M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 399
Presented at the Third Joint Meeting of the Acoustical Societies of America and Japan, December 1996

Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing[*]

Keith D. Martin[†]
Room E15-401, The Media Laboratory
Massachusetts Institute of Technology
20 Ames St., Cambridge, MA 02139

Abstract

It is only very recently that systems have been developed that transcribe polyphonic music with more than two voices in even limited generality. Two of these systems [Kashino *et al.*1995, Martin 1996] have been built within a *blackboard* framework, integrating front ends based on sinusoidal analysis with musical knowledge. These and other systems to date rely on instrument models for detecting octaves. Recent results have shown that an *autocorrelation*-based front end may make bottom-up detection of octaves possible, thereby improving system performance as well as reducing the distance between transcription models and human audition. This report outlines the blackboard approach to automatic transcription and presents a new system based on the log-lag correlogram of [Ellis 1996]. Preliminary results are presented, outlining the bottom-up detection of octaves and transcription of simple polyphonic music.

1 Introduction

In this report, we present the basis of an automatic transcription system. We describe a computational framework (blackboard systems) that allows the integration of both bottom-up and top-down processing, both of which seem to be required to explain a broad range of human perception. We make a case for a correlation-based front end rather than the sinusoidal analysis ususally employed in transcription systems. In particular, we argue that the log-lag correlogram is an appropriate signal representation, one that makes bottom-up detection of octaves feasible without introducing explicit instrument models.

After giving a brief history of transcription systems, we introduce blackboard systems and the log-lag correlogram. We then present some implementation details of the current system, preliminary experimental results, and directions for future work.

1.1 Transcription — past, present, future

Automatic transcription, in the sense of extracting note pitches and onset/offset times from an audio signal, has interested musicians and computer scientists for over twenty-five years. Although these data are not a sufficient representation for reproduction of a perceptually equivalent "copy" of the original performance, as loudness and timbre are completely ignored (see [Scheirer 1995] for an attempt to achieve perceptual equivalence in score-guided transcriptions of piano performances), they go a long way toward forming a useful symbolic representation of the music.

Monophonic transcription (equivalently dubbed "pitch-tracking" in this context) is a mature field, with many well-understood algorithms including time-domain techniques based on zero-crossings and autocorrelation, and frequency-domain techniques based on the discrete Fourier transform and the cepstrum (c.f., [Brown and Zhang 1991, Brown 1992, Brown and Puckette 1993]). Polyphonic transcription (analysis of signals with multiple simultaneously sounding notes) has enjoyed much less relative success.

In the early 1970s, Moorer built a system for transcribing duets [Moorer 1975]. His system was limited, succeeding only on music with two instruments of different timbres and frequency ranges, and with strict limitations on the allowable simultaneous musical intervals in the performance; it was unable to detect octaves or any other intervals in which the fundamental frequency of the higher note corresponds to the frequency of one of the overtones of the lower note.

In 1993, Hawley described a system which he purported could transcribe polyphonic piano performances [Hawley 1993]. His approach was based on a differential spectrum analysis (similar to taking the difference of two adjacent FFT frames in a short-time Fourier transform) and was reported to be fairly successful, largely because piano notes do not modulate in pitch. His system is an example of a transcription engine with very limited scope, showing that transcription systems can be successful by narrowing the range of input signals they consider and by relying on special characteristics of those signals, which may not be present in a more general class of signals.

While automatic music transcription has been a research goal for over 25 years, it is only in the last few years that systems have been demonstrated that are capable of transcribing more than two simultaneous musical voices in even limited generality (c.f., [Katayose and Inokuchi 1989, Kashino *et al.*1995, Martin 1996]). Systems to date have relied on signal processing front ends that can be characterized as extracting individual partials of musical notes by frequency-domain analysis. The transcription problem then becomes one of *explaining* the set of isolated partials as components of notes. In this report, we will try to make a case for a different approach to front end signal processing that may have more in common with human physiology.

1.2 One reason that transcription is hard

One of the fundamental difficulties for automatic transcription systems, as evidenced by Moorer's and later systems, is the problem of detecting octaves. Simple Fourier series theory dictates that if two periodic signals are related by an octave interval, the note of higher relative pitch will share all of its partials with the note of lower pitch. Without making strong assumptions about

[†]The author was supported by an NSF graduate fellowship during a portion of this research.

[*]The content of this report differs from the abstract printed in the meeting program; [Martin 1996] corresponds more closely to the printed abstract. This report highlights the differences between our approach and the one described in [Kashino *et al.*1995].

the strengths of the various partials (i.e., some kind of instrument model), it will not be possible to detect the higher-pitched note.

For this reason, it is necessary to rely upon another form of knowledge about musical signals in order to resolve the potential ambiguity. As mentioned above, one reasonable approach is to rely on instrument models (assumptions about the relative strengths of the various partials of a note played by a particular instrument, or a generative/synthetic model of the instrument sound). A second possibility is to apply musical knowledge, perhaps in the form of production rules for counterpoint music, or in simple heuristics for harmonic and melodic motion. Both approaches are equally valid.

To make use of such knowledge, it is necessary to have a computational framework capable of organizing it and applying it in the right context.

1.3 Blackboard systems in brief

Contemporaneously with early automatic transcription efforts, so-called "blackboard" systems were developed as a means to integrate various forms of knowledge for the purpose of solving ill-posed problems. The name "blackboard" comes from the metaphor of a group of experts standing in front of a physical blackboard, working together to solve a problem. The experts watch the solution evolve, and each individual expert makes additions or changes to the blackboard when his particular expertise is required. In a computational blackboard system, there is a central workspace/dataspace (the blackboard) which is usually structured in an *abstraction hierarchy*, with "input" at the lowest level and a solution or interpretation at the highest. Continuing the metaphor, the system includes a collection of "knowledge sources" corresponding to the experts. An excellent introduction to the history of blackboard systems may be found in [Nii 1986].

It is notable that the systems described in both [Kashino *et al.*1995] and [Martin 1996] are built within a blackboard framework. Music has a natural hierarchical structure which lends itself to the type of data abstraction hierarchy typically used in blackboard systems (a portion of one possible musical hierarchy is shown in Figure 1). The power of the blackboard framework for transcription is that it provides an environment in which it is possible to integrate both signal processing and musical knowledge into a single system with ease. Blackboard systems are also easy to expand — adding new knowledge amounts to coding a handful of procedural routines and registering them with the control system.

Blackboard systems are notable also for their ability to perform both bottom-up (data-driven) and top-down (expectation- or explanation-driven) processing. [Scheirer 1996] points out that top-down, or predictive, processing is necessary to account for human music perception, and as [Slaney 1995] and [Bregman 1995] have noted, both top-down and bottom-up processing appear to be necessary to explain human *auditory scene analysis*, of which music transcription can be viewed as a special case (one which requires a great deal of expert musical knowledge, however!).

1.4 A pitch perception model as front end

Human pitch perception is a complex phenomenon that has received a great deal of attention in the psychoacoustics literature. Over the years, a number of models have been proposed to account for the many known "oddities" of human pitch perception, including the missing fundamental phenomenon and weak pitch perception arising from interrupted noise. The best known of these models have been based on resolving individual partials with narrow filters, on envelope modulation due to the "beating" of multiple partials in a wider filter, on the alignment of subharmonics, and on autocorrelation (c.f., [Goldstein 1973, Terhardt 1979, Patterson 1987]).

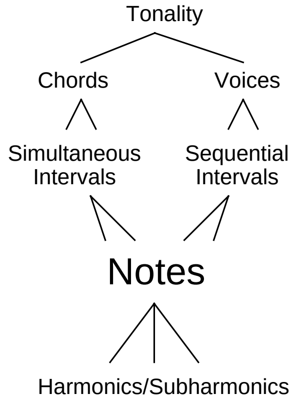

Figure 1: *A portion of one possible data abstraction hierarchy for musical signals, which might be employed within a blackboard transcription system.*

The model which seems to most compactly explain the widest range of psychoacoustic phenomena is the one proposed in [Meddis and Hewitt 1991], which is related to the "correlogram" described in [Slaney and Lyon 1993]. In the pitch perception model, the audio signal is first decomposed into frequency bands by a model of basilar membrane mechanics (implemented by a gammatone filter bank). Each filter channel is further processed by a model of inner hair cell (IHC) dynamics. The IHC model has complicated behavior, but can be viewed as half-wave rectification followed by smoothing (to eliminate carrier frequencies above the phase-locking limit of the hair cells) and onset enhancement. The output of each IHC is analyzed by short-time autocorrelation, yielding an estimate of periodic energy in each filter channel as a function of lag, or inverse pitch. Finally, the autocorrelations are summed across the filter bank channels, and the lag with the resultant largest peak is chosen as the "pitch percept". The Meddis and Hewitt model accounts not only for pitch perception of normal musical notes, but also for the missing fundamental phenomenon and several of the "weak pitch" phenomena.

In his dissertation, Ellis presents a signal processing algorithm that can be viewed as a variant of the Meddis and Hewitt model [Ellis 1996]. Ellis computes a "log-lag" correlogram, where the three axes of the the correlogram volume are: filter channel frequency, lag (or inverse pitch) on a logarithmic scale, and time (see Figure 2). The output of each frequency/lag "cell" is computed by a simple filter structure, as shown in Figure 3. To compute the "pitch percept", Ellis normalizes the output of each frequency/lag cell by the energy in that filter bank channel (given by the output for that channel at zero lag), and averages across the filter bank channels, yielding what he calls the *summary autocorrelation*, or *periodogram*. The log-lag (log-pitch) axis is an improvement over standard correlograms in that it more closely relates to the variation in pitch resolution ability of humans as a function of pitch. A variant of Ellis's model serves as the "front-end" for the system presented in this paper.

It is our contention that transcription systems built with a correlation-based front end will be more robust than systems with "sinusoid-based" front ends, in that they will not require explicit

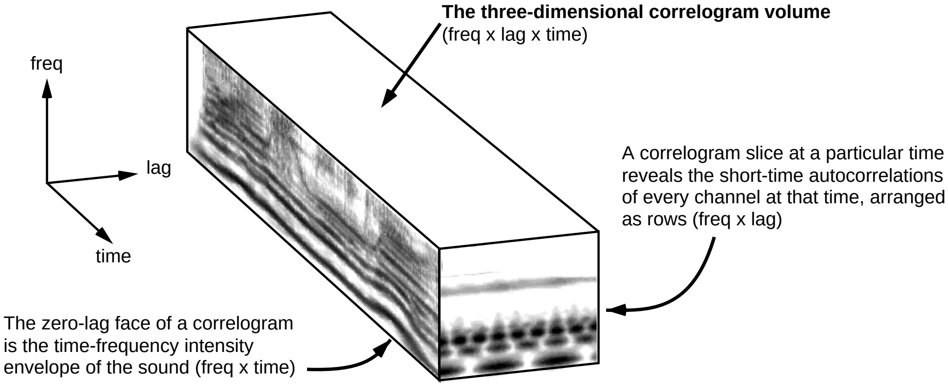
The three-dimensional correlogram volume
(freq x lag x time)
freq
lag
time
A correlogram slice at a particular time reveals the short-time autocorrelations of every channel at that time, arranged as rows (freq x lag)
The zero-lag face of a correlogram is the time-frequency intensity envelope of the sound (freq x time)

Figure 2: *A sketch showing the three axes of the correlogram volume. From [Ellis 1996], reprinted with permission.*

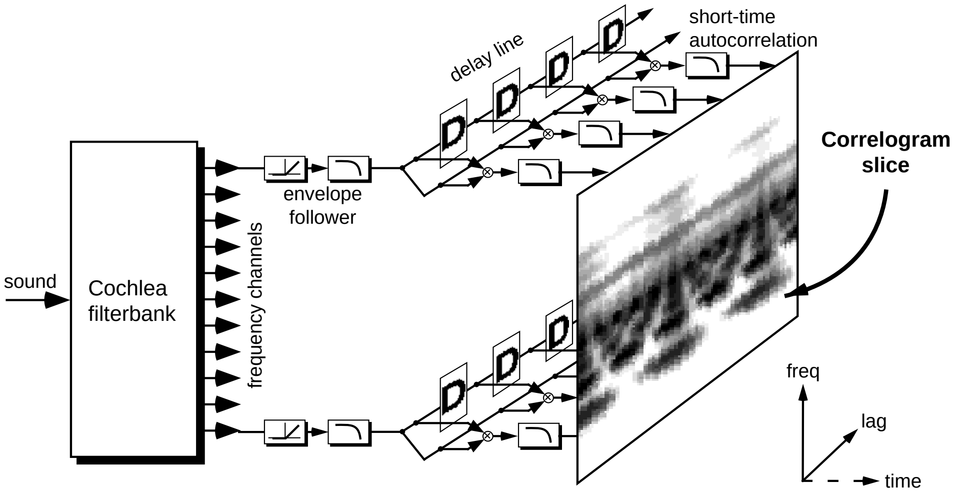

Figure 3: *A schematic drawing of the signal processing network underlying the correlogram calculation. From [Ellis 1996], reprinted with permission.*

instrument models (or may, in fact, be able to acquire their own instrument models without explicit training).

2 Implementation

In this section, the implementation details of the transcription system are presented. The signal processing underlying the system's front end is described, followed by descriptions of the blackboard system control structure, data abstraction hierarchy and knowledge base.

2.1 The front end

As described in the Introduction, the front end signal processing in the current system is modeled after the log-lag correlogram of [Ellis 1996], which may be viewed as a variant of the correlogram of Slaney and Lyon and of the pitch perception model of Meddis and Hewitt. In the current implementation, the filter bank is made up of forty gammatone filters (six per octave), with center frequencies ranging from 100 Hz to 10 kHz, spaced evenly in log frequency. The standard Patterson-Holdsworth filter parameters have been used, yielding filter bandwidths based on the ERB scale [Patterson and Holdsworth 1990].

The lag axis of the correlogram volume is sampled at 48 lags/octave, from 20 Hz to approximately 1 kHz, which yields adequate resolution for most musical signals. The time axis is downsampled to 220.5 Hz before being processed by the blackboard system. The correlogram implementation is identical to that described in [Ellis 1996], with the exception that the envelope follower lowpass filter cutoff frequency is decreased with increasing lag, such that the correlogram output is nearly critically sampled (in lag) at all lags (Ellis chose a single cutoff as a compromise between oversampling at short lags and undersampling at long lags).

As mentioned previously, a *summary autocorrelation* or *periodogram* is computed from the correlogram by normalizing each frequency/lag cell by the zero-lag energy in the same frequency band and then averaging across the frequency bands. An example of correlogram output and corresponding summary autocorrelation is shown in Figure 4.

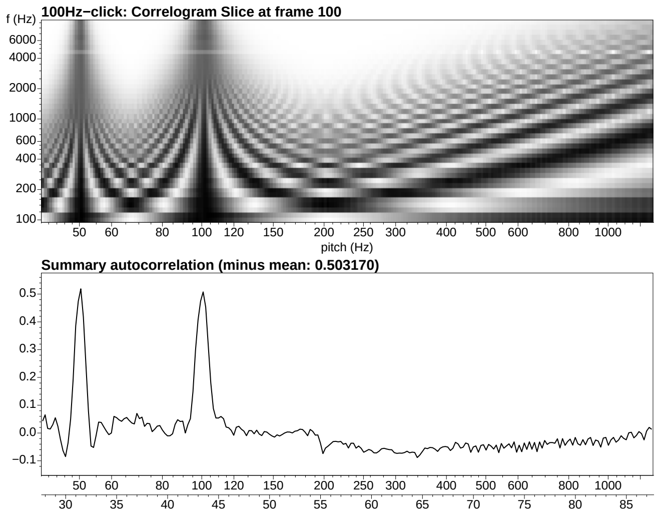

Figure 4: *Log-lag correlogram and summary autocorrelation for a 100 Hz impulse train. When a single pitch is present in the input signal, the summary autocorrelation is characterized by a sharp peak at the pitch (inverse lag) of the signal as perceived by humans, and at its subharmonics. In this figure and in all following correlogram-based figures, the lag axis has been inverted and labeled pitch for convenience. The linearly-spaced axis beneath the pitch axis corresponds to MIDI note, included for convenience in later figures.*

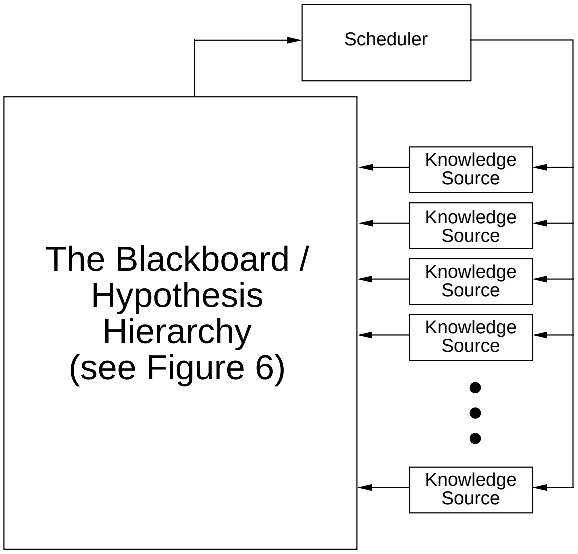

Figure 5: *The control structure of the blackboard system described in this report.*

2.2 Blackboard control structure

As described in the Introduction, blackboard systems usually consist of a central dataspace (the blackboard), a set of so-called *knowledge sources* (KSs), and a scheduler. This is the implementation style that has been adopted for the current system. It is shown schematically in Figure 5.

On a given blackboard time step[1], the control system selects a Focus of Attention (FOA), which may be a particular hypothesis currently on the blackboard, or a particular region of the blackboard (e.g., a particular node of the data abstraction hierarchy). Knowledge source (KS) preconditions are then selected and tested, based on their potential applicability to the FOA. If KSs are activated (signalled by adding their action procedures to an execution list), the action component of the KS with the highest "expected benefit rating" is executed. If no KSs are activated the FOA selection process is iterated until a suitable FOA is found.

When an acceptable KS action is found and executed, the blackboard makes note of what has changed on the blackboard and notifies all KSs that have registered interests in those types of events. In this way, KSs can be triggered in an interrupt- or event-driven fashion, which can potentially cut down on computational requirements.

As mentioned above, each knowledge source is made up of a precondition/action pair, encoded procedurally. Each KS is also responsible for registering "interests" with the blackboard (a particular KS, for example, might be interested in all new Note hypotheses, or all changes made to a particular Chord hypothesis). Each KS maintains a set of instantiations, each of which has a stimulus frame and a response frame. The stimulus/response frames are used by the blackboard to test whether a given KS instantiation is applicable to a particular FOA.

2.3 Blackboard data abstraction hierarchy

In the current implementation, the blackboard workspace is arranged in a hierarchy, with the log-lag correlogram input at the lowest (least abstract) level and notes at the highest. The general outline is shown in Figure 6 with planned extensions in dashed boxes.

Each blackboard level is home to hypotheses of a particular type. In the current system, hypotheses are implemented in a frame-like manner. All hypotheses share a common set of slots (data) and methods (code), including lists of supported and supporting hypotheses at neighboring blackboard levels. Additionally, each type of hypothesis has its own internal rating scheme, divided into two components: a support rating and an explanation rating. The internal ratings are collapsed to a six point ordinal scale, which is accessible by the KSs and the control system. Changes in the support rating of a particular hypothesis are passed upward to any supported hypotheses. Similarly, changes in explanation rating are passed downward to supporting hypotheses.

2.3.1 Correlogram Frame

At the lowest level of abstraction lie the Correlogram Frame hypotheses. Each contains a "slice" of the correlogram volume at a particular time. The correlogram is considered the "ground truth" in the system and is therefore given, by definition, a maximal support rating. Correlogram Frame hypotheses have access methods for the individual lag/frequency cells, as well as a graphical interface method currently called from a Tcl/Tk shell.

[1] Blackboard time steps have no fixed relation to time within the musical signal, though there is usually a strong correlation. Generally, the system runs through some small number of blackboard time steps for each frame of input data, however, the system is allowed to reprocess difficult portions of the input data if new information arises.

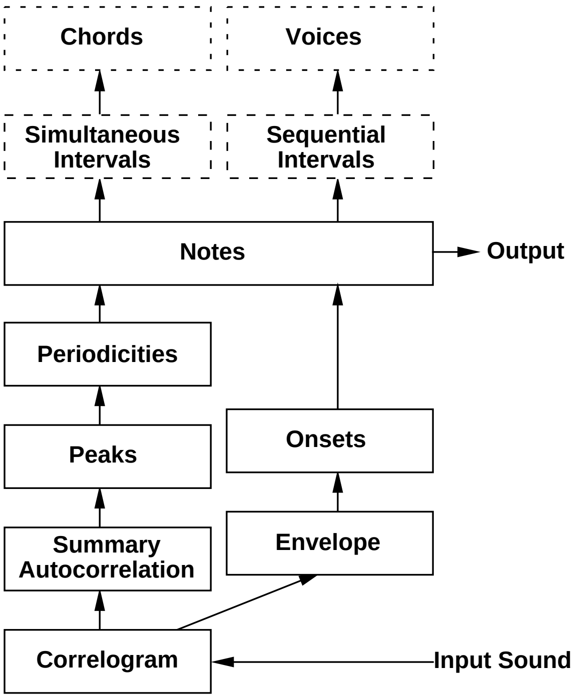

Figure 6: *The data abstraction hierarchy in the current blackboard implementation. Regions with dashed borders are planned extensions.*

2.3.2 Summary autocorrelation/periodogram

Above each Correlogram Frame lies a single Summary Autocorrelation hypothesis, formed by averaging the energy of the lag/frequency cells across frequency. Summary Autocorrelation hypotheses provide access methods for their values at particular lags. Additionally, a graphical interface method for use with Tcl/Tk has been implemented. Since the Summary Autocorrelation is derived algorithmically from the Correlogram Frame, it shares the maximal support rating of the supporting Correlogram Frame.

2.3.3 Peaks

The local maxima of each Summary Autocorrelation frame form Peak hypotheses, which have slots for the peak frequency, height, and the average height in a one-octave neighborhood around the maximum. Peaks are merely an intermediate step between the Summary Autocorrelation and Periodicity hypotheses, used for programming convenience. It is likely that they will be eliminated in future revisions of the system. The support rating of each Peak hypothesis is based on the ratio of its height to its average neighborhood height.

2.3.4 Periodicities

A Periodicity hypothesis is a collection of Peak hypotheses that may persist across multiple Correlogram frames. As mentioned in the section describing the front end, a pitched input signal will result in a subharmonic series in the Summary Autocorrelation. A Periodicity hypothesis may then be thought of as a “pitch” hypothesis, formed by gathering together Peak hypotheses which form a subharmonic series. Periodicity hypotheses have a pitch slot, as well as a “strength” score, based on the average ratio of Peak height to neighborhood average height.

2.3.5 Envelope

Envelope hypotheses are a second part of the “ground truth” derived from an input signal. For each of the correlogram filter bank channels, a zero-lag correlation is calculated, corresponding to a running estimate of the energy in that channel. As with Correlogram Frames, Envelope hypotheses have maximal support ratings.

2.3.6 Onsets

Onset hypotheses are derived directly from the Envelope signals. A first difference approximation of the Envelope slope (measured in dB) is calculated and local maxima become new Onset hypotheses. In addition to slots for onset time and envelope slope, Onset hypotheses have a slot for the energy reached at the next local maxima in the envelope signal. The Onset hypothesis support rating is based upon both the slope and the peak energy of the onset.

2.3.7 Notes

Note hypotheses consist of one or more Periodicity hypotheses, combined with one Onset hypothesis. In addition to a frequency slot filled in by a weighted average of the frequencies of the supporting Periodicity hypotheses, Note hypotheses have a pitch class method and a MIDI-note method, used for generating output in the form of a MIDI file, symbolic score output, or piano roll notation. The Note hypothesis support score is based upon the slope and maximal energy of the component Onset, as well as the support ratings of the component Periodicity hypotheses. In addition, Note hypotheses have several internal flags, which may be set by KSs, giving Note hypotheses access to limited information about their neighbors. These flags can affect ratings, as will be seen in the section describing the knowledge base.

2.4 Blackboard knowledge base

The current implementation of the transcription system is still in its infancy. It was only recently that it was thought feasible to use a correlation-based front-end, and very little of the previous implementation [Martin 1996] was reusable. At present, only five knowledge sources are present in the system, and they act almost entirely in a bottom-up, or data-driven, fashion. Of the five KSs, the first three may be combined in the next revision, as they operate together in a strictly algorithmic manner, and their combination will make it possible to eliminate the Peak hypotheses from the system altogether. The KSs are shown in a stylized representation, overlaid on the data hierarchy, in Figure 7.

As mentioned in the section describing the control system, KSs are made up of three essential components: their “interests”, a precondition component, and an action component. The KSs in the current implementation will be described from this standpoint.

2.4.1 Read Correlogram Frame

The **Read Correlogram Frame** KS has no declared interests, but rather acts as a *daemon*, remaining in the precondition queue at all times. Its precondition is satisfied whenever the blackboard FOA is a time step of the input signal for which there is not currently a Correlogram Frame hypothesis. The KSs action is simply to read the data from disk (the front end analysis is performed offline and stored in a data file), and to create a Correlogram Frame hypothesis along with a supported Summary Autocorrelation hypothesis, and to extend the Envelope hypothesis to include the new frame.

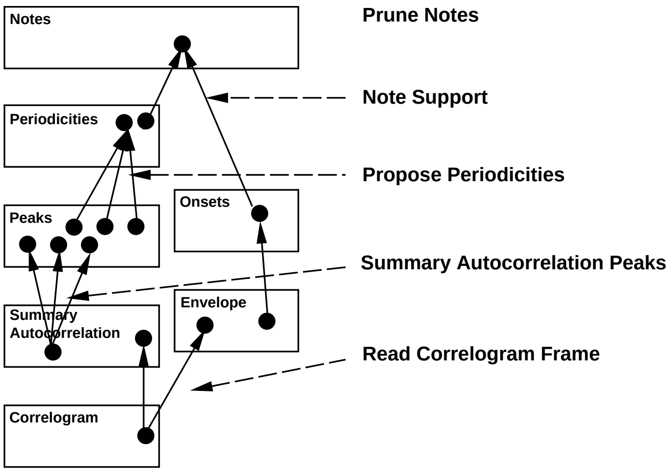

Figure 7: *A graphical representation of the knowledge base as a whole. It shows the hypothesis abstraction hierarchy used in the system with the knowledge sources overlaid. Each KS is represented as a connected graph of nodes, where each node is a hypothesis on the blackboard, and the arrows represent a support relationship.*

2.4.2 Summary Autocorrelation Peaks

The **Summary Autocorrelation Peaks** KS is interested in all new Summary Autocorrelation hypotheses. Its precondition is automatically satisfied by a new Summary Autocorrelation hypothesis, and its action is simply to propose Peak hypotheses for each local maximum in the Summary Autocorrelation.

2.4.3 Propose Periodicities

The **Propose Periodicities** KS is interested in all new Peak hypotheses. Its precondition is automatically satisfied if it has been notified of any new Peaks but hasn't yet acted upon them. Its action consists of two parts. First, the KS looks for existing Periodicity hypotheses on the blackboard. If any are found, new Peak hypotheses which fit the subharmonic series of any Periodicity are added as additional support. After this first round of analysis, remaining "unexplained" Peak hypotheses are considered for the formation of new Periodicity hypotheses. Starting with the Peak of highest frequency (shortest lag), potential subharmonic series are evaluated. Any series with sufficient support is added to the blackboard as a new Periodicity hypothesis.

2.4.4 Note Support

The **Note Support** KS is interested in all New Periodicity hypotheses. It maintains an internal list of all currently active Periodicity hypotheses. Its precondition is satisfied when a Periodicity hypothesis has ended (i.e., there has been no support for several input frames). The KSs action consists of evaluating the support of each Periodicity that satisfied the precondition and adding new Note hypotheses to the blackboard or augmenting existing Note hypotheses as appropriate. The support evaluation is heuristic; Periodicities that persist over a large number of input frames are considered to be strongly supported, as are Periodicities supported by Peak hypotheses with strong support ratings.

2.4.5 Prune Notes

It turns out that the **Note Support** KS creates many more Note hypotheses than there are notes present in a typical musical example (sometimes this is due to chance correlations in a noisy signal, but more often it is due to strong subharmonic series arising from chords and from harmonics/subharmonics of actual notes). Thus the **Prune Notes** KS is used to prune away many of the obviously incorrect note hypotheses.

The **Prune Notes** KS is interested in new Note hypotheses. Like the **Note Support** KS, its precondition is satisfied when a hypothesis it is interested in has ended (i.e., when some number of input frames has elapsed since the last time a given Note hypothesis was extended by a new Periodicity).

The **Prune Notes** action is both complex and heuristic. It is intended to eliminate both harmonics and subharmonics of actual notes without eliminating octaves, which may at first appear to be harmonics. First, the KS assembles a list of existing Note hypotheses which overlap (in time support) the Note hypothesis of interest. Next, the KS finds the maximum onset energy associated with the collection of Note hypotheses. If the onset energy of the Note hypothesis of interest is 25 dB below the maximum (a rather arbitrary threshold – one could certainly look to the psychoacoustics literature for a more perceptually relevant cutoff), the Note hypothesis is labeled with a "Too Weak" flag, which reduces its support rating.

The second portion of the KS action looks for harmonic relations between found Note hypotheses. A frequency ratio is formed between the Note hypothesis of interest and each of the overlapping Note hypotheses, and octave relations are noted. If any are found, Note durations are compared for the relevant hypotheses, and if the duration of Note hypothesis is much shorter than another, the hypothesis of shorter duration is labeled with a "Harmonic" flag, which reduces its support rating. Additionally, if the lower note in an octave relation has more component Periodicities than the upper note, and they have weaker support ratings, the lower note is labeled with a "Subharmonic" flag, which reduces its support rating. Similarly, if the upper note has more component Periodicities, and its Onset rating is less than that of the lower note, it is labeled with a "Superharmonic" flag, which reduces its support rating. These heuristics are based on empirical observation and will be developed more rigorously before the model is further extended.

3 Results

3.1 Bottom-up octave detection

The correlogram/periodogram representation may offer an advantage over sinusoidal representations for detecting the presence of octaves. As a simple example, consider two sounds: the note corresponding to MIDI note 48 (a C pitch) struck on a piano, and the same note struck simultaneously with the note corresponding to MIDI note 60 (a C, one octave higher). The difference between the two sounds is clearly audible, and a person can easily tell which one is an octave and which is a single note (it is worth pointing out that a person who is unfamiliar with the piano timbre might mistake the octave relation for a single note with pitch corresponding to that of the lower C, particularly if the context of the single note is not provided). Figures 8 and 9 show the correlogram/summary autocorrelation representations for the two cases mentioned above, based on samples from an acoustic piano.

By comparing the values of the summary autocorrelation at the subharmonics of MIDI note 60 in Figures 8 and 9 it is clear that while the presence of MIDI note 60 is not obvious at a first glance in

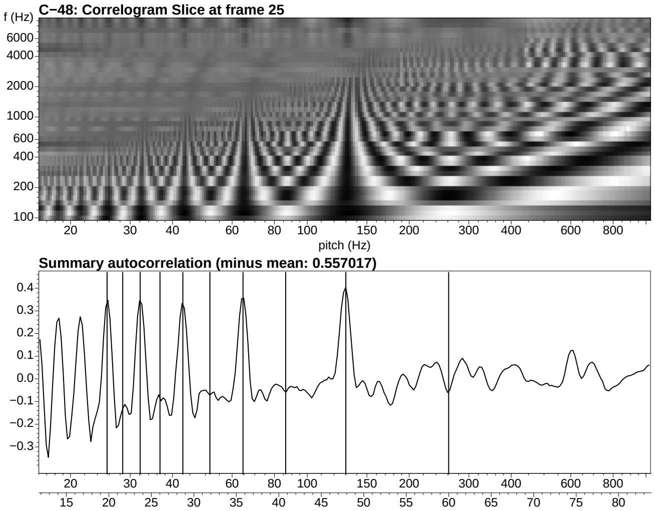

Figure 8: *Correlogram slice and summary autocorrelation of a single piano note, corresponding to MIDI note 48, with lines overlaid at the subharmonics for MIDI note 60, which is not present in the signal. Note the absence of local maxima, particularly at the fundamental frequency and second subharmonic for MIDI note 60.*

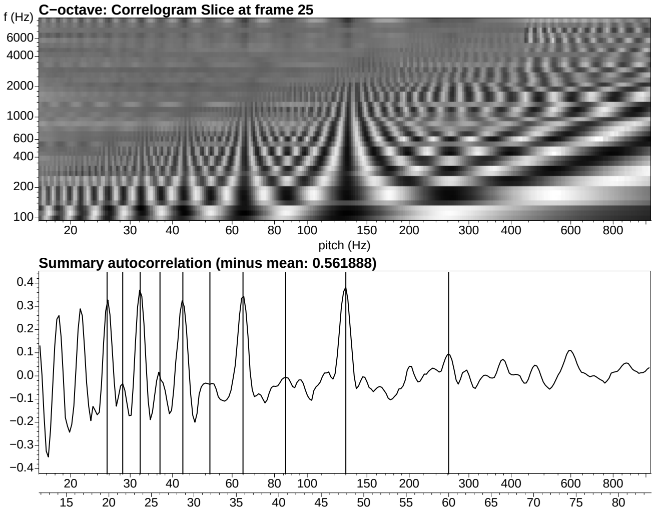

Figure 9: *Correlogram slice and summary autocorrelation of two piano notes, one octave apart, corresponding to MIDI notes 48 and 60, with lines overlaid at the subharmonics for MIDI note 60, which* is *present in the signal. Note the presence of local maxima in the summary correlation at all pitches (inverse lags) corresponding to the subharmonics for MIDI note 60.*

Figure 9, it is visible under closer scrutiny. This result is due to the effect of multiple partials contributing to the autocorrelation within a single frequency band. In the single note case, several channels exhibit beating between two adjacent partials, resulting in strong peaks at their common subharmonics, but weak and/or displaced peaks at the pitches corresponding to the partial frequencies. In the octave example, the partials belonging to the upper note reinforce the even partials of the lower note, causing them to dominate the odd partials somewhat, resulting in more clear peaks at the subharmonics corresponding to the higher C pitch.

This reasoning corresponds to an implicit instrument model, making the tacit assumption that harmonic energy varies smoothly with frequency (a reasonable assumption for many sounds). The implicit assumption is a part of the pitch perception model rather than of the system's knowledge base, however. The model makes the prediction that as the strength of a note's even partials is increased relative to the strength of the odd partials, the note will sound more and more like an octave, and will eventually (when the even partials are much stronger than the odd partials) have a perceived pitch one octave higher. This behaviour corresponds with our intuitions about pitch perception.

3.2 A monophonic transcription example

To show that the correlogram processing is extracting sufficient information for transcription, it is worth looking at its output for a monophonic signal. In this section we consider an short excerpt from a recorded performance of Bach's *Well-Tempered Clavier* (the introduction of the G-minor fugue from Book I). Music notation for the excerpt is shown in Figure 10.

As can be seen from the piano-roll output shown in Figure 12, the first eight notes have been correctly extracted by the system, along with four "extra" note hypotheses which have not been pruned. The additional hypotheses are all close harmonic relations (octaves, fourths and fifths below) of the actual notes in the recording. As the pruning algorithm was hand-tuned for the polyphonic example which follows, this encouraging but not perfect performance is to be expected. is not surprising.

3.3 A polyphonic transcription example

One of the test signals we have been working with is a short segment from the introduction of a Bach chorale (*Erschienen ist der herrlich' Tag*). Traditional music notation for the first phrase of the piece is shown in Figure 13. The sample was generated from a flat MIDI score, using samples from a Bosendorfer acoustic piano. This piece is an example of the type of simple polyphonic music that it is our ultimate goal to transcribe. Figures 14 and 15 show the correlogram/periodogram analysis at two time slices, corresponding respectively to portions of the first and second chords of the piece.

As can be seen in the piano-roll output shown in Figure 16, all of the notes in the first five beats of the piece have been correctly identified, and all extraneous note hypotheses have been pruned successfully. The successful pruning result is due to careful setting of thresholds in the **Prune Notes** KS. In the rest of the example, pruning is less successful. This result seems to be due in large part to a change from closed-form chords to open-form chords around the fifth beat. Encouragingly, nearly all of the notes in the piece appear as Note hypotheses (with the conspicuous exception of the high E [MIDI note 76], whose absence seems to be due to implicit assumptions made in the **Propose Periodicity** KS).

4 Conclusions

While the few results mentioned in the last section are hardly compelling, we are encouraged by them. The knowledge integration approach so far has ignored much of the information contained in the summary autocorrelation representation. One obvious example of this is the extremely sharp peaks exhibited at subharmonics

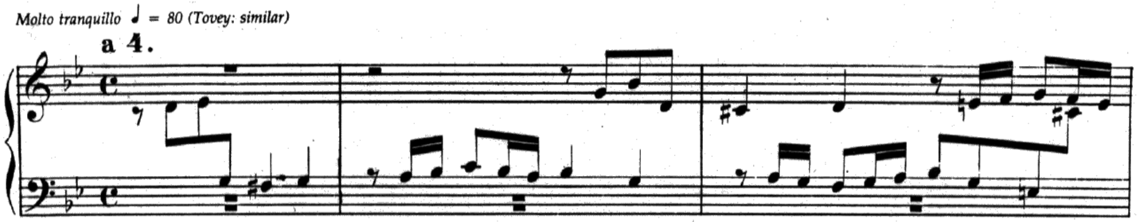

Figure 10: *Music notation for an excerpt from Bach's* Well-Tempered Clavier *(the introduction of the G-minor fugue from Book I).*

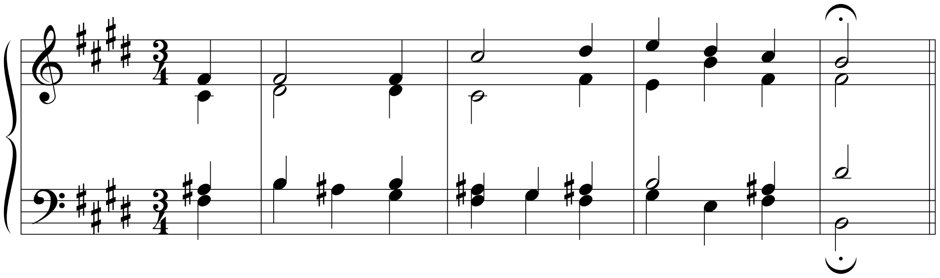

Figure 13: *Music notation for the first phrase of a Bach chorale written in the style of 18th century counterpoint. The piece is titled* Erschienen ist der herrlich' Tag.

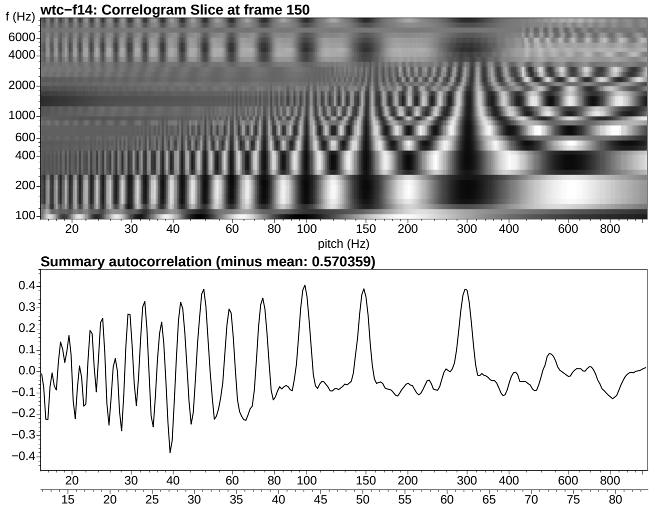

Figure 11: *Correlogram/periodogram analysis during the first note of the monophonic excerpt. It clearly represents a single pitch at MIDI note 62 (D).*

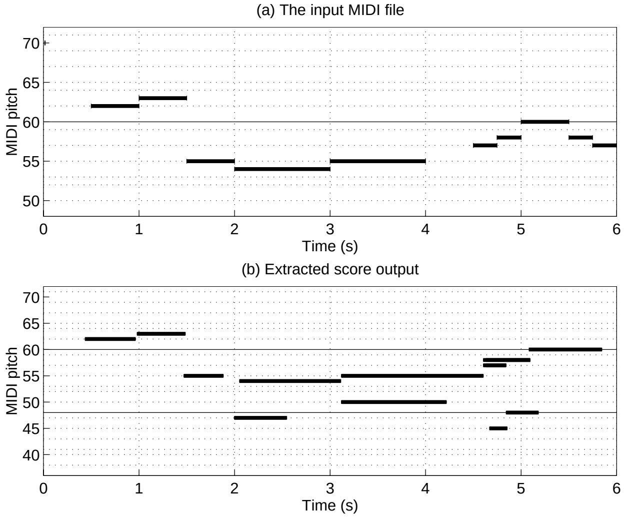

Figure 12: *MIDI score representation and partial analysis of an excerpt from a recording of Bach's* Well Tempered Clavier. *All of the notes in this portion of the recording have been identified, along with a few spurious note hypotheses, which would be pruned by applying musical knowledge in a more complete system. The time scales of the input MIDI file and extracted output have been hand-aligned for ease of comparison (the MIDI file corresponds to a flat interpretation of the score, whereas the output was extracted from a human performance, so the MIDI-file timescale is only approximate.*

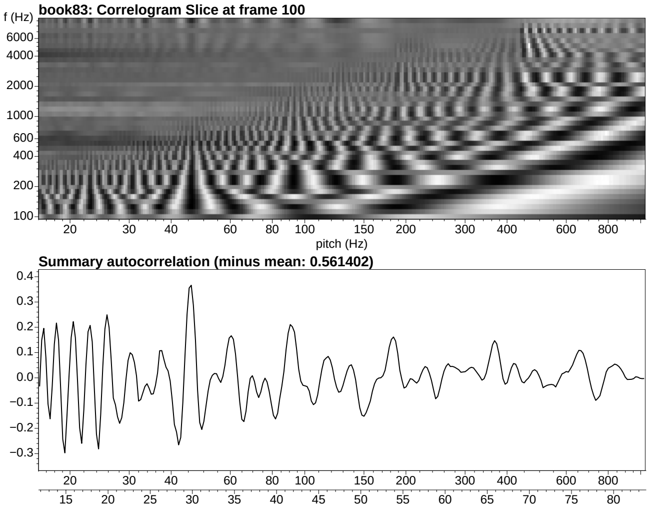

Figure 14: *Correlogram/periodogram analysis of a portion of the first chord of the Bach chorale example. There are notes in the signal corresponding to MIDI notes 54, 58, 61, and 66. The rather large peak at MIDI note 30 corresponds to a subharmonic of the chord root.*

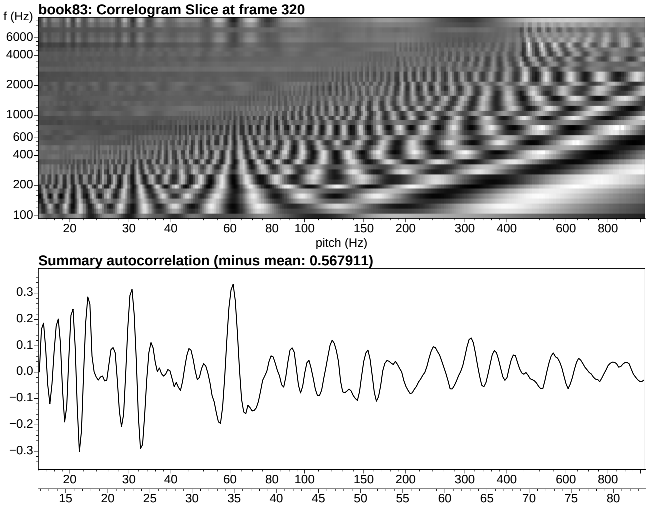

Figure 15: *Correlogram/periodogram analysis of a portion of the second chord of the Bach chorale example. There are notes in the signal corresponding to MIDI notes 59, 63, and 66. The rather large peak at MIDI note 35 corresponds to a subharmonic of the chord root.*

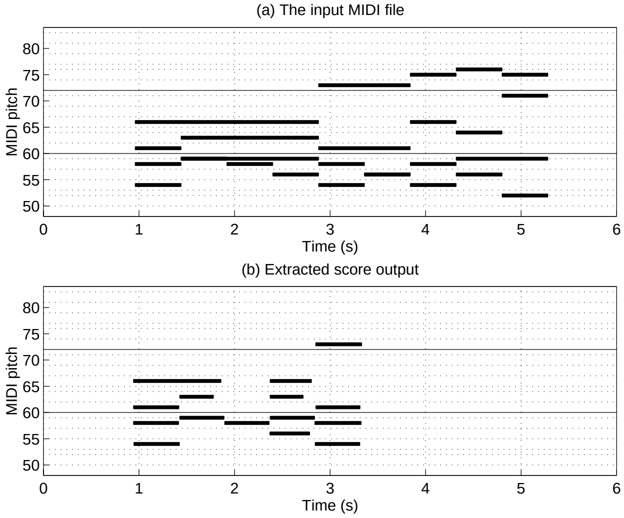

Figure 16: *MIDI score representation and partial analysis of an excerpt from a synthesized recording of Bach's* Erschienen ist der herrlich' Tag. *All of the notes in this portion of the recording have been identified, and all extraneous note hypotheses have been pruned by careful setting of thresholds in the* **Prune Notes** *KS. Analysis breaks down for the rest of the first phrase of the piece, due to the particular thresholds set in the pruning heuristics, as well as the lack of musical knowledge in the system.*

of the chord root (which are predicted by early pitch and chord perception models and clearly show up in the examples). In polyphonic transcription, it certainly makes sense to take advantage of this strong indicator of chord root in order to constrain the search for the chord's component notes.

Without doubt, many improvements could be made in the current bottom-up methods for forming Note hypotheses. We are currently investigating more robust and principled methods for performing this analysis (the present system is essentially an *ad hoc* first attempt), and we expect that performance will improve greatly.

4.1 Are we just trading harmonics for subharmonics?

A question that might be asked in reference to this work is whether the correlation-based front end merely exchanges the problem of dealing with harmonic series in the frequency domain for one of dealing with subharmonic series in "lag" — what is to be gained?

Our answer is twofold. First, even on this surface level, some ground has been gained. In sinusoidal analysis, a principal problem is that of resolving the overtones of a pitch. Often, this problem is "solved" by using extremely narrow filters (with correspondingly sluggish time-responses). The correlogram analysis does not require such narrow filters, and the pitch resolution for the musical examples we have examined (as evidenced by the figures in this report) is on the order of one semitone[2]. The "lag peaks" are quite robust, making subharmonic series detection a fairly simple task.

Second, the discussion of bottom-up octave detection reveals a distinct advantage of the correlation-based approach over sinusoidal approaches for the detection of octaves without introducing an instrument model beyond what is implied by human pitch perception.

4.2 Where do we go from here?

There are many directions in which this research can be extended. Work is planned on proving the mathematical validity of bottom-up octave detection, psychophysical validation of the implied model of human octave perception, as well as the extension of the current system's knowledge base, introducing the capacity for automatic acquisition of instrument models.

4.2.1 Integration of musical knowledge

In order to build a useful transcription system, it is necessary to incorporate a great deal of musical knowledge. Even if the goal is to generate a MIDI representation of the musical information, musical knowledge is necessary to eliminate all spurious "pitch percepts" from the correlogram analysis. This knowledge may take the form of hypotheses regarding the number and type of instruments in a performance as well as melodic/harmonic motion of the piece. Some of these ideas have been implemented in [Kashino *et al.*1995], and it will be fruitful to apply the same approach to this system.

4.2.2 Automatic acquisition of instrument models

[Ellis and Rosenthal 1995] and [Ellis 1996] describe a novel representational element for pitched signals, called the *weft*, which is based on correlogram analysis. The weft is, in essence, a source/filter model, and Ellis's extraction techniques might be used to extract excitation signals and time-varying filters (characterizing the formant structure) from simultaneous pitched sounds. It is our goal to incorporate weft analysis into a blackboard transcription system for the twofold purpose of recognizing previously heard instrument sounds and for acquiring new instrument models based on their time-varying formant structure.

[2] It should be noted that the "width" of peaks in the summary autocorrelation can be reduced by introducing additional smoothing to the correlogram calculation after the multiplication, and by reducing smoothing before multiplication (in the envelope follower).

5 Acknowledgments

This work would not have been possible without the continued support of Barry Vercoe and his Machine Listening Group at the MIT Media Lab. I am grateful to the National Science Foundation for the graduate fellowship which supported me during a portion of this research. Thanks are also due to Dan Ellis both for his significant theoretical contributions to this work and for his moral support, to Eric Scheirer for providing the initial impetus for this project and for his help along the way, and to Bill Gardner, Adam Lindsay and Paris Smaragdis for their comments, encouragement, and C++ code.

References

[Bregman 1990] Albert S. Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, MA, 1990.

[Bregman 1995] Albert S. Bregman. "Psychological Data and Computational ASA". In *Proc. of the Computational Auditory Scene Analysis Workshop; 1995 International Joint Conference on Artificial Intelligence*, August 1995.

[Brown and Puckette 1993] Judith C. Brown and Miller S. Puckette. "A high resolution fundamental frequency determination based on phase changes of the Fourier transform". *J. Acoust. Soc. Am.*, 94:662–667, 1993.

[Brown and Zhang 1991] Judith C. Brown and Bin Zhang. "Musical frequency tracking using the methods of conventional and "narrowed" autocorrelation". *J. Acoust. Soc. Am.*, 89(5):2346–2354, 1991.

[Brown 1992] Judith C. Brown. "Musical fundamental frequency tracking using a pattern recognition method". *J. Acoust. Soc. Am.*, 92(3):1394–1402, 1992.

[Dorken *et al.*1992] Erkan Dorken, Evangelos Milios, and S. Hamid Nawab. "Knowledge-Based Signal Processing Application". In Alan V. Oppenheim and S. Hamid Nawab, editors, *Symbolic and Knowledge-Based Signal Processing*, chapter 9, pages 303–330. Prentice Hall, Englewood Cliffs, NJ, 1992.

[Ellis and Rosenthal 1995] Daniel P. W. Ellis and David Rosenthal. Mid-level representation for computational auditory scene analysis. In *Proc. of the Computational Auditory Scene Analysis Workshop; 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995.

[Ellis 1996] Daniel P. W. Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, M.I.T., Cambridge, MA, June 1996.

[Goldstein 1973] J. L. Goldstein. "an optimum processor theory for the central formation of the pitch of complex tones". *J. Acoust. Soc. Am.*, 54:1496–1515, 1973.

[Hawley 1993] Michael Hawley. *Structure out of Sound*. PhD thesis, MIT Media Laboratory, 1993.

[Kashino *et al.*1995] Kunio Kashino, Kazuhiro Nakadai, Tomoyoshi Kinoshita, and Hidehiko Tanaka. Application of bayesian probability network to music scene analysis. In *IJCAI-95 Workshop on Computational Auditory Scene Analysis*, Montreal, Quebec, August 1995.

[Katayose and Inokuchi 1989] Haruhiro Katayose and Seiji Inokuchi. “The Kansei Music System”. *Computer Music Journal*, 13(4):72–77, 1989.

[Klassner *et al.*1995] Frank Klassner, Victor Lesser, and Hamid Nawab. “The IPUS Blackboard Architecture as a Framework for Computational Auditory Scene Analysis”. In *Proc. of the Computational Auditory Scene Analysis Workshop; 1995 International Joint Conference on Artificial Intelligence*, Montreal, Quebec, 1995.

[Martin 1996] Keith Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical Report #385, MIT Media Lab, Perceptual Computing Section, July 1996.

[Meddis and Hewitt 1991] Ray Meddis and Michael J. Hewitt. “Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification”. *J. Acoust. Soc. Am.*, 89(6):2866–2882, 1991.

[Moorer 1975] James A. Moorer. *On the segmentation and analysis of continuous musical sound by digital computer*. PhD thesis, Department of Music, Stanford University, Stanford, CA, May 1975.

[Nii 1986] H. Penni Nii. “Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures”. *The AI Magazine*, pages 38–53, Summer 1986.

[Patterson and Holdsworth 1990] R. D. Patterson and J. Holdsworth. A functional model of neural activity patterns and auditory images. In W. A. Ainsworth, editor, *Advances in speech, hearing and language processing vol. 3*. JAI Press, London, 1990.

[Patterson 1987] R. D. Patterson. “a pulse ribbon model of monaural phase perception”. *J. Acoust. Soc. Am.*, 82:1560–1586, 1987.

[Scheirer 1995] Eric Scheirer. Using musical knowledge to extract expressive performance information from audio recordings. In *IJCAI-95 Workshop on Computational Auditory Scene Analysis*, Montreal, Quebec, August 1995.

[Scheirer 1996] Eric Scheirer. “Bregman’s Chimerae: Music Perception as Auditory Scene Analysis”. In *Proc. 1996 Intl Conf on Music Perception and Cognition*, 1996.

[Slaney and Lyon 1993] Malcolm Slaney and Richard F. Lyon. “On the importance of time — a temporal representation of sound”. In Martin Cooke, Steve Beet, and Malcolm Crawford, editors, *Visual Representations of Speech Signals*, pages 95–116. John Wiley & Sons, 1993.

[Slaney 1995] M. Slaney. “A critique of pure audition”. In *Proc. of the Computational Auditory Scene Analysis Workshop; 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995.

[Terhardt 1979] E. Terhardt. “calculating virtual pitch”. *Hearing Research*, 1:155–182, 1979.

[Winograd and Nawab 1995] Joseph M. Winograd and S. Hamid Nawab. “A C++ Software Environment for the Development of Embedded Signal Processing Systems”. In *Proceedings of the IEEE ICASSP-95*, Detroit, MI, May 1995.

[Winograd 1994] Joseph M. Winograd. IPUS C++ Platform Version 0.1 User’s Manual. Technical report, Dept. of Electrical, Computer, and Systems Engineering, Boston University, 1994.