

Model-Based Musical Transcription

Presented at the 1999 International Computer Music Conference, Beijing, China

Andrew Sterian¹, Mary H. Simoni², and Gregory H. Wakefield¹

¹Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor MI, USA
<steriana@gvsu.edu> <<http://claymore.engineer.gvsu.edu/~steriana>>
<ghw@eecs.umich.edu> <<http://www.eecs.umich.edu/~ghw>>

²School of Music, University of Michigan, Ann Arbor MI, USA
<msimoni@umich.edu> <<http://www.music.umich.edu/faculty/simoni.mary.html>>

Abstract

We consider the problem of transcribing a musical audio waveform to its underlying note-based representation. Our approach focuses on constructing models of partial behavior both as isolated structures and as part of a note complex. These models are used to guide segmentation algorithms based on Kalman filtering and multiple-hypothesis tree search. Our data-model-based approach stands in contrast to those based upon auditory models. Our goal is to explore the degree of success attainable from careful attention to the data and the associated algorithms before we begin to consider the additional structure imposed by auditory models. We discuss results of our algorithm operating on multi-voice musical passages performed on brass instruments.

1. Introduction

Musical transcription is the transformation of an audio waveform to a parametric representation, such as a musical score. Most commonly, it is assumed that the notes of a score have been rendered to an audio waveform by one or more performers playing musical instruments. This definition of transcription is intentionally broad as there are several levels of parameterization that one can consider. Even the original score does not fully describe the performance, as there is also a performer involved who can add color to the waveform via performance gestures (e.g., vibrato, loudness dynamics, timing variations).

We consider our note-oriented view of musical transcription to be a subset of the broader class of problems in computational auditory scene analysis wherein an audio waveform may be musical in nature but may comprise features not easily described by note quanta.

Several commercial software and hardware implementations exist for monophonic transcription of note-based music. (By monophonic we mean that only one note is sounding at a given time. We do not consider stereo waveforms or the transcription of spatial position parameters). Success in this restricted class of waveforms comes from the strong constraint of knowing that only one note can be sounding at any given time. For musical notes with harmonic partial components, the monophonic constraint allows the parameters of fundamental frequency and onset/offset time to be confidently deduced from the time-varying spectrum. Artifacts in the spectrum that may be difficult to interpret can be easily rejected due to the monophonic constraint.

For polyphonic music (i.e., multiple notes may be

sounding at any one time) the problem becomes significantly more challenging. Losing the monophonic constraint (even while still assuming a harmonic partial series) handicaps our ability to interpret spectral representations of our audio waveform. As existing approaches have shown, even the transcription of the most basic note parameters (fundamental frequency, onset time, and offset time) is very difficult in this broader signal class. The transcription problem does not scale well from the monophonic to the polyphonic case, and *ad hoc* or rule-based approaches to fixing the problems that arise are generally unsuccessful (see [7] for a summary).

The recent direction in this field has been to use models of human audition to impose additional structure upon the data, or to use physiologically-motivated algorithms (such as the cochleagram), presumably so we may interpret the audio waveforms in the same way as a human [2]. While we believe that such models are necessary in, for example, separating melodic streams and evaluating the chordal context of notes, we do not believe that such models are necessary for note identification.

2. Track Formation

2.1 Peak Identification

Our transcription algorithm begins with the formation of a time-frequency representation of the acoustic waveform. We use the modal distribution [5] for its improved resolution properties when operating on the class of musical signals (i.e., those signals that are well-modeled as a sum of harmonically related sinusoids). From the time-frequency image, we identify possible partial components at each time instant by searching for local maxima in the spectrum at that time. We use an adaptive thresholding technique to

reject local maxima that appear to be artifacts of the distribution. The instantaneous power and frequency of each partial are estimated by refining the local peak's power and frequency using centroid-based estimators [5]. These estimators have been shown to have better accuracy than parabolic fitting techniques.

Our approach at this point does not differ significantly from existing approaches. However, whereas most algorithms proceed by joining adjacent peaks into tracks using a nearest-neighbor criterion, we look beyond the local neighborhood in an attempt to more robustly tolerate both misidentified peaks and missing peaks. The details of our track formation method are presented in [7]. In the next section we summarize our approach.

2.2 Partial Tracking

We make use of the Kalman filter to estimate the trajectory of a partial based upon observations of its power and frequency at discrete points in time. These observations are the refined power and frequency estimates obtained from identifying local peaks, as described in the previous section. The Kalman filter uses models of partial behavior to extract the optimal estimate of the underlying partial given a set of noisy (and possibly missing) observations. In other words, given a set of peak observations, the Kalman filter identifies the underlying partial that conforms to the model we specify and that best fits the observations.

We model the power evolution of a partial (in decibels) as a piecewise linear function, and the frequency evolution as a constant function. This instantiates our intuition of partials having constant frequency and a power behavior that can be grossly modeled with linear segments. We write our model in the form of a first-order recursion:

$$\begin{aligned} p_k &= p_{k-1} + v_k \\ f_k &= f_{k-1} + w_k \end{aligned} \quad (1)$$

where p_k and f_k are the discrete partial power (in decibels) and frequency (in Hertz) states, respectively. The v_k term represents a "velocity" (i.e., first-difference) power term, and represents an unspecified driving noise process. The velocity term is itself written as:

$$v_k = v_{k-1} + u_k \quad (2)$$

and u_k is an unspecified driving noise process.

To complete our model, we analyzed a large number of isolated single-note recordings from four members of the brass family (trumpet, french horn, tuba, and trombone) [4]. The partials of each note were identified and analyzed for velocity and frequency driving noise statistics. We concluded that the velocity driving noise can be well modeled as an AR(1) process with additional additive white noise:

$$u_k = \alpha_p u_{k-1} + G_p r_k^{(1)} + G_n r_k^{(2)} \quad (3)$$

where α_p and G_p define the pole radius and overall

power of the AR(1) process, and G_n defines the power of the additive white noise process. The terms $r_k^{(1)}$ and $r_k^{(2)}$ are uncorrelated, zero-mean, unity-variance, white Gaussian processes.

Similarly, the frequency driving noise was modeled as an AR(1) process (but with no additional white noise):

$$w_k = \alpha_f w_{k-1} + G_f r_k^{(3)} \quad (4)$$

where α_f and G_f define the pole radius and overall power of the AR(1) process. As above, $r_k^{(3)}$ is a zero-mean, unity variance, white Gaussian process.

The values of the above parameters were derived from our instrument analyses. With this model of partial behavior, we constructed a Kalman filter tracking algorithm to identify partial tracks from peak observations. Please refer to [7] or [8] for the details of this algorithm.

3. Data Association

3.1 Problem Formulation

Our partial tracker has identified a set of tracks $\mathbf{T} = \{T_i; i=1 \dots N_T\}$ where N_T represents the number of tracks. Our goal in transcription is to partition this set of tracks into N_E note events, wherein the tracks themselves suggest the note events that may explain them (and, as a result, the value of N_E as we do not know this quantity beforehand). We define a note event (or note complex) as a 4-tuple:

$$E = (\{T_{k_1}, T_{k_2}, \dots, T_{k_N}\}, f, a, b) \quad (5)$$

where the first element is a set of tracks that lend evidence to the note event, f is the fundamental frequency, and a and b are the start and end times of the note event. A partition (or hypothesis) is defined as an assignment of all N_T tracks to one of the N_E note events or as one of N_F *false-alarm tracks* F_k (i.e., tracks that are not components of any note event):

$$P_i = \left\{ E_1^i, E_2^i, \dots, F_1^i, F_2^i, \dots \right\} \quad (6)$$

The probability that a given partition is a good explanation of the identified partial tracks is defined by:

$$Pr[P_i | \mathbf{T}] = Pr[E_1^i, E_2^i, \dots, F_1^i, F_2^i, \dots | \mathbf{T}] \quad (7)$$

Our task in data association is to search over the set of probable partitions to identify the one with the maximum probability. The note events that comprise this optimum partition represent our transcription result.

Within a partition, we require the exclusive allocation of tracks to note events or to the set of false alarm tracks. We do this for two reasons. First, exclusive allocation means that the joint probability of two different partitions is zero (i.e., $Pr[P_i, P_j | \mathbf{T}] = 0$) hence we need not search over combinations of partitions. Second, exclusive allocation lowers the computational burden of the search process.

We rewrite (7) using Bayes' rule:

$$Pr[P_i|T] = \frac{Pr[T|P_i]Pr[P_i]}{Pr[T]} \quad (8)$$

Since we are only interested in finding the maximum probability over all partitions, the denominator in the above functions as a scaling constant and we can therefore determine the optimum partition number as:

$$i_{opt} = \underset{i}{\operatorname{argmax}} Pr[T|P_i]Pr[P_i] \quad (9)$$

which is the *maximum a posteriori* (MAP) estimate [6].

3.2 Note Models

The quantity $Pr[T|P_i]$ in (9) instantiates the amount of support for a particular hypothesis provided by our tracking results. For simplicity, we initially assume that the components of each hypothesis do not interact (with respect to the identified partial tracks) hence we can write:

$$Pr[T|P_i] = \left(\prod_j Pr[T|E_j^i] \right) \left(\prod_k Pr[T|F_k^i] \right) \quad (10)$$

A more intricate model would consider how combinations of note events and false alarm tracks could lead to mistracking and would give rise to joint probabilities that describe these effects.

The quantity $Pr[T|E_j^i]$ indicates the likelihood that the received tracks support the note event E_j^i . Here we can implement our observations regarding partial properties when the partials are elements of a note complex. For example, $Pr[T|E_j^i]$ should be low if there are no (or few) tracks that are harmonically related to the fundamental frequency of the note E_j^i .

We define $Pr[T|E_j^i]$ to be the product of likelihood functions:

$$Pr[T|E_j^i] = \prod_{n=1}^K L_n(T, E_j^i) \quad (11)$$

Each one of the K likelihood functions L_n describes how well the tracks associated with E_j^i appear to belong together relative to some criterion. We call these criteria *grouping properties*. Useful grouping properties, as well as the numeric parameters that define them, were suggested by note-level analyses of acoustic instrument recordings [7]. In summary, we used $K = 6$ grouping properties:

1. Common onset time
2. Common offset time
3. Harmonicity
4. Low partial support (i.e., we expected all notes to be supported by at least their first and second partials)
5. Partial gap (i.e., missing partials between the lowest and highest partials were not expected)
6. Partial density (i.e., the more partials the better)

The instantiation of the above properties as likelihood functions L_n is described in [7].

The remaining quantity of interest in Equation (10) is the *a posteriori* false alarm probability $Pr[T|F_k^i]$. For simplicity, we ignore the effect of false alarm tracks on the track identification algorithm and write $Pr[T|F_k^i] = 1$.

Finally, we need to consider the *a priori* quantity $Pr[P_i]$ in Equation (9). As before, we assume that the underlying components of the partition are independent and write:

$$Pr[P_i] = \left(\prod_j Pr[E_j^i] \right) \left(\prod_k Pr[F_k^i] \right) \quad (12)$$

The terms $Pr[E_j^i]$ allow us to ignore note events that do not correspond to notes on the 12-tone equal-tempered scale, that are of too brief duration (currently 150 ms), and that are too low or too high in frequency (currently we allow notes in the range C2 through D6, where middle C is denoted C4). The terms $Pr[F_k^i]$ compare a given track against reference values of power and duration and assign low probabilities to tracks that are either too short or of low power. Again, the details of the above are presented in [7].

Assuming independence of note events simplifies the model of Equation (12). However, we can include joint probabilities in our formulation if we want to incorporate knowledge of chords, rhythm, note intervals, and so on.

3.3 Optimum Partition Search

Given the likelihood functions defined by Equations (10) and (12), our goal is to implement Equation (9): find the partition P_i that maximizes the likelihood of receiving the identified tracks. Unfortunately, an exhaustive search over all possible partitions (as suggested by the tracks themselves) is of exponential complexity with respect to the number of tracks. Our approach to searching for the optimum partition borrows from the field of multiple-target radar tracking. In the latter, a technique known as multiple hypothesis tracking (MHT) is used to perform a manageable search over all possible assignments of radar antenna observations to target tracks [1]. We have a similar problem in our data association task, thus we propose analogies between antenna observations and partial tracks, and between target tracks and note events.

The MHT algorithm proceeds by extracting tracks, one at a time, from the list of all identified tracks T , sorted by ascending onset time. For each track, new hypotheses are formed by associating the track with existing note events, using the track to initiate new note events, or considering the track to be a false alarm. The working hypothesis set is replaced with the new set of hypotheses and the process repeats for the next track in T . Should the hypothesis set grow too large, unlikely hypotheses (as evaluated according to the likelihood functions in Equations (10) and (12)) are removed from consideration.

4. Algorithm Evaluation

In [7] we describe a nascent corpus of test cases for quantitative evaluation of automated transcription algorithms. Using the standard MIDI file format, we construct note streams that evaluate various aspects of performance, such as the ability to discern repeated notes, the ability to identify notes of varying relative power, etc. We also include several passages of well known music comprising one to four distinct voices.

Algorithm performance is measured using four primary statistics and two derived statistics. The piano roll notation of both the known score and the transcribed score are used to compute how much of the known score explains the transcription, and vice versa, as well as how many transcribed note onsets can be explained from the score, and vice versa. These four quantities give rise to two summary coefficients that attempt to quantify how “close” two passages are in piano roll notation. One of these summary coefficients corresponds to the note recognition index proposed by Kashino [3]:

$$R = \left[\frac{1}{2} \left(\frac{Right - Wrong}{Total} \right) + \frac{1}{2} \right] \cdot 100\% \quad (13)$$

Right represents the number of notes correctly transcribed, *Wrong* represents the number of spurious notes, and *Total* represents the total number of notes in the original score.

Table 1 shows the note recognition index performance of our algorithm on synthesized musical passages for up to four simultaneous voices. The musical passages are drawn from the score for “Art of the Fugue: Contrapunctus I” by J.S. Bach and rendered to acoustic waveforms using the trumpet, french horn, trombone, and tuba voices of a Kurzweil MIDI synthesizer. The reader is invited to view the web page of the first author for audio examples.

The results indicate decreasing performance with an increasing number of voices, as is to be expected. The magnitude of the numbers, however, suggests that our algorithm exhibits comparable performance relative to existing algorithms. Differences in the musical passages and timbres used to evaluate other algorithms preclude any more conclusive results. Hopefully, our corpus of test cases will be expanded, improved, and eventually adopted by the musical transcription community as a means of quantitatively comparing the performance of different algorithms.

5. Conclusions

We have presented a new approach to musical transcription that focuses upon data models and the information provided only by data analysis. Traditional *ad hoc* techniques of nearest-neighbor partial formation are replaced with a model-driven tracking algorithm based upon Kalman filtering applied to a high resolution modal time-frequency distribution. Similar rule-based and intuitive techniques of data

Number of Voices	Note Recognition Index
1	100%
2	96.2%
3	84.8%
4	79.5%

Table 1: Note recognition index scores on four passages of multi-voice, musical material

association are replaced by a more objective framework based on models and analyses of partial behavior in a note complex. Combined with the MHT search algorithm, we have implemented a transcription algorithm whose performance is comparable to that of existing algorithms. The beginnings of a new suite of test cases have been proposed in order to better quantify future comparisons of transcription algorithms.

Even with this level of performance, we believe that our algorithm is still quite naive in many respects and can act as a foundation upon which many new research directions can be built. This stands in contrast to various existing algorithms which appear to leave little room for evolution.

6. References

- [1] Blackman, S.S. (1986). *Multiple-Target Tracking with Radar Applications*, Artech House, Norwood, Massachusetts.
- [2] Brown, G.J., and M. Cooke (1994). “Perceptual Grouping of Musical Sounds: A Computational Model,” *Journal of New Music Research*, vol. 23, pp. 107-132.
- [3] Kashino, K., K. Nakadai, T. Kinoshita, and H. Tanaka (1995). “Organization of Hierarchical Perceptual Sounds,” *Proc. 14th Intl. Joint Conf. on Artificial Intelligence*, vol. 1, pp. 158-164.
- [4] Opolko, F., and J. Wapnick (1989). McGill University Master Samples (MUMS), 11 CD-ROM set, Faculty of Music, McGill University, Montreal, Canada.
- [5] Pielemeier, W. J., Wakefield, G. H., and Simoni, M. (1996). “Time-frequency analysis of musical signals,” *Proc. IEEE*, Vol. 84(9), 1216-1230.
- [6] Stark, H. and J.W. Woods (1986). *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice-Hall, New Jersey.
- [7] Sterian, A. (1999) *Model-Based Segmentation of Time-Frequency Images for Musical Transcription*. Ph.D. Dissertation, University of Michigan.
- [8] Sterian, A. and Wakefield, G.H. (1998) “A model-based approach to partial tracking for musical transcription,” *Proc. of SPIE Conf.*, San Diego, California, July, 1998.