TOWARDS REALISTIC AND NATURAL SYNTHESIS OF MUSICAL PERFORMANCES: PERFORMER, INSTRUMENT AND SOUND MODELING

Alfonso Perez, Rafael Ramirez

Music Technology Group
Universitat Pompeu Fabra, Barcelona, Spain
alfonso.perez@upf.edu

ABSTRACT

Imitation of musical performances by a machine is an ambitious challenge involving several disciplines such as signal processing, musical acoustics or machine learning. The most important techniques are focused on modeling either the instrument (physical models) or the perceived sound (signal models) and advances in the last decades in the field of sound synthesis allow for the recreation of highly realistic sounds. However, these models generally lack an explicit representation of the performer and the consequence is that despite the realistic sounds, synthetic reproductions do not sound natural as a human performance. The role of the performer is usually captured with expressivity models and more recently, on the analysis of instrumental controls acquired during real performances.

In this work we present a framework that combines the modeling of the characteristics of the sound, the instrument as well as the performer in order to render natural performances with realistic sounds automatically from a musical score. The framework is based on a source-filter model, which is driven by performer controls. Performer modeling is based on automatic rendering of perceptual features (i.e. fundamental frequency, energy, timbre and tempo contours); sound source is modeled through functions that map those perceptual features into spectral envelopes, which are translated into the temporal domain through additive synthesis; and the instrument is represented as a multi-directional filter that encodes the sound radiation properties of the instrument body. The case study is the violin, but the framework is general and could be applied to any kind of musical instrument.

1. INTRODUCTION

A musical performance can be regarded as a process in which symbolic information in the form of a musical score is transformed into sound. The main components in this process are the performer and the music instrument. The performer interprets the score and generates a sequence of actions that control the instrument, which in turn produces the sound. It is therefore of paramount importance in order to produce realistic and natural synthetic sounds, to model both the instrument sound as well as the performer. In this work, we mean by realistic, how the synthetic sound resembles that of the instrument, and natural, it is related to the sound evolution in time and the resemblance to a human performance.

Sound synthesis techniques for musical instruments [1, 2] can be divided into two main categories, physical models [3] that are focused on the sound production mechanism of the instrument, and signal models [4, 5, 6, 7] that are focused on the perceived sound. These techniques are used with great success with impulsively excited instruments such as hammered strings [8] or plucked strings [9], however, in the case of continuously excited instruments such as bowed strings or wind instruments, for which the degree of control is much higher, obtaining a natural sounding synthesis is still an open issue and it is a consequence of the lack of an explicit representation of the performer in the models.

The role of the musician in a musical performance is usually approached through computational techniques for modeling expressive performances. So called expressivity models, aim to describe and characterize deviations in a real performance from the musical score. Deviations come in the form of continuous numerical aspects of expressivity, mainly timing, dynamics and pitch. The inclusion of the performer in synthesis models has been the main subject in the recent work by Perez-Carrillo [10] and Maestre [11] with the violin, based on a representation of the performer as continuous bowing controls. Although this technique is able to render much more natural and realistic sounds than previous techniques, the performer representation is specific to bowed strings. Grounded on this previous work [12, 13, 10], we aim to provide a more general framework that can be extended to other types of instruments by representing the performer as features typical in expressivity models, instead of instrumental controls.

A musical instrument can be seen as the association of an exciter and a resonator. The exciter produces the excitation vibration (glottal chords, lips, reed or bow-string interaction) but does not produce sound and the resonator is the sounding structure (body and air cavity) that acts as a sound radiator. The excitation is typically non-linear and resonating bodies are mostly linear, passive (dissipate energy) and harmonically resonating structures. Based on this assumption, we approach the rendering of the sound of the instrument as a source-filter (exciter-resonator) model.

This works proposes a framework for realistic and natural synthesis of musical instruments by combining the modeling of the characteristics of the sound, the instrument as well as the performer. The procedure, represented in Figure 1, is based on a source-filter model, which is driven by performer controls. The first module is responsible for the representation of performer. It involves the automatic rendering of the most important perceptual features that are related to expressivity, namely, energy envelope, fundamental frequency ($f0$), tempo and timbre. These features conform the controls that drive the second module, the sound source synthesis engine. This module is responsible for mapping the timbre feature to harmonic and residual spectral envelopes. These envelopes are then filled with harmonic content corresponding to the pitch and filtered white noise for the residual part and translated to the time domain through additive synthesis. The third module is related to the acoustics of the instrument and consists of a model of sound radiation represented as filters that filter the source signal.

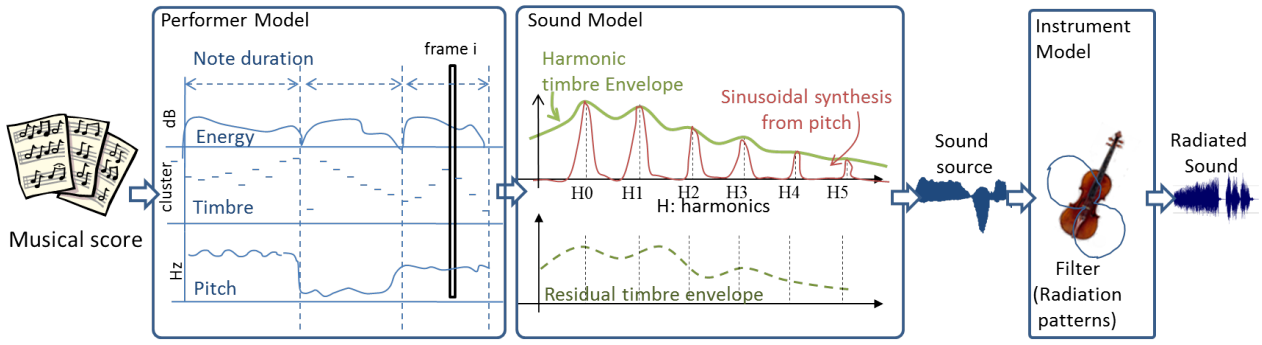

Figure 1: The synthesis framework procedure is based on a source-filter model driven by performer controls. The first module involves the automatic rendering of the most important perceptual features that are related to the performer, namely, energy, fundamental frequency ($f0$), tempo and timbre. These features conform the controls that drive the second module, the sound source synthesis engine, which is responsible for mapping the timbre feature to harmonic and residual spectral envelopes. These envelopes are then filled with harmonic content corresponding to the pitch and filtered white noise for the residual part and translated to the time domain through additive synthesis. The third module is related to the acoustics of the instrument and consists of a model of sound radiation represented as filters that filter the source signal.

2. PERFORMER MODEL

The role of the performer is represented by expressive contours of $f0$, energy, timbre and tempo. Given a symbolic representation of a musical score, the performer model generates contours for these four parameters. This process of transformation between a symbolic domain to a continuous control signal domain is represented in Figure 2. It starts by enhancing (manually or automatically) a musical score with explicit indications about the way of playing (i.e. articulations, slurs, fingering and dynamics) followed by a codification into intra-note segments for which a local envelope is generated. Finally, envelopes are stretched in time and amplitude to match the estimated segment duration 2.5. Two approaches are proposed for the generation of the segment envelopes. The first one consists of a concatenation of intra-note median contours, which is used for the generation of $f0$ (Section 2.2) and the second one is a concatenation of note envelopes retrieved from the database, used for the energy (Section 2.3). Alternative more sophisticated techniques for the rendering of such control contours are found in [11, 14].

2.1. Score codification as intra-note segments

Codification of the musical score as intra-note segments (Figure 2) is of crucial importance as the evolution of performer parameters (specially pitch and energy) largely depend on the type of segment and they show common patterns inside each of the categories. The definition of categories is based on a previous analysis of intra-note segments and bowing parameters in violin playing [15].

2.1.1. Attacks

There are two main types of attacks, *isolated attacks* and *connected attacks*. *Isolated attacks* are found after a silence and *connected attacks* are contiguous to the preceding note and therefore codified as part of a note transition. *Isolated attacks* are, in the case of the violin, divided into *on-string* and *off-string*. A *on-string attack* starts with the bow in contact with the string and it is typically found in *staccato* (and similar strokes such as *martelé*) and can also be found at the first note of a series of *legato* or *detaché*. During *off-string* attacks the bow has an air-borne phase before it gets in contact with the string. Typical bowing strokes with *off-string* attacks are bouncing bow-strokes (e.g. *spiccato*, *saltato*) and can also be found at the first note of a legato or detaché sequence.

2.1.2. Transitions

Note-transitions are segments where a change of note occurs. They are unstable parts where the first note is finishing (release) and the second is beginning (attack). The main bow strokes involved are *detaché* and *legato*. Note transitions in violin playing can be mainly caused through four types of changes in control actions (and their combinations, making a total of 15 types of transitions), a change in the of the pressing finger, a left hand position change, a change of string and a change of bowing direction.

2.1.3. Releases

We only label as releases the ones that are followed by a silence. Releases in contiguous notes are treated as part of a transition. We can identify two types, the first one is the freely-vibrating *release* that lasts until the string stops vibrating by itself, and the second type is the broken *release*, which is caused by stopping the string with the bow.

2.2. Pitch Contour Rendering

Pitch is rendered as a piecewise function of intra-note pitch segments corresponding to the note's attack, sustain, transition and release (Figure 2). A database of violin performances was automatically segmented and labeled according to those categories [15]. For each category, all the matching segments in the database are normalized in length and amplitude and the median pitch contour is used as representative. During the pitch rendering process, representatives of each segment are retrieved, time-stretched to match the note duration and amplitude-stretched to match the difference in the nominal note pitches and finally, the contours are concatenated.

Sustains are treated differently as they are highly variable in duration and may include vibrato. In this case (sustains), the pitch is rendered flat and vibrato is added in sustains longer than the vibrato offset (Figure 3). Vibrato curve is rendered by spectral synthesis of a sinusoid with associated frequency and amplitude envelopes that include fade-ins and fade-outs (Figure 3).

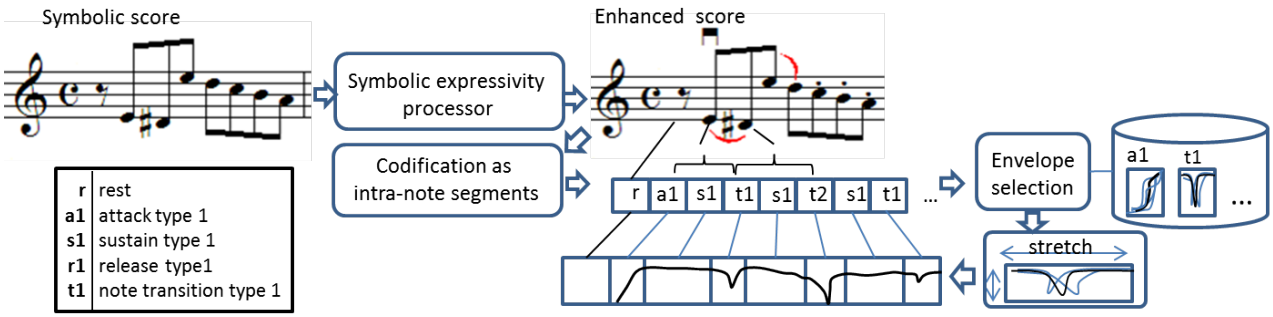

Figure 2: Process of transformation from a symbolic musical score to continuous control signals. It starts by enhancing a musical score with explicit indications about the way of playing (i.e. articulations, slurs, fingering and dynamics) followed by a codification into intra-note segments for which a local envelope is generated. Finally, envelopes are stretched in time and amplitude to match the estimated segment duration.

All parameters of the vibrato can be customized and randomly altered.

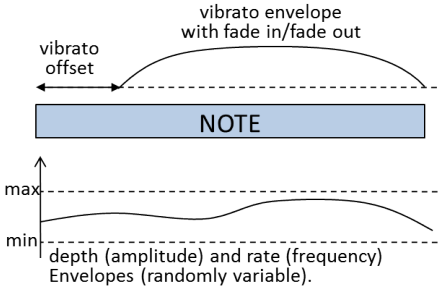

Figure 3: Vibrato is rendered by spectral synthesis of a sinusoid with associated frequency and amplitude envelopes that include fade-ins and fade-outs.

2.3. Energy Contour Rendering

The same procedure applied to the pitch based on concatenation of medians of intra-note segment categories was applied with little success. Instead of that method, sampling was used. For each note in the score, the most similar note in the database is selected according to the note context (duration, dynamics and articulation) and the context of the surrounding notes (previous and next). Finally the selected energy envelope is time-stretch to match the note duration.

2.4. Timbre Trajectory

Timbre in the database is encoded as a number corresponding to a cluster in the timbre space of the database. Details about the timbre space are given in Section 3. The timbre trajectory of a note is then represented as a series of consecutive cluster numbers for each frame in the note. The synthesis of a timbre trajectory is carried out in the same manner as the energy, that is, by retrieving from the database the timbre trajectory of the closest note. Then, for each frame of the timbre trajectory the spectral envelope of the center of the cluster is used. Finally, smoothing between consecutive frames belonging to different clusters is applied.

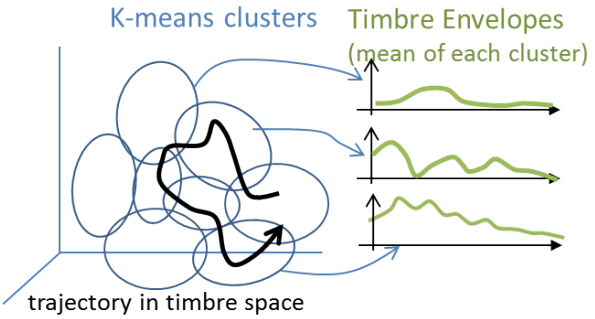

Figure 4: The timbre space is divided into clusters. A timbre trajectory is represented as a series of consecutive cluster numbers. Each sound frame in the database has an associated timbre cluster. To recreate its spectral content, it is mapped into the spectral envelopes corresponding to the enter of the cluster.

2.5. Tempo

Tempo is represented as the note duration rate respect to the nominal note duration in the score. It is computed by training feed-forward neural networks with the following inputs at the current, previous and next note: score tempo, dynamics [pp,p,mf,f,ff], note duration in beats, metrical strength [0..1] and melodic boundary [0..1]. The melodic boundary is computed following the local boundary detection model by Cambouropoulos [16]. The output of the neural network is the rate between the notes' duration in the score and in a performance of the score. Alternative methods are reported in [17, 18].

3. SOUND-SOURCE MODEL

The sound-source in bowed strings corresponds to the vibration of th string. The model is based on spectral analysis of the recordings in the database [10] based on the harmonic plus residual model [19]. Harmonic and residual components are both represented as the energy in 40 overlapping frequency bands with centers following a logarithmic scale. The band's overlapping factor is 50% and the energy of each band is estimated as the average of the corresponding frequency bins, weighted by a triangular function. The selection of the bands is inspired by perceptual models such as the Mel scale. In the case of the harmonics, the amplitude at each bin is determined by a *harmonic envelope*. This envelope is obtained at each frame by interpo-

lating harmonic peaks using a 3rd order spline (see Figure 5).

Harmonic and residual timbre space (40-dimensional) are built with all the computed envelopes in the database (normalized to the energy) and a K-means algorithm is run to automatically partition the space into 32 timbre clusters. Each cluster represents a specific envelope shape. Timbre characteristics of any audio fragment can then be represented as a succession of cluster and each cluster is represented by the 40 spectral envelope coefficients of its center as explained in Subsection 2.4 and Figure 4.

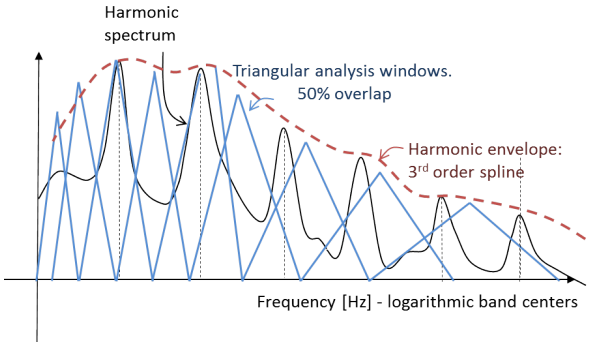

Figure 5: Harmonic and residual components are both represented as the energy in 40 overlapping frequency bands with centers following a logarithmic scale. The energy of each band is estimated as the average of the corresponding frequency bins, weighted by a triangular function. In the case of the harmonics, the amplitude at each bin is determined by a *harmonic envelope*, which is obtained at by interpolating harmonic peaks using a 3rd order spline.

4. INSTRUMENT SOUND RADIATION

The third and last module of the framework is the model of the resonator, the sound radiator structure of the instrument, which determines the characteristic color of the sound of the specific instrument. One of the advantages of modeling this part separately is the possibility of simulating different instruments of the same family as well as the simulation of directional sound effects. In a simplified model of sound production and perception, we can consider this part to be linear. It is usually modeled as a linear filter [20, 21] by measuring its response to a known excitation and computing its body transfer function (BTF) or body impulse response (BIR). There exist many different methods to obtain BTF's depending on the specific instrument.

The presented method here is based on previous work[13] where violin BIR are measured based on non-impulsive deconvolution of signals. Excitation signal was applied by bowing glissandi (sweeping the lowest octave) and the response was measured simultaneously with a bridge-pickup and a microphone in an anechoic chamber. An impulse response is obtained by performing a deconvolution between these non-impulsive bridge-pickup and microphone recordings.

The deconvolution process (Figure 6) starts with the alignment of excitation and response signals. Then, both signals are windowed and expressed in the spectral domain, obtaining two frame streams that are aligned in time. At this point, we apply a frame-by-frame deconvolution. The magnitude is weighted and averaged by the energy content of each spectral bin. Regarding the phase, due to its cyclic behavior, carrying out a classical weighted average would not provide good estimations. As a first attempt, we explored a method based on constructing a histogram of the phase values estimated for each spectral bin, weighted by their corresponding energy. However, the resulting BIR's were not causal, so finally we computed the minimum phase BIR's from the estimated BTF magnitudes by using the cepstrum and converting anti-causal exponentials to causal exponentials [22].

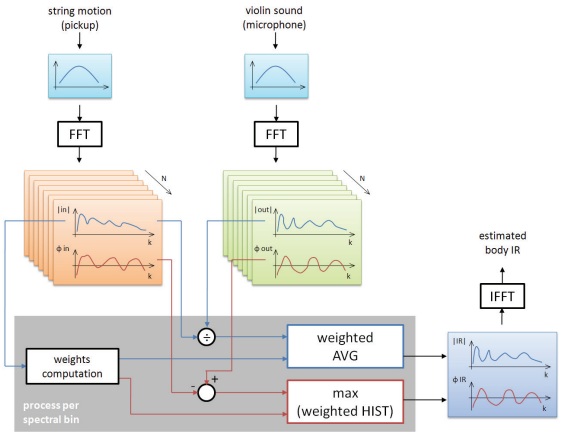

Figure 6: Schematic block diagram of the body impulse response estimation process.

Using multiple microphones, it is possible to simultaneously measure a set of impulse responses in various directions around the violin. The impulse responses can be convolved with a source signal, which improves the sound quality of the synthesizer and allows the simulation of sound directivity effects. Figure 7 shows an example of directivity patterns in the violin.

Although the method is used here for the violin, this is a general procedure that can be applied to other musical instruments as well. The requirement is that there must be a way to measure an excitation signal together with the response signals.

4.1. Directional sound effects

By computing several BTF's around the instrument, we can recreate the sound perceived by a listener at different directions and provide a better listening experience by simulating effects such as stereo or the movement of the player. Stereo can be simulated by using two different BIR's, one for each ear. The movement of the performer is another important effect. Ancillary gestures are constantly changing the position and orientation of the instrument, so the listener is perceiving the sound radiated at continuously changing directions. To simulate this motion we need to estimate the situation of the listener with respect to the sound source and apply a dynamic convolution with the corresponding BIR at each instant. Typical durations of a BIRs restrict the rate at which the listener position can be updated that may result in significant changes between consecutive convolutions. In order to avoid this artifact, an algorithm for dynamic convolution is proposed based on the fragmentation of BIR's into smaller sub-BIR's and a convolution algorithm that processes sub-BIRs concurrently and a sum of all the running convolutions is delivered.

5. CONCLUSIONS

We have presented a framework that combines the modeling of the characteristics of the sound, the instrument as well as the

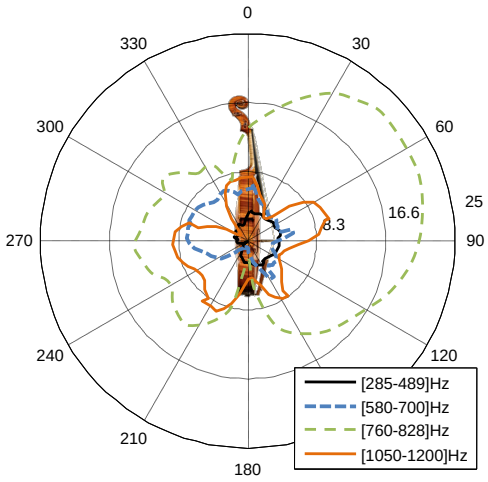

Figure 7: Directivity patterns in the x-z plane. Violin directivity at different frequencies. The patterns show the difference [dB] of sound levels depending on frequency and direction. The patterns were obtained with the performer holding the violin.

performer in order to render natural performances with realistic sounds automatically from a musical score. The framework is based on a source-filter model, which is driven by performer controls. Performer modeling is based on automatic rendering of perceptual features (i.e. fundamental frequency, energy, timbre and tempo contours); sound source is modeled through functions that map those perceptual features into spectral envelopes, which are translated into the temporal domain through additive synthesis; and the instrument is represented as a multidirectional filter that encodes the sound radiation properties of the instrument body. The case study is the violin, but the framework is general and could be applied to any kind of musical instrument.

Acknwoledgements

This work has been partly sponsored by the Spanish TIN project TIMUL (TIN2013-48152-C2-2-R).

6. REFERENCES

[1] Julius Orion Smith, "Viewpoints on the history of digital synthesis," in *Proc. Int. Computer Music Conf.*, Montreal, October 1991, pp. 1–10.

[2] Tero Tolonen, Vesa Vlimki, and Matti Karjalainen, "Evaluation of modern sound synthesis methods," Tech. Rep., 1998.

[3] Julius Orion Smith, "Physical modeling synthesis update," Tech. Rep., 2000.

[4] Jordi Bonada and Xavier Serra, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE signal processing magazine*, vol. 24, pp. 67–80, 2007.

[5] Jordi Bonada, *Voice Processing and Synthesis by Performance Sampling and Spectral Models*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[6] Eric Lindemann, "Musical synthesizer capable of expresive phrasing," *J. Acoust. Soc. Am.*, vol. 117, pp. 2700–2730, 2005.

[7] Bernd Schoner, *Probabilistic Characterization and Synthesis of Complex Driven Systems*, Ph.D. thesis, MIT Media Lab, Cambridge, Massachusetts, USA, 2000.

[8] Vesa Välimäki, Henri Penttinen, Jonte Knif, Mikael Laurson, and Cumhur Erkut, "Sound synthesis of the harpsichord using a computationally efficient physical model," *EURASIP J. on Appl. Signal Processing*, vol. 2004, pp. 934–948, 2004.

[9] Vesa Välimäki and Tero Tolonen, "Development and calibration of a guitar synthesizer," *J. Audio Engineering Soc.*, vol. 46, no. 9, pp. 766–778, September 1998.

[10] A. Pérez-Carrillo, J. Bonada, E. Maestre, E. Guaus, and M. Blaauw, "Performance control driven violin timbre model based on neural networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 1007 –1021, March 2012.

[11] Esteban Maestre, Merlijn Blaauw, Jordi Bonada, Enric Guaus, and Alfonso Pérez, "Statistical modeling of bowing control applied to sound synthesis," *IEEE Transactions on Audio, Speech and Language Processing. Special Issue on Virtual Analog Audio Effects and Musical Instruments*, vol. 18, no. 4, pp. 855–872, 2010.

[12] Alfonso Perez, *Enhancing Spectral Synthesis Techniques with Performance Gestures using the Violin as a Case Study*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2009.

[13] Alfonso Pérez-Carrillo, Jordi Bonada, Jukka Ptynen, and Vesa Välimäki, "Method for measuring violin sound radiation based on bowed glissandi and its application to sound synthesis," *J. Acoust. Soc. Am.*, vol. 130, pp. 1020–1029, 2011.

[14] Akshaya Thippur, Anders Askenfelt, and Hedvig Kjellström, "Probabilistic modeling of bowing gestures for gesture-based violin sound synthesis," in *proceedings of SMAC*, 2013.

[15] Alfonso Pérez-Carrillo, "Characterization of bowing strokes in violin playing in terms of controls and sound: Differences between bouncing and on-string bow strokes.," in *Proceedings of International Conference on Acoustics*, 2013.

[16] Emilios Cambouropoulos, "The local boundary detection model (lbdm) and its application in the study of expressive timing," in *In: Proc. of the International Computer Music Conference, Havana*, 2001.

[17] Rafael Ramirez, Esteban Maestre, and Alfonso Perez, *Guide to Computing for Expressive Music Performance*, chapter Chapter 5: Modeling, Analyzing, Identifying, and Synthesizing Expressive Popular Music Performances, pp. 123–144, Springer-Verlag, London, 2013.

[18] Rafael Ramirez, Alfonso Perez, Stefan Kersten, David Rizo, Placido Roman, and Jose M Inesta, "Modeling violin performances using inductive logic programming," *Intelligent Data Analysis*, vol. 14, pp. 573–585, 2010.

[19] Xavier Serra, *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition*, Ph.D. thesis, Stanford University, 1989.

[20] Angelo Farina, Andreas Langhoff, and Lamberto Tronchin, "Subjective comparisons of 'virtual' violins obtained by convolution," in *Proc. 2nd Int. Conf. on Acoustics and Musical Research*, Ferrara, Italy, May 1995, pp. 36–41.

[21] Lothar Cremer, *Physics of the Violin*, pp. 201–382, The MIT Press, Cambridge, MA, November 1984.

[22] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, chapter 10, pp. 480–531, Prentice Hall Press, Upper Saddle River, NJ, 1975.