Musical instrument recognition in polytimbral polyphonic music audio signals

Andrew Trahan
University of Rochester
andrew.trahan@rochester.edu

ABSTRACT

This paper gives an overview of the state of current research in recognition of musical instruments in complex audio signals. The criteria for a complex musical signal in this case are polyphony (multiple notes being present at the same time) and the occurrence of multiple instruments with their own, unique acoustic signature (otherwise known as timbre, hence "polytimbral") within a given time frame of the signal. The primary goal of this research is to gather musically meaningful information about the instruments present in a musical signal and to present an output that can be understood by the user and can also be used later by other algorithms. In order to reach this goal, recognition systems have certain requirements in order to be considered acceptable and useful for musical instrument recognition. There is also a general architecture that these systems exhibit in their implementation. Two distinct approaches to musical instrument recognition dominate the field of current research: pure pattern recognition and informed pattern recognition. In this paper, both of these approaches are explained and examined (as well as the different methods used in relevant implementations in recent articles). Studies that focus on percussive instruments exclusively are not considered, as they exhibit different challenges and algorithmic complexities. This paper concludes with observations and suggestions for further research, including reflections on the prevalence of the use of personal music databases (as opposed to public databases specifically created for instrument recognition) and the resultant issue of comparing the performance of different instrument recognition algorithms.

1. INTRODUCTION

1.1 Standards of Algorithms

Martin [20] proposed that there are six main standards that sound-source recognition systems should be evaluated by, and that implementations of such systems should strive to achieve in each category.

The first criterion is *generalization*. Generalization refers to the ability of implementations to create instrument categories that accurately model the true "general" audio fingerprint of an individual instrument, despite some variances. For example, a category that wishes to model an acoustic guitar should be able to generalize that acoustic guitars made of different types of wood should all be able to be grouped into its acoustic guitar category. Also, an example widely applicable to instruments is the variable of sound loudness, which should not affect the modeling of such categories. This criterion seeks to create stability in the structure.

The second criterion concerns *data handling*. Recognition systems should be able to process real data, not only synthesized sound sources and synthesized acoustic mixtures but real acoustic situations. The complexity of these real-world signals contains much more deviations than synthesized data, and (much like the generalization criterion) leads to increased stability in the system.

The third criterion is *scalability*. This concerns the ability of a musical instrument recognition system to learn new instrument categories in an elegant and intuitive way. A capable system should be able to be implemented with different numbers of categories and the influence of the amount of categories on the system's performance should be observable and quantifiable.

The fourth criterion is the *robustness* of the system. Robustness refers to the ability of the system to perform under non-ideal circumstances (which could include, for example, the presence of reverberation and non-source associated noise).

The fifth criterion is the *adaptivity* of the system. According to this principle, musical instrument recognition systems should employ semi-supervised learning algorithms that have to the ability to update their definitions periodically according to data gleaned from new inputs.

The sixth criterion is the ability of the implementation of the recognition system to be *processed in real time*. Since perception of sound is a time dependent process, it is important to approach the problem of instrument recognition as a processing of a sequence of time instances (i.e. frames).

1.2 Architecture of Algorithms

The general architecture of musical instrument recognition systems are similar in regards to their modular design. Fuhrmann [13] describes four general processing steps in any given algorithm: *pre-processing, feature processing, classification,* and *post processing*. As previously stated, these steps are modular and can be modi-

fied, replaced, or sometimes skipped altogether in certain implementations.

Pre-processing involves leveraging prior knowledge for the sake of better informing the musical instrument recognition system. This prior knowledge can be as simple as raw data input by a user (such as defining how many instruments are in the acoustic mixture), or it can be a separate, complex algorithm (such as source separation). Audio segmentation (windowing) is generally seen as part of the pre-processing step.

Feature processing is the procedure of extracting low-level information (features) in the audio segments. Each segment is associated with a set of values (one for each extracted feature) called a feature vector. There are many features that can be extracted; these features are briefly explained in Section 3.2: Audio Segment Features.

The *classification processing* step compares trained class models to the feature vector for each segment and gives each potential class a probability value.

Post-processing re-weights the output from the classification processing step using other information. This "external" information may be garnered from the context of previously processed audio segments. From this step, the final output information is produced for each audio segment.

1.3 Additional Considerations

There are other features to be considered when implementing algorithms for musical instrument recognition studies. One of the most important aspects to consider is whether or not any pre-processing or post-processing was involved in the implementation. Related to pre-processing is the consideration of if the implementation requires certain *a priori* knowledge.

In the classification step performance is greatly influenced by the number of different categories that the detected musical instruments can be placed into (which can be considered as a measurement for the overall scope of the implementation). For example, it can be assumed that an implementation that is only concerned with sorting inputs into two categories, woodwinds and non-woodwinds, will perform much better than one that has a separate class for each woodwind instrument.

The performance of such implementations is also greatly related to restrictions on the input. Some studies only consider a small number of different input instruments, and also a low value for the number of maximum polyphony. Another restriction to the input that is closely related to instrumentation is the processing of audio mixtures from only certain genres. This restricts acoustic variation and can lead to higher performance measures.

Perhaps one of the most important factors that can influence the performance of musical instrument recognition is the way the algorithm input is generated. This means that the way the acoustic mixtures are created can prove to be very influential. The processed mixtures can be created completely artificially by MIDI. They can also be created manually by taking "real" solo instrument sources and mixing them. These solo instrument sources can vary in nature as well, as some may have artifacts such as reverberation that can confound algorithms. The most difficult input for musical instrument recognition algorithms to process is a real-world acoustic mixture (for example, a live recording of a string quartet in a reverberant room). All of these factors must be considered when comparing different studies.

2. METHODS

2.1 Learning Algorithms

This section aims to touch upon the most commonly used learning algorithms used in current musical instrument recognition studies. It is by no means an in-depth explanation and discussion of the comparative strengths and weaknesses of each method, but instead seeks to provide a basic knowledge foundation so that the approaches mentioned in Section 4 can be understood on an elementary level.

2.1.1 Unsupervised Learning Algorithms

Unsupervised learning algorithms are techniques that create classification categories from the input data only (without any prior knowledge of class membership). Among the most popular unsupervised learning algorithms are mixture models, hidden Markov models, independent component analysis, probabilistic latent component analysis, and non-negative matrix factorization.

Mixture models are probabilistic models that represent the presence of subsets within an overall set. It seeks to make statistical inferences about the properties of subclasses, without being given information on the aforementioned subclasses [11].

Hidden Markov models (HMMs) are statistical Markov models in which the system that is to be modeled is assumed to be a Markov process (a memoryless stochastic process) with unobserved states. HMMs are often used for temporal pattern recognition [22].

Independent component analysis (ICA) separates multivariate signals into subcomponents that are additive. It operates under the assumption that all of the subcomponents are statistically independent non-Gaussian signals [4].

Probabilistic latent component analysis (PLCA) is a probabilistic model that defines spectra as distributions and extracts sets of additive components. It is an extension of probabilistic latent semantic indexing (PLSI) and exhibits sparsity and shift invariance [12].

Non-negative matrix factorization

(NMF) seeks to define a high dimensional matrix as a pair of two lower dimensional matrices. One of the matrices acts as a "dictionary" of sounds, and the other acts as an excitation detector. The requirement that the three matrices contain no negative elements makes it easier to inspect the data contained [25].

2.1.2 Supervised Learning Algorithms

Supervised learning algorithms are techniques that depend on information provided prior to evaluation. These algorithms learn relations between pre-defined categories and sample inputs. Supervised learning algorithms are usually implemented by nearest neighbor, artificial neural networks, and support vector machines.

Nearest neighbor algorithms seek to predict class memberships based on a set number of closest training examples in a multidimensional feature space. This algorithm is very simple to implement, as an object is simply classified by majority vote [7].

Artificial neural networks are systems that seek to model the central nervous system in their approach. The systems favor an interconnected approach that simulates neurons, which compute output values by sending input information through the created network [8].

Support vector machines (SVM) use sets of input data to predict which of two possible classes each input data point belongs to (otherwise known as non-probabilistic binary linear classification). An SVM algorithm is provided a training input data set (in which each provided data point is specified as belonging to a certain class) to create a model that can be used with future inputs to classify them according to where on the sample space they are located [6].

2.2 Acoustic Features

There are a plethora of acoustic features that can be extracted from audio segments, and in many musical instrument recognition algorithm implementations dozens are considered. These acoustic features can be classified into related groups. These groups are described subsequently.

Mel frequency cepstral coefficients (MFCCs) are obtained by calculating the cepstrum of energy bands created from the Mel scale [19]. The cepstrum is calculated by taking the Inverse Fourier transform (IFT) of the logarithm of the calculated spectrum of the signal. The Mel scale is a pitch scale derived from psychoacoustic perceptions of pitch height. Implementations that take into account MFCCs generally use the first 10 to 20 coefficients to create a spectral envelope estimation.

Linear prediction coefficients [23] are also used to create a spectral envelope of audio signals. The spectral envelope is created by extrapolating sample values of the signal, which is done by combining the previous samples and assigning weights to them. The coefficients calculated are these weights.

Local energies [21] are helpful and easily calculated acoustic features. Local energy is calculated by dividing the frequency spectrum into energy bands. The energies of these bands are considered as well as the total energies of groups of bands.

The pitch present in an audio signal is an important acoustic feature. Estimated pitch values can be used to determine the harmonic content of a signal. Associated features include pitch confidence levels and harmonic energy ratios [2] [21].

Spectral features [21] are a large group of features used to describe very low-level aspects of the spectrum of a signal. Among the features most used are the spectral flatness, crest, flux, roll-off, centroid, spread, skewness, kurtosis, and general spectral complexity [26].

3. STUDIES

The following section reviews some of the more recent approaches to musical instrument recognition. In light of the scope of this paper, only studies that incorporate polytimbral, polyphonic inputs are considered for review. Also, studies that focus on percussive instruments exclusively are not considered, as they exhibit somewhat different challenges and algorithmic complexities. Studies are presented by approach and within each subsection, by date of publication.

There are two main approaches to polytimbral, polyphonic instrument recognition: pure pattern recognition and informed pattern recognition. *Pure pattern recognition* algorithms are performed on unadulterated inputs (on the polyphonic signal) and identify dominant instruments in the signal (or in some cases just the most dominant instrument). Complex inputs are handled by modifying category constraints and definitions according to the inputs. *Informed pattern recognition* algorithms place emphasis on pre-processing by applying source-separation and/or multi-pitch estimation.

3.1 Pure Pattern Recognition Algorithms

Simmermacher [24] used SVM classifiers to categorize four different input instruments with polyphony of up to four simultaneous notes. The classifiers were trained by real note samples from the IOWA collection. MFCCs and MPEG-7 (multimedia content metadata) features were used in feature selection. It is reasonable to assume that the performance of the implementation was assisted by the choice of the four different input instruments (flute, piano, trumpet, violin), each of which are considered to be drastically different from each other in terms of acoustic signature (as well as theoretically motivated, since each of these instruments are regarded as belonging in different instrument "families"). Since the reported performance of the algorithm was high, it would be interest-

	Experiment Information				Algorithm Information			
Author	Polyphony	Categories	Input Type	Collection	Method	A Priori	Pre-Processing	Post-Processing
Simmermacher et al. (2006)	4	4	Real	Personal	SVM	No	No	No
Essid et al. (2006a)	4	12	Real	Personal	SVM	No	No	Yes
Little & Pardo (2008)	3	4	Artificial Mix	IOWA	SVM	No	No	Yes
Kobayashi (2009)	Not Stated	10	Real	Personal	LDA/RA	No	No	No
Giannoulis & Klapuri (2013)	4	10	Artificial Mix	MUMS	GMM	No	No	No
Eggink & Brown (2004)	Not Stated	5	Real	Personal	GMM	No	Yes	No
Kitahara et al. (2007)	4	5	Synthesized MIDI	RWC	Gaussian	Yes	Yes	Yes
Cont et al. (2007)	2	2	Real	Personal	NMF	No	Yes	No
Heittola et al. (2009)	6	19	Artificial Mix	RWC	GMM	Yes	Yes	Yes
Burred et al. (2010)	4	5	Artificial Mix	RWC	Probability Distribution	No	Yes	No
Barbedo & Tzanetakis (2011)	7	25	Real	RWC	Majority Rules	No	Yes	Yes

Table 1. Comparison of referenced approaches.

ing to record the performance of the same algorithm using more similar instruments.

The approach of Essid [10] is interesting since it seeks to identify and label the timbres of groups of instruments (by SVM) as opposed to individual instruments. This implementation poses certain philosophical issues in terms of generality and future development. The implementation supports four different instruments (and a polyphony of four) and has a relatively high amount of categories (12). As the support for more instruments is increased, the number of categories increases dramatically. This poses a problem in terms of processing time as the number of supported instrument combination classes increases.

Little & Pardo [18] used weakly-labeled data sets to obtain classifiers for instruments from polytimbral mixtures. Four different instruments were used from the IOWA collection and a polyphony of three was supported. A comparison was made between classifiers trained by artificial audio mixtures and classifiers trained by isolated notes, with those trained by the mixtures outperforming those trained by isolated notes by a significant factor.

Kobayashi [17] combined genetic algorithms with linear discriminant analysis to generate a feature set (as opposed to the SVM approach). Ten general instrument categories were created, however the supported polyphony was not reported. Although the reported recognition accuracy was quite high, this evaluation may not be accurate. Concern arises as it is clear that the music pieces used in the evaluation were also used in the training set. The music pieces were split up into 1-second windows and were randomly chosen on an individual basis to either be in the training set or in the evaluation set. This may have overfit the classifiers and artificially caused high recognition.

Giannoulis and Klapuri [14] created a novel algorithm that employs local spectral features and missing-feature techniques to run a mask estimation system. This system defines spectral regions as "reliable" or "unreliable" for the estimation of a sound source. The reliable spectral regions are then associated with a class and unreliable vector elements are treated by bounded marginalization. This approach outperformed a Gaussian mixture model (GMM) system that utilized MFCC features in polyphonies of 2 and 4. However, the novel algorithm did not perform as well on monophonic signals.

3.2 Informed Pattern Recognition Algorithms

Eggink & Brown [9] used a fundamental frequency approximation algorithm to locate the partials of the dominant instrument in a musical section. A GMM was utilized to create models for five classical instruments and for each fundamental frequency. This was done to acknowledge the fact that an instrument's timbre changes depending on the pitch of the note played. In addition, unknown frames were handled by retrieving a probabilistic measure for each potential model and choosing the one with the highest probability.

Kitahara [16] introduced a thorough probabilistic approach that utilized a fundamental frequency estimation algorithm to detect different melodic lines in audio mixture. This is augmented by an instrument probability system, which is governed by hidden Markov models that takes into account every fundamental frequency in addition to an additional 28 features. Note and instrument probabilities are multiplied together to find the highest probability possible. Unfortunately, the model only took into account four different instruments and supported polyphony of just three.

The implementation of Cont [5] focused on a NMF system for pitch and instrument estimation. The modulation spectrum of the input was used to run the NMF algorithm. Note templates were created as single basis functions in the classification matrix. Unknown inputs were then compared to the matrix, and pitch and instrument estimations were output. Unfortunately, the algorithm only supports polyphony of two notes (and consequently two categories).

Heittola [15] developed an application that utilized a polyphonic pitch estimation algorithm to run an NMF source separation algorithm. Each separated source is analyzed by pre-trained GMMs and is classified accordingly. There are a few issues with the implementation, as the number of sources present is needed as *a priori* information. Also, the audio mixtures are required to be separated into 4-second windows, which may be too long to be musically relevant. In addition, the number of sources in the 4-second window is assumed to remain constant, which is not an accurate reflection of a real-world music mixture situation.

Burred [3] created timbre models of instruments by using principal component analysis on the extracted spectral envelopes of training data inputs. Source separation was utilized before template matching to order to extract the aforementioned envelopes.

The overall goal of the system described by Barbedo and Tzanetakis [1] is slightly different than most implementations. The goal is to identify all of the instruments present in a signal, not in a single frame. All instruments detected as present in more than 5% of the frames is considered as "present" in the overall signal. Pre-processing algorithms are used for source estimation and fundamental frequency estimation for each frame. Upper partials are found by peak picking, which are then processed through a filter to isolate them. Features are then extracted and majority voting is used on each frame.

4. DISCUSSION

One of the greatest difficulties in comparing musical instrument recognition algorithm implementation is the prevalence of researchers using their own input data. A

few databases exist that, at first glance, seem to be appropriate for such studies. Such databases (also referred to as collections) include the McGill University Master Samples (MUMS), and University of Iowa Musical Instrument Samples Database (IOWA). While a few of the present databases are extensive, there exist a few issues with these. The content of the databases tend to focus on a very "clean" representation of the instruments sampled. The instruments are usually recorded in anechoic chambers and almost always as solo instruments. These samples, while helpful in some degree, are not reflective of natural acoustic situations. Natural acoustic situations would exhibit multiple instruments playing synchronously in "live" rooms (areas that have a natural reverberation). These natural situations would also include other confounds, such as noise. One of the main objectives of musical instrument detection algorithms is to be able to match a human level of ability, if not to exceed it. The process of human training for instrument detection is a lifelong process that does not depend on clean representations of instrumental mixtures. In light of this, algorithm training should aim to take into account the confounds of real-world stimuli. Including solo and group instrumental recordings may be a logical improvement to existing databases, and would also be an interesting focal point for the creation of new databases. Another improvement would be to include recordings made in multiple acoustic environments. These acoustic environments would be carefully characterized and measured prior to recording. Thus, not only would these recordings be good for comparing the performance of an algorithm in different acoustic conditions, but also these measurements may later be used in algorithms as *a priori* information to take into account real-world confounds such as reverberation and ambient room noise.

Another issue with comparing the performance of different studies is the issue of researchers not releasing their training and evaluation input data. If the data is not easily accessible, it is difficult for further studies to be done that seek to recreate previous work with their own improvements and to compare the results in a meaningful way. This relates back to the tendency for researchers creating their own input data. If the audio for their studies were available, it seems that others would be less inclined to use their own datasets as well.

Extremely limiting input dependencies are also a problem. For example, some studies only consider musical samples from certain genres (or even only one genre). It is debatable whether or not genre should even be considered an important specification, as the types of input instruments is specifically what we are trying to identify.

The ideal musical instrument recognition algorithm would create a musically meaningful instrument model to match an input to. If classes were created to be both specific enough to not output a false positive, but also general enough to be able to take into consideration small variances in timbre (inherent by the nature of a physical musical instrument), these classes could be used in many different implementations and situations. The more thorough testing of classes could lead to a better understanding of the performance of different algorithms and also the relative usefulness of different acoustic features.

5. REFERENCES

[1] Barbedo, J. & Tzanetakis, G.: "Musical Instrument Classification using individual partials", *IEEE Transactions on Audio, Speech, and Language Processing,* 19(1), 111–122, 2011.

[2] Brossier, P.: "Automatic annotation of musical audio for interactive applications", *Ph.D. thesis*, Queen Mary University, London, 2006.

[3] Burred, J., Robel, A., and Sikora, T.: "Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds", *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3), 663–674, 2010.

[4] Comon, Pierre: "Independent Component Analysis: a new concept?", *Signal Processing*, 36(3):287–314 , 1994.

[5] Cont, A., Dubnov, S., & Wessel, D.: "Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints", *In Proceedings of the Interna- tional Conference on Digital Audio Effects (DAFx*), pp. 85–92, 2007.

[6] Cortes, Corinna; and Vapnik, Vladimir N.: "Support-Vector Networks", *Machine Learning,* 20, 1995.

[7] Cover, T.; Hart, P.: "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on* , vol.13, no.1, pp.21,27, January 1967.

[8] Deng, Li; Hinton, Geoffrey; Kingsbury, Brian: "New types of deep neural network learning for speech recognition and related applications: an overview," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* , vol., no., pp.8599,8603, 26-31, May 2013.

[9] Eggink, J. & Brown, G.: "Instrument recognition in accompanied sonatas and concertos," *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 217–220, 2004.

[10] Essid, S., Richard, G., & David, B.: "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1), 68–80, 2006a.

[11] Figueiredo, M.A.T.; Jain, A.K.: "Unsupervised Learning of Finite Mixture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (3): 381-396, March 2002.

[12] Fuentes, B.; Badeau, R.; Richard, G.: "Adaptive harmonic time-frequency decomposition of audio using shift-invariant PLCA," *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE Interna-*

tional Conference on , vol., no., pp.401,404, 22-27 May 2011.

[13] Fuhrmann, F. "Automatic musical instrument recognition from polyphonic music audio signals," *Ph.D. thesis, Universitat Pompeu Fabra, Barcelona*, 2012.

[14] Giannoulis, D., and Klapuri, A.: "Musical instrument recognition in polyphonic audio using missing feature approach*," In Audio, Speech, and Language Processing, IEEE Transactions on* (Volume:21, Issue 9), 2013.

[15] Heittola, T., Klapuri, A., & Virtanen, T.: "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," *In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 327–332, 2009.

[16] Kitahara, T., Goto, M., Komatani, K., Ogata, T., & Okuno, H.: "Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps*," EURASIP Journal on Advances in Signal Processing,* 2007, 1–16, 2007.

[17] Kobayashi, Y.: "Automatic generation of musical instrument detector by using evolutionary learning method," *In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 93–98, 2009.

[18] Little, D. & Pardo, B.: "Learning musical instruments from mixtures of audio with weak labels," *In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR*), pp. 127–132, 2008.

[19] Logan, B.: "Mel frequency cepstral coefficients for music modeling," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR),* 2000.

[20] Martin, K. "Sound-source recognition: A theory and computational model*," Ph.D. thesis, Massachusetts Institute of Technology (MIT)*, MA, USA, 1999.

[21] Peeters, G.: "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," *Tech. rep,* 2004.

[22] Rabiner, L.: "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE* , vol.77, no.2, pp.257,286, Feb 1989.

[23] Schwarz, D.: "Spectral envelopes in sound analysis and synthesis,*" Master's thesis, Universität Stuttgart*, 1998.

[24] Simmermacher, C., Deng, D., & Cranefield, S. "Feature analysis and classification of classical musical instruments: an empirical study," *Lecture Notes in Computer Science*, 4065, 444–458, 2006.

[25] Smaragdis, P.; Brown, J.C.: "Non-negative matrix factorization for polyphonic music transcription," *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* , vol., no., pp.177,180, 19-22, Oct. 2003.

[26] Streich, S.: "Music complexity: a multi-faceted description of audio content," *Ph.D. thesis, Universitat Pompeu Fabra, Barcelona*, 2006.