# MUSICAL INSTRUMENT RECOGNITION USING ICA-BASED TRANSFORM OF FEATURES AND DISCRIMINATIVELY TRAINED HMMS

*Antti Eronen*
Tampere University of Technology, Institute of Signal Processing
P.O. Box 553, FIN-33101 Tampere, Finland
antti.eronen@tut.fi

## ABSTRACT

In this paper, we describe a system for the recognition of musical instruments from isolated notes or drum samples. We first describe a baseline system that uses uses mel-frequency cepstral coefficients and their first derivatives as features, and continuous-density hidden Markov models (HMMs). Two improvements are proposed to increase the performance of this baseline system. First, transforming the features to a base with maximal statistical independence using independent component analysis can give an improvement of 9 percentage points in recognition accuracy. Secondly, discriminative training is shown to further improve the recognition accuracy of the system. The evaluation material consists of 5895 isolated notes of Western orchestral instruments, and 1798 drum hits.

## 1 INTRODUCTION

Earlier work on musical instrument recognition has mainly used classifiers that are not able to effectively model the temporal evolution of spectral features. The Gaussian mixture model (GMM) ([1]) is able to effectively parameterize the distribution of observations. However, it does not explicitly model the dynamic evolution of feature values within a played note. One approach is to extract features that explicitly try to measure the temporal characteristics of isolated notes [2], or to manually segment the notes and use averages of cepstral coefficients during the onset (the beginning of a note) and steady state as features [3]. However, this has only a limited ability to model the temporal evolution even if feature variances were also used as features. Moreover, often the extraction of temporal features is computationally rather demanding and the effect is even greater if this is combined with the use of a nearest-neighbour classifier, for instance.

Hidden Markov models (HMM) are the mainstream statistical model used in the speech recognition community, and are now becoming increasingly popular also in non-speech applications. To our knowledge, Casey is the only researcher who has used HMMs to model musical instrument samples [4]. As a part of the development of the generalized audio descriptors for the MPEG-7 standard, he has evaluated the proposed methods using a database consisting of a wide variety of audio, including music, speech, environmental sounds, and different musical instrument sounds.

However, Casey's evaluation data has included examples of only a few instruments. In addition, little detail has been given on the difficulty of the evaluation material, making assessing the accuracy of his method in instrument recognition difficult. Moreover, no details were given on the topology of the resulting models, since their algorithm attempts to force some of the transition probabilities to zero during training [4].

In this paper, we take a different approach. Based on the knowledge of physical properties of musical instruments, and on the other hand the psychological studies on timbre perception, there is a clear motivation for using HMMs with a left-right topology to model isolated notes. Most musical instruments have a distinctive onset period, followed by a steady state, and finally decay (or release). For instance, some instruments are characterized by onset asynchrony, which means that the energy of certain harmonics rises more quickly than the energy at some other frequencies. Also the decay is often characterised by the prominence of certain frequencies with respect to others. This causes the features relating to the spectral shape to have different value distributions during the onset, steady state, and decay. Thus, a left-right HMM with three states might well model this temporal evolution.

This paper first describes the development of a baseline instrument recognizer that uses mel-frequency cepstrum (MFCC) and delta cepstrum (ΔMFCC) coefficients as features, and HMMs to model the feature distributions. The system is evaluated using a database consisting of isolated notes of 27 Western orchestral instruments, and a smaller database of drum hits. We propose two improvements to improve the performance of the system. First, we use the independent component analysis (ICA) to transform the feature vector consisting of catenated MFCC and ΔMFCC features to a basis with maximal statistical independence. This transform is shown to give an almost consistent improvement in recognition accuracy over the baseline with no rotation. Second, we propose using discriminative training of the HMMs. Especially with computationally attractive models with low number of components in state densities, discriminative training gives an improvement over the baseline maximum likelihood (ML) training using the Baum-Welch re-estimation algorithm.

## 2 FEATURE EXTRACTION

### 2.1 Feature extraction

Mel-frequency cepstral coefficients (MFCC) were found to be a well-performing feature set in musical instrument recognition [3], and are used as the front-end parameters in our system. The input signal is first pre-emphasized with an FIR filter having the transfer function $1 - az^{-1}$, where $a$ was between 0.97 and 0.99 in our simulations. MFCC analysis is performed in 30 ms windowed frames advanced every 15 ms for the orchestral instruments. For the analysis of short drum sounds, the frame length was reduced into 20 ms, and the hop size was 4 ms. The number of triangular filters was 40, and they occupied the band from 30Hz to half the sampling rate. For the drum sounds, the lowest frequency was 20Hz. The number of cepstral coefficients was 12 after the zeroth coefficient was discarded, and appending the first time derivatives approximated with a 3-point first-order polynomial fit resulted in a feature vector size of $n$ = 24. The resulting features were both mean and variance normalized.

### 2.2 Transforming features using independent component analysis (ICA)

Independent component analysis (ICA) has recently emerged as an interesting method for finding decorrelating feature transformations [4][5][6]. The more well-known methods for include the principal component analysis and linear discriminant analysis. The goal of ICA is to find directions of minimum mutual information, i.e. to extract a set of statistically independent vectors from the training data **X**. The use of an ICA transformation has been reported to improve the recognition accuracy in speech recognition [5]. In the MPEG-7 generalized audio descriptors, ICA is proposed as an optional transformation on the spectrum basis obtained with singular value decomposition [4], and Casey's results have shown the success of this method on a wide variety of sounds. Our approach is slightly different from all these studies. We perform ICA on concatenated MFCC and ΔMFCC features. In [4] and [5] only static features were used, and in [6] logarithmic energies and their derivatives were used.

In order to construct the $m$-by-$n$ ICA transform matrix **W**, the extracted MFCC and ΔMFCC coefficients from the training data samples are gathered into a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T]$ where each column represents the catenated MFCC (s) and ΔMFCC (d) features from the analysis frame $t$, i.e. $\mathbf{x}_t = [x_{s1}, x_{s2}, ..., x_{s(n/2)}, x_{d1}, ..., x_{d(n/2)}]^T$. The total amount of feature vectors from all recordings of all the classes in the training set is denoted by $T$. The class and recording indices are omitted here since ICA does not utilize class information. The ICA demixing matrix **W** is applied on **X** producing the transformed observation space **O** = **WX**, which is of dimension $m$-by-$T$, where $m \leq n$. The inequality is due to possible dimensionality reduction in the preprocessing step, which consists of a whitening transform.

The efficient FastICA algorithm was used for finding the ICA basis transformation [7]. It should be noted that the extra computational load caused by applying the ICA transformation occurs mainly in the off-line training phase. The test phase consists of computing the MFCC and ΔMFCC features in the usual way plus an additional multiplication with the $m$-by-$n$ matrix **W** derived off-line using the training data.

## 3 CLASSIFICATION

### 3.1 The hidden Markov model

Hidden Markov models with a left-right topology are used to model the distribution of feature vectors from each instrument category, and the classification is made with the maximum-a-posteriori rule. A continuous density hidden Markov model (HMM) with $N$ states consists of a set of parameters $\theta$ that comprises the $N$-by-$N$ transition matrix, the initial state probabilities, and the parameters of the state densities. We use diagonal-covariance Gaussian-mixture state densities which are parameterized by the weights, means, and diagonal variances. The model parameters are estimated using a training set that consists of the recordings $\mathbf{O} = [\mathbf{O}^1, ..., \mathbf{O}^R]$ and their associated class labels $L = (l^1, ..., l^R)$. Specifically, $\mathbf{O}^r = [\mathbf{o}_1, ..., \mathbf{o}_{T_r}]$ denotes the sequence of feature vectors measured from the recording $r$. The length of the observation sequence $\mathbf{O}^r$ is $T_r$. In this paper, each recording represents a single note played by an orchestral instrument, or a drum hit.

In our baseline system, the HMM parameters are iteratively optimized using the Baum-Welch re-estimation that finds a local maximum of the maximum likelihood (ML) objective function

$$F(\Theta) = \sum_{c=1}^{C} \sum_{r \in A_c} \log p(\mathbf{O}^r \mid c),$$

where $\Theta$ denotes the entire parameter set of all the classes $c \in \{1, ..., C\}$, and $A_c$ denotes the recordings from the class $c$. In the recognition phase, an unknown recording **Y** is classified using the maximum a posteriori rule:

$$\hat{c} = \arg\max_c p(\mathbf{Y} \mid c)$$

which is due to the Bayes' rule and assuming equal priors for all classes $c$. In this paper, the Viterbi-algorithm was used to approximate the above likelihoods.

### 3.2 Discriminative training

In the case that a statistical model fits poorly the data, training methods other than ML may lead into better-performing models. Discriminative training methods such as the maximum mutual information (MMI) aim at maximizing the ability to distinguish between the observation sequences generated by the model of the correct class and those generated by models of other classes [8]. The MMI objective function is given as

$$M(\Theta) = \log p(L \mid \mathbf{O}) = \sum_{r=1}^{R} \log p(l^r \mid \mathbf{O}^r)$$
$$= \sum_{r=1}^{R} \left\{ \log\left[p(l^r) p(\mathbf{O}^r \mid l^r)\right] - \log \sum_{c=1}^{C} p(c) p(\mathbf{O}^r \mid c) \right\}$$

where $p(l^r)$ and $p(c)$ are prior probabilities. Unfortunately, this requires rather complicated optimization involving the entire model set even if observations from a single class were used.

In this paper, a recently-proposed discriminative training algorithm is used. The algorithm was proposed by Ben-Yishai and Burshtein, and is based on an approximation of the maximum mutual information [9]. Their *approximated maximum mutual information* (AMMI) criterion is:

$$J(\Theta)=\sum_{c=1}^{C}\left\{\sum_{r\in A_c}\log\left[p(c)p\left(\mathbf{O}^r\middle|c\right)\right]-\lambda\sum_{r\in B_c}\log\left[p(c)p\left(\mathbf{O}^r\middle|c\right)\right]\right\},$$

where $B_c$ is the set of indices of training recordings that were *recognized* as *c*. $B_c$ is obtained by maximum a posteriori classification performed on the training set, using initial models trained with the Baum-Welch algorithm. The “discrimination rate” is controlled using the parameter $0 \le \lambda \le 1$.

The prior probabilities $p(c)$ do not affect the maximization of $J(\Theta)$, thus the maximization is equivalent to maximizing for all the classes $1 \le c \le C$ the following objective functions:

$$J_c(\Theta)=\sum_{r\in A_c}\log p\left(\mathbf{O}^r\middle|c\right)-\lambda\sum_{r\in B_c}\log p\left(\mathbf{O}^r\middle|c\right).$$

Thus, the parameter set of each class can be estimated separately, which leads to a straightforward implementation. Ben-Yishai and Burshtein have derived the re-estimation equations for HMM parameters [9]. Due to space restrictions, we present only the re-estimation equation for the transition probability from state *i* to state *j*:

$$\overline{a}_{ij}=\frac{\sum_{r\in A_c}\sum_{t=1}^{T_r-1}\xi_t(i,j)-\lambda\sum_{r\in B_c}\sum_{t=1}^{T_r-1}\xi_t(i,j)}{\sum_{r\in A_c}\sum_{t=1}^{T_r-1}\gamma_t(i)-\lambda\sum_{r\in B_c}\sum_{t=1}^{T_r-1}\gamma_t(i)},$$

where $\xi_t(i,j)=p(q_t=i,q_{t+1}=j\,|\,\mathbf{O}^r,c)$ and $\gamma_t=\sum_{j=1}^{N}\xi_t(i,j)$. The state at time *t* is denoted by $q_t$, and the length of the observation sequence $\mathbf{O}^r$ is $T_r$. In a general form, for each parameter $\nu$ the re-estimation procedure is

$$\nu=\frac{N(\nu)-\lambda N_D(\nu)}{D(\nu)-\lambda D_D(\nu)}$$

where $N(\nu)$ and $D(\nu)$ are the accumulated statistics computed according to the set $A_c$, and $N_D(\nu)$ and $D_D(\nu)$ are the statistics computed according to the set $B_c$, obtained by recognition on the training set. Thus, in a typical situation the set $B_c$ includes examples from the class $c$ and some other confusing classes. This discriminative re-estimation can be iterated in a manner similar to the standard expectation-maximisation. We typically used 5 iterations, although using just one iteration seemed to be sufficient in many situations, since the recognition accuracy did not improve much after the first iteration.

# 4 VALIDATION EXPERIMENTS

## 4.1 Validation database

Our experimental setup aimed at testing the system’s generalization ability across significant variations in recording setup and instrument instances. Samples from five different sources were used in the validation database. The sources were the McGill University Master Samples collection (MUMS) [10], the University of Iowa Electronic Music Studios website [11], IRCAM Studio Online [12], a Roland XP-30 synthesizer, and recordings arranged by Keith Martin at MIT Media Lab [2]. A total of 5895 samples of 27 Western orchestral instruments were included in the database, of which 4940 were included in the training set and 955 were tested. The division into training and test sets was done so that all the samples from a particular instrument instance in a certain recording session were either in the training or test set, i.e. the recognition was done across recordings and different instrument pieces. The recognition was performed at an intermediate level of abstraction using seven classes, which were *the brass, saxophones, single reed clarinets, double reed oboes, flutes, bowed strings,* and *plucked strings*. A random guesser would score 14% correct in these conditions. The drum database consisted of samples from 8 different synthesizer sound banks and the MUMS collection [10]. Samples of two sound banks were used in the training set (total of 1123 drum hits), and the samples of the seven remaining sources were used for testing (a total of 675). The five possible categories were *bass drum, cymbal, hi hat, snare,* and *tom-tom.*

## 4.2 Results

The Baum-Welch algorithm was used to train the baseline HMMs. The number of states (NS) and component densities per state (NC) was varied. Increasing the number of components in each state was obtained by gradually increasing the model order until the desired order NC was obtained by splitting the component with the largest weight. The state means and variances were initialized using a heuristic segmentation scheme, where each sound was segmented into as many adjacent segments as there were states in the model. The initial mean and variance for each state were estimated from the statistics accumulated from the different segments of all samples. During training, a straightforward form of regularization was applied by adding a small constant to the variance elements falling below a predetermined threshold.

Table 1 presents the results obtained using the baseline system using MFCC plus ΔMFCC features and HMMs trained using the Baum-Welch algorithm. In Table 2, the features have been ICA transformed; the HMM training is similar to the baseline. Table 3 shows the results using the MFCC plus ΔMFCC front-end, but using discriminative training of HMMs. In Table 4, both enhancements have been combined and the ICA transformed input is modelled with discriminatively trained HMMs. It can be observed

**Table 1.** Percentage correct in instrument identification, baseline system with MFCC plus ΔMFCC features and ML training.

| % correct | NC=1 | NC=2 | NC=4 | NC=6 | NC=8 |
|---|---|---|---|---|---|
| NS = 2 | 44 | 47 | 57 | **60** | 59 |
| NS = 3 | 53 | 59 | **60** | 58 | 58 |
| NS = 4 | 59 | 57 | 60 | **62** | **62** |
| NS = 5 | 56 | 60 | 60 | 60 | **62** |

**Table 2.** Percentage correct in instrument identification, ICA-based transformation applied and ML training of HMMs.

| % correct | NC=1 | NC=2 | NC=4 | NC=6 | NC=8 |
|---|---|---|---|---|---|
| NS = 2 | 48 | 56 | 60 | 63 | **66** |
| NS = 3 | 57 | 62 | 63 | 65 | **67** |
| NS = 4 | 58 | 61 | **66** | 60 | 61 |
| NS = 5 | 63 | **66** | 64 | **66** | 62 |

**Table 3.** Percentage correct in instrument identification, baseline features and discriminative training of HMMs.

| % correct | NC=1 | NC=2 | NC=4 | NC=6 | NC=8 |
|---|---|---|---|---|---|
| NS = 2 | 45 | 51 | 59 | 61 | **62** |
| NS = 3 | 58 | **63** | 59 | 59 | 58 |
| NS = 4 | 58 | 61 | 60 | 61 | **64** |
| NS = 5 | 58 | **62** | **62** | 61 | **62** |

**Table 4.** ICA-based transformation applied and discriminative training of HMMs.

| % correct | NC=1 | NC=2 | NC=4 | NC=6 | NC=8 |
|---|---|---|---|---|---|
| NS = 2 | 51 | 57 | 61 | 65 | **66** |
| NS = 3 | 57 | 64 | 64 | 66 | **68** |
| NS = 4 | 60 | 60 | **65** | 61 | 61 |
| NS = 5 | 65 | **67** | 63 | 65 | 62 |

**Table 5.** Percentage correct in drum recognition, MFCC plus ΔMFCC features.

|  | **NC = 1** | **NC = 2** | **NC = 3** | **NC = 4** |
|---|---|---|---|---|
| NS = 2 | 79 | 79 | 80 | 78 |
| NS = 3 | 76 | 77 | 79 | 81 |

**Table 6.** Percentage correct in drum recognition, ICA-based transformation applied.

|  | **NC = 1** | **NC = 2** | **NC = 3** | **NC = 4** |
|---|---|---|---|---|
| NS = 2 | 80 | 80 | 78 | 78 |
| NS = 3 | 78 | 81 | 85 | 85 |

that using the ICA transform gives an almost consistent improvement in recognition accuracy across the set of model orders tested. Using discriminative training improves the accuracy mainly with models having low number of components in state densities. This is understandable since low-order models give relatively low recognition accuracy in the training set, and there is not so much danger of over-fitting due to discriminative training as with higher order models. Different values of $\lambda$ were tested, and the results are shown for $\lambda$ =0.3.

Tables 5 and 6 show the results for the drum database using the baseline system and the ICA transformation. Here the improvement is not consistent across the different model orders evaluated, which may be partly due to the larger mismatch in training and testing conditions in this database, and the relatively smaller size of training data where examples from only two sound banks are included.

## 5 CONCLUSION

A system for the recognition of musical instrument samples was described. Applying an ICA-based transform of features gave an almost consistent improvement in recognition accuracy compared to the baseline. The accuracy could be further improved by using discriminative training of the hidden Markov models. Future work will consider the extension of these methods for monophonic phrases.

## ACKNOWLEDGMENT

We thank Assaf Ben-Yishai for his kind answers to our questions on discriminative training. The efforts of Iowa music studios, Ircam, and Keith Martin and Youngmoo Kim in making sound samples freely available are greatly appreciated.

## REFERENCES

[1] J. C. Brown, “Feature dependence in the automatic identification of musical woodwind instruments”. *J. Acoust. Soc. Am*., Vol. 109, No. 3, pp. 1064-1072, 2001.

[2] K. D. Martin, Sound-Source Recognition: *A Theory and Computational Model*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999. Available at http://sound.media.mit.edu/Papers/kdm-phdthesis.pdf.

[3] A. Eronen, “Comparison of features for musical instrument recognition”. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 19-22, Oct. 2001.

[4] M. Casey, “Generalized Sound Classification and Similarity in MPEG-7”. *Organized Sound*, 6:2, 2002.

[5] I. Potamitis, N. Fakotakis, G. Kokkinakis, ”Independent component analysis applied to feature extraction for robust automatic speech recognition”. *Electronics Letters*, Vol. 36, No. 23, Nov 2000.

[6] A. Kocsor, J. Csirik, “Fast Independent Component Analysis in Kernel Feature Spaces”. *In Proc. SOFSEM 2001*, Springer-Verlag LNCS 2234, pp. 271-281, 2001.

[7] A. Hyvärinen. “Fast and Robust Fixed-Point Algorithms for Independent Component Analysis”. *IEEE Transactions on Neural Networks* 10(3):626-634, 1999. Matlab software available at: http://www.cis.hut.fi/projects/ica/fastica/.

[8] L. R. Rabiner, B.-H. Juang. *Fundamentals of Speech Recognition*, PTR Prentice-Hall Inc., New Jersey, 1993.

[9] A. Ben-Yishai, D. Burshtein, “A Discriminative Training Algorithm for Hidden Markov Models”. Submitted to *IEEE Transactions on Speech and Audio Processing*.

[10] F. Opolko, J. Wapnick, *McGill University Master Samples* (compact disk). McGill University, 1987.

[11] The University of Iowa Electronic Music Studios, website. http://theremin.music.uiowa.edu

[12] Ircam Studio Online, website. http://soleil.ircam.fr/