DATA-DRIVEN STATISTICAL MODELING OF VIOLIN BOWING GESTURE PARAMETER CONTOURS

Esteban Maestre

Music Technology Group, Universitat Pompeu Fabra, Barcelona, SPAIN

esteban.maestre@upf.edu

ABSTRACT

We present a framework for modeling right-hand gestures in bowed-string instrument playing, applied to violin. Nearly non-intrusive sensing techniques allow for accurate acquisition of relevant timbre-related bowing gesture parameter cues. We model the temporal contour of bow transversal velocity, bow pressing force, and bow-bridge distance as sequences of short segments, in particular Bézier cubic curve segments. Considering different articulations, dynamics, and contexts, a number of note classes is defined. Gesture parameter contours of a performance database are analyzed at note-level by following a predefined grammar that dictates characteristics of curve segment sequences for each of the classes into consideration. Based on dynamic programming, gesture parameter contour analysis provides an optimal curve parameter vector for each note. The information present in such parameter vector is enough for reconstructing original gesture parameter contours with significant fidelity. From the resulting representation vectors, we construct a statistical model based on Gaussian mixtures, suitable for both analysis and synthesis of bowing gesture parameter contours. We show the potential of the model by synthesizing bowing gesture parameter contours from an annotated input score. Finally, we point out promising applications and developments.

1. INTRODUCTION

Interaction complexity between performer and musical instrument stands out when dealing with excitation-continuous musical instruments, by which sound bending is achieved with continuous modulations of the physical actions directly involved in sound production mechanisms, i.e. instrumental gestures [2]. Because of the complex and continuous nature of gestures involved in the control of bowed-string instruments (often considered among the most articulate and expressive), analysis of bowed-string instrumental gestures has been an active and challenging topic of study for several years. Particularly for the case of violin, recent studies have taken the advantage of currently available motion tracking and force sensing techniques for providing accurate gesture data [16] [10] [4].

Violin bowing gesture data analysis has been approached recently by several works. Authors in [13] used bow acceleration extrema for automatic bow stroke classification. In a similar fashion, the work in [15] extends the classification to different bowing techniques by extracting the principal components of raw acceleration and strain gage sensor data. None of these approaches is aligned towards a generative model able to also provide means for the automation of bowing gesture rendering.

A first attempt to create synthetic bowing gesture parameter contours from an annotated score is found in [3], where the author presents an algorithm for rendering a number of violin performance gesture parameter contours (including both left and right hand) by concatenating short segments following a number of hand-made rules. Following the same line, extensions dealing with left hand articulations and string changes were introduced in [7]. Both approaches lack real performance data -driven definition of segment contours parameters. A more recent study working with real data is found in [4], where bow velocity and bow force contours of different bow strokes are quantitatively characterized and reconstructed mostly using sinusoidal segments. Flexibility limitations of the proposed contour representation may impede to easily generalize its application to other bowing techniques.

Author in [9] points directions towards a general framework for the automatic characterization of real instrumental gesture cue contours using parametric Bézier curves, foreseeing them as a more powerful and flexible basis for contour shape representation (see their use for speech prosody modeling in [5]). Aimed at providing means for reconstructing contours by concatenating short curve units, implied a structured representation as opposed as the work presented in [1] dealing with audio perceptual attributes. Later, authors in [11] used Bézier concatenated curves for pursuing a model for different note-to-note articulation classes in singing voice performance.

In this paper, we present a general and extensible framework for modeling bowing gesture parameter contours (bow velocity, bow force, and bow-bridge distance) for different bowing techniques and performance contexts using concatenated Bézier cubic curves. A key aspect resides on the fact that curve parameter extraction is carried out automatically, providing a representation usable both in gesture analysis

and synthesis applications. In Section 2, we present the methodology followed for carrying out contour parameter automatic extraction. Section 3 gives details on the construction of a statistical model of contour parameters. A bowing gesture parameter contour synthesis application is outlined in Section 4. We conclude by pointing out possible extensions and applications.

2. CONTOUR ANALYSIS

2.1. Corpus

Bowing gesture data acquisition was performed by means of a commercial EMF device as reported in [10], extracting bow force by applying the techniques presented in [6] and [4]. Recording scripts (including both exercises and short musical pieces) were designed to cover four different articulation types (*détaché*, *legato*, *staccato*, and *saltato*), three different dynamics, and varied note durations in different performance contexts (attending to bow direction changes and rests). Score-performance alignment was carried automatically by means of a dynamic programming (based on the *Viterbi* algorithm [14]) adaptation of the procedure introduced in [10] plus manual correction when needed for ensuring the appropriate segmentation of bow velocity, bow force, and β ratio contours (the latter obtained from acquired bow-bridge distance and performed string and pitch) of around 10K notes.

2.2. Score Annotation -Based Note Classification

Attending to different score-annotation based characteristics of the notes in the corpus, we perform a classification that will define different classes of notes for which specific gesture models will be later constructed. The basis for classifying note samples is divided into two main groups: *intrinsic* aspects and *contextual* aspects.

Intrinsic characterstics
[ART] Articulation type: {*détaché legato staccato saltato*}
[DY] Dynamics: {*piano mezzoforte forte*
[BD] Bow direction: {*downwards upwards*}

Contextual characteristics
[BC] Bow context: {*init mid end iso*}
[PC] Phrase context: {*init mid end iso*}

Considering *intrinsic* note characteristics, first and most important is the articulation type. We have considered four different articulations: *détaché*, *legato*, *saltato*, and *saltato*. Three different dynamics are present in the corpus: *piano*, *mezzoforte*, or *forte*. The possible bow directions are *downwards* and *upwards*.

In terms of what we call *contextual* characteristics, we are considering two main aspects: which is the position of a note within a bow (e.g. in *legato* articulation, several notes are played successively without any bow direction change), and which is the position of a note with respect to *rest* segments (e.g. silences). For the case of *bow context*, we classify a note as *init* when, in a succession of notes sharing the same bow direction, is played first. A note is classified as *mid* when is played neither first nor last. The class *end* corresponds to notes played in last sequence order, while notes appearing as the only notes within a bow (e.g. in *détaché* articulation) are classified as *iso*. Analogously to the case of bow context, for what we called *phrase context*, we look at successions of notes with no *rest* segments or silences in between. Classified as *init* will those preceded by a silence and followed by another note, as *mid* those preceded by and followed by a note, as *end* those preceded by a note and followed by a silence, as *iso* those surrounded by silences.

Each feasible combination of the classes above represented will lead to a note gesture class C_i characterized by the tupla in (1). Note that not every possible combination of any of the classes considered within each of the five characteristics is feasible in practice.

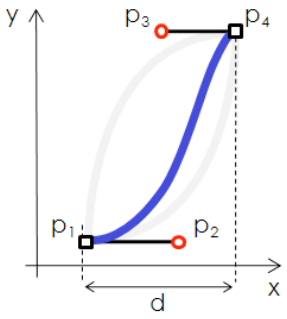
$$C_i = [\, ART_i \;\; DYN_i \;\; BD_i \;\; BC_i \;\; PC_i \,] \tag{1}$$

Since both *dynamics type* and *bow direction* will be present in any combination of *articulation type*, *bow context* and *phrase context* (the corpus covers all feasible combinations), possible classes will depend on upon the possible combinations of the last three aspects. This leads to a total of 17 combinations, which multiplied by three dynamic types and by two bow directions results into a total of 102 note gesture classes. Using the collected bowing gesture parameter cues for each class, a different note gesture model is constructed following the steps that are described in upcoming sections. More contextual variables could be taken into account, like for instance the preceding and following articulations.

2.3. Contour Representation

Bowing gesture parameterer cue contours of recorded notes are modeled in this framework by sequences of a predefined number of units (e.g. lines, curve segments). For the use case presented here, we used constrained cubic Bézier curve segments, similarly as author in [1] used for representing perceptual audio parameter contours. In contrast to the lack of consistent score-performance relation in the organization of the represenation proposed there, we define here a structured representation applying at note-level. We have represented the basic unit in Figure 1.

Even though it responds to a parametric curve defined by the x-y points p_1, p_2, p_3, p_4, the constrains found in equations (2) through (7) allow defining its shape by a vector $b = [d\ v_s\ v_e\ r_1\ r_2]$, where d represents the segment duration, v_s represents the starting y-value, v_e represents the ending y-value, and r_1 and r_2 represent the relative x-values of the attractors p_2 and p_3 respectively. Among the reasons why we choose this as the building block for modeling



$$p_{1y} = p_{2y} = v_s \quad (2)$$

$$p_{3y} = p_{4y} = v_e \quad (3)$$

$$r_1 = \frac{p_{2x} - p_{1x}}{d} \quad (4)$$

$$r_2 = \frac{p_{4x} - p_{3x}}{d} \quad (5)$$

$$0 \leq r_1 \leq 1 \quad (6)$$

$$0 \leq r_2 \leq 1 \quad (7)$$

Figure 1. Constrained Bézier cubic segment used as the basic unit in the representation of bowing gesture parameter contours.

bowing gesture parameter contours, we highlight its robustness (small changes in curve control points lead to small changes in curve contour) and its flexibility (a diverse number of shapes can be modeled by different values of r_1 and r_2, as it is illustrated by gray curves in Figure 1, which correspond to rather extreme values of r_1 and r_2).

Given a time-series bowing parameter motion cue segment $s(t)$ with $t \in [0,d]$, starting value $s(0) = v_s$, and ending value $s(d) = v_e$, optimal attractor relative x-values r_1^* and r_2^* leading to an optimal approximation $\sigma^*(t)$ of the cue segment can be found via constrained optimization (see [1]).

2.4. Grammar Definition

Contours of the bowing gesture parameter cues (bow velocity, bow force, and β ratio) corresponding to the samples present in the corpus of each note class C_i have been been carefully observed in order to foresee an optimal representation scheme by using the constrained Bézier cubic curve segments presented in previous Section. When observing the data and taking the decisions on the segment sequence arrangement, we aimed at keeping the length of the sequences at a minimum while preserving the fidelity of representation.

For each of the note classes C_i, we have defined a grammar entry ρ^i composed by three different tuplas ρ^i_V, ρ^i_F, and ρ^i_β, each one defining the number of segments $N^i_{\{V,F,\beta\}}$ and the slope sequence constrain vector $\Delta s^{i*}_{\{V,F,\beta\}}$ (see (8) through (14)) that is used when performing gesture motion cue segmentation and fitting.

$$\rho^i = \{\rho^i_V, \rho^i_F, \rho^i_\beta\} \quad (8)$$

$$\rho^i_V = \{N^i_V, \Delta s^{i*}_V\} \quad (9)$$

$$\Delta s^{i*}_V = [\delta s^{i*}_{V,1} \cdots \delta s^{i*}_{V,N^i_V-1}] \quad (10)$$

$$\rho^i_F = \{N^i_F, \Delta s^{i*}_F\} \quad (11)$$

$$\Delta s^{i*}_F = [\delta s^{i*}_{F,1} \cdots \delta s^{i*}_{F,N^i_F-1}] \quad (12)$$

$$\rho^i_\beta = \{N^i_\beta, \Delta s^{i*}_\beta\} \quad (13)$$

$$\Delta s^{i*}_\beta = [\delta s^{i*}_{\beta,1} \cdots \delta s^{i*}_{\beta,N^i_\beta-1}] \quad (14)$$

The slope sequence constrain vectors $\Delta s^{i*}_{\{V,F,\beta\}}$ define the expected sequence of slope changes for each of the gesture parameter cues. If each i-th segment is approximated linearly, a contour slope sequence $s = [s_i \cdots s_N]$ is obtained. Each pair of successive slope leads to a parameter δs_i that might take three different values: $\delta s_i \in \{-1,+1,0\}$. The value $\delta s_i = 0$ will be assigned whenever there is no clear expectancy in the relationship between successive slopes s_i and s_{i+1} (due to observations), while on the presence of a particular expectancy, the value for δs_i will be defined by equation 15.

$$\delta s_i = sign(s_{i+1} - s_i) \quad (15)$$

We have sketched in Figure 2 (Left) an hypothetic note gesture sample modeling example in order to illustrate the how we carried out grammar definition by looking at bowing gesture parameter cue contours of a particular note class: the bow velocity contour is modeled by a sequence of 3 Bézier segments with monotonically decreasing slope of their linear approximation. For modeling the force contour, a sequence of three Bézier segments is used, whose linear approximation slope value change sequence shows alternating values. The β ratio is modeled by two segments with increasing slope values. If every slope change were expected, the grammar entry ρ for the model would be defined by $N_V = 3$, $N_F = 3$, $N_\beta = 2$, $\Delta s^*_V = [-1\ -1]$, $\Delta s^*_F = [-1\ +1]$, and $\Delta s^*_\beta = [+1]$. Grammar entries defined during this analysis are available here[1].

2.5. Contour Automatic Segmentation and Fitting

Driven by each previously defined grammar entry for each note class, the acquired bowing gesture parameter cues of each note sample are automatically segmented and approximated by appropriate sequences (see Section 2.4) of Bézier cubic curve segments as the one depicted in Figure 1. The procedure for carrying out contour segmentation and curve fitting is presented next.

In Figure 2 (Right) we have sketched the bow velocity contour and its Bézier approximation of the example given in the previous section. For each one of the i segments, each one presenting a relative duration d_i, the real bow velocity contour q_i is represented by a dashed line, while the approximated Bézier contour σ_i is represented by a solid curve. We set the problem of segmentation and fitting as the optimization task of finding an optimal relative duration vector $d^* = [d^*_1 \cdots d^*_N]$ such that a total cost C is minimized while satisfying that the sum of all components of the relative duration vector must add to the unity. This is expressed in equation (16), where we defined the approximation error ξ_i for the i-th segment as the mean squared error between the real contour q_i and the optimal Bézier approximation σ^*_i

[1] http://www.iua.upf.edu/~emaestre/gestureModels/bowed/violin/grammarV1.pdf

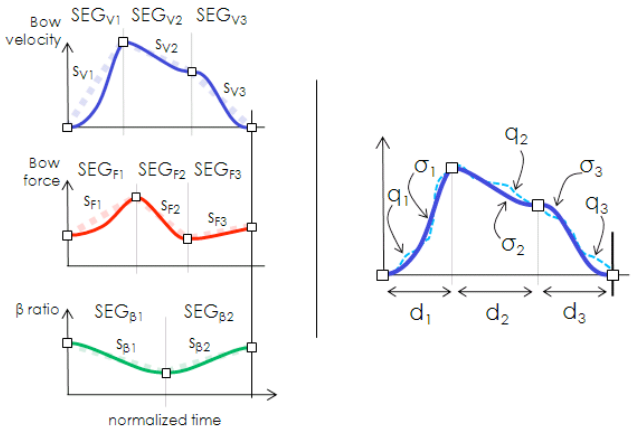


Figure 2. (Left) Schematic illustration of the note gesture model applied to a hypothetic note gesture sample in thick solid lines represent the Bézier approximation for each one of the segments $SEG_1 \cdots SEG_N$, and thick dashed lines represent the linear approximation of each one of the segments, each with its corresponding slope $s_1 \cdots s_N$. Black-border squares represent the junction between adjacent Bézier segments. (Right) Schematic representation of a gesture parameter contour and its Bézier approximation. For each one of the i segments (each one presenting a relative duration d_i), a dashed curve represents the acquired contour q_i, and a solid curve represents the approximated Bézier contour σ_i.

(see Section 2.3), and a weight w_i applied to each ξ_i.

$$d^* = [d_1^* \cdots d_N^*] = \underset{d,\sum_{i=1}^{N} d_i=1}{\operatorname{argmin}} C(d) = \underset{d,\sum_{i=1}^{N} d_i=1}{\operatorname{argmin}} \sum_{i=1}^{N-1} w_i \xi_i + \xi_N \tag{16}$$

The weight w_i applied to each of the first $N-1$ computed ξ_i will depend on the fulfillment of the slope sequence constrains defined by δs^*. Expressed in (17), the weight w_i will be set to a arbitrary value W much bigger than one in case δs_i computed from the slopes of the linear approximations of the i-th and $(i+1)$-th segments does not match the sign of its corresponding δs_i^* (see Section 2.4).

$$w_i = \begin{cases} W >> 1 & \text{if } \frac{\delta s_i}{\delta s_i^*} < 1 \text{ and } \delta s_i^* \neq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{17}$$

The solution for this problem is found by using dynamic programming techniques, in particular based on the so-called *Viterbi* decoding algorithm [14]. As a result, the whole set of note samples corresponding to each of the note classes included in the corpus is analyzed, so that the set of parameters defining the Bézier curve segments that best model each of the bowing gesture parameter contours of each note is attached to each note sample. Some examples of the results on automatic segmentation and fitting are shown in Figure 3, where acquired bowing parameter cues are compared to their corresponding Bézier approximations for *détaché* and *staccato* articulations.

3. MODEL CONSTRUCTION

3.1. Curve Parameter Vector Construction

The curve parameters of each note are represented as a vector p resulting from the concatenation of three curve parameter vectors p_V, p_F, and p_β, corresponding to the bow velocity, bow force, and β ratio contours respectively. The dimensionality of these vectors will depend on the number of segments used for modeling each bowing gesture parameter contour, which is defined by the corresponding grammar entry (see Section 2.4). Each of the three parameter vector contains three different subvectors: a first subvector p^d containing the relative durations d_i/D of each of the segments, a second subvector p^v containing the the inter-segment y-axis values (starting or ending values $v_{s,i}$ or $v_{e,i}$ of each one of the segments), and a third subvector p^r containing the pairs of attractor x-value ratios $r_{1,i}$ and $r_{2,i}$. The organization of the contour parameters is summarized by equations (18) through (30).

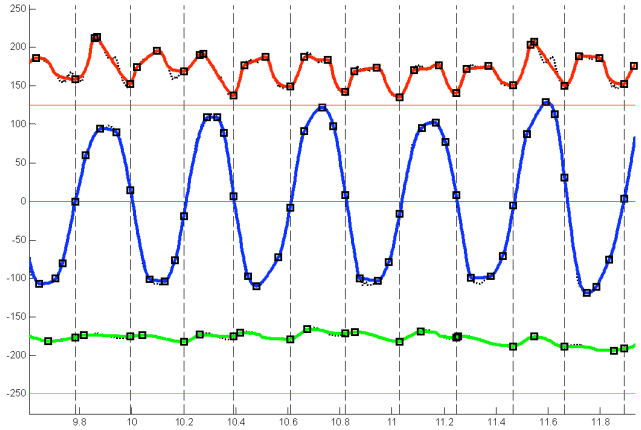
$$p = \{p_V, p_F, p_\beta\} \tag{18}$$
$$p_V = \{p_V^d, p_V^v, p_V^r\} \tag{19}$$
$$p_V^d = [d_{V,1}/D \cdots d_{V,N_V}/D] \tag{20}$$
$$p_V^v = [v_{V,s,1} \cdots v_{V,s,N_V}\ v_{V,e,N_V}] \tag{21}$$
$$p_V^r = [r_{V,1,1}\ r_{V,2,1} \cdots r_{V,1,N_V}\ r_{V,2,N_V}] \tag{22}$$
$$p_F = \{p_F^d, p_F^v, p_F^r\} \tag{23}$$
$$p_F^d = [d_{F,1}/D \cdots d_{F,N_F}/D] \tag{24}$$
$$p_F^v = [v_{F,s,1} \cdots v_{F,s,N_F}\ v_{F,e,N_F}] \tag{25}$$
$$p_F^r = [r_{F,1,1}\ r_{F,2,1} \cdots r_{F,1,N_F}\ r_{F,2,N_F}] \tag{26}$$
$$p_\beta = \{p_\beta^d, p_\beta^v, p_\beta^r\} \tag{27}$$
$$p_\beta^d = [d_{\beta,1}/D \cdots d_{\beta,N_\beta}/D] \tag{28}$$
$$p_\beta^v = [v_{\beta,s,1} \cdots v_{\beta,s,N_\beta}\ v_{\beta,e,N_\beta}] \tag{29}$$
$$p_\beta^r = [r_{\beta,1,1}\ r_{\beta,2,1} \cdots r_{\beta,1,N_\beta}\ r_{\beta,2,N_\beta}] \tag{30}$$

3.2. Performance Context -Based Clustering

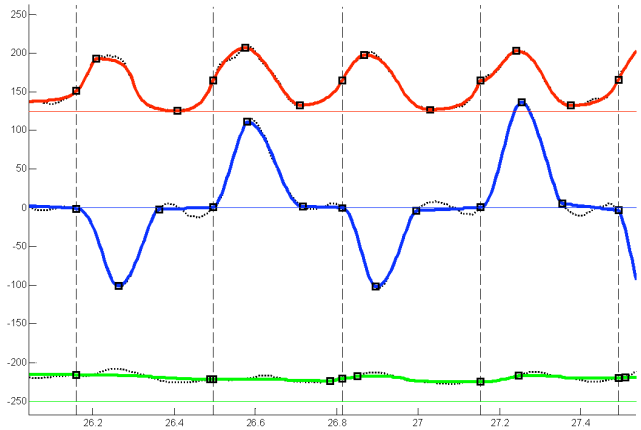
The next step toward the construction of a model for each note class C_i is to divide its notes into different clusters, each one representing notes performed in similar *performance contexts*, which we considered here defined by the note duration and the effective string length (from the bridge to the the finger/nut) after observation of the obtained data curve parameters corresponding notes of different durations or at different pitches. Even though the durations of the curve segments are coded as relative to the note durations, we have observed that their values vary significantly depending on note duration. Extending this to the other curve parameters, we find the note duration to be an important aspect not to be missed when modeling bowing gestures. Likewise, an imporant correlation between the string effective length and the curve parameters corresponding to the β parameter contour

has been observed. Analogously, furher parameters could be considered (e.g. starting/ending bow transversal position).

The procedure that follows applies to any note class C_i. Once each note sample has been annotated with its corresponding contour parameter vector p (see Section 3.1), we attach to each note a context vector $s = [\, D\ L_{st}\,]$ where D is the note duration (seconds), and L_{st} is the effective length of the string. Note clustering is performed in two steps: first the notes are grouped into different duration clusters, and then, further performance context clusters are obtained within each of the duration clusters. Note duration-based clustering is performed as a first step in order to make sure that a sufficient variety of performance contexts are found for each duration cluster.



(a) *Détaché* articulation



(b) *Staccato* articulation

Figure 3. Bowing gesture parameter cue automatic segmentation and fitting results. In each figure, from top to bottom: acquired bow force (expressed in $50N$ units), bow transversal velocity (cm/s), and bow-bridge distance ($25cm$ units) are depicted with think dashed curves laying behind the modeled, contours, represented by solid thick curves. Solid horizontal lines represent the respective zero levels. Junction points between successive Bézier segments are represented with black squares, while vertical dashed lines represent note onsets and offsets.

Step 1. In a first step, N duration clusters of notes are obtained by applying the $k-means$ clustering algorithm [8] to the note samples, based on the first component of the context vector s, i.e. the note duration D.

Step 2. In a second step, M performance context clusters of note samples are obtained by applying again the $k-means$ clustering algorithm to the notes within each of the previously obtained N duration clusters, but this time based on the 2-dimensional context vector s. Ideally, this leads to $N \times M$ performance context clusters $c_{n,m}$ per note class C_i, each one containing a number of p contour parameter vectors and s performance context vectors (we denote these sets of vectors as $p^{n,m}$ and $s^{n,m}$), each pair corresponding to a note sample. Depending on the number of samples found within each performance context cluster, the values of N and M may need to be modified in each case in order to ensure enough population of the groups.

3.3. Model Parameters Estimation

In a first step, an analysis of the correlation between the durations d_V, d_F, and d_β of the Bézier segments modeling respectively bow velocity, bow force and β ratio, and the note duration D is performed for each note class. We include such information in the model in order to be able to adequately adjust the relative duration of the segments when reconstructing contours (see Section 4). The task of this analysis is to find, for each one of the three gesture parameter contours, which segment presents the highest correlation with the note duration D. For doing so, we collect all note samples belonging to the class under analysis, compute the Pearson correlation coefficient $r_{d,D}$ between each of the segment durations and the note duration, and select the segments s_D^V, s_D^F, s_D^β presenting the highest correlation. The computation of the duration correlation vector s_D containing s_D^V, s_D^F, s_D^β is expressed in equations (31) through (35), where K is the total number of samples in the note class, and N_V, N_F, and N_β are the number of Bézier segments used for modeling each one of the three gesture parameter contours.

$$s_D = \{s_D^V, s_D^F, s_D^\beta\} \tag{31}$$

$$s_D^V = \underset{i,i=1\cdots N_V}{\operatorname{argmax}}\ r_{d_{V,i},D} \tag{32}$$

$$s_D^F = \underset{i,i=1\cdots N_F}{\operatorname{argmax}}\ r_{d_{F,i},D} \tag{33}$$

$$s_D^\beta = \underset{i,i=1\cdots N_\beta}{\operatorname{argmax}}\ r_{d_{\beta,i},D} \tag{34}$$

$$r_{d_i,D} = \frac{K\sum_{k=1}^{K} d_{i,k}D_k - \sum_{k=1}^{K} d_{i,k}\sum_{k=1}^{K} D_k}{\sqrt{K\sum_{k=1}^{K} d_{i,k}^2 - (\sum_{k=1}^{K} d_{i,k})^2}\sqrt{K\sum_{k=1}^{K} D_k^2 - (\sum_{k=1}^{K} D_k)^2}} \tag{35}$$

Then, for each of the N duration clusters, a duration distribution d^n, defined by a mean duration $\mu_{n,d}$ and a duration variance $\sigma_{n,d}^2$, is estimated from the duration of the notes

contained in the n-th duration cluster (see equation (36)).

$$d^n = \{\mu_{n,d}, \sigma_{n,d}\} \tag{36}$$

In a third step, assuming that both the curve parameter vectors $p^{n,m}$ and the context vectors $s^{n,m}$ contained in the m-th performance context cluster of the n-th duration cluster also follow a normal distribution, the estimation of $N \times M$ pairs of normal distributions $g^{n,m}$ and $v^{n,m}$, respectively corresponding to the curve parameter vector distributions (obtained from an estimation based on the set of $p^{n,m}$ vectors) and context vector distributions (obtained from an estimation based on the set of $s^{n,m}$ vectors), is carried out. This is expressed in equations (37) and (38), where $\mu^{n,m}$ and $\Sigma^{n,m}$ respectively correspond to the mean and covariance matrix of the curve parameter vector distribution of the m-th performance context cluster within the n-th duration cluster, and $\gamma^{n,m}$ and $\Omega^{n,m}$ respectively correspond to the mean and covariance matrix of the context vector distribution of the m-th performance context cluster within the n-th duration cluster.

$$g^{n,m} = \{\mu^{n,m}, \Sigma^{n,m}\} \tag{37}$$

$$v^{n,m} = \{\gamma^{n,m}, \Omega^{n,m}\} \tag{38}$$

Thus, the set of parameters describing the model for each class will contain:

- Correlation duration vector s_D.
- N duration clusters each one described by a duration distribution d^n defined by a mean duration $\mu_{n,d}$ and a duration variance $\sigma^2_{n,d}$, and containing M performance context clusters. Each of the M performance context clusters is defined by:
  - A performance context distribution $v^{n,m}$ defined by a mean $\gamma^{n,m}$ and a covariance matrix $\Omega^{n,m}$ of its context vectors.
  - A curve parameter distribution $g^{n,m}$ defined by a mean $\mu^{n,m}$ and a covariance matrix $\Sigma^{n,m}$ of its curve parameter vectors.

4. CONTOUR SYNTHESIS

As a demonstration of the potential of the modeling framework, a gesture contour preliminary synthesis application is presented here. Curve parameters of synthetic bowing gesture parameter cues are obtained for the sequence of notes of an annotated input score. First, the note class to which each note belongs is determined following the same principles that drove score annotation-based note classification during gesture data analysis (see Section 2.2). Then, a target context performance vector s^t (see Section 3.2) is determined from the score annotations. Based on the target context vector s^t, a mixed curve parameter normal distribution g^* is obtained from the curve parameter normal distributions presents in the model of the class into consideration, so that a curve parameter vector p (see equation (18)) can be drawn. Then, before using the components of p for rendering the Bézier curve segments (corresponding to the bow velocity, bow force, and β ratio countours), a number of curve parameters of must be adjusted in order to fulfill some constrains mostly related to note duration and note concatenation.

4.1. Model Mixing

We detail next the steps followed for obtaining the mixed curve parameter distribution g^* based on a target performance context vector s^t. The vector s^t is determined by the target note duration D^t (nominal duration in the input score), and the effective string length L^t_{st} (obtained from the scripted string and the pitch of the note).

1. Duration cluster selection. First, the appropriate duration cluster n^* (see Section 3.2) is selected. For doing so, we compute the distance between the target duration D^t and each cluster duration distribution using a normalized Euclidean distance in which the variance $\sigma^2_{n,d}$ of the durations within each duration cluster is taken into account (see equation (39), where $\mu_{n,d}$ is the mean duration of the n-th cluster).

$$n^* = \underset{n}{\operatorname{argmin}} \sqrt{\frac{(D^t - \mu_{n,d})^2}{\sigma^2_{n,d}}} \tag{39}$$

2. Selection of performance context clusters. Within the selected duration cluster n^*, the closest K performance context clusters (see Section 3.2) to the target context vector s^t are selected from the M subclusters n^*. For doing so, we measure the Mahalanobis distance D_M between s^t and each m-th context vector distribution $v^{n^*,m}$ in n^* (see equation (40)), and keep a vector h of length K with the indexes of the performance context clusters in increasing order.

$$D_M(s^t, v^{n^*,m}) = \sqrt{(s^t - \gamma^{n^*,m})^T (\Omega^{n^*,m})^{-1} (s^t - \gamma^{n^*,m})} \tag{40}$$

3. Curve parameter distribution mixing. The mixed curve parameter distribution g^* (from which we will draw the synthesis curve parameter vector p) is obtained as a weighted average of the K *source* curve parameter distributions corresponding to the closest K performance context distributions to the performance target s^t.

$$g^* = \{\mu^*, \Sigma^*\} \tag{41}$$

$$\mu^* = \sum_{i=1}^{K} w_i \, \mu^{n^*,h(i)} \tag{42}$$

$$\Sigma^* = \sum_{i=1}^{K} w_i \, \Sigma^{n^*,h(i)} \tag{43}$$

The mixed curve parameter distribution parameters μ^* and Σ^* (see equations (42) through (43)) respectively correspond

the weighted average of the means and covariance matrices of the *K source* curve parameter distributions. For the weights corresponding to each distribution in the mix, we have used the Mahalanobis distances computed in the previous step (see equation (44)).

$$w_i = \frac{D_M(s^t, v^{n^*,h(i)})}{\sum_{k=1}^{K} D_M(s^t, v^{n^*,h(k)})} \quad (44)$$

4.2. Adjustment of Contour Parameters

After drawing an initial curve parameter vector p from the mixed distribution g^*, its components are checked for the satisfaction of a set of constrains, some dealing with the nature of the curve segment model (for instance, attractor relative durations must be greater than zero), other dealing with the nature of the note class (e.g. articulation, bow/phrase contexts), some other dealing with the relative duration of symbols, or with note concatenation. Due to the nature of the model, some values of the curve parameters in drawn p might not respect such constrains. Here we give details on the adjustment of the components of p involved in the relative durations of the segments, and in note concatenation issues.

Segment relative durations. The relative segment durations d_i for each of the three cues must sum to the unity. In order to perform the adjustments, the duration of segment s_D (which corresponds to the one found presenting the highest correlation with the note duration, see Section 3.3) is modified for making the total sum of relative durations to be the unity (see equation (45), where D corresponds to the note target duration, and N corresponds to the number of segments used for modeling the cue contour).

$$d_{s_D}/D = 1 - \sum_{\substack{i=1 \\ i \neq s_D}}^{N} d_i/D \quad (45)$$

Note concatenation. Possible discontinuities of bowing parameter contours of successive notes are solved by setting the starting value of the first symbol of each of the three segment sequences (bow velocity, bow force and β ratio) to the ending value of the last segment of their corresponding sequence in the previously note that has been already rendered.

4.3. Rendering Results

Preliminary bowing gesture rendering results obtained by means an implementation of the framework presented here are shown in Figure 4. For rendering bowing gestures, we used existing scores in the corpus. By using note onset/offset times of the recorded performances instead of the nominal times, it is possible to visually compare the obtained contours to the acquired ones. Discontinuities in the contour of bow-bridge distance happen around note onset/offsets due to fact that for the transformation that needs to be applied to the rendered the β ratio in order to obtain the bow-bridge distance, we used flat pitch and string values over each note, leading to sudden changes not happening in reality.

5. CONCLUSION

We have presented a framework for the analysis and synthesis of violin bowing gesture parameter contours. Common patterns observed in bow velocity, bor force, and bow-bridge distance cues lead to the definition of a grammar able to dictate an automatic cue contour segmentation and fitting algorithm adapted to several articulations and contexts. Representation parameters allow the reconstruction of bowing gesture parameter contours. A statistical model based on gaussian distributions of contour parameters (suited for gesture parameter contour analysis and synthesis) is used for rendering gesture parameter contours from an input score.

Several extensions to the general methodology presented here remain clear: adding more note articulations, including left-hand gesture analysis, considering further performance context factors (e.g. playing closer to the tip or to the frog), etc. Likewise, approaches for automatic grammar definition could greatly contribute. Apart from considering the application of the modeing framework to other excitation-continuous instruments, a number of violin use-cases are to be studied. Automatic performance annotation might be useful for expressiveness or style analysis. We are currently studying applications to automate input controls for violin sound synthesis. Moreover, aspects of instrument-specific effort-based physical or biological constrains [12] might be considered in this framework.

6. ACKNOWLEDGMENTS

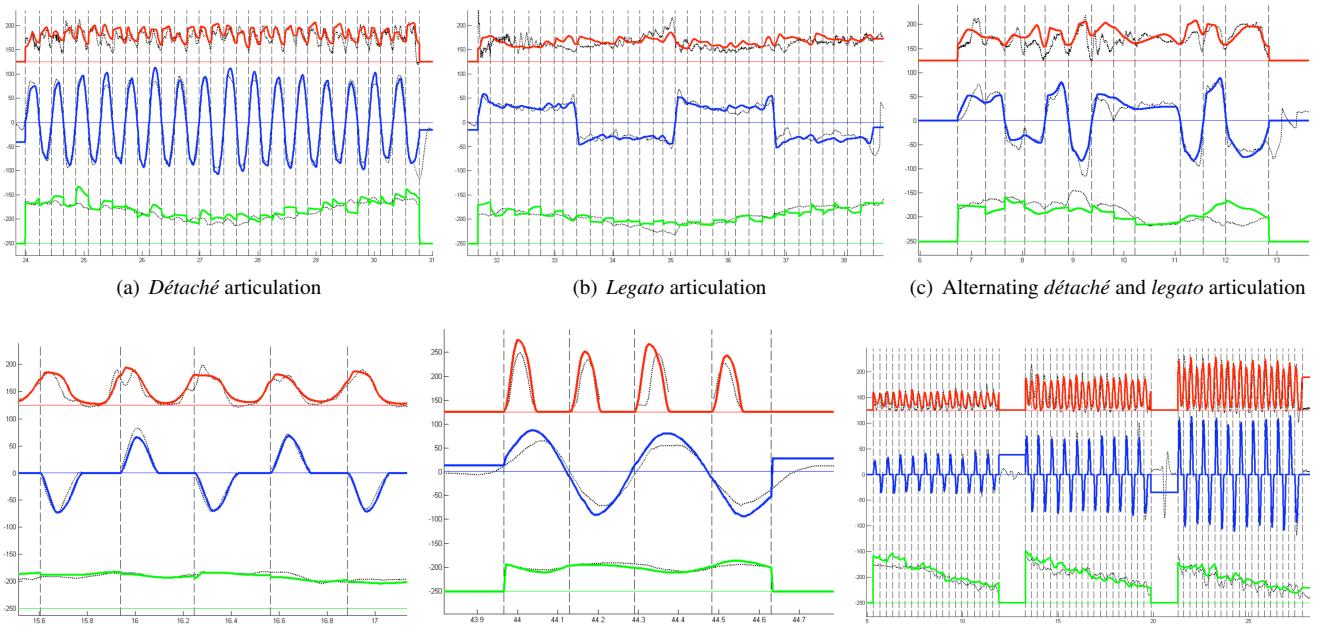
This work was supported by *Yamaha Corporation*, and by the *Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR)* of the *Generalitat de Catalunya*. I would like to thank Jordi Bonada, Merlijn Blaauw, Enric Guaus, and Alfonso Pérez for data acquisition and note segmentation. Likewise, I thank Jonathan Abel, Julius Smith, and Argyris Zymnis for inspiring discussion and helpful advice.

7. REFERENCES

[1] B. Battey, "Bézier spline modeling of pitch-continuous melodic expression and ornamentation," *Computer Music Journal 28:4*, 2004.

[2] C. Cadoz and C. Ramstein, "Capture, representation and composition of the instrumental gesture," in *Proc. of ICMC90*, Glasgow, 1990.

[3] C. Chafe, "Simulating performance on a bowed instrument," *CCRMA Tech. Rep. STAN-M48*, 1988.



(a) *Détaché* articulation (b) *Legato* articulation (c) Alternating *détaché* and *legato* articulation

(d) *Staccato* articulation (e) *Saltato* articulation (f) *Piano*, *mezzoforte*, and *forte* dynamics for *staccato* articulation

Figure 4. Bowing gesture parameter rendering results. From top to bottom: bow force ($50N$ units), bow velocity (cm/s), and bow-bridge distance ($25cm$ units). Horizontal thin lines correspond to zero levels, solid thick curves represent rendered contours, and dashed thin curves represent acquired cues. Vertical dashed lines represent note onsets and offsets.

[4] M. Demoucron, "On the control of virtual violins: Physical modelling and control of bowed string instruments," *PhD. dissertation, Universite Pierre et Marie Curie (Paris 6) and the Royal Institute of Technology (KTH)*, 2008.

[5] D. Escudero, V. Cardenoso, and A. Bonafonte, "Corpus-based extraction of quantitative prosodic parameters of stress groups in spanish," in *Proc. of ICASSP02*, Orlando, 2002.

[6] E. Guaus, J. Bonada, A. Pérez, E. Maestre, and M. Blaauw, "Measuring the bow pressing force in a real violin performance," in *Proc. of ISMA07*, Barcelona, 2007.

[7] D. Jaffe and J. Smith, "Performance expression in commuted waveguide synthesis of bowed strings," in *Proc. of ICMC95*, Alberta, 1995.

[8] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, 1967.

[9] E. Maestre, "Coding instrumental gestures: towards automatic characterization of instrumental gestures in excitation-continuous musical instruments," *DEA Doctoral pre-Thesis work, Universitat Pompeu Fabra*, 2006.

[10] E. Maestre, J. Bonada, M. Blaauw, E. Guaus, and A. Pérez, "Acquisition of violin instrumental gestures using a commercial emf device," in *Proc. of ICMC07*, Copenhagen, 2007.

[11] E. Maestre, J. Bonada, and O. Mayor, "Modeling voice articulation gestures in singing voice performance," *AES 118th Convention*, 2006.

[12] N. Rasamimanana and F. Bevilacqua, "Effort-based analysis of bowing movements: evidence of anticipation effects," *In Press*.

[13] N. Rasamimanana, E. Flety, and F. Bevilacqua, "Gesture analysis of violin bow strokes," *LNCS Vol.3881*, 2006.

[14] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. on Information Theory*, April 1967.

[15] D. Young, "Classification of common violin bowing techniques using gesture data from a playable measurement system," in *Proc. of NIME08*, Genova, 2007.

[16] ——, "A methodology for investigation of bowed string performance through measurement of violin bowing technique," *PhD Dissertation, Massachusetts Institute of Technology*, 2007.