# COMPARING SYNTHETIC AND REAL TEMPLATES FOR DYNAMIC TIME WARPING TO LOCATE PARTIAL ENVELOPE FEATURES

*Joseph Timoney*[1], *Tom Lysaght*[1], *Victor Lazzarini*[2]

Dept. of Computer Science[1], Dept. of Music[2]
NUI Maynooth, Maynooth, Ireland
jtimoney@cs.may.ie

*Lorcan Mac Manus*

School of Electronic Eng.
DIT Kevin St., Dublin, Ireland
Lorcan.macmanus@dit.ie

## ABSTRACT

In this paper we compare the performance of a number of different templates for the purposes of split point identification of various clarinet envelopes. These templates were generated with Attack-Decay-Sustain-Release (ADSR) descriptions commonly used in musical synthesis, along with a set of real templates obtained using *k*-means clustering of manually prepared test data.

The goodness of fit of the templates to the data was evaluated using the Dynamic Time Warping (DTW) cost function, and by evaluating the square of the distance of the identified split points to the manually identified split points in the test data.

It was found that the best templates for split point identification were the synthetic templates followed by the real templates having a sharp attack and release characteristic, as is characteristic of the clarinet envelope.

## 1. INTRODUCTION

The envelope can be defined as the slowly varying evolution over time of the amplitude of a sound. It is one of the most defining timbral attributes of any sound [1]. Under the assumption that a sound can be modelled as an additive combination of more elementary partials, the overall sound envelope can be interpreted as the combination of the envelopes of the partials or additive components that form the sound. The characterisation of partial envelopes is a key feature of models of musical timbre [1]. Envelopes are generally categorised into three or four parts namely: the 'attack' portion, the 'decay' portion, the 'steady-state' or 'sustain' portion, and the 'release' portion [2], typically known as the ADSR (Attack-Decay-Sustain-Release) description. Although simple, this is a well-accepted model and is valid because research on timbre perception has identified the existence of distinct perceptual attributes associated with these categorizations [3]. Such a model is useful both in applications for the analysis and synthesis of timbres. For example, instrument classification [4] and additive synthesis engines [1].

Automatic classification of the time evolution of partial envelopes into ADSR portions has not been a subject of intense study, however. Often the identification of these time segments or split-points is done by eye [4]. Of the algorithmic techniques available the most well known uses a piecewise linear approximation-based model [5]. Another two methods outlined in [1] are termed the percent method and the slope method. These are based on identifying a peak in the envelope or its smoothed derivative respectively and then working backwards to determine the start points of attack segments based on an amplitude threshold. A similar idea is used to find the release. All of these methods were examined in [6] for the analysis of the partial envelopes of Irish tin whistle sounds. It was found that they suffered from drawbacks particularly when the partial envelope was not well defined due to contamination with tremolo or noise. The piecewise-linear approximation blindly searches for four segments but unfortunately the segments it identifies do not necessarily correspond to the ADSR portions. The percent and slope methods require many heuristics in their implementation to prevent incorrect segment identification, making it difficult to have complete confidence in their results when applied automatically.

To overcome the limitations of these techniques it was proposed to use a segment identification procedure based on template matching. Given that the ADSR description is a simple model and the partial envelopes are very variable, it was felt that a template ADSR curve, stretched or compressed to fit the partial envelope in some best sense should give a good approximation to the hypothesised underlying segments. As the segment boundaries of the template are known the warping path should automatically locate the corresponding segments on the partial envelope once the matching is performed. To stretch and compress the template so that it fits the partial envelope required a non-linear warping. This was achieved using the technique of Dynamic Time Warping (DTW), an algorithm that was the cornerstone of many isolated word recognition algorithms [7]. A block diagram of the technique is illustrated in Figure 1.

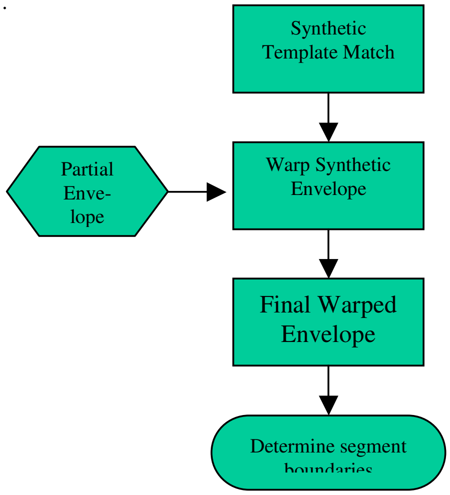

Figure 1: *Warp-based envelope analysis procedure.*

Initial experiments using a method that employed DTW for the analysis of the partial envelopes of tin whistle sounds suggested that it produced more reliable results [6]. In this work, however, only one template was used. Furthermore, the template was copied from a model described in [2]. However, the significant variability of partial envelopes suggests that one template is insufficient and it would be better to have a selection of templates of various shapes. Two possibilities are available from which a selection of templates can be created: the first possibility is to use piecewise linear descriptions of envelopes that are employed by musicians in sound synthesis applications; the second is to create templates from real sound envelope data. This paper intends to address the issue of which approach is best for partial envelope analysis using dynamic time warping. It first suggests how both types of templates can be generated. Then, experimental evaluation is employed to determine which template category provides the most satisfactory results.

## 2. TEMPLATE GENERATION

### 2.1 Synthetic Templates

The variability of partial envelopes makes it difficult to find a comprehensive rule for the creation of synthetic equivalents. Fortunately much practical investigation by musicians engaged in subtractive sound synthesis has resulted in ADSR templates that have been used for the creation of synthetic timbres equivalent to various acoustical instruments. In this work, the synthetic templates were designed following analysis of the ADSR sections of the patch diagrams for 13 well known instruments as given in [8]. A limitation of the diagrams was that only the timings for each stage of the envelope was given. Thus, although it could be assumed that the attack begins at zero amplitude and that the end of attack the envelope has reached a maximum value, there is no indication of the amplitude of the sustain portion of the envelope. Another difficulty is that the timing values given for the envelope stages are not given in absolute time units.

The templates were envisaged to be the combination of piecewise linear segments as assumed for real envelopes by [5]. This was done primarily to simplify the definition of the templates. Moreover, the key aspect of the template definitions was the relative location of the ADSR segments in time rather than their curvature as the DTW algorithm is sufficiently powerful to compensate for any inaccuracies in the template morphology. It was decided to interpret the timings for each ADSR envelope as being relative so that each timing would be viewed as a proportion of the total envelope duration (given by the sum of all the timings). To overcome the lack of a specified amplitude for the sustain, a number of envelopes were generated for each timing set with different sustain amplitudes. It was decided to calculate the sustain for 10 different amplitude levels ranging from 0.1 to 1 assuming that the maximum amplitude was 1. This resulted in a total of 130 synthetic envelope templates. A significant advantage of using such synthetic envelopes is that the length of templates can be adjusted to equal the length of the partial envelope under analysis just before the level of matching is assessed. The goodness-of-match of a template was assessed by computing the correlation coefficient between the partial envelope and the synthetic template. The best match was deemed to be the synthetic template that returned the largest correlation coefficient.

### 2.2 Templates generated from Real Data

The creation of templates from real data is of fundamental importance to those working in the field of statistical pattern recognition [9]. The choice of a most suitable procedure for building templates is dependent on whether prior knowledge is available of the number of categories or classes the data belongs to. In cases when the number of groups represented by the data is unknown this becomes an unsupervised learning problem. Such problems are usually tackled using clustering techniques. These are algorithmic approaches to partitioning the data into clusters or groups of relatedness based on some criterion. A very well-known and popular clustering algorithm is the *k*-means. This is an iterative approach whose goal is to partition the data into *k* groups such that the within-group-sum-of-squares is minimized. It requires external input to specify a value for *k*. The operation of the algorithm is as follows [10]:

Define a *d*-dimensional set of *n* data points $X = \{x_1, x_2, \ldots, x_n\}$ as the data to be clustered. Then define a *d*-dimensional set of *k* centres $C = \{c_1, c_2, \ldots, c_k\}$as the clustering solution that the iterative algorithm refines. The objective function that the *k*-means algorithm optimises is

$$KM(X,C) = \sum_{i=1}^{n} \min_{j \in \{1\ldots k\}} \left\| x_i - c_j \right\|^2 \qquad (1)$$

This objective function gives an algorithm that minimises the within-cluster variance (the square distance between each centre and its assigned data points).

The membership and weight functions for KM are

$$m_{KM}(c_l | x_i) = \begin{cases} 1 \,; if\ l = \arg\min_j \left\| x_i - c_j \right\|^2 \\ 0 \,; otherwise \end{cases} \qquad (2)$$

$$w_{KM}(x_i) = 1 \qquad (3)$$

The membership function defines the proportion of data points $x_i$ that belongs to centre $c_j$. *k*-means has what is termed a hard membership function because $m_{km}(c_l | x_i)$ can only take values of zero and unity. Thus, each data point is assigned to one centre $c_j$ exclusively which is determined by the centre that has the smallest Euclidean distance to the data point. The constant weight function of *k*-means renders all data points equally important.

The computational speed of the *k*-means algorithm can be slow for large datasets [11]. However, optimised versions have been proposed that offer significant computational savings. *k*-means has had a long association with dynamic time warping as it has been applied in procedures for the vector quantisation of feature vectors in speech recognition applications [12].

Although the popularity of *k*-means is a testament to its usefulness the one drawback of applying *k*-means to the clustering of partial envelopes is that the envelopes are rarely of equal length, a requirement of the algorithm. To impose this constraint, it has been deemed to be acceptable from other analyses of partial envelopes that faced a similar dilemma to interpolate or decimate the sustain portion of the envelope only [4]. Applying interpolation or decimation to the attack or release portions of the envelope is not justifiable as it corrupts the characteristics of the sound. Thus, before *k*-means clustering of partial envelopes to create templates the sustain portion must be identified and interpolation or decimation performed to ensure all envelopes reach a prescribed length. To ensure an accurate envelope segmentation the identification of the sustain portion must be carried out by hand.

## 3. EXPERIMENTAL SETUP

To investigate which category of template would be best in the dynamic time warping-based approach for envelope analysis the procedure outlined below was followed.

### 3.1. Envelope data compilation

A set of partial envelopes from real sounds had to be obtained first. The complete set of samples for an Eb clarinet were obtained from the University of Iowa Electronic Music Studios website [13]. There are 13 files altogether in Macintosh 'aiff' format, sampled at 44100Hz. These samples cover the range of the Eb clarinet from 'G3' up to 'C7'. They are recorded at 3 dynamic levels: '*pp*', '*mf*' and '*ff*' and are without vibrato. The sound of a clarinet can vary considerably from one note to the next, in general, however, the louder the note the faster the initial attack time. Furthermore, the clarinet is a sustaining instrument, meaning that as long as vibrating air is introduced into it, a tone will continue. Thus, a very broad characterisation of the envelope of a clarinet sound is that both the attack and release times are short and the decay and release are relatively longer [8]. As most of the sound files contained more than one note, note separation was required. This was achieved by the following method:

1) The note waveform was decimated by a factor of 2. The signal was normalised and filtered with a sixth order highpass Butterworth filter with a cutoff of 250Hz to attenuate noise introduced in the signal between the actual notes attributable to the decimation process. The signal was then quantized to a 6-bit resolution to push small signal values to zero, divided into non-overlapping frames of 75 samples in length, and the energy of each frame was found.

2) This energy representation was filtered with a 5-point median filter to smooth the energy values in the regions between the notes and to ensure a gradual roll-off from the large peaks associated with each note. All energy levels below 0.05 were located and flagged as being associated with the signal region between each note. These were marked using an indicator function that was unity in these regions and zero elsewhere.

3) The differential of the indicator function was found. Values of –1 in the differential represented the commencement of the region past the end point of each note, while values of 1 indicated the region before its beginning. Some heuristics were required to remove spurious non-beginning or endpoint values in this differential. With appropriate scaling of the differential indicator, the notes in each sound file were separated at the mid-point of each –1 and 1.

Each note was then analysed using the SMS algorithm of Serra using the implementation provided in [14]. Post-processing of the output of the SMS algorithm was done as follows:

1) The partial envelopes were converted from dB to linear scaling.

2) The matrix of partial tracks was examined to determine that the partial envelope values associated with each harmonic were consistent.

3) The envelopes whose power (given by the normalised envelope energy scaled by its length) was less than 0.15 were categorised as being ill-defined and noisy, and were excluded.

4) The beginning of each envelope was defined as the point where the differential of a 4-bit quantised version of the normalised envelope was greater then 0.0015. The inverse of this criterion was used to locate the endpoint of the envelope.

5) One observation on the output of the SMS implementation was that in the cases of very quiet or very high-pitched notes the algorithm can return envelopes that are incomplete having many zero values along their trajectory. To identify and remove these, each partial envelope was quantised to 6 bits and the number of zeros counted. If the number of zeros relative to the signal length exceeded 10% then this partial envelope was ignored.

The resulting set contained of a total of 344 clarinet envelopes. Each envelope was then analysed by eye to identify the four split-points:

- The start of the attack (SOA)
- The end of the attack (EOA)
- The start of the release (SOR)
- The end of the release (EOA)

All the information was stored in a database. Lastly, the partial envelopes were divided into equal size training and test sets. A discrete uniform distribution, scaled by the number of envelopes, was generated to provide a random set of index values for selecting which envelopes were included in each set.

### 3.2 Dynamic Time warping

Classic Dynamic Time Warping, as described by [7] aligns a template data series $\mathbf{x} = \langle x_i \rangle$, $i = \{1,\ldots,M\}$ with a reference data series $\mathbf{y} = \langle y_k \rangle$, $k = \{1,\ldots,N\}$ by finding the lowest cost path through the $\langle N \times M \rangle$ field, **F**, defined as:

$$\mathbf{F}_{i,k} = \mathrm{d}(x_i, y_k) \tag{4}$$

where $\mathrm{d}(x_i, y_k)$ is some appropriate distance metric, such as

$$\mathrm{d}(x_i, y_k) = |x_i - y_k| \tag{5}$$

To ensure only non-trivial solutions are found, a number of step constraints and path slope constraints are imposed. For each of these steps, there are two variations on how the path cost is accumulated. The symmetric form of the algorithm applies a weighting of 1 to a single vertical or horizontal step, and a weighting of 2 to a single diagonal step. Asymmetric form applies a weighting of 1 to a single horizontal or diagonal step, and a weighting of 0 to a single vertical step. Following the study [15], it was found that for Dynamic time warping applied to partial envelope analysis the best performing step constraint was the simplest, as shown in Figure 2,

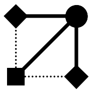

Figure 2: *Dynamic time warping step constraint.*

applied using the symmetric form of the algorithm. This meant that the matrix **G** containing the accumulated cost could be defined as:

$$\mathbf{G}_{i,j} = \min \begin{cases} \mathbf{G}_{i-1,j} + \mathrm{d}(x_i, y_j) \\ \mathbf{G}_{i-1,j-1} + 2\,\mathrm{d}(x_i, y_j) \\ \mathbf{G}_{i,j-1} + \mathrm{d}(x_i, y_j) \end{cases} \qquad (6)$$

The same Dynamic time warping was applied to the synthetic and real templates within this work.

### 3.2. Testing Procedure

To test the performance of the Dynamic time warping-based envelope analysis algorithm with both real and synthetic templates, a number of stages were devised.

The first stage was to extend or reduce the length of the training set of partial envelopes so they were of a uniform length. This was easily achieved because the envelope portion between the end of attack and beginning of release was marked. Cubic interpolation was employed to adjust the lengths. A commensurate adjustment was made to the points defining the attack and release. Following this, clustering of the envelopes using the *k*-means algorithm was performed and the attack and release segments of the clusters were identified.

The second stage proceeded in an iterative manner and was computed for each envelope in the test. Each envelopes length was calculated and from this the set of synthetic templates was generated. The best matching synthetic template to the test envelope was determined using the correlation coefficient. This synthetic envelope was then non-linearly stretched and compressed using Dynamic time warping to the test envelope. The goodness of fit between the warped envelope and test envelope was determined by evaluating (6) at $(i, j) = (M, N)$, i.e. at the final accumulated cost. Using the warping path returned from the Dynamic time warping algorithm, the location of the attack and release portions on the warped template was determined. Comparison could then be made as to the correspondence with the same points on the test envelope.

This iterative procedure was also applied to the real template envelopes that resulted from clustering. Interpolation of the sustain portion of the template envelopes was done to ensure the overall length of the template and test envelope were equal. Again, the goodness-of-fit was evaluated using (6) and the location of the points marking the attack and release portions on the warped template were found.

## 4. RESULTS

Figure 3 shows the accumulated DTW path cost for each of the template matching methods, i.e. Synthetic template and Real templates using 3, 5 and 7 clusters respectively. From this figure it can be seen that the accumulated path cost is highest with the synthetic template (solid line), followed by the real template with 3 clusters (dashed line) and the real template with 5 clusters (dot-dashed line). The lowest path cost is attained with the real templates using 7 clusters (dotted line).

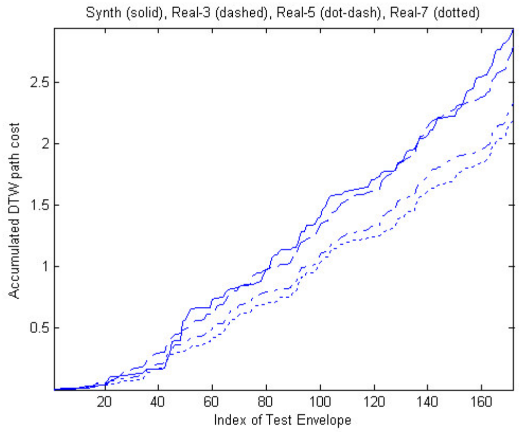

Figure 3: *Comparison of real and synthetic template fit.*

It is important to note that although the synthetic templates appear to perform poorly, this is largely due to the synthetic templates being more schematic in shape and hence having a higher DTW cost function. The real templates, being smoother in shape with less pronounced vertices require a lower overall warping to fit to the data. It is also intuitively clear that the templates having higher numbers of clusters (and hence a greater choice of matching template) should evaluate to a lower DTW path cost, and this is borne out by the data.

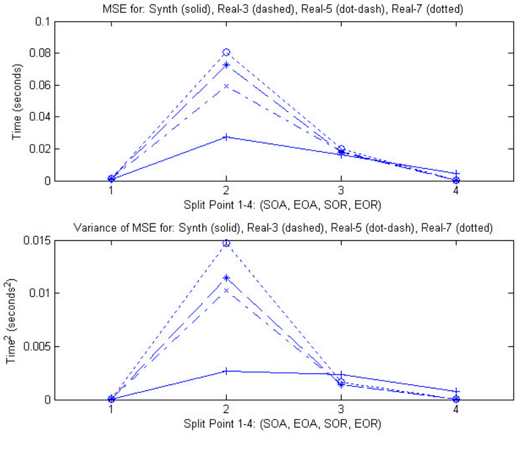

Figure 4: *MSE comparison across different template matching methods.*

The top plot in Figure 4 shows the mean squared error for the synthetic envelope (solid line), the real envelope using 3 clusters (dot-dashed line), the real envelope using 5 clusters (dashed line) and the real envelope using 7 clusters (dotted line) for each of the four split points SOA, EOA, SOR and EOR. The lower plot in Figure 4 shows the error variance for each of these envelopes (using the same legend).

As can be seen the synthetic envelope has a consistently lower mean square error and variance of error. This therefore provides the best identification of all of the split points. The real envelopes using 5 clusters are the next best performer, followed by the real envelope using 3 clusters and finally the real envelope using 7 clusters. This is particularly true for the EOA and the SOR, which are the two most pertinent points for envelope identification.

This may seem to be at odds with the results presented in Figure 3. However, it must be remembered that these results are simply stating that for the purposes of split point identification the synthetic envelopes with their sharper vertices at the split points force a truer warping to the actual split points in the original data. The synthetic envelopes therefore provide the best templates for split point identification.

Figure 5 below shows a histogram of the frequency of synthetic envelope selection. As can be seen, two envelopes in particular are most commonly chosen. For the purposes of clarifying the salient characteristic of the best-fit envelopes, these two envelops along with the two most popular envelopes from the 5-cluster set of real envelopes are plotted in Figure 6. The 5-cluster envelopes were chosen for this comparison, as these were the closest in performance to the synthetic envelopes for split point identification.

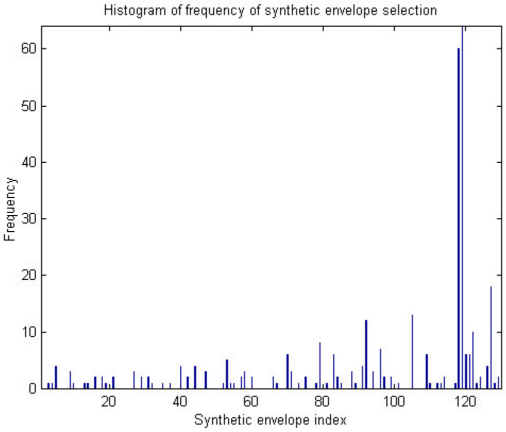

Figure 5: *Histogram of frequency of synthetic envelope selection.*

Figure 6 contains four plots. The solid lines show the most popular real and synthetic envelopes, and the dashed line shows the second most popular real and synthetic envelope. (The synthetic envelopes are the piecewise linear plots.)

As may be seen in Figure 6, all of these envelopes have a sharp attack and sharp release, along with a relatively flat sustain section. The more popular envelopes in each case can be seen to be the ones with sharper attack and release characteristics. This is characteristic of clarinet envelopes and so is consistent with what we might have expected.

## 5. CONCLUSIONS

In this paper we have compared a number of different templates for the purposes of split point identification of 172 envelopes from 119 clarinet notes. Some of these templates were synthetic templates of the type used for music synthesis. The other templates were generated using a *k*-means clustering algorithm on the original data. Of the real templates, the *k*-means algorithm was run to extract 3, 5 and 7 cluster templates.

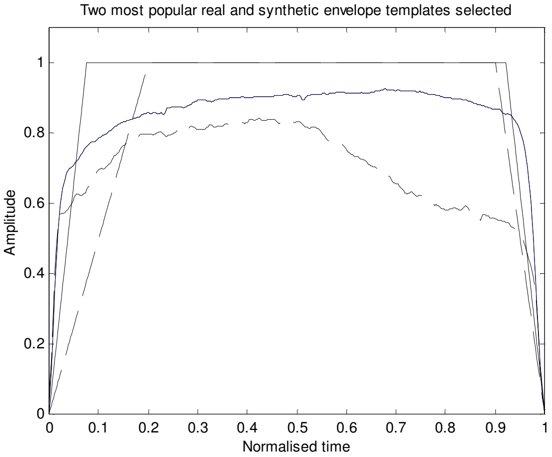

Figure 6: *Most popular envelopes selected.*

The DTW cost function appears to show that the real templates provided the best match. However, for the purposes of split point identification, the most successful templates were in fact the synthetic templates. This is due to the sharp characteristic of the split point boundaries in the synthetic envelopes. When the two most successful synthetic and real templates were plotted, it was found that they all share a sharp attack and release portion, which matches the typical clarinet envelope profile [8].

The next phase of this work will concentrate on extending this research to other families of musical instruments so as to extend the work initiated in [16] for timbre analysis of traditional instruments.

## 6. REFERENCES

[1] Jensen, K., “Timbre Models of Musical Sounds”, *Ph.D dissertation*, Department of Computer Science, University of Copenhagen, 1999.

[2] Helen, M., and Virtanen, T., “Perceptually motivated parametric representation for harmonic sounds for data compression purposes”, *DAFX03*, Queen Mary University of London, London, UK, Sept. 8-11, 2003.

[3] Grey, J. and Gordon, J., “Perceptual effects of Spectral Modifications on Musical Timbres”, *JASA*, vol 63, no. 5, 1978.

[4] McAdams, S., and Beauchamp, J.W., and Meneguzzi, S., “Discrimination of musical instrument sounds resynthesised with simplified spectrotemporal parameters”, *JASA*, vol. 105, no.2, Feb. 1999.

[5] Bernstein, A., and Cooper, E., “The piecewise linear technique of electronic music synthesis”. *Jnl. of the AES*, vol 24, no. 6 1976.

[6] Timoney, J., Mac Manus, L., Lysaght, T., and Schwarzbacher, A., “Dynamic Time Warping for Tin Whistle Partial Envelope”, *Irish Signals and Systems conference 2004*, Belfast, Northern Ireland, July 2004.

[7] Sakoe, H., and Chiba, S., ‘Dynamic programming algorithm optimization for spoken word recognition’, *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, Feb. 1978.

[8] Massey, H., et al., *A synthesist’s guide to acoustic instruments*. Amsco publications, New York, USA, 1987.

[9] Martinez, W. and Martinez, A., *Computational statistics handbook with MATLAB*. Chapman and Hall/CRC, Florida, USA, 2002.

[10] G. Hamerly and C. Elkan, “Alternatives to the k-Means Algorithm That Find Better Clusterings”, *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM'02)*, McLean, USA, November 2002, pp. 600-607.

[11] C. Elkan, “Using the Triangle Inequality to Accelerate k-Means”, *Proceedings of the Twentieth International Conference on Machine Learning (ICML'03)*, Washington, USA, August 2003, pp. 147-153.

[12] Deller, J., Proakis, J., and Hansen, J., *Discrete-time processing of speech signals*. Macmillan, New York, USA, 1993.

[13] University of Iowa Musical Instrument Samples available at http://theremin.music.uiowa.edu/MIS.flute.html

[14] Zolzer, U., ed., *DAFX Digital Audio Effects*. John Wiley and Sons, Chichester, UK, 2002.

[15] Timoney, J. et al., “An evaluation of warping techniques applied to partial envelope analysis”, submitted to *ICMC2005*, Barcelona, Spain, Sept. 2004.

[16] Timoney, J. et al., “Timbral attributes for objective quality assessment of the Irish tin whistle”, *DAFX 2004*, Naples, Italy, Sept. 2004.