Multiresolution Techniques for Audio Signal Restoration

Hugh R R Scott BSc

A thesis submitted to
The University of Warwick
for the degree of
Doctor of Philosophy

March 1995

Multiresolution Techniques for Audio Signal Restoration

Hugh R R Scott BSc

A thesis submitted to
The University of Warwick
for the degree of
Doctor of Philosophy

March 1995

Summary

This thesis describes a study of techniques for the restoration of musical audio signals using a multiresolution signal representation called the multiresolution Fourier transform (MFT), a time-frequency-scale representation. This representation allows the restoration to adapt to the local signal structure, which typically consists of a set of approximately sinusoidal partials, each consisting of an "onset" of rapid energy variation followed by more slowly varying "sustain" and "decay" phases.

It must be decided what components of a noisy audio signal are to be kept in the restored version and, conversely, which must be removed. A simple filter is introduced that retains only musical signal — that is signal which adheres to the musical model — and rejects everything else. It is shown that this filter used in conjunction with the MFT has a low computational complexity. The MFT is used to capture the transient energy present at the onset of notes by splitting the time axis of a musical signal into steady-state and transient zones using a simple onset detector, which measures the expected energy at a given time against the actual energy present.

Past audio signal restoration systems have relied on estimating a restored audio signal's spectrum from the noisy audio signal presented to the algorithm. In this thesis the idea of having more than one version of a recording is used in order to gain further information about the ideal spectrum of the noisy signal. This poses a number of problems with regards to matching the time scales of two versions of the same piece. These are addressed and solutions are offered, based on a novel multiresolution warping algorithm.

Finally, various methods for using the detected signal spectrum of a clean modern signal to restore a noisy signal using the warping techniques and musical event detection filters are shown. These account for variations in scale and input signal to noise ratio (SNR) in the noisy signal. It is also shown how the simple adaptive filter introduced earlier can be used to restore audio signals with impulse noise as well as white additive noise. This filter and the time-warping technique is compared to adaptive Wiener filtering as an audio restoration method.

Key Words:
Multiresolution, Musical Signal Restoration, Scale, Adaptive Filtering, Warping

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank the Engineering and Physical Sciences Research Council and Thorn-EMI Central Research Laboratories for their generous funding of this project.

Thanks also to the members of the image and audio signal processing group for their stimulating conversation and bright ideas which have helped to bring about the environment in which the greater part of this work has taken place. Especially, Andy King, Nicola Cross, Andrew Davies, Tao-I Hsu, Ian Levy, Peter Meulemans, Tim Shuttleworth and Horn-Chang Yang - "the lab"; and also to Jeff Smith for his invaluable technical support.

Special thanks must go to my supervisor, Roland Wilson, whose knowledge and initiative have helped motivate and guide me over the last three and a half years and also to my industrial supervisor Martin Todd for his support and interest.

Finally, I would like to express my gratitude to Vivian, my parents Hugh and Elizabeth and my brothers and sisters: George, Angela, Elizabeth, Christopher, Mhairi, Clare and Joseph; without whom none of this would have been possible.

Declaration

I declare that the material contained in this thesis is my own work and that it has neither been previously published nor submitted elsewhere for the purpose of obtaining an academic degree.

Hugh R R Scott

Chapter 1

Introduction

1.1 Noise in Music

Everybody has heard a noisy musical signal at some time, for example when tuning their radios, listening to an old, scratched vinyl record or listening to pre-recorded music on very poor equipment. The Oxford dictionary defines noise as "an unpleasant sound" [2]. In practice noise is any sound which detracts from the ideal perception of an audio signal. There are many types of noise, in musical signal restoration there are three main classes of noise, these can be classified as broadband background noise, impulse noise and harmonic distortion [10]. The first two of these are additive, that is they are *independent* of the musical signal, whereas the third is a function of the signal.

Broadband background noise is random in both time and frequency and, in audio restoration work, is generally considered to be of a Gaussian form [36] [58] [59] [22] [53]. Broadband noise occurs in recordings that have decayed as a result of their recording medium: withering of vinyl in gramophone recordings or demagnetization in tape recordings; the poor quality of a recording; the presence of uncorrelated noise in the recording process due to poor recording equipment, this noise is additive. The second type is impulse noise, which is normally present in degraded musical signals alongside broadband

background, or Gaussian, noise. Impulses are random in time only, their frequency response is a function of the noise source — degraded grooves in gramophone recordings that result in clicks when "read" by the pick-up or static present in a poor radio reception. Finally, the third noise type groups together both harmonic distortion and sound shaping. Harmonic distortion is caused by nonlinearity in the recording process, for example due to saturation. Spectral shaping may be caused by the acoustics of the recording room or inadequacies in the frequency response of the equipment and is a linear effect, but it is not additive — it is a form of filtering. Of these three classes of noise, the first two are by far the most commonly dealt with in the audio restoration literature and so methods for the removal of these types of noise are discussed next.

1.2 The Musical Signal Restoration Problem

1.2.1 Outline

The musical signal restoration problem can be formalised as follows. Let $x(t)$ denote a signal with noise present and $s(t)$ and $r(t)$ the signal and noise components respectively. Restoring $x(t)$ is equivalent to estimating $s(t)$ as accurately as possible: minimising the difference between the estimate $\hat{s}(t)$ of the clean signal and the original signal, in other words minimising $||\hat{s}(t) - s(t)||$, for some norm $||$. This is where a problem arises since neither $s(t)$ nor $r(t)$ is known *a priori*. The removal of harmonic distortion and spectral shaping lie outwith the scope of this thesis.

1.2.2 Available Audio Material

Restoration of musical signals has been made desirable in recent years by the release of old, and sometimes rare, recordings of artists on Compact Disc that have previously been available on gramophone records. Generally these gramophone recordings are quite

degraded. Most previous restoration systems assume that only one version of a recording is available. However, it is possible that more than one degraded recording may be available. Indeed this is assumed in a restoration method proposed by Vaseghi in [59], where the existence of a second equally degraded copy is assumed. In part of the work presented in this thesis it is assumed that alongside the degraded musical signal, there will be a clean, "modern" version of the same musical signal available. Even in the case when there is no clean recording of a musical signal, musicians could be paid to perform the piece, which could be recorded digitally, giving a clean version of any piece that is desired.

1.3 Previous Audio Restoration Methods and Signal Representations

Historically, there has been comparatively little activity in the field of musical signal restoration. Many of the methods used come directly from speech processing where, conversely, a lot of time and energy has been spent by a great many people over the last twenty or so years. Hence in the following overview of methods used to restore musical signals, some reference will be made to techniques employed to restore and enhance speech signals.

1.3.1 Time Representations

Viewing a musical signal as a function of time is simple: it is the way musical signals are stored, transmitted and received and is the most straightforward of all the representations. One of the least signal processing intensive methods used to restore a signal in the time domain is manual intervention [6], where the restorer views the time signal of the degraded music and decides visually what the best restored signal is. This choice is

verified subjectively by the restorer until the best effect is found. This removes both broadband noise and impulse noise partially. The problems with this are first that it is very subjective: the best restoration depends greatly on the ability of the restorer and secondly since the desired result is to be of high quality, often at, or near to, CD quality, the sampling frequency is high. For CD quality it is 44.1kHz and is generally greater than 20kHz. This means that even for a short segment of signal there is a large number of samples to be rectified manually — making manual intervention both a very skilled and labour intensive occupation.

Kalman Filtering

From a signal processing perspective, the ideal filter for a degraded time sequence such as the noisy musical signal $x(t)$ is a Kalman filter [10] [35]. A Kalman filter is a real time filter that, using a state-space signal model, estimates the best restored signal $\hat{x}(t)$, using $x(t)$ and $\hat{x}(t-1)$. It is a non-stationary filter which gradually adapts to the signal statistics as more and more samples are processed, so that for stationary signals it behaves asymptotically like the best non-causal filter — the Wiener filter. The problem in audio restoration is the highly non-stationary nature of the musical audio signal: the more or less stationary, steady-state portions of the notes occur after the transient onsets. These two phases generally require different forms of processing, which is not easily accomplished with a conventional Kalman filter. This requires a higher level model of the musical signal than is available in the sample by sample time representation available here. Kalman filtering removes impulse noise as well as additive Gaussian noise [10].

Adaptive Noise Cancelling

A time domain method used in speech enhancement for removing random noise is adaptive noise cancelling [30] [28]. Over short periods of time audio signals are periodic, whereas the noise is not. Once the periodicity of the signal is found, using autocorrelation methods for example, the difference between two signal values one period apart will equal the difference of the noise components at those times, since the clean signal values cancel out. In other words, for a degraded audio signal $s(t)$ with period T then

$$x(t) - x(t+T) = s(t) - s(t+T) + r(t) - r(t+T) \approx r(t) - r(t+T) \qquad (1.1)$$

The noise difference $r(t) - r(t+T)$ is uncorrelated with $s(t)$. This can be exploited in identifying what the noise component is at time t. For effective noise reduction, however, the spectrum of the corrupted signal needs to be known, so an estimate of the restored spectrum can be calculated. This leads to the conclusion that even though it is time information that locates the noise, it is not enough on its own to remove it. Variations in the periodicity causes implementational problems akin to those faced by Kalman filtering.

Removing Impulse Noise

The time domain lends itself to impulse noise restoration, by allowing the implementation of simple signal models. The simplest method is that used by Deutsch and Noll [17] for the removal of impulse noise in recordings held at the Phonogrammarchiv of the Austrian Academy of Science. The restorer locates the impulses visually and audibly, in a way similar to Brink, but instead of manually altering the signal, the restoration system replaces the N degraded samples with the previous N samples. They reported that there were no unpleasant side effects as long as N/S_f is less than 20ms, where S_f is the sampling frequency.

In the impulse removal system used by Vaseghi [59] [58] a simple Linear Predictor (LP) model of the signal is used. That is, for an impulse train $e(t)$, the signal at time t can be modelled as

$$s(t) = \sum_{k=1}^{p} a(k)x(t-k) + r(t) + e(t) \tag{1.2}$$

where the coefficients $a(k)$ are the LP parameters and p is the LP filter length. This can be used to create an "excitation signal" by subtracting the predicted signal from the LP output, whereupon the impulses become apparent and can be removed from the noisy signal. When these samples are so distorted that the signal cannot be corrected, the samples are just replaced using the LP model output generated from the surrounding samples. This is very similar to the method used by Cisowski [11]. The only difference between the two methods is in the approach to locating impulses in the excitation signal. Vaseghi uses a matched filter — matched to an impulse response — or just peak detection, but Cisowski uses an "outlier threshold" method that depends on the distribution of the excitation signal being approximately Gaussian. Cisowski's more sophisticated method is designed to reduce the probability of a false alarm due to a rapid musical onset or the like to less than 0.001%. A more recent method for detecting impulse noise in the time domain uses higher order statistical modelling [22]. It is assumed that the musical signals can be regarded as approximately Gaussian in distribution over a lengthy time period. This is exploited by using a third order cumulant C_3, defined as

$$C_3 = \frac{1}{N} \sum_{t=0}^{N-1} x(t)x(t+1)x(t+2) \tag{1.3}$$

which is zero for second order sample distributions. It is assumed that the impulses are not Gaussian in behaviour and so they are found by locating the points where the cumulant is significantly different from zero. These samples are then replaced using the bispectrum, or Fourier transform of the cumulant and inferring phase and magnitude of each sample

value for the affected samples.

1.3.2 Frequency Representations

In the previous section it was stated that musical signals are functions of time. Whilst this is correct, it is not helpful. Helmholtz discussed audio signals, or waves, as a tonal or frequency phenomenon [24]. Indeed it is possible to distinguish two or more notes being played simultaneously: evidence for audio signals being made up of frequency components. The spectrum of an audio signal can be generated simply by taking its Fourier transform (discussed in further detail in chapter 2), allowing simple inspection of the component frequencies in a given musical signal.

Wiener Filtering

The ideal signal processing solution to the restoration of a noisy signal in the frequency domain is the Wiener filter [43] which is based on a stationary signal model. As noted above, it can be shown that the Wiener filter is the stationary limit of a Kalman filter [10]. The problem with implementing a Wiener filter is that audio signals are generally non-stationary: their statistics alter as a function of time. The basic assumption for the use of Wiener filtering is therefore false. However, Wiener filters can be used over the stationary periods of an audio signal to good effect. This motivates the next and most commonly used representation of audio signals: time-frequency representations that are functions of both time and frequency.

1.3.3 Time-Frequency Representations

Many methods used in audio restoration use a time-frequency representation. The simplest of these representations is just the short-time spectrum [46] [9], a representation that splits

the time signal up into short regularly spaced periods and looks at the spectrum for each. Using such a representation, an estimate of the noise in each frequency band in the short-time spectrum can be made. This is done by assuming that the noise signal and noise-free signal are not correlated. The noisy signal windowed using some finite window function, $w(t)$, for example a cosine window, is represented by

$$x(t)w(t) = x_w(t) = s_w(t) + n_w(t) \tag{1.4}$$

Spectral Subtraction

The spectral subtraction method proposed by Boll [8] for speech enhancement maintains that in any degraded speech signal there is a period of non-speech activity in which the magnitude of the noise spectrum can be estimated. This estimation is computed by averaging over a number of these non-speech intervals. Boll made the phase of the estimated noise spectrum $\arg(\hat{N}_w(\omega))$ equal to that of the noisy spectrum, $\arg(S_w(\omega))$. This is then subtracted from the noisy signal spectrum, thus restoring the signal. Various improvements have been made to this method by considering the power spectra of the noisy signal [28]. More recently a method proposed by Vaseghi [58] for musical signal restoration exploits the fact that the signal and noise are uncorrelated. This means that the corresponding power spectra satisfy

$$|X_w(\omega)|^2 = |S_w(\omega)|^2 + |N_w(\omega)|^2 \tag{1.5}$$

the cross terms being zero. The amount of noise present may be overestimated, since it depends on the estimated noise in the periods of low signal activity, allowing the possibility of subtracting a high noise estimate from a signal area with a low amount of noise, creating a negative spectrum. Some solutions have been suggested to cure this, but it is worth commenting that none are entirely satisfactory. The two simplest of these

methods are: first setting all negative spectral values to zero, and secondly, changing the sign of all negative spectral values, rendering them positive. More complex methods are described in [61] [62] [31].

Deutsch and Noll [17] make similar assumptions in their background noise reduction system. By averaging three or more samples, using a window of a length equivalent to the "stationary length" of the signal, a signal estimate is gained. This is then used to calculate the impulse response of a noise suppression filter, which is multiplied with the noisy signal in the frequency domain. The problems faced by this method are twofold. First the length of signal chosen to be stationary is fixed, whereas it is well known that various parts of audio signals display more stationarity than others, implying that the window length should be allowed to vary. Secondly, as the number of windows used in the averaging process increases, the accuracy of the noise suppression filter decreases.

Comb Filters

The harmonic nature of audio signals is exploited by the adaptive comb filtering method proposed by Shields and improved by Frazier et al [28] [29]. Shields proposed constructing a set of narrowband filters spaced at equal intervals determined by the fundamental frequency of the audio signal being restored. Thus, signal would be transmitted to the restored version and all noise would be suppressed. The timbre of an audio signal alters with time, as more notes are played or, as in speech enhancement, as another word is spoken. This is compounded by the introduction of the vibrato problem, in which the timbre of the signal will vary locally. Hence Frazier proposed a more general filter that did not assume equidistance between the narrowband filters, allowing their distance to vary depending on the timbre of the signal. Comb filters have also been used in polyphonic note transcription by Moorer [38]. To accomplish this he simplified the musical transcription

problem by ruling out the possibility of such musical phenomena as partials that overlap in frequency, vibrato and glissando. The comb filter's role is to ascertain the periodicity of the signal at a given time.

Adaptive Wiener Filtering

There have been several restoration schemes that are based on Wiener filtering. A Wiener filter is a stationary filter and, since one can regard an audio signal as "locally stationary", a local Wiener filter can be applied to those signal components. Such a filter is known as an adaptive Wiener filter. Vaseghi used an adaptive Wiener filter to restore audio signals [59] [58] by assuming that there would be two degraded copies of a given performance available. An assumption was made that the noise in these two recordings would be uncorrelated — valid if one assumes that the recordings have been degraded *independently*. The signals could not just be added as their time scales would not coincide precisely: no two audio signals will have exactly the same time scale due to limitations of the performers or, in this case, the playback mechanism. It was decided that the audio signal's stationarity period was around 20ms. Hence an adaptive Wiener filter with this window length was used. To cope with the variation in time between the two recordings, it had a delay that depended on the cross-correlation of the two signals. The main drawbacks with this method are first that it is implemented in the time-domain, making the choice of adaptive delay algorithm complicated because the delay between the two recordings is a function of time. Secondly, there may not always be two degraded recordings of the same piece available. Thirdly, the filter length is fixed; this means that for the best results the stationarity of the signal should remain constant at around the filter length. It is well known that the duration of the steady-state periods of a musical signal varies with time, and is in any case punctuated by onsets. Moreover, the time scale of the piece and the speed of the performance vary

hugely, both implying a variability in the "stationary period".

Montresor et al [36] used a minimum mean square error (MMSE) method proposed by Ephraim and Mallah [18] for speech restoration. Using an MMSE approach the local spectral amplitude is calculated. The *a priori* SNR estimate is calculated using the spectral power density of a silent period in which there is no audio activity. This gives a noise estimate. Once the noisy signal is known, it is possible to improve the restored signal estimate, using a first order recursive filter on the noise and noisy signal. The local spectrum of the signal is altered so that its phase remains the same but so that it now has the new clean signal amplitude estimate. This was further improved by Simon et al [53]by using an "Octave Filterbank Implementation". The frequency domain is split into J sub-bands each of which has its own filter for estimating the *a priori* SNR. Simon et al. proposed different scales for each frequency band, choosing different time window lengths dependent on the sub-band: a small window of about 20ms for high frequencies and a long window length for low frequencies. This gives improved performance, permitting the restoration system to deal with high and low frequency noise simultaneously. This demonstrates one way in which scale can be used in time-frequency representations for audio signal restoration.

The Window of Scale

Pearson [45] worked on a similar premise in order to perform note transcription. Problems can arise when choosing a time window length for an audio signal representation. If the window is small the frequencies of notes' partials are difficult to locate. When a long window is used its position in time is imprecisely defined. This is a consequence of the Uncertainty Principle [33]. To solve this problem Pearson used a generalised wavelet transform, that has an explicit scale parameter, the Multiresolution Fourier Transform

(MFT). This allows the transcription algorithm to pick the appropriate scale at a given frequency or time, so that the position of a partial in time and frequency can be located independently of the uncertainty product associated with a fixed window and hence with high precision: such a scheme appears to offer the potential to overcome some of the limitations of previous methods.

1.4 Objectives

The work presented in this thesis intends to show how the multiresolution Fourier transform, with its free scale parameter, can be used to perform audio restoration.

1.5 Thesis Overview

The following chapter discusses some different time-frequency representations, with a view to introducing the MFT, an effective tool with which to perform audio restoration. In chapter 3, a simple adaptive filter, derived from the MFT of a musical signal, is introduced. It is shown that the filter has low complexity and can be easily applied to the restoration of noisy signals when the signal to noise ratio (SNR) is known. The benefits of combining restored signals across scale are discussed and a simple method of doing so is implemented. In chapter 4, a new method for aligning the time scales of two versions of a musical signal is introduced with a view to using the filter derived from a clean, modern recording, time-warping it and applying it to a noisy signal. In chapter 5 this is compared with other restoration methods including adaptive Wiener filtering. Finally, various methods for combining results from different scales and filters derived from the warped clean signal and the noisy signal are presented and tested. It is also shown that the adaptive filter can cope with impulsive noise as well as white additive Gaussian noise. The

thesis is concluded with a summary of achievements and suggestions for further work.

1.6 Audio Samples

1.6.1 Notes on Hardware Used

The audio samples were recorded using a Technics XL Compact Disc Player, connected to a SPARC LX workstation with a SPARC 10 DBRI audio chip for A/D conversion.

1.6.2 Measurement of Signal Accuracy

To ascertain the accuracy of the restoration algorithms presented in this thesis, a controlled amount of noise has been added to clean signals so that, on being restored, the error between the restored signal and the original clean signal can be measured precisely. This is done by calculating the signal to noise ratio (SNR) before and after restoration. The SNR is defined as the ratio of the signal variance to the noise variance of a degraded signal. The three SNR values used throughout this work are 10dB, 15dB and 20dB. Upon listening, it appears that a signal with a SNR of 15dB gives a good approximation to a heavily degraded audio signal. Signals with SNR's of 10dB and 20dB are included to demonstrate that the algorithms used in this work can function at other noise levels. Another term that is used in conjunction with SNR is *SNR gain*. This means the change in SNR of a signal upon enhancement. For example, a signal with a SNR of 10dB that is enhanced with a gain of 5dB will have an overall SNR after enhancement of 15dB.

1.6.3 Signal Samples

Four different musical signal samples have been chosen. Two extracts from Beethoven's String Quartet in G major, Adagio and Scherzo movements and Adagio and Allegro movements, from "Waldstein", a piano sonata, also by Beethoven . The Adagio string

piece is chosen as it displays extreme vibrato. This is desirable to show how the restoration system in general, and more specifically, how the warping algorithm copes with heavy vibrato. The Scherzo is used as it is very quick. The "Waldstein" was chosen because the full range of piano tempos are tested. In the Adagio section, the pianist plays very slowly, and in the Allegro movement, the pace is extremely fast. This provides a good test of the robustness of the algorithms.

For testing the warping algorithm, two versions of each piece are chosen. For the string quartets, performances by the Smithsonian string quartet and the Lindsay string quartet are used and for the piano sonata, performances by Ashkenazy and Jandö are used.

The Adagio and Allegro "Waldstien" and the Scherzo string quartet samples are approximately 12 seconds long and the Adagio string quartet sample is approximately 23 seconds long. The sampling rate used throughout this work is 44.1kHz.

Chapter 2

The Multiresolution Fourier Transform

2.1 The Requirements of a Good Representation in Audio Signal Restoration

Regarding an audio signal purely as a time function, it is difficult to make sense of what its constituent parts or features are. This problem is compounded in audio restoration since there is noise present in the time function that degrades the signal features. It is therefore helpful to have a higher level representation of such an audio signal, which facilitates the separation of musical features which would in turn enable a decision to be made as to what is noise and what is not.

For audio signal analysis it is desirable: first, that any representation is a function of time, second, that it is a function of frequency and third that this representation does not have a fixed scale [45]. The reason for wanting the higher level representation to be a function of time is obvious: audio signal features are partly separable as functions of time and so any meaningful analysis must also be a function of time. It should be a function of frequency since the features are also partly separable as a function of frequency, but why must there be a variability in the scale at which we look at these features? The reason for this is due to the Uncertainty Principle [33], which limits the simultaneous resolution of

time and frequency signal features. Thus if the scale were fixed, any analysis of an audio signal would be restricted to this fixed resolution. The problem with this is that audio signals do not have a fixed resolution. To show this, consider two notes being played on a piano. The limit on how closely in time or frequency these can be played is set by the demands of the composer, the virtuosity of the performer and the harmonic content of the notes. For example, it is possible to play two short notes less than 2Hz apart at the lower end of the piano keyboard. In order to be able to restore this audio signal well, by removing the maximum amount of noise, it would be necessary to have a high resolution in both time and frequency to detect the musical features, but this is prohibited by the uncertainty principle [45], [65].

Finally, any higher order representation of an audio signal that allows analysis of that audio signal must be accompanied by a method of inversion. This is because in audio restoration it is necessary to end up with an improved time function audio signal. A restored audio signal is of no use if it exists in an unreconstructable representation. This is a further consideration, along with the three requirements for audio restoration stated above, in choosing the representation used in this work. There are a number of audio signal representations available for analysis, whose main properties are discussed in this chapter.

2.2 Representing an Audio Signal

2.2.1 The Fourier Transform

The motivation for a Fourier representation of an audio signal is simple: a pure tone can be modelled exactly by a sine wave of the same frequency. If pure tones can be considered to be the building blocks of musical signals, then so too can sine waves, implying that any audio signal can be represented as some linear combination of sine waves. The integral

Fourier transform is defined as [43]

$$S(w) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t}dt \tag{2.1}$$

where ω is the frequency of the complex exponential. Effectively $S(\omega)$ tells us how much contribution there is for each frequency ω and, for a purely harmonic signal $s(t)$, $S(\omega)$ would consist of lines at all multiples $\eta\omega_0$ of the fundamental frequency ω_0. To return to our original analogy, this is the same as weighting each pure tone of frequency ω and then summing in frequency, creating the original time signal $s(t)$.

Whilst this gives a sufficient description of the frequency structure of a time signal $s(t)$, there is no evidence of any time structure, since the integral in (2.1) is calculated over all time. This shortcoming of the Fourier representation was noticed by Gabor [20], who formulated a representation which is a function of both frequency and time. In effect the time scale of $S(\omega)$ is infinite. Note that in this work scale is used as a term to describe a particular resolution — if it is large scale then it is coarse and, if it is small scale, it is fine.

It is well known that the Fourier transform is invertible, for completeness the inversion formula is included for signal spectrum $S(\omega)$

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)e^{j\omega t}d\omega \tag{2.2}$$

using this the time signal $s(t)$ can be obtained.

2.2.2 The Gabor Transform

Gabor realised that the Fourier representation of audio signals was less than satisfactory: it did not meet with our intuitive interpretation of time signals which was that frequency content changes as a function of time. The example that he highlighted in [20] was that of a siren moving away from a listener. The listener does not hear a constant frequency,

but one which changed as a function of time. If this were analysed using a Fourier transform, the only frequency information obtained would be spread across the range of "instantaneous frequencies". It was to the area of quantum mechanics that Gabor looked for a solution to this problem, realising that any representation of an audio signal that was both a function of time and frequency would have its accuracy in the frequency domain compromised by its accuracy in the time domain, and vice versa. This inequality is known as the uncertainty principle, and was first realised by Heisenberg in the field of wave mechanics in the 1920's [33]:

$$\Delta t \Delta f \geq \frac{1}{2} \tag{2.3}$$

where Δt is the uncertainty in time and Δf the corresponding uncertainty in frequency. Gabor's major contribution was to evaluate what he called "elementary signals" that minimised this inequality. It is well known in the field of quantum mechanics that (2.3) is minimised for a particle if the Hamiltonian describing such a particle's motion is considered to be that of a classical simple harmonic oscillator [33]. Using the Schrödinger equation it is possible to find a solution for these wave functions, known as Hermite functions. It was in this way that Gabor approached the problem of finding the most spatially and temporally compact wave functions. These wave functions were considered by Gabor to be the quanta of time-frequency representation and were accordingly named "logons". They take the form of time and frequency shifted Gaussian functions and, although this set of functions does not form an orthogonal basis, as the sine and cosine functions do, it does give the maximum time and frequency concentration, as limited by (2.3). The notation used in [45] is

$$g_{kl}(t) = g(t - kT)\exp(j(2\pi lFt + \phi)) \tag{2.4}$$

for the "logon" at time index k and frequency index l on the time frequency plane, with frequency sampling interval of $2\pi F$, and time sampling interval of T. The function $g(t)$ is a Gaussian, of the form

$$g(t) = \exp(-\alpha^2 t^2) \tag{2.5}$$

the parameter α is the scale parameter, which determines how much time definition and, correspondingly, how much frequency definition is to be allowed. These are related via

$$\Delta t = \sqrt{\frac{\pi}{2}}\frac{1}{\alpha} \quad \Delta f = \frac{\alpha}{\sqrt{2\pi}} \tag{2.6}$$

Gabor states that, although these functions are not orthogonal, it is possible to approximate an expansion of a general time signal in terms of them with weighting coefficients C_{kl}. Thus, time signal $s(t)$ can be represented as

$$s(t) = \sum_{k,l=-\infty}^{\infty} C_{kl}\, g(t - kT) \exp(j(2\pi lFt + \phi)) \tag{2.7}$$

For a representation of an audio signal, Gabor envisaged a time-frequency plane that was split into "logons", each multiplied by the coefficients C_{kl}, as in figure 2.1, with the weighting corresponding to the amount of contribution there was from each "logon" at that time and frequency. The main problem with the Gabor representation is the difficulty in calculating the coefficients C_{kl}, which is often done using recursive iteration, which is expensive in computational terms when compared with the Fast Fourier Transform, [43], used to calculate the Fourier Transform of a signal. This difficulty in calculating the Gabor coefficients is related to the stability of the inverse, as explained by Daubechies [14]. Moreover, this choice of sampling intervals in time and frequency, which is known as critical sampling, leads to a highly unstable inverse, unsuitable for audio restoration work.

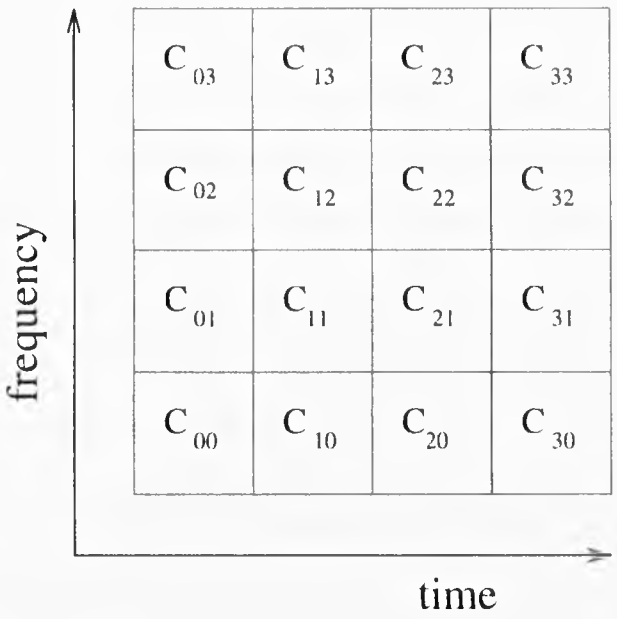

Figure 2.1: Time-frequency plane with Gabor's "logon" weightings.

2.2.3 The Short Time Fourier Transform

A simpler way of looking at the problem of creating a frequency representation of a signal that is also a function of time, is to split the time signal up into time segments, and look at the Fourier transform of each of these segments. This representation is a short time Fourier representation called the Short Time Fourier Transform (STFT). So for time signal $s(t)$, the STFT, is denoted by $S(\omega, t)$, where ω represents the spectrum of the STFT at time t [46].

In order to localise the STFT, it is necessary to modulate the signal with a window function, $g(t)$, leading to the definition of the STFT as a Fourier transform of the windowed signal, where the windowed position is t:

$$S(\omega, t) = \int_{-\infty}^{\infty} s(\tau) g(\tau - t) e^{-j\omega\tau} d\tau \tag{2.8}$$

As the position of the window $g(t)$ varies, then so does the "local" spectrum $S(\omega, t)$. If the window is too long in time, then it will be ineffective at "tracking" non-stationary signals

such as the frequency modulated siren of Gabor, while if it is too broad in frequency it will not be possible to resolve nearby spectral components. The uncertainty principle (2.3) is again the limiting factor. To illustrate this, consider a simple case when the window function is just a cosine, defined in the region $[-\tau, \tau]$, so that it is of length 2τ and has centre at the origin.

$$g(t) = \begin{cases} \cos(\frac{\pi t}{\tau}) & \text{if } t \in [-\tau, \tau] \\ 0 & \text{else} \end{cases} \tag{2.9}$$

The frequency response of this function is $G(\omega)$

$$G(\omega) = \int_{-\infty}^{\infty} \cos(\pi t/\tau) e^{-j\omega t} dt \tag{2.10}$$

which reduces to

$$G(\omega) = \int_{-\tau}^{\tau} \cos(\pi t/\tau) e^{-j\omega t} dt \tag{2.11}$$

it can be shown that this is

$$G(\omega) = 2\cos(\tau\omega)\frac{\pi\tau}{\tau^2\omega^2 - \pi^2} \tag{2.12}$$

The window and signal are convolved in the time domain, which is equivalent to the product $S(\omega, t)G(\omega)$ in the frequency domain [43]. There is a τ term in the numerator of (2.12), and a τ^2 in the denominator. Therefore as the energy concentration increases in one domain, it decreases in the other. The trick is to find the best trade-off for one's needs. If finding the precise location of a feature in time is more important than ascertaining a precise picture of its spectrum, then a small time window should be used. Conversely, if a precise spectrum is required, at the expense of the localisation in time of any feature detected, a large time window should be used. Clearly the choice of the window scale is crucial. It will affect the outcome of any analysis algorithms greatly. Note that "window scale" here means the size of the window — a large window scale means coarse resolution and a small window scale means fine resolution.

Like the Fourier transform, the STFT can be reconstructed exactly, as long as the window function used in the inverse transform $h(t)$ is related to the forward transform window $g(t)$ via [46]

$$\int_{-\infty}^{\infty} h(t)g(-t) = 1 \tag{2.13}$$

The forward transform window $g(t)$ is often called the analysis window and the inversion window $h(t)$ a synthesis window. The inverse STFT is defined in the corresponding manner to (2.8)

$$s(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t-\tau)S(\omega,t)e^{j\omega t}d\tau dt \tag{2.14}$$

2.2.4 The Wavelet Transform

One method, advanced in recent times, which tries to solve the scale problem that the STFT suffers from, is the wavelet transform (WT). This was motivated by the work done by Gabor splitting the time-frequency plane into discrete "logons". The WT generalises this approach by allowing the analysis function, or wavelet, to be shifted and dilated, permitting an alteration in the scale of the frequency representation. This permits some frequencies to be seen at one scale and some at another [25] [14] [15] [48] [36] [53].

A wavelet is a function $g(t)$, with Fourier transform denoted by $G(\omega)$; the conditions that it must satisfy to be a wavelet are [14]

$$0 < \int \left|\frac{G^2(\omega)}{\omega}\right| d\omega < \infty \qquad \int g(t)dt = 0 \tag{2.15}$$

The WT of a function $s(t)$ is then

$$S(b,a) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} g\left(\frac{t-b}{a}\right) s(t)dt \tag{2.16}$$

where the co-ordinates (b,a) give a position on a "shift-dilation" plane [25] . The variable b can be interpreted as being a time or position co-ordinate, whereas a is a frequency/scale

co-ordinate, where the amount of scale at a given frequency is determined by the choice of function $g(t)$. For example, it is shown in [25] that for a Gaussian function as a choice of wavelet, the frequency resolution increases as a function of frequency, with a proportional decrease in time resolution. It is shown in [14] that this is true for any choice of wavelet function.

Although the WT has an advantage over the STFT, in that scale is not the same for all frequencies, once a scale has been chosen, any analysis algorithm is restricted to those scales for those frequencies. This was shown to be a problem in the work done by Pearson [45] on note transcription, where the scale needed for a given frequency range was a function of the piece being transcribed, and so could not be determined beforehand. The main problem with the WT, as far as this work is concerned, is that scale and frequency are interdependent, but unless the scale is varied *independently* of frequency, it cannot be chosen to avoid interference of close partials from different notes [45]. This problem is addressed in the next section.

Since the wavelet basis functions are typically over-complete [14], many inverse transforms exist [25]. One widely used in audio signal processing for continuous signals is

$$s(t) = \frac{1}{c_g} \int \int \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) S(b,a) \frac{1}{a^2} da db \tag{2.17}$$

which reconstructs exactly. The constant c_g depends on the wavelet chosen.

2.2.5 The Multiresolution Fourier Transform

A transform based on the Fourier transform, which is a superset of both the Short Time Fourier transform and the WT, is the Multiresolution Fourier Transform (MFT) [9], [66]. It has definition

$$S_{t,\omega,\sigma} = \sigma^{\frac{1}{2}} \int_{-\infty}^{\infty} s(\tau) g(\sigma(\tau - t)) e^{-j\omega\tau} d\tau \tag{2.18}$$

It is easy to see that this is an STFT with an explicit scale parameter in the transform domain, known as the MFT domain, as it has the same structure as an STFT with the window function scaled by a factor σ. It is also possible to view (2.18) as a generalised WT. It is a condition of the definition of the MFT that the window function $g(t)$ obeys certain conditions. The first is that the window is non-zero at the origin

$$g(0) > 0 \tag{2.19}$$

and secondly, that

$$0 < \int_{-\infty}^{\infty} g(t)dt < \infty \tag{2.20}$$

which is the condition required to make the MFT a frame (see Daubechies for the definition of frame [14]). It can be shown that the MFT behaves like a WT. If one considers the definition of the WT (2.16) and the definition of the MFT (2.18), then it can be shown that the wavelets, denoted for clarity here by $\gamma(t)$, are of the form

$$\gamma(\tau, \omega) = g(\sigma(\tau - t))e^{-j\omega\tau} \tag{2.21}$$

It can be shown that these functions when dilated, shifted, or translated, give another function of the same type. In other words, if $\mathcal{G}(\tau, t, \omega)$ denotes the set of all functions of the form $\gamma(\tau, \omega)$ then any shift, translation or dilation to a function in $\mathcal{G}(\tau, t, \omega)$ results in another function that is in $\mathcal{G}(\tau, t, \omega)$. That is to say that the set $\mathcal{G}(\tau, t, \omega)$ is closed under the operations of shifting, translation and dilation. This is shown in [66].

Thus, unlike any of the previous transforms, the MFT has complete freedom in the three axes of time, frequency and scale. The algorithms employed by the user can choose precisely which scale is desired and not be limited by a transform function which has fixed scale, or a choice of window function that has been chosen for a specific scale, as is true of the previous transforms. Looking at (2.18), it can be seen that there is a great deal of

similarity with (2.8). If one set the scale parameter σ equal to some constant, then the MFT reduces to an STFT. It can be concluded therefore that the MFT is an STFT that is a function of scale also. This will be discussed again in the discrete domain.

The MFT has been used for digital audio signal processing before [45], where it was employed in note transcription algorithms with some success. The fact that there was a free choice of scale, and one other parameter was exploited in this work, so that a musical notes constituent parts — partials — could be located as accurately as desired in frequency. These were then tracked down through scale space, until their onset in time was of a desired accuracy. Hence by using the MFT, the problems inherent to the uncertainty principle can be avoided with reasonable effect.

Until now, only the continuous case has been considered. Since we are concerned with digital signals, it is necessary to consider the case for the discrete Multiresolution Fourier Transform.

As stated above the MFT is equivalent to STFT's with a continuous scale parameter. As stated in section 2.2.3 each STFT has a complete inverse. It can be concluded therefore that, in the continuous domain, the MFT is constructed from an infinite number of STFT's, each having an exact inverse. In other words there are an infinite number of inverses. The MFT can be described as being over-complete, since there exists more than one way to invert from the transform domain exactly. It is sufficient however to invert for each scale σ according to the inverse STFT given in (2.14).

2.3 The Discrete Multiresolution Fourier Transform

2.3.1 Definition

The discrete MFT coefficient of a signal $x(t)$ at time t, frequency f and scale 2^n is given by

$$X_{t,f,n} = \Xi^{\frac{1}{2}}(n) \sum_{k=0}^{N_{\text{total}}-1} x_{k-t\Xi(n)} g_n(k - t\Xi(n)) \exp(-j\frac{kf2\pi}{N_{\text{total}}}) \tag{2.22}$$

where $\Xi(n)$ is the window size, or time bin size for scale n, over which the sample values must run for each time window and n denotes the discrete scale parameter (replacing σ in the previous continuous transform). The sample size $\Xi(n)$ in the time domain is chosen to be a power of two, so that the MFT can be implemented efficiently using the Fast Fourier Transform algorithm (FFT) [43]. This is defined in terms of the level n, so that the size of the time window is directly a function of scale, or as it is known in the discrete case, level. Its definition is

$$\Xi(n) = 2^n \tag{2.23}$$

It is shown in [66] that this factor of two between levels is not only attractive in terms of being able to use the FFT algorithm, but also maximises the likelihood of features being tracked across levels. As can be seen from (2.22), the number of time samples is determined by the choice of sampling window: for critical sampling, i.e. sampling without redundancy to satisfy the equality in (2.3), it is just the total number of samples in the signal divided by the time bin size. The number of frequency samples is determined by the time bin window function size as well and, in order to invert each time bin, it is necessary to have at least as many frequency coefficients as there are samples in the time bin. If the total number of samples in the time signal is denoted by N_{total} then the number

of MFT samples on the time axis is given by

$$N_t(n) = \frac{N_{\text{total}}}{\Xi(n)} \tag{2.24}$$

where, $N_t(n)$ denotes the number of time samples in the MFT. If the signal were to be oversampled by a factor l then this expression would become

$$N_t(n) = 2^l \frac{N_{\text{total}}}{\Xi(n)} \tag{2.25}$$

accordingly. Note that in this work the term oversampling is used to refer to the amount of overlap between neighbouring windows. So that if $l = 1$ in (2.25) there would be twice as many time windows on the time axis of the MFT's time-frequency plane as each window would overlap its neighbour by 50%. For the moment however the case $l = 0$ is considered. The number of samples that are on the frequency axis of the MFT can be determined, as discussed above, by the bin size thus

$$N_f(n) = \Xi(n) \tag{2.26}$$

where $N_f(n)$ denotes the number of frequency samples in the MFT domain. If one considers the MFT plane in a similar manner to Gabor's time-frequency plane, it is easy to show that on every level, for a given input signal, there will be the same total number of coefficients determined by the product of the number of time coefficients and the number of frequency coefficients.

$$N_t(n)N_f(n) = \frac{N_{\text{total}}}{\Xi(n)}\Xi(n) = N_{\text{total}} \tag{2.27}$$

which equals the total number of coefficients in the time signal. This is reasonable, since if

$$N_t(n)N_f(n) < N_{\text{total}} \tag{2.28}$$

then the original signal could not be recovered exactly.

We shall now consider what happens to these coefficients on the time-frequency plane, as a function of level. The area of the time frequency plane and the number of coefficients on each level are invariant with level, see (2.27), so the only thing that must change is the size of each "cell" in time and frequency. This information is actually contained in the definition of the time bin size (2.23). Let $\Omega(n)$ denote the frequency sampling interval: the size of the bin in the frequency direction on the MFT plane. This is also a function of level, since the number of coefficients in the frequency plane is proportional to $\Xi(n)$. It follows that because the number of coefficients on the time-frequency plane is constant, the size of the MFT bin in the frequency direction varies inversely to the time bin size. This is a consequence of the uncertainty principle, shown in section 2.2.3 for a simple window function in time. The size of the frequency sampling interval is given by

$$\Omega(n) = \frac{2\pi}{N_f} = \frac{2\pi N_t}{N_{\text{total}}} = \frac{2\pi}{\Xi(n)} \tag{2.29}$$

It can be seen that to simplify any implementation of the MFT, if the total number of samples transformed from the time signal is a power of two, then each time bin, which is necessarily a power of two for the FFT algorithm, can divide the total number of samples exactly. From (2.29), the frequency sampling size will be a power of two also. If

$$N_{\text{total}} = 2^M \tag{2.30}$$

then the sample size in frequency can be described as

$$\Omega(n) = \frac{2\pi}{\Xi(n)} = 2^{n-M+1}\pi \tag{2.31}$$

and the number of time samples in the MFT becomes in general, from (2.25),

$$N_t(n) = 2^{M-n+1} \tag{2.32}$$

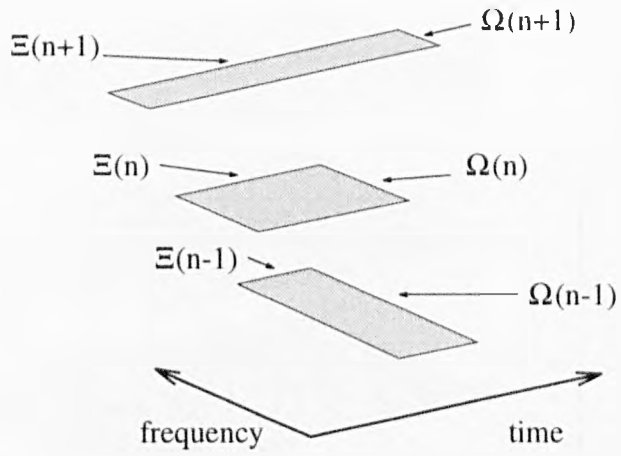

Figure 2.2: Tiles for different levels on the MFT's time-frequency plane.

An arbitrary time-frequency tile is shown in figure 2.2. where, as above, the number n denotes the level. We now have a description of the shape of the coefficients or *tiles* on the time-frequency plane, highlighted in the simple examples in figures 2.2 and 2.3. The total number of samples in the time signal represented in figure 2.3 is 4, and there are accordingly three levels shown. As can be seen, level 0 is equivalent to the original time sampled signal, as there is complete sampling in time; level 2 represents the Fourier transform of the whole signal, as there is no sampling in time, only frequency. This is a graphic illustration of the time-frequency plane for a level of the MFT with critical sampling. At each level, the total number of coefficients will be the same, with only their width and length varying as a function of level, thus altering their various resolutions in accordance with the uncertainty principle. Table 2.1 shows the time bin and frequency bin sizes in physical units for an audio signal sampled at 44.1kHz which is the sampling rate used in Compact Disc recordings and in this work, for the various MFT levels used.

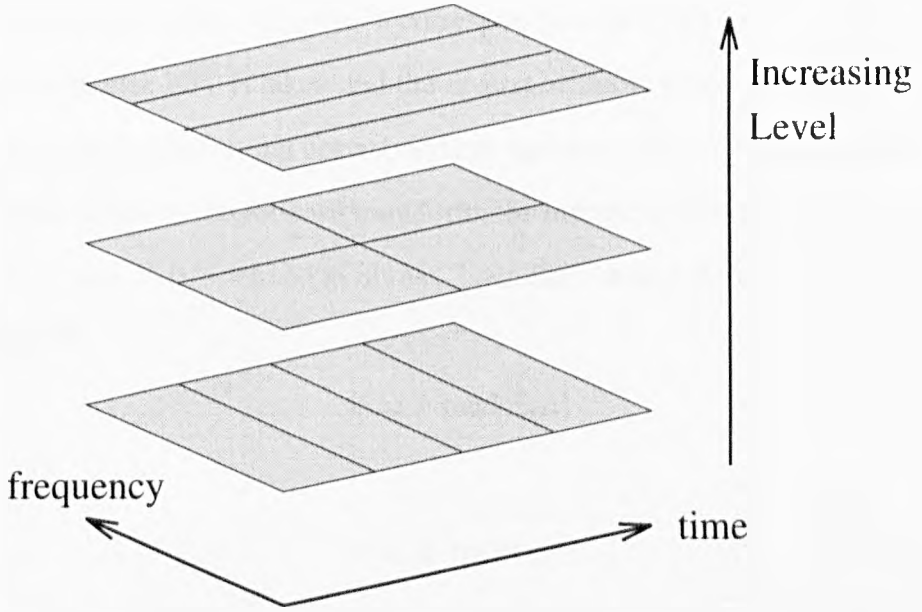

Figure 2.3: A time-frequency plane tessellation for a critically sampled MFT.

2.3.2 The Inverse Transform

As stated previously, it is of crucial importance to audio signal restoration that any analysis of the audio signals being restored be complemented by a resynthesis. This means that for each transform there must be an exact inverse.

Since the discrete MFT is a finite number of STFT's in scale space there is an inverse

level (n)	*time bin* (ms)	*frequency bin* (Hz)
8	5.804	172
9	11.61	86.1
10	23.22	43.1
11	46.4	21.5
12	92.8	10.7
13	186	5.38
14	371	2.69

Table 2.1: Time and frequency bin sizes for various levels, critically sampled with a sampling rate of 44.1kHz.

for each of these levels [9], [45], which corresponds to an STFT inverse. To perform this inversion, the inverse FFT is taken and the inverse window function applied which gives the block of signal samples that corresponds to that time bin in the MFT domain. Using the same notation as for the forward transform, the inverse is shown for level n, with time bins size $\Xi(n)$ and $N_f(n)$ defined as above. Then the value of the time signal at sample k can be found if

$$k' = k \bmod \Xi(n) \tag{2.33}$$

and

$$t = k \operatorname{div} \Xi(n) \tag{2.34}$$

then the signal is synthesised using

$$x(k) = \frac{1}{g_n(k)} \sum_{f=0}^{\Xi(n)-1} X_{t,f,n} \exp\left(j \frac{fk'2\pi}{\Xi(n)}\right) \tag{2.35}$$

which is just the inverse discrete Fourier transform for MFT bin t, determined by the position of the required time sample k. This is why $g_n(t)$ is required to be non-zero everywhere.

2.3.3 The Choice of Window Function and Sampling Interval

The MFT sampling scheme discussed in the previous section is critical sampling [14], in this each time signal sample corresponds to one sample on the time frequency plane. It can be shown, however, that having oversampling by a factor of two which corresponds to $l = 1$ in (2.25), gives a more stable representation of the time-frequency plane [14]. In this application, the question of oversampling can be considered a practical one, since the final aim of performing restoration of a noisy signal in the MFT domain will be to remove some of the coefficients. If the MFT were critically sampled restoration would cause artifacts, due to the changes in energy between adjacent time-frequency bins. This

is a problem which does not arise when the MFT is oversampled by a factor of two, as the energy from one bin will be present in the adjacent time bins. In the language of [66] and [14], oversampling is necessary to get a well behaved, or "snug", localised frame or representation. The values for $N_t(t)$ now increase by a factor of two,

$$N_t(t) = 2\frac{\Xi(n)}{N_{\text{total}}} \tag{2.36}$$

and so

$$N_t(n)N_f(n) = 2N_{\text{total}} \tag{2.37}$$

i.e. there are now twice as many coefficients in one level of the MFT domain as there are in the time domain signal.

One of the aims of this work is to transform the MFT domain structure of one audio signal, the prototype, to fit another, the target, by warping the time-axis. In so doing, it is of primary importance that the location of features in both signals be represented as accurately as possible in the MFT domain. This has implications for the choice of window function $g_n(t)$. It would make sense to choose a function that was limited in time. This would mean that the function would have only *finite extent* in the time domain. A window, or analysis function, with this property can also be described as being *truncated* in the time domain. The effect that this has in the frequency domain is to introduce sidelobes in the frequency response of the window. This effect is directly attributable to the uncertainty principle, as the energy concentration in time is inversely proportional to that in the frequency domain [65] [54] [43] .

The window functions used in this work are specifically designed to combat this problem. They are functions of the Finite Prolate Spheroidal Sequence (FPSS) class [67] [54] [43]. More precisely, in the notation of [67], they are the functions that offer a

solution to the eigenvalue equation

$$\mathbf{I}(\Xi(n))\mathbf{B}(\Omega(n))\mathbf{I}(\Xi(n))g_n = \lambda g_n \tag{2.38}$$

Here, the operators $\mathbf{I}(\Xi(n))$ and $\mathbf{B}(\Omega(n))$ are defined as square $M \times M$ matrices and the window function g_n is regarded as a column vector. This equation produces an FPSS that is time-truncated, but that has maximum energy concentration in the frequency domain. The operator $\mathbf{I}$ is

$$I_{kl}(\Xi(n)) = \begin{cases} \delta_{kl} & |x_k| < \frac{\Xi}{2} \\ 0 & \text{else} \end{cases} \tag{2.39}$$

and with $\mathbf{F}$, the Discrete Fourier transform operator, defined as

$$F_{kl} = \frac{1}{\sqrt{M}} \exp(-j\frac{2\pi kl}{M}) \tag{2.40}$$

then the bandlimiting operator $\mathbf{B}$ can be defined as

$$\mathbf{B}(\Omega(n)) = \mathbf{F}^*\mathbf{I}(\Omega(n))\mathbf{F} \tag{2.41}$$

This produces a time-truncated window function, but a function with minimum sidelobe energy in the frequency domain. The sidelobes have been measured [66] [16]. The method used to gauge the size of the sidelobes is to measure the peak ratio of the magnitudes of the centre lobe and first sidelobe. This ratio has been shown to be about 15dB for the window function generated via (2.38). However, through experimentation with these FPSS functions it is found that if the truncation operator is *relaxed* — that is, allowed to extend over a period, twice as long, then a more satisfactory window function in the frequency domain is obtained. (2.38) then becomes

$$\mathbf{I}(2\Xi(n))\mathbf{B}(\Omega(n))\mathbf{I}(\Xi(n))g_n = \lambda g_n \tag{2.42}$$

These window functions, the solutions to (2.42), have peak to first sidelobe magnitude of 26.3dB, which is much lower than suggested previously. This can be verified by examining

figures 2.6, 2.7 and 2.8. These FPSS's are called *relaxed* FPSS's, since their truncation is relaxed. This relaxation by a factor of two gives the best compromise, between the extra computation required to calculate the oversampled MFT, and the benefits of the reduced sidelobes [66]. Having a relaxed FPSS ties in with the method of oversampling: the FPSS is relaxed by a factor of 2 and the MFT is oversampled by a factor of 2.

2.3.4 Implementing the MFT

In previous implementations of the MFT [9] [16] [45], the window function has been truncated in frequency, or bandlimited. This means that, for simplicity, the implementation of the MFT needs to be in the frequency domain. This is done by taking the Fourier transform of the entire input array and the window function $g_n(t)$. This creates a single column on the time-frequency plane. The window function is shifted to cover each member of this column, and the two are multiplied together. The output of this multiplication is inverse transformed, so creating each of the rows on the time-frequency plane.

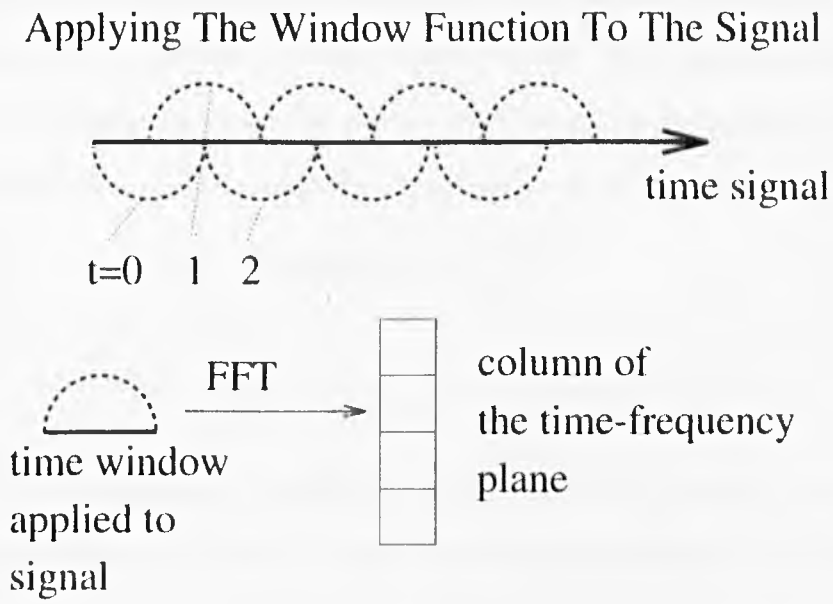

Figure 2.4: An illustration of how the discrete MFT is implemented in this work.

The implementation used in this work is much simpler since, as stated above, the window functions here are time limited. The basic algorithm is:

1. the window is applied to the start of the audio signal, covering $\Xi(n)$ samples;
2. the FFT is taken of these windowed samples, creating one column on the time-frequency plane;
3. the window is shifted by $\frac{\Xi(n)}{2}$;
4. steps 2 and 3 are repeated until all of the desired signal samples have been windowed and transformed.

This algorithm is described pictorially in figure 2.4. The advantage of this implementation is that there is no need to take any large FFT's as in Pearson [45], where an FFT was taken of the entire input array of 2^{16} samples and then the inverse FFT taken of each row.

It is important to note that there is a phase difference inherent between the implementation here and (2.22). Strictly speaking, performing the above implementation does not give an MFT. Consider the phase for the two MFT time bins according to (2.22), for a complex exponential signal of frequency f, time bins t and $t+1$, with $0 \leq k < \Xi(n)$

$$\arg(X_{t,f,n}) = \frac{kf2\pi}{\Xi(n)} \tag{2.43}$$

and

$$\arg(X_{t+1,f,n}) = \frac{(k+\frac{\Xi(n)}{2})f2\pi}{\Xi(n)} \tag{2.44}$$

where the $\frac{\Xi(n)}{2}$ is introduced by the shift in time of half a bin, between successive MFT samples. Then the phase difference can be calculated by subtracting (2.43) from (2.44) giving

$$\arg(X_{t+1,f,n}) - \arg(X_{t,f,n}) = f\pi \tag{2.45}$$

between successive time bins. In other words, for a constant frequency index f, there should be a constant change of phase between successive bins of π. The implementation above does not change phase between successive bins as the FFT is applied separately to each bin, so that in this implementation, there is constant phase between successive time bins. This has no significant consequences in the restoration application . The FPSS generation algorithm employed here is a slightly modified form of that used previously in audio signal analysis [45]. A general algorithm is used that can create FPSS's efficiently, for relatively small input sequences. In [45] the size of the FPSS required was halved; this was then created efficiently using the algorithm suggested in [67]. The FPSS's are then "grown" by recursively oversampling, time truncating and bandlimiting until the required size is achieved. This recursive approximation is a stable operation, as has been shown in [67]. For a further description of this algorithm, see [45]. A relaxed FPSS generated using this technique is shown in figure 2.5, while figures 2.6 and 2.7 show the corresponding time and frequency profiles: figure 2.5 is the product of the two which indicates the time and frequency extent of the window. In figure 2.8 the sidelobes can be seen to be over 20dB below the main peak.

2.3.5 Inverting the Oversampled Discrete MFT

To explain the inversion of an oversampled discrete MFT consider figure 2.4. It can be seen that the time samples in the window at $t = 1$ are present also in the time windows at $t = 0$ and $t = 2$. But both the windows at $t = 0$ and $t = 2$ contain information not needed to reproduce the samples in the time window at $t = 1$. The method used to invert the discrete MFT relies on the fact that the window function $g_n(t)$ approximates a cosine function. Using the function

$$h_n(t) = \frac{\cos^2(t-1)}{g_n(t)} \tag{2.46}$$

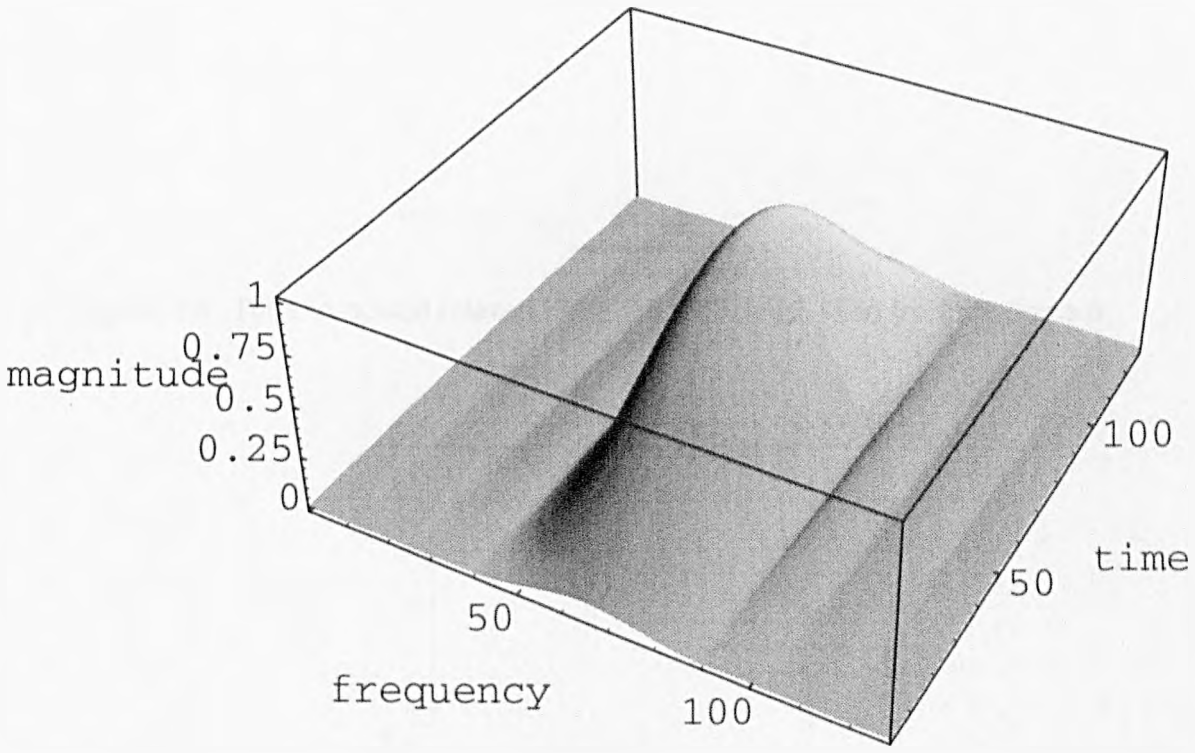

Figure 2.5: Time truncated relaxed FPSS for MFT level 11.

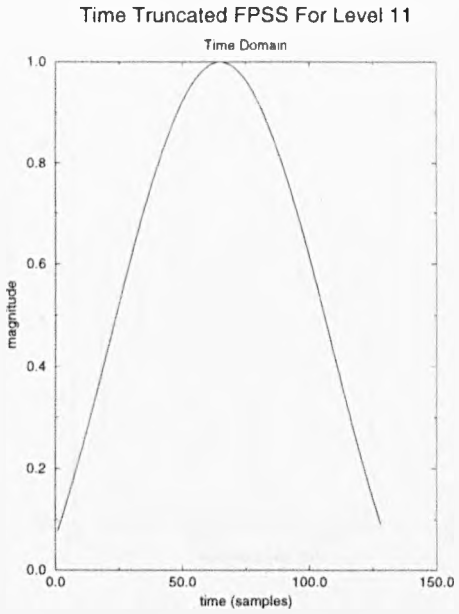

Figure 2.6: Time truncated relaxed FPSS for MFT level 11 in the time domain.

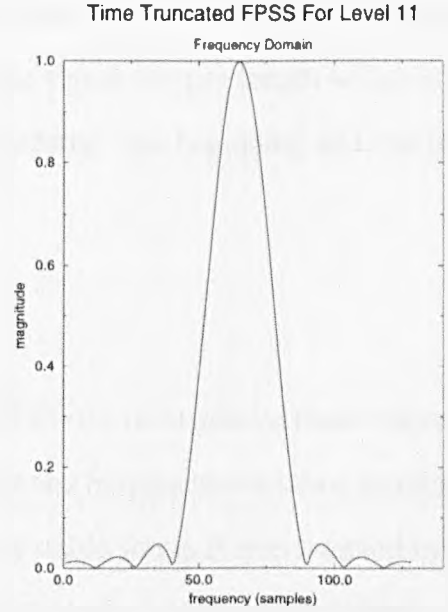

Figure 2.7: Magnitude of the time truncated relaxed FPSS for MFT level 11 in the frequency domain.

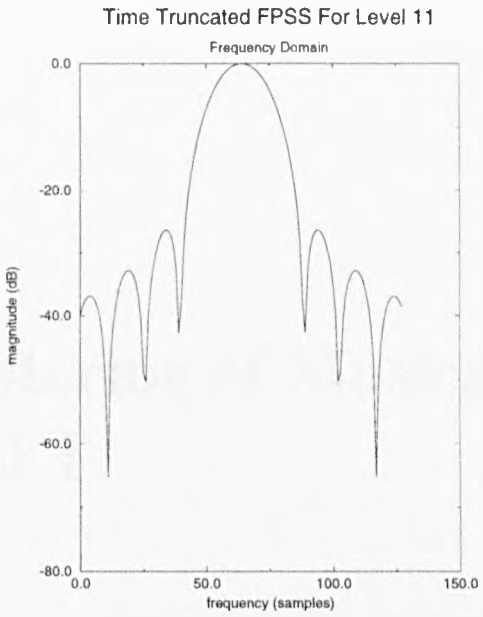

Figure 2.8: Time truncated relaxed FPSS for MFT level 11 in the time domain with a log scale.

results in adjacent windows summing to unity on inversion. This process is continued for all windows. One problem caused by this is that the first half window length samples at the beginning and end of the signal sample length will not be invertible. This problem can be easily avoided by "padding" the beginning and end of all MFT samples by half a window length of zeros.

2.4 Summary

It has been shown that the MFT is the most general time-frequency representation available: allowing a free choice of time and frequency resolution as a function of scale. Furthermore, the discrete MFT provides a stable frame if oversampled by a factor of 2 and as such is completely invertible from any level. An efficient and simple time domain implementation of the MFT for $1-d$ signals was given. In conclusion, the MFT is a versatile analysis tool and it can be implemented without difficulty.

Chapter 3

Adaptive Filtering of Musical Signals using the MFT

3.1 Introduction

In restoring a musical signal, it is necessary to determine what is music and what is not. The listener can distinguish between background sound — sound that is not musical, for example the sound of traffic or rain — and a musical composition. There must be something about music — it must have some property or properties that background noise does not have. A signal model would allow a better understanding of the structure of musical signals, and so aid in the recognition, detection and ultimately, the restoration of audio musical signals in a noisy environment.

3.2 A Musical Signal Model

There are many types of music. Indeed there exist musical genres whose aim it is to blur the border between music and sound, for example Schaeffer in his "Musique Concrète" used background sounds and decontextualised them, integrating them into a musical performance [42]. So, when deciding on a musical model, it is helpful to be sure of what type of music is to be modelled. It is a condition of this work that any music used

and therefore modelled here, must be such that there exist two performances of the same musical piece that can be mapped, or warped, one onto the other. Therefore any type of music that is to be modelled should have some fairly rigid rules as to how any given musical piece is performed. It would be rather difficult to warp one jazz performance to another as there will generally be not only large variation in timing and possibly pitch, but also in the notes being played. The type of music that is of interest will need to have some written score that is followed without too much interpretation. One such type of music is classical music, where classical music can be defined to be "any type of music written to conform to traditions of the past", or "music that is written to bring order to life" [5]. This musical genre generally has a comprehensive score for each piece and, although there is always room for interpretation between performances, any two performances of a piece will not vary unrecognizably. Fortunately, this is also the area where there is much interest in the restoration of old recordings.

Because the score is what makes different performances of a musical piece similar, it is to the score that we look when deciding how to build a musical signal model. The score is essentially a list of instructions to the performers and as such can be considered to be a collection of notes with different timings, strengths and pitch. It then follows that if a signal model can be designed for a musical note, it is only a simple extension from this to a whole performance.

For the purposes of this work, a note consists of three main parts. The first is the tone of the note, which is the frequency at which the note is heard. The second is the envelope of the note, which is the shape of the attack and decay. This is a function of the instrument played and the musician playing. For example, a piano's attack is sharp, but can be dampened slightly by the pianist pressing the key more slowly, a violin has a slow attack, but this can be quickened by the violinist. The third property of a note is its

timbre. The timbre is a function of the instrument being played, and is the distribution of the energy of a note in frequency. A note is made of partials, or harmonics, of some fundamental, where the fundamental is the base frequency and each partial's frequency is a multiple of the frequency of the fundamental. As the timbre alters, the distribution of the note's energy changes through the partials, so that in some instruments certain partials will have more prominence than others. It is this distribution of energy across frequency that gives an instrument at least part of its distinctive sound.

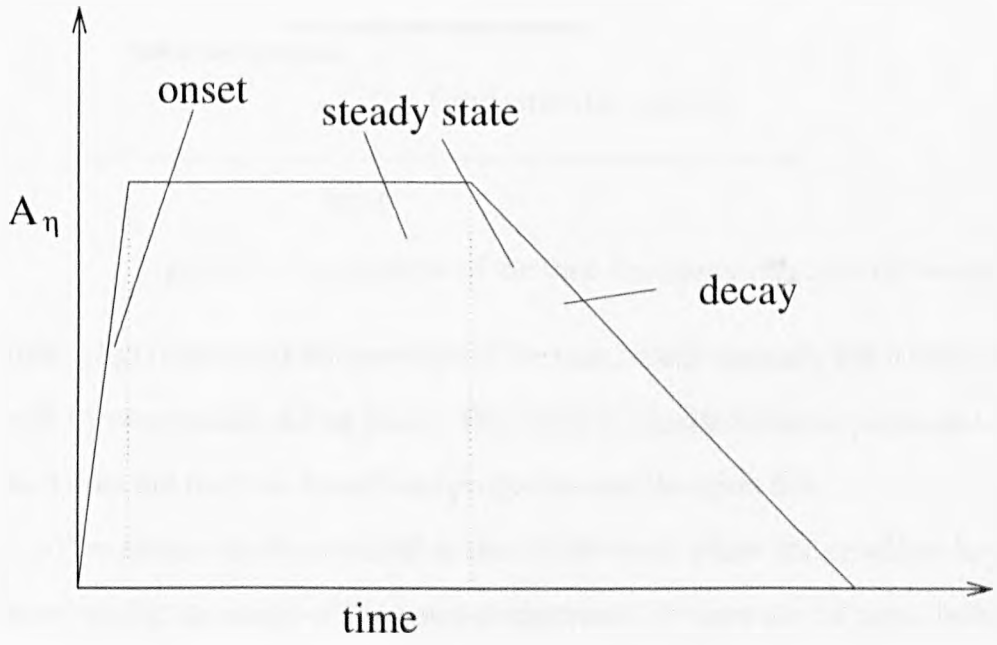

Figure 3.1: An example of the envelope of a note.

A partial can be assumed to be a quasi-sinusoid, [45]. If a note is just a sum of partials, or harmonics, then it can be modelled as

$$s(t) = \sum_{\eta=1}^{N} A_\eta(t) \sin(\omega_0 \eta t + \phi(\eta)) \tag{3.1}$$

where N is the number of partials present, ω_0 is the fundamental frequency or tone, $\phi(\eta)$ is an offset phase, and $A_\eta(t)$ can be defined as an amplitude function varying as a function of

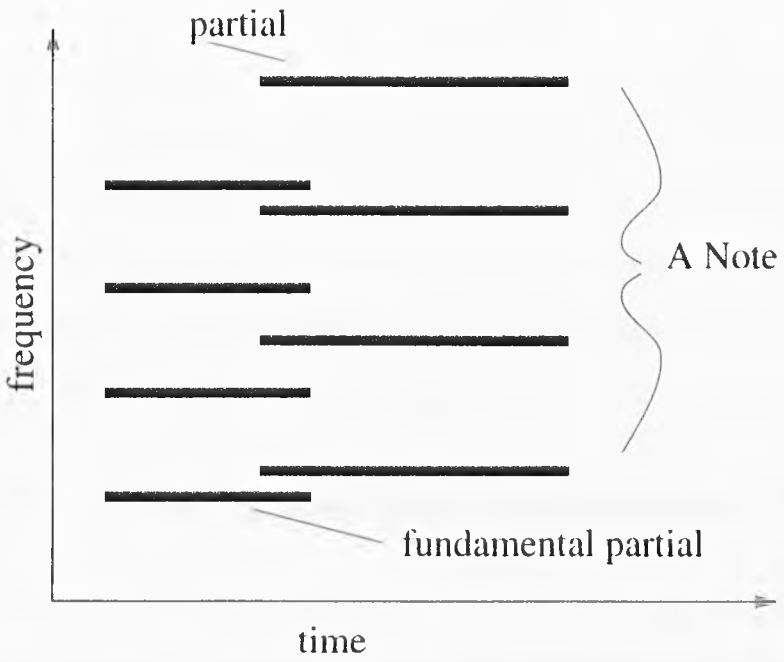

Figure 3.2: An example of the time-frequency structure of two notes.

time. A_η(t) represents the envelope of the note, which typically has a short attack or onset and, by comparison, a long decay. The decay is classified for our purposes as steady-state, as it does not have the broadband properties that the onset has.

This notion can be extended to the whole score, where the envelope function is set to zero outside the scope of the notes it represents. If there are M notes being played, and each note indexed with m is of the form in (3.1), then each note in a first approximation of a musical score could be

$$s_m(t) = \sum_{\eta=1}^{N} A_{\eta m}(t) \sin(\omega_0(m)\eta t + \phi(\eta)) \tag{3.2}$$

$A_{\eta m}(t)$ is a weighting function that encompasses both the distribution of energy through the partials η, and the time course of each. The set of all such notes, representing the score, would be

$$s(t) = \sum_{m=1}^{M} s_m(t) \tag{3.3}$$

Thus (3.3) shows that a simple musical signal model can be described as a collection of amplitude modulated sine waves in frequency, at multiples of some fundamental frequencies, occurring as a function of time. Let us now consider some of the properties that notes and their partials have in the time-frequency plane. Although each partial will occupy two frequency bins because of the sampling and the windowing of the MFT implementation (cf chapter 2), the general shape of the signal is shown in idealised form in figure 3.1 and a practical example is shown in figure 3.4.

3.3 Audio Restoration using Lowpass Filtering

One way to restore noisy signals is to use a lowpass filter, which is easy to implement, using the MFT, by selecting a cut-off frequency and setting all MFT coefficients above that cut-off equal to zero before reconstruction. Figure 3.5 shows how, for a given MFT level, coefficients at frequency bins less than the cut-off frequency f_c are included and the rest are ignored. The cut-offs used in this work are at 1.5kHz, 3kHz and 4.5kHz. 1.5kHz and 4.5kHz are included to show that by raising and lowering the cut-off frequency too much, the lowpass filter's performance deteriorates. The filtering is done on the "Waldstein" Adagio Molto and String Quartet Scherzo, with white additive noise. The gain in SNR is plotted against level for the three cut-off values. As might be expected, because the filter is constant throughout the sample, there is little dependence of the gain on the choice of level. (Note that level 11 has a time window duration of 21.5 ms). Figure 3.6 is a plot of gain in SNR against level with an input SNR of 10dB. Each curve in this plot represents one of the three cut-offs discussed above. The two extreme cut-offs, namely 1.5kHz and 4.5kHz both give a lower gain than 3.0kHz confirming that the best place to have the cut-off is where the signal energy is lower than the noise energy. For example, on a piano keyboard there are only five keys that produce a note with a fundamental frequency greater

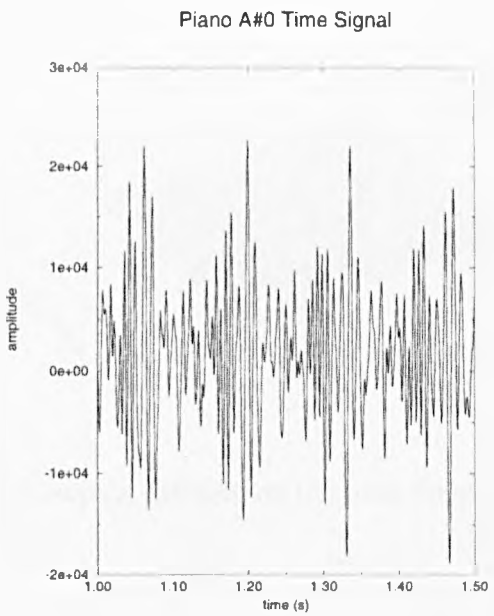

Figure 3.3: A section of time signal for piano note A#0.

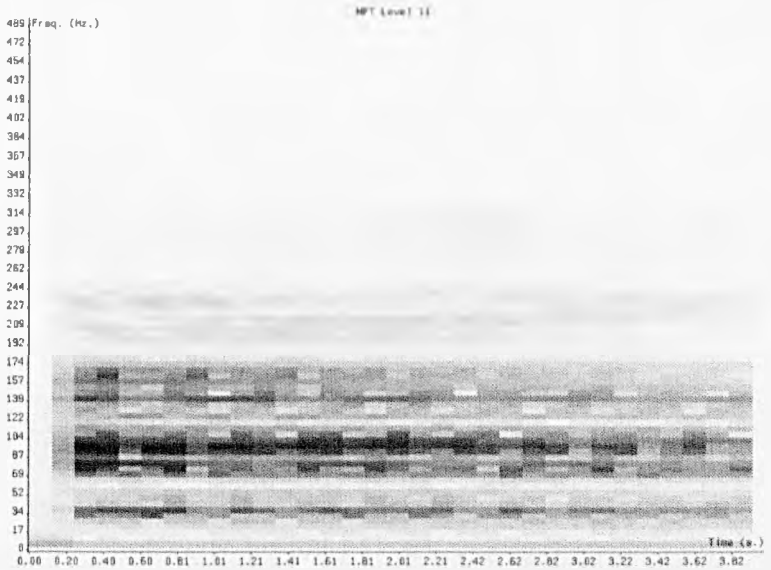

Figure 3.4: MFT time-frequency plane for the piano note A#0 at MFT level 11.

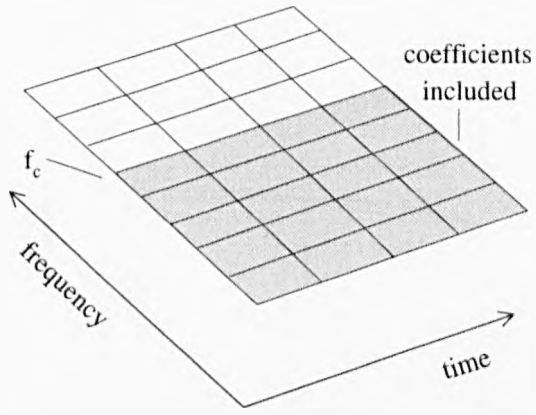

Figure 3.5: Lowpass filtering on the time-frequency plane.

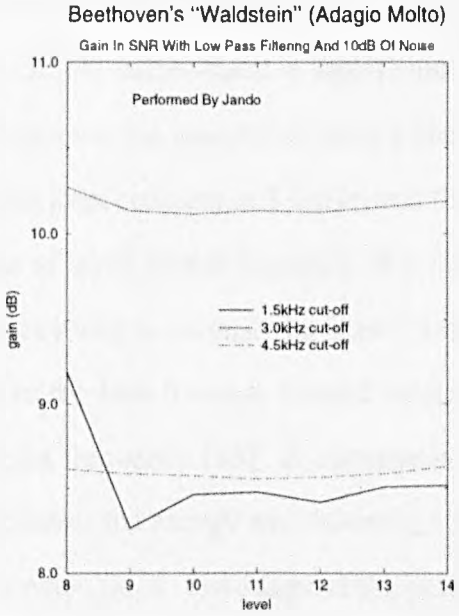

Figure 3.6: Beethoven's "Waldstein" (Adagio Molto) restored using a lowpass filter at various cut-off frequencies, with 10dB input SNR, based on various MFT levels.

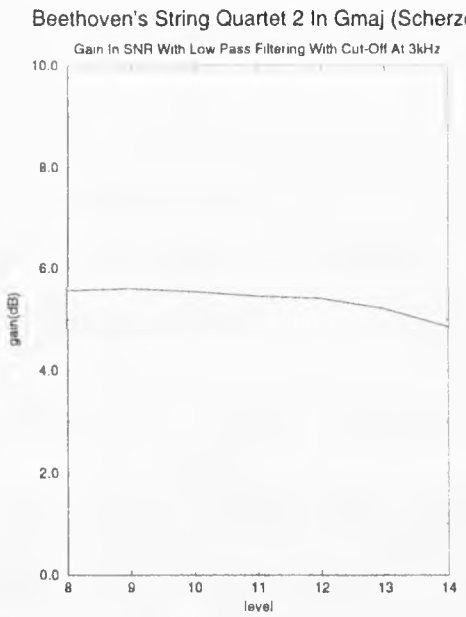

Figure 3.7: Beethoven's String Quartet no 2 in Gmaj (Scherzo) restored using a lowpass filter with a cut-off frequency of 3kHz, with 10dB input SNR, using various MFT levels.

than 3.0kHz. The shape of the curve for the cut-off at 1.5kHz is different from the other two because at such a low cut-off value, there is significant signal missing, whereas in the other two the main difference is the amount of noise removed. For the String Quartet Scherzo, the cut-off value was kept constant at 3.0kHz, and the level was varied, showing that the gain is not a function of level, as was expected. It is obvious that as the input SNR decreases, the cut-off frequency will necessarily increase: if there is no noise, the amount of "noisy signal" included in the best lowpass filtered output signal will have a cut-off frequency equal to the Nyquist frequency [43]. To summarise, the lowpass filter requires a choice of cut-off that maximises the energy and minimises the noise transmitted, and is a function of input signal to noise ratio. The shape of the restoration curve does not vary as a function of the MFT level.

3.4 Audio Restoration using Wiener Filtering

A Wiener filter is an optimal filter, which minimises the mean square restoration error. An approximate Wiener filter was used by Vaseghi [59] to restore gramophone recordings, by assuming that the noise and the signal were not correlated. A Wiener filter $h(\tau)$ can be derived from the Wiener-Hopf equation [43]

$$R_{xx}(t) = \int_{-\infty}^{\infty} R_{sx}(t-\tau)h(\tau)d\tau \tag{3.4}$$

where the noisy signal $x(t) = s(t) + r(t)$ as previously, and R_{sx} is the cross-correlation of $x(t)$ and $s(t)$. This can be solved in the Fourier domain to give

$$H(\omega) = \frac{S_{sx}(\omega)}{S_{xx}(\omega)} \tag{3.5}$$

where $H(\omega)$ is the Fourier transform of the optimal Wiener filter, $S_{xx}(\omega)$ is the power spectral density of signal $x(t)$ and $S_{sx}(\omega)$ the cross-spectral density of signals $x(t)$ and $s(t)$. It is assumed that the noise and signal elements of $x(t)$ are not correlated and so $S_{xx}(\omega)$ reduces to

$$S_{xx}(\omega) = S_{ss}(\omega) + S_{nn}(\omega) \tag{3.6}$$

where $S_{nn}(\omega)$ is the Fourier transform of the noise signal's auto-correlation function, and

$$S_{sx}(\omega) = S_{ss}(\omega) \tag{3.7}$$

(3.5) now becomes

$$H(w) = \frac{S_{ss}(\omega)}{S_{ss}(\omega) + S_{nn}(\omega)} \tag{3.8}$$

which is just the signal spectrum of the clean signal divided by itself plus the spectrum of the noisy signal. This simple structure results because, for a stationary signal, the Fourier basis diagonalises the correlation function. Suppose, for the moment, that the spectrum

of the signal and noise are known, allowing calculation of the Wiener filter. It can be seen from (3.8) that for a given frequency w, if there is a low signal spectral energy, the Wiener filter value will be low also. If, conversely, the spectral energy is significantly larger than that of the noise, then the Wiener filter has a gain of unity. Implementing the Wiener filter in the MFT domain is simple, as the signal spectrum $S_{xx}(w)$ can be estimated by summing the MFT coefficients for a given frequency, over all time. If the discrete spectrum estimate is $S_{xx}(f)$, then

$$S_{xx}(f) = \sum_{t=0}^{T} |X_{t,f,n}|^2 \tag{3.9}$$

where T is the interval over which the filter is computed. An example of the frequency response of a Wiener filter is shown in figure 3.8, which is the filter designed for Beethoven's "Waldstein" Adagio Molto with 10dB of white additive noise. One thing to note is that the filter decays rapidly at about 4kHz, confirming the supposition made in the previous section that the signal values dominate only until about that point in the frequency spectrum. The results for restoration with 10dB, 15dB and 20dB of white additive noise added to Beethoven's "Waldstein" (Adagio Molto), and String Quartet 2 in Gmaj (Scherzo) are shown in figures 3.10 and 3.11 respectively, where the gain in SNR is plotted against MFT level. One point that is obvious upon inspection is that for high MFT levels there is a larger gain on both restorations than for low levels. This can be explained simply by considering (3.9), which shows that the signal spectrum estimate is a function of level. The bandwidth for coefficients is narrower for high MFT levels: the frequency resolution increases, as does the accuracy of the Wiener filter, since all the time coefficients are summed. In figure 3.12 a plot is made of filter length (shown in seconds) against gain using the Wiener filter. This shows that as the length T of the window used to compute the Wiener filter increases beyond an optimal length there is a deterioration in the gain. There are two main variables affecting the optimal interval in this implementation of Wiener filtering. The

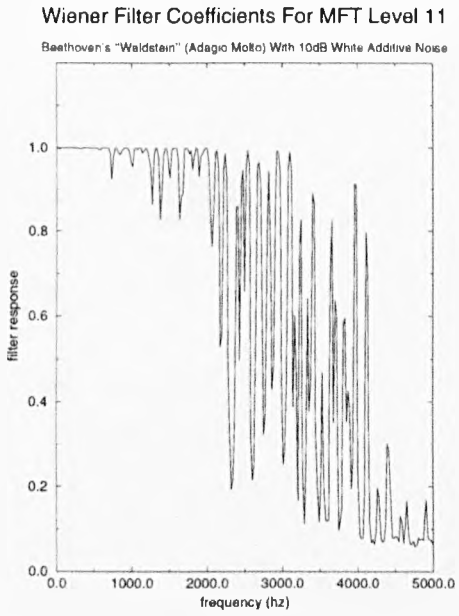

Figure 3.8: Wiener filter frequency response for Beethoven's "Waldstein" (Adagio Molto), with 10dB input SNR.

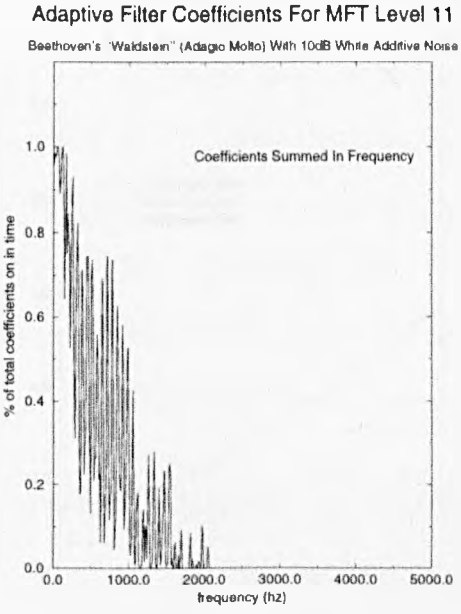

Figure 3.9: The adaptive filter $H_{t,f,n}$ summed in time to compare with Wiener filter for Beethoven's "Waldstein" (Adagio Molto), with 10dB input SNR (see section 3.5).

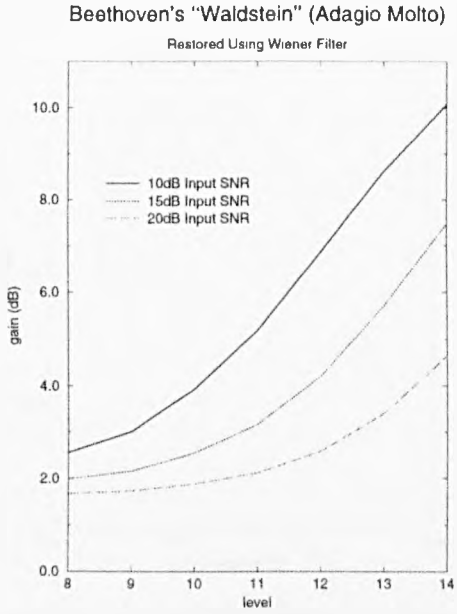

Figure 3.10: Beethoven's "Waldstein" (Adagio Molto) performed by Jandó restored using a Wiener filter for various input SNR's.

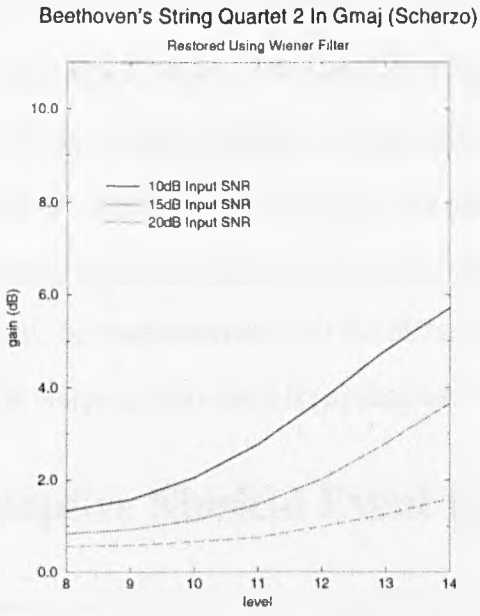

Figure 3.11: Beethoven's String Quartet no 2 in Gmaj (Scherzo) performed by The Smithsonian String Quartet restored using a Wiener filter for various input SNR's.

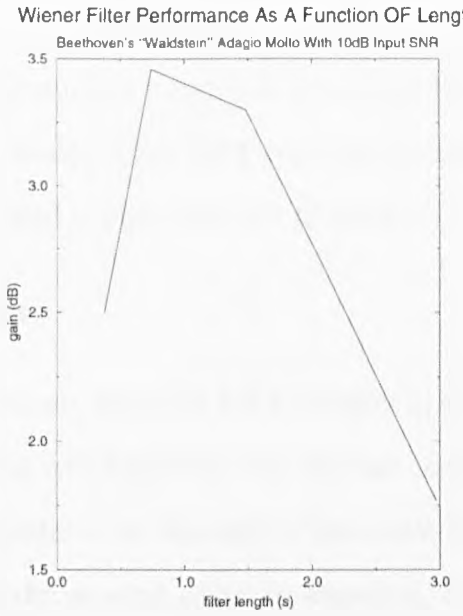

Figure 3.12: Gain on restoring Beethoven's "Waldstein" (Adagio Molto) with input SNR of 10dB using a Wiener filter as a function of filter length for intermediate MFT level 10.

first is the signal spectrum estimate, $S_{xx}(\omega)$ which, as the period of time that it pertains to decreases, becomes more adaptive but, as the filter length increases, becomes more of an average estimate for that length of signal. The second is the noise estimate, $S_{nn}(\omega)$, whose accuracy decreases if the window length becomes too short. There is therefore a trade-off between a good noise estimate and an accurate estimate of the spectral values for that time period. This problem will be addressed further in chapter 5, where an adaptive form of the Wiener filter will be implemented for two different audio signals, the target and the prototype, using the warping algorithm from chapter 4.

3.5 A Simple Adaptive Musical Event Detector

The aim of event detection is to detect the meaningful parts of musical signals: those which correspond to the partials of the notes being played. This is done by thresholding the signal at a level based on an estimate of the noise variance. This creates a binary output

that can be used to adapt the restoration filter to the spectral and temporal properties of the signal. This idea is illustrated in figure 3.2, which can be interpreted as the structure of the partials of a piece of music. Each MFT coefficient which is turned "on" in this plot represents a filter concentrated in both time and frequency.

3.5.1 The Signal Filter

As stated in the previous section, using the MFT (chapter 2) on musical signals shows that each partial occupies at least two frequency bins and has constant relative phase. Section 3.2 explained the signal model to be the sum of the constituent partials of the notes in a performance. Detecting the musical signal is, therefore, equivalent to detecting all of the partials. The detection filter acts in both time and frequency. The time filter is a first order recursive filter, which has been shown to be optimal for transient detection [51], the filter sums adjacent frequency pairs of MFT coefficients, so that its peak output is when the phase of both frequency bins is the same. This filter was used in [45] for detecting partials for note transcription. In order to keep the same absolute length of memory, the recursion coefficient, α, is changed according to n.

The output of the filter at time t, $V_{t,f,n}$ is given by

$$V_{t,f,n} = \alpha(n)(X_{t,f,n} + X_{t,f+1,n}) + (1 - \alpha(n))V_{t-1,f,n} \tag{3.10}$$

where $X_{t,f,n}$ is as before, the MFT coefficient at time t, frequency f and level n.

3.5.2 The Noise or Background Level Estimator

An estimate of the level of the background noise is found by using a separable first order recursive filter. This has a long memory in time and frequency, so that events in the past and higher (and usually lower energy) partials contribute to the estimate of the noise level.

The output of filter at time t is given by

$$W_{t,f,n} = \beta(n)(\gamma(n)X^2_{t,f,n} + (1-\gamma(n))W_{t-1,f,n}) + (1-\beta(n))(\gamma(n)X^2_{t,f+1,n} + (1-\gamma(n))W_{t-1,f+1,n}) \tag{3.11}$$

this is in effect an estimate of the noise variance. The noise standard deviation estimate is therefore just the square root of this, for time t and f

$$\sigma = \sqrt{W_{t,f,n}} \tag{3.12}$$

for coefficients t, f and n. Note also that $\gamma(n)$ and $\beta(n)$ are functions of level for the same reason as $\alpha(n)$. It is important to note that in this instance, and this instance only, "noise" means that which does not adhere to the signal model, in other words the "background" level. The template is actually derived from a clean signal, which should have minimal noise.

3.5.3 The Binary Template

The output from the signal and noise filters can be combined to detect the signal. The most important value in this combined filter is the global threshold ν, defined in terms of the maximum coefficient value

$$\Theta_n = \max_{t,f} X_{t,f,n} \tag{3.13}$$

see section 3.5.5 for how ν is chosen empirically and section 3.8 for a preliminary analytical method of choosing ν.

If μ is the activity threshold, the adaptive filter can be determined from

$$\begin{aligned} H_{t,f,n} &= 1 \quad \text{if} \quad V_{t,f,n} > \mu\sqrt{W_{t,f,n}} \text{ and } V_{t,f,n} > \nu\Theta_n \\ H_{t,f,n} &= 0 \quad \text{otherwise} \end{aligned} \tag{3.14}$$

μ serves as a local threshold, so that even if the signal filter output is large, it has to be larger than the local noise estimate. The values for μ, α, γ and β are fixed, and vary only

as a function of level. This is because the noise estimation filter requires that γ and β are large (> 0.9) to estimate over a wide range of frequencies and α must be reasonably large (typically > 0.8), to reduce false alarms. The value for μ is typically fixed at about 0.001. It is important to note that the number of coefficients switched "on" in the filter depends on the number of coefficients greater than $\nu\Theta_n$.

3.5.4 Computational and Storage Complexity

In each level of the adaptive filter, that is for each value of n, there are $N_t(n)$ time bins and $N_f(n)$ frequency bins, whose product is equal to twice the number of samples in the input audio signal. In the implementation of the filter algorithm for a single value of n there are two passes through the data: one to calculate the signal filter values and the noise filter values, and one to compare the signal values with the two thresholds, $\mu\sqrt{W_{t,f,n}}$ and $\nu\Theta_n$. In the first loop there are 6 multiplications and 8 additions — to normalise and implement the filters and in the second loop there is only one multiplication. Thus the filter's computational complexity is proportional to $2N_{\text{total}}$ for each level — as the number of multiplications and additions as stated above vary linearly with N_{total}. The number of MFT levels is

$$N_{\text{levels}} = \log_2(N_{\text{total}}) + 1 \tag{3.15}$$

So the computational complexity for the adaptive filter for an audio signal with N_{total} samples for each level is

$$N = 2N_{\text{total}}N_{\text{levels}} \tag{3.16}$$

which can be written in standard "big O" notation [1]

$$N = O(N_{\text{total}} \log_2(N_{\text{total}})) \tag{3.17}$$

Since this filter is derived from the MFT, the MFT's computational complexity will also be calculated. Consider each level of the MFT with $N_t(n)$ time bins each $N_f(n)$ wide. Calculation of each of these time bins involves both a forward and inverse FFT. Since the computational complexity of the FFT is well known as $N\log(N)$ multiplications [43], the computational complexity for each level of the MFT is

$$N_n = O(N_{\text{total}} \log_2(N_{\text{total}})) \tag{3.18}$$

Since the number of levels of the MFT, as stated above, is $\log_2(N_{\text{total}})$ then the total computational complexity for the entire MFT derived filter structure is

$$N = O(N_{\text{total}} \log_2^2(N_{\text{total}})) \tag{3.19}$$

This is illustrated graphically in figure 3.13. In a timed experiment for an audio signal

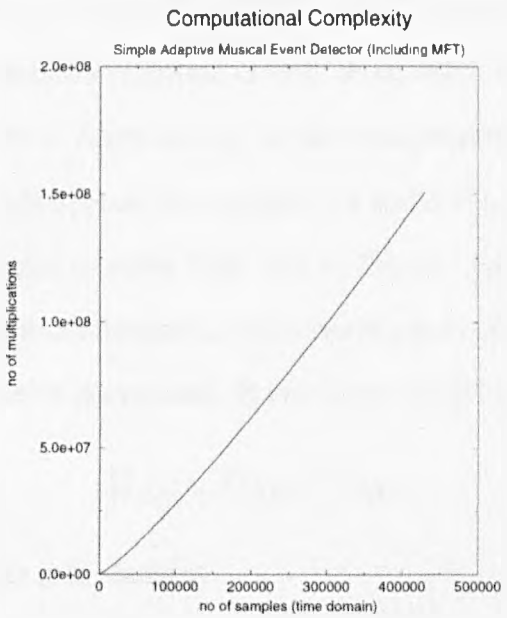

Figure 3.13: Number of multiplications as a function of sample size for the simple adaptive filter algorithm.

11.8 seconds long, sampled at 44.1kHz, it took 18 seconds to calculate one level of the

MFT and a further 19 seconds to calculate the simple adaptive filter, the hardware used was discussed in section 1.6.1.

Storage of the filter in its raw form obviously requires a total of $2N_{\text{total}} \log_2(N_{\text{total}})$ bits — 1 bit per coefficient. This can be further reduced by run length coding in time, if required.

3.5.5 Choice of Global Threshold ν

The filter, or template, is an array of MFT coefficients that, when applied to a noisy signal, permits only those coefficients with significant energy to be allowed through. This is the basis of the adaptive estimation procedure. It can be seen that as the number of coefficients in the template increases, so too does the potential gain in the signal to noise ratio (SNR) of the enhanced signal. Any signal restoration or enhancement scheme implies a trade-off between noise elimination and signal distortion, (see for example [43]). In the present system, a given template implies retention of only those MFT coefficients exceeding the threshold. The distribution of signal energy in the time-frequency plane is non-uniform but the noise energy is evenly spread (see figures 3.14 and 3.15), so using such a threshold an improvement in the signal to noise ratio will be found. For a given input SNR and MFT level, there is an optimum threshold, such that the sum of the signal rejection error and the noise retention error is minimised. If the noisy signal has MFT

$$X_{t,f,n} = S_{t,f,n} + N_{t,f,n} \tag{3.20}$$

then the error variance after smoothing is

$$E(e_{t,f,n}) = H_{t,f,n} E(N^2_{t,f,n}) + (1 - H_{t,f,n}) E(S^2_{t,f,n}) \tag{3.21}$$

where $H_{t,f,n}(\nu) = 1$ if the signal is above the threshold and $H_{t,f,n}(\nu) = 0$ otherwise. Because the implementation of the MFT ensures that the representation is a tight frame,

[14], [66], it follows that the error variance in the signal reconstructed using the template on one level is simply the mean of $e_{t,f,n}$, which depends on the threshold, ν: a larger threshold implies more signal rejection, but also more noise rejection. At present there is no adequate theory for threshold selection, and so the thresholding is done empirically. However there is a preliminary analytical method that exploits the errors in (3.21) in section 3.8.

3.6 The Structure of the Simple Adaptive Filter

The adaptive filter described above is called a template filter since it represents a "template" of the signal in the time-frequency plane at a given resolution. Its structure reflects the signal from which it was derived, namely a musical signal. It therefore has rows of coefficients switched on where the energy in the signal is concentrated. Since a musical signal's energy is distributed throughout its partials, the binary template has coefficients switched on in a pattern reminiscent of figure 3.2. Furthermore, since a musical signal is a function of time, so too is its template. In other words a binary template filter can be considered as a bank of time-varying narrowband filters reflecting the structure of the musical signal. Two templates are shown for Beethoven's "Waldstein" (Allegro Con Brio) in figures 3.14 and 3.15. These show how the coefficients vary as a function of time and frequency, at two scales. In both figures the x-axis represents time in seconds, the y-axis frequency in Hertz and the intensity of the image is proportional to the magnitude of the coefficients in the MFT, in this case black represents "on", and white "off". Note also how the visible structures of the two filters differ. That in figure 3.14 appears to be more horizontal, whilst in figure 3.15, there is a distinct vertical structure. It is easy to see from these two filters how a low MFT level or scale could capture more transient energy than the higher level. This will be discussed later. It is interesting to consider what the filter

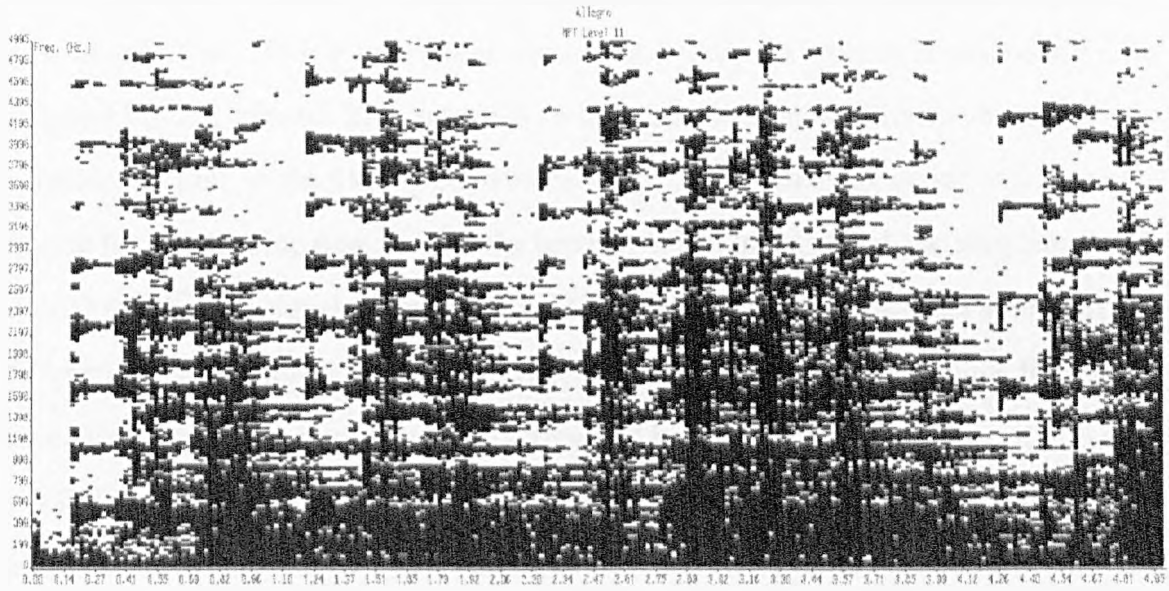

Figure 3.14: A simple adaptive filter for Beethoven's "Waldstein" (Allegro Con Brio) for MFT level 11. Note that black indicates a coefficient that is switched "on".

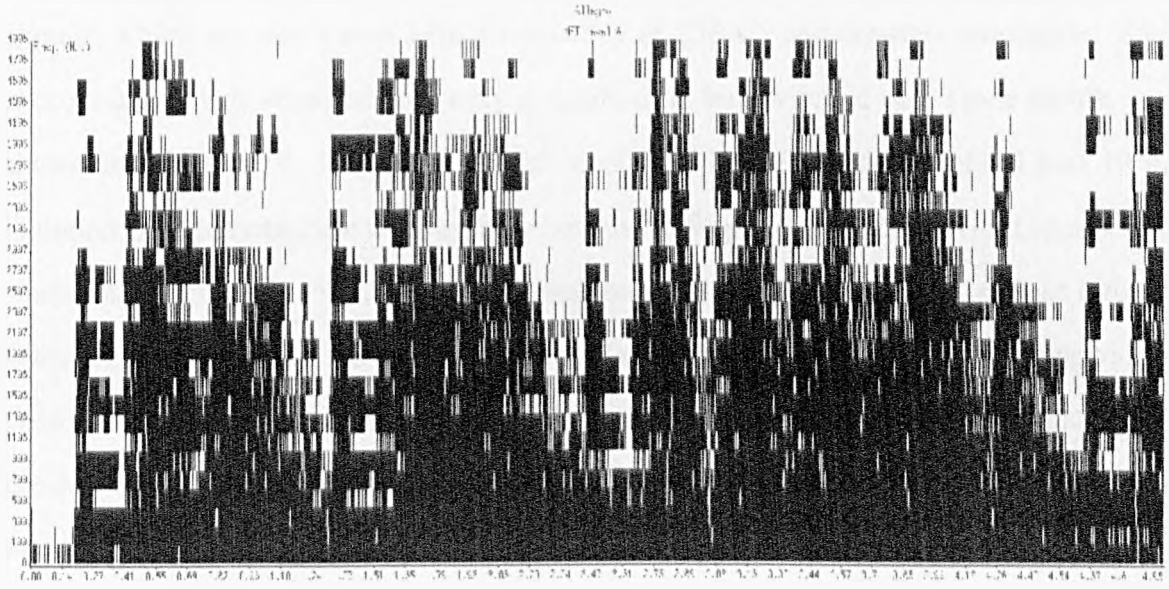

Figure 3.15: A simple adaptive filter for Beethoven's "Waldstein" (Allegro Con Brio) for MFT level 8.

would look like in the time domain. This is possible since the MFT is invertible. Note that it is not strictly an MFT, as it has been created artificially. In order to invert the MFT, the standard inverse (chapter 2) is amended so that there is no multiplication by the inverse window function, as the filter is not windowed, but the cosine windowing is still used to exploit the overlapping time bins for the inverse. When transformed, the template sounds similar to the clean signal in structure, but without the musical texture because there is a different phase structure. It does show that there are no discontinuities in the filter in the time-frequency plane, implying that filtering will be smooth and consistent.

To illustrate the point, a simple example has been created for two levels: level 8 and level 12. For all time, a frequency band was switched on by setting the value of the MFT array to one at that frequency band. The inverse MFT was applied without the inverse FPSS function but with the usual ($\cos^2$) window function on which the inverse MFT is based. Figures 3.16 and 3.17 show the resulting time signals obtained for MFT levels 12 and 8 respectively. The frequency band selected was 258 Hz, as can be seen in both time signals, which are sine waves with a frequency of 258 Hz and constant amplitude. The process above was repeated with only a single time bin switched on. These results are shown in figures 3.18, 3.19 and 3.20, for levels 12, 10 and 8 respectively. Level 10 is included as an intermediate level to show how the wave packets produced from inverting a single MFT bin vary in scale from high levels to low levels. The structure in these figures is that of a wave packet function, with sinusoidal "carrier" structure, corresponding to the position on the frequency axis that they were on the time-frequency plane, modulated by the cosine squared window function. By inspecting figures 3.18 and 3.20, it is possible to see that by adding the wave packet to a neighbour shifted by half the window size and having the same phase, that these wave packets will interfere constructively and create the sine wave shown in figures 3.16 and 3.17. It is reassuring that the time signal

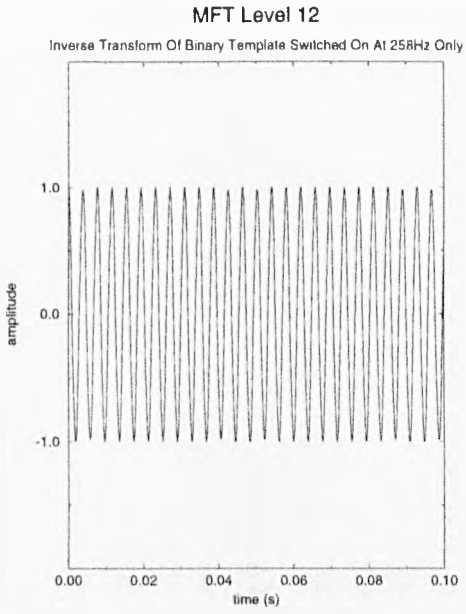

Figure 3.16: The MFT inverse for a simple adaptive filter structure with frequency bin turned on at 258Hz throughout time at MFT level 12.

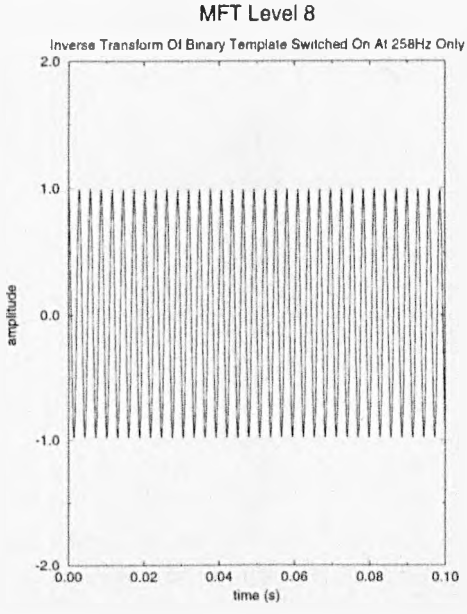

Figure 3.17: The MFT inverse for a simple adaptive filter structure with frequency bin turned on at 258Hz throughout time at MFT level 8

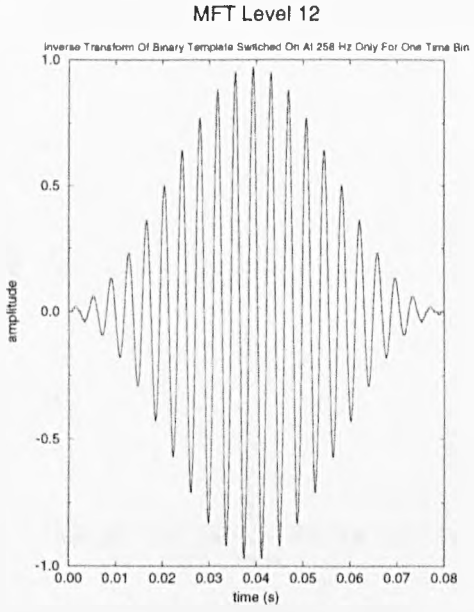

Figure 3.18: The MFT inverse for a simple adaptive filter structure with frequency bin turned on at 258Hz for one time bin at MFT level 12.

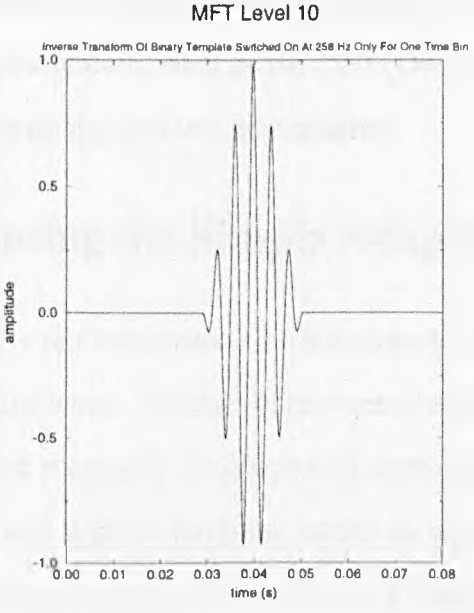

Figure 3.19: The MFT inverse for a simple adaptive filter structure with frequency bin turned on at 258Hz for one time bin at MFT level 10.

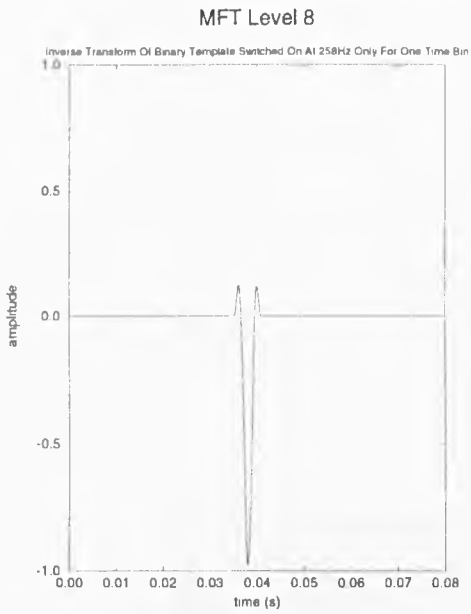

Figure 3.20: The MFT inverse for a simple adaptive filter structure with frequency bin turned on at 258Hz for one time bin at MFT level 8.

corresponding to the constituents of a binary template should look like simple functions. Note that this is very different from the result of choosing a single frequency turned on in a critically sampled representation, such as the DFT. Oversampling allows the use of smooth windows that minimise the artifacts at transients.

3.7 Restoration using the Simple Adaptive Filter

Before using the MFT fully to perform restoration across scale, results are shown for audio restoration done on individual levels. These will then set a base level for restoration using the adaptive filter (3.14), and eventually (in chapter 5) multiresolution restoration.

One way to find how well a given template works as an audio signal restorer is to obtain the template from a clean signal, $s(t)$ and restore a version of the same signal with noise added. This method of templating should give no error due to inaccuracies in time and frequency of features in the template. The MFT of the restored signal is the product

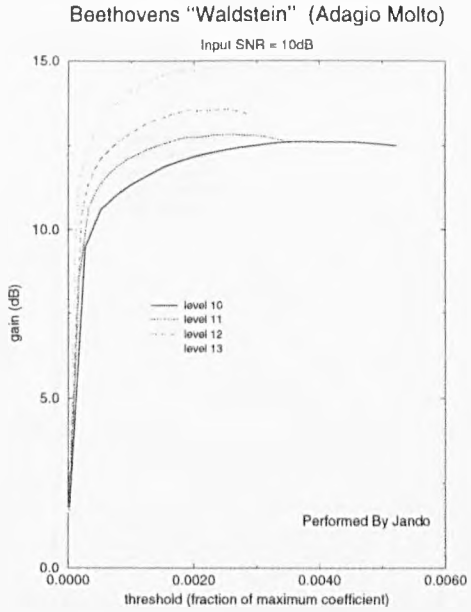

Figure 3.21: The variation of gain in SNR as a function of threshold for various levels with 10dB input SNR on Beethoven's "Waldstein" (Adagio Molto).

of the binary template and the noisy signal in the MFT domain

$$\tilde{X}_{t,f,n} = H_{t,f,n}(s(t)).X_{t,f,n} \tag{3.22}$$

3.7.1 Results and Analysis

The optimum threshold ν is found by minimising the errors in (3.21) due to the absence of signal and presence of noise in the restored signal. An estimate of the noise variance present is calculated for the noisy (target) signal, and the threshold is chosen accordingly.

Figure 3.21 shows how the best threshold varies with level and SNR. There are two main points to be taken from figure 3.21, the first is that the threshold is different for each level and the second that the maximum SNR is different for each level. The first point highlights the fact that the threshold ν is a function of level n and input SNR so $\nu = \nu(\rho, n)$, a fact that will be exploited in chapter 5. The second point, the fact that the best SNR's are different for different levels, illustrates the multiresolution nature of

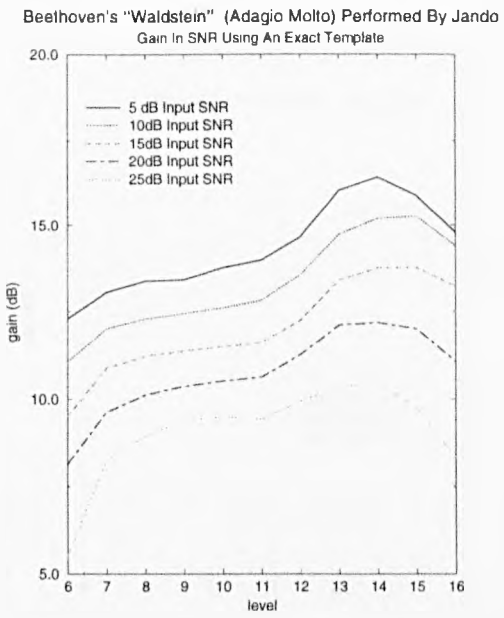

Figure 3.22: Gain as a function of level and input SNR for Beethoven's "Waldstein" (Adagio Molto).

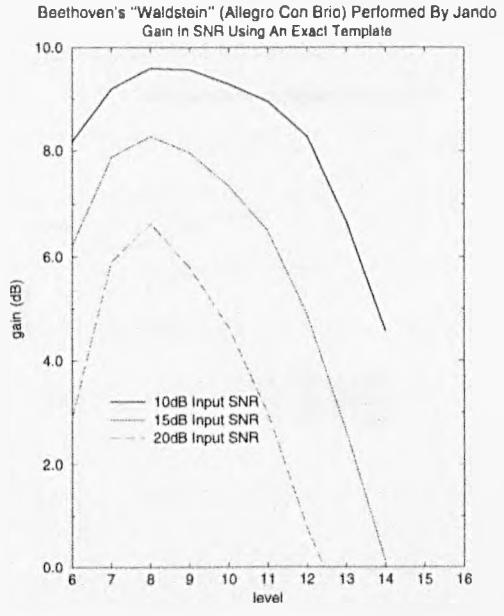

Figure 3.23: Gain as a function of level and input SNR for Beethoven's "Waldstein" (Allegro Con Brio).

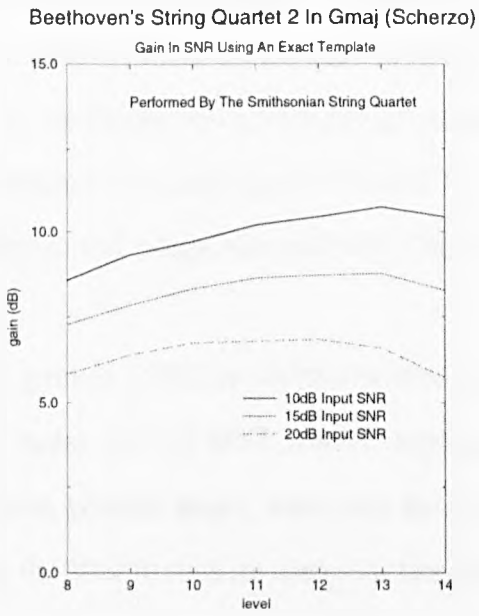

Figure 3.24: Gain as a function of level and input SNR for Beethoven's String Quartet no 2 in Gmaj (Scherzo).

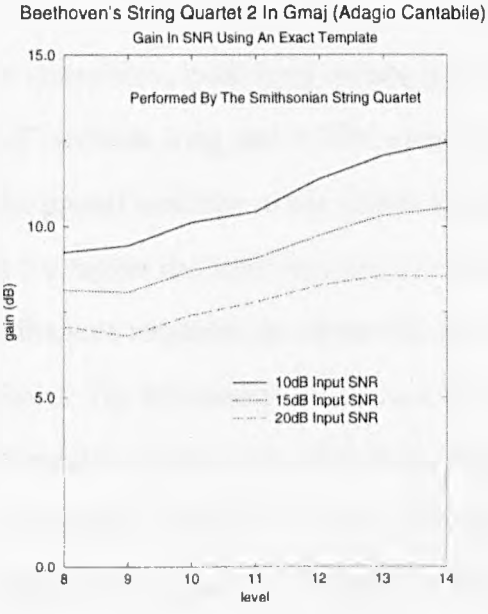

Figure 3.25: Gain as a function of level and input SNR for Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile).

audio signal enhancement, and the main reason that the MFT is used. To investigate the effects of resolution on audio restoration the gain in SNR is plotted as a function of level in figures 3.22 and 3.23 for Beethoven's "Waldstein" Adagio Molto and Allegro Con Brio performed by Jandö, respectively, and figures 3.24 and 3.25 show the results from the String Quartet in Gmaj Scherzo and Adagio Cantabile by The Smithsonian String Quartet, respectively.

Figure 3.22 shows the gain in SNR for various amounts of input noise, from 5dB to 25dB of white additive noise, for 10 MFT levels, ranging from level 6 to level 16. All the curves have the same general shape, with only minor differences. This can be interpreted as showing that the filter in each instance is retaining roughly the same number of coefficients, presumably these are the same coefficients for each level, which would mean that the filter was acting consistently across levels. The explanation for this is simply that the filter is optimised in each instance, by retaining only those coefficients that have significant energy. Obviously the Adagio Molto section of "Waldstein" is a slow section, so the best level for restoration, in all levels of noise, is a high one, level 14, where each bin in the MFT is 0.37 seconds long and 2.7Hz wide. It is natural in steady-state portions of signal, where the partial structure of the signal resembles a straight line on the time-frequency plane, that the higher the level, the more restoration there will be, as the proportion of the MFT coefficients required for reconstruction of steady-state portions is reduced by increasing the level. The limitations are that a level may be too high to capture the transients caused by the onsets of notes, or it may even be longer than the duration of some notes, missing them altogether. This is why the gain decays beyond level 14. For the faster section of the "Waldstein", from the Allegro Con Brio, it can be seen that the best level is significantly lower than level 14, at level 8. This is because the piano is being played very quickly and loudly, so the transients are prevalent and the notes are very close

in time. The problem with there being a high concentration of rapid notes and therefore transients is that the peak restoration gain is lower than for a slower piece where a much higher level can be used. As a result, the peak gain in SNR for the Allegro Con Brio is only 10dB at 10dB input SNR, compared with 15dB for the Adagio Molto.

This pattern is repeated in the two string pieces with one fast, the Scherzo, and one slow, the Adagio Cantabile. Since these are performed by bowed instruments, whose attack is not as fast and whose steady state not as sinusoidal as the piano's, due to the vibrato, the difference in the gain due to restoration between the two is not as noticeable. The gain for the Adagio Cantabile is a peak at level 14 and the Allegro Con Brio at level 13. The difference in gain between the two is only 2dB, much less than that for the piano pieces.

The main conclusions to be drawn from these results are first that the best MFT level or scale is a function of the particular piece of music being restored. It depends on whether it is fast or slow, or whether the instrument's attack is short or long, and whether the instrument's steady-state is sinusoidal like the piano. The second point is that the gain in SNR also depends on the amount of noise present in the signal, with the gain being around 10dB for the string quartet pieces and between 10dB and 15dB for the piano pieces.

It can be concluded that the best gain for lowpass filtering is significantly worse (3dB) than for templating increasing to in excess of 6dB for low input SNR. It is also worth commenting that lowpass filtering an audio signal reduces its "brightness", making it a perceptually ineffective solution. Whereas the adaptive filtering technique relies on the masking effect of allowing areas of large SNR through, at the same time as areas with low SNR's to be included in the restored signal. Finally, it has been shown also that the template performs better than the Wiener filter. From figure 3.10 the best gain for the Wiener filter is 4dB less than the best value for the template filter and, similarly, the largest

gain in figure 3.11 is worse by 5dB than the best for the template. Wiener filtering, when implemented in the MFT domain, has nothing to gain from using a low time resolution, since the only way to increase the accuracy of the filter's spectral estimate is to increase the level, and thus increase the frequency resolution.

3.8 Selecting the global threshold ν analytically

In (3.21) the error present in a thresholded MFT level is described as the sum of the errors due to the noise retained and the signal lost. This can be modelled statistically if the signal and noise energies are given probability densities $p_s(v)$ and $p_n(v)$ respectively. The energy threshold on level k is given by

$$\tau = (\nu\Theta_k)^2 \tag{3.23}$$

then the mean square errors in the restored signal can be defined as

$$\epsilon_s(\tau) = \int_0^{\tau} v p_s(v) dv \tag{3.24}$$

$$\epsilon_n(\tau) = \int_{\tau}^{\infty} v p_n(v) dv \tag{3.25}$$

The threshold τ should obviously be chosen to minimise the total error $\epsilon(\tau) = \epsilon_s(\tau) + \epsilon_n(\tau)$, i.e.

$$\frac{\partial \epsilon(\tau)}{\partial \tau} = 0 \tag{3.26}$$

Figures 3.26 and 3.27 show the signal as a histogram and its approximation by exponential and Cauchy densities over a finite range of energies. Clearly the Cauchy density gives a reasonable fit to the data, but this has the unfortunate consequence of modelling the signal density by a process with unbounded moments. To overcome this problem, the density is modelled over a limited range of values, from 0 to 10 times the noise variance and to get

a reliable estimate of the density from the data, the histogram bin width is chosen to be 1/10th of the noise variance.

To estimate the parameters of the density, it is necessary to get a reliable estimate of the signal and noise variances from the noisy data. The target signal's SNR can be estimated by exploiting the fact that nearly all the uncorrupted signal energy present in a noisy target signal is contained between the frequencies 0 and F_s, where it is also assumed that there is a proportion, $(\frac{F_s}{F_{Nyq}-F_s})$ of the total noise energy present. For most musical applications F_s is about 3kHz. Furthermore, in the frequency range F_s to F_{Nyq}, it is assumed that the only energy is the remaining noise energy. This is enough information to ascertain the noise energy present in a given noisy signal. Thus if we define

$$\Sigma_1 = \sum_{f=0}^{F_s} |X_{t,f,n}|^2 \tag{3.27}$$

$$\Sigma_2 = \sum_{f=F_s}^{F_{Nyq}} |X_{t,f,n}|^2 \tag{3.28}$$

Σ_1 is a combination of signal and a fraction, $\frac{F_s}{F_{Nyq}-F_s}$, of the noise energy for time bin t and Σ_2 is a proportion $(1 - \frac{F_s}{F_{Nyq}})$ of the noise energy present. The amount of noise and signal energy present for time bin t, can be defined in terms of (3.27) and (3.28) as,

$$\Sigma_{\mathrm{S}} = \Sigma_1 - \frac{F_s}{F_{Nyq} - F_s}\Sigma_2 \tag{3.29}$$

and

$$\Sigma_{\mathrm{n}} = \Sigma_1 - \Sigma_{\mathrm{S}} \tag{3.30}$$

Using these, an estimated signal to noise ratio ρ' can be calculated using

$$\rho' = 10\log_{10}\left(\frac{\sum_{t=0}^{T}\Sigma_{\mathrm{S}}}{\sum_{t=0}^{T}\Sigma_{\mathrm{n}}}\right) \tag{3.31}$$

using (3.29) and (3.30). Note that (3.29) and (3.30) are both normalised so that the signal energy estimate does not swamp the noise energy estimate when F_s is high. Values of the

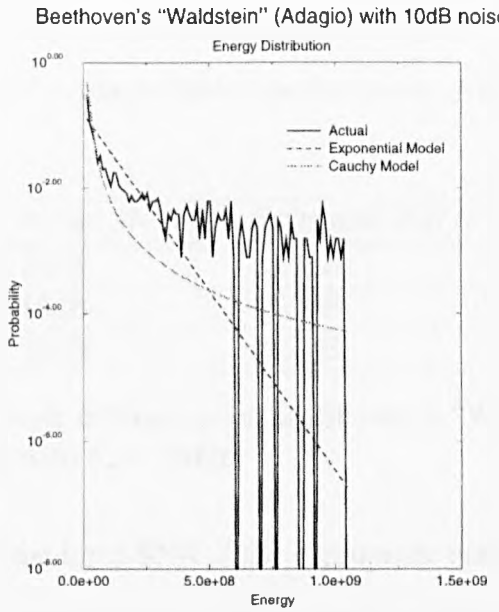

Figure 3.26: The energy distribution for Beethoven's "Waldstein" (Adagio Molto) 10dB away from the mean energy, modelled by both the exponential and Cauchy distributions on a log-linear scale.

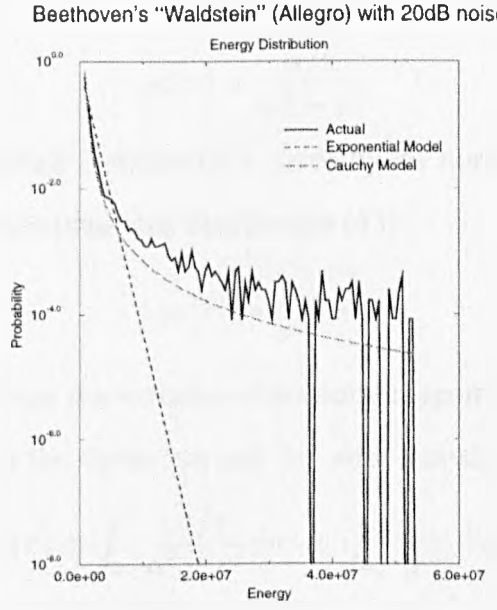

Figure 3.27: The energy distribution for Beethoven's "Waldstein" (Allegro con Brio) 20dB away from the mean energy, modelled by both the exponential and Cauchy distributions on a log-linear scale.

estimated signal to noise ratio ρ' are shown along with the actual signal to noise ratio (ρ) for Beethoven's "Waldstein" (Adagio Molto) performed by Ashkenazy in table 3.1 where $F_s = 3kHz$.

Actual SNR (ρ)	*Estimated SNR (ρ')*
10dB	10.7dB
15dB	15.9dB
20dB	20.7dB

Table 3.1: Results for noise estimation on Beethoven's "Waldstein" (Adagio Molto) performed by Ashkenazy, with F_s = 3kHz.

Using the estimate of the input SNR ρ' the appropriate energy distribution histogram is plotted alongside the approximate exponential and Cauchy distributions. This is shown in figures 3.26 and 3.27 for Beethoven's "Waldstein" Adagio and Allegro respectively. As can be seen the Cauchy distribution, whilst not perfect, matches the energy distribution most closely. This means that the signal's probability distribution is well approximated by

$$p_s(v) = \frac{\alpha/\pi}{\alpha^2 + v^2} \tag{3.32}$$

and since the noise magnitude distribution is given by the normal distribution, its energy can be modelled using the exponential distribution [43]

$$p_n(v) = \frac{1}{\sigma} e^{-\frac{v}{\sigma}} \tag{3.33}$$

where σ here is equal to twice the variance of the noise magnitude distribution. Returning to the definitions given for the signal lost and the noise included

$$\epsilon(\tau) = \int_0^\tau \frac{v\alpha/\pi}{\alpha^2 + v^2} dv + \int_\tau^\infty \frac{v}{\sigma} e^{-\frac{v}{\sigma}} dv \tag{3.34}$$

which when differentiated with respect to τ gives

$$\frac{\partial \epsilon(\tau)}{\partial \tau} = \frac{\alpha\tau}{\pi(\alpha^2 + \tau^2)} - \frac{e^{\frac{-\tau}{\sigma}}\tau}{\sigma} \tag{3.35}$$

hence

$$\tau = \sqrt{\alpha\left(\frac{\sigma}{\pi}e^{\frac{\tau}{\sigma}} - \alpha\right)} \tag{3.36}$$

which can be solved using iterative substitution giving a value for τ [63]. This process was undertaken for Beethoven's "Waldstein" Adagio by Jandó with 10dB of noise. Figure 3.28 shows results for restoration using these values of τ alongside results for restoration using the empirically chosen threshold from section 3.7. As can be seen quite clearly,

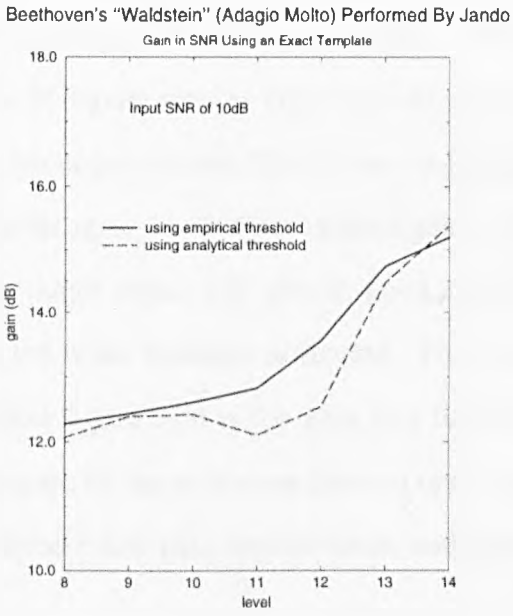

Figure 3.28: Gain as a function of level and input SNR for Beethoven's "Waldstein" (Adagio Molto) using both analytical and empirical choices of threshold.

there is at most a 1dB decrease in performance and, for level 14, a slight improvement in performance. This shows that this method is effective in estimating values for the global threshold ν analytically.

3.9 Comparing Filters Derived from Target and Prototype Signals

To illustrate the value of using a prototype signal, filters derived from the target signal can be compared with those discussed previously in section 3.7.1 — filters derived from the prototype signal. This was done using the same method as before. In figure 3.29 the gain in SNR is plotted against input SNR for the best single level in the range of levels from level 8 to level 14 for Beethoven's "Waldstein" by Jandö. The prototype derived signal gives a straight line, with a 5dB gain over the target derived restored signal for 10dB input SNR. The gain curve for the target derived filter is less co-linear, its gradient decreasing with gain. This is because the noise level of the target signal will fall off and, eventually, the filter derived from the target signal will give as good a performance as that derived by the prototype filter as the noise variance decreases. This can be seen by comparing figures 3.30 and 3.22, where figure 3.30 is the gain as a function of level for the target derived filter and 3.22 the gain for the prototype derived one. As can be seen their shape is similar, in that they both have low gain for low levels and, both have high gain around levels 13 and 14.

3.10 Multiresolution Templating

In section 3.7, audio restoration was performed on four pieces of music for various MFT levels and the gain plotted as a function of input signal to noise ratio for each. It was demonstrated that some levels are better for some pieces of music than others, for example for the "Waldstein" Adagio Molto, the best level was level 14, but for the Allegro Con Brio it was level 8. As was explained, this is due to the relative amounts of steady-state and onsets present in each piece. It follows that if a piece could be segmented into steady-state

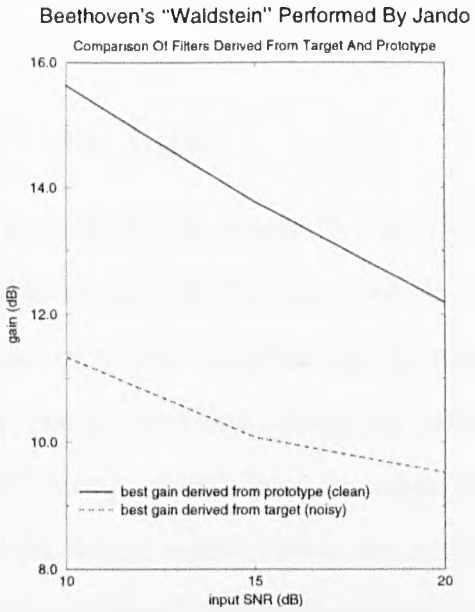

Figure 3.29: A Comparison of gain for filters derived from the prototype and target signals.

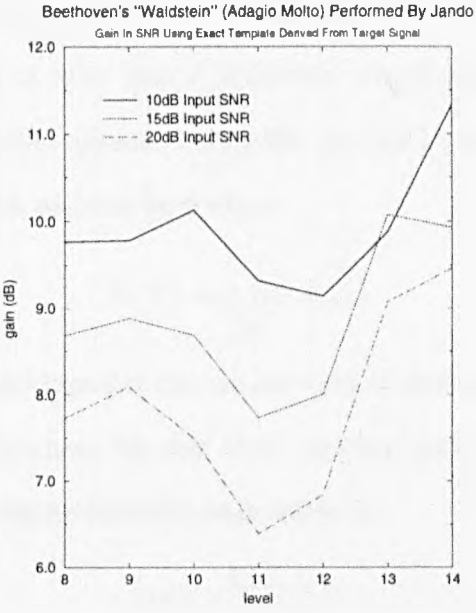

Figure 3.30: Gain as a function of level and input SNR for Beethoven's "Waldstein" (Adagio Molto) using a target derived filter.

and onset periods, and the best level for each segment could be chosen, better results might be obtained.

3.10.1 Segmentation of the Signal

It is becoming accepted that music has an inherently multiresolution structure [45] [25]. For example, it is impossible to ascertain the start time and frequency of a note to an arbitrary accuracy due to the uncertainty principle (see for example [65] [33] [19]). The results in section 3.7 show that the best restorations for various pieces of music are at different resolutions or MFT levels. A high level should enhance continuous sinusoidal signals better than a low level, which would, conversely, enhance transient signals best. This idea can be compared to ideas of "best fit" compression. In [12], for example, compression is performed on a signal by adding signals from different scales and getting the "best fit", at different intervals in the time domain, for the compressed signal by minimising the Shannon entropy of the time-frequency energy distribution.

Suppose the time axis is split into I segments where each segment is an interval indexed with i, so that the ith segment will be the interval $[t_i, t_{i+1}]$. The time axis is then the union of these segments, and can be written

$$[0, T] = \bigcup_{i=0}^{I-1} [t_i, t_{i+1}] \tag{3.37}$$

The best level of restored signal is chosen for each of these intervals by finding which level gives the best enhancement for that time segment and the final reconstruction is simply the sum of the reconstructions for each segment

$$\tilde{x}(t) = \sum_{i=0}^{I-1} \tilde{x}_i(t) \tag{3.38}$$

The time domain signal is used rather than the MFT for a number of reasons. If the signals were combined in the MFT domain then the segmentation level would necessarily

be equal to the highest level in the MFT used for the enhancement. Generally the length of an onset will be much less than the sampling interval of the highest level used, as the highest level is more suited to steady states. Moreover, inversion of a signal defined in multiple levels of the MFT is not straightforward: artifacts may be introduced around coefficients whose neighbours are represented on a different level [9]. It follows that combining the results in the time domain is the most effective solution.

In order to do this, the onsets of notes must be identified. The energy profile $E(t,n) = \sum_f |X_{t,f,n}|^2$ is used for this purpose. Although not every peak in the energy profile is an onset, in this application it is more important not to miss an onset than to avoid false alarms. The problem of onset detection is thus equivalent to peak detection in the energy profile of MFT level $X_{t,f,n}$. To find the peaks, $E(t,n)$ is smoothed with a first order recursive filter, to give

$$E'(t,n) = \xi E(t,n) + (1-\xi)E'(t-1,n) \tag{3.39}$$

This is then subtracted from the original, creating an energy difference profile, $D(t,n)$ which is non-zero only at those parts of the signal that correspond to positive transients.

$$D(t,n) = \begin{cases} E(t,n) - E'(t,n) & \text{if } E(t,n) \geq E'(t,n) \\ 0 & \text{else} \end{cases} \tag{3.40}$$

Examples of $E(t,n)$, $E'(t,n)$ and $D(t,n)$ are shown in figures 3.31, 3.32 and 3.33 respectively.

The set of onsets $\{O_i\}$ can be defined in terms of the difference $D(t,n)$. If the intervals where $D(t,n)$ is greater than zero, $D(t,n) > 0$, are indexed with i, then interval $\Delta_i(n)$ has definition

$$\Delta_i(n) = [t_{i_s}, t_{i_e}] : \text{ where } t_{i_s} \leq t \leq t_{i_e}\,,\, t_{i_s} > t_{(i-1)_e} \text{ and } D(t,n) > 0 \text{ for } t \in \Delta_i(n) \tag{3.41}$$

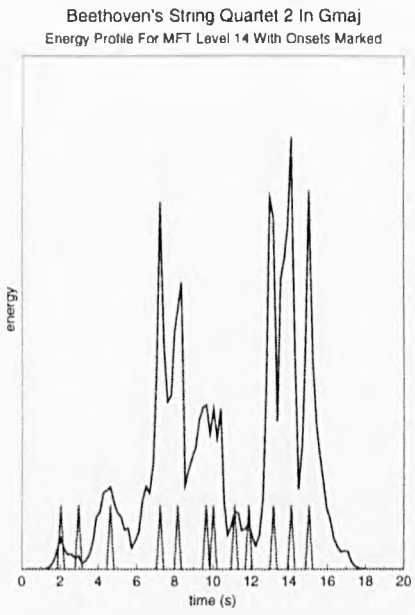

Figure 3.31: Energy profile at MFT level 14 of Beethoven's String Quartet no 2 in Gmaj with onsets marked.

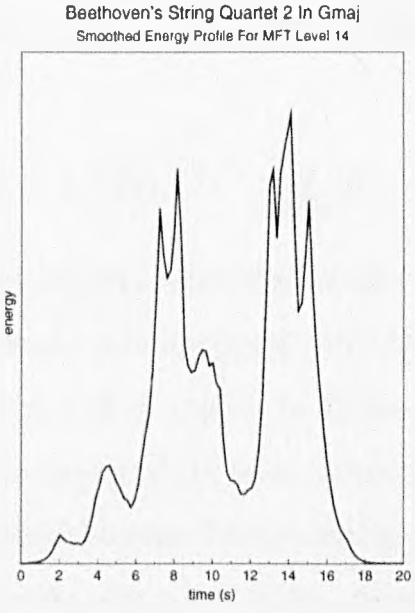

Figure 3.32: Output of the first order recursive filter where $\xi = 0.6$, with energy profile at MFT level 14 of Beethoven's String Quartet 2 in Gmaj as input.

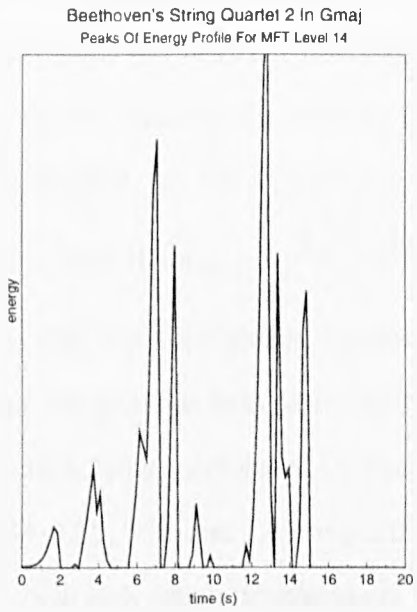

Figure 3.33: Positive transients found for energy profile at MFT level 14 of Beethoven's String Quartet no 2 in Gmaj.

where t_{i_s} and t_{i_e} denote the start and end of time interval t_i respectively.

The onsets can be found simply by maximising the difference energy profile within each of these intervals

$$O_i = t \text{ if } D(t,n) = \max_{s \in \Delta_i(n)} D(s,n) \tag{3.42}$$

The above definition of onsets, or peaks, allows the the size of the onset to vary according to n. A value of n must be chosen to be sufficiently high to allow a large onset area, and sufficiently low that there are at least as many peaks as onsets. The decision as to which level to use is dependent on the music. If the music is fast, then a high time resolution is necessary. If the music is slow, then a low level is used so that secondary bumps on the peaks in the energy profile are not detected as onsets. Normally n is chosen to be in the range $11 < n < 14$.

Once the time axis has been segmented, the decision as to which level is to be chosen

in each segment must be made. By comparing the enhanced signal $\hat{x}(t,n)$ against the ideal restored signal $s(t)$ — which is known *a priori* at this stage — the best restored signal can be chosen by minimising (3.21). This can be done by using the SNR, and choosing the level n_{optimum} to give maximum SNR, i.e. for segment i with $t \in [t_i, t_{i+1}]$

$$\rho_i(n_{\text{optimum}}) = \max_n 10\log_{10} \frac{\sum_{t=t_i}^{t_{i+1}} |s(t)|^2}{\sum_{t=t_i}^{t_{i+1}} |s(t) - \hat{x}(t,n)|^2} \tag{3.43}$$

For all four test pieces, the gain has been plotted against input SNR, for Beethoven's "Waldstein" Adagio Molto and Allegro Con Brio performed by Jandó, and String Quartet 2 in Gmaj Adagio Cantabile and Scherzo performed by The Smithsonian String Quartet. These are shown in figures 3.34, 3.35, 3.36 and 3.37 respectively, with the best "per level" templating result included to show how much improvement there is by combining across resolutions. As can be seen, there is in most cases a significant improvement in the SNR by combining across level, or scale. The one case where the increase in gain is negligible is for the very slow "Waldstein" (Adagio Molto) (3.34). This is most likely because the multiresolution method only gives an improvement if there is a mixture of steady state and onset. As there are only a few notes being played, and they are played very slowly, the amount of transient, as a proportion of the total time is negligible. It is also worth noting that the single level gain is very high already ($>$ 15dB) and so any further improvement may be hard to find. For the two string pieces, there is a marked improvement in both the Adagio Cantabile and the fast Scherzo pieces. The Scherzo at 20dB input SNR is improved by more than 1dB by combining levels. The surprise is the improvement in the Adagio Cantabile, which shows 1dB improvement also. There are two things to be considered here: first, the "Waldstein" Adagio Cantabile is not as slow as the Adagio Molto; second, the presence of vibrato in the strings could imply more variation in the optimum level.

It can be concluded from these results that audio restoration is a multiresolution

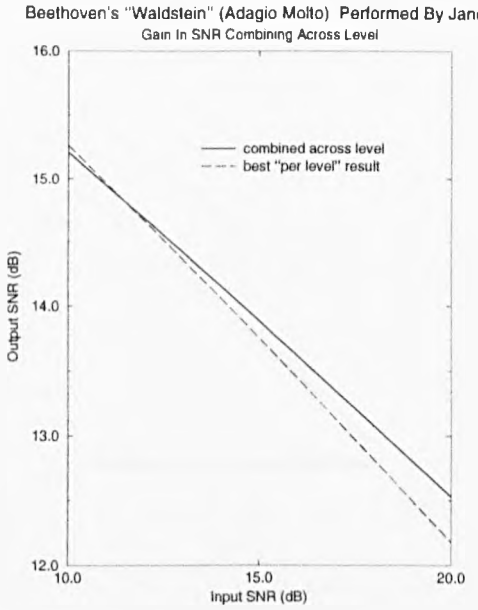

Figure 3.34: Gain against input SNR for Beethoven's "Waldstein" (Adagio Molto) performed by Jandö combining across level.

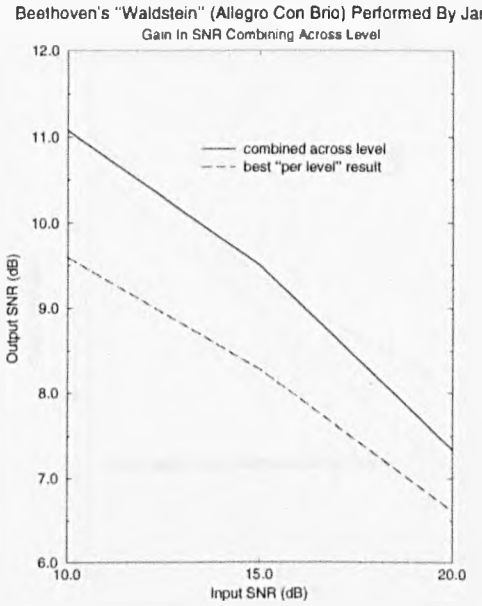

Figure 3.35: Gain against input SNR for Beethoven's "Waldstein" (Allegro Con Brio) performed by Jandö combining across level.

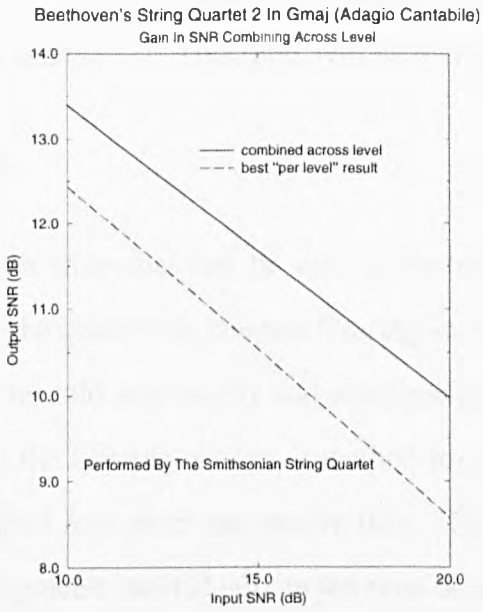

Figure 3.36: Gain against input SNR for Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile) performed by The Smithsonian String Quartet combining across level.

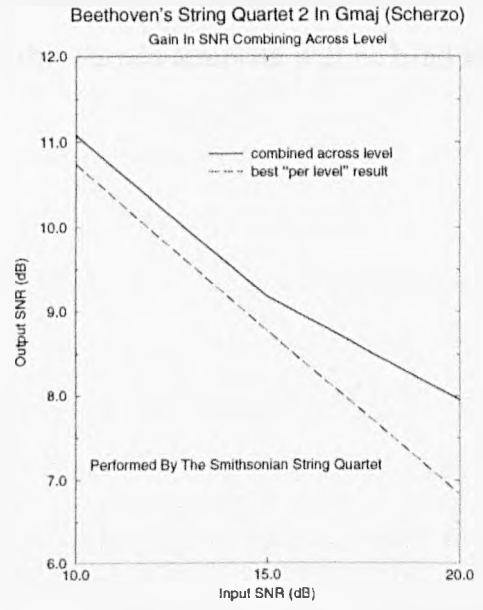

Figure 3.37: Gain against input SNR for Beethoven's String Quartet no 2 in Gmaj (Scherzo) performed by The Smithsonian String Quartet combining across level.

problem and that by combining signals from different scales, an even better enhancement can be found than those in section 3.7. This topic will be revisited in chapter 5.

3.11 Conclusion

In this chapter, an adaptive filter that can be used to restore audio signals has been described. It was shown to be better than lowpass filtering and Wiener filtering. Methods for choosing the filter's threshold empirically and analytically were described. In order to vary the scale on which the filtering is done, a method for segmenting the signal was described that split the signal into onset and steady state. The optimum level was then chosen for each of these segments individually in the time domain using (3.43), giving a signal restored using more than one scale or resolution. The implementation of the filter using the MFT was shown to involve a simple "templating" operation, based on detection of the signal in the MFT representation.

In the next chapter a method for warping the template to fit a target signal will be described and in chapter 5 the warped template will be used in much the same way as it has been in this chapter.

Chapter 4

A Warping Algorithm for Enhancement

4.1 Introduction

In order to obtain a good enhancement of a noisy signal there must be some way of deciding what is signal and what is noise. In Vaseghi's work [59], for example, this was done using two noisy recordings of the same performance. It was assumed that the noise in the two recordings would be uncorrelated and that the signals would be correlated, after a small amount of time warping. Other methods involve deciding what is noise and what is signal using empirical methods, [6] for example. In the present work, the aim is to enhance a noisy recording by discarding the noise without altering the signal significantly. This is done by finding a modern, clean recording of the piece to be enhanced and using the event detector (cf chapter 3) to give a template of the signal in the MFT domain. How this template can be used to restore a noisy version of the clean recording is discussed in chapter 3. This chapter aims to explain how, using warping techniques in time and frequency, this template can be made to fit the noisy recording for enhancement.

This chapter defines warping functions, explains a method for approximating warping functions, gives the background information necessary for the warping of the simple adaptive filter, gives a time warping algorithm for that filter and also explains a frequency

warping algorithm. It concludes with results for warping of various signals.

4.2 Warping Audio Signals

Audio signals are, or have been, warped for many purposes. In the early days of Elektronische Musik, Stockhausen realised the connection between the timbre and duration of a musical note by varying dramatically the speeds at which a note was played [32]. This technique was most famously employed in his piece "Gesang Der Jünglinge" to give the listener the effect of a note "dropping from the sky". In Paris, Schaeffer was also varying the speeds of audio signals, although, in keeping with the theory of Musique Concrète, these sounds were everyday sounds, like steam trains in "Études Aux Chemins De Fer". The time signal was warped in order to change the way the listener perceived it: from an everyday sound to what Schaeffer called an "objet sonore" or abstract musical object [42]. In both of the above examples, the warping of audio signals was experimentally aesthetic. There was no target to warp towards and the signals warped were generally analogue.

A more scientific approach to warping digital audio signals is taken in the field of speech recognition [57] [23] [47] [40]. There are many types of speech recognition systems and warping is used in all of them. The basic approach is to construct a set of target or template signals that any incoming signal can be referenced to, to find which it matches best. Each of these target signals is a recording of a word spoken by a single speaker in speaker-dependent speech recognition systems, or an amalgam of recordings of a word spoken by more than one speaker in speaker-independent systems. In both systems the input signal is compared in turn with each of the target signals. At this stage a "best warp" is found, so that the current input signal can be warped to fit the target signal as well as possible. The warping is necessary because the signals will probably be of different lengths since humans generally do not speak at a consistent speed. That is, no

speaker can be expected to speak any word in such a way as to always reproduce exactly the same time signal. For speaker-independent systems it is easy to see that reason for having an amalgam of speakers for every word is that the idiosyncrasies of speech can be "averaged out" as much as possible. The warping must be done under the constraint of certain rules, otherwise there would be an infinite number of changes that could be made to the signal. These rules are: that the warped signal must be the same length as the target signal in time, to within a tolerance; there must be some features that can be identified and used as fixed points where the gradient of the warp can be altered. In speech recognition these points are often consonants, chosen because there is very little change in the speed of speech in consonants, as opposed to spoken vowels, where the speed varies as a matter of course between speakers. Once the "best warp" is found, the goodness of fit values for each warping are compared and the target signal with the best fit is chosen. Figure 4.1 shows a schematic description of these processes. It can be seen from this diagram that the problem falls into three categories: matching, warping and the estimation of how well an input word matches each candidate word.

Warping a digital audio signal is a non-trivial exercise [3]. In order to perform an operation on the time-frequency plane of an audio signal $x(t)$, a time-frequency representation must be used. In this work the MFT is used, due to its generality (cf chapter 2). Warping the magnitude of such a time-frequency representation can be done by using a linear interpolation algorithm, that is for MFT coefficient $X_{t,f,n}$, where

$$|X_{t,f,n}| = A_{t,f,n} \tag{4.1}$$

$$\arg(X_{t,f,n}) = \phi_{t,f,n} \tag{4.2}$$

and warping coefficient ϵ gives

$$A_{\epsilon t_1,f,n} = A_{t_1,f,n} + (\epsilon t_1 - t_1)\frac{A_{t_2,f,n} - A_{t_1,f,n}}{t_2 - t_1} \text{ where } t_1 \leq \epsilon t_1 \leq t_2 \tag{4.3}$$

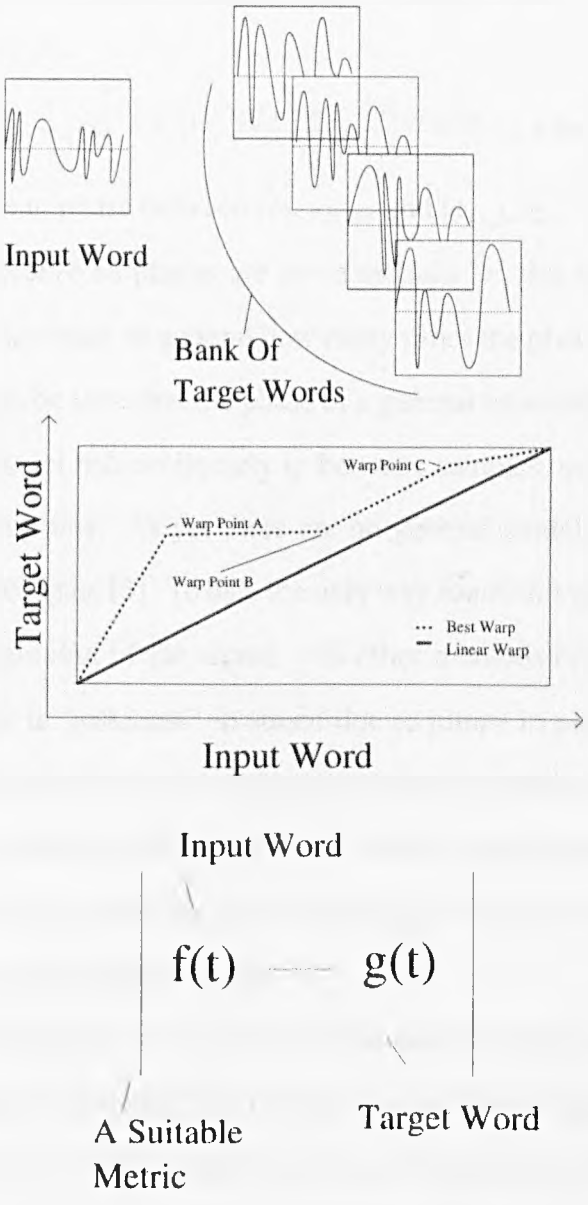

Figure 4.1: How speech recognition works using warping.

This method assumes that the behaviour of the magnitude between samples is linear, which is not necessarily a satisfactory model. Let us now consider the interpolation of the phase using a similar method, with the only difference being that the phase will be represented modulo 2π [3].

$$|\phi_{\epsilon t_1,f,n}|_{2\pi} = |\phi_{t_1,f,n}|_{2\pi} + (\epsilon t_1 - t_1)|\frac{|\phi_{t_2,f,n}|_{2\pi} - |\phi_{t_1,f,n}|_{2\pi}}{t_2 - t_1}|_{2\pi} \text{ where } t_1 \leq \epsilon t_1 \leq t_2 \quad (4.4)$$

Consider the difference in phase between $|\phi_{t_2,f,n}|_{2\pi}$ and $|\phi_{t_1,f,n}|_{2\pi}$. The difference may be greater than 2π, but because all phases are given modulo 2π, this would be unknown. In fact there is no way of knowing in general how many times the phase has looped round 2π between samples. It can be seen that the phase of a general time-frequency representation of an audio signal does not behave linearly in between samples, and therefore that linear interpolation is inappropriate. As yet there are no general models for interpolating the phase of a warped audio signal [3]. To date the only way round this problem is to "unwrap" the phase from the beginning of the signal. All other methods of phase warping create artifacts. One example is "jerkiness" in sound due to jumps in phase between samples; another is a "fan" type sound due to warping the phase by guessing how much the phase should be warped by multiplying the phase by the warping factor and taking it modulo 2π. It can be concluded that in general this type of warping is costly in terms of computation, and may not give totally satisfactory results.

The technique of warping one signal to fit another has been used previously in the field of audio restoration by Vaseghi [58] [59], for sound signals with small differences in time. The approach was to build in a time warping component into a Wiener filter that was used to estimate the power spectra of the noise by looking at the noise in the two signals, both of which should be roughly equally degraded. Using this estimate, the noise was then subtracted from the signal, thereby restoring it. This warping is, however, extremely limited. The warping only works if the two signals are aligned within the duration of

the filter, which was only 20ms long. This means that any warping that is done alters the time domain of the signal by less than 20ms at any one time. Compared to the work done in this chapter, the amount of warping done by Vaseghi was minimal. One method discussed by Vaseghi was to add the time aligned signals to form a hybrid signal. The problem with this method is one of integrity. Even though superficially it may appear to be effective since the sound signals (being correlated) add coherently and the noise signals (being uncorrelated) do not, small errors in the phase interpretation are likely to reduce the coherency between the signal components.

4.3 Describing a Warping Function

A warping function is a simple mapping from one signal to another. Generally the type of warping considered in time signals is time warping, though warping in frequency may be necessary. This is a simple extension of the algorithm used for time warping using Fourier techniques. The easiest way to think of warping is of a stretch or of a compression. The difficulty is in finding where to stretch or compress, and by how much.

For two time signals $x(t)$ and $y(t')$, where $x(t)$ is to be warped to $y(t')$, the warp function is a function that maps from the prototype time domain t to the target time domain t', so that the features in the two signals are time-aligned. The main requirement of a warping function is that the inverse warping must exist, hence that it must be a strictly monotonic increasing function. If an inverse did not exist then the prototype could not be derived from the warped prototype: in other words there would be a loss of information after warping. The warp function is defined by

$$t' = \phi(t) \tag{4.5}$$

with conditions

$$\phi(t_1) > \phi(t) \Leftrightarrow t_1 > t \tag{4.6}$$

$$t = \phi^{-1}(\phi(t)) \tag{4.7}$$

The simplest example of such a function would be

$$t' = \phi(t) = t \tag{4.8}$$

where both time domains are already aligned. The warping function can be considered, in general, as a curve. The curve corresponding to (4.8) would be that shown in figure 4.3, with gradient $\pi/4$. For a more complex relationship between the prototype time and target time consider the warp function in figure 4.4 where the gradient varies according to how fast or slow the target signal time moves relative to the prototype signal.

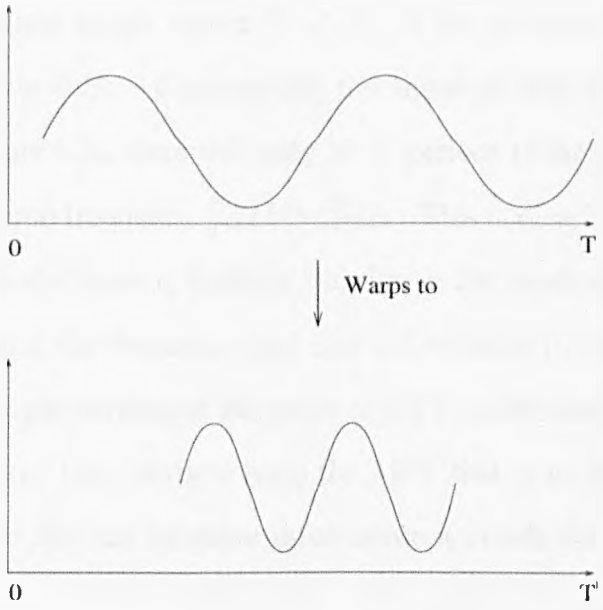

Figure 4.2: Warping a simple wave function.

In the case of two musical signals representing different performances of the same piece, a warping function exists but will be unknown initially. To find it, a series of linear

warps are fitted to the unknown warp function curve and the error between the two are measured using a suitable norm. Recursive bisection [56] is performed on both signals at breakpoints that correspond to similar features and the prototype signal is warped so that these breakpoints coincide. This process continues until the warp function or curve is well approximated by the piecewise linear warp. Finding a good approximation to the warp function can be classified as a curve fitting problem, where the norm is defined to be a measure of 'sameness' or fit between the warped prototype and target signals. This method of piecewise linear binary recursive curve matching is used in many areas, including image processing [44] and numerical analysis [50].

Warping the time signals directly alters their pitch and phase information, since the time signal is typically a sum of harmonic or partial sinusoids. Consider the case of $x(t)$, $0 \leq t < T$, a sinusoid with constant amplitude and pitch, being compressed until $T = T'$ from an original length where $T' < T$. If the periodicity of $x(t)$ is $\frac{T}{N}$ then the frequency of $x(t)$ is $\frac{N}{T}Hz$. Compressing the signal so that its end points occur at $t = T'$ and $t = 0$ (figure 4.2), there will only be N periods in the signal but it will have periodicity $\frac{N}{T'}$ and altered frequency $\frac{N}{T'}Hz > \frac{N}{T}Hz$. This is clearly inappropriate in this application. The method chosen to perform warping in this work uses an MFT structure (section 4.4.3), in which the frequency and time information in the signal can be dealt with separately. Although warping of the phase of MFT coefficients can cause problems, it is the simple adaptive filter derived from the MFT that is to be warped to perform enhancement. Because this has no phase information it avoids the difficulties described above.

Since the warping is to be done using an MFT structure, it would be impractical to use a norm that relied on warping the time signals. The norm chosen is discussed further in section 4.4.3. It is worth commenting that the choice of norm is crucial because this

determines the accuracy with which a given warp function fits the prototype to the target signal. If a norm was chosen that relied on the time signals then the warped prototype could have altered phase and pitch and therefore a low goodness of fit, even if the notes were exactly aligned.

Figure 4.3: A linear zeroth order warping function.

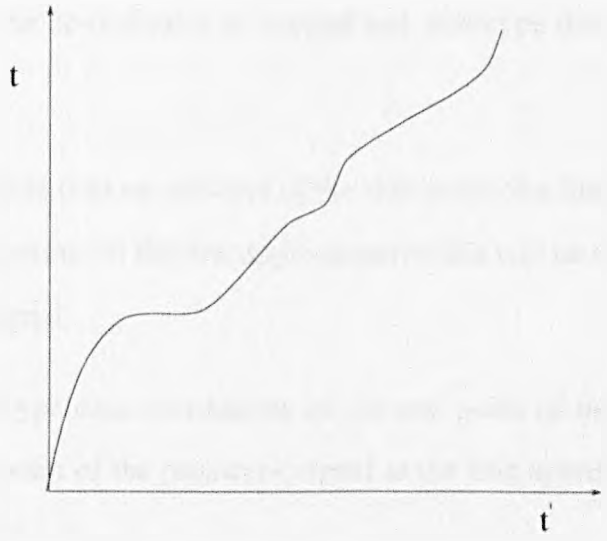

Figure 4.4: A continuous warping function.

4.4 Break Points and Warp Factors

Consider two signals: the prototype signal $x(t), 0 < t < T$ and the target signal $y(t'), 0 < t' < T'$, where $T \neq T'$ in general. The method used for analysis of warp functions as discussed in section 4.3 consists of fitting straight lines to the warp function (see figure 4.4) by recursively bisecting each line and refining each half until the best fit is found. For a zeroth order warping function, the prototype signal is warped to have the same length as the target signal. The amount by which it is warped is known as the warp factor W_f and in this case is

$$W_f = \frac{T}{T'} \tag{4.9}$$

Consider two different performances of the same musical piece. The musicians will not, in general, play with a constant ratio of speeds — they will vary their speed depending on which part of the music they wish to emphasise or heighten dramatically. A higher order approximation is necessary for the best match between two such signals: it is necessary to use more than one straight line to approximate the continuous warping function matching two performances. The co-ordinates in warped and prototype time are explained in the list below.

1. w_s is the prototype time co-ordinate of the start point of a line approximating the warping function. At the first approximation this will be the start point of the prototype signal.

2. w_e is the prototype time co-ordinate of the end point of the line estimate, and is the end point of the prototype signal at the first approximation of the warping function.

3. m_s, the same as w_s but for the target or matched signal.

4. m_e, the same as w_e but for the target or matched signal.

5. m_p is the matched point, a break point chosen in the target signal correponding to some prominent feature.

6. w_p is the warp point, a break point in the prototype signal that corresponds to the same feature as m_p does in the target signal.

w_s,w_e,m_s and m_e are used for the zeroth order warping approximation and in higher orders of warping, the choices for w_p and m_p dictate their value. It is therefore important to ensure that the right choices for w_p and m_p are made. Once these points have been chosen for order $n > 0$ the signal is warped with one warp factor to the left of the warp point and another to the right. These are called the left and right warp factors, W_f^{L} and W_f^{R} respectively. They are defined by the choice of data points with the formulae

$$W_f^{\mathrm{L}} = \frac{m_p - m_s}{w_p - w_s} \tag{4.10}$$

and

$$W_f^{\mathrm{R}} = \frac{m_e - m_p}{w_e - w_p} \tag{4.11}$$

These are applied to the appropriate sub-segment of the prototype signal, so that the matched point m_p and the warp point w_p will occur simultaneously in both signals. A diagramatic explanation of the data points for a target and prototype signal profile is shown in figure 4.5.

This section and the previous section have explained the piecewise recursive linear approximation to the warping function, where the warping function is approximated by finding break points in both signals and then warping the prototype signal so that these breakpoints coincide, and continuing this process by bisecting each half in turn until a good approximation of the warp function is found.

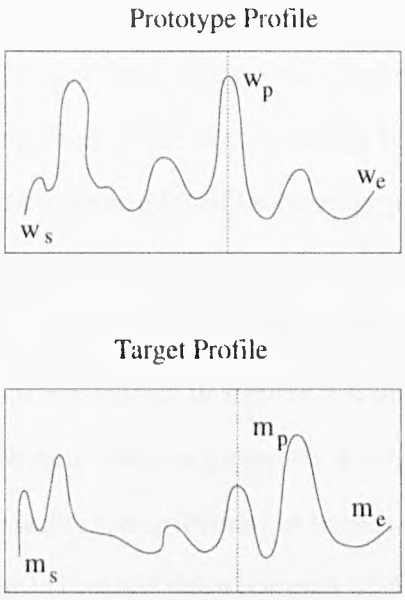

Figure 4.5: An illustration of data points.

4.4.1 The Representation of Prominent Features

The features of interest must occur as a function of time, as it is time warping that is of most importance. Frequency warping is relevant but is much simpler, as performers do not retune their instruments during a movement. Since it is time information that is of crucial importance here, the energy distribution across the frequency axis can just be summed: the energy of the signal at a given time is used as a source of features. The time variation of signal energy is appropriate for matching because most notes cause a noticeable peak in the energy profile, see for example figure 4.8. To identify matching features, the energy function $E(t,n)$ is used where, as in chapter 3,

$$E(t,n) = \sum_{f=0}^{F} |X_{t,f,n}|^2 \tag{4.12}$$

for MFT coefficients $X_{t,f,n}$ at level n, where F is the maximum frequency. The energy function is a time profile of the transformed signal $X_{t,f,n}$, and may sometimes be referred to as such. One advantage in using this structure over one that has both time and frequency

components is that in a signal with noise, noise energy is usually more evenly distributed than the signal energy across the spectrum. Therefore a significant amount of noise energy can be ignored whilst capturing most of the signal energy by bandlimiting to a frequency lower than the Nyquist frequency. Examples of this energy profile structure are given for a performance by Ashkenazy of Beethoven's "Waldstein" (Adagio Molto) in figure 4.8, and for a performance of the same piece by Jandö in figure 4.10. The MFTs from which these two representations are derived are shown in figures 4.6 and 4.7 respectively. It is easy to see that at the onset of each note there is generally a large peak in the energy profile, and that the energy profile contains the information necessary for time warping. Shown in figure 4.9 is the energy profile derived from a signal with additive noise at a power of 0dB relative to the signal, bandlimited at 5kHz, demonstrating that even in extreme levels of noise using the energy profile reveals the onset features clearly.

4.4.2 Warping An MFT Time-Frequency Plane In Time

Consider an MFT level that is to be time warped by a factor W_f. Starting at the lowest frequency sample and warped time sample, the equivalent sample in target time is calculated by dividing the prototype time t' by the warp factor W_f and using linear interpolation to calculate the warped sample value. If the MFT structure is considered as a set of linear arrays in frequency, then $x(t')$ is the warped array and $x(t)$ is the unwarped array.

$$t = \frac{t'}{W_f} \tag{4.13}$$

$$dt = t - \lfloor t \rfloor \tag{4.14}$$

$$x(t') = x(\lfloor t \rfloor) + dt(x(\lceil t \rceil) - x(\lfloor t \rfloor)) \tag{4.15}$$

where "$\lfloor\ \rfloor$" denotes floor and "$\lceil\ \rceil$" denotes ceiling.

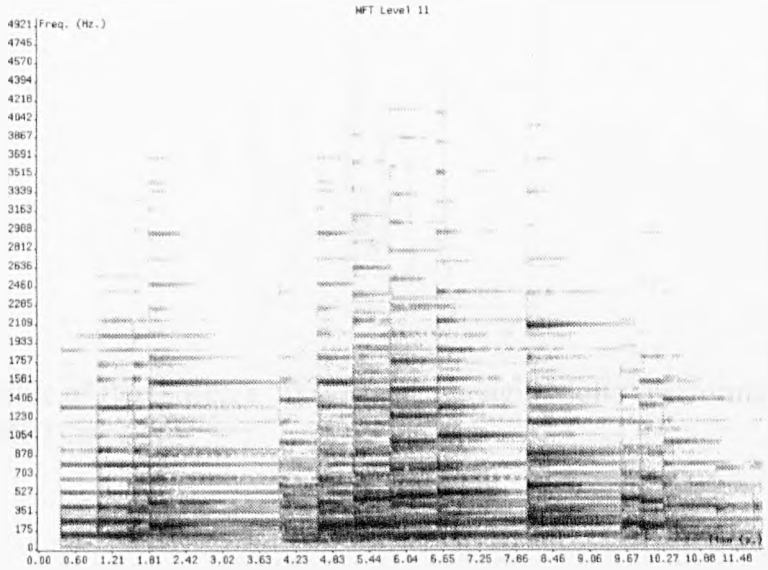

Figure 4.6: MFT level 11 of Beethoven's "Waldstein" (Adagio Molto) performed by Ashkenazy.

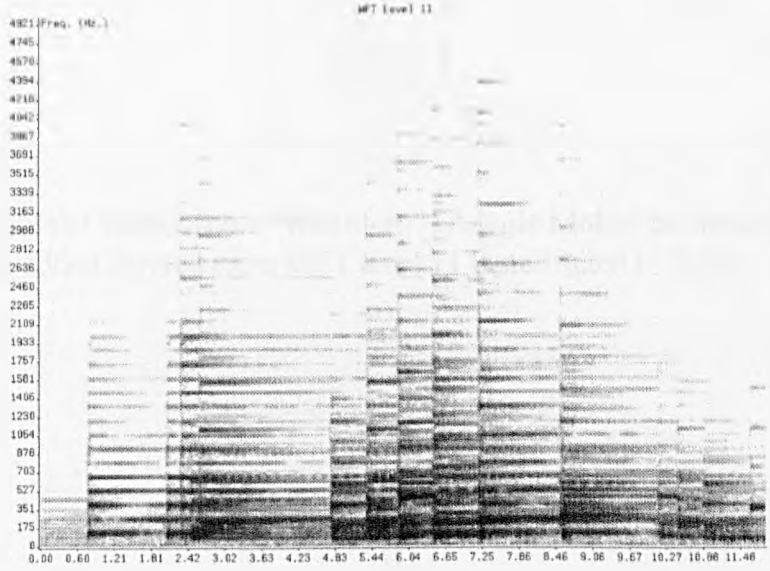

Figure 4.7: MFT level 11 of Beethoven's "Waldstein" (Adagio Molto) performed by Jandó.

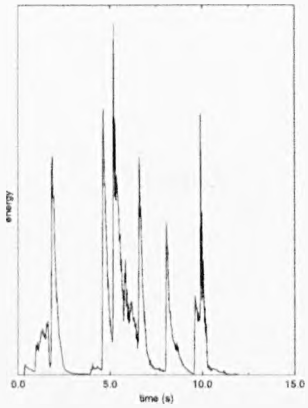

Figure 4.8: Profile of Beethoven's "Waldstein" (Adagio Molto) performed by Ashkenazy derived from MFT level 11.

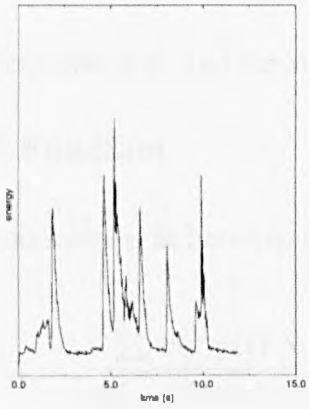

Figure 4.9: Profile of Beethoven's "Waldstein" (Adagio Molto) performed by Ashkenazy with 0dB noise added derived from MFT level 11 bandlimited to 5kHz.

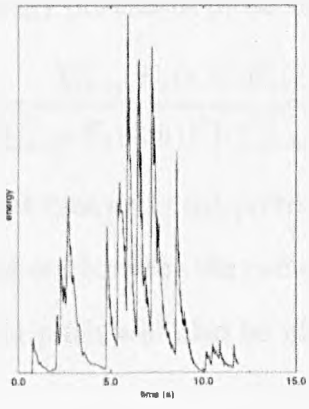

Figure 4.10: Profile of Beethoven's "Waldstein" (Adagio Molto) performed by Jandó derived from MFT level 11.

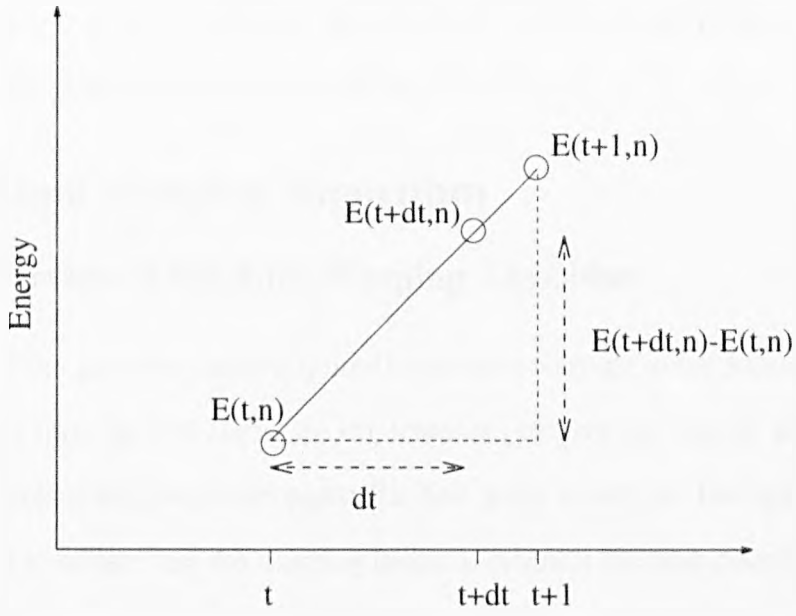

Figure 4.11: Using linear interpolation to find the value of a warped coefficient.

4.4.3 The Goodness of Fit Function

One natural way to measure the goodness of fit between two real signals is the correlation coefficient [26]

$$r_{xy} = \frac{\sum_t x(t)y(t)}{\sqrt{\sum_{t,s} x^2(t)y^2(s)}} \tag{4.16}$$

This is just the normalised dot product of the two signals, and it is simple to see that it has a maximum when the two signals are identical, and minimum when $x(t) = -y(t)$.

In this work, the correlation coefficient is not calculated from the time signals, but instead from the MFT derived energy profiles of those signals.

$$r_{xy}(n) = \frac{\sum_{t=0}^{T}(E_x(t,n)E_y(t',n))}{\sqrt{(\sum_{t=0}^{T}(E_x(t,n))^2)(\sum_{t=0}^{T}(E_y(t',n))^2)}} \tag{4.17}$$

where t' is used to denote the target time and t the prototype time. The use of the energy profile ensures that the phase variations between the two signals do not affect the goodness of fit. Similarly, any variations in pitch will also be eliminated. The warping process

thus requires solution of two problems: the search for suitable break points, m_p, w_p, and measurement of the match using the correlation of (4.17).

4.5 The Time Warping Algorithm

4.5.1 An Overview of the Time Warping Algorithm

Having discussed the necessary tools required to perform warping in the previous sections, our attention now turns to how these are implemented to give the overall scheme. This process can be broken into two main parts: the first is the search for the best data points and the second is to ensure that the warping function found is the best possible.

4.5.2 Searching for Break Points

The best break points are the ones that occur at the most prominent features because they are easy to find and have a higher probability of occurring in all versions of a given piece. The search method both tries to find the best break points, and attempts to ensure their even distribution over the duration of the two samples. At each step an approximately even split is made, constrained by a search tolerance S_t, to limit how far the data point searcher is allowed to look for break points. The choice of tolerance is important, since if it is too small the chances of finding the same data points in both the target and prototype signals are limited, while if it is too large there will be the possibility of bunching.

At a given order of warping, each signal is split into a number of segments. At the start of the warping process there is only one segment, namely the whole of the signal, after one iteration there will be two and so on. The prototype signal is split and a search is made from this point, $w_m = \frac{w_s + w_e}{2}$, within a window $S_t w_m$, until the most prominent feature is found. If there are a number of features that have equal peak energy, then the point that is closest to the mid point w_m is taken in order to prevent bunching of the points.

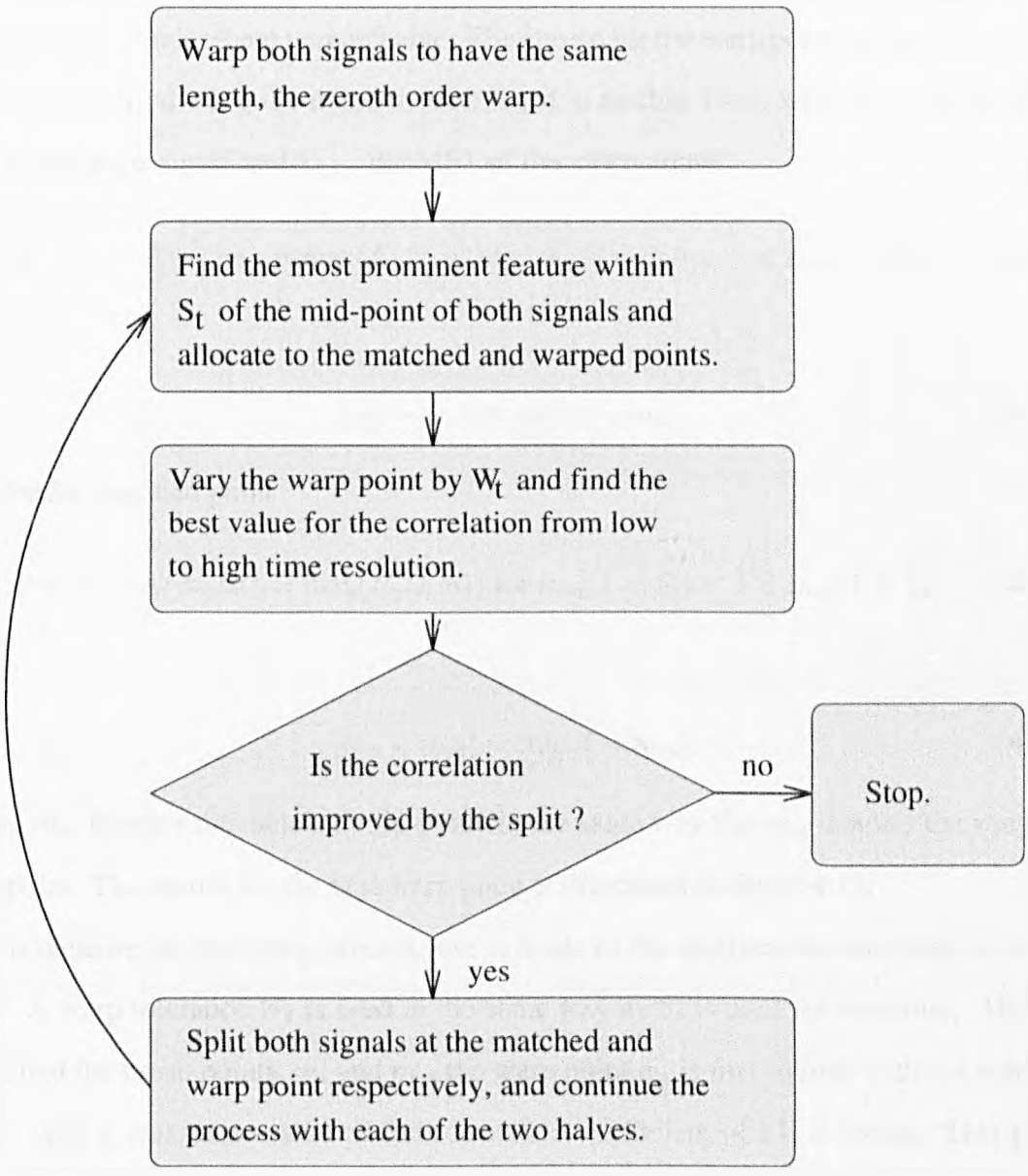

Figure 4.12: A flow chart describing the warping algorithm.

This process is repeated in the target signal, giving points that should correspond to the same feature in each signal, w_p and m_p. This should work if the speed and emphasis of two performances are similar: they are never exactly the same, but they are never so vastly different as to render them unmatchable. The choice for the warp point and matched point can be formalised using the notation introduced in section 4.4.1, with $X_{t,f,n}$ as the MFT of the prototype signal and $Y_{t,f,n}$ the MFT of the target signal.

$$S_p = w_p : E(w_p, n) = \max(E_x(t, n)) \text{ for } w_m(1 - S_t) < t < w_m(1 + S_t) \tag{4.18}$$

and

$$|w_p - w_m| = \min_{t \in S_p} |t - w_m| \tag{4.19}$$

and for the matched point

$$S'_p = m_p : E(m_p, n) = \max(E_y(t, n)) \text{ for } m_m(1 - S_t) < t < m_m(1 + S_t) \tag{4.20}$$

and

$$|m_p - m_m| = \min_{t \in S'_p} |t - m_m| \tag{4.21}$$

where m_m denotes the matched mid-point, in the same way that w_m denotes the warping mid-point. The search for the best warp point is illustrated in figure 4.13.

To improve the matching process, use is made of the multiresolution structure of the MFT. A warp tolerance W_t is used in the same way as S_t is used for searching. Having identified the break points, w_p and m_p, the warp point w_p is first refined within a window $w_p W_t$ until a maximum value of the correlation coefficient (4.17) is found. This point is then extrapolated to the next MFT level and the process is repeated until the highest resolution candidate is found. The warping tolerance is a ratio, so that as the search level decreases, the time resolution increases and the search takes place in a narrower window.

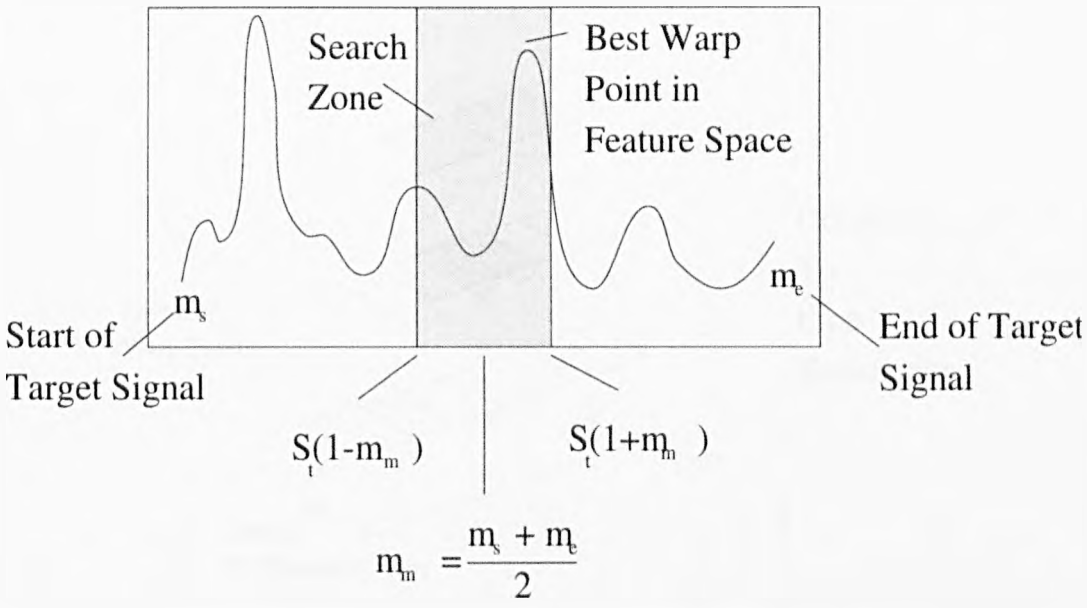

Figure 4.13: Data point searching on the prototype profile.

Once the point is found in the highest resolution it becomes the warping point for that segment of signal (see figure 4.14).

A formalisation of this process can be made by considering (4.17), and using the notations given in (4.12),(4.10),(4.11) and in the list in section 4.4. The warp point is refined at each level using

$$r_{xy}(w_p, n) = \max_{w'_p} r_{xy}(w'_p, n) \tag{4.22}$$

where w'_p is the candidate warp point found from the feature space search method described in the previous section. The correlation function $r_{xy}(w_p, n)$ is defined as

$$r_{xy}(w_p, n) = \frac{\sum_{t=w_s}^{w_p}(X_{W_f^L(w_s-t)+w_s,f,n} Y_{t-w_s+m_s,f,n}) + \sum_{t=w_p}^{w_e}(X_{W_f^R(w_p-t)+w,f,n} Y_{(t-w_p+m_p),f,n})}{\sqrt{(\sum_{t=w_s}^{w_p} X^2_{W_f^L(w_s-t)+w_s,f,n} + \sum_{t=w_p}^{w_e} X^2_{W_f^R(w'_p-t)+w'_p,f,n})(\sum_{t=m_s}^{m_p} Y^2_{t,f,n} + \sum_{t=m_p}^{m_e} Y^2_{t,f,n})}}$$

This process continues recursively, until the maximum correlation is identified at the highest time resolution.

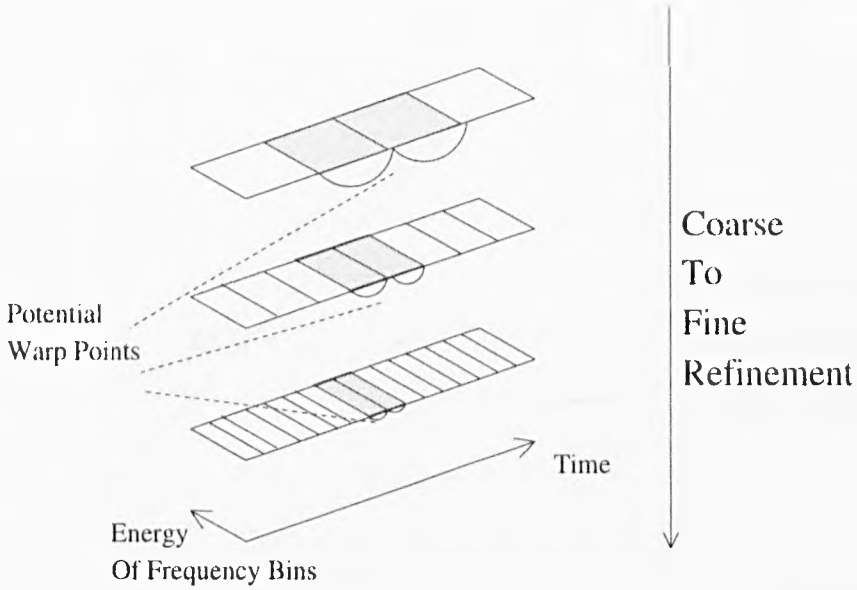

Figure 4.14: Multiresolution warp point search refinement.

4.5.3 The Stopping Criterion for the Refinement Process

If the bisection process continues unchecked, the warp function could contain a list of ordered pairs that would be only one less than the total number of points in the target signal. This is clearly impractical, given that changes in tempo occur comparatively seldom (certainly less than once per second on average). There will therefore be some order of warping which will give a sufficiently good approximation to any continuous warping function so that they will be indistinguishable. This is the point at which nothing would be gained from splitting a given segment further.

After each split, the correlation for the whole segment is compared with that of the two sub-segments and, if there is a decrease in either, the split is rejected and the recursion stops. If, however, there is an improvement, then the segment is bisected and the process continues. It is also desirable to be able to stop the process from reaching too small a scale. There is nothing to be gained from warping and matching only a few coefficients, so a minimum segment length can be set. An illustration of the process is shown in figure

4.15.

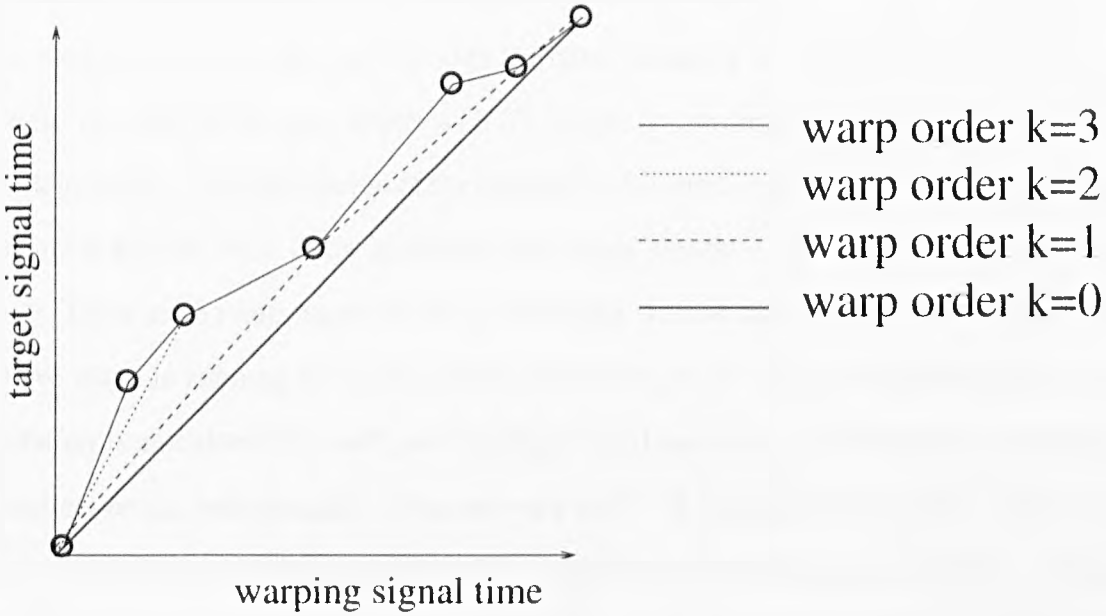

Figure 4.15: A warping function stopped at different orders of warping with warp points marked.

4.5.4 Computational Complexity

The computational complexity of the warping algorithm is not simple to calculate due to its recursiveness. The approach taken here is to calculate the computational complexity for one bisection and then generalise from this to gain an expression for an upper bound on the computational complexity for the whole algorithm.

The warping algorithm (see figure 4.12) chooses candidate break points and refines them by maximising the correlation measurement. The selection of candidate points is a bounded linear search on each MFT level of the energy profiles, which for the purposes of calculating the computational complexity of the algorithm, are considered as one dimensional lists. The next stage, the refinement of these candidate break points, relies on choosing a point, warping the prototype according to this point, and calculating

the correlation coefficient (4.17) with a bounded linear search on each MFT level. The bounded linear searches are $O(N)$ since these are searches on two $1-d$ lists. The warping (cf section 4.4.2) is a single pass through the data, changing the indices on the prototype list to the new warped indices, which is $O(N)$. There is 1 multiplication and 3 additions for each stage of this. The calculation of the correlation is found by multiplying corresponding elements in the two lists — the prototype and target signals — and normalising for their energy. There are 3 multiplications of $3N$ additions. Hence, the correlation measurement is $O(N)$ also. In refining the break points, the prototype list is first warped and then the correlation is calculated for each point within a fixed tolerance. Note that the number of multiplications in both instances does not vary as N. In other words, at each stage of an $O(N)$ process (the searching), another $O(N)$ process (the correlation calculation) takes place. Therefore the complexity for a single scale and a single break point is $O(N^2)$. If we assume that all levels of the MFT are used, then the complexity for a single node is $O(N^2 \log_2(N))$. If the depth of recursion is K then an upper bound on the complexity of the algorithm is $O(N^2 \log_2(N) 2^K)$. In practice, $K \leq 4$ for the examples used in the experiments. A graphical representation of this is shown in figure 4.16.

In practise this algorithm takes around 70 seconds to implement for an audio signal of length 11.8 seconds, sampled at 44.1kHz with a depth of $K = 4$ on the hardware discussed in section 1.6.1.

4.6 Results of Time Warping

The effectiveness of the warping algorithm has been tested on four movements, taken from two pieces. For each of the four examples the warping function is given, along with a plot of the warped, unwarped and target profile — the same time scale is used in each to facilitate comparison.

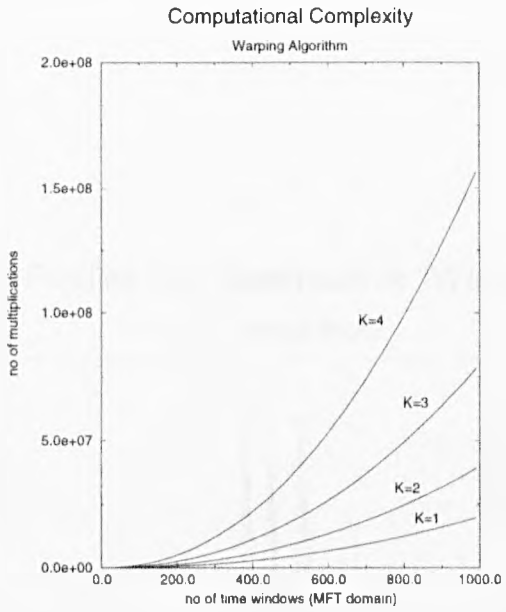

Figure 4.16: Number of multiplications as a function of sample size for the warping algorithm.

In figure 4.18, the warping function has been calculated in both directions to show that the algorithm is symmetric with respect to the target and prototype. Figures 4.21, 4.24 and 4.27 show the other warping functions and figures 4.17, 4.20, 4.23 and 4.26 show the time profiles for the warped, unwarped and target signals. Figures 4.19, 4.22, 4.25 and 4.28 show how the goodness of fit between the two varies as a function of warp order. Note that in figure 4.19 the maximum order of warping is 3. This is due to the extremely slow nature of the piano in the "Waldstein" Adagio. Nothing was gained by the order of warping increasing it beyond three.

Although the final test of the warping algorithm rests in the efficacy of the enhancements obtained using it, visual inspection of the energy profiles shows that the warping does align the major peaks quite successfully. Although the correlations vary considerably between pieces, there is a clear improvement in all cases as the warp order increases.

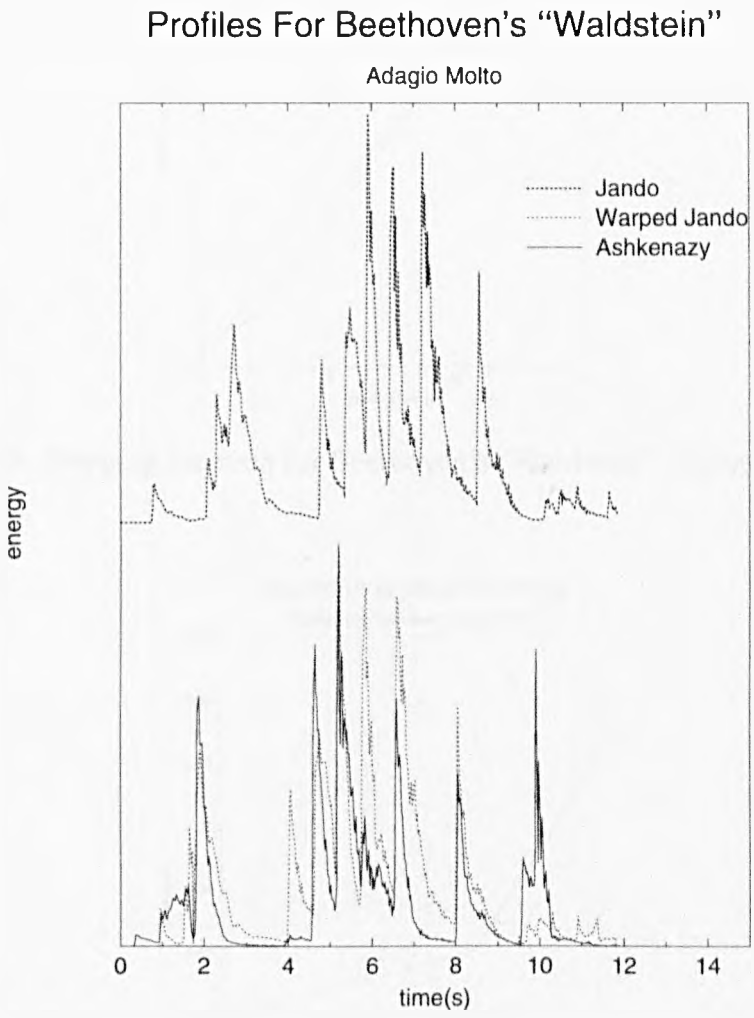

Figure 4.17: Profiles of Beethoven's "Waldstein" (Adagio Molto) showing warping.

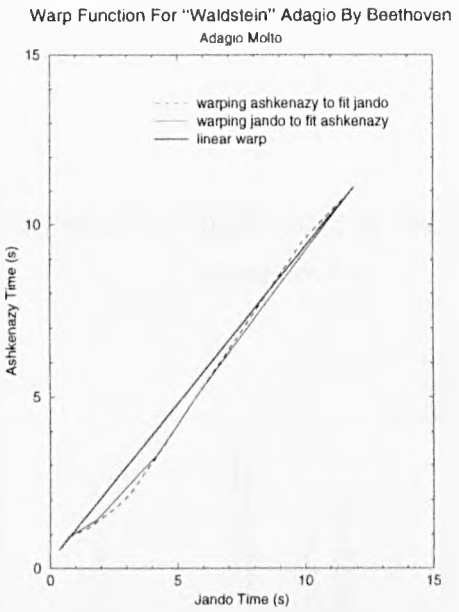

Figure 4.18: Warping function for Beethoven's "Waldstein" (Adagio Molto).

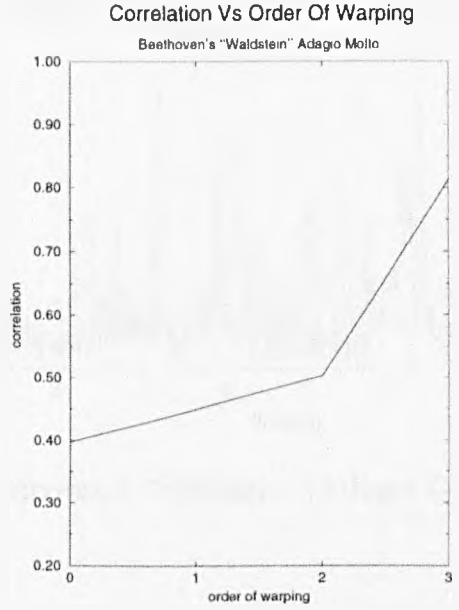

Figure 4.19: Correlation as a function of warp order for Beethoven's "Waldstein" (Adagio Molto).

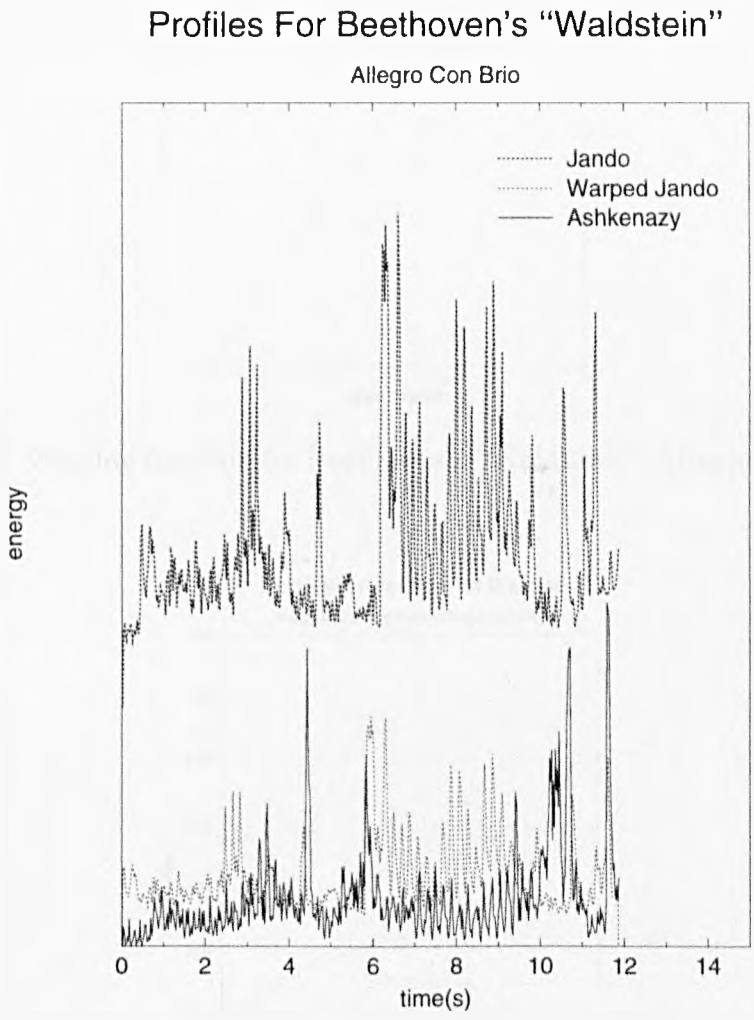

Figure 4.20: Profiles of Beethoven's "Waldstein" (Allegro Con Brio) showing warping.

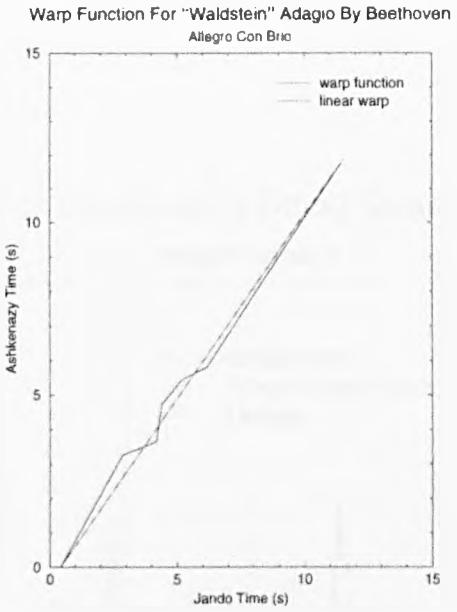

Figure 4.21: Warping function for Beethoven's "Waldstein" (Allegro Con Brio).

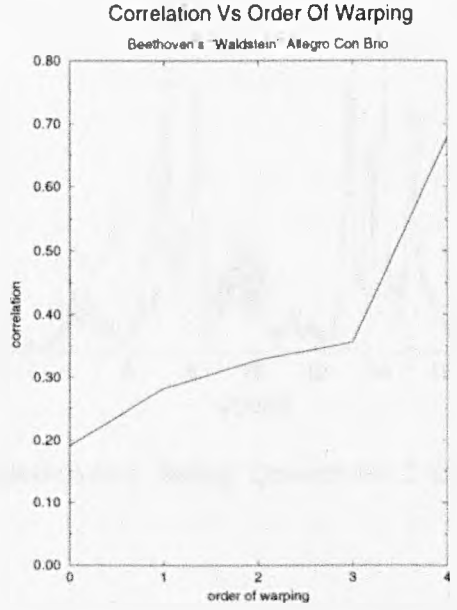

Figure 4.22: Correlation as a function of warp order for Beethoven's "Waldstein" (Allegro Con Brio).

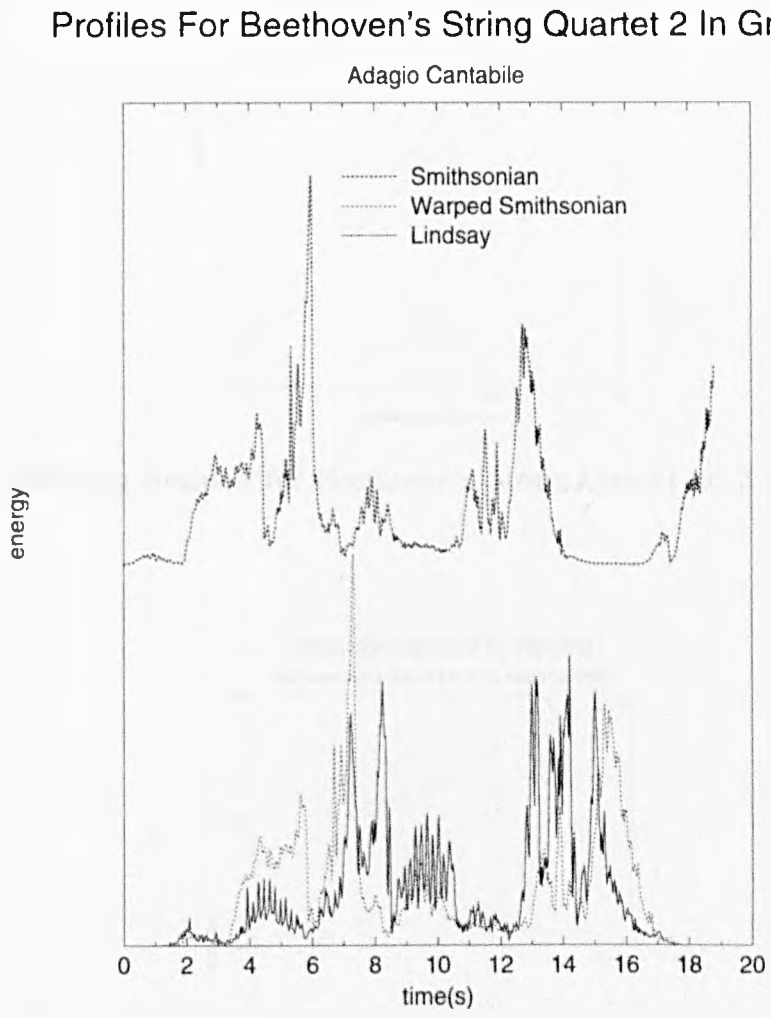

Figure 4.23: Profiles of Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile) showing warping.

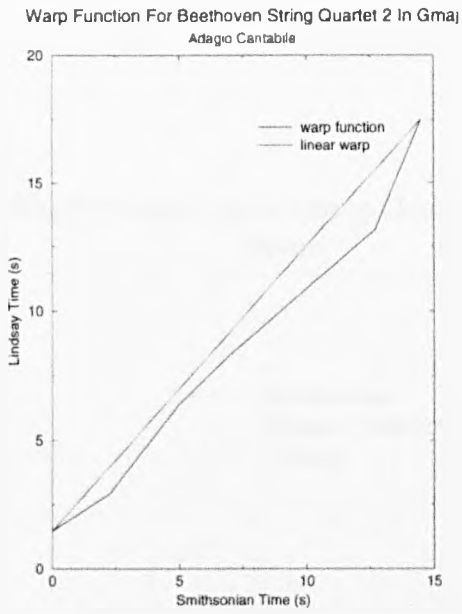

Figure 4.24: Warping function for Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile).

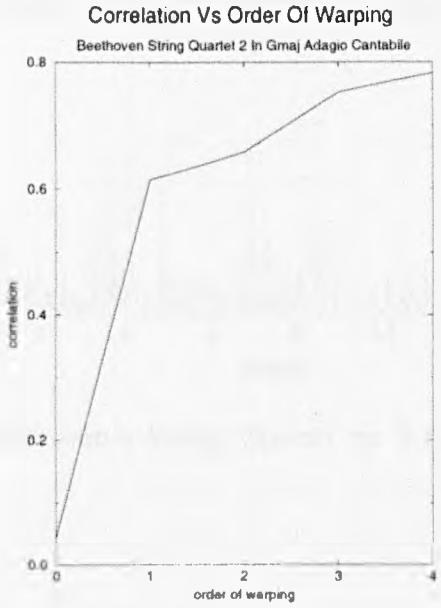

Figure 4.25: Correlation as a function of warp order for Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile).

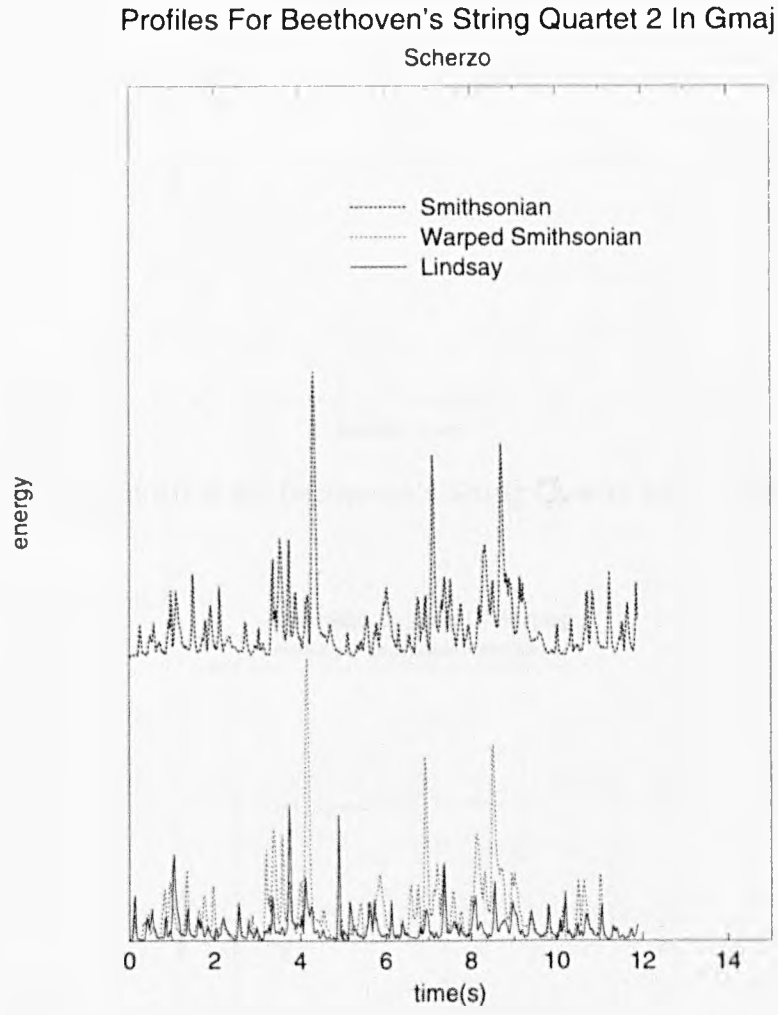

Figure 4.26: Profiles of Beethoven's String Quartet no 2 in Gmaj (Scherzo) showing warping.

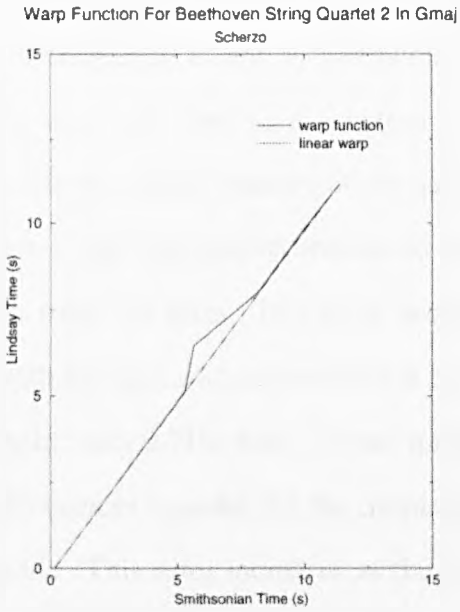

Figure 4.27: Warping function for Beethoven's String Quartet no 2 in Gmaj (Scherzo).

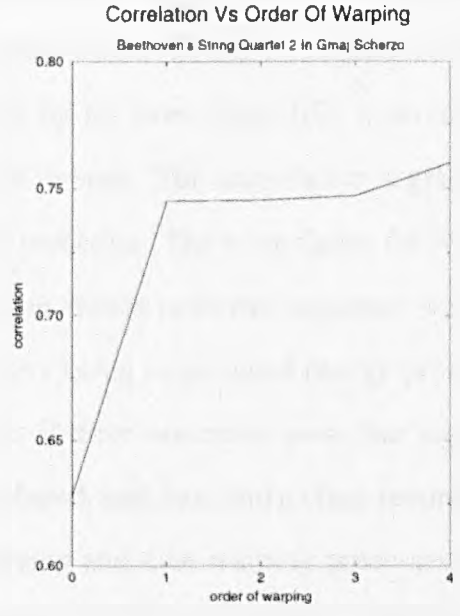

Figure 4.28: Correlation as a function of warp order for Beethoven's String Quartet no 2 in Gmaj (Scherzo).

4.7 Frequency Warping the Template

Warping a musical signal in frequency is not as complicated as warping in time. In general, instruments' tuning does not alter as a function of time or, for that matter, frequency. Whilst theoretically two performances of the same piece should have exact correspondence in frequency — all instruments should be tuned precisely to the same frequency — there is always room for error. This error becomes more noticeable when templating the noisy MFT with the time warped template at high MFT levels. Level 14 of the MFT has frequency samples only 2.7Hz wide, so that middle C must be tuned to less than 1% error between performances in order for the frequency samples of the template and the noisy MFT to coincide. This error increases as the notes played go up in scale, for example at C6 (2046Hz) the accuracy of tuning must be less than 0.25%.

The frequency warping on the MFT template is applied after the time warping so that all the features are time aligned. The MFT and the template are multiplied together so that energy is transmitted only where the MFT template is switched on. The template is then warped in frequency by no more than 10%, a percentage from which any two performances are known not to vary. The warp factor is gradually increased in steps of about 0.001, for reasonable precision. The warp factor for which the maximum amount of energy is transmitted is then chosen to be the frequency warp factor. It is worth noting that this method relies on there being more signal energy present than noise energy. This should generally be the case: if there was more noise than signal it may become difficult to identify the piece being played, and thus find a clean recording to match it to.

Once the frequency warping and time warping processes are complete, the template is ready to be used for enhancing the noisy, prototype signal. In figure 4.29, the template for Beethoven's "Waldstein" by Jandó has been time warped and is being used to restore

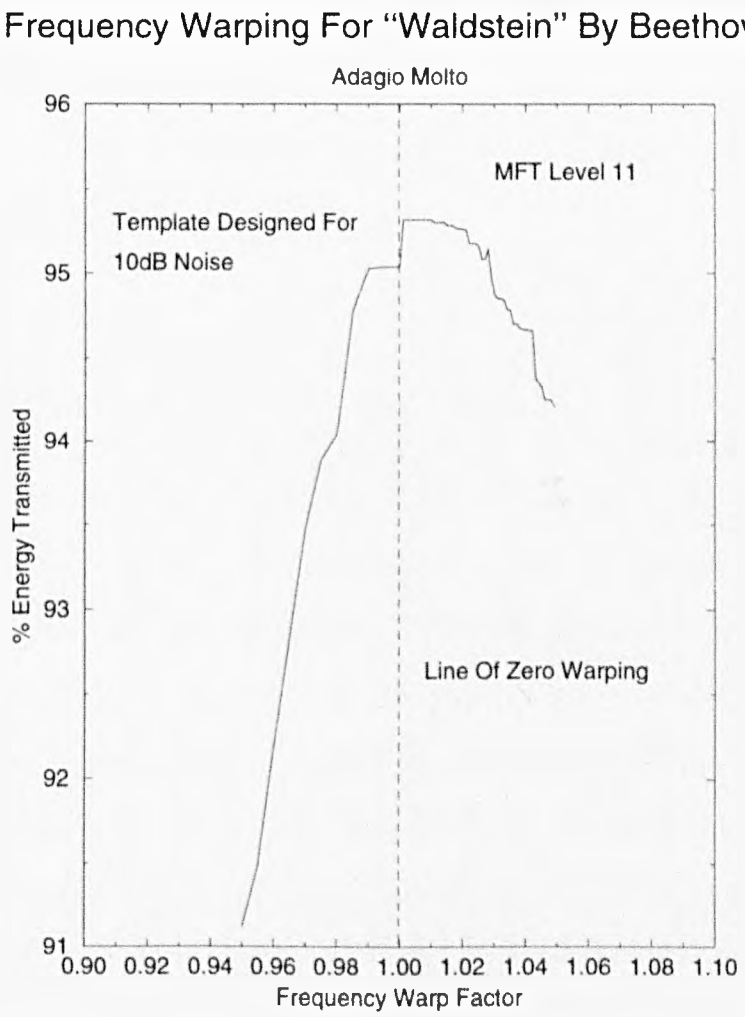

Figure 4.29: Energy transmitted by a time and frequency warped template derived from Beethoven's "Waldstein" performed by Jandö through the Ashkenazy performance with 10dB of noise

a performance by Ashkenazy with 10dB of noise added. As the template is warped in frequency the energy transmitted when the template and the noisy Ashkenazy are overlaid is calculated and plotted. As can be seen there is a plateau with a warping factor of just over 1. This ties in with what one may expect intuitively since pianos can be tuned with a high degree of accuracy.

Chapter 5

Enhancing Noisy Musical Signals using a Warped Template

5.1 Introduction

This chapter is concerned with work which draws on the simple adaptive filter described in chapter 3, and the warping method (of chapter 4), to create a warped adaptive filter of a prototype signal to be used for enhancing a noisy target signal. The threshold parameter of the event detector is a function of both the MFT level at which the thresholding takes place and the noise present in the target signal. The clean prototype is thresholded and used to template the noisy target signal. Results for this will be shown. The time axes of signals restored at various resolutions are then segmented and marked as "steady-state" or "onset" using the onset detector described in chapter 3. Methods for finding the best level for each zone are described, along with results. It is shown that the warped templating process can be used to eliminate impulse noise as well as white additive noise. Finally, the techniques used in chapter's 3 and 4 are brought together to demonstrate a simple implementation of an adaptive Wiener filter, like that used by Vaseghi [59], along with a comparison of results for this method and that of warping the simple adaptive filter.

5.2 Choosing a Template to Enhance Noisy Signals

It was shown in chapter 3 that the template threshold is a function of the SNR of the noisy signal. The threshold ν is a function of the noise present in the target signal, although the template is derived from the prototype. In figure 5.1 the threshold varies as a function of the input signal to noise ratio, which is the amount of noise present in the target signal before filtering. ν can therefore be denoted as $\nu(\rho)$, as it varies as ρ. In section 3.8 a method for estimating the SNR (ρ') of a target signal was described. In practice $\nu(\rho)$ is found empirically by filtering a prototype signal with ρ' noise added to it and varying $\nu(\rho)$ until the gain in SNR is maximised.

5.3 Restoration using a Warped Template on Different Levels

Once the template filter for the prototype signal is found, it is warped and used to enhance the noisy signal by filtering the noisy signal. This allows only those MFT coefficients of significant signal energy to be included in the reconstruction.

If the time warping function is denoted by $\phi(t)$, and the frequency warping function by $g(f)$, the binary template created from the prototype signal's MFT level by $H_{t,f,n}(\hat{y}(t))$, the noisy target signal's MFT level by $X_{t,f,n}$ and the enhanced MFT level by $\tilde{X}_{t,f,n}$ then the enhanced MFT for each level n is given by

$$\tilde{X}_{t,f,n} = \left(H_{\phi(t'),g(f'),n}(\hat{y}(t))X_{t,f,n}\right) \tag{5.1}$$

in keeping with previous notation, t' and f' are the prototype's time and frequency co-ordinates, which are warped to the target time t and frequency f. $H_{t,f,n}(\hat{y}(t))$ is the template for the warped prototype signal, warped to fit the target $x(t)$ using the warping methods described in chapter 4, and $X_{t,f,n}$ is the MFT for target signal $x(t)$. Note that the

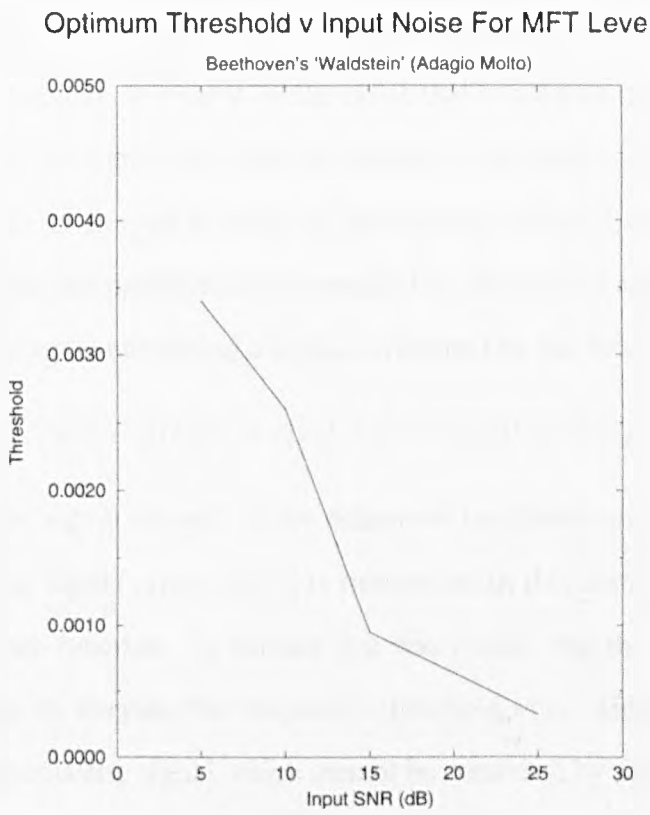

Figure 5.1: A plot of the optimum threshold $\nu(X'_{t,f,n}, \rho')$ for different values of input noise.

tilde denotes warping, and the hat denotes enhancement. The inverse MFT is taken of the enhanced MFT levels producing an enhanced time signal $\tilde{x}(t)$ for MFT level n. Since the enhanced signal is a function of level, the enhancement of noisy signal $\tilde{x}(t)$, created using (5.1) and the inverse MFT, will be denoted by $\tilde{x}(t, n)$.

5.3.1 The Warped Simple Adaptive Filter

The simple adaptive filter is warped using the warping algorithm in chapter 4, from a clean prototype signal to a noisy target. Ideally when the template is applied to the target all the noise will be excluded leaving only signal. If the target signal is $x(t) = s(t) + r(t)$, where $s(t)$ is the clean signal and $r(t)$ is the noise, then the enhanced signal should equal the

noise free signal. That is $\tilde{x}(t,n) = s(t)$. The problem with this is twofold. First, there is noise present in the signal, so even if all the noise that is not part of the signal is ignored, the noise present in the signal must still be included. Secondly, when warping, or even designing a template (cf chapter 3) not all of the signal is included, as some will be missed by thresholding. This last problem is compounded by any lack of accuracy in the warping process. If the error upon enhancing a signal is denoted by the function $\epsilon[\tilde{x}(t,n)]$ then

$$s(t) - \tilde{x}(t,n) = \epsilon[\tilde{x}(t,n)] = \epsilon[s(t)] + \epsilon[r(t)] \tag{5.2}$$

where $\epsilon[s(t)]$ is the signal missed in the enhanced templated signal and $\epsilon[r(t)]$ is the noise included. The signal error, $\epsilon[s(t)]$ is minimised in this instance by increasing the accuracy of the warp function. In chapter 3 it was shown that the error due to missing signal is minimised by varying the template's threshold $\nu(\rho)$. However in this instance there is error due to missing signal which cannot be corrected by adjusting $\nu(\rho)$. There is a trade-off between the two errors in (5.2) which was discussed more fully in chapter 3. Note that $r(t)$ is used here to denote the noisy signal so as not to confuse the reader with n which is the discrete MFT level.

5.3.2 Results for Filtering Noisy Signals using the Warped Filter

For the four samples used throughout this work, noisy input signals for one performance were restored using (5.1). The prototypes were performances of the same pieces by different artists. For the "Waldstein", the prototype performance was by Jandó and the target performance by Ashkenazy. The prototype for the string quartet was performed by The Smithsonian String Quartet and the target performance was by The Lindsay String Quartet. The target signals were corrupted by additive white Gaussian noise giving signal to noise ratios of 10dB, 15dB and 20dB respectively. For each of the restored signals $\tilde{x}(t,n)$ the SNR was measured.

The results are shown in figures 5.2, 5.3, 5.4 and 5.5 for the gain in signal to noise ratio against MFT level. Figure 5.2 includes the results shown in chapter 3 for filtering using the unwarped adaptive filter for comparison, to demonstrate how the gain is altered by warping the filter. The best enhancement is, unsurprisingly, for the slow movement of the "Waldstein" (Adagio Molto), which had the best enhancement in chapter 3 also. The gain in SNR is more than 10dB at most levels, tailing off to about 8dB's at level 14. The result for filtering using a filter derived from the clean target signal shows that there is a slight loss introduced by warping the filter, which worsens as the level increases. The Adagio piano movement gave the best restoration in chapter 3 as well because it consists of long piano notes, which have partials that closely resemble sinusoids. The next best result is for the Allegro, the gain being high for low MFT levels, but decreasing with increasing level, more severely than the decrease in gain for the Adagio. This is due to the speed of the playing. As the level increases the time resolution decreases, which implies that the relatively high proportion of transient energy will not be dealt with adequately. Since the music is played much more quickly in an Allegro piece than in an Adagio piece, the warping would necessarily be more accurate, as there is more chance of placing a warped note in the wrong place — due to the quantity of notes being played in a given time. There is still a gain of about 10dB with an input SNR of 10dB, a substantial amount of improvement.

The two string pieces (Adagio Cantabile and Scherzo) from Beethoven's String Quartet No 2, both give reasonable results for restoration using warping, but are not as good as their piano counterparts. This is partly because of the variability of the performances in terms of attack, decay and vibrato. The slow piece, Adagio Cantabile, gives a low gain as there is a heavy amount of vibrato present in the Lindsay Quartet's performance which is not as noticeable in the performance by the Smithsonian String Quartet. The

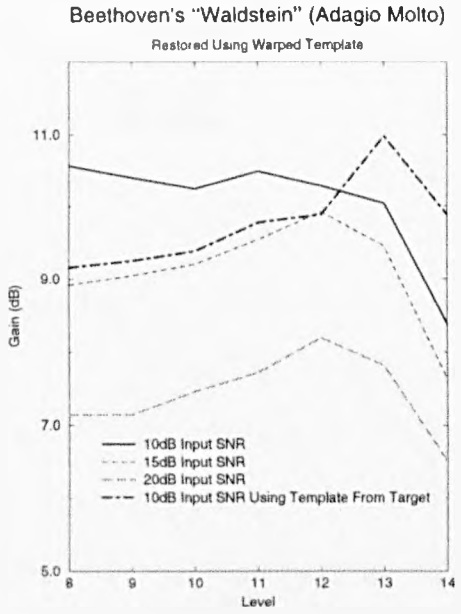

Figure 5.2: Gain in SNR for Beethoven's "Waldstein" (Adagio Molto) for different levels and input SNR's.

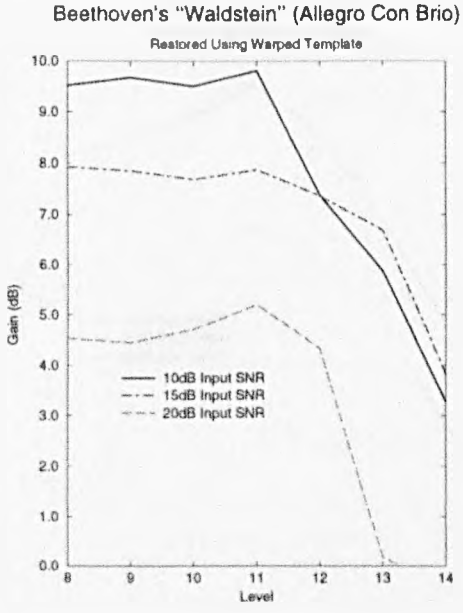

Figure 5.3: Gain in SNR for Beethoven's "Waldstein" (Allegro Con Brio) for different levels and input SNR's.

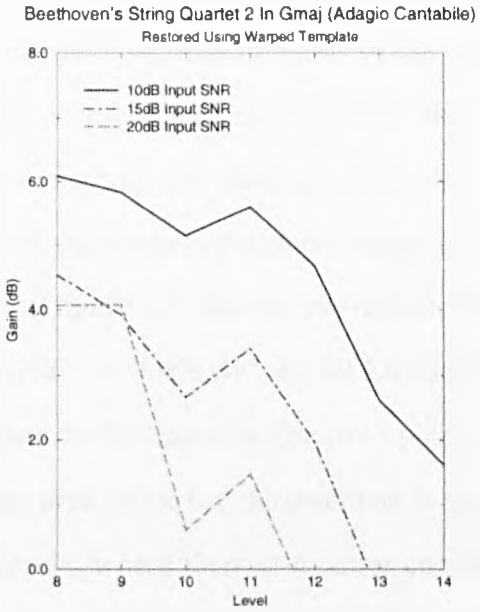

Figure 5.4: Gain in SNR for Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile) for different levels and input SNR's.

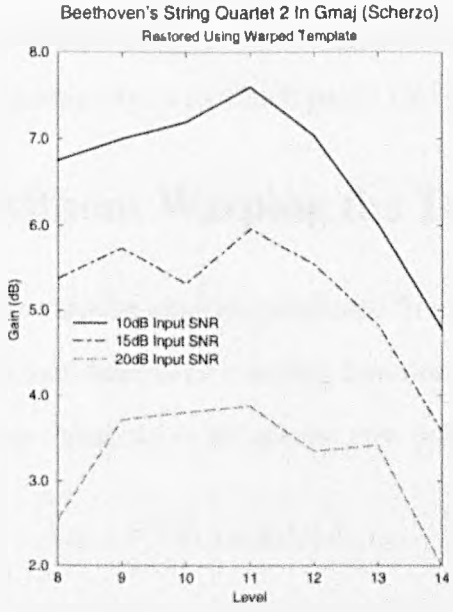

Figure 5.5: Gain in SNR for Beethoven's String Quartet no 2 in Gmaj (Scherzo) for different levels and input SNR's.

difference between the two performances for the Adagio can be seen in figures 5.6 and 5.7. These are MFTs for the same three seconds of signal in the Lindsay String Quartet's and Smithsonian String Quartet's performances, and they show the difference in the amount of vibrato present. It is worth remarking that although for the high levels ($n > 12$) in figure 5.4 the gain in dB is negative, the sound represents a subjective improvement. The quick string piece, results shown in figure 5.5, has the instruction "Scherzo" to the performer. Translated, this means "playful" or "jokingly". In the Lindsay Quartet's performance the attack is faster and louder than the Smithsonian Quartet's piece, and since the Smithsonian is warped to the Lindsay the peak value for enhancement is at an MFT level between the maximum and minimum levels, where there is a better compromise between capturing the loud attack and having the minimum number of MFT coefficients turned on to capture the signal energy. It may also be recalled from chapter 4 that correlation measurements for the warping were lower for both extracts from Beethoven's String Quartet No 2 when compared with the "Waldstein" extracts. Furthermore the correlation was lowest for the string Scherzo. It can be seen by inspecting the energy profiles visually in chapter 4 that there is a certain amount of ambiguity as to which peaks should align.

5.4 Restoration without Warping the Template

To show that it is necessary to warp the template produced from one performance to restore another, (5.1) is modified so that there is no warping function applied. The restoration is performed using the optimum threshold as before but now using

$$\tilde{X}_{t,f,n} = (H_{t,f,n}(y(t))X_{t,f,n}) \tag{5.3}$$

instead of (5.1). Results are shown in figure 5.8 for the restoration of "Waldstein" Adagio Molto using both (5.3) and (5.1), with an input SNR of 20dB. As the level increases, the

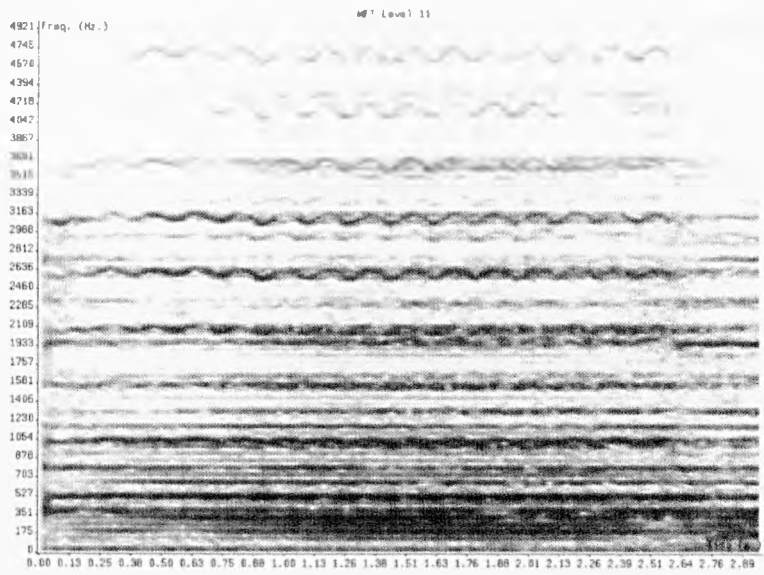

Figure 5.6: Three seconds of Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile) performed by The Lindsay String Quartet, showing the effects of vibrato on the musical partials.

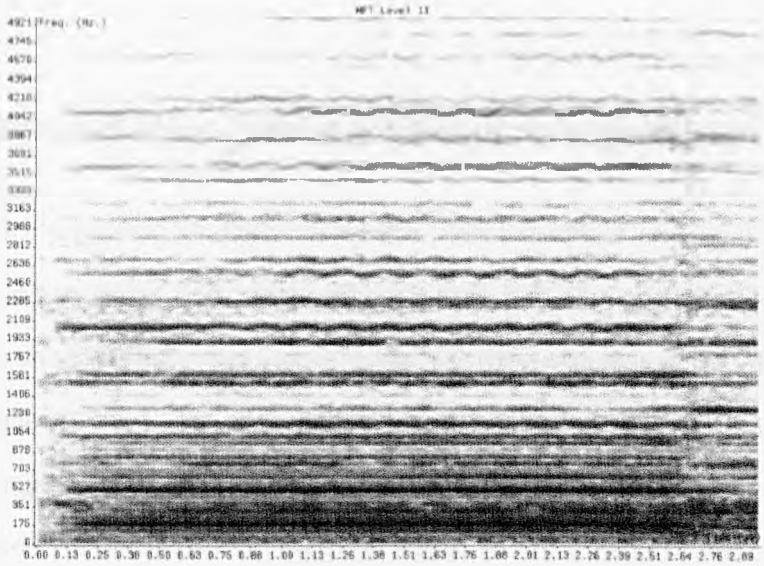

Figure 5.7: Three seconds of Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile) performed by The Smithsonian String Quartet, showing the effects of vibrato on the musical partials.

error between the warped and unwarped enhancements increases also. This is due to the transients in the template taking up wider frequency bins at the wrong time and degrading the restoration. It is easy to see that warping gives a much improved result.

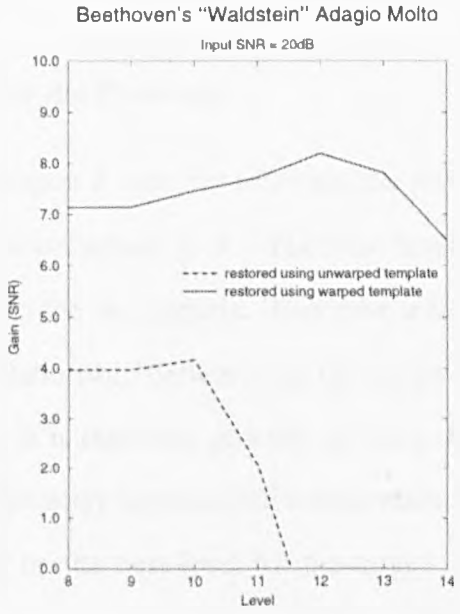

Figure 5.8: The difference between using a warped template and an unwarped template to perform restoration.

5.5 Multiresolution Enhancement using Warping

In section 3.10.1 a method of segmenting the signal in the time domain was discussed. This method used an event detector to detect onsets, which were labelled transients, and everything else was labelled steady-state. Once the time axis was segmented a "best level" choice was made for each segment, and the various "best level" choices for restoration were spliced together to give a multiresolution enhancement. In chapter 3 the choice of level was determined by maximising the signal to noise ratio of the enhanced signals $\tilde{x}(t,n)$, since the clean signal was known *a priori*. In this application, the clean target

signal will not be known, so some method for choosing the best resolution for restoration for a given segment must be found. Since the clean target signal is actually known, it is possible to perform the multiresolution enhancement in the way it was done in chapter 3 to provide a "benchmark" for the methods discussed here.

Warping the Best Levels for the Prototype

The best levels chosen in chapter 3 were for restoring the prototype signal and a version of the prototype that had noise added to it. The best level chosen for each segment maximised the SNR between the two signals. This gave a list of levels as a function of the prototype time. The relationship between the prototype time and the target time is now known (cf chapter 4). It is therefore possible to warp the best level choice for the prototype signal to match the noisy target signal's time scale. This assumes that the best level for the prototype will be the best level for the target. On first consideration this seems a valid assumption: the target and prototype should not vary greatly structurally. The problem is partly to do with error introduced during warping, as the two signals are not matched 100% there will be some noise present in the enhanced signal that should not be there and some signal missing. The ramifications for the transferability of the best level choice between the two signals becomes obvious if the results for the restoration on single levels are considered. The results for the unwarped filter acting on "Waldstein" Adagio in chapter 3 shows that level 14 has maximum gain but for the warped filter level 11 is the best. This is shown in section 5.3. Also, if the two Adagio String Quartet performances are considered, it can be seen that whilst they do not vary musically, the structure of their partials is not the same. The Lindsay Quartet's performance has both audibly and visibly more vibrato present than the Smithsonian's, making the best choice of level for the Lindsay performance a low one. Hence, even if the warping correlation was 100%,

the best levels are not necessarily the same for the prototype and target signals. It can be concluded that the underlying assumption in transferring the choice of levels from one signal to the other is not necessarily true. This was confirmed by experiment. So some method must be found that does not require any reference signal, but works only on the different signals enhanced at different resolutions of the MFT.

The Average of Levels

It is assumed that in all the restored signals $\tilde{x}(t,n)$ there is more signal energy present than there is noise. Furthermore, it is expected that low levels will have better transient enhancement and high levels better steady state enhancement. The average of all these signals for a given time segment should have some of the qualities of the low levels and some of the qualities from the high levels. The "best" signal for a given segment should also contain as much of the high level steady state enhancement and low level transient enhancement as possible. If a contribution is taken from all of the different signals, using some weighting function α_n, that is a function of level, with

$$\sum_n \alpha_n = 1 \tag{5.4}$$

The level can be chosen so that the enhanced signal at that level is the closest to the weighted composite signal, for each segment i with $t \in [t_i, t_{i+1}]$, choosing n such that $\delta(n)$ is minimised

$$\delta(n_{\text{optimum}}) = \min_n \sum_{t=t_i}^{t_{i+1}} |\tilde{x}(t,n) - \bar{\tilde{x}}(t,n)|^2 \tag{5.5}$$

where

$$\bar{\tilde{x}}(t) = \sum_n \alpha_n \tilde{x}(t,n) \tag{5.6}$$

The problem with this method is that the values of α_n need to be chosen, according to some model of signal behaviour, which is lacking at present. However a more simple

solution is to use the average signal. This meets the criteria stated above for an ideal reference signal. In other words $\bar{\tilde{x}}(t,n)$ is defined as

$$\bar{\tilde{x}}(t) = \frac{\sum_n \tilde{x}(t,n)}{N_{\text{levels}}} \tag{5.7}$$

and using (5.5) as above. Using this method, the signal chosen should be a compromise between features at high and low levels.

Minimum Variance

This method assumes that the further the signal values for a chosen level are from the ideal signal's values, the less likely it is to be the best level for that segment. In other words, it is assumed that the best level for the segment is the one for which the restoration is most like its neighbours in scale. Using the same notation as above, the level chosen using this method will be given by

$$\delta(n_{\text{optimum}}) = \min_n \sum_{t=t_i}^{t_{i+1}} (\tilde{x}(t,n) - \tilde{x}(t,n-1))^2 + (\tilde{x}(t,n) - \tilde{x}(t,n+1))^2 \tag{5.8}$$

This method creates a problem at the maximum and minimum values of n where either $n+1$ or $n-1$ may not be available. The problem is solved by simply counting the contribution of the available neighbour twice.

5.5.1 Results for Multiresolution Enhancement

Using the onset detector, the time axis is segmented into onset or steady state. The levels chosen for each segment are shown for both of the above methods for the slow piano piece (Beethoven's "Waldstein" Adagio Molto) and the quick string piece (Beethoven's String Quartet No2 in Gmaj) in figures 5.10 and 5.11 respectively. Results for all four pieces are shown in figures 5.12, 5.13, 5.14 and 5.15, where the x-axis is the input signal to noise

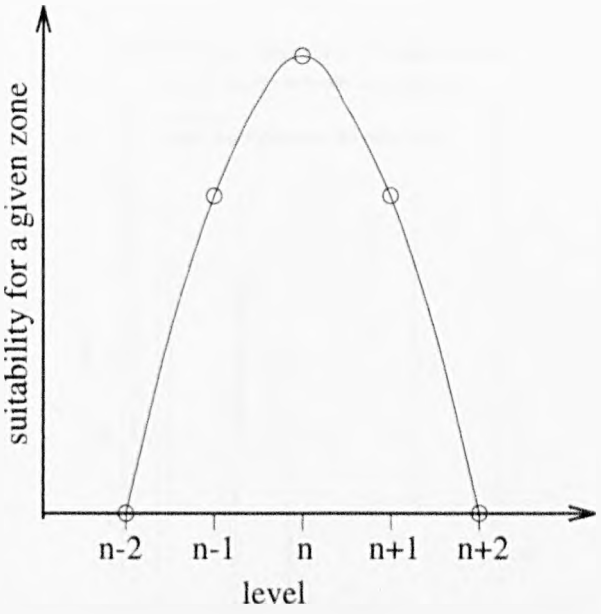

Figure 5.9: Demonstrating the motivation for using a least square difference approach to choosing levels.

ratio and the y-axis represents the gain in SNR in the output. In figure 5.12 the best result for single level enhancement (cf section 5.3) is included. In 5.12, 5.13 and 5.14 the gain is also plotted against input SNR for the *actual best choice* of level. The best choice is found using the target signal without noise added to determine which of the restored signals is closest to it — as was done in chapter 3.

The results show that in all but one piece, Beethoven's "Waldstein" Allegro Con Brio, the average method works best. In the case where the minimum variance method performs better it gives only a slight improvement on the average method. Hence of the two methods discussed above, the method of averaging over levels gives the best performance. The main advantage of using the multiresolution method is that it gives a consistently good result for the enhancement and does not force the user to consider which level would be the best for a given piece of music. The results do not show large improvements in gain

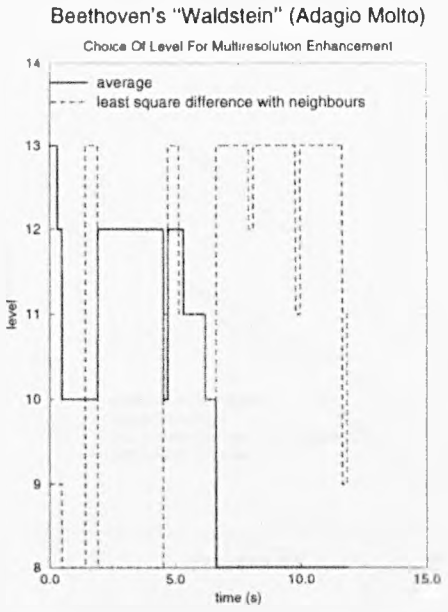

Figure 5.10: The choice of levels for Beethoven's "Waldstein" (Adagio Molto).

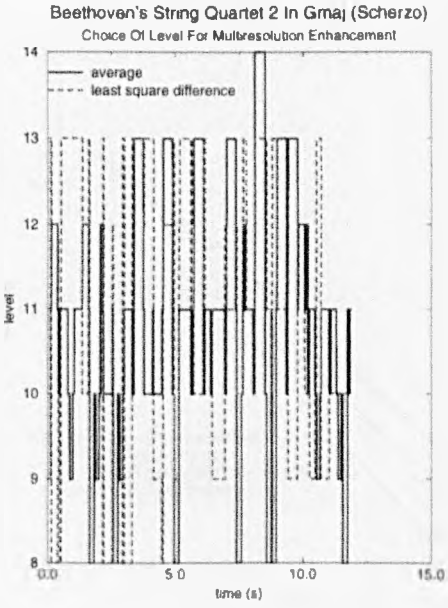

Figure 5.11: The choice of levels for Beethoven's String Quartet no 2 in Gmaj (Scherzo).

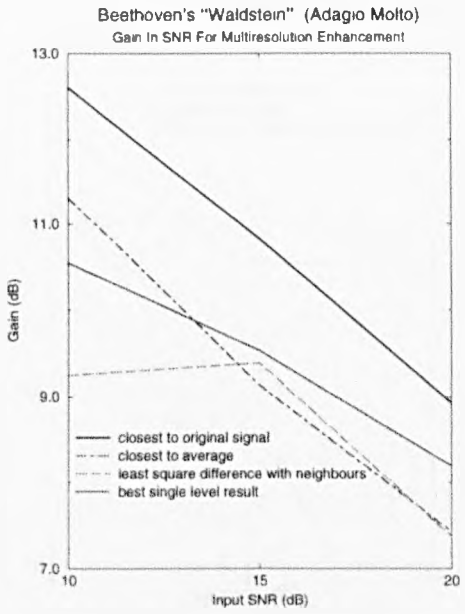

Figure 5.12: Gain against input SNR for two multiresolution enhancement methods for Beethoven's "Waldstein" (Adagio Molto).

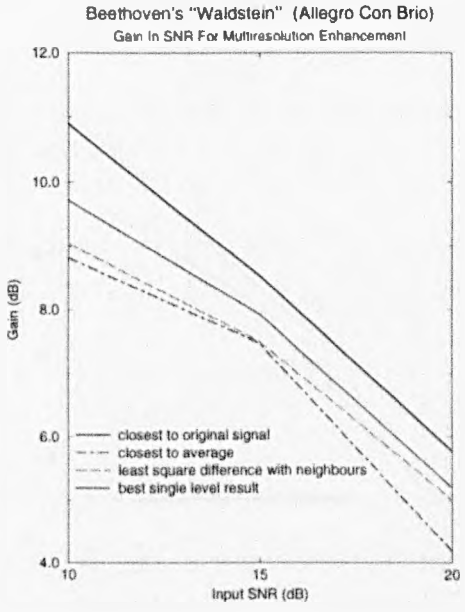

Figure 5.13: Gain against input SNR for two multiresolution enhancement methods for Beethoven's "Waldstein" (Allegro Con Brio).

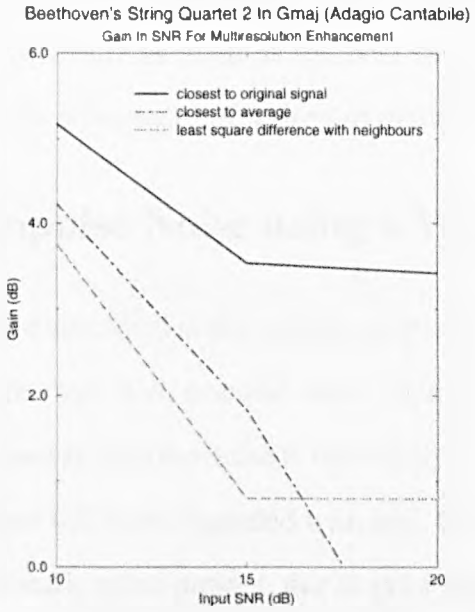

Figure 5.14: Gain against input SNR for two multiresolution enhancement methods for Beethoven's String Quartet no 2 in Gmaj (Adagio Cantabile).

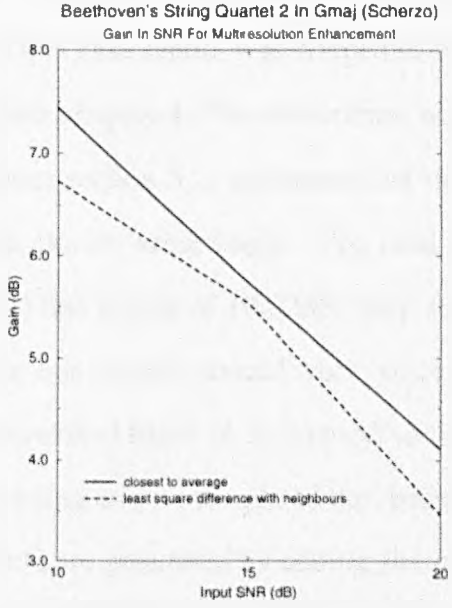

Figure 5.15: Gain against input SNR for two multiresolution enhancement methods for Beethoven's String Quartet no 2 in Gmaj (Scherzo).

found by using the multiresolution method, as they did in chapter 3, but the fact that the results are consistent and generally as good as the best values for individual levels, is enough to convince one of the advantages of using this method.

5.6 Removing Impulse Noise using a Warped Template

Whilst the templating method described in this chapter is designed to be used in broadband Gaussian noise, it also works well with impulse noise. "Clicks" on a record are usually caused by surface damage, and so impulse noise is very likely to occur on old gramophone recordings, where the surface will have degraded with age. On some radio transmissions of recordings, there will be static noise present, due to poor reception, which is also well modelled by impulses of the kind used here.

Figure 5.16 shows a section from "Waldstein" (Adagio Molto) with 10dB of white additive Gaussian noise as well as impulses present. The method used for restoring this is exactly the same as that in section 5.3. The noise was estimated and a template derived from the clean prototype, in this case Jandó, was warped to fit the target signal using the techniques and algorithm from chapter 4. The restoration was performed on MFT level 11. The noise estimator, from section 5.2, estimated that the input SNR was just over 11dB and the threshold was chosen accordingly. The final signal to noise ratio of the enhanced signal (figure 5.17) had a gain of 10.02dB, very similar to that in section 5.3. It is reasonable to expect that this method should work, since each template consists of a number of time varying narrowband filters in frequency, so that any "clicks" that are not present in the prototype recording will be templated out, because they are broadband.

The impulses, in this work, are generated by adding sharp increases of fixed amounts to a musical piece, at a random place determined by the Poisson distribution [21]. The intensity of the Poisson distribution determines how many impulses there will be per unit

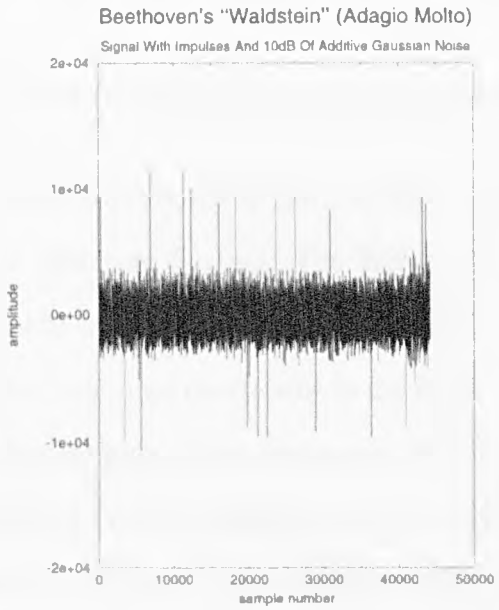

Figure 5.16: A section of Beethoven's "Waldstein" (Adagio Molto) with impulse noise added.

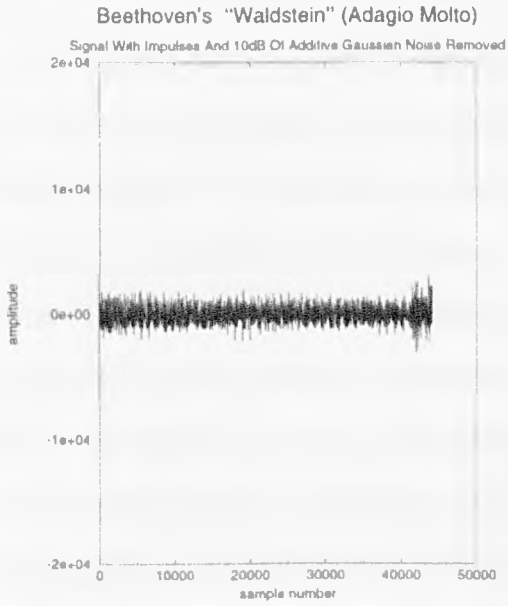

Figure 5.17: A Section of Beethoven's "Waldstein" (Adagio Molto) with impulse noise removed.

time.

5.7 Adaptive Wiener Filtering using Time Warping

In section 3.4, a Wiener filter was shown to have a response that varied as the signal spectrum $S_{ss}(\omega)$ and noise spectrum $S_{nn}(\omega)$. The Wiener filter can be made adaptive by varying the filter response through time. Using the MFT, it is possible to have a Wiener filter that varies from one time coefficient to the next. This can be implemented easily. Consider the implementation of the stationary Wiener filter in chapter 3. The spectral estimate was the energy of the prototype over all time. If the time over which the spectrum is estimated is restricted to, say, one time window, then the Wiener filter's frequency response will change between subsequent time bins. So for time t the adaptive Wiener filter's frequency response can be written

$$H(\omega,t) = \frac{S_{ss}(\omega,t)}{S_{ss}(\omega,t) + S_{nn}(\omega,t)} \tag{5.9}$$

since both the spectrum and the noise are estimated for each time bin, t. This is a naïve way to estimate a signal's spectrum, as it permits abrupt changes in the Wiener filter's frequency response from time t to time $t+1$. To avoid this, a sliding window is introduced: for a window of length $2\zeta(n)+1$ on level n, the time bins $t-\zeta(n) \leq t \leq t+\zeta(n)$ contribute to the signal and noise estimates. A cosine-squared window was used to reduce any artifacts that may be caused by taking spectral contributions over several time bins, giving a smoother transition in the frequency response of the Wiener filter from one time bin to the next. The noise estimate is made for each time bin using (3.30) and smoothed with the same window that smoothes the spectral values. To simplify the notation, $\cos^2_\zeta(t)$ will denote the cosine squared window function centred at t, extending to $t \pm \zeta$. Using the notation of section 5.2, the spectral and noise energy estimates for time t and window

length $2\zeta(n)+1$ are

$$S_{nn}(\omega,t) = \sum_{t'=t-\zeta(n)}^{t+\zeta(n)} \cos_\zeta^2(t')|\Sigma_n|^2 \tag{5.10}$$

$$S_{ss}(\omega,t) = \sum_{t'=t-\zeta(n)}^{t+\zeta(n)} \cos_\zeta^2(t')|X_{t',f,n}|^2 \tag{5.11}$$

where $x(t)$ is the prototype signal, $y(t)$ the target and Σ_n is the amount of noise estimated in $x(t)$ using (3.30). For the case considered here, where the two signals $x(t) \neq y(t)$ the spectral values $X_{t,f,n}$ must be normalised to be used as an estimate of $Y_{t,f,n}$. This is done by scaling for equal variance, using

$$\sigma_{\text{scale}}^2 = \frac{\sum |Y_{t,f,n}|^2}{\sum |X_{t,f,n}|^2} \tag{5.12}$$

which modifies the spectral estimate to

$$S_{ss}(\omega,t) = \sum_{t'=t-\zeta(n)}^{t+\zeta(n)} \cos_\zeta^2(t')|X_{t',f,n}|^2\sigma_{\text{scale}}^2 \tag{5.13}$$

The noise variance is taken from the target signal and so no scaling is required. The Wiener filter now uses the normalised prototype signal spectrum as its estimate for the target's spectral values. This assumes that they are aligned, which is not the case. They can be aligned, by using the warping functions $\phi(t)$ and $g(f)$ as above. Hence the spectral estimate using a clean unaligned prototype is

$$S_{ss}(\omega,t) = \sum_{t'=t-\zeta(n)}^{t+\zeta(n)} \cos_\zeta^2(t')|X_{\phi(t'),g(f),n}|^2\sigma_{\text{scale}}^2 \tag{5.14}$$

which is easy to implement, once the warping functions $\phi(t)$ and $g(f)$ are known. Thus the frequency response for an adaptive Wiener filter restoring degraded audio signals using a sliding window of length $2\zeta(n)+1$ and the warping function from chapter 4 is

$$H(\omega,t) = \frac{\sum_{t'=t-\zeta(n)}^{t+\zeta(n)} \cos_\zeta^2(t')|X_{\phi(t'),g(f),n}|^2\sigma_{\text{scale}}^2}{\sum_{t'=t-\zeta(n)}^{t+\zeta(n)} \cos_\zeta^2(t')|X_{\phi(t'),g(f),n}|^2\sigma_{\text{scale}}^2 + S_{nn}(\omega)} \tag{5.15}$$

5.7.1 Results

Results for restoration of Beethoven's String Quartet (Scherzo) and Beethoven's "Waldstein" (Adagio Molto) are shown in figures 5.24 and 5.23 respectively for the gain in SNR as a function of both MFT level and input SNR. The gain is plotted as a function of sliding window length $\zeta(n)$ in figure 5.18 for the Adagio "Waldstein" and in figure 5.19 for the Scherzo String Quartet piece for $n = 11$. This is the best compromise level, as the results of this work and others have shown [59]. As the window length increases from $\zeta(n) = 3$, there is an improvement in the gain for both pieces. The "Waldstein", which had the most accurate warping correlation of the four pieces, has gain plotted as a function of window length for a lower level, $n = 8$. For level 11 the gain peaks for $\zeta(n) = 3$ then decays, whereas for level 8 there is a steady increase as the window lengthens. This is common sense — the windows on level 11 are 8 times as long as their counterparts on level 8. The gain in a noisy signal due to adaptive Wiener filtering using warping is attributable to two things: the estimate of the noise spectrum, and the estimate of the local signal spectrum. The noise spectrum estimate's accuracy increases as the window length increases, as there are more values for the estimator to use, but the local spectrum estimate's accuracy will decrease as its locality decreases. In other words there is a trade-off between estimating the noise and estimating the local spectrum. There will, therefore, be an optimal length of window over which to estimate these values, which is why the maximum peaks in figure 5.18 are different for different levels. Also the optimal adaptive Wiener filter length varies as a function of the piece of music that is being played. For faster pieces, the filter will necessarily become less stationary to obtain a better restoration as the rate at which the notes are played in the music increases. Figures 5.20 and 5.22 are plots of gain against input SNR for MFT level 11 for both pieces used in the analysis here. Both figures display clearly that the gain of the adaptive Wiener filtering is a function of input SNR — as the

SNR increases, the gain decreases. The results plotted here are for the *optimal filter length*, found empirically. In chapter 3, it was shown that the stationary Wiener filter, lowpass filter, and filtering using the simple adaptive filter all varied as a function of input SNR. Figures 5.23 and 5.24 show how the gain varies as a function of both input SNR, and MFT scale. Unlike the stationary Wiener filtering in chapter 3, the gain does not necessarily increase as the frequency resolution does. This is because of the time-varying nature of the filter, which allows the best spectral and noise estimate for a given section — $\zeta(n)$ — of the time axis. Again, the results plotted are for the *optimal filter length* or value of $\zeta(n)$ that gives the maximum gain. The first thing to note is that the shapes of figures 5.23 and 5.24 are similar to those gain curves plotted using the simple adaptive filter in figures 5.5 and 5.2 for the string quartet and "Waldstein" pieces respectively. Both methods use the same warping function found using the algorithm of chapter 4. The "Waldstein" Adagio does best at high levels, because it is slow, and the String Quartet Scherzo does best at low levels, because it is fast. The Scherzo shows best filtering results with a highly adaptive filter, whereas the Adagio shows best results when the filter is less adaptive. Again, both graphs illustrate clearly how the gain decreases as the input SNR increases.

The results in figures 5.23 and 5.24 can be compared with the results in figures 5.12 and 5.15 for the "Waldstein" Adagio and String Quartet Scherzo respectively. The values for the gain in dB's are given in tables 5.1 and 5.2. For the String Quartet piece, the simple adaptive filter does better than the Wiener filter, by about 1 or 2 dB, even when not combined through scale, whereas the "Waldstein" Adagio is 1dB better when restored using the Wiener filter. It can therefore be concluded that both methods give roughly the same gain in SNR.

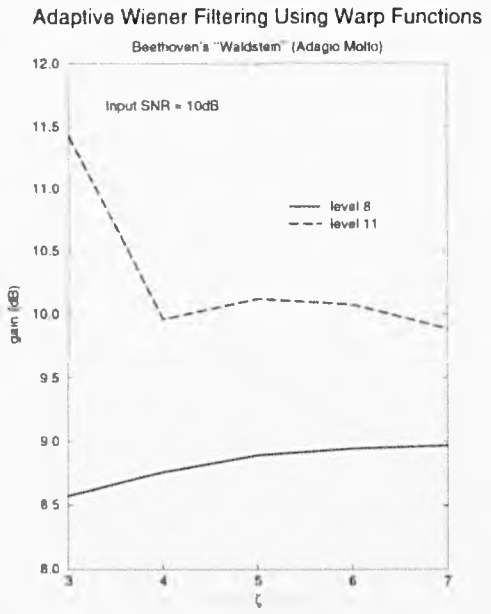

Figure 5.18: Gain as a function of window length for Beethoven's "Waldstein" (Adagio Molto) for MFT level's 8 and 11.

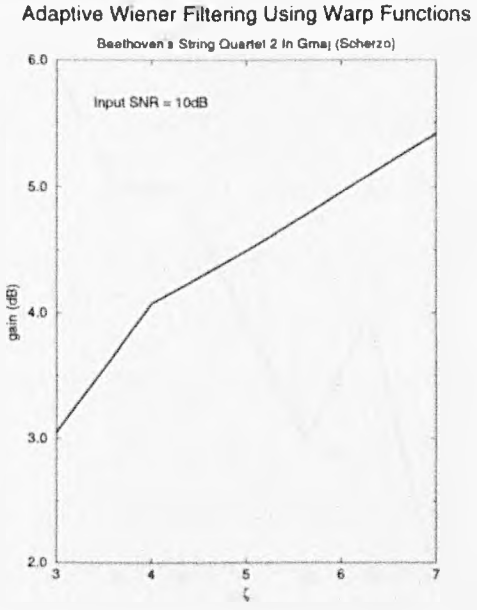

Figure 5.19: Gain as a function of window length for Beethoven's String Quartet no 2 in Gmaj (Scherzo) for MFT level 11.

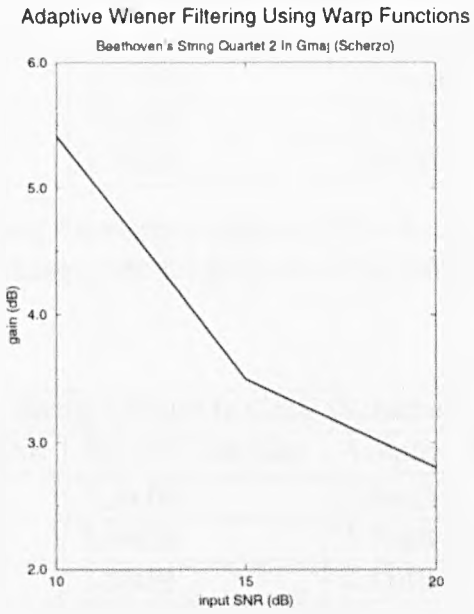

Figure 5.20: Gain as a function of input SNR for Beethoven's String Quartet no 2 in Gmaj (Scherzo) for MFT level 11.

Figure 5.21: Best window length as a function of level for Beethoven's "Waldstein" (Adagio Molto) with an input SNR of 10dB.

"Waldstein" (Adagio)		
Input SNR	*Warped Template*	*Adaptive Wiener*
10dB	10.56dB	11.02dB
15dB	9.91dB	9.73dB
20dB	8.19dB	7.83dB

Table 5.1: Gain in SNR using the warped adaptive filter and the warped Wiener filter for Beethoven's "Waldstein" (Adagio Molto) performed by Ashkenazy.

String Quartet In Gmaj (Scherzo)		
Input SNR	*Warped Template*	*Adaptive Wiener*
10dB	7.59dB	6.66dB
15dB	5.96dB	5.73dB
20dB	3.88dB	4.11dB

Table 5.2: Gain in SNR using the warped adaptive filter and the warped Wiener filter for Beethoven's String Quartet no 2 in Gmaj (Scherzo) performed by The Lindsay String Quartet.

5.8 Combining Warped Prototype and Target Derived Templates

Having compared the restoration gained by using an adaptive Wiener filter to that obtained from the simple adaptive filter of chapter 3, our attention returns to the notion of using the target signal as a filter template. This was investigated in chapter 3, for Beethoven's "Waldstein" by Jandó, where it was easily seen that there was a significant improvement in the gain if the prototype signal was used instead of the noisy target. This was obviously going to be the case since, as the SNR of the target signal increases, it became increasingly like the prototype, meaning that in the limit, the gain in restoration using the target would, at best, only ever be as good as the gain found by using the prototype. However, when using a warping function and two different recordings the story is more complicated. The problem is compounded by such matters as the accuracy of the warping algorithm as well

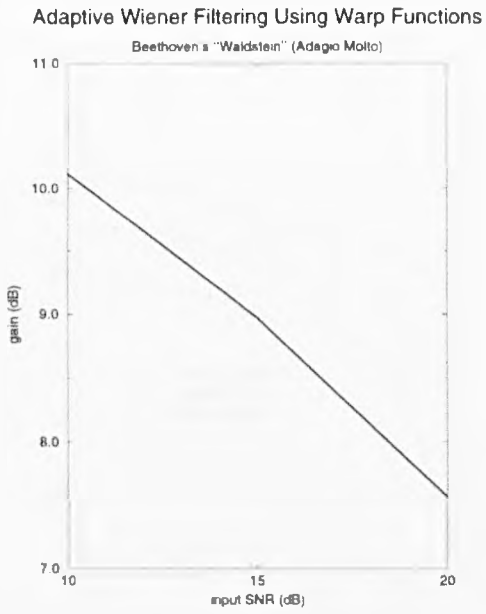

Figure 5.22: Gain as a function of input SNR for Beethoven's "Waldstein" (Adagio Molto) for MFT level 11.

as the amount of noise in the target signal. To investigate this, SNR's are shown in figures 5.25, 5.26 and 5.27, for the noisy Ashkenazy Adagio signal with 10dB, 15dB and 20dB respectively. The results show the gain when using the warped prototype filter, the filter derived from the noisy signal and linear combinations of the two.

The signals were added in the time domain for simplicity — the respective results for the warped prototype and noisy target are stored in the time domain. Combinations of the two were calculated using the simple formula

$$z(t) = p\tilde{x}(t) + (1-p)\tilde{y}(t) \tag{5.16}$$

where p denotes the proportion of the target signal included in the final signal. As can be seen in all of figures 5.25, 5.26 and 5.27, the target derived signal has a higher peak, at level 14 and, as the SNR increases, the difference between the peak target derived filter restoration and the prototype restoration decreases. Eventually, as the SNR becomes

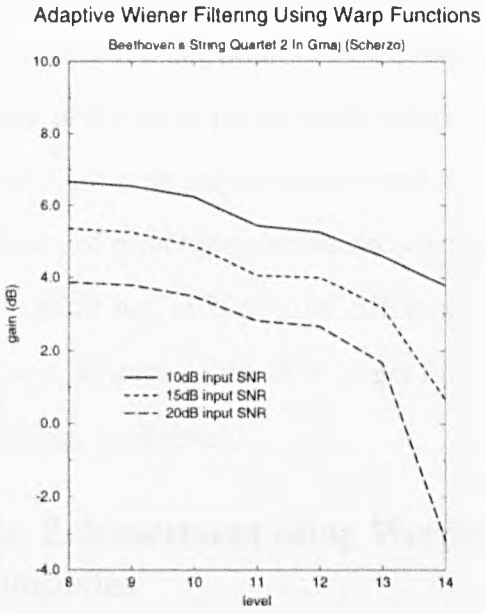

Figure 5.23: Gain as a function of level for Beethoven's String Quartet no 2 in Gmaj (Scherzo) for 10dB, 15dB and 20dB input SNR.

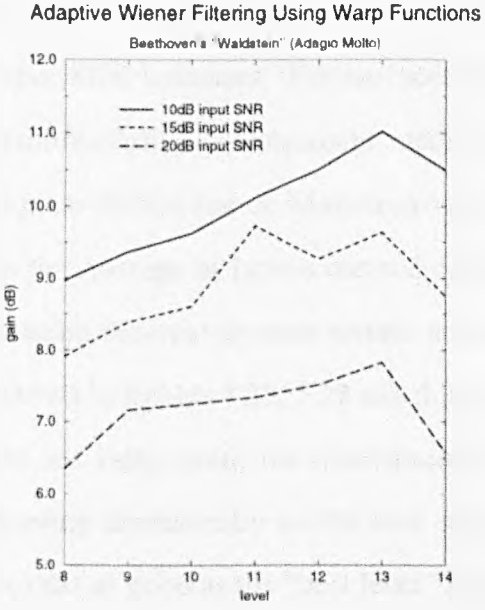

Figure 5.24: Gain as a function of level for Beethoven's "Waldstein" (Adagio Molto) for 10dB, 15dB and 20dB input SNR.

infinite and there is no noise in the target signal to restore, the ideal filter will be target derived. In practice this will happen at the point at which the noise introduced by using a warped template exceeds that of the noise in the target signal. The important point is that by combining the two filters there is an improvement overall in the SNR gain. Again, as the SNR increases, the amount of prototype needed becomes less. At 10dB having 50% of each filter gives the best result but, at 20dB, the difference between the 50% and the 70% target filter narrows until, at level 14 the 70% target filter overtakes the peak of the 50%. This bears out the findings in chapter 3.

5.8.1 Multiresolution Enhancement using Warped Prototype and Target Derived Templates

The restored time signals derived as above can be combined across scale employing the same methods used in section 5.5. The results for this are plotted in figures 5.28, 5.29 and 5.30 for $p = 0.5, 0.7$ and 0.9 respectively. Once more by only combining half and half from both signals, the best results are gained for $p = 0.5$ in all cases, but the difference in gains reducing as the input SNR increases. Furthermore there are similarities in the performance of the two multiresolution enhancement methods employed, the Average of Levels and the Least Square Difference or Minimum variance methods. In all but one of the examples shown the Average of Levels method outperforms the Least Square Difference method in combining the time-domain signals across scale to the best effect. By comparing the results shown in figures 5.28, 5.29 and 5.30 with those in figures 5.25, 5.26 and 5.27, it is easy to see that, again, the multiresolution enhancement methods employed, whilst not improving dramatically on the best single level result, do give a result that is consistently at least as good as the "best level" result.

5.9 Summary

This chapter has shown results for restoration of noisy audio signals at various resolutions and using multiresolution methods. These results show that whilst there is a degradation in the improvement of a noisy signal using the warping method of chapter 4 compared to using an exact template as in chapter 3, restoration using the warped template gives an improvement over that achieved using information derived solely from the target. Methods were discussed that combined signals restored at different levels, depending on whether the signal was steady state or an onset as used in chapter 3. The two main methods relied on different approaches, with different results. The first used a weighted composite of all the signals as a reference signal and picked the resolution of signal that was closest. This gave a consistent improvement over the best single resolution result, but was not quite as good as using the original clean signal as a reference. The second used the variance of all the signals with their nearest neighbours, and chose the best signal as the one having the lowest such mean variance. This was not as good as the composite weighted, or average, method. It was also shown that the warped template, as well as restoring broadband white Gaussian noise, copes well with impulses which would occur due to surface degradation in a gramophone recording or static in a radio transmission. The idea of adaptive Wiener filtering was introduced and, using the warping function from chapter 4, was implemented on the noisy target signal without any *a priori* information about its spectrum. This is a simpler implementation of the filter used by Vaseghi [59], made so by using the MFT and the warping functions. The results using this method were shown to be no better than the results using the computationally simple adaptive filter used earlier on, and discussed in chapter 3. Finally, in section 5.8, a general method was displayed that allowed the restoration of signals that are degraded by less than the error of the warping function. This

was achieved by combining the target derived filters and the warped prototype filter. To avoid repetition of results, cases for when the inverse warping function is used and the prototype and target signals are reversed have not been included in this chapter, although it is useful to note that the conclusions are the same.

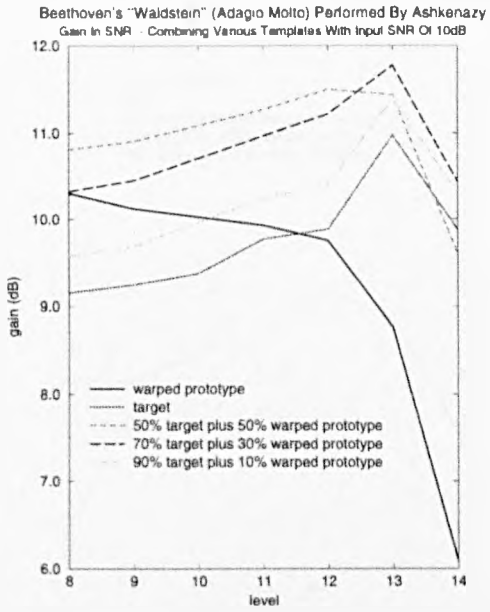

Figure 5.25: Beethoven's "Waldstein" performed by Ashkenazy (Adagio Molto) with an input SNR of 10dB, showing the gains for signals restored using a combination of warped prototype templates and target derived templates.

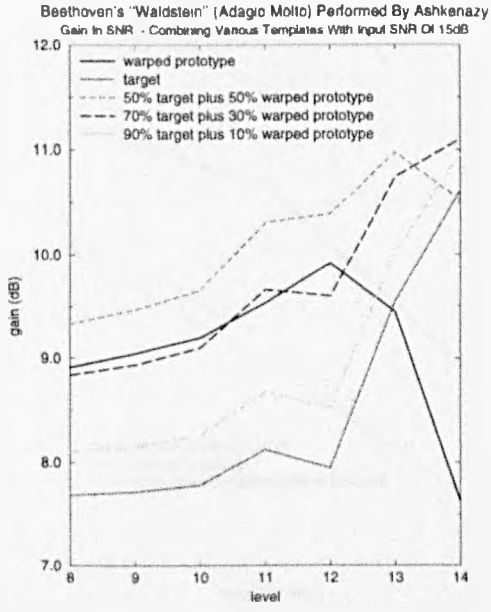

Figure 5.26: Beethoven's "Waldstein" performed by Ashkenazy (Adagio Molto) with an input SNR of 15dB, showing the gains for signals restored using a combination of warped prototype templates and target derived templates.

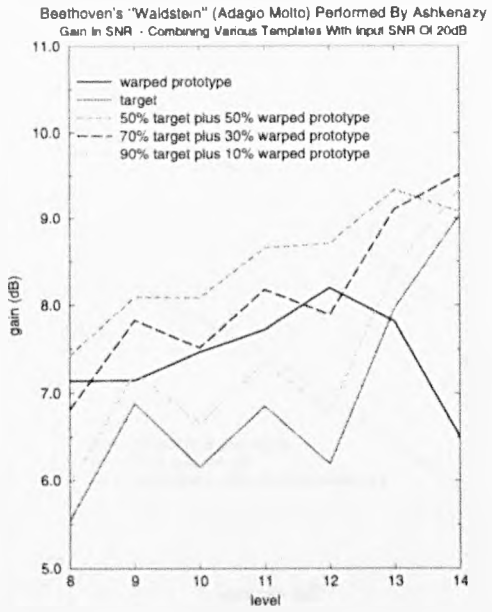

Figure 5.27: Beethoven's "Waldstein" performed by Ashkenazy (Adagio Molto) with an input SNR of 20dB, showing the gains for signals restored using a combination of warped prototype templates and target derived templates.

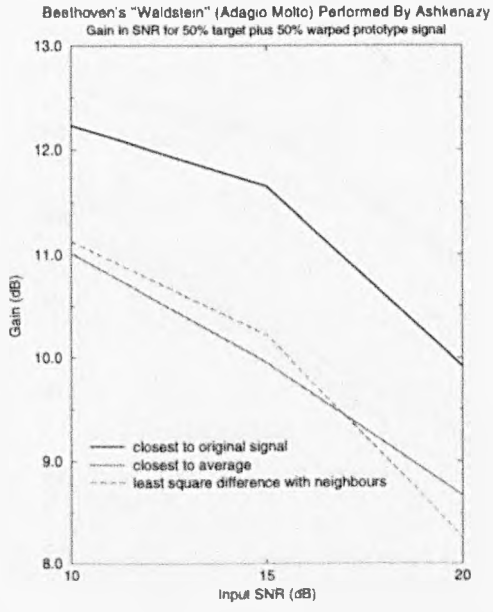

Figure 5.28: Gain against input SNR using multiresolution enhancement methods for Beethoven's "Waldstein" performed by Ashkenazy (Adagio Molto) with 50% target derived signal and 50% warped prototype derived signal.

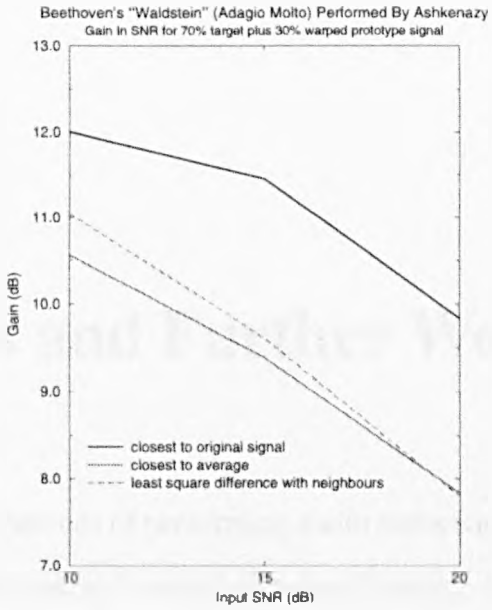

Figure 5.29: Gain against input SNR using multiresolution enhancement methods for Beethoven's "Waldstein" performed by Ashkenazy (Adagio Molto) with 70% target derived signal and 30% warped prototype derived signal.

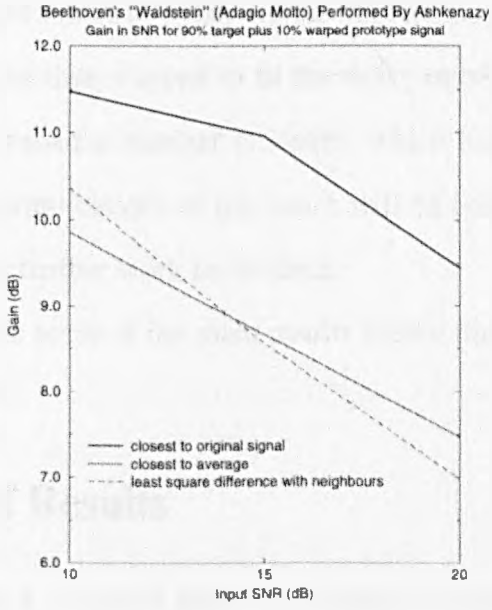

Figure 5.30: Gain against input SNR using multiresolution enhancement methods for Beethoven's "Waldstein" performed by Ashkenazy (Adagio Molto) with 90% target derived signal and 10% warped prototype derived signal.

Chapter 6

Conclusions and Further Work

This thesis has described methods of performing audio restoration using a generalised WT (the MFT). This has demonstrated the advantages of using a multiresolution approach, both in terms of algorithm design and implementation, and benefits to the gain in SNR of the restored signal. In so doing, a method of warping the features of one performance to coincide with the features in another was described. The initial method of restoration used a simple and efficient adaptive filter, which filtered the audio signal in the time-frequency domain, this was then warped to fit the noisy musical performance or target signal. This process has raised a number of issues, which have been addressed in this work. In conclusion, the implications of this work will be considered in a wider sense, along with suggestions for further work in the field.

In the following section some of the main results shown throughout this work will be summarised for the reader.

6.1 Summary of Results

All of the results in tables 6.1 and 6.2 have been shown previously in sections 3.10, 5.7 and 5.8.1. The numbers shown are the gain in SNR upon filtering using the various

"Waldstein" (Adagio)				
Input SNR	*Template*	*Warped Template*	*Adaptive Wiener*	*Target/Prototype*
10dB	15.21dB	10.56dB	11.02dB	12.22dB
15dB	13.89dB	9.91dB	9.73dB	11.65dB
20dB	12.53dB	8.19dB	7.83dB	9.92dB

Table 6.1: A brief summary of some main results for Beethoven's "Waldstein" (Adagio Cantabile).

String Quartet In Gmaj (Scherzo)			
Input SNR	*Template*	*Warped Template*	*Adaptive Wiener*
10dB	11.08dB	7.59dB	6.66dB
15dB	9.18dB	5.96dB	5.73dB
20dB	7.95dB	3.88dB	4.11dB

Table 6.2: A brief summary of some main results for Beethoven's String Quartet no 2 in Gmaj (Scherzo)

methods described. The column headed "Template" gives the results for templating with *a priori* information on the clean signal in section 3.1. "Warped Template" and "Adaptive Wiener" are the results from tables 5.1 and 5.2 in section 5.7 using the adaptive Wiener filter method and simple adaptive filter with warping. The "Target/Prototype" header gives the results for the best combination (p=0.5) of the target and prototype templates in section 5.8.1. Note that it is difficult to compare these results with those of other audio restoration methods as no numerical results are given in the literature. However, it is worth noting that the Adaptive Wiener method implemented in section 5.7 is very similar to the method proposed by Vaseghi [58] [59].

6.2 Filtering Degraded Audio Signals

In chapter 2, time-frequency representations of audio signals were considered as tools with which to perform audio restoration. In some previous work in this field [59] [58]

[6] [29], the problem of restoring degraded audio signals was approached in the time domain. In [6] [11] [22], for example, the approach taken was purely that of intervening manually in the time-domain structure of audio signals: altering them to sound better. The advantages of using a time-frequency representation such as the MFT were discussed and, after considering the most widely used methods of audio signal analysis, the MFT was shown to be the most general available. It allowed the free variation of scale with the signal features — high frequency resolution of steady-state features such as decay and high time resolution of transient features such as note onsets. This is a freedom not available to users of other time-frequency analysis tools such as the STFT, where the scale is fixed by the window size or the WT where the scale varies as a function of frequency but is fixed by the choice of wavelet. The MFT has been used before in audio signal processing by Pearson [45] [66] for note transcription. In this thesis an efficient and simple implementation of the MFT, which is not limited in the size of time signal that it could process, was introduced and its properties discussed.

In chapter 3, a simple adaptive filter was introduced. The motivation underlying this filter was that of simplicity and integrity. Instead of changing the values of the MFT domain coefficients that contained the audio signal information, a simple decision was made as to whether something was structured and therefore signal, or not and therefore assumed to be noise. This filter is simple because of its low computational complexity — $O(N \log^2(N))$ — and because of the smaller amount of storage space required when compared to any other adaptive filter, such as an adaptive Wiener filter [59]. Such a Wiener filter has complex coefficients $H(\omega)$ for each frequency value for each time, whereas in the simple adaptive filter coefficients are binary and can therefore be stored cheaply. The integrity of this adaptive filter arises from the premise that ideally there should be no alteration in the signal values — the noise should be removed but the signal should be

retained. This works in practice because although there is noise in the coefficients, the signal energy masks it [55] [37]. It was shown in chapter 3 that this filter worked well when the signal spectrum was known *a priori* and also in chapter 5 when the adaptive filter was derived from a prototype, warped and then applied to the noisy signal.

The problem of audio restoration can be approached in two equivalent ways: estimating the noise spectrum to be removed [58] [39], or estimating the signal spectrum to be retained [34] [36] [17] [18]. Once one of these is known then the other can be derived. In this work it was assumed that a clean spectral estimate was available, using the warping functions determined in chapter 4, from an uncorrupted recording of the same piece of music as the noisy signal. In choosing the threshold value for the adaptive filter, it was shown that the threshold was a function of the input SNR. A simple method was employed that allowed the amount of noise to be estimated by assuming that it was broadband — a more general assumption than that made in, for example, [36] where the noise values are assumed to have a Gaussian density. In this work, the assumption of a broadband noise distribution was used to give an accurate estimate of the input SNR, and hence the adaptive filter's threshold. An analytical method for choosing this threshold was discussed in section 3.8 and preliminary results were presented with some success. Thus the template filter uses both estimates of the signal spectrum, from the clean recording, and an estimate of the noise variance found by assuming that such a spectrum is flat, to calculate the template threshold. Once the threshold is chosen, features are detected if they span more than one (generally two) frequency bins for some length of time. This thesis has shown a method that combines both noise and signal spectral information using a musical signal model to good effect.

Most audio restoration techniques rely on the stationarity — over some time period — of the audio or musical signals for their enhancement [17] [58] [59] [36] [34] [18] [53].

This requires windowing in the time domain to isolate the period of time over which the signal is stationary. If this window is too wide, then the transients, or onsets, are missed or smeared, whereas if the window is too small any estimates suffer from high variability. This motivates the use of the MFT for audio restoration. Because of the range of scales available to the MFT user, an adaptive filter can be obtained which can be matched to both transients and steady-state periods in most musical pieces [64]. In practice the scale is varied in the time domain for ease of implementation and to avoid artifacts. To identify the regions of transient or steady-state features, a note onset detector was introduced in chapter 3. The motivation for this was simply that transients in musical signals will occur at the note onset, when the spectrum of the note changes the most rapidly in time. The onset detector split the time-domain of the signals into segments that were labelled either onset or steady-state. In chapter 3, where the signal was known *a priori*, the SNR of each segment of enhanced signal was used to determine the best MFT level or scale. In chapter 5, however, two heuristic methods were used because the uncorrupted signal was unavailable. It was concluded that the more reliable method was the one that chose the enhanced signal closest to the average of all the enhanced signals for that time frame.

If the audio signal is stationary over short periods of time only then these short periods taken together constitute a non-stationary system, the "state" of the audio signal, that alters from period to period. Furthermore, if we assume that each of these states is degraded by noise, then one method for restoring such a signal would be to use Kalman filtering as discussed briefly in chapter 1 [35] [10]. A Kalman filter is one designed to use the statistical information present in the changes of state in a non-stationary system, the degraded signal, to build an increasingly accurate picture of the uncorrupted signal. The problem with using such a method is the decision as to what constitutes a state: would such states require spectral information, how long would each state last and, since musical

signals are being considered here as multiresolution, how would the scale of such states be altered as a function of time? These questions are difficult to answer at the present time. The use of a Kalman filter would become simpler if a higher level model of the signal and its structure was being used, such as the representation employed by Pearson which used a Markov model of the musical notes [45], where the information could be manipulated more effectively. However, within the stationary periods Wiener filtering can be employed. Moreover, as the stationary states change, a time varying Wiener filter, one whose values alter as a function of time, can be introduced.

The adaptive filter was compared with a variety of methods including a lowpass filter, a stationary Wiener filter and, after the warping algorithm had been introduced, a time varying Wiener filter. The latter used warping of the clean prototype signal for its spectral estimate. The first two methods of filtering were shown to be ineffective tools for audio restoration compared with the simple adaptive filter — the lowpass filter rendered the restored signals dull, while the stationary Wiener filter performed badly because it was estimating over too long a time period, and consequently made an inaccurate estimate of the signal spectrum. The adaptive Wiener filter was implemented simply using the MFT. The time-varying Wiener filter used by Vaseghi relied on there being two recordings of the same piece available from which to estimate the local signal spectrum and the local noise spectrum [59]. This was done in the time domain, with a 20ms FIR filter that was varied so as to maximise the correlation between one noisy recording and an appropriately delayed second noisy recording of the same piece (delayed at most by one filter length). This caused a number of implementational problems not present in this work. The first problem is the assumption that two noisy recordings, with a variation in time scale always less than 20ms, are available. This implies that these would have to be two records that are identical apart from being degraded differently. The implementation

by Vaseghi is further complicated by the fact that it is based on the time domain signal. Versions were implemented that relied on Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithms to match the two signals and hence estimate a local signal spectrum. The LMS method was found to be numerically unstable, not always guaranteeing the convergence of the two recordings, whereas the RLS method was found to be computationally intensive and very sensitive to non-Gaussian noise. Because the MFT is used in this thesis, once the local noise and signal spectra are known, implementing a Wiener filter becomes equivalent to weighting each of the coefficients on one level of the MFT of the noisy signal appropriately. The results for enhancing degraded audio signals using the warped adaptive filter and the warped adaptive Wiener filter — which requires more storage and computation — were shown to be comparable. Indeed, only one instance was found where the Wiener filter performed better than the simple adaptive filter. It can be concluded that using the simple, computationally efficient adaptive filter is as good as the adaptive Wiener filter.

In terms of facilitating the filtering of degraded musical signals it has been shown in this thesis that the MFT allows not only the use of the simple adaptive filter introduced in this work, but allows a simple implementation of the adaptive Wiener filter. It can be concluded therefore that the MFT is a very useful tool with which to analyse and restore musical signals.

6.3 Warping

When a filter derived from one performance is to be used to restore a second performance, a method of a local stretching of the time axis of the former to fit the time scale of the latter is required. The warping algorithm in chapter 4 finds the best piecewise linear approximation to the generally continuous warping function. To place this algorithm in a

wider context, it is necessary to look further afield than the warping of audio signals. There is very little activity in the field of audio signal warping [59] [3] and so it is to the area of speech recognition that one is drawn for comparison. The Dynamic Time Warping (DTW) algorithm favoured in many speech recognition systems [23] [27] [47] [40] assumes the principle of optimality [13] [60]. Once the end-points of the warp path have been found, the optimal warp path is found by back-tracking to the beginning, using some distance measure to constrain the warping. The warping algorithm presented in this work allows a more general approach to the warping of an audio signal by allowing more than one measure to find the best warp path. In this work the prototype is warped linearly so that both the target and the prototype signals have the same length. Next, the candidate break points are found in the signal using musical feature matching, based on peaks in the time profile of the signal energy. Once these have been identified they are further refined using a correlation measure. This gives a more accurate warping function. This is in contrast to the warping methods utilising the DTW approach, which usually uses one metric with which the accuracy of the warping function at each choice of warp path co-ordinate is gauged. One main difference between warping audio signals and warping speech signals is the length of the signal segments — in speech recognition a word will last at most one or two seconds, but musical performances last orders of magnitude longer. Audio signal "warping" was employed in the work by Vaseghi [59] [58]. This was warping in the broadest sense: one signal was delayed relative to another by a maximum of 20ms. The amount of delay was found by using a least squares algorithm that maximised the correlation between target and prototype signals. It is difficult to see how such a method of warping could be applied to two different performances of the same piece where it has been shown that the delay is regularly greater than 20ms. Indeed it is not difficult to tell that one performance is slower than another simply by listening. Furthermore,

the warping algorithm in this work uses a feature model that is robust in additive white noise. It has been shown to work in the application of audio restoration, where the simple adaptive filter was warped and used successfully to restore degraded signals. In chapter 5 the effects of enhancement on the target signal using a template derived from the prototype signal that was not warped to match the target signal were shown to be significantly worse than the enhancements gained by using a warped one. Furthermore, the warping function was shown to be successful in spectral matching for use in the adaptive Wiener filter.

It can be concluded that the warping algorithm presented here can successfully match audio signals that have large discrepancies in their time scales by using piecewise linear approximations to the continuous warping function and a recursive binary search. It avoids some of the assumptions behind DTW and is more general than the warping method employed by Vaseghi. For any given degraded recording of a classical piece, a clean modern recording will be available, if there is none immediately available an artist can be employed to produce one.

6.4 Suggestions for Further Work

This thesis has displayed and investigated the general principles of audio restoration using a multiresolution time-frequency representation. However, in order to produce an automated restoration system further work is, however, required.

In chapter 4, the end points of the warping function were given as initial conditions to the warping algorithm. Obviously if these are in error, then the warping function will not be correct: the algorithm has no capacity to recover from bad initial conditions. If, however, the algorithm could be generalised to search for these starting points, then the problem of finding the correct start and end points manually would, by definition, disappear. One method used in speech recognition for finding the endpoints for a warping

function is to find points between words [47]. This may be possible in musical signal processing, since some musical signals will have silences between movements or musical phrases. A more general method could be employed to find the initial breakpoints of faster sections, such as the String Quartet Scherzo and "Waldstein" Allegro used as examples in this work. This method could rely on estimating the beat or tempo of the performance to find roughly where the start and end points are and then, using a similar refinement process to that used for the candidate warp points, vary the start and end point of one piece until the maximum correlation is found.

In chapter 5 it was noted that the differences in the results from the two Adagio String Quartet pieces was caused by the varying amounts of vibrato between the two performances. This was noted visually, but is also audibly perceptible. If the adaptive filter's structure is likened to a set of narrowband filters, it would be possible to use this visualisation to vary the structure of the narrowband filters sinusoidally, with those filters occurring at a higher frequency having greater variation than those at lower frequencies. Applying this to different sinusoidal perturbations to the filterbank structure would cause the adaptive filter derived from one piece to emulate the vibrato — if different — in another, capturing more of the signal and thus increasing the enhancement of the degraded target signal. This would require a more sophisticated form of frequency warping than that used in this thesis.

In chapter 5, methods for combining audio signals enhanced using different levels of the MFT on the time-frequency plane across scale were suggested. Although the method of averaging across levels was shown to be the most consistent performer of the two methods given, it was not wholly satisfactory. When compared to the *a priori* combination of levels in chapters 3 and 5, it is evident that the averaging across levels can be improved upon. This method can be generalised to use some linear combination of the enhanced signals

other than the average. Other weightings are possible, for example a weighting could be used that depended on whether the variance or energy concentration of those signals *suggested* a low or a high MFT level. Another such weighting could give low MFT levels a higher weighting for onset segments and, conversely, high levels a higher weighting for steady-state segments. This might be advantageous as the transients should occur at the note onsets.

A preliminary method of automatically choosing the global threshold ν was introduced in section 3.8 which relied on modelling both the noise and signal's energy distribution with a Cauchy distribution curve. This could be further improved for Chapter 5 if the choice of ν somehow depended on the accuracy of the warping function.

In chapter 3 an adaptive filter was derived from a noisy target signal and results for restoration were shown. In chapter 5 the signals enhanced using such a target derived filter and the signals enhanced using a warped prototype filter were combined. Obviously, as the SNR increases then the signal enhanced using a target derived filter will become more accurate and the warped prototype's filter less so. If the prototype is clean then, even if the warping were perfect, the best that the warped filter could do would only ever be as good as the target derived one. Hence using a combination of the two would sufficiently generalise the process of restoring noisy signals to allow the restoration of both extremely noisy signals and signals only slightly degraded by noise.

It is assumed that the noise present in the signal coefficients retained by the adaptive filter will masked by the noise [39] [37] [55]. However, no attempt has been made here to exploit the phase properties of the MFT coefficients, as was done in Pearson's work. These matters deserve further consideration.

6.5 Concluding Remarks

This thesis has shown that the MFT is a versatile tool with which to tackle the musical audio signal restoration problem. A computationally efficient and easily stored adaptive filter has been introduced with which to perform audio restoration. A method for describing the difference in time scales of two performances of the same piece of music has also been introduced. This warping function facilitates the application of the adaptive filter to a noisy signal without requiring estimates of the noisy signal's spectrum. It has been shown that the filter works with impulse noise as well as the white additive Gaussian noise used in the examples. Because of the versatility of the MFT, an adaptive Wiener filter, similar to that used by Vaseghi [59], was implemented simply. It was shown that the simple adaptive filter was better in all but one out of six cases where these two filters were compared. This showed the simple adaptive filter to be as good as the adaptive Wiener filter, but saving greatly on storage space required. Filters derived from the target signal and prototype signal were combined linearly in section 5.8 to improve the performance of the simple adaptive filter in low noise. Finally it can be said that the MFT, a generalised WT [66], has been shown to be a time-frequency representation that can be used for implementing restoration algorithms simply and to good effect.

Bibliography

[1] Alfred V Aho, John E Hopcroft, Jeffrey D Ullman, *"Data structures and algorithms"*, Addison-Wesley, USA, 1983.

[2] R E Allen, *" The pocket Oxford english dictionary"*, Clarendon Press, Oxford, 1984.

[3] Daniel Arfib and Nathalie Delprat, *"Musical transformations using the modifications of time-frequency images"*, The Computer Music Journal, 17(2), 1993.

[4] Daniel Arfib, *"Analysis, transformation, and re-synthesis of musical sounds with the help of a time-frequency representation"* in *"Representations of musical signals"* pp87-118, editors, Giovanni De Poli, Aldo Piccialli, Curtis Roads, The MIT Press, Cambridge Ma, 1991.

[5] D Arnold, *"The new Oxford companion to music"*, Oxford University Press, Suffolk, 1994.

[6] Roy S Brink Jnr, *"Empirical methods in restorative processing of historical recordings"*, Proceedings of the 92nd Audio Engineering Society Convention, Vienna, 1992.

[7] R Bellman, *"Dynamic programming"*, Princeton University Press, Princeton, New Jersey, 1957.

[8] Steven F Boll, *"Suppression of acoustic noise in speech using spectral subtraction"*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol ASSP-27(2), April 1979.

[9] Andrew Calway, *"The multiresolution Fourier transform: a general purpose tool for image analysis"*, PhD thesis, Department of Computer Science, The University of Warwick, UK, September 1989.

[10] James V Candy, *"Signal processing"*, McGraw-Hill International, Singapore, 1988.

[11] Krysztof Cisowski, *"Efficiency of impulsive noise detection in audio recordings using the adaptive filtering method"*, Preprint 3464 (B1-4) from The Proceedings of the 94th Audio Engineering Society Convention, Berlin, 1993.

[12] R R Coifman, M V Wickerhasuer, *"Entropy based algorithms for best basis selection"*, IEEE Transactions on Information Theory, 38(6) No 2 pp713-319, 1992.

[13] L Cooper, W Cooper, *"Introduction to dynamic programming"*, Pergamon Press, Hungary, 1981.

[14] I Daubechies, *"The wavelet transform, time-frequency localisation and signal analysis"*, IEEE Transactions on Information Theory, Vol 36 pp961-1005, 1990.

[15] I Daubechies, *"Ten lectures on wavelets"*, SIAM Press, Philadelphia, 1992.

[16] Andrew R Davies, *"Image feature analysis using the multiresolution Fourier transform"*, PhD thesis, Department of Computer Science, The University of Warwick, UK, August 1993.

[17] W A Deutsch, A Noll, *"Restoration of historical recordings by means of digital signal processing"*, Preprint 2091(H2) from The Proceedings of the 75th Audio Engineering Society Convention, Paris, 1984.

[18] Y Ephraim, D Mallah, *"Speech enhancement using a minimum mean-square short-time spectral amplitude estimator"*, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP 32(6), December 1984.

[19] D Gabor, *"Acoustical quanta and the theory of hearing"*, Nature, No 4404 pp591-594, May 1947.

[20] D Gabor, *"Theory of communications"*, Proceedings of the IEE, Vol 9(3) pp429-441, 1946.

[21] W H Hays, R L Winkler, *"Statistics:probability, inference, and decision"*, Holt, Rinehart and Winston Inc, USA, 1971.

[22] F L Hui, W H Lau, *"The removal of impulse noise in music signals using higher order spectra"*, Preprint 3846 (P11.6) from The Proceedings of the 96th Audio Engineering Society Convention, Amsterdam, 1994.

[23] Fumitada Itakura, *"Minimum prediction residual applied to speech recognition"*, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-23(1), 1975.

[24] Russel Kahl (editor), *"Selected writings of Hermann von Helmholtz"*, Wesleyan University Press, Middletown, Connecticut, 1971.

[25] R Kronland-Martinet, *"The wavelet transform for analysis, synthesis, and processing of speech and music sounds"*, Computer Music Journal, Vol12 (4) pp11-19, 1988.

[26] Murat Kunt, *"Digital Signal Processing"*, Artech House, Norwood MA, 1986.

[27] Stephen E. Levinson, *"A unified theory of composite pattern analysis for automatic speech recognition"*, pp243-275 Computer Speech Processing, J Vaissière editor, Prentice Hall, Exeter, 1985.

[28] Jae S Lim, *"Speech enhancement"*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[29] Jae S Lim, Alan V Oppenheim, *"Evaluation of an adaptive comb filtering method for enhancing speech degraded by white noise addition"*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-26(4), August 1978.

[30] Jae S Lim, Alan V Oppenheim, *" Enhancement and bandwidth compression of noisy speech"*, Proceedings of the IEEE, Vol 67(12), December 1979.

[31] R J McAulay, M L Malpass, *"Speech enhancement using a soft-decision maximum likelihood noise suppression filter"*, Technical Report 1979-31, MIT Lincoln Lab, Lexington, MA, June 1979.

[32] Peter Manning, *"Electronic and computer music"*, Clarendon Press, Oxford, 1987.

[33] P M Mathews, K Venkatesan, *"A textbook of quantum mechanics"*, Tata McGraw-Hill, New Delhi, 1988.

[34] Thomas E Miller, Jeffrey Barish, *"Optimizing sound for listening in the presence of Road Noise"*, Preprint 3760 (B3-PM-8) from The Proceedings of the 95th Audio Engineering Society Convention, New York, 1993.

[35] Richard E Mortensen, *"Random signals and systems"*, John Wiley & Sons, Singapore, 1987.

[36] S Montresor, J C Valière, J F Allard, M Baudry, , *"The Restoration of old recordings by means of digital techniques"*, Preprint 2915 (G4) from The Proceedings of the 88th Audio Engineering Society Convention, Montreux, 1992.

[37] F Richard Moorer, *" Elements of computer music"*, Prentice Hall, New Jersey, 1990.

[38] James A Moorer, *"On the segmentation and analysis of continuous musical sound by digital computer"*, report no STAN-M-3, Center for computer research in music and acoustics, Stanford University, Palo Alto, USA, 1975.

[39] J Mourjopoulos, G Kokkinakis, M Paraskeyas, *"Noisy audio signal enhancement using subjective spectra"*, Preprint 3240 from The Proceedings of the 92nd Audio Engineering Society Convention, Vienna, 1992.

[40] C S Myers, L R Rabiner, A E Rosenberg, *"Performance tradeoffs in dynamic time warping algorithms for isolated word recognition"*, IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-28, December 1978.

[41] Alan V Oppenheim, Ronald W Schafer, *"Discrete-Time Signal Processing"*, Prentice-Hall, New Jersey, 1989.

[42] Carlos Palombini, *"Machine songs V: Pierre Schaeffer — from research into noises to experimental music"*, Computer Music Journal 17(3) pp14-19, 1993.

[43] Athanasios Papoulis, *"Signal analysis"*, McGraw-Hill, Singapore, 1977.

[44] Theo Pavlidis, *"Curve fitting as a pattern recognition problem"*, Proceedings of the 6th International Conference on Pattern Recognition, Munich, 1982.

[45] Edward R.S. Pearson, *"The multiresolution Fourier transform and its application to polyphonic audio analysis"*, PhD thesis, Department of Computer Science, The University of Warwick, UK, December 1992.

[46] Michael R Portnoff, *"Time-frequency representation of digital signals based on short-time Fourier analysis"* IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-28(1), 1980.

[47] Lawrence Rabiner, Stephen Levinson, *"Isolated and connected word recognition — theory and selected applications"* IEEE Transactions on Communications COM-29(5), 1981.

[48] O Rioul, M Vetterli, *"Wavelets and signal processing"*, IEEE Signal Processing Magazine, pp14-38, October 1991.

[49] Curtis Roads, *"Granular synthesis of sounds"*, Foundations of Computer Music, MIT Press, 1985.

[50] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery *"Numerical recipes in C"* Cambridge University Press, Cambridge 1992.

[51] Hugh Scott, *"A comparison of filters for audio signal segmentation in audio restoration"* Technical Report RR231, Department of Computer Science, The University of Warwick, 1992.

[52] Xavier Serra, *"A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition"*, PhD thesis, Technical Report STAN-M-58, The Department of Music, Stanford University, 1989.

[53] L Simon, J C Valière, C Depollier, *"New contribution on noise reduction using wavelet techniques: application to the restoration of old recordings"*, Preprint 3461 (B1-1) from The Proceedings of the 94th Audio Engineering Society Convention, Berlin, 1993.

[54] David Slepian, *" On bandwidth"*, Proceedings of The IEEE, pp292-300, 64, 1976.

[55] Thomas Sporer, Holger Schröder, *"Measuring tone masking noise"*, Preprint 3349 (A-2) from The Proceedings of the 93rd Audio Engineering Society Convention, San Francisco, 1992.

[56] Ivan Tomek, *"Two algorithms for piecewise-linear continuous approximation of functions of one variable"* IEEE Transactions on Computers, C-23, 1978.

[57] J Vaissière, *"Speech recognition: a tutorial"*, pp191-242 Computer Speech Processing, J Vaissière editor, Prentice Hall, Exeter, 1985.

[58] S V Vaseghi, R Frayling-Cork, *"Restoration of old gramophone recordings"*, Journal of The Audio Engineering Society, 40(10) pp791-801, 1992.

[59] S V Vaseghi, *"Algorithms for restoration of archived gramophone recordings"*, PhD thesis, The Department of Engineering, The University of Cambridge, UK, February 1988.

[60] D White, *"Dynamic Programming"*, Aberdeen University Press, Aberdeen, 1969.

[61] M R Weiss et al, *"Processing speech signals to attenuate interference"*, presented at the IEEE Symposium on Speech Recognition, April 1974.

[62] M R Weiss, E Aschkenasy, T W Parsons, *"Study and development of the INTEL technique for improving speech intelligibility"*, Nicolet Scientific Corp., Report NSC-FR/4023, December 1974.

[63] K Weltner, J Grosjean, P Schuster, W J Webster, *"Mathematics for scientists and engineers"*, Stanley Thornes, Cheltenham, 1986.

[64] Roland Wilson, *"Scale in transient detection"*, Technical Report 18/1993, IMSOR, The Technical University of Denmark, 1993.

[65] Roland Wilson, Goesta H Granlund, *"The uncertainty principle in image processing"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (PAMI-6(6)) pp758-767, 1984.

[66] Roland Wilson and Andrew D. Calway and Edward R.S. Pearson, *"A generalised wavelet transform for Fourier analysis: the multiresolution Fourier transform and its application to image and audio signal analysis"*, IEEE Transactions on Information Theory, 38(6) No 2 pp674-691, 1992.

[67] Roland Wilson, *"Finite prolate spheroidal sequences and their applications 1: generation and properties"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol PAMI-9(6) pp787-795, 1987.