Acknowledgements

Funding for this work was provided by teh SERC and CRL Limited. I would like to acknowledge the helpful advice of my supervisor, Dr. Roland Wilson and Dr. Martin Todd at CRL.

1 Musical Signals and Music Transcription

Musical sounds may be produced in an infinite variety of ways, both by traditional instruments (violins and so forth) or more modern devices, such as synthesizers. As a result there is an infinite number of possible sounds which may be employed in a musical piece. However, for the purposes of this work it is assumed that all sounds employed in the sorts of music to be analysed may be adequately described by four physical parameters, which have corresponding psychological correlates [14].

- Repetition rate or fundamental frequency of the sound wave, correlating with pitch.
- Soundwave amplitude, correlating with loudness.
- Soundwave shape, correlating with timbre.
- Sound source location, with respect to the listener, correlating with the listener's perceived sound localisation.

Of these parameters, the latter will be ignored in this work as it is assumed that it should have no significant effect on the system's ability to transcribe a piece, and that the effect on the single-channel sound signal will be minimal. It may appear that the timbre of a sound is not important, since it is not explicitly represented in the Common Practice Notation into which we (ultimately) with to have the sound transcribed. However, in practice timbre (ill-defined as it is mathematically) is not orthogonal to pitch and loudness (for example, the timbral quality of *vibrato* is a variation in pitch). Furthermore the timbral characteristics of individual instruments may be of some use in segmenting the audio signal in a polyphonic context.

The audio signal that is the input to the system to be developed is a digitally sampled signal representing the analogue sound waveform, such as that encoded on a Compact Disc or Digital Audio Tape. It is assumed that the signal is the sum of the contributions from one or more instruments (i.e. the music may be polyphonic). Note that non-linear distortions and noise are assumed to be minimal (Figure 1).

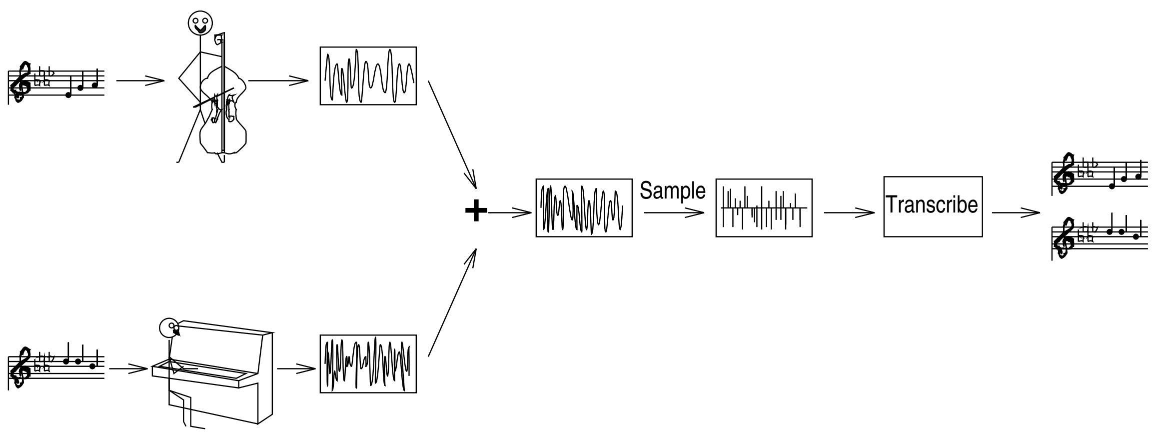


Figure 1

The major source of difficultly in this work arises from the fact that the musical input signals are allowed to be polyphonic. Indeed, solutions to the problem for monophonic signals are currently in existence which may operate in real-time and with few restrictions on the type of instrument which may be transcribed (e.g. the human singing voice can be dealt with) [9]. However, to date as far as the author is aware, there are *no* systems in existence which offer such flexibility for polyphonic music, even in non-real-time form.

The reason, perhaps, that solutions for the problem for monophonic sounds seem so much easier (though they are not trivial) is that such solutions may rely on the fact that the input signal contains a single harmonic structure for that instrument, plus some (usually small) inharmonic components and an additional quantity of unwanted noise. Hence the signal may be approximately described by,

$$x(t) = \sum kA_k(t)\sin(k\omega t + \phi_k) + e(t) \tag{1.1}$$

where

$x(t)$ is the sound signal in the time domain
ω is the fundamental frequency of the current note
$A_k(t)$ is the amplitude of the kth harmonic at time t
ϕ_k is the phase of the kth harmonic
$e(t)$ is a residual term (inharmonic components and noise)

Furthermore the structure will change in more or less discrete steps as the instrument changes pitch from note to note. The problem the for a monophonic transcriber is to identify the fundamental frequency of the signal, find the corresponding pitch class and pitch height (allowing for the fact that the instrument may be somewhat de-tuned), and measure the duration of the note (which should be approximately some simple rational multiple of a unit of beat) to find the note type (crotchet, quaver etc.)

The same problems are present in the polyphonic case. But now the harmonic structure is the sum of the structures of the individual sound sources. The notes of the instruments are not likely to all start and end together in general, and thus the harmonic structure, while perhaps varying discretely for individual instruments, in general may contain overlapping sets of harmonics from discrete time periods (Figure 2).

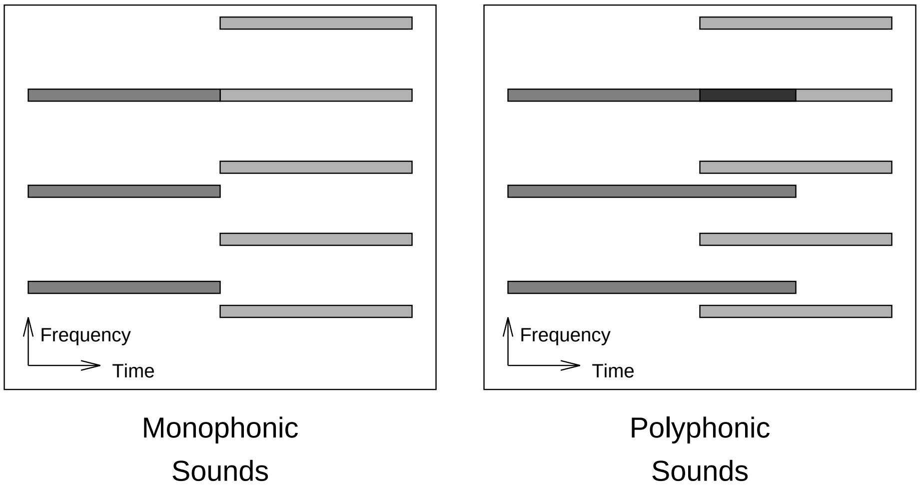


Figure 2 The Structure of Musical Signals

A more insidious problem, which in general does not apply at all to the monophonic case, is that harmonics which are close together in frequency interfere to create a beating effect which approximates an amplitude modulated sinusoid. In this case two harmonic elements merge into one, which has characteristics (the amplitude modulation) of neither individually. Unfortunately, such cases are all too common in traditional music. For example, two tones a fifth apart at 200Hz and 300Hz would both have harmonics at 600Hz. In the worst case, two instruments playing a note an octave apart would have *all* the harmonics of the lower note very close (theoretically identical to) those of the higher note. (The chances of the harmonics exactly canceling is negligible, however, since the two sets of harmonics are not likely to be exactly in phase, nor at exactly the same relative strengths.)

Clearly, therefore, there is a need for some analytical tool which can separate the features of each instrument, both in frequency and time. For this work that tool is the Multiresolution Fourier Transform.

2 The Multiresolution Fourier Transform (MFT)

The Multiresolution Fourier Transform (hereafter abbreviated to MFT) has been described in some detail in other works [16,21,25]. However, a brief description will be given here for the benefit of those readers unfamiliar with those references.

2.1 The Continous MFT

The continuous MFT is a complex-valued function of three continuous parameters: time, t, frequency, ω, and scale, σ. The MFT $\hat{x}(t,\omega,\sigma)$, of a given 1-D input signal $x(t)$ is given by [25],

$$\hat{x}(t,\omega,\sigma) = \sigma^{\frac{1}{2}} \int_{-\infty}^{\infty} x(\chi) w(\sigma(\chi - t)) e^{-j\omega\chi} d\chi \tag{2.1}$$

There are a number of ways one may view the effect of the MFT. One of the more more intuitive is to consider that the complex exponential term $e^{-j\omega\chi}$ acts as a modulator on the input signal $x(\chi)$ [14]. If the input contains a sinusoidal component of angular frequency ω_i then the modulated signal $x(\chi)e^{-j\omega\chi}$ will contain components at frequencies $\omega_i + \omega$ and $\omega_i - \omega$.† From (2.1) it can be seen that this modulated signal is then convolved in the time domain (equivalent to multiplication in the frequency domain) with the function $w(t)$. If $w(t)$ acts as a low-pass filter then the $\omega_i + \omega$ components will be attenuated and only those components in the region of ω (such that $\omega_i - \omega \approx 0$) will be retained. Furthermore, if $w(t)$ is also band-limited in time the MFT at time t and frequency ω will give an estimate of the spectrum of the signal around t and ω. The scale parameter σ has the effect of shortening or lengthening the time response of the window function (consequently lengthening and shortening it in the frequency domain). Hence, doubling σ will cause the time domain response to be twice as long, and the frequency domain bandwidth half as wide. These considerations lead to a number of constraints on the choice of window function $w(t)$ [21,25]. Firstly, it must have finite energy,

$$\int_{-\infty}^{\infty} w^2(\eta) d\eta < \infty \tag{2.2}$$

Secondly, the window and its Fourier Transform, W must be smooth,

$$\frac{\partial^2}{\partial\eta^2}(w(\eta)) \quad \text{and} \quad \frac{\partial^2}{\partial\eta^2}(W(\eta)) \quad \text{are continuous} \tag{2.3}$$

Thirdly, it must be even,

$$w(t) = w(-t) \tag{2.4}$$

Apart from these constraints the window may be freely chosen for a particular application. For the purposes of this work functions from the class of Finite Prolate Spheroidal Sequences are used [26].

Due to the trigonometrical identity,

$$\cos A \cos B = \tfrac{1}{2}\left[cos(A+B) + \cos(A-B)\right]$$

2.2 The Discrete MFT

In order to be implementable on a modern digital computer a discrete form of the MFT must be used. In this case the MFT becomes a complex-valued discrete function of the discrete parameters time t, frequency ω and scale σ. The definition of this is,

$$\hat{x}(i, j, n) = \sum_{k=0}^{N-1} w_n(t_k - t_i(n))x(t_k)e^{-jt_k\omega_j(n)} \tag{2.5}$$

where

N is the total number of sample points of the original signal $x(t)$
i is the time index
j is the frequency index
n is the scale index (or level)
t_k is the kth sample of the original signal
$t_k(n)$ is the kth time sample of scale n
$\omega_j(n)$ is the jth frequency sample at scale n

The value N is subject to the condition $N_\omega N_t \geq N$ where N_ω is the frequency sampling resolution and N_t is the time sampling resolution of the MFT. In other words, the number of coefficients in a given MFT level must be greater than or equal to the number of sample points in the original signal. If $N = 2^M$ samples then suitable values for the frequency and time sampling intervals for level n may be given by [25],

$$\Xi(n) = 2^{M-k-n}, \text{ for the temporal sampling interval} \tag{2.6}$$

$$\Omega(n) = 2^{n+1-M}\pi, \text{ for the frequency sampling interval} \tag{2.7}$$

For some value k which denotes the amount of temporal oversampling. For the purposes of this work a value of $k = 1$ is used as this improves numerical stability in the computation of the inverse transform and allows as relaxed form of the FPSS window function to be used which has reduced magnitude temporal sidelobes [25]. These values of the sampling intervals give rise to the following values for the number of temporal and frequency samples,

$$N_t = 2^{n+k} \tag{2.8}$$

$$N_\omega = 2^k N \tag{2.9}$$

The ratio of scales between levels is fixed for this work at 2 (meaning that level $n+1$ has twice the frequency resolution of level n, but half the temporal resolution) (Figure (3)). This has been found to maximise the consistency between levels and has the added benefit of allowing the MFT to be implemented in terms of the FFT [25].

2.3 MFT Summary

The MFT thus provides us with an analysis tool which overcomes many of the disadvantages of other popular signal representations for polyphonic audio analysis, such as the Short-Time Fourier Transform (STFT) which corresponds to a single level of the MFT (where the scale must be chosen *a priori*—problematic since the optimum scale is signal dependent) or the Wavelet Transform where scale is related to analysis frequency, and the ability to resolve two partials relies upon them being further apart at higher frequencies (which is not the case for polyphonic music).

Perhaps the biggest disadvantage of the discrete MFT is its size. For example, with an oversample in time of 2 and in input signal of length 2^M each MFT level requires 2^{M+1} complex

Input length = 4, 2x Temporal Oversampling used

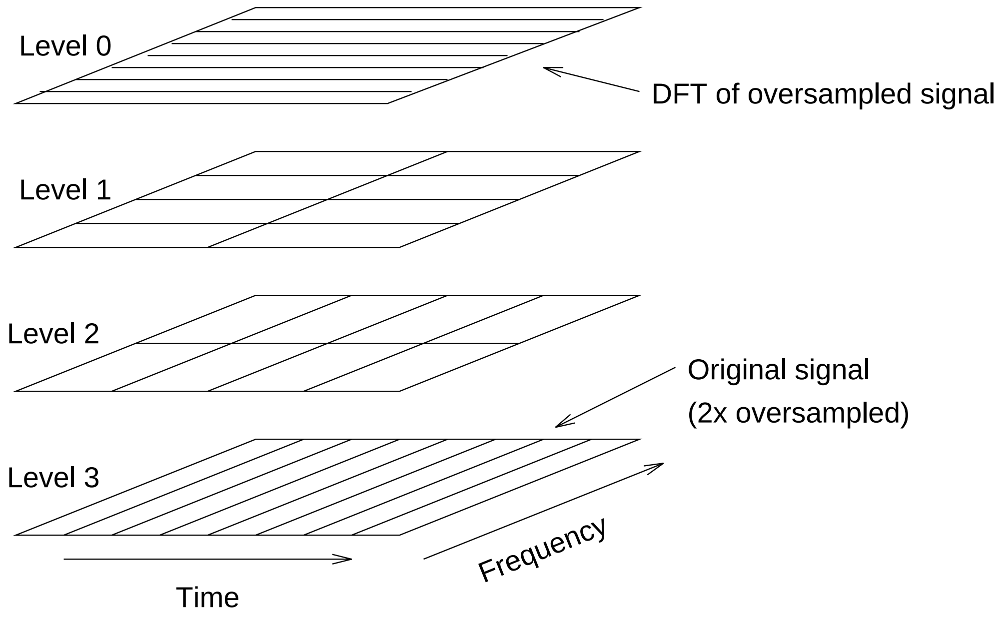


Figure 3 Structure of the discrete MFT

coefficients, and since there are M levels $M2^{M+1}$ complex coefficients are required for the whole MFT. Even taking into account the fact that, since the input is real, the MFT is symetrical about 0Hz the MFT of 10 seconds of DAT quality sound (48kHz sample rate) with complex values represented as a pair of 32-bit floating point numbers would require 38Mb of storage. Fortunately most of the very lowest and highest levels suffer from considerable amounts of temporal and frequency domain interference between signal features and hence only perhaps three or four central levels would actually contribute anything significant to processing. Even so this means that megabytes of data must be processed, even for musically short periods, and with current computers the kinds of processing that might be involved in later stages would not be practically feasible. Therefore some form of pre-processing of the raw MFT data is required to reduce the data throughput demands on later stages.

3 Beat Detection

Although the musical signal as a sound wave is continuous, at the level of the score the start of each note is quantised into a discrete time interval and the length of each note is quantised to the nearest simple rational multiple (e.g. ¼, ½, ¾) of some fixed note length. The fundamental unit of time in a piece is known as the *beat* [6]. By assuming that significant musical events (the onsets of notes) will only occur at these simple sub-divisions of the beat one may reduce the amount of data the system is required to handle (by only dealing with data from periods around the beat sub-divisions) and also decrease the susceptibilty of the system to making false inferences concerning events which occur between beats (i.e. hypothesising the presence of notes at inappropriate places in time). In order to achieve this, some estimate of the beat must first be computed.

3.1 Onset Detection

It is assumed that the beat is primarily defined by the onset times of the notes, although this can give false estimates if the first few notes of the piece are not representative of the piece as a whole. However, by computing the beat over a period of time it is assumed that a good average will be arrived at. The onsets of the notes are estimated by filtering each frequency bin of each level of the MFT through time with a high-pass filter. In previous work [16,21] the complex MFT coefficients were filtered, and use was made of the fact that, because of the 100% oversampling used, if a partial with frequency ω_i lies within frequency bin j the frequency bins $j \pm 1$ will have components of ω_i which differ in phase by π. Further, the difference in phase between adjacent time bins is a constant. However, for this work the magnitudes of the coefficients were filtered (thus losing the phase information), as suggested in [16]. This is because, in cases where there are two coincidental partials in one bin, or an excess of noise, the phase-coherence on which the algorithms rely is destroyed. That is, the adjacent bins will not in fact related in the above manner.

Using the results from [21] the data is first smoothed using a filter with impulse response (Figure 4,),

$$f(t) = e^{\frac{-|t|}{\alpha}} \tag{3.1}$$

and frequency response,

$$f(\omega) = \frac{1}{\alpha\pi} \frac{1}{\frac{1}{\alpha^2} + \omega^2}, \text{where } \omega \text{ is angular frequency} \tag{3.2}$$

This filter has the advantage that it can be split into causal and anti-causal components,

$$f_c(t) = e^{\frac{t}{\alpha}}, \quad t < 0 \tag{3.3}$$

$$f_a(t) = e^{\frac{-t}{\alpha}}, \quad t \geq 0 \tag{3.4}$$

which may be implemented as first-order recursive filters, and hence programmed efficiently. In [21] it was found that the most appropriate value of α was 2, and so it was used for this work.

The output of the smoothing filter is then used to form a first-order approximation to the gradient of the signal at time i, using the approximation,

$$\frac{\partial^+ \hat{x}(i,j,n)}{\partial t} \approx \begin{cases} \frac{|\hat{x}(i,j,n)| - |\hat{x}(i-1,j,n)|}{\Xi(n)} & , for\ |\hat{x}(i,j,n)| - |\hat{x}(i-1,j,n)| \geq 0 \\ 0 & else \end{cases} \tag{3.5}$$

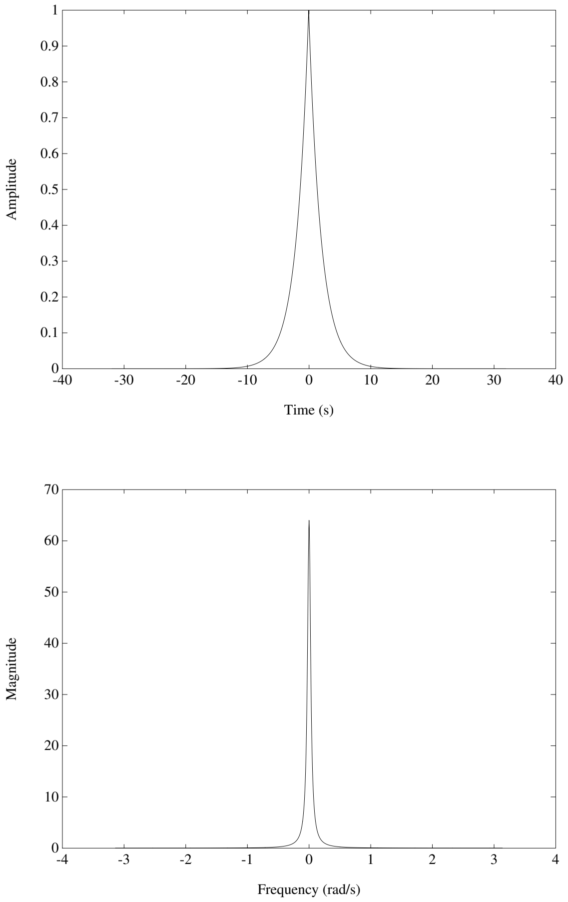


Figure 4 Time and Frequency Response of Exponential Filter

Since only onsets are of interest (positive gradients), the filter returns 0 for negative gradients.

The results of the above filtering is then summed across all frequency bins, to give a measure of the total onset energy at time t. The final stage of onset detection is to threshold this data so that peaks may be extracted (corresponding to the onsets of notes). It was found that in certain pieces the dynamic range of the music was such that the amplitude of the louder notes could create peaks of a magnitude much greater than those of the quieter notes. Therefore the sum was subjected to a form of dynamic range compression such that, for the t th value of the sum,

$$s(t) = \begin{cases} s_{\max} 10^{\gamma \log_{10}\left(\frac{s(t)}{s_{\max}}\right)} - s_{\min} & , \textit{if } s(t) > s_{\min} \\ 0 & , \text{otherwise} \end{cases} \tag{3.6}$$

where

$s(t)$ is the tth sum value

$s_{\max}$ is the maximum sum value

$s_{\min}$ is $s_{\max} 10^{\frac{-f_{dB}}{20}}$

γ is the compression factor (< 1.0)
f_{dB} is the estimated noise floor in dBs.

Once this is done, a running average is computed both forwards and backwards in time, and the average of the forward and backward averages taken,

$$s_{avg_f}(t) = \frac{\sum_{i=0}^{t} s(i)}{t+1} \tag{3.7}$$

$$s_{avg_b}(t) = \frac{\sum_{i=t}^{T} s(i)}{T-t+1} \tag{3.8}$$

$$s_{avg}(t) = \tfrac{1}{2}\left(\frac{(T-t)}{T}\, s_{avg_f}(t) \;+\; \frac{t}{T}\, s_{avg_b}(t)\right) \tag{3.9}$$

where $T + 1$ is the number of time bins.

This is done in an attempt to take account of the fact that the average signal amplitude will vary over time (for example, increasing during a *crescendo*), and hence a simple threshold based purely on the signal peak value will miss all notes in the quieter periods of the piece. Note that the point $\frac{1}{2}T$ is the mean of $s(\cdot)$.

Finally, the onsets may be computed, using $s_{avg}(t)$ as a threshold for $s(t)$. That is, an onset is detected if,

$$s(t-1) < s_{avg}(t) \;\wedge\; s(t) \geq s_{avg}(t) \tag{3.10}$$

3.2 Finding the Beat

Once the times of the onsets have been computed the beat may be estimated. Unfortunately, the time difference between successive estimates of the onsets suffers from too much variation to be used directly. It is assumed, though, that the beat is not likely to change very rapidly over a short period of time. Hence a phase-locked loop may be used to track the beat over a period of time, immune to small random variations in the timing of individual onsets, but sensitive to its longer-term trend. Figure 5 shows the phase-locked-loop used for this work. The input signal, $v(i)$ is passed through a thresholding function $\mathrm{sgn}(\cdot)$, where,

$$\mathrm{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \tag{3.11}$$

This is then multiplied with the value $\mathrm{sgn}[\sin((2\pi\omega_0 + Gv_2(i-1))i)]$, (yielding a value of 1, 0 or -1) and passed through a low-pass filter such that,

$$v_3(t) = \alpha v_2(t) + (1.0 - \alpha)v_3(i-1) \tag{3.12}$$

A delayed version of this output is fed back into the $\mathrm{sgn}[\sin(\ldots)]$ function for combination with the next input value.

Since the circuit contains non-linear sgn functions, it is hard to study analytically. However, it may be intuitively explained thus: Suppose $v^{\pm}(i)$ and $v_1^{\pm}(i)$ are in-phase (that is, both negative or both positive), and $v(i)$ is a steady signal of a given frequency. Then the

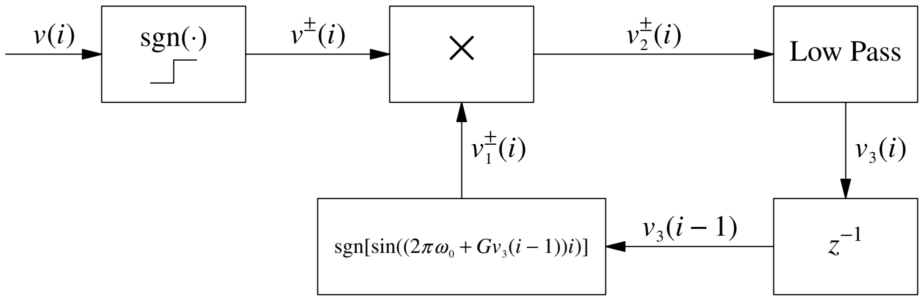


Figure 5 The Phase-Locked Loop

input to the low-pass filter would be a steady sequence $\{1, 1, 1, 1, \ldots\}$. Over time the output of the filter would rise towards a value of 1, and the value $2\pi\omega_0 + Gv_3(i-1)$ would rise, increasing the frequency of $v_1^{\pm}(i)$. This would cause $v^{\pm}(i)$ and $v_1^{\pm}(i)$ to become out of phase. Consequently a value of -1 would be input to the low-pass filter, which would cause $v_3(i)$ to fall slightly, decreasing the frequency of $v_1^{\pm}(i)$. In a steady state the value $2\pi\omega_0 + Gv_3(i-1)$ will oscillate around the frequency of the input $v(i)$

The circuit has two parameters, G and α. It has been found empirically that for this work G should be large (≈ 10) and α small ($\approx 10^{-6}$).

3.3 Onset Detection Detection Results

The above method of beat detection was applied to a number of pieces of music, with mixed results. Figure 6 shows the magnitude of level 11 of the MFT of 5 seconds of a piece by Schubert. The image (along with most of the images in this section) has been histogram-equalised in order to bring out the partials, particularly the ones at higher frequency, and hence cannot be relied up for estimating magnitudes. However, the basic structure of the MFT is clear. The piece consists of eight notes, two longer ones (at roughly times of 1 second and 2 seconds), followed by six notes in rapid succession (at times of approximately: 2.5 seconds, 3 seconds, 3.25 seconds (quite hard to discern), 3.5 seconds, 3.75 seconds (also quite hard to make out) and 4 seconds).

Figure 7 shows 7 seconds from a work by Tchaikovsky. Here the notes are much easier to discern (except perhaps the first notes after the accented notes). The notes occur at roughly: 0.2 seconds, 0.6 seconds, 1.25 seconds, 1.8 seconds, 2.7 seconds, 3.25 seconds (hard to discern), 3.8 seconds (also hard to discern), 4.4 seconds, 4.75 seconds and 5.75 seconds.

Figure 8 shows the difference-filtered version of the Schubert, with the sum across frequency $s(t)$ and thresholding average $s_{avg}(t)$ overlaid (the magnitude of the overlay is not significant—it is scaled to fit the image). The peaks corresponding to the notes in the piece are shown quite clearly. The threshold function also seems to pick out the notes quite well, apart from after the first note, where it encounters some apparent noise. This is actually due to an amount of vibrato in the first note. It is played by a violin, and since it is long, it is difficult for the performer not to introduce vibrato. The effect on the MFT of vibrato (a slight variation in the frequency of a note through time) is to cause the magnitudes to rise and fall as the note “leaves” a frequency bin and returns to it a short period later. The

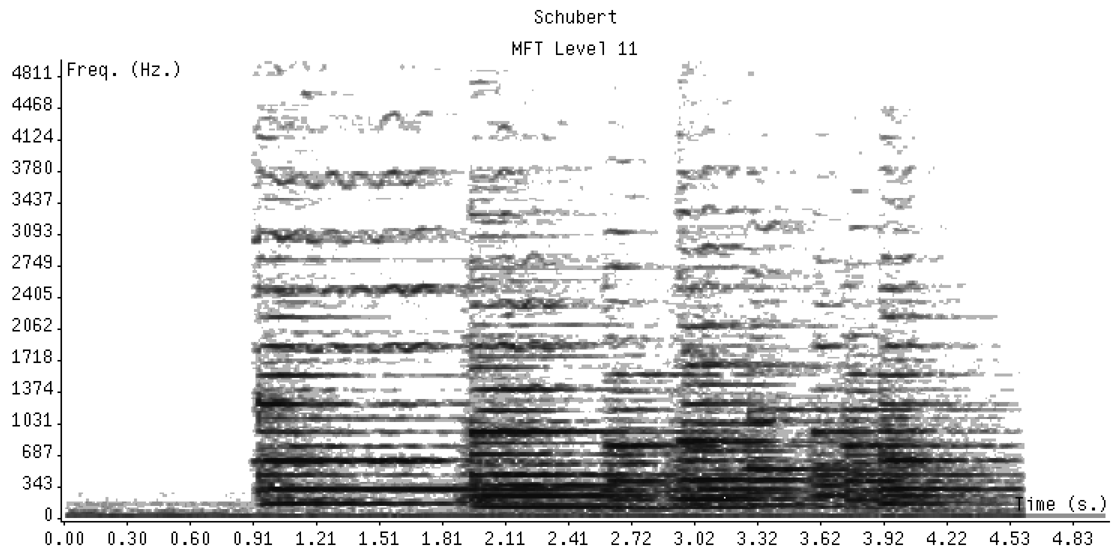


Figure 6 Schubert Initial MFT Magnitudes

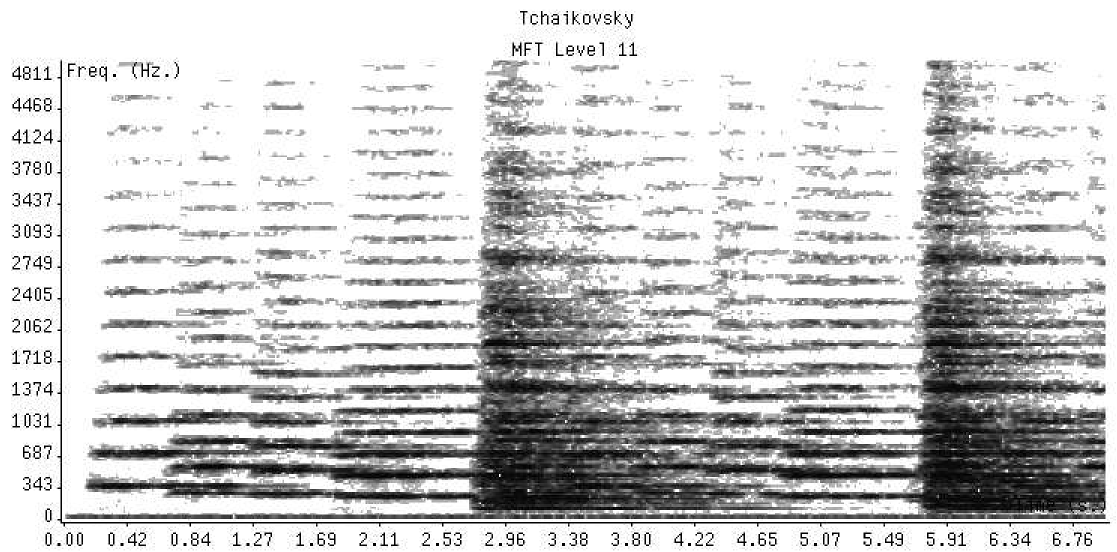


Figure 7 Tchaikovsky Initial MFT Magnitudes

difference-filter therefore sees the vibrato as a rapid succession of onsets. The fact that the vibrato spikes rise and fall may also indicate that a certain amount of amplitude modulation is also present at that time, again common for sustained String Section notes.

For the Tchaikovsky the results are less good. Figure 9 shows the difference-filtered MFT magnitudes with sums overlaid. It is immediately apparent that the peaks for majority

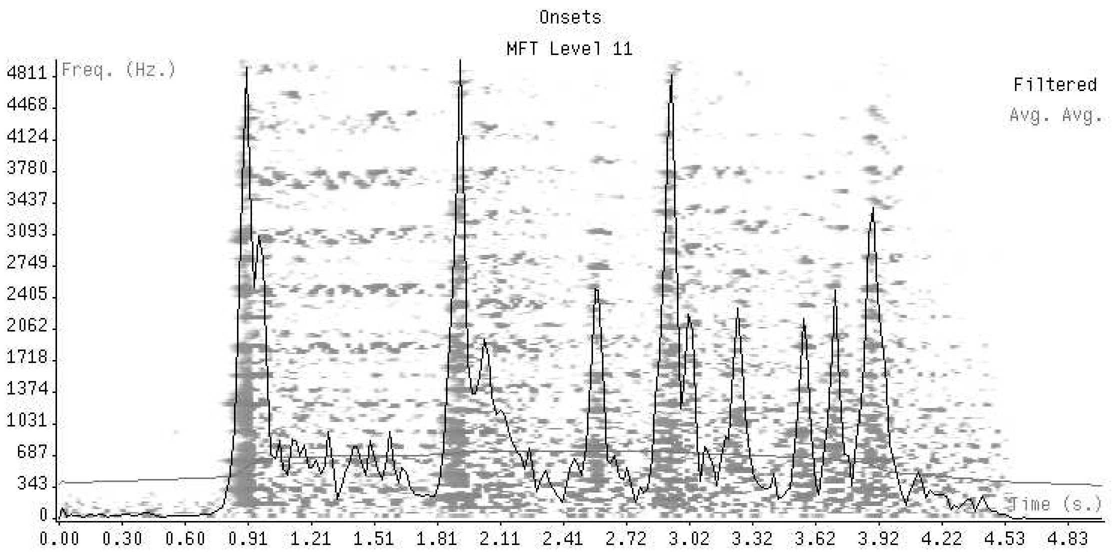


Figure 8 Schubert Difference-Filtered MFT Magnitudes with Across–Frequency Sums Overlaid

of the notes are dwarfed by the peaks for the two accented notes. However, at this stage, no compression has been applied to the sums. With a compression factor, γ, of 0.3 the result is as shown in figure 10. Here it is clear that the peaks have been compressed, but unfortunately the noise has been increased and thus a number of spurious additional spikes introduced.

An alternative approach to the normalisation process was attempted. Instead of compressing the dynamic range of using (3.6), the MFT coefficients were normalised (divided by) the output of a filter which gave a weighted measure of the signal energy looking forward and backward through time. Because of the temporal length of this filter (around 1 second either side of the current time bin), it's value is relatively stable over time, and thus the division operation is not overly sensitive to noise in the input. The particular filter chosen was a Gaussian filter, though this particular choice is not thought to be critical. The impulse response of the filter is

$$f(t) = -e^{\frac{-t^2}{\alpha^2}} \tag{3.13}$$

This response is truncated to the required length, and the filter implemented as an FIR filter. The effect of filtering with the normalisation filter for the Tchaikovsky work is shown in figure 11. In addition to the normalisation process, a mild filtering of the computed frequency sums is also performed, to attenuate some of the 'spikes' in the sums, which can cause an occasional incorrect onset detection.

3.4 Phase-Locked Loop Results

The system diagram of Figure 5 leads to a quite straightforward implementation of the phase-locked loop. However, it suffers from the disadvantage that it can take a considerable time to converge on the required frequency. Varying the parameters α and G goes some way toward improving the situation, but the system is somewhat susceptible to becoming

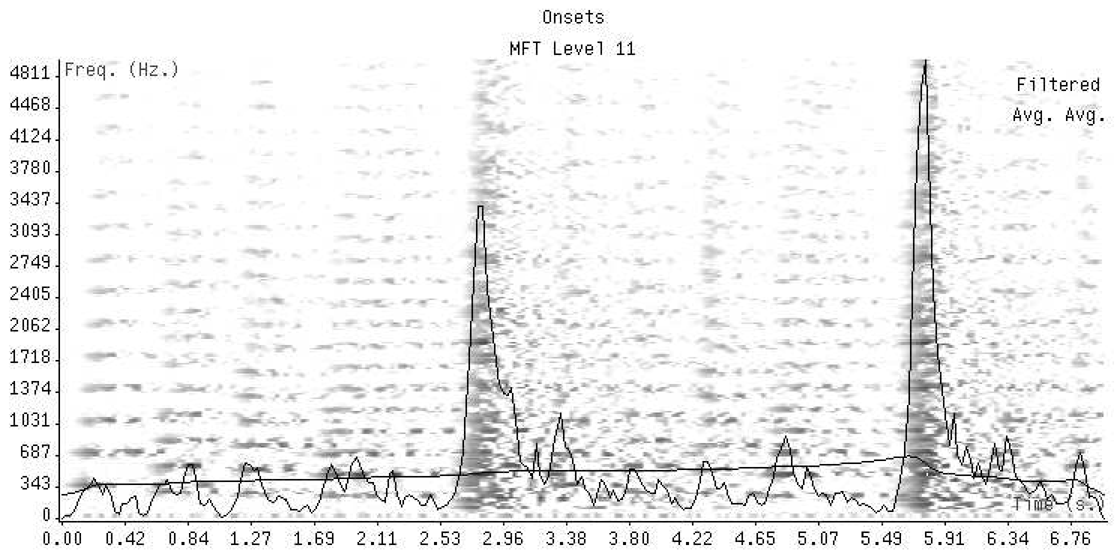


Figure 9 Tchaikovsky Difference-Filtered MFT Magnitudes with Across–Frequency Sums Overlaid

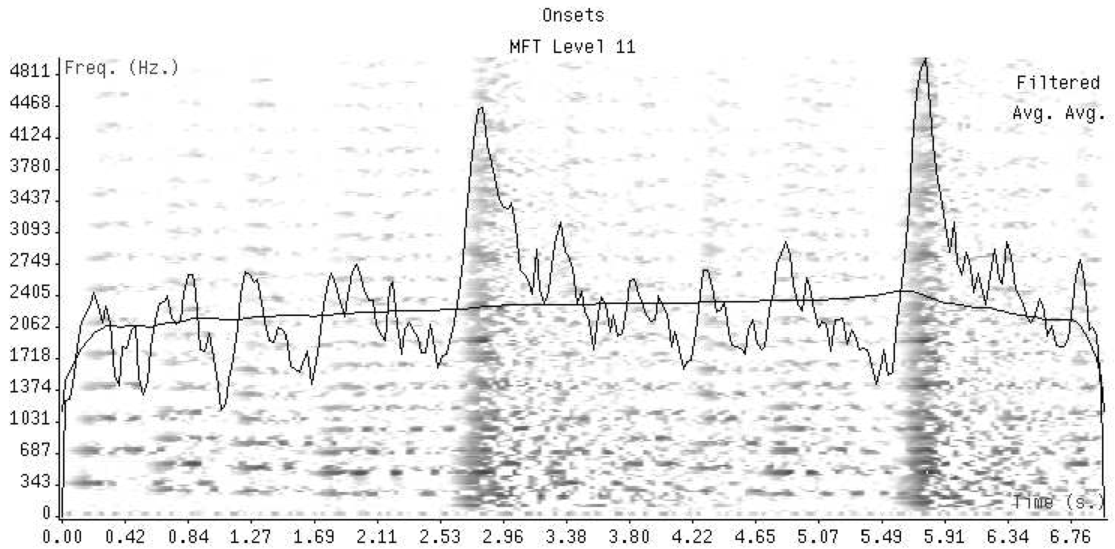


Figure 10 Tchaikovsky Difference-Filtered MFT Magnitudes with Compression Applied to Across–Frequency Sums

unstable if they are varied too much. In order to overcome the problem a technique of oversampling the input $v(i)$ was adopted, whereby each input is presented to the circuit a fixed number of times in succession. Figure 12 shows the effect of increasing the amount of oversample. It shows the output frequency of the phase-locked loop over time for an input

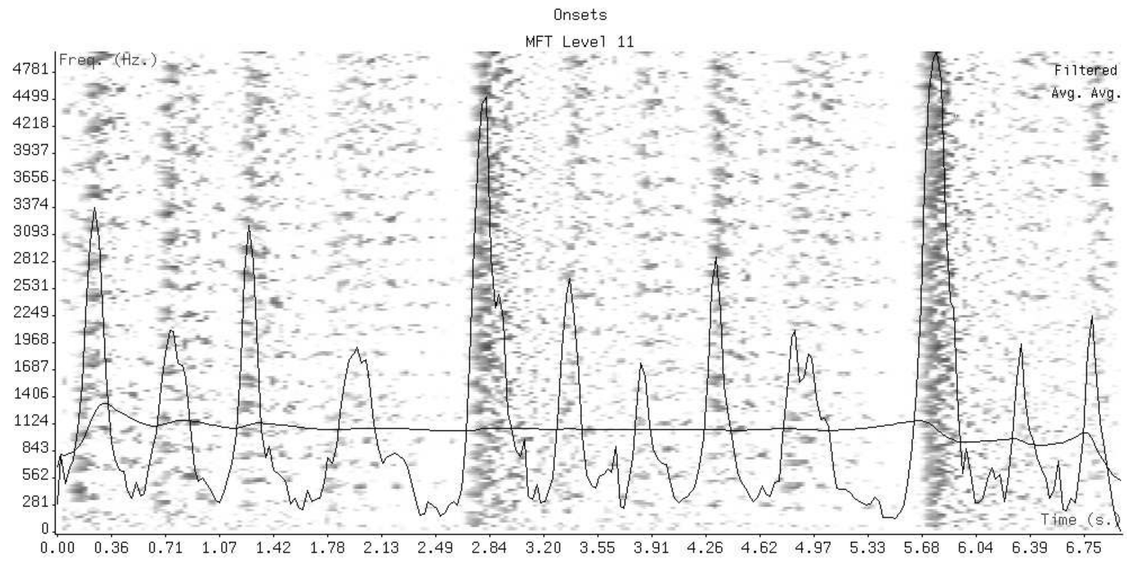


Figure 11 Tchaikovsky Filter Normalised Magnitudes with Across–Frequency Sums

frequency of 150Hz. for oversampling at 16 and 64 times. The higher oversample rate approaches the desired frequency more quickly. Of course, there is a limit on the amount of oversample which can be applied, since doubling the amount of oversampling doubles the amount of computation for each output value of the phase-locked loop.

Figure 13 shows the output of the phase-locked loop for both Schubert and Tchaikovsky. Unfortunately, due to the previously discussed imperfections in the onset detection stage, coupled with the relative shortness of the sequences (the phase-locked-loop normally taking a few thousand samples to lock onto the frequency of the signal) the results cannot be taken to be very reliable. It is interesting to note, however, that the result for the Tchaikovsky (computed using onsets detected without compressed dynamic ranges) approaches a value quite close to the reciprocal of the arithmetic mean of the interval between the roughly estimated onset times of section 3.3. Though the Schubert does not reach such a similar state with respect to its average inter-onset times, it must be noted that the particular piece of Schubert is taken from the start of the piece and a firm beat is not really established. (The standard deviation of the inter-onset intervals, based on approximate measurements from the MFT is significantly higher in the case of the Schubert than of the Tchaikovsky.)

3.5 Further Work on Beat Estimation

Secondly, the use of a phase-locked loop for beat estimation does not appear promising, since it takes a large number of samples to converge on the correct frequency. Human listeners seem to have little difficulty in finding the beat of most pieces on the basis of perhaps two or three notes. It is anticipated that, with the reduced susecptibility to false onset detections afforded by the use of the normalisation filtering described above, a simpler beat detection algorithm may be used which simply measures the time interval between consecutive onsets and quantizes to the nearest musical time division of a moving average beat estimate. Thus, if an onset is detected at approximately half the current beat estimate it

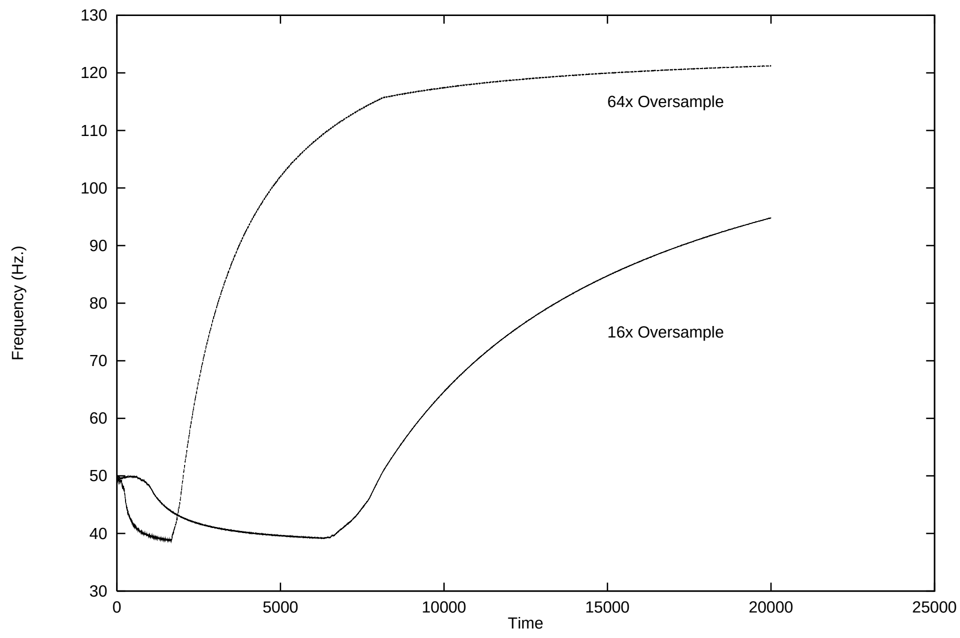
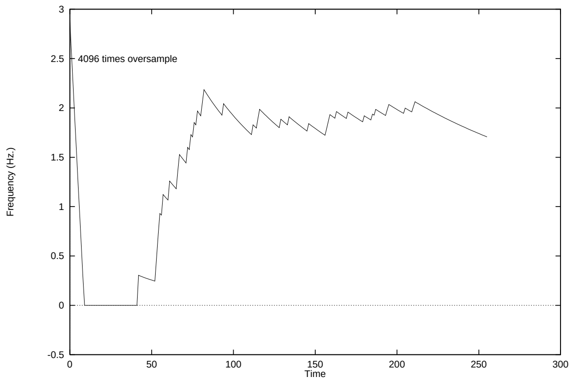
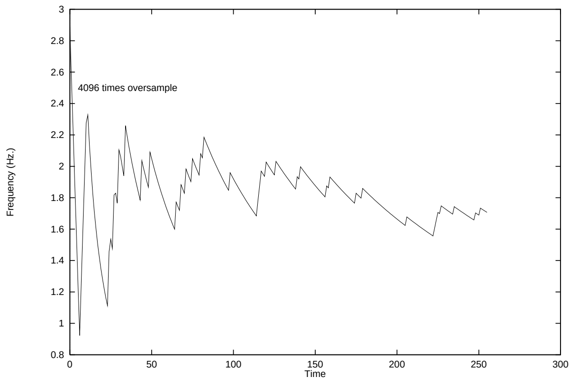


Figure 12 PLL Output Frequencies for Different Amounts of Oversample. (Input is at constant 150 Hz.)

is assumed that it occurs half way between two beats. The smallest possible subdivision of the beat is the *demisemiquaver*, which is $\frac{1}{32}$ of a beat, and the largest multiple is the *breve*, which lasts 8 times the beat, assuming in both instances that the beat is the length of a crotchet. However, in any one piece it is unlikely that such a range of durations would be encountered, and it may be possible in most instances to deal with quantisations of the order of ¼ of the beat to 2 times the beat.



Schubert



Tchaikovsky

Figure 13 Estimated Beat Frequencies

4 Pitch Tracking

A small amount of preliminary work was performed on the tracking of pitch trends in the input signal. Such information would be useful in giving an indication of the "melody countour" of piece. A great many pieces of music follow patterns of rising and falling sequences of notes. The starting point for this work is to consider the centroid frequency of the signal in each time bin. The centroid of the sequence s(i), $0 \le i < N$, is defined as,

$$\sum_{i=0}^{i=N} i\ s(i) \ \Big/ \ \sum_{i=0}^{i=N} s(i) \qquad (4.1)$$

Using (4.1) it is possible to compute the centroid frequency for each time bin, and thus track the centroid through time. Figure 14 shows the results of applying the above computation to the time bins of a simple sequence of time-separated tones, of random pitch. For each tone the strength of the harmonics decreases in inverse proportion to the harmonic number, and the centroid is thus centered near the second harmonic.†

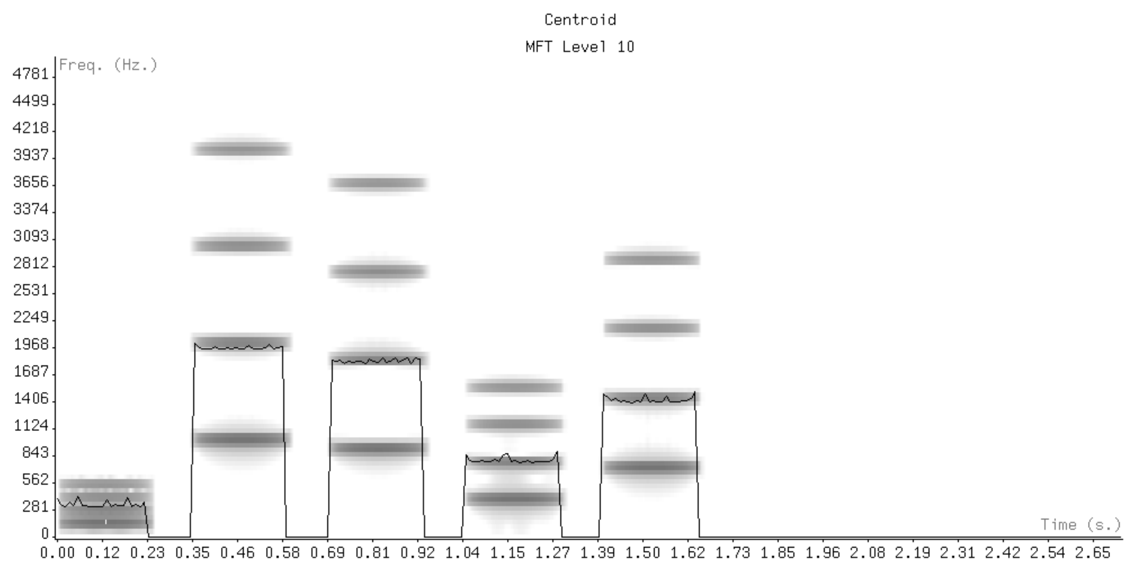


Figure 14 Centroid frequencies for a set of simple tones

Figure 15 shows the centroid of one MFT level of a section of Schubert's "The Trout" [20]. In the this section the violin is playing pairs of (relatively long duration) notes starting on higher and higher notes, while the piano plays a more rapid sequence of descending notes, repeated in runs (just visible as a "staircase" effect on the MFT).

Although it is possible to visually discern the runs of piano notes, the centroid in general does not follow them, nor does it rise with the sequences of violin notes (though it must be said that the temporal resolution of this level is better suited to discerning features of the piano's performance). The reason for this is that energy from the violin notes is still present while the piano notes are being played, and this energy distribution has its own

Since the centroid = $k(1.1 + 2.\tfrac{1}{2} + 3.\tfrac{1}{3} + 4.\tfrac{1}{4}) / (1 + \tfrac{1}{2} + \tfrac{1}{3} + \tfrac{1}{4}) = 1.92k$

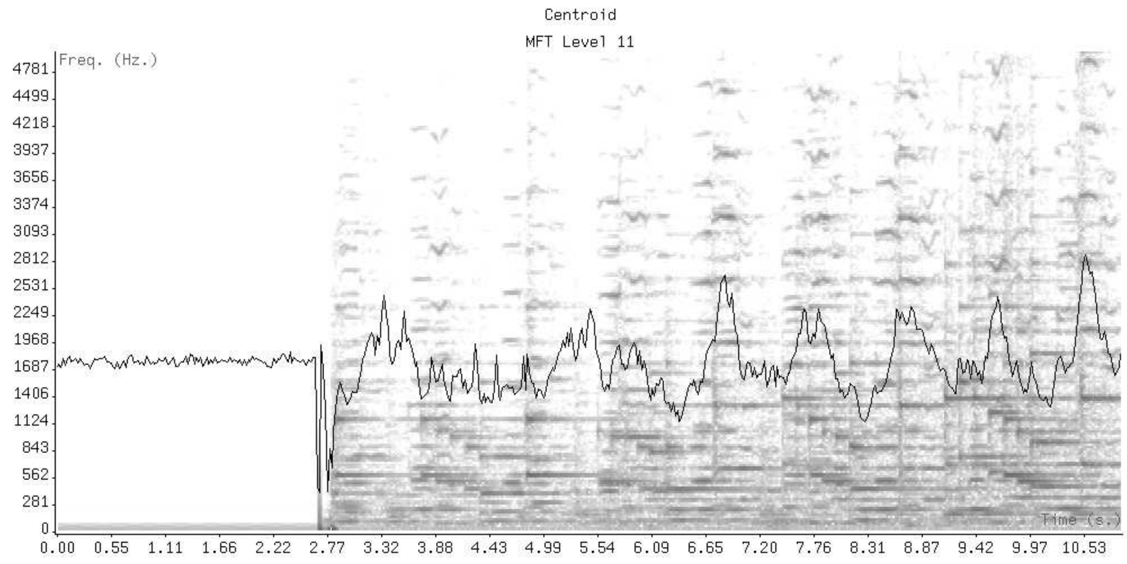


Figure 15 Centroid computation for Schubert's "Trout"

harmonic structure whose centroid is moving upward with each new note, and interfering with the downward motion of the piano's centroid. However, by only considering the onsets of the notes (detected as described in the previous chapter), the energy of the violin notes can be better separated from that of the piano. Unfortunately, as figure 16 illustrates, a new problem emerges. At times when there is no onset present, and hence little or no energy in the filtered MFT coefficients, the energy distribution is similar to non-uniform broad-band noise. Perceptually, such noise would not have a strongly defined pitch, and the centroid provides little useful information (see figure 17).

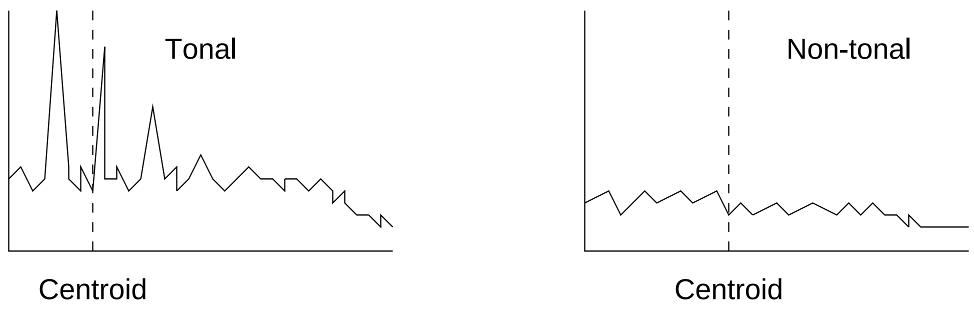


Figure 17 The centroid of tonal and non-tonal components

In order to prevent the centroid of a non-tonal component being erroneously interpreted as providing pitch information, a "tonality detector", could be employed, perhaps similar to the one described as part of Annex D of [7]. This would allow the system to consider the centroid only in instances where there appears to be a definite harmonic structure. Currently this has not been tested, but since the energy in the non-tonal

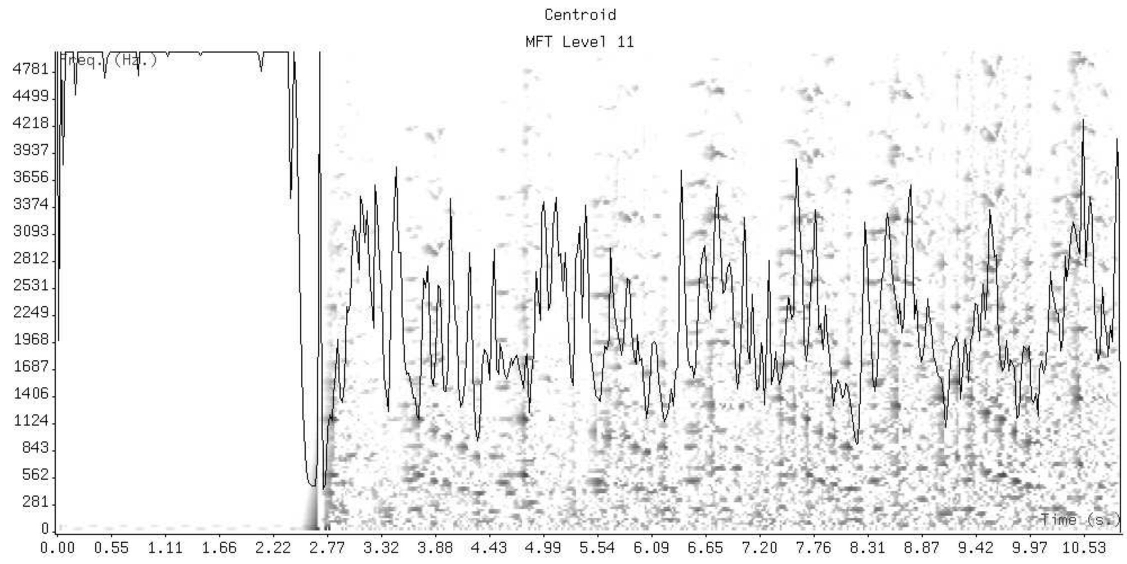


Figure 16 The centroid of the onsets

components is small compared with that in the tonal components, a crude attempt at a solution has been tried, whereby the centroid is not considered unless the total energy across frequency in each time bin exceeds some (manually set) threshold. In figure 18 it can be seen that this yields some improvement (for example in the first piano run at the start of the piece), but it is still, perhaps, not immediately useful as it stands.

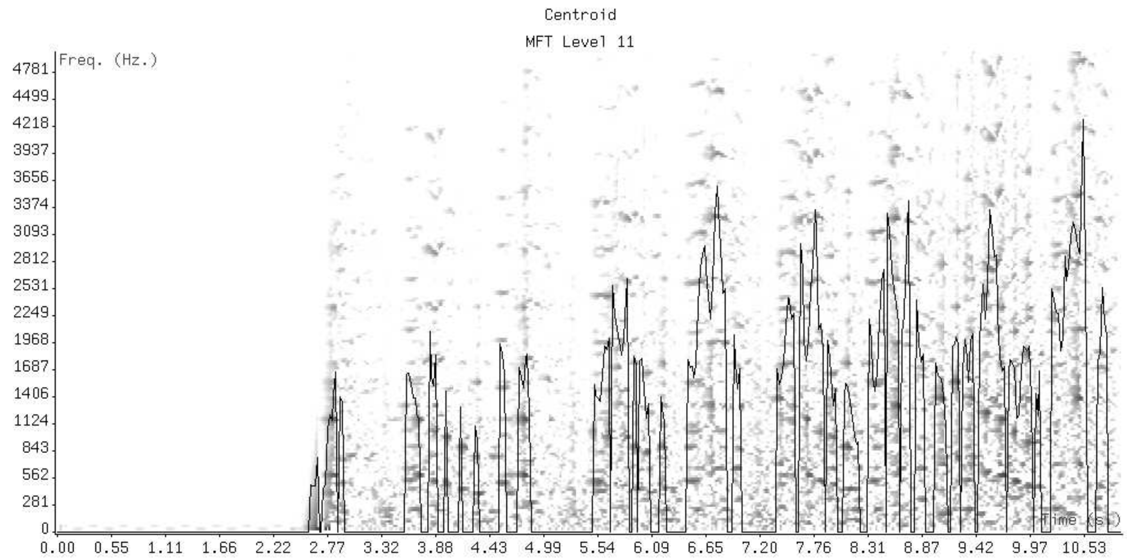


Figure 18 The centroid with non-tonal components eliminated by thresholding

5 Higher Level Structure Representation

Once the incoming audio signal has been pre-processed to reduce amount of information to be dealt with, some method must be used to segment the sets of harmonics (partials) into individual notes and chords. This process is complicated by two main facts. Firstly, there may be several polyphonic note structures corresponding to one set of partials (e.g. one note, two notes an octave a part, three notes in each an octave from the next and so on). Some of these structures are obviously nonsense (for example, it is highly unlikely that a piano concerto would contain nine notes played in octaves—the pianist would require fingers long enough to span the entire keyboard!). However, this obviousness is due to knowledge of musical structure, rather than any *a priori* property of the signal (one could, for instance, readily program a piano synthesiser to play 9 or more notes in octaves). A second complication arises from the fact that, in general, the lower level processing will not give rise to perfect results. Partials will be missed (perhaps because they contain little energy, or they interfere with harmonics from other notes), and spurious partials will be detected (again, because of interference, which causes a beating effect that can 'look' like an onset).

Any representation of musical structure must deal with at least two major aspects of music: temporal structure and tonal structure [5]. The representation of timbral structure is also an important and complex issue in general, but since timbral information is not explicitly represented in a score (other than by the naming of the instruments in a pieces orchestration), much of the work on musical representation essentially ignores this aspect.

5.1 Representations of Tonal Structure

The term 'tonal structure' is used rather loosely here. It is a generally accepted fact that music is arranged hierarchically [1,11,24]. At the lowest level there is the structure of individual notes. Pitched instruments exhibit a harmonic structure at this level (although real instruments also contain inharmonic elements, particularly during the transient onsets). The fundamental frequency, of which all the other harmonics are integer (or, more realistically, very nearly integer) multiples of defines the pitch of a note. Groups of notes are arranged into a small number of distinct notes, known as a scale. The structure of the scale plays an important part in defining the kinds of structures which will occur at higher levels in the music. Most scales choose notes so that the ratios of the fundamental frequencies of the notes in the scale correspond to small integer rationals. For example, the fifth note of the standard diatonic scale based on C has a ratio of $\frac{3}{2}$ to the fundamental. Indeed the standard Western scale has a number of interesting Group theoretic properties and geometric properties, and a considerable amount of research has concentrated on the structure of musical scales [2,22].

The next level of structure, that of the interval and chord comes into play when two or more notes from the scale are to be played together. Due to the harmonic structure of each note, and the fact that the notes on the scale are chosen to approximate simple rational numbers, any two notes taken from the scale and played together as an interval will share all or some of their harmonics. Depending on which harmonics are shared any interval will sound *dissonant* or *consonant* [6]. In the standard diatonic scale and with an equal-tempered tuning, unisons, octaves, perfect fifths and perfect fourths are known as *perfect consonances*, whereas major seconds, minor seconds, major sevenths, minor sevenths and augmented and diminished intervals are known as *dissonances*. All other intervals are *imperfect consonances*. For example, a perfect fifth shares harmonics at all of the first six harmonics of the root, whereas a major second shares *no* harmonics up the the 7th. The interplay of consonance and dissonance, and in particular the smooth resolution of dissonance is one of the major structural aspects of a musical work.

At the highest level†, the score becomes much more significant, and the temporal aspect of music comes into play. However, as a purely static symbolic representation, the score can be broken up into a hierarchy of levels, such as sections, movements, phrases and so on. In this regard, many authors have seen a similarity between musical structure and the syntactic structure of written language [4,11]. Although this work is not overtly aimed at producing a psychologically plausible account of the human listening process, because the music of interest is written by human composers for human listeners (both groups being limited by similar constraints on short-term memory capacity, although the former being in the advantageous position of being able to compose in non-real-time and make use of external memory aids, such as paper on which to write the score), the structure of music is in part defined by the psychological limitations of human beings. As noted in [11], “One can imagine some mathematical relationship to obtain between every tenth note of a piece, but such a relationship would in all likelihood be perceptually irrelevant and musically unenlightening”. However, by organising a piece in a hierarchical manner, the listener is able to ‘compress’ the representation of a piece, and hence become aware of relationships between notes considerable distances apart. Clearly a music transcription system could make use of this hierarchical structure information, but it is most likely that the greatest amount of useful structural information for the note recognition task will reside at lower levels of the structural hierarchy, covering short segments of a piece, perhaps no more than a bar or two. This is also advantageous from a processing point of view, since it implies that the piece might be processed in sequential, overlapping sections.

Although the different levels of the hierarchy are in many ways distinct, and it is often useful to think of them as operating in isolation, there are a great many ways in which the levels interact. For example, the structure of the musical scale may be viewed both in terms of a set of constraints on a Group-theoretic structure, as in [2], but it is equally (and historically) due to the requirement of having an acoustically pleasing set of notes with which to build intervals and chords. Because each level of the above hierarchy is in itself quite complicated, most of the practical work done on music representation has concentrated more or less on one particular level or another, and in parallel with the development of techniques in Artificial Intelligence, (which appear well suited to the description of the hierarchical relationships involved in music representation), work in the area has predominantly concentrated on symbolic, ‘lisp-like’ representations. However, such representations, whilst powerful in themselves, are difficult to integrate with one another between levels. Therefore for this work it is proposed that the underlying architecture for representation of music will be in the form of artificial neural networks [17]. The use of neural networks in music representation is relatively recent, and there are a number of important issues still to be addressed with regarding how best to represent musical objects in neural architectures, in particular the the representation of sequential, temporal event sequences. However, there now follows a brief summary of some of the existing practical work on representation of tonal structure, with an emphasis on work of a connectionist flavour, highlighting the basic structural devices used, and the conclusions presented as to the suitability of the representation. The general conclusions from this work are then drawn.

Beyond the purely syntactic level, there appears to be little research aimed at studying the emotive aspects of music, although some interesting work on the interpreting the emotion in traditional Japanese music is briefly described in [8].

5.1.1 Sano and Jenkins' Pitch Perception Model

Sano and Jenkins, [18] develop a neural model based upon the structure of the human cochlea. They note that the essential structure of the cochlea is that it contains a number receptor cells connected to hairs which are tuned to different frequencies on a logarithmic frequency scale, and each receptor cell can be modeled as a low-order, linear, bandpass filter, centered on a characteristic frequency x and of bandwidth $\frac{x}{10}$. Each hair is has a characteristic frequency 0.02% higher than the one below. However, the *just noticeable difference* versus frequency curve for humans is approximately 0.3% in the range 500Hz–2kHz. Therefore groups of cells in the model cochlea are fed into 'JND buckets', such that the output of cells that fall within the ±0.3% of the centre frequency of each JND bucket are fed to that bucket. The buckets produce an excitational output value if more than half the feeding input units are active. The JND buckets are then fed to a layer of semitone buckets, with an overlap so that JND buckets at positions near to half-way between two semitones excite both semitones. The semitone buckets represent the fact that the interval of a semitone corresponds to a 6% frequency spacing. At this point the semitone buckets may be labeled with the pitch class and height of the note to which they correspond. The semitone buckets are also provided with laterally inhibitory connections. The semitone buckets are then connected to a set of note neurons, so that the fundamental and its harmonics are tied together. Finally the notes are split into a pitch class and pitch height representation. Figure 19 illustrates the complete architecture.

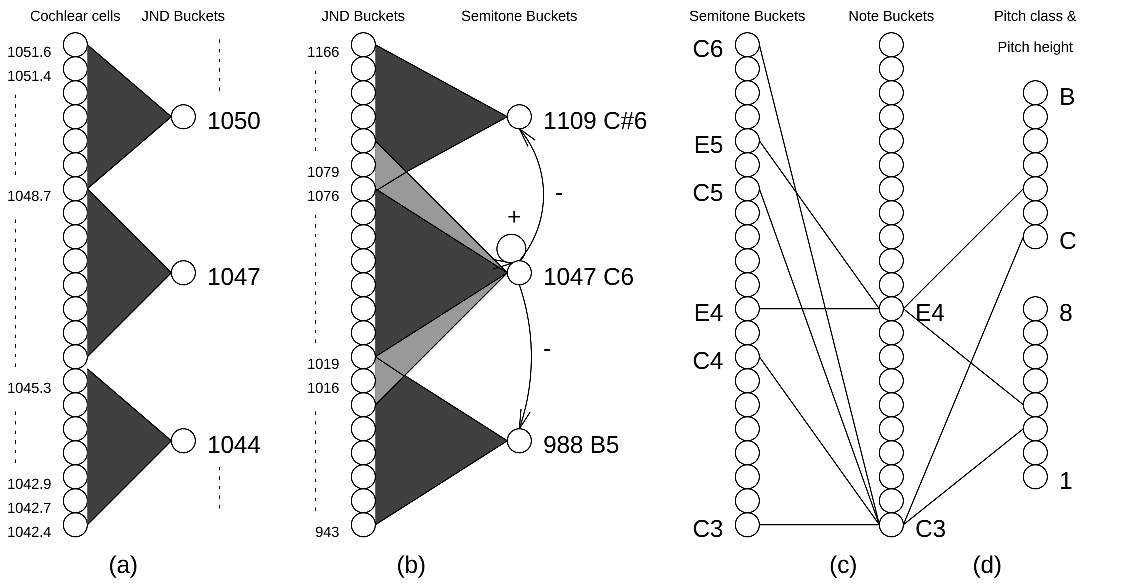


Figure 19 Sano and Jenkins' Model (Source: [18])

This model has several useful features. Firstly, it allows for the fact that for real instruments, the harmonics of a note may not correspond exactly to their ideal (integer multiple) values, either due to the characteristics of the instrument, or the fact that it is imperfectly tuned. On the negative side, the models requires a large number of neural units (around 7,500), although it is intended as a biologically plausible model, rather than an efficient computational device. Further, it can only deal with monophonic inputs. In the addendum to [18] extensions to the model which would allow it to deal with multiple tones (such as chords) are described. This involves adding further laterally inhibitory connections at the pitch unification stage (stage (c) in figure 19). These connections cause a note to inhibit the notes which correspond to its harmonics, with the inhibitory strength decreasing with the harmonic number). The idea behind this is that a note effectively subtracts its harmonics from the output. However, this relies upon having a lateral inhibition in which the inhibitory weights are related to the relative strengths of the harmonics, a factor which varies from instrument to instrument.

5.1.2 Scarborough, Miller and Jones' Tonal Analysis Model

The model described by Scarborough, Miller and Jones in[19] is a linear network for tonal induction. The basic architecture is show in figure 20. The network consists of three layers of linear neurons. The input layer has 12 units, one for each note of the chromatic scale. A neuron in this layer is excited by the presence of an input containing a frequency component which corresponds to the fundamental frequency of the corresponding note (note that that the octave information is not significant. Thus the A unit will be activated by the presence of frequencies of approximately 27.5Hz, 55Hz, 110Hz, 220Hz, 440Hz and so on). The level of activation in the input units is proportional to the duration of the note, since longer notes tend to have a greater effect on the perceived tonality. Furthermore, the activation of an input unit does not stop when a note stops, but rather decays with time. The middle layer contains one unit for each of the major and minor chords (only the major ones are shown in the diagram). Each of these units connects to the input units which correspond to the notes of the corresponding chord. The chords are then connected to a layer of units for each of the possible keys. Because of the decay in activation through time, the output at the key layer represents a weighted sum over all notes that have occurred, with more emphasis given to recent notes.

As noted in [19] this network has the advantage that it is easy to understand, and computationally efficient. However, its main disadvantage is that the temporal element of the music is only represented through the decay in activation of the units. It is more likely that timing aspects, such as whether a chord falls on a strong or weak beat, will have a significant effect on the decision about the key of piece. At present there is no way to incorporate these aspects into the model.

5.1.3 Laden and Keefe's Chord Classifier Network

Laden and Keefe describe work done on the classification of chords as major, minor or diminished triads. Initially they used an architecture like that of figure 21. An input unit in the network is activated if the chord to be classified contains the corresponding note. Since there are 36 possible major, minor and diminished triads, the system can only be considered to have learned something 'deep' about the structure of chords if it contains less than 36 hidden units. Various numbers of hidden units were tried, ranging between 3 and 25. It was found that for smaller numbers of hidden units, the successful classification rate was unsatisfactorily low. Therefore attention was switched to alternative representations of the input.

By adopting a harmonic complex representation, such as that illustrated in figure 22 it was possible for the network to correctly classify 35 out of 36 of the input chords (97% success rate) after only 5000 epochs of training. Furthermore, the network was able to correctly identify chords when one or more of the harmonics was missing, or when input activations were made proportional to the relative harmonic strengths for actual instruments.

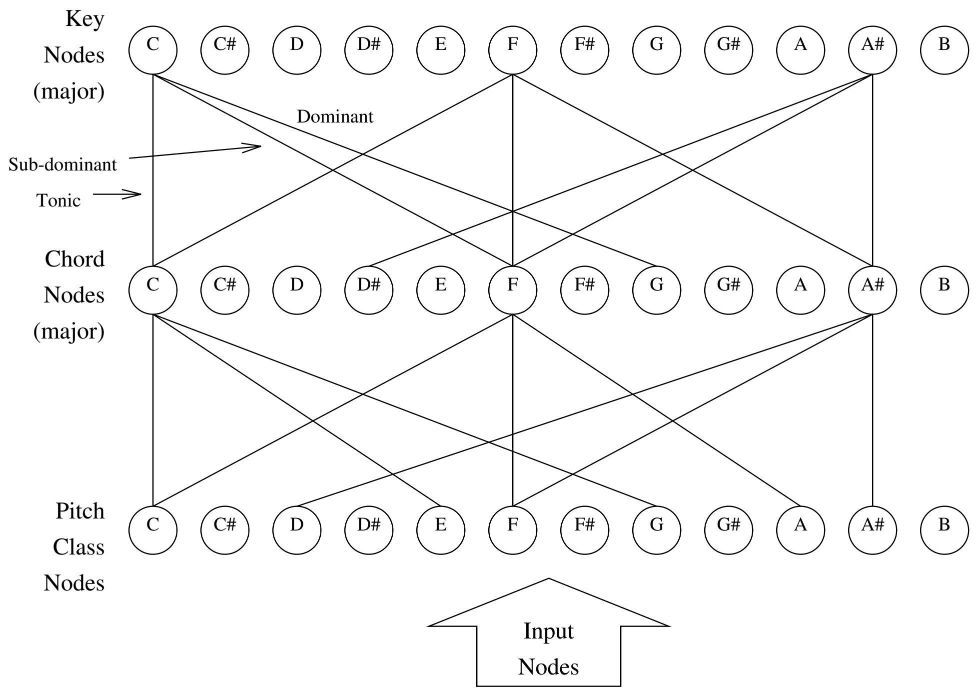


Figure 20 The Scarborough, Miller and Jones Network (Source[19]):

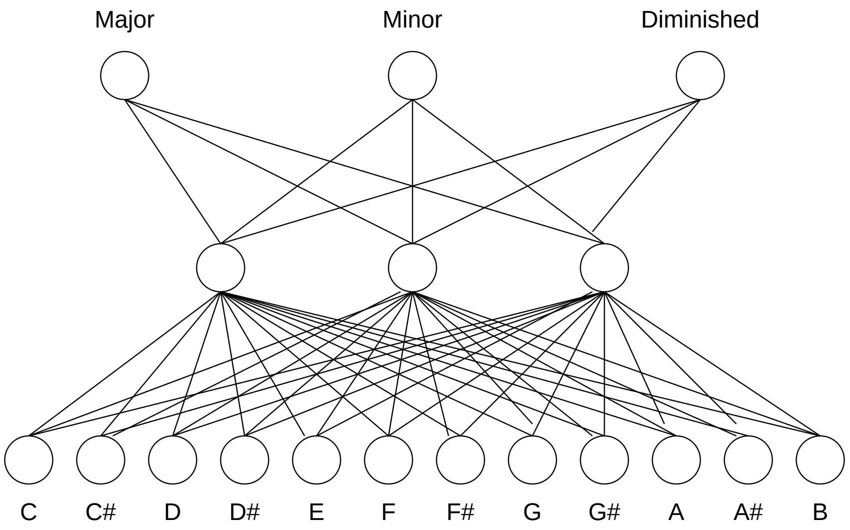


Figure 21 Laden and Keefe's Initial Network (Source: [10])

5.1.4 General Conclusions about Tonal Structure

The most striking similarity between the various approaches to the representation of tonal structures is that almost all of them use a harmonically rich representation. Perhaps

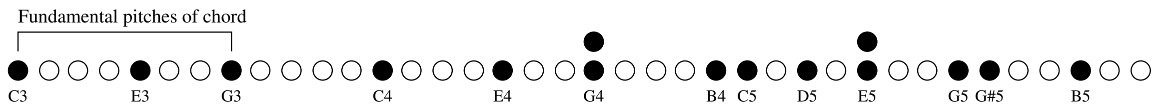


Figure 22 Harmonic Complex Representation (Source: [10])

this is not surprising, since much of the richness of the tonal structures in music derives from the interactions of the harmonics in intervals and chords. The fact that a harmonic structure appears to be most useful also fits in nicely with the fact that the MFT can provide a set of partials which correspond to the harmonics of the incoming signal. There seems, however, to be a lack of consensus as to whether pitches should be represented independent of octave (as in the Scarborough, Miller and Jones network), or dependent of octave (as in the Harmonic Complex representation of Laden and Keefe). To a certain extent the answer to this question would depend upon the type of problem the representation is to be put to use in solving. Both representations are essentially one-dimensional, and as has been shown in the work of Shepard [22] such representations may be thought of as projections of a more complex multi-dimensional structure. Another disadvantage with these representations is that they give no real clue as to how one might elaborate them to incorporate temporal aspects of music. The key recognition network of Scarborough, Miller and Jones does have some representation of time in the decay parameters for the activation of the units, but as the authors freely admit this is hardly an adequate representation of the undoubtedly complex phenomenon of musical time. One possible architecture, not generally couched in neural terms, but expressible in such terms, which goes some way to bridging the gap between the static tonal structure and the temporal structure of music is the Hidden Markov Model, already in common use for the recognition of speech.

5.2 Hidden Markov Models

Hidden Markov Models (HMMs) are described in many places in the speech processing literature, such as [12]. A brief summary of the underlying terminology used will be given here.

Consider an observation O_t made at time t. Let $\mathbf{O}$ represent the sequence of observations up to time T, i.e. $\mathbf{O}$ is the sequence $O_1, O_2, \ldots, O_T$. Consider also a set of states $q_1, q_2, \ldots, q_n$. Let $b_j(O_t)$ be the probability that the t^{th} observation is made in the j^{th} state. Suppose that there are a finite number, m, of events which may be observed. Then $b_{jk} = b_j(O_t)$ when O_t is an observation of the k^{th} event, $1 < k \leq m$. Let $\mathbf{B}$ be the matrix $[b_{jk}]$, $1 < j \leq n; 1 < k \leq m$. Let $\mathbf{A}$ be the matrix of probabilities $[a_{ij}]$, $1 < i, j \leq n$, where a_{ij} is the probability of a state transition from q_i to q_j. See figure 23.

The matrices $\mathbf{A}$ and $\mathbf{B}$ are the *parameters* of the HMM. By computing $\mathbf{A}$ and $\mathbf{B}$ for each class of input sequence (each word in the vocabulary, if the HMM is applied to speech processing), one can classify a unknown input sequence $\mathbf{O}$ as the word w such that $P(\mathbf{O}|\mathbf{A}_w, \mathbf{B}_w)$ is a maximum. However, the problem remains to compute $\mathbf{A}$, $\mathbf{B}$ and $P(\mathbf{O}|\mathbf{A},\mathbf{B})$. Fortunately, a quite efficient method exists for the computation of $P(\mathbf{O}|\mathbf{A},\mathbf{B})$, known as the "Forward–Backward Algorithm". A number of local optimisation methods may be used to compute $\mathbf{A}$ and $\mathbf{B}$, a common one being the "Baum–Welch" algorithm. Both algorithms are described in [12].

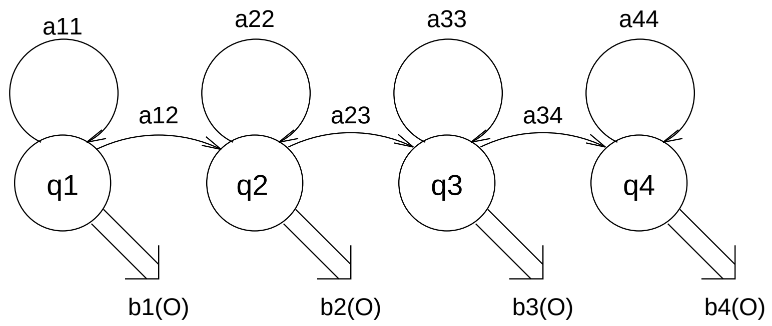


Figure 23 A Hidden Markov Model (source: [12])

Niles and Silverman [15] demonstrate that a HMM can be viewed as a class of recurrent neural network, in which the connection weight matrices correspond to the matrices **A** and **B**. Because these matrices contain probabilities they must all be positive and sum to 1. However, relaxing these constraints yields a neural network which can no longer be seen as a HMM, but which can still effectively classify. In [15] it is found that the most interesting results are obtained by relaxing the constraint that the weights must be positive. This introduces inhibitory connections into the network, allowing the presence of certain input features to preclude the incorrect interpretation of other input features. Without the relaxation of this constraint, the best that could be done was for input features to not contribute positively to the activations of other output features. Because the weight matrices can now contain negative numbers, they can no longer be viewed as probabilities, however Niles and Silverman show that an analogy can be made with the mathematics of quantum mechanics. Whether this interpretation offers any additional theoretical advantage is not clear, and no experimental results are given in [15] , so it is not clear that it offers any practical benefit, either. However, it seems likely that such a network would offer some sort of performance benefit over a pure HMM.

5.3 Applying HMMs to Music Recognition

In speech processing the HMM is typically trained on spectra corresponding to the sub-parts of speech (phonemes, syllables etc.) A word is modeled by a sequence of states. The input spectra features are usually vector quantized, so that there are a finite number of them (the number m in the above discussion of HMMs). The forward–backward algorithm can be applied and the word with highest probability accepted. However, it is also possible to build a hierarchy of HMMs, with additional levels representing the syntax and basic semantics of the language being recognised [12].

There are two major problems preventing the direct application of this kind of structure to polyphonic music recognition. Firstly, there is the question of what to use as the musical equivalent of notes. A first thought would be to have a HMM for each note on the scale. However, because the music is polyphonic the recogniser would fail when two notes were played together. The HMMs for each of the constituent notes might respond, but because of the presence of partials from the other note (which would be seen as a form of ‘noise’ in the input), the HMMs may actually respond with quite low probabilities. It would be equally easy to have HMMs trained to recognise chords, but this would only be viable for a restricted set of chords, such as triads, since there are many thousand possible distinct chords. Currently there is no obvious solution to this problem.

Secondly, the representation of musical time is still somewhat impoverished. It is commonly accepted that Markov Models in their standard form are rather poor at representing the possible temporal structures of speech. For example, the probability of staying in the current state for n steps is the probability on the arc from the current state to itself to the power n. In speech there are several instances where this is not a good model of the temporal structure, and although the structure of music is perhaps a little more like this (for example, the probability that note will continue decreases as the note lasts for longer and longer periods of time), it would still be desirable to have a greater degree of control over the temporal aspects. One possible augmentation of the standard HMM model is suggested in [13].

6 Conclusions and Further Work

Progress has been made towards the low-level representation of incoming musical signals. The MFT provides an analysis tool with some opportunity to circumvent the problems caused by the Uncertainty Principle which affect work which assumes a fixed temporal/frequency window size. In combination with the techniques developed in [16] and the work described herein on beat estimation, it should be possible to extract sets of partials which will mainly correspond to the harmonics of notes. Work must now focus on the next higher levels of representation, and decisions must be made about an architecture which will allow knowledge of both high level and low-level musical structure to interact and provide adequate system performance. It would be unwise to preclude building useful deep knowledge of musical structures to the architecture of any neural network, since the learning algorithms available are limited in their power, and since there is not an arbitrarily large amount of training data available, nor the time in which to perform the training. Furthermore, there are many apparently simple functions which existing learning algorithms cannot learn if the problem is badly represented (for example the ‘x divides y’ decision problem [23]). Hence training should perhaps be best viewed as a tuning process. On the other hand, it would be equally unwise to make too many unrealistic assumptions about the structure of the music to be transcribed (as in [3] which relies upon the harmonic structure of piano notes), since this will lead to a system of too restricted a scope, and will avoid confronting many of the most important issues in music transcription.

References

1. M. BALABAN, "Music Structures: Interleaving the Temporal and Hierarchical Aspects in Music" in *Understanding Music with AI.*
2. G. J. BALZANO, "The Group-theoretic Description of 12-Fold and Microtonal Pitch Systems," *Computer Music Journal,* Vol. 4, No. 4, p. 66–84, Massachusetts Institute of Technology, Cambridge, Mass. (Winter 1980).
3. C. CHAFE, D. JAFFE, K. KASHIMA, B. MONT-REYNAUD, AND J. SMITH, "Techniques for Note Identification in Polyphonic Music," *Proceedings of the 1985 International Computer Music Conference* (1985).
4. D. COPE, "On Algorithmic Representation of Musical Style" in *Understanding Music with AI.*
5. R. B. DANNENBERG, "Music Representation Issues: A Position Paper," *Proceedings of the 1989 International Computer Music Conference,* Ohio State University, Columbus, Ohio (November 1989).
6. I. HOLST, *An ABC of Music,* Oxford University Press, Oxford (1963).
7. ISO/IEC JTC 1/SC 29/WG 11, *Final Text for ISO/IEC 11172-3, Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s—Part 3: Audio,* ISO (May 1993).
8. H. KATAYOSE AND S. INOKUCHI, "The Kansei Music System," *Computer Music Journal,* Vol. 13, No. 4, Massachusetts Institute of Technology, Cambridge, Mass. (Winter 1989).
9. W. B. KUHN, "A Real-Time Pitch Recognition Algorithm for Musical Applications," *Computer Music Journal,* Vol. 14, No. 3, p. 60–71, Massachusetts Institute of Technology, Cambridge, Mass. (Fall 1990).
10. B. LADEN AND D. H. KEEFE, "The Representation of Pitch in a Neural Net Model of Chord Classification" in *Music and Connectionism.*
11. F. LERDAHL AND R. JACKENDOFF, *A Generative Theory of Tonal Music,* The MIT Press, Cambridge, Mass. (1983).
12. S. E. LEVINSON, "A Unified Theory of Composite Pattern Analysis for Automatic Speech Recognition," *?.*
13. S. E. LEVINSON, "Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition," *Computer Speech and Language,* No. 1, p. 29–45 (1986).
14. R. F. MOORE, *Elements of Computer Music,* Prentice Hall, Englewood Cliffs, New Jersey (1990).
15. L. T. NILES AND H. F. SILVERMAN, "Combining Hidden Markov Model and Neural Network Classifiers," *Proceedings of the 1990 International Conference on Acoustics, Speech and Signal Processing,* Volume 1: Speech Processing 1, IEEE Signal Processing Society (April 1990).
16. E. R. S. PEARSON, "The Multiresolution Fourier Transform and its Application to Polyphonic Audio Analysis," *PhD. Thesis,* University of Warwick, Coventry, England (September 1991).
17. D. E. RUMELHART AND J. L. MCCLELLAND, *Parallel Distributed Processing—Explorations in the Microstructure of Cognition,* MIT Press, Cambridge Mass. (1986).
18. H. SANO AND B. K. JENKINS, "A Neural Network Model for Pitch Perception" in *Music and Connectionism.*

19. D. L. SCARBOROUGH, B. O. MILLER, AND J. A. JONES, "Connectionist Models for Tonal Analysis" in *Music and Connectionism.*

20. F. SCHUBERT, *Piano Quintet in A major, D 667 ("The Trout"),* Ernst Eulenburg Ltd, London.

21. H. SCOTT AND R. G. WILSON, "A Comparison of Filters for Audio Signal Segmentation in Audio Restoration," *Department of Computer Science Research Report,* RR231, University of Wariwck, Coventry, England (October 1992).

22. R. N. SHEPARD, "Geometrical Approximations to the Structure of Musical Pitch," *Psychological Review,* Vol. 89, No. 4, American Psychological Association (July 1982).

23. C. THORNTON, "Why Connectionist Learning Algorithms Need To Be More Creative," *University of Sussex Cognitive Science Research Report,* CSRP 218, University of Sussex, Brighton, England (1992).

24. P. M. TODD, "A Connectionist Approach to Algorithmic Composition" in *Music and Connectionism.*

25. R. G. WILSON, A. D. CALWAY, E. R. S. PEARSON, AND A. R. DAVIES, "An Introduction to the Multiresolution Fourier Transform and its Applications," *Department of Computer Science Research Report,* RR204, University of Warwick, Coventry, England (January 1992).

26. R. G. WILSON AND M. SPANN, "Finite Prolate Spheroidal Sequences and Their Applications II: Image Feature Description and Segmentation," *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence,* Vol. 10, No. 2 (March 1988).