f11022

Examensarbete 30 hp
Februari 2011

Audio editing in the time-frequency domain using the Gabor Wavelet Transform

Ulf Hammarqvist

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Audio editing in the time-frequency domain using the Gabor Wavelet Transform

Ulf Hammarqvist

Visualization, processing and editing of audio, directly on a time-frequency surface, is the scope of this thesis. More precisely the scalogram produced by a Gabor Wavelet transform is used, which is a powerful alternative to traditional techinques where the wave form is the main visual aid and editting is performed by parametric filters. Reconstruction properties, scalogram design and enhancements as well audio manipulation algorithms are investigated for this audio representation.

The scalogram is designed to allow a flexible choice of time-frequency ratio, while maintaining high quality reconstruction. For this mean, the Loglet is used, which is observed to be the most suitable filter choice. Re-assignment
are tested, and a novel weighting function using partial derivatives of phase is proposed. An audio interpolation procedure is developed and shown to perform well in listening tests.

The feasibility to use the transform coefficients directly for various purposes is investigated. It is concluded that Pitch shifts are hard to describe in the framework while noise thresh holding works well. A downsampling scheme is suggested that saves on operations and memory consumption as well as it speeds up real world implementations significantly. Finally, a Scalogram 'compression' procedure is developed, allowing the caching of an approximate scalogram.

Handledare: Erik Wernersson
Ämnesgranskare: Anders Brun
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F11022

Introduction

This thesis covers the design of Wavelet based filters for audio analysis and also contains applications where these filters are used for audio restoration.

The material is best understood with knowledge about signal processing and audio analysis. The reader is directed to literature [1, 2] for a more comprehensive background on the mathematical and engineering aspects, and for an explanation of basic terminology.

Background

Traditionally, user interfaces in audio editing software presents multiple channels or clips of waveforms parallel along a time line, which gives a work flow analogous to editing with multiple tape-recorders.

With such approach, the visual representation of the audio, where the amplitude of the waveforms are plotted over time, has a vague perceptual connection to the produced audio signal, as it only gives a rough idea of the instantaneous sound pressure but not the signal, spectral, content. This is why using some time-frequency representation, usually a spectrogram or a spectrometer, becomes a vital analysis and visual reference tool.

A time-frequency representation can also be used for interactive editing as long as the generating transformation has an inverse. Changing the coefficients from the transformation is conceptually the same as applying time dependent filters controlled by user input. Some intricate operations, for example smoothing of transients, normally done by specialized algorithms can now just as well be done by manual user interaction. Another example is the removal of specific overtones in speech, which this approach significantly simplifies.

The problems stated in the thesis are based on the needs of Sonic AWE [3], a piece of software that allows editing audio in a time-frequency representation. The software has its focus on interactive direct editing of audio but here the focus is on the transform that maps the sound into a time-frequency representation and a few specific algorithms using the transformed data. The thesis will not deal with the specifics of how to implement these interactive filter operations, even if some design guidelines are discussed briefly, but in-

stead focuses on both a lower and a higher level. Even though the thesis is associated with a specific software project, the methods are general.

Goal

The problems reached for by this thesis are both theoretical, concerning the actual transformation, as well as practical, developing some specific end user audio processing tools that uses the time-frequency data as input.

Three main objectives were set for the thesis.

1. Design of the time-frequency representation to:
    - Find a way to change the time-frequency resolution of the representation by a user parameter while ensuring stable reconstruction and keeping filter the amount of overlap low, and,
    - see if there are any possible enhancements to the visual representation supporting interpretation.
2. Develop tools and algorithms that serve to:
    - interpolate audio to replace missing or damaged sections,
    - remove noise by spectral thresh holding, and,
    - move audio signals perceptually in frequency and stretch them in time.
3. Take a closer look at implementation details of the transformation to:
    - investigate ways to compute it faster, and,
    - ensure low reconstruction errors.

Method

The work started with a literature study in order to get an overview of the vast amount of previously published material in the related fields, audio engineering, signal processing, image analysis and wavelet theory.

Since hearing is a non-linear process and audio quality a subjective measure, performance of the audio tools are hard to measure with numerical means. In order to test methods found in literature as well as the authors own ideas a lot of time was spent on implementation. A formal listening test has also been conducted in order to evaluate one of the methods.

The code for all experiments and algorithms, except the listening test, including the Wavelet transform was implemented by the author. To this end MATLAB was used.

Contents

Chapter 1

Theory

Audio is, in the physical and biological sense, pressure oscillations picked up by our auditory system. Roughly speaking, the perceptual system only perceives how the air pressure changes. A slow increase in pressure, or a static pressure, has no real meaning for the hearing perception. Exactly how the conversion to a pressure wave to our perception works is a subject of biological and neural research itself. In fact, the nerves seem to constitute some sort of (non-linear) filter bank [4].

A repeating structure has a frequency - the rate of repetition. The smoothest form of repeating signal is a harmonic function. In the one dimensional case, this is mathematically expressed by a cosine (or sine),

$$f(t) = \cos(\omega t + \Phi), \tag{1.1}$$

where ω is the frequency and Φ a phase offset.

A Fourier transform conceptually expresses a signal as sum of such, complex valued and harmonic waves. The mathematical details, and history, of Fourier transforms and relations to Fourier series can be found in literature [5]. In the continuous case, the Fourier transform of an analytical function is given by,

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\omega x}\, dx. \tag{1.2}$$

This is one of many definitions, this one is the unitary transform expressed with angular frequencies (as opposed to ordinary frequencies). A unitary transformation means that $||\hat{f}|| = ||f||$. In the digitized world signals have to be sampled discretely in time and amplitude. In this context the Discrete Fourier Transform is used. It is expressed as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \tag{1.3}$$

where x_n are the samples of a discrete signal of length N.

1.1 Time-frequency analysis

If the goal is to analyze audio in a sense that relates to our hearing, so that an intuitive connection can be made, computing and visualizing the Fourier transform of the signal is a poor choice. We need a signal representation that shows spectral content as a function of time. A straightforward remedy is to make the Fourier analysis over short consecutive time segments of the signal instead of the whole signal. This is called Short Time Fourier Transform, STFT, and generalized in Gabor Frame theory as a Gabor Transform [6].

The Discrete Fourier transform, Eq. 1.3, of a signal of N samples produces N frequency bins, N/2 positive and N/2 negative frequencies. The frequency localization of each bin depends on time windowing used. A square window will create *sinc* shaped frequency bins. To get smoother frequency localization a smoother time window is often used, which in turns means that, in practice, the time windows have to overlap or the signal cannot be completely reconstructed. For a deeper discussion about window shapes and their Fourier transforms see literature, such as textbooks on spectral analysis [7].

If a signal, or signal component, fluctuates in frequency over the time span of the window this is generally obscured. This uncertainty is usually explained in the form of the Heisenberg uncertainty principle. If denoting the time uncertainty as σ_t and the frequency uncertainty as σ_ω for a given point in the time-frequency plane, the uncertainty principle can be explained with the relation

$$\sigma_\omega \sigma_t \propto 1.$$

This concept is very important to understand as it applies to all time-frequency analysis.

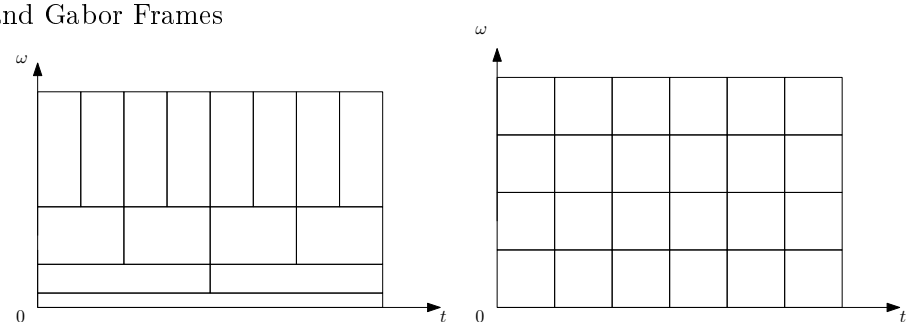
Other time-frequency representations are produced by the family of Wavelet Transforms. From an engineering standpoint, filter banks are also at the disposal. However, both STFT and Wavelet Transforms can be interpreted as filter banks as well. It is just a matter of mathematical formulation and implementation aspects. Comprehensive reviews and overviews of time-frequency distributions can be found in litterature [8, 9, 10, 1].

1.1.1 Time-frequency vs Time-Tonal bandwidth

The Fourier Transform maps the signal into a combination of basis functions linearly spaced in frequency. The Mel scale [11], which is derived as the perceived unit distance between frequencies, follows something more like a logarithmic scale. Notes in music follow a logarithmic scale - an octave step means the double (or half) frequency. The scale axis in the Wavelet domain is analogous to a logarithmic frequency axis. The time-frequency bandwidth in the Short Time Fourier domain is exchanged for a period-tonal bandwidth

in the Wavelet domain. Fig. 1.2(b) and 1.2(a) illustrates this by with boxes depicting the time and frequency spread - so called Heisenberg boxes.

Figure 1.1: Heisenberg box representation comparing Gabor Wavelet Frames and Gabor Frames



(a) Time frequency spread of Gabor Wavelet Atoms

(b) Time frequency spread of Gabor Atoms

This shows how the wavelet transform has, relatively, increasingly better frequency localization for lower frequencies (but worse time localization) as compared to the STFT.

1.2 Complex Gabor Wavelet Transform

Generally speaking a Wavelet Transform describes a signal as combination of scaled and spatially shifted versions of some function. This function is called the Mother Wavelet and the scaled versions are called Child Wavelets.

The formulation of a Wavelet transform is split up into a series of theorems, which are needed in order to ensure that the wavelet decomposition completely describes the signal and allows a reconstruction from the coefficients (a way back). Most noteworthy is the necessity to split the transformation up into an analysis part and a reconstruction part. This is explained as frames and their corresponding dual frames, which is a generalization of bases and dual bases [6, 1]. This is a very general form that allows more freedom in the wavelet shape - as long as a dual wavelet can be found reconstruction is possible.

The specific wavelet used in this thesis is the Complex Gabor Wavelet. In a sense, it is one of the simplest wavelets and also one of the original wavelets, closely related to the Morlet [12, 1, 13]. In the time domain it can be described as a complex harmonic wave multiplied by some window function. The most common choice for window is a Gaussian, as it is very well localized in both time and frequency. It can also be explained in the frequency domain by it's Fourier transform - in this case a Gaussian shaped band pass filter.

The equations involved for the Continuous Gabor Wavelet Transform are

presented below for completeness. The equations and notations are taken from Stephane Mallat's textbook [1].

A Wavelet transform is defined by it's Mother Wavelet. The Complex Gabor Mother Wavelet is defined as:

$$\psi(t) = g(t)e^{i\eta t},$$

where

$$g(t) = \frac{1}{(\sigma^2\pi)^{\frac{1}{4}}} e^{\frac{-t^2}{2\sigma^2}}, \tag{1.4}$$

is a Gaussian bell, and it's Fourier transform is:

$$\hat{g}(\omega) = (4\pi\sigma^2)^{\frac{1}{4}} e^{-\frac{\sigma^2\omega^2}{2}}. \tag{1.5}$$

The reason for expressing the wavelets in the Fourier domain is two-fold, understanding the how they 'cover' the frequency axis and the actual calculation, a common choice for the redundant Gabor Wavelet is to compute the coefficients via multiplications in the Fourier domain.

The Mother Wavelet is scaled into the so called Child Wavelets. A Child Wavelet at scale j is, in Fourier domain, given by:

$$\hat{\psi}_j(\omega) = \sqrt{a^j}\hat{\psi}(a^j\omega) = \sqrt{a^j}\hat{g}(a^j\omega - \eta),$$

$$\hat{\psi}_j(\omega) = (4\pi(a^j\sigma)^2)^{\frac{1}{4}} e^{-\frac{(a^j\sigma)^2(\omega - \frac{\eta}{a^j})^2}{2}}.$$

In this sense, a Child wavelet can be understood as a filter and the transform as a filter bank – a collection of filters. This analogy will be clearer in the discrete case.

The Wavelet transform is then defined as:

$$Wf(t, a^j) = \int_0^\infty f(u)\psi_j^*(u - t)du = f \circledast \overline{\psi}_j(t) \tag{1.6}$$

and the inverse as:

$$f(t) = \frac{2}{C_\psi} Re\left\{ \int_0^\infty \int_{-\infty}^\infty Wf(u, a^j)\psi_a^j(t - u)du\frac{ds}{s^2} \right\},$$

with

$$C_\psi = |\hat{\phi}(0)|^2,$$

$$|\hat{\phi}(\omega)|^2 = \int_\omega^\infty \frac{|\hat{\psi}(\xi)|^2}{\xi} d\xi. \tag{1.7}$$

where $\phi(t)$ is so called 'scaling function', or sometimes 'father wavelet'.

The discrete version of Eq. 1.6 is:

$$f[n] \approx \frac{log_e a}{C_\psi} \sum_{j=1}^{J} \frac{1}{a^j} Wf\left[., a^j\right] \circledast \psi_j[n] + \frac{1}{C_\psi a^j} Lf\left[., a^j\right] \circledast \Phi_j[n], \tag{1.8}$$

where $\Phi_j[n]$ is the concatenation of all scales larger than J and:

$$\begin{aligned} Wf\left[.,a^j\right] &= f \circledast \bar{\psi}_j[n] \\ Lf\left[.,a^j\right] &= f \circledast \bar{\Phi}_J[n]. \end{aligned} \tag{1.9}$$

1.2.1 The scalogram

The scalogram is a visual representation of the Wavelet coefficients constructed by mapping the absolute values onto a two-dimensional plane. The first dimension is scale, analogous to a logarithmic frequency, and expressed by the scale number j. The second dimension is dilation, the time shift of the Child wavelets, which is interpreted as time. In this thesis the Complex Gabor Wavelet is used to create the scalograms. The motivation behind using a complex valued wavelet is best understood by investigating it's Fourier transform and by explaining the idea behind the analytical signal representation – the Gabor Wavelet transform is essentially a quadrature filter bank.

1.2.2 Analytical signal representation

Any real valued signal can be expressed by:

$$f(t) = Re\left\{f_a(t)\right\},$$

where $f_a(t)$ is the analytical signal form of $f(t)$. In some formulations a factor 2 is seen in the right hand side. This depends on how $f_a(t)$ is defined. In this thesis the following is used:

$$f_a(t) = f(t) + i\hat{f}(t), \tag{1.10}$$

where $\hat{f}(t)$ is the Hilbert transform of $f(t)$.

In order to express a discrete real valued signal $f[n]$ of length $2N$ on analytical form $f_a[n]$, the Discrete Fourier Transform $\hat{X}[m]$ is computed and modified, i.e. multiplied with a Heavy-side window, so that:

$$\hat{X}[m] = \begin{cases} \hat{X}[m] & \text{if } m \in \{1, N+1\} \\ 2\hat{X}[m] & \text{if } m \in \{2, \ldots, N\} \\ 0 & \text{otherwise} \end{cases}. \tag{1.11}$$

and then inverse transformed.

The Complex Gabor Wavelet is in this context the impulse response of a band-pass filter that transforms the filtered signal onto an analytical representation. The equivalent to a power spectrum is then given by taking the absolute value of the coefficients. These types of filters are sometimes called Quadrature filters.

Informally, the Hilbert transform of a signal is the signal itself phase shifted by π and multiplied by the complex number $1i$. Thus, the analytical signal representation can be used to extract an estimate of the instantaneous phase and amplitude. The phase is extremely useful when analyzing narrow band signals. For a cosine this phase is a very good approximation of the actual phase. This also makes it possible to express signals on polar form as:

$$\begin{aligned} f(t) &\approx A(t)e^{-ß\Phi(t)} \\ A(t) &= |f_a(t)| \\ \Phi(t) &= \angle f_a(t). \end{aligned} \tag{1.12}$$

The instantaneous frequency is then:

$$\omega(t) = \frac{\delta \Phi_{uw}(t)}{\delta t}, \tag{1.13}$$

where $\Phi_{uw}(t)$ is the unwrapped phase. The phase is a discontinuous function ranging from $-\pi$ to π. Unwrapping it means adding a function to the phase so that the discontinuities disappear. If considering the phase of the last point in the first phase cycle, ϕ_0, and the phase of the first point in the second cycle, ϕ_1, unwrapping these two phase cycles then means adding 2π to the phase values of the second cycle so that $\phi_1 = \phi_0 + \phi_\delta$, where ϕ_δ is small. Extending this to n phase cycles means that the n:th phase cycle has a term $2\pi n$ added added to the phase.

1.2.3 The Gabor Wavelet Transform as a filter bank

It has been demonstrated by Ingrid Daubechies that using a Gaussian as window function the resulting filters do not constitute a proper wavelet frame [13], however it is possible to construct an almost tight frame, which makes it possible to approximate the dual frame with the original frame. Another option is synthesizing a dual frame [1]. It is argued however that such a dual frame does not really improve the reconstruction [14], so approximating the dual frame with the frame itself is reasonable.

Two consecutive filter operations, convolutions, with the same filter are equivalent to one filter constructed from multiplying the filters in Fourier domain. Since the Fourier transform of the Gabor Mother Wavelets are symmetrical around their center frequency, the Gabor Wavelet transform can be simplified into one filter bank and the scalogram is constructed from these coefficients instead. The inversion is then the real part of the summation of the coefficients for all scales.

The number of scales per octave is expressed as v. This parameter controls the density of the filters in frequency domain, how many Child Wavelets

that are constructed for every octave. The coefficients from filtering $f(t)$ with the filter at scale j is now expressed as:

$$f(t)_{a^j} = \frac{ln(2^{1/v})}{C_\psi} F^{-1}\{\hat{f}\hat{\psi}^\star_{a^j}\} \tag{1.14}$$

$$f(t) = Re\left\{\sum_0^J f(t)_{a^j} + \phi^\star_J\right\},$$

where

$$\hat{\psi}^\star_{a^j}(\omega) = \frac{\hat{\psi}^2_{a^j}}{a^{2j}},$$

and

$$\phi^\star_J = \sum_{J+1}^{M}, M \to \infty \tag{1.15}$$

is the sum of all filters corresponding to scales larger than J.

The scale parameter a^j is chosen as $2^{1/v}$ in order to discretize the scales as fractions of octaves, and the ratio $\frac{1}{\sigma}$ is sufficiently large so the Fourier transform of the Child Wavelets (approximately) sums to a constant for the choice of v.

An example of a Gabor Wavelet filter bank is shown in Fig. 1.2.

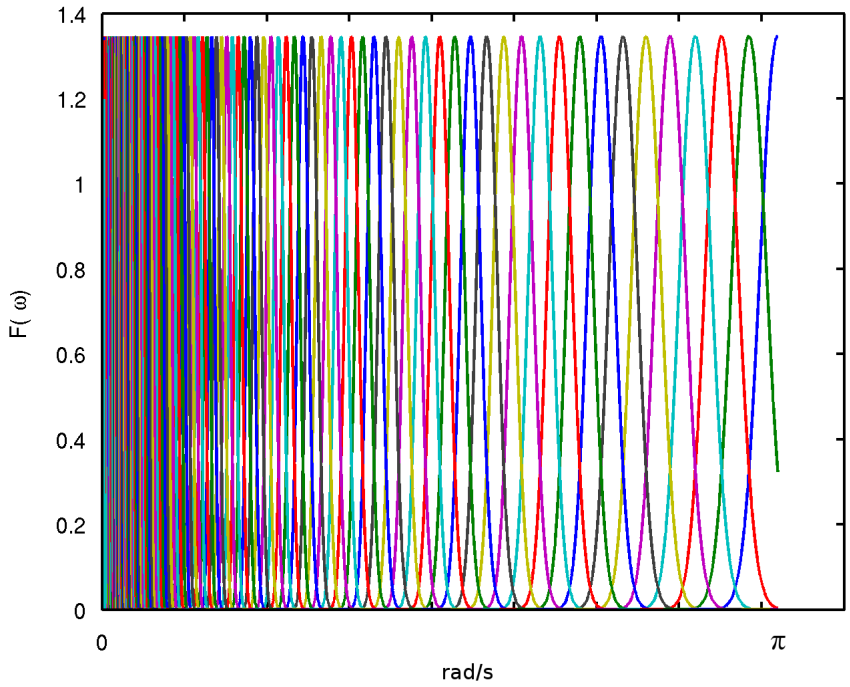


Figure 1.2: The Fourier transform of Gabor Wavelet filter bank, a collection of Child Wavelets, plotted on a linear frequency axis.

Chapter 2

The time-frequency surface

In this section two aspects are covered. First the impact of different time-frequency resolutions is illustrated. Following that attempts at refining the scalogram, essentially seeking to bend the uncertainty principle, are investigated. This involves the introduction of novel, to the author's knowledge, weighting function.

As there is no generic 'best' choice of basis (or frame) for a generic audio signal it is an inherit design of choice to offer multiple choices. The user can then pick a representation that works best for the task at hand, and additionally tweak parameters such as time-frequency resolution ratio to get a better understanding and precision for tools acting on the underlying data.

The methods that seek to refine the scalogram are judged by *visual improvement*, *flexibility* and *computational burden*. Visual improvement is a subjective measure. Flexibility in this context is whether there can be intermediate results of a normal and refined scalogram. Computational burden is the amount of extra computations needed (in rough terms).

2.1 Time-Frequency resolution ratio

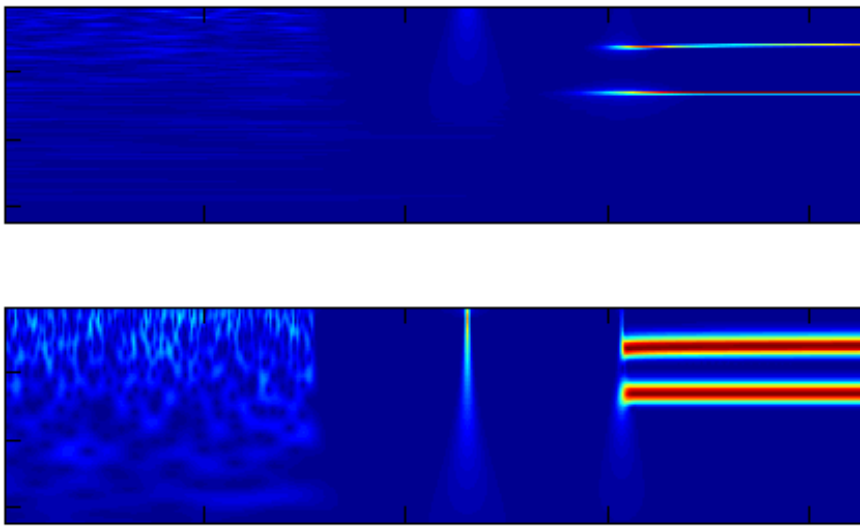
The first design objective was to allow a scalable time frequency resolution. This is achieved by modifying the mother wavelets time support, by widening and narrowing the time domain window given in Eq. 1.4. Equivalently this is a narrowing and widening of the frequency support respectively, from Eq. 1.5. The relationship between the center frequencies $\frac{\eta}{a^j}$ and the width of the Gaussian shaped filters, expressed in terms of $\frac{a^j}{\sigma}$, is so that the bells overlap 'enough' to still constitute a (approximately) tight frame. This is discussed and addressed numerically in Chapter 5. A test signal consisting of noise, a transient and two sinusoids is presented in Fig. 2.1(a) and Fig. 2.1(b), in the form of scalograms with different time-frequency resolutions.

2.1.1 Linear combinations of resolution ratios

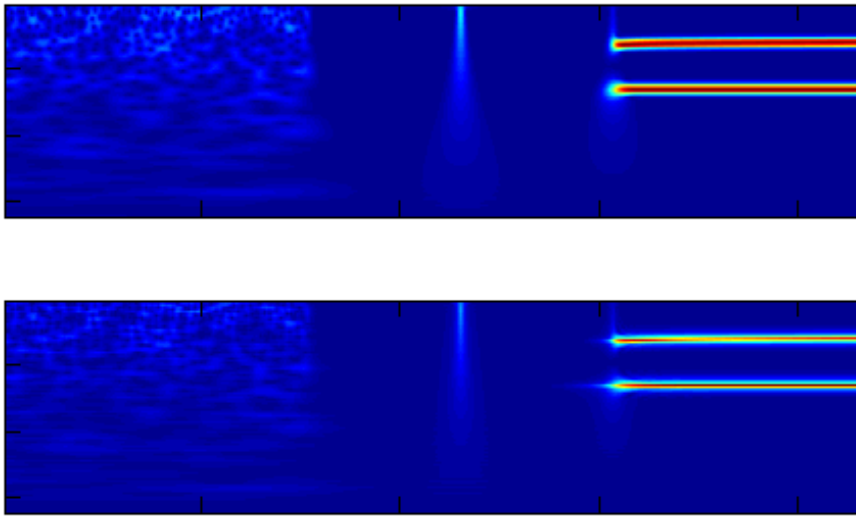
An interesting idea is to construct a linear combination of several resolution ratios - thus effectively creating a new wavelet that is a linear combination of other wavelets. The idea is that traits from several resolution ratios is combined - thus concentrating the energy to some degree in both time and frequency. This can be achieved by freezing v, but modifying the resolution parameter (in this case, σ). This way the coefficients will effectively be linear combinations from other time-frequency ratios. The resulting wavelet is then also a Gabor Wavelet, with an additional parameter (or set of parameters) that controls what other resolution ratios to include (and how much of them, if constructing the combination as a weighted sum). The result is a scalogram where sinusoids are better localized in frequency, and impulses better localized time. However, the vice versa also applies. In Fig. 2.1(b) such a surface constructed with $\sigma/4$, $\sigma/2$, σ, 2σ, 4σ is presented and compared with a normal scalogram using σ.

Result and Discussion

Combining several choices of time-frequency resolutions (effectively forming a new wavelet) can be a useful to the user interpretation - as the resulting wavelet have an increased localization of transients in time direction and sinusoidal components in frequency direction. This can be implemented as two additional parameters to change the shape of the time-frequency plane, while still retaining a perfect inverse (if each frame itself allows perfect inverse, that is).



(a) Top: Scalogram using 4 σ. Bottom: Scalogram using $\sigma/4$



(b) Top: Scalogram using σ. Bottom: Linear combination of time-frequency ratios, effectively forming a new wavelet.

Figure 2.1: Four scalograms with different time-frequency resolutions. The test signal is composite signal of a noise burst, followed by a silent period with a sharp transient, and last two sinusoids. The y-axis (bottom to top) is *log* frequency, and x-axis (left to right) is time. Notice that the higher frequency sinusoid changes frequency slightly over time.

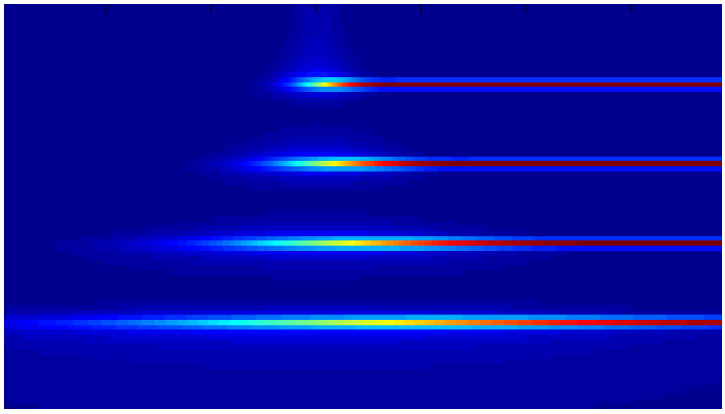
The linear combination of wavelets will inherent both the worst and best traits of the wavelets it is constructed from - it is effectively just pushing the time-frequency uncertainty into a star like pattern when constructed as in the example. To the author's knowledge this type of filter is not used in time-frequency analysis of audio. This combination can be generalized in many ways and the specifics was left as an implementation detail. The simplest form however would be to do as in the example shown in Fig. 2.1(b) - adding a few extra resolution ratios symmetrically around a center resolution.

- Visual improvement. There seems to be some benefit in creating a combinational frame. The uncertainty principle is naturally still present, however manifests itself in a different manner than the usual Gaussian blob.
- Flexibility. The amount of extra frames, or rather how the filters are reshaped, can be controlled freely as it's defined as a mean representation of other frames.
- Computational burden. The extra computation is low, as this is simply another filter shape than the usual Gaussian. Any additional computation is therefor in more terms in the filter equation. However, more filters needs to be computed if adding terms of more narrow band, in order to retain the reconstruction properties.

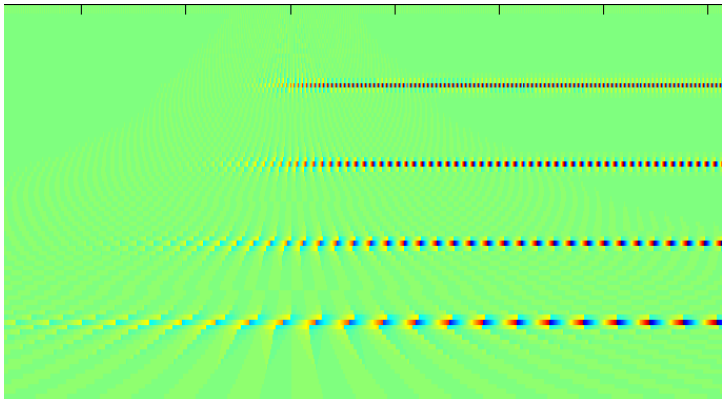
2.2 Restructuring the time-frequency distribution

Since the the spectrogram and scalogram both can be seen as smoothed variants of the Wigner Ville distribution (formalized as Cohen's class) [15], many authors seek to 'sharpen' the time-frequency representations – for better localization of partials and transients for feature extraction or sinusoidal model construction. As an illustration of this smearing, the scalogram of a set of sinusoid is presented in Fig. 2.2(a). This is obscured in the scalogram, as that only shows the absolute value. The phase plot of the same signal shows this a little better, this is presented in Fig. 2.2(b). Notice how the intermediate scales drift apart.

The ideas presented here all makes use of the phase information to alter the scalogram.



(a) Instantaneous amplitude of sinusoids of different frequencies. (Scalogram)



(b) Instantaneous phase of sinusoids of different frequencies (weighted by amplitude for visual localization)

Figure 2.2: Example signal that shows how the scalogram obscures essential information, the phase, of the signals. The phase gives a hint how the scales can add up to form a perfect reconstruction even if they appear to have been 'smeared' in the scalogram.

2.2.1 Re-assignment

Re-assignment of a time-frequency distribution is moving coefficients to other coordinates as defined by the partial derivatives [16], in respect to time and frequency, of the instantaneous phase. If this is a theoretically sound approach or not is not discussed here, but as several authors seems to have benefited from this approach [17] [18] we were curious to see what it would do for the visual interpretation in the case of the scalogram. The equations were translated from a spectrogram context into a Gabor Wavelet context:

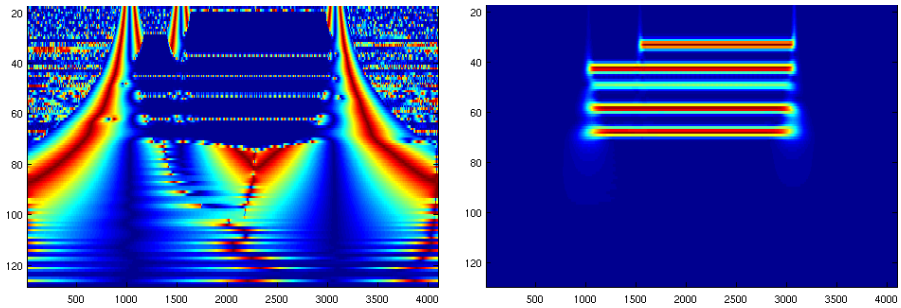
$$\text{IF} = \frac{d}{dt}\angle F(\omega, t),$$

$$\text{LGD} = -\frac{d}{d\omega}\angle F(\omega, t),$$

where ω is the center frequency of Child Wavelet at a certain scale.

IF is the instantaneous frequency. The LGD is interpreted as a timing error. The re-assignment 'coordinates' are then given by the scale corresponding to IF and time to t - LGD.

For reference, the absolute value of the LGD is shown next to the scalogram of the same signal in Fig. 2.3



(a) Absolute value of LGD (phase derivative in scale direction), blue region are close to zero.

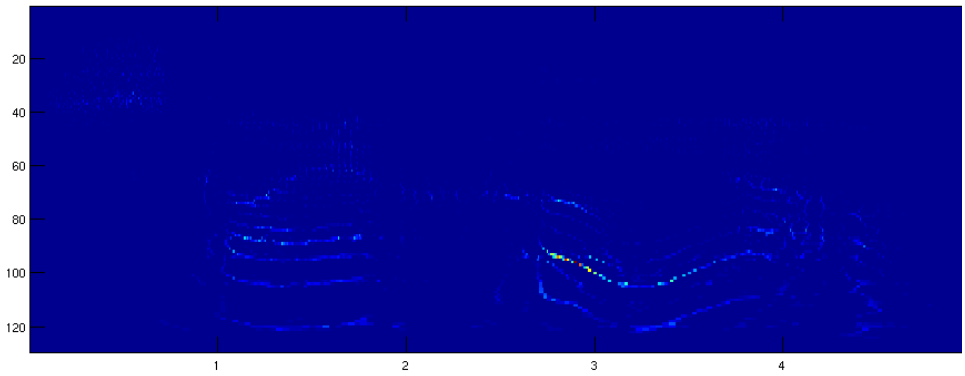
(b) Scalogram reference

Figure 2.3: Exploiting information about the derivatives of phase in the coefficients, common onset, true time support and similar features can be found. Here the phase derivative in scale direction is shown next to a scalogram for the same signal. The signal is five sinusoids, four have a common onset in time.

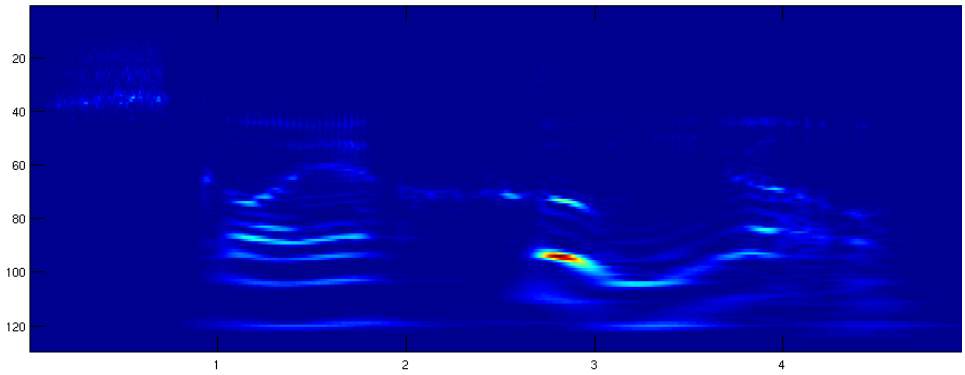
Result and Discussion

An experiment of reassignment of coefficients for the scalogram of a speech signal is presented in 2.4, where the derivatives have been approximated with finite differences and the congregation points are where the underlying

coefficients have a certain phase value (π in this case). These clustering of points is then represented by a rectangle with timespan equal to a child wavelet at the particular scale.



(a) Re-assigned scaleogram



(b) Scaleogram, reference

Figure 2.4:

- Visual improvement. It is questionable if there is any visual improvement.
- Flexibility. Re-assignment is non-flexible. Intermediate forms can be done by only taking into account one of the derivatives, however that is not the flexibility that was sought.
- Computational burden. Re-assignment involves several steps. Two set of derivatives and time-differences are calculated and lastly the congregation points have to be chosen.

Re-assignment in general will reasonably only work well when the underlying data is well separated (low level of interference between signal compo-

nents) already. This raises the question if it is motivated at all for interpretation purposes. The attempts to re-assign the coefficients in scale direction were motivated by the fact that such a re-assignment could possibly used for clustering of partials for tonal signals – in a sense segmenting the scalogram by creating a sinusoidal model of sorts. However, several interesting schemes have been developed for the purpose of sinusoidal modeling STFT [19], [20], [21] so any further work in this direction should start with evaluating those and other similar methods first.

2.2.2 Wavelet Ridges

In wavelet literature, much attention is given to the ridges corresponding to the maximum points in the scalogram [1], where they are said to give a representation of the underlying data. With narrow bandwidth filters the resulting scalogram is so smeared in time that the ridge points based solely on the maximum becomes a poor representation of the underlying data. This does not correlate with the ridges except well into the timespan of stationary signals (corresponding to the time support of the child wavelet). This is why using the phase derivatives becomes important.

Constructing a new time-frequency representation by moving and adding coefficients in scale direction does not void the final reconstruction summation in Eq. 1.15, it merely gathers the coefficients in partial sums for each time step. Thus, a re-assign method acting this way could potentially serve to both improve the readability while still be a valid representation of the signal in the reconstruction sense – and possibly decompose the signals in a meaningful way. Motivated by this some experimentation was conducted with combining the ridges and derivatives. For simple signals of only a few sinusoids, it is possible to decompose the components by moving all the coefficients to the closest ridge point that also fulfills a threshold on LGD (inter scale phase derivative) – if this value is small 'enough' the ridge point has true time support.

Result and Discussion

The result of an experimental algorithm that moves coefficients to local maximum points in scale direction that also fulfill a criterion on the phase derivative, is shown in Fig. 2.5. The signal is constructed from sinusoids and broadband noise and the result is presented in the form of scalograms.

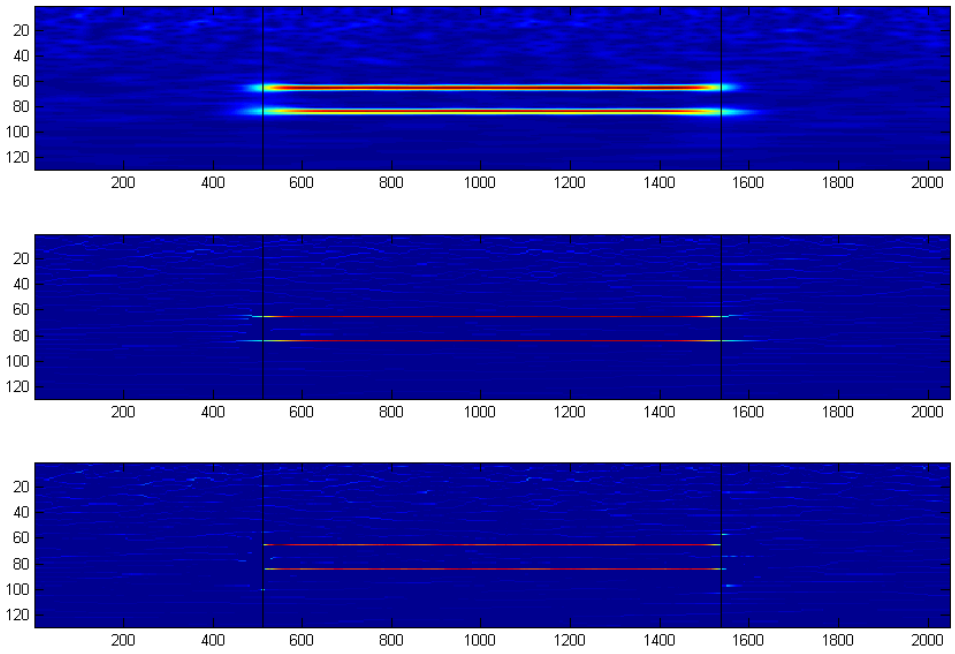


Figure 2.5: Scalogram, Ridge points and discarded Ridge segment reassignment. The true time support is shown by horizontal black lines. Taking advantage of the phase derivatives makes it possible to discard ridge points that are a result of time-smearing.

For simple signals it was possible to separate the signal components, however for more complex signals this method did not give a robust decomposition. It is believed that more work regarding how the coefficients are re-distributed could turn this into a fairly robust decomposition algorithm, but further work warrants also comparing to related decomposition methods and sinusoidal models.

- Visual improvement. Pure ridges is a poor representation, however exploiting the phase derivatives makes it somewhat more readable.
- Flexibility. Just like re-assignment, it is not flexible. It cannot be applied in intermediate forms.
- Computational burden. One phase derivative for every coefficient and the local maximum points in scale direction are computed for every time step. Some search method finds the closest peak that fulfills the phase derivative criteria.

2.2.3 Weighted Scalogram

To apply a re-assignment procedure after the wavelet transform requires many additional computations – two phase derivatives for every coefficient and then an efficient algorithm to cluster the re-assigned coefficients in a meaningful way. The benefit was not clear. A simpler, and more flexible solution was sought. This was found by using the values from the derivatives themselves, to create a weight function. A weighting function is proposed, constructed as:

$$W_{LGD}(t, a^j) = a^{-|(LGD(t,a^j))|^b},$$

where a and b control the amount of weight.

The LGD value is close to zero when near to the true support of a signal component. This means that the resulting weight W_{LGD} will be (almost) 1 on the true time support and (almost) 0 outside, effectively zeroing the components that do not belong to any true time support.

Result and Discussion

From the experimentation with re-assignment a simpler method is proposed, that to some degree achieves a similar result. Based on the inter scale phase difference, a weight overlay is suggested as a tool to aid the user interpretation of the scalogram. The degree of 'sharpening' can be altered seamlessly.

The visual result of using $a = e$ and $b = 1$ is presented in Fig. 2.6 showing that the visual effect of the weighting function on a speech signal.

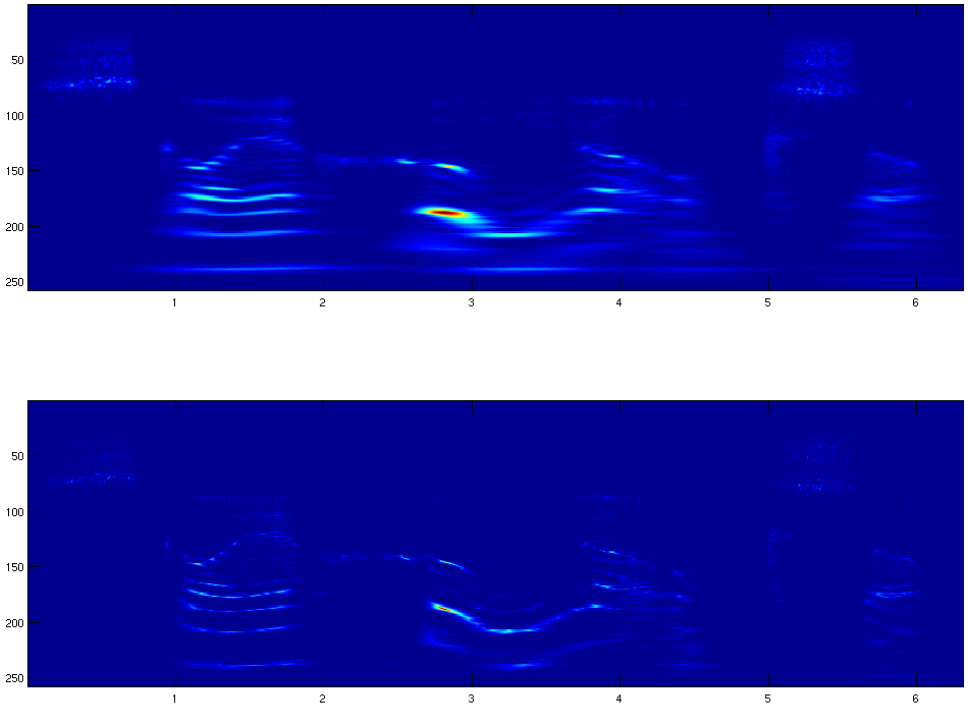


Figure 2.6: Top: Scalogram of a speech signal, Bottom: Weighted scalogram using the proposed method based on the phase derivative in scale direction.

As the W_{LGD} weight also punishes the coefficients near the edge of the time support of a component, the weight has to be used with moderation or there is a great risk to instead degrade the visual representation. However it is a very simple method that seems to be a very useful addition to a creative environment for getting a better idea of where the signal components true support are in the time-frequency plane.

- Visual improvement. Due to the weighting, the resulting scalogram is still smooth. This makes it easier to maintain readability. However, regions where signal components interfere can however show strange or misleading results. In this sense it suffers from the same problems as re-assignment.

- Flexibility. As this is a weight, the amount of weighting as well as the shape of the weighting function can be adjusted freely. In that sense it is very flexible.

- Computational burden. One phase derivative and a weight function based on an exponential, as well as a multiplication with said weight, is computed for every coefficient.

The smoothness combined with the flexibility makes this an interesting candidate for 'enhancing' the scalogram.

Using the IF value in a similar manner was tested briefly but the visual effect was only marginal, so it was not presented here.

2.3 Conclusion

From the experiments on 'refining' the scalogram, two methods stand out as promising: the combination of frames and the phase derivative weighting function.

The first is simply a combination of several time-frequency resolutions and while it is to the author's knowledge novel in the audio-scalogram context, filters with such shapes are used in engineering. The fact that the result is still a proper scalogram from which an inverse can be computed is attractive in the audio editing context. (Modifying coefficients as explained in the Introduction and Chapter 4)

The second method is the weight method proposed, that was inspired by re-assignment, but exhibits a much smoother and most importantly a flexible result. The amount of 'sharpening' is controlled by the weighting parameters. To the author's best knowledge, this type of weight have not been used before.

Re-assigning the scalogram did not impress the author. Partial summing towards ridges in the scalogram can prove to be useful with more work in regards on how to take advantage of the phase derivatives.

The final conclusion is that regardless of what method is used, the result will be poor if the signal is not well separated in the time-frequency plane.

Chapter 3

Audio interpolation

A challenging problem in audio restoration is to replace larger segments of noisy or damaged data with something meaningful. For instance, burst noise or unwanted signal components during a instrumental tone.

This chapter outlines a method that uses the Gabor Wavelet Transform for this purpose. Some other audio manipulation and restoration methods are covered in Chapter 4.

The performance was measured with a listening test.

3.1 Method

There are several ways to approach the problem as it is the perceived result that matters. Linear prediction [22] , [23], [24], [25], non-linear prediction [26], filter banks and sinusoidal modeling [27] are all examples on how this problem can be approached. Since the Gabor Wavelet Transform is a collection of narrow band pass filters, it should be possible to use the instantaneous information around a 'damaged' segment and fill in sinusoidal content in a way so that they fit the boundary values on phase and amplitude (and their derivatives) for every scale - just as can be done for every bin in the STFT case.

Interpolation of the instantaneous values

Eq. 1.13 shows how phase and amplitude values are obtained from the complex wavelet coefficients. The most straightforward way of using this information is to let sinusoids propagate from all scales and both ends of the segment and linearly interpolate these. If both ends have a very similar spectral content, this will work well. However, if there is an ever so slight phase shift or pitch shift there will unavoidably be undesirable interference occurring over the interpolated region. A solution to this problem is to interpolate the arguments themselves, i.e. phase and amplitude.

A small mismatch in phase or amplitude will give rise to a subtle, but very noticeable click. Therefore great care has to be taken as to avoid shifting any of the arguments. When considering an analytical signal with only one harmonic component, but a sharp transition, it is apparent the instantaneous values oscillate. Fig. 3.1 shows an example of this. In the case of filter responses from the scales the values can be assumed to be smooth enough, as the narrowness of the filters ensures a smoothness in time, so this is of no real concern in the interpolation procedure.

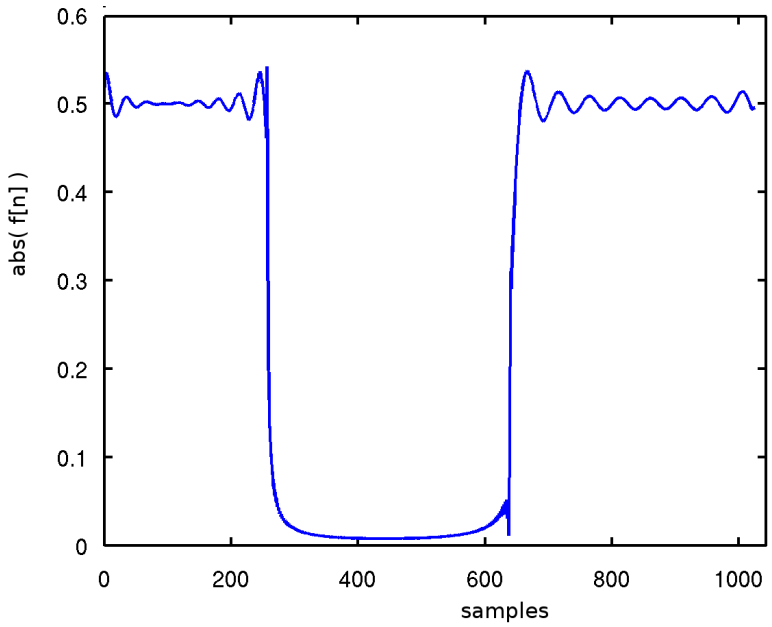


Figure 3.1: An illustration of oscillations in instantaneous amplitude as a function of time when calculated via the analytical signal form. The signal in question are two sinusoidal tones of constant amplitude, separated by a silent gap.

The wrapped phase issue

The phase values are known only in a modulus sense. Unwrapping over the region to get the relative phase offset will not work, as the coefficients have been heavily influenced by the damage in an unknown way - thus any attempt to interpolate such a phase will cause a shift in frequency over the interpolated region as compared to the boundaries.

In order to get an unwrapped phase that can be used, a few steps have to be taken. To simplify the notations, a wrapping operator is introduced:

$$W_{\angle}\Phi = \begin{cases} \Phi^{\star} - 2\pi & \text{if } \Phi^{\star} \geq \pi, \\ \Phi^{\star} & \text{otherwise} \end{cases} \tag{3.1}$$
$$\Phi^{\star} = \text{mod}(\Phi, 2\pi).$$

The target phase, $\hat{P_1}$, can then be derived as follows:

$$\begin{aligned} \hat{P_1} &= \bar{P_1} - \epsilon, \\ \bar{P_1} &= W_{\angle}P_0 + \Delta_t \frac{(\omega_1 + \omega_2)}{2}, \\ \epsilon &= W_{\angle}(\bar{P_1} - W_{\angle}P_1). \end{aligned} \tag{3.2}$$

Matching both phase and frequency

Assuming that a phase shift has occurred in the actual data (not caused by the damage) but both sides having the same frequencies, linearly interpolating the phase values will still cause a slight shift in frequency. Thus, a higher order interpolation has to be used to meet the boundary conditions on frequency. The design choice is to take into account the boundary values in frequency and phase, which means that the frequency must be allowed to drift slightly from the boundary values. This can be achieved by using the target phase from the linear case and fitting a third degree polynomial (there are 4 degrees of freedom):

$$\begin{aligned} P(t) &= w_0 t + \frac{b}{2}t^2 + \frac{c}{3}t^3 + P_0, \\ b &= \frac{6}{\Delta_t^2}(\hat{P_1} - W_{\angle}P_0 - \frac{\Delta_t(w_0 + w_1)}{3}), \\ c &= \frac{w1 - w0}{\Delta_t^2} - \frac{b}{\Delta_t}. \end{aligned} \tag{3.3}$$

The interpolated signal for one scale, on polar form, is then given by:

$$f_a^{\Delta_t}(t) = A^{\Delta_t}(t)e^{-iP(t)}, \tag{3.4}$$

where $A^{\Delta_t}(t)$ is the linearly interpolated amplitude arguments.

This lends itself to three slightly different ways of interpolating audio - the choice is in how the instantaneous values should be faded. Fig. 3.2 shows the result in waveform for interpolation between two sinusoidal segments that has a relative shift in frequency (and different phase offsets).

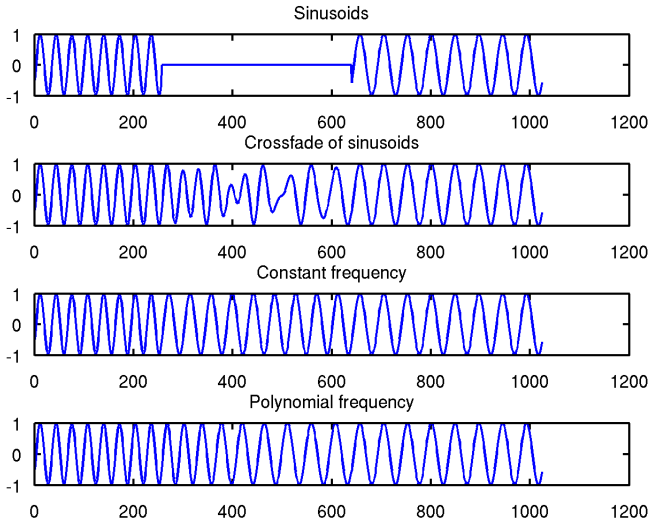


Figure 3.2: Top: Two sinusoids of different frequency and phase offset, with a silent gap between. Below: The silent gap is replaced by three different interpolation approaches using instantaneous values on amplitude, phase and frequency.

Amplitude interpolation limitations

When considering amplitude interpolation it is tempting to use a higher order of interpolation than a linear one when considering long segments. However, initial tests showed clearly that such an interpolation scheme for the amplitude causes very intrusive interference. This is explained by the relationship between scales being changed slightly so that interference gives rise to short tones. This point was not investigated further and amplitude interpolation was restricted to linear.

Time-frequency uncertainty

The next issue is related to the uncertainty principle. In order to get very narrow band signals in the scales the scales per octave parameter, v, has to be chosen fairly large. Since this fires back as an increased time smearing, the 'sample' points for the instantaneous values have to be moved further away from the actual damage. The distance in time for a given scale depends on its corresponding Child Wavelet's effective time support. In Fig. 3.3 an example of damaged sound is presented along with the result of the three interpolation methods.

If the damage is of high intensity and over the whole frequency region, it should prove beneficial to zero the affected samples in the time domain,

thus making the instantaneous values usable closer to the 'damage'.

The final method

The interpolation method given by Eq. 3.4 yields a method that should work well for semi-static sounds dominated by sinusoids, such as musical tones and portions of speech. Informal tests show that it works very well and can even work to some degree on chopped up speech signals (filling in the gaps). To investigate its feasibility a more formal listening test was conducted.

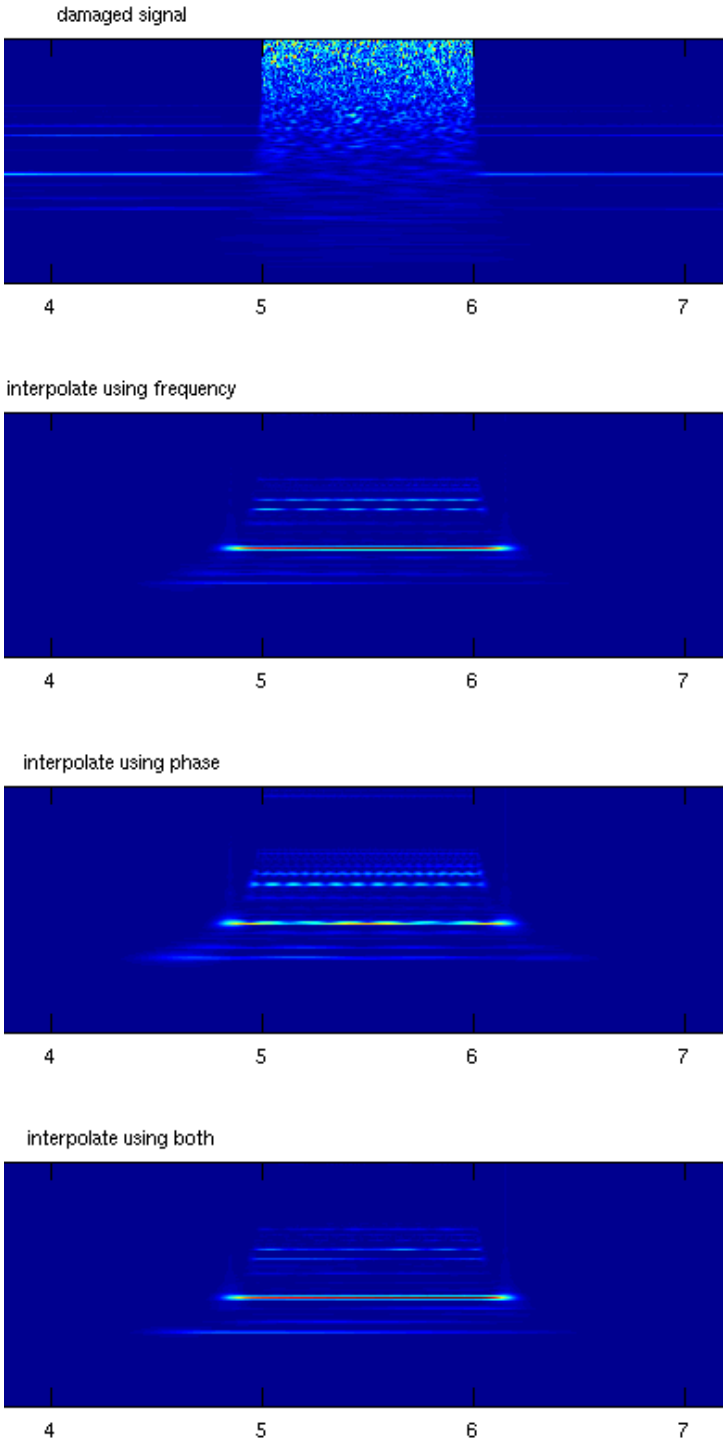


Figure 3.3: Top: A scalogram of a musical audio segment with added white noise added in a time segment. Below: The results of three different interpolation procedures using instantaneous values of amplitude and phase for all scales.

3.1.1 Listening test for performance evaluation

The first question to be asked when designing a listening test is what type of sound is to be used. Synthetic tests show that the method performs almost perfectly on simple sinusoidal combinations. It is also clear that a very rich sound containing various noise processes, transients, chirps etc will not work very well if the goal is to compare with a ground truth. With that in mind, the target was set for some sort of balance - a fairly static sound but not one constructed from synthetic signals. A sound was chosen from Creative Commons [28]. It is a polyphonic signal consisting of sampled signals played on a keyboard. The melody is played with a flute - thus it is not a perfect sum of sinusoids but rather a very narrow band process. Furthermore, there are traces of broad band noise, possibly caused by breath noise, with low energy. The target region was chosen to start at $27s$ (by the sample) into the clip as the transition between notes was to be avoided. Only the left channel was used to avoid the complexity with correlation, or lack thereof, between the stereo channels.

The actual evaluation method was chosen as a threshold test, usually used in psycho-acoustics for finding limits of perception [29]. The concept is to present the test subject with a set of sounds where one is different in some way. The test subject must listen carefully, and choose the one that is different from the rest. In order to rule out guesses, the process is repeated a number of times. If the consensus is that the subject can distinguish what sound is different, a new set with a smaller difference is presented. If the test subject chooses the wrong sound, a set with a larger difference is presented. The method counts the number of turns of right and wrong answers and an estimated threshold is produced.

Preparatory testing showed that an octave bandwidth of $1/32$ ($v = 32$) gave good results. To be sure, a 64th octave bandwidth was chosen. Distances from 10 samples up to 22000 samples were interpolated from the starting point. The limit was set at 22000 because longer interpolations did not make sense - if the method performed well at such a range that was evidence enough. Any longer ranges requested by the software would instead yield a completely different sound as to make sure that the testing procedure actually stopped.

The hypothesis was that some participants would get down to a thousand samples as any shorter interval was difficult to perceive even by the author. It also seemed very likely that some participants would have issues perceiving any but the longest intervals, as their hearing might not be trained for such small details.

3.2 Results and Discussion

Seventeen participants with varying degree of audio expertise participated. The result is presented in the form of a histogram over in Fig. 3.4. The fallout was such that no concise threshold could be drawn, most participants could not perceive the difference between the original and synthesized segment even for the longest interpolation length.

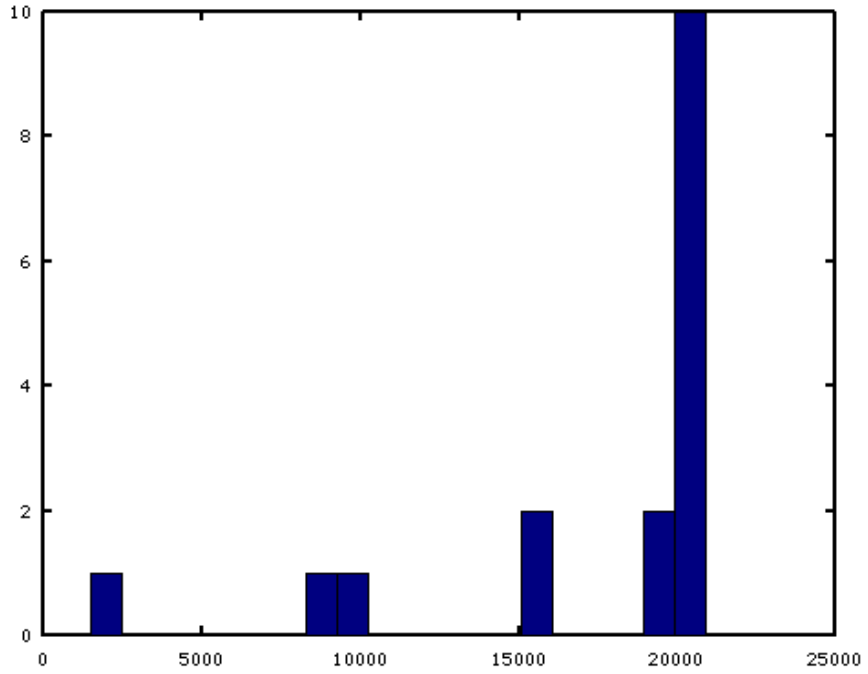


Figure 3.4: A histogram showing the result of the listening test. The x-axis is the number of samples the participants managed to distinguish as 'different'. The large cluster to the right are participants that could not even distinguish the largest length considered in the test.

The results were unexpectedly good for this particular sample - all but a few outliers had issues even with the longest interpolation region.

After asking the outliers how they were able to perceive the difference it was clear that they had perceived the gap in the low energy broad band noise for the interpolated segment.

The excellent result shows that the method works. However, some explanation for the result lies in the sound not being challenging enough and the bulk of test participants not having high audio expertise. However, the goal was to see if this worked at all. In real applications the choice is between a noisy burst, silence or an interpolation. In that perspective, this interpolation approach should be readily applicable.

The informal tests on speech also suggest that the method can produce

meaningful results even on portions of signals that are not strictly a sum of sinusoids.

A good complement for the sinusoidal interpolation is attempting to estimate any noise processes and to excite them as well over the interpolation region. This was left as a future addition as it was not clear as to how to tackle the problem, as the noise processes are generally unknown. Experimentations with thresholding the coefficients in order to extract some sort of sample of the noise process, and later adding this to the interpolated area, showed improved perceived results. If this could instead be done with linear prediction that estimates the parts of the signal best described by noise processes and somehow combines this with the interpolation procedure, filling gaps of more noisy data would be possible.

Furthermore, it must be said that even if this method proved successful, it could be improved to mimic the method constructed by Lagrange et al [27]. In this method a sinusoidal model is employed by tracking the partials over time and storing the instantaneous frequency and amplitude as separate time series for ever partial. With this approach, pitch derivatives, frequency and amplitude modulations become approachable. However, this assumes a robust sinusoid model technique and this seemed out of reach for the timespan of the thesis.

In closing it must be said that the approach of interpolating the arguments instead of two sinusoids from each direction might not always improve the result - sometimes an oscillating interference fits better to the neighboring regions.

3.3 Conclusion

The method that was developed worked very well. The result is a well performing and robust interpolation scheme for long gaps in audio. More testing can be conducted to see how well it performs for more challenging sounds, but in order to make such an investigation fruitful it should be compared to the performance of more elaborate methods. The author's belief is that in most cases this type of interpolation will work very well from a perceptual standpoint.

Chapter 4

Audio restoration and manipulation

In this chapter, two standard operations for audio editing purposes were developed and tested, using the Gabor Wavelet coefficients.

- Noise reduction
- Pitch-shift

Before covering those topics in details, we start off with some background.

As stated in the introduction the effort put into the thesis work is largely motivated by the wish to edit audio directly on in a time-frequency representation. The simple but perhaps most important operations, i.e. multiplications, are not a focus of the thesis but a few key points are pointed out here.

Editing the coefficients directly

Firstly, multiplying only some of coefficients belonging to a signal component may produce unexpected results. For instance, if editing outside the true time support of a single component, on the part belonging to the time 'smearing' of the filters, artificial components may arise as the coefficients no longer cancel out. For this particular example, the weighting method suggested in section 2.2.3 may be beneficial as a visual cue on what coefficients are on the true time support is shown more clearly.

Secondly, in order to get a WYSIWYG [1] editing environment the operations on a group of coefficients should adhere, roughly, to the time-frequency spread at those scales in the time-frequency domain. Windowing, with a smooth window, in time direction is essential or the sharp change will give rise to a very undesirable impulse. Since each scale is a band pass filter

[1] what-you-see-is-what-you-get

there is no such requirement in scale direction however in order to achieve WYSIWYG it may still be desirable. True WYSIWYG editing is only achieved when the edited time-frequency plane corresponds to a transformed signal.

4.1 Noise reduction by spectral thresholding

In this section a method for removing broad band noise by thresholding the Wavelet coefficients is presented. It is based on references [1], and mainly serve to show how such a thresholding procedure can be realized.

A common problem for home recordings is broad band electrical noise caused by the equipment, or faint background noise caused by fans, radiators etc in the vicinity. A common solution is to use a gate that zeros the sound when the intensity of the sound is under a certain value. A more involved approach to this is to use a frequency dependent threshold. This is possible in several ways, either based on a known threshold using the time-frequency representation at hand, or by using multiple transforms to get the 'best' coefficients that maintains the signal but suppresses the noise [1]. Using just one time-frequency representation is straight forward and required little extra work, so the experiment was restricted to this approach.

Method

In order to determine a noise threshold a segment of pure noise is needed. A profile of the spectral shape is approximated by finding the mean and standard deviations of amplitude for each scale over this segment. Once the profile is found, it is applied to the rest of the data using a soft threshold as a hard threshold introduces a risk of causing impulsive burst noise. Setting the threshold low will cause some of the noise coefficients to be unaltered, causing an annoying so called musical noise [1].

Let $f_j(t)$ be the complex coefficients at scale j and time t and $w(x)$ be a weight function. The segment of noise used to construct the threshold is denoted *training region*, and t_0 and t_1 are the start and stopping points, respectively, in time. $E_{j,train}$ is the mean value of the absolute value of the coefficients for scale j over the training region, and $\sigma_{j,train}$ is the standard deviation.

The thresholded coefficients are then given by:

$$f_{w,j}(t) = f_j(t) \times w(|f_j(t)|),$$

where,

$$w(x) = \begin{cases} 0 & x < w_0, \\ x\frac{w_1}{w_1 - w_0} & w_0 \leq x \leq w_1, \\ 1 & x > w_1, \end{cases} \tag{4.1}$$

and,

$$w_0 = E_{train} + A \times \sigma_{train},$$
$$w_1 = E_{train} + B \times \sigma_{train},$$

where A and B then controls the lower and upper threshold points.

The weight function is shown in Fig. 4.1.

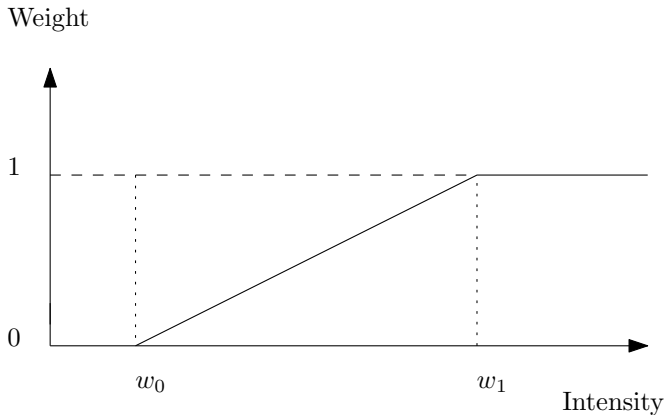


Figure 4.1: Smooth thresholding function, expressed as function of intensity. w_0 is the lower limit, and w_1 the higher.

Result and Discussion

A test using a simpler signal with added noise is shown in Fig. 4.2.

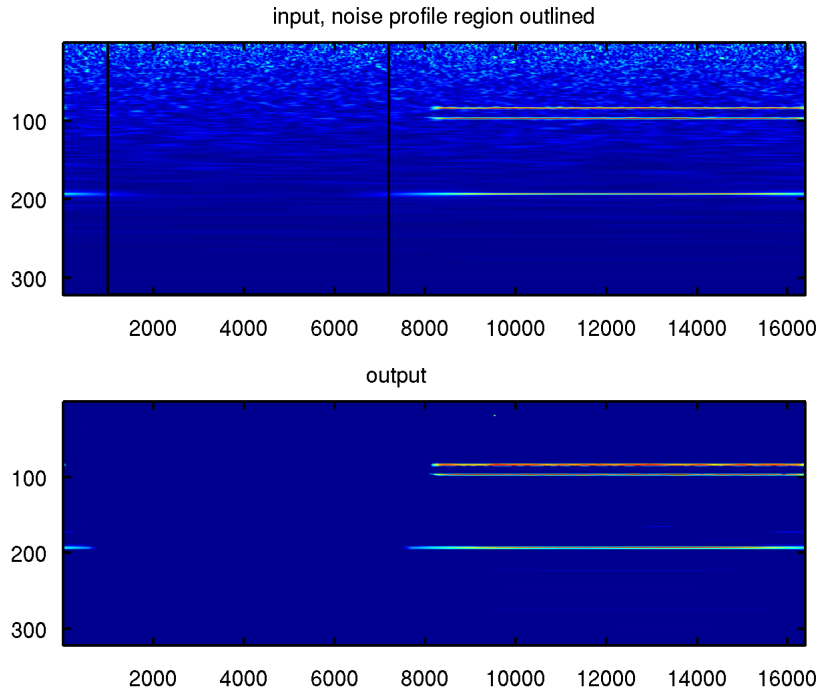


Figure 4.2: Noise removal by using a thresh hold on the coefficients based on an approximation of the spectral profile of noise.

Although the method was not extensively tested, it is believed that it will work well in an interactive environment. The parameters for controlling the upper and lower limit, as well as the way user chooses what region to be used for building the threshold, allows freedom and flexibility that should enable removal the influence of stationary noise processes. First the user selects a part of scalogram that is perceived as a good representation of the noise, then the resulting threshold limits are adjusted to produce a satisfactory result. Additionally for broad band noise processes, altering the time-frequency resolution towards worse time resolution may be beneficial as then slow moving components protrude clearer than fast moving components, such as the noise. This can be seen in Fig. 2.1.

4.2 Pitch-shift and time stretch

A very useful addition to a creative environment is the possibility to select regions in the time-frequency plane and make them more elongated/compact

in time and also, perceptually, moved to other frequency ranges. For instance, changing the pacing of speech while maintaining the pitch, or the other way around - change the pitch while maintaining the timing of the events. Commercial algorithms exists that does this in many ways using different transforms and techniques. The goal here is to see if the coefficients from the Gabor Wavelet transform can be used for this purpose.

Phase vocoder

The phase vocoder is an algorithm usually associated with the STFT. The idea is to exploit the possibility to separate an estimate of phase and amplitude for all signal components from the coefficients and modulate the phase. New coefficients are constructed that has a new instantaneous frequency corresponding to a pitch shift and computing an inverse using these a shifted signal has been constructed. Time stretch is constructed in a similar way, either by re-sampling a pitch shifted signal or by interpolating the arguments. This method is known to have problems with smearing of transients and 'reverberation' but the extent of this was not known if instead using the Gabor Wavelet.

Method

Since the instantaneous phase and energy can be approximated for every scale and sample in the Gabor Wavelet transform it is not unreasonable to think that modulating the values in the similar manner should produce, roughly, the same result. In order to highlight that this method is defined in the discrete case, the time domain is swapped for the sample domain, measured in n. Using the expressions from Eq. 1.13 a pitch shifted signal $f_{pitch}[n]$ from phase vocoder operation is calculated as:

$$f_{pitch}[n] = Re\left\{\sum_j A_j[n]e^{-ß\Phi_{pitch}[n]}\right\}. \tag{4.2}$$

where $\Phi_{pitch}[n]$ is the modulated phase.

In order to control the amount of pitch shift over time a function $p[n]$ is introduced. The modulated phase is then given as:

$$\Phi_{pitch}[n] = \sum_0^n \frac{d\Phi}{dt}[n]2^{p[n]}. \tag{4.3}$$

As example, $p[n] = 0$ is no shift, $p[n] = 1$ is an octave up, $p[n] =$ -1 is an octave down.

Eq. 4.3 was derived from the relationship between phase and frequency, shown in Eq. 1.13.

Achieving time-stretch and time-compression is done by re-sampling the arguments of amplitude and phase. The phase also needs to be modulated to compensate or the signal is simply re-sampled as a whole. Another option is to perform a pitch shift and then re-sample the result.

Result and Discussion

The method performed well on simple signals, and an arbitrary pitch shift was possible as shown in Fig. 4.2. The result for more complex signals is discussed and judged subjectively as no measure of quality was used, or comparison to other methods were made.

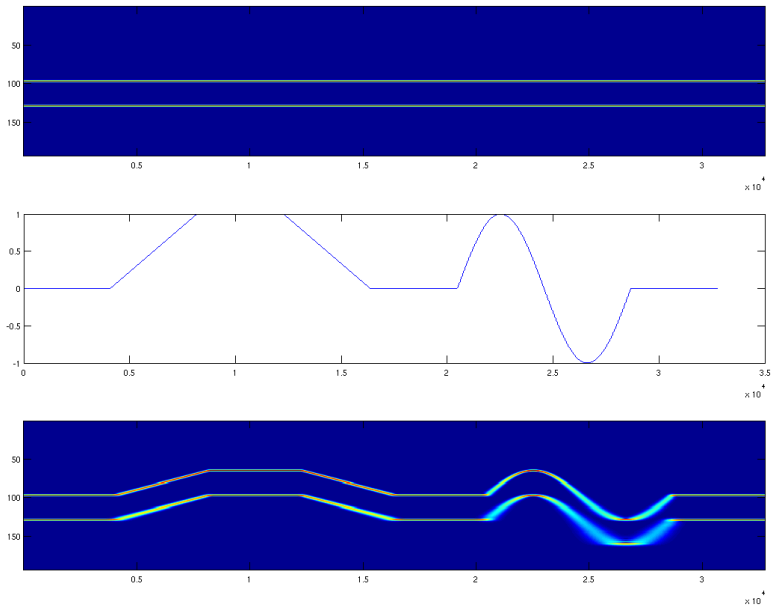


Figure 4.3: Pitch shift of signal consisting of two sinusoids. Top: Scalogram of original signal, Middle: Pitch function, Bottom: Scalogram of the shifted signal, using the pitch function.

Judging the performance is difficult. The best way would have been to compare with existing methods and use some sort of listening test. However, the author claims that the method will work well for simple signals, or signals dominated by musical tones. More complex sounds, especially speech, will suffer from interference artifacts that cause unwanted 'phasing' effects. The coefficients can be used for the purpose of pitch-shifting, but the result is far from state of the art.

The explanation for why these phasing issues occur is hard to find, but here follows an attempt at one.

For a simple signal like a sinusoid that is periodic over an interval the phase vocoder work very well. If the sinusoid has a distinct onset, then the unwrapped phase will not quite work.

A sinusoid will have most of it's energy in the scale whose center frequency is closest to the frequency of the sinusoid, but nearby bins will have a portion of the energy as well. In a small time segment around the onset the coefficients interfere in such a way that the onset is produced. Modulating the phase alters their relationship thus producing smeared transients. This means that any real world signal these problems arises all over the time-frequency plane when trying to modulate the phase, thus causing the mentioned reverberation and smearing issues.

The phase vocoder based on the STFT is used regardless of these issues. Even if no comparison was made between a phase vocoder using the Gabor Wavelet transform and using the STFT, it is believed that the scale dependent time support in the Wavelet transform makes these issues worse.

The ridge reassignment discussed in Section 2.2.2 was tested for pitch shift purposes, as modulating these re-assigned ridges are similar in spirit to locking the phase, a workaround found in literature for the STFT phase vocoder [30]. As expected the simple signals that were well separated by the approach could readily be shifted, the smearing issue was gone. This approach did not work very well for a more complex signal, like speech, however as problems with coefficients being assigned to wrong scales caused discontinuities in the instantaneous values. The result was induced impulsive burst noise.

It has been suggested that pitch and time-scale modifications can be performed on a ridge representation [1]. Since an approximate reconstruction of a signal from the ridges of a scalogram can be calculated via frame synthesis, it is suggested that modulating the phase over the ridges should produce a shifted signal. This was never tested, as it was believed that even if the ridge inversion worked, the modulated ridge would not be the same as the 'target' signals ridge - inter scale interference is likely reduced but the time smearing would still be present. No evidence in literature was found that showed that the method did perform well.

Using a ridge based method could maybe produce passable results but that requires a more work in how the ridges are constructed.

4.3 Conclusion

In this chapter, the applicability of using the coefficients, from the Gabor Wavelet transform, were investigated on two common audio manipulation applications – noise thresholding and pitch-shift.

Using the Gabor Wavelet scalogram to threshold noise seems readily applicable. The suggested way of estimating a noise profile and using this as a soft threshold are simple but effective.

Pitch-shift and time-stretch using the coefficients via a phase-vocoder produces passable, but not impressing, results. Taking into account that both faster and more pleasing algorithms exists it is not seen as beneficial to use the coefficients for this purpose.

Chapter 5

Computational aspects

This chapters covers two central questions when calculating the Gabor Wavelet coefficients: reconstruction error and computational speed. The last section outlines a novel 'compression' algorithm that shrinks the size of the scalogram *drastically* while maintaining perceptual quality.

Please be reminded that when referring to the Gabor Wavelet, it's the result of the 'analysis' filter-bank and 'reconstruction' filter bank combined, as explained in Section 1.2.3. Furthermore, we refer to the coefficients as the results of Fourier multiplications with the Child wavelets.

The Fourier multiplication approach of calculating the coefficients is equivalent to a circular convolution in the time domain. When considering a longer signal it is not feasible to perform this operation for all samples due to the high memory requirement - rather the signal has to be split into computational blocks. Due to how the circular convolution wraps around in time extra samples, corresponding to the time support of the wavelet, has to be included before and after the wanted block. (This is mostly a visual aspect as the coefficients should still sum correctly in the inverse.) Consider Eq. 1.2, the equation for a Child Wavelet at scale j. For a scale j the extra samples needed are a number of σ_{a^j}. Due to the notation used for the Gabor Wavelet based filter bank, the standard deviation in time for scale j is $\sigma_{t,j} = \sigma_{a^j}/2$. A reasonable suggestion is time support of 3 to 4 times $\sigma_{t,j}$ to minimize this wrap-around effect.

5.1 Reconstruction error

In this section the relation between reconstruction error and frequency redundancy is investigated, as well as the construction of the so called residual filters.

Theoretically, the equations listed in section 1.2 will allow perfect reconstruction. However, as mentioned in section 1.2.3 the Gabor Wavelet Transform is not, at least when using a Gaussian, a proper tight frame. The

remedy is then to have more 'overlap' between the filters. How much more that is *enough* is not clear, so this is investigated in this section. Ideally, this frequency redundancy should also be unrelated to the choice of parameter v.

Consider this simplified expression for the Child Wavelet:

$$\hat{\psi}_j(\omega) = Ae^{-\frac{(a^j\sigma)^2(\omega-\frac{\eta}{a^j})^2}{2}},$$

where A is a constant and $a^j = 2^{\frac{j}{v}}$.

We want v to be the design parameter. It controls the number of 'voices per octave', that is the spacing of the filters on a logarithmic frequency axis. This should then naturally control the total number of filters. Likewise, it should govern the value of σ so that the filters overlap enough.

The question then is how σ and v should be related so that the end result is as close to a tight frame as possible. As a starting point, it was assumed that σ can be chosen approximately proportional to v - a change in the number of scales per octave should reasonably also change width of the Gaussian bells proportionally. If σ is chosen as $\sigma_n(v) = v/n$ then a larger n results in more overlap between the filters. This leads to the final definition of the Child Wavelet:

$$\hat{\psi}_j(\omega) = Ae^{-\frac{(a^j\sigma_n(v))^2(\omega-\frac{\eta}{a^j})^2}{2}}, \tag{5.1}$$

where A is a constant,
$a^j = 2^{\frac{j}{v}}$, and,
$\sigma_n(v) = v/n$, where n is 'large enough'.

A measure of the error is then the magnitude of oscillation in the passband, explained by the total contribution of the filters not covering the frequency axis evenly, there's a distinct drop between the center frequencies of adjacent scales. This oscillation was measured with standard deviation as an indication of the reconstruction error, and is presented in table 5.1, for a few different choices of n.

Besides the flaw in the passband of the filters, the filter bank viewpoint requires two residual term filters in order to cover the whole frequency axis evenly. One is the concatenation of all lower scales mentioned in literature - the scaling function in Eq.1.15. Another term is needed to take care of the high frequency scales not summing to a constant close to the Nyquist frequency. The details of this is depicted in Fig. 5.2 and Fig. 5.1 respectively. Modifying the reconstruction formula, defined in Eq. 1.15, we get:

$$f(t) = Re\left\{\sum_0^J f(t)_{a^j} + \phi_J^\star + \phi_{high}\right\},$$

where $\phi_J^\star$ is the low frequency residual, and,
ϕ_{high} is a high frequency residual.

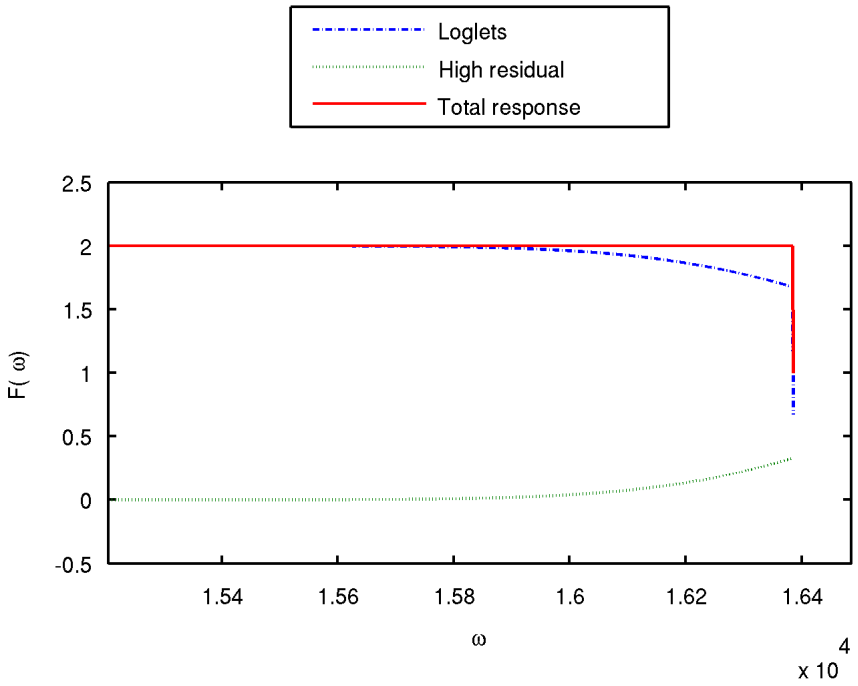


Figure 5.1: Detail of the crossover to high frequency residual

The formal definition of C_Ψ and the low frequency residual suggests that they can be derived from the integral in Eq. 1.7. Attempting this proved fruitless as to the author's knowledge the integral does not converge for the Gabor Wavelet case. In the end the formal definition was skipped. Instead, all the filters were scaled so that they would sum to 2. That is, the sum of all the filters is an all pass filter as given by Eq. 1.11. This new scaling factor is the constant A in Eq. 5.1. The residual filters were then simply the difference between 2 and the sum of the Gabor Wavelet filter bank, and split into two filters to separate the low and high residuals. Where to split is fairly arbitrary - any point in the passband should do.

In summary, in order to construct a Gabor Wavelet filterbank a few simple steps are taken:

1. Decide upon the parameter v, and the overlap factor n.
2. Compute the Child Wavelets in fourier domain using Eq. 5.1, ignoring the constant A.

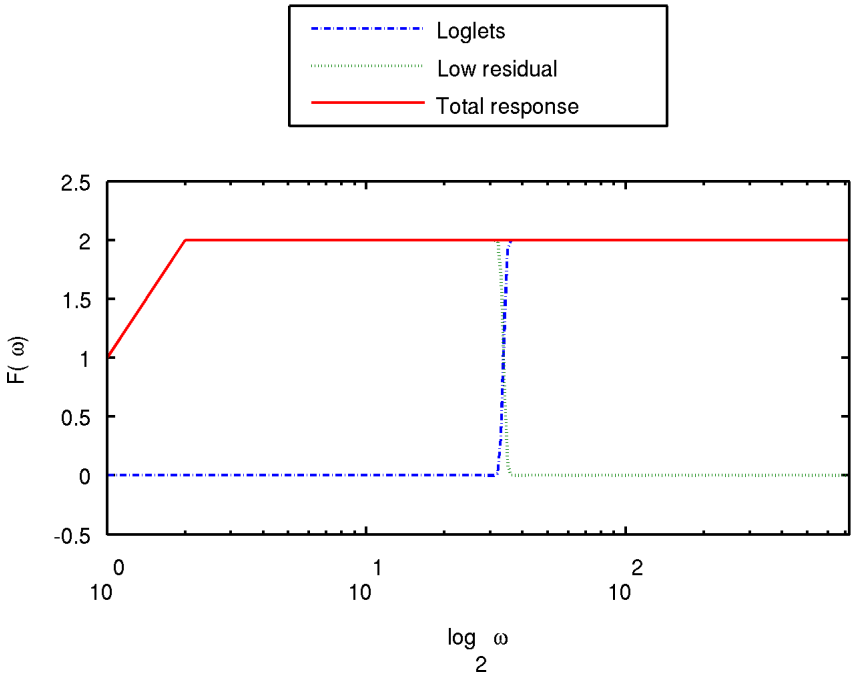


Figure 5.2: Detail of the crossover to low frequency residual

3. Sum the Child Wavelets and get the maximum, $B = \max \sum_j \hat{\psi}_j$.
4. Find A, by $A = 2/B$. Multiply the child wavelets with A.
5. Get the residuals, by $\hat{\phi}_J + \hat{\phi}_{high} = 2 - \sum_j \hat{\psi}_j$. Optionally split into 2 residuals.
6. Make sure that the sum of all these filters fulfill Eq. 1.11.

5.1.1 Loglet based filters as an alternative

The reason why the Gabor Wavelet does not allow perfect reconstruction can be described quite intuitively. A filter bank, with narrow band pass filters of compact support, placed on a logarithmic scale requires the filters to be symmetrical on a logarithmic scale. The Gabor Wavelets are not of this shape. In order to achieve perfect reconstruction some other filter shape has to be found. One such filter is the *Loglet* [31], originally derived for image analysis purposes. Only the radial part is relevant when considering one-dimensional signals, and is given by:

$$R_s(\rho) = erf(\alpha \log \frac{\beta^{s+\frac{1}{2}}}{\rho_0}\rho) - erf(\alpha \log \frac{\beta^{s-\frac{1}{2}}}{\rho_0}\rho), \tag{5.2}$$

where,
s is the scale number, equivalent to $j-1$, β 1 is equivalent to a in Wavelet notation, and α denotes the filter shape or overlap.

The resulting filter bank will, for $\beta > 1$, constitute a tight frame [31]. Large choices of α will make the filters overlap more and choices of $\alpha < 1$ make them more square shaped.

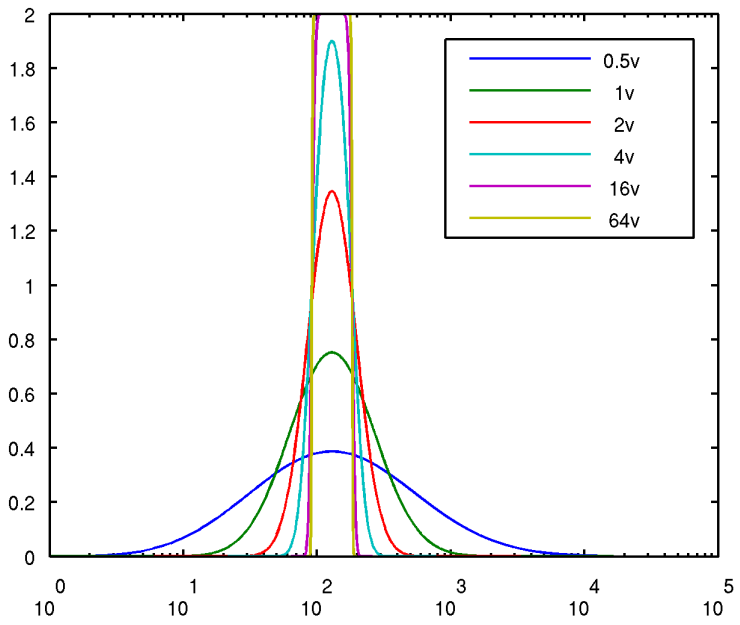


Figure 5.3: Loglet with different choices of $\alpha = n \times v$, resulting in differently shaped 'bells'. Regardless of this choice, the Loglet based filters still constitute a 'tight frame'.

The analytical time and frequency support for any choice of α is needed, or at least a good analog to it, in order to be able to make a comparison of redundancy and error properties of the Gabor Wavelet and Loglet. Since this was not given by the authors, this was instead approximated by experimenting.

When choosing the parameters as $\alpha = 2v/log_2(n)$ and $\beta = 2^{j/v}$, the resulting filter bank resembles the Gabor Wavelet with $\sigma = v/n$. As this relationship was found through experimentation, it can only be said to hold for the ranges of design parameters tested. Fig. 5.4 and 5.5 displays the time and frequency support of these two choices of Gabor Wavelet and a Loglet filter using the proposed relationship.

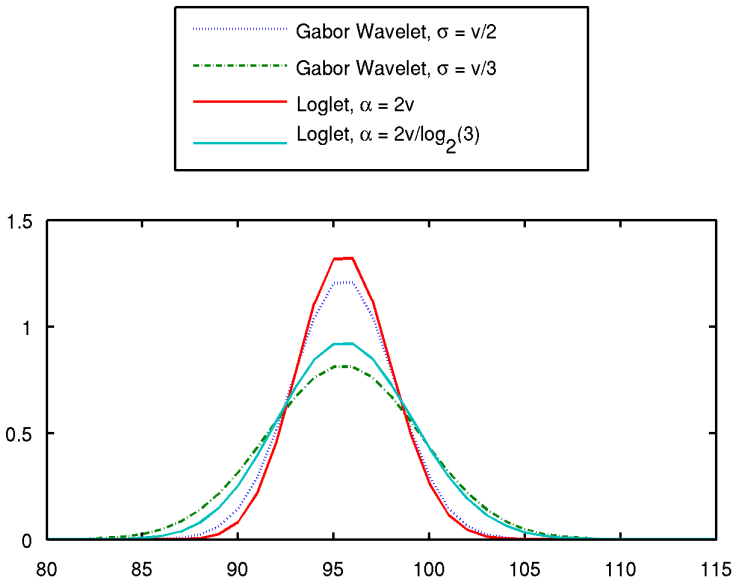


Figure 5.4: Two Loglets compared to two Gabor Wavelets in frequency domain, testing the suggested relationship of the Loglet and Gabor Wavelet parameters.

This shows that the choice of $\alpha = 2v/log_2(n)$ lends a time support slightly wider than a Gabor Wavelet with $\sigma = v/n$, and a frequency support slightly narrower for the main bell.

To compare the reconstruction properties to that of the Gabor Wavelet transform, the standard deviation over the passband of the sum of the filters was computed and is presented in table 5.1.

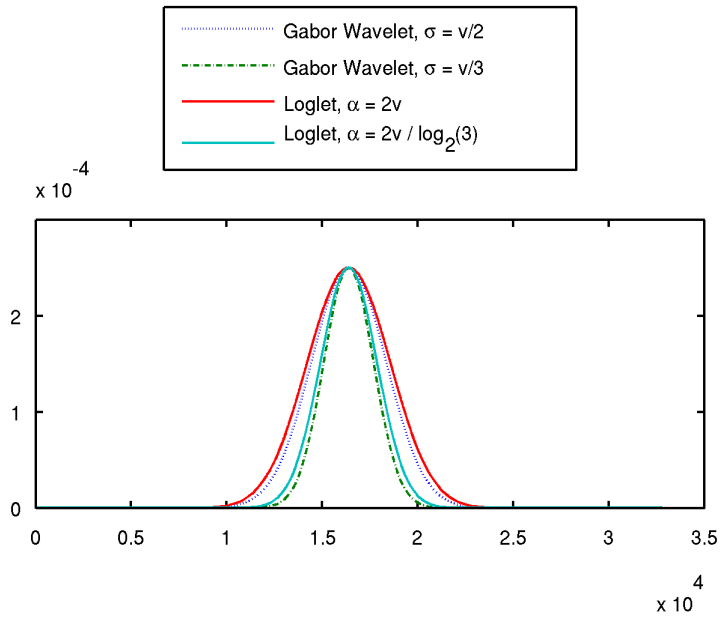


Figure 5.5: Two Loglets compared to two Gabor Wavelets in time domain, testing the suggested relationship of the Loglet and Gabor Wavelet parameters.

Result and Discussion

A simplification of the calculation Gabor Wavelet related definitions was proposed that will make sure they sum correctly to achieve good reconstruction, relying on the definition of a perfect filter bank.

The reconstruction error was investigated by deriving the standard deviation of the passband of the filters, for a few choices of n, to see the impact of the overlap amount. A Loglet filter bank was included for comparison. This is presented in Table 5.1.

Gabor Wavelet			Loglet
$\sigma = v/2$	$\sigma = v/2.5$	$\sigma = v/4$	$\alpha = 2v$
7.85e-04	1.08e-05	9.00e-08	1.37e-16

Table 5.1: Error in the passband for filter banks expressed as the standard deviations, as a measure of the oscillation cause by the filters not overlapping to sum to a constant. The numbers should not be taken as a direct measure of the error, but rather as an illustration of the impact of increasing the overlap.

The proposed simplification to force the Gabor Wavelet Transform to sum to 2 will not minimize the error - for that purpose scaling so that the summation oscillates around 2 would be a better option. However, discarding the Gabor Wavelet completely in favor of the Loglet based filters renders the reconstruction errors virtually negligible.

The proposed relationship is in a way oversimplifying the situation, as the bell tails are skewed meaning that the Loglet will have a much longer spread towards higher frequencies as compared to a Gabor Wavelet of similar shape. That makes the analog to a standard deviation weak, at least in frequency domain. It can still be used, in fact there is not any choice, but it is important to point this out if considering the down sampling scheme in the next section.

5.2 Implicit down sampling

Since the filters are narrow band pass filters, they have almost zero energy at 4 $\sigma_{j,f}$ from their center frequency, where $\sigma_{j,f} = \frac{2}{\sigma_j}$. This means that they can be down sampled without any great risk of aliasing effects using a Nyquist frequency slightly over this frequency. If sticking to down sample ratios of 2, the scales can be down sampled as depicted as in Fig. 5.6. The benefit is a substantial decrease in memory needed to store the resulting coefficients, as well as fewer operations to compute them.

If the time blocks are chosen as a power of two, the subsequent down sample blocks will also be power of two, and thus the FFT computation will

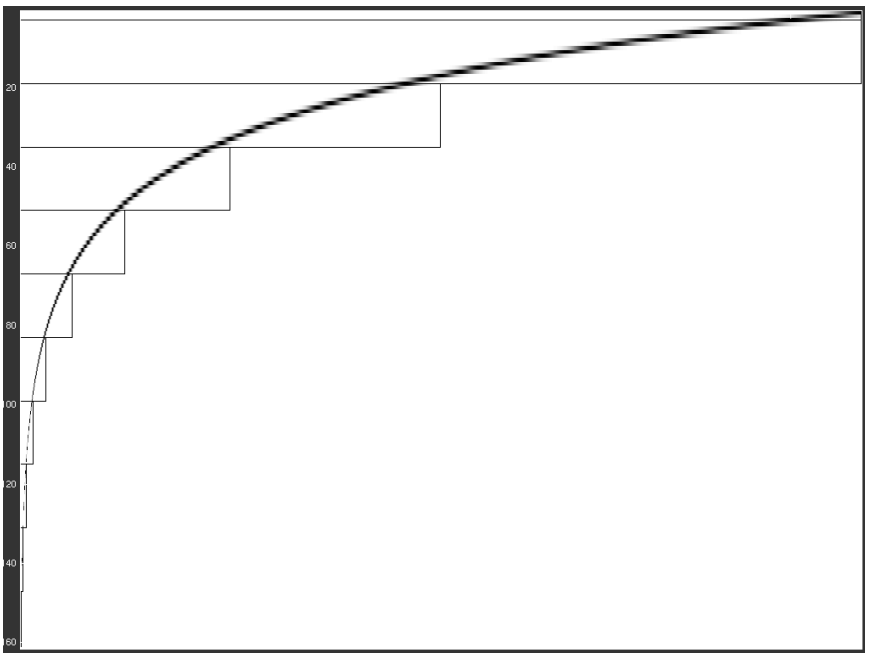


Figure 5.6: Visualization of the scales frequency support. The Gaussian bells for the different scales can be seen as the black curve. The boxes show how these can be implicitly down sampled, thus the area to the lower right is proportional to the memory and operations saved in the ideal case.

be fast. (Other prime number ratios are possible too.)

In order to get the coefficients at all sample points the down sampled block has to be up sampled before summation. All the scales down sampled with the same factor can be summed and up sampled together. This can, with some care, be done in Fourier domain as well by padding the DFT of the down sampled coefficient signal with zeros.

The problem is the transient that inevitably will occur on the edges of the summed signals due to them not being periodic of the length of the block - padding the DFT with zeros will not recreate an up sampled transient. If taking enough samples before and after the block the influence of the transient is however negligible. Other options are up sampling procedures such as convolving with reconstructing filters.

Result and Discussion

Some preliminary numbers on computational speed-up reported from Sonic AWE [3] is presented in table 5.2.

With rendering	
No downsampling	2.52038×
With downsampling	40.6933×
No rendering	
No downsampling	13.837×
With downsampling	50.590×

Table 5.2: The speed gain of using the suggested down-sampling scheme when calculating the Wavelet transform. The numbers are reported from implementation in Sonic AWE [3] and expressed as factor faster than real time. The specific test was made using one channel of audio with sample rate of 44 kHz, 16 bit (CD quality) and with 40 scales per octave and 10 octaves.

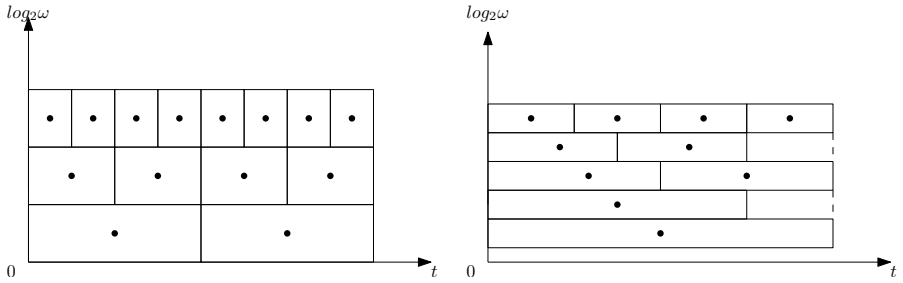
The block wise down sample procedure proved very beneficial to the real world implementation in Sonic AWE [3]. The numbers in table 5.2 are illustrative, as the results are implementation and hardware dependent, but they clearly show that is a very important result.

There are other candidates for fast calculation. Especially the Oblique Projection [32] seems worth investigating closer as they boast $O(N)$ complexity, however such a comparison is incomplete without also taking into account the relative reconstruction error as the referenced methods relies on approximations of the filters. In other words, compare the aliasing errors introduced with this scheme to their approximation errors as well as the speed gain in both methods. This was not seen as a priority by the author or the developers of Sonic AWE [3] and was left as future work.

5.3 Compressing the scalogram

If returning to the issue of the theoretical sparseness of the coefficients, it is possible to create a complete representation with a time step proportional to the octave bandwidth. This is called sub sampling in some literature. A narrower bandwidth will move the critical time spacing of the coefficients further apart, as shown in Fig. 5.7. Such a sparse representation can be cached for very long segments of audio as the amount of time redundancy is much lower, thus the number of data points in total is dramatically lower. In literature how to do this was only found for Wavelet frames with $v = 1$. Here v is higher, meaning that the resulting time steps will not be integers but fractions which makes it impossible to use the fast wavelet scheme [1].

A method similar in spirit is derived here, developed when dealing with the re-assign and pitch shift problems (see previous chapters for these problem statements).



(a) Time frequency spread of a Wavelet, v=n

(b) Time frequency spread of a Wavelet, v=2n

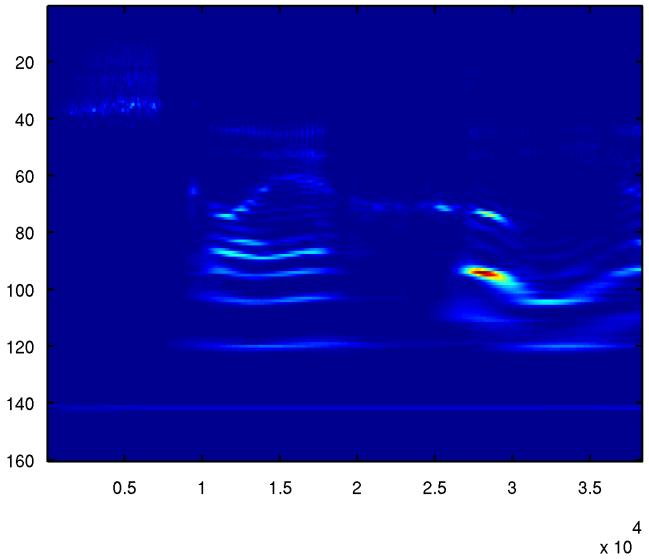
Figure 5.7: Illustrative Heisenbergbox representation of two different Wavelet Frames. The dots indicate the minimum sample points needed over the time-frequency plane. Twice as dense in scale direction means twice as sparse in time.

The proposed method is to only express phase and energy for each scale at a certain phase cycle, and every n:th cycle. So, it is in a sense signal dependent sub-sampling as these are signal dependent. The procedure first finds each point for each scale for a certain phase angle. Numerically π can be found with ease, as it can be located by finding the discontinuity in the phase derivative. Once this point has been found, the timing of the wrap from $-\pi$ to π is located more precisely via interpolation. The corresponding interpolated time and amplitude is then stored. Now each scale is expressed once per cycle with two values, thus information density is in a sense the same to a critical down sampling. As the next step, all but every $\sigma(v)/m$ of these points for each scale are discarded. The choice of m depends on the choice of $\sigma(v)$.

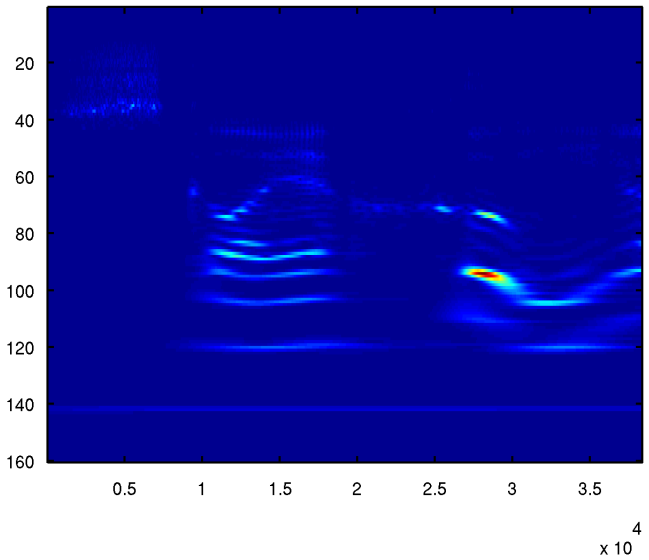
When reconstructing the scales, the values of phase and energy can now be interpolated from the two neighboring π-phase points. The assumption being made is that the the amplitude and frequency move slowly between these points, thus a linear interpolation should work. Since the amplitude envelope depends on the shape of the wavelet in time domain, in this case a Gaussian, some higher order interpolation could improve the result. Both linear and spline interpolations were tested for phase and amplitude. Spline interpolation for phase introduced interference terms, so this option was discarded and is not included in the comparison.

Such a scalogram is presented in Fig. 5.8 and an illustration of the error is presented in Fig. 5.9. Notice how similar the scalograms are. The error is presented in absolute terms, it's the difference between the normal inverse and the compressed inverse. Even if it seems to be fairly large, the

perceptual difference between the normal inverse and the compressed inverse was minimal (the author could not tell the difference).



(a) Scalogram synthesized from approximate sub sampling, effectively using approximately 3N datapoints



(b) Scalogram, reference, using the full $2 \times 16 \times 10 \times N$ datapoints.

Figure 5.8: Two scalograms of the same speech signal.

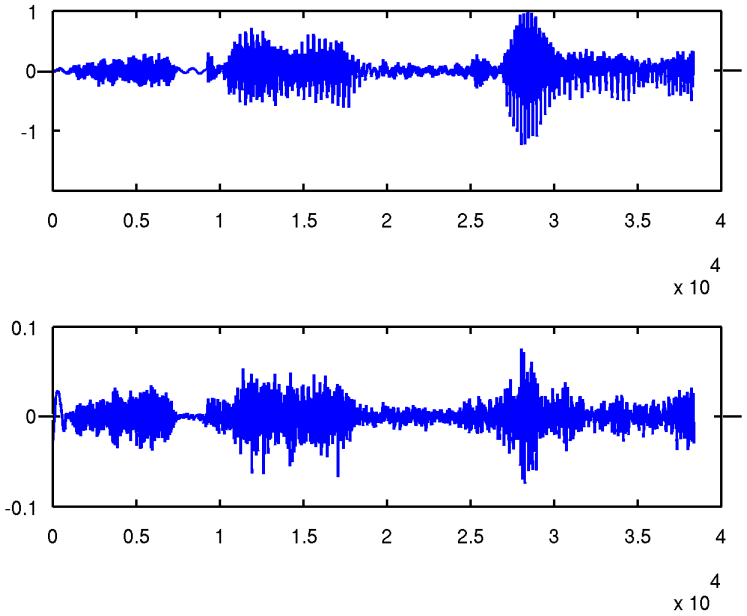


Figure 5.9: Top: Normal inverse waveform, Bottom: Error of reconstruction from approximate sub sampling using approximately only $3 \times N$ data points. Even if the error is numerical large, the author could not distinguish between them. The original inverse was constructed from $2 \times 16 \times 10 \times N$ data points.

Result and Discussion

An compression scheme for the scalogram was developed, reducing the number of data points needed as far as $3N$, where N is the amount of samples in the transformed signal. Compare this to the usual scalogram that uses $2N \times v \times O$ data points, where O is the number of octaves. Here $O = 10$, which is a common choice to cover the whole audible frequency range.

The numerical error introduced by the approximation scheme is presented in table 5.3. To make sure the scheme was unaffected by the choice of v, two tests were made with $v = 16$ and $v = 32$. Additionally, a comparison was made using linear and spline interpolation for the amplitude argument.

spline amplitude		linear amplitude		
std	max	std	max	$\times N$
v=16				
0.0122	0.0754	0.0166	0.1146	2.9487
0.0050	0.0370	0.0059	0.0373	5.8952
0.0038	0.0315	0.0039	0.0316	11.788
v=32				
0.0131	0.0628	0.0192	0.1202	2.9181
0.0075	0.0455	0.0087	0.0456	5.8320
0.0048	0.0441	0.0050	0.0441	11.660

Table 5.3: Reconstruction errors for sub sample approximation scheme and resulting effective redundancy. This test was performed on a speech signal, using 10 octaves, with two different choices for v. Notice how the effective redundancy (total number of data points needed), $\times N$, is unrelated to the choice of v. The error is the difference between the normal inverse and the inverse from the 'compressed' scalogram, and presented in terms of it's maximum value as well as standard deviation. Using spline interpolation, the error is somewhat smaller, but not drastically so.

The errors induced by this approximation scheme are mostly related to the assumptions of phase and amplitude not holding around transients. Even if the errors are non-negligible numerically informal listening suggests that they are perceptually negligible, thus using the approximated scalogram could be a strong candidate for preview purposes as the memory requirement is drastically lower. The cost is extra initial computation and interpolations when computing the inverse. For every point in the scalogram linear or spline interpolation is needed for the energy argument. For inversion, the phase and the evaluation of the cosine using this multiplied with the energy argument is needed for every time step and scale.

A proper sub sampled Scalogram could be achieved computing the coefficients from the definition in Eq. 1.6 with fractional delays in ψ_j to achieve

the non-integer sampling grid. This was never implemented however so no comparison can be made on the speed and reconstruction properties.

5.4 Conclusion

In this chapter three main topics were covered.

Firstly, it was shown how to design the Gabor Wavelet filter bank to both be adjustable in terms of the time-frequency ratio while keeping the reconstruction error low. Furthermore, it was discovered that better alternatives exist, such as the Loglet. The superiority of the Loglet for reconstruction properties was shown numerically.

Secondly, a down sampling scheme was introduced which decreased the computational time drastically. In a real world implementation this amounts to around 20 times increase in speed, compared to without using the scheme.

Lastly a 'compressing' scheme was suggested. While not a compression in the normal sense, it compresses the scalogram to a *fraction* of it's original size. Reconstruction is far from perfect, but perceptually identical by the author's judgment. This scheme will be beneficial in scenarios where computing the scalogram is expensive, and there are memory constraints. The 'compressed' scalogram can then be cached, and used for audio editing or analysis purposes, freeing up substantial amounts of $<$memory.

Acknowledgements

I want to thank a number of people that supported me in various ways during my work with the thesis.

Arne Leijon at *KTH Speech, Kungliga Tekniska Högskolan*, for letting me use his unpublished code for the listening test.

Johan Gustavsson at *Sonic AWE* for ideas and support, and his parallel work implementing stuff from the thesis. Most notably the numbers from an real world implementation of down sampling scheme!

I extend a thanks to the people at *Centre for Image Analysis (CBA), Uppsala University*, for great company and rewarding discussions - I throughly enjoyed my time there. Notably my mentor **Erik Wernersson**, and my supervisor **Anders Brun**! I also want to extend a special thanks to **Milan Gavrilovic**, **Khalid Niazi**, **Filip Malmberg**, **Robin Strand** and **Chris Luengo**, for valuable discussions and advice in different stages of the thesis.

I would also like to thank **Adrian Bahne**, **Lars-Johan Brännmark** as well as **Thomas Olofsson** at *Signals And Systems, Uppsala University* for ideas and discussions. Special thanks to Adrian for the suggestion on the listening test procedure!

Thanks to **Hans Knutsson** and **Mats Andersson** for the opportunity to present my ideas and the suggestions you gave. And naturally, thanks for inventing the Loglet! The name choice might be questionable, but the filter is excellent!

Finally, the reason I chose these topics in the first place: **Sonic AWE**. Thanks to my colleagues in the project!

Bibliography

[1] Stephane Mallat. *A Wavelet Tour of Signal Processing - The Sparse Way*. Academic Press, 2009. ISBN: 978-0-12-374370-1.

[2] Philip Denbigh. *System analysis and Signal processing*. Addison-Wesley, 1998. ISBN: 9780201178609.

[3] www.sonicawe.com. Sonic AWE, 2010.

[4] Richard F Lyon. Machine Hearing: An Emerging Field. *IEEE Signal Processing Magazine*, 27:131–139, September 2010.

[5] Anders Vretblad. *Fourier analysis and its applications*. Springer, 2003. ISBN: 0-387-00836-5.

[6] Jelena Kovačević and Amina Chebira. An Introduction to Frames. *Foundations and Trends in Signal Processing*, 2(1):1–94, 2007.

[7] Petre Stoica and Randolph Moses. *Spectral Analysis of Signals*. Pearson Prentice Hall, 2005. ISBN: 0131139568.

[8] Leon Cohen. Time-Frequency Distributions - A review. *Proceedings of the IEEE*, 77:941–981, July 1989.

[9] S. Qian and D. Chen. Understanding the nature of signals whose power spectra change with time. *IEEE Signal Processing Magazine*, 16(2):52–67, 1999.

[10] Karlheinz Gröchenig. *Foundations of time-frequency analysis*. Birskhäuser, 2001. ISBN: 0-8176-4022-3.

[11] Stanley Smith Stevens, John Volkman, and Edwin Newman. A Scale For The Measurement Of The Psychological Magnitude Of Pitch. *Journal Of The Acoustical Society Of America*, 8(3):185–190, 1937.

[12] Ingrid Daubechies. Wavelets and Applications. In Timothy Gowers, editor, *The Princeton companion to mathematics*, chapter VII.3. Princeton, 2008. ISBN: 978-0-691-11880-2.

[13] Ingrid Daubechies. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. In *IEEE Transactions on Information Theory*, volume 36, pages 961–1005, 1990.

[14] Ole Christensen and Richard S Laugesen. Approximately dual frame pairs in Hilbert spaces and applications to Gabor frames. *arXiv: 0811.3588v1*, 2008.

[15] K.R. Fitz and S.A. Fulop. A Unified Theory of Time-Frequency Reassignment. *arXiv:0903.3080v1*, 2005.

[16] D Nelson. Instantaneous Higher Order Phase Derivatives. *Digital Signal Processing*, 12(2-3):416–428, 2002.

[17] Sylvain Lagrange, Mathieu; Marchand. Estimating the instantaneous frequency of sinusoidal components using phase-based methods. *J. Audio Eng. Soc*, 55(5):385–399, 2007.

[18] Timothy J Gardner and Marcelo O Magnasco. Sparse time-frequency representations. *Proceedings of the National Academy of Sciences of the United States of America*, 103(16):6094–9, April 2006.

[19] Michael Klingbeil. Software for spectral analysis, editing, and synthesis. In *Proceedings of the International Computer Music Conference*, pages 107–110. Citeseer, 2005.

[20] Teresa H. Y. Meng Tony S. Verma. Extending Spectral Modeling Synthesis with Synthesis Transient Modeling. *Computer*, 24(2):47–59, 2010.

[21] JJ Wells. *Real-time spectral modelling of audio for creative sound transformation.* PhD thesis, University of York, 2006.

[22] Simon J Godsill and Peter J W Rayner. A Bayesian approach to the restoration of degraded audio signals. In *IEEE Transactions on Speech and Audio Processing*, volume 3, pages 267–278, 1995.

[23] Sylvian Marchand Mathie Lagrange and Jean bernard Rault. Long Interpolation of Audio Signals Using Linear Prediction in Sinusoidal Modeling. *Audio Engineering Society*, 53:891–905, 2005.

[24] Paulo A A Esquef, Vesa Välimäki, Kari Roth, and Ismo Kauppinen. Interpolation of long gaps in audio using the warped Burg's method. In *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, pages 1–6, September, 2003.

[25] LWP Biscainho, PSR Diniz, and PAA Esquef. ARMA processes in subbands with application to audio restoration. In *2001 IEEE International Symposium on Circuits and Systems*, page 258, 2001.

[26] G. Cocchi and A. Uncini. Subbands audio signal recovering using neural nonlinear prediction. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, pages 1289–1292, 2000.

[27] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Enhancing the Tracking of Partials for the Sinusoidal Modeling of Polyphonic Sounds. *IEEE Transactions on Audio, Speech and Language Processing*, 15(5):1625–1634, July 2007.

[28] http://www.freesound.org/samplesviewsingle.php?id=82454, August 2010.

[29] Stanley A. Gelfand. *Hearing: An Introduction to Psychological and Physiological Acoustics, Fourth Edition.* Monticelo, New York:Marcel Deker Inc, 1998. ISBN: 0-8247-5652-2.

[30] J. Laroche and M. Dolson. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, May 1999.

[31] Hans Knutsson and Mats Andersson. Loglets: Generalized Quadrature and Phase for Local Spatio-Temporal Structure Estimation. In Josef Bigun and Tomas Gustavsson, editors, *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 107–108. Springer Berlin / Heidelberg, 2003.

[32] Michael J Vrhel, Chulhee Lee, and Michael Unser. Rapid Computation of the Continuous Wavelet Transform by Oblique Projections. In *IEEE Transactions on Signal Processing*, volume 45, pages 891–900, 1997.