MUSICAL INSTRUMENT IDENTIFICATION IN CONTINUOUS RECORDINGS

Arie A. Livshin and Xavier Rodet

Analysis/Synthesis Team
Ircam, Paris, France
{livshin|rod}@ircam.fr

ABSTRACT

Recognition of musical instruments in multi-instrumental, polyphonic music is a difficult challenge which is yet far from being solved. Successful instrument recognition techniques in solos (monophonic or polyphonic recordings of single instruments) can help to deal with this task.

We introduce an instrument recognition process in solo recordings of a set of instruments (bassoon, clarinet, flute, guitar, piano, cello and violin), which yields a high recognition rate. A large and very diverse solo database (108 different solos, all by different performers) is used in order to encompass the different sound possibilities of each instrument and evaluate the generalization abilities of the classification process.

First we bring classification results using a very extensive collection of features (62 different feature types), and then use our GDE feature selection algorithm to select a smaller feature set with a relatively short computation time, which allows us to perform instrument recognition in solos in real-time, with only a slight decrease in recognition rate.

We demonstrate that our real-time solo classifier can also be useful for instrument recognition in duet performances, and improved using simple “source reduction”.

1. INTRODUCTION

Most works on instrument recognition have dealt with classification of separate musical tones taken from professional sound databases, e.g. McGill, Studio Online, etc.

Instrument recognition in solo performances (monophonic or polyphonic musical phrases performed by a single instrument) is different and more complicated than dealing with separate note databases, as the time evolution of each sound (attack, decay, sustain, release) is not well defined, the notes are not separated, there are superpositions of concurrent sounds and room echo, different combinations of playing techniques, etc. Marques and Moreno [1] classified 8 fairly different instruments (bagpipes, clarinet, flute, harpsichord, organ, piano, trombone and violin) using one CD per instrument for learning and one for classification. They compared 3 feature types using 2 different classification algorithms, and achieved 70% recognition rate. Brown Houix and McAdams [2] classified 4 wind instruments (flute, sax, oboe and clarinet), compared 4 feature types and reached 82% recognition rate with the best combination of parameters and training material. Martin [3] has classified sets of 6, 7 and 8 instruments, reaching 82.3% (violin, viola, cello, trumpet, clarinet, and flute), 77.9% and 73% recognition rates respectively. He used up to 3 different recordings from each instrument; in each experiment one recording was classified while the rest were learned. The feature set was relatively large and consisted of 31 one-dimensional features. For a comprehensive review of instrument recognition, see [4].

The work on solo recognition is not yet exhausted. Although it seems that there are not many applications which actually require solo recognition, yet as we shall demonstrate at the end of this paper, knowledge of how to deal well with solos can also help in recognition of multi-instrumental music (where several instruments play concurrently). The subject of musical instrument recognition in multi-instrumental music is difficult and is just beginning to get explored (e.g. [5]).

We begin the paper by presenting a process for recognition of a set of instruments (bassoon, clarinet, flute, guitar, piano, cello and violin) which yields a high average recognition rate - 88.13% when classifying 1-second pieces of real recordings.

A large and very diverse solo database is used for learning and evaluating the recognition process. It contains 108 solo performances, all by different musicians, and apparently supplies a good generalization of the different sound possibilities of each instrument in various recording conditions, playing techniques, etc., thus providing a good generalization of the sounds each instrument is capable of producing in different recordings - what we call the “concept instrument”. In order to evaluate the generalization ability of the classifier, the same solos are never used both in the learning and test sets; we have proved that a classification evaluation process in which the training and learning sets both contain samples recorded in very similar conditions is likely to produce misleading results [6].

We use a very large collection of features for solo recognition – 62 different feature types [7] which were developed and used in the Cuidado project. Using our GDE feature selection algorithm, we select a smaller feature set best suited for solo recognition in real-time (of our 7 instruments), with only a small reduction in recognition rate (85.24%) compared to the complete feature set. We present the features of this real-time feature set, which was actually implemented in a real-time solo recognition program.

We end the paper by demonstrating that the same features and techniques we used for real-time solo recognition can also help to perform instrument recognition in duet performances. We use the same solo recognition program and the real-time feature set we used for solos, first to directly perform instrument recognition in duets, and then improve the results using a multiple-f0 detection program by C. Yeh [12] and simple “source reduction”.

2. SOLO DATABASE

Our sound database consists of 108 different ‘real-world’ solo performances (by “solo” we mean that a single instrument is playing, in monophony or polyphony) of 7 instruments: bassoon, clarinet, flute, classical guitar, piano, cello and violin. These performances, which include classical, modern and ethnic music, were gathered from commercial CD’s (containing new or old recordings) and MP3 files played and recorded by professionals and amateurs.

Each solo was performed by a different musician and there are no solos taken from the same concert. During the evaluation process we never use the same solo, neither fully or partly, in both the learning set and the test set. The reason for these limitations is that we need the evaluation process to reflect the system’s ability to generalize – i.e. classify new musical phrases which were not learned, and were recorded in different recording conditions, different instruments and played by different performers than the learning set. We have proved [6] that the evaluation results of a classification system which does learn and classify sounds performed on the same instrument and recorded in the same recording conditions, even if the actual notes are of a different pitch, are much higher than when classifying sounds recorded in different recording conditions. The reason is that such an evaluation process actually shows the system’s ability to learn and then recognize specific characteristics of specific recordings and not its ability to generalize and recognize the “concept instrument”.

2.1. Preprocessing

All solos were downsampled to 11Khz, 16bit. Only the left channel was taken out of stereo recordings[1]. A 2-minute piece was taken from each solo recording and cut into 1-second cuts with a 50% overlap – a total of 240 cuts out of each solo.

3. FEATURE DESCRIPTORS

The computation routines for the features we use in the classification process were written by Geoffroy Peeters as part of the Cuidado project. Full details on all the features, can be found in [7].

The features are computed on each 1-second solo-cut separately. Besides several features[2] which were computed using the whole signal of the 1-second cut, most of the features were computed using a sliding frame of 60 ms with a 66% overlap. For each solo-cut of 1 second, the average and standard deviation of these frames were used by the classifier.

Initially, we used a very large feature collection – 62 different features of the following types [8]:

a) Temporal Features.

Features computed on the signal as a whole (without division into frames), e.g. log attack time, temporal decrease, effective duration.

b) Energy Features.

Features referring to various energy content of the signal, e.g. total energy, harmonic energy, noise part energy.

c) Spectral Features.

Features computed from the Short Time Fourier Transform (STFT) of the signal, e.g. spectral centroid, spectral spread, spectral skewness.

d) Harmonic Features.

Features computed from the Sinusoidal Harmonic modelling of the signal, e.g. fundamental frequency, inharmonicity, odd to even ratio.

e) Perceptual Features.

Features computed using a model of the human hearing process, e.g. mel frequency cepstral coefficients, loudness, sharpness.

Later in the paper we shall use our GDE feature selection algorithm to reduce the number of features in order to perform instrument recognition in real-time.

4. “MINUS-1 SOLO” EVALUATION METHOD

After the features are computed, they are normalized to the range of 0 – 1. For every solo in its turn, its 1-second solo-cuts are removed from the database and classified by the rest of the solos. This process is repeated for all solos, and the average recognition rate for each instrument is reported along with the average recognition rate among all instruments. These results are more informative than the average recognition rate per solo, as the number of solos performed by each instrument might be different.

The classification is done by first performing Linear Discriminant Analysis (LDA) [9];[10] on the learning set, multiplying the test set with the resulting coefficient matrix and then classifying using the K Nearest Neighbours (KNN) algorithm. For the KNN we use the “best” K from a range of 1 - 80 which is estimated using the leave-one-out method on the learning set[3] [11].

5. FEATURE SELECTION

After computing the recognition rate using the full feature set, we use our Gradual Descriptor Elimination (GDE) feature selection method [11] in order to find the most important features. GDE uses LDA repeatedly to find the descriptor which is the least significant and remove it. This process is repeated until no descriptors are left. At each stage of the GDE the system recognition rate is estimated.

In this Section we have set the goal to achieve a smaller feature set which will be quick to compute - allowing us to perform solo recognition in real-time, and will compromise the recognition rate as little as possible, compared with the results obtained by using

[1] It could be argued that it is preferable to use a mix of both channels. Which method is actually better depends on the specific recording settings of the musical pieces.

[2] Some features contain more than a single value, e.g. the MFCC’s; we use the term “features” regardless of their number of values.

[3] The “best K” for our database was estimated as 33 for the full feature set and 39 for the real-time set. Experiments with solo-cuts using an overlap of 75% instead of 50% (resulting in 480 solo-cuts per solo instead of 240), reported a “best K” of 78 for the full feature set and 79 for the “real-time” set.

the complete feature set. By "real-time" we mean here that while the solo is recorded or played the features of each 1-second fraction of the music are computed and classified immediately after it was performed, before the following 1-second has finished playing/recording[4].

We removed the most time-consuming features and used GDE to reduce the feature-data until the number of features went down from 62 to 20. Using these features we have actually implemented a real-time solo phrase recognition program which works on a regular Intel Processor and is written in plain Matlab code (without compilation or integration with machine language boost routines).

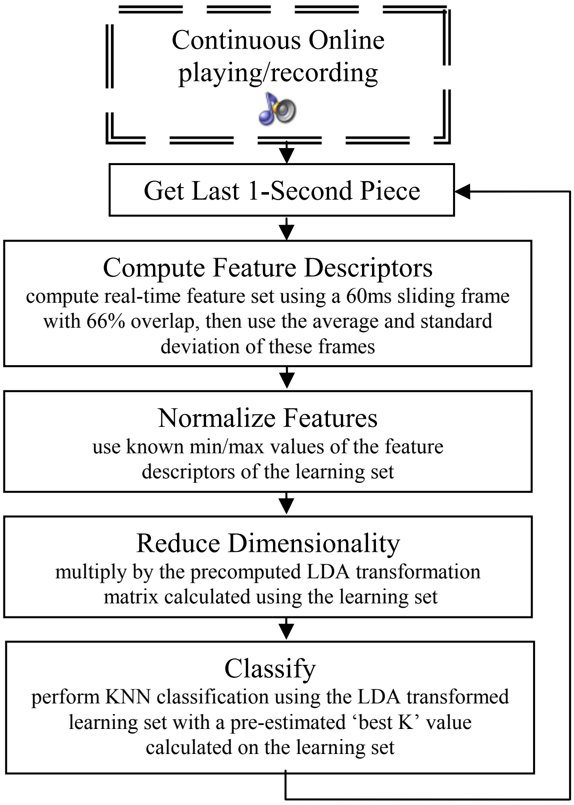
Continuous Online playing/recording
Get Last 1-Second Piece
Compute Feature Descriptors
compute real-time feature set using a 60ms sliding frame with 66% overlap, then use the average and standard deviation of these frames
Normalize Features
use known min/max values of the feature descriptors of the learning set
Reduce Dimensionality
multiply by the precomputed LDA transformation matrix calculated using the learning set
Classify
perform KNN classification using the LDA transformed learning set with a pre-estimated 'best K' value calculated on the learning set

Figure 1: *Real-time solo recognition process*

Naturally, this program uses a precomputed LDA matrix and pre-estimated 'best K' for the KNN classification, as the learning set remains constant and should not depend on the solo input.

We can see in Figure 1 that the classification process uses at each round the last 1-second of the recording, which makes the recognition 'resolution' increase in direct relation to the hardware speed and efficiency of the sub-algorithms being used.

6. RESULTS

We can see in Table 1 that the "Real-Time" average recognition rate is indeed rather close to the "Complete Set". It is interesting to note that while reducing the feature set we have actually improved the recognition rate of the flute; LDA does not always eliminate confusion caused by interfering features.

	"Real-Time" 20 features	**"Complete Set" 62 features**
Bassoon	86.25 %	90.24 %
Clarinet	79.29 %	86.93 %
Flute	83.33 %	80.87 %
Guitar	86.34 %	87.78 %
Piano	91.00 %	93.88 %
Cello	82.18 %	88.72 %
Violin	88.27 %	88.47 %
Average	**85.24 %**	**88.13 %**

Table 1: *Minus-1 Solo recognition results*

6.1. The Real-Time Feature Set

We bring in Table 2 the resulting 20 feature list for real-time classification of solos, sorted by importance, from the most important feature to the least.

1. Perceptual Spectral Slope	2. Perceptual Spectral Centroid
3. Spectral Slope	4. Spectral Spread
5. Spectral Centroid	6. Perceptual Spectral Skewness
7. Perceptual Spectral Spread	8. Perceptual Spectral Kurtosis
9. Spectral Skewness	10. Spectral Kurtosis
11. Spread	12. Perceptual Deviation
13. Perceptual Tristimulus	14. MFCC
15. Loudness	16. Auto-correlation
17. Relative Specific Loudness	18. Sharpness
19. Perceptual Spectral rolloff	20. Spectral rolloff

Table 2: *A sorted list of the most important features for real-time solo classification (of our 7 musical instruments)*

We can see in Table 2 that the 10 most important features are the first 4 Moments and the Spectral Slope, computed in both the perceptual and spectral models. See [7] for a full explanation of each feature.

7. MULTI-INSTRUMENTAL EXAMPLES

In Table 3 we bring some examples for instrument recognition in real performance duets (where 2 instruments are playing concurrently) using our solo-recognition process with the real-time feature set. This Section is not pretending to be an extensive research of multi-instrumental classification, but rather comes to demonstrate that successful solo recognition might actually be useful for instrument recognition in multi-instrumental music.

From each real performance duet, a 1-minute section was selected in which both instruments are playing together.

Each row of Table 3 shows two kinds of results; the upper results are of "Unmodified Recognition" and the bottom of recognition using simple "Source Reduction".

[4] Because the classified 1-second solo pieces can partially overlap, the theoretical upper limit for the recognition resolution is 1 sample.

	Bassoon	Clarinet	Flute	Guitar	Piano	Cello	Violin	Total Correct
Castelnuovo: Sonatina	16.2 % 51.2 %		 2.7 %		83.8 % 44.3 %	 1.8 %		V 100.0 % V 95.5 %
Stockhausen: Tierkreis		50.0 % 50.0 %	50.0 % 50.0 %					V 100.0 % V 100.0 %
Scelsi: Suite		28.9 % 35.0 %	71.1 % 51.8 %				 13.2 %	V 100.0 % V 86.8 %
Carter: Esprit rude		52.5 % 47.3 %	45.0 % 41.8 %			2.5 % 1.7 %	 9.2 %	V 97.5 % V 89.1 %
Kirchner: Triptych		 1.4 %	 1.5 %	2.5 % 7.3 %		60.0 % 41.0 %	37.5 % 48.8 %	V 97.5 % V 89.8 %
Ravel: Sonata		 4.7 %	 3.1 %	2.6 % 1.5 %		59.0 % 44.5 %	38.5 % 46.2 %	V 97.5 % V 90.7 %
Martinu: Duo		 1.8 %	 5.3 %	2.7 % 1.1 %		70.3 % 56.0 %	27.0 % 35.8 %	V 97.3 % V 91.8 %
Pachelbel: Canon in D		 10.6 %	17.6 % 41.0 %	5.0 % 7.3 %		77.5 % 38.9 %	 2.2 %	V 95.1 % V 79.9 %
Procaccini: Trois pieces	44.4 % 49.0 %		5.6 %		50.0 % 49.0 %	 2.0 %		V 94.4 % V 98.0 %
Bach: Cantata BWV		9.6 % 18.2 %	45.2 % 35.5 %	 4.1 %		45.2 % 34.9 %	 7.2 %	V 90.4 % V 70.4 %
Sculptured: Fulfillment		11.1 % 10.2 %	25.0 % 25.7 %	 5.2 %		63.9 % 54.6 %	 4.3 %	V 88.9 % V 80.3 %
Ohana: Flute duo		13.2 % 24.6 %	86.8 % 72.1 %				 3.3 %	V 86.8 % V 72.1 %
Bach: Cantata BWV				13.6 % 5.0 %		79.5 % 62.3 %	6.8 % 32.7 %	~~V~~ 86.3 % V 95.0 %
Pachelbel: Canon in D		 9.2 %	 5.1 %	15.4 % 17.8 %		84.6 % 37.2 %	0.0 % 30.6 %	~~V~~ 84.6 % V 67.8 %
Idrs: Aria	43.2 % 26.9 %	2.7 % 6.9 %	16.2 % 50.6 %	 0.5 %	37.8 % 13.9 %	 1.1 %		~~V~~ 59.4 % V 77.5 %
Feidman: Klezmer		6.7 % 37.1 %		40.0 % 36.3 %	50.7 % 13.5 %	2.7 % 13.1 %		~~V~~ 46.7 % V 73.4 %
Copland: Sonata	 1.3 %	0.0 % 13.7 %	 14.2 %	29.5 % 17.7 %	45.5 % 30.0 %	25.0 % 23.1 %		~~V~~ 45.5 % ~~V~~ 43.7 %
Guiliani: Iglou	 3.9 %	 17.8 %	8.1 % 35.1 %	32.4 % 23.5 %	10.8 %	48.6 % 19.7 %		~~V~~ 40.5 % V 58.6 %

Table 3: *Duet Classification using our real-time solo recognition program.*

7.1. Unmodified Recognition

Each consecutive 1-second piece of the duet, without any modifications, is classified by our real-time solo recognition program.

7.2. Source Reduction

With this proposed technique, we reduce the volume of one of the playing instruments, and thus make it easier for our real-time solo-recognition program to recognize the other instrument.

First we use a multiple-f0 detection program by Chunghsin Yeh [12] (this article is also presented in DAFX'04), to get an estimation of two[5] f0s and their corresponding harmonics for every frame of the duet[6]. The frames are 180 ms each with a 75% overlap.

Next, consecutive frames with the same pitch (we use the estimated f0s, quantized to half-tones), are grouped together into "chunks" of at least ½ second in length[7]. Thus, each chunk of frames contains one note (at least) which is sustained throughout the chunk and presumably performed by a single instrument, and some other notes playing along with it.

Next, each chunk is used twice; in the first case we keep the harmonics of the sustained note and filter out everything else out of the chunk. In the second case, we filter out the sustained note harmonics but keep everything else - we call this the "anti-chunk". This filtering process is performed in the frequency domain using a phase vocoder and the same frame sizes which were used in the

[5] In cases of an instrument accompanied by a highly polyphonic instrument – guitar or piano, the "duets" might have many more concurrent notes than just 2. By using f0 detection of just 2 notes, we rely on the accompanied instrument to have a relatively high volume compared to the accompanying one [5], and thus be recognized as one of the 2 f0s.

[6] The algorithms for the f0 estimation and their evaluation are out of the scope of this article.

[7] Not all the musical piece is covered by chunks, as there are some sections where both instruments play notes that are shorter than the chosen ½ second.

multiple-f0 estimation. Overlapping harmonics of the two notes are not filtered out[8].

The filtered chunks and the anti-chunks are classified by our real-time solo-recognition program.

Note about the "scoring": the Source Reduction results in table 3 (the bottom part of every row) are the percentage of time in which a specific instrument is recognized out of the total time of all the instruments. These are "net results" - in cases where some filtered chunks (or anti-chunks) overlap in time and are classified as the same instrument, we count the time of the overlapping part only once. For example, if two 1-second chunks overlap over ½ second, and are both classified as violin, it will count only as 1½ seconds for the violin and not 2 seconds. Each result in the 'Total Correct' column is the percentage of time in which the correct instruments were recognized out of the total time of all the instruments recognized in that duet.

7.3. Duet Recognition Results

The first column in Table 3 contains the partial name of the musical piece. Columns 2 to 8 contain the percentage of the total classification time which was classified as the corresponding instrument. The black cells indicate correct classifications – recognition of the instruments which actually played in the corresponding cuts, while the white cells indicate misclassified cuts. The last column is the total percentage of time in which correct instruments were recognized.

'V' beside the score in the 'Total Correct' column, indicates that the 2 most popular instrument recognitions are actually the 2 instruments playing in the duet. ~~V~~ means that only one of the 2 most popular recognitions is a correct instrument. This statistic is important; if the maximum number of instruments in a musical piece is known and we can rely on our classifier to correctly recognize them (e.g. the two correct instruments in a duet), then we can afterward limit the learning set to just these two instruments and perform the recognition process again, this time getting a much more precise segmentation of the musical piece into the correct playing instruments.

As already mentioned, each row of table 3 shows at the top the results of Unmodified Recognition and at the bottom the results of recognition using Source Reduction. We immediately can see that there is a considerable number of duet examples where Unmodified Recognition produced correct classifications, although, as we know, this classifier is very naïve and just attempts to classify unmodified duet pieces using a solo classifier.

Looking at the 'V' signs in the 'Total Correct' column, we see that the Source Reduction performed better than the Unmodified Recognition in finding which two instruments play in each duet, and except in one case, always correctly recognized them. On the other hand, just looking at the numbers in the 'Total Correct' column create the impression that in many cases the Unmodified Recognition outperforms the Source Reduction. Looking more carefully on these results along with the recognition rates for each instrument, we can conclude that in duets there is usually one instrument which is more dominant and relatively easier to recognize (e.g. with a higher volume). We see that the Source Reduction recognizes better the weaker instrument than the Unmodified Recognition (which has 'no choice' really, as it classifies each duet-cut only once), and that the recognition results of the Source Reduction for both instruments are more even. For example, if we look at Pachelbel's Canon in D for Cello and Violin, we see that the total Unmodified Recognition score is 84.6% while the Source Reduction score is only 67.8%. However, with the Unmodified Recognition, the violin was not recognized at all. The Source Reduction, on the other hand, recognized the Cello with 37.2% and the Violin with 30.6%, and although the total score is lower than the Unmodified Recognition due to the fact that the source reduction process is not perfect and results in some misclassifications, still the correct instruments are the most popular ones – cello and violin.

We have demonstrated in this Section that a good solo recognizer can be useful for instrument recognition in multi-instrumental music.

8. SUMMARY

We presented a process for continuous recognition of musical instruments in solo recordings which yields a high recognition rate. Our results are based on evaluation with a large and very diverse solo database which allowed us a wide generalization of the classification and evaluation processes using diverse sound possibilities of each instrument, recording conditions and playing techniques.

We used our GDE feature selection algorithm with a big feature set and considerably reduced the number of features, down to a feature set which allowed us to perform real-time instrument recognition in solo performances. This smaller feature set delivers a recognition rate which is close to that of the complete feature set.

Lastly, we have shown that our recognition process and real-time feature set can also be useful for instrument recognition in duet music. This exemplifies our initial claim that learning to achieve high recognition rates in solos could also be useful for instrument recognition in multi-instrumental performances.

9. FUTURE WORK

We shall continue improving the solo recognition process in parallel to working on recognition in multi-instrumental music; we have shown that the first can help to achieve the second.

We will study the reasons why specific instrument combinations produce high recognition errors and how to better differentiate between these instruments.

A confidence level estimation could be added to the solo classifier. Instead of just reporting one instrument, it could give an estimated weight to each instrument, so each 'solo' classification will produce several recognition candidates.

New features will be developed and used in the feature selection process; some of them especially designed with multi-instrumental recognition in mind.

[8] That is why we prefer the term "source reduction" instead of "source separation". We do not attempt to perform full separation of the sources, but just to "reduce" one of them as much as possible without altering the second one.

The multiple-f0 estimation and the source reduction algorithms still need to be improved, especially in order to provide an accurate segmentation into musical instruments of highly multi-instrumental music.

10. ACKNOWLEDGMENTS

Thanks to Geoffroy Peeters for using his feature computation routines and sharing his knowledge and experience.
Thanks to Chunghsin Yeh for using his multiple-f0 detection program.
Thanks to Emmanuel Vincent for sharing his solo database.

11. REFERENCES

[1] J. Marques and P. J. Moreno, “A study of musical instrument classification using Gaussian mixture models and support vector machines,” *Cambridge Research Laboratory Technical Report Series,* CRL/4, 1999.

[2] J. C. Brown, O. Houix and S. McAdams, “Feature dependence in the automatic identification of musical woodwind instruments,” *Journal of the Acoustical Society of America*, vol. 109, no. 3, pp. 1064–1072, 2001.

[3] K. Martin, *Sound-source recognition: A theory and computational model,* PhD Thesis, MIT, 1999.

[4] P. Herrera, G. Peeters and S. Dubnov, “Automatic Classification of Musical Sounds,” *Journal of New Music Research,* vol. 32, no. 1, pp. 3–21, 2003.

[5] J. Eggink and G. J. Brown, “Instrument recognition in accompanied sonatas and concertos,” to appear in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP’04),* 2004.

[6] A. Livshin and X. Rodet, “The Importance of Cross Database Evaluation in Musical Instrument Sound Classification,” in *Proc. Int. Symp. on Music Information Retrieval (ISMIR’03),* 2003.

[7] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” 2003. URL: `http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf`

[8] G. Peeters and X. Rodet, “Automatically selecting signal descriptors for Sound Classification,” in *Proc. Int. Computer Music Conf. (ICMC’02),* 2002.

[9] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition.* New York, NY: Wiley Interscience, 1992.

[10] K. Martin and Y. Kim, “Musical instrument identification: a pattern-recognition approach,” in *Proc. 136th Meeting of the Acoustical Society of America,* 1998.

[11] A. Livshin, G. Peeters and X. Rodet, “Studies and Improvements in Automatic Classification of Musical Sound Samples,” in *Proc. Int. Computer Music Conf. (ICMC’03),* 2003.

[12] C. Yeh, A. Röbel, “A new score function for joint evaluation of multiple F0 hypotheses,” in *Proc. of the Int. Conf. on Digital Audio Effects (DAFX’04),* 2004.