IDEA League

Master of Science in Applied Geophysics

Research Thesis

A seismic-inspired denoising method for online recorded music

Floris Bastiaan van den Broek

August 11th, 2017

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A seismic-inspired denoising method for online recorded music

Master of Science Thesis

for the degree in Master of Science in Applied Geophysics at

Delft University of Technology
Swiss Federal Institute of Technology Zürich
RWTH Aachen
by

Floris Bastiaan van den Broek

August 11th, 2017

Department of Geosciences and Engineering — TU Delft
Department of Earth Sciences — ETH Zürich
Faculty of Georesources and Material Engineering — RWTH Aachen

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

Delft University of Technology, The Netherlands
Swiss Federal Institute of Technology Zürich, Switzerland
RWTH Aachen, Germany

Dated: *August 11th, 2017*

Supervisor(s):

Prof. Dr. Johan O.A. Robertsson - ETH Zürich

Dr. Dirk-Jan van Manen - ETH Zürich

Björn Melinder, CTO - Soundtrap AB

Committee members:

Prof. Dr. Johan O.A. Robertsson - ETH Zürich

Dr. Dirk-Jan van Manen - ETH Zürich

Björn Melinder, CTO - Soundtrap AB

Prof. Dr. Ir. Kees P.A. Wapenaar - TU Delft

Abstract

In this work, a novel subspace-based algorithm is presented for automated random noise reduction in online recorded music. Musical signal enhancement is a separate issue from the well-studied speech enhancement problem due to the particularly wide range of signal characteristics encountered, and thus requires a very general approach. Because similar issues drive denoising advances in seismic signal processing, it is argued that an algorithm can be developed through a cross-disciplinary approach. Inspired by an enhancement method for seismic sections, noise reduction is achieved by applying a singular value decomposition-based image enhancement technique, known as eigenimage filtering, to the time-frequency representation of the musical signal. Classic eigenimage filtering approximates a full-rank matrix by its closest rank-deficient approximation; the preserved and discarded parts of the matrix correspond to the signal and noise subspaces, respectively. Under the assumption of a quasi-stationary signal, this technique is applied to the short-time Fourier transform of the signal. However, because the standard eigenimage filtering approach results in unwanted residual noise characteristics when applied in this domain, an adapted version of the technique is used. In this adaptation, all singular values are altered but none are set to zero, and the alteration is dependent on the singular values encountered. Therefore, the method is data-adaptive. Subjective and objective performance measures indicate that the method is capable of improving the quality of noisy recordings, and that its quality is competitive compared with an open-source noise reduction algorithm whilst having the advantages of automation and fewer user-defined parameters.

Acknowledgements

First and foremost, I would like to thank my committee of supervisors, without whom this work would not exist. This includes my academic supervisors, Johan Robertsson and Dirk-Jan van Manen, whose ideas, suggestions and critical reviews provided the direction this thesis needed, but who gave me all the necessary freedom to make this work my own; and my company supervisor, Björn Melinder, who not only provided me with the opportunity to do this fantastic project, but who also involved me in all other aspects of the work in the wonderful world of Soundtrap. It has been a wonderful experience, and one that I will never forget.
I am also grateful to Frank Scherbaum, who selflessly offered to lend me his own microphones, and who flew from Germany to Sweden on multiple occasions to join our meetings and give his opinion, fueled by nothing but academic interest.

Furthermore, this thesis would have been impossible without the logistical help provided by a number of people. Specifically, my gratitude goes out to Joakim Persson, who let me use his studio, equipment, instruments, voice, as well as multiple hours of his own time, in order to record the reference audio for the algorithm to be tested on. My sincere thanks also goes out to all colleagues, friends and family who (semi-)voluntarily endured and rated twenty-five minutes of noise, and in doing so helped the results of this work take shape.

Finally, I want to thank my friends and family, who have supported and visited me this semester as they have for years, regardless of distance; and all my colleagues at Soundtrap, who helped me settle, showed me around, let me in on some of the peculiar traditions of the Swedes, and in general, took me in as one of their own. You have been great, and my time here would not have been the same without you.

Floris Bastiaan van den Broek

August 11$^{\text{th}}$, 2017

TABLE OF CONTENTS

List of Figures

List of Tables

Acronyms

	Acronym	Meaning
General signal processing		
	AWGN	Additive white Gaussian noise
	COVA	Constant overlap add
	DFT	Discrete Fourier transform
	FFT	Fast Fourier transform
	GSNR	Global signal-to-noise ratio
	ISTFT	Inverse short-time Fourier transform
	OVA	Overlap add
	PSD	Power spectral density
	SNR	Signal-to-noise ratio
	SSNR	Segmental signal-to-noise ratio
	STFT	Short-time Fourier transform
	TFR	Time-frequency representation
	WF	Wiener filter
	WOVA	Weighted overlap add
Audio-specific signal processing		
	AEF	Adapted eigenimage filtering
	ITU	International Telecommunication Union
	MIR	Music information retrieval
	NSA	Noise suppression algorithm
	SNG	Spectral noise gating
	SS	Spectral subtraction
	STSA	Short-time spectral amplitude
	VAD	Voice activity detector
Mathematical concepts		
	EVD	Eigenvalue decomposition
	KLT	Karhunen-Loèwe transform
	MMSE	Minimum mean square error
	NMF,NNMF	Non-negative matrix factorisation
	SVD	Singular value decomposition
Miscellaneous		
	AC	Alternating current
	MOS	Mean opinion score

Chapter 1

Introduction

Music is omnipresent in our modern society. With an ever-growing number of both consumers and producers, there is an increasing demand for innovations that enhance the musical experience. In particular, this requires improvements to the stage during which audio is recorded, the stage in which it is played, or the processing stage in between. However, a large number of aspiring musicians do not have the option to make improvements in the former two stages, as this requires upgrading to expensive higher-quality hardware or recording in a professional studio. One of the main purposes of the processing stage for this group of users is thus to compensate for the adverse effects and limitations of their suboptimal equipment. A straightforward example of such a processing solution is audio equalisation, in which individual frequency bins can be either attenuated or enhanced, for instance to compensate for the often biased frequency response of the microphone and speakers used. A much more complex issue, however, is that of general background noise. In the absence of a soundproofed recording studio and professional recording hardware, a variety of unwanted sounds may appear in recordings. This is particularly the case for those recordings made using accessible and easy-to-use music-making applications. These platforms are specifically designed for use at home, in classrooms or even outside, and are intended to be used with mobile phones, tablets, or laptops, thus mostly relying on unsophisticated built-in microphones. Removing or reducing the noise from this category of audio therefore poses a particular challenge. It is however also an appealing challenge, both from a theoretical and a practical point of view, and will constitute the main goal of this thesis.

Over the past decades, many methods for removing noise from audio have been developed. Although partially successful, many of these methods either leave too much residual noise, introduce new (musical) noise (Inoue et al., 2011; Malca & Wulich, 1996) or distort the signal considerably (J. Chen et al., 2006). Moreover, techniques like spectral noise gating are prone to personal error, because they require the user to carefully select a noise-only interval upon which to base the noise removal. Hence, new methods are required that remove more noise without audible distortion of the signal, and that are generally applicable, preferably with as little parameterisation and user intervention as possible.

The noise reduction problem is not unique to audio, but shared in most disciplines that involve signal processing. One approach for the development of denoising methods is thus to adapt and apply processing algorithms used in other fields of research. Examples of such interdisciplinary works include the application of the (originally geophysical) Stockwell transform to enhance the signal in electrocardiograms (Huang et al., 2009), the usage of a Voice Activity Detector (VAD)

to determine the presence or absence of wind noise in volcanic tremors (Cabras et al., 2014), and the usage of Music Information Retrieval (MIR) methods to define and extract attributes from seismic data (Amendola et al., 2017).

Out of all disciplines that include signal enhancement stages, the combination of audio and seismics seems particularly well-suited for a cross-disciplinary approach for a number of reasons. First, both disciplines deal with the physics of sound wave propagation. Naturally, of main interest in music is the direct wave from sound source to microphone, whereas the reflected (and sometimes refracted) waves generally constitute the desired signal in seismic exploration. Nevertheless, the methods used to separate desired from undesired waves may be cross-field compatible. Second, seismic data volumes are generally large, typically including thousands to tens of thousands of recordings. It is thus impractical to find the optimal processing solution for each individual data trace by hand. Instead, the objective is to find a general and efficient way to enhance the signal in a large number of channels, without having to intervene and change parameters. Similarly, the data volumes in online music recording applications quickly become too large to allow for manual noise removal, such that automated methods are required.

In this thesis I present an interdisciplinary approach, in which seismic noise-reduction algorithms serve as a source of inspiration for musical signal enhancement methods. In order to enable a proper assessment of the possibilities and limitations of such an approach, it is paramount to have an overview of the signal and noise types encountered in both fields, as well as conventional processing algorithms. Hence, the following sections of this chapter will provide a general overview of common denoising practices in audio (Sec. 1.1) and seismic data processing (Sec. 1.2), which provides the framework to define the scope and outline of this work (Sec. 1.3).

1.1 Denoising techniques in audio signal processing

Noise reduction in audio is a very active field of research, with successful algorithms finding use in speech processing (Benesty et al., 2011; Parchami et al., 2016), automatic speech recognition (Hermus et al., 2007), hands-free communication (Preuss, 1979; Boll, 1979), hearing implants (Yousefian et al., 2014), and to a lesser degree, musical applications (Bassiou et al., 2014). Besides sorting methods by their field of application, noise suppression algorithms (NSA) can be classified according to the type of noise they are designed to reduce. The most extensively studied problem is that of speech corrupted by additive and uncorrelated white noise; that is, the following equations are assumed to hold:

$$x(n) = s(n) + v(n) \tag{1.1a}$$

$$X(k,\tau) = S(k,\tau) + V(k,\tau) \tag{1.1b}$$

where $x(n)$, $s(n)$ and $v(n)$ correspond to the noisy signal, clean signal and noise sampled at discrete time n, the capitals X, S and V denote their respective discrete Fourier transforms, and k and τ represent the frequency bin and time frame indices of a time-frequency representation. Though other options exist, audio signal processing conventionally takes place in the time-frequency domain, as it is particularly well suited for representing and analysing non-stationary signals. Note that (1.1b) only holds if the transform that connects it with (1.1a) is linear, which in audio is generally the case.

The denoising techniques associated with this particular additive noise model are commonly categorised into four subclasses: spectral subtractive, Wiener, statistical estimation (or Bayesian), and subspace methods. For extensive literature reviews and discussions of each of these classes, the reader is referred to Upadhyay & Karmakar (2015), J. Chen et al. (2006), Ephraim (1992) and Hermus et al. (2007), respectively. A concise overview is provided here for the purpose of reference.

1.1.1 Spectral subtraction (SS)

The spectral subtraction class, which still sees extensive use today, was introduced by Boll (1979) for the purposes of speech signal enhancement, and operates in the time-frequency domain obtained by the short-time Fourier transform (STFT). A time interval, assumed to be devoid of signal, is used to obtain an estimate of the amplitude spectrum of the noise. Under the assumption of locally stationary, uncorrelated and additive noise, this estimate is subsequently subtracted from the noisy audio to yield the desired 'clean' signal amplitude spectrum:

$$|\hat{S}(k,\tau)| = |X(k,\tau)| - |\hat{V}(k,\tau)| \tag{1.2}$$

or, in case the noise is estimated in terms of its power spectral density (PSD):

$$|\hat{S}(k,\tau)|^2 = |X(k,\tau)|^2 - |\hat{V}(k,\tau)|^2 \tag{1.3}$$

where the hat symbol is used to denote an estimate. The cross-terms $X(k,\tau)\hat{V}^*(k,\tau)$ and $X^*(k,\tau)\hat{V}(k,\tau)$ (* denoting conjugate transpose) are omitted in (1.3) because the noise is assumed zero-mean and uncorrelated with the signal. If a pause in the signal is detected (by using a voice activity detector), it is used to update the noise amplitude spectrum estimate accordingly. It is common practice to subsequently combine the processed spectral amplitudes with the original (noisy) phase spectrum, which was proven by Ephraim & Malah (1984) to be the optimal phase estimate when the STFT coefficients are mutually uncorrelated. Hence, the processed STFT coefficients are conventionally obtained as follows:

$$\hat{S}(k,\tau) = |\hat{S}(k,\tau)|e^{j\varphi_x(k,\tau)} \tag{1.4}$$

where j is the imaginary unit and $\varphi_x(k,\tau)$ is the phase spectrum of the original noisy speech. The last step is to retrieve the estimate of the clean signal $\hat{s}(n)$ by applying the inverse short-time Fourier transform (ISTFT).

Though effective in its aim of reducing background noise, the residual noise was found to possess undesirable characteristics (Preuss, 1979; Berouti et al., 1979). Because an average noise amplitude spectrum is subtracted from the stochastic noise process, the method is prone to noise residuals that are confined in both frequency and time. Converted back to the auditory domain, these are experienced as short, random-pitch sounds, leading to the phenomenon's designation as *tonal* or *musical* noise. Considered by many to be more intrusive than the original noise, subsequent research efforts where dedicated to alleviating this problem, which led to improved versions with different noise spectrum estimators (Preuss, 1979), over-subtracting based on the noise amplitudes (Berouti et al., 1979; Lorber & Hoeldrich, 1997) and subsequent processing steps designed to deal with residual musical noise (Haulick et al., 1997; Malca & Wulich, 1996).

1.1.2 Wiener Filter (WF) methods

The Wiener methods also filter the signal under the assumption of uncorrelated and additive Gaussian noise, and are in fact related to spectral subtraction. The aim is to improve the output signal-to-noise ratio (SNR) under the constraint of minimum mean square error (MMSE) between the desired (clean) and estimated signal. This criterion can be applied to the time series in time domain Wiener filtering (Benesty & Chen, 2011) or, more commonly, to the signal's amplitude spectrum in the STFT domain (Benesty et al., 2011; Parchami et al., 2016, and references therein). For the latter, the Wiener filters take the shape of a two-dimensional gain function $W(k,\tau)$:

$$\hat{S}(k,\tau) = W(k,\tau)X(k,\tau) \tag{1.5}$$

and, defining the error matrix as

$$\epsilon(k,\tau) = \hat{S}(k,\tau) - S(k,\tau) = W(k,\tau)X(k,\tau) - S(k,\tau) \tag{1.6}$$

the problem becomes to find the Wiener gain that minimises the frequency sub-band MSE criterion

$$E_\tau[|\epsilon(k,\tau)|^2] \tag{1.7}$$

where $E_\tau[.]$ denotes the mathematical expectation operator over the index τ.
It can be derived (Benesty et al., 2011) that the coefficients of the Wiener gain are given by:

$$W(k,\tau) = \frac{(\sigma_s^2/\sigma_v^2)}{1+(\sigma_s^2/\sigma_v^2)} = \frac{iSNR(k,\tau)}{1+iSNR(k,\tau)} \tag{1.8}$$

where $iSNR(k,\tau)$ is the local input signal to noise ratio, defined as the ratio of the signal and noise variances. As can easily be deduced from (1.8), the gain end-members for the cases of infinitely high and low $iSNR$ are 1 and 0, respectively. Clearly, the success of a Wiener method relies on the accuracy of the estimate of the local signal to noise ratio, which is a whole branch of research in itself. For the purposes of audio signal enhancement, examples of valuable contributions are cepstro-temporal SNR estimation (Breithaupt et al., 2008), two-step STFT-domain SNR estimation (Plapous et al., 2006) and STFT-domain SNR estimation conditioned on all previous frames (Cohen, 2005).
The Wiener gain includes an implicit trade-off between noise reduction and signal distortion. This can be derived by rewriting equations (1.5) and (1.6) using (1.1b). For the former, this leads to:

$$\begin{aligned}\hat{S}(k,\tau) &= W(k,\tau)X(k,\tau)\\ &= W(k,\tau)S(k,\tau) + W(k,\tau)V(k,\tau)\\ &= S_f(k,\tau) + V_r(k,\tau)\end{aligned} \tag{1.9}$$

which indicates that the result of filtering is a superposition of the filtered signal S_f and the residual noise V_r. The formulation of the error matrix can be reformulated in similar fashion:

$$\begin{aligned}\epsilon(k,\tau) &= W(k,\tau)X(k,\tau) - S(k,\tau)\\ &= W(k,\tau)\,[\,S(k,\tau) + V(k,\tau)\,] - S(k,\tau) &\text{(1.10a)}\\ \epsilon(k,\tau) &= S(k,\tau)\,[\,W(k,\tau) - 1\,] + W(k,\tau)V(k,\tau) &\text{(1.10b)}\end{aligned}$$

and, using the definitions of S_f and V_r as in (1.9):

$$\begin{aligned}\epsilon(k,\tau) &= [\, S_f(k,\tau) - S(k,\tau)\,] + V_r(k,\tau) \\ &= S_d(k,\tau) + V_r(k,\tau)\end{aligned} \tag{1.11}$$

The difference between the filtered and true signal, denoted S_d, is indicative of the signal distortion due to the filtering. Equation (1.11) illustrates that the error matrix always consists of a signal distortion and a residual noise component. It is insightful to consider the effect of the aforementioned gain end-members of 1 and 0. In the case of infinitely high $iSNR$, the Wiener gain is equal to one, and (1.10b) reduces to $\epsilon(k,\tau) = V(k,\tau)$. In this case, there is no signal distortion, but the noise is equally unaltered. Conversely, if $iSNR$ is infinitely low, the gain is equal to zero, and (1.10b) reduces to $\epsilon(k,\tau) = -S(k,\tau)$. Now, there is no residual noise, but the signal is fully distorted.

After estimating the a priori SNR, there are different approaches in defining the Wiener gain itself. Aside from the standard formulation in (1.8), common varieties include square-root Wiener filtering (Inoue et al., 2011), which is in fact equal to PSD domain spectral subtraction, and parametric varieties (Inoue et al., 2011; Fan, 2004) that allow for adjustments in the trade-off between noise reduction and signal distortion. In general, the greatest advantage of the Wiener methods is the simplicity of implementation, whereas the major disadvantage is the inherent trade-off between signal distortion and residual noise.

1.1.3 Statistical estimator methods

The third branch of denoising methods has its roots in probability theory. Research in this area gained momentum after Ephraim & Malah (1984) argued that although both the spectral subtractive and the Wiener methods are implemented in the STFT domain, neither method actually attempts to estimate the individual STFT coefficients in an optimal sense. This observation initiated the class of methods known as Bayesian short-time spectral amplitude (STSA) estimators. As the name implies, these methods focus on altering the perceptually dominant amplitude values, and like most of the classical spectral subtractive and Wiener methods, they are usually combined with the noisy phase spectrum afterwards as in (1.4).
The required inputs of an algorithm in this class comprise the input SNR, for which estimation methods were discussed in section 1.1.2, and the probability distributions of the Fourier coefficients of both the noise and the clean signal. These coefficients cannot be accurately measured from the signal because music and speech (and possibly the noise) are non-stationary processes. Hence, a specific statistical model is assumed for the Fourier coefficients, and the corresponding probability distribution is used. The most common choice is to assume that the Fourier coefficients are statistically independent, mutually uncorrelated, and Gaussian distributed. When the Fourier real and imaginary parts are both Gaussian distributed, the corresponding amplitudes are Rayleigh distributed. As this corresponds to a negligible probability at low amplitudes, which can lead to reduced noise suppression in the absence of signal, an additional signal presence probability factor complements the algorithm.
The actual implementation then involves minimising some Bayesian cost function of distance between the estimated and clean signal STSA values. The original algorithm by Ephraim & Malah (1984) uses the standard MMSE criterion

$$E\{\ (\,|S| - |\hat{S}|\,)^2\ \} = \mathcal{C}_{MMSE}\{\ |S|, |\hat{S}|\ \} \tag{1.12}$$

where $\mathcal{C}_{MMSE}\{.,.\}$ denotes the MMSE cost function between its arguments. Though this was found to yield better results in comparison with the standard spectral subtraction and Wiener filtering methods, it was already known at that time that the method could be further improved by using a different cost function based on the human auditory system. Specifically, because the relation between amplitude and perceived loudness is more logarithmic than linear, it is perceptually more meaningful to compare the log-amplitude spectra. This lead to the so-called MMSE log-STSA estimator (Ephraim & Malah, 1985):

$$E\{\ (\ log|S| - log|\hat{S}|\ )^2\ \} = \mathcal{C}_{Log-MMSE}\{\ |S|, |\hat{S}|\ \} \tag{1.13}$$

which was found to yield better results in comparison with the standard implementation. Further research efforts led to an even more fine-tuned weighted power law cost function (Plourde & Champagne, 2008):

$$E\left\{\ \left(\ \frac{|S|^\beta - |\hat{S}|^\beta}{|S|^\alpha}\ \right)^2\ \right\} = \mathcal{C}_{W\beta-SA}\{\ |S|, |\hat{S}|\ \} \tag{1.14}$$

that takes into account both the non-linear amplitude-loudness relation (denoted by β) and the frequency-sensitive masking properties of the human ear (denoted by α).
The greatest strength of the Bayesian class is a good overall performance, and particularly the relative absence of musical noise after application. The larger amount of required a-priori knowledge and the slightly increased computational load constitute the largest drawbacks.

1.1.4 Subspace methods

The last denoising class for suppressing uncorrelated white noise comprises the so-called subspace methods. Subspace methods have been applied in a plethora of fields for the purposes of denoising (Hu & Loizou, 2003; Jones & Levy, 1987), data compression and pattern recognition. An impractical consequence is that there exist some differences in definitions and nomenclature between fields. This work will follow the definitions as in Ulrych & Sacchi (2005). The generally adopted approach in subspace methods is to decompose the noisy signal in a number of weighted and mutually orthonormal constituents as

$$x = K\,y \tag{1.15a}$$

$$x(k) = \sum_{i=1}^{M} K_{ki}\,y_i \tag{1.15b}$$

where $x = [\ x(1)\ x(2)\ ..\ x(N)\ ]^T$ is an N-point noisy audio vector, K is a $(N \times M)$, $N > M$ matrix with the M constituent vectors k_i ordered as columns, and $y = [\ y(1)\ y(2)\ ..\ y(M)\ ]^T$ is a M-point vector containing the weights. In this context, the linearly independent constituents are also commonly referred to as the basis functions. Making use of the orthonormal nature of the basis functions, the inverse of (1.15a) can be written as

$$y = K^T\,x \tag{1.16}$$

This process is known as the principal component or Karhunen-Loèwe transformation (KLT) of the vector x, and y is known as the principal component projection of x. The transformation matrix K is obtained by means of eigenvalue decomposition of the covariance matrix of the

signal x:

$$R_x = K\,\Lambda\,K^* \tag{1.17}$$

with $R_x = E[x\,x^*]$ denoting the noisy signal covariance matrix, Λ a diagonal matrix containing the corresponding eigenvalues λ_i ordered by decreasing magnitude, and K a matrix with the eigenvectors k_i as its columns, ordered accordingly.

The approach adopted in subspace methods is now to assume that the signal component s of the noisy signal vector x can be reconstructed by using only the first p basis functions and weights, whereas to completely reconstruct the noise, all basis functions are required. Put differently: R_x is assumed to be of full rank M, whereas the clean signal covariance matrix $R_s = E[s\,s^*]$ is assumed to be of rank p, with $p < M$. If we assume, as for the other methods in this section, that the noise is Gaussian, white, and uncorrelated with the signal, R_x can be written as

$$R_x = R_s + R_v = R_s + \sigma_v^2\,I \tag{1.18}$$

and the components of Λ in (1.17) can be written as

$$\lambda_i = \begin{cases} \lambda_i^s + \sigma_v^2, & 1 \le i \le p \\ \sigma_v^2, & p < i \le M. \end{cases} \tag{1.19}$$

where λ^s denote the clean speech eigenvalues, i.e. those resulting from the decomposition of the clean speech covariance matrix $R_s = K\,\Lambda^s\,K^*$. As (1.19) illustrates, noise occupies the whole M-dimensional space, whereas the signal resides in the p-dimensional subspace. Noise removal can thus be achieved by nulling the (M-p)-dimensional *noise subspace*, and optionally, by reducing its contribution in the *signal subspace* (or, more accurately, the *signal-plus-noise subspace*). The former procedure can be achieved in terms of the Karhunen-Loèwe transform by using only those eigenvectors k_i that correspond with the signal subspace in (1.15b):

$$\hat{s}(k) = \sum_{i=1}^{p} K_{ki}\,y_i \tag{1.20}$$

The removed signal, or nulled subspace, is given by

$$x(k) - \hat{s}(k) = \sum_{i=p+1}^{M} K_{ki}\,y_i \tag{1.21}$$

which, when p is chosen correctly, does not contain any signal components.

One of the earliest contributions in this field was provided by Dendrinos et al. (1991), who showed how the singular value decomposition (SVD), a factorisation technique closely related to the KLT, can be used to effectively remove white noise. Shortly after, Ephraim & Trees (1995) demonstrated the applicability of the KLT to the additive white noise problem, as well as the method's superiority over conventional methods at that time. Later, it was found that the subspace approach can be extended to the case of coloured noise through use of implicit (Hermus et al., 2007) or explicit (Hermus et al., 2007; Hu & Loizou, 2003) noise pre-whitening. Particularly taking the last point into account, one of the main strengths of the subspace class is its general applicability and lack of required a-priori knowledge. The greatest shortcoming is the higher computational complexity, which arises from the need to perform a singular value or eigenvalue decomposition.

1.2 Denoising techniques in seismic signal processing

There are many different types of seismic surveys, each with a specific purpose. Examples include 1) the check-shot survey, designed to correct sonic logs and calibrate synthetic seismograms; 2) vertical seismic profile methods, used (amongst other reasons) to get high-resolution images in vicinity of a borehole, for subsequent correlation with surface seismic data; 3) refraction surveys, whose main purpose is to create images of subsurface seismic velocities; 4) reflection surveys, devised to locate subsurface reflectors associated with lithological boundaries or changes in pore fluid content; and 5) monitoring surveys, whose purpose is to detect tremors and (induced or natural) earthquakes. The corresponding processing sequences depend on the type of survey, the target of investigation and the acquisition characteristics.
In conventional seismic data processing, the majority of the energy that is removed from the records is not of random, but of coherent nature. For instance, in a reflection survey, the data recorded at an individual measurement station will generally consist of a superposition of the desired primary reflection, the direct arrival, refracted waves, multiple reflections, as well as random noise. Moreover, ground roll commonly poses a problem in land surveys, whereas marine surveys cope with additional complexities originating from ocean wave swells, cable towing and source/receiver ghosts. In order to keep a review of seismic processing standards tractable, this section considers only those methods that do not divert far from the low-dimensional field of audio. In particular, this section will focus on single-channel noise removal and the enhancement of stacked sections.

1.2.1 Single-channel signal enhancement

The main objective in the pre-stack processing phase is to remove any noise that may sum constructively in the subsequent stacking stage. Particularly, the coherent noise from the various unwanted wave types needs to be reduced. This issue can be addressed in multiple ways.
The conventional approach is to separate these wave types by exploiting differences in their velocity or move-out behaviour. Aside from the standard t-x plane, interfering wave types are commonly addressed by filtering in a transformed domain, in which they are more clearly separated from the signal of interest. Examples of signal enhancement methods in such filtering dimensions include projection filtering (K. Chen & Sacchi, 2017) and eigenimage noise suppression (Trickett, 2002, 2003) in the f-x (or f-xy) domain; multiple reflection removal (Zhou & Greenhalgh, 1994b) in the f-k domain; automated thresholding for large-amplitude noise (Elboth et al., 2009) in the t-f domain; and wave separation and filtering (Zhou & Greenhalgh, 1994a) in the τ-p domain. Furthermore, the use of multiple-component receivers opens up another class of signal enhancement techniques, as they allow for processing based on polarisation and the direction of wave propagation. Though this has multiple uses, it is particularly valuable in surface wave related problems such as building a near-surface velocity model (Socco et al., 2010) and the identification and removal of (potentially scattered) surface waves (Kragh & Peardon, 1995).

The exploitation of spatial and/or directional information offers limited opportunity to be extended to online recorded audio, which is generally recorded using a single microphone. However, some seismic signal enhancement procedures do operate on a single trace basis. An example of such a technique is non-negative matrix factorisation (NMF or NNMF), which can be performed on the TFR of an individual recording (as amplitude spectra are by definition non-negative).

The concept is that a non-negative matrix can be factorised as follows:

$$|X(k,\tau)|^{2\beta} = \bar{X} = DH + E \tag{1.22}$$

where $X(k,\tau)$ is a time-frequency representation of size $(N \times M)$, the amplitude of which is raised to the exponential 2β to enhance NMF performace. The variable β usually takes the value of $1/3$ in seismic literature (e.g. Cabras et al. (2014); Vaezi & Kazemi (2016)). D is the features or dictionary matrix of size $(N \times k)$, H is the weighting or code matrix of size $(k \times M)$, and E is the approximation error of the same size as X. Both D and H are strictly positive matrices. The factor k is smaller than M, and dictates how many features are extracted.
If an additive model is assumed for the various constituents of the signal, (1.22) can be reformulated:

$$\bar{X} = X_1 + X_2 = \begin{bmatrix} D_1 & D_2 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} + E \tag{1.23}$$

The theory is that the matrices X_1 and X_2 relate to the amplitude spectra of the distinct constituents, provided that the approximation error E is insignificant. A Wiener filter can then be constructed, depending on which of the constituents in desired. If X_1 represents the signal to be extracted, the corresponding time-domain signal $x_1(t)$ is given by:

$$x_1(t) \approx iSTFT\{\, G(k,\tau)\, X(k,\tau) \,\} \tag{1.24}$$

with

$$G(k,\tau) = \left(\frac{X_1}{X_1 + X_2} \right)^{\frac{1}{2\beta}} \tag{1.25}$$

and similarly for X_2. This procedure has found application in the field of seismology, where volcanic tremors were separated from wind noise (Cabras et al., 2014), as well as from one another (Cabras et al., 2012) using only a single-sensor recording. More recently, Vaezi & Kazemi (2016) found the technique successful in removing swell noise from marine seismic data, and suggest to extend its use to the reduction of ground roll in the future.

A second single-channel approach is to apply some form of thresholding to the coefficients of a time-frequency representation. The principle of thresholding methods is to specify a number of ranges, which are bounded by threshold values. An alteration rule is defined for each range; depending on in which range a coefficient falls, it is altered according to said rule. An example of such a procedure is the customised thresholding rule provided by Yoon & Vaidyanathan (2004):

$$fc(x) = \begin{cases} x - sgn(x)(1-\alpha)\lambda, & \text{if } |x| \geq \lambda \\ 0, & \text{if } |x| \leq \gamma \\ \alpha\lambda \left(\frac{|x|-\gamma}{\lambda-\gamma}\right)^2 \left\{ (\alpha-3)\left(\frac{|x|-\gamma}{\lambda-\gamma}\right) + 4 - \alpha \right\}, & \text{otherwise.} \end{cases} \tag{1.26}$$

Parolai (2009) applied the customised thresholding rule in (1.26) to the real and imaginary coefficients of a seismic signal's S-transform, which is a TFR with frequency-dependent resolution (Stockwell et al., 1996). By relating the value of λ in (1.26) to the measured noise variance, and by empirically determining suitable values for γ and α, Parolai (2009) reports a clear isolation of dispersive waves, even at low signal to noise ratios.

A further noise reduction method is median filtering, which is a nonlinear smoothing operation. It has been applied extensively in image processing to remove impulse noise (e.g. Chan et al.

(2005)) whilst preserving edges, and as will be discussed in section 1.2.2, these characteristics make it suitable for enhancing stacked seismic sections. However, the technique can also be performed on 1-dimensional data, such as a time series. In its simplest form, a median filter replaces each sample of a trace by the median value of samples in its vicinity. Like a running average, a running median tends to smooth the data, and for Gaussian distributed data, the results are equal in statistical sense (Bednar, 1983). The difference is that sharp discontinuities are much better preserved using a median filter. These characteristics make it well-suited for removing spiking noise from acoustic impedance logs (Bednar, 1983) and seismic traces (Liu et al., 2009).
The major influencing factor in median filters is the filter length, since the degree of smoothing increases as more samples are considered. For this reason, Liu et al. (2009) propose a two-step algorithm for removing random, spike-like noise. In the first stage, the filter length is determined by comparing the local median value to the global average median value. When the local value exceeds this threshold, the filter length is set to a smaller value; otherwise, the length is increased. In the second step, the signal is filtered using the updated filter lengths. Thus, regions assumed to consist mostly of noise are smoothed more heavily than portions with a stronger signal content. Later, Y. Chen (2015) adopted the same data-adaptive median filtering approach, but defined the filter length based on the degree of similarity between the original and median-filtered signal (a measure denoted *signal reliability*). This leads to smoother filter length variations and more stable results.

1.2.2 Enhancement of seismic sections

The majority of the unwanted signal is removed prior to and during stacking in reflection seismic processing. Nevertheless, there exist some additional noise suppression algorithms that are designed to operate on seismic sections. These methods are potentially applicable for audio denoising purposes, because the previously mentioned time-frequency transforms allow us to create a 2-D image of the recording of interest.

In the previous section, methods of median filtering a single trace were discussed. However, as previously mentioned, the expansion to a higher-order dimension is straightforward. The edge-preserving characteristic is particularly useful for application in seismic sections, as discontinuities can correspond to important geological features such as faults. To smooth out the noise whilst preserving such discontinuities, Aqrawi et al. (2013) opt to employ either a weighted mean or a median filter, depending on the local coherency of the seismic response. They use a filter length that increases with depth, as well as a dip estimator to guide the direction of filtering. An alternative approach is to alter the filter length based on an estimate of the local noise level. Al-Dossary (2014) used this approach in a seismic volume to adapt the size of a 3-D median filter to the data automatically, which filter sizes ranging from (3x3x3) for low-noise portions to (7x7x7) for high-noise sub volumes.

A second approach to enhance seismic sections is eigenimage filtering. This is a subspace technique that relies on the decomposition of a data matrix X using singular value decomposition (SVD) as follows:

$$X = U\Sigma V^* \tag{1.27}$$

where $X(N \times M)$ is the seismic section in matrix format, with the M traces x_j ordered as its columns; $U(N \times N)$ and $V(M \times M)$ are square matrices with the eigenvectors u_i of XX^* and v_i of X^*X ordered as their columns, respectively; and $\Sigma(N \times M)$ is a diagonal matrix containing

the so-called *singular values*, sorted by decreasing magnitude. Vice versa, the synthesis of the data matrix X from these matrices can be viewed as a summation rank-one constituent matrices, weighted by their singular value σ_i:

$$X = \sum_{i=1}^{min(N,M)} \sigma_i \, u_i \, v_i^* \tag{1.28}$$

The result of the outer dot product $u_i \odot v_i^*$ is referred to as the i^{th} *eigenimage* of X (Ulrych & Sacchi, 2005, and references therein). Each eigenimage is a rank one matrix of the same size as X. Much like the previously discussed Karhunen-Loèwe transform (Sec. 1.1.4), the reconstruction can be truncated after p eigenimages to approximate X. Signal with pronounced trace-to-trace coherency will tend to reside in the dominant first few eigenimages (for an elaborate discussion, see section 2.4), whereas incoherent noise will spread the entire domain. Noise can be reduced by removing the contribution from the last eigenimages in the reconstruction of X. The amount of energy that is removed in this process depends on the relative magnitude of the singular values and the value of p, and can be expressed as a ratio of the squared singular values (Ulrych & Sacchi, 2005):

$$\epsilon = \frac{\sum_{i=p+1}^{min(M,N)} \sigma_i^2}{\sum_i \sigma_i^2} \tag{1.29}$$

When $\epsilon = 0$, the original and processed sections are equal; when $\epsilon = 1$, all energy has been removed.

As the method is biased toward coherent events, eigenimage filtering in the t-x domain is most suitable for regions where dipping events are largely absent. If dipping layers are known to be present, the best practice is to look at the removed section (i.e. the difference between the original and reconstructed data), and choose the lowest value of p for which no information is visible in the removed section. Alternatively, the technique can be applied in the f-x (Trickett, 2002) or f-xy (Trickett, 2003) domain.

1.3 Scope and aim of the thesis

With the framework for potential denoising techniques in mind, it is essential to now carefully formulate the problem that needs solving. In the broadest sense, the aim is to develop an algorithm capable of removing noise from musical recordings. These recordings are made using the online studio application developed by Soundtrap, which at the time of writing has 1.5 million users. To determine what constitutes the most common quality issue in user projects, a preliminary study was carried out among 400 randomly selected recordings. These recordings were classified by the main quality issue in them, as summarised in table 1.1. Random noise (hiss), leakage of background music, and coherent noise (mostly AC hum) were found to be both very frequent as well as intrusive. Compared to the two noise issues, the problem of background music leakage can be addressed relatively easily by suggesting users to check their volume and to lower it if necessary. To constrain the scope, this work thus focuses on the most frequently occurring noise issue, which is the presence of random noise.

Table 1.1: Frequency of occurrence of quality issues in a group of 400 randomly selected user projects.

Quality issue	**Occurrence** (n)	**Occurrence (%)**
Amplitude clipping and clicks	16	4.0
Background music leaking from headphones	102	25.5
Random background noise (hiss)	96	24.0
Coherent background noise (hum)	38	9.5
Echoes	4	1.0
Background noise (TV, mumble, etc.)	16	4.0
Other	38	9.5
No noticeable issues	90	22.5
Total	**400**	**100**

Three particular constraints need to be taken in account regarding the nature of the project. First, because the field of application is online recorded music, the final goal is to develop a workflow with which all recordings can be automatically processed. Therefore, the algorithm needs to be data-adaptive: nothing should be removed from noise-free recordings, and as much noise as possible should be removed otherwise. Second, the application to music leads to strict limits in terms of signal distortion. A small amount of distortion may lead to music that sounds out-of-tune or unnatural; particularly so for vocal recordings, since a user will easily be able to notice any changes to his or her own voice. In terms of the previously discussed signal distortion-versus-noise reduction trade-off, higher residual noise levels are thus preferred over signal distortion. Finally, since the algorithm will potentially process a vast number of recordings, it should be as cheap as possible in a computational sense. The aim of this thesis can thus be concretely formulated as:

"To develop an algorithm that effectively removes random noise from musical recordings, without noticeable distortion of the signal; and that can process all recordings, regardless of initial quality."

With these requirements in mind, the general applicability of the subspace methods discussed in section 1.1.4 appears particularly appealing. However, rather than applying a Karhunen-Loève transform on the time-domain signal, the approach used here is to perform a singular value decomposition of the signal time-frequency representation. If performed in a small enough frame, a linear approximation to the signal may be valid; we can then enhance these linear features using the eigenimage filtering technique from seismic processing.

The remainder of this thesis is organised as follows. Chapter 2 will consider the theory of transforming a signal to a time-frequency representation, as well as the inverse transform back to a waveform. Next, the singular value decomposition on which the algorithm relies will be further elaborated on. In chapter 3, the implementation of the processing sequence will be explained in detail. The performance of the proposed algorithm will be discussed in chapter 4, both by means of objective and subjective evaluation. Finally, in chapter 5, I will provide the key conclusions of this study and indicate in which direction further work may proceed.

Chapter 2

Theoretical principles

In this chapter, the theory behind the proposed noise removal method is described. First, a concise summary of the assumptions made with regard to the music and noise signals is provided (section 2.1). Afterwards, the theory of two main procedures needs to be considered. The first of these comprises the specifics of converting a signal to the time-frequency domain and back to a waveform; the corresponding forward and inverse transforms will be outlined in sections 2.2 and 2.3, respectively. Second, the procedure used to remove noise from the time-frequency representation using singular value decomposition will be derived and discussed in section 2.4. The practical implementation is explained in detail in chapter 3.

2.1 Signal model

After recording, the sampled total signal $x(n)$ is assumed to be a superposition of uncorrelated music and noise constituents, which can be written as

$$x(n) = s(n) + v(n) \tag{2.1}$$

where $s(n)$ and $v(n)$ represent the clean music and noise signals respectively, sampled at discrete time n. The assumption of no correlation between signal and noise is common in the case of white noise problems (for example, see Ephraim & Trees (1995); Plourde & Champagne (2008); Hermus et al. (2007), amongst many others). The clean signal is considered a non-stationary process, because of the time variations in volume and pitch in music. The noise is assumed to be a zero-mean, wide-sense stationary, random process with a Gaussian probability distribution, i.e.:

$$v(n) = \mathcal{N}\left(0, \sigma^2\right) \tag{2.2}$$

2.2 Short-Time Fourier Transform (STFT)

As indicated by our signal model, the noise and clean signals overlap in time, which makes the task of separating them difficult in this domain. It is thus desirable to transform the time-domain waveform to difference space, in which the constituents are easier to distinguish from

each other. A well-known example is the Fourier transform, which transforms a signal to its frequency-domain representation:

$$X(f) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi ft} dt \tag{2.3}$$

where $x(t)$ denotes the time-domain signal to be transformed; $X(f)$ is its Fourier transform, a function of frequency f; and j denotes the imaginary unit. This transformation allows a signal to be evaluated based on its frequency content, which is meaningful in many disciplines of science and engineering.
Though useful in theoretical derivations, equation (2.3) is a continuous-time expression and thus not applicable to sampled waveforms $x(n)$. Instead, the discrete Fourier transform (DFT) is used:

$$X(k) = \mathcal{DFT}\{x(n)\} = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi \frac{nk}{N}} \tag{2.4}$$

in which n represents the time or sample index as before, $k \in [0, 1, 2, \ldots, N-1]$ is the frequency index of the DFT result, and N is the number of samples in $x(n)$. In analogy with the continuous-time formulation, equation (2.4) allows a discrete-time waveform to be represented as a summation of a finite number of discrete frequencies, given by:

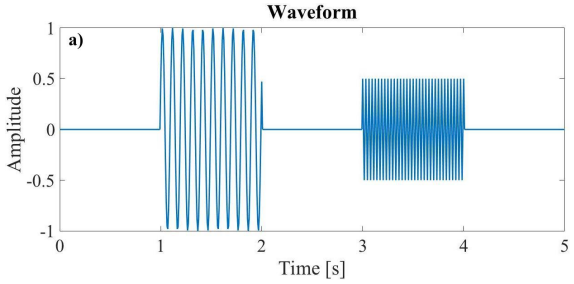
$$f(k) = \frac{f_s k}{N} \tag{2.5}$$

where f_s denotes the sampling frequency of the discretised waveform. In modern audio, this is usually 44100 Hz or 48000 Hz, such that the entire human auditory range 20 Hz – 20 kHz is sampled unaliased in accordance with the Nyquist sampling theorem. It is important for the discussion to follow to realise that the frequency resolution of the DFT is a function of N.

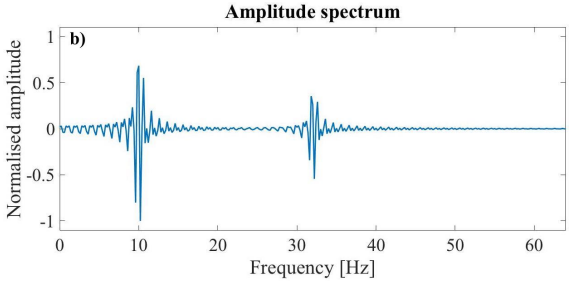
2.2.1 Motivation for a time-frequency representation

Although the Fourier transform is extremely useful in a wide range of applications, it is not particularly well-suited to the analysis of audio signals. This is due to the fact that all temporal information is lost, which is a consequence of the integration or summation over all time. Consider the waveform shown in figure 2.1a, which consists of a windowed 10 Hz sine wave of unit magnitude between 1 and 2 seconds and a windowed 32 Hz sine wave of amplitude 0.5 between 3 and 4 seconds, sampled at $f_s = 128$ Hz. Although the amplitude spectrum clearly shows both peaks around the correct frequencies, as well as the difference in magnitudes, it is impossible to infer the time relationship from figure 2.1b. For example, if the time axis is reversed before taking the discrete Fourier transform, a very similar looking result is obtained (Fig. 2.1d). Although this may not be a problem in some applications, it is clearly undesirable for musical purposes.

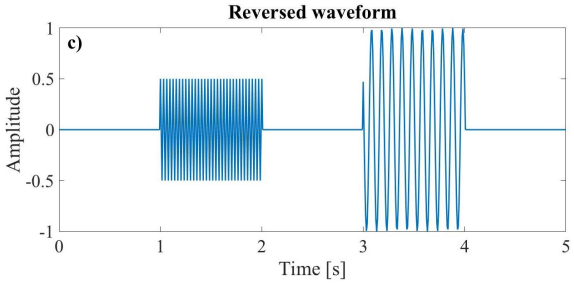
Intuitively, this issue can be addressed by taking the DFT of a limited sub-section of the waveform, and moving the corresponding window along the time axis. This is the procedure by which the short-time Fourier transform operates. In practice, this can be achieved by successively multiplying different sections of the waveform with a windowing function, and taking the DFT of



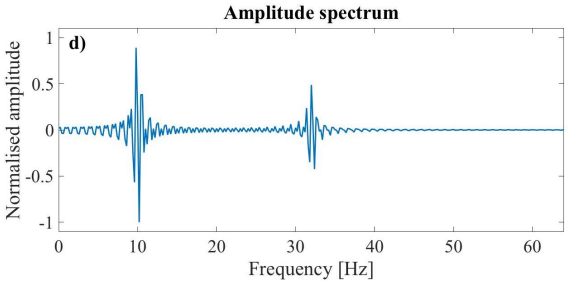
(a) Time-domain waveform.



(b) Corresponding amplitude spectrum.



(c) The waveform shown in a), but time-reversed.



(d) Amplitude spectrum corresponding to c).

Figure 2.1: Top: a discrete Fourier transform pair. The waveform consists of 1-second intervals of two sine waves at frequencies of 10 and 32 Hz, separated by a 1-second interval of zeros. The sampling rate is 128 Hz. **c)** the time-reversed equivalent of the waveform in a). **d)** Amplitude spectrum of the time-reversed waveform.

each multiplication. Mathematically, this is formulated as:

$$\begin{aligned} X(k,\tau) = \mathcal{STFT}\{x(n)\} &= \sum_{n=0}^{N-1} x(n)\, w(n-\tau h)\, e^{-j2\pi \frac{nk}{N}} \\ &= \mathcal{DFT}\{x(n)w(n-\tau h)\} \end{aligned} \tag{2.6}$$

Here, τ denotes the reintroduced time index, h represents the hop size (in samples of x) between successive windows, $w(n)$ is the windowing function, and N now represents the window size (or number of frequency bins) rather than the length of the entire audio array. Figure 2.2 shows the result of applying equation (2.6) to the waveform in figure 2.1a.

A number of important remarks can be made with respect to this image. First, it is clear that the aforementioned problem of time-ambiguity has been resolved, and that this new representation is better suited for signals whose frequency content varies over time. However, there now is not only leakage into nearby frequency bins (an issue originating from the discretisation of the frequency axis), but also a smearing effect in the time direction. The original waveform had non-zero values only in the [1s, 2s] and [3s, 4s] time intervals, but in figure 2.2, that clearly is not the case. This is a consequence of the large width of the analysis window. For example, at the time index $\tau = 9$, corresponding to a time of 2.25 seconds, the window spans a time range of [1.75s, 2.75s]. Consequently, a 0.25-second portion of the 10 Hz sinusoid is included in the windowed frame, resulting in a nonzero amplitude at this point in the time-frequency space. To address this issue, the window width can be reduced from $N = 128$ to 64 to double the time resolution. However, as indicated by equation (2.5), this will double the sampling interval along the frequency axis. Thus, the product of the frequency and time resolution is a constant, and the overall resolution cannot be increased by tuning these parameters. Figure 2.3 illustrates

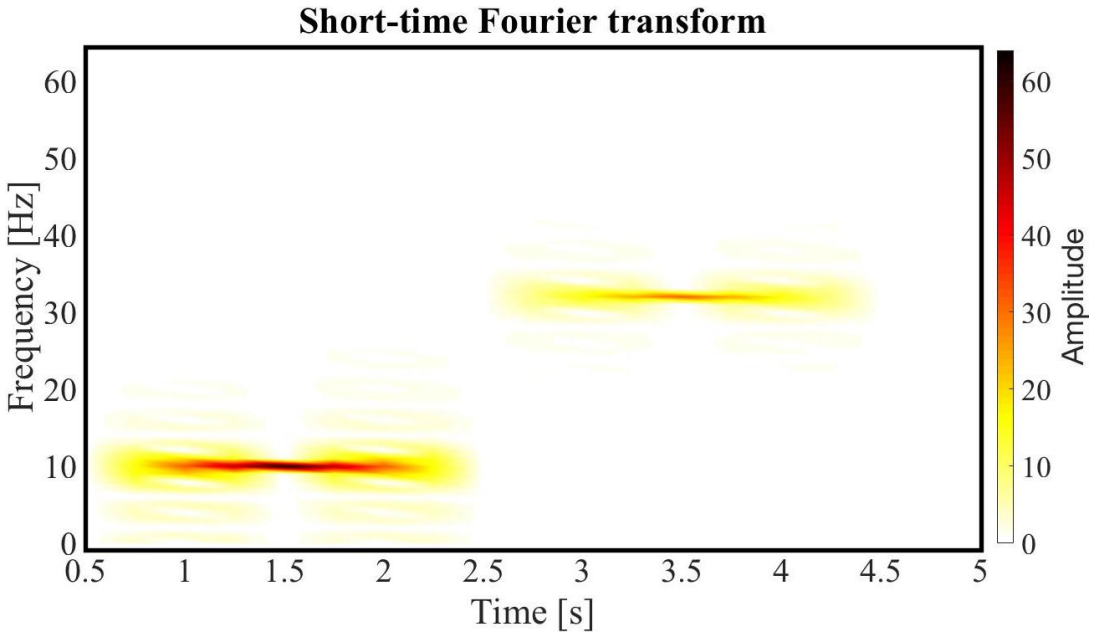
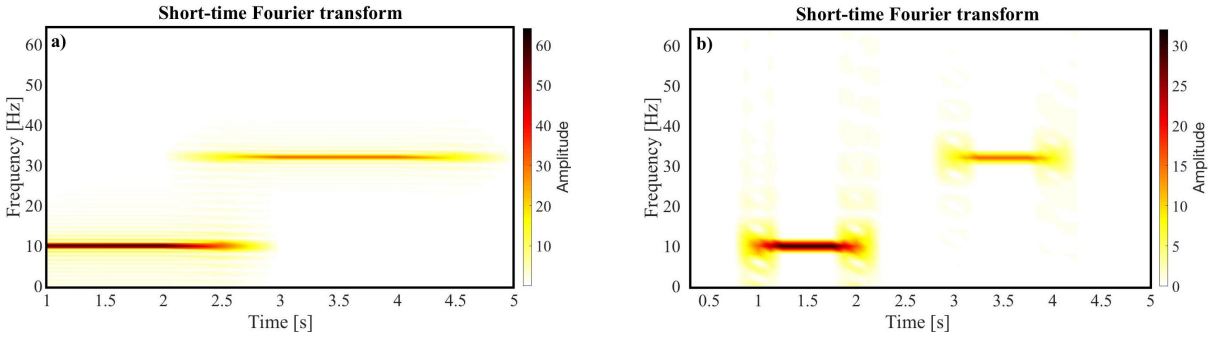


Figure 2.2: The short-time Fourier transform amplitudes of the waveform shown in figure 2.1a, using a rectangular analysis window. The window size N and hop size h to create this image were 128 and 32 samples, respectively, corresponding to a 75% overlap between adjacent frames.

this aspect by showing the same signal as in figure 2.2, but with double and half the frequency resolution. In practice, the resolution in one dimension will usually be chosen based on the specifics of the problem at hand, which automatically determines the resolution in the other direction. The window length is generally chosen to be an integer power of 2, as this allows for the use of computationally efficient radix-2 Fast Fourier Transform (FFT) algorithms. Finally, note that despite their similar appearance, the result of figure 2.2 and the so-called *spectrogram* (commonly used in audio) are not equivalent. A spectrogram shows the variation of the power spectral density (PSD) over time; it is obtained by multiplying each element of the STFT result by its complex conjugate.



(a) $N = 256$, $h = 64$.

(b) $N = 64$, $h = 16$.

Figure 2.3: Illustration of the constant-resolution concept of a STFT. The figure on the left allows us to more precisely indicate the frequencies of the sine waves, but blurs their temporal behaviour. Conversely, the figure on the right shows the times of onset and termination more clearly, at the cost of a loss in frequency resolution.

2.2.2 Analysis windows

As discussed above, a problem with time-frequency images (e.g. figure 2.2) is that they clearly suffer from the adverse effects of spectral leakage. This phenomenon occurs when the signal contains frequencies other than the analysis frequencies given by equation (2.5). It manifests itself as as a pattern of non-zero amplitudes in other frequency bins. Although it is not possible to completely alleviate this issue, its behaviour can be controlled through the use of windowing functions. The previously discussed figures were all created by taking successive DFT's of portions of the array $x(n)$. This is equivalent to multiplication with the rectangular windowing function, which is equal to 1 in a range of N samples, and 0 outside this range (top left in figure 2.4). Although this window allows for a very precise determination of the highest-amplitude frequency, it has unfavourable characteristics in terms of spectral leakage. The occurrence of spectral leakage is best understood by considering the effects of windowing a function on that function's Fourier transform. When two functions are multiplied in one domain, the corresponding action in the transformed domain is convolution. Hence, multiplication of a time-domain signal by a window function (i.e. truncating the time-domain signal) corresponds to convolution of the Fourier transforms of that signal and the window function. The transform pair of a rectangular function is the sinc function; therefore, applying this window to the time-domain signal corresponds to convolving the Fourier transform of the signal with the sinc function. The sinc function reaches its maximum when the argument is zero (the so-called *main lobe*), and has periodic maxima and minima (denoted *side-lobes*) that diminish in amplitude as the function argument increases. This periodic character is imposed on the result of the convolution and thus appears (in sampled form) in the DFT result, causing spectral leakage.

For this reason, it is important to evaluate magnitude responses when choosing a windowing function. In the case of the discrete-time rectangular window (shown in the top left of figure 2.4), the magnitude response is a sampled version of the sinc function (figure 2.4, top right). The side-lobes are of relatively high amplitude: the first side-lobe has a magnitude of -13dB compared to the main lobe. This is usually undesirable, because when a low and high-amplitude signal exist at the same time, the former could potentially be masked by the leakage of the latter. There are a multitude of windows with a magnitude response more suitable to audio signal processing. In choosing an appropriate analysis window, three characteristics are of main interest:

- Height of the side-lobes
- Width of the main lobe
- Constant overlap add (COVA) property of the window

The third point will be discussed in the section on the inverse STFT (Sec. 2.3). For the first two, an improvement in the one generally comes at the expense of the other. As mentioned, the rectangular window has high side-lobes (prominent leakage), but a narrow main lobe (high frequency resolution). In audio signal processing, it is generally preferred to reduce the amplitude of the spectral leakage into distant frequency bins, as it could potentially mask lower-amplitude features. This generally comes at the cost of a wider main lobe. A good compromise for audio

applications are the Hann and Hamming windows (Lyons, 2011):

$$w_{Hann}(n) = 0.5 - 0.5cos\left(\frac{2\pi n}{N}\right) \tag{2.7a}$$

$$w_{Hamming}(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N}\right) \tag{2.7b}$$

$$\text{for } n = 0, 1, 2, ..., N-1.$$

As can be derived from their mathematical formulations, the difference is that the first and last points of the Hann window just reach zero, whereas those of the Hamming window do not. The Hamming window was designed specifically to minimise the amplitude of the nearest side-lobe, whereas the side-lobes of the Hann window diminish in amplitude more quickly (Fig. 2.4). The difference is subtle, and in audio processing literature, both the Hamming window (Cohen, 2005; Hu & Loizou, 2003; Breithaupt et al., 2008) and the Hann window (Boll, 1979; Ephraim & Trees, 1995; Yousefian et al., 2014) are very common choices.

Figure 2.5 shows the short-time Fourier transform for the same waveform as before, but using the three different analysis windows of figure 2.4. The effect of using a window with a wider main lobe are clearly visible; the frequency of the sine wave is not as well resolved when using the Hann or Hamming windows. However, when compared to the maximum amplitude at the true frequencies 10 Hz and 32 Hz, the leakage into other frequency bins is of much lower amplitude.

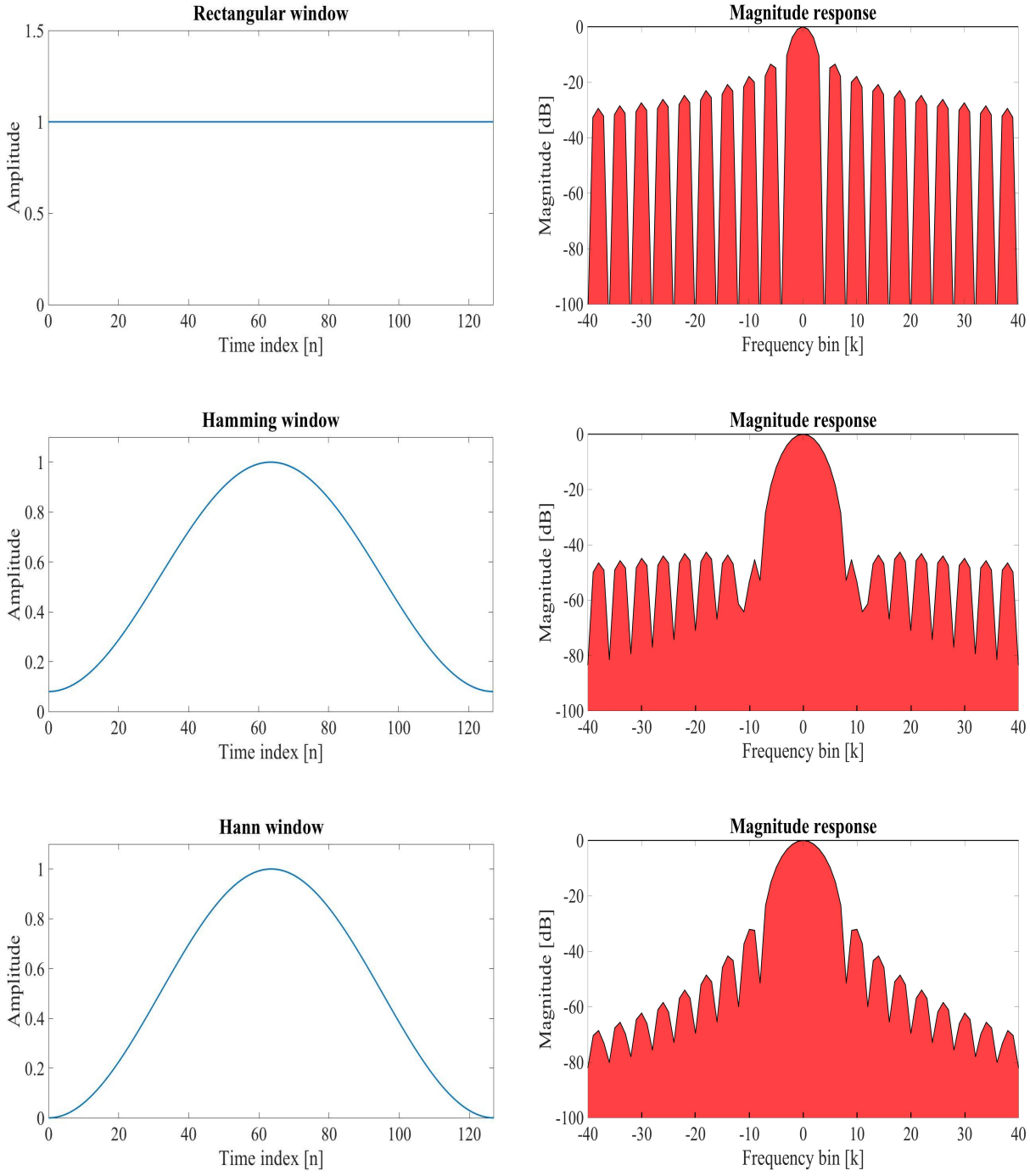


Figure 2.4: Analysis windows (left) and their corresponding magnitude responses (right). Top: rectangular window. Middle: Hamming window. Bottom: Hann window. The maximum side-lobe levels for the rectangular, Hamming and Hann windows are -13.3dB, -42.7dB, and -31.5dB, respectively.

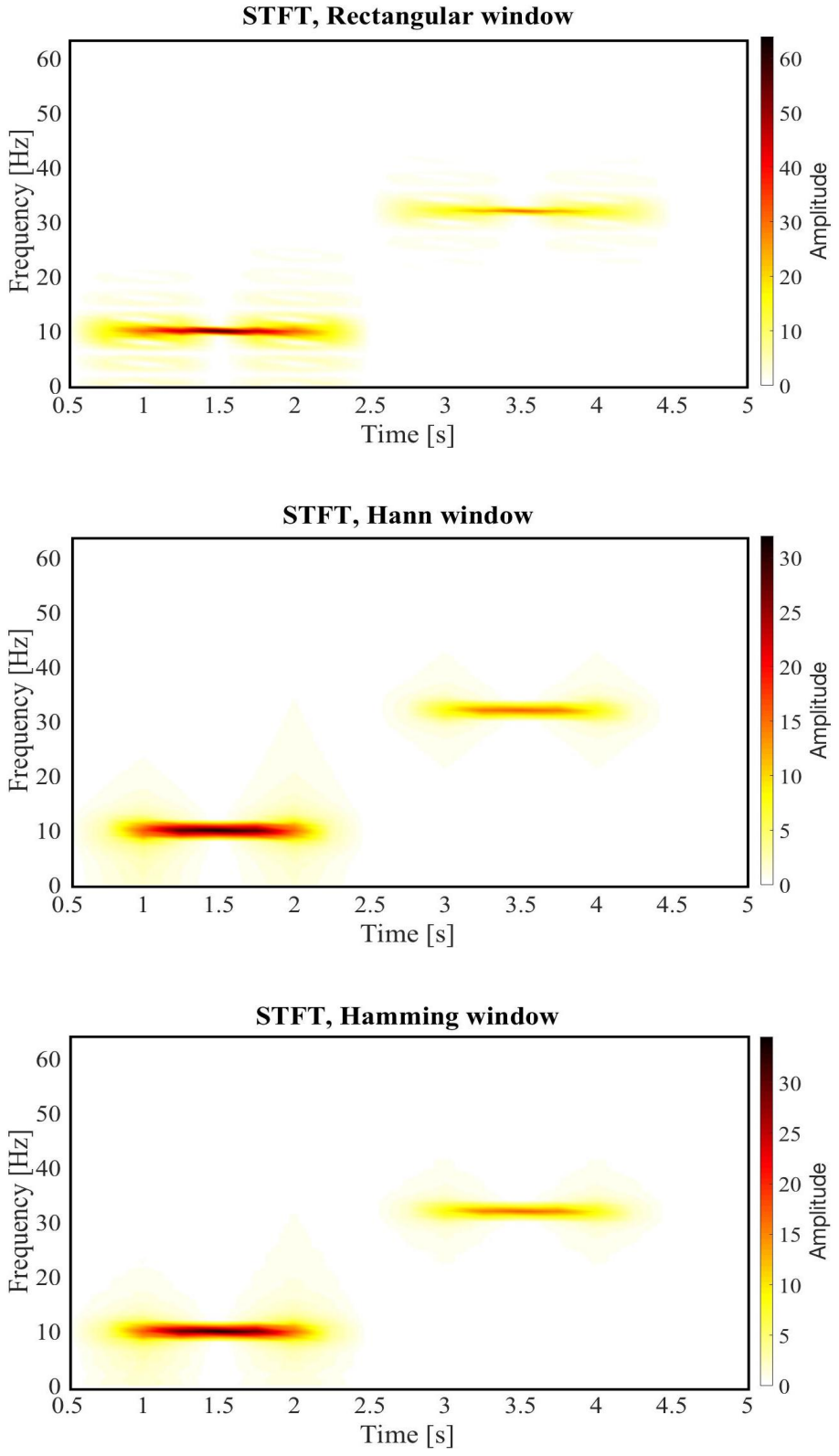


Figure 2.5: The STFT results using three different analysis windows. Top: Rectangular window (same as in figure 2.2). Middle: Hamming window. Bottom: Hann window. The other parameters (window width, hop size) have been kept the same as in figure 2.2.

2.3 Inverse Short-Time Fourier Transform (ISTFT)

Although the STFT clearly provides a very convenient way to represent a musical signal, its usage for filtering purposes is only justified if an inverse transform can be formulated. Ideally, the inverse would be exact, meaning that $x(n)$ can be exactly retrieved from just its short-time Fourier transform $X(k,\tau)$. However, this is not the case for the inverse short-time Fourier transform, which is not exact. In particular, alteration of some coefficients in $X(k,\tau)$ may lead to an invalid result (Griffin & Lim, 1984). This means that a real signal $y(n)$ whose STFT $Y(k,\tau)$ is exactly equal to the modified version of $X(k,\tau)$ may not exist. Therefore, in order to allow for filtering in the time-frequency space, Griffin & Lim (1984) derived a procedure to obtain the signal that corresponds best to the invalid STFT. This is the signal $y(n)$, whose short-time Fourier transform $Y(k,\tau)$ is closest to the modified $X(k,\tau)$ in the minimum mean square error sense. In mathematical terms, this can be expressed as:

$$\text{minimise } \epsilon = \sum_{\tau=-\infty}^{\infty} \sum_{k=0}^{N-1} |\, X_w(k,\tau h) - Y_w(k,\tau h) \,|^2 \tag{2.8}$$

where the subscript $_w$ denotes that a specific analysis window function $w(n)$ was used in creating the STFT. Using Parseval's theorem for the DFT:

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \tag{2.9}$$

equation (2.8) can be reformulated as:

$$\text{minimise } \epsilon = \frac{1}{N} \sum_{\tau=-\infty}^{\infty} \sum_{n=0}^{N-1} |\, x_w(n,\tau h) - y_w(n,\tau h) \,|^2 \tag{2.10}$$

where

$$x_w(n,\tau h) = w(n-\tau h)x(n) \tag{2.11a}$$

$$y_w(n,\tau h) = \frac{1}{N} \sum_{k=0}^{N-1} Y_w(k,\tau h) e^{j2\pi \frac{nk}{N}} = \mathcal{IDFT}\left\{Y_w(k,\tau h)\right\} \tag{2.11b}$$

Equation (2.10) is a quadratic equation of $x(n)$. The minimum can thus be found by finding the derivative with respect to $x(n)$, equating this expression to zero, and solving for $x(n)$. The result of this process is (Griffin & Lim, 1984):

$$x(n) = \frac{\displaystyle\sum_{\tau=-\infty}^{\infty} w(n-\tau h) y_w(n,\tau h)}{\displaystyle\sum_{\tau=-\infty}^{\infty} w^2(n-\tau h)} \tag{2.12}$$

This expression describes how the filtered audio $x(n)$ (or, more accurately, its best approximation) can be retrieved from a modified short-time Fourier transform. Specifically, it states that the final value of a sample in $x(n)$ is the sum of a number of contributions. Each contribution is the product of the inverse DFT of a column of Y_w at time τh and a window function $w(n-\tau h)$.

The sum of these contributions is normalised by the sum over the squared window function. In the context of the inverse transform, w is usually referred to as the synthesis window; however, as indicated by the notation, the synthesis and analysis windows are equivalent.

As can be deduced from equation (2.12), the hop size value h influences the number of values that are superposed to yield $x(n)$. If $h = 1$ sample, a total of N samples will be added at a given location to construct the result, and equation (2.6) will contain the expression of discrete convolution. Although this is not a problem from a theoretical point of view, three practical consequences should be kept in mind. First, the reconstruction as given by equation (2.12) will require a factor of τ more computations, which will slow down processing. Second, the number of points in the STFT is inversely proportional to the value of h, so each reduction in hop size is associated with an increase in memory requirements. Third, by using more points used to represent $x(n)$ in the time-frequency space, the information content is not increased. Instead, the redundancy of the representation is increased, as is the correlation between adjacent spectra (Benesty et al., 2011).

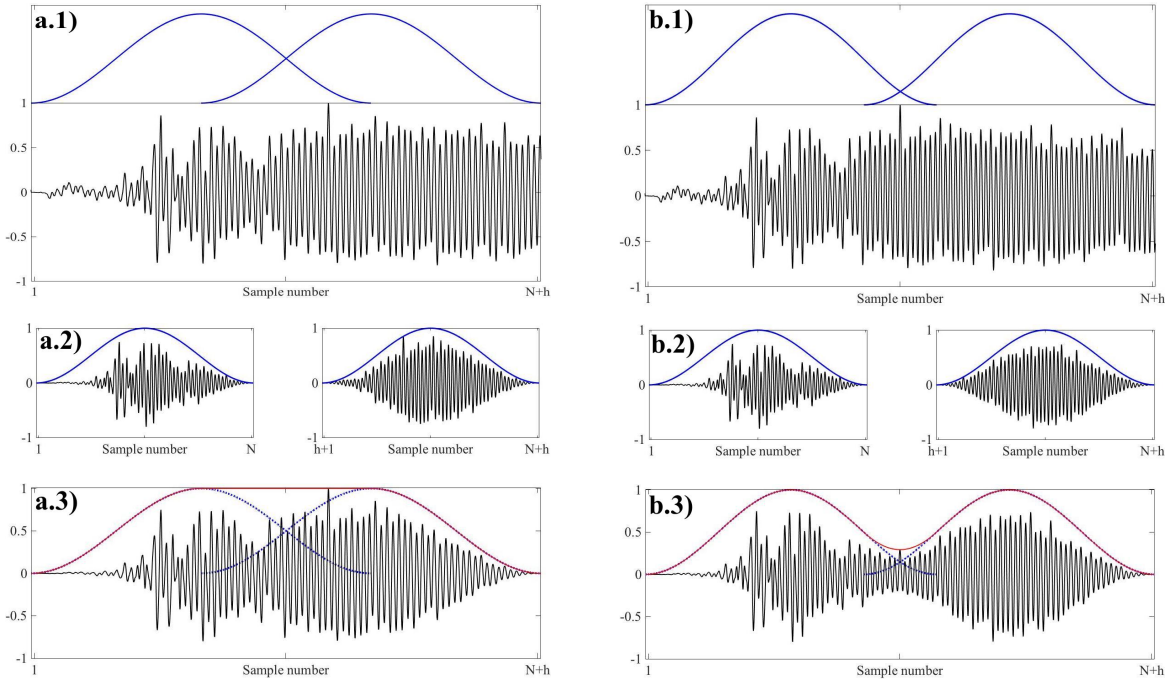
2.3.1 COVA-analysis and exactness of the ISTFT

Suppose now that a simple STFT-ISTFT combination is performed on the original audio array. In equation (2.12), we substitute $x_w(n, \tau h) = w(n - \tau h)x(n)$ for $y_w(n, \tau h)$, and the resynthesis is done by superposition of all windowed vectors x_w in the summation over τ. However, if the window hop size h is not chosen carefully, the windowing in equation (2.12) may cause reconstructions errors. This is illustrated in figure 2.6, which shows how two vectors x_w are extracted from a larger vector x using a Hann window. If h is chosen correctly, as shown in figure 2.6a, the sum of the two Hann windows (shown in red) is constant for the entire range in which they overlap. Within this range, the superposition of the two windowed vectors x_w will thus yield the original result. However, if h is not chosen correctly, as shown in figure 2.6b, the sum of the Hann windows is not a constant. In this case, not all samples n will be weighted equally and the synthesis will not yield the original vector x, because windowing has modulated the signal's envelope. This combination of windowing function and hop size is then said to not satisfy the so-called *constant overlap add (COVA)*-constraint.

Exactly which combinations of windowing function and hop size satisfy the COVA-constraint can be formulated mathematically. The scaling factor $\sum_{\tau=-\infty}^{\infty} w^2(n - \tau h)$ in (2.12) needs to be a constant; not just for all integer multiples of h, but for all points n. Using this rule, it is now possible to define how to choose the value of h: it should be as small as possible (to avoid high redundancy and computational requirements), subject to the constraint that the overlap-add of the squared windows is a constant:

$$\sum_{\tau=-\infty}^{\infty} w^2(n - \tau h) = c \quad \text{for all } n \tag{2.13}$$

As one would expect, the choice of h is intertwined with the choice of $w(n)$. Without exception, all windows satisfy the COVA-constraint in (2.13) in the maximum-redundancy case where h is equal to 1. However, for the common windowing functions, h may be much larger, while still leading to a constant sum. Figure 2.7 illustrates the sum of the squared Hann window for different values of h. Both the squared Hann and Hamming windows are COVA(h=N/4), which indicates that an exact inverse can be obtained when four windows overlap at any point. Any



(a) COVA-compliant superposition of windowed signals.

(b) Superposition of windowed signals that does not satisfy the COVA-constraint.

Figure 2.6: Synthesis of a total signal from windowed segments, performed correctly on the left and incorrectly on the right. In **a.1)** and **b.1)**, the original signal x is shown (in black) alongside two adjacent Hann windows (in blue). In **a.2)** and **b.2)**, the windowed signals x_w are shown. Finally, **a.3)** illustrates how the correct, original amplitudes are retrieved within the overlap range when the sum of the overlapping windows (shown in red) is a constant. Conversely, in **b.3)**, the reconstructed signal is amplitude modulated, because the sum of the windows is not a constant for all n, and this combination of window function and hop size does not satisfy the COVA-constraint.

integer multiple of four will also satisfy the COVA-constraint.

With the COVA-constraint in mind, it should be possible to obtain an exact (within numerical precision) inverse of $x(n)$. Figure 2.8 shows the waveform of a three-second audio file (top left), alongside the waveform obtained by doing an STFT-ISTFT combination (top right). A Hann window of length 1024 was used, along with 75% overlap as dictated by the COVA-constraint. For the vast majority of samples, the error is on the order of 10^{-15}, which is the order of magnitude to be expected for rounding errors. However, there are significant errors at the start and end of the difference series. This can be understood by looking at figure 2.7: for the first and last samples, the total sum is not a constant, because the number of overlapping windows has not reached its maximum yet. For the 1024-point Hann window with 75% overlap, we find that the index at which the maximum is attained is $1 + 1024 * 0.75 = 769$. For the case of 87.5% overlap, the maximum is reached at sample $1 + 1024 * 0.875 = 897$. With the usual sampling rate of 44100 Hz, this corresponds to only a 17ms or 20ms interval of incorrect values. Nevertheless, it is good practice to address this issue by padding both ends of the audio array with zeros, such that the first and last non-zero values of the audio array fall within the constant-sum range. These values can easily be removed after processing. When this simple workaround is used, the reconstruction error is on the order of the numerical precision for the entire audio length.

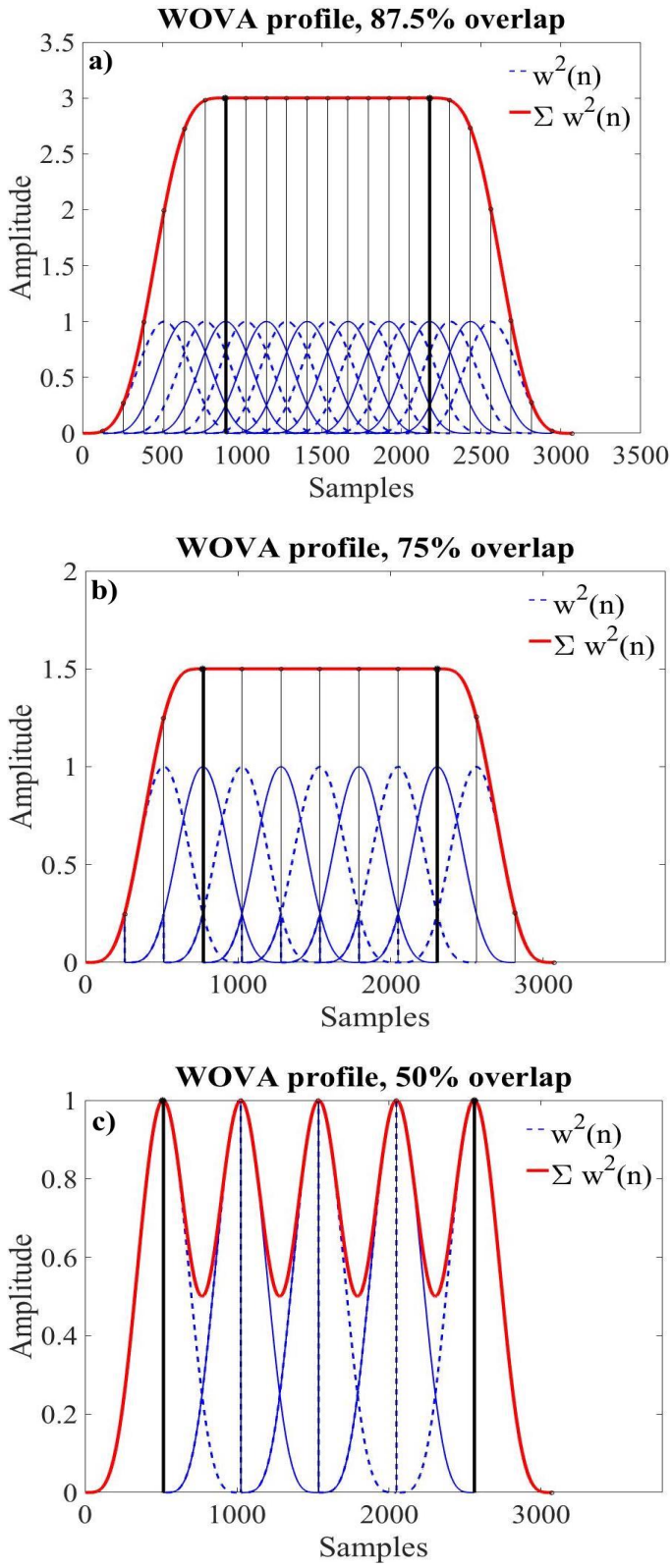


Figure 2.7: Weighted OVA-profile (red) for the 1024-point squared Hann window (blue) for different overlap values. For both figures **a)** and **b)**, which have 8 and 4 overlapping windows ($h = 128, 256$), the COVA-constraint is met. Thus, all samples between the thick vertical black lines will be weighted equally. As shown in **c)**, 50% overlap between squared Hann windows ($h = 512$) does not meet the COVA-constraint.

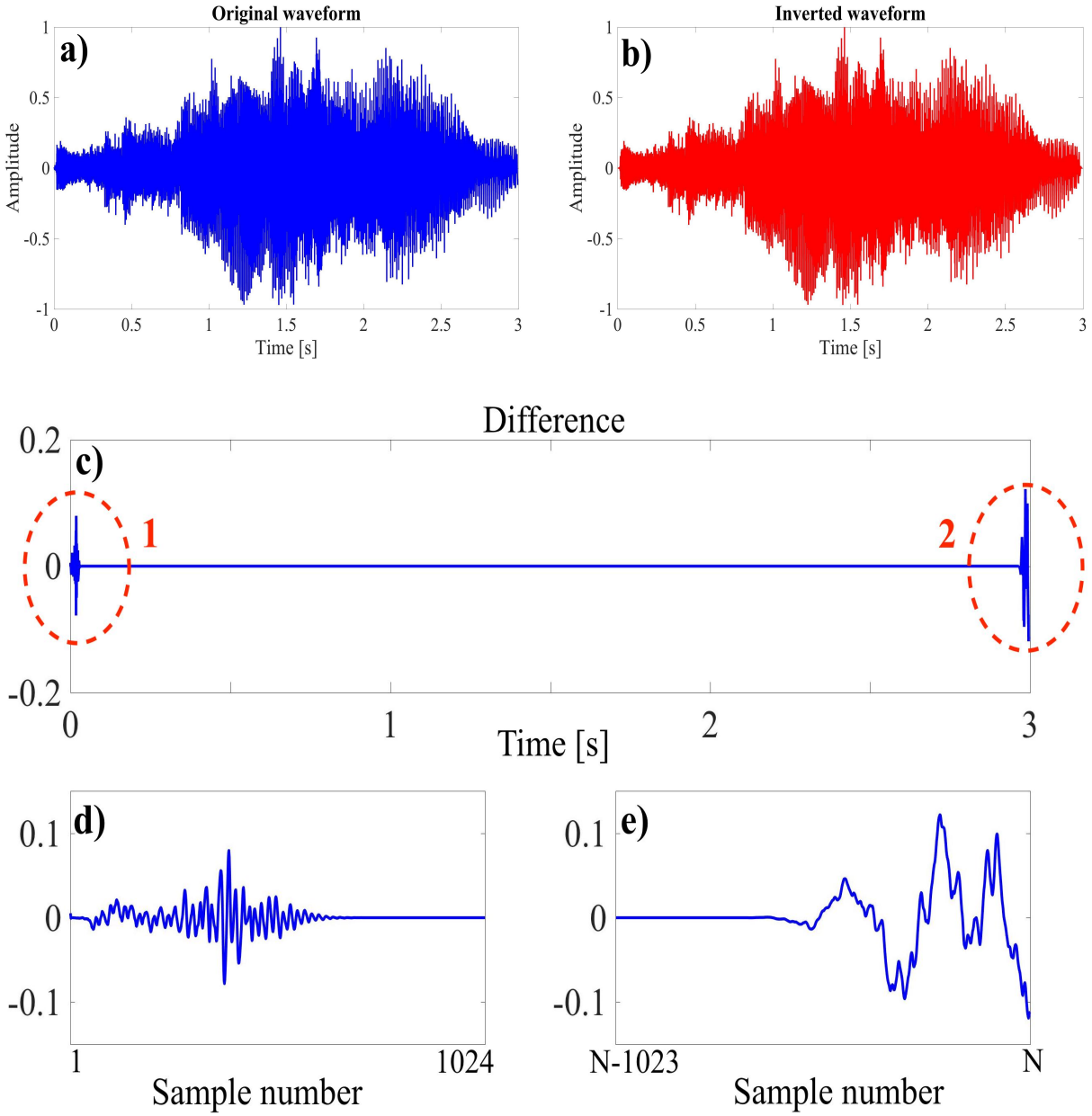


Figure 2.8: The exactness of the inverse short-time Fourier transform. **a)** The original, three-second waveform; **b)** The waveform obtained after performing the ISTFT on the short-time Fourier transform of the original waveform; **c)** The difference plot between the two waveforms at the top. The difference is negligible for the majority of the waveform, but significant at the edges. **d)** Close-up view of the encircled area 1 in c). The difference is significant for the first samples, but decreases rapidly as progressively more windows are overlapping. **e)** Same as d), but for encircled area 2. Note y-axis scales.

2.4 Singular Value Decomposition (SVD)

The noise reduction algorithm proposed in this work relies on a matrix factorisation technique called singular value decomposition (SVD). SVD enables the separation of a signal into signal-plus-noise and a noise-only components. The algorithm is closely related to the eigenimage filtering technique discussed in section 1.2.2, and thus belongs to the class of subspace methods discussed in section 1.1.4. Similar to section 1.1.4, I will follow the notation and definitions of Ulrych & Sacchi (2005).

2.4.1 Mathematical concept

The SVD-theorem states that a general complex data matrix X of size $(N \times M)$ can be uniquely factorised into a product of three matrices as follows:

$$X = U\Sigma V^* \tag{2.14}$$

where the superscript * denotes the conjugate transpose. $U\,(N \times N)$ and $V\,(M \times M)$ are complex, unitary matrices, i.e.:

$$U^*U = UU^* = I, \;\; V^*V = VV^* = I \tag{2.15}$$

and $\Sigma\,(N \times M)$ is a rectangular diagonal matrix that contains the singular values σ_i, sorted by decreasing magnitude. The singular values are always positive real numbers. The matrices U, Σ and V can be understood by considering the relation between SVD and eigenvalue decomposition (EVD). Making use of the unitary nature of U and V, it can be derived from equation (2.14) that:

$$\begin{aligned} XX^* &= [U\Sigma V^*]\;[U\Sigma V^*]^* \\ &= [U\Sigma V^*]\;[V\Sigma^* U^*] \\ &= U(\Sigma\Sigma^*)U^* \\ &= U\Lambda\, U^* \end{aligned} \tag{2.16a}$$

$$\begin{aligned} X^*X &= [U\Sigma V^*]^*\;[U\Sigma V^*] \\ &= [V\Sigma^* U^*]\;[U\Sigma V^*] \\ &= V(\Sigma^*\Sigma)V^* \\ &= V\Lambda\, V^* \end{aligned} \tag{2.16b}$$

Since Σ is a rectangular diagonal matrix, the matrix products $\Sigma^*\Sigma$ and $\Sigma\Sigma^*$ are square, diagonal matrices that contain the entries of Σ squared. Thus, equations (2.16a) and (2.16b) show the eigenvalue decomposition of the square matrices XX^* $(N \times N)$ and X^*X $(M \times M)$, and the singular values are the positive square roots of the eigenvalues in the matrix Λ. Furthermore, U and V are matrices containing the eigenvectors u_i and v_i of the matrices they decompose. In the context of singular value decomposition, these matrices are said to contain the *left-singular* and *right-singular vectors*, respectively. Note that regardless of a potential difference in size, XX^* and X^*X are of equal rank and have the same eigenvalues. The extra diagonal entries of the larger of the matrices $\Sigma^*\Sigma$ and $\Sigma\Sigma^*$ are all equal to zero.
Qualitatively speaking, equations (2.16a) and (2.16b) describe a principal component analysis of the sample covariance matrices of X. The first entries u_1 or v_1 are those vectors that have

the largest variance when XX^* or X^*X is projected onto it. Each of the following entries u_i and v_i then accounts for as much of the remaining variance as possible, under the constraint of orthogonality with all preceding vectors.

2.4.2 Eigenimage filtering a time-frequency representation

As explained briefly in section 1.2.2, noise can be removed from a data matrix X using the SVD in a technique referred to as eigenimage filtering. This involves approximating a full-rank matrix by a rank-deficient reconstruction: in the process, a specific part of the information in X is kept, and the rest is discarded. Assume that X is a full-rank matrix of size $(N \times M)$, with $M < N$. The complete reconstruction of X is given by:

$$X = \sum_{i=1}^{M} \sigma_i \, u_i \, v_i^* \tag{2.17}$$

The outer dot product $u_i \odot v_i^*$ between the two singular vectors creates a so-called eigenimage, which is a rank-one matrix that will be weighted by the associated singular value σ_i in the reconstruction of X. Suppose now that the signal in X, which could be represented by a number of similar columns or rows, is dominant over the noise component. This will have two important consequences. First, these rows or columns are then approximately linear combinations of one another, and the matrix X will be rank-deficient. Second, a series of similar rows or columns will show up prominently in the sample covariance matrices XX^* or X^*X. Since the eigenvalues in equations (2.16a) and (2.16b) are ordered by magnitude, the first eigenvectors u_i or v_i will correspond to the same signal when these matrices are diagonalised. Combined, these two features ensure that dominant linear features in X will reside in the first p eigenimages of X, whereas the reconstruction of random features will require all M eigenimages to be summed. Equation (2.17) can thus be expanded:

$$X = \sum_{i=1}^{p} \sigma_i \, u_i \, v_i^* + \sum_{i=p+1}^{M} \sigma_i \, u_i \, v_i^* \tag{2.18}$$

where the first term corresponds to the signal subspace to be kept, and the second term corresponds to the noise-only subspace to be discarded. This is the principle of eigenimage filtering.

Let us now consider the case where $\boldsymbol{X}$ is the short-time Fourier transform of a noisy signal $\boldsymbol{x}(n)$. In this section, bold symbols (e.g. $\boldsymbol{S}(k,\tau)$, $\boldsymbol{x}(n)$) will be used to denote signal components, to distinguish them from the matrices and vectors (e.g. V, u_i) used in the notation of the singular value decomposition. In order for eigenimage filtering to be an effective noise removal method in the time-frequency domain, there must be a distinction in rank between the time-frequency representations of the signal $\boldsymbol{S}(k,\tau)$ and the noise $\boldsymbol{V}(k,\tau)$, such that truncation of the reconstruction after p components will lead to noise reduction.

First, consider a Gaussian white noise vector $\boldsymbol{v}(n)$ with variance σ_v^2. Since the discrete Fourier transform is an orthogonal transformation of $\boldsymbol{v}(n)$ to $\boldsymbol{V}(k)$, the Fourier coefficients $\Re\{\boldsymbol{V}(k)\}$ and $\Im\{\boldsymbol{V}(k)\}$ will both be independent Gaussian random variables with the same variance σ_v^2. Therefore, the assumption that $\boldsymbol{V}(k,\tau)$ is a full-rank matrix is valid. Let us now consider a superposition of a musical signal $\boldsymbol{s}(n)$ and $\boldsymbol{v}(n)$. For the musical signal, the same waveform as in figure 2.1a is used. Figure 2.9b shows the result of adding Gaussian white noise to this waveform

at a global signal to noise ratio of 0dB. Upon transformation to the time-frequency domain (figure 2.9d), the noise spreads out across all frequency and time indices as expected. Since the signal $\boldsymbol{S}(k,\tau)$ in this example consists of two perfectly linear features, it should be reasonably well reconstructed when only the first two eigenimages are summed. Figure 2.10a shows the reconstruction of this time-frequency representation, using equation (2.18) with $p = 2$. In addition, the removed information is shown, obtained from the second term in equation (2.18).

A number of important characteristics of the method are visible in figures 2.9 and 2.10. First, it is evident that the method has removed a large amount of noise from the noisy waveform. Particularly the second, lower-amplitude waveform has become much more apparent after processing. Second, the residual noise in the TFR is clearly not random. Because the filtered matrix in 2.10a is specified to be of rank two, only those noise coefficients $\boldsymbol{V}(k,\tau)$ that can be obtained by linear combination of the first two eigenimages have been reconstructed. Therefore, the residual noise tends to 'shadow' the strongest signals, meaning that it is concentrated at those frequencies and time intervals that contain strong signal components. As a consequence, the residual noise in the time intervals without signal (e.g. 4-5 seconds) appears to be periodic in the waveform view. Conversely, the difference matrix shown in 2.10b is of near-full rank, and the corresponding waveform does not appear to have any particular structure.

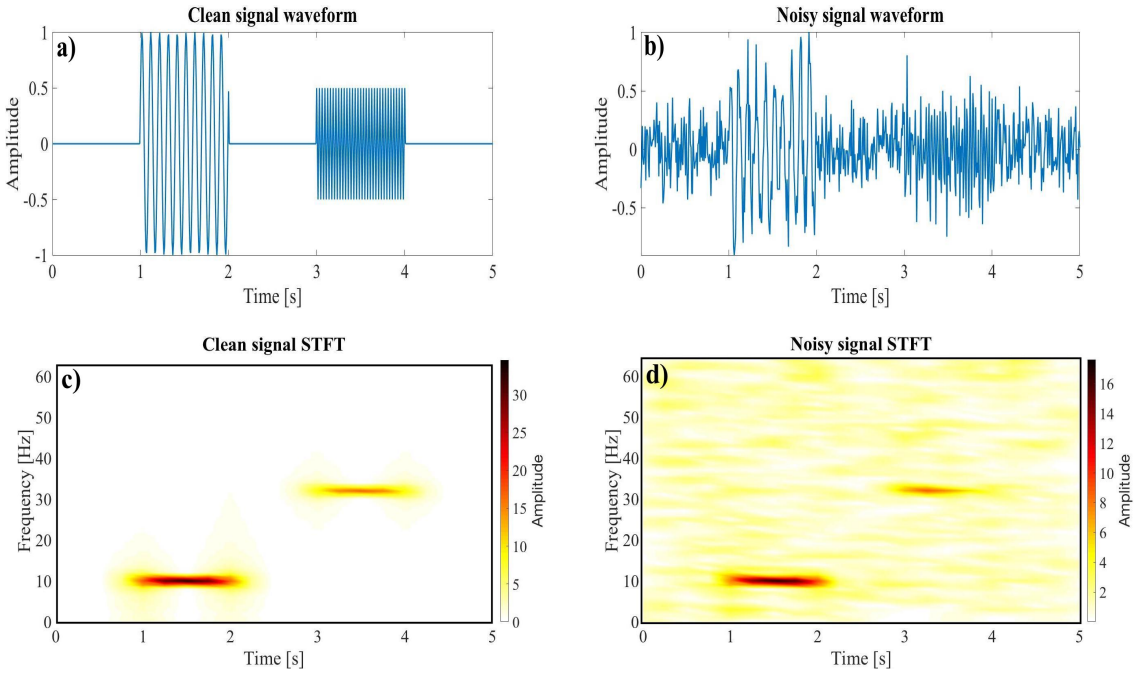


Figure 2.9: Comparison of the waveforms and time-frequency representations of a clean and noisy signal. **a)** The clean signal $\boldsymbol{s}(n)$, the same as in figure 2.1a. **b)** The same waveform, but with added white Gaussian noise $\boldsymbol{v}(n)$ at 0dB SNR. **c)** STFT of the signal shown in a), using a Hamming window. All other parameters equal to those in figure 2.2. **d)** STFT of the signal shown in b), using the same parameters.

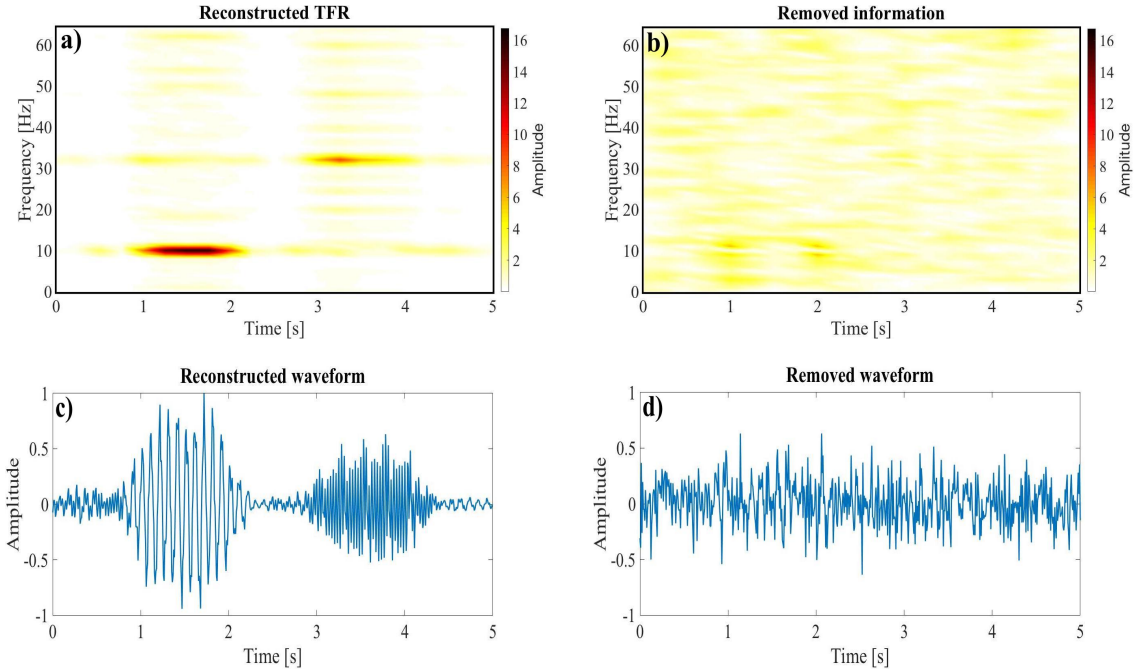


Figure 2.10: The results of eigenimage filtering a signal's short-time Fourier transform. **a)** The reconstruction of the STFT shown in figure 2.9d, obtained by summation of the first two eigenimages. **b)** The corresponding removed STFT, equal to the difference between a) and 2.9d, or the second term in equation (2.18). **c)** The filtered waveform, obtained by performing the inverse STFT of the TFR shown in a). **d)** The removed waveform, obtained by subtracting the result in c) from 2.9b.

The results shown in figures 2.9 and 2.10 illustrate that eigenimage filtering is in principle applicable to the time-frequency representation of a signal. However, when applied to the short-time Fourier transform of a real musical signal, some shortcomings of the method become apparent. Consider the TFR of a real, 8-second recording shown in figure 2.11. Whereas the synthetic example before could conveniently be reconstructed using only 2 eigenimages, this will clearly not be the case here. The structure in this image is considerably more complex, and although a significant part of the signal could be considered approximately linear, a number of features are visible for which this assumption does not hold. For instance, the signal in the area encircled in black (which corresponds to a vibrato) has a sinusoidal appearance. Since the method relies on enhancing linear features, delicate structures such as these are likely to be distorted during filtering.

Figure 2.12 shows that this indeed is the case. Although the highest-amplitude features in the lower frequencies have been reconstructed close to perfectly, the region that is encircled in figure 2.11 has been negatively affected. As the difference image in figure 2.12b indicates, a significant part of the signal has been removed, particularly at the higher frequencies. The obvious solution to this problem is to keep a larger number of eigenimages in the reconstruction. However, the performance in terms of noise reduction is already suboptimal in this example, and keeping more eigenimages will reintroduce more of the original noise. Another issue that needs to be resolved is the selectivity of the noise reduction. As mentioned in the discussion of figure 2.10a, the residual noise tends to shadow the dominant signal components. In the TFR shown in figure 2.11, the vast majority of the signal energy falls in the frequency range of 0 to 2 kHz. Consequently, the noise in the same frequency range has been left almost untouched by eigenimage filtering. Although the noise will perceptually be masked by the much louder audio in some portions, it will dominate the time intervals without signal (for instance, around 3 and 8 seconds).

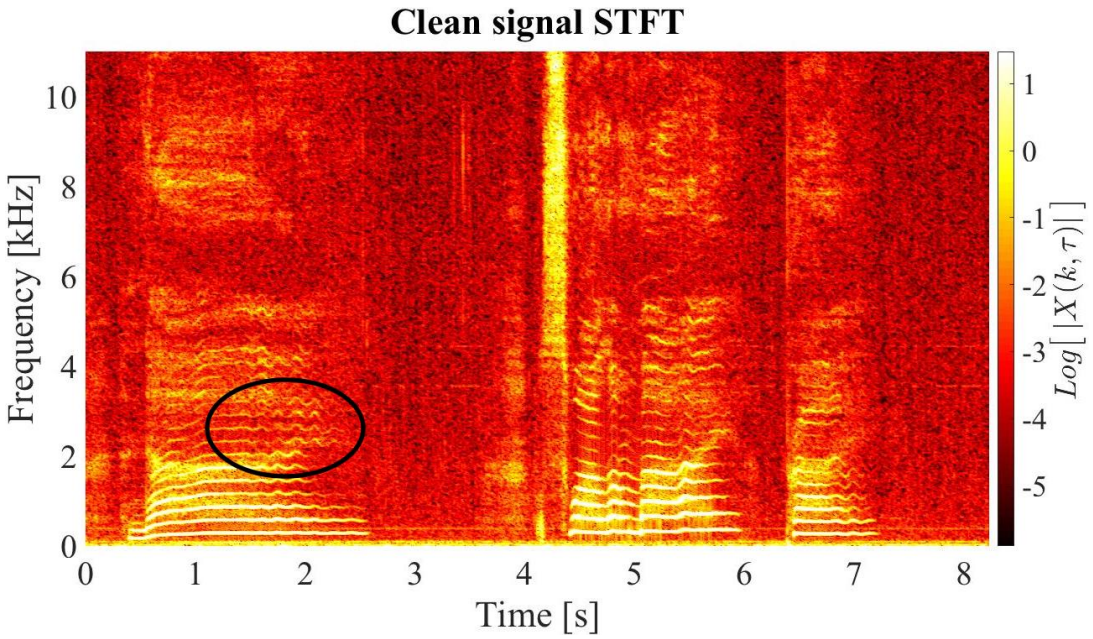
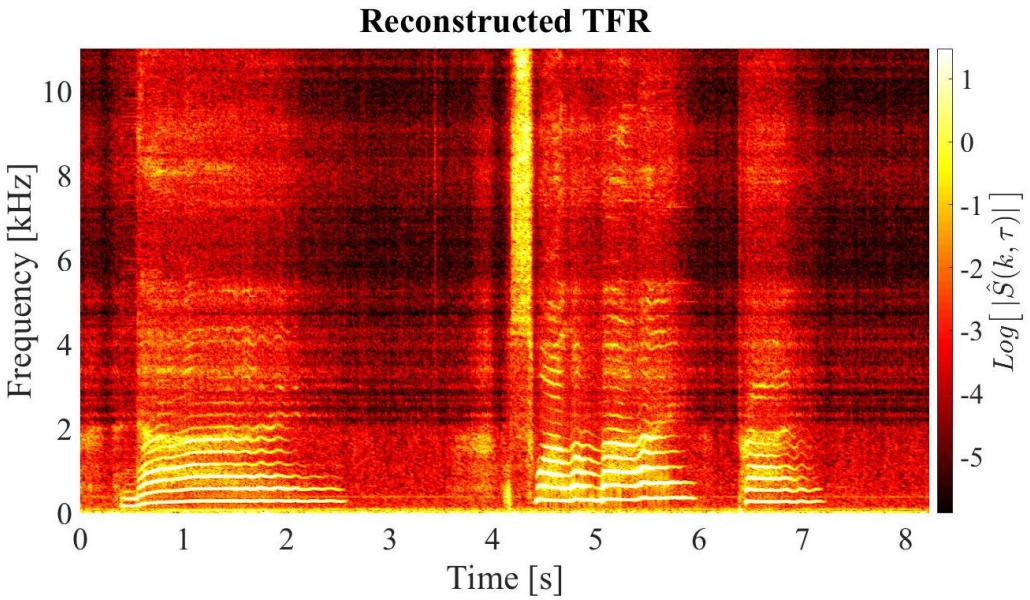
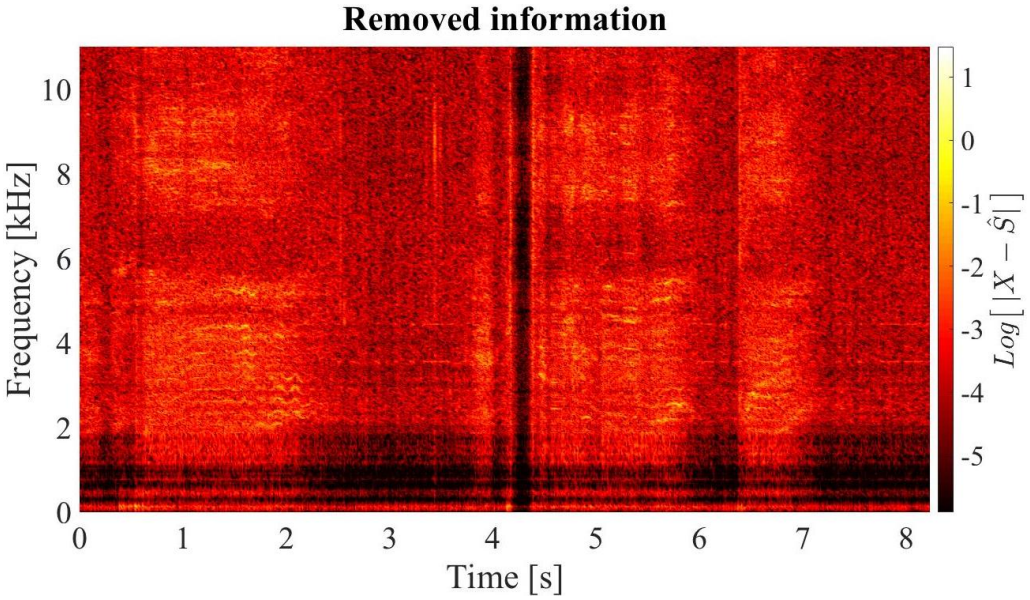


Figure 2.11: The time-frequency representation of a song recorded by a female singer ($f_s = 44100Hz$). The area encircled in black corresponds to a vibrato and is discussed in the text. The Hamming window length and hop size used to create this image were 2048 and 512 samples, respectively. Note that the natural logarithm of the amplitude is plotted, that only the frequencies between 0-10kHz are shown, and that the colour scale has been reversed, for the sake of clarity.



(a) Reconstruction of the TFR shown in figure 2.11, using 125 out of 710 possible eigenimages.



(b) The information removed in the reconstruction, corresponding to the sum of eigenimages 126 to 710.

Figure 2.12: The results of applying eigenimage filtering to the time-frequency representation of a musical signal. Although a substantial amount of noise has been removed, some of it remains, and part of the signal is lost.

2.4.3 Enhancing performance by frame-wise processing

The issues of signal loss and suboptimal noise reduction can be simultaneously resolved by processing the time-frequency representation in a frame-wise manner, rather than all in one go. In speech processing literature, a common assumption is to consider speech a stationary signal over a time range of 30-50 milliseconds (Ephraim & Malah, 1984; Ephraim & Trees, 1995; Lorber & Hoeldrich, 1997). Extraction of a frame of approximately this length from the encircled area in 2.11 yields the results shown in figure 2.13 (in order to increase the time resolution and image quality, the analysis window length and hop size were reduced to 1024 and 128 samples, respectively).

Within this frame, the signal is indeed approximately stationary, despite its sinusoidal appearance on the larger scale. Given the results previously obtained for a time-frequency representation containing signal of linear shape (figure 2.10), one would expect that eigenimage filtering in this case does much less harm to the signal. Moreover, if the next frame were to contain only noise, it would not suffer from the previously mentioned shadowing issue, since the two frames are processed independently. This holds true for both the next frame in the time direction, as well as the adjacent frame in the frequency direction. There is no significant increase in computational cost associated with frame-wise processing, because the singular value decomposition of the smaller matrices converges much more quickly. In terms of memory requirements, dividing the matrix into smaller frames is the preferable approach.

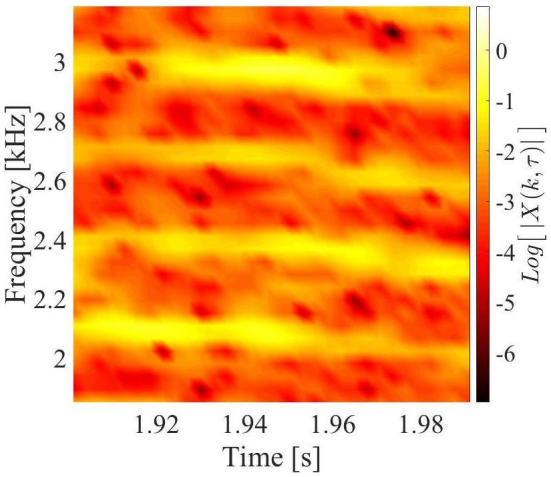


Figure 2.13: A 32-by-32 pixel processing frame, taken from the centre of the encircled area in figure 2.11. Within the frame, the signal is stationary by approximation.

The idea of frame-wise processing gives rise to the question: how can the eigenimage filtering be made adaptive to the contents of the frame? The amount of noise reduction should be dependent on the signal content of the frame; if a frame consists predominantly of signal, the number of eigenimages p should be large, such that the signal in the frame can be properly reconstructed. Conversely, if a frame consists mostly of noise, p must be a smaller number, such that the noise is significantly reduced in volume. To avoid sudden volume jumps between two frames that have been reconstructed using different values of p, adjacent processing frames overlap in the time direction, and a weighted overlap add is used to construct the filtered image.

2.4.4 Removing the noise contribution from the singular values

The key to frame-adaptive noise reduction lies in the singular values. Aside from the fact that they are always real, positive, and sorted by decreasing magnitude, the singular values have a physical interpretation. This is most easily understood by again considering the relation between SVD and EVD. A property of EVD is that the trace of the eigenvalue matrix Λ is identical to the trace of the matrix ($\boldsymbol{X}\boldsymbol{X}^*$ and $\boldsymbol{X}^*\boldsymbol{X}$) that has been factorised. Proceeding with the matrix $\boldsymbol{X}^*\boldsymbol{X}$, which is the most intuitive, the entries on the main diagonal are given by:

$$\begin{aligned}(\boldsymbol{X}^*\boldsymbol{X})_{\tau\tau} &= \sum_{k=1}^{N} (\boldsymbol{x}^*)_{\tau k}\, \boldsymbol{x}_{k\tau} \\ &= \sum_{k=1}^{N} (\bar{\boldsymbol{x}})_{k\tau}\, \boldsymbol{x}_{k\tau} \\ &= \sum_{k=1}^{N} |\boldsymbol{x}_{k\tau}|^2 \end{aligned} \tag{2.19}$$

where the notation $\bar{\boldsymbol{x}}$ is used to denote the scalar complex conjugate, to distinguish it from the matrix conjugate transpose operator. Since $z\bar{z} = |z|^2$, each value $(\boldsymbol{X}^*\boldsymbol{X})_{\tau\tau}$ corresponds to the summed squared moduli of the entries in the column at index τ. Because each column of $\boldsymbol{X}$ contains the local spectrum, equation (2.19) is equal to the power spectral density at time index τ. The trace of the matrix $\boldsymbol{X}^*\boldsymbol{X}$ (and Λ) is therefore equal to the sum of the power spectral densities of all columns, a measure of the total energy contained within the frame $\boldsymbol{X}$.

Since the singular values are the positive square roots of the eigenvalues, they are also related to the energy content, albeit via a square-root relation. This is illustrated in figure 2.14, which shows the magnitude of the singular values for three frames of different content: one with a strong signal presence, one with a much weaker signal, and one containing only noise. Two important characteristics of the singular values are visible here. First, the previously discussed relation with energy content is obvious: the squared singular values for the noise, mixed, and signal frames add up to 1, 5.7, and 639, respectively. Second, all graphs tend toward the same magnitudes at the higher indices, regardless of the signal content in the corresponding frame. This observation corroborates the assumption of distinct signal and noise subspaces: provided that the noise is wide-sense stationary, it should be represented by singular values of similar magnitude across all analysis frames.

The conventional procedure to reduce the noise using eigenimage filtering would be to truncate the reconstruction after p singular values, essentially nulling the $N - p$ last values. There are multiple possibilities in defining the value of p. It could for instance be decided based on a threshold, where the value of σ_i is unchanged if it is larger than this threshold, and set to zero if it is smaller. Since we assume that the noise is wide-sense stationary, the use of a single threshold value seems justified. Alternatively, a threshold could be defined on the basis of the slope of the singular value plot, because in theory, the slope should flatten out in the noise subspace. The problem with both these approaches is that they define the threshold in a binary sense, which can result in the unwanted residual noise characteristics discussed previously (section 2.4.2). Moreover, they remove the noise component in the noise-only subspace, but leave the noise that resides in the signal subspace intact. Especially in the cases were p is a large number, this implies that only a small portion of the noise is removed.

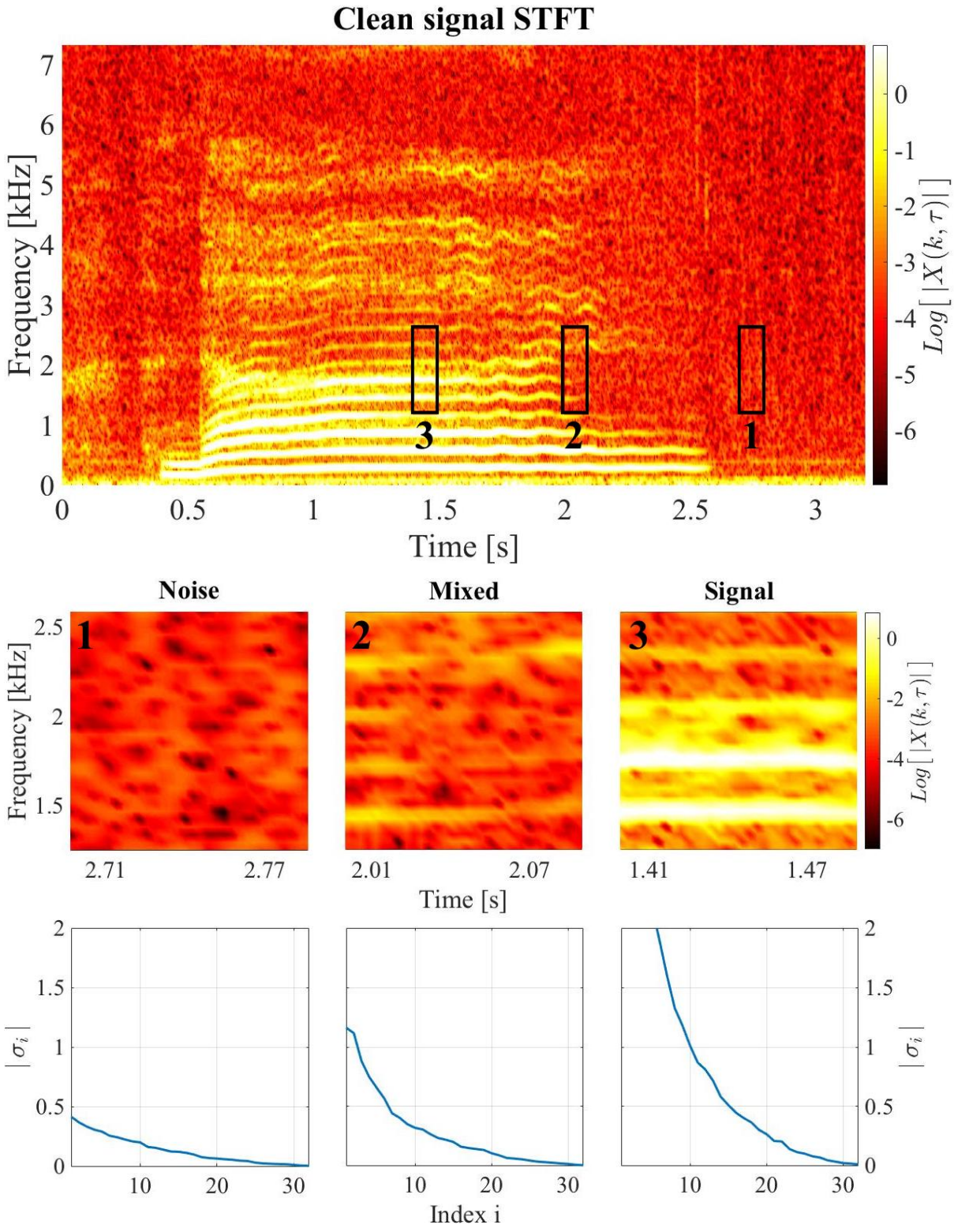


Figure 2.14: **Top:** Zoomed-in view of the time-frequency representation shown in figure 2.11, with three 32-by-32 pixel analysis frames outlined. **Middle:** Detailed view of the three analysis frames. Frames 1, 2 and 3 correspond to noise-only, mixed, and strong-signal frames, respectively. **Bottom:** Singular value magnitudes for the three analysis frames. In the presence of signal, the first singular values will be of high magnitude. The maximum value in the signal frame (bottom right) is 19,2. The last values correspond to noise, and are of similar magnitude regardless of the frame considered.

To circumvent this issue, I use an adapted version of eigenimage filtering, in which the singular values are altered, but none are set to zero. This way, it is ensured that the residual noise is of full rank, whilst still being greatly reduced in volume. The concept is that in the p-dimensional signal subspace, an estimate of the noise contribution is subtracted, whereas in the noise subspace, the singular values are reduced as opposed to nulled. In order to apply this method, three quantities must be determined:

- An estimate of the noise contribution to subtract from the signal subspace, σ_v;
- The dimensionality of the signal subspace, p;
- A factor by which to reduce the noise subspace, α.

An estimate of the noise contribution is obtained from a processing frame that is completely devoid of signal. It is assumed that at least one (90ms) frame without signal always exists, and that this is the frame with the smallest first singular value (an exception to this assumption will be discussed later). Theoretically, all singular values for such a noise-only frame should be of the same magnitude, and equal to the standard deviation of the Fourier coefficients of that noise. This is equivalent to the fact that the eigenvalues of the covariance matrix of a random process are equal to the variance of that process. In practice, however, the singular values will always decay in magnitude as seen in the bottom left in figure 2.14. This is partly a consequence of the correlation between adjacent Fourier coefficients, and partly due to the stochastic nature of the noise. It was empirically determined that setting the threshold σ_v equal to the maximum singular value of the noise-only frame yields the best results.
When the threshold has been determined, the signal subspace dimensionality p for a given analysis frame is defined by the number of singular values greater than σ_v in that frame. The noise energy can then be removed by altering the singular values. However, because the sum of the eigenvalues corresponds to the total energy in the analysis frame $\boldsymbol{X}$, the removal of the noise energy also has to be expressed in terms of the eigenvalues. The removal of the noise energy is achieved as follows:

$$\sigma_i^2 = \lambda_i = \begin{cases} \lambda_i - \sigma_v^2, & \text{if } \lambda_i > \sigma_v^2 \\ \dfrac{\sigma_v^2}{\alpha}, & \text{if } \sigma_v^2 > \lambda_i > \dfrac{\sigma_v^2}{\alpha} \\ \lambda_i, & \text{if } \lambda_i < \frac{\sigma_v^2}{\alpha} \end{cases} \tag{2.20}$$

Figure 2.15 illustrates how the singular values of the mixed frame in figure 2.14 are altered using this procedure. First, σ_v is set to 0.41, which is the first singular value of the noise-only frame in figure 2.14. In the signal subspace, denoted by I, the updated value is obtained using the first case of equation (2.20). In the noise subspace, the values are altered according to the second case of equation (2.20), unless this leads to an increase; in which case, the value is unaltered.

The shape of the singular value plot depends in part on the choice for the tuning parameter α, which enables a degree of freedom in the noise reduction algorithm. A larger value will lead to more noise reduction, but may result in unpleasant-sounding residual noise due to rank-deficiency; a smaller value will reduce the noise to a lesser degree, but ensures that the background sounds more like white noise.

Two final remarks should be made with regard to the proposed method, particularly concerning the threshold value σ_v. First, the analysis frames do not span the entire frequency range. In figure

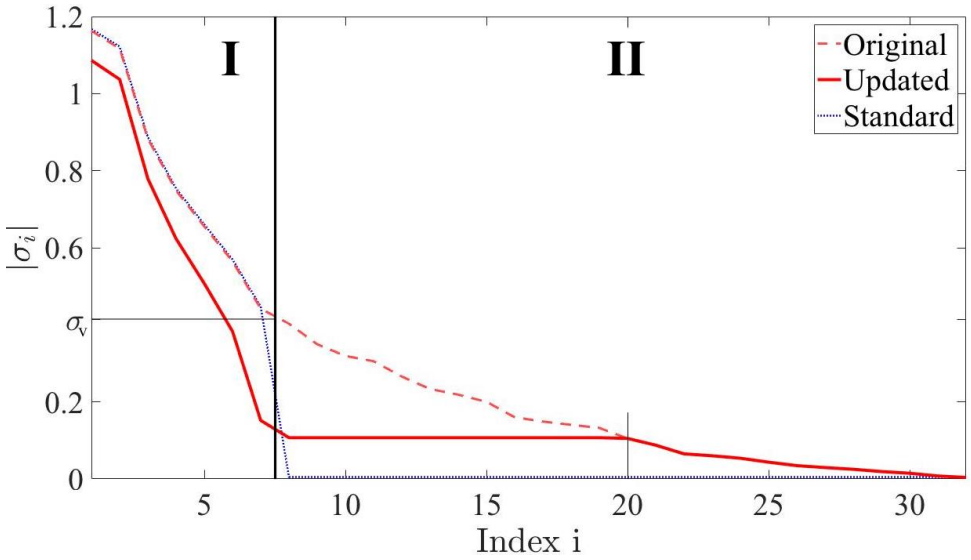


Figure 2.15: Singular value alteration for the mixed frame in figure 2.14, using a value of $\alpha = 15$ for the tuning parameter. **Dashed red:** Original singular value profile. **Solid red:** Altered singular value profile. **Dotted blue:** Singular values as used in standard eigenimage filtering. I and II denote the signal and noise subspaces, respectively.

2.14, for instance, the frames outlined are in the second frequency band (1.25kHZ - 2.5kHz) out of a total 16. By determining a threshold value σ_v for each frequency band separately, the proposed method can enhance recordings with coloured noise as well.
Second, there is an inherent practical danger associated with the way the threshold value is defined. In practice, many recordings are not only contaminated by random noise, but also contain noise of coherent nature (examples are AC hum or noise from the computer fan). This poses a problem for the way the threshold value is defined, as it is assumed that there exists a frame containing only random noise. Because coherent noise is linear in the time-frequency space, it will emerge prominently in the singular values. For this reason, the threshold values of all frequency bands are compared to one another; if the value for one frequency band is excessively high, its value will be changed based on an extrapolation from the other values.

Chapter 3

Implementation

The algorithm was developed in MathWorks' MATLAB_R2016b. Using a regular personal computer with a 1.8 GHz processor, it takes 29 seconds to process a one-minute audio file. The implementation can be subdivided in five main processing stages, as shown in figure 3.1. The following sections of this chapter discuss these stages in chronological order. In the last section of this chapter, the parameter settings are summarised.

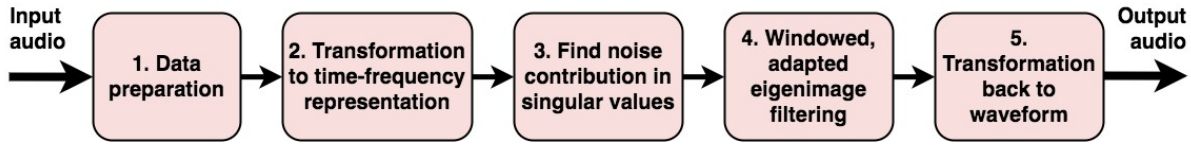


Figure 3.1: Processing sequence of the proposed denoising algorithm.

3.1 Data preparation

The first stage starts by reading in the processing parameters from an input file. These include the following:

- The STFT parameters: window length N and hop size h;
- The SVD parameters: processing frame size L and processing frame overlap β;
- The noise reduction parameter α.

Next, the audio $x(n)$ is read in and converted to mono if it was a stereo recording. Both the start and the end of the array are then padded with $N-h$ zeros on both ends to avoid the end-of-array errors from the inverse STFT (i.e. those shown and discussed in figure 2.8). Additional zeros are padded at the end until the array length is an integer multiple of h: this ensures that the audio array length is unchanged by the STFT and ISTFT. The number of padded zeros (denoted c in figure 3.2) is stored, so that they can be correctly removed at the end of the processing sequence. Finally, the audio is normalised by dividing it by its maximum absolute value.

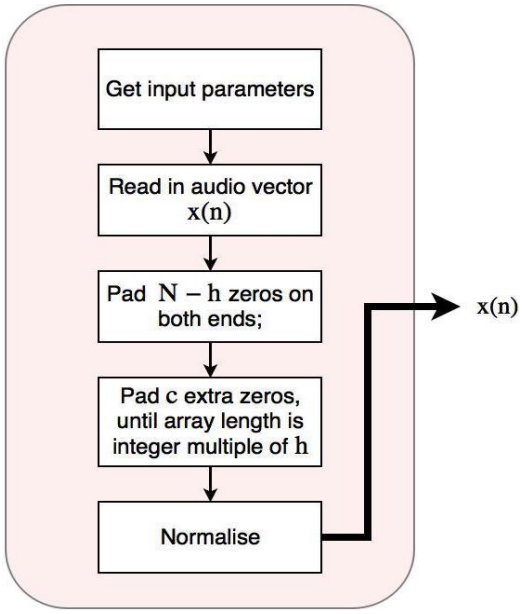


Figure 3.2: Block diagram of the data preparation stage.

3.2 Transformation to time-frequency representation

In the second stage, the audio is converted to the time-frequency space by taking the short-time Fourier transform. The analysis window $w(n)$ is chosen to be a Hamming window of length $N = 1024$ (23 ms at the sampling rate of 44100 Hz), which results in 513 unique frequency points since the redundant negative frequencies are discarded. Thus, the frequency axis of the TFR runs from 0 to 22050 Hz at a bin spacing of 43 Hz.

In choosing the hop size h, two competing interests have to be taken into consideration. On the one hand, h should be as large as possible to minimise the redundancy of the time-frequency representation. This helps to avoid excessive requirements in terms of storage and computational power. On the other hand, the eigenimage filtering frame time length scales proportionally to the value of h, because the frame size is defined in terms of TFR-points (the corresponding parameter L is discussed in section 3.3). These frames should not exceed a length of 50 ms by too much, because the assumption of a stationary musical signal becomes less valid as the frame length increases. Thus, from this perspective, h should be a small value. It was found that using a value of $h = 128$ samples is a good compromise in terms of these considerations. This re-

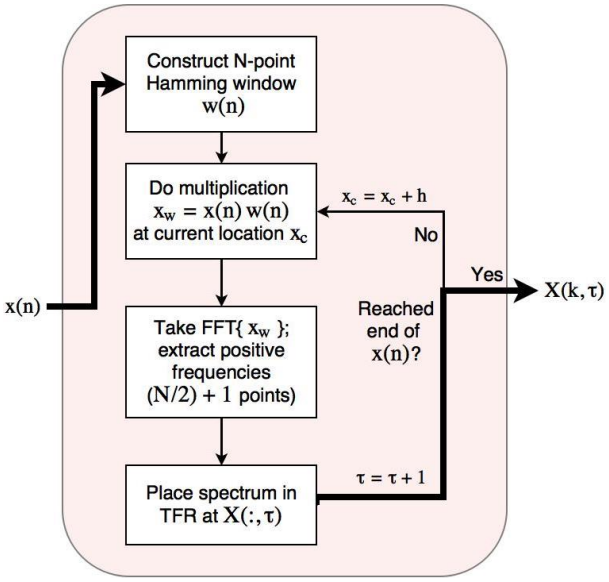


Figure 3.3: Block diagram of the transformation to the time-frequency domain using the STFT.

sults in a TFR time sample spacing of $128/44100 = 2.9$ ms, or 344 points per second. Using these values for N and h, adjacent analysis windows have 87.5% overlap, and the TFR (which is stored as a complex, double precision matrix) requires 160 megabytes of storage per minute of audio.

3.3 Find noise contribution in singular values

After conversion to the time-frequency domain, the noise contribution to the singular values is determined. As this requires doing singular value decomposition, the filter frames have to be defined concretely first. As mentioned in the previous section, the frame size is controlled by the parameter L. Its value should be large enough to allow for distinct signal and noise subspaces to take shape. However, the product $L * h$ should be kept as small as possible, so that the assumption of stationary signal is justified. With $h = 128$, the best results were obtained for $L = 32$ points, which corresponds to a frame length of $(128 * 32)/44100 = 93$ ms. Although this is in slight violation of the recommended $30 - 50$ ms maximum length, no signal distortion was observed using these values.

The processing frames, denoted X^F, overlap in the time direction to ensure a smooth volume profile (see section 2.4.3). An overlap of $\beta = 50\%$ between adjacent frames (i.e. a frame spacing of $L/2$) was found to be optimal: more overlap does not lead to a noticeably smoother result, but does increase the number of frames in the time direction (denoted NT), and thus the computational time. Finally, the extent of the processing frames in the frequency direction is set to the same value of L points, which results in a total of $NF = N/(2 * L) = 16$ frequency bands. However, to account for the fact that the number of frequency bins is 513 rather than 512, the frames in the first frequency band have a height of L+1 points instead.

With the extent in the time and frequency directions defined, all processing frames are factorised using SVD, and the singular values σ_o^F are stored for all frames (the subscript o is used to indicate that these are the original singular values). As discussed in section 2.4.4, the noise threshold σ_v for the frequency band fb is then set to the minimum first singular value of all frames in that frequency band.

The final step is to check the 16-point vector $\sigma_v(fb)$ for abnormally high values, assumed to correspond to coherent noise (see the concluding remarks in section 2.4.4). If any particular value σ_v is more than twice the average of the σ_v's of the two nearest frequency bands, it is judged unreliable. In this case, it will be replaced by a linear interpolation or extrapolation based on those two nearest σ_v values. The resulting vector $\sigma_v(fb)$ is then stored for use in the eigenimage filtering stage.

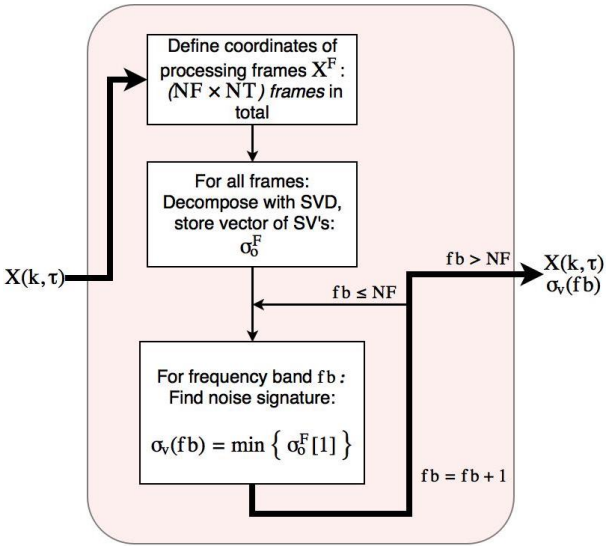


Figure 3.4: Block diagram of the processing stage in which the noise contribution is determined.

3.4 Windowed, adapted eigenimage filtering

The noise filtering takes place in the fourth stage. In a loop over all $(NF \times NT)$ processing frames X^F, each frame is again factorised using singular value decomposition to obtain the original singular value vector $\sigma_o^F = trace(\Sigma^F)$ and the singular vector matrices U^F and V^F[1]. In the next step, the singular values are altered according to equation (2.20), using the correct noise threshold value $\sigma_v(fb)$ for the current frequency band. The filtered frame is then constructed by adding all eigenimages using the new singular values σ_n^F as weights (i.e. equation (2.17)). Finally, the reconstructed processing frames are combined by weighted overlap add, to create the filtered time-frequency representation $\hat{\boldsymbol{S}}(k, \tau)$.

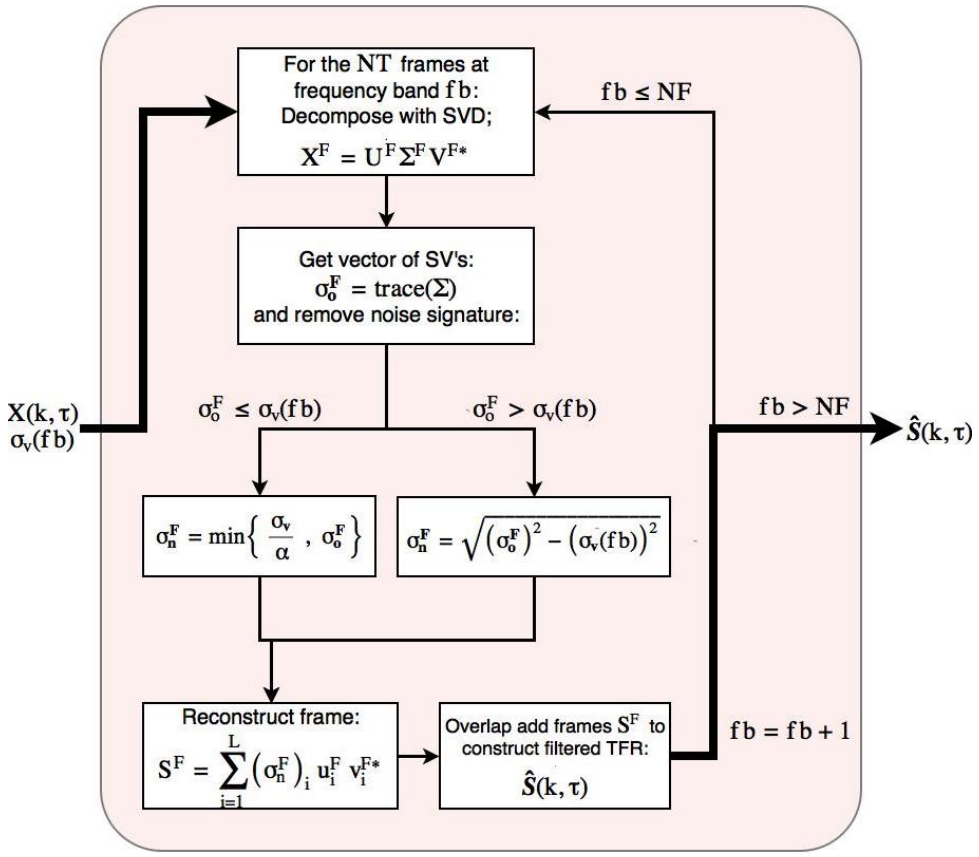


Figure 3.5: Block diagram of the adapted eigenimage filtering stage .

[1] Alternatively, these matrices could be stored for all processing frames in section 3.3 to save computation time in the filtering stage. This speeds up the algorithm as a whole by roughly 25%, but requires 300 megabytes of extra memory space per minute of audio at the discussed settings.

3.5 Transformation back to waveform

The last stage involves using the ISTFT to transform the filtered time-frequency representation to the filtered waveform, denoted $\hat{s}(n)$. The column in $\hat{\boldsymbol{S}}(k,\tau)$ at index $\tau = 1$ is extracted, and the negative frequencies are appended in reverse order (by taking the complex conjugate of the positive frequencies). This yields the full spectrum Y, which has the 0 Hz frequency centered. Next, an inverse FFT is taken of Y to obtain the 1024-point time-domain vector $y(i)$. This vector is multiplied by the synthesis Hamming window of the same length, and the result is added to the array $\hat{s}(n)$ at entries $n = 1$ to 1024. The procedure is repeated for the column at index $\tau = 2$, but it is added at entries $n = 1 + 1h$ to $n = 1024 + 1h$. This procedure is repeated until the last column of the TFR is reached. The last step of the inverse STFT is to divide all values of $\hat{s}(n)$ by the appropriate COVA-constant for the used window and overlap. Finally, the zero padding applied in the first stage is undone, which results in a filtered audio vector $\hat{s}(n)$ of the same length as the original input vector $x(n)$.

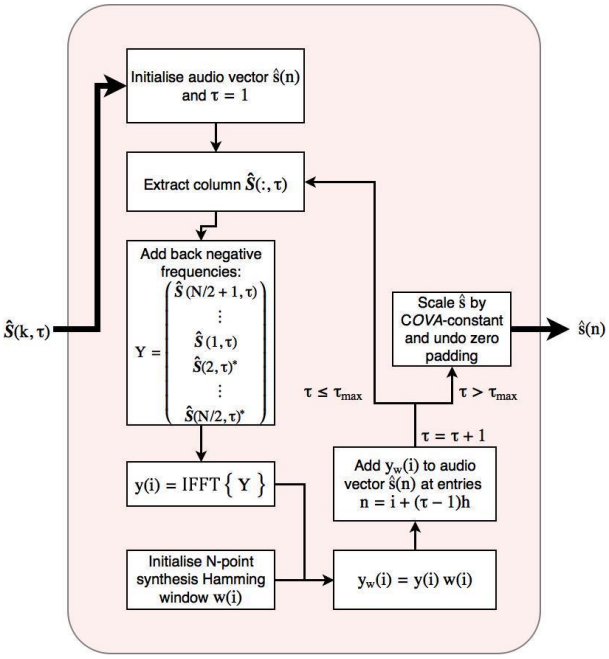


Figure 3.6: Block diagram of the transformation back to a waveform.

3.6 Choice of parameters

The outcome of the proposed algorithm is affected by the choice of parameters. The settings summarised in table 3.1 are considered the standard settings, for the reasons discussed in the previous sections. These are the values that are chosen in the evaluation of the algorithm's performance in chapter 4.

Table 3.1: Recommended parameter settings for the proposed denoising method.

Parameter	**Symbol**	**Value**
STFT window length	N	1024 [samples]
STFT analysis window hop size	h	128 [samples]
SVD processing frame size	L	32 [samples]
SVD processing frame overlap	β	50 [%]
Noise reduction tuning parameter	α	$1 \leq \alpha \leq \infty$

Chapter 4

Results

In this chapter, the results of the proposed method are presented and discussed. A three-step approach is used to assess the performance of the algorithm. First, a description is provided of the library of musical recordings used in the performance assessment (section 4.1). Second, I outline how the effectiveness of the method is quantified using objective quality measures in section 4.2. Finally, in section 4.3 the outcome of a subjective test based on the comparative test methodology ITU-T P.835 (2003) is presented.

4.1 Musical recording library

Musical signals are complex and can have widely varying characteristics, for instance depending on what instrument is played. For example, the sound of a snare drum is relatively broadband but persists for only a short period of time, whereas a triangle or glockenspiel produces sound of a specific frequency with a long decay. Moreover, the signal of instruments that include a sound box (including the human voice) generally contains a large number of harmonic overtones, while the sound of a tuning fork predominantly comprises the fundamental frequency. Hence, it is important to evaluate the performance of the proposed algorithm on the basis of different types of recordings.

For this reason, a musical signal database was created that consists of altered versions of four distinct reference recordings. The music was recorded in a studio using an AKG C414 XLS microphone on cardioid setting, which has a near uniform frequency response[2] and 88dB self-noise SNR[3]. The following instruments were recorded:

- Claves (a percussive instrument consisting of two hollow wooden tubes)
- Acoustic guitar
- Male singing voice
- Glockenspiel (a metallic version of a xylophone)

[2]Frequency response available online at: http://demandware.edgesuite.net/aauj_prd/on/demandware.static/-/Sites-masterCatalog_Harman/default/dwaaaeaa69/pdfs/AKG_C414XLS_Polar_Patterns.pdf [Accessed July 26th, 2017].

[3]Microphone specifications available online at: http://www.akg.com/Microphones/Condenser%20Microphones/C414XLS.html [Accessed July 26th, 2017].

All of these instruments have different characteristics in the time and time-frequency domains, as illustrated in figure 4.1. In musical terms, these instruments are said to have different *timbre*, which is the property that allows the human auditory system to distinguish them from one another. Claves produce a bright, impulsive sound that resembles a person snapping their fingers. The signal of an acoustic guitar consists of abrupt onsets followed by long decays, and contains a very large number of harmonic overtones. The human voice has particularly complex time-frequency characteristics, which include relatively gradual rises and falls in frequency (or pitch), overtones, as well as impulsive, broadband features that are mostly associated with the 's' and 'sh' sounds. The glockenspiel has a comparatively small number of overtones, but a particularly long decay. In addition to their varying signal properties, it was ensured that each of the reference recordings contains a time interval that is largely devoid of signal as well, to allow for evaluation of how the method performs in noise-dominated conditions.

Subsequently, Gaussian white noise was added to the reference recordings at three signal-to-noise ratios. The noise intensity is specified in terms of the global signal-to-noise power ratio (GSNR), given by:

$$GSNR = 10\,log_{10}\left(\frac{\sum_{n=1}^{N} s^2(n)}{\sum_{n=1}^{N} v^2(n)}\right) = 20\,log_{10}\left(\frac{\sum_{n=1}^{N} s(n)}{\sum_{n=1}^{N} v(n)}\right) \tag{4.1}$$

where $s(n)$ and $v(n)$ denote the clean signal and noise vectors of length N. The noise intensities added to the signal are 25dB, 30dB and 35dB; it is judged that noise levels higher than 25dB are unrealistically noisy for musical recording purposes, whereas values below 35dB are perceptually almost irrelevant. Figure 4.2 shows the waveforms and time-frequency representations for the clean and noisy guitar recordings; I refer to appendix A for the figures corresponding to the other reference recordings.

When the waveform figures are considered, the effects of adding noise are visually underwhelming. The time-frequency images, on the other hand, illustrate much better why noise is harmful to the musical signal. Whereas the highest-amplitude signal components are left relatively intact, the much lower-amplitude overtones and decays are partially or completely masked by the raised noise floor. These signal components play an important part in the perception of the previously mentioned timbre: therefore, in addition to being an unpleasant background sound, noise alters how an instrument's sound is perceived.

The noisy recordings were then processed in four ways. Each recording was processed with the proposed adapted eigenimage filtering (AEF) method, using three different values for the noise reduction tuning parameter: a conservative setting $\alpha = 10$, a high-reduction setting $\alpha = 25$, and an intermediate setting $\alpha = 17.5$. In addition, each of the recordings was processed using the spectral noise gating (SNG) method implemented in the popular open-source software package Audacity[4] to set a standard for comparison. The settings used were 20dB noise reduction at a sensitivity of 5 and with one-bin frequency smoothing; note however that the results of this method also vary slightly with the selection of the noise interval. This should not compromise the quality of the results obtained, because each recording includes a noise-only section, which allows for the selection of a signal-free noise sample.

[4] Available online at: http://www.audacityteam.org [Accessed July 26th, 2017].

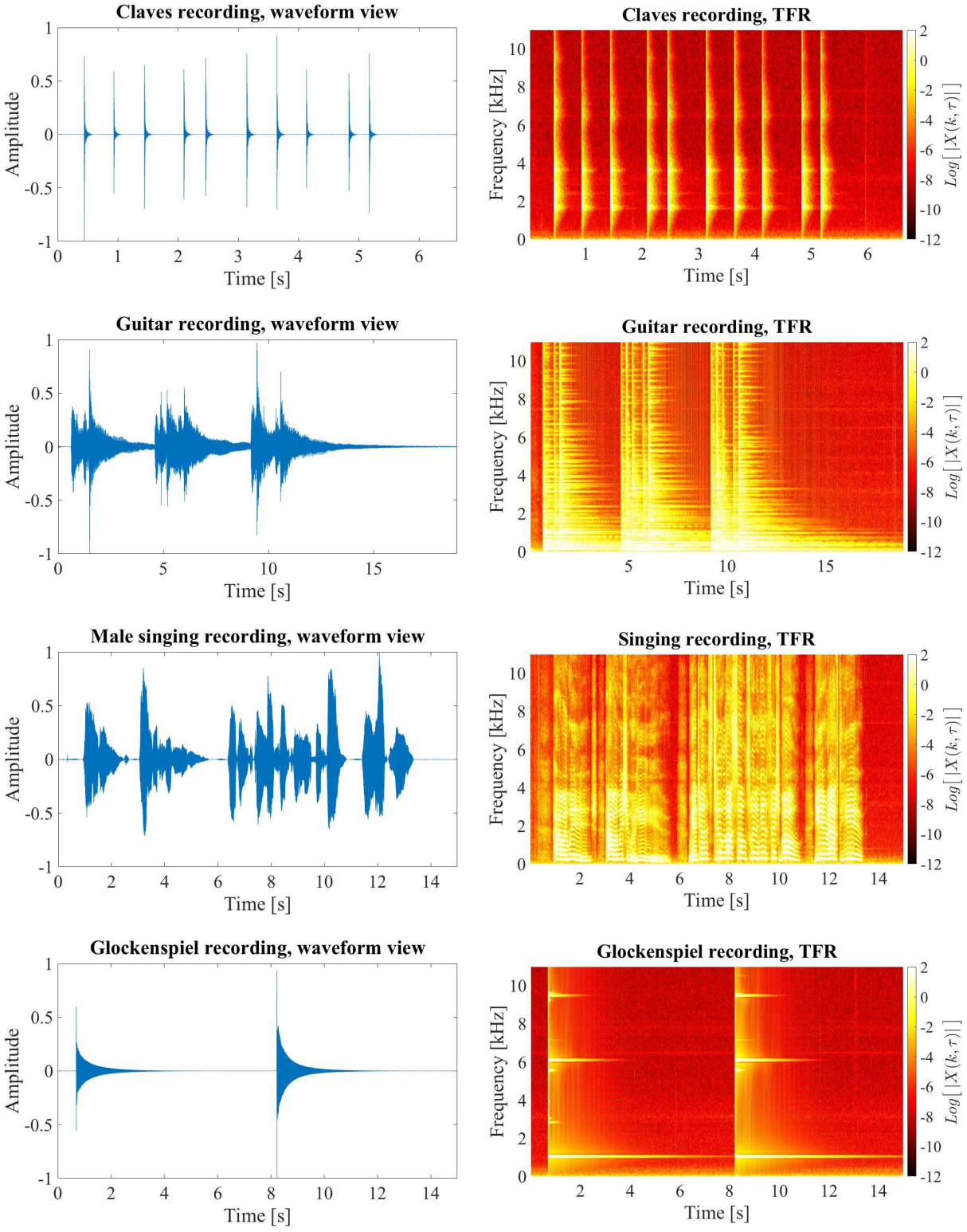


Figure 4.1: Waveform and time-frequency views of the four reference recordings used in performance evaluations. **Top:** Claves. **Second row:** Acoustic guitar. **Third row:** Singing voice (male). **Bottom:** Glockenspiel. The time-frequency representations are shown up to 10 kHz to clarify the most important structures, but extend up to 22050 Hz.

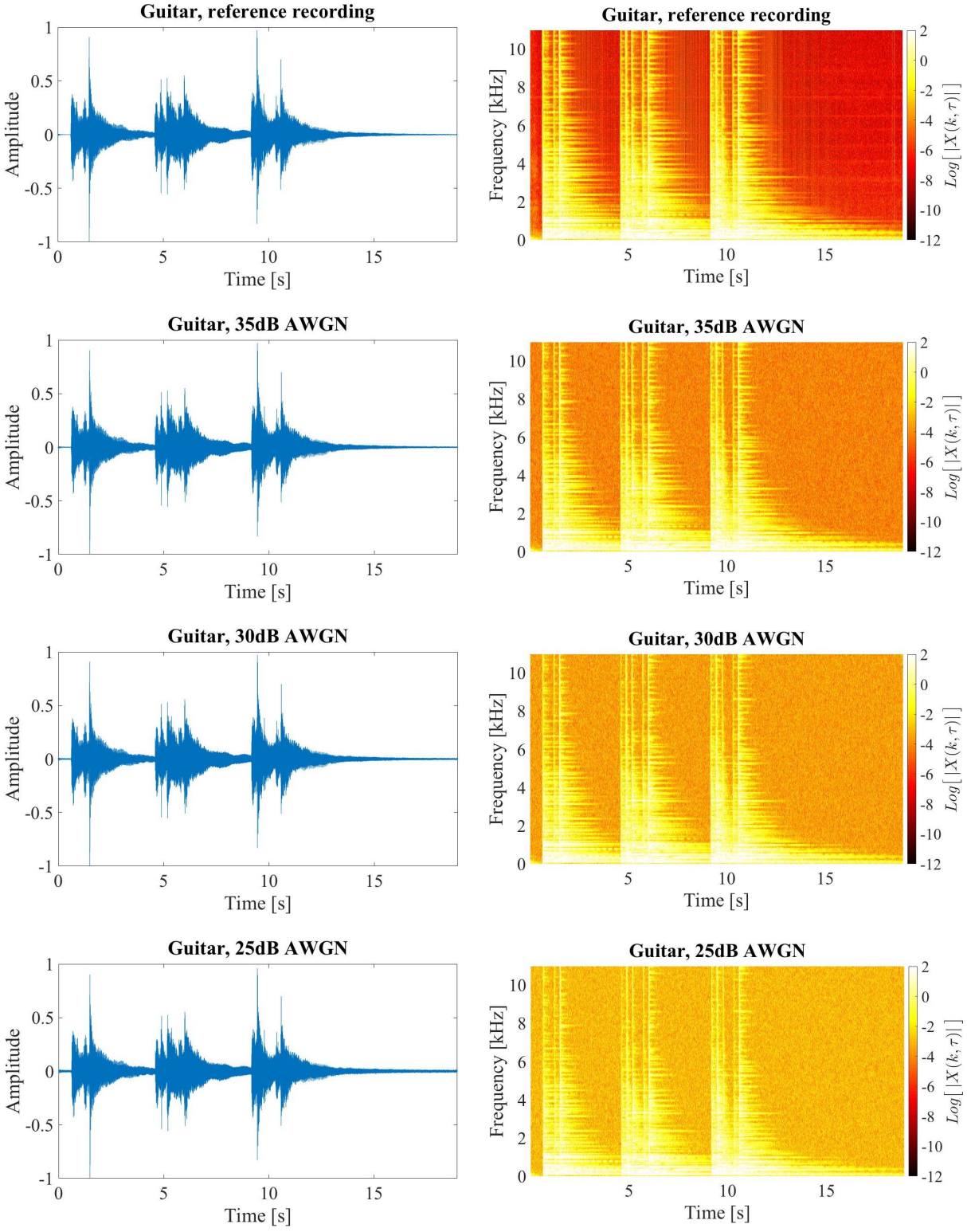


Figure 4.2: Waveforms and corresponding time-frequency representations of the Guitar recording at various noise levels. **Top:** Reference recording, which has a SNR of approximately 88 dB according to the microphone specifications. **Second row:** Reference recording with additive white Gaussian noise (AWGN) at 35dB GSNR. **Third row:** 30dB GSNR. **Bottom:** 25dB GSNR, the highest noise level considered in this work.

This results in 16 total recordings per instrument (one noise-free reference recording, three noisy references, and four processed versions per noisy reference). The time-frequency representations of all these recordings are included in appendix B. The audio library used in testing thus encompasses 64 recordings in total.

4.2 Objective results

The best way to evaluate the effectiveness of audio processing algorithms is to conduct a subjective questionnaire. However, as this can be time-consuming and potentially expensive, it is difficult to conduct such a test on a sufficiently large scale. Consequently, the subjective results may not always be statistically reliable, and a variety of objective measures have been developed to (partially) replace or complement subjective results.
Objective measures for audio processing algorithms work by quantifying some measure of distance between the noise-free and noisy recordings. Examples of such measures are the global SNR and the average segmental SNR in the time-domain, and frequency-weighted segmental SNR in the spectral domain. In their study of objective quality measures, Hu & Loizou (2008) compare how these objective measures correlate to opinion scores given by listeners in subjective tests. They found that whereas some objective measures correlate well with opinion scores on signal distortion, others correlate better with perceived noise reduction or background distortion. Therefore, the best practice is to use multiple objective measures in evaluating the results, or alternatively, to use a composite measure. The results of this study will be quantified using the global and segmental SNR measures, because they are intuitive to understand as well as easily computed. In addition, Hu & Loizou (2008) found that the segmental SNR correlates best out of all simple measures with opinion scores given on background distortion. Since the proposed algorithm contains an explicit trade-off between background noise reduction and background distortion, this measure appears particularly appropriate.

4.2.1 Global signal-to-noise ratio

As discussed in section 4.1, white noise was added to the reference recordings of the four instruments in terms of three global signal-to-noise ratios. After processing these recordings using the proposed method and spectral noise gating, the GSNR was computed again as follows:

$$GSNR = 10\,log_{10}\left[\frac{\sum_{n=1}^{N} x^2(n)}{\sum_{n=1}^{N}[x(n)-\hat{x}(n)]^2}\right] \tag{4.2}$$

where x denotes the noise-free reference recording, and $\hat{x}$ denotes the altered recording of which to determine the GSNR. Naturally, it is critical that both vectors have been aligned and normalised prior to this calculation. The outcomes for all 48 processed recordings are summarised per instrument in tables 4.1-4.4. They are interpreted by comparison to the initial GSNR, in terms of the influence of the tuning parameter α, as well as compared with the results obtained using spectral noise gating.

Dependence on the initial GSNR

The results in tables 4.1-4.4 quantify the GSNR gain obtained by processing the recordings in

various ways. As to be expected, any form of processing usually leads to a marked improvement in the global signal-to-noise ratio. The size of the gain in GSNR itself decreases as the input signal-to-noise ratio increases. For example, the mean GSNR improvement of all four processing methods for the guitar recording is 4.95 at 25 dB, 4.14 at 30 dB, and 3.00 at 35 dB. Note also that the increase in GSNR is larger for the claves and glockenspiel recordings than for the guitar and singing recordings. This is a consequence of the fact that these recordings contain relatively many processing frames without any signal (compare the TFR's in figure 4.1). As a result, a larger portion of the singular values encountered during processing fall in the noise subspace, and are thus reduced more drastically than if they were part of the signal subspace.

Influence of the tuning parameter

The effects of changing the tuning parameter α on the GSNR are unambiguous. All values in tables 4.1-4.4 indicate that higher values lead to increased GSNR gains, with the exception of the change from $\alpha = 17.5$ to 25 at 30 dB GSNR in table 4.3. This is as to be expected, since higher settings increase noise reduction at the cost of noise distortion. Furthermore, the increase from $\alpha = 10$ to 17.5 consistently yields a larger GSNR improvement than that from $\alpha = 17.5$ to 25. This too can be intuitively understood: since the tuning parameter appears in the denominator in equation (2.20), its behaviour should be asymptotic.

Comparison with spectral noise gating

In terms of the global signal-to-noise ratio, the proposed AEF method consistently compares favourably with spectral noise gating at all signal-to-noise ratios and for all instruments. The difference in GSNR gain is approximately 2 dB on average for all instruments at 25 dB noise, but interestingly, the discrepancies are progressively larger as the noise intensity is lowered. At 30 dB, the difference in GSNR gain is 3.5 dB on average, and at 35 dB, it has increased to 7.5 dB on average. Furthermore, the application of the SNG method has actually lead to a GSNR decrease for the guitar and glockenspiel recordings at a noise intensity of 35 dB. Both of these observations are likely not a consequence of poor noise reduction with the SNG method, but rather, they indicate that some degree of signal distortion has occurred. This could cause the numerator in equation (4.2) to decrease, leading to a lower GSNR measure than one would expect.

4.2.2 Segmental signal-to-noise ratio

The segmental signal-to-noise ratio (SSNR) of a recording is obtained by computing the 'global' signal-to-noise ratio's of smaller segments, followed by averaging the scores of all segments. Following the notation of Cohen (2005), the segmental signal-to-noise ratio (SSNR) is computed as follows:

$$SSNR = \frac{1}{L}\sum_{l=0}^{L-1}\mathcal{T}\left\{10\,log_{10}\left[\frac{\sum_{n=1}^{N}x^2(n+lN/8)}{\sum_{n=1}^{N}[x(n+lN/8)-\hat{x}(n+lN/8)]^2}\right]\right\} \tag{4.3}$$

where L is the number of windows in the signal, N is the number of samples per window (1024), and x and $\hat{x}$ denote the reference and noisy audio vectors, respectively. The L windows are the same as those used in the calculation of the short-time Fourier transform, and the division by eight arises from the 87.5% overlap used in the STFT. The operator $\mathcal{T}$ is defined as

Table 4.1: Global signal-to-noise (GSNR) power ratios (in dB) for the claves recordings.

Added GSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
35	41.21	41.74	42.06	36.19
30	36.53	37.11	37.60	36.08
25	32.06	32.73	33.21	31.38

Table 4.2: Global signal-to-noise (GSNR) power ratios (in dB) for the guitar recordings.

Added GSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
35	39.68	39.87	40.09	32.37
30	34.88	35.33	35.50	30.84
25	30.33	30.68	30.87	27.90

Table 4.3: Global signal-to-noise (GSNR) power ratios (in dB) for the singing recordings.

Added GSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
35	38.76	38.83	39.02	35.56
30	34.33	34.38	34.00	31.83
25	30.32	30.68	30.78	29.61

Table 4.4: Global signal-to-noise (GSNR) power ratios (in dB) for the glockenspiel recordings.

Added GSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
35	42.62	43.42	43.82	33.43
30	37.91	38.70	39.28	33.20
25	33.17	34.03	34.64	32.58

$\mathcal{T}\{\, x \,\} = \max[\, \min[35,\, x],\, -10 \,]$, and serves the purpose of confining the SNR of a segment to the perceptible range between -10 dB and +35 dB (Cohen, 2005; Hu & Loizou, 2008). As is the case for the global signal-to-noise ratio, both vectors should be aligned and normalised prior to this calculation.

The results of this calculation are shown in tables 4.5-4.8. Note that the SSNR was calculated not only for the 48 processed recordings (as done for the GSNR), but additionally for the 12 noisy reference recordings. This is necessary because the noise was added to the recordings in terms of the global signal-to-noise ratio; the corresponding SSNR values depend on the characteristics of the recordings. This is clear when the first columns of each table are considered: whereas adding noise at a GSNR of 35 dB leads to a SSNR of 1.46 dB for the claves recording, it results in a SSNR of 24.78 dB for the guitar recording. The results are interpreted on the same three levels as before.

Table 4.5: Segmental signal-to-noise (SSNR) ratios (in dB) for the claves recordings. From top to bottom, rows correspond to GSNR's of 35, 30 and 25 dB.

Noisy SSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
1.46	8.19	8.96	9.47	5.67
-2.21	3.70	4.79	5.25	5.43
-4.24	-0.32	0.51	1.16	3.65

Table 4.6: Segmental signal-to-noise (SSNR) ratios (in dB) for the guitar recordings.

Noisy SSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
24.78	29.28	29.61	29.82	23.53
20.70	26.10	26.56	26.85	22.43
15.93	21.90	22.40	22.65	20.06

Table 4.7: Segmental signal-to-noise (SSNR) ratios (in dB) for the singing recordings.

Noisy SSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
19.36	22.89	23.15	23.36	21.56
15.67	19.91	20.27	20.28	19.65
11.87	16.35	16.85	17.12	17.72

Table 4.8: Segmental signal-to-noise (SSNR) ratios (in dB) for the glockenspiel recordings.

Noisy SSNR	Proposed method $\alpha = 10$	$\alpha = 17.5$	$\alpha = 25$	SNG
14.14	20.70	21.40	21.73	13.94
9.67	16.90	17.64	18.18	13.71
5.20	12.78	13.64	14.22	13.32

Dependence on the initial GSNR

The relation between initial noise level and SSNR after processing is more ambiguous compared to the case for the GSNR. For the claves recording, the gain in SSNR is largest for the lowest noise level (e.g. from 1.46 to 8.19 dB using α-10 AEF) and smallest for the highest noise level (-4.24 to -0.32 dB) when AEF is used, but the reverse relation holds for the SNG method. For the other instruments, the SSNR gain does increase with increasing noise intensity similar to what was observed for the GSNR. Aside from this discrepancy, the results of the SSNR calculation are comparable to those obtained for the GSNR. The magnitude of the SSNR gain is again larger for the claves and glockenspiel recordings than for the other two instruments, and the gains of in both measures are of similar magnitude under the same circumstances.

Influence of the tuning parameter

Similar to the previously observed behaviour, higher values of α result in larger improvements in the segmental signal-to-noise ratio, and the difference between the lower two settings (α = 10-17.5) is larger than the difference between the higher two settings (α = 17.5-25).

Comparison with spectral noise gating

In terms of the segmental signal-to-noise ratio, the comparison between the proposed method and SNG is not as one-sided as was the case for the GSNR. Whereas the proposed method outperforms spectral noise gating at 35 dB additive noise at all settings and for all instruments, SNG performs better at the higher noise levels for the claves and singing recordings, and comparable for the glockenspiel recording. Judging by the trends observed for both methods with different SNR's, it appears likely that SNG would outperform the proposed method at higher noise levels (e.g. 20 dB GSNR), whereas AEF performs better at lower noise levels.

4.3 Subjective results

The field of musical enhancement is a relatively small discipline within the field of audio signal processing, and as such, it does not have standardised testing procedures and libraries. For this reason, the subjective listening test in this work is designed to comply with the ITU-T P.835 (2003) testing methodology, which is intended for speech processing algorithms, where possible.

4.3.1 Test design

The 64 audio fragments of the musical recording library discussed in section 4.1 were divided into 20 3-piece trials and one 4-piece trial. Within each trial, listeners are required to give their opinion in three different categories:

- The **M**-score, a measure of musical signal distortion;
- The **B**-score, a measure of background intrusiveness;
- The **O**-score, the rating of the overall quality of the fragment.

The motivation for this three-step rating approach is that some some listeners may naturally

Table 4.9: Meaning of opinion scores in subjective tests

	M-score	B-score	O-score
5	Not distorted	Not noticeable	Excellent
4	Slightly distorted	Slightly noticeable	Good
3	Somewhat distorted	Noticeable, but not intrusive	Fair
2	Fairly distorted	Somewhat intrusive	Poor
1	Very distorted	Very intrusive	Bad

pay more attention to either the background or the musical signal in forming their opinion of the overall quality. Requiring them to explicitly formulate their opinion in terms of each of these three measures takes this variability out of the results.
Listeners were instructed to formulate their opinion using a scale from 1 to 5, 5 being the best score, using table 4.9. For the three files in each trial, listeners have to rate one on the basis of the M-measure, another on the B-score, and the last one on the overall quality. In the last, four-segment trial, one of the categories is repeated. To ensure that each audio fragment receives a rating in each of the three categories, the group of listeners is divided in three teams: the members of different teams rate the same audio file by a different category.

As the test takes approximately 25-30 minutes in its entirety, listeners were advised to take a break when they deemed it necessary. Although the use of high-quality headphones was also strongly encouraged, not all listeners had these available to them, and the listening environment is not controlled enough to fully comply with ITU-T P.835 standards (see also section 4.3.3). A copy of the test instructions handed out to listeners is included in appendix D.

4.3.2 Discussion of results

A total of 24 listeners participated in the subjective test. Their scores were averaged per audio file to yield the three scores (M, B and O) for all 64 audio fragments. The complete results of all listeners for all audio files are available in appendix C. In figure 4.3, the scores of audio fragments are averaged and summarised per added noise intensity. From left to right, it shows the average ratings of the unprocessed, noisy reference recording, the proposed AEF method for the three different α settings, the spectral noise gating method, as well as the noise-free original recording. In evaluating these results, comparisons are made between processed and unprocessed recordings, between the different settings for the noise reduction parameter α, between the proposed method and spectral noise gating, as well as between different noise levels.

Processing versus no processing

Amongst other things, figure 4.3 shows the adverse effects of noise on the perceived audio quality. A comparison of the unprocessed recording to the noise-free reference reveals that the B- and O-scores decrease by up to 3 points upon addition of white noise at various intensities. The M-score, although not affected as markedly as the other two categories, also decreases by approximately 0.5-0.75 points at all noise levels. This finding is compatible with the signal masking property of broadband noise mentioned in section 4.1.

Moreover, figure 4.3 shows clearly the benefits of noise reduction algorithms. Almost all processed audio fragments compare favourably to their unprocessed counterparts in all categories and at all added noise levels. In particular, the B- and O-scores are generally raised by a full point after processing. The M-score, on the other hand, shows much less improvement and may

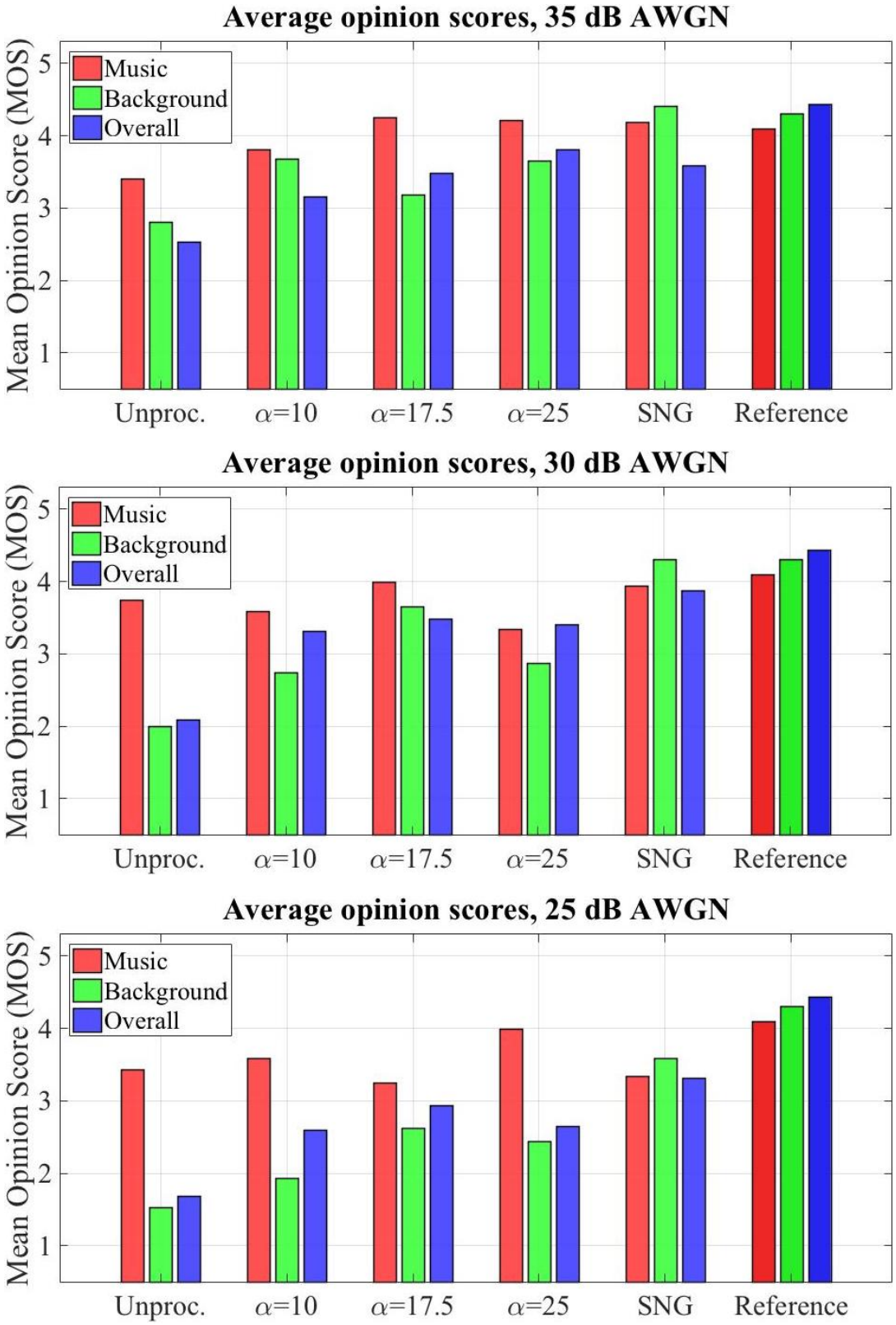


Figure 4.3: Average opinion scores per method and per added noise intensity. The results were obtained by averaging the scores given by the participants, and subsequently taking the mean of all four instrument recordings processed in that particular way. In addition to the four processing approaches, the ratings of the unprocessed noisy reference (far left, denoted 'Unproc.) and noise-free reference (far right, denoted 'Reference') are shown for comparison.

even decrease: these are the cases of α-17.5 AEF and SNG at 25 dB noise, which score 0.2 and 0.1 points lower on average than the unprocessed noisy recording, and the cases of α-10 AEF and α-25 AEF at 30 dB noise, whose scores are 0.15 and 0.4 points lower, respectively. These results indicate that the benefits of these noise reduction algorithms in terms of overall audio quality are mostly in reducing the intrusiveness of the background noise, rather than (perceptually) enhancing the musical signal.

Influence of the tuning parameter

From theory, a higher value of the tuning parameter α leads to more background noise reduction at the cost of increased noise distortion. Since no explicit distinction is made between these two processes in the listening test, their effects are combined in the opinion scores, particularly the B-category. Although the results are ambiguous and the differences generally small, the results obtained using a value of α=10 appear to be the worst in general out of the three options. The O-ratings at all noise levels using this value are the lowest of the three options considered, and the background rating is the lowest at both 30 and 25 dB by a relatively considerable margin. Instead, the results appear to suggest that the value of α should be approximately 17.5. At the two highest noise intensities, this setting yields the best ratings in both the background and overall categories, whereas at the lowest noise level, it scores in the middle between the other two options in terms of the overall rating. However, as the α=25 setting scores better at 35 dB and somewhat comparably at 25 dB, the optimal value may lie somewhere in between. Alternatively, more testing may be required to reveal which is the superior choice.

Comparison with spectral noise gating

Contrary to in the objective results, the spectral noise gating method compares favourably with the proposed method in most aspects of the subjective test. In particular, SNG is more effective in reducing the background intrusiveness. This is a consequence of the flexibility of the method in this respect: whereas SNG allows for a trade-off between noise reduction and signal distortion, the trade-off in the proposed AEF approach is between noise reduction and noise distortion. This statement is supported by the M-ratings, which are slightly lower for SNG than for the best AEF result at all noise levels. In terms of the overall score, SNG is the preferred method, particularly at higher noise levels. It should be noted, however, that the proposed method is fully automated and requires the tuning of only one parameter, whereas the SNG algorithm considered involves setting three parameters as well as manual selection of a noise-only interval.

4.3.3 Comments on the subjective results

Although the outcome of the subjective test makes intuitive sense, a number of matters should be kept in mind in its interpretation. First and foremost, the fact that each fragment is rated by 24 participants, or 8 individuals per category per fragment, means that the results should be seen as indicative, not as statistically significant. Upon closer inspection of the results in figure 4.3, some inconsistencies can be detected that likely stem from the small number of listeners. For instance, in the case of 35 dB noise, some of the processed fragments are rated better on average than the noise-free reference recording, which is highly improbable. Additionally, one would expect the ratings to gradually decrease with increasing noise level; however, the B-score for α-17.5 AEF at 35 dB is nearly half a point lower than the corresponding score at 30 dB. These inconsistencies are likely to gradually disappear as the number of test participants increases.

A second point should be made with regard to the listening conditions. As mentioned briefly in section 4.3.1, logistics and time constraints prohibited the construction of a controlled environment with equal listening conditions for all test participants. The listening conditions include the direct environment (i.e. the degree of soundproofing in the room and its acoustics), the volume at which the audio is played, as well as the device (headphones, earphones, speakers) over which it is played. Although the majority of these characteristics are unlikely to affect the relative scores given by a single participant, and therefore do not compromise the reliability of the results presented in figure 4.3, it does unequivocally result in different average scores per participant. Indeed, the tester with the lowest average score rated the files 2.6 out of 5 on average, whereas the tester with the highest average score rated them just over 4 points on average. Therefore, the absolute values of the presented subjective results are less meaningful, and the results should only be interpreted in a comparative sense.

Chapter 5

Conclusions and future work

In the previous chapters, the theory, practical implementation and results of an algorithm for denoising music were discussed. In summary, the approach taken is to transform the time-domain signal to the time-frequency space, and to remove noise from the resulting 2-D image. This is achieved by using an adapted version of the eigenimage filtering technique, which is also used in the enhancement of seismic sections. After filtering, the image is converted back to a waveform, which yields an enhanced version of the input signal. Although the proposed method does not unequivocally outperform the noise suppression method called spectral noise gating, with which it was compared directly, it does significantly improve the recording quality, and has the additional advantages of being fully automated and requiring only a single parameter to be tuned. For the intended application to an online platform for music creation, whose community comprises hundreds of thousands of amateur musicians, these are essential features that can outweigh the importance of enhanced noise reduction. The sample study conducted at the start of this work indicated that approximately 25% of user projects contain noticeable random noise. Particularly these recordings should see great improvements in quality upon processing. Therefore, the developed algorithm should hold some company value.

For the purpose of further improving the average quality of the user projects, the installment or development of a follow-up filter is worth considering, specifically a coherent noise filter. Presently, coherent noise will be perceptually enhanced upon filtering with the proposed method. A filter designed to remove coherent noise would thus enhance not only the 9.5% of recordings in which this was found to be the major quality issue, but many of the other recordings in which it is contained. An additional benefit to this second filter would be that the proposed method is more reliable if coherent noise is removed from recordings first, as its presence complicates the step in which the noise threshold is determined. If this coherent noise filter were to be included, it would thus be advised to install it before the developed denoising method in the filter sequence.

Although the results with adapted eigenimage filtering are generally promising, they are also ambiguous. Therefore, it would be preferable to put the algorithm to further tests. The objective results, which have been quantified with two relatively simple measures, could be complemented with another distance measure that correlates well with overall quality or signal distortion, such as the log-likelihood ratio or frequency-weighted SSNR. Moreover, the subjective results would benefit from having more opinion scores given by additional testers, as this would likely reduce the ambiguity of the outcome. With these efforts, the optimal value for the tuning parameter α

(presently presumed to lie between 17.5 and 25) could be decided on in a more conclusive manner. Furthermore, the performance of the algorithm in removing non-white noise, for instance pink noise, was not included in the tests. This choice was made to ensure that the length of the subjective test would not serve as a barrier for the listeners, all of which participated on a voluntary basis. Nevertheless, the algorithm is theoretically capable of suppressing noise types other than white. If a follow-up test is to be conducted, the addition of these noise types is worth considering.

General applicability was an important requirement that influenced the design of the algorithm. Nevertheless, some limitations to its applicability exist as a consequence of the assumptions made on the signal and noise characteristics. For the proposed method, one of the key simplifications is that the noise is wide-sense stationary: this justified the usage of a single threshold value in defining the signal and noise subspaces throughout the entire recording. Although this assumption appears reasonable for many noise sources (for instance, the noise generated by the electrical circuits in a computer), it is inevitably not valid for all noise sources that users may encounter. A periodic re-evaluation of the noise threshold, which would constitute a relatively simple addition to the algorithm, would likely be effective in addressing this issue and may broaden the method's field of applicability to include recordings with slowly-varying noise.

Another feature of the method, which could be seen as either a strength or as its second limitation, is the absence of an explicit trade-off between noise reduction and signal distortion. Most of the common noise reduction methods do include this trade-off, which enables their users to modify the degree of noise reduction based on how much signal distortion they can tolerate. From the subjective test results, it became clear that most listeners do not notice any significant signal distortion with the proposed method in its current form, whereas the background was still considered intrusive to a varying degree. This observation suggests that by alteration of the algorithm, such that it removes more noise at the cost of increased signal distortion, overall ratings on its performance might be further improved. This could for instance be achieved through the introduction of a second tuning parameter. Inclusion of a factor by which to multiply the noise threshold value α would enable the user to influence the dimensionality of the signal and noise subspaces. This would introduce the classic trade-off between signal distortion and noise reduction, albeit at a loss of simplicity.

To conclude, it can be said that the disciplines of seismic and audio signal processing are similar enough for methods to be applicable to both fields. Some of such methods, such as Wiener filtering and non-negative matrix factorisation, have already been extensively published on in literature. At the same time, it appears that the fields are far enough apart such that not all options have been fully explored yet. The proposed eigenimage filtering method, which to my knowledge has not been applied to the denoising of music before, is a testament to this case. This finding suggests that there is potentially more to gain from the combination of these disciplines in the future.

References

Al-Dossary, S. (2014). Random noise cancellation in seismic data using a 3-d adaptive median filter. *SEG Technical Program Expanded Abstracts*, *Vol. 33*, pp. 2512–2516.

Amendola, A., Gabbriellini, G., Dell'Aversana, P., & Marini, A. J. (2017). Seismic facies analysis through musical attributes. *Geophysical Prospecting*, *Wiley 2017*, pp. 1–10.

Aqrawi, A. A., Barka, D., & Weinzierl, W. (2013). A hybrid and adaptive attribute for noise reduction in post-stack seismic data. *75th European Association of Geoscientists and Engineers Conference and Exhibition 2013 Incorporating SPE EUROPEC 2013: Changing Frontiers*, pp. 5163–5167.

Bassiou, N., Kotropoulos, C., & Pitas, I. (2014). Greek folk music denoising under a symmetric α-stable noise assumption. *Proceedings of the 2014 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, *QSHINE 2014*, pp. 18–23.

Bednar, J. B. (1983). Applications of median filtering to deconvolution, pulse estimation, and statistical editing of seismic data. *Geophysics*, *48(12)*, 1598-1610.

Benesty, J., & Chen, J. (2011). *Optimal time-domain noise reduction filters - A theoretical study.* Springer-Verlag, Berlin Heidelberg.

Benesty, J., Chen, J., & Habets, E. A. (2011). *Speech enhancement in the STFT domain.* Springer-Verlag, Berlin Heidelberg.

Berouti, M. G., Schwartz, R. M., & Makhoul, J. I. (1979). Enhancement of speech corrupted by acoustic noise. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, *ICASSP 1979(4)*, pp. 208–211.

Boll, S. F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *Vol. ASSP-27(2)*, pp. 113–120.

Breithaupt, C. C. C., Gerkmann, T., & Martin, R. (2008). A novel a priori SNR estimation approach based on selective cepstro-temporal smoothing. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, *ICASSP 2008*, 4897–4900.

Cabras, G., Carniel, R., & Jones, J. P. (2012). Non-negative Matrix Factorization: An application to Erta 'Ale volcano, Ethiopia. *Bollettino di Geofisica Teorica ed Applicata*, *Vol. 53(2)*, pp. 231–242.

Cabras, G., Carniel, R., Jones, J. P., & Takeo, M. (2014). Reducing wind noise in seismic data using non-negative matrix factorisation: an application to Villarrica volcano, Chile. *Geofísica International*, *Vol. 51*, pp. 77–85.

Chan, R. H., Ho, C.-W., & Nikolava, M. (2005). Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*, *Vol. 14(10)*, pp. 1479–1485.

Chen, J., Benesty, J., Huang, Y. A., & Doclo, S. (2006). New insights into the noise reduction Wiener filter. *IEEE Transactions on Audio, Speech and Language Processing*, *Vol. 14(4)*, pp. 1218–1234.

Chen, K., & Sacchi, M. D. (2017). Robust f-x projection filtering for simultaneous random and erratic seismic noise attenuation. *Geophysical Prospecting*, *Vol. 65*, pp. 650–668.

Chen, Y. (2015). Deblending using a space-variant median filter. *Exploration Geophysics*, *Vol. 46(4)*, pp. 332-341.

Cohen, I. (2005). Relaxed statistical model for speech enhancement and a priori SNR estimation. *IEEE Transactions on Speech and Audio Processing*, *Vol. 13(5)*, pp. 870-881.

Dendrinos, M., Bakamidis, S., & Carayannis, G. (1991). Speech enhancement from noise: A regenerative approach. *Speech communication*, *Vol. 10*, pp. 45–57.

Elboth, T., Presterud, I. V., & Hermansen, D. (2009). Time-frequency seismic data de-noising. *Geophysical Prospecting*, *Vol. 58*, pp. 441-453.

Ephraim, Y. (1992). Statistical-model-based speech enhancement systems. *Proceedings of the IEEE*, *Vol. 80(10)*, pp. 1526–1555.

Ephraim, Y., & Malah, D. (1984). Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *ASSP-32(6)*, pp. 1109-1121.

Ephraim, Y., & Malah, D. (1985). Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *ASSP-33(2)*, pp. 443-445.

Ephraim, Y., & Trees, H. L. V. (1995). A signal subspace approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, *Vol. 3(4)*, pp. 251-266.

Fan, N. (2004). Low distortion speech denoising using an adaptive parametric Wiener filter. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, *ICASSP 2004(1)*, pp. 309-312.

Griffin, D. P., & Lim, J. S. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *Vol. ASSP-32(2)*, pp. 236-243.

Haulick, T., Linhard, K., & Schrogmeier, P. (1997). Residual noise suppression using psychoacoustic criteria. *EUROSPEECH 1997*, pp. 1395–1398.

Hermus, K., Wambacq, P., & hamme, H. V. (2007). A review of signal subspace speech enhancement and its application to noise robust speech recognition. *EURASIP Journal on Advances in Signal Processing*, *2007*, Art. No. 45821, p. 195.

Hu, Y., & Loizou, P. C. (2003). A generalized subspace approach for enhancing speech corrupted by colored noise. *IEEE Transactions on Speech and Audio Processing*, *Vol. 11(4)*, pp. 334-341.

Hu, Y., & Loizou, P. C. (2008). Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on Audio, Speech and Language Processing*, *Vol. 16(1)*, pp. 229–238.

Huang, C.-C., Liang, S.-F., Young, M.-S., & Shaw, F.-Z. (2009). A novel application of the S-transform in removing powerline interference from biomedical signals. *Physiological Measurement*, *Vol. 30(1)*, pp. 13-27.

Inoue, T., Saruwatari, H., Shikano, K., & Kondo, K. (2011). Theoretical analysis of musical noise in Wiener filtering family via higher-order statistics. *IEEE International Conference on Acoustics, Speech and Signal Processing*, *ICASSP 08-12*, pp. 2870–2874.

ITU-T P.835. (2003). Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm. *ITU-T Recommendation P.835*.

Jones, I. F., & Levy, S. (1987). Signal-to-noise ratio enhancement in multichannel seismic data via the Karhunen-Loéwe transform. *Geophysical Prospecting*, *Vol. 35*, pp. 12-32.

Kragh, E., & Peardon, L. (1995). Ground roll and polarization. *First Break*, *Vol. 13(9)*, pp. 369–378.

Liu, Y., Liu, C., & Wang, D. (2009). A 1-D time-varying median filter for seismic random, spike-like noise elimination. *Geophysics*, *Vol. 74(1)*, pp. V17–V24.

Lorber, M., & Hoeldrich, R. (1997). A combined approach for broadband noise reduction. *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics*, *WASPAA 1997*(Session 8).

Lyons, R. G. (2011). *Understanding digital signal processing* (Third edition ed.). Prentice Hall, Boston.

Malca, Y., & Wulich, D. (1996). Improved spectral subtraction for speech enhancement. *Signal Processing VIII, Theories and Applications. Proceedings of EUSIPCO-96*, *Vol. 2*, pp. 975–978.

Parchami, M., Zhu, W.-P., Champagne, B., & Plourde, E. (2016). Recent developments in speech enhancement in the short-time Fourier transform domain. *IEEE Circuits and Systems Magazine*, *Vol. 16*, pp. 45-77.

Parolai, S. (2009). Denoising of seismograms using the S transform. *Bulletin of the Seismological Society of America*, *Vol. 99(1)*, pp. 226-234.

Plapous, C., Marro, C., & Scalart, P. (2006). Improved signal-to-noise ratio estimation for speech enhancement. *IEEE Transactions on Audio, Speech and Language Processing*, *Vol. 14(6)*(2098–2108).

Plourde, E., & Champagne, B. (2008). Auditory-based spectral amplitude estimators for speech enhancement. *IEEE Transactions on Audio, Speech and Language Processing*, *Vol. 16(8)*, 1614–1623.

Preuss, R. D. (1979). A frequency domain noise cancelling preprocessor for narrowband speech communication systems. *International Conference on Acoustics, Speech, and Signal Processing*, *ICASSP 1979(4)*, 212–215.

Socco, L. V., Foti, S., & Boiero, D. (2010). Surface-wave analysis for building near-surface velocity models – established approaches and new perspectives. *Geophysics*, *Vol. 75(5)*, pp. 75A83–75A102.

Stockwell, R. G., Mansinha, L., & Lowe, R. P. (1996). Localization of the complex spectrum: The S transform. *IEEE Transactions on Signal Processing*, *Vol. 44(4)*, pp. 998-1001.

Trickett, S. R. (2002). F-x eigenimage noise suppression. *SEG Technical Program Expanded Abstracts*, *Vol. 21(1)*, pp. 2166–2169.

Trickett, S. R. (2003). F-xy eigenimage noise suppression. *Geophysics*, *Vol. 68(2)*, pp. 751–759.

Ulrych, T. J., & Sacchi, M. D. (2005). *Information-based inversion and processing with applications*. Elsevier, Amsterdam.

Upadhyay, N., & Karmakar, A. (2015). Speech enhancement using spectral subtraction-type algorithms. a comparison and simulation study. *Procedia Computer Science*, *Vol. 54*, pp. 574–584.

Vaezi, Y., & Kazemi, N. (2016). Attenuation of swell noise in marine streamer data via nonnegative matrix factorization. *SEG Technical Program Expanded Abstracts*, *Vol. 35*, pp. 4633–4638.

Yoon, B.-J., & Vaidyanathan, P. P. (2004). Wavelet-based denoising by customized thresholding. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, *ICASSP 2004*, pp. II925–II928.

Yousefian, N., Loizou, P. C., & Hansen, J. H. L. (2014). A coherence-based noise reduction algorithm for binaural hearing aids. *Speech Communication*, *Vol. 58*, pp. 101–110.

Zhou, B., & Greenhalgh, S. A. (1994a). Linear and parabolic τ-p transforms revisited. *Geophysics*, *Vol. 59(7)*, pp. 1133–1149.

Zhou, B., & Greenhalgh, S. A. (1994b). Wave-equation extrapolation-based multiple attenuation: 2-D filtering in the f-k domain. *Geophysics*, *Vol. 59(9)*, pp. 1377–1391.

Appendix A:

Noise-free and noisy reference recordings

A.1 Reference recordings: claves

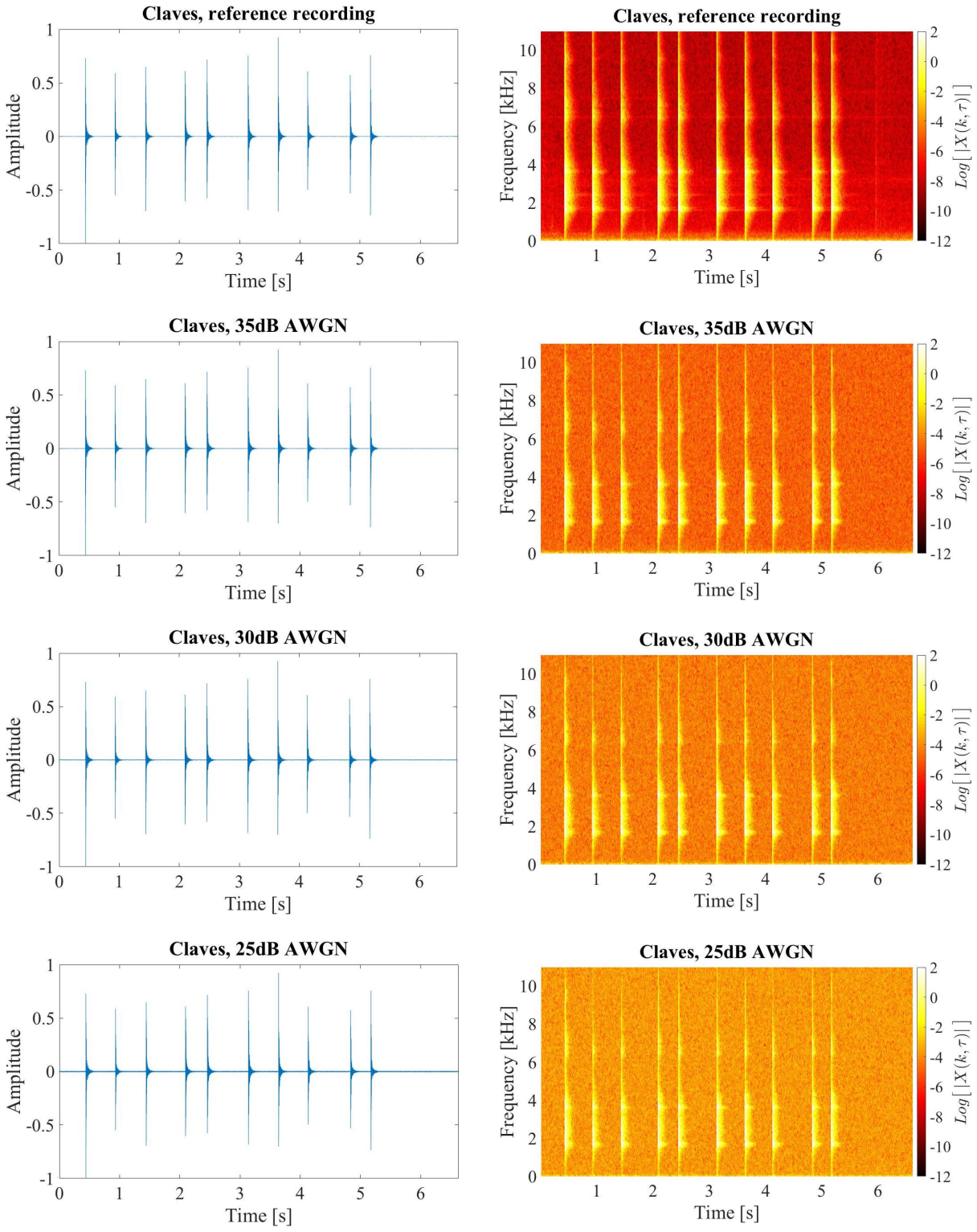


Figure A.1: Waveform and time-frequency views of the claves recording at different noise levels. **Top:** Noise-free reference recording. **Second row:** Noisy reference at a global signal-to-noise ratio of 35 dB. **Third row:** Noisy reference at a global signal-to-noise ratio of 30 dB. **Bottom:** Noisy reference at a global signal-to-noise ratio of 25 dB.

A.2 Reference recordings: guitar

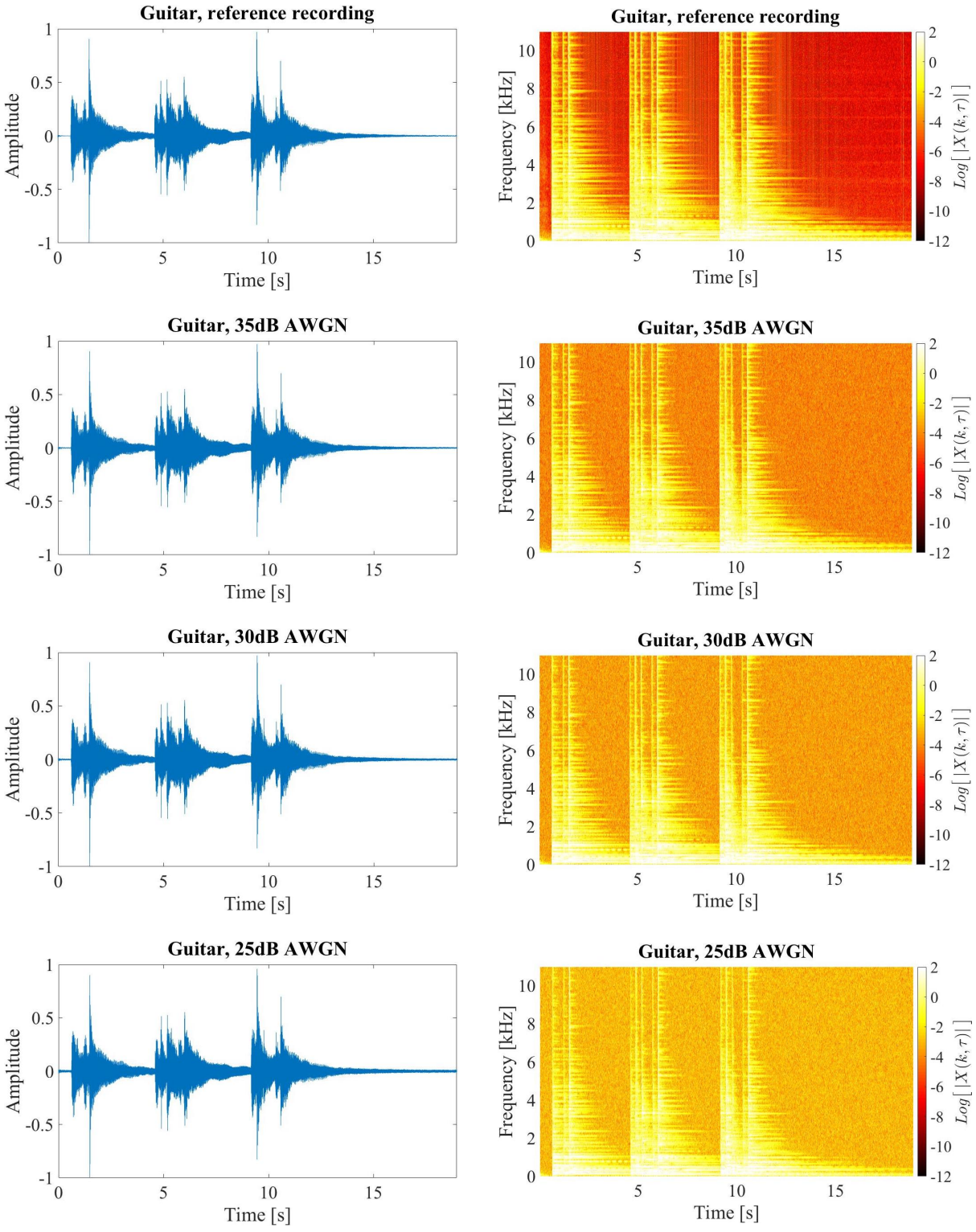


Figure A.2: Waveform and time-frequency views of the guitar recording at different noise levels. **Top:** Noise-free reference recording. **Second row:** Noisy reference at a global signal-to-noise ratio of 35 dB. **Third row:** Noisy reference at a global signal-to-noise ratio of 30 dB. **Bottom:** Noisy reference at a global signal-to-noise ratio of 25 dB.

A.3 Reference recordings: singing

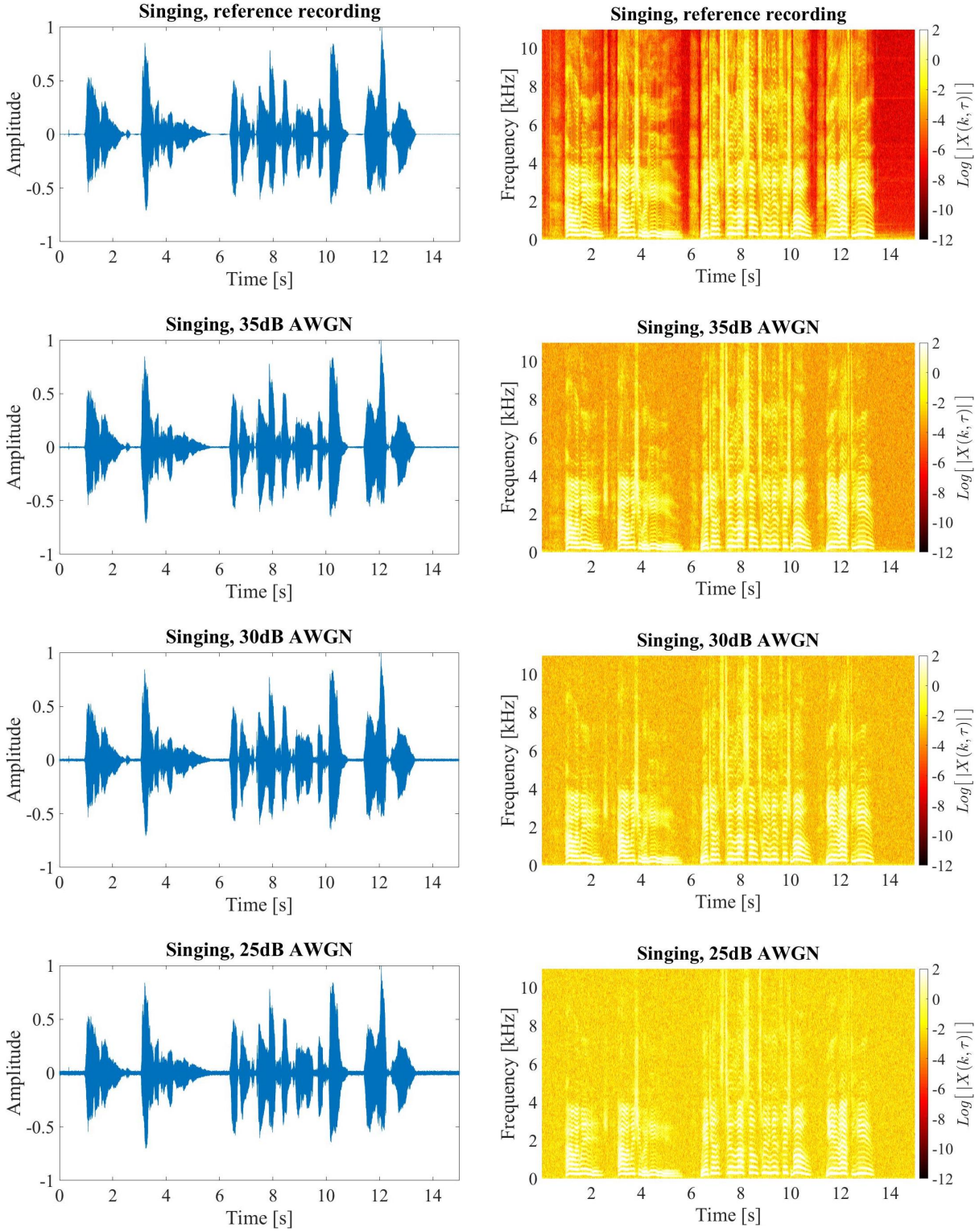


Figure A.3: Waveform and time-frequency views of the singing recording at different noise levels. **Top:** Noise-free reference recording. **Second row:** Noisy reference at a global signal-to-noise ratio of 35 dB. **Third row:** Noisy reference at a global signal-to-noise ratio of 30 dB. **Bottom:** Noisy reference at a global signal-to-noise ratio of 25 dB.

A.4 Reference recordings: glockenspiel

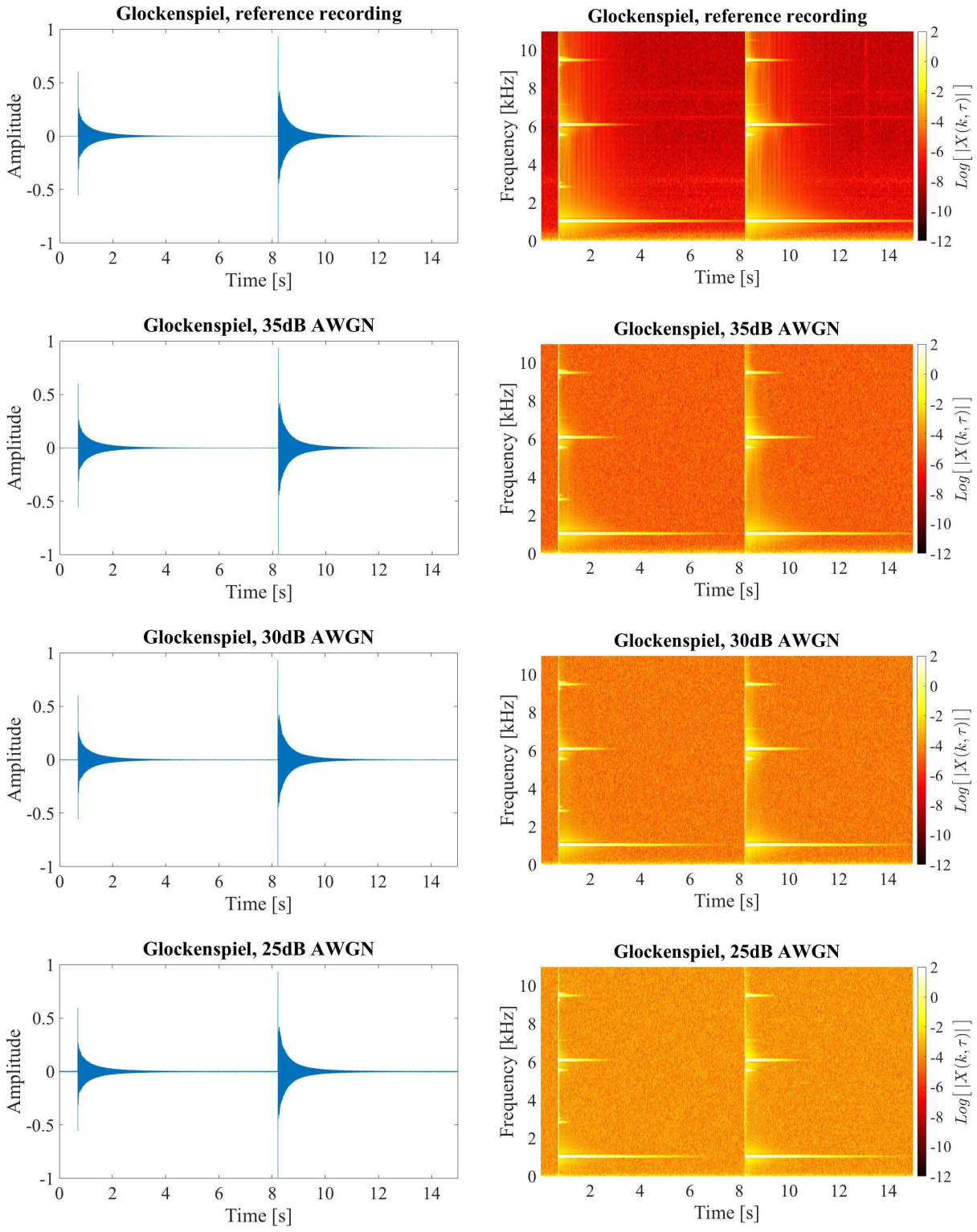
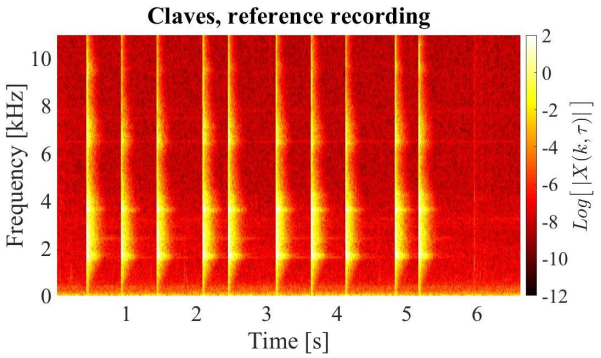


Figure A.4: Waveform and time-frequency views of the glockenspiel recording at different noise levels. **Top:** Noise-free reference recording. **Second row:** Noisy reference at a global signal-to-noise ratio of 35 dB. **Third row:** Noisy reference at a global signal-to-noise ratio of 30 dB. **Bottom:** Noisy reference at a global signal-to-noise ratio of 25 dB.

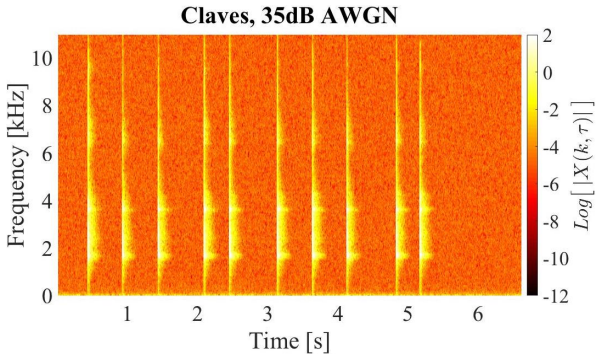
Appendix B:

Objective denoising results

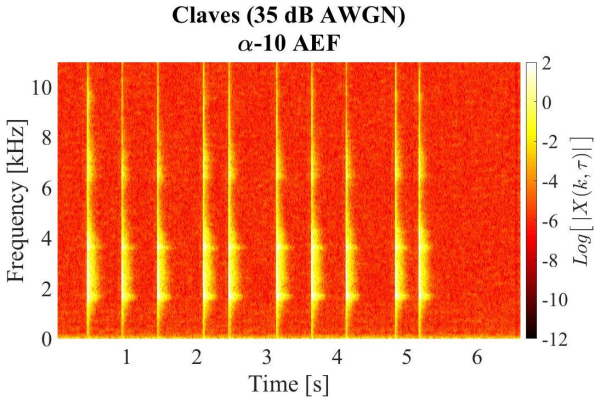
B.1 Claves recording, 35 dB AWGN



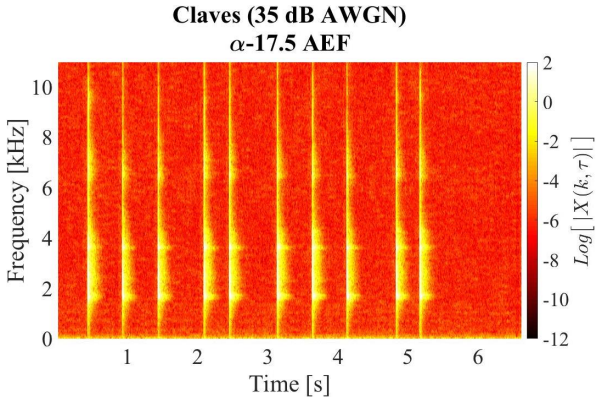
(a) Noise-free reference recording.



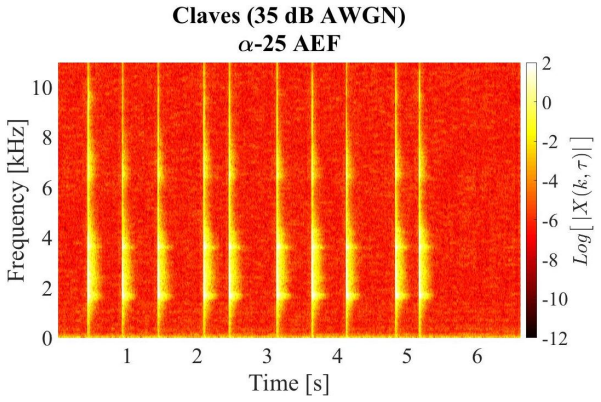
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



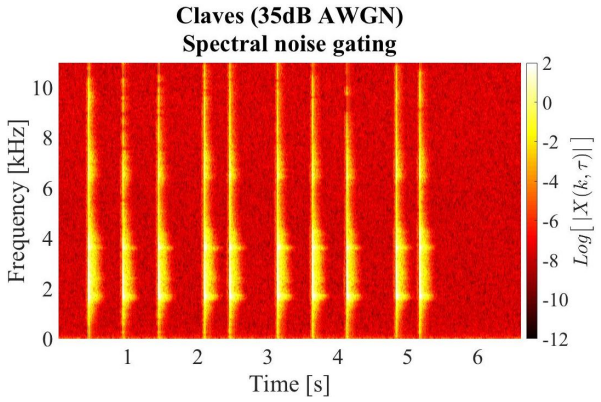
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



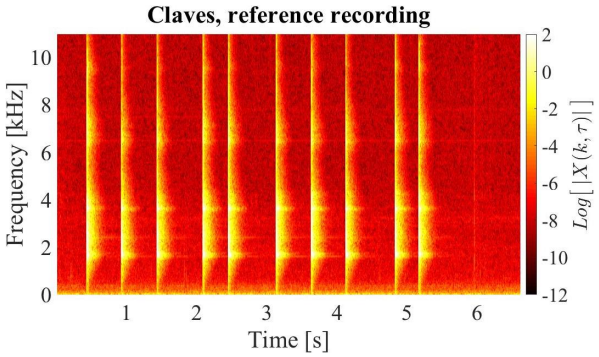
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



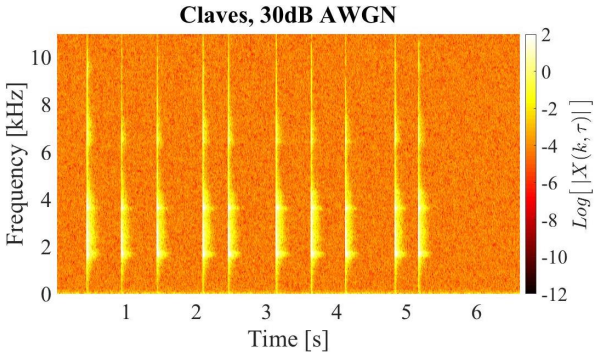
(f) The recording in (b), processed using spectral noise gating.

Figure B.1: Time-frequency representations of the claves recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 35 dB.

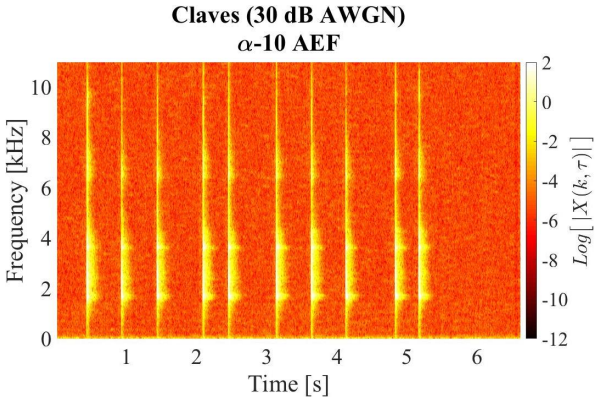
B.2 Claves recording, 30 dB AWGN



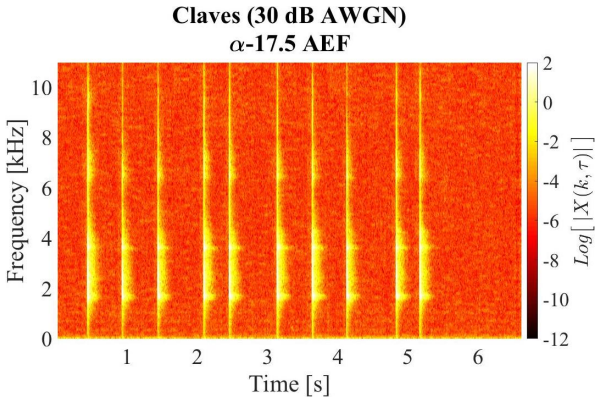
(a) Noise-free reference recording.



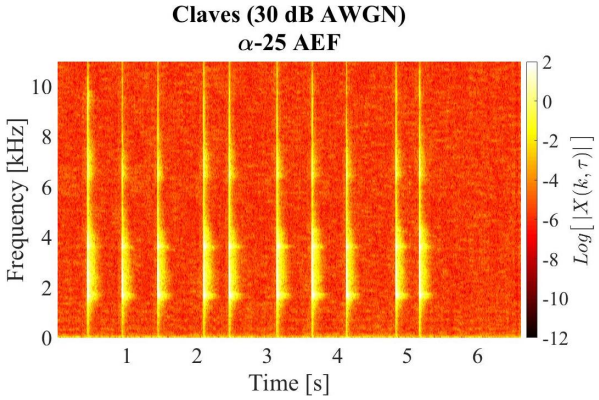
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



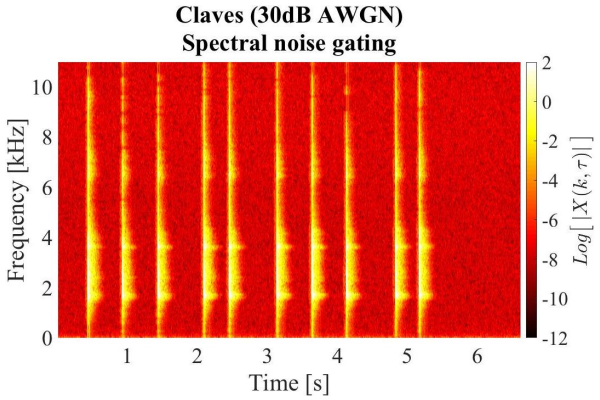
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



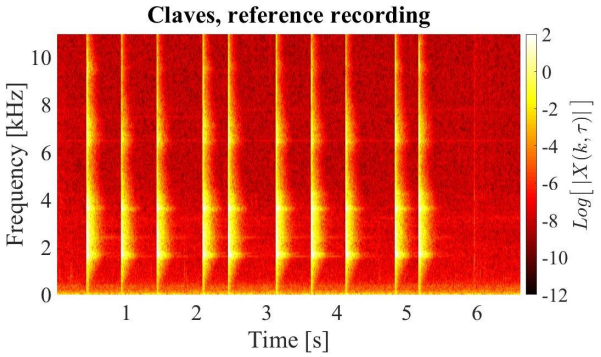
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



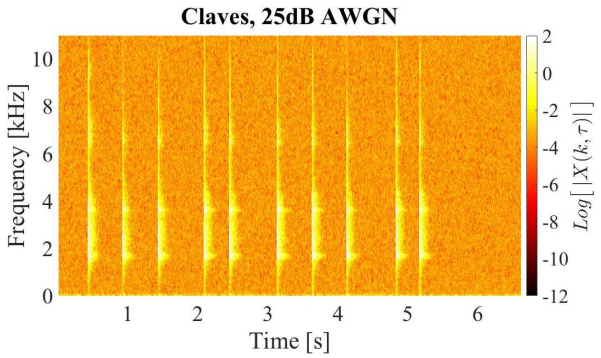
(f) The recording in (b), processed using spectral noise gating.

Figure B.2: Time-frequency representations of the claves recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 30 dB.

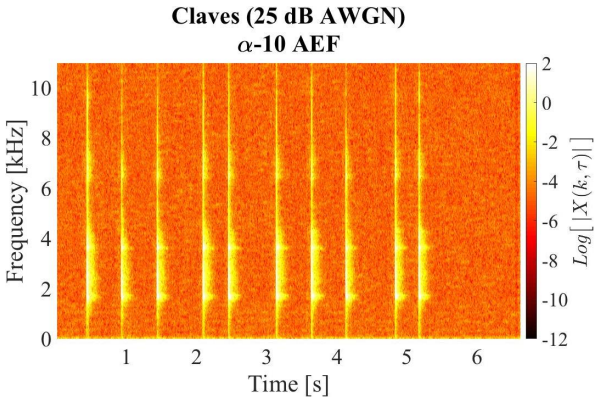
B.3 Claves recording, 25 dB AWGN



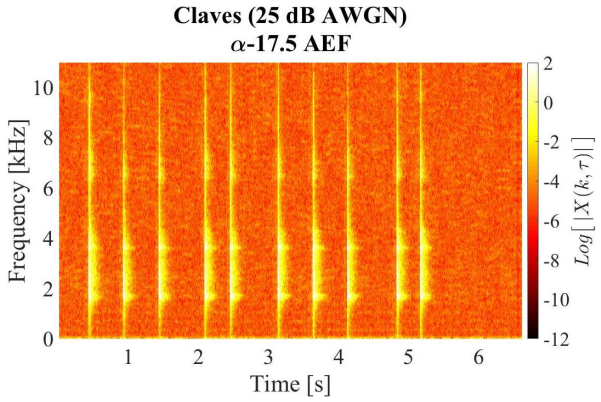
(a) Noise-free reference recording.



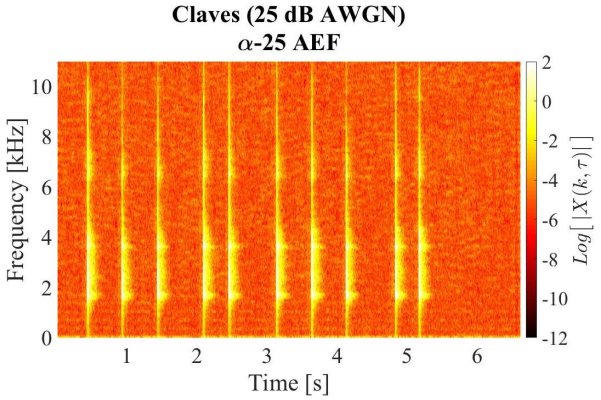
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



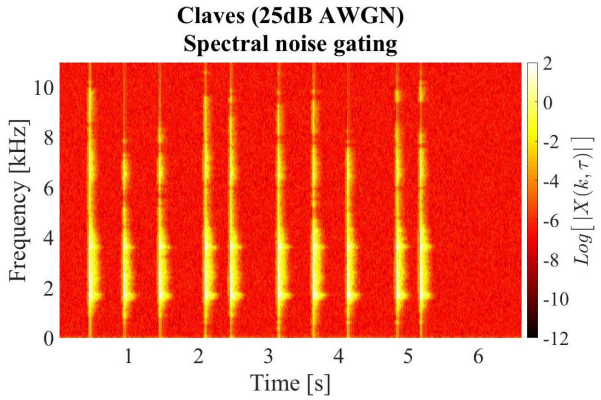
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



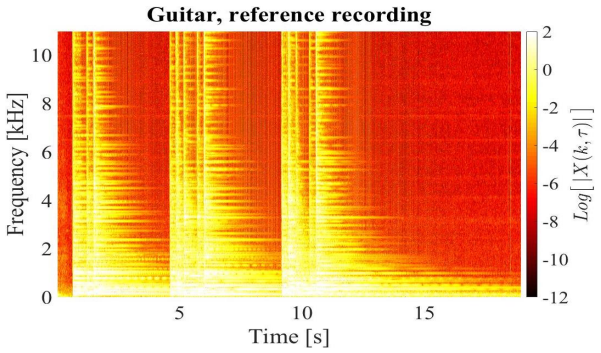
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



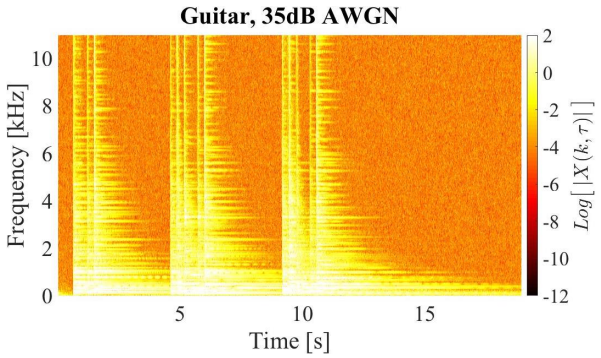
(f) The recording in (b), processed using spectral noise gating.

Figure B.3: Time-frequency representations of the claves recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 25 dB.

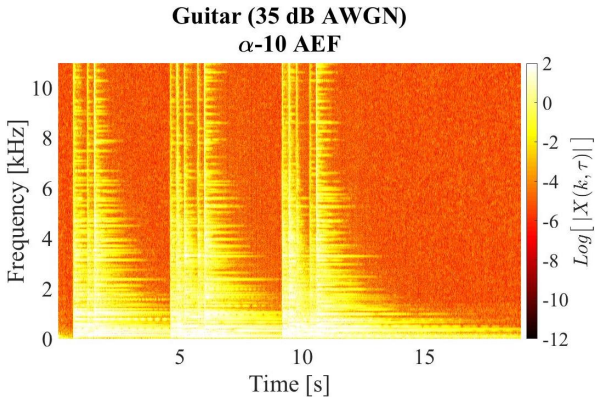
B.4 Guitar recording, 35 dB AWGN



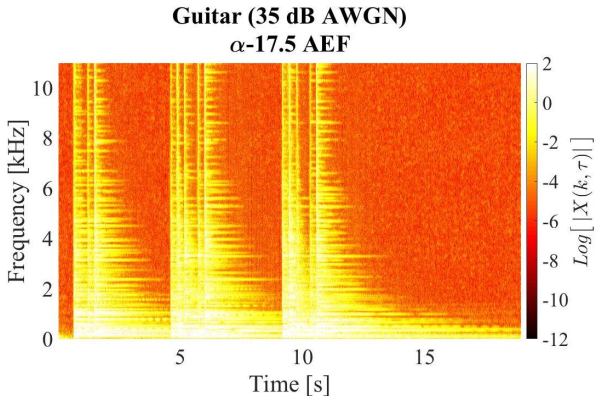
(a) Noise-free reference recording.



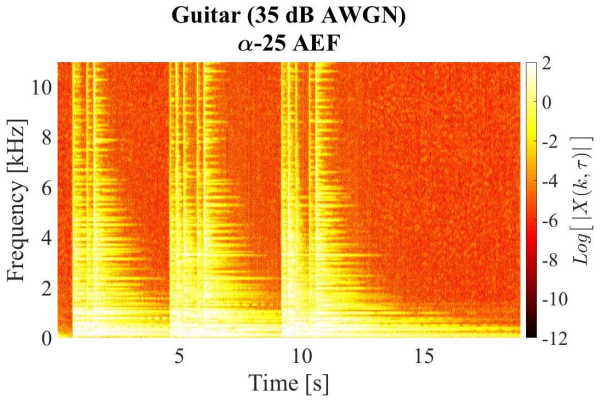
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



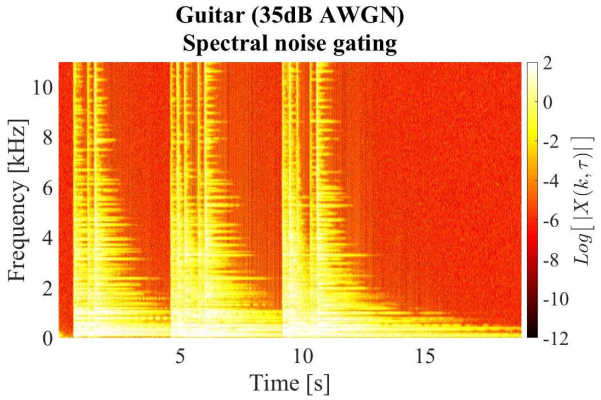
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



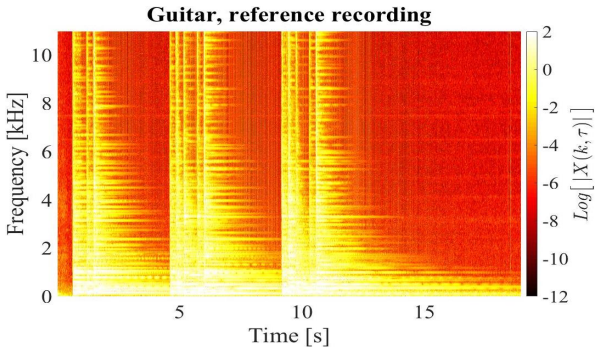
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



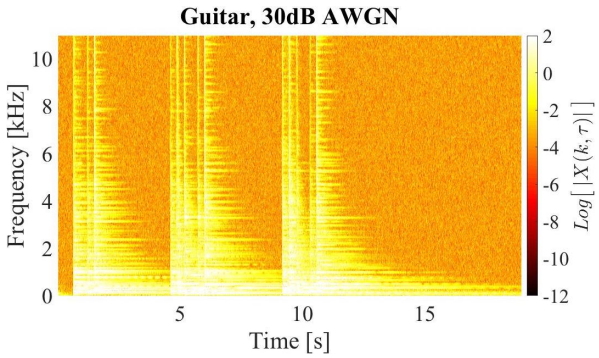
(f) The recording in (b), processed using spectral noise gating.

Figure B.4: Time-frequency representations of the guitar recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 35 dB.

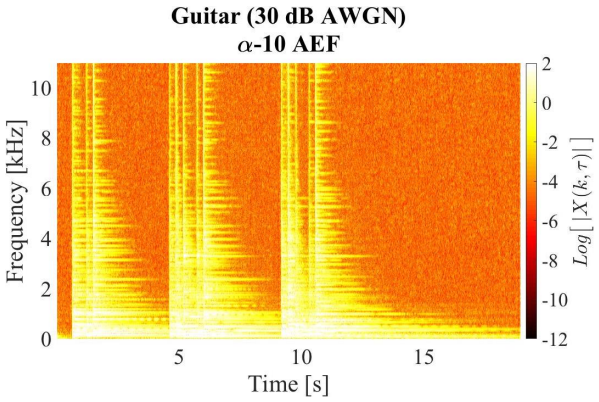
B.5 Guitar recording, 30 dB AWGN



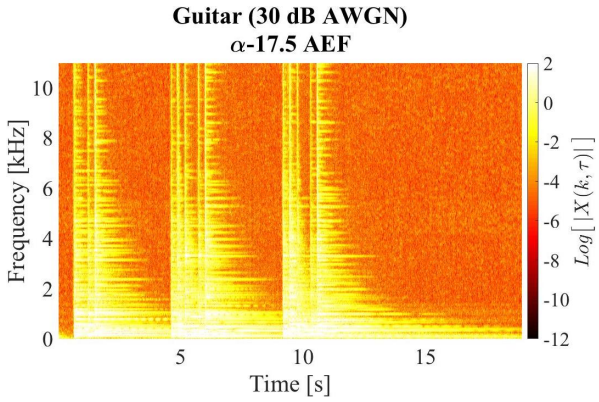
(a) Noise-free reference recording.



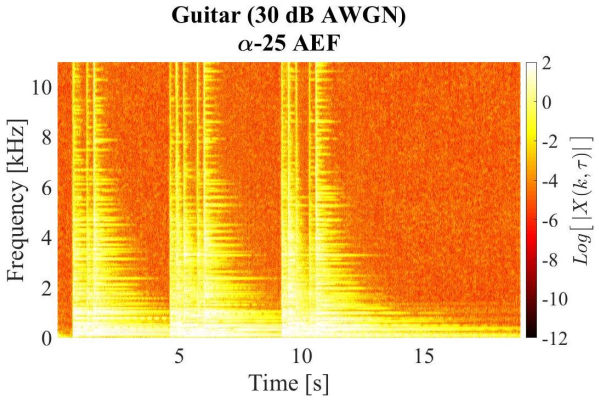
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



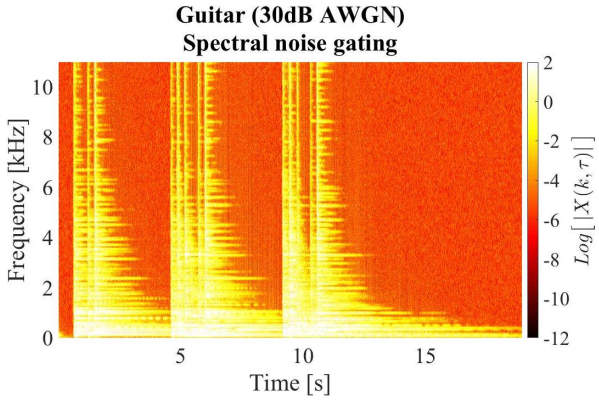
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



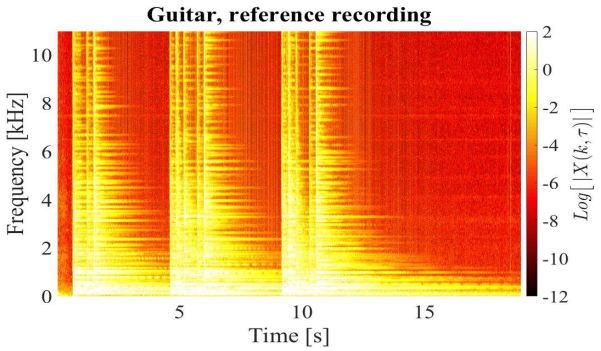
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



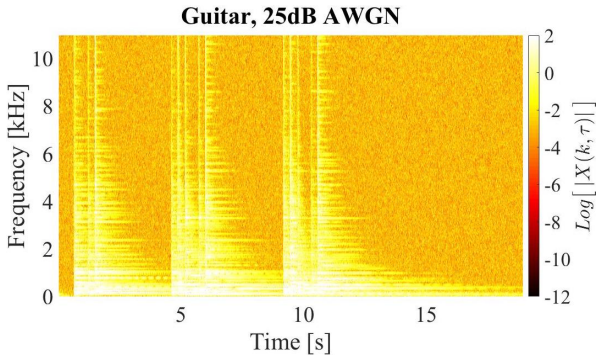
(f) The recording in (b), processed using spectral noise gating.

Figure B.5: Time-frequency representations of the guitar recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 30 dB.

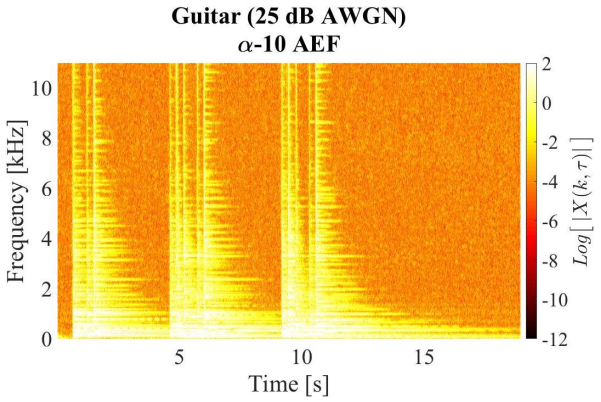
B.6 Guitar recording, 25 dB AWGN



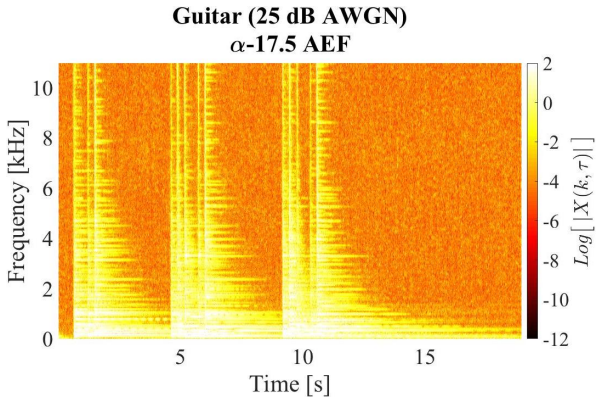
(a) Noise-free reference recording.



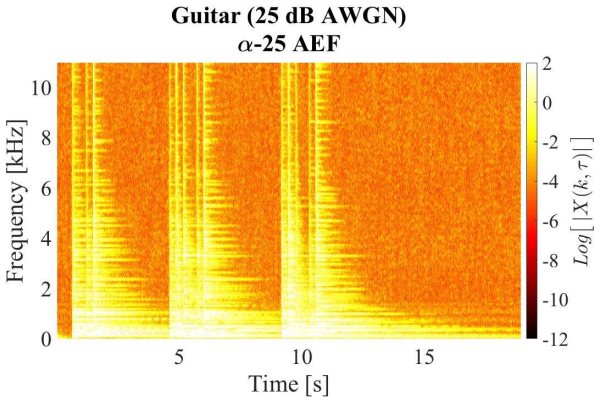
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



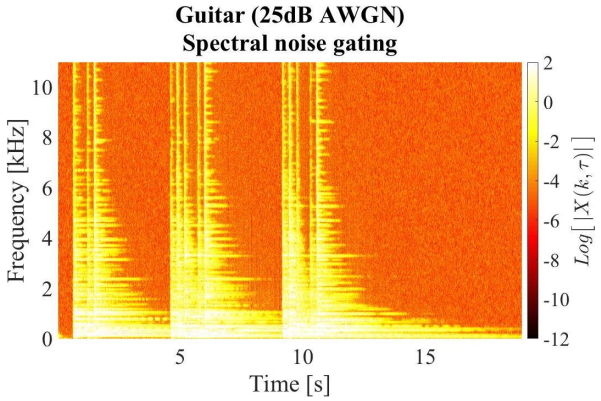
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



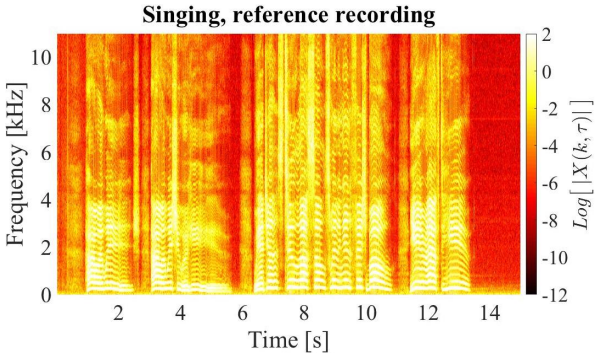
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



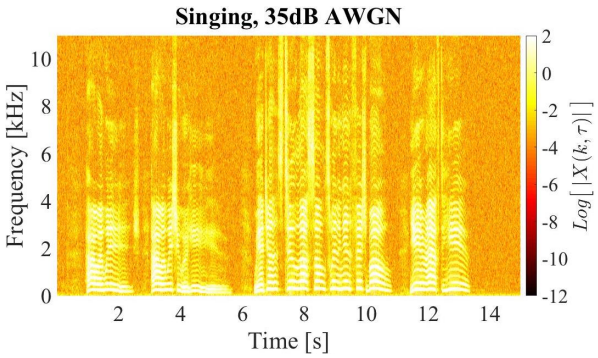
(f) The recording in (b), processed using spectral noise gating.

Figure B.6: Time-frequency representations of the guitar recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 25 dB.

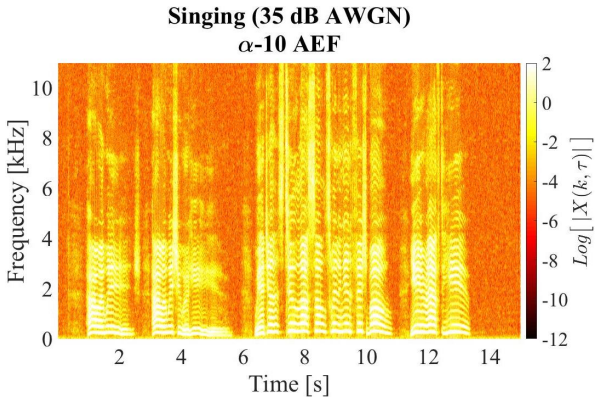
B.7 Singing recording, 35 dB AWGN



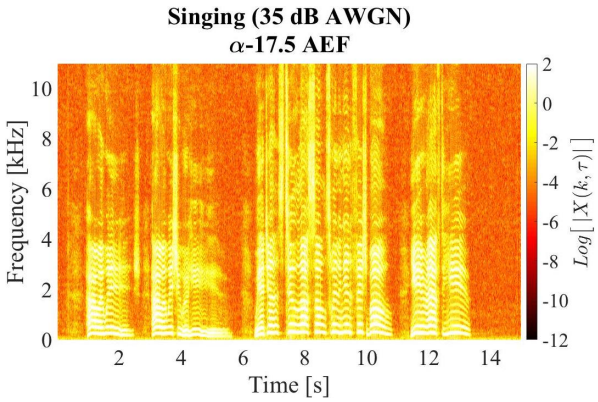
(a) Noise-free reference recording.



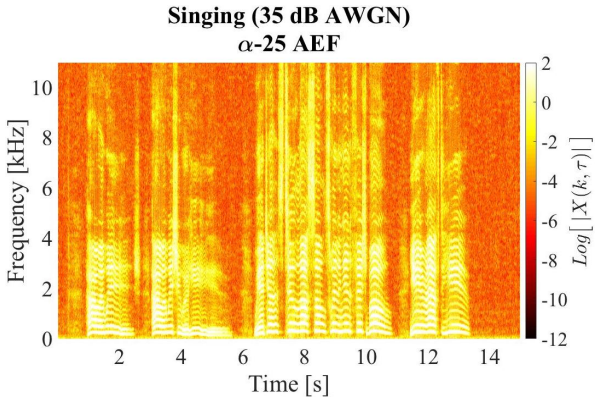
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



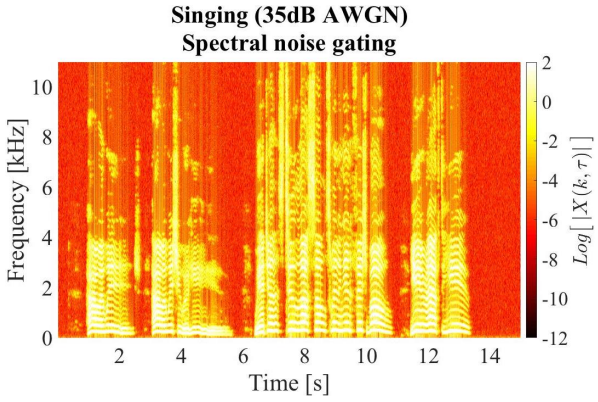
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



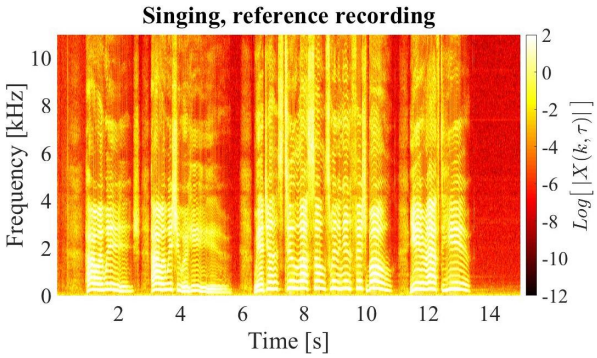
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



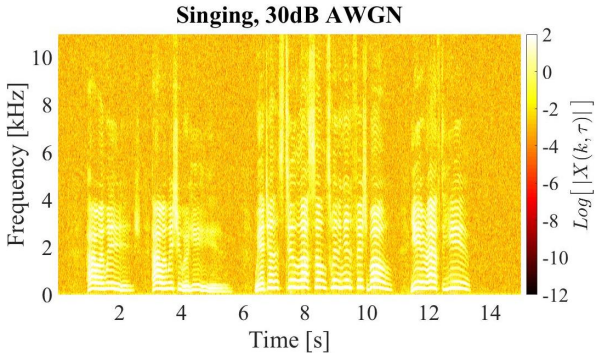
(f) The recording in (b), processed using spectral noise gating.

Figure B.7: Time-frequency representations of the singing recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 35 dB.

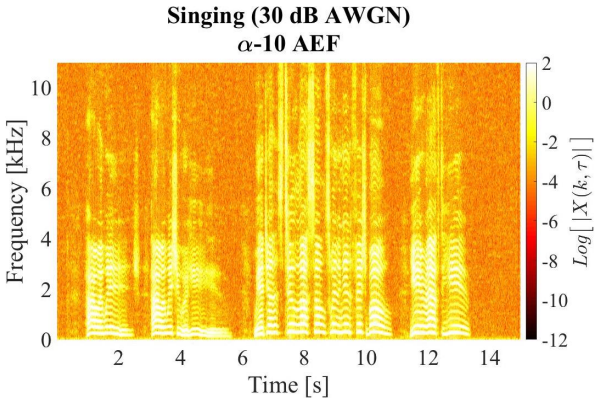
B.8 Singing recording, 30 dB AWGN



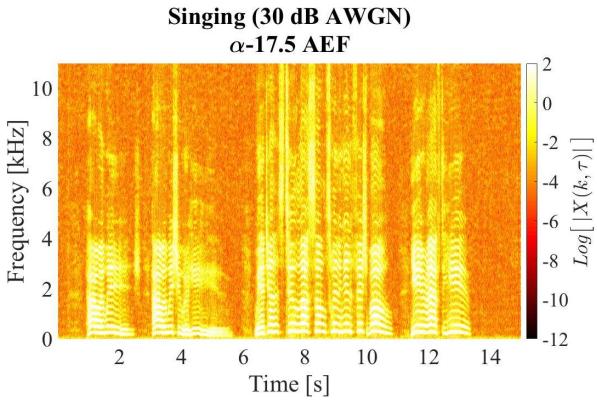
(a) Noise-free reference recording.



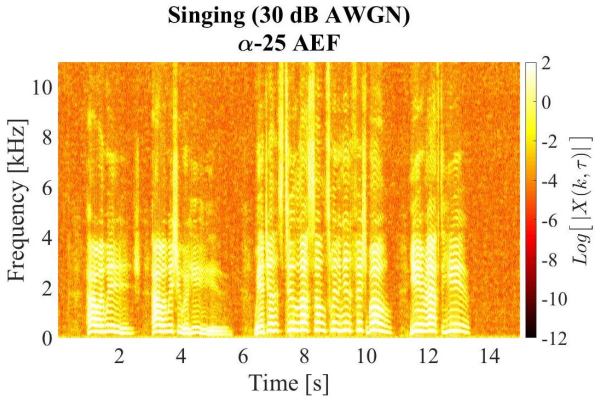
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



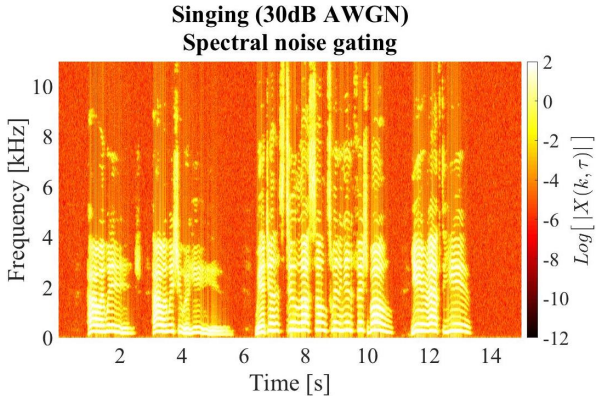
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



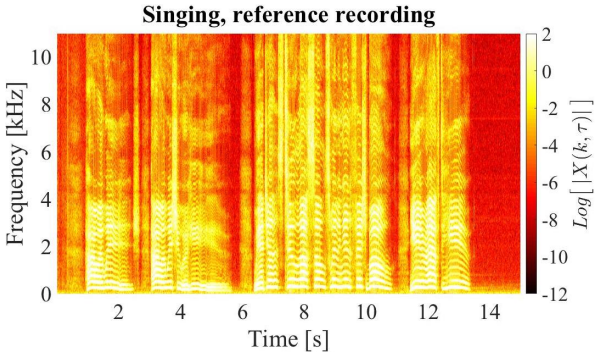
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



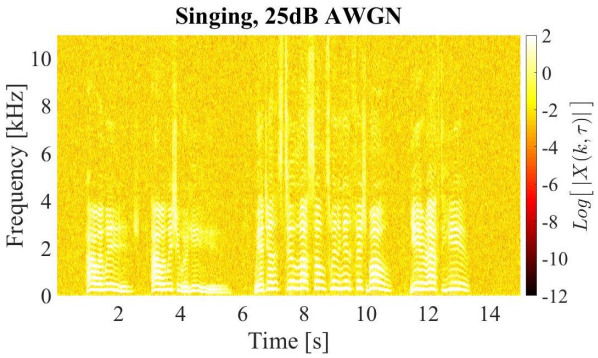
(f) The recording in (b), processed using spectral noise gating.

Figure B.8: Time-frequency representations of the singing recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 30 dB.

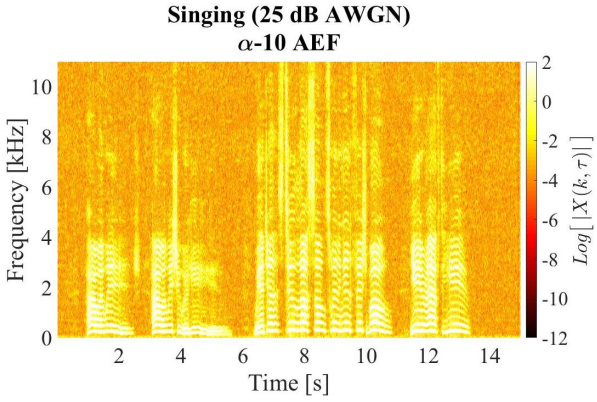
B.9 Singing recording, 25 dB AWGN



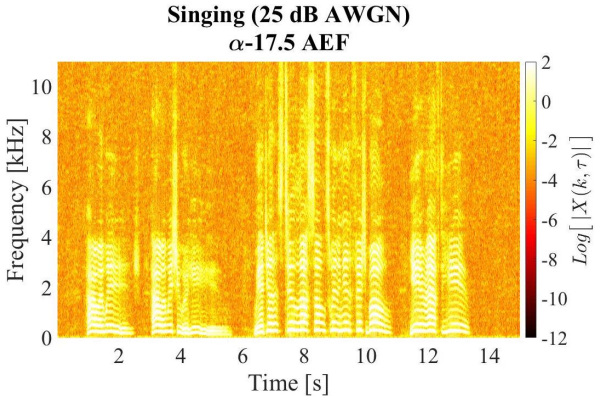
(a) Noise-free reference recording.



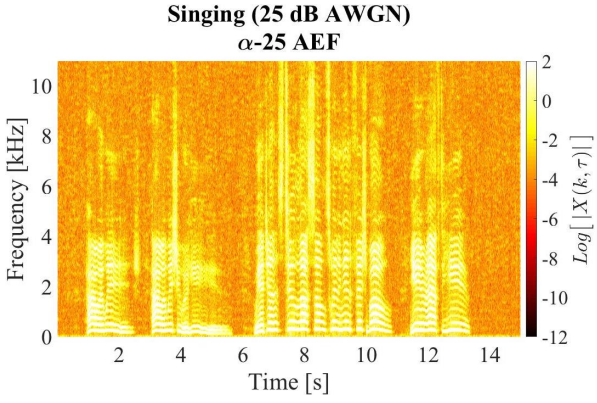
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



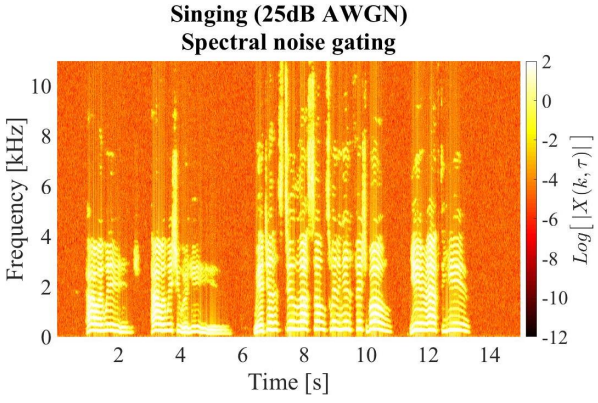
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



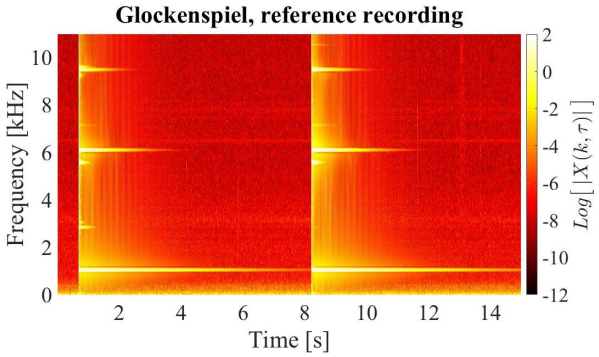
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



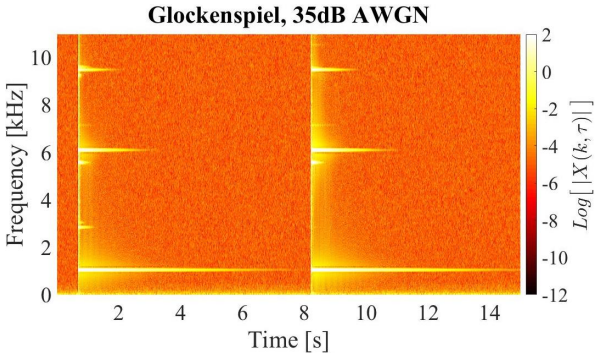
(f) The recording in (b), processed using spectral noise gating.

Figure B.9: Time-frequency representations of the singing recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 25 dB.

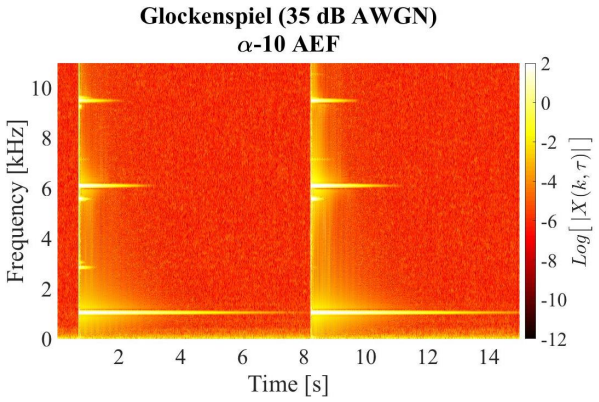
B.10 Glockenspiel recording, 35 dB AWGN



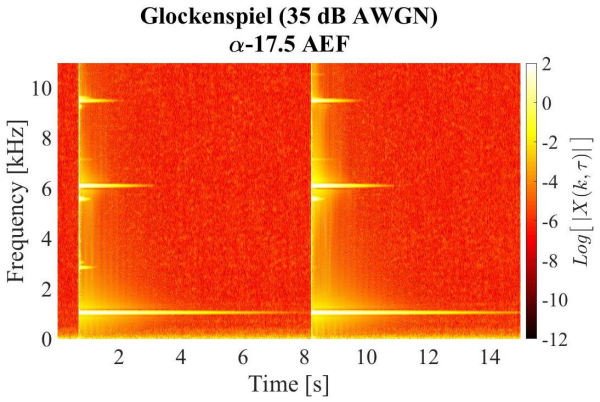
(a) Noise-free reference recording.



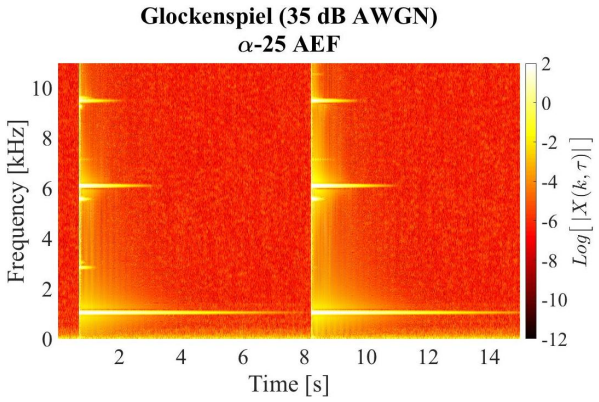
(b) Noisy reference recording with a global signal-to-noise ratio of 35 dB.



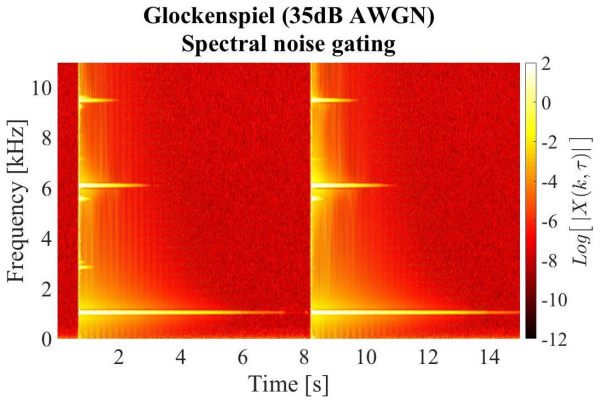
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



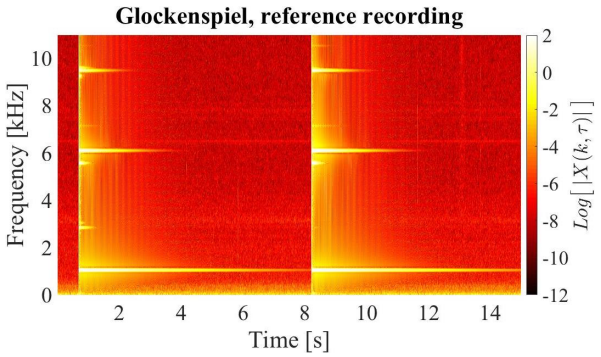
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



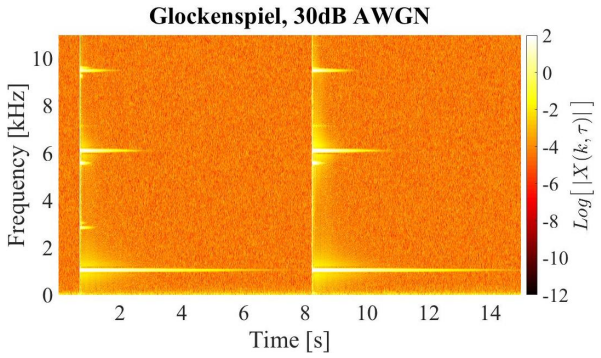
(f) The recording in (b), processed using spectral noise gating.

Figure B.10: Time-frequency representations of the glockenspiel recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 35 dB.

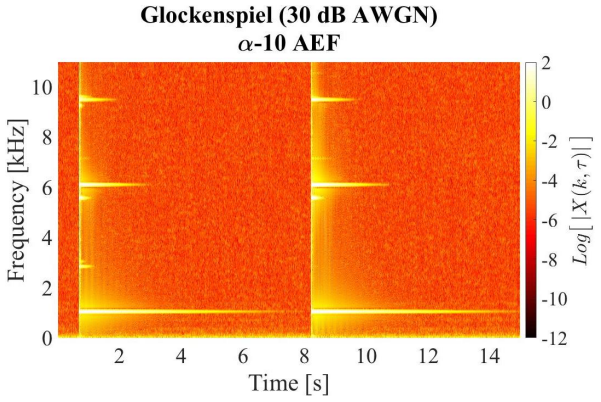
B.11 Glockenspiel recording, 30 dB AWGN



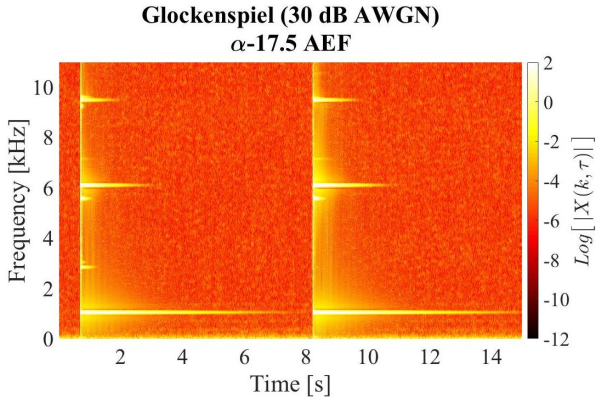
(a) Noise-free reference recording.



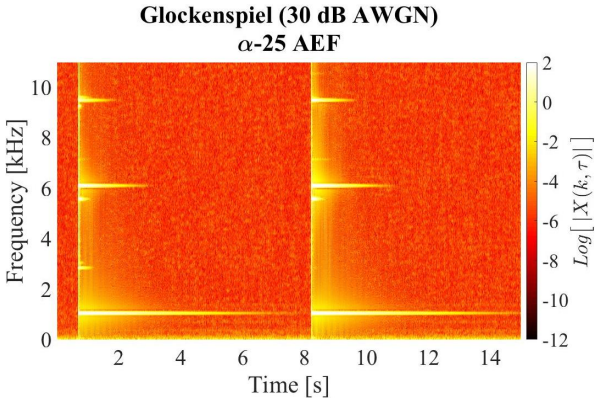
(b) Noisy reference recording with a global signal-to-noise ratio of 30 dB.



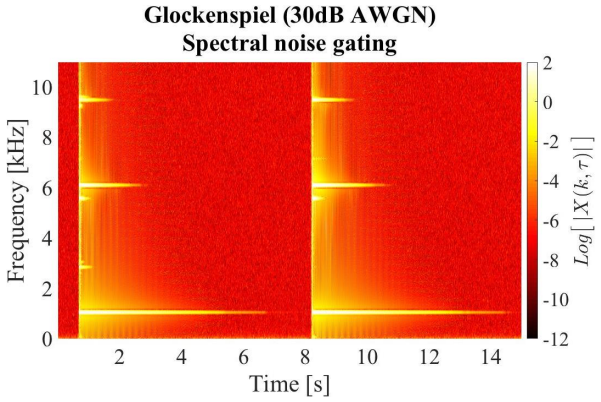
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



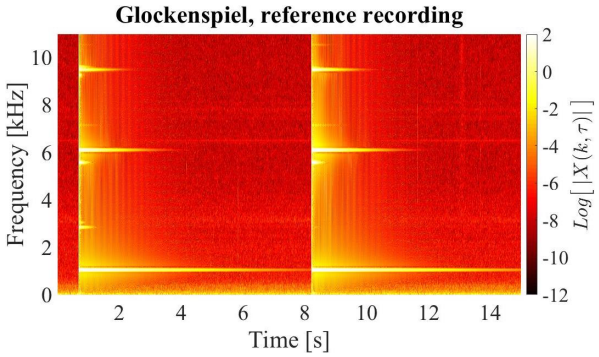
(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



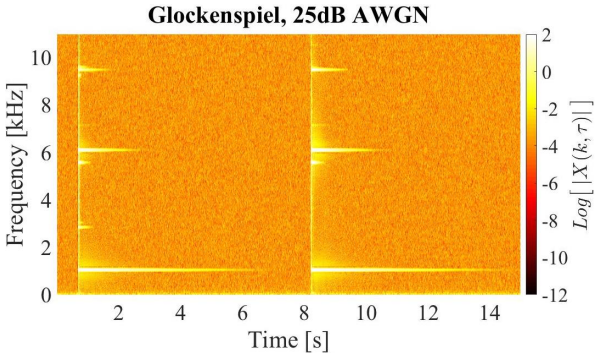
(f) The recording in (b), processed using spectral noise gating.

Figure B.11: Time-frequency representations of the glockenspiel recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 30 dB.

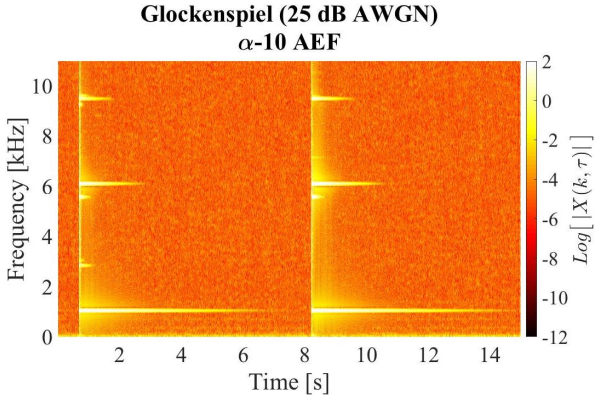
B.12 Glockenspiel recording, 25 dB AWGN



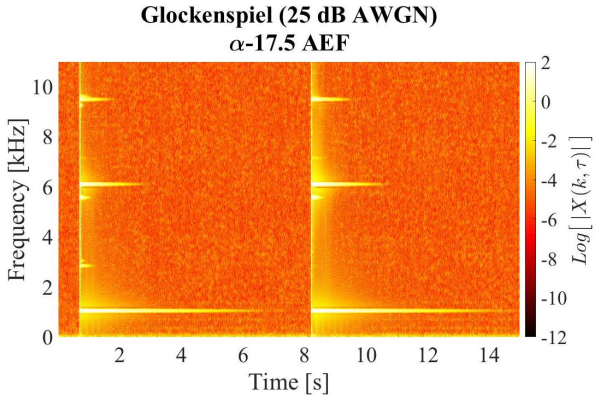
(a) Noise-free reference recording.



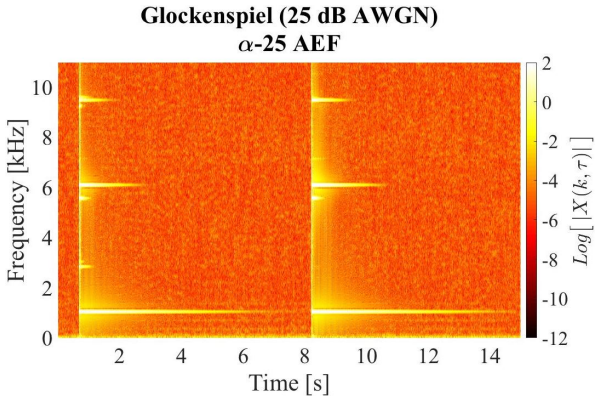
(b) Noisy reference recording with a global signal-to-noise ratio of 25 dB.



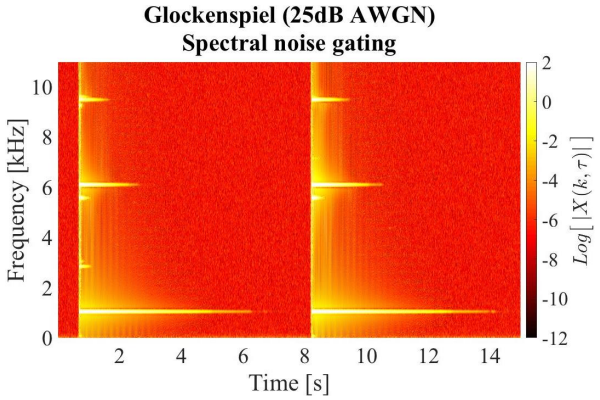
(c) The recording in (b), processed using adapted eigenimage filtering with α=10.



(d) The recording in (b), processed using adapted eigenimage filtering with α=17.5.



(e) The recording in (b), processed using adapted eigenimage filtering with α=25.



(f) The recording in (b), processed using spectral noise gating.

Figure B.12: Time-frequency representations of the glockenspiel recordings associated with additive white Gaussian noise (AWGN) at a global signal to noise-ratio of 25 dB.

Appendix C:

Subjective denoising results

C.1 Musical opinion scores

Table C.1: Individual opinion scores in the M-category given for all files in the audio library. The average M-score is shown in bold text. Note that participants rated only one third of the files on the M-category, based on which team they were in; the results of all three teams are bundled here for convenience of presentation.

Instrument	**Added noise**	**Processing**	**Musical score**								
			Participant								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Claves	None	None	5	5	4	3	4	5	2	4	**4.00**
	35 dB	None	4	4	5	4	4	4	4	3	**4.00**
		α-10 AEF	3	5	4	4	3	5	4	3	**3.88**
		α-17.5 AEF	5	4	5	4	4	5	4	4	**4.38**
		α-25 AEF	5	4	4	4	5	4	5	4	**4.38**
		SNG	3	5	5	5	3	5	4	4	**4.25**
	30 dB	None	5	4	5	5	3	4	4	5	**4.38**
		α-10 AEF	4	5	5	5	3	4	3	5	**4.25**
		α-17.5 AEF	4	3	4	3	3	4	4	3	**3.50**
		α-25 AEF	4	2	2	3	4	4	3	3	**3.13**
		SNG	5	5	4	5	2	5	4	5	**4.38**
	25 dB	None	4	5	3	1	3	4	4	3	**3.38**
		α-10 AEF	5	2	3	4	4	4	3	3	**3.50**
		α-17.5 AEF	3	3	4	2	3	2	3	2	**2.75**
		α-25 AEF	4	5	4	4	5	4	4	5	**4.38**
		SNG	5	3	3	5	1	5	2	4	**3.50**
Guitar	None	None	5	5	5	1	5	4	5	5	**4.38**
	35 dB	None	5	5	4	3	5	4	4	2	**4.00**
		α-10 AEF	4	5	5	4	5	4	4	3	**4.25**
		α-17.5 AEF	5	2	5	5	4	5	5	4	**4.38**
		α-25 AEF	5	5	4	5	5	4	5	5	**4.75**
		SNG	5	5	3	3	5	5	4	5	**4.38**
	30 dB	None	4	4	4	5	4	4	5	2	**4.00**
		α-10 AEF	5	3	3	1	4	3	4	3	**3.25**
		α-17.5 AEF	5	3	5	5	5	4	4	3	**4.25**
		α-25 AEF	4	2	4	2	3	3	4	2	**3.00**
		SNG	4	5	3	3	4	5	4	4	**4.00**
	25 dB	None	5	5	4	2	5	4	4	3	**4.00**
		α-10 AEF	5	2	3	1	4	4	4	1	**3.00**
		α-17.5 AEF	5	3	2	2	4	3	2	3	**3.00**
		α-25 AEF	4	5	3	3	3	4	4	3	**3.63**
		SNG	5	4	2	5	3	5	4	3	**3.88**

Instrument	Added noise	Processing	Musical score								
			Participant								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Singing	None	None	5	5	5	2	4	5	4	5	**4.38**
	35 dB	None	5	3	4	1	4	2	3	2	**3.00**
		α-10 AEF	5	5	5	2	5	4	4	3	**4.13**
		α-17.5 AEF	3	4	4	4	4	5	4	5	**4.00**
		α-25 AEF	4	4	5	4	5	5	4	3	**4.25**
		SNG	5	3	4	1	4	2	3	2	**3.50**
	30 dB	None	4	5	5	3	5	5	4	3	**4.25**
		α-10 AEF	5	1	4	2	2	4	4	3	**3.13**
		α-17.5 AEF	4	3	3	4	4	4	4	4	**3.75**
		α-25 AEF	3	3	3	4	5	5	4	3	**3.75**
		SNG	3	3	3	3	3	3	3	3	**3.00**
	25 dB	None	4	5	5	4	5	4	4	3	**4.25**
		α-10 AEF	5	5	5	2	4	4	4	2	**3.88**
		α-17.5 AEF	4	3	4	4	3	4	3	4	**3.63**
		α-25 AEF	4	5	4	3	4	4	4	4	**4.00**
		SNG	3	2	2	3	2	1	1	1	**1.88**
Glockenspiel	None	None	3	2	4	5	4	5	3	3	**3.63**
	35 dB	None	4	2	2	1	4	2	4	2	**2.63**
		α-10 AEF	4	3	3	1	3	3	4	3	**3.00**
		α-17.5 AEF	4	2	4	5	4	5	5	5	**4.25**
		α-25 AEF	4	4	4	1	4	3	5	3	**3.50**
		SNG	4	5	4	5	4	5	5	5	**4.63**
	30 dB	None	4	1	2	2	3	2	3	2	**2.38**
		α-10 AEF	5	2	3	5	5	4	3	3	**3.75**
		α-17.5 AEF	3	5	5	3	5	5	4	4	**4.25**
		α-25 AEF	4	1	3	5	4	4	4	3	**3.50**
		SNG	4	4	4	5	5	5	5	5	**4.63**
	25 dB	None	4	1	1	2	3	1	4	1	**2.13**
		α-10 AEF	4	2	4	5	4	4	5	4	**4.00**
		α-17.5 AEF	4	2	3	4	5	3	4	4	**3.63**
		α-25 AEF	4	4	4	3	4	5	4	4	**4.00**
		SNG	4	4	4	5	4	5	2	5	**4.13**

C.2 Background opinion scores

Table C.2: Individual opinion scores in the B-category given for all files in the audio library. The average B-score is shown in bold text. Note that participants rated only one third of the files on the B-category, based on which team they were in; the results of all three teams are bundled here for convenience of presentation.

Instrument	**Added noise**	**Processing**	**Background score**								
			Participant								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Claves	None	None	5	5	4	1	4	2	1	2	**3.00**
	35 dB	None	3	5	4	4	3	2	4	2	**3.38**
		α-10 AEF	4	5	3	5	4	3	4	4	**4.00**
		α-17.5 AEF	3	5	4	5	5	3	5	4	**4.25**
		α-25 AEF	3	4	1	2	3	2	1	4	**2.50**
		SNG	5	5	3	1	4	3	4	5	**3.75**
	30 dB	None	3	3	3	3	1	2	1	1	**2.13**
		α-10 AEF	3	3	3	4	4	5	3	3	**3.50**
		α-17.5 AEF	4	4	3	4	5	3	5	2	**3.75**
		α-25 AEF	3	4	2	3	4	4	3	2	**3.13**
		SNG	3	5	5	5	4	5	5	5	**4.63**
	25 dB	None	1	3	2	4	3	2	2	3	**2.50**
		α-10 AEF	2	3	2	2	2	2	2	1	**2.00**
		α-17.5 AEF	3	3	2	2	3	2	4	2	**2.63**
		α-25 AEF	2	2	1	2	2	1	2	2	**1.75**
		SNG	5	5	4	5	4	3	5	4	**4.38**
Guitar	None	None	5	5	5	5	5	5	5	5	**5.00**
	35 dB	None	3	4	4	3	4	2	2	3	**3.13**
		α-10 AEF	4	4	4	4	4	2	4	3	**3.63**
		α-17.5 AEF	2	4	3	2	2	3	2	3	**2.63**
		α-25 AEF	4	5	5	5	4	3	4	4	**4.25**
		SNG	5	5	5	5	5	5	5	4	**4.88**
	30 dB	None	1	2	3	5	3	2	1	1	**2.25**
		α-10 AEF	3	2	2	3	2	3	4	2	**2.63**
		α-17.5 AEF	3	3	3	3	3	3	3	3	**3.00**
		α-25 AEF	2	2	2	2	4	3	2	2	**2.38**
		SNG	5	5	4	5	4	4	5	3	**4.38**
	25 dB	None	3	1	1	1	1	1	1	1	**1.25**
		α-10 AEF	2	2	1	1	1	2	1	1	**1.38**
		α-17.5 AEF	1	2	2	1	2	3	2	2	**1.88**
		α-25 AEF	2	4	3	2	3	2	2	2	**2.50**
		SNG	3	2	3	3	2	2	4	3	**2.75**

Instrument	Added noise	Processing	Background score								
			Participant								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Singing	None	None	5	5	4	5	5	5	5	5	**4.88**
	35 dB	None	1	1	1	2	1	4	2	2	**1.75**
		α-10 AEF	4	3	3	5	4	3	2	4	**3.50**
		α-17.5 AEF	3	3	3	3	3	3	2	3	**2.88**
		α-25 AEF	4	4	5	5	3	3	3	3	**3.75**
		SNG	5	5	2	5	5	4	4	4	**4.25**
	30 dB	None	2	1	2	1	1	2	1	1	**1.38**
		α-10 AEF	1	1	2	1	1	4	3	1	**1.75**
		α-17.5 AEF	3	3	5	4	4	4	2	4	**3.63**
		α-25 AEF	2	2	4	4	3	2	3	2	**2.75**
		SNG	4	5	3	5	5	4	4	4	**4.25**
	25 dB	None	1	1	1	1	2	1	1	1	**1.13**
		α-10 AEF	2	1	2	1	2	2	1	1	**1.50**
		α-17.5 AEF	2	3	5	4	4	3	3	3	**3.38**
		α-25 AEF	1	1	4	4	4	3	2	2	**2.63**
		SNG	3	4	2	4	3	3	2	4	**3.13**
Glockenspiel	None	None	5	4	5	3	5	4	5	4	**4.38**
	35 dB	None	3	4	3	3	3	3	3	2	**3.00**
		α-10 AEF	4	5	3	3	5	3	4	2	**3.63**
		α-17.5 AEF	4	2	3	2	4	2	4	3	**3.00**
		α-25 AEF	5	5	3	4	5	3	4	4	**4.13**
		SNG	5	5	4	5	4	5	5	5	**4.75**
	30 dB	None	2	2	2	2	2	4	2	2	**2.25**
		α-10 AEF	3	2	5	2	4	4	3	2	**3.13**
		α-17.5 AEF	4	5	4	5	5	3	4	4	**4.25**
		α-25 AEF	4	3	4	2	2	4	4	3	**3.25**
		SNG	5	5	4	1	4	4	5	4	**4.00**
	25 dB	None	1	2	1	1	1	2	1	1	**1.25**
		α-10 AEF	2	2	3	3	4	3	3	3	**2.88**
		α-17.5 AEF	3	3	2	3	2	2	4	2	**2.63**
		α-25 AEF	3	4	3	3	4	2	2	2	**2.88**
		SNG	4	5	3	5	3	4	5	4	**4.13**

C.3 Overall opinion scores

Table C.3: Individual opinion scores in the O-category given for all files in the audio library. The average O-score is shown in bold text. Note that participants rated only one third of the files on the O-category, based on which team they were in; the results of all three teams are bundled here for convenience of presentation.

Instrument	**Added noise**	**Processing**	**Overall score**								
			Participant								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Claves	None	None	4	3	4	5	4	4	4	5	**4.13**
	35 dB	None	3	3	3	2	2	3	2	2	**1.63**
		α-10 AEF	3	3	3	2	2	2	3	4	**3.25**
		α-17.5 AEF	3	4	3	2	4	3	4	3	**3.25**
		α-25 AEF	5	5	3	5	4	4	4	4	**3.13**
		SNG	5	4	3	5	3	5	4	5	**4.13**
	30 dB	None	2	2	2	4	3	2	2	3	**2.50**
		α-10 AEF	4	4	2	4	2	4	4	3	**3.38**
		α-17.5 AEF	4	5	5	5	5	4	4	5	**4.63**
		α-25 AEF	4	5	4	5	4	2	4	4	**4.00**
		SNG	4	5	4	2	5	3	5	4	**4.00**
	25 dB	None	1	3	1	1	1	3	2	1	**1.63**
		α-10 AEF	4	4	3	5	4	1	2	3	**3.25**
		α-17.5 AEF	3	4	3	5	4	2	2	3	**3.25**
		α-25 AEF	3	4	2	2	2	5	4	3	**3.13**
		SNG	5	5	4	3	4	3	5	4	**4.13**
Guitar	None	None	5	5	5	5	5	5	4	5	**4.88**
	35 dB	None	2	2	2	3	3	3	3	3	**2.63**
		α-10 AEF	3	3	2	3	4	4	3	3	**3.13**
		α-17.5 AEF	4	5	3	3	4	4	4	4	**3.88**
		α-25 AEF	3	5	3	3	3	3	4	4	**3.50**
		SNG	4	5	4	1	5	2	5	4	**3.75**
	30 dB	None	1	4	1	1	1	2	2	1	**1.63**
		α-10 AEF	4	3	5	5	5	2	4	4	**4.00**
		α-17.5 AEF	2.5	3	3	3	3	5	5	3	**3.44**
		α-25 AEF	4	5	4	4	4	3	3	3	**3.75**
		SNG	4	5	4	2	4	3	4	4	**3.75**
	25 dB	None	2	1	2	4	2	1	1	1	**1.75**
		α-10 AEF	3	3	3	3	3	2	1	2	**2.50**
		α-17.5 AEF	3	3	3	3	4	2	2	3	**2.88**
		α-25 AEF	3	1	3	3	4	4	3	2	**2.88**
		SNG	4	3	2	3	4	4	2	2	**3.00**

Instrument	Added noise	Processing	Overall score								
			Participant								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Avg.**
Singing	None	None	4	4	4	5	5	5	4	4	**4.38**
	35 dB	None	3	2	2	3	2	1	1	1	**1.88**
		α-10 AEF	2	3	4	4	2	4	3	2	**3.00**
		α-17.5 AEF	2.5	4	2	2	3	4	3	2	**2.81**
		α-25 AEF	3	4	3	3	5	1	3	4	**3.25**
		SNG	3	3	3	5	5	5	2	3	**3.63**
	30 dB	None	1	1	2	4	2	2	1	1	**1.75**
		α-10 AEF	3	2	2	2	4	2	1	3	**2.38**
		α-17.5 AEF	3	3	1	2	2	3	4	3	**2.63**
		α-25 AEF	3	3	1	3	2	3	3	3	**2.63**
		SNG	2	2	3	4	4	4	5	3	**3.38**
	25 dB	None	1	1	2	5	2	1	1	1	**1.75**
		α-10 AEF	2	1	3	4	3	3	1	2	**2.38**
		α-17.5 AEF	2	3	1	2	1	3	2	2	**2.00**
		α-25 AEF	2	2	1	2	1	3	3	2	**2.00**
		SNG	2	3	2	4	3	2	3	2	**2.63**
Glockenspiel	None	None	5	5	4	4	4	4	4	5	**4.38**
	35 dB	None	3	3	4	5	3	2	3	2	**3.13**
		α-10 AEF	5	3	4	5	4	3	3	3	**3.75**
		α-17.5 AEF	4	5	4	3	5	4	4	3	**4.00**
		α-25 AEF	4	3	4	5	5	4	5	4	**4.25**
		SNG	5	2	3	1	3	3	3	2	**2.75**
	30 dB	None	3	2	3	4	2	2	2	2	**2.50**
		α-10 AEF	3	4	3	2	4	4	5	3	**3.50**
		α-17.5 AEF	3	2	5	3	3	5	3	2	**3.25**
		α-25 AEF	3	4	3	3	4	3	3	3	**3.25**
		SNG	5	5	4	4	3	5	4	5	**4.38**
	25 dB	None	3	2	1	2	2	1	1	1	**1.63**
		α-10 AEF	2	3	2	2	1	3	2	3	**2.25**
		α-17.5 AEF	4	4	3	4	5	3	2	4	**3.63**
		α-25 AEF	2	1	3	3	4	3	2	3	**2.63**
		SNG	4	4	3	1	5	3	5	3	**3.50**

Appendix D:

Subjective test instructions

Audio evaluation test

In this experiment you will be evaluating and rating the quality of audio samples. The audio consists of 4 short recordings of acoustic guitar, claves (a wooden percussive instrument commonly used in Cuban music), a glockenspiel (related to the xylophone), and the sung voice. You will hear multiple versions of the same recordings: some have been submerged in noise of different volumes, and some have been processed.
The file names are structured, and could for example look like '**5_B.wav**'. The first number shows you which 'trial' the file is in, the letter shows you what you should be rating the audio on.
The test is subdivided in 21 trials. Trials consist of three short audio fragments (except for the last one, which has four), so within each trial, you will give three (or four) ratings; **one for each** of the audio fragments.
For the sample whose name ends with an **M**, you are instructed to pay attention **only to the music signal**, and rate how distorted or unnatural it sounds to you. From the list below, please choose the numbered phrase that best corresponds to your opinion of the music alone, and type this in the cell with the same name on the excel sheet.

Attending only to the music signal, select which of the following best describes the music fragment you just heard.

the music signal in this sample was

5. - Not distorted, completely natural
4. - Slightly distorted, almost natural
3. - Somewhat distorted or unnatural
2. - Fairly distorted or unnatural
1. - Very distorted or unnatural

For the sound fragment whose name ends with a **B**, you are instructed to pay attention **only to the background** and rate how noticeable or intrusive the background sounds to you. Use the rating scale provided in the second table, and pick the numbered phrase from the list that best describes how you perceive the background alone. Please fill in this number in the cell with the same name in the excel rating sheet.

Attending only to the background, select which of the following best describes the music fragment you just heard.

the background in this sample was

5. - Not noticeable

4. - Slightly noticeable

3. - Noticeable, but not intrusive

2. - Somewhat intrusive

1. - Very intrusive

Finally, the last file in each trial has a name that ends with an **O**. For these files, I want you to listen to the audio fragment and rate your opinion on the **overall quality** of the sound fragment. Maybe pretend that someone is showing you their recording in Soundtrap, so take both the music signal and the background noise into account. As before, please fill your rating in the corresponding space in the rating sheet.

Select which of the following best corresponds to your overall quality rating of the audio file you heard, for the purposes of a user's project in Soundtrap.

the overall recording quality in this sample was

5. - Excellent

4. - Good

3. - Fair

2. - Poor

1. - Bad

The test as a whole should take around 25-30 minutes. Please do it at your convenience; you are allowed to do it in one or multiple sittings, so feel free to take a break in between when necessary. I do strongly encourage the use of headphones or external speakers over the use of lower-quality earphones or computer built-in speakers, but use whatever you have available. You are allowed to play the files multiple times and jump back and forth if that helps you form your opinion, just make sure you fill in your rating in the right place. If you have any questions or if anything seems weird, don't hesitate to let me know.
Thank you very much for participating!