MUMT 611 - Final Report

Paul C. Kosek
Department of Music Technology
McGill University
Montreal, Quebec
Email: kosek@music.mcgill.ca

Currently, there exist many different flavors of automatic extraction of piano music. Considerable success has been claimed by many, yet none has shown to perform with consistent near perfect accuracy. To this end I propose an alternative method of feature extraction.

The highest level of classification for a piano transcription algorithm is whether it is informed or not. An informed algorithm is one that takes the score of the music being transcribed into consideration. This is limited and less flexible, but arguably more helpful in the determination of an algorithm for note detection than actual note detection itself.

Possibly the most successful implementation of polyphonic piano transcription is *Sonic* [6]. *Sonic* is a project from the University of Ljubljana which uses neural networks to drive adaptive oscillators. Other methods incorporate fuzzy logic and hidden Markov Models into blackboard systems. Regardless of the implementation, feature extraction is a necessary first step of analysis.

The most common technique used for determining frequency components over a short duration of time is the Short Time Fourier Transform [STFT]. The STFT is limited by the fact that the shortest practical duration is 256 samples. Two hundred fifty-six samples at a sampling rate of 44100 Hz equates to slightly more than five ms. Five ms in piano music is an extremely long amount of time in that it renders trills and mordents into indiscernible blobs of sound. This lengthy duration also adds to the difficulty of octave detection. Pianists rarely strike all notes of a chord at the same sample, and this can be used to differentiate the amount of notes that have been struck at any given time. Pianists will not however take more than five ms to play a chord, which translates into a loss of information when the STFT is used.

To account for this significant loss of musical information, I propose to use a sample-wise frequency and amplitude analysis technique known as the first Discrete Energy Separation Algorithm [DESA-1], a function derived from the Teager Energy Operator.

Teager Energy Operator, a time domain nonlinear function of three contiguous samples, developed by Teager and introduced by Kaiser [1] is defined in the digital domain to be:

$$\Psi_d[x(n)] \triangleq x^2(n) - x(n-1)x(n+1) \tag{1}$$

The Teager Energy Operator, (TEO) is a very local function of the energy present in a signal. Various properties of the TEO have been revealed by Kaiser [2] and from them useful energy separation algorithms have been derived.

An energy separator can resolve an AM modulated, FM modulated signal into its respective AM modulated and FM modulated components. Maragos, Kaiser and Quatieri have introduced energy separators such as the DESA-1 [3] based upon the TEO. The implementation of the DESA-1 is as follows:

$$\Omega_i(n) = \arccos\left(1 - \frac{\Psi[y(n)] + \Psi[y(n+1)]}{4\Psi[x(n)]}\right) \tag{2}$$

$$|a(n)| = \sqrt{\frac{\Psi[x(n)]}{1 - \left(1 - \frac{\Psi[y(n)]+\Psi[y(n+1)]}{4\Psi[x(n)]}\right)^2}} \tag{3}$$

where

$$y(n) = x(n) - x(n-1) \tag{4}$$

The DESA-1 algorithm was used by [3] in devising frequency and amplitude envelopes for harmonics of solo acoustic instruments. Amplitude envelopes of different harmonic components of solo piano notes were shown to be distinct in this study. From these results, I speculated that that the frequency and amplitude envelopes as determined by this algorithm could be used as features in a classification algorithm for automatic transcription.

The main limitation of the TEO is that the assumptions from which it is derived break down for frequencies above a quarter of the Nyquist Rate. Another major challenge associated with practical implementation of the algorithm is filtering the signal down to the practical bandwidth. Butterworth and Chebychev filters are capable of producing narrow bandwidth signals, but still allow enough energy to pass through at frequencies beyond the bandwidth that they are impractical in this situation. Maragos et al achieve this through convolution with an extremely narrow banded signal. To generate the narrow banded signal they use a truncated discrete Gabor Impulse. The algorithm for the Gabor Impulse is:

$$h(n) = e^{-(bn)^2}\cos(2\pi f_c nT); -N \leq n \leq N \tag{5}$$

The designer can easily choose the center frequency, and bandwidth using the relationship:

$$b = BW \times T \times \sqrt{2\pi} \tag{6}$$

Convolution is a computationally expensive operation. To reduce the number of computations necessary, fast convolution can used instead. Fast convolution is made possible by choosing the length of the impulse response to be a power of two. In fast convolution, the *2N* point FFT is taken on two signals of length *N* zero padded out to length *2N*. The resultant transformed signals are multiplied together and the *2N* point IFFT is taken. Typically one signal is a fixed impulse response (h), and the other is the signal upon which filtering is desired (x). The same operation is then performed using x(N:2N-1) as the x data, and the IFFT result is added to the last *N* samples of the previous output appended with *N* zeros. This is an efficient operation with little degradation of the result as compared to standard convolution.

I set out to extract the harmonic features of the piano music through an exhaustive iterative process. The first step is to choose an appropriate length for the impulse response. The minimum attainable bandwith for the frequency response will decrease as the length of the impulse response increases, enabling more precise filters, yet as the length of the impulse response increases, characteristic features of the harmonic envelopes become heavily damped. The output of this initial filtering process will be input to the DESA-1 algorithm, thus it is imperative that the signal have a very narrow bandwidth, as the output of the DESA-1 algorithm will be used to determine the frequency components of the signal. It turns out that shorter impulse response has enough energy outside of the bandwidth to make it difficult, if not impossible, to use the DESA-1 algorithm as a frequency detector.

The impulse response that will be input into the fast convolution operation was chosen to be 16834 samples long. The next step was to choose the frequencies to be analyzed. The DESA-1 algorithm is extremely noisy for samples under 40Hz, and inapplicable for samples above 5000Hz. This range would not allow us to use a common practice in piano transcription; correlation of high frequency components to note onsets. It was speculated that this would not be significant, yet if it were desired both analysis techniques could be lumped into a blackboard system.

Given the bandwidth of total analysis, a practical peak picking implementation becomes necessary. This process is complicated only by the amount of data involved.

Fast convolution and the DESA-1 algorithm are computationally inexpensive. They can be implemented several times in parallel in real time.

Convolution and energy separation for each frequency band for the entire sample can be computed in an acceptable amount of time. This operation requires that the data be stored before it is analyzed. This is impossible to accomplish today by just keeping the contents in RAM, as almost five thousand times number of samples are needed to be stored. This data would most likely be stored as a floating point variable which would further increase the amount of storage necessary. This data can be stored to disk, but again, the amount of storage necessary makes this process impractical.

Ideally, we would pick peaks as each sample is computed. This is not possible, however, as the fast convolution generates samples in batches of length *N*, in this case 16384. (The IFFT generates batches of 32768 samples, but only the first half is complete for analysis.) Even five thousand times sixteen thousand samples is a major burden on the memory capabilities of a computer.

The solution I have currently implemented is not efficient, but allows the computations to take place. For each batch of N samples, I compute the fast convolution and energy separation for each frequency and store the results to disk. After all convolutions have taken place, I concatenate the results into matrices of a size that can be stored in memory and minimizes the number of disk read/write operations.

These matrices are examined sample by sample for amplitudes above a threshold. If data is found to be above a threshold, the rising and falling edge frequency bands are stored. If any bands are determined to be touching any bands from the previous sample, the band from the previous sample's limits are extended to include the sample number and frequency limits (if necessary) of this band. This is an extremely fast operation, and readily picks all ridges out of the spectrum of the sample.

This ridge detection algorithm can be made much faster, I presume, by computing the amplitude envelope of a single frequency for the entire sample and storing the indices at which it crosses the threshold. Then these indices could be examined for all frequencies using a fraction of the disk read/write operations, which are the major time consumers in this process.

Given the ridges, the next step, which has not been included in process yet, is to analyze the sample with a shorter, windowed impulse response at only the sample indices and frequency bands as determined by the ridges. This shorter, windowed impulse response accentuates the characteristic envelopes of each harmonic. These samples could be fed to an existing classification algorithm, such as the adaptive oscillators of SONIC.

The process of feature extraction is necessary for any implementation of automatic transcription. The process suggested and implemented is relatively simplistic, yet the ability of a convolution filter to take a single narrow slice of the frequency spectrum, and the ability of the energy separation algorithm to accurately capture the envelope of a narrow band signal provide a large amount of information with resolution beyond that which is available via other methods.

The primary aspect that needs addressing in this algorithm is a reduction in computation time. Much effort was spent on rearranging various aspects for efficient and complete implementation. The most significant introduction was the introduction of index cells in MATLAB to reduce the amount of data stored. The intent of these storage structures is essentially to make the data sparse, and as samples of interest are converted to indices of interest, the speed of the operation will

only increase.

REFERENCES

[1] J. F. Kaiser. On a simple algorithm to calculate the 'energy' of a signal. In *Proceedings of ICASSP*, pp. 381-384, April 1990

[2] J. F. Kaiser. Some useful properties of Teager's energy operators. In *Proceeedings of International Conference on Acoustics Speech and Signal Processing*, pp. III:149-152, April 1993.

[3] Petros Maragos, J. F. Kaiser and Thomas Quatieri. On amplitude and frequency demodulation using energy operators. *IEEE transactions on Signal Processing*, 41(10):1532-1550, April 1993.

[4] S. Dixon, 'On the computer recognition of solo piano music.' in *Proceedings of Australasian Computer Music Conference*, Brisbane, Australia, 2000.

[5] E.D. Scheirer. 'Using musical knowledge to extract expressive performance information from audio recordings.' In H. Okuno and D. Rosenthal (editors), Readings in Computational Auditory Scene Analysis. Lawrence Erlbaum, 1997

[6] Marolt, M., 'SONIC: Transcription of Polyphonic Piano Music With Neural Networks.' in *Workshop on Current Research Directions in Computer Music*, pp. 217-224. Barcelona 2001.

[7] C. Raphael, 'Automatic Transcription of Piano Music.' in *Proceedings of the third International Symposium on Music Information Retrieval (ISMIR-03)*, Paris, France. October 2002.

[8] Kaiser, J. F. *On a Simple Algorithm to Calculate the 'Energy' of a Signal* pp. 381-384 Proc. IEEE ICASSP, Albuquerque, N.M, 1990.

[9] J. P. Bello, G. Monti and M. Sandler, 'Techniques for Automatic Music Transcription'. in *Proceedings of the first International Symposium on Music Information Retrieval (ISMIR-01)*, Plymouth, Massachusetts, USA. October 2000.

[10] G. Monti and M. Sandler, 'Automatic Polyphonic Piano Note Extraction using Fuzzy Logic in a Blackboard System,' 5th International Workshop on Digital Audio Effects, DAFx 02, Hamburg, September 2002, 39-44.