Multi-label Ferns for Efficient Recognition of Musical Instruments in Recordings

Miron B. Kursa[1] and Alicja A. Wieczorkowska[2]

[1] Interdisciplinary Centre for Mathematical and Computational Modelling (ICM),
University of Warsaw, Pawińskiego 5A, 02-106 Warsaw, Poland
[2] Polish-Japanese Institute of Information Technology, Koszykowa 86,
02-008 Warsaw, Poland
M.Kursa@icm.edu.pl, alicja@poljap.edu.pl

Abstract. In this paper we introduce multi-label ferns, and apply this technique for automatic classification of musical instruments in audio recordings. We compare the performance of our proposed method to a set of binary random ferns, using jazz recordings as input data. Our main result is obtaining much faster classification and higher F-score. We also achieve substantial reduction of the model size.

1 Introduction

Music Information Retrieval (MIR) is a hot research topic last years [23], [26], with quite a successful solving of such problems as automatic song identification through query-by-example, also using mobile devices [25], [28], and finding music works through query-by-humming [18]. Still, one of the unattainable goals of MIR research is automatic score extraction from audio recordings, which is especially difficult for polyphonic data [8], [12]. Multi-pitch tracking combined with assignment of the extracted notes to particular voices (instruments) is a way to approach score extraction. Therefore, identification of instruments can be used to assign each note in a polyphonic and polytimbral sound to the appropriate instrument. However, the recognition of all playing instruments from recordings in polyphonic environment is still a challenging and unsolved task, related to multi-label classification of audio data representing a mixture of sounds.

In our work, the target is to recognize all instruments playing in the analysed audio segment. No initial segmentation nor providing external pitch is required. The instruments identification is performed on short sound frames, without multi-pitch tracking. In our previous works, we were using sets (which we called batteries) of binary classifiers to solve the multi-label problem [13], [30] of identification of instruments in polyphonic environment. Random forests [2] and ferns [21], [22] were applied as classification tools. Recently, we have shown that random ferns are a good replacement for random forests in music annotation tasks, as this technique offers similar accuracy while being much more computationally efficient [15]. In this paper we propose a generalized version of random ferns, which can natively perform multi-label classification. Using real musical recording data, we will show that our approach outperforms a battery of binary random ferns classifiers in every respect: in terms of accuracy, model size and prediction speed.

1.1 Background

The difficulty level of automatic instrument recognition in audio data depends on the polyphony level, and on the preprocessing performed. For single isolated sounds the instrument identification can even reach 100% for a few classes, but it decreases to about 40% for 30 or more classes [8]). For polyphonic input even labelling of ground truth data is difficult, so mixes and single sounds are commonly applied to facilitate the research on polyphonic audio data. The identification of instruments in polyphony is often supported with external provision of pitch data, but automatic multi-pitch tracking problem is addressed too [7]. Another simplified approach aims at the identification of a predominant instrument [1]. Multi-target identification of multiple instruments is performed as well, although this research is done on various sets of data, so the results cannot be directly compared. This section presents a general view of the state of the art in this area, shows classification methods applied, and sketches the broad range of approaches used in this research.

The simplest polyphonic research case is instrument identification in duets (2 instruments) [4], [10], [29], and the most complex one for symphonies, with high polyphony level (i.e. high number of instrument sounds played together). In all cases the sound waves of instruments overlap, and so harmonic spectral components (partials) do, to a certain — sometimes large — extent. One of the approaches consists in omitting overlapped partials [4], resulting in about 60% accuracy using Gaussian Mixture Models for duets from 5-instrument set. Another interesting approach to multiple-instrument recognition is presented in [3]; their approach was inspired by non-negative matrix factorization, with an explicit sparsity control.

Audio data are usually parametrized before further processing in the classification procedure, and pure data representing amplitude changes of a complex audio wave are rarely used. Preprocessing consists in calculation of parameters describing audio signal, or (more often) spectral features. Still, direct spectrum/template matching can be also for instrument identification, without feature extraction [10], [11]. This approach can result in good accuracy; in [11], 88% was obtained for the polyphony of 3 instruments: flute, violin and piano, supported with integrating musical context into the system.

The accuracy of instrument identification usually drops with increasing polyphony level and number of instruments considered in the recognition procedure. In [12], 84.1% was obtained for duets, 77.6% for trios, and 72.3% for quartets, using LDA (Linear Discriminant Analysis) based approach. In [31], LDA yielded 60% average precision for instrument pairs (300 pairs, 25 instruments), and much a higher recall of 86–100%. Other techniques used in multiple instrument identification include SVM (Support Vector Machine), decision trees, and k-nearest neighbour classifiers [5], [16]. For the polyphony of up to four jazz instruments, the average accuracy of 53% was obtained in [5], whereas [17] obtained 46% recall and 56% precision for the polyphony of up to 4 notes for 6 instruments, based on spectral clustering, and PCA (Principal Component Analysis). The research in [17] aimed at sound separation, which is also performed as an intermediate step in automatic music transcription, and then each separated sound can be independently labelled. Semi-automatic music transcription is addressed in [32] through shift-variant non-negative matrix deconvolution (svNMD) and k-means clustering; the accuracy dropped below 40% for 5 instruments, analysed in form of mixes.

2 Data

The data we use originate from various recordings, all recorded at 44.1kHz/16-bit, or converted to this format. Testing is also performed on recordings, not on mixes. This was possible because we used recordings especially prepared for research purposes, the original tracks for each instruments were available, and thus ground truth labeling was facilitated. Both training and testing data were used as mono input, although some of them were originally recorded in mono or stereo format. In the case of stereo data, mixes of the left and right channel (i.e. the average value of samples in both channels) were taken.

Sound parametrization was performed as a preprocessing in our research, for 40-ms frames. Spectrum was calculated first, using Fourier transform with Hamming window, and various spectral features were extracted. No pitch tracking was performed nor required as preprocessing. Both training and testing data were labelled with the instrument or instruments playing in a given segment. In the testing phase, the identification of instruments is performed on frame by frame basis, for consequent 40-ms frames, with 10 ms hop size (75% overlap).

2.1 Feature Set

The feature vector consists of parameters describing properties of a 40-ms audio frame, and differences of the same parameters but calculated between for a 30 ms sub-frame starting from the beginning of the frame and a 30 ms sub-frame with 10 ms offset. The features we used are mainly MPEG-7 low-level audio descriptors, are often used in audio research [9], and other features applied in instrument recognition research. The following 91 parameters constitute our feature set [13], [30]:

- *Audio Spectrum Flatness*, $\mathit{flat}_1, \ldots, \mathit{flat}_{25}$ — features parameter describing the flatness property of the power spectrum within a frequency bin for selected bins; we used 25 out of 32 frequency bands;
- *Audio Spectrum Centroid* — the power weighted average of the frequency bins in the power spectrum, with coefficients scaled to an octave scale anchored at 1 kHz [9];

- *Audio Spectrum Spread* — RMS (root mean square) of the deviation of the log frequency power spectrum wrt. *Audio Spectrum Centroid* [9];
- *Energy* — energy of the spectrum, in log scale;
- *MFCC* — 13 mel frequency cepstral coefficients. The cepstrum was calculated as the logarithm of the magnitude of the spectral coefficients, and then transformed to the mel scale, reflecting properties of the human perception of frequency. 24 mel filters were applied, and the results were transformed to 12 coefficients, and the logarithm of the energy was taken as 13^{th} coefficient (0-order coefficient of MFCC) [19];
- *Zero Crossing Rate*, where zero-crossing is a point where the sign of the sound wave in time domain changes;
- *Roll Off* — the frequency below which an experimentally chosen percentage (85%) of the accumulated magnitudes of the spectrum is concentrated; parameter originating from speech recognition, applied to distinguish between voiced and unvoiced speech;
- *NonMPEG7 - Audio Spectrum Centroid* — a linear scale version of *Audio Spectrum Centroid*;
- *NonMPEG7 - Audio Spectrum Spread* — a linear scale version of *Audio Spectrum Spread*;
- changes (differences) of the above features for a 30 ms sub-frame of the given 40 ms frame (starting from the beginning of this frame) and the next 30 ms sub-frame (starting with 10 ms offset);
- *Flux* — the sum of squared differences between the magnitudes of the DFT points calculated for the starting and ending 30 ms sub-frames within the main 40 ms frame; this feature works on spectrum directly, not on its parameters.

2.2 Audio Data

In our experiments we focused on wind instruments, typically used in jazz music. Training data for clarinet, trombone, and trumpet were taken from three repositories of single, isolated sounds of musical instruments: McGill University Master Samples (MUMS) [20], The University of Iowa Musical Instrument Samples (IOWA) [27], and RWC Musical Instrument Sound Database [6]. Since so sousaphone sounds were available in these sets, we additionally used sousaphone sounds recorded by R. Rudnicki[24]. Training data were in mono format in RWC data and for sousaphone, and in stereo for the rest of the data. Training was performed on single sounds and mixes. Our classifiers were trained to work on larger instrument sets, so additionally sounds of 5 other instruments were used in the training. These were instruments also typical for jazz recordings: double bass, piano, tuba, saxophone, and harmonica. RWC, IOWA and MUMS repositories were used to collect these sounds. The testing data were taken from the following jazz band stereo recordings by R. Rudnicki [13], [24]:

- *Mandeville* by Paul Motian,
- *Washington Post March* by John Philip Sousa, arranged by Matthew Postle,

- *Stars and Stripes Forever* by John Philip Sousa, semi-arranged by Matthew Postle — Movement no. 2 and Movement no. 3.

These recordings contain pieces played by clarinet, trombone, trumpet, and sousaphone, which are our target instruments.

3 Classification

In the previous works, we have been solving the multi-label problem of recognising instruments with the standard binary relevance approach. Namely, we were building a battery of binary models, each capable of detecting the presence or absence of a single instrument; for prediction, we were applying all the models to the sample and combining their predictions.

Unfortunately, this approach is not computationally effective, ignores the information about instrument-instrument interactions and requires sub-sampling of the training data to make balanced training sets for each battery member. Thus, we attempted to modify the random ferns classifier used in our methodology to natively support multi-label classification.

3.1 Multi-label Random Ferns

Random ferns classifier is an ensemble of K ferns, simple base classifiers equivalent to a constrained decision tree. Namely, the depth of a fern (D) is fixed and the splitting criteria on a given tree level are identical. This way, a fern has 2^D leaves and directs object x into a leaf number $F(x) = 1+\sum_{i=1}^{D} 2^{i-1}\sigma_i(x) \in 1..2^D$, where $\sigma_i(x)$ is an indicator variable for a result of the i-th splitting criterion. We are using the rFerns implementation of random ferns [14] which generates splitting criteria entirely at random, i.e. randomly selects both a feature on which the split will be done and the threshold value. Also, rFerns builds a bagging ensemble of ferns, i.e. each fern, say k-th, is not directly build not on a whole set of objects but on a *bag* B_k, a multiset of training objects created by random sampling with replacement the same number of objects as in the original training set.

The leaves of ferns are populated with *scores* $S_k(x, y)$, indicating the confidence of a fern k that an object x falling into a certain leaf $F_k(x)$ belongs to the class y. The scores are generated based on a training dataset $X^t = \{x^t_1, x^t_2, \ldots\}$, and are defined as

$$e^{S_k(x,y)} = \frac{1 + |L_k(x) \cap Y_k(y)|}{C + |L_k(x)|} \cdot \frac{C + |B_k|}{1 + |Y_k(y)|}, \tag{1}$$

where $L_k(x) = \{x^t \in B_k : F_k(x) = F_k(x^t)\}$ is a multiset of training objects from a bag in the same leaf as a given object and $Y_k = \{x^t \in B_k : y \in Y(x^t)\}$ is a multiset of training objects from a bag that belong to a class y. $Y(x)$ denotes a set of true classes of an object x, and is assumed to always contain a single element in a many-classes case; C is the number of all classes. The prediction of the whole ensemble for an object x is

$$Y^p(x) = \arg\max_y \sum_{k=1}^{K} S_k(x, y). \tag{2}$$

Our proposed generalisation of random ferns for multi-label classification is based on the observation that while the fern structures are not optimised to a given problem, the same set of F_k functions can serve all classes rather than being re-created for each one of them. In the battery classification, we create virtual *not-class* classes to get a baseline score value used to decide whether class of a certain score value should be reported as present or absent. With multi-class random ferns, however, we can incorporate this idea as a normalisation of scores so that the sign of their value will become meaningful indicator of a class presence. We call such normalised scores *score quotients* $Q_k(x, y)$, and define them as

$$e^{Q_k(x,y)} = \frac{1 + |L_k(x) \cap Y_k(y)|}{1 + |L_k(x) \setminus Y_k(y)|} \cdot \frac{1 + |B_k \setminus Y_k(y)|}{1 + |Y_k(y)|}. \tag{3}$$

The prediction of the whole ensemble for an object x naturally becomes

$$Y^p(x) = \{y : Q_k(x, y) > 0\}. \tag{4}$$

4 Experiments

When preparing training data, we start with single isolated sounds of each target instrument. After removing starting and ending silence [13], each file representing the whole single sound is normalized so that the RMS value equals one. Then, we create the training set of sounds by mixing random 40 ms frames extracted from the recordings of 1 to 4 randomly chosen instruments; the mixing is done with random weights and the result is normalized again to get the RMS value equal to one. Finally, we convert the sound into a vector of features by applying previously described sound descriptors. The multi-label decision for such an object is a set of instruments which sounds were used to create the mix. We have repeated this procedure 100 000 times to prepare our training set.

This set is used directly to generate the model with the multi-label random ferns approach. When creating the battery of random ferns, we are splitting this data into a set of binary problems. Each is devoted to one instrument and contains 3000 positive examples where this instrument contributed to the mix and 3000 negative when it was absent.

In both cases, we used $K = 1000$ ferns and scanned depths $D = 5, 7, 10, 11, 12$. As the random ferns is a stochastic algorithm, we have replicated training and testing procedure 10 times.

Both models are tested on a true jazz recordings described in Section 2.2 and their predictions assessed with respect to the annotation performed by an expert. The accuracy was assessed via precision and recall scores; these measures were weighted by the RMS of a given frame, in order to diminish the impact of softer frames which cannot be reasonably identified as their loudness approaches the noise level. Our true positive score T_p for an instrument i is a sum of RMS of frames which are both annotated and classified as i. Precision is calculated by dividing T_p by the sum of RMS of frames which are classified as i; respectively, recall is calculated by dividing T_p by the sum of RMS of frames which are annotated as i.

As a general accuracy measure we have used F-score, defined as a harmonic mean of such generalised precision and recall.

5 Results

The results of accuracy analysis are presented in a Figure 1. One can see that for fern depth larger than 7 the multi-label ferns archived both significantly better precision and recall that the battery classifier; obviously this also corresponds to a higher F-score. The precision of both methods seems to stabilise for larger depth, while the recall and so F-score of multi-class ferns raise steadily and may be likely further improved. The variation of the results is also substantially smaller for multi-class ferns, showing that the output of this approach is more stable and thus more predictable.

Table 1 collects the sizes of created models and the speed with which they managed to predict the investigated jazz pieces. One can see that the utilisation of multi-label ferns results in a substantially greater prediction speed, on average 7 times better than this achieved by the battery of binary ferns.

The difference between model sizes is less pronounced, with the battery actually producing a smaller model for fern depths 5 and 7 and only about 2 times larger for fern depth 12.

There is a negative correlation between the achieved F-score and both prediction speed and model size, though, with the fern depth controlling the speed-quality trade-off. However, this way an user may use this parameter to flexibly adjust the model to the constraints of the intended implementation.

6 Summary and Conclusions

In this paper we introduce multi-label random ferns as a tool for automatic identification of musical instruments in polyphonic recordings of a jazz band. The comparison of performance of multi-label random ferns and sets of binary ferns shows that the proposed multi-label ferns outperform the sets of binary ferns in every respect. Multi-label ferns are much faster, achieve higher F-score, and the model size increase with increasing complexity also compares favorably with the set of binary random ferns. Therefore, we conclude that multi-label random ferns can be recommended as a classification tools in many applications, not only for instrument identification, and this technique can also be applied on resource-sensitive devices, e.g. mobile devices.

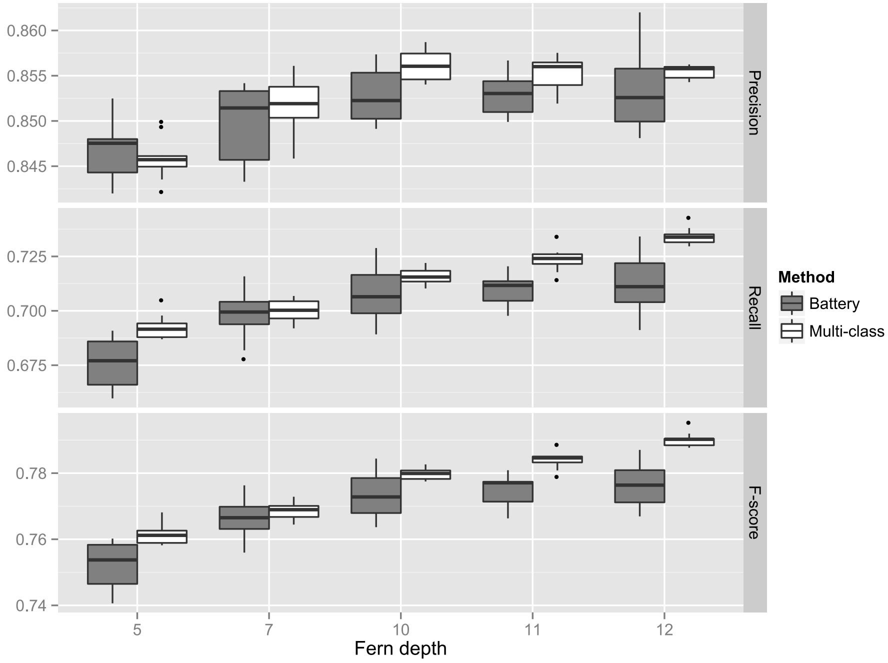

Fig. 1. Overall precision, recall and F-score for all the investigated jazz recordings and all the instruments for a battery of binary random ferns and for multi-label ferns.

	Model size		Prediction speed	
Fern depth	Battery	Multi-label	Battery	Multi-label
5	6MB	13MB	54×	359×
7	20MB	19MB	42×	301×
10	143MB	81MB	33×	238×
11	284MB	151MB	30×	216×
12	565MB	292MB	26×	204×

Table 1. Comparison of model size and prediction speed for a random ferns battery and multi-label random ferns. The speed is expressed as the total playing time of all investigated jazz recordings divided by the CPU time required to classify them.

Acknowledgments. This project was partially supported by the Research Center of PJIIT, supported by the Ministry of Science and Higher Education in Poland, and the Polish National Science Centre, grant 2011/01/N/ST6/07035. Computations were performed at the ICM UW, grant G48-6.

References

1. Bosch, J.J., Janer, J., Fuhrmann, F., Herrera, P.: A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals In: 13th International Society for Music Information Retrieval Conference (ISMIR), 559–564 (2012)
2. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
3. Cont, A., Dubnov, S., Wessel, D.: Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negativity constraints. In: Proc. 10th Int. Conf. Digital Audio Effects (DAFx-07), 85–92 (2007)
4. Eggink, J., Brown, G.J: Application of missing feature theory to the recognition of musical instruments in polyphonic audio. In: 4th International Conference on Music Information Retrieval ISMIR (2003)
5. Essid, S., Richard, G., David, B.: Instrument recognition in polyphonic music based on automatic taxonomies. IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 1, 68–80 (2006)
6. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In: 4th International Conference on Music Information Retrieval ISMIR, 229–230 (2003)
7. Heittola, T., Klapuri, A., Virtanen, A.: Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation. In: Proc. 10th Int. Society for Music Information Retrieval Conf. (ISMIR 2009) (2009)
8. Herrera-Boyer, P., Klapuri, A., Davy, M.: Automatic Classification of Pitched Musical Instrument Sounds. In: Klapuri, A., Davy, M. (eds.) Signal Processing Methods for Music Transcription. Springer Science+Business Media LLC (2006)
9. ISO: MPEG-7 Overview, `http://www.chiariglione.org/mpeg/`
10. Jiang, W., Wieczorkowska, A., Ras, Z.W.: Music Instrument Estimation in Polyphonic Sound Based on Short-Term Spectrum Match. In: Hassanien, A.-E., Abraham, A., Herrera, F. (Eds.): Foundations of Computational Intelligence Vol. 2. Approximate Reasoning. Studies in Computational Intelligence Vol. 202, 259-273, Springer (2009)
11. Kashino, K., Murase, H.,: A sound source identification system for ensemble music based on template adaptation and music stream extraction. Speech Commun., vol. 27, 337-349 (1999)
12. Kitahara, T., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. EURASIP J. Appl. Signal Process., vol. 2007, 1-15 (2007)
13. Kubera, E., Kursa, M.B., Rudnicki, W.R., Rudnicki, R., Wieczorkowska, A.A.: All That Jazz in the Random Forest. In: Kryszkiewicz, M., Rybiński, H., Skowron, A., Raś, Z.W. (eds.): ISMIS 2011. LNAI, vol. 6804, pp. 543–553. Springer, Heidelberg (2011)
14. Kursa, M.B.: Random ferns method implementation for the general-purpose machine learning (2012), `http://arxiv.org/abs/1202.1121v1`, submitted

15. Kursa, M.B.: Robustness of Random Forest-based gene selection methods. BMC Bioinformatics, Vol. 15(1), No. 8, pp. 1–8 (2014)
16. Little, D., Pardo, B.: Learning Musical Instruments from Mixtures of Audio with Weak Labels. 9th International Conference on Music Information Retrieval ISMIR (2008)
17. Martins, L.G., Burred, J.J., Tzanetakis, G., Lagrange, M.: Polyphonic instrument recognition using spectral clustering. 8th International Conference on Music Information Retrieval ISMIR (2007)
18. MIDOMI: Search for Music Using Your Voice by Singing or Humming, `http://www.midomi.com/`
19. Niewiadomy, D., Pelikant, A.: Implementation of MFCC vector generation in classification context. J. Applied Computer Science, Vol. 16, No. 2, pp. 55–65 (2008)
20. Opolko, F., Wapnick, J.: MUMS — McGill University Master Samples. CD's (1987)
21. Özuysal, M., Fua, P., Lepetit, V.: Fast Keypoint Recognition in Ten Lines of Code. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2007)
22. Özuysal, M. Calonder, M., Lepetit, V., Fua, P.: Fast Keypoint Recognition using Random Ferns. Image Processing `http://dx.doi.org/10.1109/TPAMI.2009.23` (2008)
23. Ras, Z.W., Wieczorkowska, A.A. (eds.): Advances in Music Information Retrieval. Studies in Computational Intelligence, Vol. 274, Springer Heidelberg (2010)
24. Rudnicki, R.: Jazz band. Recording and mixing. Arrangements by M. Postle. Clarinet — J. Murgatroyd, trumpet — M. Postle, harmonica, trombone — N. Noutch, sousaphone – J. M. Lancaster (2010)
25. Shazam Entertainment Ltd `http://www.shazam.com/`
26. Shen, J., Shepherd, J., Cui, B., Liu, L. (eds.): Intelligent Music Information Systems: Tools and Methodologies. Information Science Reference, Hershey (2008)
27. The University of IOWA Electronic Music Studios: Musical Instrument Samples, `http://theremin.music.uiowa.edu/MIS.html`
28. TrackID, `https://play.google.com/store/apps/details?id=com.sonyericsson.trackid`
29. Vincent, E., Rodet, X.: Music transcription with ISA and HMM. In: 5th Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA) 1197–1204 (2004)
30. Wieczorkowska, A.A., Kursa, M.B: A Comparison of Random Forests and Ferns on Recognition of Instruments in Jazz Recordings. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W (eds.): ISMIS 2012. LNAI, vol. 7661, pp. 208–217. Springer, Heidelberg (2012)
31. Barbedo, J.G.A., Tzanetakis, G: Musical Instrument Classification Using Individual Partials. IEEE Transactions on Audio, Speech & Language Processing Vol. 19 No. 1, 111–122 (2011)
32. Kirchhoff, H., Dixon, S., Klapuri, A.: Multi-Template Shift-Variant Non-Negative Matrix Deconvolution for Semi-Automatic Music Transcription. In: 13th International Society for Music Information Retrieval Conference (ISMIR), 415–420 (2012)