INTEGRATION OF DSP ALGORITHMS AND MUSICAL CONSTRAINTS FOR THE SEPARATION OF PARTIALS IN POLYPHONIC MUSIC

Ramamurthy Mani and S. Hamid Nawab

ECE Department, Boston University
Boston, MA 02215
email: hamid@bu.edu

ABSTRACT

We illustrate how high-level knowledge from the musical domain may be integrated with sophisticated signal processing algorithms within a system for separating possibly overlapping partial frequency components from polyphonic music. Musical knowledge utilized in our system is in the form of constraints on the time-frequency behaviors of musical signals such as the frequency locations of notes on the western musical scale and the presence or absence of vibrato in each note. For any given signal scenario, these constraints help in appropriately initializing and adjusting a set of algorithms for constant-Q processing, spectral peak picking, and multihypothesis tracking through Kalman filtering. As demonstrated by the evaluation of our system with a variety of signals containing two simultaneously played violin notes, the application of these algorithms results in the accurate separation of individual partials.

1. INTRODUCTION

The problem of isolating individual partial frequency components of simultaneously occurring musical notes arises in a variety of applications such as automatic music transcription, intelligent music editing, pitch tracking, and rhythm tracking [1, 2]. Signal processing solutions involving a combination of time-frequency analysis, spectral peak picking, and tracking may be brought to bear on this problem. While solutions that rely on time-frequency analysis performed using uniform or constant-Q filterbanks [3] are suitable for isolating partials within single musical notes, these solutions generally fail when applied to polyphonic music. The primary cause for failure is the fixed tradeoff between time and frequency resolution at any given point in the time-frequency plane.

A remedy for the drawbacks of fixed resolution time-frequency analysis techniques is to adapt the analysis filters at each point in the time-frequency plane in a data-dependent manner. One such adaptation strategy developed by Parks and Jones [4] involves choosing at each time-frequency point a filter that maximizes a kurtosis-like local energy concentration measure. This strategy suffers from the drawback that it is computationally very expensive due to the exhaustive search over $\mathcal{O}(10^2)$ candidate filters performed at each time-frequency point. It should be noted that for each candidate filter a new time-frequency analysis has to be performed over a subregion surrounding the time-frequency point under consideration. Another major drawback in using the concentration measure is that it does not utilize any knowledge about the time-frequency behaviors of musical signals (such as the harmonic relationship between partials of a note).

We have designed and implemented a system that adopts an alternative analysis filter adaptation strategy based upon utilizing higher level musical knowledge. This strategy relies on an iterative procedure that follows the Integrated Processing and Understanding of Signals (IPUS) paradigm [6]. We describe the components of our IPUS-based system in Section 3 after providing a brief overview of the separation problem in Section 2. Details pertaining to the implementation of our system and its performance for signals with two simultaneous violin notes are presented are presented in Section 4. Finally, we demonstrate in Section 5 that even with the added advantage of using musical knowledge, time-frequency analysis performed using our system is still an order of magnitude computationally more efficient than the kurtosis-based filter adaptation strategy.

2. SEPARATION PROBLEM

The problem of separating individual musical partial frequency components from polyphonic music may be formally described by considering the following commonly-used model for a musical note with M partials:

$$x(t) = \sum_{k=1}^{M} a_k(t) cos\left[\omega_k(t) + \phi_k\right]. \qquad (1)$$

Here, $a_k(t)$, $\omega_k(t)$, and ϕ_k are the instantaneous amplitude, instantaneous frequency, and instantaneous phase, respectively, of the k-th partial frequency component of $x(t)$. The separation problem involves estimating $a_k(t)$ and $\omega_k(t)$ for each partial within a mixture of simultaneously occurring musical notes. In our research, we have assumed that there are at most two notes occurring simultaneously at any given time. This assumption was made on the basis that the separation problem is fairly complex even in two note scenarios and offers ample scope for demonstrating the benefits of our novel system.

3. SYSTEM FOR SEPARATING PARTIALS

We have addressed the separation problem through the development of a system that utilizes a combination of time-frequency analysis, spectral peak picking, and tracking. Time-frequency

This work was sponsored in part by USAF Rome Laboratory under Contract No. F30602-95-C-0204 and in part by the Department of the Navy, Office of the Chief of Naval Research, contract number N00014-93-1-0686 under the Advanced Research Projects Agency's RASSP program.

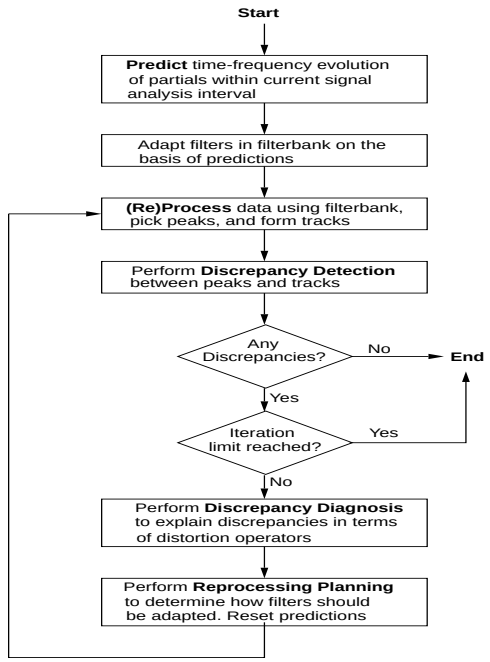

Figure 1: IPUS-based strategy for adapting the filters used in the time-frequency analysis of polyphonic music signals.

analysis is performed using a filterbank to resolve spectral contributions from each partial in the time-frequency plane. The analysis filters in the filterbank are adapted in a manner which allows the separation of partials that are possibly overlapping in the time-frequency plane. Significant spectral peaks are identified in the filterbank outputs and the peaks are then used to form time-frequency-amplitude trajectories corresponding to the partials present in the signal. The *multihypothesis tracking* algorithm from the area of *multitarget tracking* [7] is utilized in forming these trajectories. The tracking algorithm involves modeling the time-frequency-amplitude evolution of a partial by means of state equations, associating peaks with partials on the basis of this model, and finally utilizing Kalman filtering to form state trajectories for each partial from its associated peaks.

The key component of our overall system is the strategy utilized for adapting the filters in the time-frequency analysis filterbank. We have designed this strategy to follow the IPUS blackboard paradigm. The IPUS paradigm facilitates the systematic incorporation of knowledge regarding local and global constraints on the time-frequency behaviors of musical signals into the filter adaptation process. Within our IPUS-based strategy, the polyphonic musical signal is processed in a block-wise manner through the iterative procedure outlined in Figure 1. We now describe the major steps within an iteration of this procedure.

3.1. Prediction

Prediction is the process of using previously identified partials in order to hypothesize their time-frequency evolution in the current processing block. We perform prediction on a partial by first carrying out a least-squares straight-line fit to its corresponding track. The standard deviation of the partial's instantaneous frequencies is also computed. The straight line is then extrapolated into the current processing block and a frequency subregion which is 3 standard deviations to either side of this line is identified. Assuming a normal distribution for the partial's instantaneous frequency, this subregion represents the 99% confidence region for the evolution of the trajectory corresponding to the partial.

3.2. Adjusted Constant-Q Processing

The predicted time-frequency behavior of partials in the current processing block is utilized to alter the analysis filters in a constant-Q filterbank with Gaussian filters. Gaussian filters were chosen on the basis of the well-known property that they have the least time-frequency uncertainty. The center frequencies of the filters in the filterbank are uniformly spaced along the frequency axis and the impulse response of the i-th filter is given by:

$$h_i[n] = \begin{cases} A \exp\{-\alpha n^2\} \exp\left\{j \frac{2\pi f_i}{f_s} n\right\}, & |n| \leq n_i \\ 0, & |n| > n_i \end{cases} . \quad (2)$$

Here f_i is the center frequency of the filter, f_s is the sampling rate, n_i is the time index before which the magnitude of the impulse response decays to a small value ϵ, A is a scaling factor which ensures that the filter has unit energy, and α is a parameter that controls the bandwidth of the filter. In their default setting, the α parameter for the filters are adjusted to follow a constant-Q rule. We utilized the filter's frequency response [8] to arrive at the following relation:

$$\alpha = \frac{1}{2\ln(\epsilon)} \left[\frac{\pi f_i}{Q f_s}\right]^2 , \quad (3)$$

where Q is the Q-factor of the filterbank (which in our case is 34 because it corresponds to quarter-tone spacing [3]). Once predictions have been carried out in the current processing block, the default α settings of the filters are adjusted such that they have enough frequency resolution to separate the predicted partials.

3.3. Discrepancy Detection

Discrepancy detection is the process of identifying mismatches between the predictions and the tracks formed from the results of adjusted constant-Q processing. During this process, we only need to compare a prediction with tracks which are within its time-frequency vicinity. Therefore, we perform "clustering" to first group tracks and predictions which are in close spectral proximity in the current analysis interval. Discrepancy detection is then performed within each cluster. Besides the obvious benefit that clustering simplifies the process of discrepancy detection, it also greatly aids the diagnosis or explanation of these discrepancies. This is because discrepancies which are close to each other in the time-frequency plane generally have a common cause.

During the process of discrepancy detection, the following three different types of discrepancies are identified within each cluster: (a) *Missing data:* Discrepancy when no matching track is found for a prediction. (b) *Missing prediction:* Discrepancy when no matching prediction is found for a track. (c) *Missing consistency:* Discrepancy when a prediction-track pair do not match in terms of statistics such as frequency variance or frequency modulation rate.

3.4. Discrepancy Diagnosis

The process of discrepancy diagnosis attributes probable causes to the discrepancies identified in the current processing block. Dis-

crepancies within a cluster are all diagnosed simultaneously since they are most likely to have a common cause. For each cluster, the diagnostic process involves the utilization of a set of "distortion operators" to provide a mapping between an "initial state" consisting of the predictions within the cluster and a "goal state" made up of the tracks within the cluster. Distortion operators serve the twofold purpose of identifying missing information in the initial state and identifying improper processing of the data. In particular these operators are used to indicate (a) the commencement and termination of partials, (b) filters with improper time-frequency resolution tradeoff or filters that are improperly matched to partials with frequency modulation (vibrato), and (c) improper processing during spectral peak picking. The diagnostic process now involves using a combination of these distortion operators to explain the discrepancies with a cluster.

The central idea used in diagnosis involves hypothesizing the true nature of partials within a cluster on the basis of the processed data. This requires knowledge about all possible cluster scenarios that can arise in the context of musical signals with at most two notes. Through an analysis of two-note signals, we have identified 45 different cluster scenarios [5]. It is now the job of our four-step diagnostic process to prune this list of 45 scenarios down to a few specific scenarios as the likely candidates for expressing the true nature of the partials within a cluster. This process, which is outlined in Figure 2, uses criteria such as number of predictions, the nature of the discrepant cluster, and the number of notes found in the signal. It should be noted that the numbers shown in Figure 2 at the ends of the branches correspond to index numbers for our cluster scenarios. The top candidate from the list of pruned scenarios is chosen as the most likely scenario if the cluster is being diagnosed for the first time. During subsequent diagnoses, different candidate scenarios may be chosen if discrepancies still remain.

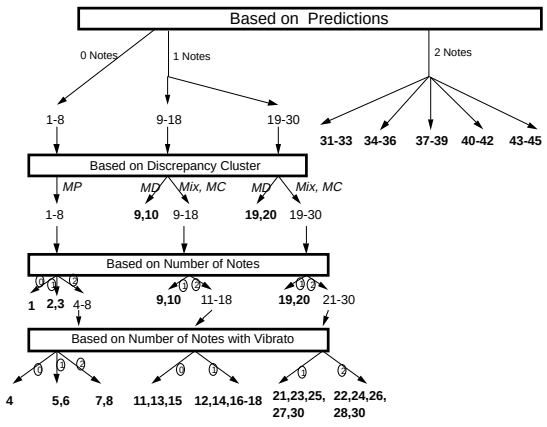

Figure 2: Diagnostic Process.

Based upon the chosen scenario, we hypothesize a set of possible distortion operators that could potentially cause the discrepancies found in the cluster. This is performed by using a lookup table that is indexed by a combination of the number of predictions in the cluster, the number of partials within the chosen scenario, the number of partials with vibrato in the chosen scenario and the types of discrepancies identified in the cluster.

3.5. Reprocessing Planning and Reprocessing

The sequence of diagnostic operators associated with a cluster are used to choose a suitable reprocessing plan from a library of plans. The execution of this plan results in a modification of the predictions within the cluster (if necessary) and a reprocessing of the data (if necessary) through an altered set of analysis filters, an altered peak picking algorithm, and finally the tracking algorithm.

Once reprocessing has been completed, the processes of discrepancy detection, diagnosis, and reprocessing are again carried out on the cluster. This is repeated until the cluster is devoid of discrepancies or until five iterations of the reprocessing loop are performed on the discrepant cluster. Persisting discrepancies are dealt with by replacing the tracks with plausible time-frequency trajectories that are obtained on the basis of other harmonically related partials present in the processing block.

4. IMPLEMENTATION

We have implemented the system described in the previous section within the recently introduced IPUS C++ Platform (ICP) [9]. ICP is a software environment which provides inherent support for the main components of the IPUS architecture. Within any IPUS-based system, problem solution involves a series of transformations between multiple abstract signal representations (such as waveform and tracks) that are stored on a hierarchical *blackboard database*. Signal processing algorithms which help carry out these transformations are executed in accordance with a set of control plans. The appropriate control plans that need to be invoked for a particular signal scenario are chosen using a *RESUN control planner* that relies on decomposing the system's goals into a goal/plan/subgoal hierarchy. ICP provides a rich library of base classes which may be utilized to derive and maintain application-specific versions of the blackboard database and the RESUN control planner. Additionally, ICP also includes a trace facility that allows all internal structures and operations of the system to be monitored using textual and graphical displays.

4.1. Evaluation

We have systematically evaluated the performance of our system using a variety of signal scenarios involving two simultaneously played violin notes. These scenarios, which were generated using the `Ensemble` music synthesis system [10], had note combinations corresponding to intervals of a fifth, a fourth, a major third, a major sixth, a minor third, a minor sixth, and a second. Each of these combinations was generated both with notes having vibrato and notes having no vibrato. Our system accurately separated the partials in each of the generated scenarios.

We now present an example from this evaluation that illustrates the sophisticated nature of the processing that is carried out within our system. The two-note signal scenario used in this example consists of a violin note C (fundamental frequency = 523 Hz) already playing when the system begins processing the data. Another violin note, the E with a fundamental frequency of 659 Hz and having a vibrato, comes on at time 0.32 secs. The results of processing this scenario are shown in Figure 3. These results show that our system successfully separated the partials in the musical signal.

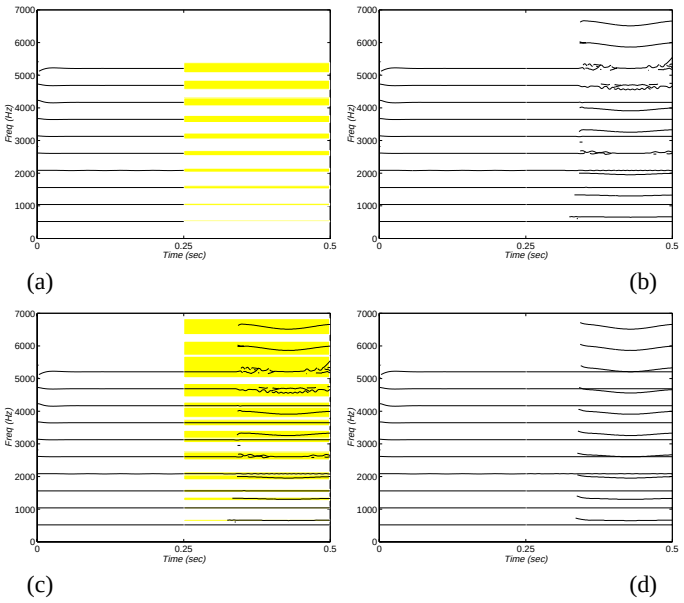

Figure 3: Processing of the signal block from 0.25 sec to 0.5 sec in the example scenario. Predictions made on the basis of partials identified in the first signal block from 0 to 0.25 secs are shown in plot (a). The gray areas indicate possible regions over which the time-frequency trajectories corresponding to the predicted partials are most likely to evolve. Plot (b) shows the tracks obtained from the results of constant-Q processing where the analysis filters have been adjusted according to the predictions. The gray regions in plot (c) indicate clusters of tracks formed using the clustering algorithm. Finally in plot (d), we show the results of reprocessing clusters which had discrepancies within them.

5. COMPUTATIONAL COMPLEXITY

An alternative to our knowledge-based approach to time-frequency analysis is a data-dependent approach that relies on a numerical measure of time-frequency energy concentration [4]. This numerical approach involves first performing hundreds of time-frequency analyses each with a different set of analysis filters. At each time-frequency point in each analysis a two-dimensional energy concentration measure, which is similar to the kurtosis measure used in statistics, is then evaluated. Finally, a composite time-frequency representation of the signal is formed by picking time-frequency points from different time-frequency analyses in a manner such that the kurtosis value at each point is maximized. In comparing this kurtosis-based filter adaptation strategy with our knowledge-based strategy, we note that our strategy enjoys the advantage of incorporating musical knowledge into the search for the appropriate analysis filters. Furthermore, we have been able to demonstrate that our strategy is also *at least* an order of magnitude more computationally efficient.

In our knowledge-based system, the bulk of the multiplications are utilized for the time-frequency processing and reprocessing of the signal through the analysis filterbank. While the peak picker and Kalman tracker require less than 1% of the multiplications used in the time-frequency processing, the other parts of the processing such as prediction, discrepancy detection, and diagnosis have negligible computational requirements. The number of multiplications required for processing the signal through a filterbank of N filters over M temporal points is:

$$C_2 = MN^2 \tag{4}$$

For an identical number of analysis filters and temporal points, the number of multiplications utilized in the kurtosis-based adaptation strategy has been shown to be [4]:

$$C_1 = W\left[8N + 2NM\log_2 N + 22NM + 24NM(\log_2(2N) + \log_2(2M))\right] \tag{5}$$

where W is the total number of analysis filters over which the search is performed at each time-frequency point. We see from (4) and (4) that for $M = 256$ and $N = 1024$, C_2 is 2% of C_1. Therefore, even if the data is reprocessed 5 times entirely (a highly unlikely scenario), our knowledge-based system would still use only a tenth of the computation demanded by the numerical approach. This computational efficiency along with the advantages of basing the processing on musical knowledge makes our novel system highly attractive for separating musical partials.

6. REFERENCES

[1] C. Chafe and D. Jaffe, "Source Separation and Note Identification in Polyphonic Music," *Proc. ICASSP 1986,* pp. 1289-92, Tokyo, May, 1986.

[2] E. R. S. Pearson and R. G. Wilson, "Musical Event Detection from Audio Signals within a Multiresolution Framework," *Proceed. ICMC 1990,* pp. 156-58, Glasgow, 1990.

[3] Brown, "Calculation of a Constant Q Spectral Transform," *J. Acoust. Soc. Am.*, vol. 89, no. 1, 425-34, 1990.

[4] D. L. Jones and T. W. Parks, "A High Resolution Data-Adaptive Time-Frequency Representation," *IEEE Trans. Acoust. Speech Sig. Proc.*, vol. 38, no. 12, pp. 2127-35, Dec. 1990.

[5] R. Mani and S. H. Nawab, "Knowledge-Based Processing of Multicomponent Signals in a Musical Application," submitted to *Signal Processing*.

[6] V. R. Lesser, S. H. Nawab, and F. I. Klassner, "IPUS: An Architecture for the Integrated Processing and Understanding of Signals," *Artificial Intelligence*, vol. 77, pp 129-171, 1995.

[7] Y, Bar-Shalom, T. E. Fortmann, "Tracking and Data Association," Academic Press, 1988.

[8] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete-Fourier Transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51-83, 1995.

[9] J. M. Winograd and S. H. Nawab, "A C++ Software Environment for the Development of Embedded Signal Processing Systems," *Proc. ICASSP 1995*, vol. 4, pp. 2715-19, Detroit, May 1995.

[10] *Ensemble* music synthesis system available at `http://www-ccrma.stanford.edu`.