Multiphonic Note Identification

Simon Dixon

Department of Computer Science
Flinders University of South Australia
GPO Box 2100, Adelaide SA 5001

email: dixon@cs.flinders.edu.au

Abstract

This paper describes work in progress which addresses several aspects of perception of sound by a computer. This work forms part of a project to build an automatic music transcription system. In recent years there has been considerable advances in the area of note identification (frequency tracking), to the extent that there are commercial systems which perform well on the task of monophonic note identification. The problem of identifying multiple simultaneous notes remains mostly unsolved, due to the difficulty of separating the partials into their component notes. We present an approach to multiphonic note identification, drawing on research in speech recognition and psychoacoustics. The acoustic data is processed according to a model of human auditory perception and dynamic models of the sources.

Keywords Music perception, note identification, automatic transcription, frequency tracking.

1 Introduction and Background

Devices for analysing acoustic signals have been available for most of this century, but it is only in the last twenty years that significant progress has been made in the perception of musical information via machines. This has largely been brought about by the advances in digital computers and the corresponding progress in digital signal processing.

One of the most commonly addressed problems is that of note identification, or frequency tracking[1]. This is an essential element of an automatic transcription system, and also has application in such diverse areas as psychoacoustics and instrument design as well as considerable commercial application in electronic music.

Many instrumentalists want to take advantage of the wide range of previously unavailable tone colours which are now found on commercial music synthesisors. But those who do not play the keyboard would prefer to access these sound sources via the familiar interface of their own instrument. Several products have been developed which achieve this goal. These devices operate in one of the two following ways: by direct detection of control information, or by acoustical analysis.

The most simple way to control a synthesisor module from another instrument is to detect the control information directly on the instrument, before it is converted to sound. In this approach, events such as striking and releasing keys on a keyboard, opening and closing valves on a brass instrument, and air pressure variations in the mouthpiece of a wind instument, are translated directly into control information for the synthesisor. The advantage of this method is its relative simplicity: it requires no acoustic analysis. However, there is a drawback with this approach: some of the subtleties of the performance (for example, small pitch variations in wind instruments) are lost when a note is mapped to a discrete value. The result of this loss of information in the translation process is that musicians do not have control over the same range of expression that they have on their own instruments.

In the second approach, musical sounds are detected via a microphone, transducer or electromagnetic pickup, and then the electronic signal is analysed for frequency (and possibly amplitude) information which is then converted to a digital code which is used to drive the synthesisor. This approach is much more difficult, as a single note from an instrument is a complex waveform which is only quasi-periodic, and which must be approximated by a group of sinusoidal partials. Although most instruments produces notes with partials which are approximately harmonic (integer multiples of a common fundamental frequency), which aids in the detection process, the non-stationary nature of the waves reduces the accuracy of attempts to capture the perceived features of the notes.

Commercial systems which convert acoustic waveforms to a synthesisor control signal are available, but in general these systems only work

[1] We avoid the term *pitch tracking*, as pitch is a perceptual quantity.

Proceedings of the 19th Australasian Computer Science Conference, Melbourne, Australia, January 31–February 2 1996.

for monophonic instruments, due to the difficulty of separating multiple notes in an acoustic waveform. This problem can be circumvented in some multiphonic instruments such as guitar and piano, by attaching a separate pickup to each string (or set of strings), and converting the signal to control information before mixing it with the signals from other strings. This effectively treats the instrument as a collection of monophonic instruments, one for each string. The advantage of knowing that a signal is monophonic is that it greatly simplifies the detection of notes, as all of the partials must then come from the same note, whereas for multiphonic instruments the partials of different notes are mixed together, and there may not even be a unique way of decomposing the combination of partials into their constituent notes[2].

Current research in music perception has not succeeded in tracking multiple simultaneous sounds, whether on the same or different instruments or voices. To successfully decompose multiple notes requires a sufficiently accurate model of each sound source which can be used to distinguish the different sources. Tanguiane [13] advocates a statistical pattern-matching method using Boolean spectra, a binary representation indicating the presence of partials at various frequencies but ignoring their amplitude. Although this approach improves the computational aspects of note decomposition, it does not allow for adequate identification of the source of the component notes. Research in psychoacoustics indicates that even if amplitude information is retained, the spectral envelope still does not provide sufficient detail to identify instruments. Therefore, the initial transient of a note must also be an important acoustic cue.

Many other acoustic cues are discussed in the auditory streaming literature, particularly in the work of Bregman [1]. This work uses principles from Gestalt psychology to explain the brain's ability to segregate and regroup sounds according to their sources. A computational approach to this work is also being developed [2]. Other related work is found in the speech processing community, in attempts to separate the voices of multiple simultaneous speakers.

In current work, we are investigating the creation of dynamic models of individual musical voices to enable separate tracking of these voices. These dynamic models will each represent an auditory stream, and component sounds which fit a model well will be adsorbed into that stream, and the model will be adjusted to reflect the stream's new component.

In this paper we do not address the problem of rhythm perception, a necessary constituent of an automatic transcription system. A dynamic modelling approach can also be used at this level to automatically extract such features as the bar and phrase structure of musical pieces [12]. A similar approach is proposed by Tanguiane [13], and is also used in extracting heirarchical structure from language [6].

2 Limitations of Fourier Analysis

Standard Fourier analysis is not sufficient for note identification in music. It is not possible to obtain sufficient resolution in both the time and frequency domains simultaneously, as there is an inherent time delay in note identification which is inversely proportional to the frequency of the note. The parameters chosen for a Fourier transform provide a tradeoff between the resolution in the frequency domain and resolution in the time domain.

For example, the two lowest notes on a piano, A_0 (27.5 Hz) and $B\flat_0$ (29.1 Hz) differ in frequency by 1.6 Hz. A short-time discrete Fourier transform of N samples taken with sampling rate S will give $\frac{N}{2}+1$ equally spaced frequency bands, each with bandwidth $\frac{S}{N}$. Thus the frequency resolution is $\frac{S}{N}$, which is just the reciprocal of the sample length in seconds (independent of the sampling frequency). So to distinguish the two lowest notes of a piano, a sample length of $\frac{1}{1.6} = 0.6$ seconds is required. In a piece of music, a reasonably fast scale or arpeggio passage will have around 10 notes per second, requiring a time resolution of 0.1 seconds.

At the other end of the frequency range there is a different problem: the frequency bands produced by the Fourier transform are too widely spaced. Again taking the piano as our example, the two highest notes B_7 (3951 Hz) and C_8 (4186 Hz) differ by 235 Hz, or 146 frequency bands, if we keep to a resolution of 1.6 Hz. The intermediate bands give very little useful musical information, yet most of the computation time is spent in calculating the content of these bands.

A solution to this problem is the Constant Q Transform [3], which calculates the Fourier transform with a variable window size for different frequency bins. This can be calculated one octave at a time, starting from the top octave with a relatively small window size, and subsampling the signal over a larger time-frame for lower octaves to save on computation time. This does not solve the resolution problem at the low end of the spectrum, because each time the signal is subsampled, a longer signal must be used in order to achieve the same number of frequency bands (usually 12 or 24 per octave, corresponding to semitones or quartertones).

[2] See Tanguiane [13] for a mathematical analysis and discussion of this problem.

The above approach is essentially that of the channel vocoder in speech analysis. An alternative approach which achieves much better resolution is based on the phase vocoder [9, 8]. The channel vocoder calculates the magnitude of the complex values returned by the Fourier transform, but ignores the phase information. This is justified by the observation that absolute phase does not appear to be relevant to the perception of sound. The idea of the phase vocoder is to use the phase information to improve on the frequency estimate of the energy captured in the frequency bin. Normally, the energy in a particularly frequency bin is assumed to correspond to the centre of the given band in which the spectral energy was detected. But by measuring the phase change between successive windows (usually only one sample apart), a much more precise value of frequency can be determined [5].

Various alternative techniques have been proposed in the literature, such as wavelets, autocorrelation and cross-correlation. The statistical correlation methods are particularly useful for processing harmonic sounds, such as those produced by the voice and all string and wind instruments, as the frequencies of the spectral components of these waveforms are in simple integer ratios. In our approach, the models created specifically for harmonic instruments are able to take advantage of the same information. Using these techniques, we have been reasonably successful in solving the problem of mistaking harmonics for notes and vice versa, and also avoiding problem of erroneous notes caused by heterodyne components (difference and summation tones).

But even without an expected model for the pattern of partials of a note, Tanguiane [13] claims that using autocorrelation, it is still possible to track such notes and resolve them into partials. Such an approach also aids the separation of signal from noise, and the resolution of signals which are masked by the presence of other signals, but fails to model the expected relationships between and within notes which aid human perception of music[3].

3 Experimental Method

The project began with a preliminary investigation of note identification in music using standard Fourier techniques. Software was developed to create digital samples of sounds entered either via a microphone or a line input (from tape or CD). Two different sampling rates were used — 8 kHz and 44.1 kHz. Although the 8kHz sampling rate limits the bandwidth to 4kHz (just below C_8, the top note on the piano), the only noticeable performance difference is the time saving of over 80% when using the lower sampling rate.

[3] c.f. the schema-based stream formation of Bregman [1]

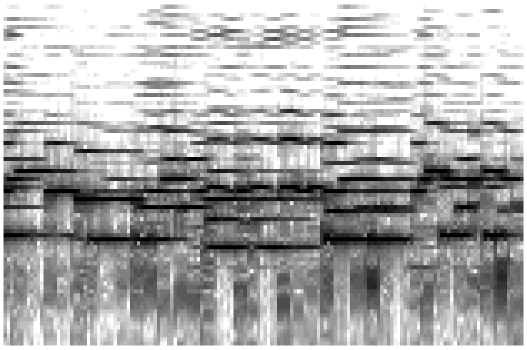

Figure 1: Spectrogram showing spectral leakage due to a rectangular window. The vertical axis represents frequency (one semitone per pixel), the horizontal axis represents time (64ms per pixel), and the amplitude is represented by darkness.

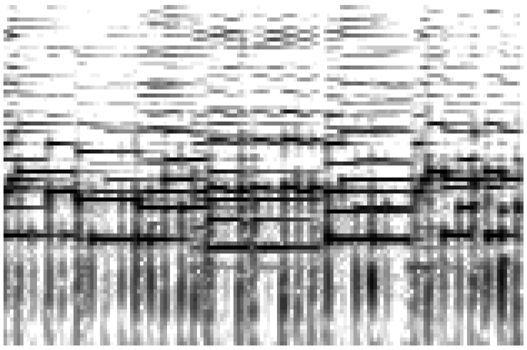

Figure 2: Reduction in spectral leakage due to a Hamming window.

Before performing the Fourier transform on the data, a suitable window size and shape had to be chosen. The suitability of a particular window size depends on a number of factors: the sampling rate, the speed of the music (which determines the smallest time resolution required), the lowest frequency of interest (which determines the required frequency resolution), and the time to calculate the Fourier transform. The frequency resolution required for an automatic transcription system (for Western music) is one semitone, which is just under 6% of the frequency. For a linear frequency scale, the allowable error is then $\pm 3\%$ of the lowest frequency. The time resolution must be well below 100ms to distinguish the notes of a fast melodic passage, but may have to be significantly smaller to transcribe rhythms precisely. The present results use a time resolution of 64ms.

Initially, a rectangular window shape was chosen, but this gives poor results as it has maximal "leakage" at its boundaries (figure 1). That is, spectral information gets smeared across adjacent frequency bins. Better results were obtained using

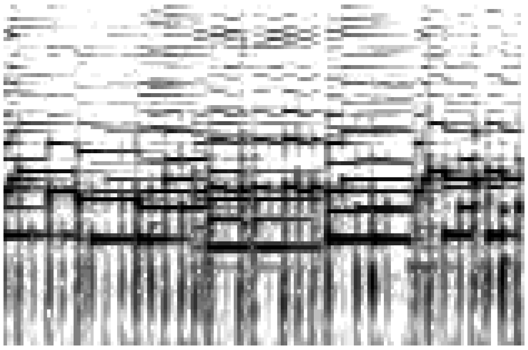

Figure 3: Results of using a 3rd-order Blackman-Harris window.

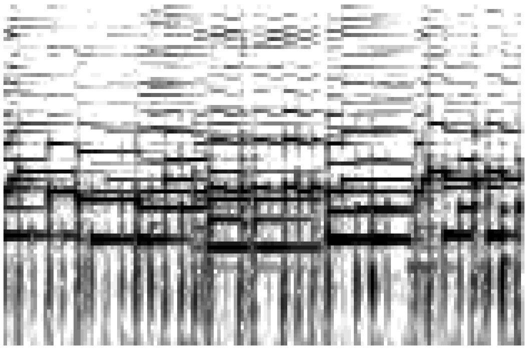

Figure 4: Results of using a 4th-order Blackman-Harris window.

a Hamming window (figure 2), which reduces side-lobe leakage to 43dB below the level of the main lobe. Higher order windows with optimal properties [10] were also tried (figures 3 and 4), but these did not improve results noticeably. These windows have much lower sidelobes (-67dB and -92dB lower than the main lobe), but this is obtained at the cost of a wider main lobe. The lack of improvement using higher order windows indicates that the remaining spectral noise is not due to leakage in the transform, but due to the non-periodicity of the signal.

After performing the Fourier transform, the complex values for the frequency bins were converted from rectangular to polar coordinates, giving amplitude and phase values. The amplitude values were normalised to correct for the gain of the window, and and then converted to a logarithmic (dB) scale. This is a reasonable first approximation for a simple auditory model, where the perceived loudness is proportional to the logarithm of the amplitude. This method also compresses the amplitude values into a manageable range.

In current work we are refining these results by scaling the spectra based on the the non-linear characteristics of the human ear, to reflect more accurately the perception of sounds arriving at the

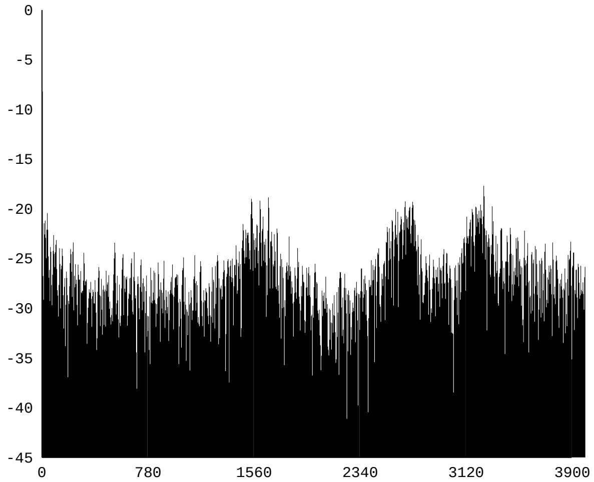

Figure 5: Fourier transform of guitar chord (3 notes ringing). Vertical axis is amplitude (dB) and horizontal axis is frequency (Hz).

ear. This will model both the frequency and amplitude response of the ear, as found in work on speech recognition [7, 14].

The next stage of processing determines the amplitude peaks in the spectral data. Without this step, very little useful information would be obtained from the Fourier transform. Unlike artificial sound data, the short-time Fourier transform of natural acoustic data does not produce a discrete line spectrum but a continuous and noisy spectrum (figure 5).

By displaying the data graphically it was possible to determine spectral peaks visually (figures 1 to 4), and these peaks corresponded to the audible notes and their harmonics. Using an on-centre off-surround peak detection algorithm, we were also able to detect peaks automatically with a high degree of accuracy. A peak was detected if there was a local maximum in amplitude within a certain band of points. It was found that a small bandwidth (1-3 points each side) gave best results, because larger values allowed notes to be masked by other nearby notes. The small bandwidth also produced a greater number of spurious peaks, but these were relatively easy to detect by the source model later in the processing.

Other refinements were also tried, using statistical methods such as auto-correlation and cross-correlation [4]. It was found that these approaches were not useful until after the peak-detection step (figures 6 and 7), at which point they were useful in removing spurious peaks from the data. This relies on the fact that musical sounds have a predictable composition of partials at harmonically related frequencies. A correlation with a harmonic spectrum confirmed the identification of a majority of notes, and best performance was obtained by correlating with a short sequence of harmonics (8). This agrees

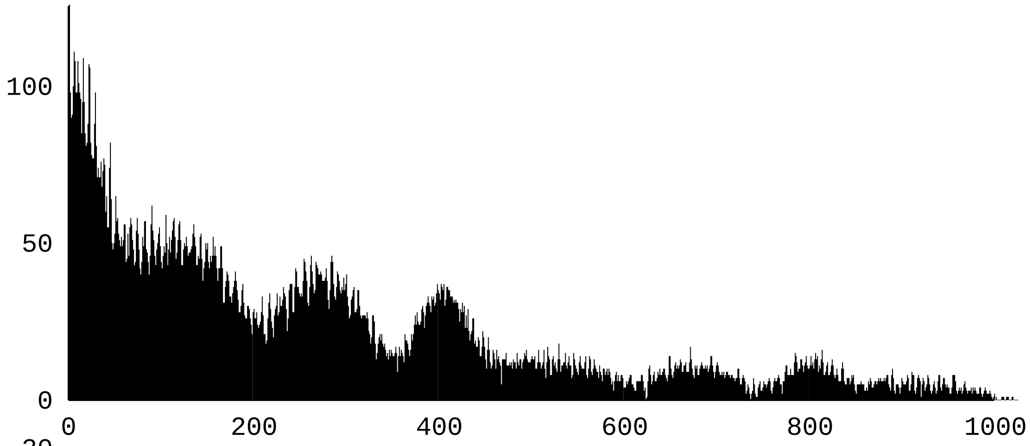

Figure 6: Autocorrelation before peak detection

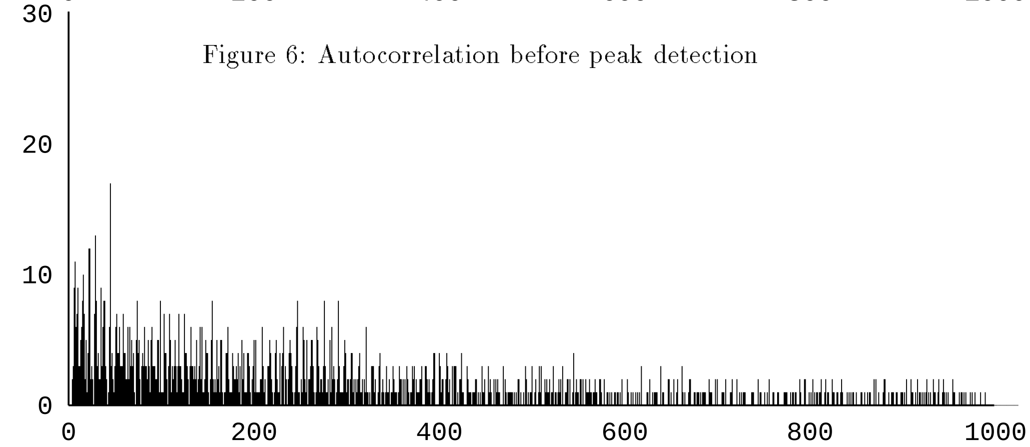

Figure 7: Autocorrelation after peak detection

with the claim of Plomp [11] that partials after the 7th are "rather unimportant".

The amplitude peaks were then discretised on a logarithmic frequency scale, identifying the peaks with musical notes, resolving to the nearest semitone (encoded in MIDI representation). Using the assumption that notes are composed of harmonic partials, a "best fit" decomposition of the partials was found, using a predictive model for the spectral envelope. In present work, this model is specific to the percussive string instruments such as guitar and piano, which produce notes whose amplitudes decrease over time, and whose partials are harmonic and have succcessively lower amplitudes as the partial number is increased. In further work, dynamic models of instruments and voices will be created to generalise this procedure.

The high degree of continuity between successive frames is also used to aid in the correct decomposition of sounds. This is particularly important to deal with the changing waveform of notes as they decay. By comparing the amplitude of each partial with its value in previous and successive frames, it is possible to detect and remove further spurious notes and fill in missing notes using a simple smoothing algorithm.

The final step of assigning a musical rhythm to the detected notes has not been addressed in this work to date. A review of work in this area can be found in [13].

4 Results and Discussion

Preliminary results show approximately 95% success rate in note identification for multiphonic guitar music. This is a promising result, as the rapid decay and weak harmonics of plucked strings make them more difficult to detect than most other instruments which have a less transient response. The peak detection algorithm currently does not make use of phase information from the Fourier transform; this information will allow a much higher resolution of frequency, or can be traded for a greater time resolution, depending on the needs.

It is surprising that many of the approaches used in musical applications such as note identification are not based on any underlying model of the human auditory system. The historical evolution of music is intrinsically related to human perception, and there is already much research by speech scientists which could be applied in this direction.

At the same time, we do not limit this system to the physical limits of human perception; there may be features of sounds which are imperceivable by humans but are nevertheless useful in disambiguating complex sounds. But we do observe that human perception is close to attaining the mathematical limits of precision in some areas of note identification. In fact, sometimes human ability is so surprising that it is evident that further information is being extracted from the environment (possibly via predictive models of sources) which goes beyond what we have examined in this paper.

5 Conclusions and Further Work

This paper demonstrates the feasibility of automatic transcription of multiphonic music. The project is still in early stages, yet provides promising results for the data on which it has been tested. Further work will investigate the effects of more accurate models of the auditory system such as the Bark transform [14], the use of phase information to improve frequency resolution, and the use of dynamic modelling to recognise and separate sounds from different sources. We also intend to implement ideas from auditory scene analysis [1, 2]. The perception of rhythm will also be addressed.

Acknowledgement

The author wishes to thank David Powers for many useful ideas and discussions.

References

[1] A.S. Bregman. *Auditory Scene Analysis: The Perceptual Organisation of Sound.* Bradford, MIT Press, 1990.

[2] G.J. Brown and M.P. Cooke. Computational auditory scene analysis. *Computer Speech and Language*, Volume 8, pages 297–336, 1994.

[3] J.C. Brown. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, Volume 89, Number 1, pages 425–434, 1991.

[4] J.C. Brown. Musical fundamental frequency tracker using a pattern recognition method. *Journal of the Acoustical Society of America*, Volume 92, Number 3, pages 1394–1402, 1992.

[5] J.C. Brown and M.S. Puckette. A high resolution fundamental frequency determination based on phase changes of the fourier transform. *Journal of the Acoustical Society of America*, Volume 94, Number 2, pages 662–667, 1993.

[6] W. Daelemans and D.M.W. Powers (Eds). *Background and Experiments in Machine Learning of Natural Language: First SHOE Workshop.* Tilburg University NL, itk proceedings 92/1 edition, 1992.

[7] P.B. Denes and E.N. Pinson. *The Speech Chain.* Doubleday Anchor, 1963.

[8] M. Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, Volume 10, Number 4, pages 14–27, 1986.

[9] J.L. Flanagan and R.M. Golden. Phase vocoder. *Bell System Technical Journal*, Volume 45, pages 1493–1509, 1966.

[10] F.J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, Volume 66, Number 1, pages 51–83, 1978.

[11] R. Plomp. The ear as a frequency analyser. *Journal of the Acoustical Society of America*, Volume 36, pages 1628–1636, 1964.

[12] D. Rosenthal. Emulation of human rhythm perception. *Computer Music Journal*, Volume 16, Number 1, pages 64–76, 1992.

[13] A.S. Tanguiane. *Artificial Perception and Music Recognition.* Springer-Verlag, 1993.

[14] E. Zwicker and E. Terhardt. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *Journal of the Acoustical Society of America*, Volume 68, pages 1523–1525, 1980.