Submitted for Publication in Cybernetics and Systems. Prepared with the T&F Journal Template.

Automatic music transcription and audio source separation

M D Plumbley, S A Abdallah,
Department of Electronic Engineering, King's College London, Strand, London WC2R 2LS, UK

J P Bello, M E Davies, G Monti and M B Sandler
Department of Electronic Engineering, Queen Mary, University of London, Mile End Road, London E1 4NS, UK

[UNTIL 31 December 2001 USE THE FOLLOWING:]

Address Correspondence to: Dr M D Plumbley, Department of Electronic Engineering, King's College London, Strand, London WC2R 2LS, UK. Email: mark.plumbley@kcl.ac.uk

[AFTER 1 January 2002 USE THE FOLLOWING:]

Address Correspondence to: Dr M D Plumbley, Department of Electronic Engineering, Queen Mary, University of London, Mile End Road, London E1 4NS, UK. Email: mdp@ieee.org

In this article, we give an overview of a range of approaches to the analysis and separation of musical audio. In particular, we consider the problems of automatic music transcription and audio source separation, which are of particular interest to our group. Monophonic music transcription, where a single note is present at one time, can be tackled using an autocorrelation-based method. For polyphonic music transcription, with several notes at any time, other approaches can be used, such as a blackboard model or a multiple-cause/sparse coding method. The latter is based on ideas and methods related to independent component analysis (ICA), a method for sound source separation.

Keywords: Automatic music transcription, blind source separation, ICA, sparse coding, computational auditory scene analysis

Over the last decade or so, and particularly since the publication of Bregman's seminal book on Auditory Scene Analysis (Bregmann 1990), there has been an increasing interest in the problem of Computational Auditory Scene Analysis (CASA): how to design computer-based models that can analyze an auditory scene. Imagine you are standing in a busy street among a crowd of people. You can hear traffic noise, footsteps of people nearby, the bleeping of a pedestrian crossing, your mobile phone ringing, and colleagues behind you having a conversation. Despite all these different sound sources, you have a pretty good idea of what is going on around you. It is more than just a mess of overlapping noise, and if you try hard you can concentrate on one of these sources if it is important to you (such as the conversation behind you).

This has proved to be a very difficult problem. It requires both separation of many sound sources, and analysis of the content of these sources. However, a few authors have begun to tackle this problem in recent years, with some success (see e.g. Ellis 1996).

One particular aspect of auditory scene analysis of interest to our group is *automatic music transcription*. Here, the sound sources are one or more instruments playing a piece of music, and we wish to analyze this to identify the instruments that are playing, and when and for how long each note is played. From this analysis we should then be able to produce a written musical score that shows notes and the duration of each on a written conventional music notation (for conventional western music, at least). In principle, this musical score could then be used to recreate the musical piece that was played.

We are also interested in the ability to separate sound sources based on their different locations in an auditory scene. This capability, known as *blind source separation* (Bell & Sejnowski 1995), can be useful on its own if we wish to eliminate some sound sources and concentrate on others. It also offers the potential to be combined with automatic music transcription systems in the future, to improve transcription performance based on the location of instruments as well as the different sounds they make.

Automatic music transcription

As we mentioned above, the aim of automatic music transcription is to analyze an audio signal to discover the instruments and notes that are playing, and so be able to produce a written transcription of the piece of music. Some pieces of music are *monophonic*, having only one instrument playing only one note at a time, such as a trumpet solo. On the other hand, most piece of music are *polyphonic*, having one or more instruments playing several notes at once, such as a piece for piano or full orchestra. (The use of the term *monophonic* here should be distinguished from a *mono* - as opposed to *stereo* – recording, which may be polyphonic music, but has been recorded with just a single audio channel.)

While the general auditory scene analysis is something we would expect most human listeners to have reasonable success at, this is not the case for the automatic music transcription problem. Most people, even musicians, do not have the ability to name the pitch of a note heard in isolation (so-called *perfect pitch*). Furthermore, only after special training are people able to perform musical transcription, and then they typically need to listen to the piece repeatedly, writing out one musical line (*voice*) at a time. Even the best human music transcribers are typically limited to pieces with less than 5 notes at once. For general listeners, the identity of individual notes does not seem to be important: rather it is the overall musical quality conveyed by the combinations of notes (*chords*). The composer may also deliberately be using the features and limitations of human hearing to create special effects, such as a single instrument alternating between high and low notes to give the perception of two separate musical lines or *streams*.

Thus the automatic music transcription problem is somewhat artificial: it is not something that we would expect an average human to be able to do easily. Nevertheless, there is much that we can learn from the human auditory system, and some of the approaches that have been used are based on or inspired either by what is know about its operation, or what is known about its ability to learn. We shall draw a distinction between two types of approach in the descriptions that follow: *knowledge-based* models and *learning* models.

In the knowledge-based approach, we try to use information we have about the physics of the musical generation process and about the human auditory system in order to build our analysis model. For monophonic music transcription, for example we can use our knowledge of the physics of vibrating objects such as the repetition of near-identical sound pressure waveforms in the auditory signals. For polyphonic music transcription, we might use time-frequency analysis techniques inspired by the cochlea in the human ear, followed by a knowledge integration system allowing us to piece together partial evidence into our proposed transcription.

In the learning approach, we may still use knowledge about the music generation process (sometimes called *prior knowledge*), but we also have adjustable parameters in our model which we allow to adapt (or *learn*) on the basis of the data that we are given. This approach is therefore sometimes called a *data-driven* approach, since we can be considered to be extracting knowledge from the audio signal (*data*) that is input to the model. For polyphonic music transcription, we might use this approach to learn the 'shapes' (frequency content) of the various notes and instruments, at the same time as the 'amounts' (volumes) of each note at each time.

Monophonic Transcription

While monophonic (single-note) music transcription, sometimes called *pitch tracking*, might seem like just a special case of polyphonic (many-note) music transcription, it is an important case that is worth treating separately. From the practical point of view, the knowledge that only one note is being played at any one time can help make the transcription process both simpler and more robust. In addition, there are many potential applications for monophonic transcription, from recognizing a hummed tune to helping a trumpet player to write down the notes they have just played.

One method that has been used for monophonic transcription is *autocorrelation* (Brown & Zhang 1991). Our approach is based on an initial autocorrelation analysis, together with detection of note onsets and approximation to the nearest MIDI note (Figure 1).

<Insert Figure 1 here>

The sampled audio signal is initially divided up into frames of *N* samples (in our case *N*=4096) and the autocorrelation function can be calculated as:

$$r_{xx}(n) = \frac{1}{N} \sum_{t=0}^{N-t-1} x(t)x(t+n)$$

In practice we calculate a similar function more efficiently via the fast Fourier transform (FFT). The autocorrelation measures the similarity between shifted version of the time waveform. The delay corresponding to the highest peak in the autocorrelation gives the period of the waveform, thus giving us the pitch of the signal (Figure 2). For higher frequencies, where the duration is only a few samples, and many waveforms feature in each frame, more accurate results can be obtained by performing a frequency transform of the frame (Bello, Monti & Sandler 2000).

<Insert Figure 2 here>

A complete transcription system needs more than just a running estimate of the current pitch: this needs to be converted into a representation of a succession of notes and durations. To this end, we use an onset detector to determine the start of new notes. The onset detector filters the audio signal to emphasize its high frequency content (which is normally present in the transients at the start of a new note): relative differences in this energy produces a sharp peak at note onset transients. Together with the pitch information and overall envelope (volume) this is fed into a *collector*. The collector uses prior knowledge about musical notes to produce a sequence of notes and durations. For example, notes must have a minimum duration of 100ms: apparent notes shorter than this will be smoothed out using a memory inside the collector. Since the system is to produce a MIDI sequence, pitches will be assigned to the nearest MIDI note. The end of notes also have special handling: they are determined either by the onset of a new note (at most one note is allowed at a time), or by the envelope falling below an audibility threshold (interpreted as no note playing). The pitch at the end of a note is held flat if it shows a tendency to drift, which is common in forced-vibration instruments.

Example results (Figure 3) show the transcription of a jazz solo trumpet piece. In fact, part of the character of jazz music is the ‘bending’ of pitch away from a normal note pitch, and this is clearly visible in the figure. Thus the transcription, while successfully transcribing the notes into a standard format, will have lost something of the original rendition.

<Insert Figure 3 here>

Polyphonic Transcription 1: Blackboard System

Transcription of polyphonic music introduces a number of new complexities that are not present in the monophonic version of this problem. Since we have more than one possible note at once, we can no longer be sure that there will be a single delay at which the whole waveform will repeat, so the autocorrelation approach is no longer suitable. Instead, we base our approaches on an initial time-frequency analysis of the signal, such as that from a fast Fourier transform (FFT).

In order to transcribe the audio signal correctly, we must identify which frequency components are due to each individual note. Each note played will typically generate frequency components at multiples of its fundamental frequency: these frequency components are called *harmonics*. If we can correctly identify the harmonics which are due to each note, we will have gone some way to solving the transcription problem. However, this is not easy for a number of reasons. For example, the frequency range of the harmonics can be very wide, so some frequency ranges can contain harmonics from two or more different notes.

A more awkward problem is that some frequency components may be caused by harmonics of more than one note. In fact, composers often use chords containing notes that have a simple ratio between their fundamental frequencies, such as 2:1 (an *octave*), 3:2 (a *fifth*) or 4:3 (a *third*), since these typically produce a sound which is pleasing to the human ear. If one note of 100Hz is played at the same time as another one octave up, at 200Hz, then every harmonic of the upper note will correspond to the even harmonics of the lower note. For example, the component at 400Hz will be due to both the 2nd harmonic of the note at 200Hz and the 4th harmonic of the note at 100Hz (the "1st harmonic" is the fundamental frequency). In fact, since the 200Hz note produces no other frequency components than might be produced by a 100Hz note, we will have to use some other information (such as the energy expected in each harmonic for a particular instrument) to infer the presence of the 200Hz note.

<Insert Figure 4 here>

To integrate these sources of information, and to make inferences about the likely notes present, we adopted the use of a *blackboard* system as used by e.g. Mellinger (1991) and Martin (1996). The outline of the blackboard system we used (Bello & Sandler 2000) is shown in Figure 4. The first stage is a signal processing stage, which performs a time/frequency analysis of the audio signal. There are many possible approaches here, including the well-known FFT, where we transform short frames of audio into a representation of the amplitude and phase of the frequency content of the signal within each frame. Alternative time-frequency analysis techniques can be used, such as the

Multi-resolution Fourier transform (MFT), Wavelets, and “ear models” built to emulate the human auditory system. These typically permit finer time resolution (and consequently coarser frequency resolution) at higher frequencies, where the normal FFT has the same resolution in time and frequency for all frequency bands, and may help to distinguish note onsets or other fast changes (Pielemeier et al 1996).

The blackboard system proper is a knowledge-based inference engine that can incorporate information from a variety of sources to produce hypotheses about (in our case) the notes present in the audio signal. The *blackboard* is a hypothesis database upon which initial input observations are first written. There is also a set of expert agents, or *knowledge sources* (KSs) which are able to make inferences about some of the hypotheses that may appear on the blackboard. When a knowledge source recognizes a set of hypotheses present on the blackboard, it may fire, writing a new hypothesis onto the blackboard. This new hypothesis may in turn be used by another KS to make a further hypothesis. A scheduler determines the order in which knowledge sources are allowed to fire.

As a simple example, suppose that the signal processing stage had determined that frequency components (*tracks*) at 100Hz, 200Hz, 400Hz and 500Hz were present. Suppose also that there is one KS that knows that the presence of frequency tracks at f and $2f$ is partial evidence for a note at f, and another KS that knows that tracks at $4f$ and $5f$ is also evidence for a note at f. Suppose also that a third KS knows that two or more hypothesis of partial evidence for a note at f is good evidence for a note at f. Then the first two KSs are able to fire, each generating a partial hypothesis of a note at 100Hz. The final KS can then integrate these two partial hypotheses into a good hypothesis of a note at 100Hz, even though its third harmonic (300Hz) was missing, perhaps due to noise.

Note that the first KS can also generate a partial hypothesis of a note at 200Hz (since frequency tracks at 200Hz and 400Hz are present), but this is not supported by further evidence so is likely to be rejected in this simple example. Depending on the design of the blackboard system, hypotheses may be “used up” on firing, which can reduce confusion, but may prevent correct operation where a single frequency track is caused by more than one underlying note.

The initial blackboard system we used had three levels of hypotheses: frequency tracks, partials, and notes, with knowledge sources tuned designed for western piano music. We are also experimenting with additional knowledge sources based on Elman Networks and Time-Delay Neural Networks (TDNNs) for detection of chords from polyphonic notes, and prediction of melodic lines (Figure 5).

<Insert Figure 5 here>

<Insert Figure 6 here>

An example of the analysis is shown in Figure 6., where a graphical transcription is shown for part of two piano solo pieces. For the Liszt Etude, with 2 to 3 notes present at any time, the transcription (Figure 6b) is very close to the original (Figure 6a). The Mussorgski Promenade is a “richer” piece, with 4 to 6 notes present at any time. While most of the notes here are recognized in some form, the duration of many notes in the transcription (Figure 6d) is shorter than the original (Figure 6c), particularly for the higher frequency notes. Further investigation indicates that these notes decay away relatively quickly, and we have found that it is difficult to set an appropriate threshold that will retain these notes while suppressing noise.

Polyphonic Transcription 2: Multiple-Cause Model

One of the issues to be addressed in polyphonic music transcription is the issue of timbre modeling. If we knew that the shapes of the notes were like, in the frequency domain, we could find a combination of notes that exactly matched the frequency content of the input. However, this is somewhat of a chicken-and-egg problem: until we know what the note shapes are for a particular piece, it seems that we cannot find the notes, and if we do not know the notes, we cannot find the note shapes. One approach would be to obtain initial samples of all instruments and notes in isolation (Kashino et al 1998), but this may not always be possible, perhaps if a single recording of unusual instruments was all that was available. In this case, one possible solution to this is to use a type of neural network called the *Multiple-Cause Model* (Saund 1995).

<Insert Figure 7 here>

The multiple-cause model (Figure 7) searches for representations of the underlying causes of the input data, together with amounts of each ‘cause’, which take account of the input data as closely as possible. This model is designed to cope with input data that is composed of several causes active at the same time. In contrast to many neural networks, this network does not operate in a simple feed-forward manner: rather the encoding layer and connections are adjusted until the encoding forms a good reconstruction of the observed data.

The multiple-cause model was originally proposed for analyzing images into constituent causes. A classic example is the ‘bars’ problem, where each image is composed of a number of horizontal and vertical black bars on a white background. Thus an image is ‘caused’ by several bars, maybe overlapping, and each pixel in the input image may be ‘caused’ by one or more bars being present in the image. The multiple-cause model is trained by minimizing an error function, such as negative log likelihood or mean squared error, by adjusting both the underlying measurements of each cause, and the patterns due to each cause.

The multiple-cause model is underdetermined, due to interdependence between the measurements of each cause and the underlying patterns. Depending on the problem, some form of constraints may need to be imposed (Harpur & Prager 1995; Charles & Fyfe 1998). For a musical example, we would expect positive amounts of each underlying note. On the assumption that notes and instruments were independent sources, each would then contribute approximately positive amount to the received spectrum.

Artificial spectra

As an initial step, we next applied the multiple-cause model to artificial spectra produced from synthesized sounds (Klingseisen & Plumbley 2000). We first tested the ability to separate spectra from different notes from the same instrument. We used linear mixtures of spectra, downsampled to 30 bins, of a synthesized clarinet playing one of 8 notes (G_3, C_4, A_3, D_4, F_4, G_4, A_4, E_4), with a probability of 0.4 of each spectrum appearing in each mixed pattern. We repeated this for notes from a synthesized violin and also for an alto recorder: in each case presentation of about 800 training patterns were needed for successful learning. We also explored separation of synthesized spectra of different instruments playing the same note, with similar results, although it appeared to take longer to learn than the experiments with different notes on the same instrument. This is probably due to the similarity between the patterns representing the same note, particularly the alignment of the fundamental and first few harmonics into the same bins in each pattern.

Combining these two approaches, patterns composed from the spectra of three instruments (Clarinet, Oboe, Trumpet) playing each of three notes were used for the training patterns. The spectra used to form the training patterns were downsampled into log-scaled bins, with a relative scaling of $2^{1/12}$ (equivalent to one semitone) between the bins. With synthesized instruments, different notes on the same instrument would then appear simple as shifted along this log-frequency scale, with 1 bin shift equivalent to 1 semitone. A multiple-cause model learned the basic pattern spectra after about 700 presentations of the training patterns. After this, we were able to post-process the resulting patterns to identify which patterns were relative shifts of each other: this correctly identified that 3 instruments were used, with relative semitone shifts of (0, +2, +4), (0, +1, +5) and (0, +2, +3).

Real Sounds

We constructed an audio signal composed of a linear time-domain addition of pulses of notes played on different instruments from the University of Iowa musical instrument samples web page (University of Iowa 1999). This time the audio signals (rather than the spectra) were added, to be closer to the situation in real audio mixing. The multiple-cause model found most of the nine underlying patterns after 300 presentations of the set of training patterns, equivalent to 20 hour's learning using Matlab on a 350MHz Pentium II (Klingseisen & Plumbley 2000).

In Figure 8 we see that the sounds have been separated such that the input sound is represented by a small number of measurement units, with other units off. This indicates that the output units have found a *sparse coding* (Field 1994), i.e. a coding with many units 'off', even without penalty terms that have been used to encourage sparse output distributions (Harpur & Prager 1996; Olshausen & Field 1996).

<Insert Figure 8 here>

We can see that are some instruments that are not completely separated: in particular, it seems that separation of flute and clarinet is difficult. In this example, Flute E_{4b} (input 8) and Clarinet G_{4b} (input 2) are poorly separated, with parts of each instrument found in the corresponding outputs (the label on the right hand side indicates the closest instrument/note). Also, some of the Flute B_4 (input 9) is still mixed with the Clarinet B_{4b} (input 3), with the attack phase of the flute being ‘picked up’ by the Clarinet output. This difficulty may be due to the relatively pure waveforms, and therefore dominant fundamentals, that these instruments have, although more investigation is needed to confirm this.

While this multiple-cause model does not yet give us a quality of analysis that we would need for accurate transcription, the concepts it embodies do crop up again in the approach we describe at the end of this article. However, before we go on to describe this, we will discuss some of the concepts and technology in *blind source separation* that we have used elsewhere, and that will form part of our final polyphonic transcription method.

Blind Source Separation and ICA

So far we have concentrated on the transcription problem of analyzing a musical audio signal, normally a single channel, to reveal the underlying notes that caused the signal that was observed. Another task we are investigating is the problem of separating out the sounds from several sources, when more than one observed signal (microphone) is available. This is sometimes called the *cocktail party problem*, after the apparent ability of two-eared human listeners to separate the various conversations going on in a cocktail party, and concentrate on just one. This problem of *blind source separation* (“blind” because the mixing process is not assumed to be known) has recently been the subject of much research, particularly since the paper of Bell & Sejnowski (1995) brought this problem to the attention of the neural networks community.

<Insert Figure 9 here>

In the simplest form of this problem (Figure 9), we suppose that we have *n* observations containing different instantaneous linear mixtures of *n* original sources, and also that the original sources are independent from each other. In this case we can use *Independent Component Analysis* (ICA) to tackle this problem. Mathematically, we express this as $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ where **x** and **s** are *n*-dimensional real vectors, and **A** is a non-singular mixing matrix. The probability density of the source components s_i are assumed to be independent, so that $p(\mathbf{s}) = \prod_i p(s_i)$.

<Insert Figure 10 here>

A typical ICA method proceeds in two stages (Figure 10). Firstly, covariances in the observations **x**(t) are removed by *whitening*, i.e. linearly transforming the observations to give **z**(t) = **Qx**(y) where **Q** is a linear matrix and **z** has identity covariance matrix, and is therefore *decorrelated*. Secondly, the axes are rotated until the outputs become truly independent. Here this second stage relies on higher-order statistics, such as 4th order.

The ICA procedure has a number of limitations. For example, it cannot separate sources with more than one Gaussian (normally distributed) source: these will be decorrelated, but the original source 'directions' will remain unknown. Another is that the estimated sources obtained are subject to an unknown scaling and perturbation. We cannot tell what the 'amplitude' (variance) of the original sources were (including knowing whether our estimate is an inverted version of the source), nor can we tell in which order they originally appeared. In many cases, this latter issue, the so-called *permutation problem*, does not really matter, but it can be important if we are considering several linked problems where the relative order is critical.

Convolutive mixing

<Insert Figure 11 here>

In realistic auditory environments, room acoustics can introduce delays, echoes and reverberation, leading to convolutive mixing of the sources on their way to the microphones (Figure 11). We can either view this mixing in the time domain, as $\mathbf{x}(t) = \sum_{t} \mathbf{A}(t)\mathbf{s}(t-t)$ or in the frequency domain as $\mathbf{X}(w) = \mathbf{F}(w)\mathbf{S}(w)$ where **F**(ω) is the Fourier transform of **A**(τ), and **X** (resp. **S**) is the Fourier transform of **x** (resp. **s**). Thus in the frequency domain, we can consider this to be many parallel source separation problems, for each different value of *w*.

Several authors have proposed algorithms to tackle this problem (Cichocki, Amari & Cao 1996; Lee, Bell & Lambert 1997; Smaragdis 1997). Many of these approaches can be viewed as attempting to perform ICA on different frequency bands, with some coupling between the frequency components. The coupling is necessary to overcome the permutation problem that we mentioned earlier. If we simply attempted to separate each frequency band independently, we may get e.g. frequency band w_a of source 1 presented on output 1, but frequency band w_b of the same source being presented instead on output 2.

Our approach to this problem (Davies 2001) relies on the tendency for the amplitudes of the various frequency bands of a particular source to vary together. In a Bayesian modeling framework, we introduce an extra scale parameter b_k for each source k, which scales all frequency components of that source, and is assumed to vary slowly. In the different frequency bands we then use a *likelihood ratio jump test* to try out the possible permutations, and choose the permutation with maximum likelihood. We compared our algorithm with the Smaragdis (1997) algorithm, this time on speech sources. For simple room acoustics, we found that either approach worked well. However, for room acoustics with non-trivial echo, for example at 5-10ms, we found that the Smaragdis algorithm failed to separate the sources, while our likelihood ration jump test was still successful (Figure 12). We also found that it was possible to obtain a similar result on simpler acoustics by randomizing the initial mixing matrices. Thus it may be that the Smaragdis algorithm (which uses a gradient descent method) may be trapped in local minima, while the likelihood ratio jump test is able to “jump” out of these if required.

Transcription from ICA and sparse coding

In the previous section, we considered how to separate mixtures of original source signals, where different amounts of each source signal arrive in each microphone due to an unknown linear mixing process, and where we have N microphones to separate N sources. But consider again the transcription problem, where notes of unknown timbre (spectral shape) are mixed together to give the audio signal we listen to. If we consider our N observations to be different frequency components of the signal, we should be able to use an ICA-like approach to separate out the note activities as if they were different sources. Here the ‘mixing matrix’, or *basis*, that we find will be the spectral shape of each note, which should all be different (even if some are just a frequency-scaled version of others).

One other principle will help us. For most musical pieces, even if they are polyphonic, only a few of the many possible notes are being played at any one time. If we wish to describe the volume, or amplitude, of all notes at any particular time, most of these will be zero. This leads to the concept of *sparse coding* (Field 1994), where each item is coded using a representation where only a few values are non-zero. We came across this in our discussion of the multiple-cause model, and sparse coding has also been suggested as suitable for representing natural images (Olshausen & Field 1996).

This sparse coding approach can help to overcome some of the limitations associated with standard ICA: for example, it is possible to separate more sources than the observations available (the *overcomplete basis* case), and it is capable of handling noise on the observations (Lewicki & Sejnowski 2000). Thus it may be able to separate an audio signal into more possible notes than the frequency components available (See also: Bofill & Zibulevski 2000). The sparse coding method is a Bayesian, probabilistic method. The 'mostly zero' expectation is modeled as a prior probability for note activation which is sharply peaked around zero. Otherwise, the structure of the sparse coding model is similar to the Multiple-Cause Model described earlier. It attempts to find a set of sparse and independent notes, together with their spectral shapes, which will reconstruct (or regenerate) the observed signal. This is therefore sometimes referred to as a *generative model* or *analysis-by-synthesis* approach.

We tested our algorithm on real sounds generated by a MIDI synthesizer and then resampled. We used Bach: *Partita in A minor for keyboard* (BWV827), which was chosen since it mainly consists of two or three independent lines, with a few block chords. We used magnitude spectra rather than power spectra as input to the model, so that the noise is approximately Gaussian, as assumed in our model (Abdallah & Plumbley 2001). The model learned 55 basic spectra, 49 of which were note spectra, and others appeared to represent transitions (e.g. note onsets) rather than notes. When the output activations of these basis vectors are plotted, they show a good 'piano-roll' representation for the music contained in the input spectrogram (Figure 13).

The note shapes obtained can clearly be seen in Figure 13(c). Due to the permutation problem, the sparse-coding method cannot automatically identify which note shape corresponds to which note pitch. In this example, the notes were previously re-ordered by eye, but we anticipate that it should be possible to construct automatic re-ordering methods in future.

To check that the analysis was reasonable, we performed a simple re-synthesis by feeding the identified notes into a MIDI piano synthesizer. After some trial and error adjustments, this resulted in a passable rendition of the original piece. A few notes can be heard to be missing here and there, and timing jitters are present that made the resynthesis sound a little like a slightly nervous piano student, although these are more likely to be created by different relative alignment of frame boundaries with note onset times.

Discussion

Analysis and separation of musical audio is still in relative infancy at present, compared with e.g. automatic speech recognition. However, we have tried to give an indication of the variety of techniques being used to tackle this problem, from one particular perspective. Other approaches to music analysis and transcription include the Bayesian approach (Kashino et al 1998; Walmsley et al 1999), estimation of multiple sinusoids using fast block-based algorithms (Macleod & Tsatsoulis 1999), the number theoretic approach (Klapuri 1998), the use of Kalman filters (Sterian 1999), or the comodulation and dynamical clustering (Scheirer 2000). We feel that this problem is a very interesting and fruitful area for future research.

In blackboard systems, knowledge sources can be quite complex, and may rely on many sets of expert “knowledge”. For example, Godsmark & Brown (1999) describe a blackboard model which integrates evidence from several grouping principles, such as temporal and frequency proximity, common onset and offset times, common frequency modulation, and similarity of timbre (the frequency “shape” of each note). Each principle is handled by a special hypothesis formation expert, with the parameters of the knowledge sources set to be consistent with known psychophysical measurements.

There are a number of issues that are outside the scope of this article, although they are likely to be important for the analysis of musical audio (See also: Byrd & Crawford 2001). We have, for example, largely ignored the temporal structure of the musical signals so far, concentrating to a large extent on the frequency content of the current audio signal. However, music includes many types of time structure, from small, fast variations of pitch (e.g. vibrato), through rhythmic beat structure, to the structuring of verses, and these are likely to have to be an important part of any practical music transcription system.

Another issue that is likely to complicate practical music transcription systems is that the timbre of a note (its frequency content) changes over time. For example, in a plucked-string instrument such as the harpsichord, when the string is first plucked there is a wide range of harmonics present in the sound, and a “bright” sound is heard. However, the high frequency harmonics decay relatively quickly, as they lose their energy, so the note becomes “darker”, leading to the concept of *timbre tracks* traced out by a note as its sound changes (Godsmark & Brown 1999). Our current sparse coding method relies on the timbre remaining relatively constant, so this issue will need tackling before we move on to more complex instruments. In passing, it is worth mentioning that monophonic music transcription does not need information about the timbre of notes to analyze the pitch, so it is likely to be more robust against changing timbre than many polyphonic methods.

There are also instruments that do not have a fundamental simple harmonic structure, such as percussion instruments like drums or cymbals, and the polyphonic analysis methods we have outlined in this article would not be suitable for these. Even apparently straightforward instruments, such as the piano, may be more complex than appears at first. Due to the different modes of vibration of the strings within the piano, some harmonics occur at slightly non-integer multiples of the fundamental frequency: in effect, their phase drifts (i.e. they drift “out of sync”) relative to the fundamental as time proceeds (Daudet: personal communication).

Although we have concentrated here on the concept of automatic music transcription for writing down a printed score, one potential application for automatic music transcription is in the area of audio coding for compression. Currently most audio compression methods, such as the currently popular MP3 format (MPEG-1 Layer 3), attempt to reduce the number of bits used to transmit certain frequency bands to match their audibility. An alternative approach would be to analyze the audio to uncover the “objects” that gave rise to the audio sound to be coded, and send a coded description of those objects instead of the audio itself. The recent MPEG-4 Structured Audio (MP4-SA) standard (Vercoe et al 1998) would offer a possible mechanism to transmit these audio objects, by sending an *orchestra* description, describing what instruments (sound sources) are available, followed by a *score*, specifying how these should be played. Currently these MP4-SA descriptions are authored directly (e.g. from a human playing at a keyboard), but if high quality polyphonic music transcription is possible, these structured descriptions could be created automatically from audio signals. With the continual need for bandwidth to store and deliver music over the Internet, including to mobile devices, it may be that this object-based coding application turns out to be more important than the production of musical scores.

Conclusions

We have given an overview of a range of approaches to analysis and separation of musical audio used within our group. In particular, we have investigated the problems of automatic music transcription and audio source separation. Monophonic music transcription, where a single note is present at one time, can be tackled using an autocorrelation-based method. For polyphonic music transcription, with several notes at any time, other approaches can be used, such as a blackboard model or a multiple-cause/sparse coding method. The latter is based on ideas and methods related to independent component analysis (ICA), a method for sound source separation.

We feel that the research area of musical audio analysis and separation is a very interesting and fruitful area for future research, and may lead to improved methods for audio coding and compression as well as tools to assist musicologists.

Acknowledgements

SA is supported by an EPSRC Postgraduate Studentship. JPB and GM are supported by the OMRAS project, jointly funded by JISC (UK) and NSF (USA).

References

Abdallah, S. A., and Plumbley, M. D. 2001. *Sparse coding of music signals*. Submitted for publication.

Bell, A. J. & Sejnowski, T. J. 1995. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7, pp. 1129-1159.

Bello, J. P., Monti, G. and Sandler M. B. 2000. Techniques for Automatic Music Transcription. In *Proceedings of the International Symposium on Music Information Retrieval (MUSIC IR 2000), Plymouth, MA, 23-25 Oct*.

Bello, J. P. and Sandler, M. B. 2000. Blackboard System and Top-Down Processing for the Transcription of Simple Polyphonic Music. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 7-9.*

Bofill, P. & Zibulevski, M. 2000. Blind separation of more sources than mixtures using sparsity of their short-term Fourier transform. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation, 19-22 June 2000, Helsinki, Finland*, pp 87-92.

Bregman, A. 1990. *Auditory Scene Analysis*. MIT Press.

Brown, J C & Zhang, B. 1991. Musical frequency tracking using the methods of conventional and 'narrowed' autocorrelation. *Journal of the Acoustical Society of America*, 89, 2346-2354.

Byrd, D. & Crawford, T. 2001. Problems of music information retrieval in the real world. Submitted to *Information Processing and Management*.

Charles, D. & Fyfe, C. 1998. Modelling multiple-cause structure using rectification constraints. *Network: Computation in Neural Systems*, 9, pp 167-182.

Cichocki, A, Amari, S., and Cao, J. 1996. Blind separation of delayed and convolved signals with self-adaptive learning rate. In *Proc. NOLTA'96.*

Davies, M. E. 2001. Audio Source Separation. Submitted to *Mathematics in Signal Processing V*, Editors McWhirter and Proudler.

Ellis, D. P. W. 1996. *Prediction-driven Computational Auditory Scene Analysis*. PhD Thesis, Department of Electrical Engineering and Computer Science, MIT.

Field, D. J. 1994. What is the goal of sensory coding? *Neural Computation*, 6:559-601.

Godsmark, D., & Brown, G. J. 1999. A blackboard architecture for computational auditory scene analysis. *Speech Communication* 27, 351-366.

Harpur, G. F. and Prager, R. W. 1995. *Techniques for low entropy coding*. Technical Report CUED/F-INFENG/TR. 197, Engineering Department, Cambridge University, UK.

Harpur, G. F. and Prager, R. W. 1996. Development of low entropy coding in a recurrent network. *Network: Computation in Neural Systems*, 7:277-284.

Kashino, K., Nakadai, K. , Kinoshita, T. & Tanaka, H. 1998. Application of the Bayesian probability network to music scene analysis. In *Computational Auditory Scene Analysis*, D. F. Rosenthal and H. Okuno, Eds., Lawrence Erlbaum, pp. 115-137.

Klapuri, A. 1998. *Automatic Transcription of Music.* Master of Science Thesis, Department of Information Technology, Tampere University of Technology.

Klingseisen, J. and Plumbley, M. D. 2000. Towards musical instrument separation using multiple-cause neural networks. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation, (ICA’2000), 19-22 June 2000, Helsinki, Finland*, pages 447-452.

Lee, T. W., Bell, A. J. and Lambert, R. H. 1997. Blind separation of delayed and convolved sources. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 758-764. MIT Press, Cambridge, MA, 1997.

Lewicki, M. S. & Sejnowski, T. J. 2000. Learning overcomplete representations. *Neural Computation*, 12, 337-365.

Macleod, M. D. & Tsatsoulis, C. I. 1999. Approaches to music analysis using the sinusoid plus residual model. Presented at the *Cambridge Music Processing Colloquium, 30 September 1999, Cambridge, UK.*

Martin, K. 1996. *A Blackboard System for Automatic Transcription of Simple Polyphonic Music*. Technical Report No. 385, Perceptual Computing Section, MIT Media Laboratory.

Mellinger, D. K. 1991. *Event Formation and Separation in Musical Sound*. PhD Thesis, Center for Computer Research in Music and Acoustics, Stanford University.

Olshausen, B. A. and Field, D. J. 1996. Natural image statistics and effcient coding. *Network: Computation in Neural Systems*, 7:333-339, 1996.

Pielemeier, W. J., Wakefield, G. H. & Simoni, M. H. 1996. Time-Frequency Analysis of Musical Signals. *Proceedings of the IEEE, Special Issue on Time-Frequency Analysis*, 84, 1216-1230.

Saund, E. 1995. A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7, pp. 51-71.

Scheirer, E. D. 2000. *Music-Listening Systems*. Ph.D. dissertation, MIT Media Laboratory.

Shuttleworth, T. & Wilson, R. G. 1993. *Note Recognition in Polyphonic Music using Neural Networks*. Research Report CS-RR-252, Department of Computer Science, University of Warwick, Coventry, UK.

Smaragdis, P. 1997. *Information theoretic approaches to source separation*. Master's thesis, MAS Department, Massachusetts Institute of Technology.

Sterian, A. D. 1999. *Model-Based Segmentation of Time-Frequency Images for Musical Transcription*. PhD Thesis, Department of Electrical Engineering, University of Michigan.

University of Iowa. 1999. Musical instrument samples web page. URL http://theremin.music.uiowa.edu/~web/sound/

Vercoe, B. L., Gardner, W. G. & Scheirer, E. D. 1998. Structured Audio: Creation, transmission, and rendering of parametric sound representations. *Proceedings of the IEEE*, 86, 922-940.

Walmsley, P. J., Godsill, S. J. & Rayner, P. J. W. 1999. Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters. *In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Oct 17-20 1999, New Paltz, New York.*

[FIGURE CAPTIONS]

Figure 1. Block diagram of monophonic transcription system.

Figure 2. Output from the autocorrelation stage of the monophonic transcription system, revealing a pitch period of about 76 samples. The output corresponding to small delays have been set to zero to suppress the large initial peak at the zero-shift $\tau=0$ position.

Figure 3. Results of the monophonic transcription process, showing (a) the original trumpet audio waveform, (b) the output from the pitch tracker, and (c) the conversion to the nearest midi notes, indicated by start time ('o') and duration ('+'s).

Figure 4. Outline of polyphonic transcription system based on a blackboard model.

Figure 5. Hypothesis levels within the blackboard system.

Figure 6. Results of polyphonic music transcription using the blackboard system. (a) Extract of *Liszt: Etude No. 5 aus Grandes Etudes de Paganini*.(b) Transcription of (a). (c) Extract of *Mussorgski: Promenade - Ballett der Küchlein in ihren Eierschalen*. (d) Tanscription of (c). This rendition of the extracts is copyright Bernd Krueger.

Figure 7. Multiple-cause model.

Figure 8. Results of multiple-cause model separation of audio sounds.

Figure 9. The simple blind source separation problem.

Figure 10. A typical ICA procedure takes the original data (a), performs a whitening stage (b) to remove covariances, and finally rotates the data axes to produce the independent outputs (c).

Figure 11. Convolutive mixing. The observations are composed of various delayed versions of the original sources, which we can consider to be filtered versions of the original sources.

Figure 12. Separation results for mixtures of speech ("A-B-C-D-E" mixed with "1-2-3-4-5") with non-trivial echo for (a) the Smaragdis (1997) algorithm and the Likelihood Ration Jump Test (b). We can see that the Smaragdis algorithm appears to have extracted the low frequency components from one source, together with the high frequency components of the other, and vice-versa.

Figure 13. Polyphonic transcription of extract of *Bach: Partita*, showing (a) input spectrogram and (b) output note activations, which is visibly sparser than the input. The note shapes (c), which had to be ordered manually, clearly show the harmonic structure, except for the "notes" above 43 which probably represent transitions.