REAL-TIME DTW-BASED GESTURE RECOGNITION EXTERNAL OBJECT FOR MAX/MSP AND PUREDATA

Frédéric Bettens
Faculty of Engineering (FPMs) - TCTS Lab
7000 Mons, BELGIUM
frederic.bettens@fpms.ac.be

Todor Todoroff
ARTeM
1030 Bruxelles, BELGIUM
Faculty of Engineering (FPMs) - TCTS Lab
7000 Mons, BELGIUM
todor.todoroff@skynet.be

ABSTRACT

This paper focuses on a real-time Max/MSP implementation of a gesture recognition tool based on Dynamic Time Warping (DTW). We present an original "multi-grid" DTW algorithm, that does not require prior segmentation. The num.dtw object will be downloadable on the `numediart` website both for Max/MSP and for Pure Data. Though this research was conducted in the framework described below, with wearable sensors, we believe it could be useful in many other contexts. We are for instance starting a new project where we will evaluate our DTW object on video tracking data as well as on a combination of video tracking and wearable sensors data.

1 INTRODUCTION

The "Dancing Viola" project, described in more details in [7], was led at the Faculté Polytechnique de Mons within the `numediart`program and is linked to viola player Dominica Eyckmans. It covers some of the aspects of the long-term project "Extension du corps sonore" launched by Musiques Nouvelles, a contemporary music ensemble in Mons, that aims at giving intrumental music performers an extended control over the sound of their instrument. The intention is to extend the understanding of the sound body from the instrument only to the combination of the instrument and the whole body of the performer. Whereas usual augmented instruments designs track the gestures used to play the instrument to expand its possibilities, this specific project focuses on using non-musical gestures to transform the sound of the instrument. Our approach is dictated by the nature of Dominica's project: she is actually dancing while playing the viola and we track her dancing movements rather than her hands movements. But the recognition algorithm we present here is not limited in any way by this specific context, as we have successfully demonstrated using a database of hands gestures measured with sensors placed on the hands. Gesture recognition is a welcome addition to an interactive performance and can be used to trigger events, to adapt the response of the virtual instruments according to the detected gestures, or to move through the various steps of a performance. As other modules like hit detection, mapping, interpolation (also developed within the "Dancing Viola" project and described in [8]) or sound synthesis and transformations, must be running simultaneously on the same computer, it is essential to minimize the computational load. This Max/MSP object is being integrated in the ARTeM software framework for the concerts with Dominica Eyckmans, as well as for other artistic works.

While using similar hardware (cf. 2.1), the atomic gesture recognition algorithm developed by Benbasat and Paradiso [1] is not suitable in our project: as dance movements are usually chained without pauses and cannot be decomposed in a concatenation of elementary movements along one accelerometer axis only, we have to consider an algorithm that can deal with unconstrained fluid motions, without the knowledge of the start and end of a gesture.

As for Automatic Speech Recognition (ASR) applications, the most popular algorithms used for gesture recognition are Dynamic Time Warping (DTW) [5, 4] and Hidden Markov Models (HMMs) [2]. In our framework, the aim is to develop a user-dependent recognition system with a small gesture vocabulary and a database of limited size. As some gestures should be added, removed, enabled, or disabled easily and quickly, without any training procedure, we chose for the DTW algorithm, which we adapted to make it usable in real-time without the need for segmentation.

This report is divided in following sections: after a brief description of the system, we present the gesture recognition module, by describing our "multi-grid" DTW algorithm and its real-time Max/MSP implementation, as well as some preliminary results, and we conclude with future investigations.

SMC 2009, July 23-25, Porto, Portugal

2 SYSTEM

2.1 Sensors

The sensor system allows for the data of two sensors (each a combination of a 3-axes accelerometer and a 2-axes gyroscope), placed on both ankles of the performer, to be transmitted every 8ms wirelessly over Wi-Fi. More details on the sensors can be found in [7]. The placement on the ankles presents a minimal hinderance even for movements on the ground. Depending on the results of further experimentation with the new software tools we will consider the need, the type, and the placing of additional sensors on Dominica's body.

2.2 Software framework

The ARTeM software, developed inside the Max/MSP environment to map sensor data to parameters of various sound transformation algorithms, is organized around a modular concept: the audio paths of the various virtual instruments are connected through a matrix, with external inputs and outputs of virtual instruments injected from the top and redirected with selectable level, to the inputs of the virtual instruments and the external sound outputs.

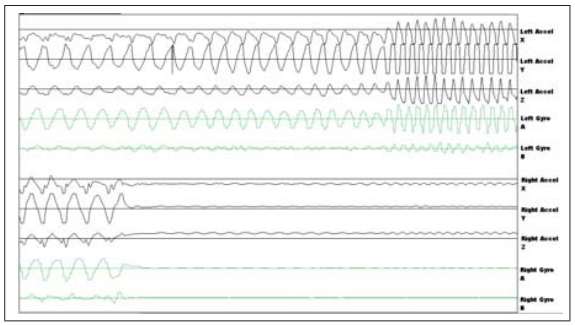

Figure 1. A data recording: 3-axes accelerometer (black) and 2-axes gyroscopic data (green) for left and right ankles.

The sensors data (Figure 1) are received as UDP packets through the normal Wi-Fi interface. An external decodes the custom protocol, scales the raw data and defines a name space depending on configuration messages sent to its input and outputs data as messages. All samples are then made available through a send/receive scheme throughout all the patches.

3 GESTURE RECOGNITION

3.1 DTW algorithm

The classical DTW algorithm uses Dynamic Programming (DP) principles to determine the best nonlinear mapping (Figure 2) between the temporal indices of the test sequence $(i = 1..I)$ and those of the reference sequence $(j = 1..J)$, assuming that both these sequences have been segmented. We denote by $d(i, j)$ the (non-negative) "local distance" (or dissimilarity measure) between the test frame T_i and the reference frame R_j (where a frame is composed of the data of all sensors and axes at a given time), and by $D(i, j)$ the "accumulated distance" along the sub-path between the origin and the current node (i, j). The algorithm aims at minimizing these accumulated distance values and/or at extracting the associated best path (i.e., the sequence of nodes) in the DTW grid (Figure 2). A classical way of computing the accumulated distance value $D(i, j)$ along a sequence of nodes (i_k, j_k) $(k = 1..K)$ consists in weighting the local distance elements $d(i_k, j_k)$ with transition costs that depend on the predecessor (i_{k-1}, j_{k-1}), and summing up the weighted values:

$$D(i,j) = \sum_{k=1}^{K} W(i_k, j_k; i_{k-1}, j_{k-1})\, d(i_k, j_k) \qquad (1)$$

These transition costs raise the issue of normalization when computing paths of different lengths (e.g. when a test gesture is compared with several reference gestures of unequal duration). Dividing the optimal distance by the "path length" (i.e. the sum of all weights along the path) leads to the mathematical expression of an average "cost per node" and, using the following symmetric transition cost type [6]:

$$W_k = (i_k - i_{k-1}) + (j_k - j_{k-1}) \qquad (2)$$

the normalization factor $(I + J)$ is path-independent. The question of the weight of the local distance corresponding to the first node is solved by computing the transition cost between a "fictitious" original node $(0, 0)$ and the first node.

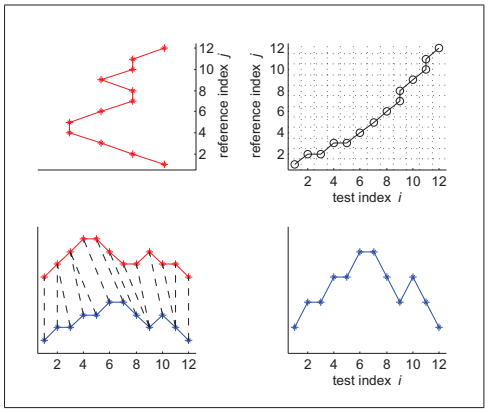

Figure 2. Mapping between two time series and DTW grid.

3.1.1 DTW search constraints

- Monotonicity and strict endpoint constraint. In its strictest form, any candidate path must not only be monotonic, meaning that $i_{k-1} \leq i_k$ and $j_{k-1} \leq j_k$, but also begin at $(1,1)$ and end at (I,J) exactly.

- Global path constraints. Itakura [3] suggests the specification of the maximum allowable compression and expansion factors ($\lambda_{max} \geq 1$ and $\lambda_{min} \leq 1$, with e.g. $\lambda_{min} = 1/\lambda_{max}$), whereby all paths must entirely lie within a parallelogram (Figure 3a). Another global constraint, proposed by Sakoe and Chiba [6], requires that the paths lie within a simple strip around a purely linear path: $|j_k - i_k| \leq R$, where R is the "window width" (Figure 3c).

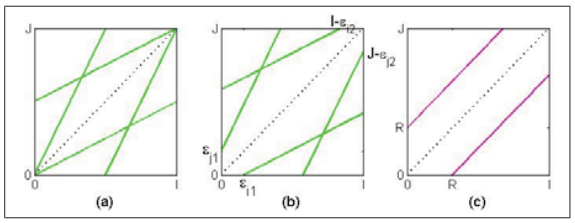

Figure 3. Global constraints: (a) Itakura, (b) Itakura (relaxed), (c) Sakoe and Chiba

- Local path constraints. The expansion or compression ratio between test and reference can also be limited locally, in the neighbourhood of each node. These local constraints are usually defined by listing the legal transitions. Equations 3 and 4 show the local path constraint implemented, where each node (i_k, j_k) can be reached from three different sets of predecessors (Figure 4):

$$D(i_k, j_k) \; = \; min(\; D_1, \; D_2, \; D_3) \qquad (3)$$

with:

$$\begin{aligned} D_1 &= D(i_k - 1, j_k - 2) + 2d(i_k, j_k - 1) + d(i_k, j_k) \\ D_2 &= D(i_k - 2, j_k - 1) + 2d(i_k - 1, j_k) + d(i_k, j_k) \\ D_3 &= D(i_k - 1, j_k - 1) + 2d(i_k, j_k) \end{aligned} \qquad (4)$$

- Relaxed endpoint constraint. To address the issue of locating accurately and in real-time the endpoints of a test sequence, the constraints are relaxed by permitting the path to start from one of the following nodes: $(1,1)$ to $(1+\epsilon_{i_1}, 1)$, or $(1,1)$ to $(1, 1+\epsilon_{j_1})$, and to end at one of the following nodes: $(I - \epsilon_{i_2}, J)$ to (I,J), or $(I, J - \epsilon_{j_2})$ to (I,J) (Figure 3b). Consequently, the different paths associated to each of the candidate

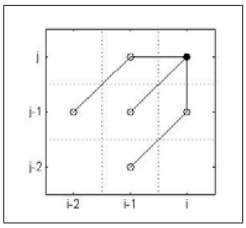

Figure 4. Local constraints

terminal nodes are compared on the basis of their normalized accumulated distances, where the global normalization factor $(i_K + j_K)$ is determined by the final coordinates only.

When only the starting point is approximately known, lower and upper bounds of the other endpoint may be found: e.g. $I_{min} = J/2$ and $I_{max} = 2J$, when the expansion/compression ratio lies in the range between 1/2 and 2 (if we neglect ϵ_{i_1} and ϵ_{j_1} values). In this context, since the ending point is *a priori* almost unknown, we decide to remove the margin parameters ϵ_{i_2} and ϵ_{j_2}, as well as to remove the global constraints that were linked to that ending point (i.e. two straight lines in Figure 3b). Finally, the gesture is restricted to end somewhere between the bounds I_{min} and I_{max} along the i axis, and strictly at J along the j axis (Figure 5). In other words, the warping consists in aligning the whole reference sequence with a test sequence (or subsequence) that may be up to twice as long or twice as short.

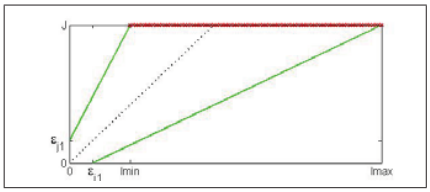

Figure 5. Final global constraints (with $R = \infty$) and set of admissible ending points

Figure 6 shows an example of local (left) and normalized accumulated (right) distance matrices for similar (top) and different (bottom) gestures, with following parameter values: $\epsilon_{i_1} = 8$, $\epsilon_{j_1} = 0$, $\lambda_{min} = 0.5$, $\lambda_{max} = 2$, and $R = \infty$. Low distance values (depicted by blue pixels) are obtained when comparing similar gestures. Conversely, high dissimilarities are observed when the tested gesture is very different from the reference one, resulting in a worse matching score.

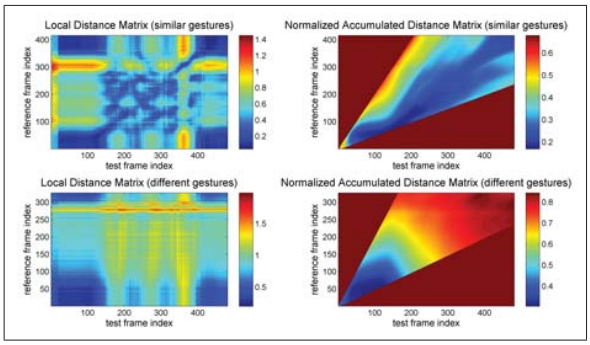

Figure 6. Local (left) and Normalized Accumulated (right) Distance Matrices for similar (top) and different (bottom) gestures

3.1.2 "Multi-grid" DTW algorithm and real-time Max/MSP implementation

Only few implementations of DTW do not require prior segmentation. Oka [5] presents a continuous DP algorithm, which is an efficient real-time method as the different paths originating from all possible starting points are simultaneously competing in the same DTW grid (one per reference gesture). However, he does not explain how to include global constraints. On the other hand, Ko [4] describes a method including these constraints, but at the cost of a higher computational load, as whole new paths are calculated from each new starting point (i.e. at every time instant) in the accumulated distance matrix (for each reference gesture).

Our "multi-grid" DTW algorithm provides a compromise solution. The method uses simultaneously a set of shifted DTW grids, each one hypothesizing another starting point (or set of consecutive starting point candidates when $\epsilon_{i1} \neq 0$) for the test sequence. The time shift between two successive DTW grids will generally be equal to $hop\_size = 1 + \epsilon_{i1}$. The number of simultaneously active grids can be limited to the following quantity: $S_{max} = ceil(I_{max}/hop\_size)$. As J may vary from one gesture to the other, I_{max} and S_{max} are also depending on the specific reference gesture. At every time instant, one best score (possibly "infinite" at the beginning) is computed in each shifted grid, and the minimum value of all these normalized accumulated distances is assigned to the given reference gesture. Despite the computation of several shifted grids, a low complexity can be achieved via an iterative implementation (like in [4]), where only one partial column $D(i, j)$ is evaluated in each grid at a given time i (for each reference gesture), instead of all (partial) preceding columns from the starting point.

Figure 7 illustrates normalized accumulated distance matrices for successive shifted DTW grids when test and reference gestures are similar. A good matching score is obtained for the low shift values, while it becomes worse when the delay increases.

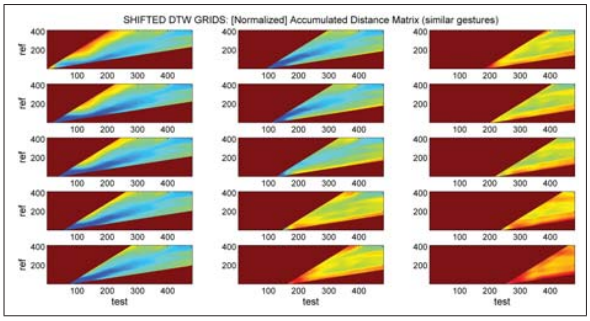

Figure 7. Normalized Accumulated Distance Matrices for successive shifted DTW grids (similar gestures)

Figure 8 also illustrates normalized accumulated distance matrices for successive shifted DTW grids, but when test and reference gestures are different. Again, the matching scores obtained in this latter figure are worse than the scores obtained in the former one.

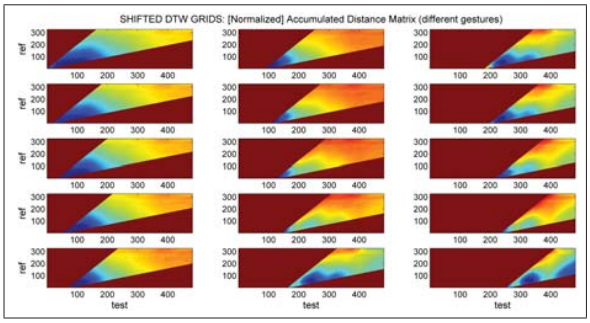

Figure 8. Normalized Accumulated Distance Matrices for successive shifted DTW grids (different gestures)

Finally, the overall gesture recognition module has been implemented as a Max/MSP external (see Figure 9), which includes the "multi-grid" DTW algorithm, as well as the pre- and post-processing stages described hereafter. It also evaluates and displays the time compression/expansion ratio, providing feedback to the artist (e.g. during rehearsals).

3.2 Pre-processing and distance metrics

The pre-processing of the sensors data and the calculation of the local distances are not part of the DTW algorithm itself, but their computation is a preliminary stage, briefly explained in this subsection.

The current version of our system implements a downsampling stage (with a factor 4), preceded by a lowpass filtering step, and uses the L_1-distance, whose computation is

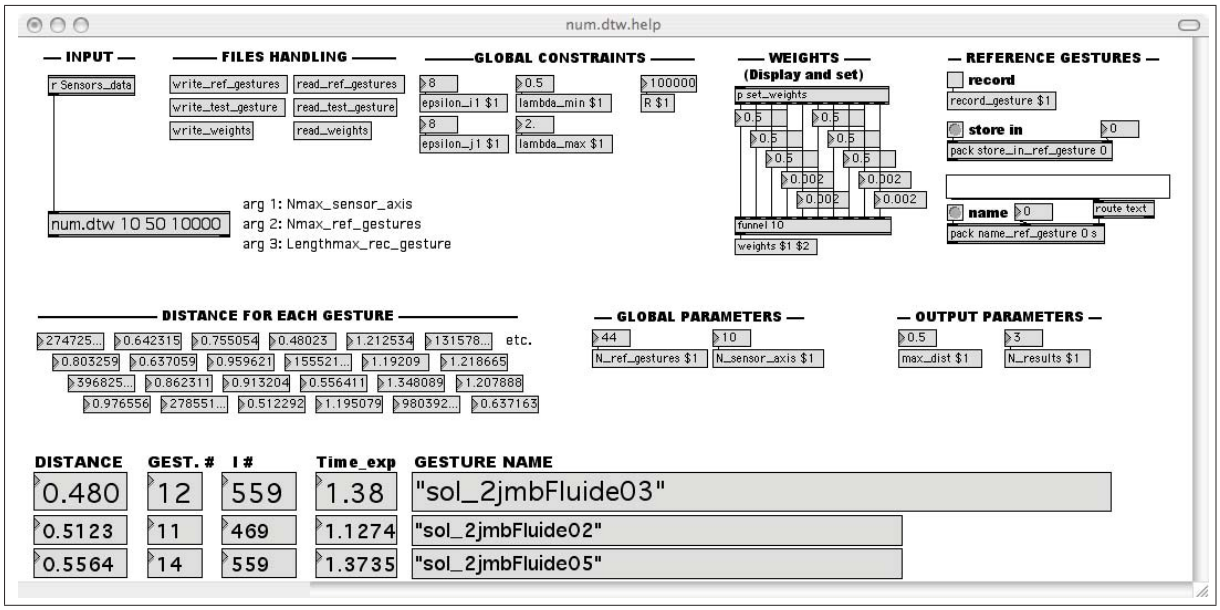

Figure 9. num.dtw Max/MSP external

very efficient as its expression is made of a (weighted) sum of the absolute value of differences. During this calculation, the sensors data are weighted, as some of them are varying within completely different ranges of values and expressed in different units (e.g. accelerometer data $\pm 2g$ and angular velocity $\pm 500°/s$). The easiest way consists in normalizing the samples axis per axis (e.g. dividing them by 2 and 500, respectively).

3.3 Post-processing

In the current Max/MSP implementation, the post-processing consists in selecting, at each moment, the gesture with the lowest normalized accumulated distance and validating its recognition if this value is below a user-defined global threshold.

3.4 Preliminary results

We first tested our "multi-grid" DTW algorithm offline, on a small database composed of recordings of 44 isolated dance gestures (with a sampling period of $8ms$). Each individual unsegmented test gesture was compared with each segmented reference gesture.

As a result of all these pair-wise comparisons, we obtained a "pseudo confusion matrix" (Figure 10), the small amount of recorded data preventing us from deriving actual statistics. However, one can see that the main diagonal is in blue colour, because each gesture is very similar to itself, and the blocks of blue pixels are explained by the presence

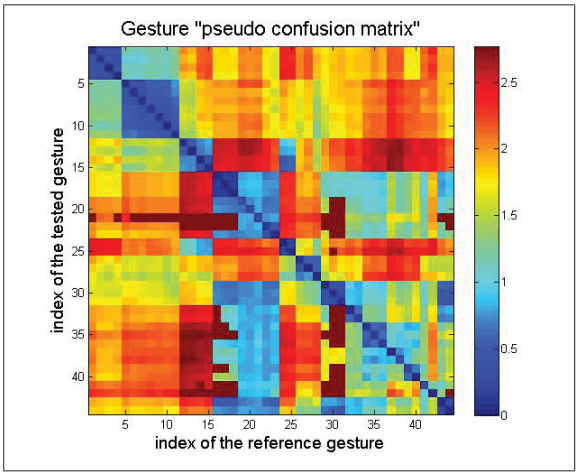

Figure 10. Gesture "pseudo confusion matrix"

of several occurences of the same gesture in our database. This representation allowed us to examine the ambiguity between some different pre-defined gestures and to get information about an appropriate fixed global threshold or a series of gesture-based threshold values.

Our DTW algorithm was also used in a second application. The sensors were attached to the wrists of the second author and a dozen of left and/or right arm movements were successfully recognized in real-time. The post-processing was slightly modified into an N-best strategy ($N = 3$), that is, displaying continuously the three best matched gestures. However, the correct gesture was always classified in first position, except when the execution was too fast (e.g. more than two times faster, while a factor 2 was the maximum fixed by local and global constraints).

4 CONCLUSION AND FUTURE WORK

A real-time DTW-based gesture recognition tool has been developed, with a great flexibility provided by its set of parameters (minimum and maximum expansion and compression ratios, "window width", sensor axes weights, user-defined global threshold, etc.) and it has been successfully tested on two different small databases. We are finalizing the port of the external to Pd.

Algorithmic improvements include the addition of other local constraints types (only equation 4 is implemented now) and the ability to activate and/or deactivate specific reference gestures on the fly.

Some investigations are worth trying as far as the preprocessing is concerned: e.g. removing the gravity component to derive tilt-invariant features, testing different levels of downsampling, applying nonlinear quantification, etc. Some work could also be accomplished to improve post-processing: the single global distance threshold might be replaced by gesture-dependent threshold values and the measured time expansion/compression ratio could be taken into account.

5 ACKNOWLEDGMENTS

Research supported by `numediart`, a long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N°716631).

6 REFERENCES

[1] A. Y. Benbasat and J. A. Paradiso, "An inertial measurement framework for gesture recognition and applications," in *Gesture Workshop*, 2001, pp. 9–20.

[2] F. Bevilacqua, F. Guédy, N. Schnell, E. Fléty, and N. Leroy, "Wireless sensor interface and gesture-follower for music pedagogy," in *Proc. NIME '07*. New York, NY, USA: ACM, 2007, pp. 124–129.

[3] F. Itakura, "Minimum prediction residual principle applied to speech recognition," vol. 23, 1975, pp. 67–72.

[4] M. H. Ko, G. West, S. Venkatesh, and M. Kumar, "Using dynamic time warping for online temporal fusion in multisensor systems," vol. 9, no. 3. Elsevier Science Publishers B. V., 2008, pp. 370–388.

[5] R. Oka, "Spotting method for classification of real world data," vol. 41, no. 8, 1998, pp. 559–565.

[6] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," in *IEEE Trans. ASSP*, vol. 26, 1978, pp. 43–49.

[7] T. Todoroff, F. Bettens, W.-Y. Chu, and L. Reboursière, "Extension du corps sonore - dancing viola," in *Proc. NIME '09*, Pittsburgh, Pennsylvania, USA, 2009, pp. xx–xx.

[8] T. Todoroff and L. Reboursière, "1-d, 2-d and 3-d interpolation tools for max/msp/jitter," in *Proc. ICMC '09*, Montreal, Quebec, Canada, 2009, pp. xx–xx.