An Accurate Timbre Model for Musical Instruments and its Application to Classification

Juan José Burred[1], Axel Röbel[2], and Xavier Rodet[2]

[1] Communication Systems Group, Technical University of Berlin,
burred@nue.tu-berlin.de,
[2] Analysis/Synthesis Team, IRCAM-CNRS – STMS, Paris,
{roebel,rod}@ircam.fr

Abstract. A compact, general and accurate model of the timbral characteristics of musical instruments can be used as a source of a priori knowledge for music content analysis applications such as transcription and instrument classification, as well as for source separation. We develop a timbre model based on the spectral envelope that meets these requirements and relies on additive analysis, Principal Component Analysis and database training. We put special emphasis on the issue of frequency misalignment when training an instrument model with notes of different pitches, and show that a spectral representation involving frequency interpolation results in an improved model. Finally, we show the performance of the developed model when applied to musical instrument classification.

1 Introduction

Our purpose is to develop a model that represents the timbral characteristics of a musical instrument in an accurate, compact and general manner. Such a model can be used as a feature in classification applications, or as a source model in sound separation, polyphonic transcription or realistic sound transformations.

The spectral envelope of a quasi-harmonic sound, which can be accurately described by the amplitude trajectory of the partials extracted by means of additive analysis, is the basic factor defining its timbre. Salient peaks on the spectral envelope (formants or resonances) can either lie at the same frequency, irrespective of the pitch, or be correlated with the fundamental frequency. In this paper, we shall refer to the former as f0-invariant features of the spectral envelope, and to the latter as f0-correlated features.

Model generalization is needed in order to handle unknown, real-world input signals. This requires a framework of database training and a consequent extraction of prototypes for each trained instrument. Compactness does not only result in a more efficient computation but, together with generality, implies that the model has captured the essential characteristics of the source.

Previous research dealing with spectral envelope modeling includes the work by Sandell and Martens [1], who use Principal Component Analysis (PCA) as

a method for data reduction of additive analysis/synthesis parameters. Hourdin, Charbonneau and Moussa [2] apply Multidimensional Scaling (MDS) to obtain a timbral characterization in form of trajectories in *timbre space*. A related procedure by Loureiro, de Paula and Yehia [3] has been recently applied to perform clustering based on timbre similarity. De Poli and Prandoni [4] propose their *sonological models* for timbre characterization, which are based on applying PCA or Self Organizing Maps (SOM) to an estimation of the envelope based on Mel Frequency Cepstral Coefficients (MFCC). These approaches are mainly intended to work with single sounds, and do not propose a statistical training procedure for a generalized application. The issue of taking into account the pitch dependency of timbre within a computational model has only been addressed recently, as in the work by Kitahara, Goto and Okuno [5].

In the present work, we combine techniques aiming at compactness (PCA) and envelope accuracy (additive analysis) with a training framework to improve generality. In particular, we concentrate on the evaluation of the frequency misalignment effects that occur when notes of different pitches are used in the same training database, and propose a representation strategy based on frequency interpolation as an alternative to applying data reduction directly to the partial parameters. We model the obtained features as prototype curves in the reduced-dimension space. Also, we evaluate this method in one of its possible applications: musical instrument classification, and compare its performance with that of using MFCCs as features.

We can divide the modeling approach into a representation and a prototyping stage. In the context of statistical pattern recognition, this corresponds to the traditional division into feature extraction and training stages.

2 Representation stage

2.1 Additive analysis

We have chosen to develop a model based on a previous full additive analysis yielding not only amplitude, but also frequency and phase information of the partials, all of which will be needed for reconstruction and applications involving resynthesis, such as source separation or sound transformations. Additive analysis/synthesis assumes that the original signal $x[n]$ can be approximated as a sum of sinusoids whose amplitudes and frequencies vary in time:

$$x[n] \approx \hat{x}[n] = \sum_{p=1}^{P[n]} A_p[n] \cos \Theta_p[n] \tag{1}$$

Here, $P[n]$ is the number of partials, $A_p[n]$ are their amplitudes and $\Theta_p[n]$ is the total phase, whose derivative is the instantaneous frequency $f_p[n]$. Additive analysis consists of performing a frame-wise approximation of this model, yielding a set of amplitude, frequency and phase information, $\hat{x}_{pl} = (\hat{A}_{pl}, \hat{f}_{pl}, \hat{\theta}_{pl})$, for each partial p and each time frame l. To that end, the successive stages of pitch detection, peak picking and partial tracking are performed. We use a standard procedure, as described in [6].

2.2 Basis decomposition of partial spectra

In its most general form, the basis expansion signal model consists of approximating a signal as a linear combination of basis vectors $\mathbf{b}_i$, which can be viewed as a factorization of the form $\mathbf{X} = \mathbf{BC}$, where $\mathbf{X}$ is the data matrix containing the original signal, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$ is the transformation basis whose columns $\mathbf{b}_i$ are the basis vectors, and $\mathbf{C}$ is the coefficient matrix. Most common transformations of time-domain signals fall into this framework, such as the Discrete Fourier Transform, filter banks, adaptive transforms and sparse decompositions.

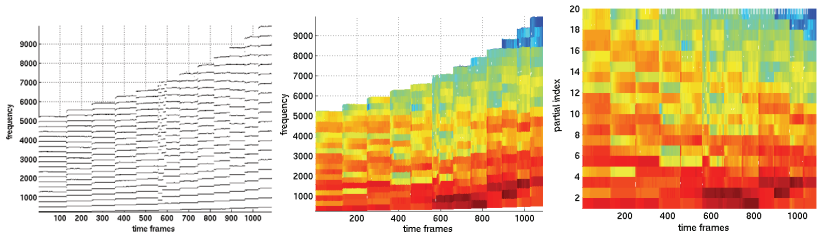
Such an expansion can also be applied to time-frequency (t-f) representations, in which case $\mathbf{X}$ is a matrix of K spectral bands and N time samples (usually $N \gg K$). If the matrix is in temporal orientation (i.e., it is a $N \times K$ matrix $\mathbf{X}(n,k)$), a temporal $N \times N$ basis matrix is obtained. If it is in spectral orientation ($K \times N$ matrix $\mathbf{X}(k,n)$), the result is a spectral basis of size $K \times K$. Having as goal the extraction of spectral features, the latter case is of interest here.

Using adaptive transforms like PCA or Independent Component Analysis (ICA) has proven to yield valuable features for content analysis [7]. In particular, PCA yields an optimally compact representation, in the sense that the first few basis vectors represent most of the information contained in the original representation, while minimizing the reconstruction error, and making it appropriate as a method for dimensionality reduction. ICA can be understood as an extension of PCA that additionally makes the transformed coefficients statistically independent. However, since the minimum reconstruction error is already achieved by PCA, ICA is not needed for our representation purposes. This fact was confirmed by preliminary experiments.

2.3 Dealing with variable supports

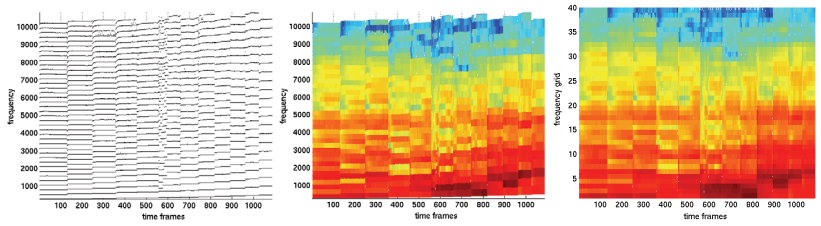
In the context of spectral basis decompositions, training is achieved by concatenating the spectra belonging to the class to be trained (in this case, a musical instrument) into a single input data matrix. As mentioned above, the spectral envelope may change with the pitch, and therefore training one single model with the whole pitch range of a given instrument may result in a poor timbral characterization. However, it can be expected that the changes in envelope shape will be minor for neighboring notes. Training with a moderate range of consecutive semitones will thus contribute to generality, and at the same time will reduce the size of the model.

In the case of additive data, the straightforward way to arrange the amplitudes $\hat{A}_{pl}$ into a spectral data matrix is to fix the number of partials to be extracted and use the partial index p as frequency index, obtaining $\mathbf{X}(p,l)$. We will refer to this as Partial Indexing (PI). However, when concatenating notes of different pitches for the training, their frequency support (defined as the set of frequency positions of each note's partials) will obviously change logarithmically. This has the effect of misaligning the f0-invariant features of the spectral envelope in the data matrix. This is illustrated in Fig. 1, which shows the concatenated notes of one octave of an alto saxophone. In the partial-indexed data



(a) Frequency support (b) Original partial data (c) PCA data matrix

Fig. 1. PCA data matrix with Partial Indexing (1 octave of an alto saxophone).



(a) Frequency support (b) Original partial data (c) PCA data matrix

Fig. 2. PCA data matrix with Envelope Interpolation (1 octave of an alto saxophone).

matrix depicted in Fig. 1c (where color/shading denotes partial amplitudes), diagonal lines descending in frequency for subsequent notes can be observed, which correspond to a misalignment of f0-invariant features. On the contrary, those features that follow the logarithmic evolution of f0 will become aligned.

We evaluate an alternative approach consisting on setting a fixed maximum frequency limit f_{max} before the additive analysis and extracting for each note the required number of partials to reach that frequency. This is the opposite situation as before: now the frequency range represented in each model is always the same, but the number of sinusoids is variable. To obtain a rectangular data matrix, an additional step is introduced in which the extracted partial amplitudes are interpolated in frequency at points defined by a grid uniformly sampling a given frequency range. The spectral matrix is now defined by $\mathbf{X}(g, l)$, where $g = 1, \ldots, G$ is the grid index and l the frame index. We shall refer to this approach as Envelope Interpolation (EI). This strategy does not change frequency alignments (or misalignments), but additionally introduces an interpolation error. In our experiments, we will evaluate two different interpolation methods: linear and cubic interpolation.

Generally, frequency alignment is desirable for our modeling approach because of two reasons. First, prototype spectral shapes will be learnt more effectively if subsequent training frames share more common characteristics. Secondly, the data matrix will be more correlated and thus PCA will be able to obtain a better compression. In this context, the question arises of which one of the alternative preprocessing methods, PI (aligning f0-correlated features) or EI

(aligning f0-invariant features), is more appropriate. In other words, we want to measure which of the two kind of formant-like features are more important for our modeling purposes. In order to answer to that, we performed the experiments outlined in the next section.

2.4 Evaluation of the representation stage

We implemented a cross-validation setup as shown in Fig. 3 to test the validity of the representation stage and to evaluate the influence of the different preprocessing methods introduced: PI, linear EI and cubic EI. The audio samples belonging to the training database are subjected to additive analysis, concatenated and arranged into a spectral data matrix using one of the three methods. PCA is then performed upon the data matrix, yielding a common reduced basis matrix $\mathbf{E}_r$. The data matrix is then projected upon the obtained basis, and thus transformed into the reduced-dimension model space.

The test samples are subjected to the same pre-processing, and afterward projected upon the basis extracted from the training database. The test samples in model space can then be projected back into the t-f domain and, in the case of EI preprocessing, reinterpolated at the original frequency support. Each test sample is individually processed and evaluated, and afterward the results are averaged over all experiment runs.

By measuring objective quantities at different points of the framework, it is possible to evaluate our requirements of compactness (experiment 1), reconstruction accuracy (experiment 2) and generalization (experiment 3). Although each experiment was mainly motivated by its corresponding model aspect, it should be noted that they do not strictly measure them independently from each other.

Here we present the results obtained with three musical instruments belonging to three different families: violin (bowed strings), piano (struck strings or percussion) and bassoon (woodwinds). The used samples are part of the RWC Musical Instrument Sound Database. We trained one octave (C4 to B4) of two

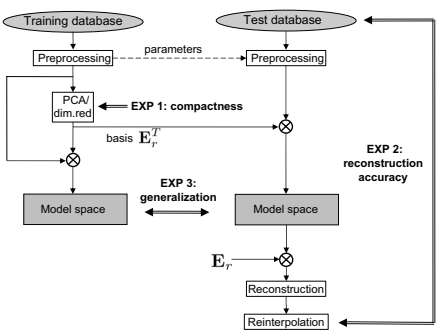


Fig. 3. Cross-validation framework for the evaluation of the representation stage.

exemplars from each instrument type. As test set we used the same octave from a third exemplar from the database. For the PI method, $P = 20$ partials were extracted. For the EI method, f_{max} was set as the frequency of the 20th partial of the highest note present in the database, so that both methods span the same maximum frequency range, and a frequency grid of $G = 40$ points was defined.

Experiment 1: compactness. The first experiment evaluates the ability of PCA to compress the training database by measuring the explained variance:

$$EV(R) = 100\frac{\sum_i^R \lambda_i}{\sum_i^K \lambda_i} \tag{2}$$

where λ_i are the PCA eigenvalues, R is the reduced number of dimensions, and K is the total number of dimensions ($K = 20$ for PI and $K = 40$ for EI). Fig. 4 shows the results. The curves show that EI is capable of achieving a higher compression than PI for low dimensionalities ($R < 14$ for the violin, $R < 5$ for the piano and $R < 10$ for the bassoon). A 95% of variance is achieved with $R = 8$ for the violin, $R = 7$ for the piano and $R = 12$ for the bassoon.

Experiment 2: reconstruction accuracy. To test the amplitude accuracy of the envelopes provided by the model, the dimension-reduced representations were projected back into the t-f domain, and compared with the original sinusoidal part of the signal. To that end, we measure the Relative Spectral Error (RSE)[8]:

$$RSE = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\frac{\sum_{p=1}^{P_l}(A_{pl} - \tilde{A}_{pl})^2}{\sum_{p=1}^{P_l} A_{pl}^2}} \tag{3}$$

where $\tilde{A}_{pl}$ is the reconstructed amplitude at support point (p, l), P_l is the number of partials at frame l and L is the total number of frames.

The results of this experiment are shown in Fig. 5. EI reduces the error in the low-dimensionality range. The curves for PI and EI must always cross because with PI, zero reconstruction error is achieved when all dimensions are present, whereas in the EI case, an interpolation error is always present, even with the full dimensionality. Interestingly, the cross points between both methods occur at around $R = 10$ for all three instruments.

Experiment 3: generalization. Finally, we wish to measure the similarity between the training and test data clouds in model space. If the sets are large enough and representative, a higher similarity between them implies that the model has managed to capture general features of the modeled instrument for different pitches and instrument exemplars.

We avoid probabilistic distances that rely on the assumption of a certain probability distribution (like the Divergence, the Bhattacharyya distance or the Cross Likelihood Ratio), which will yield inaccurate results for data not matching that distribution. Instead, we use average point-to-point distances because, since they are solely based on point topology, they will be more reliable in the

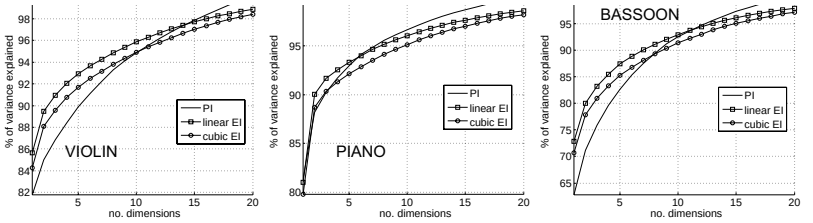


Fig. 4. Results from experiment 1: explained variance.

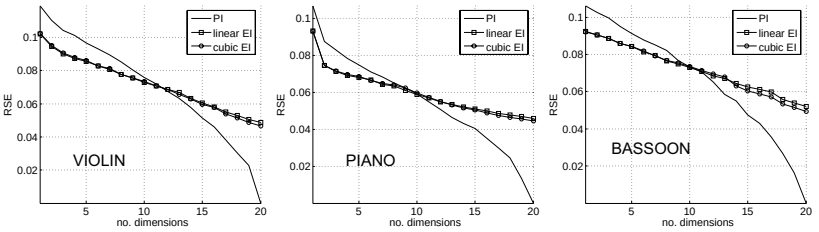


Fig. 5. Results from experiment 2: Relative Spectral Error.

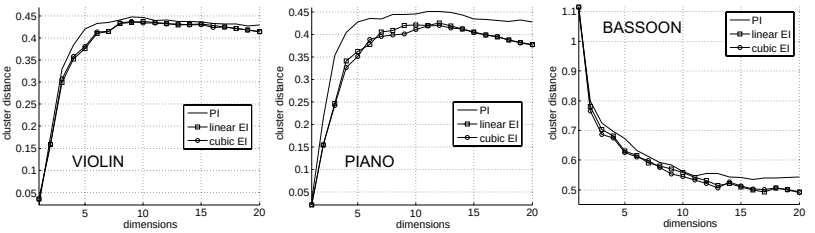


Fig. 6. Results from experiment 3: cluster distance.

general case. In particular, the averaged minimum distance between point clouds, normalized by the number of dimensions, was computed:

$$D_R(\omega_1, \omega_2) = \frac{1}{R}\left\{\frac{1}{n_1}\sum_{i=1}^{n_1}\min_{\mathbf{y}_j \in \omega_2}\{d(\mathbf{y}_i, \mathbf{y}_j)\} + \frac{1}{n_2}\sum_{j=1}^{n_2}\min_{\mathbf{y}_i \in \omega_1}\{d(\mathbf{y}_i, \mathbf{y}_j)\}\right\}. \quad (4)$$

where ω_1 and ω_2 denote the two clusters, n_1 and n_2 are the number of points in each cluster, and $\mathbf{y}_i$ are the transformed coefficients. An important point to note is that we are measuring distances in different spaces, each one defined by a different set of basis, one for each preprocessing method. A distance susceptible to scale changes (such as the Euclidean distance) will yield erroneous comparisons. It is necessary to use a distance that takes into account the variance of the data in each dimension in order to appropriately weight their contributions. These requirements are met by the point-to-point Mahalanobis distance:

$$d_M(\mathbf{y}_0, \mathbf{y}_1) = \sqrt{(\mathbf{y}_0 - \mathbf{y}_1)^T \mathbf{\Sigma}_{\mathbf{Y}}^{-1} (\mathbf{y}_0 - \mathbf{y}_1)} \quad (5)$$

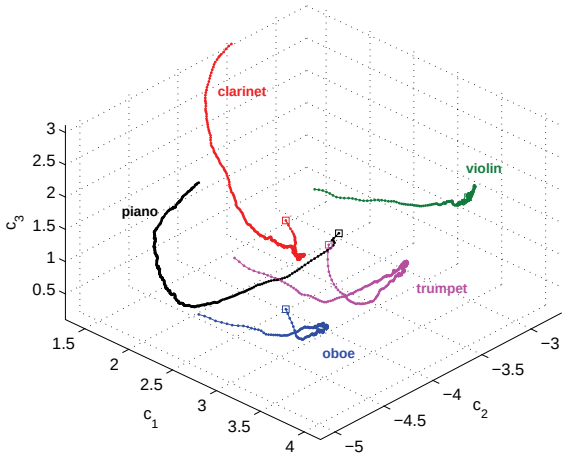


Fig. 7. Prototype curves in the first 3 dimensions of model space corresponding to a 5-class training database of 1098 sound samples, preprocessed using linear envelope interpolation. The starting points are denoted by squares.

where $\mathbf{\Sigma_Y}$ is the covariance matrix of the training coefficients. The results of this third experiment are shown in Fig. 6. In all cases, EI has managed to reduce the distance between training and test sets in comparison to PI.

3 Prototyping stage

In model space, the projected coefficients must be grouped into a set of generic models representing the classes. Common methods from the field of Music Information Retrieval include Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM). Both are based on clustering the transformed coefficients into a set of densities, either static (GMM) or linked by transition probabilities (HMM). The exact variation of the envelope in time is either completely ignored in the former case, or approximated as a sequence of states in the latter. However, we wish to model the time variation of the envelope in a more accurate manner, since it plays an equally important role as the envelope shape when characterizing timbre. Therefore, we choose to always keep the sequence ordering of the coefficients, and to represent them as trajectories rather than as clusters. For each class, all training trajectories are collapsed into a single prototype curve by interpolating all trajectories in time using the underlying time scales in order to obtain the same number of points, and averaging each point across the dimensions. Note lengths do not affect the length or the shape of the training trajectories. Short notes and long notes share the same curve in space as long as they have the same timbral evolution, the former having a smaller density of points on the curve than the latter. Fig. 7 shows an example set of prototype

Representation	Accuracy	STD
PI	74,86 %	± 2.84%
Linear EI	94,86 %	± 2.13%
Cubic EI	94,59 %	± 2.72%
MFCC	60,37 %	± 4.10%

Table 1. Classification results: maximum averaged classification accuracy and standard deviation (STD) using 10-fold cross-validation.

curves corresponding to a training set of 5 classes: piano, clarinet, oboe, violin and trumpet, in the first three dimensions of the model space. This plot corresponds to one fold of the cross-validation experiments performed in the next section.

4 Application to musical instrument classification

In the previous sections we have shown that the modeling is successful in capturing the timbral content of individual instruments. However, for most applications, dissimilarity between different models is desired. Therefore, we wish to evaluate the performance of this modeling approach when performing classification of solo instrumental samples. One possibility to perform classification using the present model is to extract a common basis for the whole training set, compute one prototype curve for each class and measure the distance between an input curve and each prototype curve. We define the distance between two curves as the average Euclidean distance between their points.

For the experiments, we defined a set of 5 classes (piano, clarinet, oboe, violin and trumpet), again from the RWC database, each containing all notes present in the database for a range of two octaves (C4 to B5), in all different dynamics (forte, mezzoforte and piano) and normal playing style. This makes a total of 1098 individual note files, all sampled at 44,1 kHz. For each method and each number of dimensions, the experiments were iterated using 10-fold cross-validation. The best classification results are given in Table 1. With PI, an accuracy of $74,86\%$ was obtained. This was outperformed by around 20% when using the EI apporach, obtaining $94,86\%$ for linear interpolation and $94,59\%$ for cubic interpolation. As in the representation stage experiments, performance does not significantly differ between linear and cubic interpolation.

4.1 Comparison with MFCC

The widely used MFCCs are comparable to our model inasmuch as they aim at a compact description of spectral shape. To compare their performances, we repeated the experiments with exactly the same set of database partitions, substituting the representation stage of Sect. 2 with the computation of MFCCs. The highest achieved classification rate was of 60,37 % (with 13 coefficients), i.e., around 34 % lower than obtained with EI. This shows that, although having

proved an excellent feature for describing overall spectral shape for general audio signals, MFCCs are not appropriate for an accurate spectral envelope model using the prototype curve approach. Also, they use the Discrete Cosine Transform (DCT) as the dimension reduction stage, which unlike PCA is suboptimal in terms of compression.

5 Conclusions and future work

Using the Envelope Interpolation method as spectral representation improves compression efficiency, reduces the reconstruction error, and increases the similarity between test and training sets in principal component space, for a low to moderate dimensionality. In average, all three measures are improved for 10 or less dimensions, which already correspond to 95% of the variance contained in the original envelope data. It also improves prototype-curve-based classification by 20 % in comparison to using plain partial indexing and by 34 % in comparison to using MFCCs as the features.

It follows that the interpolation error introduced by EI is compensated by the gain in correlation in the training data. We can also conclude that f0-invariant features play a more important role in such a PCA-based model, and thus their frequency alignment must be favored.

In a more general classification context, it needs to be verified how the model behaves with a note range larger than 2 octaves. Most probably, several models for each instrument will have to be defined, corresponding to its different registers. In any case, the present results show that the interpolation approach should be the method of choice for this and other, more demanding applications such as transcription or source separation, where the accuracy of the spectral shape plays the most important role.

Possibilities to refine and extend the model include: using more sophisticated methods to compute the prototype curves (like Principal Curves), dividing the curves into the attack, decay, sustain and release phases of the temporal envelope and modeling the frequency information. The procedure outlined here will be integrated as a source model in a source separation framework operating in the frequency domain.

6 Acknowledgments

This research was performed while author J.J.B. was working as a guest researcher at the Analysis/Synthesis Team, IRCAM. The research work leading to this paper has been partially supported by the European Commission under the IST research network of excellence K-SPACE of the 6th Framework programme.

References

1. G.J. Sandell and W.L. Martens. “Perceptual Evaluation of Principal-Component-Based Synthesis of Musical Timbres,” *J. Audio Eng. Soc.*, Vol. 43, No. 12, December 1995.

2. C. Hourdin, G. Charbonneau and T. Moussa. "A Multidimensional Scaling Analysis of Musical Instruments' Time-Varying Spectra," *Computer Music J.*, Vol. 21, No. 2, 1997.
3. M.A. Loureiro, H.B. de Paula and H.C. Yehia, "Timbre Classification of a Single Musical Instrument," *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.
4. G. De Poli and P. Prandoni, "Sonological Models for Timbre Characterization," *J. of New Music Research*, Vol. 26, 1997.
5. T. Kitahara, M. Goto and H.G. Okuno, "Musical Instrument Identification Based on F0-Dependent Multivariate Normal Distribution," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, China, 2003.
6. X. Serra, "Musical Sound Modeling with Sinusoids plus Noise," in C. Roads, S. Pope, A. Piccialli and G. De Poli (Eds.), *Musical Signal Processing*, Swets & Zeitlinger, 1997.
7. M. Casey, "Sound Classification and Similarity Tools," in B.S. Manjunath, P. Salembier and T. Sikora, (Eds.), *Introduction to MPEG-7*, J. Wiley, 2002.
8. A. Horner, "A Simplified Wavetable Matching Method Using Combinatorial Basis Spectra Selection," *J. Audio Eng. Soc.*, Vol. 49, No. 11, 2001.