Transcribing Multi-instrument Polyphonic Music with Hierarchical Eigeninstruments

Graham Grindlay, *Student Member, IEEE,* Daniel P.W. Ellis, *Senior Member, IEEE*

***Abstract*—This paper presents a general probabilistic model for transcribing single-channel music recordings containing multiple polyphonic instrument sources. The system requires no prior knowledge of the instruments present in the mixture (other than the number), although it can benefit from information about instrument type if available. In contrast to many existing polyphonic transcription systems, our approach explicitly models the individual instruments and is thereby able to assign detected notes to their respective sources. We use training instruments to learn a set of linear manifolds in model parameter space which are then used during transcription to constrain the properties of models fit to the target mixture. This leads to a hierarchical mixture-of-subspaces design which makes it possible to supply the system with prior knowledge at different levels of abstraction.**

The proposed technique is evaluated on both recorded and synthesized mixtures containing two, three, four, and five instruments each. We compare our approach in terms of transcription with (*i.e.* detected pitches must be associated with the correct instrument) and without source-assignment to another multi-instrument transcription system as well as a baseline NMF algorithm. For two-instrument mixtures evaluated with source-assignment, we obtain average frame-level F-measures of up to 0.52 in the completely blind transcription setting (*i.e.* no prior knowledge of the instruments in the mixture) and up to 0.67 if we assume knowledge of the basic instrument types. For transcription without source assignment, these numbers rise to 0.76 and 0.83, respectively.

***Index Terms*—Music, polyphonic transcription, NMF, subspace, eigeninstruments**

I. Introduction

Music transcription is one of the oldest and most well-studied problems in the field of music information retrieval (MIR). To some extent, the term "transcription" is not well-defined, as different researchers have focused on extracting different sets of musical information. Due to the difficulty in producing all the information required for a complete musical score, most systems have focused only on those properties necessary to generate a pianoroll representation that includes pitch, note onset time, and note offset time. This is the definition of transcription that we will use in this paper, although we will consider the additional property of instrument source.

In many respects music transcription resembles speech recognition: in both cases we are tasked with the problem of decoding an acoustic signal into its underlying symbolic form. However, despite this apparent similarity, music poses a unique set of challenges which make the transcription problem particularly difficult. For example, even in a multi-talker speech recognition setting, we can generally assume that when several talkers are simultaneously active, there is little overlap between them both in time and frequency. However, for a piece of music with multiple instruments present, the sources (instruments) are often highly correlated in time (due to the underlying rhythm and meter) as well as frequency (because notes are often harmonically related). Thus, many useful assumptions made in speech recognition regarding the spectro-temporal sparsity of sources may not hold for music transcription. Instead, techniques which address source superposition by explicitly modeling the mixing process are more appropriate.

A. NMF-based Transcription

Non-negative matrix factorization (NMF) [1], [2] is a general technique for decomposing a matrix $\mathbf{V}$ containing only non-negative entries into a product of matrices $\mathbf{W}$ and $\mathbf{H}$, each of which also contains only non-negative entries. In its most basic form, NMF is a fully unsupervised algorithm, requiring only an input matrix $\mathbf{V}$ and a target rank K for the output matrices $\mathbf{W}$ and $\mathbf{H}$. An iterative update scheme based on the *generalized EM* [3] algorithm is typically used to solve for the decomposition:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} \tag{1}$$

NMF has become popular over the last decade in part because of its wide applicability, fast multiplicative update equations [4], and ease of extension. Much of the recent work on NMF and related techniques comes from the recognition that for many problems, the basic decomposition is under-constrained. Many different extensions have been proposed to alleviate this problem, including the addition of penalty terms for sparsity [5], [6], [7] and temporal continuity [8], [9], [10].

In addition to other problems such as source separation [11], [12], NMF and extensions thereof have been shown to be effective for single-channel music transcription [13], [14], [15], [16]. In this situation the algorithm is typically applied to the magnitude spectrogram of the target mixture, $\mathbf{V}$, and the resulting factorization is interpreted such that $\mathbf{W}$ corresponds to a set of spectral basis vectors and $\mathbf{H}$ to a set of activations of those basis vectors over time. If $\mathbf{V}$ contains only a single instrument source, we can view $\mathbf{W}$ as a set of spectral tem-

Manuscript received September 30, 2010; revised XX 00, 20XX. This work was supported by the NSF grant IIS-0713334. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

G. Grindlay and D.P.W. Ellis are with the Department of Electrical Engineering, Columbia University, New York, NY, 10027 USA, e-mail: {grindlay,dpwe}@ee.columbia.edu

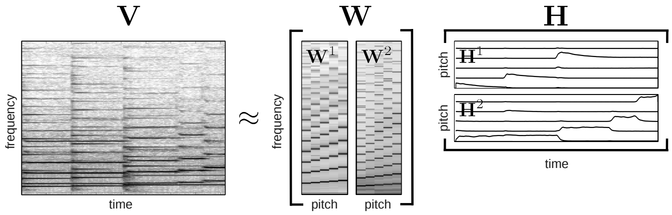


Fig. 1. Illustration of the basic NMF transcription framework. In this example two instrument sources each with five pitches are considered. This results in sub-models $\mathbf{W}^1$ and $\mathbf{W}^2$ as well as transcriptions $\mathbf{H}^1$ and $\mathbf{H}^2$.

plates, one per pitch.[1] Thus, $\mathbf{H}$ gives the degree to which each pitch is active in each time frame and represents most of the information needed for transcription. This basic formulation can be extended to handle a mixture of S instrument sources

$$\mathbf{V} \approx \sum_{s=1}^{S} \mathbf{W}^s \mathbf{H}^s \tag{2}$$

by simply interpreting the basis and weight matrices as having block forms. This concept is illustrated in Figure 1 for a mixture of synthetic piano and flute notes.

The NMF decomposition can be used for transcription in both supervised ($\mathbf{W}$ is known *a priori* and therefore held fixed) and unsupervised ($\mathbf{W}$ and $\mathbf{H}$ are solved for simultaneously) settings. However, difficulties arise with both formulations. For unsupervised transcription it is unclear how to determine the number of basis vectors required, although this is an area of active research [17]. If we use too few, a single basis vector may be forced to represent multiple notes, while if we use too many some basis vectors may have unclear interpretations. Even if we manage to choose the correct number of bases, we still face the problem of determining the mapping between bases and pitches as the basis ordering is typically arbitrary. Furthermore, while this framework is capable of separating notes from distinct instruments as individual columns of $\mathbf{W}$ (and corresponding rows of $\mathbf{H}$), there is no simple solution to the task of organizing these individual columns into coherent blocks corresponding to particular instruments. Recent work on the problem of assigning bases to instrument sources has included the use of classifiers, such as support vector machines [18], and clustering algorithms [19].

In the supervised context, we already know $\mathbf{W}$ and therefore the number of basis vectors along with their ordering, making it trivial to partition $\mathbf{H}$ by source. The main problem with this approach is that it assumes that we already have good models for the instrument sources in the target mixture. However, in most realistic use cases we do not have access to this information, making some kind of additional knowledge necessary in order for the system to achieve good performance.

One approach, which has been explored in several recent papers, is to impose constraints on the solution of $\mathbf{W}$ or its equivalent, converting the problem to a semi-supervised form. Virtanen and Klapuri use a source-filter model which constrains the basis vectors to be formed as the product of excitation and filter coefficients [20]. This factorization can result in a decomposition requiring fewer parameters than an equivalent NMF decomposition and has been used for tasks such as instrument recognition [21]. Vincent *et al.* impose harmonicity constraints on the basis vectors by modeling them as combinations of deterministic narrow-band spectra [14], [22]. More recently, this model was extended by Bertin *et al.* to include further constraints that encourage temporal smoothness in the basis activations [23].

B. Multi-instrument Transcription

Although there has been substantial work on the monophonic [24] and polyphonic [25], [26], [27], [28], [23] transcription problems, many of these efforts have ignored the important task of assigning notes to their instrument sources. Exceptions include work by: Kashino *et al.* on hypothesis-driven musical scene analysis [29]; Vincent and Rodet on multi-instrument separation and transcription using independent subspace analysis and factorial hidden Markov models [30]; Leveau *et al.* on sparse dictionary-based methods that, although tested primarily on instrument recognition tasks, could be adapted to the transcription problem [31]; Kameoka *et al.* on *harmonic temporal clustering* (HTC) [32] which defines a probabilistic model that accounts for timbre and can label notes by instrument; a system for detecting and tracking multiple note streams using higher-order hidden Markov models proposed by Chang *et al.* [33]; and the multi-pitch tracking work of Duan *et al.* [34], [35]. Duan *et al.* take a multi-stage approach which consists of multi-pitch estimation followed by segmentation and grouping into instrument tracks. The track formation stage, which they motivate using psycho-acoustic principles of perceptual grouping, is accomplished using a constrained clustering algorithm. It is important to note that this system makes the simplifying assumption that each instrument source is monophonic. Thus, it cannot be used for recordings containing chords and multi-stops.

In previous work, we introduced a semi-supervised NMF variant called *subspace NMF* [15]. This algorithm consists of two parts: a training stage and a constrained decomposition stage. In the first stage, the algorithm uses NMF or another non-negative subspace learning technique to form a model parameter subspace, $\boldsymbol{\Theta}$, from training examples. In the second stage of the algorithm, we solve for the basis and activation matrices, $\mathbf{W}$ and $\mathbf{H}$, in a fashion similar to regular NMF, except we impose the constraint that $\mathbf{W}$ must lie in the subspace defined by $\boldsymbol{\Theta}$. This approach is useful for multi-instrument transcription as the instrument model subspace not only solves the ordering problem of the basis vectors in the instrument models, but also drastically reduces the number of free parameters. Despite not meeting the strict definition of eigenvectors, we refer to these elements of the model as "eigeninstruments" to reinforce the notion that they represent a basis for the model parameter space.

Recently, it has been shown [36] that NMF is very closely related to *probabilistic latent semantic analysis* (PLSA) [37] as well as a generalization to higher-order data distributions

[1] In an unsupervised context, the algorithm cannot be expected to disambiguate individual pitches if they never occur in isolation; if two notes always occur together then the algorithm will assign a single basis vector to their combination.

called *probabilistic latent component analysis* (PLCA) [7]. Although in many respects these classes of algorithms are equivalent (at least up to a scaling factor), the probabilistic varieties are often easier to interpret and extend. In more recent work, we introduced a probabilistic extension of the subspace NMF transcription system called *probabilistic eigeninstrument transcription* (PET) [16]. In this paper, we present a hierarchical extension of the PET system which allows us to more accurately represent non-linearities in the instrument model space and to include prior knowledge at different levels of abstraction.

II. Method

Our system is based on the assumption that a suitably-normalized magnitude spectrogram, $\mathbf{V}$, can be modeled as a joint distribution over time and frequency, $P(f,t)$. This quantity can be factored into a frame probability $P(t)$, which can be computed directly from the observed data, and a conditional distribution over frequency bins $P(f|t)$; spectrogram frames are treated as repeated draws from an underlying random process characterized by $P(f|t)$. We can model this distribution with a mixture of latent factors as follows:

$$\begin{aligned} P(f,t) &= P(t)P(f|t) \\ &= P(t)\sum_z P(f|z)P(z|t) \end{aligned} \quad (3)$$

Note that when there is only a single latent variable z this is the same as the PLSA model and is effectively identical to NMF. The latent variable framework, however, has the advantage of a clear probabilistic interpretation which makes it easier to introduce additional parameters and constraints. It is worth emphasizing that the distributions in (3) are all multinomials. This can be somewhat confusing as it may not be immediately apparent that they represent the probabilities of time and frequency *bins* rather than specific values; it is as if the spectrogram were formed by distributing a pile of energy quanta according to the combined multinomial distribution, then seeing at the end how much energy accumulates in each time-frequency bin. This subtle yet important distinction is at the heart of how and why these factorization-based algorithms work.

Suppose now that we wish to model a mixture of S instrument sources, where each source has P possible pitches, and each pitch is represented by a set of Z components. We can extend the model described by (3) to accommodate these parameters as follows:

$$P(f|t) = \sum_{s,p,z} P(f|p,z,s)P(z|s,p,t)P(s|p,t)P(p|t) \quad (4)$$

A. Instrument Models

1) Eigeninstruments: $P(f|p,z,s)$ represents the instrument models that we are trying to fit to the data. However, as discussed in Section I, we usually don't have access to the exact models that produced the mixture and a blind parameter search is highly under-constrained. The solution proposed in our earlier work [15], [16], which we extend here, is

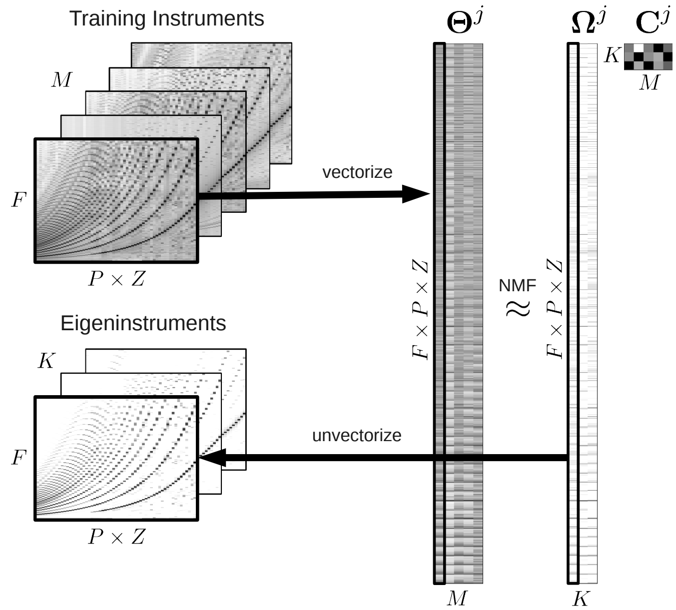


Fig. 2. Formation of the j^{th} instrument model subspace using the eigeninstrument technique. First a set of training models (shown with $Z = 1$) are reshaped to form model parameter matrix $\mathbf{\Theta}^j$. Next, NMF or a similar subspace algorithm is used to decompose $\mathbf{\Theta}^j$ into $\mathbf{\Omega}^j$ and $\mathbf{C}^j$. Finally, $\mathbf{\Omega}^j$ is reshaped to yield the probabilistic eigeninstruments for subspace j, $P_j(f|p,z,k)$.

to model the instruments as mixtures of basis models or "eigeninstruments". This approach is similar in spirit to the eigenvoice technique used in speech recognition [38], [39].

Suppose that we have a set of instrument models $\mathcal{M}$ for use in training. Each of these models $\mathcal{M}_i \in \mathcal{M}$ contains the Z separate F-dimensional spectral vectors for each of the P possible pitches as rendered by instrument i at a fixed velocity (loudness). Therefore $\mathcal{M}_i$ has FPZ parameters in total which we concatenate into a super-vector, $\mathbf{m}_i$. These super-vectors are then stacked together into a matrix, $\mathbf{\Theta}$, and NMF with some rank K is used to find $\mathbf{\Theta} \approx \mathbf{\Omega}\mathbf{C}$.[2] The set of coefficient vectors, $\mathbf{C}$, is typically discarded at this point, although it can be used to initialize the full transcription system as well (see Section III-E). The K basis vectors in $\mathbf{\Omega}$ represent the eigeninstruments. Each of these vectors is reshaped to the F-by-P-by-Z model size to form the eigeninstrument distribution, $P(f|p,z,k)$. Mixtures of this distribution can now be used to model new instruments as follows:

$$P(f|p,z,s) = \sum_k P(f|p,z,k)P(k|s) \quad (5)$$

where $P(k|s)$ represents a source-specific distribution over eigeninstruments. This model reduces the size of the parameter space for each source instrument in the mixture from FPZ, which is typically tens of thousands, to K which is typically between 10 and 100. Of course the quality of this parametrization depends on how well the eigeninstrument basis spans the true instrument parameter space, but assuming a sufficient

[2]Some care has to be taken to ensure that the bases in $\mathbf{\Omega}$ are properly normalized so that each section of F entries sums to 1, but so long as this requirement is met, any decomposition that yields non-negative basis vectors can be used.

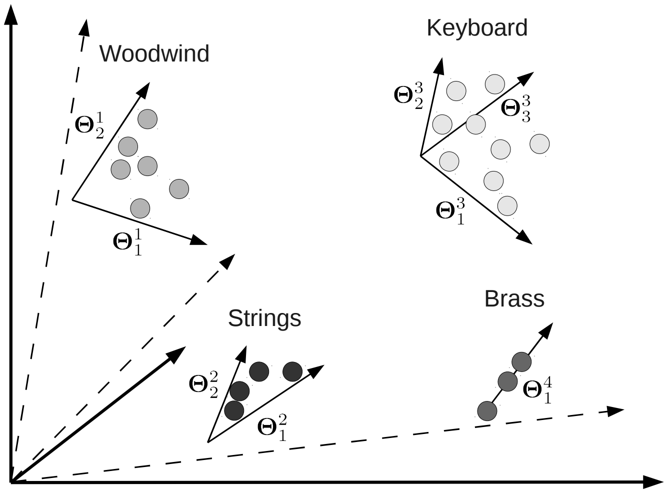


Fig. 3. Caricature of the mixture-of-subspaces model. The global instrument parameter space has several subspaces embedded in it. Each subspace corresponds to a different instrument type or family and has its own rank and set of basis vectors. Note that in practice the subspaces are conical regions extending from the global origin, but are shown here with offsets for visual clarity.

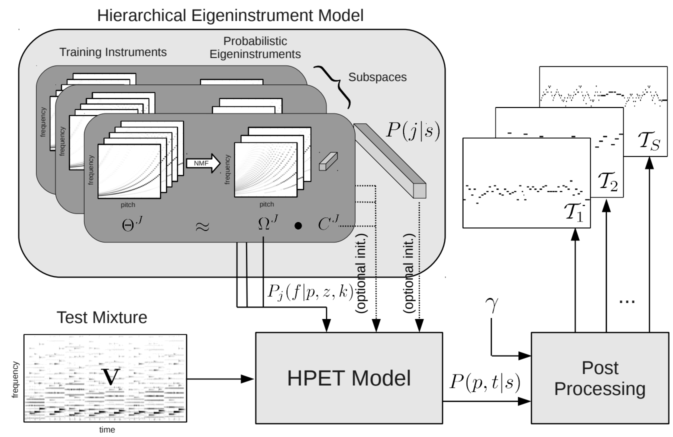


Fig. 4. Illustration of the *hierarchical probabilistic eigeninstrument transcription* (HPET) system. First, a set of training instruments is used to derive the set of eigeninstrument subspaces. A weighted combination of these subspaces are then used by the HPET model to learn the probability distribution $P(p,t|s)$, which is post-processed into source-specific binary transcriptions, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_S$.

variety of training instruments are used, we can expect good coverage. An overview of the eigeninstrument construction process is shown in Figure 2.

2) Hierarchical Eigeninstruments: Although we can expect that by training on a broad range of instrument types, the eigeninstrument space will be sufficiently expressive to represent new instruments, it is conceivable that the model may not be restrictive enough. Implicit in the model described in (5) is the assumption that the subspace defined by the training instruments can be accurately represented as a linear manifold. However, given the heterogeneity of the instruments involved, it is possible that they may actually lie on a nonlinear manifold, making (5) an insufficient model. The concern here is that the eigeninstrument bases could end up modeling regions of parameter space that are different enough from the true instrument subspace that they allow for models with poor discriminative properties.

One way to better model a non-linear subspace is to use a mixture of linear subspaces. This locally linear approximation is analogous to the *mixture of principal component analysers* model described by Hinton *et al.* [40], although we continue to enforce the non-negativity requirement in our model. Figure 3 illustrates the idea of locally linear subspaces embedded in a global space. The figure shows the positive orthant of a space corresponding to our global parameter space. In this example, we have four subspaces embedded in this parameter space, each defined by a different family of instruments. The dashed lines represent basis vectors that might have been found by the regular (non-hierarchical) eigeninstrument model. We can see that these bases define a conical region of space that includes far more than just the training points.

The extension from the PET instrument model to the mixture-of-instrument subspaces model is straightforward and we refer to the result as *hierarchical eigeninstruments*. Similar to before we use NMF to solve for the eigeninstruments, except now we have J training subsets with I_j instruments each. For each model $\mathcal{M}_i^j \in \mathcal{M}^j$, we reshape the parameters into a super-vector and then form the parameter matrix, $\mathbf{\Theta}^j$. Next, NMF with rank K_j is performed on the matrix, yielding $\mathbf{\Theta}^j \approx \mathbf{\Omega}^j \mathbf{C}^j$. Finally, each $\mathbf{\Theta}^j$ is reshaped into an eigeninstrument distribution, $P_j(f|p,z,k)$. To form new instruments, we now need to take a weighted combination of eigeninstruments for each subspace j as well as a weighted combination of the subspaces themselves:

$$P(f|p,z,s) = \sum_j P(j|s) \sum_k P_j(k|s) P_j(f|p,z,k) \tag{6}$$

In addition to an increase in modeling power as compared to the basic eigeninstrument model, the hierarchical model has the advantage of being able to incorporate prior knowledge in a targeted fashion by initializing or fixing the coefficients of a specific subspace, $P_j(k|s)$, or even the global subspace mixture coefficients, $P(j|s)$. This can be useful if, for example, each subspace corresponds to a particular instrument type (violin, piano, etc.) and we know the instrument types present in the target mixture. A more coarse-grained modeling choice might associate instrument families (brass, woodwind, etc.) with individual subspaces, in which case we would only have to know the family of each source in the mixture. In either case, the hierarchical eigeninstrument model affords us the ability to use the system with *a priori* information which is more likely to be available in real-world use cases than specific instrument models.

B. Transcription Model

We are now ready to present the full transcription model proposed in this paper, which we refer to as *hierarchical probabilistic eigeninstrument transcription* (HPET) and is illustrated in Figure 4. Combining the probabilistic model in (4) and the eigeninstrument model in (6), we arrive at the

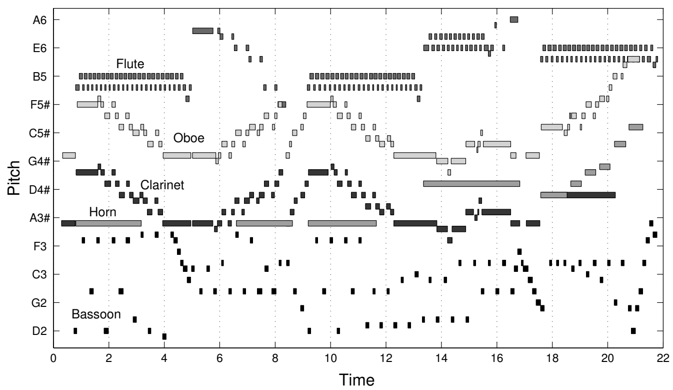


Fig. 5. Pianoroll of the complete 5-instrument mixture used in our experiments.

following:

$$P(f|t) = \sum_{s,p,z,k,j} P(j|s)P_j(f|p,z,k)P_j(k|s)P(z|s,p,t)P(s|p,t)P(p|t) \quad (7)$$

Once we have solved for the model parameters, we calculate the joint distribution over pitch and time conditional on source:

$$P(p,t|s) = \frac{P(s|p,t)P(p|t)P(t)}{\sum_{p,t} P(s|p,t)P(p|t)P(t)} \quad (8)$$

This distribution effectively represents the transcription of source s, but still needs to be post-processed to a binary pianoroll representation so that it can be compared with ground-truth data. Currently, this is done using a simple threshold γ (see Section III-D). We refer to the final pianoroll transcription of source s as $\mathcal{T}_s$.

We solve for the parameters in (7) using the *expectation-maximization* (EM) algorithm [3]. This involves iterating between two update steps until convergence (we find that $50 - 100$ iterations is almost always sufficient). In the first (expectation) step, we calculate the posterior distribution over the hidden variables s, p, z, and k, for each time-frequency point given the current estimates of the model parameters:

$$P(s,p,z,k,j|f,t) = \frac{P(j|s)P_j(f|p,z,k)P_j(k|s)P(z|s,p,t)P(s|p,t)P(p|t)}{P(f|t)} \quad (9)$$

In the second (maximization) step, we use this posterior to increase the expected log-likelihood of the model given the data:

$$\mathcal{L} \propto \sum_{f,t} V_{f,t} \log\left(P(t)P(f|t)\right) \quad (10)$$

where $V_{f,t}$ are values from our original magnitude spectrogram, $\mathbf{V}$. This results in the following update equations:

$$P(j|s) = \frac{\sum_{f,t,p,z,k} P(s,p,z,k,j|f,t)V_{f,t}}{\sum_{f,t,p,z,k,j} P(s,p,z,k,j|f,t)V_{f,t}} \quad (11)$$

$$P_j(k|s) = \frac{\sum_{f,t,p,z} P(s,p,z,k,j|f,t)V_{f,t}}{\sum_{f,t,p,z,k} P(s,p,z,k,j|f,t)V_{f,t}} \quad (12)$$

$$P(z|s,p,t) = \frac{\sum_{f,k,j} P(s,p,z,k,j|f,t)V_{f,t}}{\sum_{f,k,j,z} P(s,p,z,k,j|f,t)V_{f,t}} \quad (13)$$

$$P(s|p,t) = \frac{\sum_{f,z,k,j} P(s,p,z,k,j|f,t)V_{f,t}}{\sum_{f,z,k,j,s} P(s,p,z,k,j|f,t)V_{f,t}} \quad (14)$$

$$P(p|t) = \frac{\sum_{f,s,z,k,j} P(s,p,z,k,j|f,t)V_{f,t}}{\sum_{f,s,z,k,j,p} P(s,p,z,k,j|f,t)V_{f,t}} \quad (15)$$

III. Evaluation

A. Data

The data set used in our experiments was formed from part of the development woodwind data set used in the MIREX Multiple Fundamental Frequency Estimation and Tracking evaluation task.[3] The first 22 seconds from the bassoon, clarinet, oboe, flute, and horn tracks were manually transcribed.[4] These instrument tracks were then combined (by simply adding the individual tracks) to produce all possible 2-instrument, 3-instrument, 4-instrument, and 5-instrument mixtures and then down-sampled to 8kHz.

In addition to the data set of recorded performances, we also produced a set of synthesized versions of the mixtures described above. To produce the synthetic tracks, the MIDI versions were rendered at an 8kHz sampling rate using *timidity* [5] and the SGM V2.01[6] soundfont. Reverberation and other effects were not used.

For both the real and synthesized mixtures, the audio was transformed into a magnitude spectrogram. This was done by taking a 1024-point short-time Fourier transform (STFT) with 96ms (Hamming) window and 24ms hop and retaining only the magnitude information. The specific properties of the data set are given in Table I. Note that these numbers summarize the recorded and synthesized data sets separately and therefore are effectively doubled when both sets are considered.

TABLE I
Summary of the properties of our data set.

	# Mixtures	# Notes	# Frames
2-instrument	10	4044	22470
3-instrument	10	6066	22470
4-instrument	5	4044	11235
5-instrument	1	1011	2247

It is also important to emphasize that this data is taken from the MIREX *development* set and that the primary test data is not publicly available. In addition, most authors of other transcription systems do not report results on the development data, making comparisons difficult. We do, however, include a comparison to the multi-instrument transcription system proposed by Duan *et al.* [34] in our experiments.

[3]http://www.music-ir.org/mirex/2009/index.php/Multiple_Fundamental_Frequency_Estimation_&_Tracking
[4]These transcriptions are available from the corresponding author.
[5]http://timidity.sourceforge.net
[6]http://www.geocities.jp/shansoundfont/

TABLE II
INSTRUMENTS USED TO BUILD THE (HIERARCHICAL) EIGENINSTRUMENTS MODEL IN OUR EXPERIMENTS.

Group (J)	Rank (K_j)	Instruments
Keyboard	10	(5) Pianos
Guitar	12	(6) Guitars
Bass	8	(4) Basses
Viol	8	Violin, Viola, Cello, Contrabass
Brass	18	Trumpet, Trombone, Tuba, (2) Horns, (4) Saxophones
Reed	6	Oboe, Bassoon, Clarinet
Pipe	6	Piccolo, Flute, Recorder

B. Instrument Models

We used a set of thirty-four instruments of varying types to derive our instrument model. The instruments were divided up into seven roughly equal-sized groups (*i.e.* $J = 7$) of related instruments which formed the upper layer in the hierarchical eigeninstruments model. The group names and breakdown of specific instruments are given in Table II.

The instrument models were generated with *timidity*, but in order to keep the tests with synthesized audio as fair as possible, two different soundfonts (Papelmedia Final SF2 XXL [7] and Fluid R3 [8]) were used. We generated separate instances of each instrument type using each of the soundfonts at three different velocities (40, 80, and 100), which yielded 204 instrument models in total.

Each instrument model $\mathcal{M}_i^j$ consisted of $P = 58$ pitches (C2-A6#) which were built as follows: for each pitch p, a note of duration 1s was synthesized at an 8kHz sampling rate. An STFT using a 1024-point (Hamming) window was taken and the magnitude spectra were kept. These spectra were then normalized so that the frequency components summed to 1 (*i.e.* each spectrogram column sums to 1). Next, NMF with rank Z (the desired number of components per pitch) was run on the normalized magnitude spectrogram and the resulting basis vectors were used as the components for pitch p of model $\mathcal{M}_i^j$. Note that because unsupervised NMF yields arbitrarily ordered basis vectors, this method does not guarantee that the Z components of each pitch will correspond temporally across models. We have found that initializing the activation matrix used in each of these per-pitch NMFs to a consistent form (such as one with a heavy main diagonal structure) helps to remedy this problem.

Another potential issue has to do with the differences in the natural playing ranges of the instruments. For example, a violin generally cannot play below G3, although the model described thus far would include notes below this. Therefore, we masked out (*i.e.* set to 0) all FZ parameters of each note outside the playing range of each instrument used in training. There are other possibilities for handling these ill-defined pitch values as well. We could, for example, simply leave them in place or we could set each vector of F frequency bins to an uninformative uniform distribution. A fourth possibility is to treat the entries as missing data and modify our EM algorithm to impute their maximum likelihood values at each iteration, similar to what others have done for NMF [41]. We experimented with all of these techniques, but found that simply setting the parameters of the out-of-range pitch values to 0 worked best.

[7] http://www.papelmedia.de/english/index.htm

[8] http://soundfonts.homemusician.net/collections_soundfonts/fluid_release_3.html

Next, as described in Section II-A, the instrument models were stacked into super-vector form and NMF was used to find the instrument bases which were then reshaped into the eigeninstruments. For the HPET system, we used different ranks (values of K_j) for each group of instruments because of the different sizes of the groups. The specific values used for the ranks are given in Table II, although it is worth noting that preliminary experiments did not show a substantial difference in performance for larger values of K_j. The NMF stage resulted in a set of instrument bases, $\mathbf{\Omega}^j$ for each group j which were then reshaped into the eigeninstrument distribution for group j, $P_j(f|p, z, k)$. For the non-hierarchical PET system, we simply combined all instruments into a single group and used a rank equal to the sum of the ranks above ($K = 68$). Similar to before, the resulting instrument bases were then converted to an eigeninstrument distribution.

Note that in preliminary experiments, we did not find a significant advantage to values of $Z > 1$ and so the full set of experiments presented below was carried out with only a single component per pitch.

C. Algorithms

We evaluated several variations of our algorithm so as to explore the hierarchical eigeninstruments model as well as the effects of parameter initialization. In all cases where parameters were initialized randomly, their values were drawn from a uniform distribution.

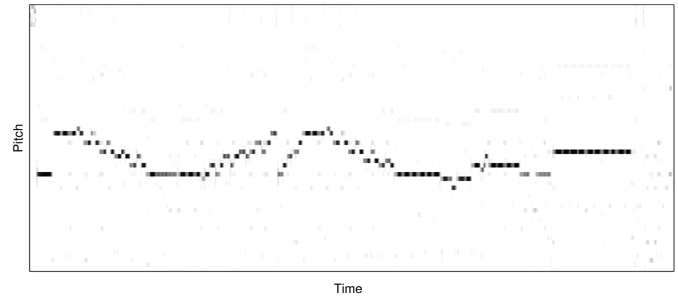
1) HPET: totally random parameter initialization
2) $\text{HPET}_{\text{group}}$: $P(j|s)$ initialized to the correct value
3) $\text{HPET}_{\text{model}}$: $P(j|s)$ and $P_j(k|s)$ initialized to an instrument of the same type from the training set

The first variant corresponds to totally blind transcription where the system is given no prior knowledge about the target mixture other than the number of sources. The second variant corresponds to providing the system with the group membership of the sources in the mixture (*i.e.* setting $P(j|s) = 1$ when s belongs to instrument group j and 0 otherwise). The third variant is akin to furnishing the system with knowledge of the correct groups as well as an approximate setting for the eigeninstrument distribution in that group (*i.e.* setting $P_j(k|s) = 1$ when s is of instrument type k in group j and setting $P_j(k|s) = 0$ otherwise). It is important to note that in this third case we determine these eigeninstrument settings using an instrument of the correct type, but whose parameters come from the training set, $\mathcal{M}$. This case is meant to correspond to knowledge of the specific instrument type, not the exact instrument model used to produce the test mixture.

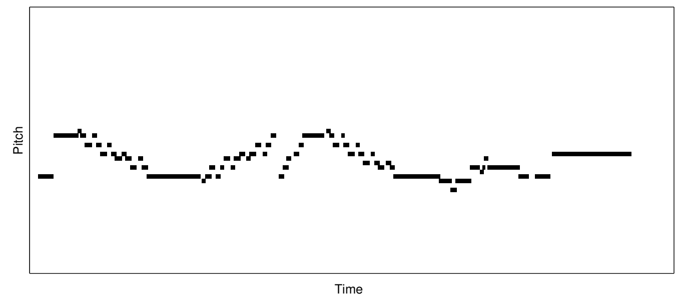
Both of the informed variants of the HPET system are only *initialized* with the settings that they receive. Intuitively, we are trying to start the models in the correct "neighborhood" of parameter space in the hope that they can further optimize these settings. We have experimented with other variations where the parameters are fixed to these values, but the results are not significantly different. Figure 6 shows an example of the raw output distribution, $P(p,t,|s)$, as generated by $\text{HPET}_{\text{model}}$. Ground truth values for the synthesized bassoon-clarinet mixture are shown as well. Although the hierarchical extension to the PET system has the advantage of providing a means by which to include prior knowledge, we were also interested in testing whether the increased subspace modeling power would have a beneficial effect. To this end, we include the original (non-hierarchical) PET algorithm in our experiments as well.

As mentioned earlier, the paucity of transcription systems capable of instrument-specific note assignment makes external comparisons difficult. We are grateful to Duan *et al.* for providing us with the source code for their multi-pitch tracking system [34] which we refer to as MPT. We used the parameter settings recommended by the authors. As with the HPET systems, we provide the MPT algorithm with the number of instrument sources in each mixture and with the minimum and maximum pitch values to consider. As part of the multi-pitch estimation front-end in MPT, the algorithm needs to know the maximum polyphony to consider in each frame. It is difficult to set this parameter fairly since our approach has no such parameter (technically it is P, the cardinality of the entire pitch range). Following the setting used for the MIREX evaluation, we set this parameter to 6 which is the upper-bound of the maximum polyphony that occurs in the data set. The output of the MPT algorithm consists of the F0 values for each instrument source in each frame. We rounded these values to the nearest semitone.

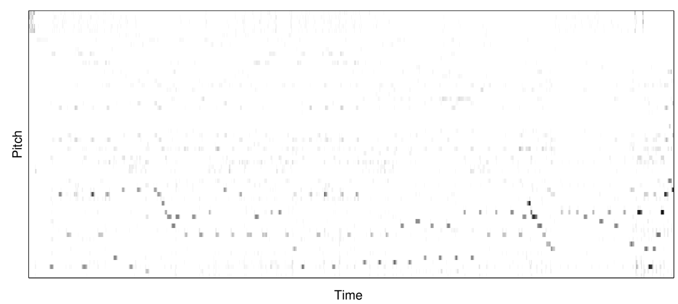
Finally, as a baseline comparison, we include a generic NMF-based transcription (with generalized KL divergence as a cost function) system. This extremely simple system had all of its instrument models (sub-matrices of $\mathbf{W}$) initialized with a generic instrument model which we defined as the average of the instrument models in the training set.



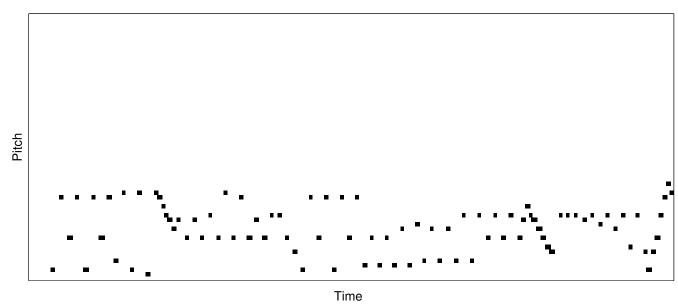
(a) Clarinet (HPET)



(b) Clarinet (ground-truth)



(c) Bassoon (HPET)



(d) Bassoon (ground-truth)

Fig. 6. Example HPET (with model initialization) output distribution $P(p,t|s)$ and ground-truth data for the synthesized bassoon-clarinet mixture.

D. Metrics

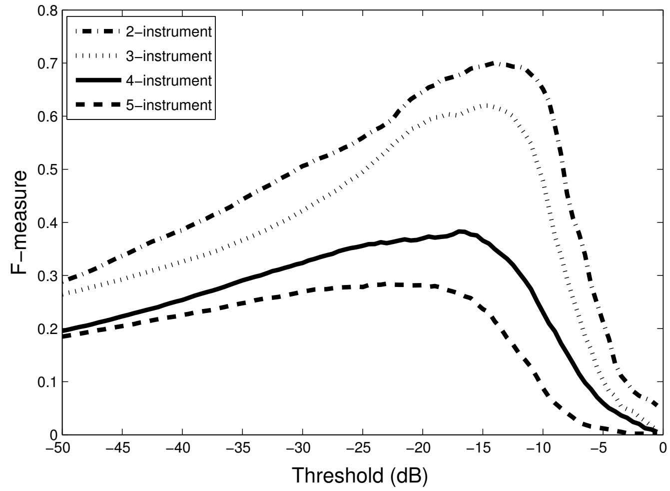
We evaluated our method using a number of metrics on both the frame and note levels. In the interest of clarity, we distilled these numbers down to F-measure [42] (the harmonic mean of precision and recall) on both the frame and note levels as well as the *mean overlap ratio* (MOR). When computing the note-level metrics, we consider a note onset to be correct if it falls within +/- 48ms of the ground truth onset. This is only slightly more restrictive than the standard tolerance (+/- 50ms) used by the MIREX community. Because of the difficulty in generating an accurate ground-truth for note offsets (many notes decay and therefore have ambiguous end times), we opted to evaluate this aspect of system performance via the MOR which is defined as follows. For each correctly detected note onset, we compute the overlap ratio as defined in [43]:

$$\text{overlap ratio} = \frac{\min\{t_a^{\text{off}}, t_t^{\text{off}}\} - \max\{t_a^{\text{on}}, t_t^{\text{on}}\}}{\max\{t_a^{\text{off}}, t_t^{\text{off}}\} - \min\{t_a^{\text{on}}, t_t^{\text{on}}\}} \tag{16}$$

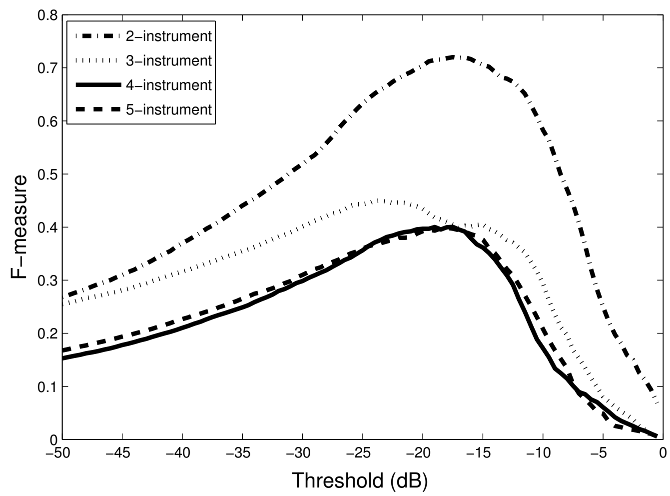
where, for each note under consideration, t_a^{on} is the onset time according to the algorithm, t_t^{on} is the ground-truth onset time, and $t_a^{\text{off}}, t_t^{\text{off}}$ are the offset times from algorithm and ground-truth, respectively. The overlap ratio is computed for all correctly detected notes and the mean is taken to give the MOR.

Note that, because the order of the sources in $P(p,t|s)$ is arbitrary, we compute sets of metrics for all possible permutations and report the set with the best frame-level F-measure.

Recall that the output of our system is a joint distribution over pitch and time (conditioned on source) and therefore must



(a) Synthetic mixtures



(b) Real mixtures

Fig. 7. Comparison of the sensitivity of the HPET algorithm at a range of threshold values for γ. Results are averaged over mixtures consisting of the same number of instruments.

be discretized before the evaluation metrics can be computed. This is done by comparing each entry of $P(p,t|s)$ to a threshold parameter, γ, resulting in a binary pianoroll $\mathcal{T}_s$:

$$\mathcal{T}_s = \begin{cases} 1 & \text{if } P(p,t|s) > \gamma \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

The threshold γ used to convert $P(p,t|s)$ to a binary pianoroll was determined empirically for each algorithm variant and each mixture. This was done by computing the threshold that maximized the area under the *receiver operating characteristic* (ROC) [44] curve for that mixture, taking source assignment into account (*i.e.* pitch, time, and source must match in order to be counted as a true positive). Although this method of parameter determination is somewhat post-hoc, the algorithm is fairly robust to the choice of γ as shown in Figure 7.

As with many latent variable models, our system can be sensitive to initial parameter values. In order to ameliorate the effects of random initialization, we run each algorithm three times on each test mixture. Evaluation metrics are computed for each algorithm, mixture, and repetition and then averaged over mixtures and repetitions to get the final scores reported in Tables III-VI.

E. Experiments

We conducted two primary experiments in this work. The first, and most important, was the comparison of the six algorithms (three HPET variants, PET, MPT, and NMF) for multi-instrument transcription. In this experimental setting we are interested in evaluating not only an algorithm's ability to detect notes correctly, but also to assign these notes to their source instruments. Therefore a pitch is only considered correct if it occurs at the correct time and is assigned to the proper instrument source. We refer to this as the *transcription with source assignment* task.

It is, however, also interesting to also consider the efficacy of each algorithm for the simpler source-agnostic transcription task as this problem has been the focus of most transcription research in recent years. We refer to this task as *transcription without source assignment*. For concision, only the average frame-level F-measures for this case are included.

The results of our experiments are summarized in Tables III-VI. As we would expect, the baseline NMF system performs the worst in all test cases – not surprising given the limited information and lack of constraints. Also unsurprising is the general downward trend in performance in all categories as the number of instruments in the mixture increases.

In terms of the frame-level results for the case with source assignment (Table III), we can see that the HPET algorithm benefited substantially from good initializations. With the exception of the outlier in the case of the real 5-instrument mixture, HPET with full model initialization performed substantially better than other systems. HPET with initialization by group performs slightly worse, although in some cases the results are very close. Interestingly, we also find that HPET does not always outperform PET, although again, the numbers are often very close. This suggests that the true instrument space may be relatively well approximated by a linear subspace. The comparison between HPET, PET, and MPT is also interesting, as these systems all make use of roughly the same amount of prior knowledge. For mixtures containing fewer source instruments, the eigeninstrument-based systems slightly out-perform MPT, although performance is essentially the same for 4-instrument mixtures and MPT does better on synthesized 5-instrument mixtures.

Turning to the note-level onset-detection metric (Table V), we find a similar trend as at the frame-level. The initialized models typically outperform all other systems by a reasonable margin, with full model initialization leading to slightly better performance than group-only initialization. The numbers for all systems were generally down for this task as compared to the frame-level analysis. MPT in particular did not perform nearly as well as it had on frame-level detection. However, the MPT numbers appear to be roughly consistent with the MIREX 2010 note-level results which suggests that MPT had difficulty with the characteristics of the woodwind data set.

TABLE III
AVERAGE FRAME-LEVEL F-MEASURES (WITH SOURCE ASSIGNMENT).

	Synthesized				Real			
	2-inst	3-inst	4-inst	5-inst	2-inst	3-inst	4-inst	5-inst
HPET	0.50	0.44	0.36	0.27	0.52	0.43	0.37	**0.40**
HPET$_{\text{group}}$	0.65	0.51	0.42	0.38	0.62	0.48	0.42	0.35
HPET$_{\text{model}}$	**0.67**	**0.54**	**0.47**	**0.42**	**0.63**	**0.50**	**0.43**	0.33
PET [16]	0.53	0.40	0.33	0.28	0.53	0.42	0.36	0.35
MPT [34]	0.49	0.40	0.36	0.35	0.49	0.40	0.36	0.35
NMF	0.34	0.28	0.23	0.19	0.28	0.24	0.20	0.18

TABLE IV
AVERAGE FRAME-LEVEL F-MEASURES (WITHOUT SOURCE ASSIGNMENT).

	Synthesized				Real			
	2-inst	3-inst	4-inst	5-inst	2-inst	3-inst	4-inst	5-inst
HPET	0.76	0.73	0.71	0.69	0.62	0.67	0.65	0.65
HPET$_{\text{group}}$	**0.83**	0.79	0.75	**0.73**	0.77	0.72	**0.70**	**0.67**
HPET$_{\text{model}}$	**0.83**	**0.80**	**0.77**	**0.73**	**0.78**	**0.73**	**0.70**	**0.67**
PET [16]	0.76	0.71	0.68	0.65	0.70	0.67	0.67	0.66
MPT [34]	0.64	0.64	0.62	0.60	0.64	0.63	0.62	0.60
NMF	0.59	0.62	0.62	0.60	0.48	0.52	0.52	0.53

TABLE V
AVERAGE NOTE-LEVEL F-MEASURES (WITH SOURCE ASSIGNMENT).

	Synthesized				Real			
	2-inst	3-inst	4-inst	5-inst	2-inst	3-inst	4-inst	5-inst
HPET	0.47	0.37	0.28	0.21	0.45	0.37	0.31	**0.36**
HPET$_{\text{group}}$	**0.61**	0.46	0.35	0.31	0.60	0.44	0.37	0.31
HPET$_{\text{model}}$	0.49	**0.54**	**0.39**	**0.37**	**0.62**	**0.47**	**0.39**	0.22
PET [16]	0.45	0.32	0.26	0.22	0.45	0.36	0.30	0.26
MPT [34]	0.21	0.14	0.10	0.10	0.19	0.14	0.10	0.10
NMF	0.27	0.22	0.17	0.11	0.23	0.19	0.15	0.11

TABLE VI
AVERAGE MEAN OVERLAP RATIOS (WITH SOURCE ASSIGNMENT).

	Synthesized				Real			
	2-inst	3-inst	4-inst	5-inst	2-inst	3-inst	4-inst	5-inst
HPET	0.49	0.47	0.45	0.43	0.49	0.47	0.43	0.43
HPET$_{\text{group}}$	0.54	0.46	0.46	0.42	0.52	0.49	0.48	0.47
HPET$_{\text{model}}$	0.54	0.51	0.49	0.47	0.52	0.48	0.48	**0.53**
PET [16]	0.51	0.46	0.43	0.37	0.47	0.45	0.42	0.40
MPT [34]	**0.58**	**0.54**	**0.55**	**0.51**	**0.59**	**0.54**	**0.55**	0.51
NMF	0.38	0.38	0.35	0.35	0.38	0.37	0.37	0.35

MPT did, however, do best in terms of MOR (Table VI) in almost all categories, although results for the fully initialized HPET variant were slightly better for the real 5-instrument case.

Next, we consider transcription without source assignment (Table IV) which corresponds to the polyphonic transcription task that has been most thoroughly explored in the literature. Again, the initialized models perform substantially better than the others. Here we see the greatest disparity between synthesized and recorded mixtures (at least for the eigeninstrument-based systems) in all of the experiments. An examination of the test data suggests that this may be largely due to a tuning mismatch between the recorded audio and synthesized training data.

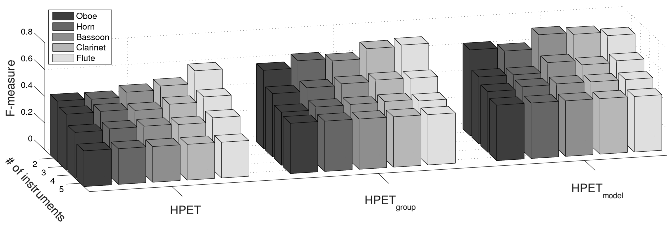
Finally, we discuss the differences in performance between the HPET variants based on the instruments in the mixture. Figure 8 shows this breakdown. For each algorithm and instrument, the figure shows the F-measure averaged over only the mixtures containing that instrument. We can see that, in almost all cases, the flute appears to have been the easiest instrument to transcribe, and the oboe the most difficult. This trend seems to have held for both synthetic as well as real mixtures, although the blind HPET variant had more trouble with real mixtures containing flute. Referring to Figure 5, we see that the flute part occupies a largely isolated pitch range. Given the limited number of harmonics present in notes at this range, it seems likely that pitch was the primary source of discriminative information for the flute part. The oboe part, however, occurs not only roughly in the middle of the modeled pitch range, but also almost entirely mirrors the clarinet part. It is therefore not surprising that mixtures containing oboe are difficult. The same line of reasoning, however, would lead us to expect that the mixtures containing clarinet would be equally difficult given the similarities between the two instrument parts. Interestingly, this does not appear to be the case as performance for mixtures containing clarinet are reasonably good overall. One possible explanation is that the clarinet model is relatively dissimilar to others in eigeninstrument space and therefore easy to pick out. This makes sense considering that the harmonic structure of the clarinet's timbre contains almost exclusively odd harmonics (for the relevant pitch range).

IV. CONCLUSIONS

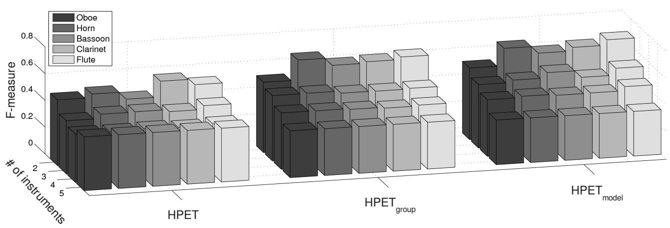
We have presented a hierarchical probabilistic model for the challenging problem of multi-instrument polyphonic transcription. Our approach makes use of two sources of information available from a set of training instruments. First, the spectral characteristics of the training instruments are used to form what we call "eigeninstruments". These distributions over frequency represent basis vectors that define instrument parameter subspaces specific to particular groups of instruments. Second, the natural organization of instruments into families or groups is exploited to partition the parameter space into a set of separate subspaces. Together, these two distributions constrain the solutions of new models which are fit directly to the target mixture.

We have shown that this approach can perform well in the blind transcription setting where no knowledge other than the number of instruments is assumed. For many of the metrics and mixture complexities considered, our approach performs as well or better than other multi-instrument transcription approaches. We have also shown that by assuming fairly general prior knowledge about the sources in the target mixture, we can significantly increase the performance of our approach.

There are several areas in which the current system could be improved and extended. First, the thresholding technique that



(a) Synthetic mixtures



(b) Real mixtures

Fig. 8. Per-instrument average frame-level F-measures (with source assignment) by algorithm and number of sources for (a) synthesized data and (b) real data.

we have used is extremely simple and results could probably be improved substantially through the use of pitch dependent thresholding or more sophisticated classification. Second, and perhaps most importantly, although early experiments did not show a benefit to using multiple components for each pitch, it seems likely that the pitch models could be enriched substantially. Many instruments have complex time-varying structures within each note that would seem to be important for recognition. We are currently exploring ways to incorporate this type of information into our system.

Acknowledgment

The authors would like to thank Zhiyao Duan for providing us with the source code for his transcription algorithm. We are also grateful for the helpful comments provided by the reviewers.

References

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[4] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[5] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[6] J. Eggert and E. Körner, "Sparse coding and NMF," in *IEEE International Joint Conference on Neural Networks*, vol. 4, 2004, pp. 2529–2533.

[7] M. Shashanka, B. Raj, and P. Smaragdis, "Probabilistic latent variable models as non-negative factorizations," *Computational Intelligence and Neuroscience*, vol. 2008, 2008.

[8] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[9] T. Virtanen, A. T. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorization for audio signal modeling," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 1825–1828.

[10] C. Févotte, N. Bertin, and J. L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[11] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *International Conference on Spoken Language Processing*, 2006.

[12] P. Smaragdis, M. Shashanka, and B. Raj, "A sparse non-parametric approach for single channel separation of known sounds," in *Advances in Neural Information Processing Systems*, 2009, pp. 1705–1713.

[13] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, pp. 177–180.

[14] E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic non-negative matrix factorization for polyphonic pitch transcription," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 109–112.

[15] G. Grindlay and D. P. W. Ellis, "Multi-voice polyphonic music transcription using eigeninstruments," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 53–56.

[16] ——, "A probabilistic subspace model for polyphonic music transcription," in *International Conference on Music Information Retrieval*, 2010, pp. 21–26.

[17] V. Y. F. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization," in *Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[18] M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *European Signal Processing Conference*, 2005.

[19] K. Murao, M. Nakano, Y. Kitano, N. Ono, and S. Sagayama, "Monophonic instrument sound segregation by clustering NMF components based on basis similarity and gain disjointness," in *International Society on Music Information Retrieval Conference*, 2010, pp. 375–380.

[20] T. Virtanen and A. Klapuri, "Analysis of polyphonic audio using source-filter model and non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2006.

[21] T. Heittola, A. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," in *International Conference on Music Information Retrieval*, 2009, pp. 327–332.

[22] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[23] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010.

[24] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 1917, pp. 1917–1930, 2002.

[25] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, November 2003.

[26] S. A. Abdallah and M. D. Plumbley, "Polyphonic music transcription by non-negative sparse coding of power spectra," in *International Conference on Music Information Retrieval*, 2004, pp. 318–325.

[27] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[28] G. Poliner and D. P. W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, 2007.

[29] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Organization of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism," in *International Joint Conference on Artificial Intelligence*, 1995, pp. 158–164.

[30] E. Vincent and X. Rodet, "Music transcription with ISA and HMM," in *International Symposium on Independent Component Analysis and Blind Signal Separation*, 2004, pp. 1197–1204.

[31] P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Transactions*

on Audio, Speech, and Language Processing, vol. 16, no. 1, pp. 116–128, 2008.

[32] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 982–994, 2007.

[33] W. C. Chang, W. Y. Su, C. Yeh, A. Roebel, and X. Rodet, "Multiple-F0 tracking based on a high-order hmm model," in *International Conference on Digital Audio Effects*, 2008, pp. 379–386.

[34] Z. Duan, J. Han, and B. Pardo, "Harmonically informed multi-pitch tracking," in *International Society on Music Information Retrieval Conference*, 2009, pp. 333–338.

[35] ——, "Song-level multi-pitch tracking by heavily constrained clustering," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 57–60.

[36] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implications," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, pp. 601–602.

[37] T. Hofmann, "Probabilistic latent semantic analysis," in *Conference on Uncertainty in Artificial Intelligence*, 1999, pp. 289–296.

[38] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker identification in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, November 2000.

[39] R. J. Weiss and D. P. W. Ellis, "Speech separation using speaker-adapted eigenvoice speech models," *Computer Speech and Language*, vol. 24, no. 1, pp. 16–29, 2010.

[40] G. E. Hinton, P. Dayan, and M. Revow, "Modelling the manifolds of images and handwritten digits," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 65–74, 1997.

[41] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *SIAM International Conference on Data Mining*, 2006, pp. 549–553.

[42] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed. London: Butterworths, 1979.

[43] M. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 319–322.

[44] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 2006, pp. 861–874, 2005.

Graham Grindlay (S'06) received the B.A. in computer science from Oberlin College, Oberlin, OH in 2000, the M.S. in computer science from the University of California, Santa Cruz, CA in 2005, and the M.S. in media arts and sciences from the Massachusetts Institute of Technology (MIT), Cambridge, MA in 2007.

He is currently working towards the Ph.D. degree in electrical engineering at Columbia University, New York, NY. He is primarily interested in the development of novel machine learning and signal processing techniques for machine listening with a focus on music applications.

Daniel P. W. Ellis (M'96-SM'04) received the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge.

He is an Associate Professor in the Electrical Engineering Department, Columbia University, New York. His Laboratory for Recognition and Organization of Speech and Audio (LabROSA) is concerned with all aspects of extracting high-level information from audio, including speech recognition, music description, and environmental sound processing. He also runs the AUDITORY e-mail list of 1700 worldwide researchers in perception and cognition of sound. He worked at MIT, where he was a Research Assistant in the Media Lab, and he spent several years as a Research Scientist at the International Computer Science Institute, Berkeley, CA.