CENTER FOR COMPUTER RESEARCH IN MUSIC AND ACOUSTICS OCTOBER 1985

Department of Music
Report No. STAN-M-29

Techniques for Note Identification in Polyphonic Music

by

Chris Chafe, David Jaffe, Kyle Kashima, Bernard Mont-Reynaud, Julius Smith

Research sponsored in part by

The National Science Foundation and Xerox Corporation

CCRMA
DEPARTMENT OF MUSIC
Stanford University
Stanford, California 94305

Techniques for Note Identification in Polyphonic Music

Chris Chafe, David Jaffe, Kyle Kashima, Bernard Mont-Reynaud, Julius Smith

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

ABSTRACT

A set of techniques for segmentation of polyphonic piano signals is described. Event detection begins with time-domain techniques. Note identification is accomplished on spectral surfaces built up via the "Bounded-Q" frequency transform. The transform method answers the need for frequency discrimination which can pull apart close-lying partials in polyphonic textures. Metrical context is used to aid signal processing algorithms after early searches reveal an outline of events. The approach of reconciling the raw results of detection algorithms with expectations from context eliminates the need for critically tuned detection thresholds.

INTRODUCTION

A fundamental task in analyzing performed music is to extract every note played, identifying timings, pitch and dynamic information and other parameters. Our goal is a competent system that can be an important tool for the study of real performance as well as for applications requiring tracking of live musicians, automatic transcription, and segmentation of digital audio recordings. Past research at CCRMA has addressed the monophonic version of this problem. Turning to the analysis of polyphonic music has presented new challenges and led to several new techniques.

The experimental analysis system relies on a hierarchy of descriptions of the input. Beginning with time and frequency domain representations, it moves through successive stages of abstraction to obtain musical representations such as scale pitches, note values and so on. The topic of this paper is the acoustic analysis used for the initial task of finding notes in the raw signal. We will discuss how knowledge about the instrument's acoustics can be combined with context-building capabilities to increase the system's level of performance.

The acoustic analysis relies both on specially adapted signal processing tools, and on data reduction ideas derived from knowledge of the source's acoustics. Signal processing methods are chosen for maximum performance in terms of bandwidth-vs-time tradeoffs and computation cost. These include applications of the Bounded-Q frequency transform [Kashima 1985] and time-domain techniques [Schloss 1985]. Event identification is accomplished with knowledge about the instrument, for example, in grouping peaks as partials which belong to the same pitch. As metrical context becomes available inconsistencies and ambiguities in the first-pass results are identified.

The system combines information at different stages to form and evaluate hypotheses about events, partials and

This work was supported by the National Science Foundation under Contracts NSF MCS-8012476 and DCR-8214350. Xerox Corporation has provided computer equipment for the research.

notes. Feedback paths are being developed in which more abstract knowledge about the signal is used to oversee the basic feature detectors. A control structure will configure the calling sequence of analytic sub-tasks. The current environment provides a laboratory for studying the adaptive manoeuvers required of a fully automatic and flexible control structure. Modules designed to meet the following needs have been incorporated:

- **Signal Transformation** High resolution in the frequency domain is required to distinguish voices in a polyhonic texture.
- **Event Detection** Time domain techniques identify points of interest in the signal.
- **Note Modelling** Detected events are examined. Chor are broken into notes and note partials are tracked in time.
- **Generation of Early Context** Event timings are used to build a metrical grid. Patterns suggest events that are rhythmically weak.
- **Recursive Sharpening** Tracked notes are eliminated from the original signal and further passes attempt to find hidden events.

FRONT-END SIGNAL TRANSFORMATION

A fixed transformation technique produces the data base on which event detection and modelling is performed. In a sense, this corresponds to the outermost periphery of the ear-brain system. At the cochlea the signal is transformed into information that is presented to first-level abstractors in the pathway. Cues for musical events in temporal, frequency and phase representations become identified in subsequent stages. The following front-end procedure creates a raw data base for our system representing the signal in these three domains.

Bounded-Q Frequency Transform

A specially adapted version of the Constant-Q frequency transform is used to obtain time-varying spectral data. Previoı work with monophonic musical input was accomplished using a straight FFT-based method which was sufficient to resolve single source harmonically-related partials [Foster 1982]. Greater frequency resolution is needed when discriminating multiple notes which may be played as close as a semitone apart. The problems in extending the original method can be shown by a calculation of requirements at the extreme ends of the range of interest.

A straight FFT yields magnitude data in a set of equally-spaced frequency bins. There are half as many bins as there are time samples taken as input (for real signals negative frequency components duplicate the positive frequency components). The width of a bin, ΔF is equal to the Nyquist rate divided by the number of bins. The resolution problem for the polyphonic case is worst at the low end of the range. Here, an extremely high-resolution straight FFT is needed to separate two tones a semitone apart, which is expensive to compute. In the second octave of the piano (for the pitch A1) the interval is 3.270 Hz. Resolving any pair of peaks requires at least one intermediate bin, indicating a minimum ΔF of 1.635 Hz. Given our analysis system's sampling rate of 25.6 kHz, the time window would be 15657 samples wide (rounded up to 16384 points for a convenient FFT block size). Since we calculate an FFT every 5 msec for minimally sufficient temporal resolution, the straight FFT method is going to produce 128 frequency points for every time sample, or an overhead of 3.276 Megasamples/second.

At the highest frequencies of interest, for example the 4th partial of a pitch 5 octaves up (the partial equals A6), frequency is greatly oversampled, with 512 bins per semitone, creating a huge amount of useless data. The basic problem is a mismatch between the logarithmic continuum of interest and the linear FFT method. This is solved by choosing a technique in which ΔF varies proportionally with frequency, such as the Constant-Q transform, in which bin widths are exponentially spaced. The implementation used in the current work is actually a hybrid technique that makes efficient use of the straight, or linear, FFT to produce a constant number of bins by octave. In the Bounded-Q method, a window size is selected which is only long enough to resolve partials in the upper octave of the signal. The signal is repeatedly transformed and down-sampled by an octave. Each octave decimation produces a set of bins with doubling resolution [Kashima 1985].

The BQFT implementation benefits efficiency-wise from being able to do its low-pass decimation filter as a multiplication in the frequency domain (instead of time domain convolution), and can be quite economical especially when the signal is hopped by full windows. Using full hops, the 5 msec temporal resolution constraint dictates an FFT size of 128 points yielding 64 bin resolution in each successive octave decimation. Within the best resolved upper half of each octave decimation the ratio of bins per semitone varies between a low bound of 1.86 and an upper bound of 3.52. Where the straight FFT above produced 128 points for every input sample, the BQFT method yields one point per sample.

Each octave decimation of the BQFT produces a time-varying spectrum of the signal in the form of a matrix. It contains magnitude information which has been thresholded above the level of transform noise and converted to dB format. A second matrix containing phase information for each bin is output as well, and along with the original time domain signal these comprise the raw data base for higher-level processing. The data base structure and its access functions are described in the Appendix.

Pre-emphasis

The input spectrum is equalized to retain high frequency information. Without pre-emphasis, higher partials are lost when thresholding spectral data to remove the transform noise floor. A pre-emphasis filter is designed which flattens out a representative average spectral envelope of the input [Smith 1983]. The energy in the input signal is modelled using a 3-pole linear predictor. A filter is then fit to the model's inverse curve and used to pre-emphasize the entire input sound file before executing the above transform.

EVENT DETECTION

Time domain amplitude information is used to identify points of interest in the transformed data base. A simple event detecting algorithm is paired with a second process which generates metrical context. A third one, the "listener" adjusts detection sensitivity until a sufficiently strong metrical scheme comes into focus. It guides the entire event detection task, signalling satisfactory completion.

An assumption is made about the source: The musical performance is presumed to be an elaboration of some meter (albeit expressive, since there is no restriction on tempo variation or rubato) and to be rich in rhythmic patterns. At this early stage only timings are available to establish a sense of musical context. These rhythmic events create a flat, pitchless projection of the piece, where even chords are indistinguishable from single notes.

Surfboard Amplitude Segmentation

The event detector accomplishes segmentation according to amplitude. By detecting abrupt attacks, the method is suited to percussive instruments in general, e.g. struck or plucked strings, drums, bells, etc. Using the low-passed and down-sampled version of the amplitude envelope of the signal, a linear-regression "surfboard" runs along its crest marking significant rises of slope [Schloss 1985]. A set of parameters determine detection sensitivity. The accuracy of the detector is limited to the sampling interval of the amplitude envelope (5 msec). Performance, in terms of number of correct detections, is better than 95% for the piano.

Smooth overlappping tones present the greatest difficulty. Strong ringing events sometimes mask the entrance of a relatively weak subsequent note. With the amplitude method, keyboard-style slurs leave slurred tones lying undetectable in the shadow of the elongated preceeding tones. In essence, these are local polyphonic moments which can occur in a single musical line. After the primary note has been spectrally modelled, a subsequent stage of the analysis can uncover the missing note, as is explained below.

Employing Early Context

Early on, the analysis is guided by higher level processing. Using only a population of first-pass detections, context is used to provide hints indicating grossly underdetected signals. Further hints help to eliminate spurious detections and identify likely points of interest to probe with further signal processing. The linked-in musical analysis capabilities are described in detail elsewhere.

Metrical context is determined using musical analysis tools developed in a parallel effort to the acoustic analysis presented here [Chafe 1982, Mont-Reynaud et al. 1984, Mont-Reynaud and Goldstein 1985, Mont-Reynaud 1985]. Briefly, the system derives a pulse from the raw event timings and tracks it across tempo variations. Timings are then "normalized," effectively the same as flattening out expressive fluctuations in the performance. Each detected event is assigned a metrically related musical duration. A heuristic measure is available which represents the confidence with which the task was completed. The heuristic can be examined to see how it varies with different detection sensitivity values.

Three detection parameters can be altered. By comparing the raw detected results at different threshold levels, strong points can be distinguished by their number of occurences. We are attempting to combine this kind of "grey-scale" information with context heuristics to automatically focus the following parameters.

- **Slope Threshold** This parameter determines a minimum rise in slope to be considered as a possible attack. Lowering the threshold detects more events.
- **Surfboard Length** The surfboard length is shortened to change the window size used in the linear regression, essentially to weaken its low pass effect.

- **Refractory Interval** A minimum event duration is specified. If shortened, attacks can follow each other more closely in time.

The second use of rhythmic context is to identify spurious detections. All detections are compared as to the number of rhythmic patterns to which they belong. Spurious detections contribute to fewer patterns in a pattern-rich musical score.

The patterns are discovered by a search through the musical, metrically normalized, durations determined above. Patterns which occur are identified and counted. Events are weighted by the number of patterns to which they belong, and those which are more or less loners are marked as questionable. All but the most ambiguous loners are separable from their peers via a heuristic measure and can be ignored.

Certain kinds of patterns are strong enough to suggest missing detections. As an example, a solid run of fast equal notes might be broken midway by a single missing down beat. If unbroken runs are typical of the example it is reasonable to infer that the down beat went undetected. A likely attack time is then inferred from the meter and added to the list of detections.

NOTE MODELLING

Events can be identified by building note models from their spectral components. The models are based on knowledge about the acoustics of the source instrument. For the piano, particular features which can be observed are stretched partials, smooth spectral rolloff and transient envelope shapes.

Picking Peaks

A list of events is now available, which have been identified either directly by event detection or suggested via rhythmic context. The spectral content of each event can be established by identifying peaks in the transform matrix. A frequency vs. amplitude slice, in the form of a set of instantaneous bin amplitudes is returned from the matrix at the indicated time, or slightly later if it is important that the event has stabilized. A latency of 100 msec ensures most transient phenomena have subsided.

In order to determine peaks with the best possible frequency resolution, a vector is formed which consists of a composite of bin amplitudes taken from the top octave of each octave decimation. These are joined into a single vector whose bins span the range from 1/2 the Nyquist rate of the lowest decimation to the Nyquist rate of the original signal. A vector of 256 points covers the range from 50 Hz. to 12.8 kHz.

Peakpicking is accomplished by searching the vector for groups of points which exceed a threshold. Each group yields a single maximum peak location which is converted into a frequency. A table provides bin center frequencies where the different octave decimations have been taken into account.

Grouping Partials

Peak data is analyzed to find groups which cohere as partials. The method starts with the lowest peak, assumes it's a fundamental and checks each above-lying peak to determine whether it could be a partial. If matches are found they are marked and eliminated from further searches. If no match is found the fundamental candidate is discarded. The search continues until no peaks remain. Chords within the peak data are separated into multiple fundamental candidates.

Matching takes into account the overtone properties of piano strings. For each fundamental, a template is generated against which peaks can be compared. It consists of a series that approximates the stretching of piano harmonics for the given pitch and register.

The search method is simple, effective and somewhat flawed. Omissions occur where one fundamental falls on the partial of another, e.g. an octave or an octave and a fifth interval. A filter for uncovering the higher tone is constructed by adding up partial amplitudes associated with each possible hidden fundamental, e.g. an octave's partials: [2, 4, 6, 8...]. Significant weighting in a such a set identifies the hidden note.

A second weakness results from unlucky coincidences between noise induced peaks. Such spuriously identified fundamentals can be identified because they are short-lived, and disappear when tracked through time.

Tracking Note Models

Starting from the detected event's attack point, each fundamental and its associated partials are tracked forward and backward to refine onset time and determine the note's duration. A minimum duration of 50 msec. is required for a candidate to qualify for true notehood. Shorter durations are kicked out as spurious fundamentals.

Percussive notes can disappear in two ways: either as freeringing exponential decays or with more abrupt damping.

The tracking mechanism infers one or the other case from the time-varying spectral envelope that it creates. With BQFT data lower partial timings are significantly blurred due to poor time resolution from the longer effective FFT window in the lower octave decimations. By studying upper partials a match is made either to the free-ringing case or the damped one. If the lower partials cannot be clearly resolved in higher decimation levels (to gain time resolution) their fate is extrapolated from their expected behavior in the chosen situation.

Looking Further and Music Minus One

Once timings for all apparent notes are known, a second matrix is constructed in which partials of found notes have been erased. Partials are subtracted from their peak bin along with neighboring skirts, if any. The blanking interval is limited to 3 bins.

This new matrix is used to conduct a second search for missing events, peaks, note candidates and so on. Instead of the original time-domain amplitude envelope, the surfer is run on an envelope created by summing all remaining bins for each point in time in the composite vector surface.

Note models available from each successive analysis pass can be used to drive two kinds of re-synthesis. The models can drive additive synthesis in a straight-forward way, with one oscillator per partial using the time-varying amplitude envelopes found by tracking. A second method uses the models to subtract notes from the original signal using an FFT-based filter. With the signal transformed into the frequency domain, the spectral components of one note or any combination of notes (a single voice, for instance) is removed and the modified spectrum is inverse transformed.

IMPLEMENTATION

The research system is currently implemented across several machines, in a networked environment. Digitizing is accomplished with a portable PCM-F1 system in 16 bits at 44.1 kHz. The recordings are digitally transferred from playback to the file system of CCRMA's time sharing facility, a Foonly F4 computer. Programs written in SAIL for the Foonly handle the front-end signal processing including sound file editing, sample rate conversion, pre-emphasis, the BQFT and amplitude envelope extraction. The facility also offers hardware synthesis capabilities and high-resolution manuscript printing. These are often used for reviewing results at different stages of the analysis.

The various representations of the signal are sent via Ethernet to the file system of a Xerox 1108 which runs INTERLISP-D. This machine is primarily used for development of the acoustic analysis system. A data base with access functions for BQFT data allows random access from files. These functions are optimized for sequential operations. In addition, interpolations of several kinds are available for "in-betweening" in different dimensions (see Appendix).

Processing which involves musical context is being developed on a second 1108. Operations which build or evaluate context can be handled as requests run on this sister processor. One machine can evaluate functions or obtain data resident on the other. From this capability we are developing a flexible environment for the system's control structure. An analysis system is evolving with different layers of the system running in parallel and communicating in the networked environment. UNIX-like pipes are the software channels which link points in the system hierarchy within and between machines.

SUMMARY

The modules presented are building blocks of a future fully-automated analysis system. We are integrating them into an environment with software for recognition of many other musical contructs. This will extend the principle that perfomance of the initial detection task is improved by building expectation into the system. The current use of rhythmic recognition capabilities will be augmented by parsers for melodic and harmonic constructs.

Brute force detection schemes are less likely to succeed in analyzing polyphonic musical textures than they are for monophonic input. The balance between undertection and overdetection becomes more difficult to achieve. We have taken the approach that conservative, less sensitive thresholds can be augmented by using metric information to point to likely event positions. Spurious events from overdetection are conversely reducible by evaluating rhythmic conformity. Thus, threshold tuning becomes less critical. Further sharpening results from modelling notes according to expectations of acoustic features in the source.

The musical applications of this work potentially involve many instruments and musical textures. Non-percussive instruments require other techniques soon to be incorporated. We can imagine for example, amplitude "windsurfers" that are keyed to recognize the different classes of note onsets of wind instruments. In any case, pure time-domain techniques will generally be less effective for such cases than they are for the piano. A detector in the frequency-domain is being engineered based on the Maximum A-Prior Line Estimation (MAPLE) method for tracking significant spectral lines in the matrix [Wolcin 1980].

APPENDIX

Design of the BQFT Data Base and Access Functions

The BQFT data base is organized to provide random access as well as efficient sequential access with local backtracking. The access is demand-driven. The basic unit is the "block", an array of FFTsize pieces of data of a given type. The type is either log-magnitude or phase for a given decimation level. These types are called "basic types" because the data comes directly from a binary file. There are also "derived types" which are computed from the basic types on the basis of demand. The derived types currently supported are linear-magnitude, real and imaginary.

For each basic type, the data is stored in memory in a circular buffer which contains contiguous blocks of data. Each basic type also has some local state indicating the current contents of the buffer along with information about the previous access and the resolution of the data. The buffer is updated from the binary file only when a block which is not currently in memory is requested. When this occurs, the requested data is positioned in the buffer according to a user-settable variable and only as much data as is necessary to fill out the buffer is read in. Thus the ratio of future data to past data can be set by the user to an optimal value for his access pattern, presumably a value which triggers the fewest number of disk accesses. Note that when a buffer is updated, it is only the given basic type for the given decimation level which is altered. Thus it is quite normal for different decimation levels to have in memory, at the same time, data from different times in the signal. The principal motivation for the circular buffer arrangement is the kind of access pattern required to do sinc function interpolation (described below), in which local backtracking is required in the context of a general forward motion in time.

A single pointer provides a handle to the entire BQFT data base for a given signal. This pointer is passed to various access functions to return the appropriate data, allowing several signals to be in memory at once without interfering with one another. A header file gives information for the signal as a whole. Any decimation level or type of data may be omitted from the data base. Thus partial BQFT representations are supported.

The access functions are arranged hierarchically, so that functions dealing with higher levels of abstraction are defined in terms of functions dealing more closely with the raw data. Each access function returns a vector along either the frequency axis or the time axis of the BQFT data base. This vector is a set of data points of a given basic type and decimation level. The lowest level access functions return the basic frequency axis unit, the FFT block, or the basic time axis unit, the channel. A block consists of only those frequency points which were evaluated in the original FFT for a selected decimation level. Similarly, a channel consists of only those times at which an FFT was evaluated for the selected decimation level.

Higher level functions provide two forms of interpolation of blocks and channels. First, it is possible to interpolate to derive a block lying between two blocks or channels. Secondly, it is possible to use interpolation to resample up or down a block or channel to derive a new resolution in that domain. Of course, it is also possible to perform both of these interpolations. For example, a vector consisting of a block at an arbitrary time with an arbitrary resolution can be derived.

The interpolation may be done in one of two ways. Linear interpolation provides a quick approximation and is especially useful in generating graphic plots of BQFT surfaces Sinc function interpolation requres more computation but gives the true value between two frequency points, assuming the signal is time-limited, or between two time points, assuming the signal is band-limited. Sinc interpolation is optimal and is superior to Lagrange interpolation for acoustic signals [Schafer and Rabiner, 1973]. The sinc interpolation between two data points is evaluated by weighting the neighbors of the desired point by a sinc function, summing these values and dividing by π. The sinc function chosen is the one which is the transform of the rectangular window in the opposite domain and is centered at the desired point.

Sinc function interpolation requires a complex representation of the signal, so functions which return a complex vector are provided. Furthermore, the sinc interpolation works on a real/imaginary representation of the complex data rather than a polar representation. It can thus serve as an illustration of how derived types are computed on the basis of demand. The sinc interpolation function knows it needs a real/imaginary representation so it calls a Generate-Real and a Generate-Imaginary function. Each of these looks to see if the real or imaginary data is already present. If not it synthesizes it. First, it needs to call a Generate-Linear-

Magnitude function. This function, in turn, looks to see if a linear-magnitude representation exists. If not, it synthesizes it. Finally, the sinc function has the data it needs.

Another function answers the question of whether two neighboring frequency points with some energy represent a single or multiple sinusoidal components. An easy solution consists of looking at the phase of the two frequencies, tracking them over several time intervals, and differentiating each of these vectors with respect to time. This produces a vector tracking frequency for each channel. If these vectors are nearly identical, the information corresponds to a single sinusoid.

References

[1] K. Kashima, "The Bounded-Q Frequency Transform," *Department of Music Technical Report STAN-M-28.*, 1985.

[2] A. Schloss, "On the Automatic Transcription of Percussive Music," Ph.D. Thesis, Department of Speech and Hearing, Stanford University, Stanford University, Stanford California, June 1985. *Department of Music Technical Report STAN-M-27.*,

[3] S. Foster, J. Rockmore and W. Schloss. "Toward an Intelligent Editor of Digital Audio: Signal Processing Methods," *Computer Music Journal,* vol. 6, no. 1, 1982.

[4] J.O. Smith, "Tehcniques for Digital Filter Design and System Identification with Application to the Violin," Ph.D. Thesis, Department of Electrical Engineering, Stanford University, Stanford California, June 1983. *Department of Music Technical Report STAN-M-14.*,

[5] C. Chafe, B. Mont-Reynaud and L. Rush, "Toward an Intelligent Editor of Digital Audio: Recognition of Musical Constructs," *Computer Music Journal,* vol. 6, no. 1, 1982.

[6] B. Mont-Reynaud, M. Goldstein, "On Finding Rhythmic and Melodic Patterns in Musical Lines," *in this volume, ICMC 1985,* also as *Department of Music Technical Report STAN-M-30.*, 1985.

[7] B. Mont-Reynaud, et al. " Intelligent Systems for the Analysis of Digitized Acoustic Signals, Final Report. ," *Department of Music Technical Report STAN-M-15.*, 1984.

[8] B. Mont-Reynaud, et al. " Problem-Solving Strategies in a Music Transcription System ," *Proceedings of the IJCAI, 1985.*

[9] J. Wolcin, "Maximum A Priori Estimation of Narrow-Band Signal Parameters," *JASA,* vol. 80, no. 1, pp. 174-178, 1980.

[10] R. Schafer and L. Rabiner "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE, vol. 61, pp 692-702, June 1973.*, 1985.