary template selection method.

Gesture / Test person	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Total
Punch-Pull	100.0%	90.0%	90.0%	70.0%	70.0%	80.0%	80.0%	82.6%
Pull-Punch	100.0%	90.0%	90.0%	60.0%	90.0%	80.0%	90.0%	85.7%
Right-Left	100.0%	70.0%	100.0%	100.0%	80.0%	50.0%	90.0%	84.3%
Left-Right	100.0%	100.0%	60.0%	100.0%	100.0%	50.0%	70.0%	82.6%
Up-Down	100.0%	80.0%	100.0%	70.0%	100.0%	80.0%	100.0%	90.0%
Down-Up	100.0%	70.0%	100.0%	80.0%	100.0%	70.0%	90.0%	87.1%
Total	100.0%	83.3%	90.0%	80.0%	90.0%	68.3%	86.7%	85.5%

When the results of the best methods of the user-dependent and -independent versions are compared, it can be seen that in most cases the user-independent version using evolutionary template selection gave around 10 percentage units worse results than the user-dependent version using WDF and ED. Still, the gestures of every person were recognized with high accuracy using evolutionary selection: the recognition rates for the gestures of person 1 were in fact better using the user-independent version. The only difference was person 6, whose gestures were recognized user-independently with an accuracy of only 68.3%. Using user-dependent templates, the gestures of person 6 were recognized almost perfectly, at a rate of 98.3%. Therefore, the problem is not that the gestures of the test data of person 6 were of low quality and impossible to recognize. One explanation for the weak user-independent recognition results is that person 6 had his/her own personal way of performing the gestures; person 6 especially seemed to perform the left-right and right-left gestures differently than the others. These gestures were recognized with an the accuracy of only 50%, see Table 3. Because persons seem to have at least two different ways of performing gestures, it could be wise to choose at least two templates per gesture, and not just one as was done in this study, to make user-independent gesture recognition more reliable.

4.2 Performance Test Data

Performance test data were collected to test the performance and accuracy of the gesture recognition system. These data did not include any of the six gestures and therefore all the detected gestures could be considered as false positive.

The gesture recognition system was tested using a Pentium D (3GHz, 2GByte RAM)) powered computer, and the results presented in Table 4 show that the running time of the presented method was about 15.0% of the duration of the performance test data sequences. This means the system is over six times

Table 4: Performance and accuracy of the method.

Person	Duration of performance data	CPU time for template matching	False positive results
1	1732s	306s	0
2	1672s	296s	2
3	1604s	340s	0
4	1557s	358s	3
5	1609s	218s	0
6	1791s	297s	5
7	1609s	206s	0
Total	11574s	1745s	10

faster than real-time, without any optimization, therefore the method can be used online.

A gesture recognition system is not allowed to produce false positive results often, because it would make the user-interface very frustrating to use. Table 4 also shows that the method is very accurate, meaning that it very seldom produced false positive results. The test sequences were all together over three hours long and the number of false positive results was only 10. So, on average, the proposed method produced one false positive result per 20 minutes.

5 CONCLUSIONS

This article presented a gesture recognition method for recognizing six predefined gestures. The method is based on template matching and the results show that it can recognize gestures very accurately and in real time. Three different distance measures were tested and the best results were achieved using weighted double fold distance measure. A user-dependent version of the system can recognize gestures with an accuracy of 94.3% when WDF distance measure is used. It was also shown that the improvement gained using WDF is statistically significant. User-independent version of the method can rocognize gestures with an accuracy of 85.5%. Compared with other studies, the recognition rates are really competitive. Most other studies use more than one sensor, unlike this study, and therefore the achieved results can be considered state-of-the-art.

The presented method works really well. It seldom produces false positive results and can recognize gestures with high accuracy. Still, the accuracy of the user-independent version could be improved by choosing more class templates, because people seem to have at least two different ways of performing gestures. Now only one template per gesture was used. The problem is that this would of course make the system slower.

The presented gesture recognition system is designed to control a simple user interface, and the next task is to fuse the gesture recognition system and the interface together.

ACKNOWLEDGEMENTS

This study was carried out with financial support from the Sixth Framework Programme of the European Community for research, technological development and demonstration activities in an XPRESS (FleXible Production Experts for reconfigurable aSSembly technology) project. It does not necessarily reflect the Commission’s views and in no way anticipates the Commission’s future policy in this area.

Pekka Siirtola would like to thank GETA (The Graduate School in Electronics, Telecommunications and Automation) for financial support.

REFERENCES

Corradini, A. (2001). Dynamic time warping for off-line recognition of a small gesture vocabulary. In *RATFG-RTS ’01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 82, Washington, DC, USA. IEEE Computer Society.

Gupta, L., Molfese, D., Tammana, R., and Simos, P. (1996). Nonlinear alignment and averaging for estimating the evoked potential. *Biomedical Engineering, IEEE Transactions on*, 43(4):348–356.

Ko, M., West, G., Venkatesh, S., and Kumar, M. (2008). Using dynamic time warping for online temporal fusion in multisensor systems. *Inf. Fusion*, 9(3):370–388.

Laurinen, P., Siirtola, P., and Röning, J. (2006). Efficient algorithm for calculating similarity between trajectories containing an increasing dimension. pages 392–399. Proc. 24th IASTED international conference on Artificial intelligence and applications, February 13 - 16, Innsbruck, Austria.

Niennattrakul, V. and Ratanamahatana, C. (2007). Inaccuracies of shape averaging method using dynamic time warping for time series data. In *ICCS ’07: Proceedings of the 7th international conference on Computational Science, Part I*, pages 513–520, Berlin, Heidelberg. Springer-Verlag.

Siirtola, P., Laurinen, P., and Röning, J. (2008). A weighted distance measure for calculating the similarity of sparsely distributed trajectories. In *ICMLA’08: Proceedings of the Seventh International Conference on Machine Learning and Applications*.

Siirtola, P., Laurinen, P., and Röning, J. (2009). Mining an optimal prototype from a periodic time series: an evolutionary computation-based approach. In

Congress on Evolutionary Computation (CEC 2009), pages 2818–2824.

Stiefmeier, T. and Roggen, D. (2007). Gestures are strings: Efficient online gesture spotting and classification using string matching. In *In: Proceedings of 2nd International Conference on Body Area Networks (BodyNets)*.