Online Context Recognition in Multisensor Systems using Dynamic Time Warping

Ming Hsiao Ko [1], Geoff West [1], Svetha Venkatesh [1], and Mohan Kumar [2]
[1] *Department of Computing, Curtin University of Technology*
Perth, WA, Australia, Email: {komh, geoff, svetha}@cs.curtin.edu.au
[2] *Department of Computer Science and Engineering, The University of Texas at Arlington*
Box 19015, Arlington TX 76120, Email: kumar@cse.uta.edu

Abstract

In this paper, we present our system for online context recognition of multimodal sequences acquired from multiple sensors. The system uses Dynamic Time Warping (DTW) to recognize multimodal sequences of different lengths, embedded in continuous data streams. We evaluate the performance of our system on two real world datasets: 1) accelerometer data acquired from performing two hand gestures and 2) NOKIA's benchmark dataset for context recognition. The results from both datasets demonstrate that the system can perform online context recognition efficiently and achieve high recognition accuracy.

1. INTRODUCTION

Many computers are embedded in the world around us. These ubiquitous computers will anticipate our needs and sometimes act on our behalf. Sensor processing represents this paradigm shift in computing. By analysing the data acquired from multiple sensors, we can augment computers with awareness of their environment and situation. The process for extracting, fusing and converting relevant data from multiple sensors to a representation for situation awareness is referred to as *context recognition*. Context recognition is a challenging task since the situations are discribed by multimodal sequences acquired from different sensors over a period of time. In this paper we present our approach for online context recognition in multisensor systems.

The core of our system is the *Dynamic Time Warping* (DTW) recognizer. The DTW non–linearly warps one time sequence to match another given start and end points correspondence. First, the DTW is extended to deal with multimodal sequences acquired from multiple sensors, usually a mix of binary, discrete, and continuous variables. The DTW recognizer can recognize multimodal sequences of different lengths e.g. the same human gestures performed at different speeds, or the same batch processes of different durations. Second, we show how our system deals with unknown endpoints of multimodal sequences that are embedded in continuous data streams, for which accurate location of the endpoints is critical for reliable and robust context recognition using DTW. The endpoint detection problem is nontrivial and has been studied thoroughly in the speech recognition community, where "silence" between utterances is a useful clue [1], [2]. However, it is more difficult to develop endpoint detection algorithms for multimodal sequences, since there is no "silence" region as can be found before and after a spoken word. At any point in time, there might be some low level events detected from some sensors. We demonstrate that our system can perform online context recognition efficiently, and copes with the sheer magnitude of multi–sensory information in real–time and achieve high recognition accuracy.

The rest of the paper is organized as follows: In section 2, we review dynamic time warping. In section 3, we describe the architecture of our online context recognition system. In section 4, we evaluate the recognition accuracy and speed of the system with two real world datasets. Finally, conclusions and further work are discussed in section 5.

2. DYNAMIC TIME WARPING

A general time alignment and similarity measure for two temporal sequences is Dynamic Time Warping (DTW), which was introduced by Sakoe and Chiba [3]. DTW has been extended to deal with unknown start and end points of isolated words in speech [4], [5], and connected word recognition [6], [7], [8]. More recent research on DTW has focused on applying it to mining patterns from one–dimensional time series [9], and indexing and clustering one–dimensional time series [10], [11].

To the best of our knowledge, there has been little work on applying DTW for context recognition from sensor data. Other techniques usually used are *hidden Markov modelling* (HMM) [12], artificial neural networks [13], and self–organizing maps [14]. However, the training and recognition procedures in DTW are potentially much simpler and faster. In fact, it has been demonstrated that DTW recognition is fast for most of the time series databases available [15].

The classic DTW algorithm uses a local distance measure to determine the distance between a class sequence and a test sequence by calculating a warping path. Suppose we have a class sequence $(C(i))_{i=1}^{I}$ of length I and a test sequence $(T(j))_{j=1}^{J}$ of length J. To calculate the similarity between these two sequences, a local distance measure $d(C(i), T(j))$ between two points of these sequences is applied to calculate a warping path on an I by J matrix. A warping path W is a set of matrix elements that defines a mapping between C and T,

$$W = \left\{ w(i(q), j(q)) \;\middle|\; \begin{array}{l} q = 1, \ldots, Q, \\ \max(I, J) \leq Q \leq I + J - 1 \end{array} \right\} \quad (2.1)$$

with $i(q) \in \{1, \ldots, I\}$ and $j(q) \in \{1, \ldots, J\}$. The warping path is typically subject to several constraints: *Continuity*, *Endpoint constraints*, and *Monotonicity* [3].

The overall distance $DTW(C,T)$ between the class sequence and the test sequence is then calculated by summing the local distances over the warping path W.

$$DTW(C,T) = \arg\min_{W} \left(\sum_{q=1}^{Q} d(i(q), j(q)) \Big/ k \right) \quad (2.2)$$

where k is the normalisation factor. The warping path can be found very efficiently using dynamic programming, more information on DTW can be found in [3].

3. METHODS

A. System architecture

Fig. 1 shows the architecture of the proposed system for online context recognition of multimodal sequences. The input to the system

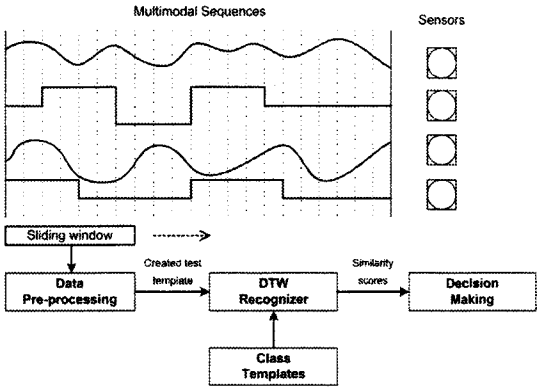

Fig. 1: The architecture of online context recognition system.

is the raw data (raw multimodal sequences acquired from sensors) that is currently buffered in the sliding window. First, the data pre–processing module converts the raw data buffered in the sliding window into the *test template* (data–reduced version of the raw multimodal sequences). This conversion process may include: *filtering*, *normalising variables*, and *feature extraction*. We have studied the influence of these data pre–processing techniques for recognition accuracy in our previous work [16]. Second, the DTW recognizer calculates the DTW distances between the class templates and the converted test template, and passes the results to the decision making procedure. Finally, the output of the system is the decision made by decision making procedure on which class template is the most similar one to the input.

B. The training of the DTW recognizer

In the training stage, the most important task is the creation of reliable class templates for each class of multimodal sequences to be recognised. Although the template can be selected randomly (i.e. *random selection*), the accuracy of the DTW–based recognition system greatly relies on the quality of the created class templates. Some better methods than random selection are:

1) *Normal selection*: The normal procedure for selecting the class templates is to use each example as a template and determine its recognition rate when classifying the other examples of that class. The example with the best recognition rate is chosen as the template.
2) *Minimum selection*: Select the class template using the *intra–class DTW distance*, that is the sum of DTW distances between the example that is the template to all other examples within the same class. The one with *minimum intra–class DTW distance* is selected as the class template. In terms of clustering , the template selected by this method is the centre of the cluster.
3) *Average selection*: Generate the class template from a set of the best templates, e.g. extract the five templates that have the best minimum inter–class DTW distances, then take the average of these to produce the final class template. Averaging a collection of time series that are not perfectly time–aligned is non–trivial [15], [17]. It has been demonstrated [17] that the speaker–dependent recognition rate is improved from 85.3%, using the normal selection, to 99%, using the average selection.
4) *Multiple selections*: Use several class templates for each class, determine the classification for each template and then combine the results. This method usually achieves better classification rates than the aforementioned methods. However, this method is computationally inefficient because it increases the number of class templates to be compared.

We will compare random selection, *minimum selection* and *multiple selections* in our experiments to demonstrate the importance of selecting good class templates. After the creation of class templates, there are two other parameters required by the DTW recognizer. For each class C^n ($1 \leq n \leq N$, N=total number of classes), these parameters will be derived from the training templates of C^n (denote as TC^n). They are:

- *End region (E1 and E2 of Eq. 3.3)*: for each class C^n, $E2$ is defined as the maximum length found in TC^n and $E1$ is defined as the minimum length found in TC^n. $E2$ and the length of class templates of C^n are used to determine the number of rows and columns of DTW distance tables of C^n respectively. $E1$ and $E2$ together define the all possible lengths of multimodal sequences of class C^n.
- *Rejection threshold*: for each C^n, we derive the rejection threshold by calculating the intra–class DTW distances between the selected class template of C^n and all TC^n. The threshold is defined as the mean of these distances plus standard deviations. During the sliding window operation, if the minimum DTW distance derived between the test template and the class template of C^n exceeds the threshold, this class template is given an infinite distance and this indicates the test template did not belong to class C^n.

C. Dynamic Time Warping recognizer

After the class templates have been created, the next step is to measure the similarity between online data and class templates. Since the length of multimodal sequences of the same class can vary greatly and the endpoints are unknown, the proposed system uses a variant of DTW to achieve both time alignment and similarity measure. In the remainder of this section, we will develop dynamic time warping for online recognition of multimodal sequences generated from multiple sensors.

1) Multi–dimensional local distance measure for DTW: The class templates $C(I \times V)$ and test template $T(J \times V)$ represent multimodal sequences where V is the number of variables. To calculated the DTW distance between the test and class templates, The system uses the extended *Euclidian distance* (Eq. 3.1) and *Cosine correlation coefficient* (Eq. 3.2) as the local distance measures to calculate the difference between the two vectors: C_i^V and T_j^V. Their recognition performances have been evaluated in [16], and they are defined as:

$$d_E\left(C_i^V, T_j^V\right) = \sqrt{\sum_{v=1}^{V} WV(v)\left(C_{i(v)} - T_{j(v)}\right)^2} \tag{3.1}$$

$$d_C\left(C_i^V, T_j^V\right) = 1 - \frac{\sum_{v=1}^{V} WV(v)\, C_{i(v)} T_{j(v)}}{\sqrt{\sum_{v=1}^{V} C_{i(v)}^2}\sqrt{\sum_{v=1}^{V} T_{j(v)}^2}} \tag{3.2}$$

where WV is a positive definite weight vector. The weight vector WV can be used in the DTW algorithm to give more weight to certain variables to improve the performance of online recognition. If every element of WV is equal to 1, we obtain the normal *Euclidian distance* and *Cosine correlation coefficient*. If prior knowledge of the importance of the sensors to the multimodal sequences under recognition are given, we can assign the elements of WV according to the importance of the sensors.

2) DTW variant for unknown start and end points: One of the major drawbacks in the original DTW algorithm is the endpoint constraints that requires the warping path to start and finish in diagonally opposite corner cells of the distance table as shown by the solid line in Fig. 2. When reasonably accurate determination of the start and end points of the multimodal sequence has been made, this constraint is acceptable and does not harm overall performance of the recognizer. However in the case of online recognition of multimodal sequences where the accurate endpoints are unknown, the original DTW algorithm is inadequate. As such, we use a variant of the original DTW algorithm that replaces the endpoint constraints to

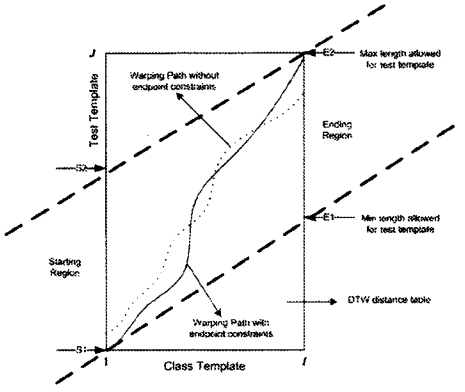

Fig. 2: DTW variant to solve the problem of unknown start or end points.

allow the start point to fall in a defined start region (i.e. between S1 and S2) and the end point to fall in a defined end region (i.e. between E1 and E2) as shown by the bold dashed lines in Fig. 2. Hence the optimal warping path starts at the first vector of the class template and within the start region of the test template, and ends at the last vector of the class template and within the end region of the test template. Given a start and end region, the purpose of the DTW algorithm is to determine (in an optimal and efficient manner) the warping path, which provides the best time alignment between the class and the test template and has the minimum cumulative distance. The searching of the optimal warping path for this variant of the DTW algorithm can be expressed as an extension of (Eq. 2.2):

$$DTW(C,T) = \min_{S_1 \le s \le S_2} \left(\min_{E_1 \le e \le E_2} \left(\sum_{q=1}^{Q} d(i(q), j(q)) \Big/ k \right) \right) \tag{3.3}$$

where s and e represent the start and end points of a warping path respectively.

3) Online DTW: The proposed system uses a sliding window based approach to achieve online recognition of multimodal sequences as depicted in Fig. 1. For each class template, the DTW recogniser maintains a distance table (see Fig. 2). The online recognition proceeds by constructing the distance table while the data is acquired. When a new vector of the test template is produced by the data pre–processing module, the distance table expands by one row and only the local distances for the corresponding cells in the distance table are calculated, i.e. when j^{th} vector of the test template is produced, $d(C_i, T_j)$ is calculated for all $i(1 \le i \le I)$. Once the distance table reaches the maximum buffer size defined during the training stage, then it becomes a sliding window. At each new time instance, the first vector of the test template and the first row of the distance table is deleted and a new vector is added into the j^{th} vector of the test template. The local distances of the j^{th} row of the distance table are re–calculated accordingly. In this way, the algorithm is efficient since it uses previous calculated values. At each new time instance, the online DTW recogniser calculates a new warping path in the updated DTW distance table by fixing the start point (i.e. $S2$=$S1$=1 in Eq. 3.3) and relaxing the end point (i.e. $E_1 \le e \le E_2$ in Eq. 3.3). For each class to be recognised, the DTW distance is obtained by calculating the cumulative distance along the new warping path. If the lengths of class templates (between classes or within the class) are quite different, we need to normalize the returned DTW distance. We would expect longer class templates to have higher DTW distances than short class templates, since they have longer warping paths on which to accumulate errors. The system uses the following normalization policies:

- N=1, no normalization on the distance.
- N=2, normalised by the length of the optimal warping path.
- N=3, normalised by the length of the longer template between class and test templates.
- N=4, normalised by the length of the shorter template between class and test templates.
- N=5, normalised by the sum of the lengths of both class and test templates.

At each new time instance, the normalised DTW distances are calculated for each class template and passed to the decision making procedure to perform classification.

D. Decision making procedure

The last major step in the proposed system (see Fig. 1) is the decision making procedure which chooses the class template that most closely matches the unknown test template. There are two decision rules used in the system depending on which method is used to create the class template.

For *Normal*, *Minimum*, and *Average* selections where a single class template is used for each class, the *nearest neighbour* decision rule (NN rule) is used. Assume we have N classes , C^n , $1 \le n \le N$, and for each test template we obtain the DTW distance D^n using our online algorithm. Then the decision rule is simply:

$$n* = \arg\min_{n} [D^n] \tag{3.4}$$

and the class template C^{n*} is chosen as the winning class template.

For *Multiple* selections in which each class is represented by two or more class templates, the *K–nearest neighbour* (KNN) decision rule can be used. Thus if we assume there are M class templates for all N classes, and we denote the m^{th} class template of C^n as $C^{n,m}$, where $1 \le n \le N$, $1 \le m \le M$, then the DTW distance for the m^{th} class template of C^n is $D^{n,m}$. If we select K $(1 \le k \le M)$ nearest neighbours out of M class templates of C^n that have the least DTW distances to the unknown test template (centre K samples in the cluster), and let it denotes the DTW distance of the k^{th} neighbour to the unknown test template as $D^{n,k}$, then for each test template, we obtain the DTW distance for D^n as:

$$D^n = \frac{1}{K} \sum_{k=1}^{K} D^{n,k} \tag{3.5}$$

and we choose the index of the recognized class using Eq. 3.4. It should be noted that for $K = 1$, the K–nearest neighbour decision rule becomes the nearest neighbour rule.

4. CASE STUDIES AND RESULTS

A. Experimental setup and evaluation

We use two real world datasets to explore the performance of the proposed online context recognition system. First, we demonstrate the accuracy and speed of the system for performing online recognition of shorter test templates. The templates are converted by data pre–processing module from the raw data acquired from homogenous sensors, and consist of continuous variables with significant variability. Second, we show that the system can achieve good results from longer test templates that are converted from the raw data acquired from heterogeneous sensors, and have larger dimensions with a mix of discrete and continuous variables. All the experiments in this paper were conducted on a Pentium-4 3.2 GHz with 1 GB of RAM running Windows XP professional. Most software components in our system are currently implemented in MATLAB, except the DTW recognizer which is implemented in C++ for computational efficiency.

We evaluate the performance of the online recognition system with *accuracy rate (AR)* and *local minimum deviation (LMD)* as illustrated in Fig. 3. Fig. 3a shows examples of multimodal sequences to be recognised, embedded in a continuous data stream. Fig. 3b shows the normalised DTW distance of all the classes for all the sliding windows starting at every time instance. The ground truth defines the start and end times of all the multimodal sequences to be recognised (depicted as ST and ET in Fig. 3a). The winning class

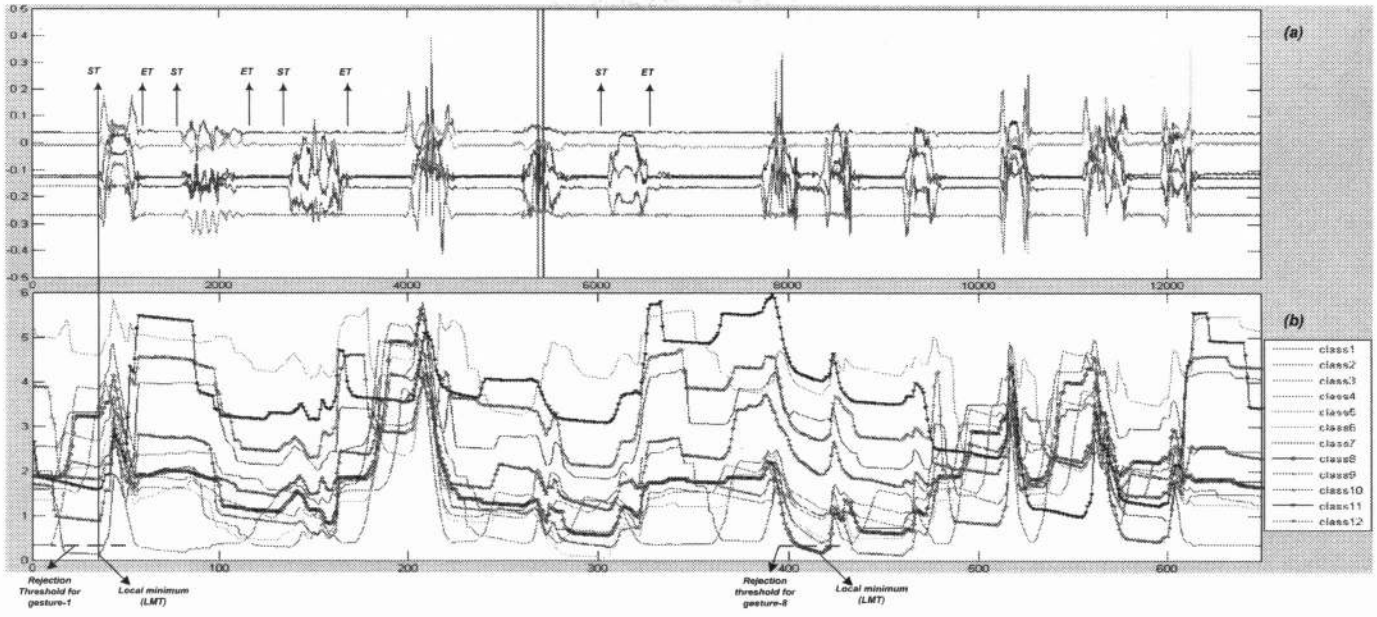

Fig. 3: (a) The multimodal sequences to be recognized, embedded in the six channels of continuous data. (b) The DTW distances of all classes calculated at every time instance.

for the multimodal sequence buffered in the current sliding window is defined as the one with the lowest DTW distance that is also lesser than the rejection threshold (the red dashed line at the right bottom of Fig. 3b). Hence, the AR is the ratio of the number of multimodal sequences, of which the winning class at time ST matches the ground truth, to the total number of multimodal sequences to be recognised in the data. When the winning class matches the ground truth, we define: the $LMD \triangleq \frac{|ST-LMT|}{ET-ST}$. LMT is defined as the time instance near ST where the local minimum of the DTW distances occurs, and indicates the time instance where the best match occurs. As depicted at the right bottom of Fig. 3b, all the time instances near ST of the first multimodal sequence that have the DTW distances lesser than the rejection threshold (the red dashed line) are correctly classified as *class-1* (the blue line), and LMT is the time instance where the local minimum of these DTW distances occurs. A reliable online recognition system will have high AR and low LMD.

B. Actor-independent online recognition of two hand gestures

Six channels of continuous data are acquired from two sensors (consisting of accelerometers: Analog Devices ADXL202 [18]) that can each measure acceleration of up to $\pm$2g at 150 samples/second in 3–D space. They are housed in small wristwatch sized enclosures worn in the form of a wristband on both wrists. The dataset provided [19] consists of four people mimicking the 12 gestures of a cricket umpire: *Cancel Call, Dead Ball, Four, Last Hour, Leg Bye, No Ball, One Short, Out, Penalty Runs, Six, TV Replay, and Wide* [20]. Fig. 3a shows the raw data form of the 12 gestures in the same order. Gestures were captured over different days with four different actors to introduce variability between the movements. The sequences of the same gesture are also different in length, because the same movements can be performed at different speeds.

The training data consists of 65 sequences for each of the 12 gestures, and segmented from continuous data streams. The raw data form of each gesture is a six dimensional sequence which has about 450$\sim$1200 vectors long (3$\sim$8 seconds). From the training data, we selected four different sets of class templates to represent each of the 12 gestures, and we also derived the *End region* (E1 and E2) and the *Rejection threshold* respectively for each gesture as described in section 3-B. The test data is another set of four continuous sequences that are captured by four actors respectively, performing each of the 12 gestures once (48 gestures in total which are embedded in four continuous data streams). Unlike the segmented sequences, the test data is a continuous data stream that consists of both gesture sequences and "no-activity" sequences. Fig. 3a shows one of the test sequences. Although we have demonstrated that the DTW recognizer can classify the gestures in raw data without feature extraction and yielded good results, the high computation required is the trade-off [16]. To speed up online gesture recognition, our data pre-processing module extracts features (i.e. Extracting *Mean* and *Standard deviation* within sliding windows of size 50 samples and overlap 30) from the data buffered in the current sliding window as well as the class templates. Therefore, the online DTW recognizer ends up dealing with shorter test and class templates, and is more efficient.

TABLE 1: THE ARS ACHIEVED BY USING FOUR SETS OF CLASS TEMPLATES: **Min–4**, **RD–4**, **Min–1**, AND **RD–1**.

	Min–4	RD–4	Min–1	RD–1
K = 1, N = 1	97.92	93.75	89.58	83.33
K = 1, N = 2	93.75	93.75	91.67	77.08
K = 1, N = 3	100.00	97.92	91.67	83.33
K = 1, N = 4	100.00	97.92	93.75	83.33
K = 1, N = 5	100.00	97.92	93.75	87.50
K = 2, N = 1	95.83	95.83	92.71	82.81
K = 2, N = 2	93.75	93.75		
K = 2, N = 3	97.92	97.92		
K = 2, N = 4	97.92	97.92		
K = 2, N = 5	97.92	97.92		
K = 3, N = 1	93.75	91.67		
K = 3, N = 2	93.75	93.75		
K = 3, N = 3	95.83	95.83		
K = 3, N = 4	95.83	93.75		
K = 3, N = 5	97.92	95.83		
K = 4, N = 1	93.75	95.83		
K = 4, N = 2	91.67	93.75		
K = 4, N = 3	95.83	95.83		
K = 4, N = 4	95.83	95.83		
K = 4, N = 5	95.83	95.83		
Avg. time (sec.)	44	44	17	17

The ARs are shown in Table 1. The ARs were obtained for different sets of class templates, different values of K (in the K-nearest neighbour decision rule r.f. section 3-D) from K=1 to K=4, and different normalisation methods from N=1 to 5 (r.f. section 3-C.3). First, we compare the *random*, *minimum*, and *multiple* selection methods (r.f. section 3-B) by using the four different sets of class templates. They are selected from training data by: using minimum selection to select four class templates per gesture that have least *intra-class DTW distances* (Min–4), selecting four class templates per gesture randomly (RD–4), using minimum selection to select one class template per gesture (Min–1), and selecting one class template per gesture randomly (RD–1). Both ARs obtained by using one template per gesture (Min–1 and RD–1) are worse than ARs using

four templates per gesture (Min–4 and RD–4). This shows that the AR of DTW recognizer can be improved by increasing the number of class templates per class, but the trade–off is that the computational time is raised from an average of 17 seconds to 44 seconds. Second, the method used to select the class template is also critical. Using minimum selection outperformed random selection in both cases as the average ARs of Min–1 and Min–4 are about 10% and 1% higher than RD–1's and RD–4's respectively. The ARs achieved by using both *normal* and *average* selection methods are not shown, because they produced similar ARs as minimum selection method. Third, the value for K should be carefully chosen as the number of class templates increases per class. For four class templates per class, the K=1 and K=2 rules yield higher ARs than K=3 or K=4 rules. For K=1, both Min–4 and RD–4 yield best ARs of 100% and 98% respectively. On the other hand, the best ARs of Min–4 and RD–4 for K=4 declined to 95.8%. Finally, applying different normalisation methods also affects the AR (r.f. section 3-C.3). For K=1, Min–4, RD–4 and Min–1 yield higher ARs by applying N=3, 4 or 5 than N=1 (no normalisation), increasing about 2% to 4% in AR. For K=2, 3, and 4 of Min–4, RD–4, applying N=3, 4 or 5 always yields better AR than N=1. In this experiment, applying normalisation did not improve the AR dramatically (compare to N=1); this is because the 12 gesture sequences have similar lengths in the test data.

TABLE 2: THE LMDS CALCULATED FOR ALL 12 GESTURES BY USING **Min–4**, **RD–4**, **Min–1**, AND **RD–1**.

	Min–4 K=1 N=5	RD–4 K=1 N=5	Min–1 K=1 N=5	RD–1 K=1 N=5
Gesture1	0.83	0.83	0.83	4.39
Gesture2	3.03	2.02	4.01	11.94
Gesture3	2.86	3.81	3.81	3.81
Gesture4	4.25	3.33	5.00	3.33
Gesture5	9.50	13.17	12.00	8.67
Gesture6	3.96	3.96	5.28	1.92
Gesture7	4.46	4.63	4.46	7.83
Gesture8	5.63	7.49	5.63	7.51
Gesture9	4.32	2.27	4.32	5.16
Gesture10	2.50	2.63	3.54	2.22
Gesture11	3.36	4.22	3.36	2.26
Gesture12	6.34	2.50	2.50	7.50

The local minimum deviations (LMDs) of all 12 gestures from Min–4, RD–4, Min–1, and RD–1 are shown in Table 2. Most of the LMDs are less than 7%, except two special cases. First, it should be noted that RD–1 has several large LMDs that are greater than 7%. This is because one class template is selected randomly for each gesture, and the selected one might have large DTW distance to the test sequences and result in higher LMDs. Second, the LMDs for the 5^{th} gesture are greater than 8% for all four sets of class templates. This is because the 5^{th} gesture sequence performed by the third actor is quite different from the training data, and it has LMD greater than 30% for all four sets of class templates. From the LMDs between the selected class templates and test data, we can determine if the selected class templates are good representatives.

The total length of the test data is about 56000 samples (around 14000 samples for each), corresponding to approximately 400 seconds duration. Using Min–1 or RD–1, the online recognition of the test data took about 17 seconds and achieved highest AR of 93.75% and 87.5% respectively. The highest ARs of 100% and 97.92% were achieved in about 44 seconds using Min–4 and RD–4 respectively, which only took about 0.016 second (performing 48 DTW comparisons) to classify the test templates starting at every time instance. The results demonstrate that the DTW recognizer is both fast and accurate for online recognition of gesture sequences acquired from the accelerometers.

C. Online context recognition of mobile devices

The data used here have been presented at the workshop for "*Benchmarks and a database for context recognition*" [21], and proposed as a suitable benchmark for evaluating context recognition algorithms. The data is obtained from sensors placed in a small sensor box which is then attached to a mobile handheld device. The sensors used to capture this data include *accelerometers with 3 axes*, an *illumination sensor*, a *thermometer*, a *humidity sensor*, a *skin conductivity sensor*, and a *microphone*. The data is collected by two users carrying the mobile device and repeating each of the five predefined user scenarios for less than 25 times (r.f. Table 2 of [21] for details). The duration of each scenario varies from 1 to 5 minutes and the sensor data is sampled at every second. The data set is provided in an ASCII flat file, and there are a total of 32 columns and 46045 rows which corresponds to approximately 12.8 hours of data. The first three columns include: the ID number of the scenario (SID), the ID number of the repetition of the scenario (RID) and time from the beginning of repetition in seconds. The rest of the 29 columns are *context atoms*. They have pre-processed the raw sensor signals into context atoms, i.e. extracting features from the raw signals and then quantising the dynamic range of feature value into the range [0, 1] with *fuzzy sets*. More details on the processing of sensor data into context atoms is available in [22]. Hence, each user scenario is described by a multimodal sequence with 29 dimensions of feature values per second over a duration of 1 to 5 minutes. From both SID and RID, we know the start and end time for each repeated user scenario so that we can define the ground truth. More detailed information on this dataset is available in [21], [22].

Since the raw data form of the benchmark data is not released for public use, we cannot apply our own data pre-processing module. The online DTW recognizer has to deal with longer test templates that have about 80∼300 feature vectors (1∼5 minutes) and larger dimensions (29 dimensions). To train the DTW recognizer, we selected four different sets of class templates from the benchmark data to represent each of the 5 user scenarios, and we also derived the *End region* (E1 and E2) and the *Rejection threshold* respectively for each user scenario as described in section 3-B. In the first test case, we use the benchmark data as it is in the ASCII flat file to test our online recognition system. There are a total of 240 repetitions of the five user scenarios and each repetition is followed by another repetition without any other data in between. Since the feature vectors in the ASCII file are annotated with picture sequences [21], we can identify the feature vectors while the mobile device is on the table and doing nothing. To make the online recognition task more complicated and closer to real situations, we created the second test data by rearranging the order of the 240 repetitions of five user scenarios randomly and also inserting 1∼3 minutes of "on the table" feature vectors in between each repetition. The total length of the second test data becomes 75687 rows, corresponding to approximately 21 hours of data.

TABLE 3: THE ARS FOR BOTH T1 AND T2 ACHIEVED BY USING DIFFERENT SETS OF CLASS TEMPLATES, K VALUES, AND N VALUES.

	T1–Min–5	T1–Min–1	T2–Min–5	T2–RD–5	T2–Min–1	T2–RD–1
K = 1, N = 1	71.25	67.92	70.83	69.58	67.92	71.67
K = 1, N = 2	93.75	91.25	80.83	80.00	77.50	72.92
K = 1, N = 3	98.33	96.67	94.58	92.50	93.33	88.75
K = 1, N = 4	98.33	98.33	92.50	94.58	92.08	90.42
K = 1, N = 5	98.33	97.92	94.17	93.33	92.50	88.75
K = 2, N = 1	70.83		70.42	66.25		
K = 2, N = 2	92.50		77.50	71.67		
K = 2, N = 3	97.92		94.58	91.67		
K = 2, N = 4	98.33		92.92	92.92		
K = 2, N = 5	98.33		94.17	92.92		
K = 5, N = 1	68.33		67.50	60.83		
K = 5, N = 2	92.92		77.92	69.58		
K = 5, N = 3	97.92		94.58	92.50		
K = 5, N = 4	97.92		94.17	93.33		
K = 5, N = 5	98.33		94.17	92.50		
Avg. time (sec.)	13334	2564	23312	23241	5545	5844

In Table 3, the 2^{nd} and 3^{rd} columns are the ARs for the first test data (T1), and from the 4^{th} to 7^{th} columns are the ARs for the second test data (T2). The ARs were obtained for four different sets of class templates: Min–5 (use five class templates to represent each scenario), RD–5, Min–1, and RD–1, different values of K, and different normalisation methods from N=1 to 5. The best AR on T1 is 98.33%, on the other hand, the best AR on T2 is 94.58%. Comparing the same settings of class templates, K and N, most ARs obtained on T1 are better then T2. This is because the scenarios

in T1 are not preceded and followed by any noise, whereas the scenarios in T2 are separated by 1∼3 minutes of "on the table" feaure vectors. In the real situation, the multimodal sequences under recognition will always be preceded and followed by noise, and the AR obtained will also be affected by the noise. Second, we can see that applying normalisation on DTW distances improves the AR dramatically (e.g. comparing the ARs from N=3, 4, or 5 to N=1). This is because the average length of all repetitions of the 4^{th} user scenario is only 100 seconds compared to the average length of the other user scenarios (238, 206, 217, and 205 seconds for scenarios 1, 2, 3 and 5 respectively), for which the differences are high. From the observation on the confusion matrices of all ARs, most mis–classifications occurred when the 4^{th} user scenario are misclassified as the 5^{th} user scenario. Therefore, normalisation on the DTW distance is required if the lengths of multimodal sequences are very different between the classes. Normalisation methods, N=3, 4, and 5 always have similar ARs and are much higher than N=1 and 2. We found that normalising the DTW distances by the length of the optimal warping path (N=2) gave low ARs as no normalisation has applied (N=1) in both case studies.

TABLE 4: SOME LMDS FOR THE FIVE SCENARIOS.

	Scenario1	Scenario2	Scenario3	Scenario4	Scenario5
T1–Min–5, K=1, N=3	0.77	0.35	0.30	1.07	0.78
T1–Min–1, K=1, N=4	2.80	0.43	0.41	2.64	0.79
T2–Min–5, K=1, N=3	3.60	0.49	0.35	0.83	2.63
T2–RD–5, K=1, N=4	2.63	0.46	0.36	0.82	3.50
T2–Min–1, K=1, N=3	0.60	0.58	1.70	1.11	1.10
T2–RD–1, K=1, N=5	5.83	0.70	5.08	3.40	2.66

Table 4 shows some of the local minimum deviations (LMDs) of all five user scenarios. Most of the LMDs are less than 3%, except T2–RD–1 has several LMDs that are greater than 3%, which again shows that the randomly selected class templates are not good representatives. The LMDs calculated from these two test data (T1 and T2) are low, corresponding to the high ARs obtained.

For T1, the online context recognition took about 13334 seconds and achieved the highest AR of 98.33%. When the DTW recognizer receives each new feature vector, it only takes 0.29 seconds (13334/46045=0.29) to calculate new warping paths and classify the data currently buffered in the sliding window. For T2, the recognition process took about 23241∼23312 seconds to achieve the best AR of 94.58%. To classify each updated sliding window, it only takes 0.31 seconds (less than one second). Compared to the 0.016 seconds required in the last case study, the online recognition of this benchmark takes 19 times longer (0.31/0.016=19), which is expected as the multimodal sequences to be recognised have more feature vectors and larger dimensions. The results again demonstrate that the DTW recognizer is both fast and accurate for online context recognition of multimodal sequences.

This benchmark dataset has also been analysed using the *Symbol Clustering Map* (SCM) technique [23], [24], [25]. This method is similar to *Self-Organizing Maps* [26]. Two levels of SCM are used to classify the five scenarios. The first level clusters the instantaneous patterns and then the second level finds the temporal relationships. From the confusion matrix of [25], we know that using two levels of SCM can only identify four clusters from the five classes of user scenarios, and the method can not distinguish well between scenarios one and two. However, SCM has the advantage of performing unsupervised clustering of the user scenarios. Although we have to train and find the best class template for each of the five user scenarios for the DTW recognizer, the classification results are better.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an online context recognition system using DTW. Our system is able to recognise multimodal sequences of different lengths generated from multiple sensors with both discrete and continuous outputs. We have demonstrated the accuracy and speed of the system for online context recognition on two real word datasets. These datasets generally consist of sub-events that show significant variation in their durations and sensor outputs. Importantly the DTW complexity is linear with the number of dimensions in the data and hence can deal with high dimensional time varying data. In future work, we would like to extend *dynamic time warping* for recognising more complex multimodal sequences such as: interleaved sequences, sequences with gaps, and missing sub-sequences.

REFERENCES

[1] R. E. Crochiere, J. M. Tribolet, and L. R. Rabiner, "An improved endpoint detector for isolated word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. COM-29, no. 5, pp. 621–659, 1981.

[2] J. G. Wilpon and L. R. Rabiner, "Application of hidden markov models to automatic speech endpoint detection," *Computer Speech and Language*, vol. 2, no. 3/4, pp. 947–954, 1987.

[3] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics,Speech, and Signal Proc.*, vol. 26, pp. 43–49, 1978.

[4] C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 6, pp. 623–635, 1980.

[5] L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-26, no. 4, pp. 575–582, 1978.

[6] C. S. Myers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 2, pp. 284–297, 1981.

[7] L. R. Rabiner and C. S. Myers, "Connected digit recognition using a level building dtw algorithm," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 3, pp. 351–363, 1981.

[8] C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "An investigation of the use of dynamic time warping for word spotting and connected speech recognition," in *IEEE ICASSP*, 1980, pp. 173–177.

[9] E. Keogh, "A fast and robust method for pattern matching in time series databases," in *IEEE ICTAI*, Newport Beach, CA, 1997, pp. 578–584.

[10] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems: An International Journal (KAIS)*, pp. 358–386, 2004.

[11] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos, "Iterative incremental clustering of time series," in *EDBT*, Crete, Greece, 2004.

[12] R. DeVaul, M. Sung, J. Gips, and A. Pentland, "Mithril 2003: applications and architecture," in *IEEE ISWC*, New York, 2003.

[13] M. C. Mozer, "The neural network house: An environment that adapts to its inhabitants," in *Proc. of the AAAI Spring Symposium on Intelligent Environments*, Menlo, Park, CA, 1998, pp. 110–114.

[14] A. Krause, D. Siewiorek, A. Smailagic, and J. Farringdon, "Unsupervised, dynamic identification of physiological and activity context in wearable computing," in *IEEE ISWC*, New York, 2003.

[15] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping." in *SIAM 2005 Data Mining Conference*, CA, 2005.

[16] M. H. Ko, G. West, S. Venkatesh, and M. Kumar, "Temporal data fusion in multisensor systems using dynamic time warping," in *SENSORFUSION 2005 co-located with WICON 2005*, Budapest, Hungary, 2005.

[17] W. H. Abdulla, D. Chow, and G. Sin, "Cross-words reference template for dtw based speech recognition systems," in *IEEE TENCON*, 2003.

[18] "Analog devices inc., adxl202 dual axis accelerometer," 2005.

[19] G. Chambers, S. Venkatesh, G. West, and H. Bui, "Segmentation of intentional gestures for sports video annotation," in *IEEE MMM*, 2004.

[20] D. Shepherd, "BBC sport academy cricket umpire signals," 2005.

[21] J. Mntyjrvi, J. Himberg, P. Kangas, U. Tuomela, and P. Huuskonen, "Sensor signal data set for exploring context recognition of mobile devices," in *PERVASIVE*, Linz/Vienna, Austria, 2004.

[22] J. Mntyjrvi, "Sensor-based context recognition for mobile applications," Ph.D. Thesis, Ph.D. Thesis, VTT Publications, 2003.

[23] J. Flanagan, J. Himberg, and J. Mntyjrvi, "A hierarchical approach to learning context and facilitating user interaction in mobile devices," in *AIMS in conjunction with Ubicomp*, 2003.

[24] J. A. Flanagan, J. Mntyjrvi, and J. Himberg, "Unsupervised clustering of symbol strings and context recognition," in *IEEE ICDM*, 2002.

[25] J. Himberg, J. A. Flanagan, and J. Mntyjrvi, "Towards context awareness using symbol clustering map," in *WSOM*, Kitakyushu, Japan, 2003.

[26] T. Kohonen, *Self-Organizating Maps*, 2nd ed. Springer-Verlag, 2000.