Purging Musical Instrument Sample Databases Using Automatic Musical Instrument Recognition Methods

Arie Livshin, Xavier Rodet

Purging Musical Instrument Sample Databases Using Automatic Musical Instrument Recognition Methods

Arie Livshin and Xavier Rodet, *Member, IEEE*

***Abstract*—Compilation of musical instrument sample databases requires careful elimination of badly recorded samples and validation of sample classification into correct categories. This paper introduces algorithms for automatic removal of bad instrument samples using Automatic Musical Instrument Recognition and Outlier Detection techniques. Best evaluation results on a methodically contaminated sound database are achieved using the introduced MCIQR method, which removes 70.1% "bad" samples with 0.9% false-alarm rate and 90.4% with 8.8% false-alarm rate.**

***Index Terms*—Instrument recognition, multimedia databases, music, music information retrieval, pattern classification.**

I. Introduction

A Musical Instrument Sample Database of Isolated Notes (MISDIN) is a collection of sound samples of one or more musical instruments where each sample contains a recording of a single note played by one instrument. MISDINs are commonly used by electronic musical instruments, such as synthesizers and samplers, to reproduce sounds of other instruments. MISDINs are also utilized by the majority of music information retrieval (MIR) algorithms, including pitch estimation [1], music representation [2] and others, as evaluation data for experiments and for modeling sounds of different musical instruments. Having badly recorded or incorrectly labeled samples in a MISDIN may therefore cause incorrect sounds to be played by an electronic instrument, or produce erroneous computation results in scientific MIR experiments.

In the pattern recognition field, erroneous samples in a database are usually called "outliers." For a thorough summary on outliers see [3].

Manual removal of outliers from a MISDIN by listening to each individual sample is a hard and time-consuming task. In this paper, we introduce and evaluate techniques for automatic removal of outliers from MISDINs using automatic musical instrument recognition (AMIR) and outlier detection methods.

In several papers, including [4]–[7], we have used a variation of the classical pattern recognition approach for AMIR first employed in [8]. A large collection of feature descriptors was computed on the sound samples of each musical instrument in a Learning Set in order to capture the different characteristics of each instrument class. The feature descriptors were then weighted and computed on unlabeled samples in a Test Set. Next, the Test Set was classified using a classifier trained on the Learning Set. The paper shows that the same descriptors we used in [4]–[7] and similar techniques can be used successfully in order to automatically detect outliers in a MISDIN.

This paper significantly extends ideas we presented briefly in [4], by introducing new and improved algorithms, methodical evaluation and thorough discussions and conclusions[1]

Manuscript received September 07, 2008; revised February 25, 2009. Current version published June 10, 2009. This work was supported in part by the Chateaubriand scholarship of the French Ministry of Foreign Affairs and by the "ACI Masse de Données" project "Music Discover." The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Susanto Rahardja.

The authors are with the Institut de Recherche et Coordination Acoustique/Musique (IRCAM), 75004 Paris, France. (e-mail : arie.livshin@gmail.com; xavier.rodet@ircam.fr).

Digital Object Identifier 10.1109/TASL.2009.2018439

II. Feature Descriptors

In order to encapsulate characteristic attributes of sound signals of different instruments, an extensive feature set consisting of 45 different feature types is computed on each sound sample. Some of these feature computations produce a vector of values and some are computed using a selection of different parameters. For example, Spectral Kurtosis feature variations include Kurtosis computed on the linear spectrum, the log-spectrum, the harmonics envelope, etc. A total of 162 feature descriptor values are computed per sample.

Most of the feature descriptors are "frame based"—the feature is computed on each frame of a short-time Fourier transform (STFT) of the signal [9], using a sliding window of 60 ms with a 66% overlap, then the average over all these frames is used as a feature descriptor. The feature descriptors are normalized to the range of $[0, 1]$.

The feature computation routines were written by G. Peeters of IRCAM. A full description of each feature can be found in [10].

Feature List

A. Temporal Features

Features computed on the whole signal (without division into frames): Log Attack Time, Temporal Decrease, Temporal Centroid, Effective Duration, Signal Auto-Correlation, Zero-Crossing Rate.

B. Energy Features

Features referring to the energy content of the signal: Total Energy Modulation, Harmonic Energy, Noise-Part Energy.

[1]More precisely, the current paper introduces the Self-Consistency Outlier removal algorithm (SCO), the Self-Consistency-Rate (SCR), and the IQR and MCIQR algorithms which are non-iterative versions of the Interquantile Range (IQR) and Modified IQR (MQR) algorithms first presented in [4]. Unlike in [4], where we have just briefly demonstrated our AMIR outlier removal algorithms of that time, the algorithms here are systematically evaluated using a methodically contaminated MISDIN.

C. Spectral Features

Features computed from the STFT of the signal: Spectral Centroid, Spectral Spread, Spectral Skewness, Spectral Kurtosis, Spectral Slope, Spectral Decrease, Spectral Rolloff, Spectral Variation, Spectral Flatness, Spectral Crest.

D. Harmonic Features

Features computed from the Sinusoidal Harmonic modeling of the signal: Fundamental Frequency (f0), Noisiness, Inharmonicity, Harmonic Spectral Deviation, Odd to Even Harmonic Ratio, Harmonic Tristimulus, Harmonic Centroid, Harmonic Spread, Harmonic Skewness, Harmonic Kurtosis, Harmonic Slope, Harmonic Decrease, Harmonic Rolloff, Harmonic Variation.

E. Perceptual Features

Features computed using a model of the human hearing process (see [11] for the Mel scale and [12] for the Bark scale): Mel Frequency Cepstral Coefficients (MFCCs), Delta MFCC, Delta-Delta MFCC, Loudness, Relative Specific Loudness, Fluctuation Length, Mean Fluctuation Length, Roughness, Sharpness, Spread.

III. Self Consistency Rate

The main reasons for outliers in MISDINs are as follows.

1) Attribute Noise: Badly sampled sounds or garbled data.

2) Class Noise: Samples mislabeled as belonging to the wrong instrument.

3) Sparse Region Samples: Samples correctly recorded and labeled but still differing very much from other samples in their instrument class.

As already noted, when performing AMIR using a classical pattern-recognition approach, one MISDIN or more are used by the classification algorithm as a Learning Set, i.e., for capturing typical sound characteristics of the different instruments. The Learning Set is then used for classifying the Test Set which contains new, unlabeled sounds. The presence of outliers in the Learning Set can therefore lead to inflated error rates and substantial distortions of parameter and statistic estimates when using either parametric or nonparametric tests [13].

We propose to use these inflated error-rates for measuring the effectiveness of outlier removal methods by introducing a Self-Consistency Rate (SCR) for a MISDIN, which is computed before and after removing the outliers.

The SCR computation uses Self-Classification [5], formerly a common evaluation method for AMIR. In Self-Classification, a MISDIN is split into a Learning Set, containing a certain percentage of randomly selected samples from each instrument class, and a Test Set which contains the remaining samples. The Learning Set is then used to classify the Test Set. In order to eliminate the dependency of the resulting recognition rate on a specific random split into Learning and Test sets, this process is repeated a number of times, and the average, and optionally the standard deviation and confidence intervals of the recognition rates, are reported. While it was demonstrated that Self-Classification is not appropriate for generalized evaluation of AMIR [5] as one MISDIN does not typically model a general, "concept" musical instrument, Self-Classification is very suitable for computing the Self-Consistency Rate of a specific MISDIN.

For computing the SCR, 50 Self-Classification rounds are performed with a 66%/34% split into Learning and Test sets. In each classification round, after selecting the Learning and Test sets, a linear discriminant analysis (LDA) [14] transformation matrix is computed using the Learning Set and then used to transform both the Learning and Test sets. The K-Nearest-Neighbors (KNN) algorithm is used for classification. The best K value for the whole process is selected from a range of $[1, 80]$ after completing the 50 rounds. LDA + KNN has been demonstrated in [6] and [7] as an effective classification algorithm for performing AMIR.

The reported MISDIN SCR is the average recognition rate of these 50 Self-Classification rounds. See Fig. 1 for a flowchart of the SCR computation process.

The SCR measures the success with which samples of an instrument in the MISDIN can be used for recognizing each other, and thus, how consistent are the representations of the different instruments in the MISDIN.

While in this paper we evaluate MISDIN purging methods knowing *a priori* which samples constitute the outliers, SCR could be used as well in real-world situations to estimate whether am MISDIN is likely to contain unknown outlying samples.

IV. MISDIN Purging Methods

A. Interquantile Range

Interquantile Range (IQR) is a commonly used outlier detection approach.

Algorithm: Given a sample database $\boldsymbol{S}$ with $\{\boldsymbol{D}\}$ feature descriptors computed on each sample:

- For every descriptor $\boldsymbol{d} \in \boldsymbol{D}$:
  - — let $\boldsymbol{P_1}$ be the $\boldsymbol{X}$th percentile of the values of $\boldsymbol{d}$ in $\boldsymbol{S}$;
  - — let $\boldsymbol{P_2}$ be the $\boldsymbol{Y}$th percentile of the values of $\boldsymbol{d}$ in $\boldsymbol{S}$ where $\boldsymbol{X} > \boldsymbol{Y}$ (for example: $\boldsymbol{X} = 90\%, \boldsymbol{Y} = 10\%$);
  - — remove all samples where the value of $\boldsymbol{d}$ "falls out" of the defined range, that is:

$$\boldsymbol{d} > \boldsymbol{P_1} + (\boldsymbol{P_1} - \boldsymbol{P_2}) * \boldsymbol{C} \quad or \quad \boldsymbol{d} < \boldsymbol{P_2} - (\boldsymbol{P_1} - \boldsymbol{P_2}) * \boldsymbol{C}$$

where $\boldsymbol{C}$ is a scalar selected for interval scaling (for example: $\boldsymbol{C} = 1.5$).

In this paper, $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{C}$ are empirically selected according to the permitted false-alarm ratios.

Instead of using percentiles, a common modification of IQR is to calculate the mean and standard deviation (STD) of every descriptor, and then remove the samples where a descriptor is distanced from its mean by more than several times its STD [15].

Note that IQR is not a supervised method—it does not utilize class information. While this has the advantage that IQR could be used even with nonlabeled sound collections, when present with a sound collection which is labeled, IQR has the disadvantage that it ignores available information about the different spread of descriptor values in each class.

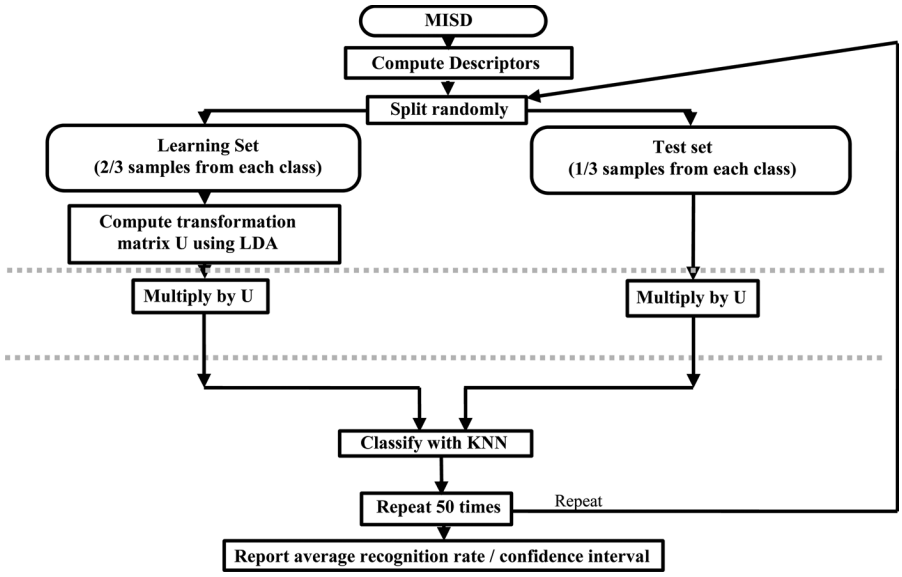


Fig. 1. Computing self-consistency rate (SCR) using self-classification and LDA + KNN.

IQR assumes "weak noisiness," in the sense that samples are considered outliers even if only one of their descriptors has outlying values.

B. Multiclass IQR

The introduced Multiclass IQR (MCIQR) method for removing outliers is a supervised generalization of IQR.

Algorithm:

- Perform IQR on each class separately.
  - When a sample with an outlier descriptor is found, do not remove it immediately, but rather count for every sample its number of outlying descriptors.
- At the end of the process, remove the samples which have more outlying descriptors than a specified threshold. The threshold is selected according to the permitted false-alarm ratio.

As noted, MCIQR is a generalization of IQR. By artificially labeling all samples in the MISDIN as belonging to a single class and setting to 1 the outlying descriptors threshold, i.e., the number of outlying descriptors a sample should have to be considered an outlier and removed, MCIQR becomes IQR.

C. Self-Consistency Outlier Removal

The introduced Self-Consistency Outlier Removal technique (SCO) is a "wrapper method" in the sense that it utilizes for outlier detection the same classification algorithm used for computing the SCR, that is, Self-Classification.

Algorithm:

- Repeat $\boldsymbol{N}$ times.
  - — Let ***Learning-Set*** $= \boldsymbol{L}\%$ of the samples from each instrument class in the MISDIN, selected randomly.
  - — Let ***Test-Set*** $=$ remaining samples in the MISDIN, i.e., those not in ***Learning-Set***.
  - — Classify the ***Test-Set*** using the ***Learning-Set***.
  - — Record indices of misclassified samples.
- Samples misclassified in at least $M\%$ of the experiments are marked as outliers (and removed).

TABLE I
REMOVAL OF OUTLIERS FROM THE CONTAMINATED SOL EXCERPT WITH UP TO 1% FALSE ALARMS

	Clean MISDIN	Contaminated MISDIN	IQR	MCIQR	SCO
Self-Consistency Rate	**92.7** (92.0 – 93.4)	**79** (78.6 – 79.5)	**88.1** (87.7 – 88.4)	**91.6** (91.3 – 91.8)	**86.2** (85.8 – 86.6)
Removed %:					
Class Noise	N/A		0	53	51.5
Random256	N/A		100	100	39
Random Bound	N/A		81.8	100	43.9
Random Class Bound	N/A		12.5	31.8	7.6
Total Bad Removed	N/A		**49.6**	**70.1**	**35.5**
Total Good Removed	N/A		0.9	0.9	0.95

In this paper, we use $L = 66\%$, which is a common split ratio for AMIR Self-Classification experiments ([4], [5], [8], and others), and $N = 50$ which results in a confidence interval of less than 2%. M is selected according to the permitted false-alarm ratio.

Note that SCO uses partial, randomly selected groups of samples in the Learning Set at each classification step, thus creating a "bagging" effect and lowering the overall distortion in classifications caused by the outliers present in the Learning Set [16].

V. EVALUATION

Each of the outlier removal algorithms is performed using the AMIR descriptor set computed on a methodically contaminated MISDIN. There is a tradeoff between the number of "bad" samples and "good" samples ("false alarms") removed by the algorithms; therefore, each algorithm is evaluated twice; first allowing up to 1% of "good" samples to be removed (Table I), and another time allowing up to 10% "good" samples removed (Table II).

A. Contaminated MISDIN

The proposed techniques are evaluated using an excerpt from the extensive IRCAM Studio On-Ligne (SOL) MISDIN [17]. This excerpt contains 1325 sound samples of 20 "musical instruments": guitar, harp, violin (pizzicato and sustained), viola (pizzicato and sustained), cello (pizzicato and sustained), contrabass (pizzicato and sustained), flute, clarinet, oboe, bassoon,

TABLE II
REMOVAL OF OUTLIERS FROM THE CONTAMINATED SOL EXCERPT WITH UP TO 10% FALSE ALARMS

	Clean MISDIN	Contaminated MISDIN	IQR	MCIQR	SCO
Self-Consistency Rate	**92.7 (92.0 – 93.4)**	**79 (78.6 – 79.5)**	**89.8 (89.5 – 90.2)**	**92.3 (91.5 – 93.1)**	**96.8 (96.4 – 97.1)**
Removed %:					
Class Noise	N/A		18.2	75.7	100
Random256	N/A		100	100	100
Random Bound	N/A		100	100	98.5
Random Class Bound	N/A		50	86.4	51.6
Total Bad Removed	N/A		**67.2**	**90.4**	**87.8**
Total Good Removed	N/A		9.9	8.8	9.5

alto sax, accordion, trumpet, trombone, French horn and tuba. All the samples are two seconds long, monophonic, and sampled in 44.1 kHz with 16-bit resolution.

Our Feature Descriptor set, consisting of 162 feature descriptors, is computed on each sample.

Computing SCR on the SOL excerpt produces a consistency rate of 95.7%. Taking into consideration that as noted, LDA + KNN and the feature descriptor set have been demonstrated as robust in previous studies, this rate shows that the SOL database excerpt, which was professionally recorded and inspected, is indeed quite consistent.

In order to test our techniques for automatic "bad" instrument-sample removal, the SOL excerpt MISDIN is next "contaminated" with four kinds of outlying samples.

1) "*Class Noise*": the class labels of random 5% of the MISDIN samples are changed to different, randomly selected, instrument classes. For example, a violin sample may be intentionally mislabeled as viola.
2) "*Random256 Samples*": samples with descriptor values selected randomly from the range of $[0,\ 255]$ are added to the MISDIN with random class labels. The quantity of these samples is 5% of the original MISDIN size.
3) "*Random Bound Samples*": the minimum and maximum of each descriptor over the entire noncontaminated SOL excerpt are found. 5% of pseudorandom samples are added to the MISDIN, where each descriptor in these samples is bound by its respective minimum and maximum values in the noncontaminated SOL excerpt. For example, if the minimum value of descriptor #1 in the SOL excerpt was 0 and the maximum was 1, then in an added contaminating sample, descriptor #1 may have random values in the range of $[0,\ 1]$.
4) "*Random Class Bound Samples*": 5% of pseudorandom samples are added to each class, with descriptors bound by their respective minimum and maximum values in this class in the noncontaminated SOL excerpt. For example, if the values of descriptor #1 in the violin class were bound by $[0.5,\ 0.9]$, and in the cello class by $[0.2,\ 0.6]$, then in a contaminating sample added to the violin class, descriptor #1 may have random values in the range of $[0.5,\ 0.9]$ while in a contaminating sample added to the cello class, descriptor #1 may have random values in the range of $[0.2,\ 0.6]$.

Naturally, the outlying samples are inserted into SOL before the descriptors are normalized and LDA is computed.

B. Results

Tables I and II show the evaluation results:

1) Columns:

- "Clean MISDIN": shows the SCR of classifying only the "good" samples in the contaminated MISDIN, i.e., the contaminating samples are removed and the Self-Consistency Rate is computed for the remaining samples. Note that this "Clean" MISD has 5% less samples than the original SOL MISD excerpt due to the removal of the contaminating Class Noise samples.
- "Contaminated MISDIN": the SCR of the contaminated database.
- IQR, MCIQR, SCO—the SCRs of the contaminated database after it is purged with each of these algorithms.

2) Rows:

- "Self-Consistency Rate"—as previously noted, this is the average result of 50 self-classification rounds with 66%/34% split. The numbers in parenthesis are the 95% confidence intervals of these SCRs.
- "Removed%:" rows—"Class noise," "Random256, "Random Bound," "Random Class Bound"—the percentage of each type of contaminating samples removed by an algorithm. For example, in Table I, MCIQR has removed 53% of the Class Noise contaminating samples.

3) Lowest Rows ("Total"):

- "Total Bad Removed"—the total percentage of "bad" (contaminating) samples removed.
- "Total Good Removed"—the total percentage of "good" (original) samples removed.

N/A is short for Not Applicable.

Tables I and II reveal that the introduced methods indeed detect the bad samples rather well. Let us examine the types of contaminating samples removed by each algorithm:

4) IQR: As could be expected from its nonsupervised nature, IQR was unable to detect Class Noise. Random 256 and Random Bound outliers were removed as well as the probability of getting at least a single feature descriptor out of 162 with an "edge" value is high with these contamination types, and the presence of a single outlying descriptor is enough for IQR to remove a sample. Samples from the Random Class Bound contamination type are much more difficult for IQR to detect—many descriptors in various classes simply cannot reach outlying values due to their minimum/maximum values over the entire MISDIN. For example, suppose that in the violin class the minimum and maximum values of descriptor #1 are $[-10,\ 10]$, while the minimum and maximum values of descriptor #1 over the entire MISDIN are $[-50,\ 50]$. This means that Random Class Bound samples in the violin class will never have a globally outlying descriptor #1. As IQR does not use class information, it cannot detect samples with such descriptors even if they do have a "local" outlying value in their own class.

5) MCIQR: We can see that the MCIQR method has outperformed the other two, removing higher percentages of "bad" samples for both false-alarm thresholds. As MCIQR uses class information, it did not have the disadvantages of IQR regarding Class Noise and Random Class Bound samples. Another reason

for its higher success ratios is that unlike IQR, it did not immediately remove every sample with a single outlying descriptor, but rather removed samples which had at least "N" outlying descriptors, thus reaching slower the permitted false-alarm rates.

6) SCO: As the SCO algorithm does not have a gradual scale for how much a sample "deserves" to be removed according to its descriptor values, its behavior is the same with all types of contaminating outliers as long as they are misclassified. However, as the Random Class Bound samples had the highest probability of being actually classified by SCO as their appointed class (while possibly having outlying values which could be detected by MCIQR) SCO had the least success removing them compared to other contamination types.

In Table II, we see that SCO has produced the "purged" MISDIN with the highest SCR—96.8%, which is even noticeably higher than the SCR for the "Clean," non-Contaminated MISDIN—92.7%. This high rate was achieved while removing only 87.8% of the contaminating samples versus 9.5% of "good" samples, which is actually less "successful" than MCIQR. This apparent contradiction is actually not very surprising. SCO performs Self-Classification rounds and removes samples which are frequently misclassified. As SCR also uses Self-Classification for computing its rate, SCO actually removes directly samples which are likely to be misclassified by the SCR computation routines. The reasons SCR does not rise up to 100% is that the learning and test sets for the Self-Classification rounds are selected in a random manner, and that we limit SCO by the percentage of Good Samples it is allowed to remove.

Therefore, the relatively high score does not directly mean that SCO outperformed the other algorithms in this case. We should remember that our primary goal is not to get the highest SCR but rather to get rid of the highest number of "bad" samples for the price of a certain percentage of "good" samples removed. This has tempered SCO results in Table I, as SCO removed the allowed 1% of Good Samples relatively fast.

VI. Summary and Conclusion

We have introduced methods for automatically removing "bad" samples from MISDINs, involving computation of Automatic Musical Instrument Recognition feature descriptors on the samples and using outlier detection and classification techniques. We have also introduced the SCR measurement which helps to evaluate a MISDIN self-consistency. Evaluation on a methodically contaminated excerpt of the SOL sound-sample database has shown that these techniques indeed detect bad instrument samples rather well, with the introduced MCIQR method leading with removal of 70.1% of "bad" samples with 0.9% false-alarm rate, and 90.4% "bad" samples with 8.8% false-alarm rate.

For nonlabeled sound collections, out of the three tested algorithms, IQR is the "only way to go" as the other two algorithms require class information. For disposal of "bad" samples in instrument-labeled MISDINs, MCIQR seems to be the best as it has outperformed the other two algorithms in this respect. However, if maximally high SCRs are desired, specifically tailored wrapper-type methods may well be the answer—SCO has scored the highest SCR when allowed up to 10% false-alarm rate.

Note that not every outlier should be always removed—there are many arguments regarding the desirability of the "whole business" of removing outliers. Diversity in a database, which may lower the SCR, is not necessarily bad and may actually model a special, interesting, sample population, such as a breathing noise in a flute sample or the scraping noise of a guitar string, rather than simply indicate sampling or classification errors.

The general rule is to "know your data," thus being able to "intelligently guess" which percentage of erroneous samples could be expected. This allows providing the outlier removal algorithms with appropriate limiting parameters, such as the percentage of samples to remove, and the number of descriptors likely to go wrong. "Knowing the data" also allows tailoring special outlier removing algorithms for very specific data types, such as done in [18], where an algorithm is specifically tailored for removing outliers from different views (graphical images) of the same scenery.

VII. Future Work

The achieved "bad" sample removal rates are rather high using the contaminated SOL excerpt. However, the question still remains whether our contamination types, while methodical, indeed represent real-world errors in MISDINs.

The Class Noise outliers certainly mirror a real situation where sound-samples are classified into wrong categories. Regarding "damaged" samples, it is harder to define exactly what they are, whether these are noisy recordings, samples containing pops and clicks, samples with too much echo, or other possibilities.

A precise analysis of the outliers actually present in MISDINs during production will allow better evaluation of our outlier detection methods and verifying that the suggested AMIR descriptor set indeed models well authentic abnormalities in MISDINs. Such analysis may also allow the development of task-oriented feature descriptors for removing specific types of bad samples from MISDINs.

The SCR definition in this paper is not the only possible one. Other SCRs could be defined using various classification techniques (suitable for AMIR), producing different rates depending on the sensitivity of the algorithms to outliers in the learning set, score computation, and other factors.

References

[1] Y. Li and D. Wang, "Pitch detection in polyphonic music using instrument tone models," in *Proc. 2007 IEEE Conf. Acoust., Speech, Signal Process.*, Apr. 2007, pp. II-481–II-484.

[2] P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 116–128, Jan. 2008.

[3] J. W. Osborne and A. Overbay, "The power of outliers (and why researchers should ALWAYS check for them)," *J. Practical Assess., Res., Eval.*, vol. 9, no. 6, 2004.

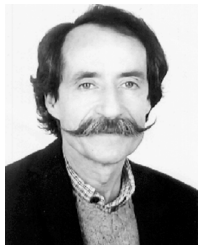
[4] A. Livshin, G. Peeters, and X. Rodet, "Studies and improvements in automatic classification of musical sound samples," in *Proc. Int. Conf. Comput. Music (ICMC)*, 2003, pp. 220–227.

[5] A. Livshin and X. Rodet, "The importance of cross database evaluation in musical instrument sound classification: A critical approach," in *Proc. Int. Symp. Music Inf. Retrieval (ISMIR)*, 2003, pp. 241–242.

[6] A. Livshin and X. Rodet, "Musical instrument identification in continuous recordings," in *Proc. Int. Conf. Digital Audio Effects (DAFX)*, 2004, pp. 222–227.
[7] A. Livshin and X. Rodet, "The significance of the non-harmonic "Noise" versus the harmonic series for musical instrument recognition," in *Proc. Int. Symp. Music Inf. Retrieval (ISMIR)*, 2006, pp. 95–100.
[8] K. D. Martin and Y. E. Kim, "2pMU9. musical instrument identification: A pattern-recognition approach," in *Proc. 136th Meeting Acoust. Soc. Amer.*, 1998, 1768(A).
[9] J. B. Allen, "Short time spectral analysis, synthesis and modification by discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 3, pp. 235–238, Jun. 1977.
[10] G. Peeters, "A Large set of audio features for sound description (similarity and classification) in the CUIDADO Project," CUIDADO I.S.T. Project Report, 2004 [Online]. Available: http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf
[11] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, vol. 8, no. 3, pp. 185–190, 1937.
[12] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. Acoust. Soc. Amer.*, vol. 68, no. 5, pp. 1523–1525, 1980.
[13] D. W. Zimmerman, "Invalidation of parametric and nonparametric statistical tests by concurrent violation of two assumptions," *J. Experimental Education*, vol. 67, no. 1, pp. 55–68, 1998.
[14] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley Interscience, 1992.
[15] Inconsistent Data, Matlab Documentation Mathworks, 2008 [Online]. Available: http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/data_analysis/f0–7275.html
[16] J. François, Y. Grandvalet, T. Denoeux, and J. M. Roger, "Resample and combine: An approach to improving uncertainty representation in evidential pattern classification," *Inf. Fusion*, vol. 4, no. 2, pp. 75–85, 2003.
[17] P. Szendy, "Vers les studios en Ligne -L'Ircam sur les autoroutes de l'information," 1997 [Online]. Available: http://mediatheque.ircam.fr/articles/textes/Szendy97a/,
[18] A. Adam, E. Rivlin, and I. Shimshoni, "ROR: Rejection of outliers by rotations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 78–84, Jan. 2001.

Arie Livshin received the B.Sc. and M.Sc. degrees in computer science from the Hebrew University of Jerusalem, Jerusalem, Israel, and the Ph.D. degree in computer science from IRCAM and the UPMC University (Paris-VI), Paris, France.

His main research interests are pattern recognition, digital signal processing applied to music and Internet usability. He has worked on numerous software projects and is currently a founder and the CTO of VisualBee.com.



Xavier Rodet (M'06) is currently with the Institut de Recherche et Coordination Acoustique/Musique (IRCAM), Paris, France. His research interests are in the areas of signal and pattern analysis, recognition, and synthesis. He has been working particularly on digital signal processing for speech, speech and singing voice synthesis, and automatic speech recognition. Computer music is his other main domain of interest. He has been working on understanding spectro–temporal patterns of musical sounds and on synthesis-by-rules. He has been developing new methods, programs, and patents for musical sound signal analysis, synthesis, and control. He is also working on physical models of musical instruments and nonlinear dynamical systems applied to sound signal synthesis.