# Sparse Denoising of Audio by Greedy Time-Frequency Shrinkage

*Gautam Bhattacharya*

Department of Music Research
Schulich School of Music, McGill University
Montreal, Canada

March 2014

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Arts.

# Abstract

Matching Pursuit (MP) is a greedy algorithm that iteratively builds a sparse signal representation. This work presents an analysis of MP in the context of audio denoising. By interpreting the algorithm as a *simple shrinkage* approach, we identify the factors critical to its success, and propose several approaches to improve its performance and robustness. We also develop several model enhancements and introduce an audio denoising approach called *Greedy Time-Frequency Shrinkage* (GTFS). Numerical experiments are performed on a wide range of audio signals, and we demonstrate that GTFS denoising is able to yield results that are competitive with state-of-the-art audio denoising approaches. Notably, GTFS retains a small percentage of a signal's transform coefficients for building a denoised representation, i.e., it produces very *sparse* denoised results.

# Résumé

L'algorithme de *Matching Pursuit* (MP) construit par itérations une représentation parcimonieuse du signal, au prix d'un coût de calcul élevé. Ce mémoire présente une analyse de l'algorithme de MP dans le contexte du débruitage audio. En interprétant l'algorithme MP comme une méthode de contraction simple (*simple shrinkage*), nous chercherons à identifier les facteurs essentiels à son succès, puis proposerons plusieurs approches afin d'en améliorer les performances et la robustesse. Plusieurs améliorations du modèle seront ainsi développées, et une approche du débruitage audio dénommée *Greedy Time-Frequency Shrinkage* (GTFS) sera présentée en détails. Des expérimentations numériques appliquées à un large éventail de signaux sonores démontrent que les résultats obtenus par débruitage GTFS s'avèrent compétitifs face aux méthodes de débruitage audio qui constituent l'état de l'art. En particulier, le GTFS ne retient qu'un faible pourcentage des coefficients de la transformée du signal pour en construire une représentation débruitée, et produit ainsi des résultats débruités très compacts.

# Acknowledgments

I would like to extend my sincerest thanks to my thesis advisor, Prof. Philippe Depalle. His insight, passion for his subject, and problem-solving approach have been an invaluable asset to me over the course of my Master's degree.

My thanks to everyone at the Music Technology department at McGill for making my time here a truly memorable experience. The opportunity to interact with researchers specializing in diverse disciplines served as a great source of inspiration throughout my own research.

Last but not least, I would like to acknowledge my biggest supporters, my family. My mother, whose persistent motivation and encouragement has been a driving force in my life. My father, to whom I have always turned to for advice and guidance. I thank my parents for all their love and support, which has allowed me to pursue my passions and interests.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Overview

## 1.1 Here Comes the Noise

Noise is an unpredictable, ubiquitous and often inevitable phenomenon present in the natural environment we live in. So much so, that human beings are adept at communicating in the presence of noise. The growing prevalence of digital communication devices makes the problem of audio noise reduction an active and important field of research. Noise can be of several types, including coloured, correlated, multiplicative and impulsive noise. However, a large percentage of noise reduction research has been focussed on the removal of *uncorrelated Gaussian or white noise.* This kind of noise is present in several speech communication systems, and manifests itself in musical recordings in the form of background 'hiss'. Researchers have noted that additive white noise is amongst the hardest to suppress [2]. Noise reduction solutions have been used to enhance several applications, including cellular phone systems, teleconferencing, noise cancellation, audio restoration and speech enhancement. Many of these applications are associated with specific noise reduction problems, such as suppressing different kinds of noise. As a result, most noise reduction approaches are application specific, and it has been pointed out that designing a general-purpose noise reduction solution that works well in all situations is extremely difficult – if not impossible. The classical (and not so classical) approaches to audio noise reduction involve transforming the noisy sound to the frequency domain and then figuring out a clever way to keep only the non-noisy components of the sound [3]. A fundamental assumption in all these denoising methods is that the meaningful parts of the sound are concentrated on a relatively small number of frequency components, and one can hence perform noise reduction on the sound in the frequency domain while leaving the meaningful part of the sound largely unaffected. In the context of this work, it might be useful to interpret noise reduction algorithms as approaches that compress or *sparsify* the time-frequency representation of a signal. By this we mean that a noise reduction algorithm will try to reduce or attenuate the noisy signals transform coefficients to zero (or to a very low value). This interpretation forms the basis of signal denoising based on L1-norm relaxation [4].

An important aspect of the denoising approaches developed in this research is that they do not remove noise from a signal in accordance with the classical approach. Firstly, classical signal denoising involves transformation of the noisy signal to the frequency domain using an orthogonal basis. In the context of signal denoising, orthogonality implies that white noise in the time domain is converted to white noise in the frequency domain. This greatly simplifies the design of statistical signal smoothing and attenuation approaches [5]. In this research we will *not* work with orthogonal basis. Rather we make use of structures known as *redundant dictionaries* for signal transformation. Researchers have shown that while orthogonality offers statistical advantages, it is not an essential condition as far as signal denoising is concerned [6]. Secondly, unlike the classical approach, we do not identify the noise and attenuate it. Instead we try to recover the 'meaningful' part of the sound directly from the noisy audio. That is, we attempt to sense or recover the deterministic audio content. While we do not expect the denoising approaches developed in this research to outperform existing (classical) approaches, we hope to produce results of comparable quality. We hypothesize that attempting to denoise a signal by recovering the deterministic content is more in agreement with the noise reduction mechanism of the human auditory process. Human beings do not perform noise reduction as a post-processing activity. Instead we actively process the noisy signal and try to retrieve as much useful information from it as possible. We are thus hopeful that this research can find application in areas related to hearing and psychoacoustic research, apart from more traditional noise reduction tasks. Another aspect of the approaches developed in this work is their sparse behaviour. As mentioned earlier, there is a strong link between noise reduction and signal sparsity. Through our experiments we will show that it is possible to sparsely recover sounds in varying noise conditions. Notably, we will show that the denoising approach developed in this research is able to produce denoising results that are comparable to those of state-of-the-art denoising algorithms. In this research we will explore audio denoising in the context of musical audio and speech. The unpredictable nature of these signals make noise reduction a challenging problem, and any a priori information about the sounds being analyzed can potentially improve the performance of a denoising algorithm.

## 1.2 Sounds & Music

Over the years noise reduction has become an essential part of communication devices such as smart phones, as well as musical audio applications such as noise cancellation headphones. In order to represent audio signals in the transform or frequency domain, it is useful to study their lower level structures.

- **Musical Audio:** At the note level, musical sounds are typically characterized by short transient component corresponding to the attack of the note, followed by a sustained tonal part and finally a release segment. Transient-Steady State separation (TSS) is a common first step in audio compression and audio manipulation / synthesis applications [7]. Apart from the tonal and transient components, most musical sounds are also associated with some form of noise – for example, breath noise in a flute sound. From a noise reduction point of view, this type of noise is 'good' noise and ideally should not be suppressed by the noise reducing algorithm. From our experiments we have noticed that the transient components of a musical sound are more difficult to accurately recover in a noisy setting. This is because transient audio components have a short time support and a proportionately wideband frequency support. As a result, the frequency content of an audio transient can be quite similar to that of a noisy sound component, and hence discriminating between the two can be challenging. Noise reduction approaches for musical sounds generally fall into one of two categories – digital audio restoration and adaptive noise cancellation. In this work we will be concerned with the former.

- **Speech:** Noise reduction for speech, or speech enhancement is a very active field of research. Speech can be modelled using a sinusoidal representation [8] or a source-filter approach [9]. Other approaches based on Gammatone and Gammachirp filter banks have also been successfully applied to speech coding. Consequently, speech enhancement algorithms have made use of these different models [10, 11, 12, 13], several of which have been adapted successfully for musical audio restoration tasks.

## 1.3 Sparse Representation

We describe the sparse representation problem with the help of a simple musical example. Let us consider a single note of a glockenspiel. We have a priori knowledge that the sound consists of a fundamental frequency and higher-order harmonics. That is, the information required to accurately describe the sound is concentrated around a relatively small number of frequency components. At the same time, apart from its harmonic content, the sound contains components related to the note's attack, as well as some noise associated with the recording process. As it turns out, if one attempts to account for all the different components, we end up with a system that has more unknowns than equations. This is because the total number of possible components is typically much greater than the length of the signal. Such a situation is known as an underdetermined system of linear equations. Solving such a system of equations requires a non-linear solver. More formally this can be stated as :

$$arg\ min_s\ \|\mathbf{s}\|_0\ subject\ to\ \|\mathbf{y} - \mathbf{D}\mathbf{s}\|_2^2 \leqslant \varepsilon \tag{1.1}$$

Where **s** is a vector of coefficients, $\|\mathbf{s}\|_0$ is the $l_0$ pseudo norm, which is a direct measure of the sparsity of **s**. That is, it represents the number of non-zero transform signal coefficients of $s$. **D** is known as an overcomplete or redundant dictionary, and in the context of the above example would ideally consists of elements related to the tonal and transient parts of the glockenspiel sound. In order to solve for **s**, one can make use of a greedy algorithm like Matching Pursuit [14] or an approach based on L1-norm relaxation [4]. The sparse representation problem can thus be described as an approach that attempts to accurately represent a signal with a collection of elements that numbers significantly less than the length of the signal. In this work we will focus on *greedy algorithms*. These methods have found widespread use in audio processing applications ranging from source separation to audio coding and compression [15], due to their computational efficiency and robust performance. We note that while algorithms based on L1-norm relaxation have received significant attention in the context of signal denoising, greedy methods have been comparatively ignored in this context. The combination of these factors serves as motivation for us to explore the use of greedy methods for audio signal denoising.

## 1.4 Noise Reduction

Noise reduction is a very challenging and complex problem due to several reasons. First of all, the characteristics of the noise change from application to application, and moreover, can vary with time. It is therefore extremely difficult to design a noise reduction technique that works well in different situations. Another factor that affects the design of a noise reduction approach is the specific context or application. For example, sometimes the application may call for the noise reduction algorithm to increase the intelligibility or improve the overall speech perception quality, while other situations may call for an improvement in the performance of an ASR (Automatic Speech Recognition) system - which would require a quite different noise reduction approach [16]. Below we list some of the principal factors that need to be considered while designing a noise reduction or denoising solution:

- The number of channels available for enhancement; i.e., single channel and multi-channel techniques. Understandably, more channels or measurement points leads to a simpler and more efficient noise reduction solution. In the simple case where there are two microphones available, one microphone can be used to capture a reference of the noise and an adaptive filtering solution can be applied. This approach is the basic technique utilized in noise reduction headsets [2].
- How the noise is combined with the audio; i.e., additive noise, multiplicative noise and convolutional noise. In this research we focus on additive white noise, which occurs commonly in many speech enhancement and audio restoration applications.
- Statistical relationship between the noise and the audio; i.e., uncorrelated or independent noise, and correlated noise, such as echo or reverberation. While we will deal with uncorrelated noise sources, we believe that the approach developed here will be able to perform well in the presence of correlated noise, provided the correlation is

not too high.

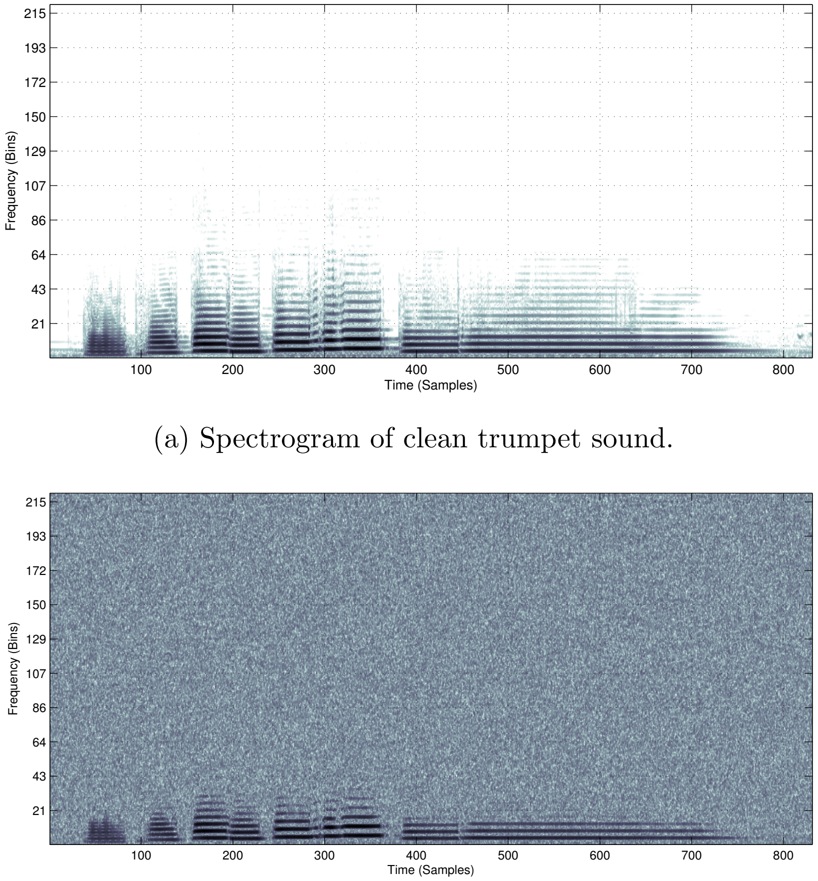
- In which domain is the processing carried out, i.e., in the time or frequency domains. While both approaches have been extensively studied over the years and have shown to produce good results, we choose to develop a frequency domain based denoising approach.

The factors mentioned above highlight the difficulty involved with designing a noise reduction solution. As a result, most, if not all noise reduction algorithms are designed for specific contexts. In this research we are concerned with noise reduction tasks in the context of audio, including digital audio restoration and speech enhancement. While several denoising approaches have been developed over the years, frequency domain or spectral methods have been the most widely used. We present the general concepts of spectral denoising techniques in the next section, as well as discussion on their application based on overcomplete signal representations.

## 1.5 Spectral Denoising

Over the years spectral domain approaches have emerged as the most popular ones for noise reduction in audio signals. These methods involve generating a time-frequency representation of the noisy sound, and then processing the coefficients of this representation to attenuate the noise. The approach takes advantage of the fact that most sounds can be represented by a relatively small number of frequency components, and hence noise reduction can be performed in the frequency domain while maintaining the relevant parts of the sound spectrum largely unaffected. The Short-Time Fourier Transform (STFT) is commonly used to generate time-frequency representations for audio signals. After the STFT coefficients have been calculated, they are attenuated by using a suppression rule. Along with classical approaches such as empirical Wiener filtering, several suppression rules have been introduced by researchers working in speech enhancement [10, 11, 12, 13]. We note that several of these methods have been successfully translated to musical audio.

Fig. 1.1 shows the effect of noise on the spectrum of a sound. The audio considered in this example is a solo trumpet recording sampled at 44100 Hz. Apart from the STFT, the orthogonal Wavelet basis can also be used to generate time-frequency signal representations.



(a) Spectrogram of clean trumpet sound.

(b) Spectrogram of noisy trumpet sound (5 dB).

**Fig. 1.1** Clean and noisy trumpet spectrograms.

Wavelets differ from the Fourier basis in that they belong to a family of time-scale atoms, as opposed to a family of time-frequency atoms. Wavelets can thus be used to perform multi-scale or multi-resolution signal analysis. This feature of the wavelet basis was used by Donoho and Johnstone to develop the wavelet threshold denoising strategy [5]. A review of Wavelet denoising is presented in Chapter 3. As mentioned previously, in this research we make use of redundant dictionaries for signal transformation. Like spectral denoising, signal representation using these structures is based on the assumption that a sound/signal can be represented by a relatively small number of components in the frequency domain. That is, dictionary based methods work on the assumption that the analysis signal is frequency sparse. In the case of signal denoising, the difficulty lies in identifying the relevant frequency components of a sound submerged in noise, and only selecting atoms that are correlated to the deterministic signal content and not the noise. Another advantage of working with redundant dictionaries is that they offer greater representational power than either of the STFT or Wavelet transform. A more detailed description and analysis of redundant dictionaries is provided in Chapter 2. Based on these factors, we are motivated to explore the use of redundant signal representations for the purpose of audio denoising.

## 1.6 Performance Metrics

Over the years several quality or performance metrics have been introduced in order to measure the quality of denoised audio. Most of these performance evaluations are based on a comparison of some sort with the original sound, i.e., the sound without any noise. In this work we will make use of two commonly used metrics, in order to collect some quantitative data regarding the performance of the various algorithms developed and tested over the course of this research.

### 1.6.1 Signal to Noise Ratio (SNR)

Signal-to-Noise-ratio (SNR) represents the distortion between the original (clean) signal, $f$, and the denoised signal, $\hat{f}$, in a norm 2 sense. The SNR of $\hat{f} \in R^p$ is given by:

$$SNR = 10\ log_{10} \left( \frac{\sum_{n=1}^{p} |f(n)|^2}{\sum_{n=1}^{p} |f(n) - \hat{f}(n)|^2} \right)$$

SNR is probably the most commonly used performance metric in audio denoising. That being said, it does not necessarily provide a clear indication to the perceptual quality of the denoised results. For instance, denoised results with residual noise and signal distortion artifacts can still yield high SNR values.

### 1.6.2 Perceptual Evaluation of Audio Quality (PEAQ)

In order to compensate for the drawbacks of the SNR metric, we also make use of the ITU-R recommendation BS. 1387 Perceptual Evaluation of Audio Quality (PEAQ) metric

[17]. As the name suggests, this metric is a perceptual measure. PEAQ attempts to model perceived audio quality differences by combining a number of psycho-acoustic features, based on a filter-bank ear model. A measure known as the *objective difference grade* is computed via a neural network featuring a quality scale ranging from -4 (very annoying) to 0 (imperceptible).

### 1.6.3 Audio Quality

While both SNR and PEAQ metrics are standard measures of audio quality, they are not accepted uncritically. Notably, SNR is based on the L2-norm and hence favours non-sparse solutions, while the PEAQ metric appears to favour sparser signal representation. Nonetheless, these metrics provide us with with tools for performing quantitative comparisons between the denoising approach developed here and state-of-the-art noise reduction algorithms. We however maintain that the best measure of audio quality is through subjective listening tests. While we performed informal testing of this nature, large-scale listening tests are beyond the scope of this work.

## 1.7 Previous Work

In his article on the relevance of simple shrinkage in redundant representations [6], Elad highlighted the use of simple wavelet shrinkage operators to solve the basis pursuit denoising problem. He also shed some light on why simple shrinkage works with redundant dictionaries despite their lack of orthogonality. This research is important in the context of this work as we will also make use of simple shrinkage rules with redundant dictionaries. Siedenburg points out that while approaches for adaptive thresholding have been described in wavelet literature, the topic is relatively untouched in the context of audio denoising with time-frequency dictionaries [18]. He introduces an adaptive thresholding approach, coupled with Empirical Wiener attenuation. This research is interesting as it is a sparse denoising algorithm, which is similar to the approach we developed.

## 1.8 Thesis Organization

Chapter 2 provides an overview of greedy atomic decomposition and sparse signal representation. We discuss the different aspects of both, as well as how they pertain to audio signal denoising. We review the Matching Pursuit algorithm, highlighting the potential advantages of the approach for signal denoising. We then test the algorithm's ability to remove noise from musical audio, and present an analysis of its performance. In Chapter 3 we review the Wavelet threshold denoising approach. Wavelet thresholding shrinkage, is a powerful signal denoising approach, that is known to produce near optimal results for white noise removal [5]. In Chapter 4 we analyze different aspects of simple shrinkage in the context of greedy atomic decomposition. We then present the Greedy Time-Frequency Shrinkage (GTFS) denoising approach, and an analysis of its performance on real world audio signals. Chapter 5 presents extensive testing of GTFS denoising on a wide variety of music and speech signals.

## 1.9 Contribution

In this work our goal is to explore the use of greedy atomic decomposition algorithms to develop adaptive thresholding solutions for the purpose of audio signal denoising. The major contribution of this work is the design and implementation of a series of greedy denoising algorithms that are able to sparsely recover audio signals in a variety of noisy settings. A significant aspect of the algorithms developed in this work rely on the sparse nature of the denoised solution they produce. We are able to show that even in adversely noisy settings, sparse signal recovery of good subjective quality is possible. We believe that this feature can be taken advantage of in applications where signal compression is desirable.

# Chapter 2

# Greedy Atomic Decomposition

## 2.1 Greed is Good [1]

A *greedy* search strategy looks for the largest value amongst a set of values. In the context of an atomic decomposition, this corresponds to identifying the time-frequency atom most correlated to the local structures of the signal being analysed [19]. Despite their simplicity, greedy algorithms remain a popular choice in image and audio processing application due to their good performance, computational efficiency and ability to produce compact signal representations. As a result, in recent years several low bit-rate audio coding and compression approaches based on greedy atomic decomposition have been introduced [20, 21, 22, 23]. One of the most significant aspects of greedy signal representation algorithms is their ability to work with redundant dictionaries. In the context of this work, the enhanced representational power of redundant dictionaries allow for better recovery of deterministic audio content. While greedy methods have received significant attention for signal coding, compression and recovery, research into their signal denoising abilities have not received as much attention. In this chapter we will start by detailing the various aspects of greedy atomic decomposition, and discuss their significance to signal denoising. In section 2.3 we discuss the Matching Pursuit (MP) algorithm, which forms the basis for the denoising strategies developed in Chapter 4. In section 2.4.1 we analyze MP's ability to denoise a real world audio signal. We reiterate that in this research we are interested in denoising music and speech signals, and we have developed our approach taking into account the unique nature of our data. Apart from sparse signal representation, greedy approaches have also been studied for the purpose of sparse signal recovery from noisy or incomplete measurements [24]. This problem falls under the umbrella of compressed sensing, which is essentially a sub-sampling approach [25]. We note at this point that compressed sensing can also be looked at as a denoising approach, in which several random measurements (less than the length of the signal) of the signal are denoised to sparsely recover the signal. At this point, we note that while there are similarities, the approach developed in this work is not a compressed sensing solution [26]. We are however interested in the use of random measurement matrices for signal denoising, and hope to explore this idea in future work.

## 2.2 Atomic Decomposition

The idea of decomposing a signal over a family of functions that are well localized both in time and frequency was first proposed by Gabor [27]. The functions being considered here are known as time-frequency atoms; so called because they represent a fundamental unit of sound. Along with a mathematical proof of concept, Gabor conducted a series of psychoacoustic experiments that helped establish the connection between time-frequency atomic decomposition and the psychology of hearing [28]. As a result, the idea of atomic decomposition has formed one of the cornerstones of digital audio processing, manifesting itself as the Short-Time Fourier Transform (STFT). Over the years, the idea has been extended and enhanced, most notably with the Wavelet transform [29] and redundant dictionaries [14]. A signal can greedily be decomposed over a set of atoms by selecting the atoms that are the most correlated with the signal in an iterative manner. The weight of each selected atom is given by its correlation with the signal being analysed. In the following sections we will discuss the various constituents of an atomic decomposition in a greedy approximation setting.

### 2.2.1 Time-Frequency Atomic Families

The set of functions over which a signal is decomposed is known as a time-frequency (or time-scale) atomic family. Depending on the choice of atoms or indeed their parameterization, the decomposition can have very different properties. In recent years, the properties of several atom families have been studied. These include damped sinusoids, chirps, harmonic atoms, gammatones and even a family atoms specifically designed for audio noise reduction called speclets [30, 31, 32, 33, 34]. These atomic families are parametric, in the sense that the entire family may be generated from a single atom. Apart from parametric atoms, research has also been active in the use of instrument specific atoms for musical applications, as well as learned dictionaries. In this research we exclusively make use of STFT or Gabor atoms, which are widely used in audio processing. STFT atoms belong to the so called Weyl-Heisenberg group, which have a time-frequency structure. In Chapter 3 we review simple shrinkage, which is a classical denoising approach based on the Wavelet transform. Wavelets belong to the affine group of atoms, which have a time-scale structure.

A parametric time-frequency atomic family refers to a sampling of the time-frequency plane by a mother atom. That is to say, without any a priori information of the analysis signal, the entire time-frequency plane is sampled by moving a single atom to different locations in time and frequency. A significant advantage of using such atoms lies in the fact that the entire family may be generated from a single atom, i.e., all the atoms do not need to be explicitly generated. More formally, a general family of time-frequency atoms can be generated by translating and modulating a single window function $g \in L^2$. Here g(t) may be real or complex, and we assume that it is continuously differentiable. Using complex atoms provides the advantage of not requiring explicit definition of atom phases, which are obtained as a result of the decomposition [30]. Other conditions on g(t) include that its integral is non - zero and that $||g|| = 1$

For any scale $s > 0$, frequency modulation $\xi$ and translation $\upsilon$, we denote $\gamma = (s, \upsilon, \xi)$ and define :

$$g_\gamma(t) = \frac{1}{\sqrt{s}}\, g\left(\frac{t-u}{s}\right) e^{i\xi t} \tag{2.1}$$

The above equation represents a complex time-frequency atom. For a real valued time-frequency atom, there would be a sinusoid multiplied by the window function instead of a complex exponential. The parameter $u$ represents the translation of the window along the time axis, $\xi$ is the atom's frequency, and $s$ is the scale of the atom. A time-frequency dictionary is a single scale dictionary with $s = 1$, however it is also possible to have time-scale dictionaries, i.e., a dictionary containing atoms of several scales.

### 2.2.2 Redundant Dictionaries

Much like an atomic family, a redundant dictionary also refers to a sampling of the time-frequency plane. However, in order for a dictionary to be redundant, it *must* be over-sampled. Redundancy is directly proportional to the sampling rate, which is determined by the hop size used between adjacent windows or atoms, and the frequency resolution of the Fast Fourier Transform (FFT) used to compute correlations. Starting with a Weyl-Heisenberg family of atoms (such as the STFT one) it is possible, through the introduction of a scaling parameter, to build a dictionary that allows for time-scale analysis as well. This is achieved by *scaling* a mother atom over the time-frequency plane, in addition to translation and modulation. This multi-scale analysis ability is one of the main advantages of dictionary-based methods. Another significant advantage of redundant dictionaries is that different kinds of atomic families can be combined together in a single dictionary. In the context of audio, this offers advantages such as being able to use asymmetrical atomic shapes for accurate transient estimation, along with symmetrical atoms. A key feature of redundant dictionaries is their link with sparse signal representation. In general, the more redundant the dictionary, the sparser the resulting signal representation. The need for a very large dictionary is perhaps best explained in opening sentence of the original Matching Pursuit paper [14] – “ We can express a wide range of ideas and at the same time easily communicate subtle differences between close concepts, because natural languages have large vocabularies, that include words with close meanings.” The result is a low-level signal representation that is more compact than that obtained by using decomposition over an orthogonal basis, while preserving information content. Due to the relative compactness of the representation, atomic decompositions over redundant dictionaries are also known as sparse atomic decompositions. We believe that redundant dictionaries can be useful for an audio noise reduction approach. As we try to recover the deterministic audio rather than directly attenuate the noise, it seems logical that a larger vocabulary or dictionary would improve the chances of success.

It should be noted that most, if not all noise reduction approaches are based on signal transformation with orthogonal bases or functions that behave like orthogonal basis. This is because an orthogonal basis transforms white noise into white noise, which simplifies several of the processing steps. Redundant dictionaries are in general not orthogonal and hence classical denoising approaches do not necessarily translate to redundant signal representations. Thus, redundant dictionaries appear to offer a tradeoff in terms of signal denoising – better representational power, but lacking the optimality of orthogonality.

### 2.2.3 Greedy Parameter Estimation

Parameter estimation in greedy iterative algorithms like Matching Pursuit is achieved by means of *analysis-by-synthesis*. In this approach, successive parameters are estimated from a residual or error signal after all previously estimated atoms have been synthesized and subtracted from the original signal. This approach has been shown to perform well in the presence of noise for audio compression applications, and has also been used successfully for audio analysis, modification and synthesis [35, 36].

Accurate parameter estimation depends upon several factors. Amongst the most significant of these is the nature of the signal being analysed. This is because we can use this information, amongst other things, to design our dictionary. Dictionary design refers to choosing atomic families that are highly correlated with the signal being analyzed, and determining how they sample the time-frequency plane. This topic has been exhaustively studied over the last thirty years, and several dictionaries for speech and music have been proposed [20]. In this research however, we are interested in noisy speech and music. As a result, apart from having a dictionary that is highly correlated to the deterministic audio, it must also not be correlated with the noise. In theory this should not be a problem, as we are dealing with the removal of *uncorrelated* additive white noise. However from numerical experiments we can confirm that this is not the case. The unpredictable nature of the greedy atomic decomposition process often leads to noisy audio components being transformed along with deterministic audio. We also note that short atoms are more likely to be matched to noisy audio. Mismatch between dictionary atoms and a sound can lead to audible artifacts such as pre-echo [37]. In the case of noisy sounds, we found that a greedy iterative procedure often produces a signal distortion commonly known as *musical noise*. A possible reason for this is that the dictionary is too redundant, or too correlated to the noisy signal, leading a greedy algorithm like MP to match dictionary atoms to noisy audio components.

As highlighted in section 2.2.1, several different functions and signals have been used to construct dictionaries for signal decomposition.The important point is that the atoms of the dictionary are well correlated with the possible local time-frequency structures of the signal. The importance of this fact is heightened when the signal to be decomposed consists of signal components whose localizations vary widely in time and frequency. Music and speech sounds are good examples of such signals.When the atoms of a dictionary are not sufficiently correlated to the local time-frequency structures of the sound, the structures are subdecomposed over several atoms and their information is diluted, as in signal decomposition via orthonormal basis. This presents a problem in terms of extracting said time-frequency structure from the analysis representation, as the decomposition dilutes the information rather than provide information on how to easily identify it.

### 2.2.4 Stopping Condition

Greedy atomic decomposition is an iterative process which insures that the energy of the residual signal decreases continuously. The process is repeated until some stopping criteria is met. Signal-to-Residual ratio (SRR) is a parameter commonly used to stop a greedy atomic decomposition like MP. It indicates the power of the reconstructed signal compared to the residual MP signal. Ideally, the stopping condition should be based on *change* in SRR. When the change in SRR falls below a predetermined threshold, i.e., the SRR becomes approximately constant, the MP process can be stopped. This approach insures that MP represents as much of the deterministic audio content as possible, which is a very important factor in the context of signal denoising. In the noise reduction framework, our goal is to recover the meaningful audio while leaving out the noise. As a result, we use the SRR-based stopping criteria for all the denoising approaches developed and tested in this work. An arbitrary low value of $5 \times 10^{-5}$ was selected as the threshold for the SRR level.

### 2.2.5 Sparsity

In section 2.2.2 we stated that the redundancy of the dictionary used for atomic decomposition is intrinsically linked to the sparsity of the signal representation. That is to say, the larger the dictionary, the smaller the number of transform coefficients needed to represent the signal. As a result sparse representation algorithms like MP have been widely used for audio coding and compression. In the context of audio denoising, the aim is to recover as much deterministic audio content, irrespective of sparsity. That is to say, achieving high signal sparsity is not a priority. That being said, from numerical experiments we noted that the quality of the denoised sounds produced by MP based denoising is directly linked to the sparsity of the denoised signal representations. We consider a signal representation to be *sparse*, if it requires less than 5% of the signals transform domain coefficients are non-zero. In general, sparse output sounds imply denoised results of good subjective quality. However if the output signal representation is *too sparse*, it might mean that denoising algorithm has not been able to completely recover the signal. This typically occurs when the dictionary is not sufficiently correlated with the deterministic signal content, and can be largely avoided by making use of highly redundant dictionaries.

## 2.3 Matching Pursuits (MP)

The Matching Pursuit (MP) algorithm is the original greedy strategy for time-frequency atomic decomposition [14]. In their pioneering work, Mallat & Zhang proved that when a signal is decomposed over a redundant dictionary, a simple greedy approach will converge to a solution in a least squares sense. How 'good' this solution is, depends on the dictionary and stopping criteria used. More specifically, this means selecting a dictionary that is highly correlated with the analysis signal, and ensuring that the decomposition process is not stopped before all the deterministic audio content is matched. An important feature of the MP algorithm that motivates us to explore it in the context of audio denoising, is its adaptive nature. Matching Pursuit can adapt to the local time-frequency structures of the signal. An important aspect of this adaptive nature is the algorithm's ability to work with redundant dictionaries, whose enhanced representational power has been highlighted earlier. MP falls into the category of *sparse signal representation* algorithms, however sparsity is not strictly enforced by MP, but rather a consequence of the greedy search over a redundant dictionary. Another adaptive aspect of MP is its flexible structure. In the next section we will briefly describe some variations of the MP algorithm that change the algorithm in terms of different parameters. These modifications of MP point to the robustness of the algorithm. By this we mean that even with relatively drastic changes to its basic structure (usually to suit a specific application), the modified MP algorithm will usually also converge in a mean square sense. A major aspect of this work will involve taking advantage of the flexible nature of MP, and adapt the algorithm for signal denoising.

We now proceed to describe the various steps involved in Matching Pursuit atomic decomposition. In chapter 3 we will develop denoising approaches through modifications of the MP structure, and thus a thorough understanding of its structure is crucial to our work. At each stage of the iteration, the atom that is the most correlated with the signal being analyzed is selected by the algorithm. Then the weighted contribution of this atom to the signal is subtracted and the iteration proceeds on the residual. The weight given to this atom is determined by the value of the correlation between the signal and the atom. Mathematically, the task at the $i$th stage is to find the atom $g_{m(i)}$ that minimizes the 2-norm of the residual $r_{i+1}$,

$$r_{i+1}[n] = r_i[n] - \alpha_i g_{m(i)}[n] \tag{2.2}$$

Where $\alpha_i$ is a weight that describes the contribution of the atom to the signal, i.e., the expansion coefficient, and $m(i)$ is the dictionary index of the atom. The process begins with $r_0[n] = x[n]$. Treating the signals as column vectors, the energy of the 2-norm of the residual $r_{i+1}$ is minimum when orthogonal to the atom:

$$\langle r_i - \alpha_i g_{m(i)}, g_{m(i)} \rangle = (r_i - \alpha_i g_{m(i)})^H g_{m(i)} = 0 \tag{2.3}$$

$$\Rightarrow \alpha_i = \frac{\langle g_{m(i)}, r_i \rangle}{\langle g_{m(i)}, g_{m(i)} \rangle} = \langle g_{m(i)}, r_i \rangle$$

Where the last step follows from restricting the atoms to be unit-norm. Then the energy $\langle r_{i+1}, r_{i+1} \rangle$ of the error is,

$$\langle r_i, r_i \rangle - \frac{|\langle g_{m(i)}, r_i \rangle|^2}{\langle g_{m(i)}, g_{m(i)} \rangle} = \langle r_i, r_i \rangle - |\alpha_i|^2 \tag{2.4}$$

This energy is minimized by choosing the atom $g_{m(i)}$ that has the largest magnitude correlation with the signal, and the expansion coefficient for that atom is $\langle g_{m(i)}, r_i \rangle$. In deriving a signal decomposition, Matching Pursuit is iterated until the residual energy is below some threshold, or until some other, often perceptual stopping criteria is met. After $I$ iterations the decomposition corresponds to the estimate,

$$x[n] \approx \sum_{i=1}^{I} \alpha_i g_{m(i)} \tag{2.5}$$

The mean-squared error of this sparse approximation, namely the energy of the residual, converges to zero as the number of iterations approaches infinity [14]. This convergence property implies that $I$ iterations will provide a reasonable $I$-term estimate. This $I$-term approximation however, is in general not optimal in a mean-squared sense. Since the dictionary is not orthogonal, the standard Matching Pursuit algorithm does not find the optimal $I$-term expansion. In order to do so, it requires finding the minimum projection error over all $I$-dimensional dictionary subspaces, which is not computationally feasible for large $I$.

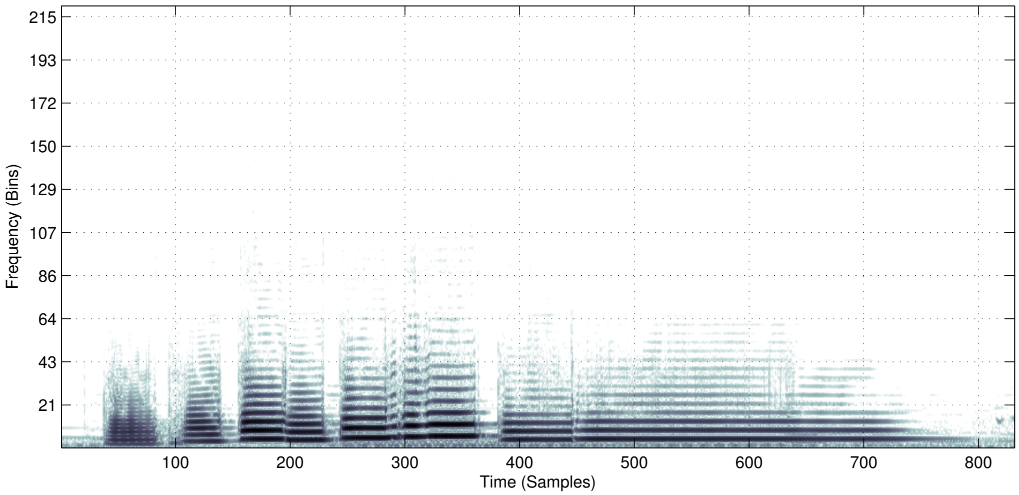


**Fig. 2.1** Spectrogram of clean trumpet sound.

Figs. 2.1 & 2.2 displays the spectrogram of a trumpet recording and its MP reconstruction. The MP decomposition process was stopped when a SRR of 20 dB was achieved. A three scale dictionary with Gabor atoms corresponding to window lengths of 4096, 2048 and 512 samples was used for signal decomposition. From the figure above it is clear that

MP has not completely recovered the sound, with certain spectral elements present in the original recording, that are absent from the reconstruction. We can however confirm that the reconstructed sound is of good quality, and greater spectral detail can be achieved by allowing the algorithm to reach a higher SRR.

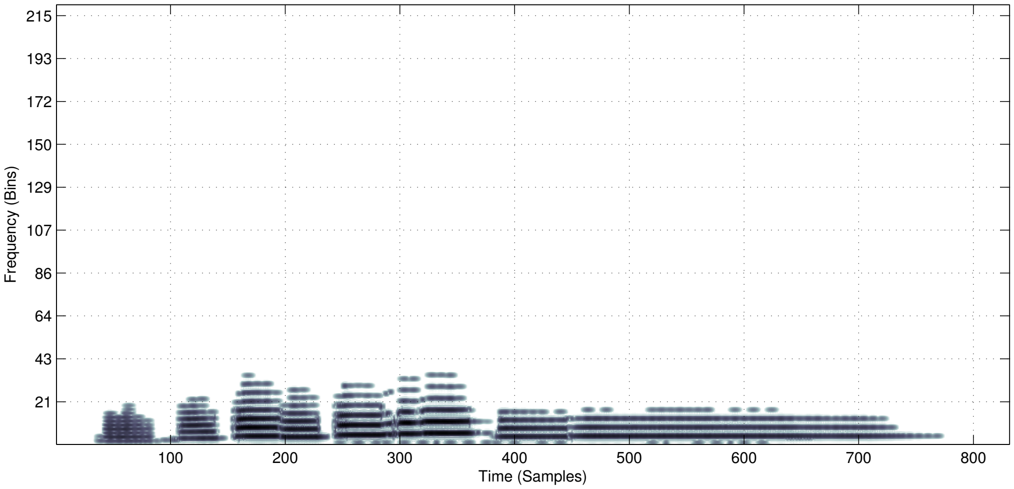
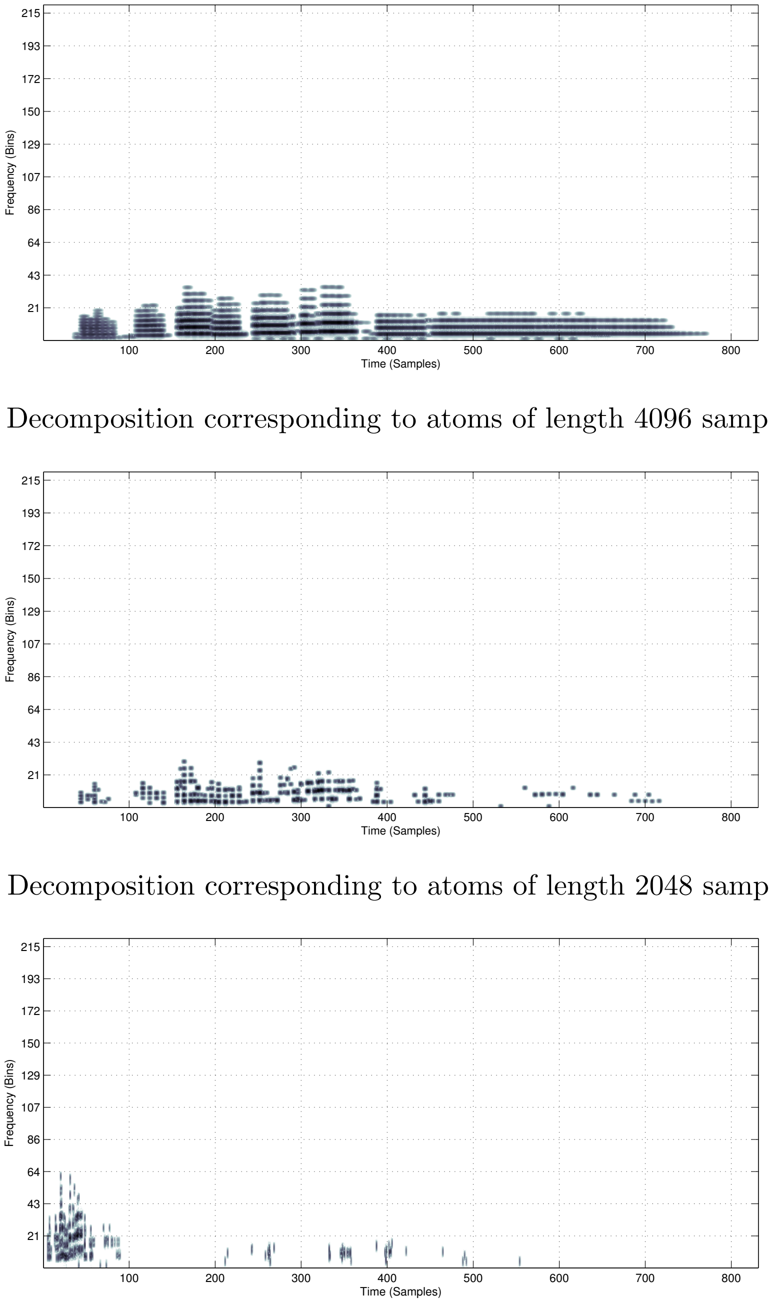


**Fig. 2.2** MP reconstructed trumpet spectrogram.

Fig. 2.3 illustrates the multi-layer nature of a MP decomposition. We see that most of the atoms selected in the representation correspond to longer atoms of length 4096 and 2048 samples, while a small number of 512 length atoms are also selected. Fig. 2.3 highlights the ability of MP to analyze a signal at different resolutions. This is the main advantage of MP over classical approaches for signal transformation. Our goal is to take advantage of this enhanced representational power of MP in order to develop an audio denoising strategy.



(a) Decomposition corresponding to atoms of length 4096 samples.

(b) Decomposition corresponding to atoms of length 2048 samples.

(c) Decomposition corresponding to atoms of length 512 samples.

**Fig. 2.3** MP: multi-layer atomic decomposition.

## 2.4 MP Variations

Over the last decade and a half MP has been widely used in several audio coding and compression applications. One of the reasons for this is the simple and extendible structure of the MP algorithm. As a result, several variants of the MP algorithm have been proposed, each usually tailored to a specific application. We do not attempt to list or review all of these MP variants here, rather we refer the reader to the literature [38, 22, 23, 39, 32]. Some of these extensions include dictionary weighting, integrating psychoacoustic information and improving computational efficiency. While several variants of MP exist, none of them, to our knowledge have been designed to specifically tackle the audio denoising problem.

### 2.4.1 MP Denoising

Most modern signal denoising approaches are equipped with an assumption of sparsity, i.e., they make use of a *prior*, which imposes certain conditions on the transform coefficients of the signal. As stated in the section 2.3, signal sparsity in MP is a result of the greedy search over a redundant dictionary. Specifically, MP builds a signal representation by greedily selecting the dictionary atoms most correlated to the signal, which often forms a sparse set. In the case of signal denoising, MP will first match dictionary atoms to the deterministic parts of the signal, before encoding noisy audio. Thus, the idea would be to stop the decomposition process before this occurs. This not a trivial problem, as one of the main issues with MP is that the algorithm does not know when to stop, i.e., it will continue to remove atoms from the residual signal, possibly indefinitely. This fact is amplified in a noisy setting as it leads to noisy audio components being captured and transformed. As mentioned previously, we make use of a stopping criteria based on the SRR value. In order to illustrate MP for audio denoising, we use the trumpet recording from Fig. 2.1, which we deteriorate with 5 dB of additive white noise. The spectrum of the noisy trumpet sound is shown in Fig. 2.4. A two scale dictionary with Gabor atoms corresponding to window lengths of 4096 and 2048 samples is used for signal transformation. We note that in section 2.3, a three scale dictionary was used for signal decomposition, however from Fig. 2.3c we know that a very small number of 512 length atoms were used in signal representation. Moreover, the shorter length of these atoms makes them more susceptible to being matched to noisy audio content.

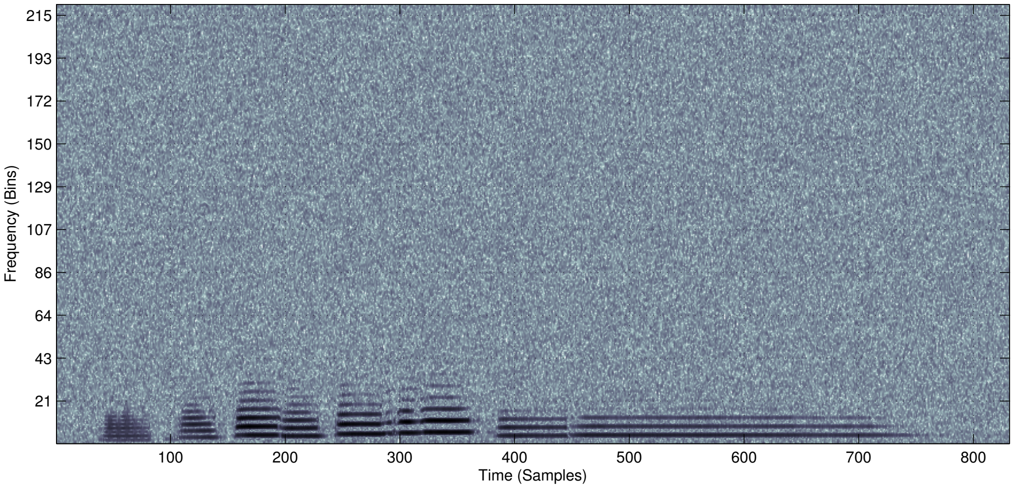


**Fig. 2.4** Spectrogram of trumpet deteriorated by 5 dB white noise.

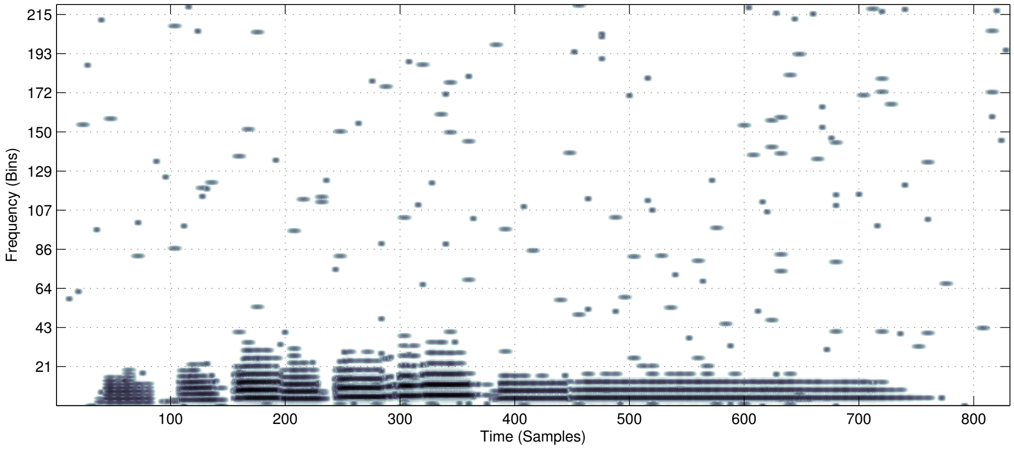


**Fig. 2.5** Denoised trumpet spectrogram.

Fig. 2.5 displays the spectrogram of the denoised sound output by Matching Pursuit. From the figure it is clear that MP was unable to stop the decomposition process before some part of the noisy audio content was transformed. The noisy artifacts are most prominent in the high frequency range, and is commonly referred to as *musical noise*. We note that for an arbitrary SRR threshold of $5 \times 10^{-5}$, the MP algorithm appears to continue indefinitely. In order to generate Fig. 2.5 the algorithm was run for 1724 iterations, which was the number of times the algorithm was run in section 2.3. A simple approach for improving MP's performance in terms of attenuating musical noise would be to increase the SRR threshold. The problem with this approach is that selecting an appropriate value can be challenging, as too high a threshold would lead to underfitting, i.e., part of the deterministic audio content would not be recovered. Such an approach would also not be taking full advantage of the representational power of redundant dictionaries. As a result, we choose to persist with very low SRR threshold ($5 \times 10^{-5}$) throughout this work. In chapter 3 we will develop ideas for MP-based audio denoising, that try to tackle the problem in a different way, by identifying when noisy audio content is being matched. By working with a low SRR threshold and using highly redundant dictionaries, we are able to avoid the underfitting problem in most cases.

## 2.5 Summary

In this chapter we presented an overview of greedy atomic decomposition using redundant dictionaries. We highlighted the advantages of dictionary-based methods over traditional signal transformation, in terms of signal representation and sparsity. We then discussed parameter estimation in the context of greedy atomic decomposition, and how it relates to signal denoising. Then we presented the Matching Pursuit (MP) algorithm, which forms the basis of the denoising approaches developed later in this work. We highlighted the algorithms ability to adapt to the local time-frequency structures of the analysis signal, as well as the ability to easily manipulate the algorithms structure. We then illustrated that when Matching Pursuit is used to denoise a corrupted musical signal, the algorithm is able to recover the deterministic audio, but is not able to stop the decomposition process before some of the noise is also recovered. This noise manifests itself in the denoised signal in the form of isolated time-frequency artifacts called musical noise. While this is the case, we believe that the algorithms performance can be improved, based on our analysis of greedy atomic decomposition and redundant dictionaries. In order to develop ideas to do so, we review a classical signal denoising approach in the next chapter.

# Chapter 3

# Simple Shrinkage

## 3.1 Introduction

*Simple Shrinkage* is a classical denoising approach in signal processing. The approach is based on the idea of signal sparsity in the wavelet or transform domain. Signal denoising is carried out with the help of an oracle, that furnishes information regarding a spatially adaptive estimator. Wavelet shrinkage has been shown to produce close to optimal results for the removal of white Gaussian noise [5]. Apart from being a powerful denoising strategy, one of the main reasons that we choose to study simple shrinkage is that while the approach was designed for unitary transforms and orthogonal basis, shrinkage has been used successfully with no-unitary transforms and even redundant dictionaries [6]. This is an important factor, as we deal with MP-based redundant signal representations in this work. Our goal is to improve MP-based audio denoising by incorporating different aspects of simple shrinkage into the approach. In this chapter we review wavelet shrinkage denoising, highlighting various aspects of the approach that will be further analyzed and enhanced in the next chapter.

## 3.2 Wavelet Threshold Denoising

### 3.2.1 Introduction

A commonly used approach for the removal of additive Gaussian noise from a signal $\mathbf{y}$, is via the minimization of the function

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda P_r(\mathbf{x}) \tag{3.1}$$

This equation emerges from a Baysian formulation by means of maximum a posteriori probability (MAP) estimation. The first term is referred to as the log - likelihood, and describes the relationship between the desired (clean) signal $\mathbf{x}$ and its noisy version $\mathbf{y}$. The term $P_r(\mathbf{x})$ represents a prior posed on the unknown signal $\mathbf{x}$. Over the years numerous expressions have been proposed and utilized for different signal types [6]. When a signal prior can be expressed as a linear combination, it can also be described as a filter, i.e., it describes an approach to filter the transform coefficients of the noisy signal so as to retrieve the clean signal or sound. In this work we are interested in one such filtering approach, related to the wavelet transform [40]. In a series of publications, Donoho and Johnstone advocated the sparsity of the wavelet coefficients of a signal as the driving force in recovering the desired signal. Interestingly, the use of a wavelet filter or prior in eq. 3.1 leads to a simple closed form solution, known as *shrinkage*. The wavelet denoising approach is known to be ideal for Gaussian white noise provided that sparsity is enforced on the signal's representation by means of a unitary transform.

### 3.2.2 Problem Statement

Given N noisy samples of a function $s$,

$$y_i = s(t_i) + z_i, \qquad i = 1, .....N \tag{3.2}$$

with $t_i = \frac{i-1}{n}$ and $z_i$ is given by independent, identically distributed (i.i.d) white noise.

The goal of the wavelet shrinkage signal denoising approach is to estimate the function $\mathbf{s}$, with small mean square error (MSE). That is, to find an estimate $\hat{\mathbf{s}}$ from the data $y_1, ....y_n$

with small risk,

$$R(\hat{s}, s) = N^{-1}.E[(\hat{s} - s)^2] \tag{3.3}$$

with a high probability that $\hat{\mathbf{s}}$ is at least as smooth as $\mathbf{s}$.

In order to solve this problem, one usually specifies a fixed class of functions to which the signal belongs. The Fourier and Wavelet basis are amongst the more commonly chosen sets of functions.Then one may seek an estimator $\hat{\mathbf{s}}$ that attains minimax, or close to minimax risk over the function class. In order to apply this principle in practical situations, we assume that $\mathbf{s}$ is an unknown member of a scale of function classes and may attempt to behave in a way that is simultaneously near minimax across the entire scale [41].

### 3.2.3 Spatial Adaptation via Wavelet Shrinkage

As per the idea of ideal spatial adaptation, an oracle furnishes information about how best to adapt a spatially variable estimator, to an unknown function [40]. Donoho and Johnstone introduced this concept through wavelet shrinkage, and let us quote them: "Orthonormal bases of compactly supported wavelet provide a powerful complement to traditional Fourier methods in that they permit an analysis of a signal into localized oscillating components, due to the time-scale nature of the wavelet basis." The principle idea in wavelet shrinkage is to use this kind of spatially varying (varying scale) decomposition to build algorithms that adapt their effective 'window width' to the amount of oscillation in the data. The primary steps involved in a wavelet denoising solution are summarized below :

- *Discrete wavelet transform of noisy data.* The $N$ noisy data samples are transformed via the discrete wavelet transform, to obtain $N$ noisy wavelet coefficients.
- *Thresholding of noisy wavelet coefficients* via either soft or hard thresholding.
- *Inverse wavelet transform* coefficients to obtain a denoised signal approximation.

### 3.2.4 Transformation of Data

As mentioned previously, in their original paper the authors take advantage of the time-scale nature of the wavelet transform and its multi-resolution analysis ability. The choice of wavelets, amounts to the choice of a filter [42]. This is interesting, as overcomplete or redundant representations can also analyze a signal at different scales. It is worth noting that the principles of wavelet shrinkage apply to other unitary transforms like the Fourier transform, however the denoising results are not as appealing when single resolution analysis methods are used. We will discuss simple shrinkage in the context of redundant and multi-scale dictionaries in greater detail in a later section; for now it is safe to say that these denoising rules have been found to produce good results even when overcomplete signal representations are used [6].

### 3.2.5 Threshold Selection

In order to shrink the wavelet coefficients, one needs to compute a threshold value. Donoho and Johnstone proposed several threshold selection rules, including the minimax, universal and the Stein Unbiased Risk Estimate (SURE) thresholds. The SURE threshold differs from the former two as it is computed from the actual noisy data, while the others only depend on the length of the signal. This data adaptive behaviour of the SURE threshold motivates us to explore it further in order to develop dynamic thresholding ideas for the GTFS denoising approach.

#### Universal Threshold

The Universal threshold denoising algorithm was proposed in [40], and is also known as VisuShrink. This approach adheres to a fixed threshold form given by,

$$\lambda_{uni} = \sigma \ \sqrt{2 \ log(N)} \tag{3.4}$$

Where N is the signal length and $\sigma$ is the noise level estimation based on equation 3.7. The Universal threshold guarantees to attenuate all the noise, but in doing so it often underfits the data [43]. This leads to very sparse denoising results.

### Minimax Threshold

The Minimax threshold was also proposed in [40]. It is an optimal threshold that is derived from minimizing the constant term in an upper bound of the risk involved in the estimation of the function. The optimal threshold is defined as,

$$\lambda_M = \sigma \lambda_n^* \tag{3.5}$$

where $\lambda_n^*$ is defined as the value of $\lambda$ satisfying

$$\lambda_n^* := inf_\lambda \ sup_d \left\{ \frac{R_\lambda(d)}{n^{-1} + R_{oracle}(d)} \right\} \tag{3.6}$$

where $R_\lambda(d) = E(\delta_\lambda(d) - d)^2$ and $R_{oracle}(d)$ is the ideal risk achieved with the help of an oracle. Donoho and Johnstone considered two oracles, *diagonal linear projection* – an oracle that tells you when to 'keep' or 'kill' each empirical wavelet coefficient, and a *diagonal linear shrinker* – an oracle that tells you how much to shrink from each coefficient. We do not go into details regarding oracles here and rather refer the reader to [40].

### SURE Threshold

In 1981 Stein introduced an approach to estimate the mean, $\hat{\mu}$, of a multivariate normal distribution in an unbiased fashion. Stein showed that for a given multivariate normal observation $y \sim \mathcal{N}(\mu, I), \mu \in \mathbb{R}^p$ and a *near* arbitrary estimator $\hat{\mu} = y + g(y)$, with $g : \mathbb{R}^p \to \mathbb{R}^p$ being weakly differentiable, the risk of $\hat{\mu}$ is given by

$$\mathbb{E}\|\hat{\mu} - \mu\|^2 = p + \mathbb{E}[\|g(y)\|^2 + 2\nabla g(y)]$$

where $\nabla g = \sum_{i=1}^{L} \frac{\partial}{\partial i} g_i$, cf. [44]

Where $L$ is the length of the analysis signal.

### 3.2.6 Shrinkage Operators

Once a threshold has been selected, the next step in the wavelet denoising procedure is to shrink the wavelet coefficients and threshold them according to a given rule. Motivated by the fact that only a few wavelet coefficients contribute to the signal, i.e., the signal is sparse in the wavelet domain, Donoho and Johnstone introduced two non-linear thresholding rules for wavelet shrinkage :

- **Hard Thresholding**: $\eta_H(w, \lambda) = w \ \{|w| > \lambda\}$
- **Soft Thresholding**: $\eta_S(w, \lambda) = sgn(w)(|w| - \lambda)_+$

Where $(.)_+$ is max(0,.).
It is worth noting that the hard thresholding operator does not actually attenuate or shrink the value of atomic coefficients like the soft thresholding rule. As a result the two operators display markedly different behaviour, each with their own advantages and disadvantages. The soft thresholding estimate is a continuous function and tends to have bigger bias due to the shrinkage of larger coefficients. The hard thresholding estimate on the other hand is a discontinuous function and thus tends to have bigger variance and can be unstable. Over the years several other non-linear estimators have been proposed that offer a tradeoff between the hard and soft shrinkage estimates [45]. In this work, along with hard and soft shrinkage, we experiment with the Empirical Wiener attenuation rule [46], which has been shown to produce results that are somewhat in between the hard and soft thresholding rules [47].

### 3.2.7 Noise Variance Estimation

An important factor in the wavelet denoising approach is the estimation of the noise power or variance. In the original papers Donoho and Johnstone proposed the use of a robust median filter to provide an estimation of the noise variance. The noise variance estimate is calculated as

$$\sigma = \frac{MAD}{0.6745} \tag{3.7}$$

Where MAD is the absolute median estimation of the wavelet (transform) coefficients. We note that while this noise level estimate has been shown to work well for several denoising applications, researchers have pointed out, for example, that in the context of speech enhancement, this noise level estimate is not robust enough [48]. Through our experiments we hope to evaluate the effect of noise variance estimation on our denoising solution.

### 3.2.8 Level Dependent Thresholding

Johnstone and Silverman introduced a wavelet shrinkage approach for signals corrupted with correlated noise [49]. In their approach, noise variance is estimated individually at each level (scale) and level dependent thresholds are obtained. The GTFS denoising approach tries to take advantage of the representational power of multi-scale overcomplete dictionaries, and thus a level or scale dependent thresholding approach can be incorporated into its framework.

# Chapter 4

# Greedy Time-Frequency Shrinkage

## 4.1 Introduction

In the previous chapter we analyzed the signal denoising capabilities of Matching Pursuit. We established that MP starts out by selecting only atoms correlated to the deterministic parts of the audio, however when used in conjunction with a SRR based stopping criteria, MP will match atoms to the noisy signal content as well. In order to achieve denoised audio reconstructions of good quality, the algorithm needs to be able to determine when it should stop, i.e., when all the deterministic audio components have been recovered. A simple approach to make MP more robust in this regard is to implement a stopping criteria based on thresholding the correlation value between the dictionary atoms and the residual MP signal. When MP is used for signal denoising in conjunction with this kind of stopping criteria, the method can be interpreted as a *simple shrinkage* approach. In this chapter we present the *Greedy Time-Frequency Shrinkage* (GTFS) denoising approach. We analyze different aspects of wavelet shrinkage in the context of MP based audio denoising and propose several strategies to improve the performance of the approach.

## 4.2 Greedy Time-Frequency Shrinkage

The denoising principle in MP is based on the fact that the algorithm selects time-frequency atoms that are highly correlated with the signal in order to build a signal representation.

Thus as the noise we wish to remove is uncorrelated, MP will first select the correlated deterministic atoms before it selects the noisy ones. The success of the approach is based on MP being able to identify when a noisy atom is selected, and stopping the decomposition process. When this is achieved by thresholding the correlation value between the dictionary atoms and the residual signal, MP denoising can be viewed as a *simple shrinkage* approach. In this interpretation, MP iteratively applies the *hard thresholding* rule to the greedily selected atomic coefficient. We hence dub MP denoising performed in this way – *Greedy Time-Frequency Shrinkage (GTFS).*

**Algorithm 1: GTFS**

**Task:** Denoise the signal y by $\hat{x} = argmin_z \ .\|Dz - y\|_2^2 < \varepsilon$

**Parameters:** $D$ is a redundant dictionary and $y$ is the noisy signal, k is the iteration count and $\varepsilon$ is a predetermined threshold.

**Initialization:** Set $residual = y, z_k = 0 \ and \ k = 0$

**Threshold Selection:** Determine a threshold value using either *universal, minimax or SURE thresholds*

**Main Iteration:** Set k =1 and apply,

- **Weights**: Calculate correlations between y and D and select the most correlated atom using a standard greedy procedure.
- **Shrinkage**: Shrink the coefficient of the selected atom using the hard thresholding rule:
  $\alpha_H(\alpha_i, \lambda) = \alpha_i \ \{|\alpha_i| > \lambda\}$
- **Return**: Set k = k + 1 and repeat.
- **Stop**: Stop iterating when the change in residual energy falls below the threshold $\varepsilon$.

**Finalize**: The denoised output is $\hat{x} = Dz$.

The algorithm above illustrates how GTFS or MP denoising based on correlation thresholding may be interpreted as a simple shrinkage approach. MP iteratively applies the *hard thresholding* rule on the greedily selected atom. Ideally, the algorithm should stop when all the deterministic audio components have been recovered. However this is not always the case, as GTFS denoising is sensitive to the shrinkage operator, the threshold value $\lambda$, and the dictionary used for signal analysis.

### 4.2.1 Shrinkage Operator

GTFS iteratively applies the hard thresholding rule on the selected atomic coefficient. The hard thresholding operator does not actually attenuate or shrink atomic coefficients, i.e., it is a 'keep' or 'kill' rule, and hence the success of the approach depends on dictionary configuration, correlation thresholds, and stopping thresholds.The hard thresholding operator is a discontinuous function, and produces results with high variance [45]. Moreover, hard thresholding is a *diagonal estimation* approach, wherein each time-frequency coefficient is processed independently. One of the known drawbacks of diagonal estimators in audio denoising is that they produce isolated time-frequency artifacts that are perceived as *musical noise* [50].

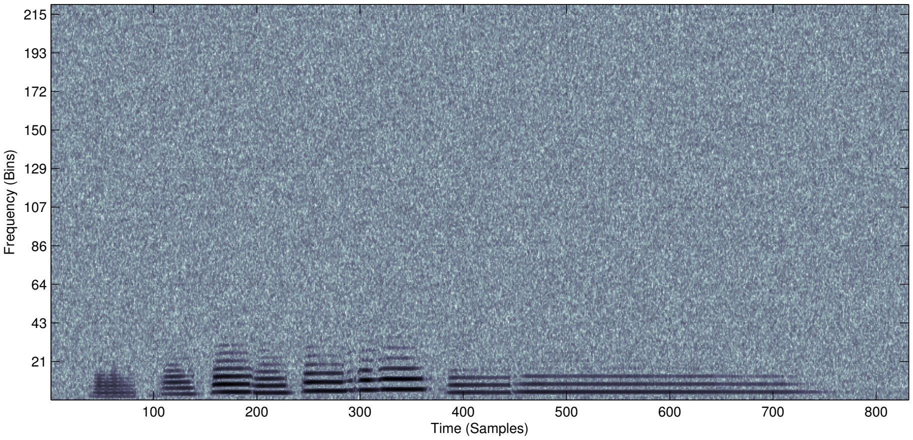


**Fig. 4.1** Spectrogram of trumpet audio deteriorated by 5 dB white noise.

Fig. 4.1 displays the spectrogram of a trumpet sound sampled at 44.1 kHz, deteriorated by 5 dB of additive white noise. Fig. 4.2 displays the spectrogram of the denoised sound produced by GTHS in conjunction with the Minimax threshold.The dictionary used for signal transformation is of two scales with atoms corresponding to lengths of 4096 and 2048 samples respectively.One can clearly see isolated time-frequency artifacts in the denoised

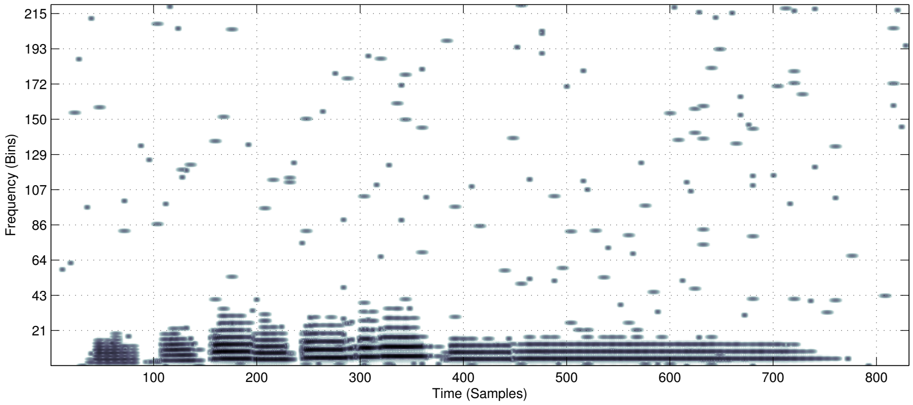


**Fig. 4.2** GTFS denoised trumpet spectrogram.

reconstruction. As could be expected, the hard thresholding diagonal estimation approach produces these musical noise artifacts in the denoised sound.

### 4.2.2 Threshold Selection

At each iteration of GTFS the correlation of the selected dictionary atom is checked against a threshold. As stated in the previous section, GTFS does not actually attenuate atomic coefficients and hence an appropriate threshold is needed to avoid both under and overfitting of data.Table 3.1 compares the output SNR achieved by GTFS in conjunction with the different wavelet thresholds, for the noisy trumpet sound in Fig. 4.1.

| Threshold | SNR (dB) |
|---|---|
| Universal | 16.16 |
| Minimax | 18.61 |
| SURE | 17.70 |

**Table 4.1** Denoising results for different thresholds.

Fig. 4.3 compares the absolute value of the atomic coefficients selected by GTFS with the different wavelet thresholds. In this example, GTFS has been modified by replacing the hard thresholding rule with soft thresholding, in order to visualize how the different thresholds affect coefficient attenuation. A more detailed analysis of GTFS in conjunction with the soft thresholding operator is presented in section 4.3.1.

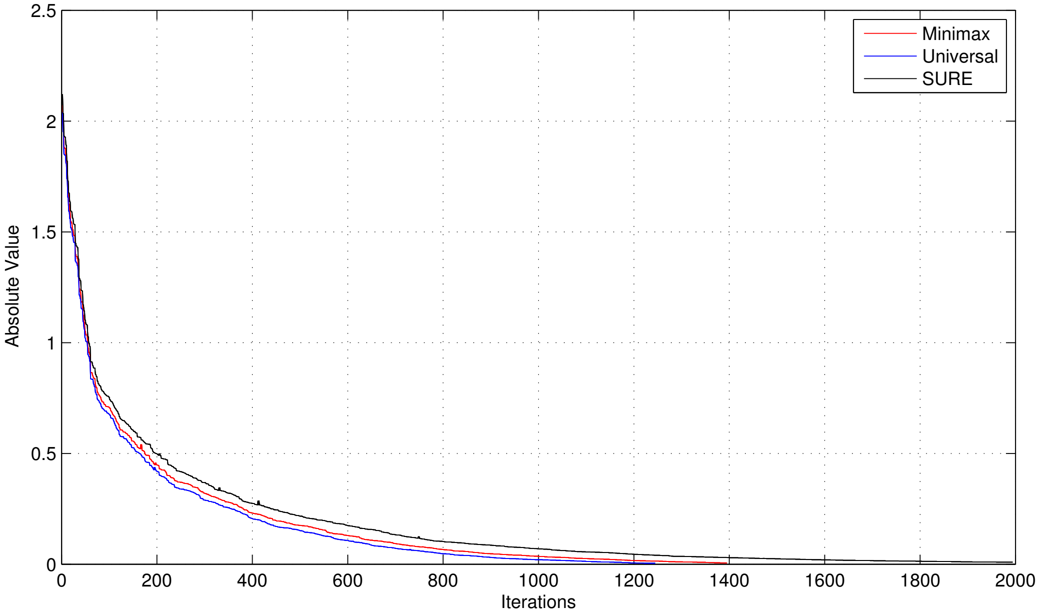


**Fig. 4.3** Atomic coefficients selected by GTFS using different thresholds.

The Universal threshold (0.1386) is the largest of the three, and attenuates the time-frequency coefficients the most, as illustrated by the blue curve in Fig. 4.3. The SURE threshold (0.0569) is the lowest and accordingly causes the leases coefficient attenuation (black curve). The Minimax curve displays intermediate coefficient attenuation (red curve), as suggested by its value of 0.1076. As the Universal threshold is considerably larger than the Minimax and SURE thresholds, it stops the signal decomposition process the earliest, producing the sparsest result. While it is able to completely attenuate the noise, the Universal threshold underfits the data, i.e., it is unable to *completely* recover the sound. As a result it yields the lowest output SNR of the three thresholds. That being said, both the Minimax and SURE thresholds used in conjunction with GTFS produce high-frequency musical noise artifacts.

### 4.2.3 Dictionary Design

Signal transformation in GTFS is achieved by decomposing a signal over a redundant dictionary. The main assumption is that the signal under analysis can be represented by a small number of dictionary elements, i.e., an assumption of *sparsity*. In the context of audio denoising, in addition to sparsely representing the sound, the dictionary should not be correlated to the noisy signal content. Thus, GTFS denoising is highly dependent on the dictionary, and the diagonal estimation approach often leads to data overfitting and musical noise artifacts. Numerical experiments have shown that GTFS produces the best denoising results when a dictionary with relatively long atoms is used. This is because long (tonal) atoms are well correlated to the tonal content of a sound. Longer atoms are also less likely to be matched to noisy signal content than shorter atoms. Another advantage of using a dictionary with longer atoms is that it implies sparser denoising results. In general, sparse denoised results imply good results in terms of musical noise attenuation and output SNR. However in extreme cases, underfitting leads to results that are *too* sparse, and generally unacceptable. We make use of a three-scale dictionary with atoms corresponding to window lengths of 8192, 4096 and 2048 samples for the majority of our experiments. It should be noted that while this kind of dictionary worked well for the majority of our experiments, it underfits the data for sounds with a large amount of transient content, i.e., sharp attacks, frequency modulation, breath noise etc. For such sounds we repeatedly perform a decomposition on the residual signal, using shorter (transient) atoms of lengths corresponding to 512, 256 and 128 samples.

#### Atomic Shape

Apart from the length of the atoms in the dictionary, windows of different shapes may be used to generate time-frequency atoms. In this research we work exclusively with Gabor or STFT atoms, which are widely used in audio processing. Atomic families like Gammatones, damped sinusoids and Formant Wave Functions (FOF) have also been shown to work well with speech and musical audio, and may potentially offer certain advantages over Gabor atoms. We will consider GTFS denoising with different atomic families in future works.

### 4.2.4 Discussion

In this section we analyze GTFS denoising in terms of coefficient attenuation and threshold selection. GTFS uses the hard thresholding rule to attenuate atomic coefficients, which often overfits the data leading to musical noise artifacts. Shrinkage or attenuation of atomic coefficients plays a central role in signal denoising and hence we are motivated to explore alternate attenuation rules as a means to improve the performance of GTFS denoising. In terms of threshold selection, the wavelet thresholds were shown to yield problems related to both over and underfitting of data. The underfitting problem can be avoided by increasing dictionary redundancy, however, this in turn increases the risk of overfitting.

## 4.3 Model Enhancements

Having identified the primary factors on which GTFS denoising depends, we present several approaches to improve its performance in terms musical noise attenuation, output SNR and algorithmic stability. We begin by considering two alternate attenuation rules in conjunction with GTFS – Soft Thresholding and Empirical Wiener filtering. We then develop an adaptive thresholding approach, based on the SURE estimate.

### 4.3.1 Attenuation Rule

GTFS makes use of the hard thresholding rule, that does not attenuate a coefficient if it satisfies the threshold, i.e., it is a 'keep' or 'kill' rule.Thus if an erroneous atom gets past the correlation threshold, there is no provision to reduce or nullify its influence. In order to tackle this problem, we consider two alternate attenuation rules.

**Soft Thresholding**

Along with hard thresholding, Donoho and Johnstone introduced a soft thresholding operator for the shrinkage of wavelet coefficients [42]. The GTFS-ST algorithm maintains the same GTFS denoising structure, replacing the hard thresholding step with a soft threshold

$$\alpha_S(\alpha_i, \lambda) = sgn(\alpha_i)(|\alpha_i| - \lambda)_+ \tag{4.1}$$

Where $(.)_+$ represents $max(0, .)$. Unlike hard thresholding, the soft thresholding operator is a continuous function, and produces results with high bias due to attenuation of large coefficients [45].

### Empirical Wiener Attenuation

Like simple shrinkage, Empirical Wiener filtering is a signal denoising approach based on diagonal estimation [46]. The Empirical Wiener attenuation rule is given by

$$\alpha_{EW}(\alpha_i, \lambda) = \alpha_i \left(1 - \left[\frac{\lambda}{|\alpha_i|}\right]^2\right)_+ \tag{4.2}$$

In relation to simple shrinkage, Empirical Wiener filtering has been shown to produce results that are somewhat in between the hard and soft thresholding operators [47].

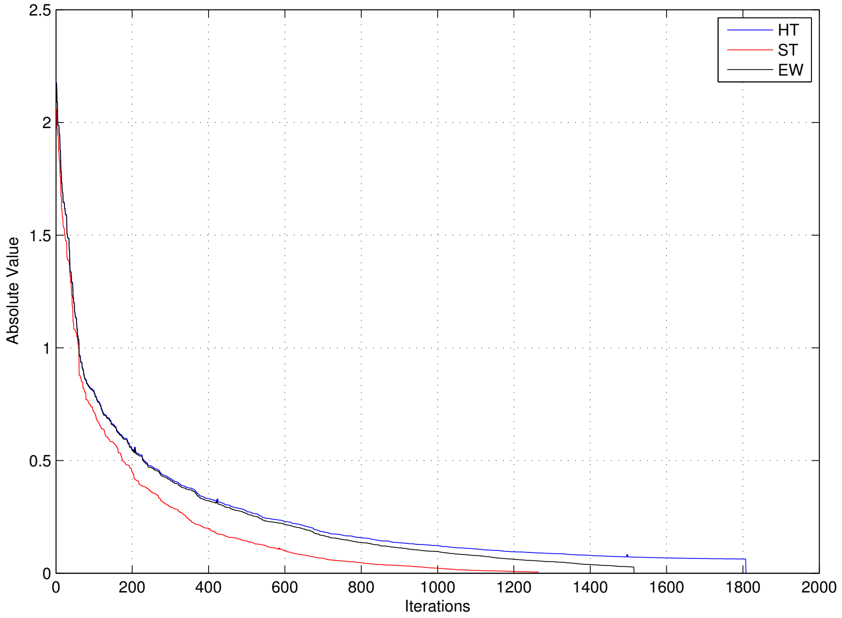
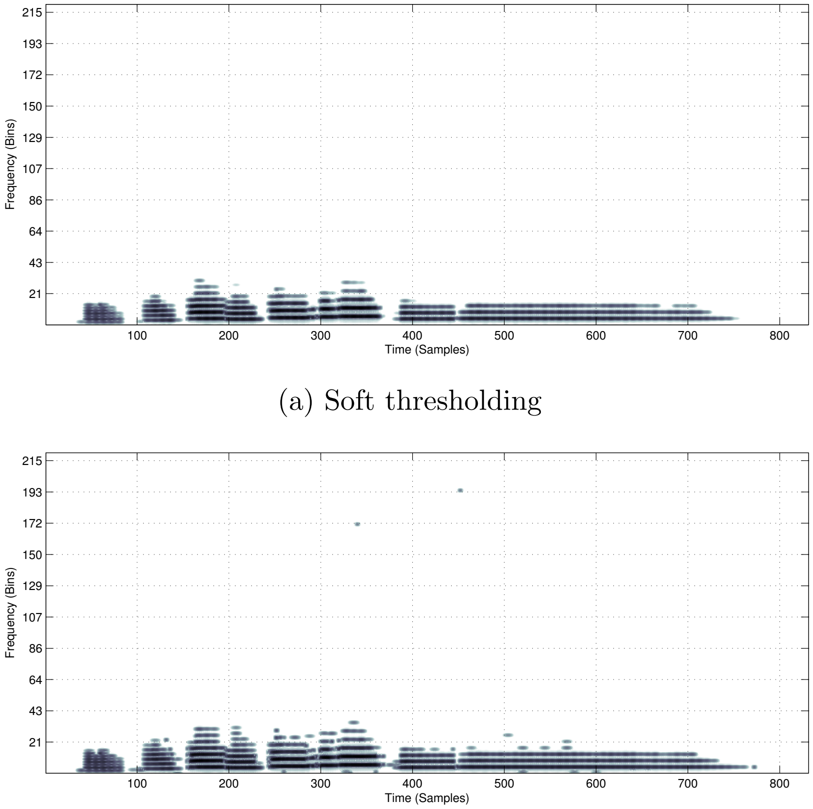


**Fig. 4.4** Atomic coefficients selected by GTFS using different attenuation rules. HT = Hard Thresholding. ST = Soft Thresholding. EW = Empirical Wiener.

Fig. 4.4 compares the evolution of the atomic coefficients selected by GTFS and those selected when the hard thresholding rule is replaced by soft thresholding and empirical Wiener attenuation rules. The SURE estimate was used as the threshold in each case. The empirical Wiener curve (black) starts out at a similar slope as the hard thresholding curve (blue), but starts to move in between it and the soft thresholding curve (red). GTFS in conjunction with the empirical Wiener rule also ran for a duration (iterations) that was in between the other two attenuation rules.



(a) Soft thresholding

(b) Empirical Wiener filtering

**Fig. 4.5** Denoised spectrograms produced by GTFS in conjunction with different attenuation rules.

Fig. 4.5 displays the spectrograms of the denoised sounds output by GTFS using hard thresholding, soft thresholding and the empirical Wiener attenuation rules respectively. All algorithms make use of the SURE threshold. The audio used for this example was a solo trumpet recording sampled at 44.1 kHz, deteriorated by 5 dB of additive white noise. We make use of a two-scale dictionary, with Gabor atoms corresponding to window lengths of 4096 and 2048 samples respectively. Figure 4.2 clearly shows that the hard thresholding operator overfits the data and creates musical noise artifacts. These artifacts, represented by isolated time-frequency structures in Fig. 4.2 exist primarily in the high frequency range (above 4000 Hz). Despite audible artifacts in the reconstruction, GTFS with hard thresholding yielded a SNR of 17.54 dB. Both the soft thresholding operator and the empirical Wiener rule were able to completely attenuate any musical noise. However in doing so, the soft thresholding operator underfit the data, i.e., did not recover the sound completely. As a result GTFS with soft thresholding yielded a relatively low SNR output of 13.52 dB. The empirical Wiener rule performed the best in this case study, producing a SNR of 17.87 dB. We note that, while the empirical Wiener rule produces the best result for this sound, the same is not necessarily true for a different sound, or when a different dictionary is used.

**Algorithm 2: GTFS Soft Thresholding / Empirical Wiener**

**Task:** Denoise the signal y by $\hat{x} = argmin_z \; \|Dz - y\|_2^2 < \varepsilon$

**Parameters:** D and y are given, k is the iteration count and $\varepsilon$ is a predetermined threshold.

**Initialization:** Set $residual = y, z_k = 0 \; and \; k = 0$

**Threshold Selection:** Determine a threshold value using either *universal, minimax or SURE thresholds*

**Main Iteration:** Set k =1 and apply,

- **Weights**: Calculate correlations between y and D and select the most correlated atom using a standard greedy procedure.
- **Shrinkage**: Shrink the coefficient of the selected atom using either of:
  - Soft Thresholding: $\alpha_S(\alpha_i, \lambda) = sgn(\alpha_i)(|\alpha_i| - \lambda)_+$
  - Empirical Wiener: $\alpha_{EW}(\alpha_i, \lambda) = \alpha_i \left(1 - \left[\frac{\lambda}{|\alpha_i|}\right]^2\right)_+$
- **Return**: Set k = k + 1 and repeat.
- **Stop**: Stop iterating when the change in residual energy falls below the threshold $\varepsilon$.

**Finalize**: The denoised output is $\hat{x} = Dz$.

Algorithm 2 presents an overview of the GTFS denoising algorithm when used in conjunction with either the soft thresholding or empirical Wiener attenuation rules. The only difference with standard GTFS is that the hard thresholding operator is substituted by an alternate attenuation rule. Apart from affecting the contribution of each atom, the choice of attenuation rule also affects the GTFS denoising process in terms of frequency of the atoms selected. Interestingly, the three attenuation rules select mostly the same set of frequencies – roughly 90%. This implies that a very small fraction of time-frequency coefficients influence whether the algorithm will overfit the data or not.

## 4.4 Dynamic Thresholding

Apart from coefficient attenuation, selecting a suitable threshold for denoising is another important factor. In section 4.2.2 we showed that the basic wavelet thresholds suffer from both under and overfitting problems. In the context of GTFS denoising, the wavelet thresholds are *static*, i.e., they remain the same for every iteration. While the threshold remains the same, the GTFS residual signal changes continuously at each iteration. In order to tackle these problems, we propose taking advantage of the iterative structure of GTFS denoising via SURE estimate. We are motivated to consider the SURE estimate for several reasons.

### 4.4.1 Data Dependency

GTFS denoising is an iterative process, and the residual signal changes at every iteration (as an atom is subtracted from it). As more and more deterministic audio content is removed from the residual, it is likely that a correlation threshold computed on the initial residual signal would differ from that computed on an intermediate residual. As GTFS iteratively minimizes the energy of the residual signal in the mean square sense, that is to say, $E\left[e^2(n)\right] \sim 0$. Another way to say this is that the residual energy becomes constant. We hence expect the Stein's Unbiased Risk Estimate on the transform coefficients at each iteration to have a generally decreasing trend.

Fig. 4.6, shows the SURE estimate on the transformed residual at each iteration of GTFS.

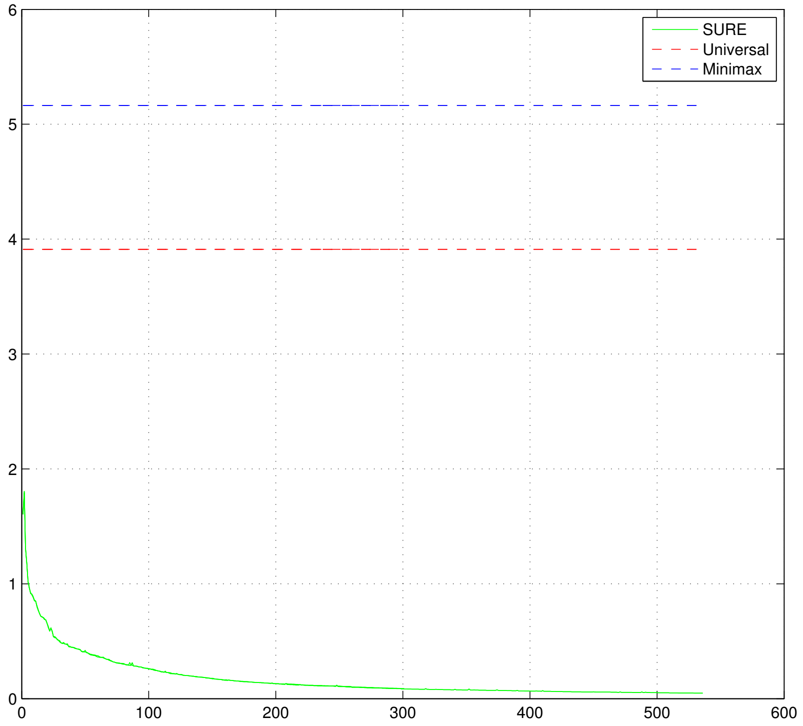


**Fig. 4.6** Change in residual risk during MP decomposition process (Trumpet Audio).

As expected, the SURE threshold computed iteratively on the residual signal shows a decreasing trend.Thus, perhaps unsurprisingly, the *static* wavelet thresholds are not able to perform well, especially in conjunction with the hard thresholding operator. The SURE threshold is data dependent, i.e., it is calculated using the transform coefficients of the noisy signal [51]. Thus it can be used to update threshold values based on the evolving GTFS residual signal.

### 4.4.2 Adapting to Functions of Unknown Smoothness

Donoho and Johnstone made use of Stein's Unbiased Risk Estimate (SURE) in their SUREshrink denoising approach [44]. In this work they introduce the idea of finding level dependent thresholds by considering the different resolution levels of the wavelet transform as independent multivariate normal estimation problems. Stein's unbiased estimate gives an estimate of the risk for a particular threshold value $t$; minimizing this in $t$ gives a selection of the threshold level for that resolution level. Thus, each resolution level has a specific threshold. The main advantage of this method over other wavelet denoising approaches is its ability to adapt to signals of unknown smoothness.

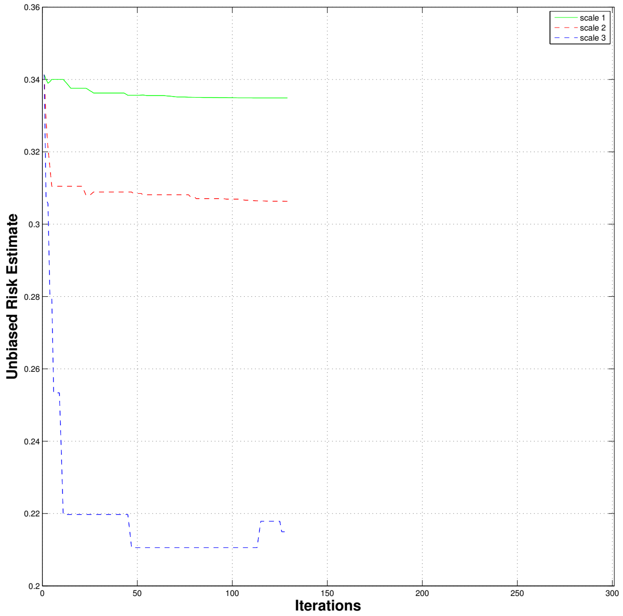


**Fig. 4.7** Evolving SURE threshold for different scales (Trumpet Audio).

In a similar fashion to the wavelet transform, overcomplete dictionaries can also analyze a signal at multiple scales. Fig. 4.7 shows the changing value of the SURE threshold for each scale GTFS decomposition. The audio used to produce this figure is the noisy trumpet from Fig. 4.2. In order to generate the individual curves, the SURE threshold was evaluated independently on the transform coefficients of each scale, at each iteration of GTFS. A three scale dictionary with atoms corresponding to window lengths of 4096, 2048 and 512 samples was used. From the figure we see that the SURE estimate adapts differently according to the scale of the atoms.

### 4.4.3 GreedySURE Thresholding

As highlighted in the introduction to this section, GTFS is an iterative denoising approach and the residual signal keeps changing at each iteration. In order to adapt to this continually evolving residual signal, rather than compute the SURE threshold on initial transform coefficients of the signal alone, we recompute the threshold at every iteration. The blue curve in Fig. 4.8 represents the GreedySURE threshold. In section 4.4.1 we showed that the SURE estimate on the MP residual shows a decreasing trend, and predictably the GreedySURE approach displays similar behaviour.

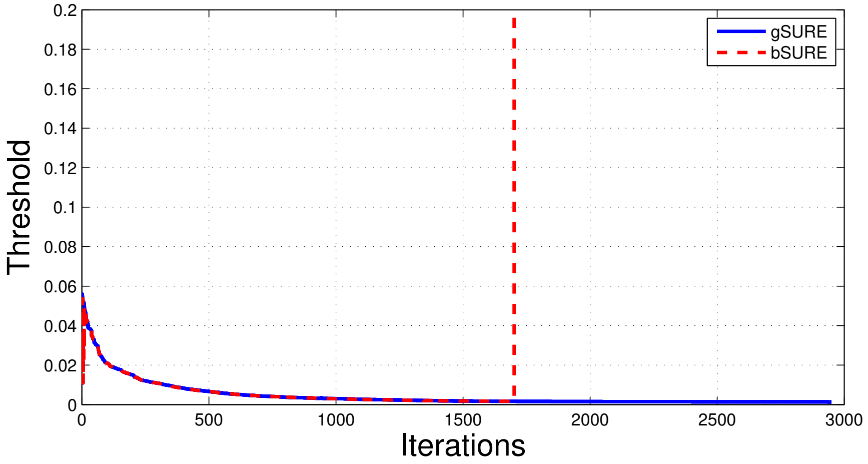


**Fig. 4.8** Comparison of the GreedySURE and BlockSURE thresholds.

In order to illustrate GreedySURE thresholding, we persist with the solo trumpet recording used in previous sections. The audio is corrupted with 5 dB of additive noise. From Fig. 4.9 we see that the denoised spectrogram produced by GTFS in conjunction with GreedySURE thresholding produces a large amount of musical noise artifacts. We hypothesize that one of the possible reasons for this is that the SURE threshold is generally lower than the Universal and Minimax thresholds. This, combined with the fact that the GreedySURE displays a decreasing trend, could potentially lead to dictionary atoms being matched to noisy audio by GTFS. A simple approach to help avoid overfitting would be to simply add a bias value to the GreedySURE threshold at each iteration. From experimental analysis we can confirm that while there is some marginal improvement in performance, this simple approach does not help counter the overfitting problem. Another approach would be to use GreedySURE thresholding in conjunction with attenuation rules apart from hard thresholding. However, once again numerical results have suggested that the thresholding scheme produces musical noise artifacts in this situation as well.

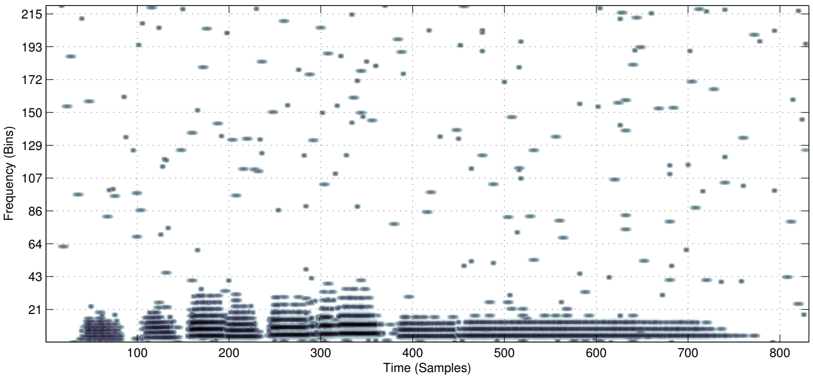


**Fig. 4.9** GreedySURE denoised trumpet spectrogram.

### 4.4.4 BlockSURE Thresholding

The BlockSURE thresholding approach is a refinement on the GreedySURE thresholding approach.The BlockSURE approach aims to take more local information into account while determining threshold values. This is achieved by constructing a *time-frequency block* around the atom selected by GTFS at each iteration. The SURE estimate is then computed on the block to determine a suitable threshold value. For atoms of length 2048 samples, this corresponds to 743 ms time blocks, and a frequency range of 172.26 Hz. The BlockSURE threshold is represented by the red curve in Fig. 4.8, and overlaps with the GreedySURE curve (blue) most of the time. Fig. 4.10 displays the spectrogram of the denoised trumpet sound using the BlockSURE threshold in conjunction with the hard thresholding operator. It is clear that the BlockSURE approach is able to attenuate musical noise to a much larger degree than GreedySURE, although a small amount of visible artifacts remain. The output sound yielded a SNR of 17.53 dB.

Fig. 4.11 displays the spectrogram for the denoised sounds produced with the different attenuation rules. The soft thresholding rule produced very similar results to the hard thresholding operator, with a similar output SNR of 17.64 dB, and a similar amount of

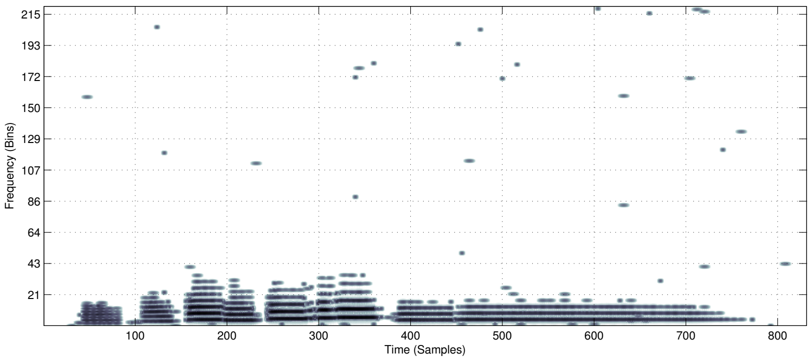


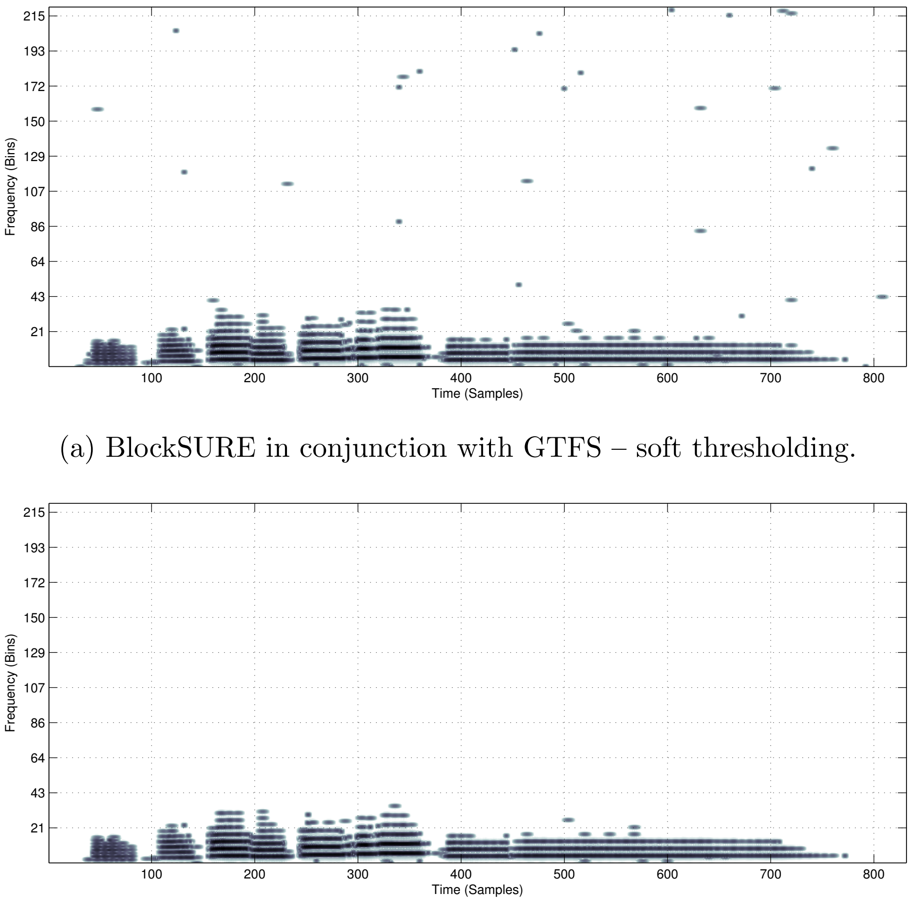
**Fig. 4.10** BlockSURE in conjunction with GTFS – hard thresholding.

musical noise artifacts. The Empirical Wiener attenuation rule was able to yield the best result in terms of musical noise attenuation, and produced an output SNR of 17.34 dB.

Fig. 4.8 compares the threshold values generated by the GreedySURE and BlockSURE thresholding approaches. It can be seen that the two curves overlap each other until approximately 1650 iterations, at which point GTFS with BlockSURE thresholding stops the denoising process. At its last iteration, BlockSURE produced an extremely high threshold, indicated by the vertical dashed line, which caused the process to stop. We hypothesize that this may occur, when a considered time-frequency block contains a large number of atoms that are matched to noisy audio. On the other hand, GreedySURE thresholding works with the entire residual signal, and is unable to increase its threshold value once the deterministic audio components have been recovered. As a result, GreedySURE thresholding is more inclined to produce musical noise artifacts than the BlockSURE approach. From numerical experiments we can confirm that this trend is followed for different kinds of musical audio with varying noise level.

### Further Enhancements

A potential extension of the BlockSURE approach involves computation of a priori SNR values for each block, as demonstrated by Cai [52]. Such a denoising framework would resemble the audio Block Thresholding algorithm [53], which has been shown to greatly attenuate musical noise artifacts. Another way in which the approach can potentially be



(a) BlockSURE in conjunction with GTFS – soft thresholding.

(b) BlockSURE in conjunction with GTFS – Empirical Wiener.

**Fig. 4.11** BlockSURE thresholding with different attenuation rules.

enhanced is by creating molecules [54], or collections of atoms that are subtracted from the residual signal at the same time. By building a molecule from within a time-frequency block, each atom of the molecule can be attenuated based on the block threshold and a priori SNR. Each of these enhancements will be considered in future works.

### 4.4.5 NaiveSURE Thresholding

The NaiveSURE thresholding approach aims to take advantage of the evolving nature of the GTFS residual signal. However, rather than analyze the signal in the spectral domain like GreedySURE, the NaiveSURE approach calculates threshold values based on the time-domain residual signal. A noteworthy aspect of the NaiveSURE thresholding approach is that the algorithm does *not* require noise variance as an input parameter, making the approach more robust.

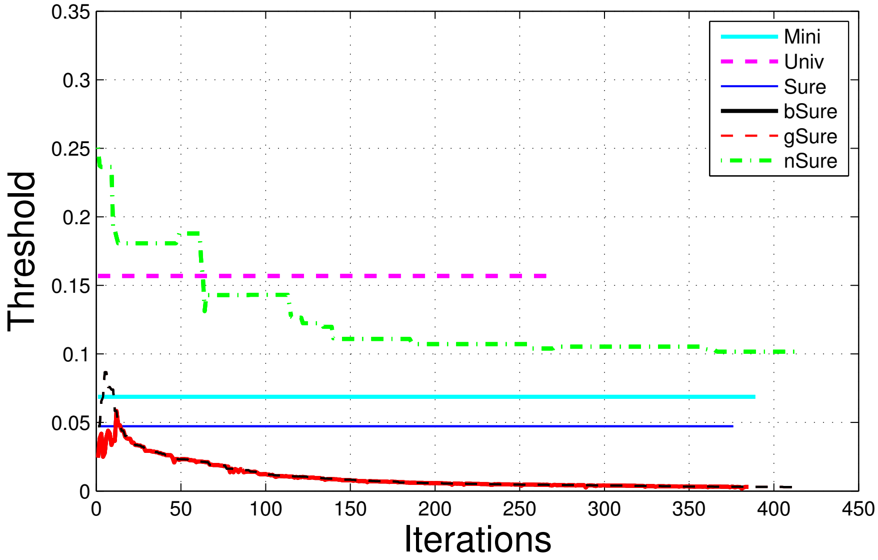


**Fig. 4.12** Evolution of static and dynamic thresholds over the course of a GTFS denoising process (Trumpet Audio).

Fig. 4.12 compares the different static and dynamic thresholds over the course of a GTFS denoising process. It can be seen that the NaiveSURE threshold is considerably higher than the GreedySURE and BlockSURE dynamic thresholds. Interestingly, the NaiveSURE threshold starts out with a higher value than the Universal threshold, and ends up at a value that is somewhere in between the Universal and Minimax thresholds for the signal. Algorithm 3 presents an overview of the GTFS algorithm in conjunction with the various dynamic thresholding strategies. The main difference between this method and the static thresholding approaches is that the threshold calculation step is included inside the main loop, i.e., thresholds are re-computed at every iteration. Fig. 4.13 displays the denoised spectrogram of the audio output by GTFS using NaiveSURE thresholding. Notably, the approach is the only one to *not* produce any musical noise artifacts when used in conjunction with the hard thresholding operator. While this behaviour is partly explained by the fact that the NaiveSURE threshold is higher than the GreedySURE threshold, we note that simply adding a bias value to the GreedySURE threshold was unable to improve its performance. This point only serves to further highlight the adaptive nature of the NaiveSURE approach.

**Algorithm 3: GTFS Dynamic Thresholding**

**Task:** Denoise the signal y by $\hat{x} = argmin_z \,.\|Dz - y\|_2^2 < \varepsilon$

**Parameters:** D and y are given, k is the iteration count and $\varepsilon$ is a predetermined threshold.

**Initialization:** Set $residual = y, z_k = 0 \; and \; k = 0$

**Main Iteration:** Set k =1 and apply,

- **Threshold**:
  - GreedySURE: Determine SURE threshold on transform coefficients.
  - BlockSURE: Determine SURE threshold on time-frequency block.
  - NaiveSURE: Determine SURE threshold on time-domain residual signal.
- **Weights**: Calculate correlations between y and D and select the most correlated atom using a standard greedy procedure.
- **Shrinkage**: Shrink the coefficient of the selected atom with the SURE estimate on the residual using either the hard or soft thresholding operators.
- **Return**: Set k = k + 1 and repeat.
- **Stop**: Stop iterating when the change in residual energy falls below the threshold $\varepsilon$.

**Finalize**: The denoised output is $\hat{x} = Dz$.

In Fig. 4.14 both GreedySURE and NaiveSURE thresholding have been used in conjunction with the soft thresholding operator. As the NaiveSURE threshold is larger than GreedySURE, the approach expectedly causes greater attenuation to the atomic coefficients. We recall that GreedySURE thresholding in conjunction with soft thresholding was not able to attenuate musical noise artifacts. The NaiveSURE threshold with soft thresholding on the other hand was able to attenuate musical noise to a large extent. Fig. 4.15 displays the spectrograms of the denoised sounds output by the NaiveSURE approach, in conjunction

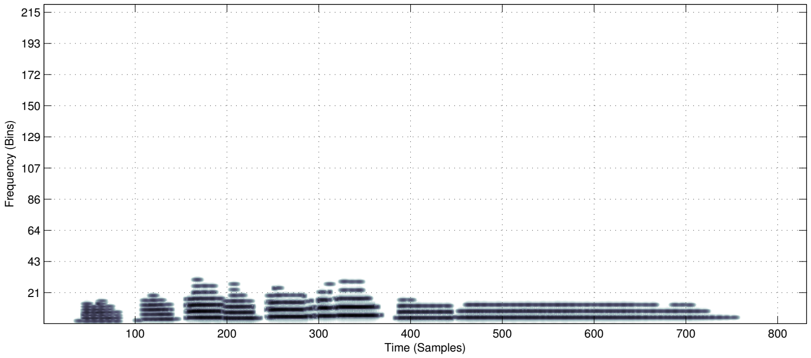


**Fig. 4.13** NaiveSURE in conjunction with GTFS – hard thresholding.

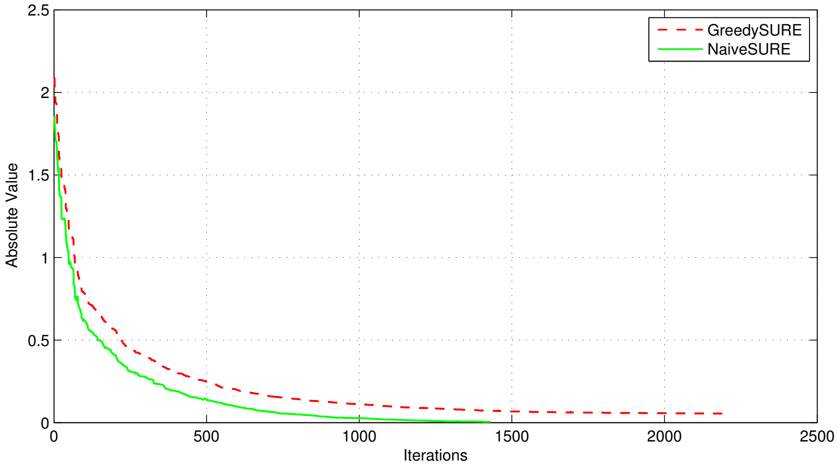
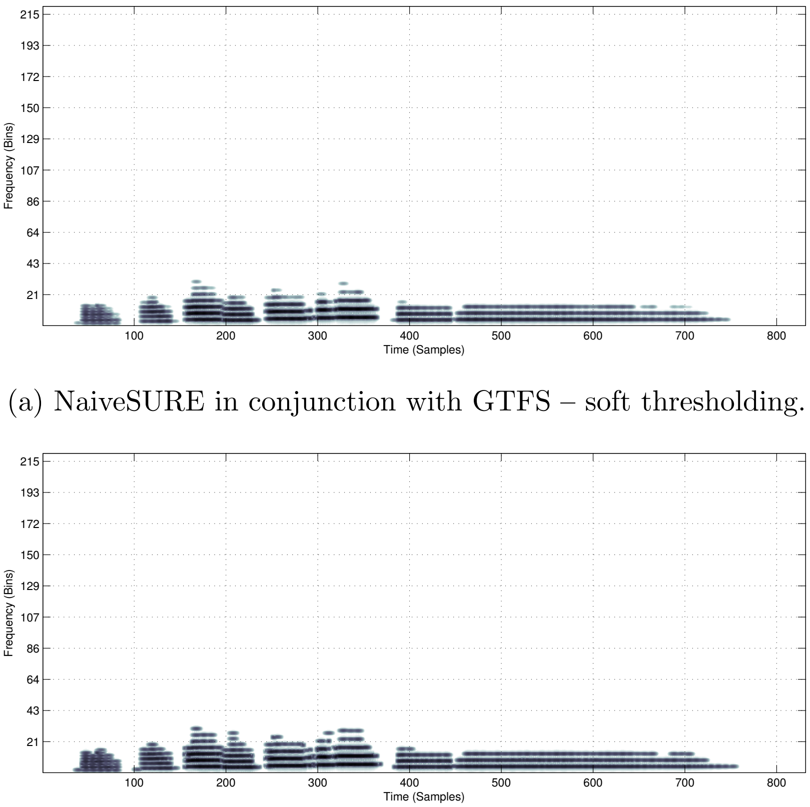


**Fig. 4.14** Comparison of atomic coefficients selected by different dynamic thresholding strategies.

with the soft thresholding and empirical Wiener rules respectively.

Although not clearly visible from the figure, the soft thresholding operator underfits the data, achieving an output SNR of 12.25 dB. The same can be said about the hard thresholding operator and Empirical Wiener rules, which produced SNR's of 16.10 dB and 14.40 dB respectively. These results are considerably less than the BlockSURE approach, which yielded a SNR of 17.44 dB. While the NaiveSURE approach does not produce optimal results in terms of SNR, the advantage of the approach lies in the fact that it is more robust to musical noise. In the event that denoised sounds with higher SNR are required, better



(a) NaiveSURE in conjunction with GTFS – soft thresholding.

(b) NaiveSURE in conjunction with GTFS – Empirical Wiener filtering.

**Fig. 4.15** NaiveSURE thresholding with different attenuation rules.

results can be achieved by increasing the redundancy of the dictionary.

## 4.5 Summary

In this chapter we presented Greedy Time-Frequency Shrinkage (GTFS), which interprets MP-based denoising as a simple shrinkage approach. We analyzed different aspects of the GTFS approach, including coefficient thresholds, shrinkage operators and signal representation via redundant dictionaries. Having identified the factors on which GTFS denoising depends we developed model enhancements in order to ameliorate the algorithm's denoising performance. Through these enhancements we were able to show that GTFS denoising is able to recover deterministic audio content from noisy signal without the creation of musical noise artifacts. In the next chapter we will perform comprehensive testing of the GTFS approach with a variety of musical sounds and speech. So far we have experimented much with the dictionary in terms of composition or redundancy. We will explore these aspects of GTFS denoising in greater detail in the next chapter.

# Chapter 5

# Experiments and Results

## 5.1 Introduction

In the previous chapter we presented GTFS denoising, which combines various aspects of simple shrinkage with Matching Pursuit-based greedy atomic decomposition. The unpredictable nature of the decomposition process makes experimental testing and analysis an important aspect of developing and improving the approach. In this chapter we present extensive testing and analysis of GTFS denoising on a wide range of audio, including a broad range of musical sounds, and speech. Quantitative data is gathered in terms of output SNR and PEAQ scores. Apart from these performance metrics, we analyze the output (denoised) signal representations in terms of sparsity as well. Along with GTFS, the results of two other state-of-the-art denoising algorithms are also presented.

## 5.2 Comparison of Denoising Algorithms

For the purpose of comparison we will test the performance of the algorithms developed in the previous chapter with two other denoising approaches that also make use of redundant dictionaries – the audio Block Thresholding (BT) approach [53], and the Persistent Empirical Wiener filtering (PEW) approach [18]. The Block Thresholding has been shown to outperform several well established audio denoising approaches and may be considered as the state of the art in audio noise reduction. More recently, Persistent Empirical Wiener filtering was introduced, and was shown to produce comparable results to Block Thresholding. The approach is based on the idea of mixed norms and structured sparsity [55]. To our knowledge this is the only denoising approach that produces a sparse solution – with approximately 10-15 percent of the transform coefficients being retained in the denoised signal. In our experiments with the two algorithms, Block Thresholding consistently performs better than the Empirical Wiener approach in terms of SNR. It should be noted that the difference is usually small and often marginal. At the same time, the PEW approach outperforms BT in terms of PEAQ scoring. Again, the difference is usually quite small. An important point to note is that at higher noise levels, the Block Thresholding solution produces noticeable artifacts, for certain sounds. That being said, from informal listening tests we can confirm that the approach produces excellent results for most sounds. On the other hand, the PEW approach produces much fewer of these kinds of artifacts. We feel that this point is an important factor in the context of our work. We will show in the following sections of this chapter, that GTFS can be tailored to produce different denoised results. For instance, sometimes it may be desirable to obtain the most compact denoised result possible. We will also show that GTFS denoising can produce results that are comparable to those of both Block Thresholding and Persistent Empirical Wiener filtering in terms of PEAQ and SNR, while being significantly more compact or sparse.

## 5.3 Testing & Analysis

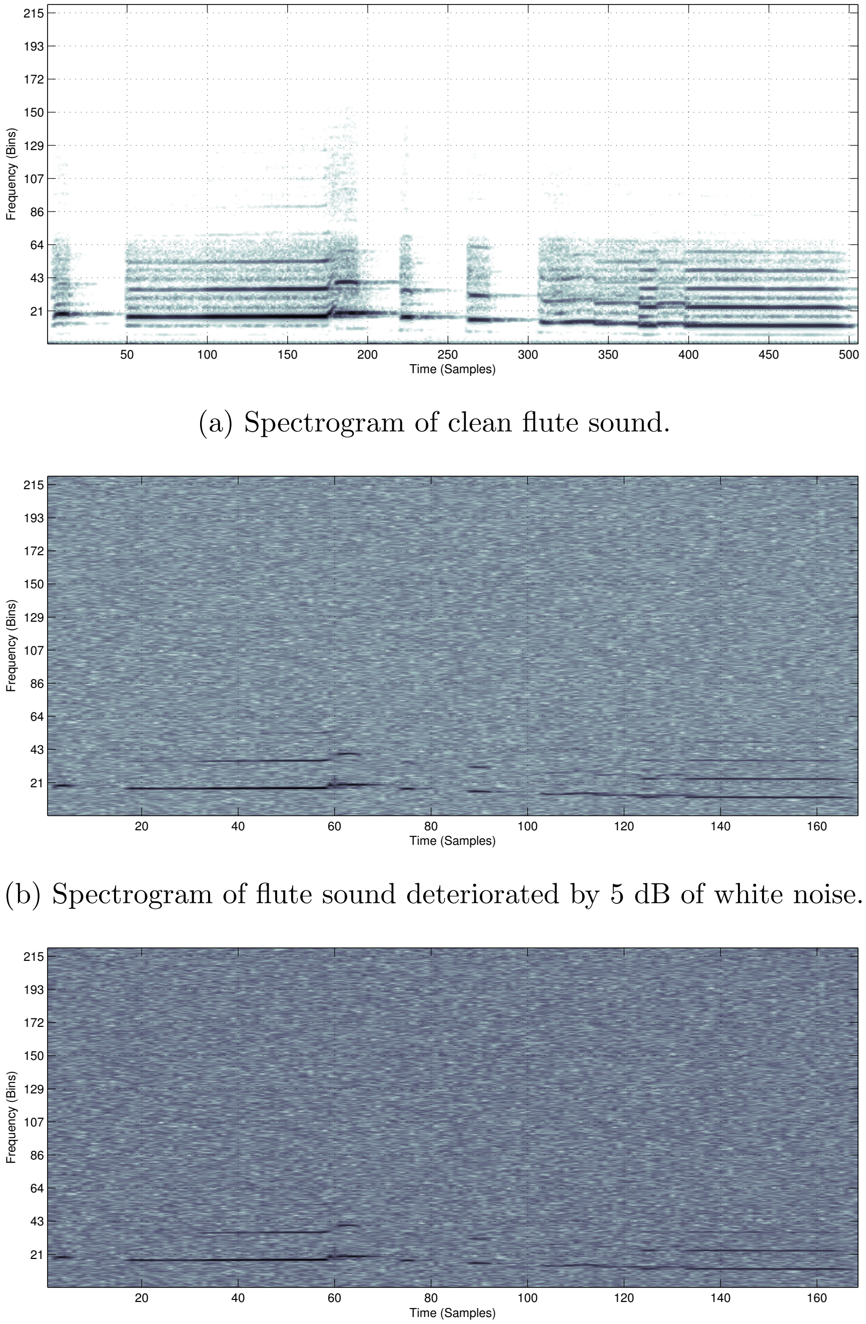
An important factor to consider while designing an application to process musical audio, is the diverse and variable nature of these signals. These features make musical audio quite challenging to work with. In order to account for this variability, we test GTFS denoising extensively, on a broad range of sounds, in order to be able to make some generalizations about the approach. Testing is performed on recordings of wind and brass instruments, plucked and bowed strings, and human voice. All the audio is sampled at 44100 Hz. Testing is performed at noise levels of 5 dB and 0 dB respectively.

### 5.3.1 Dictionary Design

Throughout this work we have stressed on the importance of the dictionary in the greedy atomic decomposition process. The strategies developed in Chapter 4 are aimed at reducing the dependence of GTFS on the dictionary. Provided the algorithm does not underfit the data, a sparse denoised result generally implies a denoised sound of good subjective quality. As mentioned previously, the redundancy of the dictionary is intrinsically linked to signal sparsity. Based on numerical experiments we report that audio signal like music and speech are best represented by multi-scale redundant dictionaries that oversample the time-frequency plane. Using dictionaries of this kind often leads to extremely sparse denoised signal representations. In the case of audio denoising, an important point to note is that the dictionary used from sound to sound can change. While this is not the case for the majority of the sounds tested in this work, we make note of the situations in which it is. We primarily make use of two and three scale redundant dictionaries of STFT atoms, with atomic lengths corresponding to 8192, 4096 and 2048 samples. As stated previously, shorter atoms are more susceptible to being matched to noisy audio content.

### 5.3.2 Wind & Brass

We begin our testing of musical sounds by considering the family of wind and brass instruments. For our testing we will consider recordings of the flute, saxophone and trumpet. These instrumental sounds are characterized by a large amount of tonal content, and audio components related to performer breathing. In the context of noise reduction, it becomes a challenge to differentiate between the noise that needs to be removed, and the components of the sound that are related to performer's breathing. While a relatively accurate denoised reconstruction is relatively simple to achieve, capturing aspects of the breath noise is essential for the denoised result not to sound synthetic. Fig. 5.1 displays the spectrogram of the flute sample used in our testing is displayed. The piece is a short monophonic recording. In this work we have restricted our testing to monophonic audio, in order to better understand the behaviour of MP-based audio denoising techniques. We will consider polyphonic recordings in future works. The lower two figures show the spectrograms for the flute signal when it is corrupted with white noise of varying intensities. It is clear from the figure that depending on the noise level, several of the spectral components of the original sound are either partially or completely blurred out by the noise.



(a) Spectrogram of clean flute sound.

(b) Spectrogram of flute sound deteriorated by 5 dB of white noise.

(c) Spectrogram of flute sound deteriorated by 0 dB of white noise.

**Fig. 5.1** Clean and noisy flute spectrograms.

In terms of setting up the experiment, our main concern involves designing the dictionary. As mentioned earlier, wind instrument sounds have large tonal content and we accordingly choose a dictionary that is suited for the extraction of tonal content.We hence construct a multi scale Gabor dictionary containing atoms corresponding to window lengths of 8192, 4096 and 2048 samples respectively. It should be noted that we will consider the same dictionary for the two different noise conditions.

| **SNR** | **Flute** | **Sax** | **Trumpet** |
|---|---|---|---|
| **GTFS** | 19.21 | 17.37 | 18.06 |
| **BT** | 20.31 | 19.76 | 19.48 |
| **PEW** | 18.67 | 16.60 | 17.22 |

| **PEAQ** | **Flute** | **Sax** | **Trumpet** |
|---|---|---|---|
| **GTFS** | -2.62 | -2.21 | -2.24 |
| **BT** | -3.06 | -2.94 | -2.66 |
| **PEW** | -2.39 | -2.02 | -1.97 |

**Table 5.1** Comparison of output SNR & PEAQ scores. Noise level = 5 dB.

Table 5.1 compares the output SNR and PEAQ scores produces by GTFS. The results of the audio Block Thresholding and Persistent Empirical Wiener denoising approaches have also been provided. From the results we see that Block Thresholding consistently yields the highest SNR, although the difference between the three algorithms is minimal. Notably, the Block Thresholding approach produces audible low frequency artifacts in its denoised sounds. Both PEW and GTFS denoising do not produce these kinds of artifacts. This perhaps explains why both GTFS and PEW filtering perform better than Block Thresholding in terms of PEAQ scores.

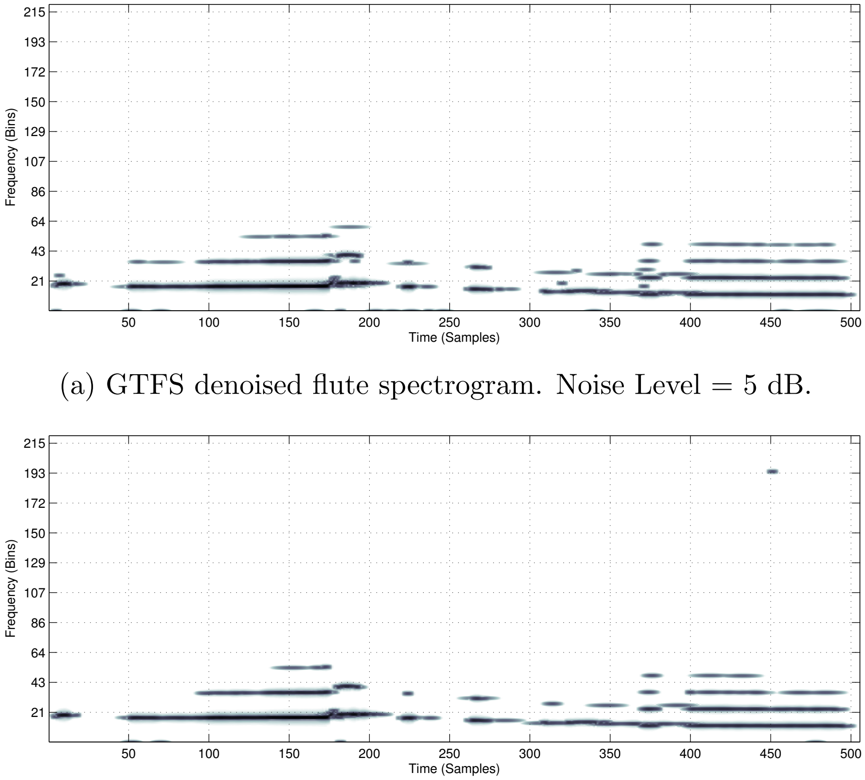
Table 5.2 displays the denoising results for a noise level of 0 dB. We see that the algorithms follow the same trend as in the 5 dB noise case. Again, BT is able to consistently achieve the highest SNR, while GTFS and PEW filtering yield better PEAQ scores. From the tables we notice that GTFS denoising produces results that are somewhat in between the Block Thresholding and Empirical Wiener approaches.

Fig. 5.2 displays the output spectrograms produced by GTFS at different noise levels. Notably, we see that in both noise conditions GTFS is able to greatly attenuate any musical noise artifacts. We also note that GTFS was able to attenuate musical noise in a similar

| SNR | Flute | Sax | Trumpet |
|---|---|---|---|
| GTFS | 16.47 | 14.85 | 15.21 |
| BT | 17.09 | 16.22 | 16.45 |
| PEW | 16.20 | 12.45 | 13.72 |

| PEAQ | Flute | Sax | Trumpet |
|---|---|---|---|
| GTFS | -2.83 | -2.55 | -2.52 |
| BT | -3.36 | -3.17 | -2.87 |
| PEW | -2.64 | -2.41 | -2.30 |

**Table 5.2** Comparison of output SNR & PEAQ scores. Noise level = 0 dB.



(a) GTFS denoised flute spectrogram. Noise Level = 5 dB.

(b) GTFS denoised flute spectrogram. Noise Level = 0 dB.

**Fig. 5.2** GTFS denoised flute spectrograms.

fashion for the trumpet and saxophone recordings. Throughout this work, we have stressed on the fact that one of the main features of working with redundant dictionaries is their ability to produce sparse signal representations. In table 5.3 we document the average sparsity of the signal representations produced by the three algorithms. We recall that by sparsity, we mean the number of non-zero transform signal coefficients in the denoised output sound.

| Algorithm | 5 dB Noise | 0 dB Noise |
|---|---|---|
| BT | 97.62 | 94.45 |
| PEW | 14.33 | 11.04 |
| GTFS | 00.19 | 00.07 |

**Table 5.3** Average number of transform coefficients retained (%).

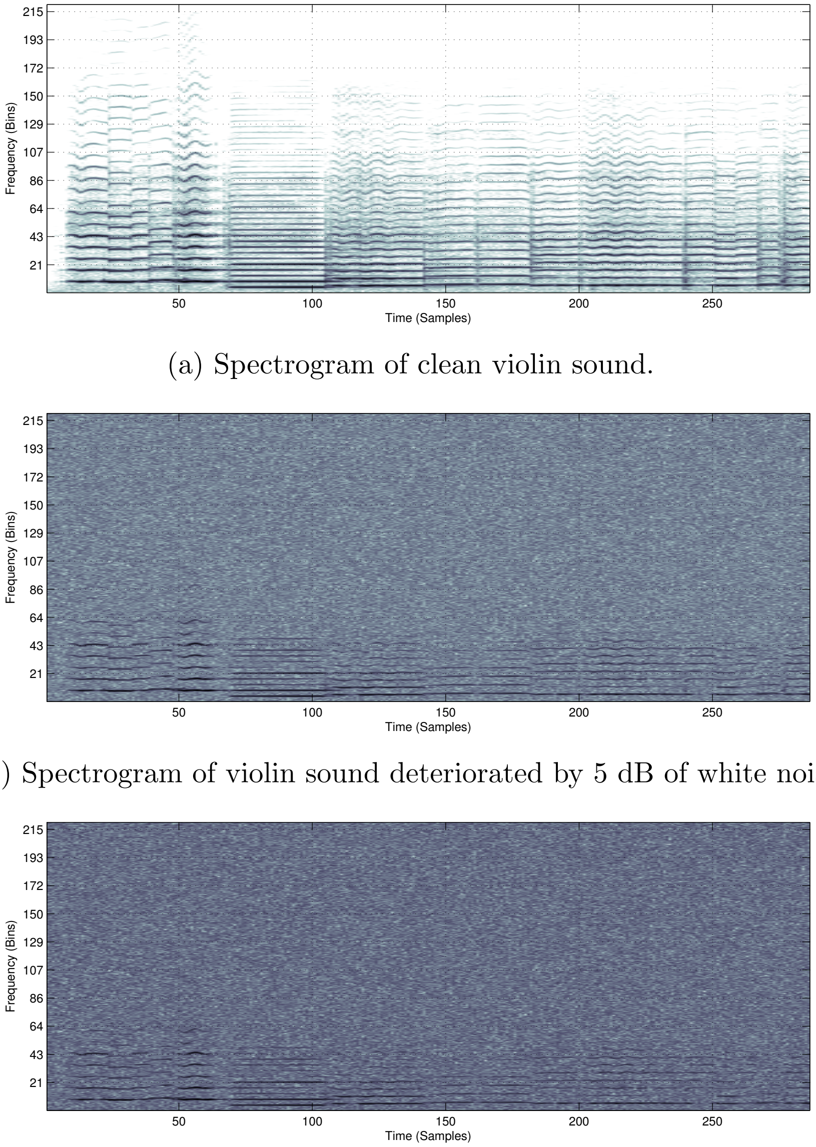
From Tables 5.1, 5.2 & 5.3, we see that GTFS denoising produces competitive results with both BT and PEAQ in terms of SNR and PEAQ scores, at different noise levels. While there is much debate regarding the merits of different performance metrics, we can confirm that the denoised results produced by all three algorithms are of comparable quality[1]. This assessment is based on informal listening tests. Notably, GTFS retains a significantly smaller number of the transform signal coefficients than both BT and PEW, retaining between 0.07-0.19% of the transform coefficients. In our testing the PEW denoising approach retained 11.04-14.33% of the coefficients on average. BT is not a sparse denoising approach and retains most of the transform coefficients.

### 5.3.3 Plucked & Bowed Strings

The family of string instruments is able to produce a wide range of sounds. String instruments can essentially be divided into two sub-families – plucked and bowed strings. While plucked strings sounds are relatively simple to represent, bowed strings present a more challenging problem. This is because bowed strings are often associated with noisy transient audio content. We will show that accurate recovery of these kinds of sounds requires additional processing. In this section we consider violin and cello for bowed strings, and the piano in the case of plucked strings. Both kinds of sounds have some similar characteristics – large tonal content, with transient parts around the note attacks. Fig. 5.3 displays the spectrogram of the clean violin recording, as well those of the clean audio deteriorated by

[1] denoised examples of all three approaches are available at :
`http://music.mcgill.ca/~gautam/ICASSP/icassp2.html`

noise of varying intensities. Unlike the flute recording in the previous section, the violin spectrogram contains a large number of short vertical spectral lines. This confirms the presence of transient content.



(a) Spectrogram of clean violin sound.

(b) Spectrogram of violin sound deteriorated by 5 dB of white noise.

(c) Spectrogram of violin sound deteriorated by 0 dB of white noise

**Fig. 5.3** Clean and noisy violin spectrograms.

In terms of time-frequency representation, atoms of short length have been shown to be best suited for transient audio content. This could be a challenge in a signal denoising setting, because, as mentioned previously, short atoms are more susceptible to being matched to noisy audio by GTFS. From numerical experiments we have found that when the noisy signal contains a *significant* amount of transient audio content, GTFS is still able to match them to deterministic audio and avoid recovery of noisy content. After experimenting with a number of dictionary configurations, we settled on a two-scale dictionary with atoms corresponding to window lengths 2048 and 256 samples respectively – long and short atoms to represent the tonal and transient components of the sound.

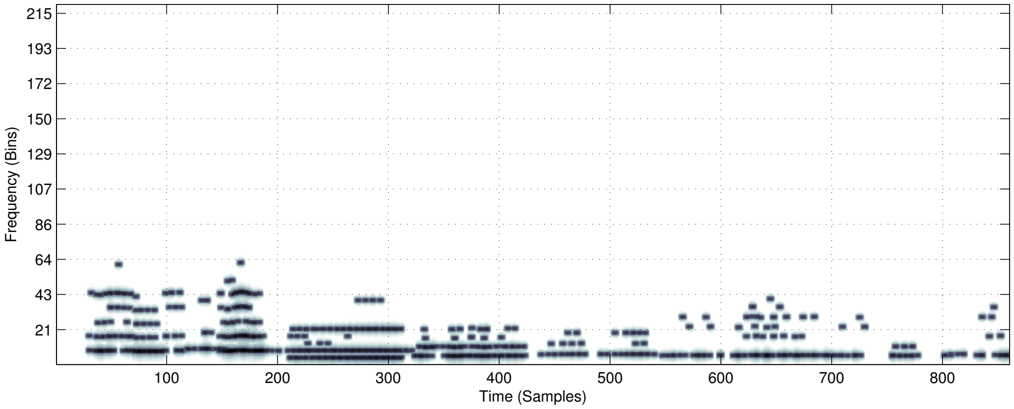
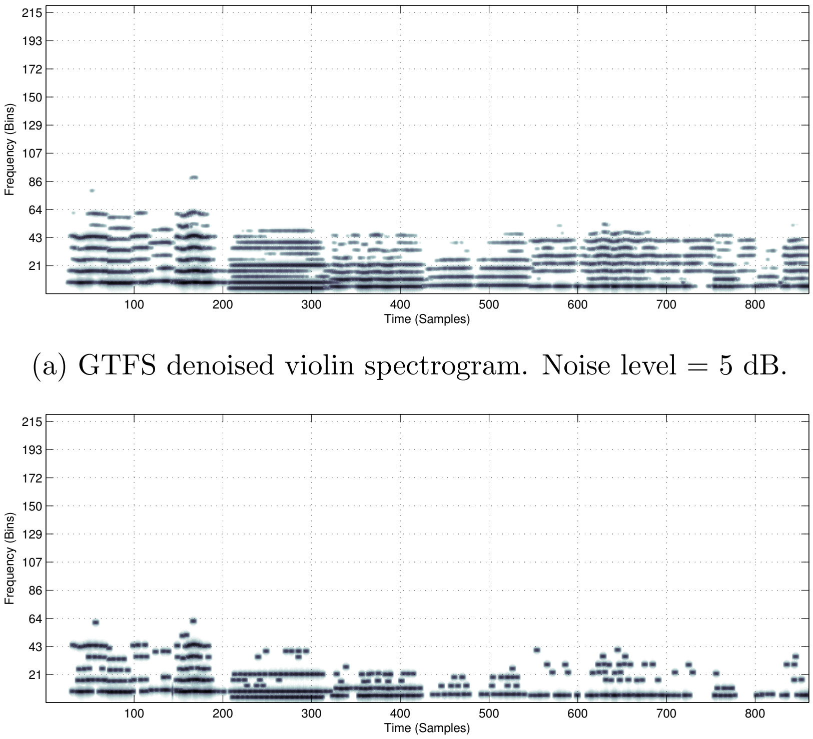


**Fig. 5.4** GTFS denoised violin spectrogram (single processing).

Comparing Fig. 5.4 to Fig. 5.3a, it is clear that GTFS has not been able to recover all the deterministic audio content. That is, the two scales included in the dictionary were not sufficient to represent the signal. We note that amongst all the sounds tested in this work, this was the only one for which GTFS consistently underfit the data. The SNR of the denoised sound is 8.95 dB. In order to tackle the underfitting problem, we devised a simple approach. After the GTFS denoising process is completed, we run the algorithm again using a new or enhanced dictionary. From numerical experiments we found that using a single scale dictionary yielded the best results. GTFS is run on the residual signal over and over, until signal recovery is deemed to be sufficient. At each instance, GTFS makes use of a different dictionary. Fig. 5.5 displays the spectrograms of the denoised sounds obtained after repeated residual processing. In the 5 dB noise case, GTFS was run on the residual signal 12 times, increasing the SNR to 14.21 dB. Interesting, GTFS was only run 3 times in the 0 dB noise case, achieving an output SNR of 9.02 dB.



(a) GTFS denoised violin spectrogram. Noise level = 5 dB.

(b) GTFS denoised violin spectrogram. Noise level = 0 dB.

**Fig. 5.5** GTFS denoised violin spectrograms at different noise levels (repeated processing).

This example brings up an interesting point regarding time-frequency signal representation. The original two-scale dictionary we made use of would be adequate to accurately represent the clean violin recording, with the longer atoms being matched to the tonal content and the shorter atoms to the transient content. However in the context of signal denoising, this same dictionary proves to be insufficient to *recover* the audio. The results of this experiment suggest that audio recovery from noise is strongly influenced by transient audio content. Tables compare the output SNR and PEAQ scores of GTFS with BT and PEW filtering on the different string instruments at different noise levels. The string instruments display a similar trend as the wind and brass instruments in terms of algorithmic performance. Block Thresholding consistently achieved the highest SNR while GTFS and Persistent Empirical Wiener filtering yield better PEAQ scores. Once again, all differences remain small to marginal. There are no audible musical noise artifacts in the denoised results of each algorithm, however the Block Thresholding approach produces low frequency artifacts.

| SNR | Violin | Cello | Piano |
|---|---|---|---|
| GTFS | 14.18 | 17.54 | 16.74 |
| BT | 14.87 | 15.92 | 18.40 |
| PEW | 14.10 | 14.22 | 16.56 |

| PEAQ | Violin | Cello | Piano |
|---|---|---|---|
| GTFS | -3.01 | -2.47 | -2.42 |
| BT | -3.63 | -3.01 | -3.04 |
| PEW | -2.83 | -2.12 | -2.20 |

**Table 5.4** Comparison of output SNR & PEAQ scores. Noise level = 5 dB.

In terms of output signal sparsity, the trend is also the same. GTFS produces significantly sparser denoised signal representations than either Block Thresholding or Persistent Empirical Wiener filtering. The PEW approach retains between 12-14% of the transform signal coefficients, while Block Thresholding retained 96-98%. GTFS denoising retained 0.3-2.5% of the transform signal coefficients. We note that the violin recording which required repeated processing was the least sparse result produced in this work, retaining 2.5% of the transform coefficients in the denoised signal representation.

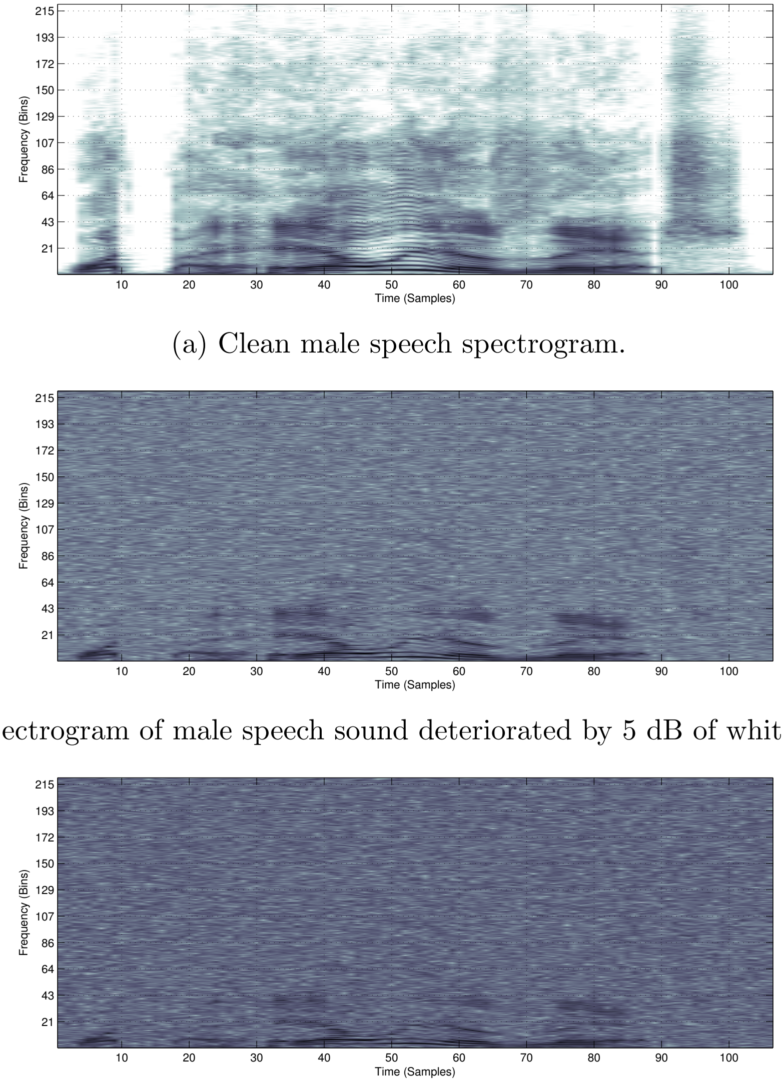
| SNR | Violin | Cello | Piano |
|---|---|---|---|
| GTFS | 09.73 | 12.19 | 15.18 |
| BT | 12.34 | 13.68 | 13.72 |
| PEW | 10.04 | 10.68 | 12.84 |

| PEAQ | Violin | Cello | Piano |
|---|---|---|---|
| GTFS | -1.78 | -2.51 | -2.58 |
| BT | -3.40 | -3.13 | -3.06 |
| PEW | -2.68 | -2.15 | -1.95 |

**Table 5.5** Comparison of output SNR & PEAQ scores. Noise level = 0 dB.

### 5.3.4 Speech

In this case study we also tested the GTFS denoising approach on noisy speech recordings. Speech enhancement has been an active field of research for the last thirty years, with several pioneering noise reduction algorithms emerging from this domain. In terms of dictionary design, we recall that we make use of STFT atoms in this work. The sinusoidal model has been extensively researched in the context of speech [56], and hence STFT atoms should be able to accurately represent speech sounds. We use a two-scale dictionary with atoms corresponding to window lengths of 4096 and 2048 samples respectively.



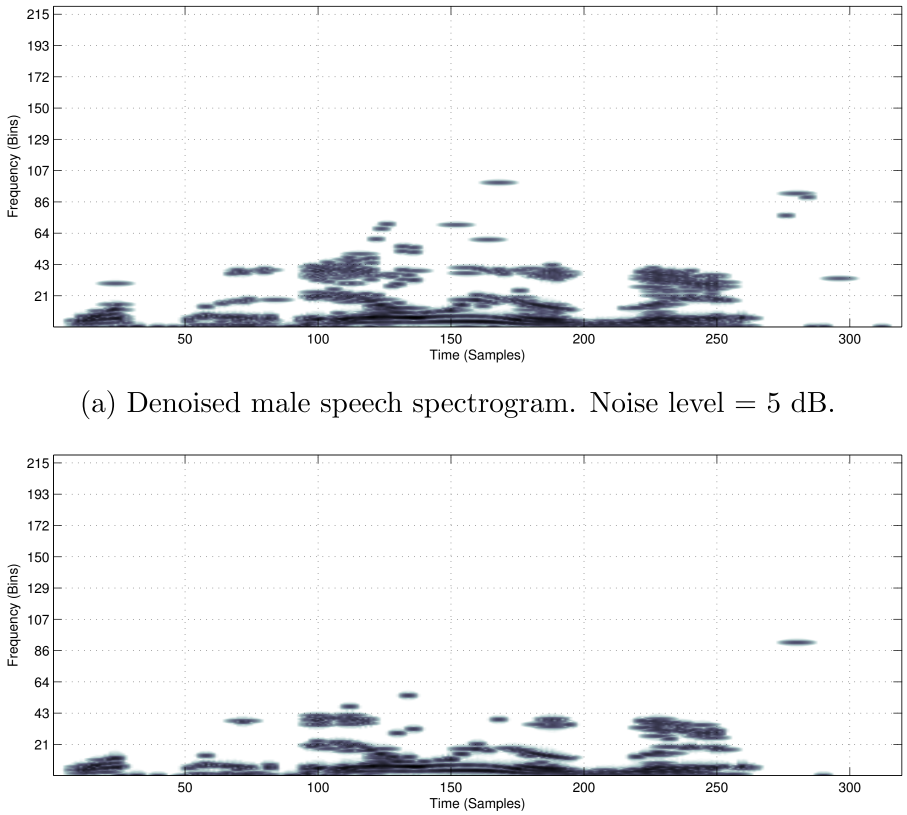
(a) Clean male speech spectrogram.

(b) Spectrogram of male speech sound deteriorated by 5 dB of white noise.

(c) Spectrogram of male speech sound deteriorated by 0 dB of white noise.

**Fig. 5.6** Clean and noisy male speech spectrograms.

Fig. 5.6a displays the spectrogram of a male speech recording. Unlike the musical sounds tested in the previous sections, speech sounds have well defined formant structures, as is visible from the figure. The lower two figures display the same speech spectrograms submerged in noise of varying intensities.



(a) Denoised male speech spectrogram. Noise level = 5 dB.

(b) Denoised male speech spectrogram. Noise level = 0 dB.

**Fig. 5.7** GTFS denoised male speech spectrograms.

Fig. 5.7 displays the spectrograms of the denoised sounds produced by GTFS. As was the case with musical sounds, GTFS was able to greatly attenuate any musical noise artifacts. In the 5 dB noise case GTFS yielded a SNR of 12.23 dB, while in the 0 dB case produced an output SNR of 9.93 dB.

In terms of comparison with Block Thresholding and Persistent Empirical Wiener filtering, the denoising results follow the same trend as for the musical sounds. Block Thresholding once again achieves the highest SNR, while GTFS and PEW filtering yield better PEAQ scores. The difference between the three approaches remains marginal.

| SNR | M-Voice | F-Voice |
|---|---|---|
| **GTFS** | 12.00 | 15.60 |
| **BT** | 12.94 | 16.95 |
| **PEW** | 11.26 | 15.28 |

| PEAQ | M-Voice | F-Voice |
|---|---|---|
| **GTFS** | -2.55 | -2.29 |
| **BT** | -2.92 | -3.30 |
| **PEW** | -2.39 | -2.36 |

**Table 5.6** Comparison of output SNR & PEAQ scores. Noise level = 5 dB.

The denoised speech sounds also show a similar trend in terms of signal sparsity, with GTFS retaining a significantly smaller percentage of the transform signal coefficients than both Block Thresholding and Persistent Empirical Wiener filtering.

| SNR | M-Voice | F-Voice |
|---|---|---|
| **GTFS** | 09.26 | 12.20 |
| **BT** | 10.30 | 13.74 |
| **PEW** | 08.50 | 11.89 |

| PEAQ | M-Voice | F-Voice |
|---|---|---|
| **GTFS** | -2.00 | -2.17 |
| **BT** | -2.86 | -3.31 |
| **PEW** | -2.36 | -2.42 |

**Table 5.7** Comparison of output SNR & PEAQ scores. Noise level = 0 dB.

# Chapter 6

# Conclusions

## 6.1 Introduction

This research presented an analysis of greedy atomic decomposition in the context of audio signal denoising. While classical audio denoising approaches take advantage of this fact by attenuating noise in the transform domain without affecting the deterministic audio, Greedy Time-Frequency Shrinkage (GTFS) tries to recover the deterministic audio by means of a redundant dictionary. GTFS is an interpretation of Matching Pursuit denoising (based on correlation thresholding) as a simple shrinkage approach. This enables us to systematically analyze the algorithms denoising performance, and allows us to embed different aspects of wavelet shrinkage into GTFS denoising. We now summarize our key findings, and propose directions for future research.

## 6.2 Summary

As highlighted throughout this work, one of the primary considerations in GTFS denoising is the redundant dictionary used for signal transformation. As the quality of denoised results depends on the algorithms ability to recover the useful parts of the audio, the dictionary *needs* to be able to accurately represent the signal. If not, GTFS will underfit, or incompletely recover the signal. We note that while this situation can be improved by running the algorithm again on the residual signal (as we did for the violin example in chapter 5), our goal remains to develop a denoising approach that is able to accurately recover the sound through a single processing. The underfitting problem can be mitigated by using highly redundant dictionaries, however as we showed with the violin example, this is not always the case. On the other hand, stopping the decomposition process before noisy audio components are transformed, is a more challenging problem. As previously highlighted, when GTFS is not stopped in time, it results in musical noise artifacts in the denoised sound.

In order to improve the performance and robustness of GTFS, we introduced several model enhancements. The first was based on substituting the hard thresholding attenuation rule used in basic GTFS with alternate attenuation rules. Both the soft thresholding rule and empirical Wiener rules perform better than hard thresholding, and produce better denoising results. The second model enhancement we introduced was a dynamic thresholding approach that makes the GTFS approach more signal adaptive. By taking advantage of the iterative structure of GTFS, we developed a thresholding strategy that adapts to the evolving GTFS residual signal. Through numerical experiments we showed that these enhancements were able to greatly improve the performance of GTFS, particularly in the context of musical noise attenuation.

One of the main features of GTFS denoising is the *sparse* nature of the output representation it produces. On average, the algorithm retains less than 2% of a signal transform coefficients. This is significantly sparser than the audio Block Thresholding (BT) and Persistent Empirical Wiener (PEW) algorithms tested in this work. More importantly, the results produced by GTFS are comparable to those of BT and PEW in terms of SNR and PEAQ scores. While this is the case, we note that GTFS is more sensitive to the specificities of the data than the other algorithms we tested. By this we mean that underfitting or overfitting of the data by GTFS depends (to some extent) on the sound/signal being analysed. As we were primarily interested in musical audio, this sensitivity is explained by the variable nature of our analysis signals. We note that the algorithm's data dependancy was far less important for the speech signals tested.

## 6.3 Future Work

In the context of this work, one of the key aspects of greedy atomic decomposition is its flexible algorithm structure. That is, the model enhancements we introduced were primarily based on modification to the structure of basic MP/GTFS. We hope to introduce future enhancements based on similar modifications.

### 6.3.1 Non-Diagonal Estimation

Denoising strategies based on non-diagonal estimation estimate the signal's transform coefficients attenuation by considering information from multiple coefficients that are close together in time and frequency [57]. Non-diagonal estimators are often based on the calculation of an a priori SNR [57, 52], and have been shown to greatly outperform standard diagonal estimators in terms of musical noise attenuation. We hope to integrate non-diagonal estimators such as block thresholding and Ephraim and Malah's MMSE-LSA [58] into the GTFS framework, in order to further improve robustness against musical noise.

### 6.3.2 Molecular Approaches

Molecular Matching Pursuit is a greedy atomic decomposition approach that constructs a molecule of atoms to subtract from the residual signal at each iteration [38]. Molecules can be constructed based on closeness in time or frequency. The method can be considered as a *structured* sparsity approach, as it can be tailored to account for the persistence of atoms in time or frequency. This is similar to the principle on which Persistent Empirical Wiener denoising is based, and hence we are motivated to explore the algorithm's denoising potential.

# References

[1] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[2] S. V. Vaseghi, *Advanced digital signal processing and noise reduction.* Wiley, 2008.

[3] S. Godsill, P. Rayner, and O. Cappé, “Digital audio restoration,” in *Applications of Digital Signal Processing to Audio and Acoustics* (M. Kahrs and K. Brandenburg, eds.), Springer-Verlag, 1998.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[5] D. L. Donoho and J. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[6] M. Elad, “Why simple shrinkage is still relevant for redundant representations?,” *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5559–5569, 2006.

[7] L. Daudet, “A review on techniques for the extraction of transients in musical signals,” in *Computer Music Modeling and Retrieval*, pp. 219–232, Springer-Verlag, 2006.

[8] R. McAulay and T. Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 4, pp. 744–754, 1986.

[9] J. E. Markel and A. H. Gray, *Linear prediction of speech.* Springer-Verlag New York, Inc., 1976.

[10] Y. Ephraim and I. Cohen, “Recent advancements in speech enhancement,” *The Electrical Engineering Handbook*, pp. 12–26, 2006.

[11] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 2, pp. 443–445, 1985.

[12] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” *Acoustics, Speech and Signal Processing,*

*IEEE Transactions on*, vol. 32, no. 6, pp. 1109–1121, 1984.

[13] I. Cohen, “Speech enhancement using a noncausal a priori SNR estimator,” *Signal Processing Letters, IEEE*, vol. 11, no. 9, pp. 725–728, 2004.

[14] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[15] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, “Sparse representations in audio and music: from coding to source separation,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[16] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.

[17] P. Kabal, “An examination and interpretation of ITU-R BS. 1387: Perceptual evaluation of audio quality,” *TSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University*, pp. 1–89, 2002.

[18] K. Siedenburg, “Persistent empirical Wiener estimation with adaptive threshold selection for audio denoising,” in *Proceedings of the 9th Sound and Music Computing Conference, Copenhagen (July 11-14th)*, 2012.

[19] G. M. Davis, S. G. Mallat, and Z. Zhang, “Adaptive time-frequency decompositions,” *Optical Engineering*, vol. 33, no. 7, pp. 2183–2191, 1994.

[20] E. Ravelli, G. Richard, and L. Daudet, “Union of MDCT bases for audio coding,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 1361–1372, 2008.

[21] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, F. Lopez-Ferreras, and J. Curpian-Alonso, “New matching pursuit based sinusoidal modelling method for audio coding,” in *Vision, Image and Signal Processing, IEEE Proceedings*, vol. 151, pp. 21–28, IET, 2004.

[22] M. G. Christensen and S. H. Jensen, “The cyclic matching pursuit and its application to audio modeling and coding,” in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*, pp. 550–554, IEEE, 2007.

[23] R. Heusdens, R. Vafin, and W. B. Kleijn, “Sinusoidal modeling using psychoacoustic-adaptive matching pursuits,” *Signal Processing Letters, IEEE*, vol. 9, no. 8, pp. 262–265, 2002.

[24] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[25] E. J. Candès, “Compressive sampling,” in *Proceedings oh the International Congress of*

*Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pp. 1433–1452, 2006.

[26] D. L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[27] D. Gabor, “Theory of communication. part 1: The analysis of information,” *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, vol. 93, no. 26, pp. 429–441, 1946.

[28] D. Gabor, “Acoustical quanta and the theory of hearing,” *Nature*, vol. 159, no. 4044, pp. 591–594, 1947.

[29] S. Mallat, *A wavelet tour of signal processing: the sparse way.* 2008.

[30] M. Goodwin, “Matching pursuit with damped sinusoids,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 3, pp. 2037–2040, IEEE, 1997.

[31] R. Gribonval, “Fast matching pursuit with a multiscale dictionary of gaussian chirps,” *Signal Processing, IEEE Transactions on*, vol. 49, no. 5, pp. 994–1001, 2001.

[32] R. Gribonval and E. Bacry, “Harmonic decomposition of audio signals with matching pursuit,” *Signal Processing, IEEE Transactions on*, vol. 51, no. 1, pp. 101–111, 2003.

[33] E. C. Smith and M. S. Lewicki, “Efficient auditory coding,” *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.

[34] M. Moussallam, P. Leveau, and S. Aziz Sbai, “Sound enhancement using sparse approximation with speclets,” in *Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference on*, pp. 221–224, IEEE, 2010.

[35] E. B. George and M. J. Smith, “Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model,” *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 5, pp. 389–406, 1997.

[36] E. B. George and M. J. Smith, “Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones,” *Journal of the Audio Engineering Society*, vol. 40, no. 6, pp. 497–516, 1992.

[37] B. L. Sturm, J. J. Shynk, L. Daudet, and C. Roads, “Dark energy in sparse atomic estimations,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 671–676, 2008.

[38] L. Daudet, “Sparse and structured decompositions of signals with the molecular matching pursuit,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1808–1816, 2006.

[39] T. S. Verma and T. H. Meng, “Sinusoidal modeling using frame-based perceptually weighted matching pursuits,” in *Acoustics, Speech, and Signal Processing, 1999. Pro-*

*ceedings, 1999 IEEE International Conference on*, vol. 2, pp. 981–984, IEEE, 1999.

[40] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[41] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet shrinkage: asymptopia?," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 301–369, 1995.

[42] D. L. Donoho, "De-noising by soft-thresholding," *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, 1995.

[43] A. G. Bruce and H.-Y. Gao, "Understanding waveshrink: variance and bias estimation," *Biometrika*, vol. 83, no. 4, pp. 727–745, 1996.

[44] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[45] H.-Y. Gao, "Wavelet shrinkage denoising using the non-negative garrote," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 469–488, 1998.

[46] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 2, pp. 137–145, 1980.

[47] K. Siedenburg and M. Dörfler, "Audio denoising by generalized time-frequency thresholding," in *Audio Engineering Society Conference: 45th International Conference: Applications of Time-Frequency Processing in Audio*, Audio Engineering Society, 2012.

[48] S.-F. Lei and Y.-K. Tung, "Speech enhancement for nonstationary noises by wavelet packet transform and adaptive noise estimation," in *Intelligent Signal Processing and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium on*, pp. 41–44, IEEE, 2005.

[49] I. M. Johnstone and B. W. Silverman, "Wavelet threshold estimators for data with correlated noise," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 2, pp. 319–351, 1997.

[50] O. Cappé, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 345–349, 1994.

[51] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[52] T. Cai, "Adaptive wavelet estimation: a block thresholding and oracle inequality approach," *The Annals of Statistics*, vol. 27, no. 3, pp. 898–924, 1999.

[53] G. Yu, S. Mallat, and E. Bacry, “Audio denoising by time-frequency block thresholding,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1830–1839, 2008.

[54] L. Daudet, “Sparse and structured decompositions of signals with the molecular matching pursuit,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1808–1816, 2006.

[55] M. Kowalski and B. Torrésani, “Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients,” *Signal, image and video processing*, vol. 3, no. 3, pp. 251–264, 2009.

[56] R. McAulay and T. Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 4, pp. 744–754, 1986.

[57] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 6, pp. 1109–1121, 1984.

[58] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 2, pp. 443–445, 1985.