Audio Engineering Society

Convention Paper

Presented at the 121st Convention
2006 October 5–8 San Francisco, CA, USA

Modeling musical articulation gestures in singing voice performances

Esteban Maestre[1], Jordi Bonada[1], and Oscar Mayor[1]

[1]*Music Technology Group, Institut Universitari de l'Audiovisual, Universitat Pompeu Fabra, Ocata 1, 08003 Barcelona, SPAIN*

Correspondence should be addressed to Esteban Maestre (emaestre@iua.upf.edu)

ABSTRACT

We present a procedure to automatically describe musical articulation gestures used in singing voice performances. We detail a method to characterize temporal evolution of fundamental frequency and energy contours by a set of piece-wise fitting techniques. Based on this, we propose a meaningful parameterization that allows reconstructing contours from a compact set of parameters at different levels. We test the characterization method by applying it to fundamental frequency contours of manually segmented transitions between adjacent notes, and train several classifiers with manually labeled examples. We show the recognition accuracy for different parameterizations and levels of representation.

1. INTRODUCTION

Singing voice is known to be the most complex musical instrument in terms of timbre variability and articulation control, so natural singing voice becomes one of the most challenging aspects of musical instrument synthesis. Recent research on sample-based singing voice synthesis has reached high quality voice models [2], but the inclusion of a model for the articulation gestures, understood for instance as transitions from note to note, remains still an open problem. In response to this, different articulation gestures (e.g. legato, staccato) are sampled and stored in the database, while some other (e.g. vibrato) are generated by adding them to the melody contour. These limitations restrict the naturalness and flexibility of synthesis, making difficult to transform stored articulations or create new ones for a specific context, due mainly to the lack of a quantitative description. It is found also that in the case of a physical model-based synthesizer [5], where the input controls are more related to the voice production and articulation, it is hard to render the nuances taking place during performance. Here we focus on the analysis and characterization of continuous parameters extracted from real audio performance recordings, important for any improvement of current syn-

thesis methods.

Several approaches have been taken in order to model nuances of audio perceptual parameters in voice analysis. Author in [1] used cubic Bézier to model the evolution of amplitude, fundamental frequency and spectral centroid. However, the author does not focus on any structured analysis of the performance, such as transitions or attacks. Authors in [10] modified the content of a performance in order by removing some gestures such as overshoots or vibratos from a real performance. Still, they don't describe quantitatively those gestures, but find a mathematical formulae able to produce restricted shapes in fundamental frequency contours in order to measure how the presence of those gestures affected the mood perception in some listening tests. The idea of using Bézier curves has been succesfully used for modeling prosodic features in spoken language [6].

Restricting musical articulation gestures in singing voice to be mainly conveyed by pitch and amplitude modulations, we present here a quantitative characterization of fundamental frequency and energy contours of manually segmented articulations, understood as note-to-note transitions. We model contours by a set of piece-wise fitting techniques and a meaningful parameterization, consisting in linear approximation and Bézier spline approximation, both with several levels of representation. One important aim of the characterization method is the possiblilty to be used both in recognition and synthesis stages, so that still keeping a reduced number of parameters, contours can be reconstructed with enough accuracy. As an experiment, we use the characterization method for the classification of transitions in terms of fundamental frequency contours of manually labeled transitions between adjacent notes. We show the recognition accuracy for three different classification methods and four different parameterizations.

The structure of the paper is as follows. First, we give a short overview of Bézier curves and Bézier splines. Then, we give some details about the database we used for our experiments, and we describe shortly how we prepare our data. In Section 4, we give the details of our description procedure. After that, an experiment on transition classification is reported in Section 5. Finally, some conclusions and further work are pointed out in Section 6.

2. BÉZIER SPLINES

Since in the 70's were used for body design in automotive industry, *Bézier curves* have obtained dominance in the typesetting industry. Consider $N+1$ control points $p_k,\ k \in \{1, 2, .., K\}$ in n-dimensional space. The Bézier parametric curve function $B(u)$ is of the form in equation 1, where $0 \leq u \leq 1$.

$$B(u) = \sum_{0 \leq k \leq N} p_k \frac{N!}{k!(N-k)!} u^k (1-u)^{N-k} \qquad (1)$$

$B(u)$ is a continuous function in n-dimensional space defining the curve with $N+1$ discrete control points $P_K = \{p_0, p_1, .., p_N\}$. $u = 0$ at the first control point ($k = 0$) and $u = 1$ at the last control point ($k = N$). The curve in general does no pass through any of the control points except the first and last. From the formula, $B(0) = p_0$ and $B(1) = p_N$. The curve is always contained within the convex hull of the control points, so it never oscillates wildly away from the control points. For the case of four control points, curves are usually called *cubic Bézier curves*, and the formula reduces to the expression of equation 2. An example showing a two-dimensional Bézier curve with four control points is shown in Figure 1.

$$B(u) = p_o(1-u)^3 + 3p_1 u(1-u)^2 + 3p_2 u^2(1-u) + p_3 u^3 \qquad (2)$$

As the number of control points increases, it is necessary to have higher order polynomials and possibly higher factorials. It is common therefore to piece together small sections of Bézier curves to form a longer and more complex curve. This also helps control local conditions, because normally in Bézier curves, changing the position of one control point will affect the whole curve. Of course since the curve starts and ends at the first and last control point it is easy to physically match the sections. Multiple curve pieces can be joined together to form longer C_1-continuous curves. The curve is made C_1-continuous by the setting the tangents the same at the join, as it is illustrated in Figure 2. In this case, each piece of the curve is defined by u ranging from 0 to 1. See Figure 2. A complete overview of Bézier curves and splines can be found in [2].

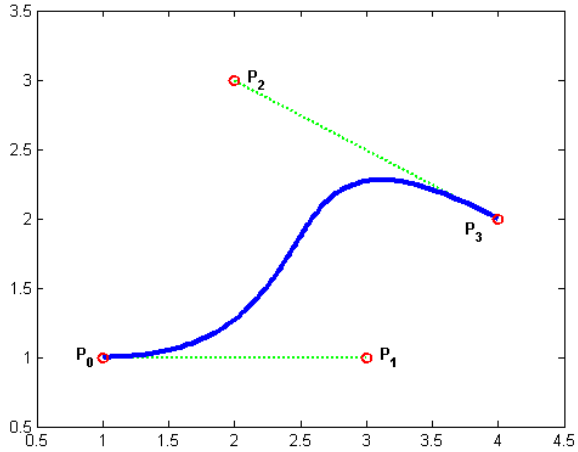

Figure 1: Two-dimensional cubic Bézier curve defined by the start and end points $p_0 = \{1, 1\}$ and $p_3 = \{4, 2\}$, and two control points $p_1 = \{3, 1\}$ and $p_2 = \{2, 3\}$.

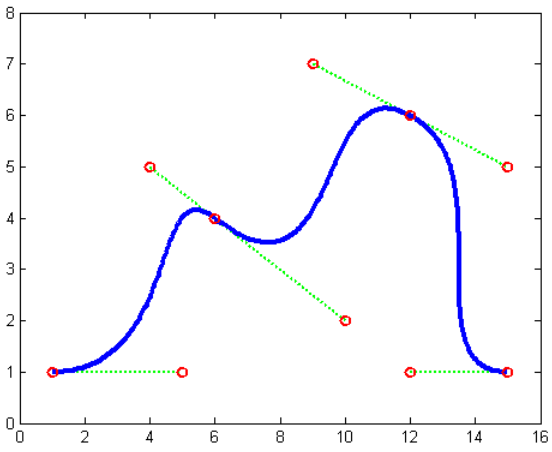

Figure 2: Piecewise cubic Bézier curve with three concatenated cubic curves. Note the smooth junctions between curves, by setting control points to be on the same tangent.

3. DATA PREPROCESSING

For carrying out this work, we used a small database of 4 popular song excerpts, performed by different trained singers. For each excerpt we have the voice solo recording and its corresponding original score, stored in MIDI format. The segmentation has been carried out as follows. First, the performances have been automatically segmented into notes by means of an alignment to the original score. In a second step, note-to-note transition articulations are manually segmented following well defined criteria. The total number of transitions is 332. We used the software SMSTools2[1] for carrying out the automatic alignment and segmentation, and also for the extraction of energy and fundamental frequency frame-by-frame contours, hop-size of 512 samples. We express fundamental frequency in *cents* and energy in *dB*. We also smoothed contours and applied some preprocessing them in order to avoid some discontinuities and modulations due to unvoiced consonants.

3.1. Note segmentation

Performance recordings are automatically segmented into notes using Hidden Markov Models (HMMs) with note duration models, similar to the method described in [4], but including some heuristic rules [9]. One of the principles of this segmentation algorithm is the assumption that the singer does not make use of ornaments, nor consolidation of notes. Once note boundaries are obtained, they are revised manually in order to correct possible errors of the algorithm. Then, transitions are manually segmented looking mainly at the fundamental frequency contour and note boundaries. Roughly, the criteria used can be summarized as the following:

- We consider a transition as a portion of sound shared by two successive notes.
- We assume that the singer is not inserting silences between notes that appeared together in the original score.
- The note boundary is always included in the transition segment.
- Fundamental frequency values at transition boundaries are close to the quantized MIDI pitch of the notes involved.
- Fundamental frequency discontinuities due to phonetics (see Section 3.2) are always included in the transition.

[1] developed in the Music Tenchology Group, Universitat Pompeu Fabra, as an internal project

An example of transition segmentation is depicted in figure 3, where we can see the audio waveform and note segmentation at the top, and pitch contour and transition segmentation at the bottom.

3.2. Contour preparation

The gross contour of the voice pitch, which we may view as a sort of macro-intonation, is decided by the composer. The singer adds emotion in the domain of the micro-intonation. However, some micro-intonation nuances are not caused by the performer in order to give any expression or style, but by the voice production mechanisms in some phonetic contexts [11]. Thus, singing voice presents some special characteristics, due mainly to phonation, that might be considered in a special way. Some consonants constrain the contours of amplitude and fundamental frequency independently from the intentions of the singer. The most clear example of this is the case of the pronunciation of non-pitched consonants as it is the case of fricatives or plosives. Minor modulations due to voiced consonants have not bee taken into account. That is what some people from the speech community call the 'micro-prosody' [7]. An example showing this effect is depicted in Figure 4, where both fundamental frequency and energy contour have been interpolated by means of a cubic Bézier curve (see below for a detailed explanation of the interpolation procedure).

Contour preparation consists of two main steps, carried out for the whole utterance at a time. The first step consists in smoothing contours. For doing so, we filter the contours by applying a gaussian window of a size of five frames. Then, in a second step, we deal with the non-pitched regions by applying some interpolation to the contours. This is carried out for refilling the missing values of fundamental frequency contours and for avoiding important nuances in the energy contour due to pronunciation of unvoiced consonants. We interpolate contours by means of a cubic Bézier curve (see Section 2). We set the tangents at the initial and final points of the non pitched region as the regression slope of the contour beyond the left and right transition boundaries, respectively. We compute these slopes by taking a small window of three frames before the boundary and after the boundary. Then, we set the position of each inner control point of the cubic Bézier along its corresponding tangent (computed before), at a distance from its outer point equal to the distance between the the limits of the non-pitched region. We found this approach resulting in well continued contours. Additionally, in order to remove the nuances that the pronunciation of unvoiced consonants produce in the energy envelope, we substitute the energy contour in the non-pitched region by a cubic Bézier computed in the same manner as for fundamental frequency contour (see Figure 4).

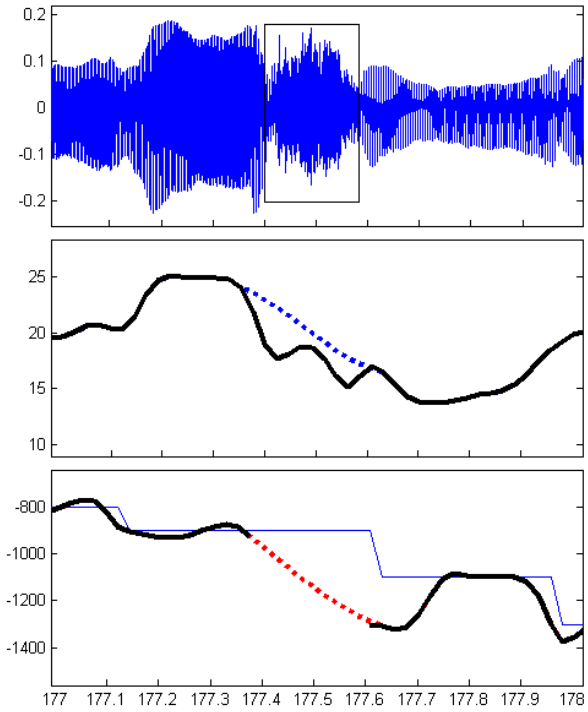

Figure 4: Interpolation of fundamental frequency and energy contours in a non-pitched region. From top to bottom: audio and square indicating an unvoiced consonant, energy contour, and fundamental frequency contour over the quantized MIDI notes. Thick sloid lines represent the original contour, while dashed lines represent the interpolation.

4. DESCRIPTION

When designing the representation procedure, we aimed at getting a meaningful and compact representation that allows to reconstruct easily a contour from the model parameters in order to apply it for

Figure 3: Screenshot of SMSTools2 and note and transition semgentation. At the top, it is shown the waveform and the note boundaries. At the bottom, one finds both fundamental frequency contour, and transitions represented as a rhombus.

synthesis. This led us to decide a piece-wise representation using constrained Bézier splines, always ensuring continuity between adjacent pieces. We discarded the use of polynomials due to the lack of local control. The number of approximating pieces is supplied in advance, so a multi-level representation is obtained for each contour. The fitting algorithm is fed with the normalized contours as they have been extracted from the segmented performances. Before that, a set of landmarks is extracted from each contour by studying extrema and curvature characteristics. During the fitting process, piece-to-piece divisions are allowed to fall only at landmark positions, in order to reduce the search.

4.1. Landmark extraction

In order to reduce the computational cost needed for fitting the pieces, we decided to extract a set of landmarks from the contour by studying zero-crossings of smoothed versions of the first and third derivatives. We take this decision based on the assumption that piece-to-piece boundaries will be situated at points with high curvature, thus we look for extrema, both along the contour, and along the second derivative of the contour. Then, during the fitting process, these boundaries are allowed to fall only at landmark positions (see next Section). The procedure is carried out in two steps, performed again to the whole utterance. In a first step, we compute the first three derivatives and find the zero-crossings of the first derivative and third derivative, projecting their positions onto the original envelope. These points on the envelope constitute a first set of landmarks. A detailed explanation of the first procedure of landmark extraction can be found at [8].

Then, in a second step, we perform cleaning of landmarks based on their importance given their context. For each landmark, we join a line connecting its predecessor and its successor. Then, we measure the y-axis distance Δy between the landmark and its projection on such line. We discard those landmarks for which Δy is less than a pre-fixed threshold (see Figure 5). We have set this threshold empirically. For the fundamental frequency we set this threshold to be equal to 25 *cents*, i.e. a quarter of a semitone. For the case of energy, we have set the threshold to 1 dB.

4.2. Fitting process

For approximating contours, we chose two differ-

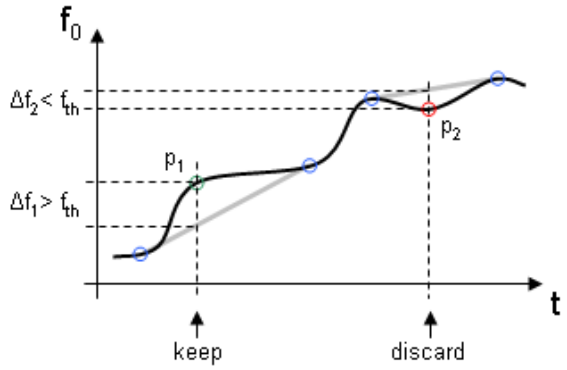

Figure 5: Illustration of the landmark cleaning step for the case of fundamental frequency contour, where two landmarks p_1 and p_2 are considered. Landmark p_1 is kept.

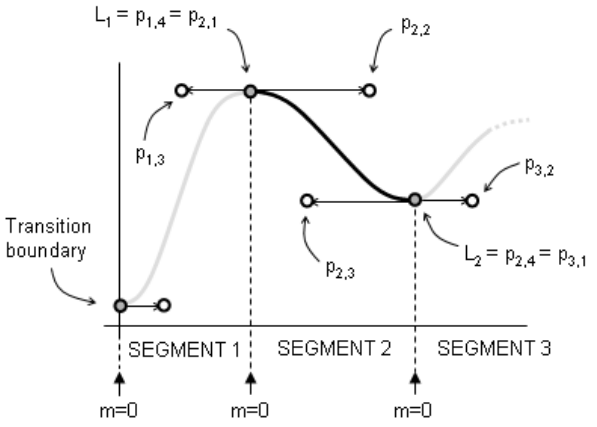

Figure 6: Illustration of the concatenation of several Bézier curves, and the tangent constrain at each junction point

ent representations from which we are able to reconstruct the approximated contours with different accuracy. We approximate contours by (1) cubic Bézier segments, and (2) linear segments. Each contour (energy and fundamental frequency) is approximated by a finite number of concatenated segments. The number of concatented segments is supplied in advance. In the second case, which we will explain in detail next, contours are reconstructed by a constrained set of Bézier curves concatenated as explained in Section 2. The constrains, along with the fitting algorithm, are stated next.

Each concatenation point between segments will correspond to one of the previously extracted landmarks L. The tangents at each junction point are set to present zero-slope, so that the closest inner control points of adjacent curves present the same y-value, and equal to the value of the contour at that landmark. This is illustrated in Figure 6 by the two control points $p_{1,3}$ and $p_{2,2}$, conforming the tangent at the landmark L_1, and the two control points $p_{2,3}$ and $p_{3,2}$, conforming tangent the landmark L_2. This is expressed as in equation 3 and 4, where y represents the y-value of the position af junction points and control points.

$$p_{i+1,2,y} = p_{i,2,y} = L_{i,y} \quad (3)$$

$$L_{i,y} = p_{i,4,y} = p_{i+1,1,y} \quad (4)$$

For each junction point L_i, we define the x-distance between the control points defining the tangent (and thus how much it is approached by the approximating curve), by means of a strength s_i, being $0 \leq s_i \leq 1$. This only parameter influences both x-distances between junction points L_i and its adjacent control points $p_{i-1,3}$ and $p_{i,2}$, i.e, the x-position of the second control point of the $(i+1)-th$ Bézier curve will range from the x-position of the $i-th$ junction point L_i to the x-value of the next junction point L_{i+1}, depending on the value of S_i. This is expressed for both control sides of the junction point L_i as in equation 5 and 6.

$$p_{i+1,2,x} = L_{i,x} + S_i(L_{i+1,x} - L_{i+1,x}) \quad (5)$$

$$p_{i,3,x} = L_{i,x} - S_i(L_{i,x} - L_{i-1,x}) \quad (6)$$

The strengths at the boundaries of the transitions are set to $S_0 = 0.25$ and $S_{N+1} = 0.25$, being N the number of approximating segments. This has been decided empirically with the aim of keeping the number of parameters sufficiently low, and taking into account that these two strengths do not play an important role in the description of the shape.

Then, given the boundaries of the transition, a representation of each approximated contour will depend on the number of segments N used for the approximation, the $N-1$ junction points L, and the

corresponding strengths S. This is expressed as in equation 7.

$$BC = f(N, L, S) \quad (7)$$
$$L = L_1, .., L_N$$
$$S = S_1, .., S_N$$

The number of segments used to approximate contours is set in advance, and will serve as an algorithm parameter, setting the level of representation. Then, for fitting each contour, we look for (1) the N positions of the junction points L among the set of extracted landmarks L, and (2) the values of the strengths S applied to each point, that minimize (see equation 8) the squared approximation error ϵ_C of the original contour C.

$$\underset{L,S}{\operatorname{argmin}}\, \epsilon_C(L, S) \quad (8)$$

The error approximation will then depend just on the selected junction points L and the corresponding strengths S, as expressed in equation 9.

$$\epsilon_C(L, S) = \sum_{1 \le t \le T} (BC(L, S, t) - C(t))^2 \quad (9)$$
$$L, S fixed$$

In order to find a solution for this optimization problem, we have proceeded with an iterative fitting of both L and S for a given N. The possible values of each pair x and y (one pair defining each junction point) correspond exactly with some of the positions of the already extracted landmarks, so there are finite possibilities where to look for. Conversely, the possible values of each strength S range continuously from 0 to 1, so we have quantized them linearly to ten values in order to be able to find an approximate solution.

As already pointed out above, we also computed a linear approximation of the contours, by setting all strengths S equal to zero and removing them from the search. In such case, we join each of the junction points L by means of linear segments. An example of reconstructed approximation of a fundamental frequency contour for a particular transition is depicted in Figure 7, where N+1=3 and N+1=4 approximating segments have been used, being N the number of landmarks used.

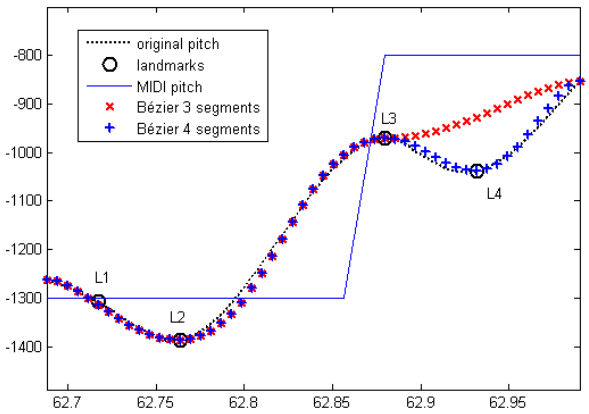

Figure 7: Reconstruction of fundamental frequency contour. For the case of 3 concatenated Bézier curves, selected landmarks have been L_2 and L_3, while for the case of 4 concatenated Bézier curves, L_2, L_3, and L_4 have been selected.

5. EXPERIMENT

As an experiment, we classify transitions depending based on fundamental frequency contour description. By exhaustive observation of fundamental frequency contours of the transitions present in our performance, we observed three main different classes of transitions between notes. We assigned to them the following labels: *normal*, *portamento*, and *scoop*. Next we describe briefly what characterizes these empirically distinguished transition types.

- **Normal:** No extra pitch fluctuation is found, apart from the path from one note to the following during the whole transition.
- **Portamento:** Target pitch is already reached before vowel onset corresponding to note change.
- **Scoop:** After the vowel onset, pitch is maintained for several hundreds of milliseconds at a higher or lower value before reaching target pitch.

Before characterizing the segments, we join the outer boundaries of the contour by means of a linear segment with initial and final values equal to the initial and final values of the contour. Then, for each of the selected landmarks, the pitch value of its position is subtracted from pitch value of the linear segment at the same position (x-value). This ad-hoc normalizing preprocessing is carried out in order to avoid the possible differences among transitions in which origin and target pitch differences are just one semitone, or maybe a complete octave.

Feature vectors consist first in the duration of the transition, and the relative position of the note change. Then, depending on the level of representation, $N-1$ sub-vectors of features are added, being N the number of segments used for approximate the segment. This subset will consist in the value x and y of the junction points selected by the fitting algorithm detailed above. The x-value is taken as relative to the position of the note change, while for the y-value, we use the normalized values as explained in previous paragraph. For the case of Bézier-spline representation, $N-1$ strength values are added to each subset.

We trained three different supervised classifiers by means of the tool WEKA [12]. The classification methods we used are: Naïve Bayes Classifier (NB), Multi-layer Perceptron Neural Network (MP), and k-Nearest Neighbor Classifier (k-NN). For the MP classifier, we used a hidden layer of twenty perceptron neurons, while for the k-NN classifier we set the number of neighbors to three. No significant improvements were found by modifying algorithm parameters. We used a total of 332 transitions, from which we labeled by hand 258 articulations as *normal*, 59 as *scoop*, and 15 as *portamento*. This leads to a baseline of 73.2%. We show in Table 1 the recognition accuracy for each one of the three classifiers and four different levels of representation, including two and three linear segments (Lin-2 and Lin-3), and two and three cubic Bézier segments (Bez-2 and Bez-3).

We found high recognition results for all representation levels and classification methods, with no very significant differences. Multi-layer Perceptron classifier (MP) showed to perform slightly better. In terms of representation level, it is found that the more complex is the description, the worst is the performance of the classifier. This can be understood as an explanation of the low complexity of the visually perceptual attributes used when looking at pitch contours during manual labeling. If we look into individual class recognition results, it is found high recognition accuracy for both *normal* and *scoop* classes. Conversely, class *portamento* performed poorly in all experiments. We show an example in Table 2. We can explain this fact by looking at the low number of instances belonging to this class.

Classification accuracy (%)			
Representation	**NB**	**MP**	**k-NN**
Lin-2	90.37	91.26	90.96
Lin-3	86.44	90.36	89.45
Bez-2	90.06	90.36	90.67
Bez-3	87.95	88.26	87.65

Table 1: Classification accuracy for different classification methods (NB: Naïve Bayes, MP: Multi-layer Perceptron, k-NN: K-Nearest Neighbor) and different levels of representation.

k-NN, Bez-2	**normal**	**scoop**	**portamento**
normal	243	10	5
scoop	8	51	0
portamento	12	0	3

Table 2: Confusion matrix for the case of k-NN classifier and using two cubic Bézier segments to represent the contour.

6. CONCLUSION AND FUTURE WORK

We have presented a method for the characterization of fundamental frequency and energy contours of singing voice musical performance articulations. However, this procedure could be used to model the temporal evolution of other performance-related parameters. We have outlined the different steps of data preparation, and the fitting procedure to represent contours by means of several concatenated Bézier curves. We have also shown the results obtained by testing the representation procedure in a transition classification experiment.

The phonetic transcription has not been taken into account, but it is pretended to include it as a way to help in an automatic segmentation process. Moreover, the analysis of micro-prosodic features related

to phonetics will definitely help in improving the classification accuracy, since this issue is currently introducing some noise in the contours, which reduces the recognition accuracy. More extended tests, with more different labels and other segments of interest, and dealing also with energy representation are to be carried out. Increasing database size is an important need for improving the results. We have used some adhoc definitions for different transitions types, but the intention is to increase the database and discover transition types by means of clustering techniques.

Possible applications of this modeling procedure, still in a very preliminary stage, are tools for singing education, natural singing voice synthesis, singer identification tasks, and also tools for emulating a reference singer. We plan to transform some real performances, by means of pitch shifting and amplitude scaling, for applying the articulation description obtained by our procedure, and compare the resulting performances to the original recordings.

7. ACKNOWLEDGMENT

This work has been partially funded by the Spanish TIC2003-07776-C02-02 project PROMUSIC. The authors would like to thank Perfecto Herrera for his useful advices.

8. REFERENCES

[1] Battey, B., "Bezier Spline Modeling of Pitch-Continuous Melodic Expression and Ornamentation", Computer Music Journal 28,4, 2004

[2] Bonada, J. and Loscos, A., "Sample-based singing voice synthesis based in spectral concatenation", Stockholm Music and Acoustics Conference, 2003, Stockholm, Norway

[3] P. Bourke, "Computer Graphics. Lecture notes", Centre for astrophysics and supercomputing, Swinburne University of Technology, 2000, Melbourne, Australia

[4] Cano, P. and Loscos, A. and Bonada, J., "Score-Performance Matching using HMMs", Proceedings of International Computer Music Conference, 1999, Beijing, China

[5] P. R. Cook, "Identification of Control Parameters in an Articulatory Vocal Tract Model, With Applications to the Synthesis of Singing", Department of Electrical Engineering. Stanford Center for Computer Research in Music and Acoustics, USA, 1991

[6] D. Escudero and V. Cardeñoso, "Corpus based extraction of quantitative prosodic parameters of stress groups in Spanish", Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2002

[7] Fujisaki H. and Ohno S., "Prosodic parameterization of spoken Japanese based on a model of the generation process of F0 contours", Proceedings of the International Conference on Spoken Language Processing, 1996

[8] Maestre, E. and Gómez, E.,"Automatic characterization of dynamics and articulation of monophonic expressive recordings", Procedings of the 118th AES Convention, 2005, Barcelona, Spain

[9] Mayor, O., Bonada, J. and Loscos, A. "The singing tutor: Expression Categorization and Segmentation of the Singing Voice", Proceedings of the AES 121st Convention, 2006. San Francisco, USA

[10] T. Saitou and M. Unoki and M. Akagi, "Extraction of F0 dynamic characteristics and development of F0 control model in singing voice", Proceedings of the International Conference on Spoken Language Processing, 2002

[11] Sundberg, J., "The science of the singing voice", Northern Illinois University Press, 1987.

[12] Witten, I.H. and Eibe, F., "Data Mining, Practical Machine Learning Tools and Techniques with Java Implementation", Morgan Kaufmann Publishers Inc., 1999