universität
wien

DISSERTATION

Titel der Dissertation

"Theory and implementation of adaptive time-frequency transforms"

verfasst von

Dipl.-Math. Nicki Holighaus

angestrebter akademischer Grad

Doktor der Naturwissenschaften (Dr.rer.nat)

Wien, 2013

Studienkennzahl lt. Studienblatt:	A 791 405
Studienrichtung lt. Studienblatt:	Mathematik
Betreuer:	a.o. Prof. Dr. Hans Georg Feichtinger

Contents

Acknowledgments

To call the contents of this thesis the solitary work of a single person would be a terrible injustice towards all the great people that have provided essential parts through their support, inspiration, motivation, collaboration, criticism, mentor- and friendship to name only a few of the multitude forms of input I have been granted during my time in Vienna so far. When I came to Vienna on my first visit to the Numerical Harmonic Analysis Group (NuHAG) in January 2010, the city, the Austrian dialects, the people and most of all the intricacies of scientific work were a mystery to me. Still, Prof. Feichtinger welcomed me into the group wholeheartedly, with a friendliness and enthusiasm that has continued to both inspire and astound me. While pushing me in the right direction when truly necessary, Prof. Feichtinger always gave me the opportunity to pursue the various problems that piqued my interest. I'm truly thankful for his support, without which this thesis would not exist in its current form. My heartfelt thanks also go to Monika Dörfler and Peter Balazs, who shared the regrettably inofficial position of co-advisor. Through support and guidance, as well as, sometimes quite strict, formal criticism, they shaped my work and improved both the intuition and rigor of my mathematical pursuits.

I'm grateful for the incredible community of great local and international co-workers at NuHAG and the Acoustics Research Institute (ARI) that enrich my life on a professional, but also a personal level as collaborators, friends and party animals. They are too many to name and picking some few over the others seems ungrateful to those left out. I trust that they will know that I did not forget the the joy of many fruitful discussions, laughter, cake, games and gossip we shared.

Moreover, I want to thank my family for always supporting me and my decision to continue studying mathematics instead of getting a "proper job". Thanks for occasionally asking what I do and listening to my desperate attempts at explanation, trust me I know how boring it can sound. Finally, my deepest thanks to Marion. It sounds simple enough: You live with me and keep me sane. But I'm well aware that yours is the hardest job of them all. Most incredibly, you seem to enjoy it. Thank you.

During the course of this dissertation project, the author was supported by the Austrian Science Fund (FWF): START-project FLAME [Y551-N13] and NFN SISE [S10602-N13], the Vienna Science and Technology Fund (WWTF): Audiominer (MA09-24) and MulAc (MA07-025), as well as the EU FET Open grant UNLocX (255931).

Abstract

The thesis at hand has the goal of presenting and extending modern methods of applied, mathematical time-frequency analysis. To this end, we investigate two different types of flexible time-frequency representations, their properties and efficient implementation. Our approach builds on the underlying concepts of frames to construct stable and invertible systems of time-frequency atoms.

Firstly, we recall some fundamental mathematical theory that is required for the presentation of our results. This is followed by a section that develops efficient methods for computation and inversion of discrete Gabor transforms on arbitrary, nonseparable lattices. We start from an elegant, unique representation of two dimensional lattices by means of three intuitively meaningful parameters. From this representation, we can easily determine the number of all possible lattices, or those of a particular cardinality, for a given signal length. Furthermore, it allows the straightforward description of any lattice as either a union of rectangular lattices or a geometric transformation of a single rectangular lattice. This in turn enables the use of preexisting, optimized algorithms designed for rectangularly sampled Gabor transforms for computations on arbitrary lattices.

In the second part, we develop a theory of nonstationary Gabor transforms. These nonstationary systems admit the variation of both the window function and the sampling density along either the time or the frequency axis. We investigate the structure of this type of system and their associated frame operators. Among other results, we determine sufficient and necessary conditions for a nonstationary Gabor system to form a frame and therefore allow for stable reconstruction from the frame coefficients. A discrete variant of nonstationary Gabor systems can be used for discrete signals and numerical implementation. We describe in detail both iterative and direct approaches at efficient reconstruction as well as several exemplary implementations of time- or frequency-adaptive systems.

The presented techniques are available as MATLAB toolbox, supplied with the thesis, or in case of Gabor systems on arbitrary lattices, as part of the open source time-frequency toolbox LTFAT for MATLAB/OCTAVE.

Zusammenfassung

Die vorliegende Dissertation hat die Darlegung und Erweiterung moderner Methoden zur angewandten, mathematischen Zeit-Frequenz Analyse zum Ziel. Zu diesem Zweck betrachten wir, aufbauend auf Grundlagen der Frametheorie, zwei unterschiedliche Ansätze für flexible Zeit-Frequenz Darstellungen, sowie deren Eigenschaften und effiziente Implementierung.

Zunächst widmen wir uns der Aufarbeitung der notwendigen, zugrundeliegenden mathematischen Theorie, gefolgt von der Entwicklung effizienter Verfahren zur Berechnung und Invertierung von diskreten Gabortransformationen auf allgemeinen, nichtseparablen Gittern. Wir beginnen dabei mit einer eleganten, eindeutigen Darstellung zweidimensionaler Gitter durch drei leicht verständliche Parameter. Ausgehend von dieser Darstellung können wir nicht nur die Anzahl der möglichen Gitter, oder derer mit einer bestimmten Kardinalität, für eine bestimmte Signallänge bestimmen, sondern auch jedes nichtseparable Gitter leicht als Vereinigung rechteckiger Gitter oder geometrische Transformation eines Solchen beschreiben. Dies wiederum ermöglicht die Verwendung existierender, optimierter Algorithmen für Gabortransformationen auf Rechtecksgittern für die Berechnung auf beliebigen Gittern.

Im zweiten Teil der Dissertationsschrift entwickeln wir eine Theorie nichtstationärer Gabortransformationen. Diese nichtstationären Systeme erlauben die Veränderung der Fensterfunktion und Abtastdichte entlang der Zeit- oder Frequenzachse. Wir untersuchen die Struktur solcher Systeme und der zugehörigen Frameoperatoren. Unter Anderem betrachten wir hinreichende und notwendige Kriterien, so dass ein nichtstationäres Gaborsystem ein Frame ist und damit stabile Rekonstruktion aus den Framekoeffizienten erlaubt. Eine diskrete Variante des Konzepts nichtstationärer Gaborsysteme erlaubt uns die numerische Implementierung. Wir beschreiben sowohl iterative, als auch direkte Methoden zur effizienten Rekonstruktion und einige Beispielimplementierungen zeit- bzw. frequenzadaptiver Systeme im Detail.

Die vorgestellten Verfahren sind verfügbar als MATLAB Toolbox, welche der Dissertation beiliegt, bzw. als Teil der frei erhältlichen LTFAT Zeit-Frequenz Toolbox für MATLAB/OCTAVE.

Chapter 1

Introduction

1.1 Motivation

In the last decades, time-frequency transforms have become increasingly important both in abstract harmonic analysis and digital signal processing. In the form of, e.g. audio and image compression, which often employ time-frequency techniques in some form, they have become an invisible part of our everyday life. What once was a small area in the field of harmonic analysis, introduced by the work of D. Gabor [78], has grown into a vast field of research, populated by theoretical and applied mathematicians, engineers and other researchers. Our understanding of the intrinsic properties of standard tools such as short-time Fourier (or Gabor) transforms and Wavelet transforms has improved vastly, as compared to 20 years ago, with a number of books, e.g. [115], [47], [84], covering the most well-established results. Yet much remains to be investigated.

Both Gabor and Wavelet transforms possess a large amount of structure, giving rise to a deep, yet accessible mathematical theory. Sometimes though, especially in applications, this structure might be considered as a drawback and a less rigid framework is desired, be it purely in terms of the employed sampling strategy or the underlying representation itself. The purpose of this thesis is to investigate some interesting recent approaches to overcome the rigidity, illustrate their usefulness with some selected implementations and finally provide a comprehensive MATLAB toolbox for experiments and applications, based on the presented results.

1.1.1 Notation

$\mathbb{N}$, $\mathbb{N}_0$	the natural numbers without, respectively with, 0
$\mathbb{Z}_L$	shorthand for $\mathbb{Z}/L\mathbb{Z}$
$f(\cdot)$	Continuous function
$f[\cdot]$	Discrete function / vector
$\mathcal{F}f = \hat{f}$	Fourier transform of f (continuous/discrete)
$\mathcal{F}^{-1}f = \check{f}$	Inverse Fourier transform of f (continuous/discrete)
Φ, φ_{sub}	General system of functions / frame and its elements
Ψ, ψ_{sub}	A dual frame of Φ
$\widetilde{\Phi}$, $\widetilde{\varphi_{\text{sub}}}$	Canonical dual frame of Φ
$\mathring{\Phi}$, $\mathring{\varphi}_{\text{sub}}$	Canonical tight frame associated to Φ
$\mathbf{C}_\Phi = \mathbf{C}$	Analysis operator associated with Φ
$\mathbf{D}_\Phi = \mathbf{D}$	Synthesis operator associated with Φ
$\mathbf{S}_\Phi = \mathbf{S}$	Frame operator associated with Φ
$\mathbf{T}_{\text{sub}}$	Translation operator
$\mathbf{M}_{\text{sub}}$	Modulation operator
$\mathcal{G}(g, a, b)$, $g_{n,m}$	(Regular, separable) Gabor system
$\mathcal{G}(g, a, b, s)$, $g_{n,m}$	nonseparable Gabor system
$\mathcal{G}(g, \Lambda)$, g_λ	Gabor system on the lattice Λ
$\mathcal{G}(\mathbf{g}, \mathbf{b})$, $g_{n,m}$	(Time-side) nonstationary Gabor system
$\check{\mathcal{G}}(\mathbf{g}, \mathbf{a})$, $g_{n,m}$	(Frequency-side) nonstationary Gabor system
$\lVert \cdot \rVert_{op}$	Operator norm
$\mu(M)$	(Lebesgue-)measure of the set M
χ_M	characteristic function of the set M
$\operatorname{sgn}(t)$	sign of $t \in \mathbb{R}$, equals 1 for $t > 0$, -1 for $t < 0$ and 0 for $t = 0$
$\operatorname{mod}(k, l)$	modulo operation with respect to $l \in \mathbb{Z}$, applied to $k \in \mathbb{Z}$

1.2 Topics and Structure

The work at hand considers two somewhat separate main themes, the first being the theory and efficient implementation of Gabor systems on arbitrary lattices, foremost in a finite, discrete setting. The second theme concerns the theory and implementation of adaptive time-frequency representations and will focus on nonstationary Gabor systems in particular. These are a generalization of Gabor systems that allow for varying windows and sampling sets along time or frequency while retaining important properties like FFT-based (Fast Fourier Transform) implementation and a well-structured frame operator.

Both parts can again be divided into two sections. The first section builds the continuous theory, where basic properties are recalled and more intricate ones proven and discussed, while the second one provides the discrete theory accompanied by implementations and/or applications.

Most of the material presented in this thesis is compiled from the following publications or manuscripts the thesis author has authored or co-authored:

- L. Toth, M. Hampejs, N. Holighaus, and C. Wiesmeyr, "On the subgroups of the group $\mathbb{Z}_m \times \mathbb{Z}_n$," Preprint, submitted, arXiv: 1211.1797

- N. Holighaus, "Structure of nonstationary Gabor frames and their dual systems," Preprint, submitted, arXiv: 1306:5037

- P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. A. Velasco, "Theory, implementation and applications of nonstationary Gabor Frames," J. Comput. Appl. Math., vol. 236, no. 6, pp. 1481-1496, 2011

- T. Necciari, P. Balazs, N. Holighaus, and P. Søndergaard, "The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction," Proceedings of the 38th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013), 2013

- G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with nonstationary Gabor frames," Proceedings of DAFX11, Paris, 2011

- N. Holighaus, M. Dörfler, G. A. Velasco, and T. Grill, "A framework for invertible, real-time constant-Q transforms," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, pp. 775-785, April 2013

- N. Holighaus, C. Wiesmeyr and P.L. Søndergaard, "Efficient Algorithms for the Discrete Gabor Transform on a nonseparable lattice," to appear in IEEE Transactions on Signal Processing, 2013

The text is structured as follows.

1.2.1 Preliminaries

In this section, we recall important mathematical foundations frequently used in the course of this text. Also some of the notational conventions are clarified. Among those concepts are Banach and Hilbert spaces, in particular L^p and ℓ^p-spaces, but also the Wiener space $W(L^\infty, \ell^1, \mathbb{R}^d)$. Furthermore bounded operators, Neumann series expansions and important results from continuous and discrete Fourier analysis are reviewed.

Frames and Gabor systems form the backbone concepts of this thesis and thus the related theory is shortly discussed. After defining frames and Bessel sequences, we introduce the related operators and the concept of dual and tight frames. Subsequently, we define short-time Fourier transforms and Gabor systems, followed by recalling their most striking properties. Among them are the Walnut and Janssen representations, Ron and Shen's duality principle and the painless case result by Daubechies, Grossmann and Meyer. We close with brief sections on discrete Gabor systems and general Wavelet systems.

1.2.2 Gabor systems on lattices

In this chapter, we investigate Gabor systems on arbitrary lattices. We begin by introducing the concept of lattices and its relevance for flexibly sampled Gabor transforms, finding that most of the properties of regularly sampled Gabor systems are retained in some form, as long as the sampling set forms a group. We discuss different methods to express a lattice and the corresponding Gabor system through one or more rectangular sampling sets, thereby reducing questions about systems on general lattices to the classical case. Our main tools here are multiwindow Gabor systems and the theory of metaplectic operators.

In the next section, we discuss properties of the subgroups of the group $\mathbb{Z}_M \times \mathbb{Z}_N$, where M and N are arbitrary positive integers. Simple formulae for the total number of subgroups and the number of subgroups of a given order are deduced. The cyclic subgroups and subgroups of a given exponent are also considered.

We proceed by discussing discrete, finite Gabor analysis on arbitrary group lattices. We survey existing methods for reducing discrete, finite Gabor transforms on general lattices to the product lattice case, which is well-studied. The first is based on multiwindow Gabor schemes, while the alternative methods use metaplectic operators to transform Gabor systems on general lattices to a product lattice setting. An improvement of previous results on the latter method is presented. The fact that all finite lattices can be described by a class of lower triangular matrices is used to construct an algorithm that shows improved performance over previous methods in many cases. Comparisons are made with respect to the computational complexity, and the running time of optimized implementations in the C programming language. The new algorithms have the lowest known computational complexity for nonseparable lattices and the implementations are freely available for download. By summarizing general background information on the state of the art, this chapter can also be seen as a research survey, sharing with the readers experience in the numerical work in Gabor analysis.

1.2.3 Nonstationary Gabor theory

This and the subsequent chapters form the main body of the thesis. Nonstationary Gabor systems are a recent generalization of classical Gabor systems, allowing for adaptive time-frequency representations. This chapter begins by introducing the basic definitions and properties of such systems.

Signal analysis with classical Gabor frames leads to a fixed time-frequency resolution over the whole time-frequency plane. To overcome the limitations imposed by this rigidity, we propose an extension of Gabor theory that leads to the construction of frames with time-frequency resolution changing over time or frequency. We describe the construction of the resulting *nonstationary Gabor frames* and give the explicit formula for the canonical dual frame for a particular case, the *painless case*.

Subsequently, we investigate the structural properties of dual systems for nonstationary Gabor frames. In particular, we prove that some inverse nonstationary Gabor frame operators admit a Walnut-like representation, i.e. the operator acting on a function can be described by weighted translates of that function. This is of particular interest, when the original frame operator is not a simple multiplication operator. In this case, which only occurs when compactly supported window functions are used, the canonical dual frame partially inherits the structure of the original frame, with differences that we describe in detail. Moreover, we determine a necessary and sufficient condition for a pair of nonstationary Gabor frames to form dual frames, valid under mild restrictions. This condition is then applied in a simple setup, to prove the existence of dual pairs of nonstationary Gabor systems with coarser frequency sampling than allowed by previous results. A discussion of the results, restricted to the classical case of regular Gabor systems, precedes the statement of the general results. Here, we also explore a connection to recent work of Christensen, Kim and Kim on Gabor frames with compactly supported window function.

1.2.4 Nonstationary Gabor: Implementation

This chapter begins by introducing finite-dimensional nonstationary Gabor frames. Since most results on the continuous case can easily be derived for this case by slightly modifying the original proof, only the most important results are repeated. A method for implementing the above-mentioned transforms with perfect reconstruction is provided.

We continue to elaborate on several implementations of nonstationary Gabor frames and their application in audio signal processing, in particular a method for automatic adaptation to transients and an algorithm for an invertible constant-Q transform. First, we discuss the construction of a flexible class of nonstationary Gabor frames that can be used in a time-adaptation scheme and how to use this construction for adaptation based on an onset sequence.

A longer section discusses the *nonstationary Gabor based constant-Q transform (CQ-NSGT)* and a framework for its real-time implementation, e.g. for audio processing tasks. Audio signal processing frequently requires time-frequency representations and in many applications, a nonlinear spacing of frequency bands is preferable. This chapter introduces a framework for efficient implementation of invertible signal transforms allowing for nonuniform frequency resolution. Nonuniformity in frequency is achieved by applying *nonstationary Gabor frames* with adaptivity in the frequency domain. The realization of a perfectly invertible *constant-Q* transform is described in detail. To achieve real-time processing, independent of signal length, slicewise processing of the full input signal is proposed and referred to as *sliCQ transform.*

By applying frame theory and FFT-based processing, the presented approach overcomes computational inefficiency and lack of invertibility of classical constant-Q transform implementations. Numerical simulations evaluate the efficiency of the proposed algorithm and the method's applicability is illustrated by experiments on real-life audio signals.

Finally, we present some extensions of the framework just introduced. The efficiency of iterative reconstruction and analysis schemes is discussed at the example of a so-called ERBlet transform. This transform is, like the constant-Q nonstationary Gabor transform, an instance of the more general framework of nonstationary Gabor filterbanks. The ERBlet transform is based on the auditory ERB scale, a frequency scale adapted to human auditory perception, and thus well-suited for analysis and audio processing using perceptual considerations. The chapter is closed with a discussion of nonstationary Gabor frames constructed from a warping of the time- or frequency axis. We introduce a method to construct tight, adapted time-frequency frames from simple partitions of unity.

1.2.5 Appendix - The toolboxes

The algorithms and methods provided in this thesis have been compiled into a freely available MATLAB toolbox, respectively worked into the established MATLAB/OCTAVE toolbox LTFAT (Large Time-Frequency Analysis Toolbox). Since a central part of this thesis project was the implementation of the research conducted throughout, the documentation of the provided code is included in the text. For the code itself, we refer the reader to the respective toolbox webpages `nsg.sourceforge.net` and `ltfat.sourceforge.net`.

Chapter 2

Preliminaries

Before we explore the new results presented in this thesis, we recall a number of basic definitions and mathematical concepts that will be used repeatedly during the following chapters. Firstly, we introduce the relevant function space concepts and some important results from Fourier analysis. What follows are sections on basic frame and Gabor theory, along with a very brief description of Wavelet transforms. More detailed results on the subjects touched upon can be found in standard works on functional analysis [42], Fourier analysis [81], [104] and [75], frame theory [33] or time-frequency and Wavelet analysis [84], [47] and [116].

2.1 Basics

The function spaces we consider are usually Banach spaces $\mathcal{B}$, i.e. complete normed vector spaces, or even Hilbert spaces $\mathcal{H}$, additionally equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. In particular, L^p-spaces over $\mathbb{R}^d$, $d \in \mathbb{N}$, are of special importance.

For $X \subseteq \mathbb{R}^d$, we denote by $\mu(X)$ the usual Lebesgue measure of X. If $\mu(X) = 0$, then we refer to X as a null set. The essential infimum of a function $f \; : \; \mathbb{R}^d \mapsto \mathbb{R}$ is defined as

$$\text{ess inf} \, f = \sup\{x \in \mathbb{R} \; : \; \mu(\{t \in \mathbb{R}^d \; : \; f(t) < x\}) = 0\}$$

if $\{x \in \mathbb{R} \; : \; \mu(\{t \in \mathbb{R}^d \; : \; f(t) < x\}) = 0\}$ is non-empty and $-\infty$ otherwise, whereas its essential supremum is

$$\text{ess sup} \, f = \inf\{x \in \mathbb{R} \; : \; \mu(\{t \in \mathbb{R}^d \; : \; f(t) > x\}) = 0\}$$

if $\{x \in \mathbb{R} \; : \; \mu(\{t \in \mathbb{R}^d \; : \; f(t) > x\}) = 0\}$ is non-empty and ∞ otherwise.

The *essential support* or simply *support* $\text{supp}(f)$ of $f \; : \; \mathbb{R}^d \mapsto \mathbb{C}$ is the smallest subset of $\mathbb{R}^d$, such that $f(t) = 0$ *almost everywhere*, i.e. except for null sets, on

$\mathbb{R}^d \setminus \operatorname{supp}(f)$ or, formally

$$\operatorname{supp}(f) = \bigcap_{\substack{X \subseteq \mathbb{R}^d \\ f(t)=0 \text{ a.e. on } \mathbb{R}^d \setminus X}} X.$$

For simplicity, we use the notation $f_1 \equiv f_2$ if $f_1 = f_2$ almost everywhere and $f_1 \simeq f_2$ if there are constants $C_1, C_2 > 0$ such that $C_1 f_2 \leq f_1 \leq C_2 f_2$ almost everywhere.

Also, we denote by χ_X the characteristic function on the set X, i.e.

$$\chi_X(t) = \begin{cases} 1, & \text{for } t \in X \\ 0, & \text{else,} \end{cases}$$

and define $f\,|_X := f\chi_X$, the restriction of f to the set X.

Definition 1. Let $p \in \mathbb{N}$, then $L^p(\mathbb{R}^d)$ is defined as the Banach space of functions $f \ : \ \mathbb{R}^d \mapsto \mathbb{C}$, such that

$$\int_{\mathbb{R}^d} |f(t)|^p \ dt < \infty$$

and equipped with the norm

$$\|f\|_p := \|f\|_{L^p(\mathbb{R}^d)} := \left(\int_{\mathbb{R}^d} |f(t)|^p \ dt \right)^{1/p}.$$

For $p = \infty$, $L^\infty(\mathbb{R}^d)$ is the Banach space of essentially bounded functions, i.e.

$$\text{ess sup}\,|f| < \infty$$

and equipped with the norm

$$\|f\|_\infty := \|f\|_{L^\infty(\mathbb{R}^d)} := \text{ess sup}\,|f|.$$

The spaces $L^1(\mathbb{R}^d)$ and $L^2(\mathbb{R}^d)$ are often referred to as the spaces of integrable functions and square-integrable functions, respectively. Furthermore, $L^2(\mathbb{R}^d)$ is a Hilbert space with the inner product

$$\langle f_1, f_2 \rangle_{L^2(\mathbb{R}^d)} = \int_{\mathbb{R}^d} f_1(t)\overline{f_2(t)} \ dt, \text{ for all } f_1, f_2 \in L^2(\mathbb{R}^d).$$

Note that $f_1\overline{f_2} \in L^1(\mathbb{R}^d)$ by Cauchy-Schwarz' inequality

$$\|f_1\overline{f_2}\|_1 \leq \|f_1\|_2 \|f_2\|_2. \tag{2.1}$$

The open $L^2(\mathbb{R}^d)$-ball around t with radius δ is given by

$$B_\delta(t) := \{x \in \mathbb{R}^d \ : \ \|x - t\|_2 < \delta\}.$$

From now on, we will omit the subscript for norm and inner product associated to $L^2(\mathbb{R}^d)$.

At some point, we will use another important function space.

Definition 2. The Wiener space $W(L^\infty, \ell^1, \mathbb{R}^d)$ is the space of functions $f \in L^\infty(\mathbb{R}^d)$ such that

$$\|f\|_{W(L^\infty,\ell^1,\mathbb{R}^d)} := \sum_{k\in\mathbb{Z}^d} \operatorname*{ess\,sup}_{t\in[0,1]^d} |f(t+k)| < \infty. \tag{2.2}$$

A whole theory of so-called Wiener amalgam spaces exists, generalizing the Wiener space and considering normed function spaces combining notions of global and local function behavior, see e.g. [62], [76], [93]. The Wiener space is particularly useful when working with sampling and periodization operators.

Each continuous L^p-space has a sequence space counterpart defined in an analogous fashion.

Definition 3. Let $p \in \mathbb{N}$, then $\ell^p(\mathbb{Z}^d)$ is defined as vector space of functions $f \ : \ \mathbb{Z}^d \mapsto \mathbb{C}$, such that

$$\sum_{l\in\mathbb{Z}^d} |f[l]|^p < \infty$$

and equipped with the norm

$$\|f\|_p := \|f\|_{\ell^p(\mathbb{Z}^d)} := \left(\sum_{l\in\mathbb{Z}^d} |f[l]|^p\right)^{1/p}.$$

For $p = \infty$, $\ell^\infty(\mathbb{Z}^d)$ is the vector space of bounded sequences, i.e.

$$\sup_{l\in\mathbb{Z}^d} |f[l]| < \infty$$

and equipped with the norm

$$\|f\|_\infty := \|f\|_{\ell^\infty(\mathbb{Z}^d)} := \sup_{l\in\mathbb{Z}^d} |f[l]|.$$

Where no confusion is possible, we use the same notation for L^p- and ℓ^p-norms, as well as the L^2 and ℓ^2 inner products. Alternatively, we might use the subscript ℓ^2 for norm and inner product in $\ell^2(\mathbb{Z}^d)$.

The support of a vector $f\ :\ \mathbb{Z}^d \mapsto \mathbb{C}$ is defined as the set of non-zero values of f, defined as $\text{supp}(f) = \{l \in \mathbb{Z}^d\ :\ f[l] \neq 0\}$.

Finite dimensional vector spaces $\mathbb{C}^L$ and $\mathbb{R}^L$ are always considered circular, i.e. modulo L, and whenever ℓ^p-norms or the ℓ^2 inner product are used in that context, they are considered to be evaluated on a single period of length L. For convenience, we may switch back and forth between indexing a finite dimensional vector by $0, \ldots, L-1$ and $-\lfloor L/2 \rfloor, \ldots, \lceil L/2 \rceil - 1$.

In subsequent chapters we consider linear operators between normed function spaces, i.e. operators of the form $\mathbf{O}\ :\ V \mapsto W$ such that, for all $f_1, f_2 \in V$ and $C \in \mathbb{R}$, $\mathbf{O}(f_1 + f_2) = \mathbf{O}(f_1) + \mathbf{O}(f_2)$ and $\mathbf{O}(\lambda f_1) = \lambda \mathbf{O}(f_1)$ hold. The operator norm of $\mathbf{O}$ is defined by

$$\|\mathbf{O}\|_{op} = \sup\{\|\mathbf{O}(f)\|_W\ :\ f \in V,\ \|f\|_V \leq 1\}.$$

Whenever the operator norm is finite, e.g. $\|\mathbf{O}\|_{op} = B$, then we say that $\mathbf{O}$ is *bounded above.* If there exist $0 < A \leq B < \infty$ such that

$$A\|f\|_V \leq \|\mathbf{O}(f)\|_W \leq B\|f\|_V, \text{ for all } f \in V, \tag{2.3}$$

then we say that $\mathbf{O}$ is *bounded.* Moreover, if (2.3) holds, $\mathbf{O}$ has a bounded inverse $\mathbf{O}^{-1}\ :\ W \mapsto V$.

If $\mathbf{O}$ is a bounded, linear operator on V, i.e. W equals V, and there is a $C \in \mathbb{R} \setminus \{0\}$ such that the Neumann series

$$C \sum_{j=0}^{\infty} (\mathbf{I} - C\mathbf{O})^j \tag{2.4}$$

converges, then the series limit equals the inverse operator $\mathbf{O}^{-1}$.

2.1.1 Fourier analysis

For an integrable function $f \in L^1(\mathbb{R}^d)$, we denote its Fourier transform by

$$\mathcal{F}f(\xi) = \hat{f}(\xi) = \int_{\mathbb{R}^d} f(t)\, e^{-2\pi i \langle \xi, t \rangle}\ dt. \tag{2.5}$$

By the Riemann-Lebesgue lemma, $\hat{f} \in \mathcal{C}_0(\mathbb{R}^d)$, i.e. $\hat{f}$ is continuous and vanishing at infinity. Indeed, $\hat{f}$ is even uniformly continuous. If $\hat{f}$ is also in $L^1(\mathbb{R}^d)$, then the inverse Fourier transform is given by

$$\mathcal{F}^{-1}\hat{f}(t) = \check{\hat{f}}(t) = \int_{\mathbb{R}^d} \hat{f}(\xi)\, e^{2\pi i \langle t, \xi \rangle}\ d\xi \tag{2.6}$$

and $\check{\hat{f}} = f$.

Let $f_1, f_2 \in L^1(\mathbb{R}^d)$, then the convolution of f_1 and f_2, defined by

$$f_1 * f_2(t) = \int_{\mathbb{R}^d} f_1(x) f_2(t-x)\, dx, \tag{2.7}$$

is an element of $L^1(\mathbb{R}^d)$. Moreover, the convolution theorem

$$\widehat{f_1 * f_2} = \hat{f}_1 \hat{f}_2 \tag{2.8}$$

holds.

A more symmetric theory is obtained by extending the Fourier transform to $L^2(\mathbb{R}^d)$ employing a standard density argument. In this setting, we obtain

$$\|f\| = \|\hat{f}\|, \text{ for all } f \in L^2(\mathbb{R}^d), \tag{2.9}$$

Plancherel's theorem, showing that $\mathcal{F}$ is a unitary operator on $L^2(\mathbb{R}^d)$ and consequently *Parseval's formula*

$$\langle f_1, f_2 \rangle = \langle \hat{f}_1, \hat{f}_2 \rangle, \text{ for all } f_1, f_2 \in L^2(\mathbb{R}^d), \tag{2.10}$$

holds. Note that, through (2.8) and (2.10), the convolution definition can be extended to $L^2(\mathbb{R}^d)$.

Let us recall some of the fundamental properties of Fourier transforms on $L^2(\mathbb{R}^d)$. Define continuous *translation* and *modulation* operators by

$$\mathbf{T}_x f = f(\cdot - x) \text{ and } \mathbf{M}_\omega f = e^{2\pi i \langle \omega, \cdot \rangle} f, \tag{2.11}$$

for $x, \omega \in \mathbb{R}^d$. Both of these operators are easily seen to be linear and unitary. For $f \in L^2(\mathbb{R}^d)$, we have

$$\widehat{\mathbf{T}_x f} = \mathbf{M}_{-x} \hat{f} \quad \text{and} \quad \widehat{\mathbf{M}_\omega f} = \mathbf{T}_\omega \hat{f}.$$

Moreover,

$$\widehat{f(-\cdot)} = \hat{f}(-\cdot), \quad \hat{\bar{f}} = \overline{\check{f}}$$

and, for $s \in \mathbb{R}^+$,

$$\widehat{\mathbf{D}_s f} = \mathbf{D}_{s^{-1}} \hat{f},$$

with the dilation operator $\mathbf{D}_s f = s^{-d/2} f(\cdot/s)$.

Once we consider sampling and Fourier transforms, periodization problems and Poisson's summation formula appear naturally. In its basic form, it reads as follows.

Proposition 1. *Let $f \in L^1(\mathbb{R}^d)$ be such that $|f(t)| \leq C(1+|t|)^{-d-\epsilon}$ and $|\hat{f}(\xi)| \leq C(1+|\xi|)^{-d-\epsilon}$, for some $C, \epsilon > 0$. For all $t \in \mathbb{R}^d$,*

$$\sum_{l \in \mathbb{Z}^d} f(t+l) = \sum_{l \in \mathbb{Z}^d} \hat{f}(l) e^{2\pi i \langle l,t \rangle} \tag{2.12}$$

holds pointwise and both sums converge absolutely.

If absolute convergence and pointwise validity are not necessary, weaker conditions on $f, \hat{f}$ are sufficient for (2.12) to hold. The equality above can be shown to hold almost everywhere for $f \in L^2(\mathbb{R})$ [22] with norm convergence. With Poisson's summation formula, we transition from Fourier transforms to Fourier series.

The Fourier series of a periodic function $f \in L^1(\mathbb{T}^d)$, with the d-dimensional torus $\mathbb{T}^d = (\mathbb{R}/\mathbb{Z})^d$, at position $t \in \mathbb{T}^d$ is given by

$$\sum_{l \in \mathbb{Z}^d} c_f[l] e^{2\pi i \langle l,t \rangle}, \tag{2.13}$$

with the Fourier coefficients

$$c_f[l] = \int_{\mathbb{T}^d} f(t) e^{-2\pi i \langle t,l \rangle} \, dt. \tag{2.14}$$

While the Fourier coefficients are well-defined (f is integrable), convergence of the Fourier series is a complicated matter, see e.g. [81], [104] and [75]. However, if (2.13) converges as a function in t, then its limit equals f almost everywhere. Again, the mapping from a periodic function to its Fourier coefficients (and backwards) can be extended to a unitary mapping from $L^2(\mathbb{T}^d)$ to $\ell^2(\mathbb{Z}^d)$ (from $\ell^2(\mathbb{Z}^d)$ to $L^2(\mathbb{T}^d)$). A Fourier transform for sequences can be defined in the same manner by switching signs in the exponentials appearing in (2.13) and (2.14). If the reference is clear, we will denote the Fourier transform of a periodic function f_1 and a sequence f_2 as $\widehat{f_1}$ with $\widehat{f_1}[l] = c_{f_1}[l]$ if $f_1 \in L^1(\mathbb{T}^d)$ and $\widehat{f_2}$ with $\widehat{f_2}(t) = \sum_{l \in \mathbb{Z}^d} f_2[l] e^{-2\pi i \langle l,t \rangle}$ if $f_2 \in \ell^1(\mathbb{Z}^d)$, respectively. Inverse Fourier transforms are denoted as $\check{f_1}$ and $\check{f_2}$, if well-defined.

Finite dimensional vector spaces $\mathbb{C}^L, \mathbb{R}^L$, have their own concept of a *discrete Fourier transform (DFT)* that shares most of the basic properties of continuous Fourier transforms, such as Parseval's formula (possibly up to a normalization factor), duality of translation and modulation operators and their behavior under involution, complex conjugation and convolution. Discrete translation and modulation operators are given by

$$\mathbf{T}_n f[l] = f[l-n] \quad \text{and} \quad \mathbf{M}_k f[l] = f[l] e^{2\pi i k l/L},$$

for $n, k \in \mathbb{Z}_L$. Note that vector indices are considered circularly, i.e. modulo L, unless noted otherwise.

Definition 4. The *unitary* discrete Fourier transform of $f \in \mathbb{C}^L$ is a unitary mapping from $\mathbb{C}^L$ into itself, given by

$$\hat{f}[l] := \mathbf{DFT}_L(f)[l] = L^{-1/2} \sum_{j=0}^{L-1} f[j] e^{-2\pi i j l / L}. \tag{2.15}$$

The inverse mapping is given by

$$\check{\hat{f}}[l] := \mathbf{IDFT}_L(\hat{f})[l] = L^{-1/2} \sum_{j=0}^{L-1} \hat{f}[j] e^{2\pi i j l / L}. \tag{2.16}$$

The circular convolution $f_1 * f_2$ if $f_1, f_2 \in \mathbb{C}^L$ is defined by

$$(f_1 * f_2)[l] = \sum_{j=0}^{L-1} f_1[j] f_2[l-j]$$

and equals $\widecheck{\hat{f}_1 \hat{f}_2}$. The discrete Fourier transform can be easily generalized to multidimensional vectors, but the extension is actually equivalent to applying subsequent one dimensional DFTs to each dimension of the vector. Therefore, all the usual properties are trivially inherited.

2.2 Frames

Frame theory attempts to describe stable, potentially redundant spanning sets. As such, *frames* are a generalization of Riesz and orthonormal bases equipped with a notion of decomposing functions into a linear combination of building blocks, the *frame elements*. Introduced more than 60 years ago by Duffin and Schaeffer [57], the rise of frames in time-frequency analysis and signal processing began with the work of Daubechies, Grossmann and Meyer [48], a scientific contribution that we will stumble upon repeatedly during the course of this thesis. A good resource on the established results in general frame theory and for some special cases are Christensen's books [31, 33] or [28]. In a Hilbert space $\mathcal{H}$, frames are defined as follows.

Definition 5. Let Λ be a countable index set. Then the sequence $\Phi = (\varphi_\lambda)_{\lambda \in \Lambda}$ of functions $\varphi_\lambda \in \mathcal{H}$ is called a frame if there exist positive constants A and B such that

$$A \|f\|_{\mathcal{H}}^2 \leq \sum_{\lambda \in \Lambda} |\langle f, \varphi_\lambda \rangle_{\mathcal{H}}|^2 \leq B \|f\|_{\mathcal{H}}^2 \quad \forall f \in \mathcal{H}, \tag{2.17}$$

i.e. $\sum_{\lambda \in \Lambda} |\langle f, \varphi_\lambda \rangle_{\mathcal{H}}|^2 \simeq \|f\|_{\mathcal{H}}^2$. The constants A and B are called lower and upper frame bound, respectively. If $A = B$, then Φ is a *tight frame*. The sequence Φ is a *Bessel sequence*, if at least the upper bound in (2.17) is satisfied.

In the course of this thesis, the Hilbert spaces $\mathcal{H}$ in question will usually be $L^2(\mathbb{R})$, $L^2(\mathbb{R}^d)$ or $\mathbb{C}^L$ and $\mathbb{C}^{L_1} \times \ldots \times \mathbb{C}^{L_k}$ for some $k \in \mathbb{N}$. A number of important operators are associated to a frame.

Definition 6. Let $\Phi = (\varphi_\lambda)_{\lambda\in\Lambda}$ be a frame or Bessel sequence in $\mathcal{H}$. Then the *analysis operator* $\mathbf{C}_\Phi : \mathcal{H} \mapsto \ell^2(\Lambda)$ is defined by

$$\mathbf{C}_\Phi(f)[\lambda] = \langle f, \varphi_\lambda \rangle_{\mathcal{H}}, \text{ for all } f \in \mathcal{H},\ \lambda \in \Lambda. \tag{2.18}$$

The *synthesis* or *reconstruction operator* $\mathbf{D}_\Phi : \ell^2(\Lambda) \mapsto \mathcal{H}$ is defined by

$$\mathbf{D}_\Phi c = \sum_{\lambda\in\Lambda} c[\lambda]\varphi_\lambda, \text{ for all } c \in \ell^2(\Lambda). \tag{2.19}$$

Both $\mathbf{C}_\Phi$ and $\mathbf{D}_\Phi$ are bounded above with $\|\mathbf{C}_\Phi\| \leq \sqrt{B}$, $\|\mathbf{D}_\Phi\| \leq \sqrt{B}$. If Φ is a frame, their composition $\mathbf{S}_\Phi := \mathbf{D}_\Phi\mathbf{C}_\Phi$ is a bounded operator from $\mathcal{H}$ into itself, given by

$$\mathbf{S}_\Phi f = \sum_{\lambda\in\Lambda} \langle f, \varphi_\lambda \rangle_{\mathcal{H}} \varphi_\lambda, \text{ for all } f \in \mathcal{H}, \tag{2.20}$$

where the sum converges unconditionally. If the attribution to a certain frame is clear, we will omit the subscripts and write $\mathbf{S}$, $\mathbf{D}$ and $\mathbf{C}$ for the respective operator.

If Φ is a frame, it can easily be verified that $\mathbf{C}$ equals the adjoint $\mathbf{D}^*$ of $\mathbf{D}$, i.e. $\mathbf{S} = \mathbf{C}^*\mathbf{C} = \mathbf{D}\mathbf{D}^*$. Moreover, $\mathbf{S}$ is self-adjoint, positive and invertible, with the optimal upper and lower frame bounds given by $B_0 = \|\mathbf{S}\|_{op}$ and $A_0^{-1} = \|\mathbf{S}^{-1}\|_{op}$, respectively.

Any frame admits, possibly non-unique, *dual frames* $\Psi = (\psi_\lambda)_{\lambda\in\Lambda}$ with $\psi_\lambda \in \mathcal{H}$, such that any function $f \in \mathcal{H}$ can be perfectly reconstructed from the frame analysis. Explicitly $\mathbf{D}_\Psi\mathbf{C}_\Phi = \mathbf{I} = \mathbf{D}_\Phi\mathbf{C}_\Psi$, where $\mathbf{I}$ is the identity operator on $\mathcal{H}$. A particular dual frame is given by applying the inverse frame operator to the frame elements: $\widetilde{\varphi_\lambda} := \mathbf{S}_\Phi^{-1}\varphi_\lambda$.

Corollary 1. *Let Φ be a frame with frame bounds A, B. Then $\widetilde{\Phi} = (\widetilde{\varphi_\lambda})_{\lambda\in\Lambda}$, with $\widetilde{\varphi_\lambda} := \mathbf{S}_\Phi^{-1}\varphi_\lambda$, is a frame with frame bounds B^{-1}, A^{-1} and the following equalities hold.*

$$f = \sum_{\lambda\in\Lambda} \langle f, \widetilde{\varphi_\lambda} \rangle \varphi_\lambda = \sum_{\lambda\in\Lambda} \langle f, \varphi_\lambda \rangle \widetilde{\varphi_\lambda}, \textit{ for all } f \in \mathcal{H}, \tag{2.21}$$

where both sums converge unconditionally. We call $\widetilde{\Phi}$ the canonical dual frame *associated with Φ.*

Since $\mathbf{S}$ is bounded, self-adjoint and positive, the operator $\mathbf{S}^{-1/2}$ is also well-defined, giving rise to the *canonical tight frame* $\mathring{\Phi} = (\mathring{\varphi}_\lambda)_{\lambda \in \Lambda}$, with $\mathring{\varphi}_\lambda := \mathbf{S}^{-1/2}\varphi_\lambda$. The canonical tight frame is a tight frame with frame bound 1, i.e.

$$f = \sum_{\lambda \in \Lambda} \langle f, \mathring{\varphi}_\lambda \rangle \mathring{\varphi}_\lambda,$$

for all $f \in \mathcal{H}$, with unconditional convergence of the sum.

For any frame, the inverse frame operator admits a Neumann series representation [57]. Let $0 < A \leq B < \infty$ be the optimal frame bounds, then

$$\mathbf{S}^{-1} = 2/(A+B) \sum_{j=0}^{\infty} (\mathbf{I} - 2\mathbf{S}/(A+B))^j, \tag{2.22}$$

where $\mathbf{I}$ denotes the identity operator. The normalization factor $2/(A+B)$ yields $\|\mathbf{I} - 2\mathbf{S}/(A+B)\|_{op} \leq \frac{B-A}{B+A} < 1$ and the fastest convergence among all possible choices [82], [110], [100].

2.3 The short-time Fourier transform and Gabor systems

The classical time-frequency analysis tool is the *short-time Fourier transform (STFT)*. For a more detailed introduction to short-time Fourier transforms and Gabor analysis, see [71] or [84].

Definition 7. The short-time Fourier transform of a signal $f \in L^2(\mathbb{R}^d)$ with respect to the non-zero *window* function $g \in L^2(\mathbb{R})$ is defined as

$$\mathcal{V}_g f\,(x, \omega) = \int_{\mathbb{R}} f(t)\,\overline{g(t-x)} e^{-2\pi i \langle \omega, t \rangle}\, dt = \langle f, \mathbf{M}_\omega \mathbf{T}_x g \rangle\,. \tag{2.23}$$

The STFT of f with respect to g is a function in $L^2(\mathbb{R}^{2d})$ and defined pointwise. From the orthogonality relations, also called *Moyal's formula*,

$$\langle \mathcal{V}_{g_1} f_1, \mathcal{V}_{g_2} f_2 \rangle = \langle f_1, f_2 \rangle \overline{\langle g_1, g_2 \rangle}, \text{ for all } f_1, f_2, g_1, g_2 \in L^2(\mathbb{R}^d), \tag{2.24}$$

we can deduce $\|\mathcal{V}_g f\| = \|f\| \|g\|$. In particular, the STFT is unitary whenever $\|g\| = 1$. It is also invertible.

Corollary 2. *Let $g, h \in L^2(\mathbb{R}^d)$ and $\langle g, h \rangle \neq 0$, then for all $f \in L^2(\mathbb{R}^d)$,*

$$f = \langle h, g \rangle^{-1} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \mathcal{V}_g f(x, \omega) \mathbf{M}_\omega \mathbf{T}_x h \; d\omega\; dx, \tag{2.25}$$

where the vector valued integral is to be understood in the weak sense.

A sampled short-time Fourier transform is also referred to as *Gabor transform.* More explicitly, let $g \in L^2(\mathbb{R}^d)$ be a window function and $a, b \in \mathbb{R}^+$, then the corresponding (separable) *Gabor system* [78] is defined as

$$\mathcal{G}(g, a, b) = \{\mathbf{M}_{bm}\mathbf{T}_{an}g : m, n \in \mathbb{Z}^d\}. \tag{2.26}$$

If the reference to a Gabor system is clear, we will use the notation $g_{n,m} = \mathbf{M}_{bm}\mathbf{T}_{an}g$.

If a Gabor system forms a frame or Bessel sequence, we call it *Gabor frame* or *Gabor Bessel sequence.* A Gabor system can only be a frame if $ab \leq 1$ and is a Riesz basis if and only if it is a frame and $ab = 1$. Frames are of particular interest in the investigation of Gabor systems not only because they allow for more flexibility in the decomposition of functions, but also because no "well-localized" Gabor Riesz basis exists. This result is known as the *Balian-Low theorem.* We mainly consider Gabor systems that constitute at least a Bessel sequence, if not a frame.

Theorem 1 (Balian (1981) [12] and Low (1985) [114])**.** *If $\mathcal{G}(g, a, b)$ constitutes a Riesz basis for $L^2(\mathbb{R})$, then*

$$\int_{\mathbb{R}} t^2 |g(t)|^2 \, dt \int_{\mathbb{R}} \xi^2 |\hat{g}(\xi)|^2 \, d\xi = \infty.$$

In other words, if g generates a Gabor basis, then it maximizes Heisenberg's uncertainty principle.

Variations on the Balian-Low theorem exist, e.g. [84, Thm 8.4.1], each of them amounting to the same problem: In order to have a stable analysis/synthesis system of Gabor type, generated by a well-localized prototype function (window), redundancy is necessary. The frame (or Bessel) coefficients of the Gabor system $\mathcal{G}(g, a, b)$ correspond to samples of the short-time Fourier transform

$$\mathbf{C}_{\mathcal{G}(g,a,b)}(f)[n, m] = \langle f, \mathbf{M}_{mb}\mathbf{T}_{na}g\rangle = \mathcal{V}_g f(na, mb), \text{ for all } f \in L^2(\mathbb{R}^d).$$

The synthesis operator on the other hand constructs a function as linear combination of time-frequency shifted copies of a single prototype function

$$\mathbf{D}_{\mathcal{G}(g,a,b)}(c) = \sum_{n,m \in \mathbb{Z}^d} c[n, m]\mathbf{M}_{mb}\mathbf{T}_{na}g, \text{ for all } c \in \ell^2(\mathbb{Z}^{2d}).$$

A central property of Gabor frames $\mathcal{G}(g, a, b)$ is the fact that dual Gabor frames of the form $\mathcal{G}(h, a, b)$ with $h \in L^2(\mathbb{R}^d)$ and the same constants $a, b \in \mathbb{R}^+$ exist. In particular, the canonical dual frame of $\mathcal{G}(g, a, b)$ is generated from the *canonical dual window* $\tilde{g} = \mathbf{S}^{-1}g$ and given by $\mathcal{G}(\tilde{g}, a, b)$.

Besides this fact, Gabor systems possess many structural properties that are not inherent to more general Bessel sequences, among them two very useful representations of the *Gabor frame operator.* Their proofs can be found e.g. in Gröchenig's book [84].

Proposition 2 (Walnut (1992) [165])**.** *Let $\mathcal{G}(g,a,b)$ and $\mathcal{G}(h,a,b)$, with $g,h \in L^2(\mathbb{R})$ and $a,b \in \mathbb{R}^+$, be Gabor systems. If either*

(i) $g,\ h \in W(L^\infty, \ell^1, \mathbb{R}^d)$, or

(ii) $\mathcal{G}(g,a,b)$ and $\mathcal{G}(h,a,b)$ are Bessel sequences,

then the associated frame-type operator *(or* mixed frame operator*)* $\mathbf{S}_{g,h} := \mathbf{S}_{g,h,a,b} := \mathbf{D}_{\mathcal{G}(h,a,b)}\mathbf{C}_{\mathcal{G}(g,a,b)}$ *admits a Walnut representation of the form*

$$\mathbf{S}_{g,h}f = b^{-d} \sum_{n,k\in\mathbb{Z}^d} \mathbf{T}_{na}\left(h\mathbf{T}_{kb^{-1}}\overline{g}\right)\mathbf{T}_{kb^{-1}}f, \quad \textit{for all } f \in L^2(\mathbb{R}). \tag{2.27}$$

Setting $h = g$ yields the Walnut representation for the frame operator.

The proof in [84] only needs minor modifications to also be valid in the Bessel case.

Theorem 2 (Janssen (1995) [98])**.** *Let $\mathcal{G}(g,a,b)$ and $\mathcal{G}(h,a,b)$, with $g,h \in L^2(\mathbb{R})$ and $a,b \in \mathbb{R}^+$, be Gabor systems such that*

$$\sum_{m,n\in\mathbb{Z}^d} |\langle h, \mathbf{M}_{m/a}\mathbf{T}_{n/b}g\rangle| < \infty.$$

Then the frame-type operator $\mathbf{S}_{g,h} = \mathbf{D}_{\mathcal{G}(h,a,b)}\mathbf{C}_{\mathcal{G}(g,a,b)}$ admits a Janssen representation:

$$\mathbf{S}_{g,h}f = (ab)^{-d} \sum_{m,n\in\mathbb{Z}^d} \langle h, \mathbf{M}_{m/a}\mathbf{T}_{n/a}g\rangle \mathbf{M}_{m/a}\mathbf{T}_{n/b}f. \tag{2.28}$$

Setting $h = g$ yields the Janssen representation for the frame operator.

In words, Walnut's representation writes the frame operator as a linear combination of weighted translates, with weight functions $b^{-d}\sum_{n,k\in\mathbb{Z}^d}\mathbf{T}_{na}\left(h\mathbf{T}_{kb^{-1}}\overline{g}\right)$, while Janssen's representation is in terms of time-frequency translates on the *adjoint lattice*, see Chapter 3, with scalar weights $\langle h, \mathbf{M}_{m/a}\mathbf{T}_{n/a}g\rangle$.

The Walnut representation is closely related to duality conditions for Gabor systems [99], i.e. two Gabor Bessel sequences $\mathcal{G}(g,a,b)$ and $\mathcal{G}(h,a,b)$ form dual frames if and only if

$$b^{-d}\sum_{n\in\mathbb{Z}^d}\mathbf{T}_{na}\left(h\overline{g}\right) \equiv 1 \tag{2.29}$$

and

$$b^{-d} \sum_{n,k\in\mathbb{Z}^d} \mathbf{T}_{na} \left(h\mathbf{T}_{kb^{-1}}\overline{g}\right) \equiv 0, \text{ for all } k \in \mathbb{Z}^d \setminus \{0\}. \tag{2.30}$$

In Chapter 4 we will discuss a generalization of Walnut's representation to nonstationary Gabor systems and some of its implications on the structure of such systems.

A simple sufficient condition for a Gabor system to form a frame exists if the so-called *painless case* conditions, first introduced in by Daubechies, Grossmann and Meyer [48], are met. The painless case describes Gabor systems with compactly supported generator, i.e. $\text{supp}(g) \subseteq [c,d]$ for some $c,d \in \mathbb{R}$ with $c < d$ and dense frequency sampling $b \leq (d-c)^{-1}$. We only give the result for the one dimensional case.

Proposition 3 (Daubechies, Grossmann, Meyer (1986))**.** *Let* $g \in L^2(\mathbb{R})$ *such that* $\text{supp}(g) \subseteq [c,d]$ *for some* $c,d \in \mathbb{R}$ *with* $c < d$ *and* $b \leq (d-c)^{-1}$. *The Gabor system* $\mathcal{G}(g,a,b)$ *is a frame with frame bounds* $0 < A \leq B < \infty$ *if and only if*

$$A \leq b^{-1} \sum_{n\in\mathbb{Z}} |T_{na}g(t)|^2 \leq B, \text{ for a.e. } t \in [0,a). \tag{2.31}$$

In this setting, a quick glance at the Walnut representation shows that the frame operator is diagonal and the canonical dual window is obtained by pointwise division of g by the inner term of (2.31). In fact, all computations simplify tremendously in this setting, to the point that most uses of Gabor systems in application restrict themselves to this case.

The "right" function spaces for time-frequency analysis are the so-called modulation spaces $M^{p,q}_w$ that can be defined as the tempered distributions f such that their STFT with regards to a window g in the Schwartz space of rapidly decaying functions is in the weighted space $L^{p,q}_w$, i.e.

$$\int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} |\mathcal{V}_g f(x,\omega)|^p w(x,\omega)^p \, dx \right)^{q/p} d\omega < \infty.$$

A particularly useful space among them is $M^1 := M^{1,1}_1$, also called *Feichtinger's algebra* [61]. Choosing a function from M^1 as prototype for a Gabor system, one can avoid many of the usual technicalities involved in handling Gabor frames and show additional properties that are not valid for general window functions in $L^2(\mathbb{R}^d)$. It contains e.g. the Schwartz functions and any compactly supported and continuously differentiable function. Modulation spaces have been introduced by Feichtinger [63], for more information see [84, Ch 11].

In Chapter 3 we treat Gabor systems on lattices. In that context, we will use *multiwindow* Gabor systems, a simple generalization of Gabor systems [174, 176].

They are essentially a union of finitely many Gabor systems $\mathcal{G}(g_1,a,b),\ldots,\mathcal{G}(g_k,a,b)$ with the same translation and modulation parameters a,b. They retain most of the structure of Gabor systems, e.g. the Walnut and Janssen representations of the frame operator.

2.3.1 Discrete Gabor analysis

In the finite discrete case, we take the Hilbert space $\mathcal{H}$ to be $\mathbb{C}^L$ or a product of such spaces. For a good introduction to Gabor analysis in this setting, see [150]. We shall restrict the lattice parameters a and b to factors of L such that the numbers $N=\frac{L}{a}$ and $M=\frac{L}{b}$ are integers.

Thus, a discrete Gabor system is a finite sequence of functions of the form

$$\mathcal{G}(g,a,b)=\left\{\mathbf{M}_{bm}\mathbf{T}_{an}g : n=0,\ldots,N-1; m=0,\ldots,M-1\right\},$$

which is a collection of $M\cdot N$ vectors in $\mathbb{C}^L$. Obviously, to fulfill the frame conditions (2.17), we need at least $M\cdot N\geq L$. A multidimensional discrete Gabor system is defined analogously.

With the notation $g_{n,m}=\mathbf{M}_{bm}\mathbf{T}_{an}g$, the discrete Gabor transform of $f\in\mathbb{C}^L$ with respect to the Gabor system $\mathcal{G}(g,a,b)$ is given by

$$c[n,m]=\langle f,g_{n,m}\rangle=\sum_{l=0}^{L-1} f[l]\overline{g[l-na]}e^{-2\pi i ml/M}=\sqrt{L}\widehat{f\overline{\mathbf{T}_{na}g}}[mb] \tag{2.32}$$

and the corresponding discrete Gabor synthesis is

$$\tilde{f}[l]=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} c[n,m]g_{n,m}[l]=\sqrt{L}\sum_{n=0}^{N-1}\left(c[n,\cdot]\mathbf{T}_{na}g\right)^{\vee}[l], \tag{2.33}$$

where $\left(c[n,\cdot]\mathbf{T}_{na}g\right)^{\vee}$ is a sequence in $\mathbb{C}^M$ with cyclic indexing. If $a=b=1$, Equations (2.32) and (2.33), the latter equipped with the normalization factor $\left(L\|g\|^2\right)^{-1}$ are also referred to as the discrete short-time Fourier transform and inverse discrete short-time Fourier transform, provided g denotes the same function in (2.32) and (2.33). In this case, they provide perfect reconstruction for all $f\in\mathbb{C}^L$.

The theory of discrete Gabor frames is mostly analogous to the continuous theory presented above with the usual modifications. The painless case construction, Walnut and Janssen representations etc. can be deduced for the discrete case in much the same way, but with the simplification that function spaces do not play a role anymore. There are situations where the discreteness and the fact that discrete systems are treated periodically, present some additional problems, e.g. in the case of nonseparable Gabor schemes, see Chapter 3.

2.4 Wavelet systems

While we will not work with Wavelet systems directly, they are both an important tool for time-frequency, or rather time-scale, analysis and a particular case of a frequency-side nonstationary Gabor system, see Chapter 4. Therefore, we briefly sketch the one-dimensional continuous wavelet transform. More information can be found, e.g. in the standard works on Wavelets [47] and [116].

Let $\varphi \in L^2(\mathbb{R})$ with zero average. The wavelet transform of $f \in L^2(\mathbb{R})$ with respect to φ is defined as

$$\mathbf{W}_\varphi f(u,s) = \int_{\mathbb{R}} s^{-1/2} f(t) \overline{\varphi\left(\frac{t-u}{s}\right)}\, dt = \langle f, \mathbf{T}_u \mathbf{D}_s \varphi \rangle,\ u \in \mathbb{R}, s \in \mathbb{R}^+. \tag{2.34}$$

The formula above can also be expressed as a convolution $\left(f * \mathbf{D}_s \overline{\varphi(-\cdot)}\right)(u)$, showing that the Wavelet coefficients are the coefficients corresponding to some linear filterbank.

The collection of atoms

$$\mathcal{W}(\varphi) := \{\varphi_{u,s} = \mathbf{T}_u \mathbf{D}_s \varphi\}_{u\in\mathbb{R}, s\in\mathbb{R}^+}\,, \tag{2.35}$$

is called a (continuous) Wavelet system.

If φ is localized around t_c, then $\varphi_{u,s}$ is centered at $st_c + u$, with center frequency ξ_c/s, where ξ_c is the center frequency of $\hat{\varphi}$.

A Wavelet φ is called *admissible* with admissibility constant $C < \infty$, if it satisfies

$$\int_{\mathbb{R}} \xi^{-1} |\hat{\varphi}(\xi)|^2 \, d\xi = C. \tag{2.36}$$

For any admissible Wavelet $\varphi \in L^2(\mathbb{R})$, all $f \in L^2(\mathbb{R})$ can be reconstructed from their respective Wavelet coefficients. Explicitly,

$$f = C^{-1} \int_{\mathbb{R}^+} \int_{\mathbb{R}} s^{-2} \langle f, \mathbf{T}_u \mathbf{D}_s \varphi \rangle \mathbf{T}_u \mathbf{D}_s \varphi \; du\; ds. \tag{2.37}$$

There are two standard discretization schemes for Wavelet transforms. Let $a, b \in \mathbb{R}^+$ with $a > 1$, then the dilation-first discretization uses the atoms

$$\theta_{n,m} = \varphi_{nb,a^m} = \varphi\left(\frac{t-nb}{a^m}\right),\ \text{for } m,n \in \mathbb{Z},$$

while translation-first discretization uses the atoms

$$\phi_{n,m} = \varphi\left(\frac{t}{a^m} - nb\right) = \varphi_{a^m nb, a^m} = \mathbf{D}_{a^m} \mathbf{T}_{nb} \varphi,\ \text{for } m,n \in \mathbb{Z}.$$

The second discretization scheme seems more suited to the structure of Wavelet transforms, but the first scheme has the benefit of constituting a shift invariant system. It should be noted that, in the continuous case, most traditionally used Wavelets are constructed on the Fourier side, one reason being the ease with which the admissibility condition can be verified.

Compared to Gabor frames, the results on Wavelet frames and their implementation are relatively scarce. It is well known that the canonical dual frame to a Wavelet frame is in general not a Wavelet frame, but duality conditions similar to (2.29) and (2.30) exist [41], as well as a painless case result [48]. These results usually put conditions on the Fourier transform of the involved Wavelet, making use of Parseval's formula and the duality of translation and modulation under the Fourier transform, i.e.

$$\langle f, \mathbf{T}_u \mathbf{D}_a \varphi \rangle = \langle \hat{f}, \widehat{\mathbf{T}_u \mathbf{D}_a \varphi} \rangle = \langle \hat{f}, \mathbf{M}_{-u} \mathbf{D}_{a^{-1}} \hat{\varphi} \rangle.$$

We will also use this correspondence to express sampled Wavelet systems as nonstationary Gabor systems. This is interesting as well for obtaining a discretized Wavelet transform for signals in $\mathbb{C}^L$ that resembles a continuous Wavelet transform more than the classical dyadic discrete Wavelet transform (DWT) based on a multiresolution analysis.

Handling Wavelet transforms on $\mathbb{C}^L$ is a more delicate matter than discretizing Gabor transforms, since the dilation operator is not always well-defined on $\mathbb{C}^L$. Even if it is, it amounts to a structured permutation, which is not particularly useful from a time-frequency analysis point of view. We propose a possible workaround for these problems in Section 5.4.

Chapter 3

Gabor systems on arbitrary lattices

In Section 2.3, we introduced Gabor systems and Gabor frames on a rectangular sampling set, i.e. the Gabor coefficients were taken to be samples of the STFT on sets of the form $a\mathbb{Z}^d \times b\mathbb{Z}^d$. Alternatively, these coefficients can be interpreted as the inner products with time-frequency atoms centered at positions $(an, bm)_{n,m\in\mathbb{Z}^d}$. Now one might be interested in Gabor systems on more general sampling sets which might be better suited for commonly used window functions, see Figure 3.1.

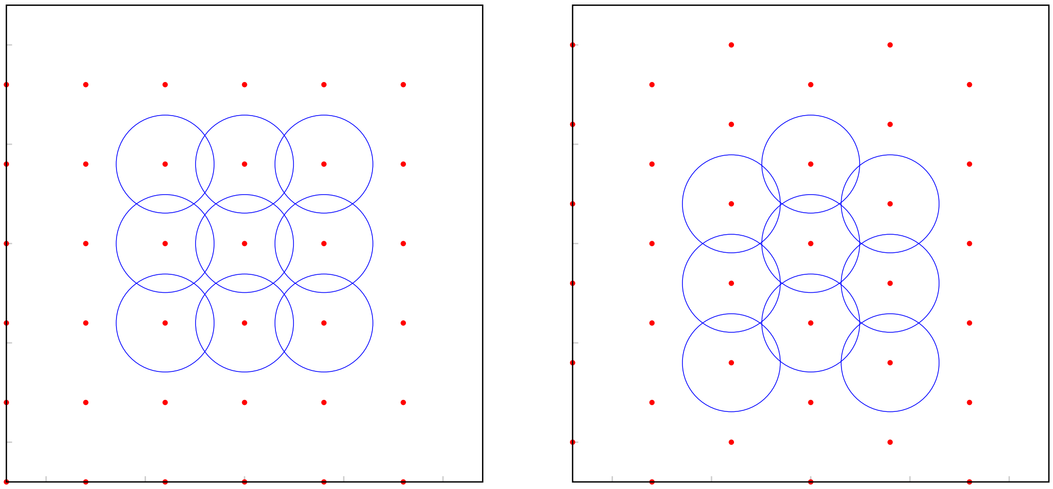


Figure 3.1: An atom with circular (Gaussian) time-frequency concentration yields a better covering of the time-frequency plane for a non-rectangular sampling.

3.1 Gabor systems on lattices

Gabor systems on arbitrary sampling sets give up a large amount of structure and while the subject has received attention [64], [72], [151], [152], [29], [83], most results amount to density conditions. On the other hand, Gabor systems on discrete subgroups, also called *lattices* [84] allow for a good amount of flexibility while being rich in structure and, considering the uniform time-frequency concentration of the Gabor atoms, also encompassing the most sensible sampling sets for Gabor analysis. In particular, it has recently been conjectured that for a standard Gaussian window the best sampling strategy is a regular hexagonal pattern [5].

As a lattice over $\mathbb{R}^{2d}$ we consider every subgroup $G \leq \mathbb{R}^{2d}$. The group G can also be written as $\Lambda\mathbb{Z}^{2d}$ with a positive, real-valued, invertible $2d \times 2d$ matrix Λ and is called *(fully) separable* if Λ can be chosen as a diagonal matrix, *time-frequency separable* if

$$\Lambda = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}, \text{ with } A, B \in \mathbb{R}^{d\times d},$$

dimensionally separable if Λ can be chosen such that

$$\Lambda[l,j] = 0 \text{ for all } l,j = 0,\dots,2d-1 \text{ with } (l-j) \bmod d \neq 0,$$

and *fully nonseparable* otherwise. We will use the notation Λ synonymous for the $2d \times 2d$ matrix Λ and the lattice $\Lambda\mathbb{Z}^{2d}$ and refer to Λ as *nonseparable* if it is not fully separable. Continuous Gabor systems on lattices have been considered, e.g. in [68], [70], [160], [17] and [84, Ch 9.4]. While our focus is on establishing state of the art techniques for the computation of Gabor systems over $\mathbb{C}^L$, we will review the essential continuous (and multidimensional) theory. Unless otherwise noted, the material in this review section can be found in [84]. We will denote by $\mathrm{GL}_{2d}(\mathbb{R})$ and $\mathrm{SL}_{2d}(\mathbb{R})$ the invertible matrices and matrices with determinant 1 over $\mathbb{R}^{2d}$, respectively.

For ease of notation, we will denote by $\pi(\lambda) = \mathbf{M}_{\lambda_2}\mathbf{T}_{\lambda_1}$ the time-frequency shift by $\lambda \in \mathbb{R}^{2d}$. The Gabor system on $\Lambda\mathbb{Z}^{2d}$ with respect to the window function $g \in L^2(\mathbb{R}^d)$ is the collection of functions

$$\mathcal{G}(g,\Lambda) := \{g_\lambda = \pi(\lambda)g \ : \ \lambda \in \Lambda\}. \tag{3.1}$$

An important connection is the one between a lattice Λ and its so-called *adjoint lattice* Λ°.

Definition 8. Let Λ be a lattice. The adjoint lattice Λ° of Λ is the set of points $\lambda^\circ \in \mathbb{R}^{2d}$ such that

$$\pi(\lambda^\circ)\pi(\lambda) = \pi(\lambda)\pi(\lambda^\circ), \text{ for all } \lambda \in \Lambda. \tag{3.2}$$

For a fully separable lattice Λ, the adjoint lattice is given by the diagonal matrix Λ° with $\Lambda^\circ[l,l] = 1/\Lambda[l+d, l+d]$ and $\Lambda^\circ[l+d, l+d] = 1/\Lambda[l,l]$ for $l = 0, \ldots, d-1$. Generally, see [70], the adjoint lattice can be obtained as

$$\Lambda^\circ = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \Lambda^{-T},$$

where Λ^{-T} denotes the inverse of the transpose Λ^T of Λ.

The adjoint lattice plays an important role in the generalized versions of both the Ron-Shen duality principle and Janssen's representation, each proven in [73]. The Ron-Shen duality principle implies that the frame property of $\mathcal{G}(g,\Lambda)$ is equivalent to $\mathcal{G}(g,\Lambda^\circ)$ being a Riesz basis for its closed span, while

$$\mathbf{S}_{g,h,\Lambda} f = |\det(\Lambda)|^{-d} \sum_{\lambda \in \Lambda^\circ} \langle h, \pi(\lambda) g \rangle \pi(\lambda) f, \text{ for all } f \in L^2(\mathbb{R}^d) \tag{3.3}$$

is the general form of Janssen's representation. There are multiple ways to express Gabor systems on general lattices as transformations of Gabor systems on a rectangular sampling set and/or in a single dimension. We will see that all dimensionally separable Gabor systems can be expressed as fully separable Gabor system and, provided that the window function g is a d-dimensional tensor product, i.e. $g(t) = g_1(t_1) \otimes \ldots \otimes g_d(t_d)$ handled with one-dimensional techniques. See [34] for more information on the correspondence between one- and multidimensional Gabor systems. On the other hand, no significant results on time-frequency separable lattices seem to exist.

We present two simple approaches to establish a separable to nonseparable correspondence, namely the multiwindow technique and the correspondence via the metaplectic representation.

The multiwindow method has been proposed by Zibulski and Zeevi [176] and Feichtinger et al. [70]. Let Λ be a lattice. $\mathcal{G}(g,\Lambda)$ can be written as a separable multiwindow Gabor system if there is a finite number of elements $\lambda_1, \ldots, \lambda_k \in \Lambda$ and a separable lattice $\tilde{\Lambda}$, such that, with $\lambda_0 = 0$

$$\Lambda = \bigcup_{j=0}^{k} \lambda_j + \tilde{\Lambda}. \tag{3.4}$$

In this case,

$$\mathcal{G}(g,\Lambda) = \bigcup_{j=0}^{k} \mathcal{G}(\pi(\lambda_j) g, \tilde{\Lambda}) \tag{3.5}$$

holds up to phase factors.

Consequently, the frame operator of $\mathcal{G}(g,\Lambda)$ is given as the sum of the frame operators $\mathbf{S}_{\pi(\lambda_j)g,\tilde{\Lambda}}$ and as such possesses a Walnut-representation and other structural properties inherited from separable Gabor systems. As for the question when a lattice can be decomposed in this way, to the best of our knowledge there has not been any exhaustive study. For the one dimensional case however, there exists a quite simple sufficient condition with a constructive proof.

Lemma 1. *Let Λ be a lattice. If lower triagonal $L \in GL_2(\mathbb{R})$ and integer valued $P \in SL_2(\mathbb{R})$ exist such that $\Lambda = LP$ and*

$$L = \begin{pmatrix} a & 0 \\ s & b \end{pmatrix},$$

with $s/b \in \mathbb{Q}$, then there exist $\lambda_0 = 0$ and $\lambda_1, \ldots, \lambda_k \in \Lambda$, $k \in \mathbb{N}_0$ and a separable lattice $\tilde{\Lambda}$, such that

$$\Lambda = \bigcup_{j=0}^{k} \lambda_j + \tilde{\Lambda}.$$

Proof. If $s = 0$, then Λ is separable and there is nothing to show. Otherwise note that P is integer-valued and $|\det(P)| = 1$, thus applying P to $\mathbb{Z}^2$ can be thought of a permutation, i.e. $P\mathbb{Z}^2 = \mathbb{Z}^2$. There are unique $p \in \mathbb{Z}$ and $q \in \mathbb{N}$ that are coprime and satisfy $s/b = p/q$. Consequently, with

$$\tilde{\Lambda} = \begin{pmatrix} qa & 0 \\ 0 & b \end{pmatrix},$$

we have the equality

$$\Lambda = \bigcup_{j=0}^{q-1} j(a,s)^T + \tilde{\Lambda},$$

concluding the proof. □

The same idea can be applied using a UP factorization with U upper triagonal and $P \in \mathrm{SL}_{2d}(\mathbb{R})$ integer valued, potentially increasing the class of lattices allowing for such a decomposition. Moreover, Lemma 1 can easily be generalized to dimensionally separable lattices, if applied to the submatrices

$$\begin{pmatrix} \Lambda[l,l] & \Lambda[l,l+d] \\ \Lambda[l+d,l] & \Lambda[l+d,l+d] \end{pmatrix}, \text{ for all } l = 0, \ldots, d-1.$$

Note that the expression of one dimensional, nonseparable Gabor systems through separable Gabor schemes is also considered in the work of Bastiaans and van Leest, e.g. [14, 160]. However, they assume that the lattice can be written in the form $\Lambda = UL$, where U is a diagonal real-valued matrix and L is an integer matrix in Hermite normal form. Consider the lattice

$$\Lambda = \begin{pmatrix} 1 & 0 \\ \sqrt{2} & 1 \end{pmatrix}$$

to see that this is not always possible.

Metaplectic transformations can also be used to provide a connection between separable and nonseparable Gabor systems. The continuous theory of metaplectic representation is well summarized in [84, Ch 9.4]. Loosely speaking, there is a subgroup of $\mathrm{SL}_{2d}(\mathbb{R})$, called *symplectic matrices*, such that their action on a vector in $\mathbb{R}^{2d}$ is, in a geometric sense, translated to the time-frequency plane by a unitary operator on $L^2(\mathbb{R}^d)$. More explicitly, a matrix $M \in \mathrm{GL}_{2d}(\mathbb{R})$ is symplectic if and only if

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \text{ for some } A, B, C, D \in \mathbb{R}^{d\times d}, \tag{3.6}$$

with $AC^* = A^*C$, $BD^* = B^*D$ and $A^*D - C^*B = I$, implying $|\det(M)| = 1$. The set of symplectic matrices forms a group under composition, denoted by $\mathrm{Sp}_d(\mathbb{R})$.

If M is symplectic, then there exists a unitary operator $\mu_M \ : \ L^2(\mathbb{R}^d) \mapsto L^2(\mathbb{R}^d)$, such that, with $(\tilde{x}, \tilde{\omega})^T = M^{-1}(x, \omega)^T$,

$$\mathcal{V}_g f(\tilde{x}, \tilde{\omega}) = e^{\pi i(\langle x,\omega\rangle - \langle \tilde{x},\tilde{\omega}\rangle)} \mathcal{V}_{\mu_M g}\mu_M f(x, \omega), \text{ for all } f, g \in L^2(\mathbb{R}^d). \tag{3.7}$$

In words, the short-time Fourier transform of f with respect to g evaluated at $\lambda \in \mathbb{R}^{2d}$ equals the short-time Fourier transform of $\mu_M f$ with respect to $\mu_M g$ evaluated at $M\lambda$. We call μ_M the *metaplectic operator* corresponding to M. This motivates the following definition.

Definition 9. A lattice $\Lambda \leq \mathbb{R}^{2d}$ is called *symplectic*, if there are $c \in \mathbb{R} \setminus \{0\}$ and $M \in \mathrm{Sp}_d(\mathbb{R})$, such that

$$\Lambda = cM\mathbb{Z}^{2d}. \tag{3.8}$$

It is easy to show that, provided $\mathcal{G}(g, c, c)$ and $\mathcal{G}(h, c, c)$ are Bessel sequences, and $\Lambda = cM\mathbb{Z}^{2d}$ is a symplectic lattice, the following holds.

$$S_{g,h,c,c} = \mu_M^{-1} S_{\mu_M g, \mu_M h, \Lambda}\ \mu_M.$$

This in turn implies that $\mathcal{G}(\mu_M g, \Lambda)$ is a frame with frame bounds A, B if and only if $\mathcal{G}(g, c, c)$ is. Therefore all the properties and structure of the separable case (Janssen's representation, Ron-Shen duality principle, etc.) immediately carry over in some sense, we even obtain a Walnut representation of some sort. If μ_M

and μ_M^{-1} are known, both analysis and synthesis on the nonseparable lattice can be performed via the separable lattice.

However, if a lattice is given in a specific form $\Lambda\mathbb{Z}^{2d}$ and Λ is not a scalar multiple of a symplectic matrix, determining whether Λ specifies a symplectic lattice is a nontrivial task and equivalent to the existence of an integer valued matrix $P \in \mathrm{SL}_{2d}(\mathbb{R})$ such that $\Lambda = cMP$ with c and M as in Definition 9.

On the other hand, the case $d = 1$ is somewhat simpler again. It is easy to see that the conditions on symplectic matrices (3.6) reduce to $\det(M) = 1$ and that any nonsingular $\Lambda \in \mathrm{GL}_2(\mathbb{R})$ can be decomposed as cM with $c \neq 0$, $|\det(M)| = 1$. Consequently, every lattice over $\mathbb{R}^2$ is symplectic and can be deformed into a separable lattice using symplectic transformations.

The question remains how to determine the metaplectic operator associated to a symplectic matrix M. There are certain elementary symplectic matrices, where the corresponding operator is explicitly known. These are the standard symplectic form

$$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix},$$

corresponding to the Fourier transform, the unitary dilations

$$\begin{pmatrix} A & 0 \\ 0 & A^{-1} \end{pmatrix},$$

with A invertible, associated with $\mathbf{D}_{A^{-1}}f = |det(A)|^{d/2}f(A\cdot)$ and the skew (or shear) matrices

$$\begin{pmatrix} I & 0 \\ C & I \end{pmatrix},$$

associated with the chirp multiplication $\mathbf{R}_C f(t) = f(\cdot)\exp(\pi i t^T C t)$. Kaiblinger and Neuhauser have shown [103, Thm 2] that the metaplectic operator associated to every symplectic matrix can be decomposed into a combination of elementary metaplectic operators. Their decomposition requires 2 Fourier transforms, 3 chirp multiplications and a dilation. For $d = 1$, a simpler procedure is possible if $\Lambda = LP$ as in Lemma 1.

Lemma 2. *Let Λ be a lattice. Λ is a symplectic lattice if lower triagonal $L \in GL_2(\mathbb{R})$ and integer valued $P \in SL_2(\mathbb{R})$ exist such that $\Lambda = LP$ and*

$$L = \begin{pmatrix} a & 0 \\ s & b \end{pmatrix}.$$

Proof. To prove the lemma, confirm $P\mathbb{Z}^2 = \mathbb{Z}^2$ and either $\tilde{L} = (ab)^{-1/2}L$ or $\tilde{L}\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ is symplectic. □

In the case $d = 1$, we will discuss multiwindow and metaplectic methods for "rectifying" lattices in more detail for the discrete, finite case, where several problems are introduced due to the circular nature of $\mathbb{Z}_L$. This is done in Section 3.3, where we also mention some simple extensions to the multidimensional case of signals in $C^{L_1} \times \ldots \times C^{L_k}$. Before that, the next section discusses some general properties of lattices over $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$.

3.2 On the subgroups of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$

This section contains partial results from joint work with Mario Hampejs, László Tóth and Christoph Wiesmeyr, available as preprint [154].

Let $\mathbb{Z}_{L_1}$ be the group of residue classes modulo L_1 and consider the direct product $G = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$, where L_1 and L_2 are arbitrary positive integers. We aim to discuss properties of the subgroups of G and to derive simple formulae for the number of certain types of subgroups of G, including the total number $s(L_1, L_2)$ of its subgroups and the number $s_k(L_1, L_2)$ of its subgroups of order k, where $k \mid L_1 L_2$.

Subgroups of $\mathbb{Z} \times \mathbb{Z}$ (sublattices of the two dimensional integer lattice) and associated counting functions were considered by several authors in pure and applied mathematics. It is known, for example, that the number of subgroups of index[1] n in $\mathbb{Z}\times\mathbb{Z}$ is $\sigma(n)$, the sum of the (positive) divisors of n. See, e.g., [80], [179], [3, item A001615]. Although features of the subgroups of G are not only interesting by their own but also have applications, one of them the description of Gabor lattices, it seems that a synthesis on subgroups of G can not be found in the literature.

In the case $L = L_1 = L_2$ the subgroups of $\mathbb{Z}_L \times \mathbb{Z}_L$ play an important role in discrete, finite time-frequency analysis. As discussed in Section 2.3, time-frequency analysis attempts to investigate function behavior via a phase space representation given by the short-time Fourier transform [84]. The phase space corresponding to discrete, finite functions (or vectors) belonging to $\mathbb{C}^L$ is exactly $\mathbb{Z}_L \times \mathbb{Z}_L$. Concerned with the question of reconstruction from samples of short-time Fourier transforms, it has been found that when sampling on lattices, i.e. subgroups of $\mathbb{Z}_L \times \mathbb{Z}_L$, the associated analysis and reconstruction operators are particularly rich in structure, which, in turn, can be exploited for efficient implementation, cf. [105], [158], [150] and references therein. It is of particular interest to find subgroups in a certain range of cardinality, therefore a complete characterization of these groups helps choosing the best one for the desired application.

It is known that for every finite Abelian group the problem of counting all

[1] The number of cosets of the subgroup contained in the group.

subgroups and the subgroups of a given order reduces to p-groups, which follows from the properties of the subgroup lattice of the group (see R. Schmidt [139], M. Suzuki [153]). In particular, for $G = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ this can be formulated as follows. Assume that $\gcd(L_1, L_2) > 1$. Then G is an Abelian group of rank two, since G is isomorphic to $\mathbb{Z}_u \times \mathbb{Z}_v$, where $u = \operatorname{lcm}(L_1, L_2)$, $v = \gcd(L_1, L_2)$. Let $u = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ and $v = p_1^{\beta_1} \cdots p_r^{\beta_r}$ be the prime power factorizations of u and v, respectively, where $\alpha_j \geq \beta_j \geq 0$ $(1 \leq j \leq r)$. Then

$$s(L_1, L_2) = \prod_{j=1}^{r} s(p_j^{\alpha_j}, p_j^{\beta_j}), \tag{3.9}$$

and

$$s_k(L_1, L_2) = \prod_{j=1}^{r} s_{k_j}(p_j^{\alpha_j}, p_j^{\beta_j}), \tag{3.10}$$

where $k = k_1 \cdots k_r$ and $k_j = p_j^{\gamma_j}$ with some exponents $0 \leq \gamma_j \leq \alpha_j + \beta_j$ $(1 \leq j \leq r)$.

Now consider the p-group $\mathbb{Z}_{p^\alpha} \times \mathbb{Z}_{p^\beta}$, where $\alpha \geq \beta \geq 0$. This is of rank two for $\alpha \geq \beta \geq 1$. One has the simple explicit formulae:

$$s(p^\alpha, p^\beta) = \frac{(\alpha - \beta + 1)p^{\beta+2} - (\alpha - \beta - 1)p^{\beta+1} - (\alpha + \beta + 3)p + (\alpha + \beta + 1)}{(p-1)^2}, \tag{3.11}$$

$$s_{p^\gamma}(p^\alpha, p^\beta) = \begin{cases} \frac{p^{\gamma+1}-1}{p-1}, & \gamma \leq \beta \leq \alpha, \\ \frac{p^{\beta+1}-1}{p-1}, & \beta \leq \gamma \leq \alpha, \\ \frac{p^{\alpha+\beta-\gamma+1}-1}{p-1}, & \beta \leq \alpha \leq \gamma \leq \alpha + \beta. \end{cases} \tag{3.12}$$

Formula (3.11) was derived by G. Călugăreanu [45, Sect. 4] and recently by J. Petrillo [126, Prop. 2] using Goursat's lemma for groups. M. Tărnăuceanu [157, Prop. 2.9], [156, Th. 3.3] deduced (3.11) and (3.12) by a method based on properties of certain attached matrices.

Therefore, $s(L_1, L_2)$ and $s_k(L_1, L_2)$ can be computed using (3.9), (3.11) and (3.10), (3.12), respectively. We deduce other formulae for $s(L_1, L_2)$ and $s_k(L_1, L_2)$ (Theorems 3 and 4), which generalize (3.11) and (3.12), and put them in more compact forms. These are consequences of a simple representation of the subgroups of $G = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$, given in Theorem 3. This representation might be known, but we could not locate it in the literature.

Our approach is elementary, using only simple group-theoretic and number-theoretic arguments. The proofs are given in Section 3.2.2.

Throughout this section we use the notation $\sigma(n)$ for the sum of the positive divisors of n.

3.2.1 Results

The subgroups of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ can be identified and visualized in the plane with sublattices of the lattice $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$. Every two dimensional sublattice is generated by two basis vectors. For example, the Figure shows the subgroup of $\mathbb{Z}_{12} \times \mathbb{Z}_{12}$ having the basis vectors $(1,2)^T$ and $(0,6)^T$.

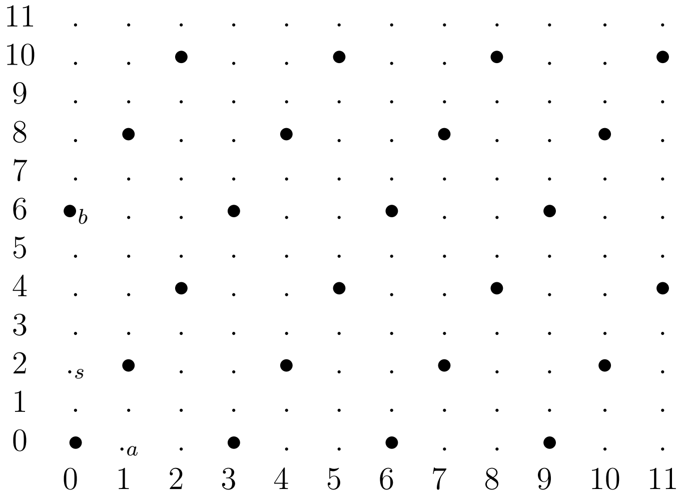


Figure 3.2: Geometric representation of a lattice.

This suggests the following representation of the subgroups:

Theorem 3. *For every $L_1, L_2 \in \mathbb{N}$ let*

$$I_{L_1,L_2} := \{(a,b,t) \in \mathbb{N}^2 \times \mathbb{N}_0 : a \mid L_1, b \mid L_2, 0 \leq t \leq \gcd(L_1/a, b) - 1\} \quad (3.13)$$

and for $(a,b,t) \in I_{L_1,L_2}$ define

$$H_{a,b,t} := \{(ia, jb + itb/\gcd(L_1/a, b)) : 0 \leq i \leq L_1/a - 1, 0 \leq j \leq L_2/b - 1\}. \quad (3.14)$$

Then $H_{a,b,t}$ is a subgroup of order $\frac{L_1 L_2}{ab}$ of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ and the map $(a,b,t) \mapsto H_{a,b,t}$ is a bijection between the set I_{L_1,L_2} and the set of subgroups of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$.

Note that for the subgroup $H_{a,b,t}$ the basis vectors mentioned above are $(a,s)^T$ and $(0,b)^T$, where

$$s = \frac{tb}{\gcd(L_1/a, b)}. \quad (3.15)$$

This notation for s will be used also in the rest of the section. Note also that in the case $a \neq L_1$, $b \neq L_2$ the area of the parallelogram spanned by the basis vectors is ab, exactly the index of $H_{a,b,t}$.

We say that a subgroup $H = H_{a,b,t}$ is a subproduct of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ if $H = H_1 \times H_2$, where H_1 and H_2 are subgroups of $\mathbb{Z}_{L_1}$ and $\mathbb{Z}_{L_2}$, respectively.

Theorem 4. *i) The exponent*[2] *of the subgroup $H_{a,b,t}$ is given by*

$$\exp H_{a,b,t} = \frac{L_1 L_2}{\gcd(L_2 a, L_1 s, L_1 b)}. \tag{3.16}$$

ii) The subgroup $H_{a,b,t}$ is cyclic if and only if $\gcd(L_1/a, L_2/b, L_1 s/ab) = 1$.

iii) The subgroup $H_{a,b,t}$ is a subproduct if and only if $t = 0$ and $H_{a,b,0} = \mathbb{Z}_{L_1/a} \times \mathbb{Z}_{L_2/b}$. Here $H_{a,b,0}$ is cyclic if and only if $\gcd(L_1/a, L_2/b) = 1$.

According to Theorem 3, the number $s(L_1, L_2)$ of subgroups of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ can be obtained by counting the elements of the set I_{L_1,L_2}.

Corollary 3. *For every $L_1, L_2 \in \mathbb{N}$, $s(L_1, L_2)$ is given by*

$$s(L_1, L_2) = \sum_{a|L_1, b|L_2} \gcd(a, b). \tag{3.17}$$

Formula (3.17) is a special case of a formula representing the number of all subgroups of a class of groups formed as cyclic extensions of cyclic groups, deduced by W. C. Calhoun [27] and having a laborious proof. Note that formula (3.17) is given, without proof, in [3, item A054584].

Corollary 4. *For every $k, L_1, L_2 \in \mathbb{N}$ such that $k \mid L_1 L_2$,*

$$s_k(L_1, L_2) = \sum_{\substack{a|L_1, b|L_2 \\ L_2 a/b = k}} \gcd(a, b). \tag{3.18}$$

For the case $L = L_1 = L_2$ relevant in time-frequency analysis, this amounts to the next Corollary.

Corollary 5. *For every $k, L \in \mathbb{N}$ such that $k \mid L^2$, the following hold*

$$\exp H_{a,b,t} = \frac{L}{\gcd(a, b, s)}, \tag{3.19}$$

where $H_{a,b,t}$ is cyclic if and only if $\gcd(a, b, s) = ab/L$, *while $H_{a,b,0}$ is cyclic if and only if* $\gcd(a, b) = ab/L$. *Furthermore,*

$$s(L) = s(L, L) = \sum_{a,b|L} \gcd(a, b) \quad \textit{and} \quad s_k(L) = s_k(L, L) = \sum_{\substack{a,b|L \\ La/b = k}} \gcd(a, b). \tag{3.20}$$

[2] The least common multiple of the order of the group elements.

3.2.2 Proofs

Proof. (for Theorem 3) Let H be a subgroup of $G = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$. Consider the natural projection $\pi_1 : G \to \mathbb{Z}_{L_1}$ given by $\pi_1(x, y) = x$. Then $\pi_1(H)$ is a subgroup of $\mathbb{Z}_{L_1}$ and there is a unique divisor a of L_1 such that $\pi_1(H) = \langle a \rangle := \{ia : 0 \le i \le L_1/a - 1\}$. Let $s \ge 0$ be minimal such that $(a, s) \in H$.

Furthermore, consider the natural inclusion $\iota_2 : \mathbb{Z}_{L_2} \to G$ given by $\iota_2(y) = (0, y)$. Then $\iota_2^{-1}(H)$ is a subgroup of $\mathbb{Z}_{L_2}$ and there exists a unique divisor b of L_2 such that $\iota_2^{-1}(H) = \langle b \rangle$.

We show that $H = \{(ia, jb + is) : i, j \in \mathbb{Z}\}$. Indeed, for every $i, j \in \mathbb{Z}$, $(ia, jb + is) = i(a, s) + j(0, b) \in H$. On the other hand, for every $(u, v) \in H$ one has $u \in \pi_1(H)$ and hence there is $i \in \mathbb{Z}$ such that $u = ia$. We obtain $(0, v - is) = (u, v) - i(a, s) \in H$, $v - is \in \iota_2^{-1}(H)$ and there is $j \in \mathbb{Z}$ with $v - is = jb$.

Here a necessary condition is that $(0, sL_1/a) \in H$ (obtained for $i = L_1/a$, $j = 0$), that is $b \mid sL_1/a$, equivalent to $b/\gcd(L_1/a, b) \mid s$. Clearly, if this is verified, then for the above representation of H it is enough to take the values $0 \le i \le L_1/a - 1$ and $0 \le j \le L_2/b - 1$.

Also, dividing s by b we have $s = bq + r$ with $0 \le r < b$ and $(a, r) = (a, s) - q(0, b) \in H$, showing that $s < b$, by its minimality. Hence $s = tb/\gcd(L_1/a, b)$ with $0 \le t \le \gcd(L_1/a, b) - 1$. Thus we obtain the given representation.

Conversely, every $(a, b, t) \in I_{L_1,L_2}$ generates a subgroup $H_{a,b,t}$ of order $L_1L_2/(ab)$ of $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ and the proof is complete. □

Proof. (for Theorem 4) The subgroup $H_{a,b,t}$ is generated by (a, s) and $(0, b)$. Hence the exponent of $H_{a,b,t}$ is the least common multiple of the orders of these two elements. The order of $(0, b)$ is L_2/b. To compute the order of (a, s) note that $L_2 \mid rs$ if and only if $L_2/\gcd(L_2, s) \mid r$. Thus the order of (a, s) is $\operatorname{lcm}(L_1/a, L_2/\gcd(L_2, s))$. We deduce that the exponent of $H_{a,b,t}$ is

$$\exp H_{a,b,t} = \operatorname{lcm}\left(\frac{L_1}{a}, \frac{L_2}{\gcd(L_2, s)}, \frac{L_2}{b}\right) = \operatorname{lcm}\left(\frac{L_1L_2}{L_2a}, \frac{L_1L_2}{L_2\gcd(L_2, s)}, \frac{L_1L_2}{L_1b}\right)$$
$$= \frac{L_1L_2}{\gcd(L_2a, L_2L_1, L_1s, L_1b)} = \frac{L_1L_2}{\gcd(L_2a, L_1s, L_1b)}.$$

ii) Now $H_{a,b,t}$ is cyclic if and only if its exponent equals its order, that is $\frac{L_1L_2}{\gcd(L_2a, L_1s, L_1b)} = \frac{L_1L_2}{ab}$, equivalent to $\gcd(L_1/a, L_2/b, L_1s/ab) = 1$.

iii) Follows at once from Theorem 3. □

Proof. (for Corollary 3) By its definition, the number of elements of the set I_{L_1,L_2} is

$$\sum_{a \mid L_1, b \mid L_2} \sum_{0 \le t \le \gcd(L_1/a, b) - 1} 1 = \sum_{a \mid L_1, b \mid L_2} \gcd(L_1/a, b) = \sum_{a \mid L_1, b \mid L_2} \gcd(a, b),$$

representing $s(L_1, L_2)$. This is formula (3.17). □

Proof. (for Corollary 4) According to Theorem 3,

$$s_k(L_1, L_2) = \sum_{\substack{a|L_1, b|L_2 \\ L_1L_2/ab=k}} \gcd(L_1/a, b),$$

giving (3.18). □

Remark 1. *For any finite groups A and B a subgroup C of $A \times B$ is cyclic if and only if $\iota_1^{-1}(C)$ and $\iota_2^{-1}(C)$ have coprime orders, where ι_1 and ι_2 are the natural inclusions ([18, Th 4.2]). In the case $A = \mathbb{Z}_{L_1}$, $B = \mathbb{Z}_{L_2}$ and $C = H_{a,b,t}$ one has $\#\iota_1^{-1}(C) = \gcd(L_1/a, L_1s/ab)$ and $\#\iota_2^{-1}(C) = L_2/b$ and the characterization of the cyclic subgroups $H_{a,b,t}$ given in Theorem 4(ii) can be obtained also in this way. It turns out that regarding the sublattice, $H_{a,b,t}$ is cyclic if and only if the numbers of points on the horizontal and vertical axes, respectively, are relatively prime. Note that in the case $L = L_1 = L_2$ the above condition reads $L\gcd(a, b, s) = ab$. Thus it is necessary that $L \mid ab$. The subgroup on the Figure is not cyclic.*

Remark 2. *Every subgroup G of $\mathbb{Z} \times \mathbb{Z}$ has the representation $G = \{(ia, jb + is) : i, j \in \mathbb{Z}\}$, where $0 \le a$, $0 \le s \le b$ are unique integers. This follows like in the proof of Theorem 3. Furthermore, in the case $a, b \ge 1$, $s < b$ the index of G is ab and one obtains at once that the number of subgroups G having index n ($n \in \mathbb{N}$) is $\sum_{ab=n} \sum_{0 \le s \le b-1} 1 = \sum_{ab=n} b = \sigma(n)$.*

3.3 Discrete Gabor systems on lattices and their implementation

This section contains results from joint work with Peter L. Søndergaard and Christoph Wiesmeyr, available as preprint [169].

Over the past 20 years the Gabor transform has become a very valuable and widely used tool in signal processing. The finite, discrete short-time Fourier transform (STFT) for a given signal f of length L is computed by testing f against shifted and modulated copies of a window function g

$$\mathcal{V}_g f[x, \omega] = \sum_{l=0}^{L-1} f[l]\overline{g[l-x]}e^{-2\pi i \omega l/L}.$$

The Gabor transform is a sampled version of the STFT and both provide the possibility to extract temporal frequency information from the signal. The space spanned by the two variables x, ω is called the time-frequency plane; more precise information can be found in Section 3.3.1. A family of translations and modulations of a window function is called *Gabor family* or *Gabor system*.

In the discrete setting efficient algorithms exist almost exclusively for sampling on separable or rectangular lattices [145], the most important ones discovered quickly after finding the Fast Fourier transform algorithm [43]. The two approaches that are most commonly used are the *overlap-add algorithm* (OLA), [88,148] and the *weighted overlap-add algorithm* [129,138]. Both of these algorithms require that the impulse response of the window is supported on a set considerably smaller than the signal length. Even though this is a slight abuse of terminology, in the finite setting it seems sensible to call such windows FIR, i.e. *Finite Impulse Response*. Fast, but less well known algorithms without this requirement have also been found [13, 147].

It is a natural question how to generalize existing algorithms for the Gabor transform and its inverse to the case of nonseparable lattices. In the early years of this century there has been a series of papers and investigations on this subject [14–17, 159] and more by the same authors, also collected in [158]. Earlier studies focus on the computation of dual Gabor windows on nonseparable lattices, using iterative methods [69] or harnessing the block structure of Gabor analysis and frame operators directly and reducing nonseparable sampling sets to a union of product lattices [70, 130]. Another contribution came some years later further investigating the discrete theory of *metaplectic operators* [66]. In this paper we present approaches from these works and propose an improved algorithm, which allows for more efficient computation.

There are two fundamentally different ways of realizing computations that we will investigate and improve upon. The first one uses a decomposition of a nonseparable lattice into the union of cosets of a sparser separable lattice similar to [70,158,161,176]. This will allow to write the Gabor family as a union of Gabor families on this sparse lattice with different windows. We call such a system *multiwindow Gabor* family, since it shares much of the structure from standard Gabor systems [173]. The details can be found in Section 3.3.2, see also Section 3.1 for the continuous case.

The second method under consideration uses the fact that any lattice can be written as the image of a rectangular lattice under an invertible lattice transform. For a special subset of these transforms, so called *metaplectic operators* on the signal space exist that allow to reduce all the computations for Gabor systems on nonseparable lattices to Gabor systems on rectangular lattices. It turns out that in the one dimensional setting the transformation to the separable case is always possible [66,103]. This method has first been described for the continuous case, see Section 3.1 for a short summary or [84] for a more comprehensive study, before it was translated into the finite discrete setting, where it takes more effort to obtain the results due to number theoretic considerations. The algorithms presented and improved in Section 3.3.2 are based on the results in [66].

In higher dimensions the class of lattices that can be reduced to a rectangular sampling strategy is expected to be a strict subset of all lattices. While the class of lattice transforms that admit a metaplectic operator, called *symplectic matrices*, can be determined explicitly it is not easy to see whether a given lattice can be transformed to rectangular shape using this class of matrices. In contrast to the difficulties with generalizing the metaplectic approach to higher dimensions, the multiwindow decomposition can be extended directly. However, the description of the multidimensional case is beyond the scope of this contribution.

After introducing the necessary basic concepts in Section 3.3.1, we mainly present the different approaches in Section 3.3.2. Section 3.3.3 describes the implementation of the different algorithms and compares their computational complexity and running time.

3.3.1 Preliminaries

Discrete Gabor frames on subgroups of the TF-plane. We recall some basics from Gabor analysis, frame theory and the theory of metaplectic operators on $\mathbb{C}^L$. A *Gabor system* in $\mathbb{C}^L$ is a set of functions of the form

$$\mathcal{G}(g,\Lambda) := \{\mathbf{M}_\omega \mathbf{T}_x g \ : \ \lambda = (x,\omega)^T \in \Lambda \subseteq \mathbb{Z}_L^2\}. \tag{3.21}$$

Thus, a Gabor system is a set of time-frequency shifts of a fixed function g. For some given x and ω we use the notation of a time-frequency shift operator

$$\pi(x,\omega) = \pi(\lambda) = \mathbf{M}_\omega \mathbf{T}_x.$$

The Gabor coefficients of some $f \in \mathbb{C}^L$, with respect to $\mathcal{G}(g,\Lambda)$ are given by samples of the short-time Fourier transform

$$\mathcal{V}_g f[x,\omega] = \langle f, \mathbf{M}_\omega \mathbf{T}_x g\rangle = \sum_{l=0}^{L-1} f[l]\overline{g[l-x]}e^{-2\pi i \omega l/L}, \ (x,\omega)^T \in \Lambda. \tag{3.22}$$

It is important to know if the signal f can be reconstructed from its transform coefficients $\{c_{x,\omega} = \mathcal{V}_g f[x,\omega]\}_{(x,\omega)^T\in\Lambda}$, i.e. when the Gabor system $\mathcal{G}(g,\Lambda)$ forms a frame. It turns out that this is equivalent to the invertibility of the so-called frame operator $\mathbf{S} = \mathbf{S}_{g,\Lambda}$ defined as

$$\mathbf{S}_{g,\Lambda} f = \sum_{(x,\omega)^T\in\Lambda} \langle f, \pi(x,\omega)g\rangle \pi(x,\omega)g. \tag{3.23}$$

By inversion of this operator we can give an explicit inversion formula

$$f = \sum_{(x,\omega)^T \in \Lambda} c_{x,\omega} \mathbf{S}^{-1} \pi(x,\omega) g.$$

The family $\{\mathbf{S}^{-1}\pi(x,\omega)g\}_{(x,\omega)^T\in\Lambda}$ is called the (canonical) dual Gabor system. If Λ is a subgroup of the phase space, then we know from standard Gabor theory that the dual system is a Gabor system itself, given by $\mathcal{G}(\mathbf{S}^{-1}g, \Lambda)$, see e.g. [84]. In the following we only consider this structured case and denote that by $\Lambda \leq \mathbb{Z}_L^2$.

It is easy to see that for any matrix $A \in \mathbb{Z}_L^{2\times 2}$ the set $A\mathbb{Z}_L^2$ forms a subgroup of the time-frequency plane. The following corollary shows that the converse is also true. Furthermore, it suggests a normal form that allows us to establish a one to one relation between lattices and generating matrices. Further implications of this statement can be found in the previous Section 3.2 or [154].

Corollary 6. *For every $\Lambda \leq \mathbb{Z}_L^2$ there exist unique $a, b|L$, $0 \leq s < b$ and $s \in \frac{ab}{\gcd(ab,L)}\mathbb{Z}$, such that*

$$\Lambda = A\mathbb{Z}_L^2 = \begin{pmatrix} a & 0 \\ s & b \end{pmatrix} \mathbb{Z}_L^2. \tag{3.24}$$

Proof. Follows directly from Theorem 3. □

With a, b, s as in (3.24), we define

$$\mathcal{G}(g, a, b, s) := \mathcal{G}(g, \Lambda) = \{g_{n,k} := \mathbf{M}_{sn+bk}\mathbf{T}_{an}g \ : \ (n,k) \in \mathbb{Z}_{L/a} \times \mathbb{Z}_{L/b}\}$$

for $\Lambda = A\mathbb{Z}_L^2$, omitting s if it equals zero. Recall the notation $M = L/b$ for the number of frequency channels, as introduced in Section 2.3.1. Analogous to Section 3.1, lattices with $s = 0$ are called *separable, rectangular* or *product lattices*, since they can be written as the direct product of two subgroups of $\mathbb{Z}_L$. If $s \neq 0$, we call a lattice *nonseparable.* It is easy to see that the unique lower triangular form can be rewritten into an upper triangular matrix.

Proposition 4. *Given a subgroup $\mathbb{Z}_L^2$ in normal form, i.e. given a, b and s. Then the following representations are equivalent*

$$\begin{pmatrix} a & 0 \\ s & b \end{pmatrix} \cdot \mathbb{Z}_L^2 = \begin{pmatrix} \tilde{a} & \tilde{s} \\ 0 & \tilde{b} \end{pmatrix} \cdot \mathbb{Z}_L^2, \tag{3.25}$$

where $\tilde{b} = \gcd(b,s)$, $\tilde{a} = ab/\gcd(b,s)$. If furthermore we use Bézout's identity to represent $k_1 s + k_2 b = \gcd(b,s)$, then $\tilde{s} = k_1 a$.

Proof. By computation one can verify that

$$\begin{pmatrix} a & 0 \\ s & b \end{pmatrix} \cdot \begin{pmatrix} b/\gcd(b,s) & k_1 \\ -s/\gcd(b,s) & k_2 \end{pmatrix} = \begin{pmatrix} \tilde{a} & \tilde{s} \\ 0 & \tilde{b} \end{pmatrix}.$$

The second matrix has determinant 1 and therefore is invertible. The assertion follows, because $Q \cdot \mathbb{Z}_L^2 = \mathbb{Z}_L^2$ for any invertible matrix over $\mathbb{Z}_L^2$. □

In some cases we will switch to another description of a subgroup as it comes up more natural in some settings. Instead of the shear parameter s, one can also use the shear relative to b, given by

$$\lambda = \frac{s}{b} = \frac{\lambda_1}{\lambda_2}, \text{ with } \lambda_1 = \frac{s}{\gcd(b,s)},\ \lambda_2 = \frac{b}{\gcd(b,s)}. \tag{3.26}$$

This easily explains how to convert s into λ_1 and λ_2 and vice versa. A visualization can be found in Figure 3.3. Unlike in the case of separable lattices, there is no immediate natural way of indexing the Gabor coefficients. However, it seems sensible to index by the position in time and counting the sampling points in frequency from the lowest nonnegative frequency upwards. Therefore we will fix

$$c\,[m,n] = \sum_{l=0}^{L-1} f[l]\overline{g[l-an+1]}e^{-2\pi i l(m+w[n])/M}, \tag{3.27}$$

for the rest of this chapter, where the additional offset w is given by $w[n] = \text{mod}(n\lambda_1, \lambda_2)/\lambda_2$. This format is also implemented in the open source MATLAB / OCTAVE Toolbox *LTFAT* [2], used for the experiments in Section 3.3.3.

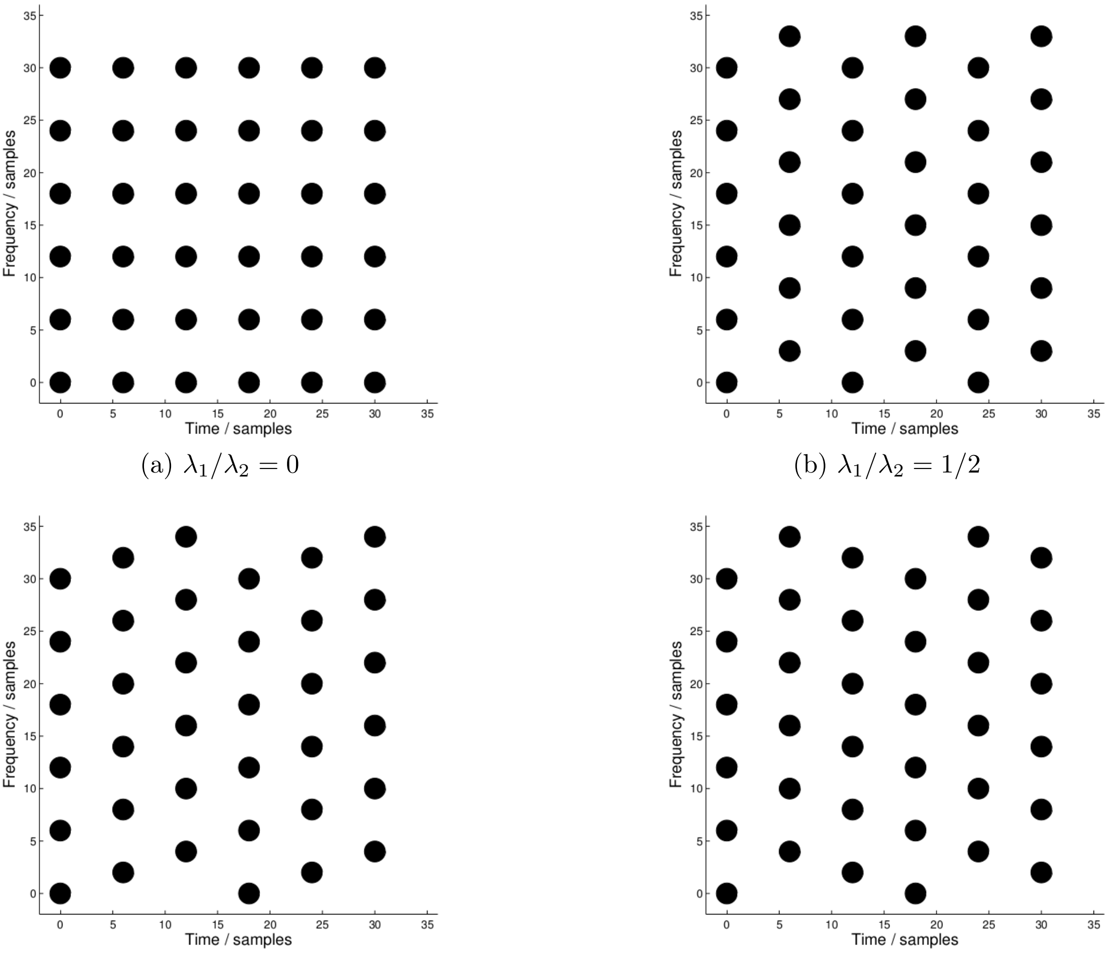
Metaplectics. A metaplectic operator, loosely speaking, is the signal domain counterpart to a symplectic transform of the lattice on phase space. A comprehensive treatment of these operators in the finite discrete setting can be found in [103]. We will be focusing on the one dimensional setting, for which the operators are described in detail in [66]. In this section we will formulate some results that will prove to be important in subsequent sections. We start by the factorization of a lattice generator into elementary matrices, which we will denote by

$$F = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad S_c = \begin{pmatrix} 1 & 0 \\ c & 1 \end{pmatrix}, \quad D_a = \begin{pmatrix} a & 0 \\ 0 & a^{-1} \end{pmatrix}, \tag{3.28}$$

where $c \in \mathbb{Z}_L$ and $a \in \mathbb{Z}_L$ invertible.

Proposition 5 (Feichtinger et al. (2008) [66])**.** *Let* $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathbb{Z}_L^2$ *with* $\det(A) = 1$*, then there exists* $m \in \mathbb{Z}$ *such that* $a_0 = a+mb$ *is invertible in* $\mathbb{Z}_L$*. Let* $c_0 = c+md$*, then*

$$A = S_{c_0 a_0^{-1}} D_{a_0} F^{-1} S_{-a_0^{-1}b} F S_{-m}.$$



(a) $\lambda_1/\lambda_2 = 0$

(b) $\lambda_1/\lambda_2 = 1/2$

(c) $\lambda_1/\lambda_2 = 1/3$

(d) $\lambda_1/\lambda_2 = 2/3$

Figure 3.3: The figure shows the placement of the Gabor atoms for four different lattice types in the time-frequency plane. The displayed Gabor system has parameters $a = 6$, $M = 6$ and $L = 36$. The lattice (a) is called *rectangular* or *separable* and the lattice (b) is known as the *quincunx* lattice.

The proof is based on Weil's decomposition of arbitrary symplectic matrices into a composition of elementary symplectic matrices as in (3.28).

Lemma 3. *For the above defined matrices we define the corresponding metaplectic operators as follows*

$$\begin{aligned} F &\mapsto \mathbf{U}_F = \mathcal{F}, \\ S_c &\mapsto \mathbf{U}_{S_c} = \left(f[\cdot] \mapsto f[\cdot]\exp(\pi i c \cdot^2 (L+1)/L)\right), \\ D_a &\mapsto \mathbf{U}_{D_a} = \left(f[\cdot] \mapsto f[a^{-1}\cdot]\right). \end{aligned}$$

With these transformations the following hold for all $\lambda \in \mathbb{Z}_L^2$

$$\begin{aligned} \mathbf{U}_F\pi(\lambda) &= \phi_F(\lambda)\pi(F\lambda)\mathbf{U}_F \\ \mathbf{U}_{S_c}\pi(\lambda) &= \phi_{S_c}(\lambda)\pi(S_c\lambda)\mathbf{U}_{S_c} \\ \mathbf{U}_{D_a}\pi(\lambda) &= \phi_{D_a}(\lambda)\pi(D_a\lambda)\mathbf{U}_{D_a}, \end{aligned}$$

where ϕ_F, ϕ_{S_c} *and* ϕ_{D_a} *are phase factors.*

Proof. Some simple calculations are sufficient to establish the result:

$$\begin{aligned} \mathbf{U}_F\pi(\lambda)f &= \mathcal{F}\mathbf{M}_\omega\mathbf{T}_x f = \mathbf{T}_\omega\mathbf{M}_{-x}\hat{f} \\ &= e^{-2\pi i x\omega/L}\mathbf{M}_{-x}\mathbf{T}_\omega\hat{f} = e^{-2\pi i x\omega/L}\pi(F\lambda)\mathbf{U}_F f, \end{aligned}$$

$$\begin{aligned} \mathbf{U}_{S_c}\pi(\lambda)f &= \mathbf{M}_\omega\mathbf{T}_x e^{\pi i c(\cdot+x)^2(L+1)/L} f \\ &= e^{\pi i c x^2(L+1)/L}\mathbf{M}_{\omega+cx}\mathbf{T}_x\mathbf{U}_{S_c} f \\ &= e^{\pi i c x^2(L+1)/L}\pi(S_c\lambda)\mathbf{U}_{S_c} f \end{aligned}$$

and

$$\begin{aligned} \mathbf{U}_{D_a}\pi(\lambda)f &= \mathbf{M}_{a^{-1}\omega} f(a\cdot - x) = \mathbf{M}_{a^{-1}\omega} f\left(a^{-1}(\cdot - ax)\right) \\ &= \mathbf{M}_{a^{-1}\omega}\mathbf{T}_{ax}\mathbf{U}_{D_a} f = \pi(D_a\lambda)\mathbf{U}_{D_a} f. \end{aligned}$$

□

The combination of the two results above immediately yields the following theorem.

Theorem 5. *For any matrix* $A \in \mathbb{Z}_L^2$ *with* $\det(A) = 1$*, there exists a metaplectic operator* $\mathbf{U}_A$*, such that for all* $\lambda \in \mathbb{Z}_L^2$

$$\mathbf{U}_A\pi(\lambda) = \phi_A(\lambda)\pi(A\lambda)\mathbf{U}_A. \tag{3.29}$$

3.3.2 Computation on nonseparable lattices

Nonseparable lattices in $\mathbb{Z}_L^2$ can be interpreted in a variety of ways. Several different approaches relate Gabor expansions on general lattices to one or several equivalent expansions on separable (or rectangular) lattices. From an algorithmic viewpoint, these are of particular interest, since a wealth of research [6, 8, 13, 129, 150, 173] has investigated efficient algorithms for analysis and synthesis using Gabor dictionaries on separable lattices. Each of the three approaches described in this section yields a simple relation between arbitrary given Gabor systems and Gabor systems on separable sampling sets that can be harnessed for efficient analysis and synthesis.

Correspondence via multiwindow Gabor. We will decompose a given lattice into a union of λ_2 cosets of a sparser separable lattice, which will allow us to use multiwindow methods [175–178] for the computation. Using multiwindow methods for computation of Gabor transforms on nonseparable lattices has been proposed in [70, 176] and implementation has been discussed in [158, 161]. However, the latter only briefly mention the computation of dual Gabor windows, not discussing efficient implementation in detail.

Proposition 6. *Given the lattice Λ in normal form specified by the parameters a, b and s, then*

$$\Lambda = \cup_{m=0}^{\lambda_2 - 1} \left((ma, \operatorname{mod}(ms, b))^T + \tilde{\Lambda} \right), \tag{3.30}$$

where $\lambda_2 = b/\gcd(b, s)$ and $\tilde{\Lambda}$ is the separable lattice generated by $(\lambda_2 a, 0)^T$ and $(0, b)$.

Proof. Let the matrix generating Λ be denoted by A and let us define $M_x = \{sx + b\omega : \omega \in \mathbb{Z}_L\}$, for $0 \leq x < L/a$. We note here, that M_x is the second coordinate of the set $A \cdot (x, \mathbb{Z}_L)^T$. Furthermore, $0 \in M_x$ if and only if x is a multiple of λ_2. To see that, we first note that λ_1 and λ_2 are relatively prime. Then the following equation has a solution if and only if x is a multiple of λ_2

$$sx + b\omega = b \left(\frac{\lambda_1}{\lambda_2} x + \omega \right) = 0.$$

This yields

$$\begin{aligned} M_x &= M_{x+\lambda_2}, \quad \text{for } x \in \mathbb{Z}_{L/a} \\ M_x &= \operatorname{mod}(sx, b) + M_0 \end{aligned}$$

This observation yields the following decomposition of the original lattice

$$\begin{aligned}\Lambda &= \bigcup_{x=0}^{L/a-1} \{ax\} \times M_x \\ &= \bigcup_{m=0}^{\lambda_2-1} \left((am, \operatorname{mod}(sm, b)) + \bigcup_{j=0}^{L/(a\lambda_2)-1} \{aj\lambda_2\} \times M_0 \right),\end{aligned}$$

which finishes the proof by observing

$$\tilde{\Lambda} = \bigcup_{j=0}^{L/(a\lambda_2)-1} \{aj\lambda_2\} \times M_0.$$

□

We can now describe a Gabor system $\mathcal{G}(g, \Lambda)$, with Λ in the form (3.24), and the related operators completely in terms of a union of Gabor systems $\mathcal{G}(g_m, \tilde{\Lambda})$ on the separable lattice $\tilde{\Lambda}$.

Proposition 7. *Let $\mathcal{G}(g, \Lambda)$, $\mathcal{G}(g_m, \tilde{\Lambda})$, with $\Lambda, \tilde{\Lambda}$ as in Proposition 6 and $g \in \mathbb{C}^L$, $g_m = \mathbf{M}_{\operatorname{mod}(ms,b)} \mathbf{T}_{ma} g$, for $0 \leq m < \lambda_2$, be Gabor systems, then*

$$\mathbf{S}_{g,\Lambda} f = \sum_{m=0}^{\lambda_2-1} \mathbf{S}_{g_m, \tilde{\Lambda}} f. \tag{3.31}$$

Moreover, the Gabor transform can be computed using the identity

$$\langle f, \mathbf{M}_{kb+\operatorname{mod}(ms,b)} \mathbf{T}_{na} g \rangle = e^{-\frac{2\pi i \tilde{n}\tilde{a} \operatorname{mod}(ms,b)}{L}} \langle f, \mathbf{M}_{kb} \mathbf{T}_{\tilde{n}\tilde{a}} g_m \rangle, \tag{3.32}$$

where $\tilde{n} = \lfloor n/\lambda_2 \rfloor$ and $m = n - \tilde{n}$.

Proof. Analogous to Lemma 3, we find that

$$\begin{aligned}\mathbf{M}_{kb+\operatorname{mod}(ms,b)} \mathbf{T}_{na} g &= \mathbf{M}_{kb} \mathbf{M}_{\operatorname{mod}(ms,b)} \mathbf{T}_{\tilde{n}\tilde{a}} \mathbf{T}_{ma} g \\ &= e^{\frac{2\pi i \tilde{n}\tilde{a} \operatorname{mod}(ms,b)}{L}} \mathbf{M}_{kb} \mathbf{T}_{\tilde{n}\tilde{a}} \mathbf{M}_{\operatorname{mod}(ms,b)} \mathbf{T}_{ma} g \\ &= e^{\frac{2\pi i \tilde{n}\tilde{a} \operatorname{mod}(ms,b)}{L}} \mathbf{M}_{kb} \mathbf{T}_{\tilde{n}\tilde{a}} g_m,\end{aligned}$$

yielding (3.32). Using $kb + \operatorname{mod}(ms, b) = kb + \operatorname{mod}(ns, b) = (k - \lfloor ns/b \rfloor)b + ns$, since $\operatorname{mod}(\tilde{n}s, b) = 0$ allows to derive (3.31) by the identity

$$\begin{aligned}&\sum_{n=0}^{L/a-1} \sum_{k=0}^{L/b-1} \langle f, \mathbf{M}_{ns+kb} \mathbf{T}_{na} g \rangle \mathbf{M}_{ns+kb} \mathbf{T}_{na} g \\ &= \sum_{m=0}^{\lambda_2-1} \sum_{\tilde{n}=0}^{L/\tilde{a}-1} \sum_{k=0}^{L/b-1} \langle f, \mathbf{M}_{kb} \mathbf{T}_{\tilde{n}\tilde{a}} g_m \rangle \mathbf{M}_{kb} \mathbf{T}_{\tilde{n}\tilde{a}} g_m.\end{aligned}$$

□

Correspondence via Smith normal form. In this and the following section, we aim to describe an arbitrary lattice as separable lattice under a symplectic deformation, i.e. we will determine a symplectic matrix P, such that $\Lambda = P\tilde{\Lambda}$ for a general lattice Λ and a separable lattice $\tilde{\Lambda}$. This problem is equivalent to decomposing the lattice generator matrix $A \in \mathbb{Z}_L^{2\times 2}$ into $A = PDV$, with a diagonal matrix D, a determinant 1 matrix V and a symplectic matrix P. We observed earlier that any determinant 1 matrix in $\mathbb{Z}_L^{2\times 2}$ is symplectic. Thus, this decomposition is accomplished by applying Smith's algorithm for matrices in $\mathbb{Z}^{2\times 2}$ to determine the Smith normal form $\tilde{D}$ of A and transformation matrices $\tilde{P}, \tilde{V}$, followed by considering the entries of $\tilde{D}, \tilde{P}, \tilde{V}$ modulo L to find D, P, V.

The following proposition by Feichtinger et al. was originally published in [66], where the proof is also presented. The procedure of computing Gabor transforms and dual windows using the methods in this section have been proposed therein, but their implementation was not discussed in detail.

Proposition 8 (Feichtinger et al. (2008))**.** *Let $\Lambda = A\mathbb{Z}_L^2$ be a lattice and $A = \tilde{P}\tilde{D}\tilde{V}$ the Smith decomposition of A. Then*

$$\Lambda = P\tilde{\Lambda}, \tag{3.33}$$

where $P = \operatorname{mod}(\tilde{P}, L)$, $D = \operatorname{mod}(\tilde{D}, L)$ and $\tilde{\Lambda} = D\mathbb{Z}_L^2$.

Using Proposition 5 and Lemma 3 one obtains the operator $\mathbf{U}_P$ corresponding to the symplectic matrix P and this leads to the final computational procedure described in the following corollary.

Corollary 7. *Let the notation be as in the previous proposition. Then one finds for the symplectic matrix P and the corresponding metaplectic operator $\mathbf{U}_P$ by setting $\tilde{g} = \mathbf{U}_P^{-1} g$*

$$\mathbf{S}_{g,\Lambda} = \mathbf{U}_P \mathbf{S}_{\tilde{g},\tilde{\Lambda}} \mathbf{U}_P^{-1}. \tag{3.34}$$

Furthermore, the Gabor coefficients can be computed using the identity

$$\langle f, \pi(z) g\rangle = \phi_P(z)\langle \mathbf{U}_P^{-1} f, \pi\left(P^{-1}z\right)\tilde{g}\rangle, \text{ for all } z = (x,\omega)^T \in \Lambda. \tag{3.35}$$

Correspondence via shearing. As detailed in the previous section, the Weil decomposition and Smith normal form can be used to show that any lattice in $\mathbb{Z}_L^2$ can be written as a separable lattice, deformed by 6 elementary symplectic matrices. This number can be reduced to 4 or less as shown in the following theorem, which we will prove at the end of this section. Reducing computations on nonseparable lattices to the product lattice case via a shear operation has been proposed earlier [158, 161], however the authors were able to describe only a subset of all lattices over $\mathbb{Z}_L^2$ as shears of rectangular lattices. In [158] the author speculates that it might be possible to describe every lattice as the image of a product lattice under a horizontal and a vertical shear. In this section, we prove that this is indeed possible.

The proper definition of discrete, finite chirps, necessary to perform time-frequency shearing, has been a matter of some discussion, see e.g. [30]. While the naive linear chirp $\exp(2\pi i s t^2/L)$ is still used by Bastiaans and van Leest [158, 161], a more appropriate definition, see Lemma 3, has been proposed by Kaiblinger [66, 102], constituting a second degree character [166].

Theorem 6. *let $A \in \mathbb{Z}_L^{2\times 2}$. There exist $s_0, s_1 \in \mathbb{Z}_L$ and $V \in \mathbb{Z}_L^{2\times 2}$ with $|\det(V)| = 1$, such that*

$$A = U_{s_0,s_1} D V, \tag{3.36}$$

where $D \in \mathbb{Z}_L^{2\times 2}$ is diagonal and

$$U_{s_0,s_1} = S_{-s_1} F^{-1} S_{s_0} F. \tag{3.37}$$

We can now rewrite Gabor transforms on nonseparable lattices in the vein of Proposition 7 using the metaplectic operator associated to U_{s_0,s_1}. Subsequently, we denote by $\mathbf{U}_{s_0,s_1}$ the metaplectic operator associated with U_{s_0,s_1}.

Proposition 9. *Let $\Lambda = A\mathbb{Z}_L^2$ be a lattice, D, U_{s_0,s_1} as in the previous theorem and $\tilde{\Lambda} = D\mathbb{Z}_L^2$. Furthermore, let $g \in \mathbb{C}^L$ and $\tilde{g} = \mathbf{U}_{s_0,s_1}^{-1} g$. Then*

$$\mathbf{S}_{g,\Lambda} f = \mathbf{U}_{s_0,s_1} \mathbf{S}_{\tilde{g},\tilde{\Lambda}} \mathbf{U}_{s_0,s_1}^{-1} f \tag{3.38}$$

and

$$\langle f, \mathbf{M}_\omega \mathbf{T}_x g\rangle = \phi_{U_{s_0,s_1}}(z) \langle \mathbf{U}_{s_0,s_1}^{-1} f, \mathbf{M}_{\omega - s1(x-s_0\omega)} \mathbf{T}_{x-s_0\omega} \tilde{g}\rangle, \tag{3.39}$$

for all $z = (x,\omega)^T$. Moreover,

$$\phi_{U_{s_0,s_1}}(z) = \exp\left(\pi i (s_0\omega^2 - s_1(x-s_0\omega)^2)(L+1)/L\right). \tag{3.40}$$

Proof. Everything but the explicit form of the phase factor $\phi_{U_{s_0,s_1}}$ is a direct consequence of Lemma 3 and Theorem 6, note

$$U_{s_0,s_1} = S_{-s_1} F^{-1} S_{s_0} F = \begin{pmatrix} 1 & -s_0 \\ -s_1 & s_0 s_1 + 1 \end{pmatrix}.$$

To complete the proof, set $y = (x - s_0\omega)$ and determine the phase factor explicitly:

$$\begin{aligned}
&\mathbf{U}_{S_{-s_1}} \mathcal{F}^{-1} \mathbf{U}_{S_{s_0}} \mathcal{F} \mathbf{M}_\omega \mathbf{T}_x f \\
&= \exp\left(\pi i s_0 \omega^2 (L+1)/L\right) \mathbf{U}_{S_{-s_1}} \mathcal{F}^{-1} \mathbf{T}_\omega \mathbf{M}_{s_0\omega - x} \mathbf{U}_{S_{s_0}} \mathcal{F} f \\
&= \exp\left(\pi i s_0 \omega^2 (L+1)/L\right) \mathbf{U}_{S_{-s_1}} \mathbf{M}_\omega \mathbf{T}_y \mathcal{F}^{-1} \mathbf{U}_{S_{s_0}} \mathcal{F} f \\
&= \exp\left(\pi i (s_0\omega^2 - s_1 y^2)(L+1)/L\right) \mathbf{M}_{\omega - s_1 y} \mathbf{T}_y \mathbf{U}_{S_{-s_1}} \mathcal{F}^{-1} \mathbf{U}_{S_{s_0}} \mathcal{F} f,
\end{aligned}$$

where we used Lemma 3 and $\exp(2\pi i m(L+1)/L) = \exp(2\pi i m/L)$ for all $m \in \mathbb{Z}$. □

For Proposition 9 to be valid, it remains to prove Theorem 6, establishing the representation of A through U_{s_0,s_1}.

Proof of Theorem 6. By Proposition 6 we can assume without loss of generality that A is in lattice normal form, i.e.

$$A = \begin{pmatrix} a & 0 \\ s & b \end{pmatrix}.$$

To prove Equation (3.36), we rewrite $U_{s_0,s_1}^{-1} A = DV$ with a diagonal matrix D and a unitary matrix V. It can be seen that

$$U_{s_0,s_1}^{-1} = \begin{pmatrix} s_0 s_1 + 1 & s_0 \\ s_1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & s_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ s_1 & 1 \end{pmatrix}.$$

Now, using Proposition 4 in the step from (3.41) to (3.42) below, we can write

$$\begin{aligned} U_{s_0,s_1}^{-1} A &= \begin{pmatrix} 1 & s_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ s_1 & 1 \end{pmatrix} \begin{pmatrix} a & 0 \\ s & b \end{pmatrix} \\ &= \begin{pmatrix} 1 & s_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & 0 \\ s_1 a + s & b \end{pmatrix} \qquad (3.41) \\ &= \begin{pmatrix} 1 & s_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{ab}{X} & a k_1 \\ 0 & X \end{pmatrix} \begin{pmatrix} k_2 & -k_1 \\ (s_1 a + s)/X & b/X \end{pmatrix} \qquad (3.42) \\ &= \begin{pmatrix} \frac{ab}{X} & s_0 X + a k_1 \\ 0 & X \end{pmatrix} \begin{pmatrix} k_2 & -k_1 \\ (s_1 a + s)/X & b/X \end{pmatrix}. \qquad (3.43) \end{aligned}$$

Here $X = \gcd(s_1 a + s, b)$ and k_1, k_2 stem from Bézout's identity when representing $\gcd(s_1 a + s, b) = k_1(s_1 a + s) + k_2 b$. It is important to note that the second matrix in the last line has determinant one. This shows that the lattice $U_{s_0,s_1} A$ is separable if and only if $\tilde{D} = \left(\begin{smallmatrix} ab/X & s_0 X + a k_1 \\ 0 & X \end{smallmatrix}\right)$ is equivalent to a diagonal matrix, i.e.

$$\operatorname{mod}(s_0 X + a k_1, ab/X) = 0. \qquad (3.44)$$

We will now deduce numbers s_0 and s_1 satisfying our needs from the prime factor decomposition of the involved quantities. Therefore we represent $L = \prod_{j=1}^{J} p_j^{n_j}$ for a fixed set of prime numbers. Since a and b are divisors of L we find their prime factor decompositions to have exponents $\{\alpha_j\}_{j=1}^{J}$ and $\{\beta_j\}_{j=1}^{J}$, where $\alpha_j, \beta_j \leq n_j$. The shearing parameter has the decomposition $s = l \prod_{j=1}^{J} p_j^{\sigma_j}$, where $\gcd(l, L) = 1$.

We choose

$$s_1 = \prod_{j=1}^{J} p_j^{\mu_j}, \text{ where } \mu_j = \begin{cases} 1 \text{ for } \alpha_j = \sigma_j, \\ 0 \text{ else.} \end{cases} \tag{3.45}$$

With this choice of s_1 we investigate

$$X = \gcd(s_1 a + s, b) = \prod_{j=1}^{J} \gcd(s_1 a + s, p_j^{\beta_j}).$$

To do so, we have to individually treat three cases:

1. $\alpha_j < \sigma_j$: Since s_1 and p_j are coprime we find $\gcd(s_1 a + s, p_j^{\beta_j}) = p_j^{\min(\alpha_j, \beta_j)}$.
2. $\alpha_j > \sigma_j$: $\gcd(s_1 a + s, p_j^{\beta_j}) = p_j^{\min(\sigma_j, \beta_j)}$, and $\min(\sigma_j, \beta_j) < \alpha_j$
3. $\alpha_j = \sigma_j$: Use Eq. (3.45) to determine that $\gcd(s_1 a + s, p_j^{\beta_j}) = p_j^{\min(\alpha_j, \beta_j)}$

The above arguments show that with the choice of s_1, we find that $X = \prod_{j=1}^{J} p_j^{\gamma_j}$, where $\gamma_j \leq \alpha_j$.

Now we turn to the choice of s_0. We first decompose $k_1 = l \prod_{j=1}^{J} p_j^{\kappa_j}$, where l and L are coprime. Let us explain how to choose the shear via the positive part of a vector

$$s_0 = \left(\prod_{j=1}^{J} p_j^{(\beta_j - \gamma_j - \kappa_j)_+} - l \right) \prod_{j=1}^{J} p_j^{\alpha_j + \kappa_j - \gamma_j}, \text{where } (x_+)_j = \max(x_j, 0).$$

A straightforward calculation shows then that

$$s_0 X + a k_1 = \prod_{j=1}^{J} p_j^{(\beta_j - \gamma_j - \kappa_j)_+ + \alpha_j + \kappa_j},$$

and we furthermore see that

$$(\beta_j - \gamma_j - \kappa_j)_+ + \alpha_j + \kappa_j \geq \beta_j + \alpha_j - \gamma_j.$$

This proofs that (3.44) is satisfied, completing the proof. □

Remark 3. *It is easy to see that X in the proof above satisfies* $\gcd(a, b) = kX$ *for some $k \in \mathbb{N}_0$ and therefore ab/X is a multiple of X. Thus, the diagonal matrix constructed above is in fact the Smith normal form of A.*

Further optimization. In this section we will first determine which signal lengths are feasible for some given choice of a, M and λ_1, λ_2. This restriction holds for all the presented methods equally and is essential to know in computations.

Particularly when using the shear method described in the previous paragraphs it is interesting to know for which signal lengths one of the two shears s_0 and s_1, preferably the frequency side shear s_0, can be chosen to be zero. This saves additional computation time.

Proposition 10. *Given the parameters $\lambda = \lambda_1/\lambda_2$, a and M. Then the minimal signal length, for which these parameters are feasible is given by $L_{min} = \lambda_2 \operatorname{lcm}(a, M)$. All the feasible signal lengths are multiples of this.*

Proof. For the parameters in combination with a given signal length L to form a lattice we require the following conditions

$$\begin{aligned} a|L,\ M|L, \\ \frac{L}{a}\lambda \in \mathbb{Z}, \\ \frac{L}{M}\lambda \in \mathbb{Z}, \end{aligned}$$

where the first conditions immediately yield $\operatorname{lcm}(a, M)|L$. From the other two conditions we can derive

$$a\lambda_2/\gcd(a, \lambda_1)|L \quad \text{and} \quad M\lambda_2/\gcd(M, \lambda_1)|L.$$

Therefore, the signal length has to be a multiple of

$$L_{\min} = \operatorname{lcm}\left(\frac{a\lambda_2}{\gcd(a, \lambda_1)}, \frac{M\lambda_2}{\gcd(M, \lambda_1)}, a, M\right).$$

We proceed to show that

$$\operatorname{lcm}\left(\frac{a\lambda_2}{\gcd(a, \lambda_1)}, a\right) = \lambda_2 a. \tag{3.46}$$

For this purpose we look at the prime factor decomposition of the involved quantities, where we denote by $\alpha_j, \gamma_j, \delta_j$ the exponents of the prime number p_j of a, λ_1 and λ_2 respectively. Then we find, since λ_1 and λ_2 are coprime, that the exponent of p_j of $\operatorname{lcm}(a\lambda_2/\gcd(a, \lambda_1), a)$ is given by

$$\max(\alpha_j - \min(\alpha_j, \gamma_j) + \delta_j, \alpha_j) = \alpha_j + \delta_j,$$

proving (3.46). The proof that $\operatorname{lcm}(M\lambda_2/\gcd(M,\lambda_1), M) = \lambda_2 M$ is completely analogous. Combine these to find

$$\begin{aligned}\operatorname{lcm}\left(\frac{a\lambda_2}{\gcd(a,\lambda_1)}, \frac{M\lambda_2}{\gcd(M,\lambda_1)}, a, M\right) &= \operatorname{lcm}(\lambda_2 a, \lambda_2 M) \\ &= \lambda_2 \operatorname{lcm}(a, M).\end{aligned}$$

□

Now we shall investigate, which multiples of the just derived minimal signal length allow for computation without the frequency shear. To do so, it is instructive to compute the set of factors l, for which $L = lL_{\min}$ needs only the time shear. We will introduce here some important constants related to the *time shift* a, the *frequency shift* b, the *number of channels* $M = L/b$ and the *number of time shifts* $N = L/a$. We define $c, d, p, q \in \mathbb{N}$ by

$$c = \gcd(a, M) \quad , \quad d = \gcd(b, N), \tag{3.47}$$

$$p = \frac{a}{c} = \frac{b}{d} \quad , \quad q = \frac{M}{c} = \frac{N}{d}. \tag{3.48}$$

With these numbers, the *redundancy* of a Gabor system can be written as $L/(ab) = q/p$, where q/p is an irreducible fraction. It holds that $L = cdpq$. Some of the introduced notation will be important in the next section.

Proposition 11. *Given λ, a and M. Let the prime factor decomposition of $c = \gcd(a, M)$ be given by*

$$c = \prod_{j=1}^{J} p_j^{\gamma_j}$$

for some set of prime factors and corresponding exponents. Let

$$c_1 = \prod_{j=1}^{J} p_j^{\sigma_j}, \quad \sigma_j = \begin{cases} \gamma_j & \text{if } \gcd(\lambda_2, p_j) = 1, \\ 0 & \text{else,} \end{cases}$$

then the frequency shear can be chosen to be 0 *if the signal length satisfies*

$$L = nL_{min}\frac{c}{c_1}, \tag{3.49}$$

for some $n \in \mathbb{N}$. In words, c_1 are factors of c that are relatively prime to λ_2

Proof. With the standard notation we easily see that the time shear is sufficient if and only if $(s + kb)/a \in \mathbb{Z}$ for some $k \in \{0, \ldots, M-1\}$. Rewriting this leads to

$$L = \tilde{l}\frac{Ma\lambda_2}{\lambda_1 + k\lambda_2} = l\frac{Ma\lambda_2}{\gcd(\lambda_1 + k\lambda_2, Ma\lambda_2)},$$

for some $l \in \mathbb{Z}$. However, these signal lengths might not be compatible with the feasibility condition from Proposition 10. Therefore we compute the ratio

$$\frac{L}{L_{\min}} = l\frac{\gcd(M,a)}{\gcd(\lambda_1 + k\lambda_2, Ma\lambda_2)}.$$

Since this fraction should be an integer number, we have to choose

$$l = n\frac{\gcd(\lambda_1 + k\lambda_2, Ma\lambda_2)}{\gcd(M, a, \lambda_1 + k\lambda_2, Ma\lambda_2)},$$

for some $n \in \mathbb{N}$. Therefore, we can compute

$$L = nL_{\min}\frac{\gcd(M,a)}{\gcd(M,a,\lambda_1 + k\lambda_2)}. \tag{3.50}$$

With the notation introduced above we are now interested in computing

$$\max_{k\in\mathbb{N}}(\gcd(c, \lambda_1 + k\lambda_2)). \tag{3.51}$$

Firstly, we rewrite

$$\gcd(c, \lambda_1 + k\lambda_2) = \prod_{j=1}^{J}\gcd(p_j^{\gamma_j}, \lambda_1 + k\lambda_2).$$

Now we will individually investigate the factors in the product above.

Case 1. $\gcd(p_j, \lambda_2) = 1$, in which case we can find numbers $k_{1,j}, k_{2,j}$, such that

$$\lambda_1 + k_{1,j}\lambda_2 = k_{2,j}p_j^{\gamma_j}.$$

Furthermore, the full set of coefficients of λ_2, for which the above equation can be satisfied is given by $K_j = \left\{k_{1,j} + mp_j^{\gamma_j} : m \in \mathbb{Z}\right\}$. Therefore, for any $k \in K_j$ we find

$$\gcd(p_j^{\gamma_j}, \lambda_1 + k\lambda_2) = p_j^{\gamma_j}.$$

Case 2. $\gcd(p_j, \lambda_2) \neq 1$, which implies directly that λ_2 is a multiple of p_j. In this case we have to argue that $\lambda_1 + k\lambda_2$ can never be a multiple of p_j. Indeed, any linear combination $k_{1,j}\lambda_2 + k_{2,j}p_j$ is a multiple of p_j and therefore not equal to λ_1, which is assumed to be relatively prime to λ_2. Consequently, for any choice of $k \in \mathbb{Z}$

$$\gcd(p_j^{\gamma_j}, \lambda_1 + k\lambda_2) = 1.$$

For all the indices j in case 1, it is easy to see that the intersection of the corresponding sets K_j is not empty. This is an immediate consequence from the fact that powers of two different prime numbers have no common divisors. Using the notation introduced above, we can conclude that there exists some $k \in \mathbb{Z}$, such that

$$\gcd(c, \lambda_1 + \lambda_2) = c_1.$$

The last argument needed is to show that $k \in \{0, \ldots, M-1\}$. By construction $s + kb = \tilde{k}a$, for some $\tilde{k} \in \mathbb{Z}$. Therefore, for any $m \in \mathbb{Z}$

$$s + \left(k + m\frac{L}{b}\right) b = \left(\tilde{k} + m\frac{L}{a}\right) a,$$

and for an appropriate choice of m, the expression in brackets on the left hand side will equal some number in the desired range.

□

Remark 4. *There are possibly other feasible signal lengths than those determined by* (3.49)*. The full set of feasible lengths is determined by*

$$\left\{ nL_{min} \frac{\gcd(M,a)}{\gcd(M,a,\lambda_1 + k\lambda_2)} : \ n \in \mathbb{N}, k \in \{0,\ldots,M-1\} \right\}.$$

This can be easily seen from (3.50) *in the proof above. For simplicity we only construct the minimal factor, that L_{min} has to be multiplied with, as stated in* (3.51)*.*

Remark 5. *Looking at* (3.49) *we see that, if we are given a certain redundancy* (3.48) *q/p and a lattice type λ_1/λ_2 and require a low value of $L_{min}c/d$, we must choose c such that it is relatively prime to λ_2. As an example, consider a common choice of $a = 32$, $M = 64$ and $\lambda_1/\lambda_2 = 1/2$ (the quincunx lattice). In this case $c = \gcd(a, M) = 32$ which is the worst possible case, as it is a power of $\lambda_2 = 2$ giving a value of $L_{min}c/d = 128 \cdot 32 = 4096$. If we instead choose $a = 27$, $M = 54$ (which is the same redundancy) we get $L_{min}c/d = 108 \cdot 1 = 108$. This illustrates that it is possible to work efficiently with the quincunx lattice by not choosing the rectangular lattice parameters to be powers of 2.*

Extension to higher dimensions

It is well known [34, 67, 125] that multidimensional Gabor transforms and dual windows can be computed using algorithms designed for the 1D case, if both the Gabor window and the lattice used can be written as a tensor product. That is, we assume that with $l = (l_1, \ldots, l_d)^T \in \mathbb{C}^{L_1} \times \ldots \times \mathbb{C}^{L_d}$,

$$g[l] = g_1[l_1] \otimes \ldots \otimes g_D[l_d]$$

and

$$\Lambda = \Lambda_1 \times \ldots \times \Lambda_d = A_1 \mathbb{Z}_{L_1}^2 \times \ldots \times A_d \mathbb{Z}_{L_d}^2$$

for some $A_j \in \mathbb{Z}_{L_j}^2 \times \mathbb{Z}_{L_j}^2$ for $j = 1, \ldots, d$, where d denotes the dimension.

Equivalently, we can say that Λ can be described by a block matrix

$$A = \begin{pmatrix} D & E \\ F & G \end{pmatrix}, \tag{3.52}$$

with diagonal blocks $D, E, F, G \in \mathbb{Z}^{n \times n}$. In this case, the multidimensional transform and dual window can be computed by subsequently applying the algorithms presented in the previous sections in every dimension. A matrix describing the lower dimensional lattice corresponding to dimension j is simply given by

$$A_j = \begin{pmatrix} D_{j,j} & E_{j,j} \\ F_{j,j} & G_{j,j} \end{pmatrix}$$

and can be transformed into lattice normal form (3.24), allowing straightforward application of the presented algorithms.

However, we are not aware of a constructive method to determine whether a lattice, given by an arbitrary matrix, can be described by a banded matrix of the form (3.52).

3.3.3 Implementation and timing

In this section we discuss the implementation and speed of the proposed algorithms.

Methodology for computing the computational complexity. To compute the discrete Fourier transform, the familiar FFT algorithm is used. When computing the flop (floating point operation) count of the algorithm, we will assume that a complex FFT of length M can be computed using $4M \log_2 M$ flops. A review of flop counts for FFT algorithms is presented in [101]. When computing the flop count, we assume that both the window and signal are complex valued.

At this point it is important to recall the notation of c, d, p, q introduced in (3.47) and (3.48). The cost of performing the computation of a DGT with a full length window on a rectangular lattice using the algorithm first reported in [147] is given by

$$8Lq + 4L \log_2 d + 4MN \log_2 d + 4MN \log_2 (M) \tag{3.53}$$

$$= L\left(8q + 4 \log_2 d\right) + 4MN \left(\log_2 L/p\right) \tag{3.54}$$

where the first terms in (3.53) come from the multiplication of the matrices in the factorization, the two middle terms come from creating the factorization of the signal and inverting the factorization of the coefficients, and the last term comes from the final application of FFTs. The terms can be collected as in (3.54), where the first term grows as the length of the signal L, and the second terms grows as the total number of coefficients MN. In the following, we refer to this as the *full window* algorithm.

If the window is an FIR window supported on an index set with width L_g which is much smaller than the length of the signal L, the *weighted-overlap-add algorithm*, first reported in [129], can be used instead. It has a computational complexity of

$$8L\frac{L_g}{a} + 4NM\log_2 M. \tag{3.55}$$

In the following, we refer to this as the *FIR window* algorithm.

A third approach to computing a DGT is a hybrid approach, where a DGT using an FIR window can be computing using a full window algorithm on blocks of the input signal. The blocks are then combined using the classical *overlap-add* (OLA) algorithm, cf. [88, 148].

The OLA algorithm works by partitioning a system of length L into blocks of length L_b such that $L = L_b N_b$, where N_b is the number of blocks. The block length must be longer than the support of the window, $L_b > L_g$. To perform the computation we take a block of the input signal of length L_b and zero-extend it to length $L_x = L_b + L_g$, and compute the convolution with the extended signal using the similarly extended window. Because of the zero-extension of the window and signal, the computed coefficients will not be affected by the periodic boundary conditions, and it is therefore possible to overlay and add the computed convolutions of length L_x together to form the complete convolution of length L.

Equations (3.54) and (3.55) are used to express the efficiency of the algorithms for the DGT on nonseparable lattices.

Implementation of the shear algorithm

The shear algorithm proposed in Proposition 9 computes the DGT on a nonseparable lattice using a DGT on a separable lattice with some suitable pre- and postprocessing steps. The computational complexity of the pre- and postprocessing steps is significant compared to the separable DGT, so we wish to minimize the cost of these steps. An implementation of the shear algorithm is presented as Algorithm 1. Note that we assume the existence of several underlying routines: an implementation DGT of the separable Gabor transform, the periodic chirp PCHIRP(L,s)= $\exp(\pi i s \cdot^2 (L+1)/L)$ and SHEARFIND, a program that determines the shear parameters s_0, s_1 and the correct separable lattice to do the DGT on, following the constructive proof of Theorem 6.

Algorithm 1 The shear algorithm: $c = \textsc{dgtns}(f, g, a, M, \lambda)$

```
$[s_0, s_1, b_r] = \textsc{shearfind}(L, a, M, \lambda)$
if $s_1 \neq 0$ then
    $p \leftarrow \textsc{pchirp}(L, s_1)$
    $g(\cdot) \leftarrow p(\cdot)g(\cdot)$
    $f(\cdot) \leftarrow p(\cdot)f(\cdot)$
end if
if $s_0 = 0$ then
    $c_r \leftarrow \textsc{dgt}(f, g, a, M)$
    $C_1 \leftarrow s_1 a(L+1) \pmod{2N}$
    for $k = 0 \rightarrow N-1$ do
        $E \leftarrow e^{\pi i (C_1 k^2 \pmod{2N})/N}$
        for $m = 0 \rightarrow M-1$ do
            $c(\left\lfloor \frac{-s_1 ka + mb \pmod{L}}{b} \right\rfloor, k) \leftarrow E c_r(m, k)$
        end for
    end for
else
    $a_r \leftarrow \frac{ab}{b_r}, \quad M_r \leftarrow \frac{L}{b_r}, \quad N_r \leftarrow \frac{L}{a_r}$
    $C_1 \leftarrow \frac{a_r}{a}, \quad C_2 \leftarrow -s_0 b_r / a$
    $C_3 \leftarrow a s_1 (L+1), \quad C_4 \leftarrow C_2 b_r (L+1)$
    $C_5 \leftarrow 2 C_1 b_r, \quad C_6 \leftarrow (s_0 s_1 + 1) b_r$
    $p \leftarrow \textsc{pchirp}(L, -s_0)$
    $g(\cdot) \leftarrow p(\cdot)\textsc{fft}(g(\cdot))/L$
    $f(\cdot) \leftarrow p(\cdot)\textsc{fft}(f(\cdot))$
    $c_r \leftarrow \textsc{dgt}(f, g, b_r, N_r)$
    for $k = 0 \rightarrow N_r - 1$ do
        for $m = 0 \rightarrow M_r - 1$ do
            $s_{q1} \leftarrow C_1 k + C_2 m \pmod{2N}$
            $E \leftarrow e^{\pi i (C_3 s_{q1}^2 - m(C_4 m + C_5 k) \pmod{2N})/N}$
            $\tilde{m} \leftarrow C_1 k + C_2 m \pmod{N}$
            $\tilde{k} \leftarrow \left\lfloor \frac{-s_1 a_r k + C_6 m \pmod{L}}{b} \right\rfloor$
            $c(\tilde{k}, \tilde{m}) \leftarrow E c_r(-k \pmod{N_r}, m)$
        end for
    end for
end if
```

A simple trick is to notice that when a frequency-side shear is needed, the DFT of the signal f and the window g are multiplied by a chirp on the frequency side, $\tilde{f} = \mathbf{U}_{s_0,s_1}^{-1} f$ and $\tilde{g} = \mathbf{U}_{s_0,s_1}^{-1} g$. The total cost of this is 4 FFTs and two pointwise multiplications. However, instead of transforming the chirped signal and window back to the time domain, we can compute the nonseparable DGT directly in the frequency domain using the well-known commutation relation of the DFT and the translation and modulation operators:

$$\langle f, M_m T_n g \rangle \quad = \quad e^{-\pi i m n / L} \langle \mathcal{F} f, M_{-n} T_m \mathcal{F} g \rangle \, . \tag{3.56}$$

This trick saves the two inverse FFTs at the expense of the multiplication of the coefficients by a complex exponential and reshuffling. As we already need these operations to realize (3.39) and (3.40), they can be combined with no additional computational complexity.

The overlap-add algorithm can be used in conjunction with the shear algorithm in the following case: we wish to compute the DGT with an FIR window for a nonseparable lattice using the shear algorithm. Because of the frequency-side shearing, the window is converted from an FIR window into a full length window, making it impossible to perform real-time or block-wise processing. However, if the shear algorithm is used inside an OLA algorithm, this is no longer a concern, as the shearing will only convert the window into a window of length $L_g + L_b$, restoring the ability to perform block-wise processing.

In total, the shear-OLA algorithm for the DGT is calculated in three steps using the three algorithms:

1. Split the input signal into blocks using the overlap-add algorithm

2. Apply the shears to the blocks of the input signal as in the shear algorithm

3. Use the full-window rectangular lattice DGT on the sheared signal blocks.

The downside of the shear-OLA algorithm is that the total length of the DGTs is longer than the original DGT by

$$\rho = \frac{L_g + L_b}{L_b}, \tag{3.57}$$

where L_b is the block length. Therefore, a trade-off between the block length and the window length must be found, so that the block length is long enough for (3.57) to be close to one, but at the same time small enough to not impose a too long processing delay.

Dual and tight windows

The shear method in Proposition 9 can also be used to compute the canonical dual and canonical tight windows, using the factorization of the frame operator given in (3.38). The complete algorithm for the canonical dual window is shown in Algorithm 2, and uses the same trick as the Gabor transform algorithm to compute the canonical dual when a frequency side shear is needed: do it in the Fourier domain without transforming back. Again, we assume the existence of an implementation GABDUAL for the computation of Gabor dual windows on separable lattices.

Algorithm 2 Dual window via shearing: $\tilde{g} = \text{GABDUALNS}(g, a, M, \lambda)$

```
$[s_0, s_1, b_r] = \text{SHEARFIND}(L, a, M, \lambda)$
if $s_1 \neq 0$ then
    $p \leftarrow \text{PCHIRP}(L, s_1)$
    $g(\cdot) \leftarrow p(\cdot)g(\cdot)$
end if
$b \leftarrow \frac{L}{M}$,    $M_r \leftarrow \frac{L}{b_r}$,    $a_r \leftarrow \frac{ab}{b_r}$
if $s_0 = 0$ then
    $g_d \leftarrow \text{GABDUAL}(g, a_r, M_r)$
else
    $p_0 \leftarrow \text{PCHIRP}(L, -s_0)$
    $g(\cdot) \leftarrow p_0(\cdot)\text{FFT}(g)(\cdot)$
    $g_d \leftarrow L \cdot \text{GABDUAL}(g, L/M_r, L/a_r)$
    $g_d \leftarrow \text{IFFT}(\overline{p_0}(\cdot)g_d(\cdot))$
end if
```

To compute the canonical dual and tight windows on a separable lattice, the matrices are first factorized as in [147, 150] and then the factorized matrices are transformed as in [100].

Analysis of the computational complexity

The flop counts of the various algorithms used for computing the DGT on a non-separable lattice is listed in Table 3.1. The additional parameters used are defined as follows. For the multiwindow algorithms we define

$$c_{mw} = \gcd(a_{mw}, M), \quad d_{mw} = \gcd(b, N_{mw}),$$
$$p_{mw} = N_{mw}/d_{mw}, \quad q_{mw} = M/c_{mw},$$

Table 3.1: Flop counts for different ways of computing the DGT on a nonseparable lattice. First column lists the algorithm, second column the flop count for the particular algorithm. Listed from the top, the algorithms are: The multiwindow algorithm using the full window rectangular lattice algorithm, the multiwindow algorithm using the FIR window rectangular lattice algorithm, the Smith normal form algorithm using the full window rectangular lattice algorithm, the shear algorithm when no frequency shear is needed, the shear algorithm including the frequency shear and finally the overlap-add versions of the shear algorithms. The term L_g denotes the length of the window used so L_g/a is the overlapping factor of the window.

Alg.:	Flop count
Multi-window	
FIR.	$8L\frac{L_g}{a} + 4NM\log_2 M$
Full.	$L\lambda_2\left(8q_{mw} + 4\log_2 d_{mw}\right)$ $+MN\left(4\log_2 L/p_{mw} + 6\right)$
SNF	$L\left(8q + 4\log_2 d_{sm} + 8\log_2 L + 18\right)$ $+MN\left(4\log_2 L/p + 6\right)$
Shear alg.	
No freq. shear	$L\left(8q + 4\log_2 d + 6k_{time}\right)$ $+MN\left(4\log_2 L/p + 6k_{time}\right)$
Freq. shear	$L\left(8q + 4\log_2 Lc_{sh} + 6 + 6k_{time}\right)$ $+MN\left(4\log_2 L/p + 6\right)$
Shear OLA	
No freq. shear	$\rho L\left(8q + 4\log_2 \rho d_{shola} + 6k_{time}\right)$ $+\rho MN\left(4\log_2 \rho L_b/p + 6k_{time}\right)$
Freq. shear	$\rho L\left(8q + 4\log_2 \rho Lc_{shola} + 6k_{time} + 6\right)$ $+\rho MN\left(4\log_2 \rho L_b/p + 6\right)$

where $a_{mw} = a\lambda_2$ and $N_{mw} = N/\lambda_2$. For the shear and shear OLA algorithms we define

$$c_{sh} = \frac{ca_{sh}}{a}, \quad d_{sh} = \frac{dM}{M_{sh}},$$

$$c_{shola} = \frac{ca_{shola}}{a}, \quad d_{shola} = \frac{dM}{M_{shola}},$$

$$k_{time} = \begin{cases} 1 & \text{if a time shear is used,} \\ 0 & \text{otherwise,} \end{cases}$$

where a_{sh}, M_{sh} and a_{shola}, M_{shola} are the parameters of the rectangular lattice the problem is reduced to in the respective algorithm.

Based on the computational complexity presented in the table, any of the algorithms may for some specific problem setup be the fastest, except for the Smith-normal form algorithm which is always slower than the shear algorithm:

- The multiwindow algorithm for FIR windows is the fastest for very short windows.
- The multiwindow-OLA algorithm is the fastest for simple lattices (λ_2 small) and medium length windows.
- The shear-OLA algorithm is the fastest for more complex lattices (λ_2 large) and medium length windows.
- The multiwindow algorithm is the fastest for simple lattices (λ_2 small) and very long windows.
- The shear algorithm is the fastest for more complex lattices (λ_2 large) and very long windows.

Numerical experiments

Implementations of the algorithms described in this paper can be found in the Large Time Frequency Analysis Toolbox (LTFAT), cf. [145], [2]. An appropriate algorithm will be automatically invoked when calling the DGT or DGTREAL functions. The implementations are done in both the MATLAB / OCTAVE scripting language and in C. All tests were performed on an Intel i7 CPU operating at 3.6 GHz.

As the speed of the algorithms depends on a large number of parameters a, M, L, L_g, c, d, s_0, s_1 and similar parameters relating to the multiwindow and shear transforms, we cannot provide an exhaustive illustration of the running times. Instead we will present some figures that illustrates the crossover point of when the the shear algorithm becomes faster than the multiwindow algorithm as the lattice complexity λ_2 increases. The behavior of the algorithms as the window length L_g increases is completely determined by the algorithms for the rectangular lattice, so we refer to [147] for illustrations.

The experiments shown in Figure 3.4 illustrate how the computational complexity of the running time of the algorithm depends on the lattice complexity λ_2: The complexity of the shear algorithm is independent of λ_2, while the complexity of the multiwindow algorithm grows linearly.

The bumps in the curves for the multiwindow algorithm are due to variations in q_{mw}: The multiwindow algorithm transforms the problem into λ_2 different DGTs that should be computed on a lattice with redundancy $q/(p\lambda_2)$. The number q_{mw}

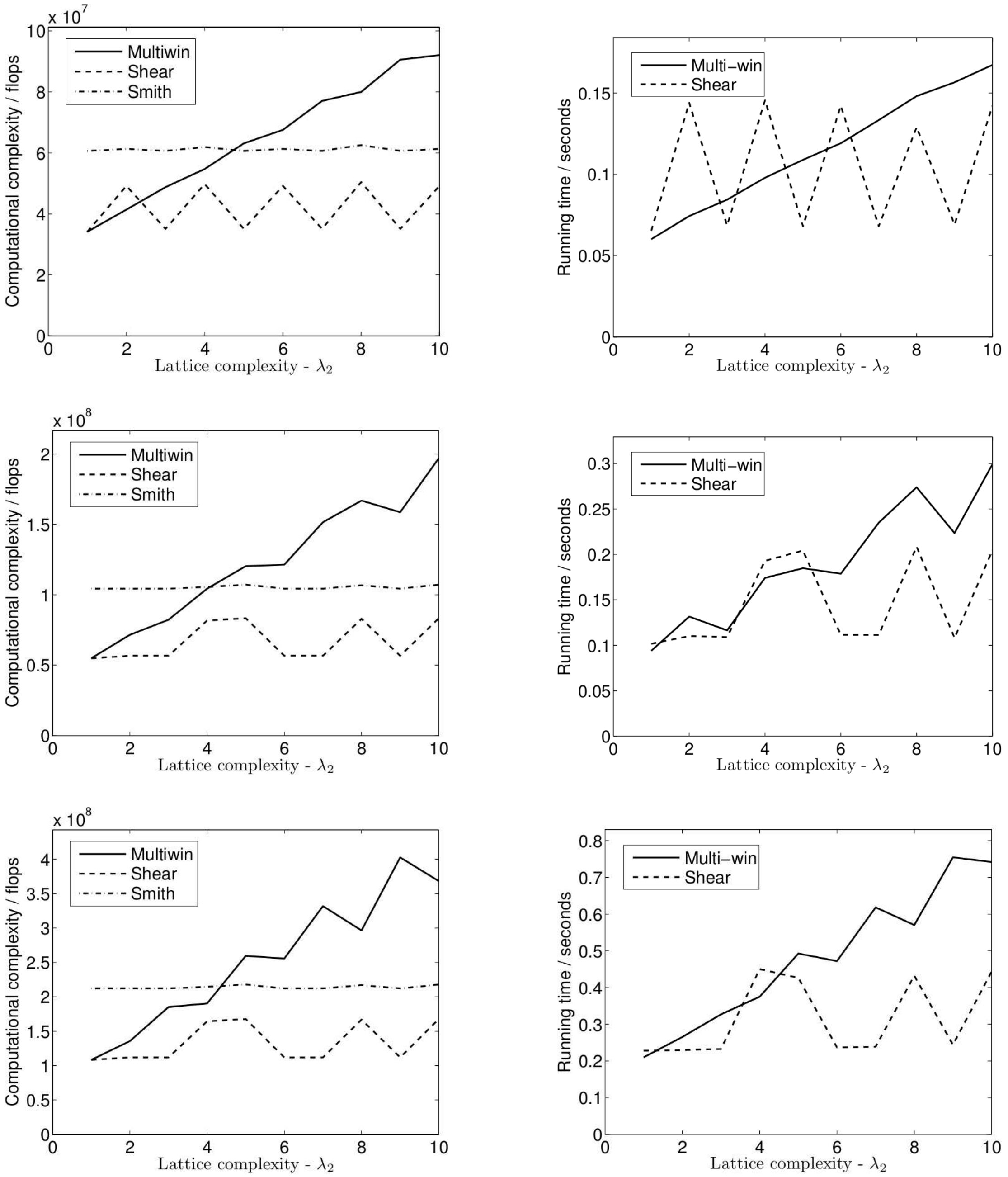


Figure 3.4: Computation of the DGT for nonseparable lattices with increasing lattice complexities, λ_2. The length is kept fixed at $L = \text{lcm}(a, M) \cdot 2520$ which is the minimal legal transform length for all the tested lattices. (left) Accurate flop counts and (right) the actual running time. The Gabor system parameters are $a = 32$, $M = 64$ ($p/q = 1/2$) (1st row), $a = 40$, $M = 60$ ($p/q = 2/3$) (2nd row) and $a = 60$, $M = 80$ ($p/q = 3/4$) (3rd row).

is the nominator of this written as an irreducible fraction, and depending on $p\lambda_2$ it may be smaller than q.

The bumps in the curves for the shear algorithm are caused by whether or not a frequency side shear is required for that particular lattice configuration, and to a lesser extent whether a time-side shear is needed. As the multiwindow algorithm is faster for simple lattices, there is a cross-over point where the shear algorithm becomes faster, but the cross-over point depends strongly on the exact lattice configuration. Just considering the flop counts would predict that the cross-over happens for a smaller value of λ_2 that what is really the case. This is due to the fact that there are more complicated indexing operations and memory reshuffling for the shear algorithm than for the multiwindow algorithm, and this is not properly reflected in the flop count.

The cross-over point where one algorithm is faster than the other is highly dependent on the interplay between the algorithm and the computer architecture. Experience from the ATLAS [168], FFTW [77] and SPIRAL [50] projects shows that in order to have the highest performance, it is necessary to select the algorithm for a given problem size based on previous tests done on the very same machine. Performing such an optimization is beyond the scope of this our study, and we therefore cannot make statements about how to choose the most efficient cross-over points.

Chapter 4

Theory of nonstationary Gabor systems

This chapter compiles results presented in the manuscript [91] and joint work with P. Balazs, M. Dörfler, F. Jaillet and G.A. Velasco, published in [10].

In this chapter, we investigate the properties of adaptive time-frequency systems that generalize classical Gabor systems. Although some of the presented results apply in a more general setting, the focus is on time-frequency systems with compactly supported generators.

Redundant short-time Fourier methods, also known as Gabor analysis [71] and previous chapters, are widely used in signal processing applications. The basic idea is the analysis of a signal f by consideration of the projections $\langle f, g_{x,\omega} \rangle$ of f onto time-frequency atoms $g_{x,\omega}$. The $g_{x,\omega}$ are obtained by translation of a unique prototype function over time and frequency: $g_{x,\omega}(t) = g(t-x)e^{2\pi i \omega t}$. Recall that, for $g \in L^2(\mathbb{R})$ and $a, b \in \mathbb{R}^+$, the corresponding *Gabor system* [78, 84] $\mathcal{G}(g, a, b)$ is the set of functions

$$g_{m,n}(t) = \mathbf{M}_{mb}\mathbf{T}_{na}g(t) = g(t-na)e^{2\pi i m b t}, \ \forall \ m, n \in \mathbb{Z}. \tag{4.1}$$

The prototype function g is also called *window* or *generator function*. This classical construction leads to a signal decomposition with fixed time-frequency resolution over the whole time-frequency plane.

Of particular interest are systems that allow for stable, perfect reconstruction of any function $f \in L^2(\mathbb{R})$ from the system coefficients, given by inner products with the system elements. Such systems are generally called *frames* [31, 57] or, when they are of the form $\mathcal{G}(g, a, b)$, *Gabor frames*. For any frame, there exists a possibly non-unique *dual frame* that enables the aforementioned perfect reconstruction. Gabor frames $\mathcal{G}(g, a, b)$ possess the nice property that, due to their highly structured nature, the existence of a dual frame with the same structure,

$\mathcal{G}(h, a, b)$ for some $h \in L^2(\mathbb{R})$, is guaranteed. This inheritance of structure from the original frame by a dual frame does not hold for more general frames and is one of the reasons why Gabor frames are so convenient to work with. One such dual frame is the *canonical dual*, obtained by applying the inverse *frame operator*, cf. Section 2 for details, to the frame elements.

One of the early and most prevalent results in the field is the theory of painless nonorthogonal expansions [48], where the authors determine a simple necessary and sufficient condition for Gabor frames $\mathcal{G}(g, a, b)$ with compactly supported generator and dense frequency sampling, i.e. small frequency step b, to constitute a frame. Then the frame operator is diagonal and thus easily inverted and the canonical dual generator $\tilde{g}$ has the same support as g. This setting is often referred to as the *painless case*, see also Section 2.3.

In applications, frames generated from compactly supported window functions are of particular interest, because they allow for the most efficient computation of the frame coefficients and reconstruction. Compact support of the frame generators is also crucial for real-time implementation. Thus, the investigation of such frames beyond the painless case is an active field, see e.g. [24,32,37,39,108] and [35,36]. In the latter two articles, Christensen, Kim and Kim prove that for any Gabor frame with $\text{supp}(g) \subseteq [1, 1]$, $a = 1$ and $b \in]1/2, 1[$, there exists a dual Gabor frame generated by a window supported on some compact set dependent only on the magnitude of b, cf. [36, Theorem 2.1, Lemma 3.2]. In fact, they show in [35] that the support condition in [36] can be further improved, for a large class of window functions g. This is also reflected in our own results in Section 4.3, although we recover only a special case of the results in [35]. The results in this manuscript are somewhat complementary to those of Christensen, Kim and Kim. To allow for a comparison, we recall some results from [36] in Section 4.3.

More results on the support of dual Gabor frames are due to Gröchenig and Stöckler [85, Theorem 9]. They prove the existence of dual frames with compactly supported, piecewise continuous generator for $\mathcal{G}(g, a, b)$ with g a totally positive function of finite type. While the class of functions treated by Gröchenig and Stöckler is quite different from the compactly supported functions in this contribution, the support size of the dual generator grows proportionally to the quotient $\frac{ab}{1-ab}$ in both cases.

The restriction of Gabor systems to a fixed resolution is often undesirable in processing signals with variable time-frequency characteristics. Alternative decompositions have been introduced to overcome this deficit, e.g. the wavelet transform [47], the constant-Q transform (CQT) [25] or decompositions using filterbanks [23], in particular based on perceptive frequency scales [86]. Adaptation over time is considered in approaches such as modulated lapped transforms [117], adapted local trigonometric transforms [167] or (time-varying) wavelet packets [131].

Most of the cited work achieves flexible tilings of the time-frequency plane, but efficient reconstruction from signal-adaptive, overcomplete time-frequency transforms is rarely addressed. One exception is a recent approach in [171], which is in fact a special case of the more general model considered in this chapter. The wealth of existing approaches to fast adaptive transforms underlines the need for flexibility arising from many applications. On the other hand, the introduction of flexibility in a transform that is based on accurate mathematical modeling causes technical complications that are not always easy to overcome.

We introduce a class of efficient adaptive time-frequency transforms, that allow for perfect invertibility under some easily verified conditions. As a straightforward generalization of Gabor transforms, they will be referred to as *nonstationary Gabor transforms* (NSGT). While classical Gabor systems are constructed from regular translations and modulations, *Nonstationary Gabor* (NSG) systems are generated by a countable set of window functions and modulations thereof, allowing for adaptivity of the analysis windows *and* the sampling points. Explicitly, we associate a sequence of pairs $\mathcal{G}(\mathbf{g},\mathbf{b}) := (g_n, b_n)_{n\in\mathbb{Z}}$, $g_n \in L^2(\mathbb{R})$ and $b_n \in \mathbb{R}^+$, with the set of functions

$$g_{m,n}(t) = \mathbf{M}_{mb_n} g_n(t) = g_n(t)e^{2\pi i m b_n t}, \quad \text{for all } m,n \in \mathbb{Z}. \tag{4.2}$$

If $\mathcal{G}(\mathbf{g},\mathbf{b})$ constitutes a frame, we call it a *nonstationary Gabor frame*. Note that a nonstationary Gabor system with $b_n = b$ and $g_n = T_{na}g$ for all $n \in \mathbb{Z}$ with $g \in L^2(\mathbb{R})$ and $a, b \in \mathbb{R}^+$ is a Gabor system.

Nonstationary Gabor frames combine the adaptivity of *local Fourier bases* [7, 117] with the flexibility of redundant systems to provide a powerful framework for time-frequency representations. Much like Gabor frames give rise to *Wilson bases* [20, 21, 49, 65], local Fourier bases can be constructed from NSG frames, although the more intricate properties of their relationship have yet to be investigated.

As a special case relevant for applications, we introduce a generalized notion of *painless nonorthogonal expansions* [48]. The central feature of painless expansions is the diagonality of the frame operator associated with the proposed analysis system, i.e. the frame operator is a simple multiplication operator. This idea is used here to yield painless nonstationary Gabor frames and will allow for both mathematical accuracy in the sense of perfect reconstruction (the frame operator is invertible) and numerical feasibility by means of an FFT-based implementation. The construction of painless nonstationary Gabor frames relies on three intuitively accessible properties of the windows and time-frequency shift parameters used.

1. The signal f of interest is localized at time- (or frequency-)positions n by

means of multiplication with a *compactly supported* (or bandlimited) window function g_n.

2. The Fourier transform is applied on the localized pieces $f \cdot g_n$. The resulting spectra are sampled densely enough in order to perfectly reconstruct $f \cdot g_n$ from these samples.

3. Adjacent windows overlap to avoid loss of information. At the same time, unnecessary overlap is undesirable. In other words, we assume that $0 < A \leq \sum_{n \in \mathbb{Z}} |g_n(t)|^2 \leq B < \infty$, a.e., for some positive A and B.

We will show that these requirements lead to invertibility of the frame operator and therefore to perfect reconstruction. Moreover, the frame operator is diagonal and its inversion is straightforward. Further, the canonical dual frame has the same structure as the original one. Because of these pleasant consequences following from the three above-mentioned requirements, the frames satisfying all of them will be called *painless nonstationary Gabor frames* and we refer to this situation as the *painless case*. Under application of a Fourier transform to the signal of interest, our approach leads to adaptivity in either time or frequency. The concept of this chapter relies on ideas introduced in [95], and presented at [96]. First, we give all formal proofs and provide the link to frame theory. The possibility to represent other analysis/synthesis systems with this approach is established. Numerical issues are investigated and several applications are presented in the following Chapter 5. Besides the applications presented therein, the painless construction is being used in realizing various time- or frequency-adaptive transforms [111,112,122,171].

In the second part of this chapter, we investigate the properties of more general NSG systems. For more results on nonstationary Gabor frames beyond the painless case, we refer to [55], [56].

Note that, in contrast to regular Gabor frames, the existence of a dual frame with the same structure, i.e. comprised of window functions h_n and modulation parameters b_n, is not guaranteed for general NSG frames. Indeed, one of the central results in this chapter details the structure of the canonical dual system under certain restrictions. These restrictions, concerning the support and overlap of the window functions g_n and the modulation parameters b_n, guarantee compact support for the elements of the canonical dual frame and a certain modulation and phase shift structure, detailed in Section 4.4. This structure can be deduced from that of the inverse frame operator, which is in turn determined using the Walnut representation of the NSG frame operator and the Neumann series representation of its inverse.

Further, we obtain a duality condition, necessary and sufficient for pairs of nonstationary Gabor systems $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$ to constitute dual frames, requiring only mild restrictions on the modulation parameters b_n. For a fixed NSG frame

$\mathcal{G}(\mathbf{g}, \mathbf{b})$, these equations might not be solvable, i.e. a dual system of the form $\mathcal{G}(\mathbf{h}, \mathbf{b})$ may not even exist. We determine a simple, yet somewhat restrictive, condition on $\mathcal{G}(\mathbf{g}, \mathbf{b})$, such that the duality conditions are solvable.

The frame and Bessel properties of generalized shift-invariant system (GSI) have been extensively studied by Hernández, Labate and Weiss [89], as well as Ron and Shen [137]. We refer to GSI systems as frequency-side NSG systems since they are equivalent to NSG systems via an application of the Fourier transform. As such, the results in [89,137] can be rewritten to give Bessel and frame duality conditions for nonstationary Gabor systems that are equivalent to our duality result, but for different technical conditions. Indeed the Dual Gramian, heavily used in [137], can be considered equivalent to the Walnut-like representations presented in Section 4.2 of this work. Nonetheless, we feel that our approach is sufficiently different from the techniques and restrictions applied in [89,137] to be interesting in its own right. Our results also apply to the classical Gabor case by choosing $b_n = b$ and $g_n = T_{na}g$ to describe the support of the canonical dual window $\tilde{g}$ in the setting considered in [36] and [35], complementing the results therein. By restricting the duality conditions for NSG systems in that way, we recover the famous duality conditions for Gabor systems [99, 133, 136] and a simple special case of a result in [35].

In this and subsequent chapters, we are mainly interested in Gabor and nonstationary Gabor systems that at least constitute a Bessel sequence. We associate $\mathbf{g}$ and $\mathbf{b}$ with the sequences $(g_n)_n$ and $(b_n)_n$, respectively.

4.1 Construction of nonstationary Gabor frames

4.1.1 Resolution changing over time

As opposed to standard Gabor analysis, where time translation is used to generate atoms, the setting of nonstationary Gabor frames allows for changing, hence adaptive, windows in different time positions. Then, for each time position, we build atoms by *regular* frequency modulation.

Definition 10. Let $\mathbf{g} := (g_n)_{n\in\mathbb{Z}}$ be a sequence of *window functions* in $L^2(\mathbb{R})$ and $\mathbf{b} := (b_n)_{n\in\mathbb{Z}}$, with $b_n \in \mathbb{R}^+$ for all n, a sequence of *frequency sampling steps*, then the associated *nonstationary Gabor (NSG)* system is the collection $\mathcal{G}(\mathbf{g}, \mathbf{b}) := \{g_{n,m}\}_{n,m\in\mathbb{Z}}$ of time-frequency atoms

$$g_{n,m}(t) = g_n(t)e^{2\pi i m b_n t} = \mathbf{M}_{mb_n} g_n(t).$$

Implicitly, we assume that the functions g_n are well-localized and centered around time-points a_n. This is similar to the standard Gabor scheme, however with the possibility to vary the window g_n for each position a_n. Thus, sampling of the time-frequency plane is done on a grid which is irregular over time, but regular over frequency at each temporal position.

Figure 4.1 shows an example of such a sampling grid. Note that some results exist in Gabor theory for semi-regular sampling grids, see e.g. Chapter 3 or [29]. Our study uses a more general setting, as the sampling grid is in general not a subgroup of the time-frequency plane and, more importantly, the window can evolve over time. To get a first idea of the effect of nonstationary Gabor frames, the reader may take a look at Figure 4.2 and Figure 4.3, which show regular Gabor transforms and a nonstationary Gabor transform of the same signal. Note that the NSGT in Figure 4.3 was adapted to transients and the components are well-resolved.

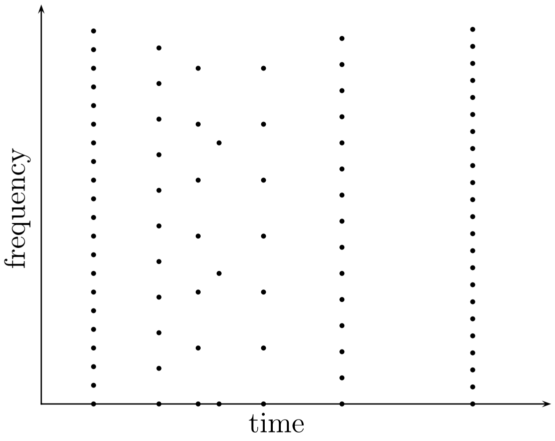


Figure 4.1: Example of a sampling grid of the time-frequency plane when building a decomposition with time-frequency resolution evolving over time

In the current situation, the analysis coefficients may be written as

$$c_{n,m} = c[n,m] = \langle f, \mathbf{M}_{mb_n} g_n \rangle = \widehat{(f \cdot \overline{g_n})}(mb_n),\ m,n \in \mathbb{Z}.$$

Remark 6. *If we set $g_n(t) = g(t - na)$ for a fixed time-constant a and $b_n = b$ for all n, we obtain the case of classical painless nonorthogonal expansions for regular Gabor systems introduced in [48].*

4.1.2 Resolution changing over frequency

An analog construction in the frequency domain leads to irregular sampling over frequency, together with windows featuring adaptive bandwidth. Then, sampling is regular over time. An example of the sampling grid in such a case is given in Figure 4.4.

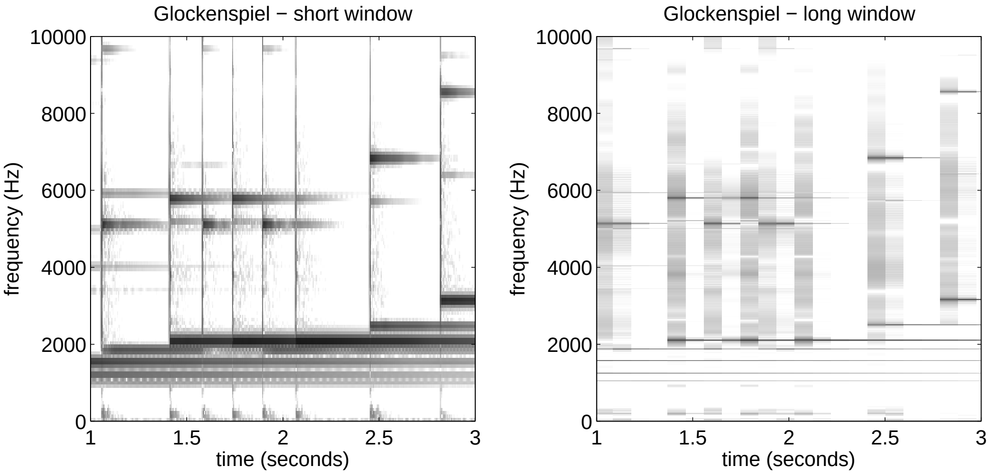


Figure 4.2: Glockenspiel (Example 1). Gabor representations with short window (11.6 ms), resp. long window (185.8 ms).

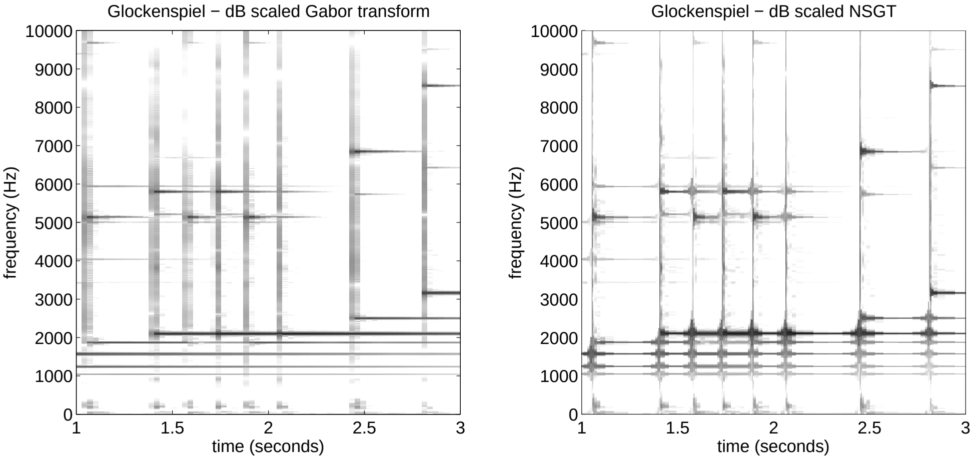


Figure 4.3: Glockenspiel (Example 1). Regular Gabor representation with a Hann window of 58 ms length and a nonstationary Gabor representation using Hann windows of varying length.

Definition 11. Let $\mathbf{g} := (g_m)_{m\in\mathbb{Z}}$ be a sequence of *window functions* in $L^2(\mathbb{R})$ and $\mathbf{a} := (a_m)_{m\in\mathbb{Z}}$, with $a_m \in \mathbb{R}^+$ for all m, a sequence of *time sampling steps*, then the associated *(frequency side) nonstationary Gabor system* is the collection $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a}) := \{g_{n,m}\}_{n,m\in\mathbb{Z}}$ of time-frequency atoms

$$g_{n,m}(t) = \mathcal{F}^{-1}\left(\mathbf{M}_{-na_m} g_m\right)(t) = \mathbf{T}_{na_m}\widecheck{g_m}(t). \tag{4.3}$$

Therefore $\widehat{g_{n,m}}(\xi) = g_m(\xi)\cdot e^{-2\pi i n a_m \xi}$ and the analysis coefficients may be written as

$$c_{n,m} = c[n,m] = \langle f, g_{n,m}\rangle = \langle \hat{f}, \mathcal{F}(\mathbf{T}_{na_m}\widecheck{g_m})\rangle = \mathcal{F}^{-1}\left(\hat{f}\cdot\overline{g_m}\right)(na_m).$$

Hence, the situation is completely analogue to the one described in the previous section, up to a Fourier transform.

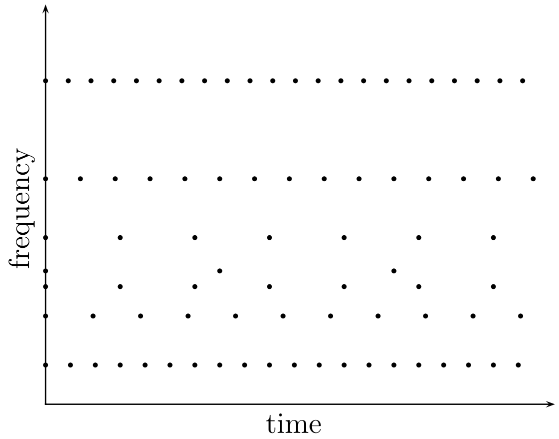


Figure 4.4: Example of a sampling grid of the time-frequency plane when building a decomposition with time-frequency resolution changing over frequency

In practice we will choose each function g_m as a well localized band-pass function with center frequency b_n.

Link between nonstationary Gabor frames, wavelet frames and filterbanks:

To obtain wavelet frames, see Section 2.4, the wavelet transform in (2.35) is sampled at sampling points (b_n, a_m). A typical discretization scheme [115] is (nb_0, a_0^m). Then, the frame elements are $\varphi_{m,n}(t) = \mathbf{T}_{nb_0}\mathbf{D}_{a_0^m}\varphi(t)$. Comparing this expression to (4.3) and setting $g_m = \mathbf{D}_{a_0^m}\varphi$ and using a regular translation step b_0, we see that a wavelet frame with this discretization scheme corresponds to a nonstationary Gabor transform.

Another possibility for sampling the continuous wavelet transform [47] uses $(nb_0a_0^m, a_0^m)$. The resulting systems are often referred to as *affine systems*. Again, we obtain a correspondence to nonstationary Gabor frames by setting $g_m = \mathbf{D}_{a_0^m}\varphi$ and translation step $b_0 \cdot a_0^m$ on scale m.

Beyond the setting of wavelets, any *filterbank* [115], even with non-constant down-sampling factors D_m, can be written as a nonstationary Gabor frame, so long as the transfer functions are known explicitly. A filterbank is a set of time-invariant, linear filters h_m, i.e. Fourier multipliers. The response of a filterbank for the signal f and sampling period T_0 is given (in the continuous case) by

$$c_{n,m} = (f * h_m)\,(nD_mT_0) = \int\limits_{\mathbb{R}} f(t)h_m\,(nD_mT_0 - t)\,dt = \langle f, g_{n,m}\rangle\,,$$

where $g_{n,m}(t) = \overline{h_m\,(nD_mT_0 - t)}$. Setting $g_m = \overline{h_m(-\cdot)}$ and choosing $a_m = D_mT_0$ this construction is realized with nonstationary Gabor frames using (4.3). If the filters are band-limited and the down-sampling factors are small enough, then the conditions for the painless case are met, see Theorem 7 in the next section. Consequently, the corresponding reconstruction procedure can be applied.

4.1.3 Painless nonstationary Gabor frames - Invertibility and Reconstruction

In this central section we give the precise conditions under which painless nonstationary Gabor frames are constructed. The first two basic conditions, namely compactly supported windows and sufficiently dense frequency sampling points, lead to diagonality of the associated frame operator $\mathbf{S}$, defined as in Section 2.2. Analogous to the discrete (matrix) case, we say that an operator is *diagonal* if it is a multiplication operator and *non-diagonal* otherwise. The third condition, the controlled overlap of adjacent windows, then leads to boundedness and invertibility of $\mathbf{S}$. The following theorem generalizes the results given for the classical case of painless nonorthogonal expansions [48, 84].

Theorem 7. *For every $n \in \mathbb{Z}$, let the function $g_n \in L^2(\mathbb{R})$ be compactly supported with* $\operatorname{supp}(g_n) \subseteq [c_n, d_n]$ *and let b_n be chosen such that $d_n - c_n \leq b_n^{-1}$. The frame operator*

$$\mathbf{S} : f \mapsto \sum_{n,m} \langle f, g_{n,m}\rangle g_{n,m}$$

of the system

$$g_{n,m}(t) = g_n(t)\,e^{2\pi i m b_n t},\ m \in \mathbb{Z} \text{ and } n \in \mathbb{Z},$$

is given by a multiplication operator of the form

$$\mathbf{S}f = \left(\sum_n b_n^{-1}|g_n|^2\right) f, \text{ almost everywhere.} \tag{4.4}$$

Proof. Note that,

$$\begin{aligned}\langle \mathbf{S}f, f\rangle &= \sum_n \sum_m \Big| \int_{\mathbb{R}} f(t)\,\overline{g_n(t)}\,e^{-2\pi imb_nt}dt\Big|^2\\ &= \sum_n \sum_m \Big| \int_{c_n}^{d_n} f(t)\,\overline{g_n(t)}\,e^{-2\pi imb_nt}dt\Big|^2,\end{aligned}$$

due to the compact support property of the windows g_n. Let $I_n = [c_n, c_n + b_n^{-1}]$ for all n and χ_I denote the characteristic function of the interval I. Taking into account the compact support of g_n again, it is obvious that

$$f\,\overline{g_n} = \chi_{I_n} \sum_k \mathbf{T}_{kb_n^{-1}}(f\overline{g_n}),$$

with the b_n^{-1}-periodic function $\sum_k \mathbf{T}_{kb_n^{-1}}(f\,\overline{g_n})$. Hence, with $W_{n,m}(t) = e^{-2\pi imb_nt}$,

$$\begin{aligned}\Big| \int_{c_n}^{d_n} f(t)\,\overline{g_n(t)}\,W_{n,m}(t)\,dt\Big|^2 &= \Big| \int_{I_n} f(t)\,\overline{g_n(t)}\,W_{n,m}(t)\,dt\Big|^2,\\ &= \big|\langle f\,\overline{g_n}, W_{n,m}\rangle_{L^2(I_n)}\big|^2\end{aligned}$$

and applying Parseval's identity to the sum over m yields

$$\begin{aligned}\langle \mathbf{S}f, f\rangle &= \sum_n \sum_m |\langle f\,\overline{g_n}, W_{n,m}\rangle_{L^2(I_n)}|^2\\ &= \sum_n b_n^{-1}\|f\,\overline{g_n}\|^2 = \big\langle \sum_n b_n^{-1}|g_n|^2 f, f\big\rangle.\end{aligned}$$

□

Provided $\mathcal{G}(\mathbf{g}, \mathbf{b})$ is a frame with $b_n = b$ for all $n \in \mathbb{Z}$, in particular if it is a regular Gabor frame, then it can easily be shown that the frame operator $\mathbf{S}_{\mathcal{G}(\mathbf{g},\mathbf{b})}$ commutes with modulations of the form $\mathbf{M}_{mb}$, with $m \in \mathbb{Z}$, and thus the canonical dual frame is of the form $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{b})$. This also holds for systems with non-diagonal frame operator. Under the label *Fourier-like systems*, NSG systems with uniform modulation parameter and their frame properties have recently been treated in [38].

The setup described in Theorem 7 is usually referred to as the *painless case* and $\mathcal{G}(\mathbf{g}, \mathbf{b})$ is called a *painless system*. While in general, the inversion of $\mathbf{S}$ can be numerically unfeasible, in the special case described in Theorem 7, the invertibility of the frame operator is easy to check and inversion is a simple multiplication.

Corollary 8. *Under the conditions given in Theorem 7, the system of functions $g_{n,m}$ forms a frame for $L^2(\mathbb{R})$ if and only if $\sum_n b_n^{-1}|g_n(t)|^2 \simeq 1$. In this case, the canonical dual frame elements are given by:*

$$\tilde{g}_{n,m}(t) = \frac{g_n(t)}{\sum_k b_k^{-1}|g_k(t)|^2} e^{2\pi i m b_n t}, \tag{4.5}$$

and the associated canonical tight frame elements can be calculated as:

$$\mathring{g}_{n,m}(t) = \frac{g_n(t)}{\sqrt{\sum_k b_k^{-1}|g_k(t)|^2}} e^{2\pi i m b_n t}.$$

Remark 7. *The optimal lower and upper frame bounds are explicitly given by $A_0 = \text{ess inf} \sum_n b_n^{-1}|g_n(t)|^2$ and $B_0 = \text{ess sup} \sum_n b_n^{-1}|g_n(t)|^2$.*

We next state the results of Theorem 7 and Corollary 8 in the Fourier domain. This is the basis for adaptation over frequency.

Corollary 9. *For every $m \in \mathbb{Z}$, let the function $\check{g_m}$ be band-limited to $\text{supp}(g_m) = [c_m, d_m]$ and let a_m be chosen such that $d_m - c_m \leq a_m^{-1}$. Then the frame operator of the system*

$$g_{n,m}(t) = \check{g_m}(t - na_m)\,, m \in \mathbb{Z}, n \in \mathbb{Z}$$

is given by a convolution operator of the form

$$\langle \mathbf{S}f, f\rangle = \langle \mathcal{F}^{-1}\big(\sum_m a_m^{-1}|g_m|^2\big) * f, f\rangle \tag{4.6}$$

for $f \in L^2(\mathbb{R})$. Hence, the system of functions $g_{n,m}$ forms a frame of $L^2(\mathbb{R})$ if and only if $\sum_m a_m^{-1}|g_m(\xi)|^2 \simeq 1$. The elements of the canonical dual frame are given by

$$\tilde{g}_{n,m}(t) = \mathbf{T}_{na_m}\mathcal{F}^{-1}\left(\frac{g_m}{\sum_k a_k^{-1}|g_k|^2}\right)(t) \tag{4.7}$$

and the canonical tight frame is given by

$$\mathring{g}_{n,m}(t) = \mathbf{T}_{na_m}\mathcal{F}^{-1}\left(\frac{g_m}{\sqrt{\sum_k a_k^{-1}|g_k|^2}}\right)(t). \tag{4.8}$$

Proof. We deduce the form of the frame operator in the current setting from the proof of Theorem 7 by setting

$$\langle \mathbf{S}f, f\rangle = \langle \widehat{\mathbf{S}f}, \hat{f}\rangle = \sum_{n,m} |\langle \hat{f}, \widehat{g_{n,m}}\rangle|^2$$

and the rest of the corollary is equivalent to Corollary 1. □

Remark 8. *As mentioned in Section 4.1.2, the NSGT is linked to wavelet frames. In the painless case it is possible to construct a dual sequence which has the same structure. Consequently, Corollary 9 is a generalization of the painless wavelet construction in [48]. For non-painless wavelet frames it is sometimes also possible to construct a dual sequence which has the same structure, see e.g. [58, 59], where alternative duals are constructed.*

4.2 Walnut and Walnut-like representations

Both regular and nonstationary Gabor frame-type operators admit a so-called *Walnut representation*, i.e. a representation purely in terms of translates of the frame generators and the function to which the operator is applied. The Walnut representation for nonstationary Gabor frames has only recently been rigorously proven for systems constructed from window functions in the Wiener space [55], recall Definition 2. Here, we also use a variant for Bessel sequences. For the proof, we refer the interested reader to [55], since the Bessel case only requires minor modifications.

Proposition 12 (Dörfler, Matusiak (2011) [55]). *Let $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$ be nonstationary Gabor systems with $b_n \in \mathbb{R}^+$ and $g_n \in L^2(\mathbb{R})$, for all $n \in \mathbb{Z}$. If either*

(i) $g_n,\ h_n \in W(L^\infty, \ell^1, \mathbb{R})$ for all $n \in \mathbb{Z}$, or

(ii) $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$ are Bessel sequences,

then the associated frame-type operator $\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} := \mathbf{D}_{\mathcal{G}(\mathbf{h},\mathbf{b})}\mathbf{C}_{\mathcal{G}(\mathbf{g},\mathbf{b})}$ admits a Walnut representation of the form

$$\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} f = \sum_{n,k\in\mathbb{Z}} b_n^{-1} h_n \mathbf{T}_{kb_n^{-1}} \overline{g_n} \mathbf{T}_{kb_n^{-1}} f, \quad \textit{for all } f \in L^2(\mathbb{R}). \tag{4.9}$$

Substituting b_n by b and g_n by $\mathbf{T}_{na} g$ for all $n \in \mathbb{Z}$ yields the Walnut representation of Gabor frame-type operators (Proposition 2). Setting $\mathbf{h} = \mathbf{g}$ yields the Walnut representation for the frame operator.

The Walnut representation shows that the frame operator maps a function $f \in L^2(\mathbb{R})$ onto a sum of weighted, translated copies of itself, where the weight functions are given by $\omega_{n,k} := b_n^{-1} g_n \mathbf{T}_{-kb_n^{-1}} \overline{g_n}$, for all $n, k \in \mathbb{Z}$ and the corresponding translates are $\mathbf{T}_{-kb_n^{-1}}$.

The *painless case* result, Theorem 7 can alternatively be derived from the Walnut representation easily: If $\operatorname{supp}(g_n) \subseteq [c_n, d_n]$ and $b_n^{-1} \geq d_n - c_n$, then $\omega_{n,k} = b_n^{-1} g_n \mathbf{T}_{-kb_n^{-1}} \overline{g_n} \equiv 0$ for all $k \neq 0$ and thus $\mathbf{S}$ is diagonal. Furthermore, boundedness of the sum in (4.4) is a necessary condition for any NSG system to constitute a Bessel sequence.

Proposition 13. *Let $\mathcal{G}(\mathbf{g},\mathbf{b})$ and $\mathcal{G}(\mathbf{h},\mathbf{b})$ be nonstationary Gabor Bessel sequences with $b_n \in \mathbb{R}^+$ and $g_n \in L^2(\mathbb{R})$, for all $n \in \mathbb{Z}$. Let B be a joint Bessel bound of $\mathcal{G}(\mathbf{g},\mathbf{b})$ and $\mathcal{G}(\mathbf{h},\mathbf{b})$. Then*

$$\sum_{n\in\mathbb{Z}} b_n^{-1}|h_n\mathbf{T}_\tau\overline{g_n}| \leq B \quad a.e. \tag{4.10}$$

In particular $\sum_{n\in\mathbb{Z}} b_n^{-1}|g_n|^2 \leq B$ almost everywhere.

Proof. To prove $\sum_{n\in\mathbb{Z}} b_n^{-1}|g_n|^2 \leq B$, we retrace the steps of a proof by Chui and Shi [40] for Wavelet frames. By the Bessel property of $\mathcal{G}(\mathbf{g},\mathbf{b})$ and Plancherel's theorem for Fourier series,

$$\begin{aligned} B\|f\|^2 \geq \sum_{n,k\in\mathbb{Z}} |\langle f, g_{n,k}\rangle|^2 &= \sum_{n,k\in\mathbb{Z}} \left| \int_0^{b_n^{-1}} \sum_{l\in\mathbb{Z}} \mathbf{T}_{lb_n^{-1}} f(t) \mathbf{T}_{lb_n^{-1}} \overline{g_n(t)} e^{-2\pi i k b_n t}\, dt \right|^2 \\ &= \sum_{n\in\mathbb{Z}} b_n^{-1} \int_0^{b_n^{-1}} \left| \sum_{l\in\mathbb{Z}} \mathbf{T}_{lb_n^{-1}} f(t) \mathbf{T}_{lb_n^{-1}} \overline{g_n(t)} \right|^2 dt. \end{aligned}$$

Observe b_n^{-1}-periodicity of the integrand. For all $0 < N \in \mathbb{Z}$, we can choose some $\epsilon > 0$, such that for all $t_0 \in \mathbb{R}$ and $f = \sqrt{2\epsilon}^{-1}\chi_{[t_0-\epsilon,t_0+\epsilon]}$

$$\begin{aligned} &\sum_{n=-N}^{N} \frac{1}{b_n} \int_{t_0-b_n^{-1}/2}^{t_0+b_n^{-1}/2} \left| \sum_{l\in\mathbb{Z}} \mathbf{T}_{lb_n^{-1}} f(t) \mathbf{T}_{lb_n^{-1}} \overline{g_n(t)} \right|^2 dt \\ &= \sum_{n=-N}^{N} \frac{1}{2\epsilon b_n} \int_{t_0-\epsilon}^{t_0+\epsilon} |g_n(t)|^2\, dt \leq B\|f\| = B \end{aligned}$$

holds. Subsequently taking limits over ϵ and N proves $\sum_{n\in\mathbb{Z}} b_n^{-1}|g_n|^2 \leq B$ almost everywhere.

The general case follows by Cauchy-Schwarz' inequality:

$$\sum_{n\in\mathbb{Z}} b_n^{-1}|h_n\mathbf{T}_\tau\overline{g_n}| \leq \Big(\sum_n b_n^{-1}|h_n|^2 \sum_l b_l^{-1}|g_l|^2\Big)^{1/2} \leq B, \tag{4.11}$$

for all $\tau \in \mathbb{R}$. □

The Walnut representation is a very handy tool, describing the action of NSG frame operators in an intuitive way. We would like to use a slightly more general definition, though.

Definition 12. Let Λ be a countable index set, X a dense subspace of $L^2(\mathbb{R})$ and $\mathbf{W} : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ a bounded linear operator. If sequences $(\omega_\lambda)_{\lambda\in\Lambda}$ and $(a_\lambda)_{\lambda\in\Lambda}$ of bounded functions $\omega_\lambda \in L^\infty(\mathbb{R})$ and scalars $a_\lambda \in \mathbb{R}$ exist such that

$$\mathbf{W}f = \sum_{\lambda\in\Lambda} \omega_\lambda \mathbf{T}_{a_\lambda} f, \text{ for all } f \in X \tag{4.12}$$

and the sum on the right-hand side is unconditionally convergent, then we say that $\mathbf{W}$ has a *Walnut-like representation* with *weights* ω_λ and *translation constants* a_λ.

Operators of this form have been introduced before, see e.g. [9], albeit not under that name, in the context of multi-window Gabor frames in amalgam spaces. The authors show that the operators with Walnut-like representation, such that the L^∞-norms of the weights ω_λ are summable in a weighted ℓ^1-sense, form a Banach $*$-algebra and that their inverse, if it exists, is an operator with Walnut-like representation. In that context, the results we give in Theorems 9 and 10 can be seen as a particular case of a larger theory investigating in the structure of the inverse of certain operators. While the results in [9], see also the references therein, are asymptotic in nature, by restricting to a very special case we obtain the form of the inverse operator more explicitly. For regular Gabor systems, the Walnut representation has been shown to be absolutely convergent by Janssen [99]. For more general NSG systems, we discuss an alternate Walnut-like representation of the nonstationary Gabor frame operator and its unconditional convergence in Section 4.4.1.

Under weak additional assumptions, we can show that in fact, the weights corresponding to a fixed translate of f in (4.12) are bounded by the operator norm of $\mathbf{W}$.

Lemma 4. *Let $\mathbf{W} : L^2(\mathbb{R}) \mapsto L^2(\mathbb{R})$ be a bounded linear operator with Walnut-like representation. If $\|\mathbf{W}\|_{op} = C < \infty$ and for all $c, d \in \mathbb{R}$ with $c < d$, $\{a_\lambda : \omega_\lambda |_{[c,d]} \neq 0\}_{\lambda\in\Lambda}$ is free of accumulation points, then*

$$\Big| \sum_{\substack{\lambda\in\Lambda \\ a_\lambda = a_{\lambda_0}}} \omega_\lambda \Big| \leq C \quad a.e., \tag{4.13}$$

for all $\lambda_0 \in \Lambda$.

Proof. Without loss of generality, assume

$$\sum_{\substack{\lambda\in\Lambda \\ a_\lambda = a_{\lambda_0}}} \omega_\lambda \geq C_0 > C \text{ a.e. on } M \text{ with } \mu(M) > 0.$$

Then for all $\delta > 0$, there exists $l \in \mathbb{Z}$, such that $\mu(M \cap B_\delta(2l\delta)) > 0$. Furthermore, since $\{a_\lambda \ : \ \omega_\lambda \mid_{[c,d]} \neq 0\}_{\lambda\in\Lambda}$ has no accumulation points for all $c < d$, we can choose a pair $\delta > 0$, $l \in \mathbb{Z}$ such that

$$\{a_\lambda \ : \ \omega_\lambda \mid_{B_\delta(2l\delta)} \neq 0\}_{\lambda\in\Lambda} \cap B_{2\delta}(a_{\lambda_0}) = \{a_{\lambda_0}\}$$

and furthermore $\mu(M_l) > 0$, with $M_l = M \cap B_\delta(2l\delta)$. Take $f = \chi_{M_l - a_{\lambda_0}}$. If $f \in X$, then

$$\left|\mathbf{W}f \mid_{B_{\delta_0}(2l\delta_0)}\right| \geq C_0 |\mathbf{T}_{a_{\lambda_0}} f| \quad \Rightarrow \quad \|\mathbf{W}f\| \geq C_0 \|f\|,$$

contradicting $\|\mathbf{W}\|_{op} = C < C_0$. If $f \notin X$, construct a sequence $(f_n \in X)_{n\in\mathbb{N}}$ converging to f. For such a sequence, some $n_0 \in \mathbb{N}$ exists, such that $\|\mathbf{W}f_n\| > C\|f_n\|$, for all $n \geq n_0$. □

Remark 9. *If on the other hand,* $\mathbf{W} : L^2(\mathbb{R}) \mapsto L^2(\mathbb{R})$ *is linear and* $\mathbf{W}f$ *can be written in the form* (4.12) *for all* f *in a dense subspace of* $L^2(\mathbb{R})$*, then* $\mathbf{W}$ *is guaranteed to be a bounded linear operator if* $\sum_{\lambda\in\Lambda} \|\omega_\lambda\|_\infty < \infty$*.*

4.3 Results in the regular case

In this section, we recall a result of Christensen, Kim and Kim [36] and state the results of the following Section 4.4 in a simplified form for regular Gabor frames. Thus, this section demonstrates the application of our results to a classical setting and eases the reader into the technicalities necessary for the description of the general case. The results discussed herein are special cases of and follow directly from the results presented in Section 4.4.

We start by fixing some notation. For the rest of this section, we assume g, as used in the Gabor system $\mathcal{G}(g, a, b)$, to be compactly supported with $\mathrm{supp}(g) = [c, d]$. Let

$$\begin{aligned} I_{n,0} &= [c, d] + na, \\ I^+_{n,k} &= [c - (k-1)a + kb^{-1}, d] + na, \\ I^-_{n,k} &= [c, d + (k-1)a - kb^{-1}] + na, \end{aligned} \tag{4.14}$$

for all $n \in \mathbb{Z}, k \in \mathbb{N}$.

These sets will be helpful in describing both the support of the weight functions of the Walnut-like representation of $\mathbf{S}^{-1}$, as well as the support of the canonical dual window $\mathbf{S}^{-1}g$ in the case that $\mathcal{G}(g, a, b)$ constitutes a frame. The conditions placed on $\mathcal{G}(g, a, b)$ in Theorems 8 and 9 will be seen to imply $I^\pm_{n,k+1} \subseteq I^\pm_{n,k} \subseteq I_{n,0}$ and $I^+_{n,1} \cap I^-_{n,1} = \emptyset$ for all $n \in \mathbb{Z}, k \in \mathbb{N}$.

The following theorem combines two results in [36], rewritten in our notation:

Theorem 8 (Christensen, Kim, Kim). *Let $g \in L^2(\mathbb{R})$ supported on $[-1,1]$ and $b \in]1/2, 1[$. Assume that $\mathcal{G}(g,1,b)$ is a frame and set $K := \lfloor \frac{b}{1-b} \rfloor$, then*

(i) [36, Th 2.1] there exists a dual window $h \in L^2(\mathbb{R})$ with $\operatorname{supp} h \subseteq [-K, K]$.

(ii) [36, Lem 3.2] If g is bounded, $K > 1$ and $\mathcal{G}(h,1,b)$, with $h \in L^2(\mathbb{R})$ supported on $[-K,K]$, is a dual frame, then h is essentially supported on a subset of

$$I_{0,0} \cup \bigcup_{k=1}^{K} \left(I^-_{-k,k} \cup I^+_{k,k} \right).$$

The main tool used in [36] is the duality condition for Gabor Bessel sequences $\mathcal{G}(g,a,b)$ and $\mathcal{G}(h,a,b)$ to form dual frames [99,133,136]

$$b^{-1} \sum_{n \in \mathbb{Z}} T_{na} h \overline{T_{kb^{-1}+na} g} = \begin{cases} 1 \ a.e. & \text{for } k = 0, \\ 0 \ a.e. & \text{else.} \end{cases} \tag{4.15}$$

In Section 4.4.1, we will discuss the existence of a similar duality condition for nonstationary Gabor systems $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$.

Our following result is a restriction of Theorem 10 to Gabor systems, showing that the canonical dual window of $\mathcal{G}(\mathbf{g}, \mathbf{b})$ satisfies the properties attributed to h in Theorem 8 (i) and (ii). The conditions on g, a and b, while written differently as a preparation for Theorem 10, are equivalent to those in Theorem 8. We note that, by restricting g to be a continuous, compactly supported function with finitely many zeros inside its support, Christensen, Kim and Kim show that the frame property of $\mathcal{G}(g,1,b)$ is equivalent to the existence of a continuous function $h \in L^2(\mathbb{R})$, with support contained in $[-K,K]$ such that g, h satisfy the duality relations above. Our result investigates the structure of the inverse frame operator and derives properties of the canonical dual frame, but we do not attempt to characterize the frame property.

Theorem 9. *Let $g \in L^2(\mathbb{R})$ with $\operatorname{supp}(g) \subseteq [c,d]$ and $d > c$. Furthermore, let $a \in [\frac{d-c}{2}, d-c[$, $b \in]0, \frac{1}{a}[$ and $K = \lfloor \frac{(d-c-a)b}{1-ab} \rfloor$. If $\mathcal{G}(g,a,b)$ is a frame, the following hold.*

(i) The inverse frame operator $\mathbf{S}^{-1}$ has a Walnut-like representation of the form

$$\mathbf{S}^{-1} f = \sum_{k=0}^{K} \omega_k \mathbf{T}_{-kb^{-1}} f, \tag{4.16}$$

with $\operatorname{supp}(\omega_k) \subseteq \bigcup_{n \in \mathbb{Z}} I^-_{n,k}$ and $\operatorname{supp}(\omega_{-k}) \subseteq \bigcup_{n \in \mathbb{Z}} I^+_{n,k}$ for all $k \in \mathbb{N}$.

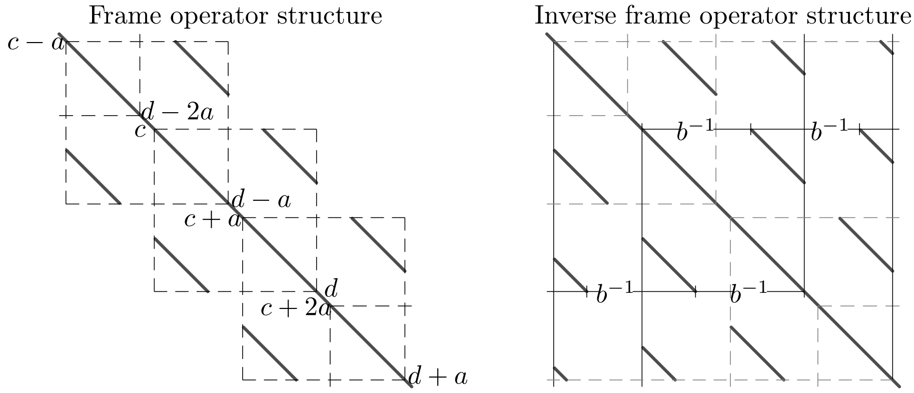


Figure 4.5: Section of a Gabor frame operator and its inverse in the setting of Theorem 9 (schematic illustration). *Left*: The weights correspond to side-diagonal entries of a matrix, with ω_0 (the main diagonal) and $\omega_{\pm 1}$ located on side-diagonal $\pm b^{-1}$. Grey diagonal lines indicate non-zero entries in the side-diagonals/weights and we see that at most 3 entries in each row are non-zero. Dashed lines indicate the support of the translates of g. *Right*: The inverse frame operator additionally possesses a regularly spaced set of weights ω_k, located on the side-diagonals kb^{-1}. Their non-zero entries are constrained by the support of the respective translates of g, indicated by horizontal and vertical lines. The parameter choice leads to shrinking support for weights located further from the main diagonal.

(ii) The canonical dual window $\tilde{g} = \mathbf{S}^{-1}g \in L^2(\mathbb{R})$ *satisfies*

$$\operatorname{supp}(\tilde{g}) \subseteq I_{0,0} \cup \bigcup_{k=1}^{K} \left(I^-_{-k,k} \cup I^+_{k,k} \right). \tag{4.17}$$

Borrowing intuition from the discrete case, the inverse frame operator $\mathbf{S}^{-1}$ can, according to Theorem 9 (i), informally be interpreted as an infinitesimal matrix, supported only on the main diagonal and a discrete set of side-diagonals which in turn are non-zero only on specific intervals. For an illustration, see Figure 4.5.

We see that Theorems 8 and 9 are complementary and shed light on the same problem from somewhat different points of view.

Example 1. Assume that $\mathcal{G}(g, \frac{7}{6}, \frac{3}{5})$, with $g \in L^2(\mathbb{R})$ continuous, $\operatorname{supp}(g) = [-1, 1]$ and $g(t) > 0$ for all $t \in]-1, 1[$, constitutes a frame. The frame operator of $\mathcal{G}(g, \frac{7}{6}, \frac{3}{5})$ can be written as

$$\mathbf{S}^{-1}f = \sum_{k=-1}^{1} \omega_k \mathbf{T}_{-kb^{-1}} f,$$

with the essential supports of ω_1 and ω_{-1} contained in $\bigcup_{n\in\mathbb{Z}}\left[-1+\frac{7n}{6}, 1+\frac{7n-10}{6}\right]$ and $\bigcup_{n\in\mathbb{Z}}\left[-1+\frac{7n+10}{6}, 1+\frac{7n}{6}\right]$, respectively. Consequently,

$$\operatorname{supp}(\mathbf{S}^{-1}g) \subseteq \left[-\frac{13}{6}, -\frac{11}{6}\right] \cup [-1,1] \cup \left[\frac{11}{6}, \frac{13}{6}\right],$$

since $I^+_{n,k} = I^-_{n,k} = \emptyset$, for all $k > 1$.

This example raises the question when $\omega_k \equiv 0$ for $|k| > 1$ can be guaranteed, i.e. the weights associated with $\mathbf{S}^{-1}$ are supported on the same set as those associated with $\mathbf{S}$. An answer is given in the following Corollary.

Corollary 10. *Let $\mathcal{G}(g,a,b)$ as in Theorem 9, with $b \in]0, \frac{2}{d-c+a}[$, then*

$$\mathbf{S}f = \sum_{k=-1}^{1} \omega_k \mathbf{T}_{-kb^{-1}} f,$$

and

$$\mathbf{S}^{-1}f = \sum_{k=-1}^{1} \widetilde{\omega_k} \mathbf{T}_{-kb^{-1}} f,$$

and $\omega_1 \equiv 0$ and $\widetilde{\omega_1} \equiv 0$ outside $\bigcup_{n\in\mathbb{Z}} I^-_{n,1}$, $\omega_{-1} \equiv 0$ and $\widetilde{\omega_{-1}} \equiv 0$ outside $\bigcup_{n\in\mathbb{Z}} I^+_{n,1}$.

Proof. To obtain the statement for $\mathbf{S}^{-1}$, apply Theorem 9(i) and simply check that $I^+_{n,k} = I^-_{n,k} = \emptyset$, for all $k > 1$. For $\mathbf{S}$, the statement follows by applying the conditions of Theorem 9 to the Walnut representation (4.9). □

Under the conditions above, it is reasonable to assume that it is possible to find a dual window with support in $[c,d]$. As can be shown by applying the duality condition (4.15), this is true in many cases.

Corollary 11. *Let $\mathcal{G}(g,a,b)$ with $g \in L^2(\mathbb{R})$ and $\operatorname{supp}(g) \subseteq [c,d]$ be a Gabor Bessel sequence as in Theorem 9 with $b \in]0, \frac{2}{d-c+a}[$.*

(a) Let $\mathcal{G}(h,a,b)$ a Gabor Bessel sequence with $h \in L^2(\mathbb{R})$, $\operatorname{supp}(h) \subseteq [c,d]$. $\mathcal{G}(g,a,b)$ and $\mathcal{G}(h,a,b)$ are dual frames if and only if the following hold:

- *For almost every $t \in [c, c+a[$:*

$$(h\overline{g})(t) + \mathbf{T}_{-a}(h\overline{g})(t) = b. \qquad \text{(a.i - Gabor)}$$

- *For almost every $t \in I^-_{0,1}$:*

$$(h\mathbf{T}_{-b^{-1}}\overline{g})(t) = 0. \qquad \text{(a.ii - Gabor)}$$

– *For almost every $t \in I^+_{0,1}$:*

$$(h\mathbf{T}_{b^{-1}}\overline{g})\,(t) = 0. \qquad \text{(a.iii - Gabor)}$$

(b) A Bessel sequence $\mathcal{G}(h,a,b)$ with $h \in L^2(\mathbb{R})$ and $\operatorname{supp}(h) \subseteq [c,d]$ *exists, such that the pair $\mathcal{G}(g,a,b)$, $\mathcal{G}(h,a,b)$ satisfy (a), if and only if there is some $A > 0$ such that the following hold:*

$$|g(t)| \geq A \quad \textit{or} \quad |\mathbf{T}_{-a}g(t)| \geq A \textit{ for a.e. } t \in [c, c+a[, \qquad \text{(b.i - Gabor)}$$

$$|\mathbf{T}_{-a}g| \geq A \textit{ a.e. on } \operatorname{supp}(\mathbf{T}_{-b^{-1}}g) \cap I^-_{0,1} \qquad \text{(b.ii - Gabor)}$$

and

$$|\mathbf{T}_{a}g| \geq A \textit{ a.e. on } \operatorname{supp}(\mathbf{T}_{b^{-1}}g) \cap I^+_{0,1}. \qquad \text{(b.iii - Gabor)}$$

Note that any real, continuous g with $0 < g(t) < 1$ for all $t \in (c,d)$ satisfies Corollary 11(b) for all $a < d-c$. Furthermore, under the assumptions $0 < g(t) < 1$ for all $t \in (c,d)$ any $h \in L^2(\mathbb{R})$ with $\operatorname{supp}(h) \subseteq [c,d]$, satisfying Corollary 11(a) must have its essential support contained in $[d - b^{-1}, c + b^{-1}]$. The result above, the restriction of Corollary 15 to the regular Gabor case, is little more than a reduction of the duality condition (4.15) to systems $\mathcal{G}(g,a,b)$ with $\operatorname{supp}(g) \subseteq [c,d]$ and $b \in]0, \frac{2}{d-c+a}[$. We see that pairs of dual frames with small support can be found if the painless case conditions are almost fulfilled.

A more general result, improving the support condition in Theorem 8, can be found in [35]. It cannot, however, easily be generalized to nonstationary Gabor frames.

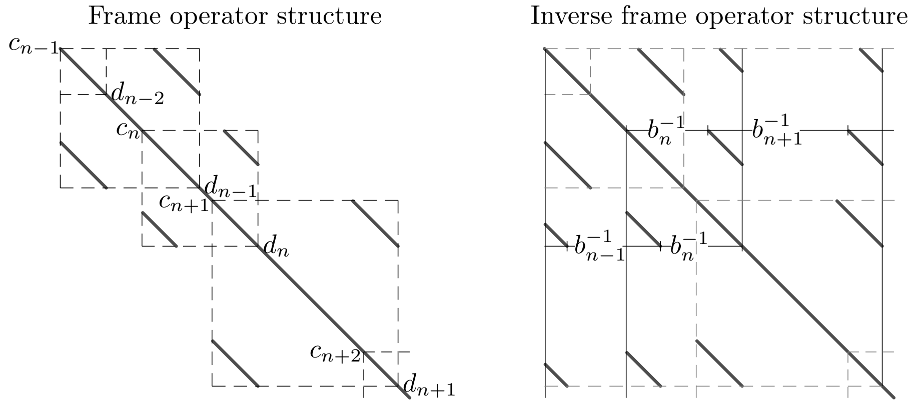
4.4 Results for nonstationary Gabor frames

We now generalize the notation used in Section 4.3 to the nonstationary setting and state our results in the general case. Since the modulation parameters b_n need not be equal anymore, we will work with

$$\begin{aligned} B^+_{n,k} &:= \sum_{j=0}^{k-1} b^{-1}_{n+j} \text{ and} \\ B^-_{n,k} &:= \sum_{j=0}^{k-1} b^{-1}_{n-j}, \ \forall\, n \in \mathbb{Z}, k \in \mathbb{N}. \end{aligned} \qquad (4.18)$$

Then, for the nonstationary Gabor system $\mathcal{G}(\mathbf{g},\mathbf{b})$, with $\operatorname{supp}(g_n) = [c_n, d_n]$ for all $n \in \mathbb{Z}$, we set

$$\begin{aligned} I_{n,0} &= [c_n, d_n], \\ I^+_{n,k} &= [c_{n-k+1} + B^-_{n,k}, d_n], \\ I^-_{n,k} &= [c_n, d_{n+k-1} - B^+_{n,k}] \end{aligned} \qquad (4.19)$$


Frame operator structure

Inverse frame operator structure

Figure 4.6: Section of a NSG frame operator and its inverse in the setting of Theorem 10 (schematic illustration). *Left*: The weights correspond to side-diagonal entries of a matrix, with ω_0 the main diagonal and $\omega_{n,\pm 1}$ located on side-diagonal $\pm b_n^{-1}$. Grey diagonal lines indicate non-zero entries in the side-diagonals/weights and we see that at most 3 entries in each row are non-zero. Dashed lines indicate the support of the individual window functions. *Right*: The inverse frame operator additionally possesses an irregularly spaced set of weights $\omega_{n,k}$ and $\omega_{n,-k}$ located on the side-diagonals $B^+_{n,k}$ and $-B^-_{n,k}$, respectively. That is, they are dependent on the non-uniform frequency steps b_n. Their non-zero entries are constrained by the support of the respective windows, indicated by horizontal and vertical lines. The parameter choice leads to shrinking support for weights located further from the main diagonal.

for n, k as before. Note that for $g_n = \mathbf{T}_{na} g$ and $b_n = b$ for all $n \in \mathbb{Z}$, these sets coincide with those in the previous section.

As before, the notational conventions above will be helpful in describing the structure inherent to the Walnut-like representation of inverse nonstationary Gabor frame operators. The conditions on $\mathcal{G}(\mathbf{g}, \mathbf{b})$ in Theorem 10 below imply $I^{\pm}_{n,k+1} \subseteq I^{\pm}_{n,k} \subseteq I_{n,0}$ and $I^+_{n,1} \cap I^-_{n,1} = \emptyset$ for all $n \in \mathbb{Z}, k \in \mathbb{N}$. Some intuition can be gained from likening the NSG frame operator and its inverse to a sparse, infinitesimal matrix with a structured set on non-zero side-diagonals that are in turn non-zero only on specific intervals. For an illustration, see Figure 4.6.

The following theorem details the structure of the inverse frame operator and the canonical dual frame $(\widetilde{g_{m,n}})_{m,n}$:

Theorem 10. *Let $\mathcal{G}(\mathbf{g}, \mathbf{b})$ be a nonstationary Gabor frame with $g_n \in L^2(\mathbb{R})$, $\operatorname{supp}(g_n) = [c_n, d_n]$, $c_n < d_n$ and $b_n \in]\frac{1}{d_n - c_n}, \infty[$ for all $n \in \mathbb{Z}$. If $\epsilon > 0$ exists such that $d_{n-1} \leq c_{n+1}$ and $b_n^{-1} \geq \max\{\frac{d_n - c_n}{2}, c_{n+1} - c_n, d_n - d_{n-1}\} + \epsilon$ for all $n \in \mathbb{Z}$, then the inverse frame operator $\mathbf{S}^{-1}_{\mathbf{g},\mathbf{b}} = \mathbf{S}^{-1}$ has a Walnut-like representation of the form*

(i)

$$\mathbf{S}^{-1}f = \omega_0 f + \sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}\left(\omega_{n,k}\mathbf{T}_{-B^+_{n,k}}f + \omega_{n,-k}\mathbf{T}_{B^-_{n,k}}f\right), \tag{4.20}$$

for all $f \in L^2(\mathbb{R})$. *Furthermore,* $\operatorname{supp}(\omega_{n,k}) \subseteq I^-_{n,k}$ *and* $\operatorname{supp}(\omega_{n,-k}) \subseteq I^+_{n,k}$ *for all* $n \in \mathbb{Z}, k \in \mathbb{N}$.

(ii) For any fixed $n \in \mathbb{N}$ *the elements* $\{\widetilde{g_{m,n}} = \mathbf{S}^{-1}g_{m,n}\}_{m\in\mathbb{Z}}$ *of the canonical dual frame satisfy*

$$\operatorname{supp}(\widetilde{g_{m,n}}) \subseteq I_{n,0} \cup \bigcup_{k\in\mathbb{N}}\left(I^-_{n-k,k} \cup I^+_{n+k,k}\right). \tag{4.21}$$

(iii) The elements $\widetilde{g_{m,n}}$, $m \neq 0$ *of the canonical dual frame can be derived from* $\widetilde{g_{0,n}}$ *by*

$$\widetilde{g_{m,n}} = \mathbf{M}_{mb_n}\Big(\widetilde{g_{0,n}}|_{I^{(0)}_n} + \sum_{k\in\mathbb{N}}\widetilde{g_{0,n}}|_{I^-_{n-k,k}}\exp(2\pi imb_nB^+_{n-k,k}) + \widetilde{g_{0,n}}|_{I^+_{n+k,k}}\exp(-2\pi imb_nB^-_{n+k,k})\Big), \tag{4.22}$$

where $\mathbf{M}_x$ *denotes modulation by* x.

(iv) For each $n \in \mathbb{Z}$, *there exists* $k_n \in \mathbb{N}$ *such that* $I^\pm_{n,k} = \emptyset$ *for all* $k \geq k_n$. *Furthermore, if a constant* $C < \infty$ *exists, such that* $\max_n(d_n - c_n - b_n^{-1}) \leq C$, *then* $I^\pm_{n,k} = \emptyset$ *for all* $n \in \mathbb{Z}$ *and* $k \geq C/\epsilon$.

Loosely speaking, the above theorem can be read as follows: Whenever a nonstationary Gabor system, comprised of compactly supported window functions with moderate overlap and sufficiently small modulation parameters, constitutes a frame, then

(i) The inverse frame operator possesses a Walnut-like representation with compactly supported off-diagonal weight functions.

(ii) Each element of the canonical dual frame is supported on a finite, disjoint union of compact intervals.

(iii) The canonical dual frame of $\mathcal{G}(\mathbf{g}, \mathbf{b})$ is "almost" a nonstationary Gabor system with the same modulation parameters. Some phase shifts may occur, though.

(iv) For fixed $n \in \mathbb{Z}$ only finitely many of the intervals $I^\pm_{n,k}$ are non-empty. If the window sizes behave nicely, there is a uniform bound on the number of non-empty sets, valid for all $n \in \mathbb{Z}$.

It is imminent from Theorem 10(ii) and (iii), that we can only guarantee the canonical dual system to be a NSG system with the same modulation parameters, if either $I^{\pm}_{n,k} = \emptyset$ for all n, k or $b_n = b$ for all $n \in \mathbb{Z}$. Other constructions that provide a canonical dual system with the same NSG structure are possible, but require some care in the choice of both window functions and parameters, see the example below. Intuitive constructions such as the choice of a uniform undersampling factor, i.e. $b_n^{-1} = \alpha(d_n - c_n)$ for some $\alpha < 1$, do not leave the structure of the original system intact.

Example 2. Let $\mathcal{G}(\mathbf{g}, \mathbf{b})$ be a NSG frame satisfying the conditions of Theorem 10. Let furthermore $\{n_j\}_{j\in\mathbb{Z}} \subseteq \mathbb{Z}$ with $n_j < n_{j+1}$ such that

(i) $b_{n_j}^{-1} \geq d_{n_j} - c_{n_j}$, for all $j \in \mathbb{Z}$ and

(ii) $b_n = b_k$ for all $n, k \in [n_j + 1, n_{j+1} - 1]$, $j \in \mathbb{Z}$.

Then it is easy to see that $I^+_{n_j,1} = \emptyset$, $I^-_{n_j,1} = \emptyset$ and the phase factors in Equation (4.22) equal 1, whenever $I^+_{n+k,k}$, respectively $I^-_{n-k,k}$, is non-empty. Consequently, the canonical dual frame is of the form $\mathcal{G}(\widetilde{\mathbf{g}}, \mathbf{b})$ for some sequence of functions $\widetilde{\mathbf{g}} = (\widetilde{g_n})_{n\in\mathbb{Z}}$.

To recover Theorem 9 from Theorem 10, combine (i) and (iv); note that $B^{\pm}_{n,k} = kb^{-1}$ for all n, k, take $\omega_k = \sum_n \omega_{n,k}$ for $k \neq 0$.

Before we prove Theorem 10, we collect some preliminary results about NSG systems $\mathcal{G}(\mathbf{g}, \mathbf{b})$ satisfying the conditions of the theorem. First, $d_{n-1} \leq c_{n+1}$ guarantees that, except possibly at endpoints, at most two adjacent windows g_n and g_{n+1} overlap. Moreover, $b_n^{-1} < d_n - c_n$ combined with $b_n^{-1} > c_{n+1} - c_n$ yields $c_{n+1} < c_n + b_n^{-1} < d_n$ and analogous, $c_n < d_n - b_n^{-1} < d_{n-1}$, implying that g_n, g_{n+1} and g_n, g_{n-1} overlap on a nontrivial interval. If $b_n^{-1} \geq d_n - c_n$, then $c_n \leq d_{n-1}$ is still a necessary condition for completeness of $\mathcal{G}(\mathbf{g}, \mathbf{b})$. Further, $b_n < \frac{2}{d_n - c_n}$ yields $[c_n, d_n] \cap ([c_n, d_n] + kb_n^{-1}) = \emptyset$ for $|k| \geq 2$.

By Corollary 12, we have $g_n \mathbf{T}_{kb_n^{-1}} \overline{g_n} \equiv 0$ for $|k| > 1$. The support of products of shifted weights $\mathbf{T}_\tau \omega_{n,k}$ will play a substantial role in proving Theorem 10. Indeed, they are the motivation behind the definition of the intervals $I^{\pm}_{n,k}$. Since a better understanding of their relations in the setting of Theorem 10 is crucial, we precede the proof with a lemma discussing these relations. The results are used, or at least considered, several times during the course of the proof of Theorem 10.

Lemma 5. *Under the conditions of Theorem 10, the following hold for all $n, m \in \mathbb{Z}, k, j \in \mathbb{N}$:*

(a) $I^-_{n,k+1} = I^-_{n,1} \cap (I^-_{n+1,k} - b_n^{-1})$, *with* $|I^-_{n,k+1}| < \min\{|I^-_{n,k}|, |I^-_{n+1,k}|\}$. *Analogous:* $I^+_{n,k+1} = I^+_{n,1} \cap (I^+_{n-1,k} + b_n^{-1})$, *with* $|I^+_{n,k+1}| < \min\{|I^+_{n,k}|, |I^+_{n-1,k}|\}$.

(b) $I_{n,k+1}^{\pm} \subsetneq I_{n,k}^{\pm}$.

(c) For $n \neq m$, $I_{n,k}^{+} \cap I_{m,j}^{+} = \emptyset$. *Analogous:* $I_{n,k}^{-} \cap I_{m,j}^{-} = \emptyset$.

(d) Whenever $I_{n,k}^{-} \cap I_{m,j}^{+} \neq \emptyset$, *it follows that* $m \in \{n-1, n-2\}$. *Furthermore* $I_{n,k}^{-} \cap I_{n-2,j}^{+} \neq \emptyset$ *implies* $c_n = d_{n-2}$.

(e) For $m \neq n$, $I_{n,k}^{-} + b_n^{-1} \cap I_{m,j}^{+} = \emptyset$ *and* $I_{n,k}^{-} \cap I_{m,j}^{+} - b_m^{-1} = \emptyset$.

(f) The following are equivalent:

(i) $I_{n-2,1}^{-} + b_{n-2}^{-1} \cap I_{n,1}^{-} \neq \emptyset$,
(ii) $I_{n-2,1}^{+} \cap I_{n,1}^{+} - b_n^{-1} \neq \emptyset$,
(iii) $c_n = d_{n-2}$.

Proof. (a) The conditions on $\mathbf{b}$ imply $c_n > c_{n+1} - b_n^{-1}$ and $d_{n+k+1} < d_{n+k} + b_{n+k+1}^{-1}$, proving the statement about the size of $I_{n,k+1}^{+}$. Further, by the same argument, $I_{n,1}^{-} \cap (I_{n+1,k}^{-} - b_n^{-1}) = [c_n, d_n - b_n^{-1}] \cap [c_{n+1} - b_n^{-1}, d_{n+k+1} - B_{n,k+1}^{+}] = [c_n, d_{n+k+1} - B_{n,k+1}^{+}] = I_{n,k+1}^{-} = I_{n,k}^{-} \cap (I_{n+1,k}^{-} - b_n^{-1})$. The proof for $I_{n,k+1}^{+}$ is analogue.

(b) Follows from (a).

(c) By (a), it is sufficient to show that $I_{n,1}^{+} \cap I_{m,1}^{+} = \emptyset$ and $I_{n,1}^{-} \cap I_{m,1}^{-} = \emptyset$ for $n \neq m$. Since the conditions on $\mathbf{b}$ guarantee $d_n - b_n^{-1} < d_{n-1} \leq c_{n+1} < c_n + b_n^{-1}$, (c) is immediate.

(d) Assume $m < n-1$ or $m > n$, then it is easy to see that $I_{n,1}^{-} \cap I_{m,1}^{+} = \{c_n\}$ if $m = n-2$, $c_n = d_{n-2}$ and otherwise $I_{n,1}^{-} \cap I_{m,1}^{+} = \emptyset$. For $m = n$ we get $I_{n,1}^{-} \cap I_{m,1}^{+} = \emptyset$ by the conditions on $\mathbf{b}$. The second part immediately follows from (b) with $I_{n,1}^{-} + b_n^{-1} = I_{n,1}^{+}$, the third part is analogue.

(e) and (f) follow from (b),(c), resp. (c),(d), together with $I_{n,1}^{-} + b_n^{-1} = I_{n,1}^{+}$. □

Lemma 5(f) and the second part of (d) are concerned with the case that $c_n = d_{n-2}$ for some $n \in \mathbb{Z}$. When we determine the Walnut-like representation of the inverse NSG frame operator in the following proof, the non-empty intersection between $I_{n,k}^{-}$ and $I_{n-2,j}^{+}$ leads to weights that are non-zero on a single point. The total number of such point weights is countable, hence their essential support is empty. Consequently the action of the corresponding operator equals that of the zero operator $\mathbf{0} : f \mapsto 0$. Thus, we will only point out those weights at the appropriate position in the proof, afterwards ignoring them entirely. However, when considering discrete NSG systems, these "point weights" influence the action of the corresponding operator and must be considered, somewhat complicating the argument. For more information regarding that case, see Remark 11 after the proof.

With Lemma 5 in place, we can now proceed to the proof of Theorem 10.

Proof of Theorem 10. (i): If $b_n^{-1} \geq d_n - c_n$ for all $n \in \mathbb{Z}$, then the frame operator $\mathbf{S}$ is diagonal and there is nothing to prove, cf. [10, Theorem 1] for more information. Otherwise, we make use of both the Neumann series (2.22) representation of $\mathbf{S}^{-1}$ and the Walnut representation of $\mathbf{S}$. Since we assume $\mathcal{G}(\mathbf{g}, \mathbf{b})$ to be a frame with frame bounds A, B, the Neumann series converges to the inverse frame operator and each of its elements defines a bounded, linear operator. The proof can roughly be structured into two parts. First, we use an induction argument to show that each element $\mathbf{N}^j$, $j \in \mathbb{N}_0$ with $\mathbf{N} := \mathbf{I} - 2\mathbf{S}/(A+B)$, of the Neumann sum $\frac{2}{A+B}\sum_{j=0}^{\infty} \mathbf{N}^j$ possesses a Walnut-like representation of the form

$$\mathbf{N}^j f = \omega_{j,0} f + \sum_{n\in\mathbb{Z}} \sum_{k=1}^{j} \left(\omega_{j,n,k} \mathbf{T}_{-B_{n,k}^+} f + \omega_{j,n,-k} \mathbf{T}_{B_{n,k}^-} f \right), \tag{4.23}$$

for all $f \in L^2(\mathbb{R})$, with $\operatorname{supp}(\omega_{j,n,k}) \subseteq I_{n,k}^-$ and $\operatorname{supp}(\omega_{j,n,-k}) \subseteq I_{n,k}^+$ for all $j \in \mathbb{N}_0$, $k \in \mathbb{N}$. Since $\mathbf{S}^{-1} = \frac{2}{A+B}\sum_{j\in\mathbb{N}_0} \mathbf{N}^j$, this establishes (a), except for the convergence of the desired ordering of the involved sums. The second part discusses this convergence of the sum of Walnut-like representations to the desired Walnut-like representation of $\mathbf{S}^{-1}$.

Similar to Equation (4.23) above, symmetric terms corresponding to $\omega_{j,n,k}$ for positive, respectively negative, k will appear throughout the proof. Both terms are generally handled in the same manner. Consequently, while we always provide both terms, studying one of the two closely should be sufficient to understand the argument of the proof. Since the identity operator $\mathbf{I}$ has a Walnut-like representation $\mathbf{I}f = \omega_{0,0} f$ with $\omega_{0,0} \equiv 1$, we can invoke the conditions on $\mathcal{G}(\mathbf{g}, \mathbf{b})$ to see that

$$\mathbf{N}^0 f = f \text{ and } \mathbf{N}^1 f = \omega_{1,0} f + \sum_{n\in\mathbb{Z}} \left(\omega_{1,n,1} \mathbf{T}_{-B_{n,1}^+} f + \omega_{1,n,-1} \mathbf{T}_{B_{n,1}^-} f \right)$$

for all $f \in L^2(\mathbb{R})$, with $\operatorname{supp}(\omega_{1,n,1}) \subseteq I_{n,1}^-$ and $\operatorname{supp}(\omega_{1,n,-1}) \subseteq I_{n,1}^+$. This proves (4.23) for $j \in \{0, 1\}$.

For the induction step, we show that (4.23) for j implies (4.23) for $j+1$ for all $j \in \mathbb{N}$. We define for $j \geq 1$

$$\mathbf{N}_D^j f = \omega_{j,0} f + \sum_{n\in\mathbb{Z}} \sum_{k=1}^{j-1} \left(\omega_{j,n,k} \mathbf{T}_{-B_{n,k}^+} f + \omega_{j,n,-k} \mathbf{T}_{B_{n,k}^-} f \right),$$

and

$$\mathbf{N}_R^j f = \sum_{n\in\mathbb{Z}} \left(\omega_{j,n,j}\mathbf{T}_{-B_{n,j}^+} f + \omega_{j,n,-j}\mathbf{T}_{B_{n,j}^-} f\right).$$

This allows us to write $\mathbf{N}^j$ as the sum of $\mathbf{N}_D^j$ and $\mathbf{N}_R^j$ and consequently

$$\mathbf{N}^{j+1} = \mathbf{N}^j\mathbf{N}^1 = \mathbf{N}_D^j\mathbf{N}^1 + \mathbf{N}_R^j\mathbf{N}^1. \tag{4.24}$$

Note that the only assumptions made on the form of $\mathbf{N}^j$ is the support of the weights, allowing us to use the induction assumption to show that $\mathbf{N}_D^j\mathbf{N}^1$ has a Walnut-like representation of the form

$$\mathbf{N}_D^j\mathbf{N}^1 f = \eta_{j,0} f + \sum_{n\in\mathbb{Z}}\sum_{k=1}^{j} \left(\eta_{j,n,k}\mathbf{T}_{-B_{n,k}^+} f + \eta_{j,n,-k}\mathbf{T}_{B_{n,k}^-} f\right), \tag{4.25}$$

for all $f \in L^2(\mathbb{R})$, with $\operatorname{supp}(\eta_{j,n,k}) \subseteq I_{n,k}^-$ and $\operatorname{supp}(\eta_{j,n,-k}) \subseteq I_{n,k}^+$ for all $k \in \{1,\dots,j\}$.

On the other hand, for all $f \in L^2(\mathbb{R})$, $\mathbf{N}_R^j\mathbf{N}^1 f$ can be written as

$$\mathbf{N}_R^j\mathbf{N}^1 f = \sum_{n\in\mathbb{Z}} \left(\omega_{j,n,j}\mathbf{T}_{-B_{n,j}^+}\mathbf{N}^1 f + \omega_{j,n,-j}\mathbf{T}_{B_{n,j}^-}\mathbf{N}^1 f\right).$$

By Lemma 5 and for all $n \in \mathbb{Z}, k = j$:

$$\operatorname{supp}(\omega_{j,n,k}\mathbf{T}_{-B_{n,j}^+}\omega_{1,0}) \subseteq I_{n,j}^-,$$

$$\operatorname{supp}(\omega_{j,n,k}\mathbf{T}_{-B_{n,j}^+}\omega_{1,\tilde{n},l}) \subseteq \begin{cases} I_{n,j}^- & l=-1,\ \tilde{n}=n+k+l, \\ I_{n,j+1}^- & l=1,\ \tilde{n}=n+k, \\ \emptyset & \text{else.} \end{cases}$$

an analogous for $k=-j$. In the final case above, we already include the situation where $\operatorname{supp}(\omega_{j,n,k}) \subseteq I_{n,k}^-$ and $\operatorname{supp}(\omega_{1,\tilde{n},l}) \subseteq I_{\tilde{n},1}^{\pm}$ such that $I_{n,k}^-$ and $I_{\tilde{n},1}^{\pm}$ intersect at a single point, confer the introductory remark preceeding the proof.

Order the appearing weights by the corresponding translate of f and take their sum to find that $\mathbf{N}_R^j\mathbf{N}^1 f$ can be written as

$$\mathbf{N}_R^j\mathbf{N}^1 f = \widetilde{\omega_{j+1,0}} f + \sum_{n\in\mathbb{Z}}\sum_{k=j-1}^{j+1} \left(\widetilde{\omega_{j+1,0,k}}\mathbf{T}_{-B_{n,k}^+} f + \widetilde{\omega_{j+1,0,-k}}\mathbf{T}_{B_{n,k}^-} f\right) \tag{4.26}$$

for all $f \in L^2(\mathbb{R})$, with $\operatorname{supp}(\widetilde{\omega_{j+1,n,k}}) \subseteq I_{n,k}^-$ and $\operatorname{supp}(\widetilde{\omega_{j+1,n,-k}}) \subseteq I_{n,k}^+$ for $k \in \{j-1,j,j+1\}$. Considering Equations (4.25) and (4.26), we conclude that (4.23) holds for $j+1$, completing the induction argument.

We combine the results so far, arriving, for all $f \in L^2(\mathbb{R})$, at

$$\begin{aligned}\mathbf{S}^{-1}f &= \frac{2}{A+B}\sum_{j\in\mathbb{N}_0}\mathbf{N}^j f \\ &= \frac{2}{A+B}\left(\sum_{j\in\mathbb{N}_0}\omega_{j,0}f + \sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}\Big(\omega_{j,n,k}\mathbf{T}_{-B^+_{n,k}}f + \omega_{j,n,-k}\mathbf{T}_{B^-_{n,k}}f\Big)\right). \qquad (4.27)\end{aligned}$$

Recall that $\|\mathbf{N}^1\|_{op} \le C$, with $C = \frac{B-A}{B+A} < 1$, where A and B are the optimal frame bounds for $\mathcal{G}(\mathbf{g},\mathbf{b})$. To conclude the proof, we want to interchange the sum over j with the sums over n and k. To achieve that, we show absolute convergence in operator norm. Observe that, by $d_n \le c_{n+2}$ we see that $\sum_{n\in\mathbb{Z}}\|f\mid_{[c_n,d_n]}\| \le 2\|f\|$ for all $f \in L^2(\mathbb{R})$ and, checking the support of $\omega_{j,n,k}$, we have

$$\omega_{j,n,k}\mathbf{T}_{-B^+_{n,k}}f = \omega_{j,n,k}\mathbf{T}_{-B^+_{n,k}}(f\mid_{[c_{n+k-1},d_{n+k-1}]}),\ \text{for } k>0$$

and

$$\omega_{j,n,k}\mathbf{T}_{B^-_{n,-k}}f = \omega_{j,n,k}\mathbf{T}_{B^-_{n,-k}}(f\mid_{[c_{n+k+1},d_{n+k+1}]}),\ \text{for } k<0.$$

By Lemma 4, we have $|\omega_{j,0}|, |\omega_{j,n,k}| \le C^j$ and consequently

$$\begin{aligned}&\left\|\sum_{j\in\mathbb{N}_0}|\omega_{j,0}|f + \sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}\Big(|\omega_{j,n,k}|\mathbf{T}_{-B^+_{n,k}}f + |\omega_{j,n,-k}|\mathbf{T}_{B^-_{n,k}}f\Big)\right\| \\ &\qquad\qquad \le \left\|\sum_{j\in\mathbb{N}_0}C^j|f| + \sum_{n\in\mathbb{Z}}\sum_{k=1}^{j}C^j f_{n,k}\right\|, \qquad (*)\end{aligned}$$

where

$$f_{n,k} = |\omega_{j,n,k}|\mathbf{T}_{-B^+_{n,k}}\left|f\mid_{[c_{n+k-1},d_{n+k-1}]}\right| + |\omega_{j,n,-k}|\mathbf{T}_{B^-_{n,k}}\left|f\mid_{[c_{n-k+1},d_{n-k+1}]}\right|.$$

In the next step, we separate the first term in the sum over j and for the remaining terms, interchange the sums over j with that over n. We also reorder the sums over j and k by the appearing restrictions of f.

$$\begin{aligned}(*) &= \left\|\frac{1}{1-C}|f| + \sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}f_{n,k}\sum_{j\ge|k|}C^j\right\| \\ &= \left\|\frac{1}{1-C}|f| + \frac{1}{1-C}\sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}C^k f_{n,k}\right\| \\ &\le \frac{1}{1-C}\|f\| + \frac{2}{1-C}\sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}C^k\left\|f\mid_{[c_n,d_n]}\right\| \\ &\le \frac{1}{1-C}\|f\| + \frac{4}{1-C}\|f\|\sum_{k\in\mathbb{N}}C^k = \frac{3C+1}{(1-C)^2}\|f\|.\end{aligned}$$

Hence, we can interchange the sums in (4.27) and find

$$\mathbf{S}^{-1}f = \frac{2}{A+B}\left(\omega_0 f + \sum_{n\in\mathbb{Z}}\sum_{k\in\mathbb{N}}\left(\omega_{n,k}\mathbf{T}_{-B^+_{n,k}}f + \omega_{n,-k}\mathbf{T}_{B^-_{n,k}}f\right)\right), \tag{4.28}$$

with $\omega_0 = \sum_j \omega_{j,0}$, $\omega_{n,k} = \sum_j \omega_{j,n,k}$ and $\|\omega_0\|_\infty \leq \frac{1}{1-C}$, $\|\omega_{n,k}\|_\infty \leq \frac{C^{|k|}}{1-C}$ for all $n \in \mathbb{Z}$, $k \in \mathbb{Z}\setminus\{0\}$. This concludes the proof of (i).

(ii): To determine the support of $\widetilde{g_{m,n}}$, we use (i) and collect the weights $\omega_{\tilde{n},k}$ such that

$$\operatorname{supp}(\mathbf{T}_{B^+_{n,k}}\omega_{\tilde{n},k}) \cap [c_n, d_n] \neq \emptyset \text{ for } k > 0,$$

respectively

$$\operatorname{supp}(\mathbf{T}_{-B^-_{n,-k}}\omega_{\tilde{n},k}) \cap [c_n, d_n] \neq \emptyset \text{ for } k < 0.$$

By checking the support properties, these can be found to be exactly the weights

$$\omega_{n-k+1,k},\ \omega_{n+k-1,-k}, \text{ for all } n \in \mathbb{Z},\ k \in \mathbb{N} \text{ and}$$
$$\omega_{n-k,k},\ \omega_{n+k,-k}, \text{ for all } n \in \mathbb{Z},\ k \in \mathbb{N}.$$

Consequently,

$$\begin{aligned}\widetilde{g_{m,n}} = \omega_0 g_{m,n} + \sum_{k\in\mathbb{N}}\Big(&\omega_{n-k+1,k}\mathbf{T}_{-B^+_{n-k+1,k}}g_{m,n} + \omega_{n-k,k}\mathbf{T}_{-B^+_{n-k,k}}g_{m,n} \\ &+\omega_{n+k-1,-k}\mathbf{T}_{B^-_{n+k-1,k}}g_{m,n} + \omega_{n+k,-k}\mathbf{T}_{B^-_{n+k,k}}g_{m,n}\Big)\end{aligned} \tag{4.29}$$

holds and

$$\operatorname{supp}(\widetilde{g_{m,n}}) = I_{n,0}\bigcup_{k\in\mathbb{N}}\left(I^-_{n-k+1,k}\cup I^-_{n-k,k}\cup I^+_{n+k-1,k}\cup I^+_{n+k,k}\right).$$

Complete the proof of (ii) by noting that $I^\pm_{n,k+1} \subseteq I^\pm_{n,k}$ and $I^\pm_{n,1} \subseteq I_{n,0}$, where we applied Lemma 5.

(iii): We know that $g_{m,n} = \mathbf{M}_{mb_n}g_n$ and $e^{2\pi imb_nt}$ is a b_n^{-1}-periodic function. Furthermore, $B^-_{n+k,k+1} = B^-_{n+k,k}+b_n^{-1}$ and analogous for $B^+_{n-k,k+1}$. Apply Equation (4.29) to $g_{0,n} = g_n$ and $g_{m,n} = g_ne^{2\pi imb_n\cdot}$ to confirm (iii).

(iv): Fix $n \in \mathbb{Z}$ and $k^-_n \in \mathbb{N}_0$ the smallest integer, such that $d_n - c_n - b_n^{-1} \leq k^-_n\epsilon$. Since $c_m + b_m^{-1} \geq c_{m+1}+\epsilon$, for all $m \leq n$, we see that $I^-_{n,\tilde{k}} = \emptyset$ for all $\tilde{k} \geq k^-_n$. Analogous, there exists $k^+_n \in \mathbb{N}_0$, such that $I^+_{n,\tilde{k}} = \emptyset$ for all $\tilde{k} \geq k^+_n$. Define $k_n := \max\{k^+_n, k^-_n\}$, then $I^\pm_{n,\tilde{k}} = \emptyset$ for all $\tilde{k} \geq k_n$. This proves the first part. To prove the second part note that $|I^+_{n,1}| = |I^-_{n,1}| \leq C$ follows from $d_n - c_n - b_n^{-1} \leq C$ for all $n \in \mathbb{Z}$. As before, $|I^\pm_{n,k+1}| \leq |I^\pm_{n,k}| - \epsilon$, concluding the proof. □

Remark 10. *Altogether, Theorem 10 tells us that, in the described case, the canonical dual frame of* $\mathcal{G}(\mathbf{g},\mathbf{b})$ *is not too different in structure from* $\mathcal{G}(\mathbf{g},\mathbf{b})$ *itself. From Theorem 10(iii) in particular, we see that a choice of constant* b_n *leads, as expected, to a canonical dual that is also a nonstationary Gabor frame. In cases where* b_n *varies in a systematic way, e.g. as powers of* 2*,* $\{\mathbf{S}^{-1}g_{n,m}\}_{n,m\in\mathbb{Z}}$ *can be interpreted as a nonstationary Gabor frame such that the functions* $\mathbf{S}^{-1}g_{n,m}$ *for fixed* $n\in\mathbb{Z}$ *are constructed from few prototypes by regular modulation with some step* $\tilde{b}_n$ *only dependent on* n*.*

Remark 11. *[Discrete NSG systems] Above, we have disregarded certain weights supported on isolated points, because isolated points are null sets in* $L^2(\mathbb{R})$*. This is not anymore true in* $\ell^2(\mathbb{Z})$ *or* $\mathbb{C}^L$ *with the usual point measure. Hence, these cases are worth some consideration. By considerations similar to those in the proof above, additional weights* $\rho^{\pm}_{n,k,l}$ *may appear for* $n\in\mathbb{Z}, k\in\mathbb{N}, l\in\mathbb{N}_0$*. All of them are supported on a single point, more explicitly* $\operatorname{supp}(\rho^{+}_{n,k,l}) = d_n - B^{+}_{n-k+1,k}$ *with corresponding translation operator* $\mathbf{T}_{-B^{+}_{n-k+1,k}-B^{+}_{n+2,l+1}}$ *and* $\operatorname{supp}(\rho^{-}_{n,k,l}) = d_n + B^{-}_{n+k+1,k}$ *with corresponding translation operator* $\mathbf{T}_{B^{-}_{n+k+1,k}+B^{-}_{n,l+1}}$*.*

However, also in the discrete setting, "smooth" window functions are preferred for their better time-frequency concentration. Therefore, assuming g_n *to be zero at the endpoints of its support is a weak restriction.*

With this caveat and the usual considerations in mind, the proof of Theorem 10 above can be directly applied to NSG systems in $\ell^2(\mathbb{Z})$*.*

Remark 12 (Frames for $\mathbb{C}^L$)**.** *For finite, discrete nonstationary Gabor transforms, Theorem 10 applies with essentially the obvious adjustments. Albeit, the circular nature of this setting introduces potential complications.*

To ensure that (iii) still holds, we must guarantee that the intervals $I^{(0)}_n$*,* $I^{+}_{n-k,k}$*,* $I^{-}_{n+l,l}$ *are disjoint. Assume the number of windows* g_n *to be* N*. Then if the nonstationary Gabor system in question satisfies* $g_n(c_n)g_{n-2}(c_n) = 0$ *for all* $n\in\{0,\dots,N-1\}$*, it is sufficient that* $\sum_{n=0}^{N-1} b_n^{-1} \geq L + \max_n |I^{+}_{n,1}|$*.*

Fast computation of the inverse frame operator can be implemented e.g. via a structured Gaussian elimination algorithm.

As in the regular Gabor case, it is reasonable to ask whether the weights of the inverse frame operator are supported on the same set as those of the original frame operator.

Corollary 12. *Let* $\mathcal{G}(\mathbf{g},\mathbf{b})$ *be as in Theorem 10, with* $b_n \in \left]0, \frac{b_{n-1}}{b_{n-1}(d_n-c_{n-1})-1}\right[$ *for all* $n\in\mathbb{Z}$*, then*

$$\mathbf{S}f = \omega_0 f + \sum_{n\in\mathbb{Z}} \left(\omega_{n,1}\mathbf{T}_{-B^{+}_{n,1}} f + \omega_{n,-1}\mathbf{T}_{B^{-}_{n,1}} f\right),$$

and

$$\mathbf{S}^{-1}f = \widetilde{\omega_0}f + \sum_{n\in\mathbb{Z}} \left(\widetilde{\omega_{n,1}}\mathbf{T}_{-B^+_{n,1}}f + \widetilde{\omega_{n,-1}}\mathbf{T}_{B^-_{n,1}}f \right).$$

Furthermore $\omega_{n,1}$ *and* $\widetilde{\omega_{n,1}}$ *are supported on a subset of* $I^-_{n,1}$*, while* $\omega_{n,-1}$ *and* $\widetilde{\omega_{n,-1}}$ *are supported on a subset of* $I^+_{n,1}$*.*

Proof. Apply Theorem 10(i) and simply check that $I^+_{n,k} = I^-_{n,k} = \emptyset$, for all $n \in \mathbb{Z}, k > 1$ to show the statement for $\mathbf{S}^{-1}$. For $\mathbf{S}$, apply the Walnut representation (4.9) and check the conditions of Theorem 10. □

So far, we have investigated the structure of inverse NSG frame operators and the canonical dual frames of NSG systems. We have seen that only few particular choices of $\mathcal{G}(\mathbf{g}, \mathbf{b})$ yield a canonical dual frame of the form $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{b})$. Yet, this does not exclude the existence of a dual system $\mathcal{G}(\mathbf{h}, \mathbf{b})$ per se. To further illuminate this problem, we will, under mild restrictions to the involved systems, deduce a necessary and sufficient condition for two NSG systems $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$ to constitute dual frames. As an illustrative example, we will apply the result in the setting of Corollary 12.

4.4.1 Towards a duality condition

The Walnut representation (4.9) is an efficient way to describe the action of a NSG frame-type operator. However, to determine duality of two NSG systems, it is beneficial to rearrange the summations ordered by the appearing translate of f. More precisely, define for any sequence $\mathbf{b} = (b_n)_n$ with $b_n \in \mathbb{R}^+$ the countable set $E_\mathbf{b}$ by

$$E_\mathbf{b} = \{\tau \in \mathbb{R} \ : \ \exists\ (m,n) \in \mathbb{Z}^2 \text{ s.t. } \tau = mb_n^{-1}\}. \tag{4.30}$$

Furthermore, to prevent pathologies, we introduce the following notion of "nice" nonstationary Gabor systems.

Definition 13. We call a nonstationary Gabor system $\mathcal{G}(\mathbf{g}, \mathbf{b})$ *well-behaved*, if either of the following holds:

(i) $E_\mathbf{b}$ is free of accumulation points.

(ii) For all $n \in \mathbb{Z}$, g_n is compactly supported on some interval $[c_n, d_n]$ and $O_n := \{l \in \mathbb{Z} \ : \ c_l < d_n \text{ and } d_l > c_n\}$ is finite.

The flexibility gained by the way a NSG system is defined allows the construction of a multitude of pathological cases that are generally not interesting for practical purposes. Note that the functions g_n and h_n are usually desired to be well concentrated in time and frequency. Further, they should be evenly distributed over time. Consequently, only finitely many compactly supported windows overlapping is a rather weak restriction. On the other hand $\{b_n^{-1} : n \in \mathbb{Z}\}$ being δ-separated, i.e. either $b_n^{-1} = b_l^{-1}$ or $|b_n^{-1} - b_l^{-1}| \geq \delta$ for all $n, l \in \mathbb{Z}$ is enough to guarantee $E_{\mathbf{b}}$ being free of accumulation points.

We can now formulate an alternative version of the Walnut representation (4.9), valid on a dense subspace of $L^2(\mathbb{R})$.

Corollary 13. *Let $\mathcal{G}(\mathbf{g},\mathbf{b})$ and $\mathcal{G}(\mathbf{h},\mathbf{b})$ be well-behaved nonstationary Gabor Bessel sequences with $b_n \in \mathbb{R}^+$ and $g_n, h_n \in L^2(\mathbb{R})$, for all $n \in \mathbb{Z}$. Then, for all $f \in L^2(\mathbb{R})$ with compact support,*

$$\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} f = \sum_{\tau \in E_{\mathbf{b}}} \omega_\tau \mathbf{T}_\tau f, \tag{4.31}$$

with

$$\omega_0 = \sum_{n\in\mathbb{Z}} b_n^{-1} h_n \overline{g_n} \text{ and } \omega_\tau = \sum_{\substack{(m,n)\in\mathbb{Z}^2 \\ mb_n^{-1}=\tau}} b_n^{-1} h_n \mathbf{T}_\tau \overline{g_n} \text{ for } \tau \neq 0. \tag{4.32}$$

Moreover, the sum in (4.31) is absolutely convergent. Consequently, the extension to $L^2(\mathbb{R})$ of the bounded, linear operator defined by the right-hand side of (4.31) equals $\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}}$.

Proof. By Proposition 13, $\sum_{n\in\mathbb{Z}} b_n^{-1} |h_n \mathbf{T}_\tau \overline{g_n}| \leq B$ almost everywhere, for any $\tau \in \mathbb{R}$. Now let I be any finite interval such that $\operatorname{supp}(f) + \operatorname{supp}(f) \subseteq I$. If $\mathcal{G}(\mathbf{g},\mathbf{b})$, $\mathcal{G}(\mathbf{h},\mathbf{b})$ are well-behaved in the sense of Definition 13(i), then $E_{\mathbf{b}} \cap I$ is a finite set and

$$\begin{aligned}
&\Big| \sum_{\tau\in E_{\mathbf{b}}} \sum_{\substack{(n,k)\in\mathbb{Z}^2 \\ \tau=kb_n^{-1}}} b_n^{-1} (h_n \mathbf{T}_\tau \overline{g_n}) \mathbf{T}_\tau f \Big| \\
&\leq \sum_{\tau\in E_{\mathbf{b}}\cap I} \sum_{\substack{(n,k)\in\mathbb{Z}^2 \\ \tau=kb_n^{-1}}} b_n^{-1} |h_n \mathbf{T}_\tau \overline{g_n}| |\mathbf{T}_\tau f| \\
&\leq B \sum_{\tau\in E_{\mathbf{b}}\cap I} |\mathbf{T}_\tau f| < \infty \text{ a.e. on } I,
\end{aligned} \tag{4.33}$$

with absolute convergence. If on the other hand, $\mathcal{G}(\mathbf{g},\mathbf{b})$, $\mathcal{G}(\mathbf{h},\mathbf{b})$ are well-behaved in the sense of Definition 13(ii), then the sum over $E_{\mathbf{b}}$ is locally finite. Thus, by the Walnut representation (4.9) of $\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}}$:

$$\sum_{\tau\in E_{\mathbf{b}}} \omega_\tau \mathbf{T}_\tau f = \sum_{n,k\in\mathbb{Z}} b_n^{-1} (h_n \mathbf{T}_{kb_n^{-1}} \overline{g_n}) \mathbf{T}_{kb_n^{-1}} f = \mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} f, \ \forall\, f \in L^2(\mathbb{R}). \tag{4.34}$$

Since $\sum_{\tau\in E_{\mathbf{b}}} \omega_\tau \mathbf{T}_\tau = \mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}}$ on a dense subspace of $L^2(\mathbb{R})$, the extension of $\sum_{\tau\in E_{\mathbf{b}}} \omega_\tau \mathbf{T}_\tau$ to $L^2(\mathbb{R})$ equals $\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}}$. □

For well behaved systems it is easy to see that $\omega_0 \equiv 1$ and $\omega_\tau \equiv 0$ for $\tau \neq 0$ is a sufficient condition for $\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} f = f$ and thus for $\mathcal{G}(\mathbf{g},\mathbf{b})$ and $\mathcal{G}(\mathbf{h},\mathbf{b})$ to be dual frames. The following result shows that it is necessary as well.

Theorem 11. *Let $\mathcal{G}(\mathbf{g},\mathbf{b})$, $\mathcal{G}(\mathbf{h},\mathbf{b})$ be well-behaved nonstationary Gabor Bessel sequences with $g_n, h_n \in L^2(\mathbb{R})$, $b_n \in \mathbb{R}^+$. Then*

$$\sum_{n\in\mathbb{Z}} b_n^{-1} h_n \overline{g_n} \equiv 1 \textit{ and } \sum_{\substack{(m,n)\in\mathbb{Z}^2\\ mb_n^{-1}=\tau}} b_n^{-1} h_n \mathbf{T}_\tau \overline{g_n} \equiv 0 \textit{ for } \tau \neq 0 \tag{4.35}$$

is equivalent to duality in the sense that

$$f = \sum_{m,n\in\mathbb{Z}} \langle f, g_{m,n}\rangle h_{m,n} \textit{ for all } f \in L^2(\mathbb{R}). \tag{4.36}$$

Proof. Assume Equation (4.35) holds. Since $\mathcal{G}(\mathbf{g},\mathbf{b})$, $\mathcal{G}(\mathbf{h},\mathbf{b})$ are well-behaved, the alternate Walnut representation (4.31) of $\mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}}$ can be evoked. We see that for all compactly supported f,

$$\begin{aligned}\sum_{m,n\in\mathbb{Z}} \langle f, g_{m,n}\rangle h_{m,n} &= \mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} f\\ &= \sum_{\tau\in E_{\mathbf{b}}} \omega_\tau \mathbf{T}_\tau f\\ &= \omega_0 \mathbf{T}_0 f = f.\end{aligned}$$

Therefore, Equation (4.36) holds for all compactly supported $f \in L^2(\mathbb{R})$ and by density for all $f \in L^2(\mathbb{R})$, proving the first inference. We prove the converse inference by contradiction, assuming (4.35) to be violated, then provide a counterexample to (4.36). Let $\mathcal{G}(\mathbf{g},\mathbf{b})$, $\mathcal{G}(\mathbf{h},\mathbf{b})$ be well-behaved in the sense of Definition 13(i), i.e. if $\omega_\tau \neq 0$ for some $\tau \in \mathbb{R}\setminus\{0\}$, we can choose $\delta > 0$ and $l \in \mathbb{Z}$, such that $E_{\mathbf{b}} \cap B_{2\delta}(\tau) = \{\tau\}$ and $\omega_\tau|_{B_\delta(2l\delta)} \neq 0$. Let $f = \chi_{B_\delta(2l\delta)-\tau}$, then

$$0 \equiv f\ |_{B_\delta(2l\delta)} \neq \mathbf{S}_{\mathbf{g},\mathbf{h},\mathbf{b}} f\ |_{B_\delta(2l\delta)} = \omega_\tau|_{B_\delta(2l\delta)},$$

proving that $\omega_\tau \equiv 0$ for all $\tau \in \mathbb{R}\setminus\{0\}$ is necessary. But then $\omega_0 \neq 1$ contradicting (4.36) can easily be seen. If instead $\mathcal{G}(\mathbf{g},\mathbf{b})$, $\mathcal{G}(\mathbf{h},\mathbf{b})$ are well-behaved in the sense of Definition 13(ii), note that

$$E_{\mathbf{b},n} = \{\tau \in \mathbb{R}\ :\ \exists\ m \in \mathbb{Z}, l \in O_n \text{ s.t. } \tau = mb_l^{-1}\} \tag{4.37}$$

is free of accumulation points and apply the reasoning above. □

Note that duality of a pair of Bessel sequences as in (4.36) implies the frame property for both involved Bessel sequences.

Corollary 14. *Let $\mathcal{G}(\mathbf{g},\mathbf{b})$ be a well-behaved nonstationary Gabor Bessel sequence with $g_n \in L^2(\mathbb{R})$, $b_n \in \mathbb{R}^+$. Then $\mathcal{G}(\mathbf{g},\mathbf{b})$ is a normalized tight frame, i.e. a frame with bounds $A = B = 1$, if and only if*

$$\sum_{n\in\mathbb{Z}} b_n^{-1}|g_n|^2 \equiv 1 \text{ and } \sum_{\substack{(m,n)\in\mathbb{Z}^2\\ mb_n^{-1}=\tau}} b_n^{-1} g_n \mathbf{T}_\tau \overline{g_n} \equiv 0 \text{ for } \tau \neq 0. \tag{4.38}$$

Remark 13. *In the context of GSI systems (generalized shift-invariant systems), equivalent to NSG systems via the application of a Fourier transform, the results presented in Theorem 11 and Corollary 14 above can be found in [89, Thm 9.1, Thm 2.1] and [137, Prop 3.44, Cor 1.16], proven under different technical conditions. However, they have not previously been connected to nonstationary Gabor systems and we provide herein very simple proofs, relying only on elementary mathematical concepts.*

Remark 14. *For systems with uniform $\mathbf{b}$, i.e. $b_n = b$, the duality conditions above reduce to the well-known conditions for Gabor frames [133, 136] or more generally, shift-invariant frames [99]. In both classical cases, the canonical dual frame inherits the structure of the original frame and thus the duality conditions are guaranteed to have a solution. This is not true for NSG systems in general. Indeed, we expect that for many choices of a NSG frame $\mathcal{G}(\mathbf{g},\mathbf{b})$, there is no system $\mathcal{G}(\mathbf{h},\mathbf{b})$ satisfying* (4.35).

Remark 15. *The restriction to well-behaved NSG systems in Theorem 11 prevents us from recovering the equivalence of the duality conditions to the frame property for Wavelet systems, proven by Chui and Shi in [41]. However, the restriction to well-behaved systems allows for a straightforward proof, once all the ingredients are in place. Also note that duality of Wavelet systems is equivalent to the duality of certain* quasi-affine systems*, as introduced by Ron and Shen [134, 135, 137]. In contrast to Wavelet systems, quasi-affine systems are well-behaved NSG systems, and therefore satisfy the restrictions of Theorem 11.*

Given a specific setup of $\mathbf{g}$ and $\mathbf{b}$, the duality conditions above may prove useful to determine the existence of a dual system that shares the modulation parameters $\mathbf{b}$. This is particularly interesting from an algorithmic point of view, since analysis and synthesis can be realized efficiently for NSG systems, but not for general frames. Here, we consider the setting of Corollary 12 and show that dual pairs of NSG frames with compactly supported generators exist. We obtain the following result.

Corollary 15. *Let $\mathcal{G}(\mathbf{g}, \mathbf{b})$ be a nonstationary Gabor Bessel sequence as in Theorem 10 with $g_n \in L^2(\mathbb{R})$, $b_n \in]0, \frac{b_{n-1}}{b_{n-1}(d_n - c_{n-1}) - 1}[$ and $c_n \leq d_{n-1}$ for all $n \in \mathbb{Z}$.*

(a) Let $\mathcal{G}(h_n, b_n)$ a nonstationary Gabor Bessel sequence with $h_n \in L^2(\mathbb{R})$, $\operatorname{supp}(h_n) \subseteq [c_n, d_n]$. $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$ are dual frames if and only if the following hold for all $n \in \mathbb{Z}$:

- *For almost every $t \in [c_n, c_{n+1}[$:*

$$b_n^{-1} \left(h_n \overline{g_n}\right)(t) + b_{n-1}^{-1} \left(h_{n-1} \overline{g_{n-1}}\right)(t) = 1. \tag{a.i}$$

- *For almost every $t \in I_{n,1}^-$:*

$$\left(h_n \mathbf{T}_{-b_n^{-1}} \overline{g_n}\right)(t) = 0. \tag{a.ii}$$

- *For almost every $t \in I_{n,1}^+$:*

$$\left(h_n \mathbf{T}_{b_n^{-1}} \overline{g_n}\right)(t) = 0. \tag{a.iii}$$

(b) A Bessel sequence $\mathcal{G}(\mathbf{h}, \mathbf{b})$ with $h_n \in L^2(\mathbb{R})$ and $\operatorname{supp}(h_n) \subseteq [c_n, d_n]$ for all $n \in \mathbb{Z}$ exists, such that the pair $\mathcal{G}(\mathbf{g}, \mathbf{b})$, $\mathcal{G}(\mathbf{h}, \mathbf{b})$ satisfy (a), if and only if there is some $A > 0$ such that the following hold for all $n \in \mathbb{Z}$:

$$b_n^{-1/2} |g_n(t)| \geq A \quad \textit{or} \quad b_{n-1}^{-1/2} |g_{n-1}(t)| \geq A \textit{ for a.e. } t \in [c_n, c_{n+1}[, \tag{b.i}$$

$$b_{n-1}^{-1/2} |g_{n-1}| \geq A \textit{ a.e. on } \operatorname{supp}(\mathbf{T}_{-b_n^{-1}} g_n) \cap I_{n,1}^- \tag{b.ii}$$

and

$$b_{n+1}^{-1/2} |g_{n+1}| \geq A \textit{ a.e. on } \operatorname{supp}(\mathbf{T}_{b_n^{-1}} g_n) \cap I_{n,1}^+. \tag{b.iii}$$

Proof. The systems $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{h}, \mathbf{b})$ are well-behaved in the sense of Definition 13(ii). Thus they form a pair of dual nonstationary Gabor frames if and only if Equation (4.35) is satisfied. Invoking the support conditions on the systems, we get

$$b_n^{-1} h_n \overline{g_n} + b_{n-1}^{-1} h_{n-1} \overline{g_{n-1}} = 1 \text{ a.e. on } [c_n, c_{n+1}[,$$

$$h_n \mathbf{T}_{-b_n^{-1}} \overline{g_n} = 0 \text{ a.e. on } I_{n,1}^+$$

and

$$h_n \mathbf{T}_{b_n^{-1}} \overline{g_n} = 0 \text{ a.e. on } I_{n,1}^-,$$

for all $n \in \mathbb{Z}$, concluding the proof of (a). We first prove that (b.i) to (b.iii) are sufficient by constructing a dual Bessel sequence $\mathcal{G}(\mathbf{h}, \mathbf{b})$ satisfying the support constraints. Let for all $n \in \mathbb{Z}$, J_n^0 be the largest open subset of $[c_n, c_{n+1}[$ such that $b_n^{-1/2} |g_n| \geq A$ almost everywhere on J_n^0 and $J_n^1 = [c_{n+1}, d_n] \setminus J_{n+1}^0$.

Furthermore let us denote, for all $n \in \mathbb{Z}$, $J_n^- = I_{n,1}^- \cap \operatorname{supp}(T_{-b_n^{-1}} g_n)$ and $J_n^+ = I_{n-1,1}^+ \cap \operatorname{supp}(T_{b_{n-1}^{-1}} g_{n-1})$. Then

$$h_n := \begin{cases} b_n/\overline{g_n}, & \text{on } (J_n^0 \cup J_n^1) \setminus (J_n^+ \cup J_n^-)\,, \\ 0, & \text{else }, \end{cases}$$

is well-defined almost everywhere for all $n \in \mathbb{Z}$. With this choice, it is easy to see that the conditions (a.i) to (a.iii) are satisfied. Furthermore, $h_n(t) < 2\sqrt{b_n}/A$ almost everywhere. Thus $h_n \in L^\infty(\mathbb{R})$ and is compactly supported, in particular $h_n \in L^2(\mathbb{R}) \cap W(L^\infty, \ell^1)$. We see that $\sum_n b_n^{-1} |h_n|^2 \leq 2/A^2$. Invoke the Walnut representation and apply the proof of Proposition 13 to see that $\mathcal{G}(\mathbf{h}, \mathbf{b})$ is a Bessel sequence.

For the converse, we assume either of (b.i) to (b.iii) to be violated. Note that h_n is uniquely determined almost everywhere on $J_{n+1}^- \cup J_{n-1}^+$. If (b.ii) or (b.iii) is violated, we can for every $\epsilon > 0$ find $n \in \mathbb{Z}$, such that $b_n^{-1/2}|g_n| < \epsilon$ and consequently $b_n^{-1/2}|h_n| > \epsilon^{-1}$ almost everywhere on a subset $M \subseteq J_{n+1}^- \cup J_{n-1}^+$ of positive measure. Therefore, $\sum_n b_n^{-1}|h_n|^2 > \epsilon^{-2}$ on a set of positive measure, contradicting the Bessel condition by Proposition 13. If on the other hand (b.i) is violated, then we can for every $\epsilon > 0$ find $n \in \mathbb{Z}$, such that $b_n^{-1/2}|g_n| < \epsilon$ and $b_{n-1}^{-1/2}|g_{n-1}| < \epsilon$ almost everywhere on a subset $M \subseteq [c_n, c_{n+1}[$ of positive measure. Assume (a.i) to be satisfied, i.e.

$$1 = |b_n^{-1} h_n \overline{g_n} + b_{n-1}^{-1} h_{n-1} \overline{g_{n-1}}| \leq \epsilon \left(b_n^{-1/2} |h_n| + b_{n-1}^{-1/2} |h_{n-1}| \right) \text{ a.e. on } M.$$

Then, almost everywhere on M, either $b_n^{-1/2}|h_n| > 1/2\epsilon$ or $b_{n-1}^{-1/2}|h_{n-1}| > 1/2\epsilon$, contradicting the Bessel condition by Proposition 13. □

Remark 16. *The proof shows that, given g_n, g_{n-1}, g_{n+1}, h_n is uniquely determined on $I_{n,1}^- \cup]d_{n-1}, c_{n+1}[\cup I_{n,1}^+$, except for a zero set. Therefore, as Christensen, Kim and Kim have observed in the regular case [35], the equation system* (4.4.1) *to* (4.4.1) *is not solvable in general, if $I_{n,1}^- \cap I_{n-1,1}^+ \neq \emptyset$ and* $\operatorname{supp}(h_n) \subseteq \operatorname{supp}(g_n)$, *for all $n \in \mathbb{Z}$. In the classical Gabor case, because $b_n = b$ for all $n \in \mathbb{Z}$, an appropriate increase of the size of* $\operatorname{supp}(h_n)$ *does the trick. We expect that this can be generalized to NSG systems with uniform* $\mathbf{b}$. *In the general case however, non-uniformity of b_n significantly complicates matters and further work is required to determine the solvability of* (4.4.1) *to* (4.4.1) *even without support constraints on the h_n, i.e. whether any NSG system $\mathcal{G}(\mathbf{h}, \mathbf{b})$, dual to $\mathcal{G}(\mathbf{g}, \mathbf{b})$, can exist.*

To recover Corollary 11, replace g_n by $T_{na}g$ and b_n by b and observe the a-periodicity of Gabor systems. Note that $b < \frac{b}{b(d-c+a)-1}$ is equivalent to $b < \frac{2}{d-c+a}$.

4.5 Conclusion and perspectives

Our approach enables the construction of frames with flexible evolution of time-frequency resolution over time or frequency. The resulting frames are well suited for applications as they can be implemented using fast algorithms, at a computational cost close to standard Gabor frames.

We presented several results on the structure of nonstationary Gabor systems with low redundancy in time and moderate redundancy in frequency, demonstrating that such systems, if invertible, possess an inverse frame operator with a distinct structure not too different from that of the original frame operator. While the canonical dual frame will be of nonstationary Gabor type only if the modulation parameters b_n are chosen uniformly, we have given a simple condition on the existence of a dual nonstationary Gabor frame satisfying the exact same support conditions. Furthermore, such a frame can be constructed by solving a simple set of equations. Reduction of our results to the case of classical Gabor systems shows that the canonical dual frame satisfies a special support condition, for which Christensen, Kim and Kim have recently shown the existence of a dual frame satisfying it. Under stronger restrictions on the redundancy of the Gabor system, we showed that this support condition can be improved to coincide with the original support.

Further, we have generalized the duality conditions for Gabor systems to the setting of well-behaved NSG systems, providing a tool for investigating the existence of dual pairs of nonstationary Gabor systems.

Future work includes the investigation of the inverse frame operator for more general NSG systems, allowing for higher overlap and/or coarser frequency sampling, although numerical experiments have shown that low redundancy systems with high overlap possess a highly non-sparse inverse frame operator. Moreover, harnessing the results in this manuscript to provide fast implementations for the inversion of certain discrete nonstationary Gabor frames, extending the flexibility of such systems in applications is planned.

Chapter 5

Discrete, finite nonstationary Gabor systems and their implementation

This section compiles results from joint work with M. Dörfler, T. Grill and G.A. Velasco [164], [92], implementation examples from joint work with P. Balazs, M. Dörfler, F. Jaillet and G.A. Velasco [10], partial results from joint work with T.Necciari, P.Balazs and P.L. Søndergaard [122] and preliminary results from unpublished joint work with C. Wiesmeyr.

In the sense of reproducible research, [163], the algorithms and scripts to reproduce the results in this chapter are available at the respective webpages

- `http://univie.ac.at/nonstatgab/` and
- `http://www.kfs.oeaw.ac.at/ICASSP2013_ERBlets`.

Please note that the latter requires the free Large Time Frequency Analysis Toolbox (LTFAT) [2, 145, 146] and Auditory Modeling Toolbox (AMToolbox) [1] for MATLAB/OCTAVE.

5.1 Discrete, finite frames

For the practical implementation, the theory of frames (see Section 2.2 for a brief introduction or [33] for a more comprehensive treatment) may be developed in a finite discrete setting using the Hilbert space $\mathbb{C}^L$. In this section, we shortly introduce frames for $\mathbb{C}^L$, i.e. vector spaces of finite, discrete signals, understood as functions f, g on $\mathbb{C}^L$. The general framework is mostly analogous to the $L^2(\mathbb{R})$ setting, save for the fact that the involved operators can now be considered as matrices. Since we consider time-frequency frames only, we introduce frames over $\mathbb{C}^L$ with a two parameter index set. Consider a collection of atoms $\Phi = \{\varphi_{n,m} \in$

$\mathbb{C}^L\}$ with $(n,m) \in \Lambda_N \times \Lambda_M$ for finite index sets Λ_N, Λ_M with cardinality N and M, respectively. For simplicity, we will assume $\Lambda_N = \{0,\dots,N-1\}$ and $\Lambda_M = \{0,\dots,M-1\}$. In this setup, the subscripts n,m allude to the position of $\varphi_{n,m}$ in time and frequency. For now, we restrict to uniform index sets, while subsequent sections will allow one index set to depend on the position in the other, Λ_{M_n} may depend on $n \in \Lambda_N$.

Let $P = NM$ be the number of frame elements. Then the analysis operator C_Φ is a $P \times L$ matrix, containing the complex conjugate of the frame elements in its rows. Canonically, the rows are in "frequency first" ordering, i.e.

$$C_\Phi[m+nM, l] = \overline{\varphi_{n,m}}[l].$$

The synthesis operator equals the adjoint (conjugate transpose) $D_\Phi = C_\Phi^*$ of the analysis operator and is given by

$$D_\Phi[l, m+nM] = \varphi_{n,m}[l].$$

Consequently, the frame operator $\mathbf{S} = D_\Phi C_\Phi$ is a $L \times L$ matrix acting on $f \in \mathbb{C}^L$ by

$$\mathbf{S}f = \sum_{n,m} \langle f, \varphi_{n,m}\rangle \varphi_{n,m}. \tag{5.1}$$

If the linear operator $\mathbf{S}$ is invertible on $\mathbb{C}^L$, then the set of functions $\{\varphi_{n,m}\}_{(n,m)\in\Lambda_N\times\Lambda_M}$, is a *frame*[1]. In this case, we may define a *dual frame* by

$$\widetilde{\varphi_{n,m}} = \mathbf{S}^{-1}\varphi_{n,m} \tag{5.2}$$

and reconstruction from the coefficients $c_{n,m} = \langle f, \varphi_{n,m}\rangle$ is straightforward:

$$\begin{aligned} f = \mathbf{S}^{-1}\mathbf{S}f &= \sum_{n,m} \langle f, \varphi_{n,m}\rangle \mathbf{S}^{-1}\varphi_{n,m} = \sum_{n,m} c_{n,m}\widetilde{\varphi_{n,m}} \\ = \mathbf{S}\mathbf{S}^{-1}f &= \sum_{n,m} \langle f, \mathbf{S}^{-1}\varphi_{n,m}\rangle \varphi_{n,m} = \sum_{n,m} \langle f, \widetilde{\varphi_{n,m}}\rangle \varphi_{n,m}. \end{aligned}$$

Note that an overcomplete frame ($L < P$) admits an infinite number of dual frames. The particular dual frame obtained via the application of the inverse frame operator to the frame elements is called the *canonical* dual frame, see Section 2.2. The optimal frame bounds of Φ are given by

$$A_0 = \inf_{f\in\mathbb{C}^L} \frac{\|C_\Phi f\|^2}{\|f\|^2} \quad \text{and} \quad B_0 = \sup_{f\in\mathbb{C}^L} \frac{\|C_\Phi f\|^2}{\|f\|^2}.$$

[1] Note that, if $\{\varphi_{n,m}, (n,m) \in \Lambda_N \times \Lambda_M\}$ is an orthonormal basis, then $\mathbf{S}$ is the identity operator.

Also recall discrete Gabor frames (Section 2.3), for which the elements $g_{n,m}$ are obtained from a single window g by time- and frequency-shifts along a lattice. Fixing a time-shift parameter a and a frequency-shift parameter b, with $L/a, L/b \in \mathbb{N}$, we call the collection of atoms $\mathcal{G}(g,a,b) = \{g_{n,m} = \mathbf{M}_{kb}\mathbf{T}_{na}g\}_{(n,m)\in\Lambda_N\times\Lambda_M}$, with $\Lambda_N \times \Lambda_M = \mathbb{Z}_{L/a} \times \mathbb{Z}_{L/b}$, a *Gabor system.* If $\mathcal{G}$ is a frame, it is called a Gabor frame. For Gabor frames, the frame coefficients are given by samples of the short-time Fourier transform of f with respect to the window g:

$$\begin{aligned} c_{n,m} = c[n,m] = \langle f, g_{n,m}\rangle &= \langle f, \mathbf{M}_{kb}\mathbf{T}_{na}g\rangle \\ &= \sum_{l=0}^{L-1} f[l]\overline{g[l-na]}e^{-2\pi i l\cdot kb/L}. \end{aligned} \tag{5.3}$$

In a general setting, the inversion of the operator $\mathbf{S}$ poses a problem in numerical realization of frame analysis. However, for Gabor frames, it was shown in [48], that under certain conditions, usually fulfilled in practical applications, $\mathbf{S}$ is diagonal, and a dual frame can be calculated easily, see also Section 4.1.3. This situation of *painless nonorthogonal expansions* can also be formulated for (nonstationary) Gabor frames over $\mathbb{C}^L$ as discussed in the following section. Generally, NSG frames with small frame bound ratio B_0/A_0 can be inverted efficiently using iterative methods, see Section 5.5.1.

5.2 Discrete nonstationary Gabor transforms

Since the development of the painless case is largely straightforward from simple matrix multiplication, we only state the result. Translating the remaining results from the previous chapter requires only the usual changes and minor modification. The periodic structure of signals on $\mathbb{C}^L$ might sometimes lead to additional restrictions and heavy notation, see Remark 12. Since the principal idea remains the same as over $\mathbb{R}$, we omit these results. From here on we use the notation $\Lambda_k := \{0,\dots,k-1\}$ for $k \in \mathbb{N}$.

Definition 14. Let $\{g_n\}_{n\in\Lambda_N}$ be a set of functions in $\mathbb{C}^L$ and $\{M_n\}_{n\in\Lambda_N}$ a set of integers associated with the set of real values $\{b_n = \frac{L}{M_n}\}_{n\in\Lambda_N}$, then the associated *discrete, nonstationary Gabor system* $\mathcal{G}(\mathbf{g},\mathbf{b})$ is given by

$$g_{n,m}[l] = \mathbf{M}_{b_n}g_n[l] = g_n[l]e^{\frac{2\pi i m b_n l}{L}},$$

for $n = 0,\dots,N-1$, $m = 0,\dots,M_n - 1$ and all $l = 0,\dots L-1$. The integer M_n denotes the number of equidistant frequency positions considered at the n-th temporal position.

Note that in practice, $g_{n,m}[l]$ will have zero-values for most l, allowing for efficient FFT-implementation: since $M_n = \frac{L}{b_n}$, we have $g_{n,m}[l] = g_n[l] \cdot e^{\frac{2\pi iml}{M_n}}$ and the nonstationary Gabor coefficients are given by an FFT of length M_n for each g_n. More explicitly, the samples $c[n,m] = (f\overline{g_n})^\wedge [mb_n]$ are obtained by taking the zero-extending $f\overline{g_n}$ to a multiple of M_n samples and computing the Fourier transform of the vector

$$\sum_{k=0}^{\lceil L/M_n \rceil -1} f\overline{g_n}[\cdot + kM_n] \ \in \mathbb{C}^{M_n}.$$

This application of Poisson's summation formula is sometimes referred to as *periodization trick*.

Consequently, we compute NSG analysis coefficients via the simple and efficient Algorithm 3. For convenience, we use the notation

$$c := \{c_n\}_{n\in\Lambda_N} := \{\{c_{n,m}\}_{m\in\Lambda_{M_n}}\}_{n\in\Lambda_N}$$

to refer to the full set of coefficients and coefficients at one time position, respectively. By abuse of notation, we indicate by $c \in \mathbb{C}^{N\times M_n}$ that c is an irregular array with N rows, the n-th row possessing M_n entries.

Algorithm 3 NSG analysis: $c = \mathbf{NSGT}_L(f, \mathbf{g}, \mathbf{b})$

```
Initialize $f, g_n$ for all $n \in \Lambda_N$
for $n \in \Lambda_N$ do
    $c_n \leftarrow \sqrt{M_n} \cdot \mathbf{FFT}_{M_n}(f\overline{g_n})$
end for
```

Here $\mathbf{(I)FFT}_N$ denotes a (inverse) Fast Fourier transform of length N, including the necessary zero-padding preprocessing to convert the input vector to the correct length N.

The remaining problem is to ascertain that $\mathcal{G}(\mathbf{g}, \mathbf{b})$ is a frame and to compute the dual frame. The following proposition is a discrete, frequency side version of an equivalent result for NSG systems in $L^2(\mathbb{R})$ and achieves both, using the painless case conditions: Let I_n denote the minimal closed interval such that $\operatorname{supp}(g_n) \subseteq I_n$ and $L_n = |I_n|$ its length. Then we call the nonstationary Gabor system $\mathcal{G}(\mathbf{g}, \mathbf{b})$ a *painless* system if and only if

$$b_n \leq L/L_n \text{ or equivalently } M_n \geq L_n, \text{ for all } n \in \Lambda_N. \tag{5.4}$$

Proposition 14. *Let* $\mathcal{G}(\mathbf{g}, \mathbf{b})$ *an NSG system satisfying* (5.4). *This system is a frame if and only if*

$$0 < \sum_{n\in\Lambda_N} M_n |g_n[l]|^2 < \infty, \qquad \textit{for all } l = 0, \dots, L-1 \tag{5.5}$$

and the windows generating the canonical dual frame $\mathcal{G}(\widetilde{\mathbf{g}}, \mathbf{b})$ are given by

$$\widetilde{g_n}[l] = \frac{g_n[l]}{\sum_{k\in\Lambda_N} M_k|g_k[l]|^2}. \tag{5.6}$$

Proof. Denote by I_n an interval of length L_n, containing the support of g_n. By assumption

$$0 < \sum_{n\in\Lambda_N} M_n|g_n[l]|^2 < \infty, \quad \text{for all } l = 0, \dots, L-1$$

and $M_n \geq L_n = |I_n|$. Note that the frame operator (5.1) can be written as follows

$$\begin{aligned}\mathbf{S}f[l] &= \sum_{n\in\Lambda_N} \sum_{m=0}^{M_n-1} \langle f, \mathbf{M}_{mb_n} g_n \rangle \mathbf{M}_{mb_n} g_n[l] \\ &= \sum_{n\in\Lambda_N} \sqrt{M_n} \sum_{m=0}^{M_n-1} \mathbf{FFT}_{M_n}(f\overline{g_n})[m] g_n[l] e^{-2\pi i m b_n/L} \\ &= \sum_{n\in\Lambda_N} M_n \mathbf{IFFT}_{M_n}(\mathbf{FFT}_{M_n}(f\overline{g_n}))[l] g_n[l], \end{aligned} \tag{5.7}$$

for all $f \in \mathbb{C}^L$. Furthermore, with χ_{I_n} the characteristic function of the interval I_n,

$$\begin{aligned} f\overline{g_n} &= \chi_{I_n} \sum_{k=0}^{b_n-1} \mathbf{T}_{kM_n}(f\overline{g_n}) \\ &= \chi_{I_n} \mathbf{IFFT}_{M_n}(\mathbf{FFT}_{M_n}(f\overline{g_n})) \end{aligned}$$

and, obviously, $g_n = \chi_{I_n} g_n$. Inserting into (5.7) yields

$$\begin{aligned} \mathbf{S}f[l] &= \sum_{n\in\Lambda_N} M_n (f\overline{g_n})[l] g_n[l] \\ &= f[l] \sum_{n\in\Lambda_N} M_n |g_n|^2[l]. \end{aligned} \tag{5.8}$$

With the sum bounded above and below, the inverse frame operator can be written as

$$\mathbf{S}^{-1} f[l] = f[l] \left(\sum_{n\in\Lambda_N} M_n |g_n|^2[l] \right)^{-1}, \text{ for all } f \in \mathbb{C}^L. \tag{5.9}$$

Since the elements of the canonical dual frame are given by (5.2), this completes the proof. □

Remark 17. *The maximum and minimum of the sum in* (5.5) *give the upper and lower frame bound B_0, A_0, respectively.*

The analysis algorithm above is complemented by Algorithm 4, an equally simple synthesis algorithm that synthesizes a signal $\tilde{f}$ from a set of coefficients c.

Algorithm 4 NSG synthesis: $\tilde{f} = \mathbf{iNSGT}_L(c, \tilde{\mathbf{g}}, \mathbf{b})$

```
Initialize $c_{n,m}, \widetilde{g_m}$ for all $m \in \Lambda_{M_n}$, $n \in \Lambda_N$
for $n \in \Lambda_N$ do
    $f_n \leftarrow M_n^{-1/2} \cdot \mathbf{IFFT}_{M_n}(c_n)$
end for
$\tilde{f} \leftarrow \sum_{n \in \Lambda_N} f_n \widetilde{g_n}$
```

Remark 18. *The algorithms in this section can also be applied for $M_n < L_n$. However, in this case, applying **(I)FFT**$_{M_n}$ may require periodization or periodic extension, respectively, to convert the input to length M_n or the output to length L_n.*

It is quite easy to see that, if $\mathcal{G}(\mathbf{g}, \mathbf{b})$ and $\mathcal{G}(\mathbf{g}, \mathbf{b})$ form a pair of dual NSG frames, then $\mathbf{iNSGT}_L(c, \tilde{\mathbf{g}}, \mathbf{b})$ applied to $c = \mathbf{NSGT}_L(f, \mathbf{g}, \mathbf{b})$ reconstructs any $f \in \mathbb{C}^L$ perfectly. For a detailed proof, see Proposition 15 in the following section.

Another way to determine the finite, discrete painless case condition is the Walnut representation for the NSG frame operator over $\mathbb{C}^L$. Let $P = \sum_{n=0}^{N-1} M_n$ be the number of elements of $\mathcal{G}(\mathbf{g}, \mathbf{b})$. Then $\mathbf{D} \in \mathbb{C}^{L \times P}$ and $\mathbf{C} \in \mathbb{C}^{P \times L}$.

Corollary 16. *The frame operator $\mathbf{S} = \mathbf{D} \cdot \mathbf{C}$ of the NSG system $\mathcal{G}(\mathbf{g}, \mathbf{b})$ is an $L \times L$ matrix with entries:*

$$\mathbf{S}_{l,k} = \sum_{n \in \mathcal{N}_{(l-k)}} M_n \, g_n[l] \, \overline{g_n}[k]$$

where $\mathcal{N}_p = \{n \in \Lambda_N \ : \ p = 0 \bmod M_n\}$ for $p \in [-L+1, L-1]$. Therefore, if appropriate support conditions are met, $\mathbf{S}$ is a diagonal matrix.

Proof. Recall the definition of the frame operator (5.1):

$$\begin{aligned}
\mathbf{S}f[l] &= \sum_{n \in \Lambda_N} \sum_{m \in \Lambda_{M_n}} \langle f, g_{n,m} \rangle g_{n,m}[l] \\
&= \sum_{n \in \Lambda_N} \sum_{m \in \Lambda_{M_n}} \sum_{k=0}^{L-1} f[k] \overline{g_n}[k] g_n[l] \exp(2\pi i (l-k) m / M_n) \\
&= \sum_{n \in \Lambda_N} \sum_{k=0}^{L-1} f[k] \overline{g_n}[k] g_n[l] \sum_{m \in \Lambda_{M_n}} \exp(2\pi i (l-k) m / M_n),
\end{aligned}$$

where the sum over m equals M_n if $l-k \bmod M_n = 0$ and 0 otherwise. Consequently, defining $\mathcal{N}_{l-k} = \{n \in \Lambda_N | \, l-k \bmod M_n = 0\}$, the entries of $\mathbf{S}$ can be written as

$$\mathbf{S}_{l,k} = \sum_{n \in \mathcal{N}_{(l-k)}} M_n \, g_n[l] \, \overline{g_n}[k].$$

Assume $L_n < M_n$, with L_n the size of the support of g_n as above, for all $n \in \Lambda_N$. Then $\mathbf{S}_{l,k} = 0$ for all $k \neq l$ and $\mathbf{S}$ is diagonal. □

If b_n is not chosen to be a divisor L, the modulations $\exp(2\pi i m \cdot /M_n)$ produce a jump on the circular border of $\mathbb{C}^L$. For prototype functions g_n supported on that border, the modulation then introduces "discontinuities", negatively affecting the frequency decay properties of $g_{n,m}$. However, restricting to $b_n \mid L$ severely limits the flexibility of choosing M_n. We can navigate around the continuity issue by using a *phaselocked* implementation of of the transform. Phaselocking is a technique frequently used in signal processing, where the modulations are considered relative to the window position instead of the signal.

More explicitly, assume the support of g_n to be smaller than L and centered at the sample a_n. Then instead of the frame elements in Definition 14, we use

$$g_{n,m}[l] = g_n[l] \exp(2\pi i (l - a_n)_{\Lambda_{L,a_n}} m/M_n), \tag{5.10}$$

where the notation $(l)_{\Lambda_{L,a_n}}$ indicates that l is considered circularly on the interval $\Lambda_{L,a_n} := [-\lfloor L/2 \rfloor + a_n, \lceil L/2 \rceil + a_n - 1]$. This modification guarantees that the modulation does not introduce "jumps" to $g_{n,m}$. Phaselocked systems come, however, with the drawback of a slightly more complicated frame operator representation.

Corollary 17. *Let $\mathcal{G}(\mathbf{g}, \mathbf{b})$ be a phaselocked NSG system as in Equation* (5.10). *Furthermore, define $x_{n,l,k}$ by*

$$x_{n,l,k} = \begin{cases} l-k, & \text{if } l,k \in \Lambda_{L,a_n} \\ l-k-L, & \text{if } -\lfloor L/2 \rfloor + a_n < 0, \; k \in \Lambda_{L,a_n}, \; l \notin \Lambda_{L,a_n} \\ & \text{or } -\lfloor L/2 \rfloor + a_n \geq 0, \; k \notin \Lambda_{L,a_n}, \; l \in \Lambda_{L,a_n} \\ L+l-k, & \text{if } -\lfloor L/2 \rfloor + a_n \geq 0, \; k \in \Lambda_{L,a_n}, \; l \notin \Lambda_{L,a_n} \\ & \text{or } -\lfloor L/2 \rfloor + a_n < 0, \; k \notin \Lambda_{L,a_n}, \; l \in \Lambda_{L,a_n}. \end{cases}$$

The frame operator $\mathbf{S} = \mathbf{D} \cdot \mathbf{C}$ of the NSG system $\mathcal{G}(\mathbf{g}, \mathbf{b})$ is an $L \times L$ matrix with entries:

$$\mathbf{S}_{l,k} = \sum_{n \in \mathcal{N}_{x_{n,l,k}}} M_n \, g_n[l] \, \overline{g_n}[k]$$

where $\mathcal{N}_{x_{n,l,k}} = \{n \in \Lambda_N \; : \; x_{n,l,k} = 0 \bmod M_n\}$ for all $x_{n,l,k}$. Therefore, if appropriate support conditions are met, $\mathbf{S}$ *is a diagonal matrix.*

Proof. Analogue to Corollary 16. □

5.2.1 Numerical complexity

Assuming that the windows g_n have support of length L_n, let $M = \max_n \{M_n\}$ be the maximum FFT-length. We consider the painless case where $L_n \leq M_n \leq M$. The number of operations is

1. Windowing: L_n operations for the n-th window.
2. FFT: $\mathcal{O}(M_n \cdot \log(M_n))$ for the n-th window.

Then the number of operations for the discrete NSGT is

$$\begin{aligned}\mathcal{O}\left(\sum_{n=0}^{N-1} M_n \cdot \log(M_n) + L_n\right) &= \mathcal{O}(N \cdot (M \log(M) + M)) \\ &= \mathcal{O}(N \cdot (M \log(M)))\end{aligned}$$

Similar to the regular Gabor case, the number of windows N will usually depend linearly on the signal length L while the maximum FFT-length M is assumed to be independent of L. In that case, the discrete NSGT is a linear cost algorithm.

For the construction of the dual windows in the painless case, the computation involves multiplication of the window functions by the inverse of the diagonal matrix $\mathbf{S}$ and results in $\mathcal{O}(2\sum_{n=0}^{N-1} L_n) = \mathcal{O}(N \cdot M)$ operations. Lastly, the inverse NSGT has numerical complexity $\mathcal{O}(N \cdot (M \log(M)))$, as in the NSGT, since it entails computing the IFFT of each coefficient vector, multiplying with the corresponding dual windows and evaluating the sum.

5.2.2 Frequency-adaptive nonstationary Gabor frames

The original motivation for the introduction of NSGT was the desire to adapt both window size and sampling density in time, cf. [10,95], in order to accurately resolve transient signal components. Here, we apply the same idea in frequency, i.e. adapt both the bandwidth and sampling density in frequency. From an algorithmic point of view, we apply a nonstationary Gabor system to the Fourier transform of the input signal.

The windows are constructed directly in the frequency domain by taking real-valued filters g_m centered at frequency ω_m (in samples). The inverse Fourier transforms $\widecheck{g_m} := \mathcal{F}^{-1} g_m$ are the time-reverse impulse responses of the corresponding (frequency-adaptive) filters. Therefore, we let $\widecheck{g_m}$, $m \in \Lambda_M$, denote the members of a finite collection of band-limited windows, well-localized in time, whose Fourier transforms $g_m = \mathcal{F}\widecheck{g_m}$ are centered around possibly irregularly (or, e.g. geometrically) spaced frequency points ω_m.

Then, analogous to (5.4) before, we select frequency dependent time-shift parameters (hop-sizes) a_m as follows: if the *support* (the interval where the vector is nonzero) of g_m is contained in an interval of length L_m, then a_m is chosen such that

$$a_m \leq \frac{L}{L_m} \text{ and } N_m = L/a_m \in \mathbb{N}, \text{ for all } m \in \Lambda_M. \tag{5.11}$$

In other words, the time-sampling points have to be chosen dense enough to guarantee (5.11). If we denote by $g_{n,m}$ the modulation of g_m by $-na_m$, i.e. $g_{n,m} = \mathbf{M}_{-na_m} g_m$, then we obtain the frame members $\varphi_{n,m}$ by setting

$$\varphi_{n,m} = g_{n,m} := \mathcal{F}^{-1}(\mathbf{M}_{-na_m} g_m) = \mathbf{T}_{na_m}\breve{g_m},$$

where $m \in \Lambda_M$ and $n = \Lambda_{N_m}$. The system $\breve{\mathcal{G}}(\mathbf{g},\mathbf{a}) := \{g_{n,m} = \mathbf{T}_{na_m}\breve{g_m}\}_{n,m}$ is a *painless frequency-side nonstationary Gabor system* (or NSG filterbank) for $\mathbb{C}^L$, as described in the previous section. We also define $\mathbf{g} := \{g_m \in \mathbb{C}^L\}_{m\in\Lambda_M}$ and $\mathbf{a} := \{a_m\}_{m\in\Lambda_M}$. By Parseval's formula, we see that the frame coefficients can be written as

$$c_{n,m} = \langle f, g_{n,m}\rangle = \langle \hat{f}, \mathbf{M}_{-na_m} g_m\rangle. \tag{5.12}$$

For convenience, we use the notation $c := \{c_m\}_{m\in\Lambda_M} := \{\{c_{n,m}\}_{n\in\Lambda_{N_m}}\}_{m\in\Lambda_M}$ to refer to the full set of coefficients and channel coefficients, respectively. By abuse of notation, we indicate by $c \in \mathbb{C}^{N_m\times M}$ that c is an irregular array with M columns, the m-th column possessing N_m entries. The NSG coefficients can be computed using the following algorithm.

Algorithm 5 Frequency side NSG analysis: $c = \mathbf{NSGTF}_L(f, \mathbf{g}, \mathbf{a})$

```
Initialize $f, g_m$ for all $m \in \Lambda_M$
$f \leftarrow \mathbf{FFT}_L(f)$
for $m \in \Lambda_M$ do
    $c_m \leftarrow \sqrt{N_m} \cdot \mathbf{IFFT}_{N_m}(f\overline{g_m})$
end for
```

The analysis algorithm above is complemented by Algorithm 6, an equally simple synthesis algorithm that synthesizes a signal $\tilde{f}$ from a set of coefficients c.

Remark 19. *Similar to the previous section, the algorithms proposed in this section can also be applied for $a_m > \frac{L}{L_m}$. However, in this case, applying **(I)FFT**$_{N_m}$ may require periodization or periodic extension, respectively, to convert the input to length N_m or the output to length L_m.*

If $\breve{\mathcal{G}}(\mathbf{g},\mathbf{a})$ and $\breve{\mathcal{G}}(\widetilde{\mathbf{g}},\mathbf{a})$ are a pair of dual frames, then we can reconstruct a function perfectly from its NSG analysis coefficients.

Algorithm 6 Frequency side NSG synthesis: $\tilde{f} = \mathbf{iNSGTF}_L(c, \tilde{\mathbf{g}}, \mathbf{a})$

```
Initialize $c_{n,m}, \widetilde{g_m}$ for all $n \in \Lambda_{N_m}$, $m \in \Lambda_M$
for $m \in \Lambda_M$ do
    $f_m \leftarrow N_m^{-1/2} \cdot \mathbf{FFT}_{N_m}(c_m)$
end for
$\tilde{f} \leftarrow \sum_{m \in \Lambda_M} f_m \widetilde{g_m}$
$\tilde{f} \leftarrow \mathbf{IFFT}_L(\tilde{f})$
```

Proposition 15. *Let* $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a}) = \{g_{n,m} = \mathbf{T}_{na_m}\breve{g_m}\}_{n,m}$ *and* $\breve{\mathcal{G}}(\widetilde{\mathbf{g}}, \mathbf{a}) = \{\widetilde{g_{n,m}} = \mathbf{T}_{na_m}\breve{\widetilde{g_m}}\}_{n,m}$ *be a pair of dual frames. If c is the output of* $\mathbf{NSGTF}_L(f, \mathbf{g}, \mathbf{a})$ *(Algorithm 5), then the output $\tilde{f}$ of* $\mathbf{iNSGTF}_L(c, \tilde{\mathbf{g}}, \mathbf{a})$ *(Algorithm 6) equals f, i.e.*

$$\tilde{f} = f, \quad \textit{for all } f \in \mathbb{C}^L. \tag{5.13}$$

Proof. By Algorithm 5, we have

$$\begin{aligned} c_{n,m} &= c_m[n] \\ &= \sqrt{N_m}\frac{1}{\sqrt{N_m}} \sum_{k=0}^{N_m-1} \sum_{l=0}^{a_m-1} (\hat{f}\overline{g_m})[k + lN_m] e^{2\pi i n k a_m / L} \\ &= \sum_{k=0}^{N_m-1} \sum_{l=0}^{a_m-1} (\hat{f}\mathbf{M}_{na_m}\overline{g_m})[k + lN_m]. \end{aligned} \tag{5.14}$$

Since $N_m \geq L_m$, only one element of the inner sum above is non-zero, for each $k \in \{0, \dots, N_m - 1\}$. It follows that

$$c_{n,m} = \langle \hat{f}, \mathbf{M}_{-na_m} g_m \rangle. \tag{5.15}$$

Inserting into Algorithm 6 yields, for all $l \in \{0, \dots, L-1\}$,

$$\begin{aligned} \hat{\tilde{f}}[l] &= \sum_{m \in \Lambda_M} \sum_{n=0}^{N_m-1} c_{n,m} e^{-2\pi i n l a_m / L} \widetilde{g_m}[l] \\ &= \sum_{m \in \Lambda_M} \sum_{n=0}^{N_m-1} \langle \hat{f}, \mathbf{M}_{-na_m} g_m \rangle \mathbf{M}_{-na_m} \widetilde{g_m}[l], \end{aligned}$$

the discrete frame synthesis formula. By assumption, $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a})$ and $\breve{\mathcal{G}}(\widetilde{\mathbf{g}}, \mathbf{a})$ are dual NSG frames and thus

$$\hat{\tilde{f}}[l] = \hat{f}[l], \qquad \text{for all } j \in \{0, \dots, L-1\}.$$

Applying the inverse discrete Fourier transform completes the proof. □

Since the DFT is a unitary operator on $\mathbb{C}^L$, it is easy to see that the painless case conditions (Proposition 14) apply to frequency side nonstationary Gabor systems with bandlimited filters in an analogue fashion.

In Section 5.4, we construct a constant-Q NSG system satisfying (5.11) and (5.5).

Remark 20. *Note that NSG frames can be equivalently used to design general nonuniform filter banks [109, 121] in a similar manner.*

5.3 Time-side systems

Technical framework: The simulations presented in this section were done in MATLAB R2009b on a 3 Gigahertz Intel Core 2 Duo machine with 2 GB RAM.

5.3.1 Automatic adaptation to transients

In real-life applications, NSGT has the potential to represent local signal characteristics, e.g. transient sound events, in a more appropriate way than predetermined, regular transform schemes. Since the appropriateness of a representation depends on the specific application, any adaptation procedure must be designed specifically. For the implementation itself, however, two observations generally remain true: First, the general nonstationary framework needs to be restricted to a well defined set of choices. Second, some measure is needed to determine the most suitable of the possible choices. For example, in the case of a sparsity measure, the most sparse representation will be chosen. To show that good results are achieved even when using quite simple adaptation methods, we describe a procedure suitable for signals consisting mainly of transient and sinusoidal components. The adaptation measure proposed is based on onset detection, i.e. estimating where transients occur in the signal. The transform setting is what we call *scale frames*: the analysis procedure uses a single window prototype and a countable set of dilations thereof.

For evaluation, the representation quality is measured by comparison of the number of representation coefficients leading to certain root mean square (RMS) reconstruction errors, for both NSGT and regular Gabor transforms. The results are especially convincing for sparse music signals with high energy transient components. Other possible adaptation methods might be based e.g. on time-frequency concentration, sparsity or entropy measures [171], [97], [112].

Scale frames: In the following paragraphs, we propose a family of nonstationary Gabor frames that allows for exponential changes in time-frequency resolution along time positions. To avoid heavy notation and since the formalism necessary for the discrete, finite case could obscure the principal idea, we describe the continuous case construction. Suitable standard sampling then yields discrete, finite frames with equivalent characteristics.

The basic idea is to build a sequence of windows g_n from a single, continuous window prototype g with support on an interval of length 1 in such a way that the resulting g_n satisfy Proposition 14. The window sequence will be unambiguously determined by a sequence of scales. Once this *scale sequence* is known, it is a simple task to choose modulation parameters b_n satisfying the necessary conditions.

As a scale sequence, we allow any integer-valued sequence $\{s_n\}_{n\in\mathbb{Z}}$ such that $|s_n - s_{n-1}| \in \{0, 1\}$, where the latter restriction is set in order to avoid sudden changes of window length. Then, g_n is, up to translation, given by a dilation of the prototype g:

$$\mathbf{D}_{2^{s_n}}(g)(t) = \sqrt{2^{-s_n}}g(2^{-s_n}t).$$

This implies that a change of scale from one time step to the next corresponds to the use of a window either half or twice as long. More precisely, for every time step n, set $s = \min\{s_{n-1}, s_n\}$ and fix an overlap of $2/3 \cdot 2^s$, if $s_n \neq s_{n-1}$ and $1/3 \cdot 2^s$, if $s_n = s_{n-1}$. Explicitly,

$$g_n = \mathcal{T}_n \mathbf{D}_{2^{s_n}}(g),$$

with recursively defined time shift operators $\mathcal{T}_n$ given by

$$\mathcal{T}_0 = \mathbf{T}_0, \mathcal{T}_n = \begin{cases} \mathbf{T}_{2^s 5/6}\mathcal{T}_{n-1}, & \text{if } s_n \neq s_{n-1} \\ \mathbf{T}_{2^{s+1}/3}\mathcal{T}_{n-1}, & \text{else.} \end{cases}$$

Defining the time shifts in this manner, we achieve exactly the desired overlap as illustrated in Figure 5.1.

By construction, each g_n has non-zero overlap with its neighbors g_{n-1} and g_{n+1} and at any point on the real line, at most two windows are non-zero. After performing a preliminary transient detection step, as explained before, the construction of the adapted frame reduces to the determination of a scale sequence.

In the subsequent figures and experiments we used the Hann window as prototype, but other window choices are possible. The described concept can easily be generalized by admitting other overlap factors and scaling ratios than the ones specified above. The parameters have to be chosen with some care, though. Otherwise the resulting frames might be badly conditioned, with a big or even infinite condition number $\frac{B}{A}$, caused by accumulation points for the time shifts or gaps between windows.

Frame construction from a sequence of onsets: In this paragraph, we assume that the signals of interest are mainly comprised of transient and sinusoidal

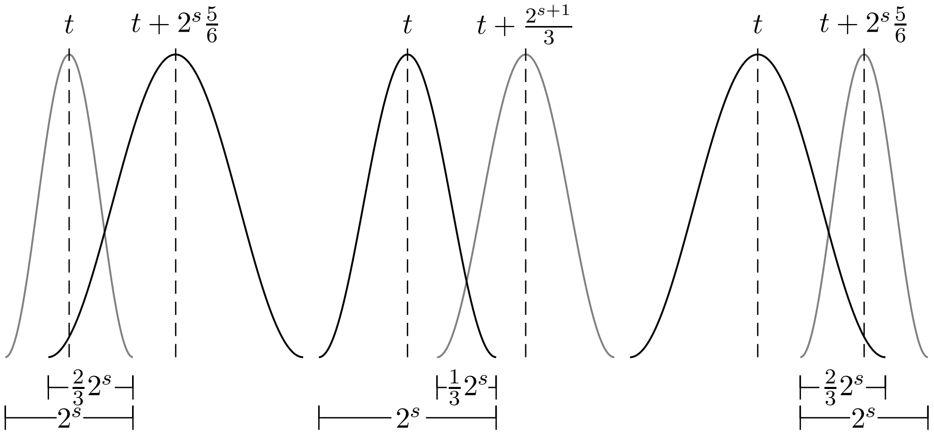


Figure 5.1: Illustration of scale frame overlaps and time shifts.

components, an assumption met, e.g. by piano music. The instant a piano key is hit corresponds to a percussive, transient sound event, directly followed by harmonic components, concentrated in frequency. An intuitive adaptation to signals of this type would use high time resolution at the positions of transients. This corresponds to applying minimal scale at the transients and steadily increasing the scale with the distance from the closest transient. The transients' positions can be determined, e.g. by so-called onset detection procedures [52] which, if used carefully, work to a high degree of accuracy. Once the transient positions are known, the construction of a corresponding scale frame yields good nonstationary representations for sufficiently sparse signals.

Application of onset-based scale frames: We applied the procedure proposed above to various signals, mainly piano music. For this presentation, we selected three examples, all of them sampled at 44.1 kHz and consisting of a single channel. Some more examples and corresponding results as well as the source sound files can be found on the associated webpage `http://univie.ac.at/nonstatgab/`.

- Example 1: The widely used Glockenspiel signal shown in Figure 4.3, Chapter 4.
- Example 2: An excerpt from a solo jazz piano piece performed by Herbie Hancock, characterized by its calmness and varied rhythmical pattern, resulting in irregularly spaced low-energy transients. See Figure 5.2.
- Example 3: A short excerpt of György Ligeti's piano concert. With highly percussive onsets in the piano and Glockenspiel voices and some orchestral background, this is the most polyphonic of our examples. See Figure 5.3.

For comparison, the plots in Figures 4.3, 5.2 and 5.3 also show standard Gabor coefficients with comparable (average) window overlap. A Hann window of 2560 samples length was chosen for the computation of regular Gabor transforms. The comparison shows that for the three signals, the NSGT features a better concentration of transient energy than a regular Gabor transform, while keeping, or even improving, frequency resolution.

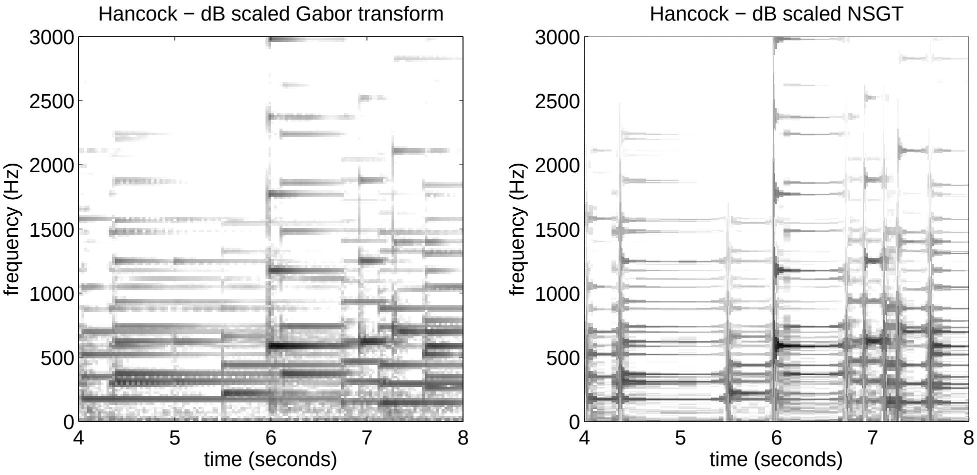


Figure 5.2: Hancock (Example 2). Regular and nonstationary Gabor representations.

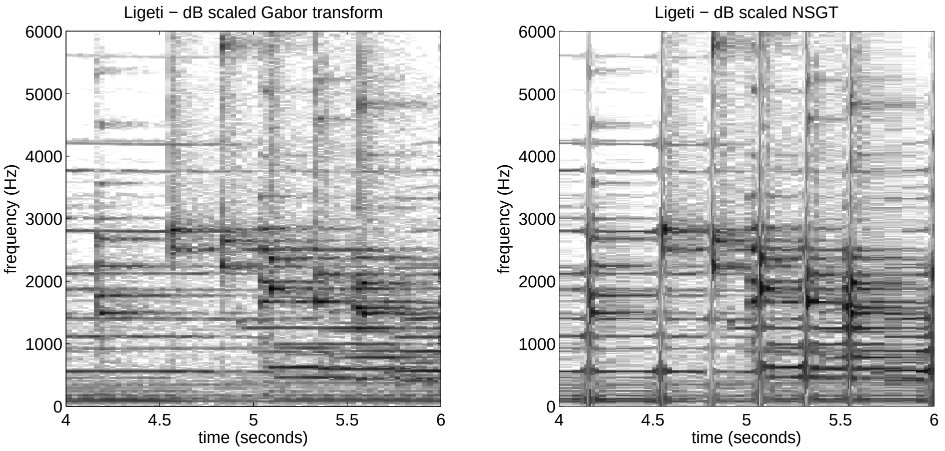


Figure 5.3: Ligeti (Example 3). Regular and nonstationary Gabor representations.

Efficiency in sparse reconstruction: The onset detection procedure and a subsequent scale frame analysis were applied, along with a regular Gabor decomposition, to the Glockenspiel and Ligeti signals. As a test of the representations' sparsity, the signals were synthesized from their corresponding coefficients, modified by hard thresholding followed by reconstruction using the canonical dual frame. Then the numbers of largest magnitude coefficients needed for a certain relative root mean square (RMS) reconstruction error for each representation were compared. The RMS difference of a vector f and its reconstruction f_{rec} is given by

$$RMS(f, f_{rec}) = \sqrt{\frac{\sum_{k=0}^{L-1} |f[l] - f_{rec}[l]|^2}{\sum_{k=0}^{L-1} |f[l]|^2}}.$$

All transforms are of redundancy about $\frac{5}{3}$. The results for NSGT and different regular Gabor transform schemes are listed in Figure 5.4. On the Glockenspiel signal the NSGT method performs vastly better than the ordinary Gabor transform. For Ligeti, the differences are not as significant, but still the NSGT-based procedure shows better overall results.

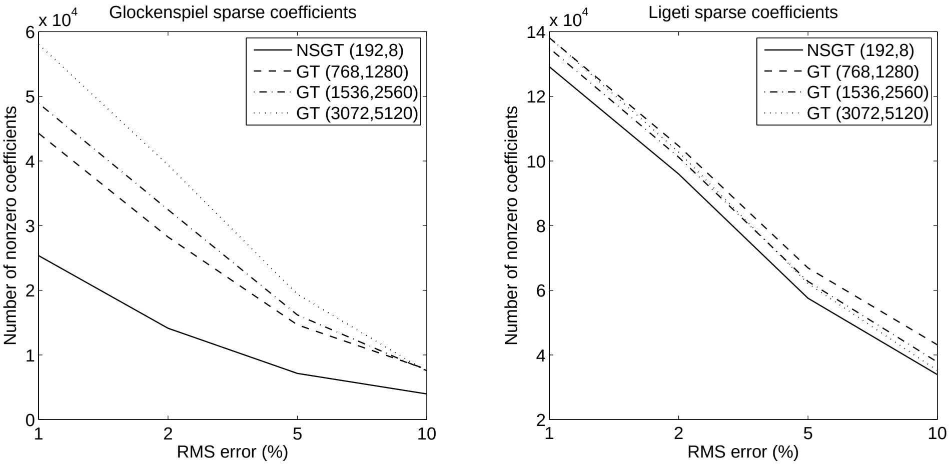


Figure 5.4: RMS error in sparse representations of Example 1 and Example 3. Parameters (in parentheses) are hop size and window length in the regular case (GT) or shortest window length and number of scales for the nonstationary case (NSGT). The values are estimated to be the optimal numbers of coefficients necessary to achieve reconstruction with less than the respective error.

Further experiments and a more exhaustive discussion of the parameters used in the experiments, can be found on the webpage `http://univie.ac.at/nonstatgab/`. Along them, examples of regular and nonstationary reconstructions from a specified amount of coefficients can be found, so the reader might get a subjective impression of perceptive reconstruction quality. In conclusion, the experiments show that for real music signals, NSGT can provide a sparser representation than regular Gabor transforms, admitting reasonable reconstruction error.

5.4 Frequency-side systems

5.4.1 A framework for invertible, real-time constant-Q transforms

Analysis, synthesis and processing of sound is commonly based on the representation of audio signals by means of time-frequency dictionaries. The short-time Fourier transform (STFT), also referred to as *Gabor transform*, is a widely used tool due to its straightforward interpretation and FFT-based implementation, which ensure efficiency and invertibility [53, 118]. STFT features a uniform time and frequency resolution and a linear spacing of the time frequency bins.

In contrast, the constant-Q transform (CQT), originally introduced in [172] and in music processing by J. Brown [26], provides a frequency resolution that depends on geometrically spaced center frequencies of the analysis windows. In particular, the Q-factor, i.e. the ratio of center frequency to bandwidth of each window, is constant over all frequency bins; the constant Q-factor leads to a finer frequency resolution in low frequencies, whereas time resolution improves with increasing frequency. This principle makes the constant-Q transform well-suited for audio data, since it better reflects the resolution of the human auditory system than the linear frequency-spacing provided by the FFT, cf. [144] and references therein. Furthermore, musical characteristics such as overtone structures remain invariant under frequency shifts in a constant-Q transform, which is a natural feature from a perception point of view. In speech and music processing, perception-based considerations are important, which is one of the reasons why CQTs, due to their previously discussed properties, are often desirable in these fields. An example of a CQ-transform, obtained with our algorithm, is shown in Figure 5.5.

The principal idea of CQT is reminiscent of wavelet transforms, compare [142]. Since they are based on iterated filter banks, these methods are computationally too expensive for long, real-life signals, when high Q-factors, such as 12-96 bins per octave, are required. As opposed to wavelet transforms, the original CQT is not invertible and does not rely on any concept of (orthonormal) bases. On the other hand, the number of bins (frequency channels) per octave is much higher in the CQT than most traditional wavelet techniques would allow for. Partly due to this requirement, the computational efficiency of the original transform as well as its improved versions, cf. [25], may often be insufficient. Moreover, the lack of invertibility of existing CQTs has become an important issue: for some desired applications, such as extraction and modification, e.g. transposition, of distinct parts of the signal, the unbiased reconstruction from analysis coefficients is crucial. Approximate methods for reconstruction from constant-Q coefficients have been proposed before, in particular for signals which are sparse in the frequency domain [44] and by octave-wise processing in [141]. In [141], Klapuri and Schörkhuber presented a computation of the CQT that shows improved efficiency and flexibility compared to the method proposed in [25], among others. However, the approximate inversion introduced in [141] still gives an RMS error of around 10^{-3}. The lack of perfect invertibility prevents the convenient modification of CQT-coefficients with subsequent resynthesis required in complex music processing tasks such as masking or transposition.

In the present contribution, we are interested in inversion in the sense of *perfect reconstruction*; to this end, we investigate a new approach to constant-Q signal processing. The presented framework has the following core properties:

1. Relying on concepts from frame theory, [118], we suggest the implementation of a constant-Q transform using the nonstationary Gabor transform (NSGT), which guarantees perfect invertibility. This perfectly invertible constant-Q transform is subsequently called *constant-Q nonstationary Gabor transform* (CQ-NSGT).

2. We introduce a preprocessing step by *slicing* the signal to pieces of (usually uniform) finite length. Together with FFT-based methods, this allows for bounded delay and results in linear processing time. Thus, our algorithm lends itself to real-time processing and the resulting transform is referred to as *sliced constant-Q transform* (sliCQ).

NSGTs, introduced in [10, 95], generalize the classical sampled short-time Fourier transform or Gabor transform [71, 118]. They allow for fast, FFT-based implementation of both analysis and reconstruction under mild conditions on the analysis windows. The CQ-NSGT was first presented in [164]; the frequency-resolution of the proposed CQ-NSGT is indistinguishable from that of the CQT, cf. Figure 5.5 for an example.

The main drawback of the CQ-NSGT is the inherent necessity to obtain a Fourier transform of the entire signal prior to actual processing. This problem prohibits real-time implementation and is overcome by a slicing step, which preserves the perfect reconstruction property. However, blocking effects and time-aliasing may be observed if the coefficients are modified in applications such as de-noising or transposition and time-shift of certain signal components. While slicing the signal naturally introduces a trade-off between delay and finest possible frequency resolution, the parameters can be chosen to suppress blocking artifacts and to leave the constant-Q coefficient structure intact.

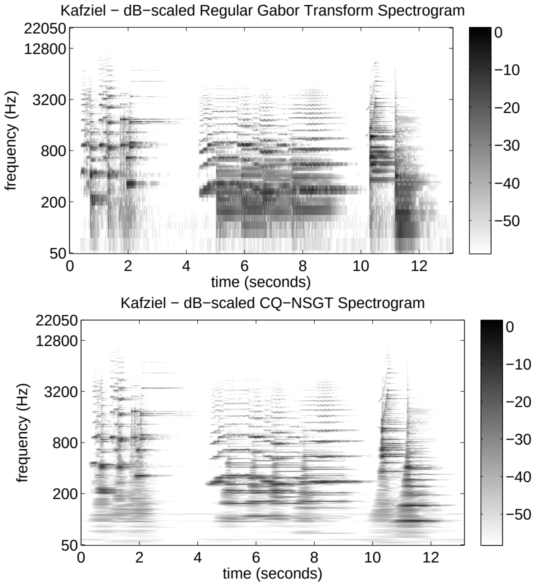


Figure 5.5: Time-frequency representations on a logarithmically scaled frequency axis: STFT spectrogram (top) and constant-Q NSGT spectrogram (bottom).

5.4.2 The CQ-NSGT parameters: Windows and lattices

The parameters of the NSGT can be designed as to implement various frequency-adaptive transforms. Here, we focus on the parameters leading to an NSGT with constant-Q frequency resolution, suitable for the analysis and processing of music signals, as discussed in the paragraphs above. In constant-Q analysis, the functions g_m are considered to be filters with support of length $L_m \leq L$ centered at frequency ω_m (in samples), such that for the bins corresponding to a certain frequency range, the respective center frequencies and lengths have (approximately) the same ratio. Using these filters, the CQ-NSGT coefficients $c_{n,m}$ are obtained via Algorithm 5, where m indexes the frequency bins, and $n \in \Lambda_{N_m}$.

As detailed in [164], the construction of the filters for the CQ-NSGT depends on the following parameters: minimum and maximum frequencies $\xi_{\min}$ and $\xi_{\max}$ (in Hz), respectively, the sampling rate ξ_s, and the number of bins per octave B.
Setting: For the frame elements in the transform, we consider functions $g_m \in \mathbb{C}^L$, $m = 1, \ldots, M$. Their center frequencies ξ_m satisfy $\xi_m = \xi_{\min} 2^{\frac{m-1}{B}}$, similar to the classical CQT in [26], for $m = 1, \ldots, M$, where M is an integer such that $\xi_{\max} \leq \xi_M < \xi_s/2$, the Nyquist frequency. Here $\xi_{\min}$ and $\xi_{\max}$ are the desired minimum and maximum frequencies, respectively. Note that the correspondence between ξ_m and ω_m below is the conversion ratio from Hz to samples, as detailed in the next paragraphs. In this case, we take $M = \lceil B \log_2(\xi_{\max}/\xi_{\min}) + 1 \rceil$, where

Table 5.1: Center frequency and bandwidth values

m	ξ_m	Ω_m
0	0	$2\xi_{\min}$
$1,\ldots,M$	$\xi_{\min}2^{\frac{m-1}{B}}$	ξ_m/Q
$M+1$	$\xi_s/2$	$\xi_s-2\xi_M$
$M+2,\ldots,2M+1$	$\xi_s-\xi_{2M+2-m}$	ξ_{2M+2-m}/Q

$\lceil t\rceil$ is the smallest integer greater than or equal to t. While in the CQT no 0-frequency is present, the NSGT provides all necessary freedom to use additional center frequencies. Since the signals of interest are real-valued, we put filters at center frequencies beyond the Nyquist frequency in a symmetric manner. This results in the following values for the center frequencies, also summarized in Table 5.1:

$$\xi_m = \begin{cases} 0, & m=0 \\ \xi_{\min}2^{\frac{m-1}{B}}, & m=1,\ldots,M \\ \xi_s/2, & m=M+1 \\ \xi_s-\xi_{2M+2-m}, & m=M+2,\ldots,2M+1. \end{cases}$$

The corresponding bandwidth Ω_m of g_m is set to be $\Omega_m=\xi_{m+1}-\xi_{m-1}$, for $m=2,\ldots,M-1$. This leads to a constant Q-factor $Q=\xi_m/\Omega_m=(2^{\frac{1}{B}}-2^{-\frac{1}{B}})^{-1}$, while Ω_1 and Ω_M are taken to be ξ_1/Q and ξ_M/Q, respectively. Since the signals are real-valued, additional filters are considered, which are positioned in a symmetric manner with respect to the Nyquist frequency. Moreover, to ensure that the union of filter supports cover the entire frequency axis, filters with center frequencies corresponding to the zero frequency and the Nyquist frequency are included. The values for Ω_m over all frequency bins are given below and summarized in Table 5.1:

$$\Omega_m = \begin{cases} 2\xi_{\min}, & m=0 \\ \xi_2, & m=1,2M+1 \\ \xi_m/Q, & m=2,\ldots,M-1 \\ (\xi_s-2\xi_{M-1})/2, & m=M,M+2 \\ \xi_s-2\xi_M, & m=M+1 \\ \xi_{2M+2-m}/Q, & m=M+3,\ldots,2M. \end{cases}$$

With these center frequencies and bandwidths, the filters g_m are set to be $g_m[l]=H((l\xi_s/L-\xi_m)/\Omega_m)$, for $m=1,\ldots,M,M+2,\ldots,2M+1$ and $l=$

$0,\ldots,L-1$, where H is some continuous function centered at 0, positive inside and zero outside $]-1/2,1/2[$, i.e. each g_m is a sampled version of a translated and dilated H. A standard choice would be setting $H = 0.5 + 0.5\cos(2\pi t)$, the Hann window. Meanwhile, g_0 and g_{M+1} are taken to be plateau functions, i.e. continuous, compactly supported functions that are constant 1 on some interval, centered at the zero and the Nyquist frequencies respectively. Thus, each filter g_m is centered at $\omega_m = \xi_m L/\xi_s$ and has support $L_m = \Omega_m L/\xi_s$.

Letting $a_m \leq \frac{\xi_s}{\Omega_m}$, we define $g_{n,m}$ by their Fourier transform $\widehat{g_{n,m}} = \mathbf{M}_{-na_m} g_m$, $n = 0,\ldots,\lfloor N_m \rfloor - 1$. Figure 5.6 illustrates the time-frequency sampling grid of the set-up, where the center frequencies are geometrically spaced and sampling points regularly spaced.

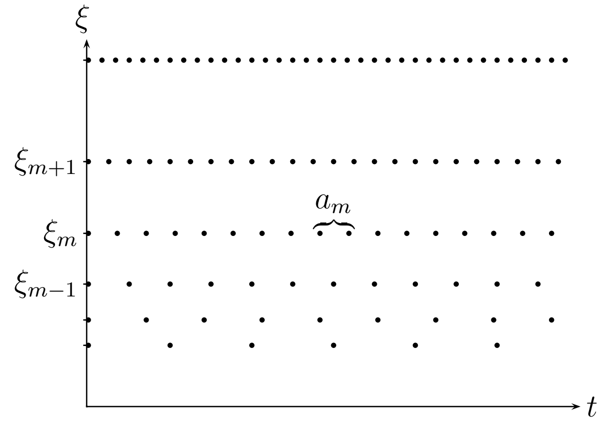


Figure 5.6: Exemplary sampling grid of the time-frequency plane for a constant-Q nonstationary Gabor system.

With the aforementioned parameters, we compute the *phaselocked* CQ-NSGT coefficients as

$$c_{n,m} = \sum_{l=0}^{L-1} \hat{f}[l]\overline{g_m[l]} e^{2\pi i (l-\omega_m)\cdot na_m/L}.$$

This phaselock convention, while slightly different from the definition (5.12) above, does not affect the frame property, yet implementation is more straightforward.

It is easy to see that the support conditions on $\widehat{g_m}$ imply that the sum $\sigma = \sum_{m=0}^{2M+1} N_m \left|\widehat{g_m}\right|^2$ is finite and bounded away from 0. Consequently, this choice of $\breve{\mathcal{G}}(\mathbf{g},\mathbf{a})$ satisfies the conditions of Proposition 14 for any sequence $\mathbf{a}$ with $N_m \geq L_m$ for all $m \in \Lambda_M = \{0,\ldots,2M+1\}$, allowing us to apply Proposition 15. Note that while a_m might be rational, N_m must be integer-valued. Consequently, perfect reconstruction of the signal is obtained from the coefficients $c_{n,m}$ by applying Algorithm 6 with a dual frame, e.g. the canonical dual given by (5.6).

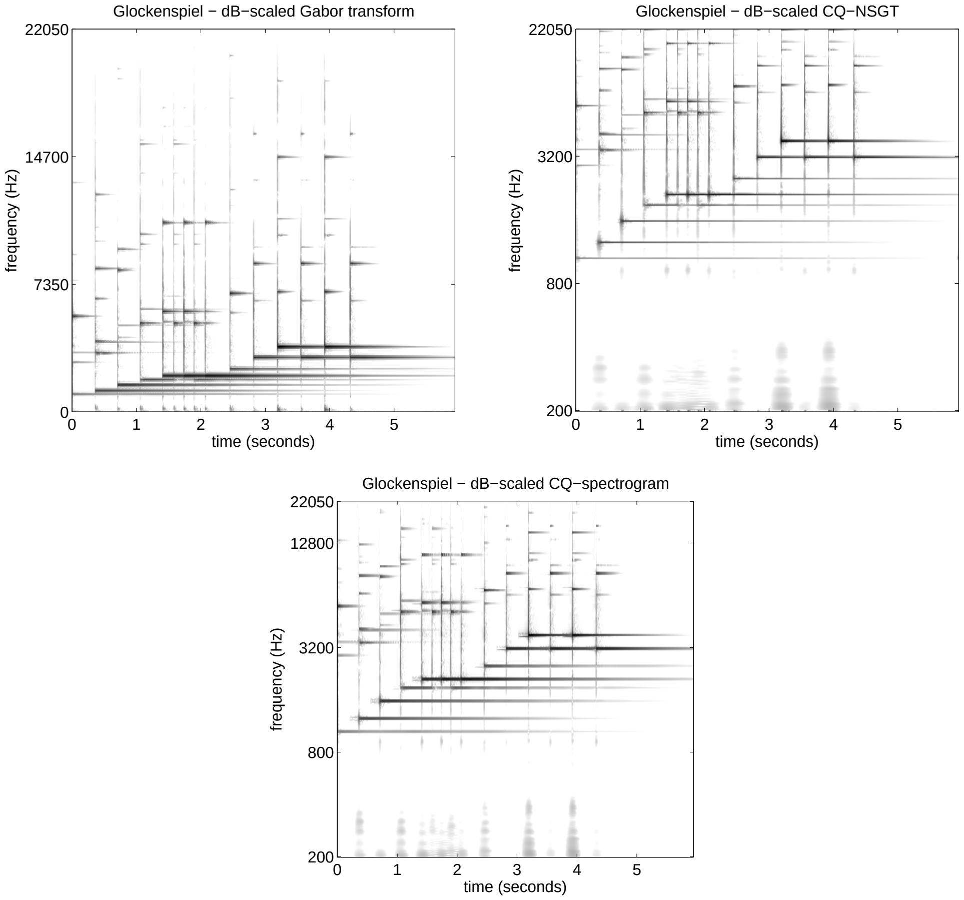


Figure 5.7: Glockenspiel (Example 1). Regular Gabor, constant-Q nonstationary Gabor and constant-Q representations of the signal. The transform parameters were $B = 48$ and $\xi_{\min} = 200$ Hz.

Note that we consider the bandwidth to be the support of the window in frequency. This makes sense in the considered painless case. Very often, see e.g. [141], the bandwidth is taken as the width between the points where the filter response drops to half of the maximum, i.e. the -3dB-bandwidth. This definition would also make sense in a non-compactly supported case. For the chosen filters, Hann windows, the Q-factor considering the -3dB-bandwidth is just double of the one considered above.

We see in Figure 5.7 the standard Gabor transform spectrogram and the constant-Q NSGT spectrogram of the Glockenspiel signal, the latter being very similar to the CQT spectrogram obtained from the original algorithm [26], but with the additional property that the signal can be perfectly reconstructed from the coefficients. Figures 5.8 and 5.9 compare the standard Gabor transform spec-

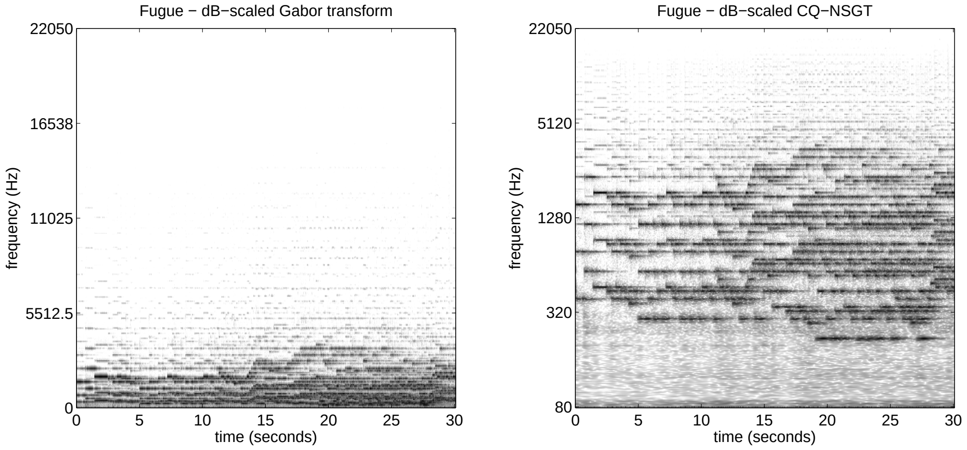


Figure 5.8: Bach's Little Fugue (Example 4). Regular and constant-Q nonstationary Gabor representations of the signal. The transform parameters were $B = 48$ and $\xi_{\min} = 75$ Hz.

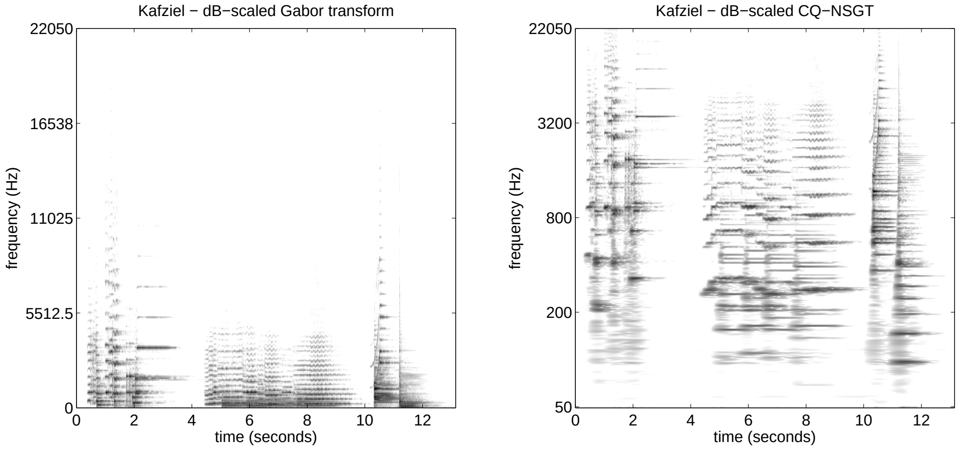


Figure 5.9: Violin and piano duet (Example 5). Regular and constant-Q nonstationary Gabor representations of the signal. The transform parameters were $B = 48$ and $\xi_{\min} = 50$ Hz.

trogram and the constant-Q NSGT spectrogram of two additional test signals, both sampled at 44.1 kHz:

- Example 4: A recording of Bach's Little Fugue in G Minor, BWV578 performed by Christopher Herrick on a pipe organ. Low frequency noise and the characteristic structure of pipe organ notes are resolved very well by a CQT. See Figure 5.8.

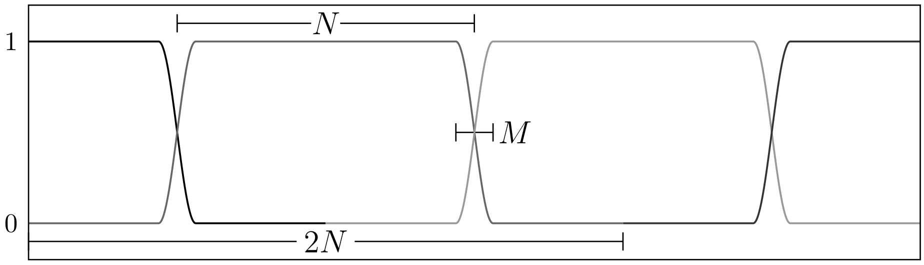


Figure 5.10: Tukey windows used in the slicing process. Note that the chosen amount of zero-padding leads to a half-overlap situation.

- Example 5: An excerpt from a duet between violin and piano. Written by John Zorn and performed by Sylvie Courvoisier and Mark Feldman, the sample is made up of three short segments: A frantic sequence of violin and piano notes, a slow violin melody with piano backing and an inharmonic part with chirp component. See Figure 5.9.

5.4.3 Real-time processing and the sliCQ

The CQ-NSGT implementation introduced in the previous sections a priori relies on a Fourier transform of the entire signal. This contradicts the idea of real-time applications, which require bounded delay in processing incoming samples and linear over-all complexity. These requirements can be satisfied by applying the CQ-NSGT in a blockwise manner, i.e. to (fixed length) slices of the input signal. However, the slicing process involves two important challenges: First, the windows h_m used for cutting the signal must be smooth and zero-padding has to be applied to suppress time aliasing and blocking artifacts when coefficient-modification occurs. Second, the coefficients issued from the blockwise transform should be equivalent to the CQ-coefficients obtained from a full-length CQ-NSGT. This can be achieved to high precision by careful choice of both the slicing windows h_m and the analysis windows g_m used in the CQ-NSGT.

Structure of the sliCQ transform

We now summarize the individual steps of the sliCQ algorithm and introduce the involved parameters.

I) Sliced constant-Q NSGT analysis:

1. Cut the signal $f \in \mathbb{C}^L$ into overlapping slices f^j of length $2N$ by multiplication with uniform translates of a slicing window h_0, centered at 0.

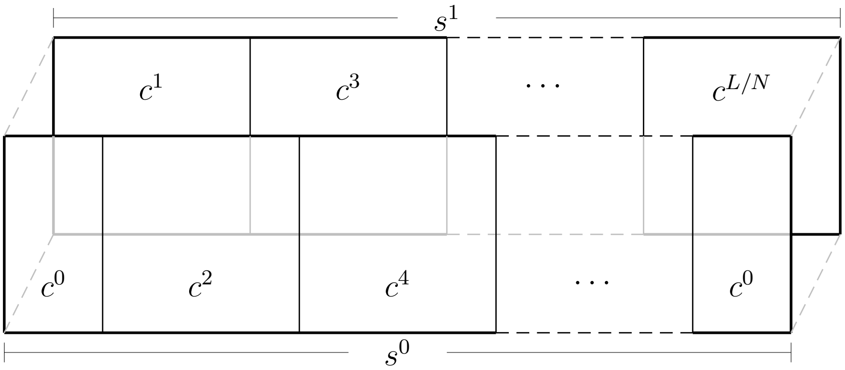


Figure 5.11: Structure of the sliCQ coefficients - schematic illustration

2. For each f^j, obtain coefficients $c^j \in \mathbb{C}^{2N/a_m \times |\Lambda_M|}$, by applying $\mathbf{NSGTF}_{2N}(f, \mathbf{g}, \mathbf{a})$ (Algorithm 5).
3. Due to the overlap of the slicing windows, cf. Figure 5.10, each time index is related to two consecutive slices. For visualization and processing, the slice coefficients c^j are re-arranged into a 2-layer array s, with $s := \{s^k\}_{k\in\{0,1\}} \in \mathbb{C}^{2\times N_m \times |\Lambda_M|}$, cf. Figure 5.11.

II) Sliced constant-Q NSGT synthesis:

1. Retrieve c^j by partitioning s.
2. Compute the dual frame $\breve{\mathcal{G}}(\tilde{\mathbf{g}}, \mathbf{a})$ for $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a})$ and, for all j, $\tilde{f}^j = \mathbf{iNSGTF}_{2N}(c^j, \tilde{\mathbf{g}}, \mathbf{a})$ (Algorithm 6).
3. Recover f by (windowed) overlap-add.

Note that L must be a multiple of $2N$; this is achieved by zero-padding, if necessary. By construction, the positions (n, m) of the coefficients in s^k reflect their time-frequency position with respect to the full-length signal, for $k = 0, 1$.

Computation of a sliced constant-Q NSGT

The *sliced constant-Q NSGT* (sliCQ) coefficients of f with respect to h_0 and $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a})$ and slice length $2N$ are obtained according to Algorithm 7.

Note that in this and the following algorithm, negative indices are used in a circular sense, with respect to the maximum admissible index, e.g. $f[-l] := f[L-l]$ or $s^k_{-n,m} := s^k_{N_m-n,m}$. As the CQ-NSGT analysis before, Algorithm 7 is complemented by a synthesis algorithm with similar structure, Algorithm 8, that synthesizes a signal $\breve{f}$ from a 2-layer coefficient array s.

The following proposition states that f is perfectly recovered from its sliCQ coefficients by applying Algorithm 8.

Algorithm 7 sliCQ analysis: $s = \mathbf{sliCQ}_{L,N}(f, h_0, \mathbf{g}, \mathbf{a})$

```
Initialize $f, h_0, g_m$ for all $m \in \Lambda_M$
$m \leftarrow 0$
for $m = 0, \dots L/N - 1$ do
    for $j = 0, \dots 2N - 1$ do
        $f^j[l] \leftarrow f\mathbf{T}_{jN}h_0[l + (j-1)N]$
    end for
    $c^j \leftarrow \mathbf{NSGTF}_{2N}(f, \mathbf{g}, \mathbf{a})$
    $l \leftarrow (j \bmod 2)$
    for $m \in \Lambda_M,\ n^s = 0, \dots, 2N/a_m - 1$ do
        $s^k_{n^s+(j-1)N/a_m,m} \leftarrow c^j_{n^s,m}$
    end for
end for
```

Proposition 16. *Let* $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a})$ *and* $\breve{\mathcal{G}}(\tilde{\mathbf{g}}, \mathbf{a})$ *be dual NSG systems for* $\mathbb{C}^{2N}$*. Further let* $h_0, \tilde{h}_0 \in \mathbb{C}^L$ *satisfy*

$$\sum_{j=0}^{L/N-1} \mathbf{T}_{jN}\left(h_0\overline{\tilde{h}_0}\right) \equiv 1. \tag{5.16}$$

If s *is the output of* $\mathbf{sliCQ}_{L,N}(f, h_0, \mathbf{g}, \mathbf{a})$ *(Algorithm 7), then the output* $\tilde{f}$ *of* $\mathbf{isliCQ}_{L,N}(s, \tilde{h}_0, \tilde{\mathbf{g}}, \mathbf{a})$ *(Algorithm 8) equals* f*, i.e.,* $\tilde{f} = f$*.*

Proof. According to Proposition 15, $\tilde{f}^j$, the output of **iNSGTF** in Step 9 of Algorithm 8 satisfies to $f^j[l] = (f \cdot \mathbf{T}_{jN}h_0)[l+(j-1)N]$. Since $\sum_j \mathbf{T}_{jN}\left(h_0\overline{\tilde{h}_0}\right) \equiv 1$ holds,

$$\tilde{f} = \sum_j (f \cdot \mathbf{T}_{jN}h_0)\mathbf{T}_{jN}\overline{\tilde{h}_0} = f \cdot \sum_j \mathbf{T}_{jN}\left(h_0\overline{\tilde{h}_0}\right) = f$$

follows. □

The relation between CQ-NSGT and sliCQ

To maintain perfect reconstruction in the final overlap-add step in Algorithm 8, we assume

$$h_j = \mathbf{T}_{jN}h_0, \text{ with } \sum_{j=0}^{L/N-1} h_j \equiv 1, \tag{5.17}$$

and use a dual window $\tilde{h}_0$ satisfying (5.16) in the synthesis process.

Another obvious option for the design of the slicing windows is to require $\sum_j h_j^2 \equiv 1$, which would allow for using the same windows in the final overlap-add

Algorithm 8 sliCQ synthesis: $\tilde{f} = \mathbf{isliCQ}_{L,N}(s, \tilde{h}_0, \tilde{\mathbf{g}}, \mathbf{a})$

```
**Initialize** $s, \tilde{h}_0, \tilde{g}_m$ for all $m \in \Lambda_M$
$m \leftarrow 0$
$\tilde{f} \leftarrow \mathbf{0}_L$
**for** $m = 0, \dots L/N - 1$ **do**
    $k \leftarrow (j \bmod 2)$
    **for** $m \in \Lambda_M,\ n^s = 0, \dots, 2N/a_m - 1$ **do**
        $c^j_{n^s,m} \leftarrow s^k_{n^s+(j-1)N/a_m,m}$
    **end for**
    $\tilde{f^j} \leftarrow \mathbf{iNSGTF}_{2N}(c^j, \tilde{\mathbf{g}}, \mathbf{a})$
    **for** $j = 0, \dots 2N - 1$ **do**
        $\tilde{f}[l + (j-1)N] \leftarrow \tilde{f}[l + (j-1)N] + \tilde{f^j}[l]\tilde{h}_0[l - N]$
    **end for**
**end for**
```

step. However, if we want to approximate the true CQ-coefficients as obtained from a full-length transform, (5.17) is the more favorable condition.

In our implementation, *slicing* of the signal is accomplished by a uniform partition of unity constructed from a Tukey window h_0 with essential length N and transition areas of length M, for some $N, M \in \mathbb{N}$ with $M < N$ (usually $M \ll N$). The slicing windows are symmetrically zero-padded to length $2N$, reducing time aliasing significantly. The uniform partition condition (5.17) leads to close approximation of the full-length CQ-NSGT by sliCQ. This correspondence between the sliCQ and the corresponding full-length CQ-NSGT is made explicit in the following proposition.

Proposition 17. *Let $\breve{\mathcal{G}}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ be a nonstationary Gabor system for $\mathbb{C}^L$. Further, let $h_0 \in \mathbb{C}^L$ be such that* (5.17) *holds and define $g_m \in \mathbb{C}^{2N}$, for all $m \in \Lambda_M$ by*

$$g_m[l] = g^{\mathcal{L}}_m[lL/(2N)].$$

For $f \in \mathbb{C}^L$, denote by $c \in \mathbb{C}^{N_m \times |\Lambda_M|}$ the CQ-NSGT coefficients of f with respect to $\breve{\mathcal{G}}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ and by $s \in \mathbb{C}^{2 \times N_m \times |\Lambda_M|}$ the sliCQ coefficients of f with respect to h_0 and $\breve{\mathcal{G}}(\mathbf{g}, \mathbf{a})$. Then

$$\begin{aligned} &|s^0_{n,m} + s^1_{n,m} - c_{n,m}| \\ &\leq \|f\| \Big(\|(1 - h_0 - h_1)\mathbf{T}_{n^s a_m} \breve{g^{\mathcal{L}}_m}\| \\ &+ \|(h_0 + h_1) \sum_{k=1}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_m + 2kN} \breve{g^{\mathcal{L}}_m}\| \Big) \end{aligned} \tag{5.18}$$

for $n = jN/a_m + n^s$*, with* $j = 0, \dots, L/N - 1$ *and* $n^s = 0, \dots, N/a_m - 1$.

Proof. Since g_m is obtained by sampling $g_m^{\mathcal{L}}$ with sampling period $L/2N$, the (inverse) Fourier transform $\breve{g_m}$ of g_m is given by periodization of $g_m^{\mathcal{L}}$ as follows:

$$\breve{g_m}[l] = \sum_{k=0}^{\frac{L}{2N}-1} \breve{g_m^{\mathcal{L}}}[l + k \cdot 2N]. \tag{5.19}$$

Recall from (5.12) that the CQ-NSGT coefficients of f with respect to $\mathcal{G}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ are given by $c_{n,m} = \langle f, \mathbf{T}_{na_m}\breve{g_m^{\mathcal{L}}}\rangle$, while the CQ-NSGT coefficients c^j of f^j are, for $j = 0, \dots, L/N - 1$, $n^s = 0, \dots, \frac{2N}{a_m} - 1$ and $m \in \Lambda_M$

$$\begin{aligned} c^j_{n^s,m} &= \langle \widehat{f^j}, g_{n^s,m}\rangle = \langle \widehat{f^j}, \mathbf{M}_{-n^s a_m} g_m\rangle \\ &= \langle f^j, \mathbf{T}_{n^s a_m}\breve{g_m}\rangle \\ &= \left\langle f, h_m \sum_{k=0}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_m + (j-1+2k)N}\breve{g_m^{\mathcal{L}}} \right\rangle, \end{aligned} \tag{5.20}$$

where the final inner product is taken over $\mathbb{C}^L$. Observe that every $n = 0, \dots, N_m - 1$ can be written as $n = j\frac{N}{a_m} + n^s$ with n^s from $0, \dots, \frac{N}{a_m} - 1$ and thus

$$\begin{aligned} s^0_{n,m} + s^1_{n,m} &= c^j_{n^s + N/a_m, m} + c^{j+1}_{n^s,m} \\ &= \left\langle f, (h_j + h_{j+1}) \sum_{k=0}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_m + (j+2k)N}\breve{g_m^{\mathcal{L}}} \right\rangle \\ &= \left\langle f, \mathbf{T}_{n^s a_m + jN}\breve{g_m^{\mathcal{L}}} \right\rangle + R[n] \\ &= \left\langle f, \mathbf{T}_{a_m(\frac{jN}{a_m} + n^s)}\breve{g_m^{\mathcal{L}}} \right\rangle + R[n] \\ &= c_{n,m} + R[n]. \end{aligned} \tag{5.21}$$

Here,

$$\begin{aligned} R[n] = &\left\langle f, (h_j + h_{j+1} - 1)\mathbf{T}_{n^s a_m + jN}\breve{g_m^{\mathcal{L}}} \right\rangle \\ &+ \left\langle f, (h_j + h_{j+1}) \sum_{k=1}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_m + (j+2k)N}\breve{g_m^{\mathcal{L}}} \right\rangle. \end{aligned} \tag{5.22}$$

Hence $s^0_{n,m} + s^1_{n,m} - c_{n,m} = R[n]$. The result follows from Cauchy-Schwartz' inequality, applied to the case $j = 0$, observing independence from j. □

Remark 21. *In practice, $\breve{g}_m^{\mathcal{L}}$ is chosen such that the translates $\mathbf{T}_{na_m}\breve{g}_m^{\mathcal{L}}$ are* essentially concentrated *in*

$$I_{N,M} = \left[-\frac{N-M}{2}, N + \frac{N-M}{2} \right],$$

i.e. $\|\mathbf{T}_{na_m}\breve{g}_m^{\mathcal{L}}\chi_{\mathbb{R}\setminus I_{N,M}}\| \ll \|\mathbf{T}_{na_m}\breve{g}_m^{\mathcal{L}}\|$, *for all* $n = 0, \ldots, N/a_m - 1$. *Therefore, the value of* (5.18) *is negligibly small. Numerical evaluation of the approximation quality is given in Section 5.4.4.*

As a consequence of the previous proposition, we define the *sliCQ spectrogram* as $|s^0 + s^1|^2$ and propose to simultaneously treat $s^0_{n,m}$ and $s^1_{n,m}$, corresponding to the same time-frequency position, when processing the coefficients.

5.4.4 Numerical analysis and simulations

In this section we treat the computational complexity of CQ-NSGT and sliCQ and how they compare to one another. We show that despite superlinear complexity, CQ-NSGT outperforms state-of-the-art implementations of the classical constant-Q transform. Since sliCQ is a linear cost algorithm, it further improves the efficiency of the CQ-NSGT for sufficiently long signals. Section 5.4.4 provides experimental results confirming the good approximation of CQ-NSGT by the corresponding sliCQ coefficients, cf. Proposition 17.

The NSG Toolbox (for MATLAB) and CQ-NSG Toolbox (for Python) used in this section are available at `http://www.univie.ac.at/nonstatgab/slicq`, alongside extended experimental results complementing those presented in Section 5.4.5.

Computational complexity:

We assume the number of filters $|\Lambda_M|$ in the CQ-NSGT to be independent of the signal length L and Proposition 14 to hold, in particular $N_m \geq L_m$. The support size L_m of each filter g_m depends on L. Hence, the number of operations for Algorithm 5 is as follows:

$$\mathcal{O}\Big(\underbrace{L \ \log(L)}_{\mathbf{FFT}_L} + \sum_{m \in \Lambda_M} \underbrace{N_m \ \log(N_m)}_{\mathbf{IFFT}_{N_m}} + \underbrace{L_m}_{f \cdot \overline{g_m}} \Big).$$

With L_m and N_m bounded by L, this can be simplified to $\mathcal{O}(L \ \log(L))$.

The computation of the dual frame involves inversion of the multiplication operator $\mathbf{S}$ and applying the resulting operator $\mathbf{S}^{-1}$ to each filter. This results in

$\mathcal{O}(2\sum_{m\in\Lambda_M} L_m) = \mathcal{O}(L)$ operations, where the support of the g_m was taken into account.

Complexity of Algorithm 6 can be derived to be $\mathcal{O}(L\ \log(L))$, analogous to Algorithm 5.

For $\mathbf{sliCQ}_{L,N}$ (Algorithm 7), we assume the slice length $2N$ to be independent of L, resulting in a computational complexity of

$$\mathcal{O}\Big(\underbrace{L/N}_{\#\text{slices}}\cdot\big(\underbrace{2N\ \log(2N)}_{\mathbf{NSGTF}_{2N}}+\underbrace{2N}_{f\cdot\mathbf{T}_{mN}\overline{h_0}}\big)\Big)=\mathcal{O}(L).$$

Both the dual frame and $\tilde{h}_0$ can be precomputed independent of L, whilst Algorithm 8 is of complexity $\mathcal{O}(L)$, analogous to Algorithm 7.

Efficiency of CQ-NSGT:

Technical framework: The simulations presented in this section were performed in MATLAB R2009b on a 3 Gigahertz Intel Core 2 Duo machine with 2 GB RAM. The constant-Q transforms were computed using the code published with [141], available for free download at `http://www.elec.qmul.ac.uk/people/anssik/cqt/`. The constant-Q nonstationary Gabor transform (CQ-NSGT) algorithms are available at `http://univie.ac.at/nonstatgab/`.

The computation time of the nonstationary Gabor transform was found to be better than a recent fast CQT implementation [141], as seen in Figure 5.12. The two plots show mean values for computation time in seconds and the corresponding variance over 50 iterations, with varying window lengths and number of frequency bins, respectively. The outlier, drawn in gray, in Figure 5.12 (left) at the prime number 600569 illustrates dependence of the current CQ-NSGT implementation on the signal length's prime factor structure, analogous to FFT.

Efficiency of sliCQ:

The following computation time experiments were performed in the same framework as before but using MATLAB R2011a.

Figure 5.13(left) reproduces and extends some of the results shown in Figure 5.12; it shows, for both the constant-Q implementation provided in [141] and CQ-NSGT, mean computation duration and variance for analysis followed by reconstruction, against signal length. The plot also illustrates the dependence of CQ-NSGT on the prime factor decomposition of the signal length L.

Figure 5.13(right) illustrates the performance of sliCQ compared to the constant-Q and CQ-NSGT algorithms shown in Figure 5.13(left). Linearity of the sliCQ

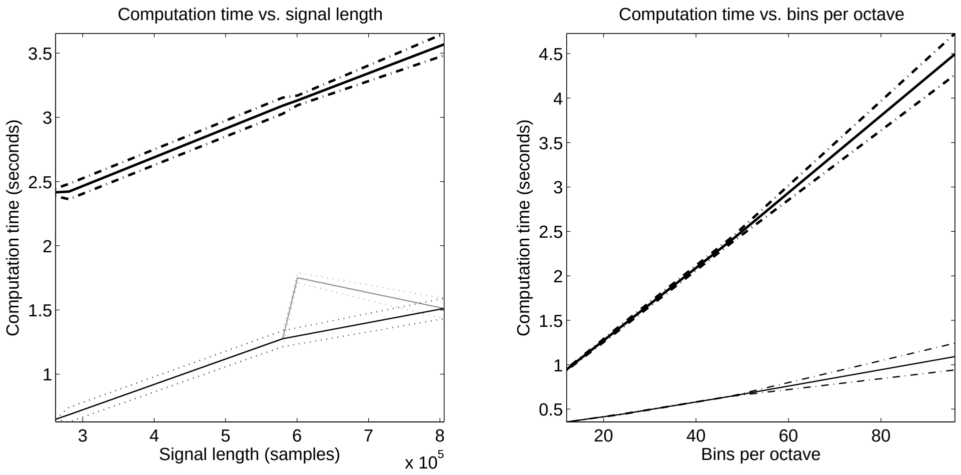


Figure 5.12: Comparison of computation time of CQT (top curves) and NSGT (bottom curves). The figure on the left shows the computation times for signals of various lengths with the number of bins per octave fixed at $B = 48$, while the figure on the right shows the computation times for the Glockenspiel signal, varying the number of bins per octave. In both figures, the solid lines represent the mean time (in seconds) and the dashed or dotted lines signify the mean time with corresponding variance. The lower left curve also shows gray solid lines indicating an outlier. The minimum frequency for all cases $\xi_{\min}$ was chosen at 50 Hz.

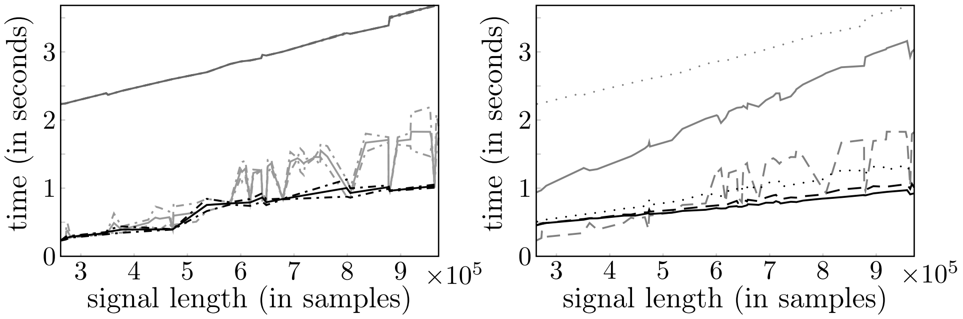


Figure 5.13: (left) Computation time versus signal length of the CQ transform (dark gray) and CQ-NSGT. For the CQ-NSGT we show separate graphs including (light gray), respectively neglecting prime signal lengths (black). Graphs show the mean performance (solid) and variance (dashed) over 50 iterations. (right) Computation time versus signal length of the CQ transform (dotted gray), CQ-NSGT (dashed gray) and various sliCQ transforms. The sliCQ transforms were taken with slice lengths 4096 (solid gray), 16384 (dotted black), 32768 (dashed black) and 65536 (solid black) samples.

algorithm becomes evident, with deviations occurring due to unfavorable FFT

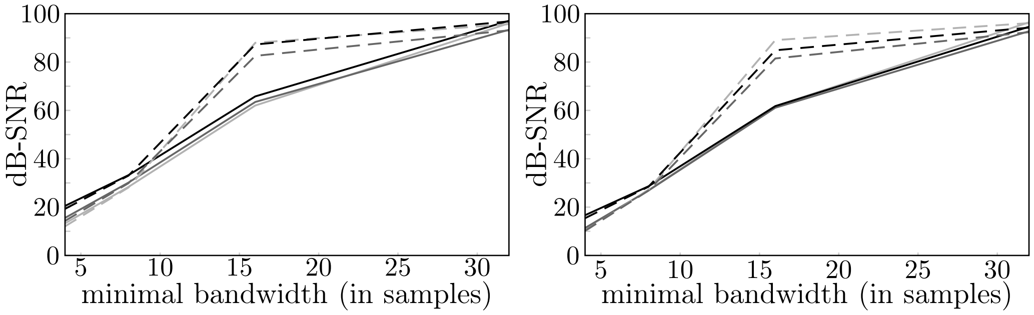


Figure 5.14: SliCQ coefficient approximation error against the minimal admissible bandwidth for Set 1 (top) and Set 2 (bottom). All transforms use Blackman-Harris windows in the CQ-NSGT step. Solid and dashed lines represent long (1/4 slice length) and short (1/128 slice length) transition areas respectively, while colors correspond to the slice length: 4096 (light gray), 16384 (dark gray) and 65536 samples (black).

lengths $2N/a_m$ in $\mathbf{(i)NSGTF}_{2N}$. Performance improvements for increasing slice length can be attributed to the advanced nature of MATLAB's internal FFT algorithm, as compared to the current implementation of the sliCQ framework.

The performance of the involved algorithms does not depend on signal content. Consequently, random signals were used in the performance experiments, although we implicitly assumed the signals to be sampled at 44.1 kHz. All the results represent transforms with 48 bins per octave, minimum frequency 50 Hz and maximum frequency 22 kHz, in Section 5.4.5 a maximum frequency of 20 kHz is used instead. Results for other parameter values do not differ drastically and are omitted.

Approximation properties

To verify the approximate equivalence of the sliCQ coefficients to those of a full-length CQ-NSGT and thus to a constant-Q transform, we computed the norm difference between $s^0 + s^1$ and c as in Proposition 17, for two sets of fundamentally different signals. Set 1 contains 50 random, complex-valued signals of 2^{20} samples length, while Set 2 consists of 90 music samples of the same length, sampled at 44.1 kHz each, covering pop, rock, jazz and classical genres. The signals of the second set are well-structured and often well-concentrated in the time-frequency plane, characteristics that the first set lacks completely.

For discretization reasons as well as to achieve good concentration of $\breve{g}_m^{\mathcal{L}}$ in Proposition 17, sliCQ implementations must impose a lower bound on the length of g_m. Approximation results for various lower bounds on the filter length are summarized in Figure 5.14, showing the mean approximation quality over the whole set.

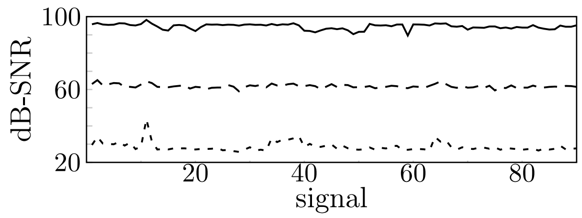


Figure 5.15: Coefficient approximation error (5.18) for all signals from Set 2 and slice and transition length of 65536, resp. 16384 samples. Line style indicates the minimal bandwidth: 8 (dotted), 16 (dashed) and 32 (solid) samples.

All errors are given in signal-to-noise ratio, scaled in dB:

$$20 \log_{10} \frac{\|c\|}{\|c - (s^0 + s^1)\|}$$

Figure 5.14 shows that, independent of other parameters, a minimal filter length smaller than 8 samples leads to a representation that is visibly different from, while values above 16 samples yield coefficients that are largely equivalent to those of a constant-Q transform. We can see that the slice length itself has rather small influence on the results, while the interplay of slicing window shape, specified by the ratio of transition area length to slice length, and minimal filter length is illustrated nicely; remarkably, this ratio influences the approximation quality mainly for moderately well localized filters. This is in correspondence with the characterization given in (5.18): the *circular overspill*, given by the second term of the right hand side in (5.18), depends on the shape and support of the sum of two adjacent slicing windows, in particular for moderately well localized filters. If the windows are very well localized, the overspill is small independent of the particular shape of the slicing area. On the other hand, very badly localized windows make the distinct influence of the slicing windows negligible. Finally, a comparison of the top and bottom graphs in Figure 5.14 shows that the approximation quality is largely independent of the signal class. For Set 1 the variance is generally negligible (< 0.1 dB) and was omitted. Despite some outliers in Set 2, we have found the approximation quality to depend on the minimal filter length in a stable way, cf. Figure 5.15. These outliers can be attributed to signals particularly sparse (smaller error) or dense (larger error) in low frequency regions, where $\breve{g}_m^{\mathcal{L}}$ is least concentrated.

5.4.5 Experiments on applications

Our experiments show applications of the CQ-NSGT and sliCQ in musical contexts, where the property of a logarithmic frequency scale renders the method

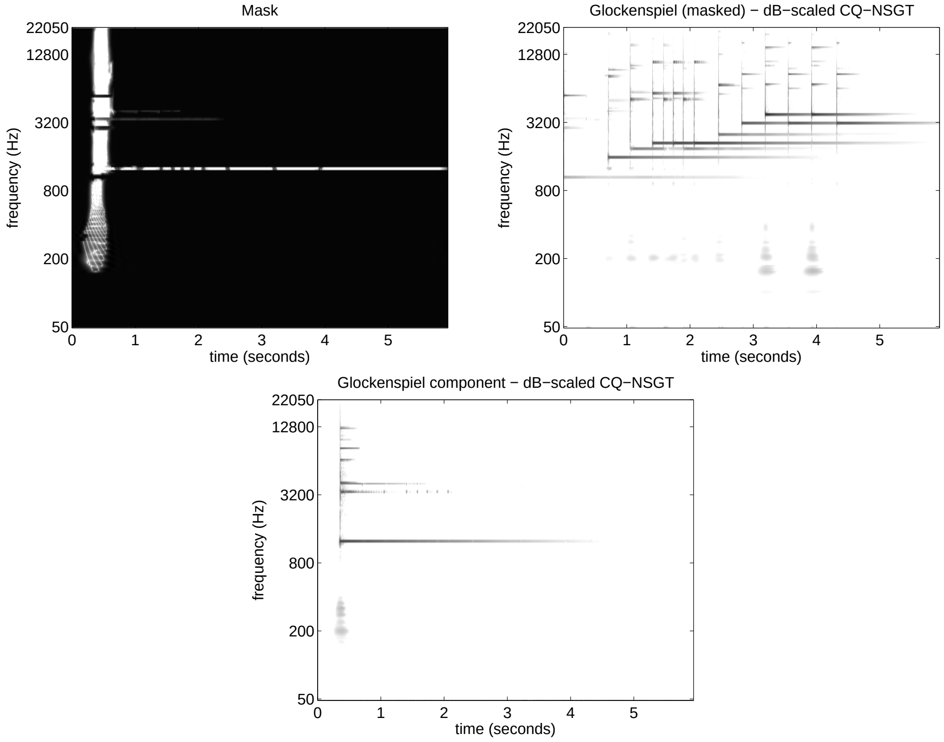


Figure 5.16: Note extraction from the Glockenspiel signal by masking. The CQ-NSGT coefficients of the Glockenspiel signal were weighted with the mask shown on top. The remaining signal and extracted component are depicted in the middle and bottom respectively. The transform parameters were $B = 24$ and $\xi_{\min} = 50$ Hz.

often superior to the traditional STFT. Sound examples for the first two experiments can be found at `http://univie.ac.at/nonstatgab/cqt/`, while those for the final experiment are provided at `http://univie.ac.at/nonstatgab/slicq/`.

Masking

In the masking experiment, we show that the perfect reconstruction property of CQ-NSGT can be used to cut out components from a signal by directly modifying the time-frequency coefficients. The advantage of considerably higher spectral resolution at low frequencies (with a chosen application-specific temporal resolution at higher frequencies) compared to the STFT, makes the CQ-NSGT a very powerful, novel tool for masking or isolating time-frequency components of musical signals. Our example shows in Figure 5.16 a mask for extracting – or inversely, suppressing – a note from the Glockenspiel signal depicted in Figure 5.7. The mask was created as a gray-scale bitmap using an ordinary image manipulation program and then resampled in order to conform to the irregular time-frequency

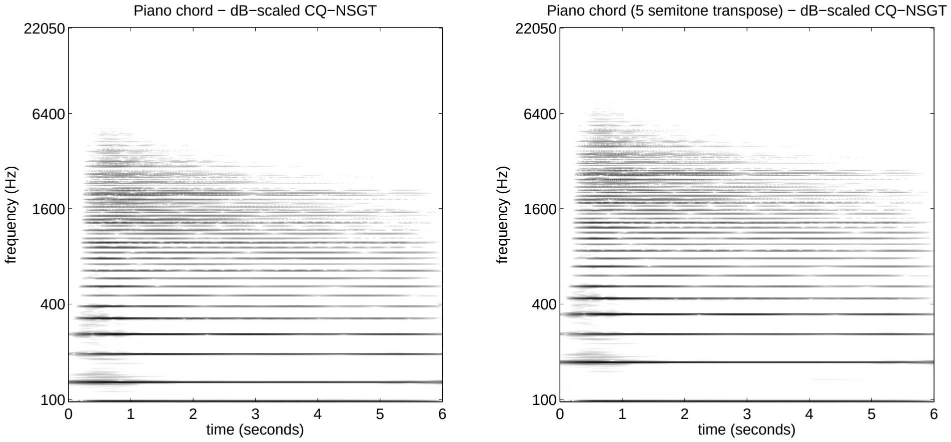


Figure 5.17: Piano chord signal and upwards transposition by 5 semitones, corresponding to a circular shift of the inner bins by 20. The transform parameters were $B = 48$ and $\xi_{\min} = 100$ Hz.

grid of the CQ-NSGT. Figure 5.16 shows the masked spectrogram, along with the spectrograms of the synthesized, processed signal and remainder.

Transposition

An interesting property of continuous constant-Q decompositions is the fact the transposition of a harmonic structure, like a note including overtones, corresponds to a simple translation of the logarithmically scaled spectrum. Approximately, this is also the case for the finite, discrete CQ-NSGT. In this experiment, we transposed a piano chord simply by shifting the inner frequency bins accordingly. By inner frequency bins, we refer to all bins with constant Q-factor. This excludes the 0-frequency and Nyquist frequency bins, as well as their direct neighbors. The onset portion of the signal has been damped, since inharmonic components, such as transients, produce audible artifacts when handled in this way. In Figure 5.17, we show spectrograms of the original and modified chords, shifted by 20 bins. This corresponds to a upwards transposition by 5 semitones.

Transposing a component

The previous experiments show how the CQ-NSGT can be applied in the processing of signals taking advantage of the logarithmic frequency scaling and the perfect reconstruction property. In particular, the transposition of a harmonic structure amounted to just a translation of the spectrum along frequency bins, while the masking of the CQ-NSGT coefficients allowed for the extraction or suppression

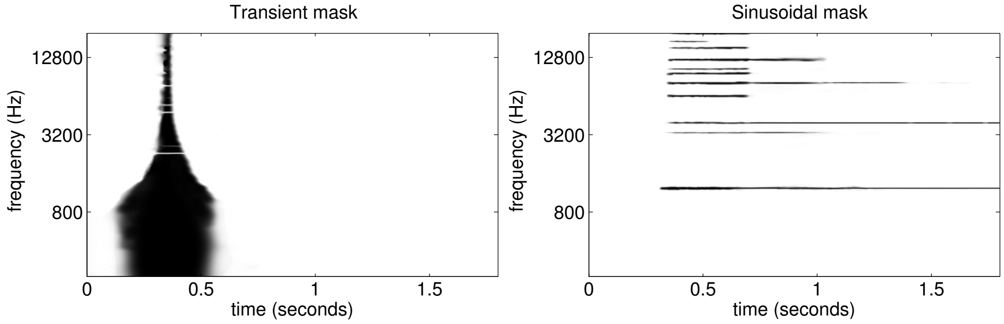


Figure 5.18: Masks for extracting a transient (left) and sinusoidal component (right) of the Glockenspiel signal. The gray level plot describes the amplitude of the mask, with black and white representing 1 and 0, respectively.

of a component of the signal. In the following experiment, we show that the two procedures can be combined to modify a portion of a signal.

The sound files for this and other transposition experiments are available at `http://www.univie.ac.at/nonstatgab/slicq`. A script for the Python toolbox that executes the experiment, is available on the same page.

Figure 5.18 shows masks for isolating a transient part and the corresponding sinusoidal part of a Glockenspiel signal, created using an ordinary image manipulation program. Therein, the layers paradigm has been used to be able to quickly switch on and off the masks in order to accurately adapt them to the CQ-NSGT representation of the audio. An "inverse mask" is also constructed for the remainder part of the signal, essentially decomposing the signal into transient, sinusoidal and background portions. The masks have been drawn in the logarithmic domain, to be able to handle the dynamics of the audio. They are linearly scaled in dB units, so that 0 in the mask corresponds to 10^{-5} (-100 dB) and 1 corresponds to 1 (0 dB).

While keeping the transient part, the isolated sinusoidal component of the signal is transposed upward by 2 semitones, corresponding to 8 frequency bins. The transient, the remainder, and the modified sinusoidal coefficients are then added and the inverse transform is applied to obtain the resulting processed signal. For ease of use, this process is done with a rectangular representation of the slices, obtained by choosing N_m constant for all frequency bands which corresponds to a sinc-interpolation of the coefficients.

Figure 5.19 compares the CQ-NSGT spectrograms of the original and the modified signal, while Figure 5.20 shows the results for the same experiment using sliCQ transforms with different slice lengths. Note that the plots show the spectrogram of the synthesized signal, not the time-frequency coefficients before synthesis. Fur-

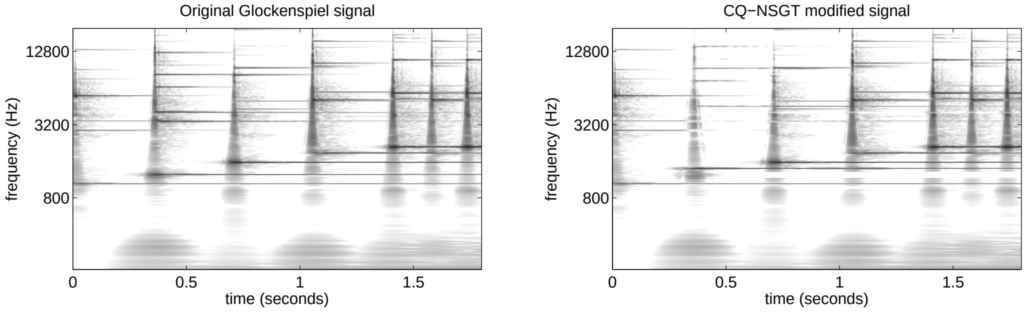


Figure 5.19: CQ-NSGT spectrograms showing an excerpt of the Glockenspiel signal before (left) and after transposition of a component (right).

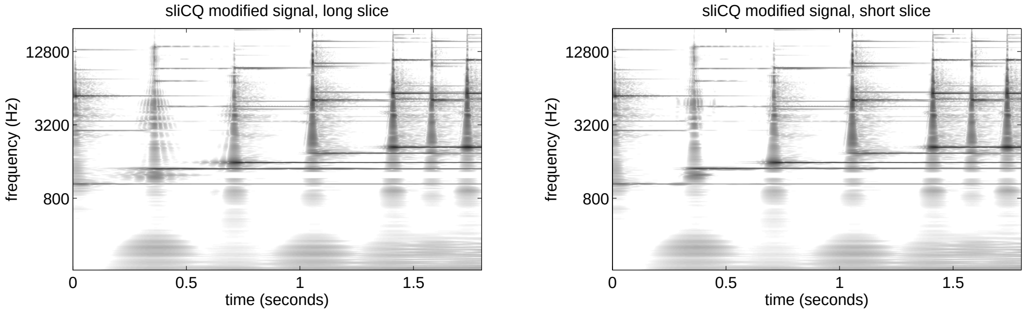


Figure 5.20: sliCQ spectrograms showing an excerpt of the Glockenspiel signal after transposition of a component. The left plot was done with a slice length of 50000 and a transition area of 20000 samples, the right plot with a slice length of 5000 and a transition area of 2000 samples.

ther, the exact same mask was used for CQ-NSGT and sliCQ transpositions.

5.5 Extensions and further implementations

5.5.1 An auditory-based transform and iterative reconstruction

The peripheral auditory system can be modeled in a first approximation as a bank of bandpass filters whose bandwidth corresponds to the spectral resolution of the ear. Many psychoacoustical studies have focused on the characterization of these “auditory filters”(see [120] for a review). The filters are commonly described by their equivalent rectangular bandwidth (ERB). The ERB (in Hz) of the auditory filter centered at frequency F (in Hz) is [79]

$$ERB(F) = 24.7 + \frac{F}{9.265}. \tag{5.23}$$

Eq. (5.23) indicates that the auditory frequency resolution as described by the ERB is approximately constant-Q only at high frequencies ($>$ 2 kHz). For the full range of audible frequencies (.02–20 kHz) the ERBs range from 27 Hz to 2.2 kHz. Using ERB units, the range of audible frequencies can be discretized as a bank of 39 adjacent filters whose ERB number is [79]

$$ERB_{\text{num}}(F) = 9.265 \ln \left(1 + \frac{F}{228.8455}\right) \tag{5.24}$$

and, reciprocally, $F = u(E_F) = 228.8455 \left(e^{(E_F/9.265)} - 1\right)$. Eq. (5.24) corresponds to the ERB scale used to plot psychoacoustical data on a perceptual frequency axis. The partition of the frequency axis into filters leads to a partition of the time axis into time windows whose widths correspond to the temporal resolution at a certain frequency. In [162] the windows' shape was estimated using Gaussian stimuli with various spectro-temporal shapes. The results indicated that the spectral width of one window corresponds to one ERB and the temporal width approximately corresponds to four periods of the carrier frequency, *e.g.*, 4 ms at 1 kHz. Overall, the data in [162] suggests that the auditory system performs a TF analysis using its own "internal" windows that are well approximated by Gaussians with frequency dependent spectro-temporal resolution.

Auditory-based TF representations

To date, two general approaches exist to achieve a perceptually motivated TF representation of an audio signal. The first approach includes models of auditory processing like in [113, 119]. Such models attempt to replicate the various stages of auditory processing and are useful to improve our knowledge about the auditory system. However, they feature many parameters, they are computationally demanding and *not invertible*. Approximately invertible models were proposed in, *e.g.*, [74, 128] as integrated audio coders. Consequently, the signal representation is not easily accessible. Overall, auditory models do not constitute TF analysis-synthesis tools. The second approach includes TF transforms tuned to mimic the auditory TF resolution (see Sec. 5.5.1). Wavelet and constant-Q transforms are used [4,127,164] in this context, but they mismatch the auditory spectral resolution at low frequencies. Further developments include a bilinear transform [124], linear [90,149] and nonlinear gammatone filterbanks [94], and auditory-based nonuniform filterbanks [19,46]. They approximate the auditory TF resolution nicely but fail at providing perfect reconstruction.

Analysis and dual windows: ERBlets

The ERBlet transform consists of filters $g_m, m = 0, \dots, M$, that are Gaussian windows constructed in the frequency domain according to

$$g_m[l] = \Omega_m^{-\frac{1}{2}} e^{-\pi\left[\frac{l-\xi_m}{\Omega_m}\right]^2} \tag{5.25}$$

where $l \in \mathbb{Z}$ is the discrete frequency variable, ξ is the center frequency (in Hz), and Ω is a shape factor that controls the effective support (bandwidth) of g (in Hz). Let $f_{\min}$ and $f_{\max}$ denote the minimum and maximum analysis frequencies, respectively. Their corresponding ERB numbers are (Eq. (5.24)) E_0 and E_M. Linearly distributing $M + 1$ filters from E_0 to E_M with a density of V filters per ERB leads to $E_m = E_0 + m/V$ with $M = V\,(E_M - E_0)$. Then $\xi_m = u(E_m)$ and $\Omega_m = ERB(\xi_m)$. The factor $\Omega_m^{-\frac{1}{2}}$ in Eq. (5.25) ensures that all filters have the same energy. Although Gaussians are not compactly-supported windows, they decay very fast. Thus, by truncating the filters so that $\mathrm{supp}(g_m) = \lceil 4\,\Omega_m \rceil$, the filters are close to zero at the borders. Finally, a_m and V can be chosen such that the frame operator associated with the ERBlets is invertible, see Section 5.2.2.

Implementation

The ERBlet algorithms and scripts are available at `http://www.kfs.oeaw.ac.at/ICASSP2013_ERBlets`. This address is referred to below as the "webpage". The algorithms are also available in the NSG Toolbox.

To process the positive and negative frequencies the $M + 1$ filters are mirrored to the negative frequency domain (note that if $f_{\min}$ and $f_{\max}$ are set at the 0 and Nyquist frequencies, respectively, then only $M - 1$ filters are mirrored). The ERBlet transform is determined by the two parameters a_m and V that provide control over the resolution and redundancy of the transform, $red = \sum_{m=-M}^{M} a_m^{-1}$. The number of time samples in each channel is given by $N_m = \lceil N_m \rceil$. Choosing a_m such that $N_m \geq \mathrm{supp}(g_m)$ results in a painless system (see Sec. 5.2.2). Otherwise $\hat{\mathbf{S}}$ is not diagonal and iterative algorithms are required for efficient inversion. To do so, we note that $\hat{\mathbf{S}}_{\check{\mathcal{G}}(\mathbf{g},\mathbf{a})} := \mathcal{F}\mathbf{S}_{\check{\mathcal{G}}(\mathbf{g},\mathbf{a})}\mathcal{F}^{-1} = \mathbf{S}_{\mathcal{G}(\mathbf{g},\mathbf{a})}$ and use the equality

$$\sum_{n,m} c_{n,m}\breve{\widetilde{g_{n,m}}} = \sum_{n,m} c_{n,m}\mathbf{S}^{-1}_{\check{\mathcal{G}}(\mathbf{g},\mathbf{a})} g_{n,m} = \mathcal{F}^{-1}\mathbf{S}^{-1}_{\mathcal{G}(\mathbf{g},\mathbf{a})} \sum_{n,m} c_{n,m} g_{n,m} \tag{5.26}$$

to solve the linear system

$$\mathbf{S}x = \sum_{n,m} c_{n,m}\,\breve{g_{n,m}} \tag{5.27}$$

with an adapted *conjugate gradients* (CG) algorithm [82,155], where the right-hand sum in Eq. (5.27) is computed by $\mathbf{NSIGTF}_L$ (Algorithm 6). If $\check{\mathcal{G}}(\mathbf{g}, \mathbf{a})$ is a frame, then so is $\mathcal{G}(\mathbf{g}, \mathbf{a})$ and $\hat{\mathbf{S}}$ is self-adjoint, leading to convergence of CG to the desired solution. The convergence speed depends on the condition number $\kappa(\hat{\mathbf{S}})$ (*i.e.*, the frame bound ratio B/A [155]) and can be improved with a preconditioning step [11]. If the $\widehat{g_m}$ decay fast enough then $\hat{\mathbf{S}}$ is diagonal dominant and the matrix

$$\mathbf{D}(\hat{\mathbf{S}})^{-1}_{m,l} = \begin{cases} \left(\sum N_m |\hat{g}_m|^2\right)^{-1}[m], & \text{if } m = l \\ 0, & \text{else} \end{cases} \tag{5.28}$$

is an efficient preconditioner. Because CG works for self-adjoint matrices only and applying $\mathbf{D}(\hat{\mathbf{S}})^{-1}$ to $\hat{\mathbf{S}}$ does not result in a self-adjoint matrix, we use

$$\mathbf{D}(\hat{\mathbf{S}})^{-1/2}\hat{\mathbf{S}}\mathbf{D}(\hat{\mathbf{S}})^{-1/2}$$

instead. Since applying $\hat{\mathbf{S}}$ to a signal f is equivalent to performing analysis followed by synthesis with the time-side NSG system $\mathcal{G}(\mathbf{g}, \mathbf{a})$, we can use $\mathbf{NSGT}_L$ (Algorithm 3) and $\mathbf{NSIGT}_L$ (Algorithm 4) to solve Eq. (5.27), requiring only one additional $\mathbf{FFT}_L$ for initialization and one $\mathbf{IFFT}_L$ to obtain the result after convergence[2]. The preconditioner in Eq. (5.28) is realized by point-wise multiplication. Thus, one CG step involves one application of $\mathbf{NSGT}_L$ and $\mathbf{NSIGT}_L$ and L scalar multiplications for the preconditioning, see Algorithm 9. As experimental results in Sec. 5.5.1 show, only a few iterations are necessary for CG to converge to the correct solution up to numerical precision.

Although we have not conducted any experiments on iterative analysis so far, the same scheme can be used by relying on the selfadjointness of $\mathbf{S}$, $\mathbf{S}^{-1}$, giving

$$\langle f, \widetilde{\widehat{g_{n,m}}} \rangle = \langle \hat{f}, \hat{\mathbf{S}}^{-1} g_{n,m} \rangle = \langle \hat{\mathbf{S}}^{-1} \hat{f}, g_{n,m} \rangle.$$

A preconditioned iterative analysis algorithm is given as Algorithm 10.

Results and discussion

Two experiments were conducted to evaluate the performance of the ERBlet transform. In Exp. 1 we tested the convergence of iterative reconstruction for several NSG ERBlet systems yielding different redundancies (see Tab. 5.2). In Exp. 2 we compared the ERBlet to two other auditory-based approaches in terms of signal representation, reconstruction error, and redundancy. The audio material consisted of a 5-sec musical excerpt from the band Manowar (song “Heart of Steel”, studio version) in mono format, sampled at 44.1 kHz, 16 bits/sample. All analyses were performed for $F_{\text{min}} = 0$ and $F_{\text{max}} = 22.05$ kHz. Complementary results, colored figures, and simulation codes are available on the webpage.

[2] The first $\mathbf{FFT}_L$ can be substituted by involution, using $\left(\mathcal{F}\left(\mathbf{NSIGTF}_L(c, \mathbf{g}, \mathbf{a})\right)\right)[\cdot] = \mathbf{NSIGT}_L(c, \mathbf{g}, \mathbf{a})[-\cdot]$. If the $\mathbf{FFT}$ implementation used is not unitary, an additional normalization might be necessary.

Algorithm 9 Iterative synthesis: $\tilde{f} = \mathbf{NSIGTF}_L^{it}(c, \mathbf{g}, \mathbf{a}, \lambda)$

```
Initialize $x_0 = 0$, $k = 0$
$r_0 \leftarrow \mathbf{NSIGTF}_L(c, \mathbf{g}, \mathbf{a})$
$r_0 \leftarrow \mathbf{FFT}_L(r_0)$
$h_0, p_0 \leftarrow \mathbf{D}(\hat{\mathbf{S}})^{-1} r_0$
repeat
    $q_k = \mathbf{NSIGT}_L(\mathbf{NSGT}_L(p_0, \mathbf{g}, \mathbf{a}), \mathbf{g}, \mathbf{a})$
    $\alpha_k \leftarrow \frac{\langle r_k, h_k \rangle}{\langle p_k, q_k \rangle}$
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    $r_{k+1} \leftarrow r_k + \alpha_k q_k$
    $h_{k+1} \leftarrow \mathbf{D}(\hat{\mathbf{S}})^{-1} r_{k+1}$
    $\beta_k \leftarrow \frac{\langle r_{k+1}, h_{k+1} \rangle}{\langle r_k, h_k \rangle}$
    $p_{k+1} \leftarrow h_{k+1} + \beta_k p_k$
    $k \leftarrow k + 1$
until $r_k \leq \lambda$
$\tilde{f} \leftarrow \mathbf{IFFT}_L(x_k)$
```

Algorithm 10 Iterative analysis: $c = \mathbf{NSGTF}_L^{it}(f, \mathbf{g}, \mathbf{a}, \lambda)$

```
Initialize $x_0 = 0$, $k = 0$
$b \leftarrow \mathbf{FFT}_L(f)$
$r_0 \leftarrow b$
$h_0, p_0 \leftarrow \mathbf{D}(\hat{\mathbf{S}})^{-1} r_0$
repeat
    $q_k = \mathbf{NSIGT}_L(\mathbf{NSGT}_L(p_0, \mathbf{g}, \mathbf{a}), \mathbf{g}, \mathbf{a})$
    $\alpha_k \leftarrow \frac{\langle r_k, h_k \rangle}{\langle p_k, q_k \rangle}$
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    $r_{k+1} \leftarrow r_k + \alpha_k q_k$
    $h_{k+1} \leftarrow \mathbf{D}(\hat{\mathbf{S}})^{-1} r_{k+1}$
    $\beta_k \leftarrow \frac{\langle r_{k+1}, h_{k+1} \rangle}{\langle r_k, h_k \rangle}$
    $p_{k+1} \leftarrow h_{k+1} + \beta_k p_k$
    $k \leftarrow k + 1$
until $r_k \leq \lambda$
$x_k \leftarrow \mathbf{IFFT}_L(x_k)$
$c \leftarrow \mathbf{NSGTF}_L(x_k, \mathbf{g}, \mathbf{a})$
```

Table 5.2: Parameters, redundancies and frame bound ratios of the NSG ERBlet systems used in Exp. 1.

$\breve{\mathcal{G}}(\mathbf{g},\mathbf{a})$	case	V	M	N_m	red	B/A
ERBlet1	painless	1	43	$\lceil 4\,\Omega_m \rceil$	4.00	1.44
ERBlet2	painless	3	129	$\lceil 4\,\Omega_m \rceil$	12.00	1.07
ERBlet3	CG	1	43	$\lceil \frac{32\,\Omega_m}{9} \rceil$	3.53	1.44
ERBlet4	CG	1	43	$\lceil \frac{8\,\Omega_m}{3} \rceil$	2.64	1.44
ERBlet5	CG	1	43	$\lceil 2\,\Omega_m \rceil$	1.98	1.52
ERBlet6	CG	1	43	$\lceil \frac{4\,\Omega_m}{3} \rceil$	1.32	2.56
ERBlet7	CG	1	43	$\lceil \frac{12\,\Omega_m}{11} \rceil$	1.08	5.88

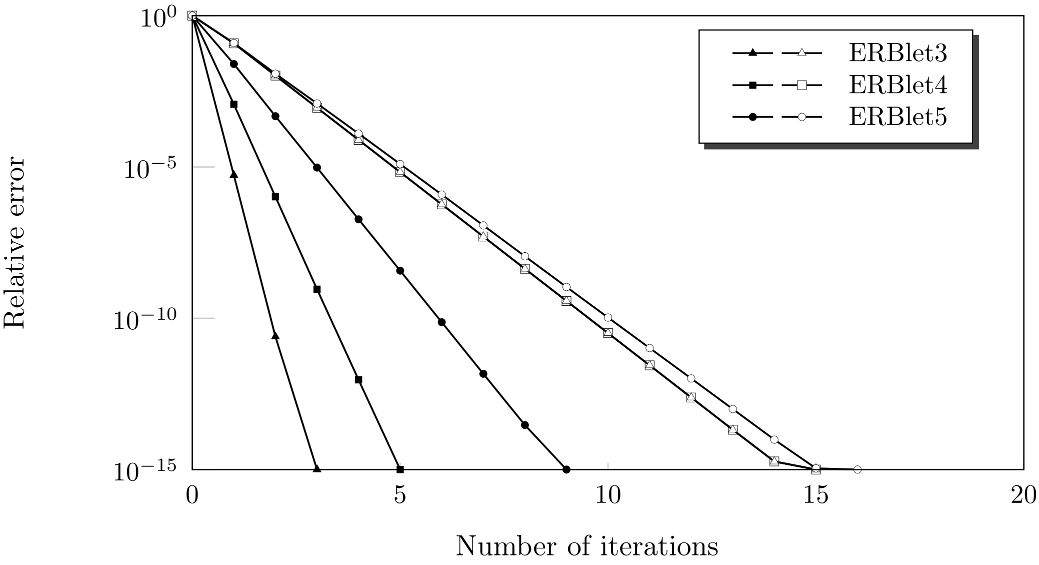


Figure 5.21: Convergence of the iterative reconstruction with (filled markers) and without diagonal preconditioning (empty markers) for various ERBlet configurations indicated in Tab. 5.2.

The results of Exp. 1 are depicted in Fig. 5.21 as a convergence plot. It can be seen that preconditioning had a considerable effect for the systems ERBlet3 to ERBlet5. Iterative synthesis for ERBlet6 and ERBlet7, not shown in the plot, converged in 21 and 45 iterations, respectively. Preconditioning had no effect in these cases. Noteworthy, the number of iterations does not depend on the signal length but only on the condition number.

In Exp. 2 we considered the system “ERBlet2” in Tab. 5.2, the NSG constant-Q transform in [164], and the linear gammatone filterbank in [90]. The constant-Q transform used 24 filters per octave distributed between 50 Hz and 22.05 kHz (212 filters in total) and Q = 9 ($\approx F/ERB(F)$ for $F > 2$ kHz). These parameters for the constant-Q were chosen so that both the constant-Q and ERBlet transforms have approximately the same number of filters in the frequency range 2–20 kHz (84) and the same redundancy over the whole TF plane (12). The gammatone filterbank used 3 filters per ERB (128 filters in total). Signal representations are depicted in Fig. 5.22. Fig. 5.22a shows that the ERBlet captured both harmonic (voice vibrato) and transient (drums) parts in the broadband, rich background generated by drums and distorted guitars. Fig. 5.22b shows that ERBlet and constant-Q representations are very similar above 500 Hz but differ below. Below 500 Hz the ERBlet has a better time resolution while the constant-Q transform has a better spectral resolution. This is due to the fact that the constant-Q transform features a larger number of filters at low than at high frequencies. Consequently, the constant-Q transform required 212 filters to achieve the same (visual) high-frequency resolution as the ERBlet with 129 filters. Both the ERBlet and constant-Q transforms led to perfect reconstruction (relative errors $< 10^{-15}$). Fig. 5.22c shows that ERBlet and gammatone representations are very similar over the whole TF plane. Since gammatone filters are auditory filter models *per se*, this result indicates that the ERBlet approximates well the auditory TF resolution. While the ERBlet achieved perfect reconstruction, the gammatone filterbank led to a relative reconstruction error of about 10^{-3}. Note however, that this error was perceptually irrelevant (as indicated by informal listening). Because the gammatone system in [90] features no downsampling option, its redundancy was 128 compared to 12 for the ERBlet.

Overall, the proposed method provides a linear, auditory-based, and perfectly invertible TF transform that can be easily integrated in audio analysis-synthesis systems. An advantage of the current implementation is that resolution and redundancy are adaptable without affecting the reconstruction error. While the ERBlet achieves a perceptually motivated TF analysis comparable to that of linear gammatone filterbanks [90, 149], it allows perfect reconstruction even with a density of 1 filter per ERB (see Tab. 5.2). In comparison, a gammatone implementation designed to achieve near-perfect reconstruction (relative error = 10^{-7}) in [149] requires a minimum density of 2.4 filters per ERB. Although our approach cannot substitute for physiologically plausible auditory models like [119], it could be useful to auditory modeling approaches in which a density of 1 filter per ERB is often desired [74].

To account for the level dependency of the auditory filters' bandwidth and the compressive response of the cochlea [120], an approximately invertible nonlinear gammatone filterbank was proposed in [94]. To further improve the match between the auditory and the transform resolutions while retaining the perfect reconstruc-

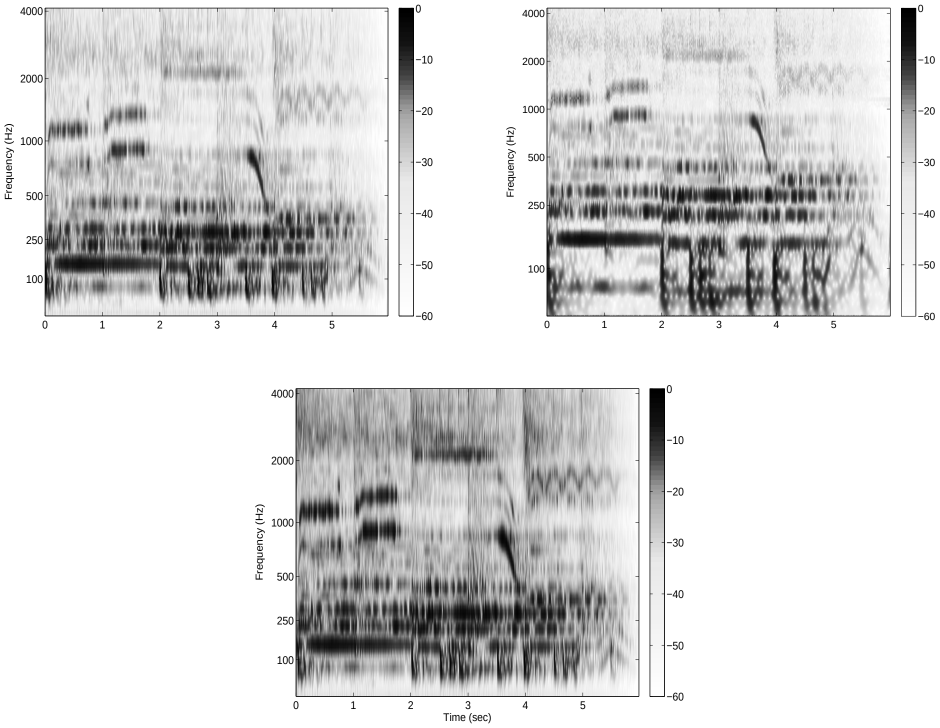


Figure 5.22: TF representations (squared moduli, in dB) for (top left) ERBlet, (top right) constant-Q transform, and (bottom) gammatone filterbank (restricted to the relevant frequency band 0–4000 Hz).

tion property of the ERBlet, future works include: inclusion of compression and use of windows with Gaussian shapes on the ERB scale (*i.e.*, using a warping function mapping the linear frequency axis to the ERB scale), see also the next section.

5.5.2 Warped tight frames

In this section we provide a method for the construction of tight nonstationary Gabor frames. In applications of frame theory, the frame elements are most often constructed from a number of explicitly known prototype functions and some transformation rule, e.g. translation and modulation for Gabor frames. The original prototype functions are designed to fulfill a number of useful properties such as time-frequency localization, fast decay or having a certain shape. In time-frequency analysis, bump functions with a distinct global maximum and good decay around the maximum are the canonical choice. Tight or well-conditioned frame constructions are very important, because they guarantee that the dual frame elements will not deviate too much from the nice properties of their originally chosen counterparts.

For Gabor systems, it is reasonably simple to construct explicit tight frames starting from a uniform partition of unity obtained from a compactly supported function. More explicitly, let $\gamma \in \mathcal{C}_c(\mathbb{R})$ be a continuous, compactly supported function such that

$$\sum_{n\in\mathbb{Z}} \mathbf{T}_{na}\gamma \equiv C, \tag{5.29}$$

for some $a, C \in \mathbb{R}^+$. For simplicity, assume $\operatorname{supp}(\gamma) = [-1/2, 1/2]$. Then $\mathcal{G}(g, a, b)$, with $g(t) = \sqrt{\gamma(t)}$ forms a tight frame (for $L^2(\mathbb{R})$) for any $b < 1$. This follows immediately from the painless nonorthogonal expansions result, see [48] or Chapter 4.

If $\tilde{L} \in a\mathbb{N}$ with $\tilde{L} > 1$, then any regular sampling[3] $g[l] = g(lD)$ with $L := \tilde{L}/D \in \mathbb{Z}$, $a/D \in \mathbb{N}$ and $l = -\lfloor L/2 \rfloor, \ldots, \lceil L/2 \rceil - 1$ gives rise to a tight painless Gabor frame $\mathcal{G}(g, a, b)$ in $\mathbb{C}^L$, for any $b^{-1} \geq \lfloor D^{-1} \rfloor$. That is, $\mathcal{G}(g, a, b)$ satisfies the discrete painless case conditions (Proposition 14) with constant frame operator diagonal.

We will now extend this construction to adapted frames by sampling the continuous partition of unity not regularly, but according to some warping function.

Tight frames via warping

A warping function is a continuous, increasing function $w : J \mapsto I$, where I, J are finite intervals in $\mathbb{R}$ and $[-L/2, L/2[$, respectively. We allow for three different cases:

(i) $J = [-L/2, L/2[$ and $w(-L/2) = ka$, $w(L/2) = ja$ for some $k, j \in \mathbb{Z}$, $[-1/2, 1/2[\subseteq [ka, ja[$,

(ii) $J = [0, A[$ and $w(0) = 0$, $w(A) = ja + 1/2$ for some $j \in \mathbb{N}$ and

(iii) $J = [A, B[$ and $w(A) = -1/2$ and $w(B) = ja + 1/2$ for some $j \in \mathbb{N}$,

where $A, B > 0$

In case (i), we would consider the interval $[ka, ja[$ circularly to construct a set of circular warped windows spanning the whole interval $[-L/2, L/2[$, while the warping function will be mirrored in a point symmetric way for the cases (ii) and (iii). In those cases, it is assumed that $j \geq \lfloor a^{-1} \rfloor$. Some examples of warping functions can be seen in Figure 5.23.

[3] By an abuse of notation, we will refer to the sampling $g \in \mathbb{C}^L$ of the continuous function $g \in \mathcal{C}_c(\mathbb{R})$ using the same notation g.

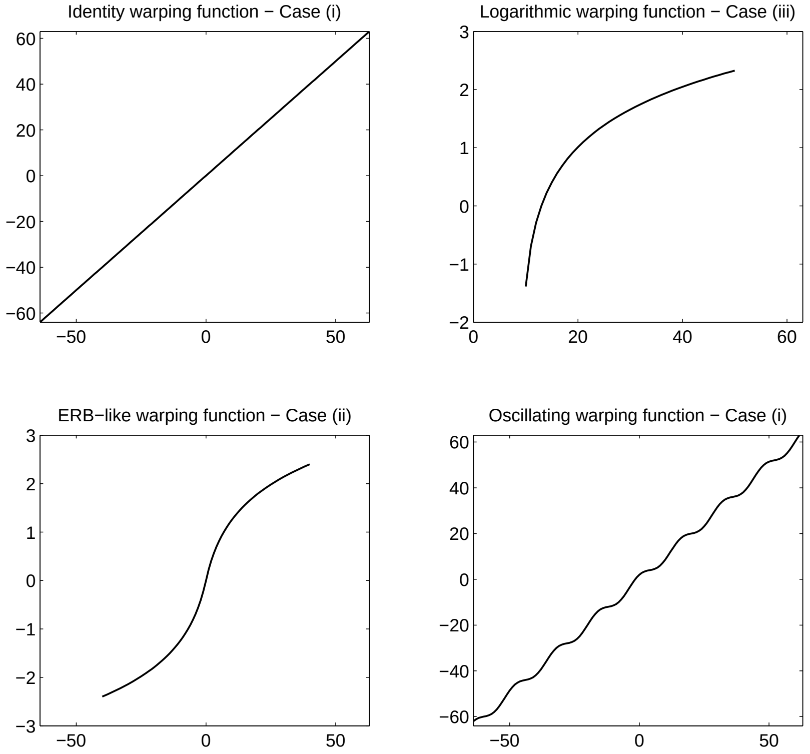


Figure 5.23: Some warping functions

We obtain the warped windows $g_n \in \mathcal{C}([-L/2, L/2[)$ as follows:

Case (i):

$$g_n(t) = (\mathbf{T}_{na}g + \mathbf{T}_{(n+(j-k)a}g)(w(t)), \text{ for } n = k, \dots, 0$$

and

$$g_n(t) = (\mathbf{T}_{na}g + \mathbf{T}_{(n-(j-k)a}g)(w(t)), \text{ for } n = 1, \dots, j-1.$$

Case (ii):

$$g_n(t) = (\mathbf{T}_{na}g)(\mathrm{sgn}(t)w(|t|)), \text{ for } n = -j, \dots, j \text{ and } g_{j+1}(t) = \sqrt{C - \sum_{n=-j}^{j-1} |g_n(t)|^2}.$$

Case (iii):

$g_n(t) = (\mathbf{T}_{(n-1)a}g)(w(t))$, for $n = 1, \dots, j, g_n(t) = g_{-n}(-t)$ for $n = -j, \dots, -1$ and g_0, g_{j+1} with $\mathrm{supp}(g_0) \subseteq [-c, c]$ and $\mathrm{supp}(g_0) \subseteq [-L/2, -c] \cup [c, L/2[$ for some $c \in [A, B[$ satisfying

$$|g_0(t)|^2 + |g_j(t)|^2 = C - \sum_{n=-j}^{j-1} |g_n(t)|^2.$$

The constant C in the equations above corresponds to the constant in Equation (5.29). Figure 5.24 shows the effect of a logarithmic warping function.

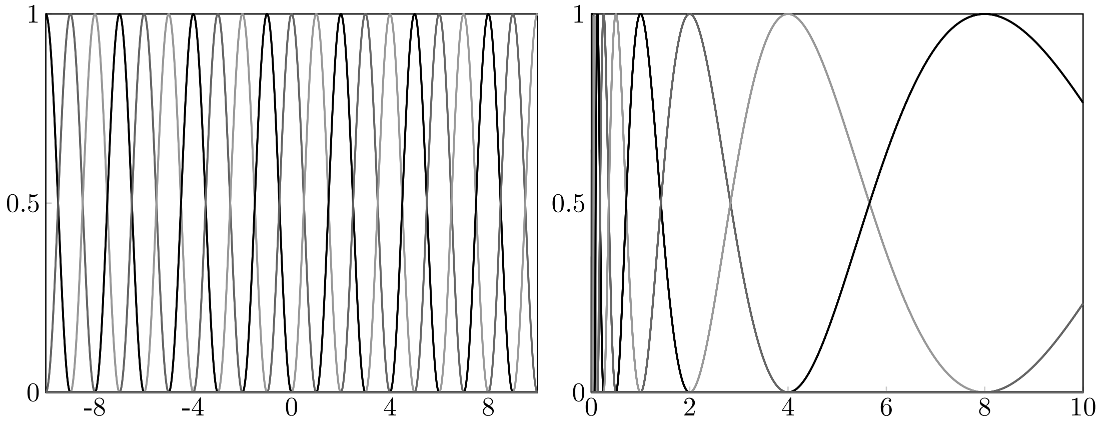


Figure 5.24: A set of regular translates (left) and their logarithmically warped counterparts (right).

By construction, every regular sampling of the functions g_n constructed above yields a set of finite, discrete function, each supported on a strict subset of $-\lfloor L/2 \rfloor, \ldots, \lceil L/2 \rceil - 1$ and with the property that their squared magnitudes have constant sum. Without loss of generality, we can assume $D = 1$, i.e. we obtain $g_n \in \mathbb{C}^L$ with $g_n[l] := g_n(l)$. Fixing some $M_n \in \mathbb{N}$ with $M_n \geq L_n$ for L_n the length of the smallest closed interval I_n containing the support of g_n, we obtain a painless, tight NSG frame $\mathcal{G}(\mathbf{g}, \mathbf{b})$, where $b_n = L/M_n$, after normalizing by M_n^{-1}.

These constructions can be used for tight frequency-side NSG frames on varying frequency scales, e.g. the CQ and ERB scales presented in this chapter, but also for time-side adaptation by controlling the derivative of the warping function dependent on signal features.

5.6 Summary and Conclusion

In this section, we discussed discrete NSG systems and some exemplary implementation alongside possible applications. One difficulty when using our approach is to adapt the time-frequency resolution to the evolution of the signal characteristics. If prior knowledge is available, this can be done by hand. We introduced a time-side adaptation scheme based on onset detection that was shown to improve sparsity in some cases. A different approach will involve the investigation of sparsity criteria as proposed in [97]. Time-side systems can be of particular interest for applications where the frequency characteristics of the signal are known to evolve significantly with time. Order analysis [143], in which the signal analyzed is produced by a rotating machine having changing rotating speed, is an example of such an application.

We have introduced a framework for real-time implementation of an invertible constant-Q transform based on frame theory. The proposed framework allows for straight-forward generalization to other non-linear frequency scales, such as mel- or Bark scale, cp. [86], [60] or ERB scale, see Section 5.5. While real-time processing is possible by means of a preprocessing step, we investigated the possible occurrence of time-aliasing. We provided a numerical evaluation of computation time and quality of approximation of the true NSGT coefficients.

In analogy to the classical phase vocoder, phase issues have to be addressed, if CQ-transformed coefficients are processed, cp. [106, 107, 132]. While preliminary experiments using the proposed framework for real-life signals were presented, undesired phasing effects, mainly due to the contribution of a signal component to several adjacent filters, will be investigated in detail in future work. This issue is discussed for unsliced constant-Q transforms in [140].

We have shown that modern implementations of classical algorithms provide efficient methods for iterative analysis or reconstruction for well-conditioned non-stationary Gabor frames with non-diagonal frame operator. Well-conditioned non-painless frames can easily be constructed by subsampling a painless, tight NSG frame, obtained from a warped set of uniform translates, as discussed in Section 5.5.

Future work will lead to adaptability in both time and frequency leading to *quilted frames* as introduced in [54], e.g. by applying the introduced slicing technique, allowing for different slice lengths and varying NSG systems on each slice. Furthermore, the idea of warped frames is currently under closer investigation, along with the development of a corresponding continuous theory.

Appendices

Appendix A

The toolboxes

An important goal of the thesis project presented in the previous chapters was the creation of a computational toolbox for experimental and application oriented work with adaptive time-frequency representations. The tools for nonstationary Gabor transforms are collected in the NSG Toolbox, available at `nsg.sourceforge.net` and containing routines for direct, as well as iterative, analysis and synthesis, creation of specific nonstationary Gabor dictionaries, setup of frame-related operators and plotting. These are accompanied by some wrapper routines that simplify the use of the different transform types and a few basic demos that introduce the syntax and usage of the Toolbox. The documentation is written to comply with mat2doc (`mat2doc.sourceforge.net`) standards and reproduced below.

The methods for nonseparable Gabor systems have been incorporated into the Large Time-Frequency Analysis Toolbox (LTFAT [2]). Note that the nonseparable functionality in LTFAT extends preexisting code and the author of this thesis was involved only in implementing the additional functionality for nonseparable Gabor systems. The documentation of the files in question is reproduced below, with the kind permission of LTFAT main author Peter L. Søndergaard.

A.1 NSGToolbox - Startup

A.1.1

NSGT_STARTUP - Set paths for using NSGToolbox

Usage

```
nsgt_startup()
```

Description This script file adds NSGToolbox folders to the MATLAB path.

A.2 NSGToolbox - Core routines

A.2.1 Forward transforms

NSGT - Nonstationary Gabor transform

Usage

```
[c,Ls] = nsgt(f,g,shift,M)
[c,Ls] = nsgt(f,g,shift)
c = nsgt(...)
```

Input parameters

f	The signal to be analyzed (For multichannel signals, input should be a matrix which each column storing a channel of the signal).
g	Cell array of frequency side analysis windows
shift	Vector of time shifts
M	Number of frequency channels (optional). If M is constant, the output is converted to a matrix

Output parameters

c	Transform coefficients (matrix or cell array)
Ls	Original signal length (in samples)

Description Given the cell array g of windows, the time shift vector *shift*, and channel numbers M, `nsgt` computes the corresponding nonstationary Gabor transform of *f*. Let $P(n) = \sum_{l=1}^{n} shift(l)$, then the output `c = nsgt(f,g,shift,M)` is a cell array with

$$c\{n\}(m) = \sum_{l=0}^{Ls-1} f[l]\overline{g\{n\}[l-P(n)]}e^{-2\pi i(l-P(n))m/M(n)},$$

where `m` runs from `0` to `M(n)-1`.

If multichannel input is used, the same nonstationary Gabor system is applied to each channel and each entry of c will be a 2D array with c{n}(:,CH) being the entries corresponding to time position n, signal channel *CH*.

If M is scalar or uniform, then c is converted into a regular array.

The choice of phase-locked coefficients (by inserting (l-P(n)) in the complex exponential) prevents border artifacts for combinations of window functions *g{n}* that do not have full support and *M(n)* that do not divide *Ls*.

References: [10]

NSGT_REAL - Nonstationary Gabor transform for real signals

Usage

```
[c,Ls] = nsgt_real(f,g,shift,M)
[c,Ls] = nsgt_real(f,g,shift)
c = nsgt_real(...)
```

Input parameters

f	A real-valued signal to be analyzed (For multichannel signals, input should be a matrix which each column storing a channel of the signal).
g	Cell array of frequency side analysis windows
shift	Vector of time shifts
M	Number of frequency channels (optional). If M is constant, the output is converted to a matrix

Output parameters

c	Transform coefficients (matrix or cell array)
Ls	Original signal length (in samples)

Description Given the cell array g of windows, the time shift vector *shift*, and channel numbers M, `nsgt_real` computes the corresponding nonstationary Gabor transform of *f*, computing only the positive frequency samples. Let $P(n) = \sum_{l=1}^{n} shift(l)$, then the output `c = nsgt_real(f,g,shift,M)` is a cell array with

$$c\{n\}(m) = \sum_{l=0}^{Ls-1} f[l]\overline{g\{n\}[l-P(n)]}e^{-2\pi i(l-P(n))m/M(n)},$$

where `m` runs from `0` to `floor( M(n)/2 )`.

For more details, see `nsgt`.

NSGTF - Nonstationary Gabor filterbank

Usage

```
[c,Ls] = nsgtf(f,g,shift,M)
[c,Ls] = nsgtf(f,g,shift)
c = nsgtf(...)
```

Input parameters

f	The signal to be analyzed (For multichannel signals, input should be a matrix which each column storing a channel of the signal).
g	Cell array of analysis filters
shift	Vector of frequency shifts
M	Number of time channels (optional). If M is constant, the output is converted to a matrix

Output parameters

c	Transform coefficients (matrix or cell array)
Ls	Original signal length (in samples)

Description Given the cell array *g* of windows, the time shift vector *shift*, and channel numbers *M*, `nsgtf` computes the corresponding nonstationary Gabor filterbank of *f*. Let $P(n) = \sum_{l=1}^{n} shift(l)$, then the output `c = nsgtf(f,g,shift,M)` is a cell array with

$$c\{n\}(m) = \sum_{l=0}^{Ls-1} \hat{f}[l]\overline{g\{n\}[l-P(n)]}e^{-2\pi i(l-P(n))m/M(n)},$$

where `m` runs from `0` to `M(n)-1`.

If multichannel input is used, the same nonstationary Gabor system is applied to each channel and each entry of *c* will be a 2D array with c{n}(:,CH) being the entries corresponding to frequency channel *n*, signal channel *CH*.

If *M* is scalar or uniform, then *c* is converted into a regular array.

The choice of phase-locked coefficients (by inserting (l-P(n)) in the complex exponential) prevents border artifacts for combinations of window functions *g{n}* that do not have full support and *M(n)* that do not divide *Ls*.

References: [10], [164]

NSGTF_REAL - Nonstationary Gabor filterbank for real signals

Usage

```
[c,Ls] = nsgtf_real(f,g,shift,M)
[c,Ls] = nsgtf_real(f,g,shift)
c = nsgtf_real(...)
```

Input parameters

f	A real-valued signal to be analyzed (For multichannel signals, input should be a matrix which each column storing a channel of the signal).
g	Cell array of analysis filters
shift	Vector of frequency shifts
M	Number of time channels (optional). If M is constant, the output is converted to a matrix

Output parameters

c	Transform coefficients (matrix or cell array)
Ls	Original signal length (in samples)

Description Given the cell array g of windows, the time shift vector *shift*, and channel numbers M, `nsgtf_real` computes the corresponding nonstationary Gabor filterbank of *f*, using only the filters with at least partially supported on the positive frequencies. Let $P(n) = \sum_{l=1}^{n} shift(l)$, then the output `c = nsgtf_real(f,g,shift,M)` is a cell array with

$$c\{n\}(m) = \sum_{l=0}^{Ls-1} \hat{f}[l]\overline{g\{n\}[l-P(n)]}e^{-2\pi i(l-P(n))m/M(n)},$$

where `m` runs from `0` to `M(n)-1` and `n` from 1 to `N`, where $g\{N\}$ is the final filter at least partially supported on the positive frequencies. All filters in g, *shift* that are completely supported on the negative frequencies are ignored.

For more details, see `nsgtf`.

A.2.2 Inverse transforms

NSIGT - Nonstationary Gabor synthesis

Usage

```
fr = nsigt(c,g,shift,Ls)
fr = nsigt(c,g,shift)
```

Input parameters

c	Cell array of nonstationary Gabor coefficients
g	Cell array of synthesis windows
shift	Vector of time shifts
Ls	Length of the analyzed signal

Output parameters

fr	Synthesized signal (Channels are stored in the columns)

Description Given the cell array c of nonstationary Gabor coefficients, a set of windows g and time shifts *shift*, this function computes the corresponding nonstationary Gabor synthesis. Let `N=numel(g)` and $P(n) = \sum_{l=1}^{n} shift(l)$, then the synthesis formula reads:

$$fr[l] = \sum_{n=0}^{N-1} \sum_{m} c\{n\}(m) g\{n\}[l - P(n)] e^{2\pi i (l-P(n))m/M(n)},$$

for $l = 0, \cdots, Ls - 1$. In practice, the synthesis formula is realized by `ifft` and overlap-add.

If a nonstationary Gabor frame was used to produce the coefficients and g is a corresponding dual frame, this function should perfectly reconstruct the originally analyzed signal to numerical precision.

Multichannel output will save each channel in a column of *fr*.

References: [10]

NSIGT_REAL - Nonstationary Gabor synthesis for real signals

Usage

```
fr = nsigt_real(c,g,shift,M,Ls)
```

Input parameters

c Cell array of nonstationary Gabor coefficients

g Cell array of synthesis windows

shift Vector of time shifts

M Number of frequency channels (vector/scalar)

Ls Length of the analyzed signal

Output parameters

fr Synthesized real-valued signal (Channels are stored in the columns)

Description Given the cell array c of nonstationary Gabor coefficients, a set of windows g and time shifts *shift*, this function computes the corresponding real-valued nonstationary Gabor synthesis. Let `N=numel(g)` and $P(n) = \sum_{l=1}^{n} shift(l)$, then the complex valued synthesis formula reads:

$$fr[l] = \sum_{n=0}^{N-1} \sum_{m} c\{n\}(m) g\{n\}[l - P(n)] e^{2\pi i (l-P(n))m/M(n)},$$

for $l = 0, \cdots, Ls - 1$. In practice, the synthesis formula is realized by `ifft` and overlap-add. In the real valued case, `ifftreal` provides the missing frequency content normally given by the coefficients `c{n}(m)` for $m \geq \lfloor M(n)/2 \rfloor$.

If a nonstationary Gabor frame was used to produce the coefficients and g is a corresponding dual frame, this function should perfectly reconstruct the originally analyzed signal to numerical precision.

Note that `nsigt_real` requires the input parameter M to guarantee that the vectors used in the overlap-add process are of the correct length.

Multichannel output will save each channel in a column of *fr*.

References: [10]

NSIGTF - Nonstationary Gabor filterbank synthesis

Usage

```
fr = nsigtf(c,g,shift,Ls)
fr = nsigtf(c,g,shift)
```

Input parameters

c Cell array of nonstationary Gabor coefficients

g Cell array of synthesis filters

shift Vector of frequency shifts

Ls Length of the analyzed signal

Output parameters

fr Synthesized real-valued signal (Channels are stored in the columns)

Description Given the cell array *c* of nonstationary Gabor filterbank coefficients, a set of filters *g* and frequency shifts *shift*, this function computes the corresponding nonstationary Gabor filterbank synthesis. Let `N=numel(g)` and $P(n) = \sum_{l=1}^{n} shift(l)$, then the synthesis formula reads:

$$\mathbf{FFT}(fr)[l] = \sum_{n=0}^{N-1} \sum_{m} c\{n\}(m) g\{n\}[l - P(n)] e^{-2\pi i (l-P(n))m/M(n)},$$

for $l = 0, \cdots, Ls - 1$. The final reconstruction step then is `fr = ifft(fr)`. In practice, the synthesis formula is realized by `fft`, followed by `ifft` and overlap-add.

If a nonstationary Gabor frame was used to produce the coefficients and g is a corresponding dual frame, this function should perfectly reconstruct the originally analyzed signal to numerical precision.

Multichannel output will save each channel in a column of *fr*.

References: [10], [164]

NSIGTF_REAL - Nonstationary Gabor filterbank synthesis for real signals

Usage

```
fr = nsigtf_real(c,g,shift,M,Ls)
```

Input parameters

c	Cell array of nonstationary Gabor coefficients
g	Cell array of synthesis filters
shift	Vector of frequency shifts
M	Number of time channels (vector/scalar)
Ls	Length of the analyzed signal

Output parameters

fr	Synthesized real-valued signal (Channels are stored in the columns)

Description Given the cell array c of nonstationary Gabor filterbank coefficients, a set of filters g and frequency shifts *shift*, this function computes the corresponding nonstationary Gabor filterbank synthesis for real valued signals.

Note that, due to the structure of the coefficient array in the real valued setting, all entries `g{n}` with $N > length(c)$ will be ignored and assumed to be fully supported on the negative frequencies.

Let $P(n) = \sum_{l=1}^{n} shift(l)$, then the synthesis formula reads:

$$fr_{temp}[l] = \sum_{n=0}^{N-1} \sum_{m} c\{n\}(m) g\{n\}[l - P(n)] e^{-2\pi i (l-P(n))m/M(n)},$$

for $l = 0, \cdots, Ls - 1$. In practice, the synthesis formula is realized by `fft` and overlap-add. To synthesize the negative frequencies, fr_{temp} is truncated to length $\lfloor Ls/2 \rfloor + 1$. Afterwards `ifftreal` implicitly computes the hermite symmetric extension and computes the inverse Fourier transform, i.e. `fr = ifftreal(`fr_{temp}`)`.

If a nonstationary Gabor frame was used to produce the coefficients and g is a corresponding dual frame, this function should perfectly reconstruct the originally analyzed signal to numerical precision.

Multichannel output will save each channel in a column of *fr*.

References: [10], [164]

A.2.3 Painless reconstruction

NSDUAL - Canonical dual NSG frame (for painless systems)

Usage

```
gd = nsdual(g,shift,M)
```

Input parameters

g	Cell array of window functions/filters
shift	Vector of time/frequency shifts
M	Number of frequency channels (vector/scalar)

Output parameters

gd	Dual window functions

Description Given a nonstationary Gabor frame specified by the windows/filters *g*, shift parameters *shift*, and channel numbers *M*, `nsdual` computes the canonical dual frame windows/filters *gd* by inverting the diagonal of the frame operator and applying the inverse to *g*. More explicitly,

$$gd\{n\} = \frac{g\{n\}}{\sum_l M(l)|g\{l\}|^2}$$

If *g*, *shift*, *M* specify a painless frame, i.e. $|supp(g\{n\})| \leq M(n) \; \forall \; n$ and

$$\sum_n M(n)|g\{n\}|^2 \simeq 1,$$

the computation will result in the canonical dual frame. If *g*, *shift*, *M* specify a frame, but the first condition is violated, the result can be interpreted as a first approximation of the corresponding canonical dual frame.

Note, the time shifts corresponding to the dual window sequence is the same as the original shift sequence and as such already given.

If *g*, *shift*, *M* is a painless frame, the output can be used for perfect reconstruction of a signal using the inverse nonstationary Gabor transform `nsigt`, `nsigtf`.

References: [10]

NSTIGHT - Canonical tight NSG frame (for painless systems)

Usage

```
gt = nstight(g,shift,M)
```

Input parameters

g	Cell array of window functions/filters
shift	Vector of time/frequency shifts
M	Number of frequency channels (vector/scalar)

Output parameters

gt	Tight window functions

Description Given a nonstationary Gabor frame specified by the windows/filters *g*, shift parameters *shift*, and channel numbers *M*, `nstight` computes the canonical tight frame windows/filters *gt* by inverting the diagonal of the frame operator and applying the square root to the inverse to *g*. More explicitly,

$$gt\{n\} = \frac{g\{n\}}{\sqrt{\sum_l M(l)|g\{l\}|^2}}.$$

If *g*, *shift*, *M* specified a painless frame, i.e. $|supp(g\{n\})| \leq M(n) \; \forall \; n$ and

$$\sum_n M(n)|g\{n\}|^2 \simeq 1,$$

the computation will result in a tight nonstationary Gabor frame. If *g*, *shift*, *M* specify a frame, but the first condition is violated, the result can be interpreted as a first approximation of the corresponding canonical tight frame.

Note, the time shifts corresponding to the tight window sequence is the same as the original shift sequence and as such already given.

If *g*, *shift*, *M* is a painless frame, the output can be used for analysis and perfect reconstruction of a signal using the nonstationary Gabor algorithms `nsgt`, `nsgtf`, `nsigt`, `nsigtf`.

References: [10]

A.3 NSGToolbox - Dictionary generators

A.3.1 Nonstationary Gabor transform dictionaries

NSGSCLWIN - Scale-frame dictionary generator

Usage

```
[g,shift,M] = nsgsclwin(positions,short,max_win,Ls)
```

Input parameters

positions A vector specifying time positions around which shortest windows are desired

short Shortest admissible window length (in samples)

max_win Maximum number of different window sizes

Ls Length of the signal to be analyzed (in samples)

Output parameters

g Cell array of window functions

shift Vector of time shifts

M Vector of window lengths

Description Create nonstationary Gabor frame from a sequence of positions, e.g. an onset sequence. This routine builds scale-type frames with fixed scale parameter $Q = 2$ and overlap parameters $O_1 = 2/3$, $O_2 = 1/3$. Currently, Hann windows of varying length are used as prototype.

The routine places short windows on the points specified by the sequence, while the space between 2 points is spanned by a sequence of that smoothly expand to longer lengths and then shrink back to a short length. We call those short sequences `short->long->short` building blocks.

The final window sequence is obtained by concatenating the sequence of building blocks.

Notes:

For the parameters `Q`, `O_1` and `O_2` as specified above being valid, the input parameter *short* must be a multiple of 6, or non integer shifts might occur.

The first value of *positions* should always be 1 to cover the complete time axis. A valid position sequence can be obtained, e.g. from `onsetdet`.

References: [10]

A.3.2 Nonstationary Gabor filterbank dictionaries

NSGCQWIN - Constant-Q/Variable-Q dictionary generator

Usage

```
[g,shift,M] = nsgcqwin(fmin,fmax,bins,sr,Ls,varargin)
[g,shift,M] = nsgcqwin(fmin,fmax,bins,sr,Ls)
```

Input parameters

fmin	Minimum frequency (in Hz)
fmax	Maximum frequency (in Hz)
bins	Vector consisting of the number of bins per octave
sr	Sampling rate (in Hz)
Ls	Length of signal (in samples)
varargin	Optional input pairs (see table below)

Output parameters

g	Cell array of constant-Q/variable-Q filters
shift	Vector of shifts between the center frequencies
M	Vector of lengths of the window functions

Description Create a nonstationary Gabor filterbank with constant or varying Q-factor and relevant frequency range from *fmin* to *fmax*. To allow for perfect reconstruction, the frequencies outside that range will be captured by 2 additional filters placed on the zero and Nyquist frequencies, respectively.

The Q-factor (quality factor) is the ratio of center frequency to bandwidth `cent_freq/bandwidth`.

To create a constant-Q filterbank with a fixed number of bins per octave, use a scalar parameter *bins*. The default parameters serve to set up a filter sequence with approximately 1/2 overlap and only approximately constant Q-factor (up to 1 sample deviation). The optional switch *fractional* can be set to 1 to allow for fractional sampling and exact constant Q-factor.

Alternatively, a vector *bins* can be supplied. In this case, successive octaves can have different numbers of filters regularly spaced on a logarithmic scale, e.g. *bins(1)* filters will be placed between `fmin` and `2*fmin`, *bins(2)* filters between `2*fmin` and `4*fmin` and so on.

For more details on the construction of the constant-Q nonstationary Gabor filterbank, please check the reference.

Optional input arguments arguments can be supplied like this:

```
nsgcqwin(fmin,fmax,bins,sr,Ls,'min_win',min_win)
```

The arguments must be character strings followed by an argument:

'min_win',min_win	Minimum admissible window length (in samples)
'Qvar',Qvar	Bandwidth variation factor
'bwfac',bwfac	Channel numbers M are rounded to multiples of this
'fractional',fractional	Allow fractional shifts and bandwidths
'winfun',winfun	String containing the desired window function name

References: [164], [92]

NSGWVLTWIN - Wavelet dictionary generator

Usage

```
[g,shift,M,fb] = nsgwvltwin(fmin,bw,bins,sr,Ls,winfun)
[g,shift,M,fb] = nsgwvltwin(fmin,bw,bins,sr,Ls)
[g,shift,M] = nsgwvltwin(...)
```

Input parameters

fmin	Desired minimum center frequency (in Hz)
bw	Desired bandwidth in the first frequency band (in Hz)
bins	Desired number of bins per octave
sr	Sampling rate (in Hz)
Ls	Signal length
winfun	String containing the window function name, see `winfuns`

Output parameters

g	Cell array of Wavelet filters
shift	Vector of frequency shifts
M	Number of time channels
fb	Frame bounds of the resulting system

Description Given the parameter set *fmin*, *bw*, *bins*, *sr* and *Ls*, this function constructs a painless system of bandlimited Wavelets spanning the range of frequencies from *fmin* to $2^{k/bins} fmin$ with k such that the dilation centered at this frequency will be completely contained in the positive frequencies, but the Wavelet at $2^{k+1/bins} fmin$ would not be. The number of scales per octave is determined by the input parameter *bins*, while the Wavelet corresponding to the largest (time-)scale will is constructed to have a bandwidth of *bw* Hz. The low and high frequencies will be spanned by a plateau-like filter each to ensure the frame property.

If you are not familiar with Wavelet systems, please use `wvlttrans` instead.

References: [10]

NSGERBWIN - ERBlet dictionary generator

Usage

```
[g,shift,M]=nsgerbwin(bins,sr,Ls,varargin)
[g,shift,M]=nsgerbwin(bins,sr,Ls)
```

Input parameters

bins	Desired bins per ERB
sr	Sampling rate of f (in Hz)
Ls	Signal length
varargin	Optional input pairs (see table below)

Output parameters

g	Cell array of ERBlet filters
shift	Vector of frequency shifts
M	Number of time channels

Description Create a nonstationary Gabor filterbank composed of filters regularly spaced on the ERB frequency scale and having constant Equivalent Rectangular Bandwidth.

The conversion formula of Hz to ERB number is given by

$$ERB_{num}(x) = 9.2645\,\mathrm{sgn}(x)\log(1 + 0.00437|x|).$$

The Equivalent Rectangular Bandwidth at frequency `x` is

$$ERB(x) = 24.7(1 + 0.00437x).$$

The filters are chosen symmetrically around the zero frequency and finally a symmetric filter is placed on the Nyquist frequency.

The result can serve as input parameters for `nsgtf` to obtain the ERBlet analysis coefficients or `nsigtf` to synthesize from coefficients, as well as their counterparts for real-valued signals.

Optional input arguments arguments can be supplied like this:

```
nsgerbwin(bins,sr,Ls,'Qvar',Qvar)
```

The arguments must be character strings followed by an argument:

'Qvar',Qvar	Bandwidth variation factor
'bwfac',bwfac	Channel numbers M are rounded to multiples of this
'winfun',winfun	String containing the desired window function name

References: [10], [122]

A.4 NSGToolbox - Operator matrices

A.4.1 Operator matrices

NSGFRMMAT - Sparse nonstationary Gabor frame operator matrix

Usage

```
S = nsgfrmmat(g,shift,M,Ls)
S = nsgfrmmat(g,shift,M)
S = nsgfrmmat(g,shift)
```

Input parameters

g	Cell array of window functions
shift	Vector of shifts between the center frequencies
M	Vector of lengths of the window functions
Ls	Signal length (optional)
steps	Maximum number of side-diagonals to compute (optional)

Output parameters

S	Sparse frame operator matrix

Description Creates the frame operator matrix of size $Ls \times Ls$ associated to the nonstationary Gabor system g, *shift*, M in sparse matrix format.

From the Walnut representation of the nonstationary Gabor frame operator S we can deduce that the discrete nonstationary Gabor frame operator is represented by a sparse matrix.

Let `N = numel(shift)` and $K_l = \{n \in [0, N-1] : l = 0 \bmod M(n)\}$, then

$$S(k,j) = \sum_{n \in K_{k-j}} M(n) g\{n\}[k] * conj(g\{n\}[j]).$$

This representation is used together with the size (support) of the windows g to compute only the relevant entries of S are computed.

The optional parameter *steps* can be used to compute approximations of the frame operator with only a certain number of side-diagonals.

References: [10], [91]

NSGANAMAT - Nonstationary Gabor analysis operator matrix

Usage

```
G = nsganamat(g,shift,M,Ls,phaselock)
G = nsganamat(g,shift,M,Ls)
G = nsganamat(g,shift,M)
G = nsganamat(g,shift)
G = nsganamat(g,shift,M,phaselock)
G = nsganamat(g,shift,phaselock)
```

Input parameters

g	Cell array of analysis windows
shift	Vector of time shifts
M	Number of frequency channels (optional)
Ls	Transform length
phaselock	This optional 0/1 switch specifies the phaselock convention: 0 (non-phaselocked), 1 (phaselocked, default)

Output parameters

G	Frame analysis operator corresponding to the input arguments

Description Computes the frame analysis matrix corresponding the nonstationary Gabor system specified by g, *shift* and M. The rows of the frame analysis matrix contain the complex conjugate of the frame elements in modulation first order, i.e. let $P(n) = \sum_{l=1}^{n} shift(l)$

$$K_n = \sum_{l=0}^{n-1} M(l)$$

and let m run from 0 to $M(n) - 1$. Then

$$G(K_n + m, cdot) = \overline{g\{n\}(\cdot - P(n-1))} * e^{-2\pi i m(\cdot - P(n-1))/M(n)}$$

The conjugate transpose of G equals the synthesis operator corresponding to g, *shift* and M. Consequently, $c = Gf$ and $fr = conj(G)^T c$.

The formulas above use the phaselocked definition of a nonstationary Gabor frame also realized by `nsgt`, `nsigt`. Alternatively, non-phaselocked frame elements can be used. Note however, that this might result in border discontinuities if $L/M(n)$ is not integer.

Note: While this routine can be used to gain some insight into the structure of frame-related operators, it is not suited for use with transform lengths over a few thousand samples.

References: [33], [10]

A.5 NSGToolbox - Iterative algorithms

A.5.1 Iterative analysis

NSGAITER - Iterative nonstationary Gabor analysis

Usage

```
[c,Ls,res,Nit]=nsgaiter(f,g,shift,M,varargin)
[c,Ls,res,Nit]=nsgaiter(f,g,shift,M)
[c,Ls,res]=nsgaiter(...)
[c,Ls]=nsgaiter(...)
c=nsgaiter(...)
```

Input parameters

f	Input signal
g	Cell array of window functions
shift	Vector of shifts between the window positions
M	Number of frequency channels
varargin	Optional input pairs (see table below)

Output parameters

c	Transform coefficients
Ls	Input signal length
res	Vector of relative residuals
Nit	Number of iterations

Description Given a function f and nonstationary Gabor frame specified by g, *shift* and M, this routine approximates the frame coefficients associated to the canonical dual frame.

The approximated coefficients are obtained by first applying the inverse frame operator to f iteratively using the conjugate gradients method, followed by computing the analysis coefficients of $S^{-1}f$ with respect to g, *shift* and M with `nsgt`, followed. The following equivalence is used:

$$c\{n\}(m) = \langle f, \mathbf{S}^{-1} g_{n,m} \rangle = \langle \mathbf{S}^{-1} f, g_{n,m} \rangle$$

The conjugate gradients algorithm uses the frame operator, or rather its efficient realization by applying `nsgt` and `nsigt` consecutively.

Convergence speed of the conjugate gradients algorithm depends on the condition number of the frame operator, which can be improved by preconditioning. Currently, only a diagonal preconditioner using the inverse of the frame operator diagonal is implemented.

Note: The algorithm only converges if g, *shift* and M form a frame.

Optional input arguments arguments can be supplied like this:

```
nsgaiter(f,g,shift,M,'tol',tol)
```

The arguments must be character strings followed by an argument:

'tol',tol	Error tolerance
'Mit',Mit	Maximum number of iterations
'prec',prec	Preconditioning switch

References: [122], [82]

NSGAITERF - Iterative nonstationary Gabor filterbank analysis

Usage

```
[c,Ls,res,Nit]=nsgaiterf(f,g,shift,M,varargin)
[c,Ls,res,Nit]=nsgaiterf(f,g,shift,M)
[c,Ls,res]=nsgaiterf(...)
[c,Ls]=nsgaiterf(...)
c=nsgaiterf(...)
```

Input parameters

f	Input signal
g	Cell array of filters
shift	Vector of shifts between the center frequencies
M	Number of time steps
varargin	Optional input pairs (see table below)

Output parameters

c	Filterbank coefficients
Ls	Input signal length
res	Vector of relative residuals
Nit	Number of iterations

Description Given a function f and nonstationary Gabor filterbank frame specified by g, *shift* and M, this routine approximates the frame coefficients associated to the canonical dual frame.

The approximated coefficients are obtained by first applying the inverse frame operator to f iteratively using the conjugate gradients method, followed by computing the analysis coefficients of $S^{-1}f$ with respect to g, *shift* and M with `nsgtf`, followed. The following equivalence is used:

$$c\{n\}(m) = \langle f, \mathbf{S}^{-1} g_{n,m}\rangle = \langle \mathbf{S}^{-1} f, g_{n,m}\rangle$$

The conjugate gradients algorithm uses the frame operator, or rather its efficient realization by applying `nsgtf` and `nsigtf` consecutively.

Convergence speed of the conjugate gradients algorithm depends on the condition number of the frame operator, which can be improved by preconditioning. Currently, only a diagonal preconditioner using the inverse of the frame operator diagonal is implemented.

Note: The algorithm only converges if g, *shift* and M form a frame.

Optional input arguments arguments can be supplied like this:

```
nsgaiterf(f,g,shift,M,'tol',tol)
```

The arguments must be character strings followed by an argument:

'tol',tol Error tolerance

'Mit',Mit Maximum number of iterations

'prec',prec Preconditioning switch

References: [122], [82]

A.5.2 Iterative synthesis

NSGSITER - Iterative nonstationary Gabor synthesis

Usage

```
[fr,res,Nit]=nsgsiter(c,g,shift,M,Ls,varargin)
[fr,res,Nit]=nsgsiter(c,g,shift,M,varargin)
[fr,res,Nit]=nsgsiter(c,g,shift,M,Ls)
[fr,res,Nit]=nsgsiter(c,g,shift,M)
[fr,res]=nsgsiter(...)
fr=nsgsiter(...)
```

Input parameters

c	Nonstationary Gabor coefficients
g	Cell array of window functions
shift	Vector of shifts between the window positions
M	Number of frequency channels
Ls	Original signal length
varargin	Optional input pairs (see table below)

Output parameters

fr	Synthesized output signal
res	Vector of relative residuals
Nit	Number of iterations

Description Given coefficients c and nonstationary Gabor frame specified by g, *shift* and M, this routine approximates the synthesis formula associated to the canonical dual frame.

The synthesized signal *fr* is obtained by first synthesizing with respect to g, *shift* and M using `nsigt` followed by iteratively applying the inverse frame operator to the result using the conjugate gradients method. The following equivalence is used:

$$fr = \sum_{n=0}^{N-1} \sum_{m=0}^{M(n)-1} c\{n\}(m) S^{-1} g_{n,m} = S^{-1} \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M(n)-1} c\{n\}(m) g_{n,m} \right),$$

where `N=numel(shift)`. The conjugate gradients algorithm uses the frame operator, or rather its efficient realization by applying `nsgt` and `nsigt` consecutively.

Convergence speed of the conjugate gradients algorithm depends on the condition number of the frame operator, which can be improved by preconditioning. Currently, only a diagonal preconditioner using the inverse of the frame operator diagonal is implemented.

Note: The algorithm only converges if g, *shift* and M form a frame.

Optional input arguments arguments can be supplied like this:

```
nsgsiter(c,g,shift,M,'tol',tol)
```

The arguments must be character strings followed by an argument:

'tol',tol	Error tolerance
'Mit',Mit	Maximum number of iterations
'prec',prec	Preconditioning switch

References: [122], [82]

NSGSITERF - Iterative nonstationary Gabor filterbank synthesis

Usage

```
[fr,res,Nit]=nsgsiterf(c,g,shift,M,Ls,varargin)
[fr,res,Nit]=nsgsiterf(c,g,shift,M,varargin)
[fr,res,Nit]=nsgsiterf(c,g,shift,M,Ls)
[fr,res,Nit]=nsgsiterf(c,g,shift,M)
[fr,res]=nsgsiterf(...)
fr=nsgsiterf(...)
```

Input parameters

c	Nonstationary Gabor coefficients
g	Cell array of filters
shift	Vector of shifts between the center frequencies
M	Number of time steps
Ls	Original signal length
varargin	Optional input pairs (see table below)

Output parameters

fr	Synthesized output signal
res	Vector of relative residuals
Nit	Number of iterations

Description Given coefficients *c* and nonstationary Gabor frame specified by *g*, *shift* and *M*, this routine approximates the synthesis formula associated to the canonical dual frame.

The synthesized signal *fr* is obtained by first synthesizing with respect to *g*, *shift* and *M* using `nsigtf` followed by iteratively applying the inverse frame operator to the result using the conjugate gradients method. The following equivalence is used:

$$fr = \sum_{n=0}^{N-1} \sum_{m=0}^{M(n)-1} c\{n\}(m) S^{-1} g_{n,m} = S^{-1} \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M(n)-1} c\{n\}(m) g_{n,m} \right),$$

where `N=numel(shift)`. The conjugate gradients algorithm uses the frame operator, or rather its efficient realization by applying `nsgtf` and `nsigtf` consecutively.

Convergence speed of the conjugate gradients algorithm depends on the condition number of the frame operator, which can be improved by preconditioning. Currently, only a diagonal preconditioner using the inverse of the frame operator diagonal is implemented.

Note: The algorithm only converges if *g*, *shift* and *M* form a frame.

Optional input arguments arguments can be supplied like this:

```
nsgsiter(c,g,shift,M,'tol',tol)
```

The arguments must be character strings followed by an argument:

'tol',tol	Error tolerance
'Mit',Mit	Maximum number of iterations
'prec',prec	Preconditioning switch

References: [122], [82]

A.6 NSGToolbox - Wrapper functions for specific transforms

A.6.1 Onset-based transform

ONSETNSGT - Onset-based nonstationary Gabor transform

Usage

```
[c,g,shift,M,Ls] = onsetnsgt(f,thre,short,max_win,win_length)
[c,g,shift,M,Ls] = onsetnsgt(f,thre,short,max_win)
[c,g,shift,M,Ls] = onsetnsgt(f,thre,short)
[c,g,shift,M,Ls] = onsetnsgt(f,thre)
c = onsetnsgt(...)
```

Input parameters

f	The signal to be analyzed (single channel only)
thre	Peak-picking threshold
short	Shortest allowed window length
max_win	Maximum number of different windows used
win_length	Window length for the onset STFT analysis

Output parameters

c	Cell array of transform coefficients
g	Cell array of analysis windows
shift	Vector of time shifts
M	Number of frequency channels
Ls	Original signal length

Description This is a wrapper function for the scaleframe nonstationary Gabor transforms with onset detection based adaptation. Given a signal *f*, this wrapper computes the spectral flux onset detection function based on a regular discrete Gabor transform with redundancy 16 using a Hann window of length *win_length*. A simple peakpicking algorithm determines the significant maxima in the spectral flux function. Those are assumed to be the onsets in *f*.

From this onset sequence, a scaleframe nonstationary Gabor system will be constructed and the corresponding analysis performed by `nsgt_real`.

Note: The current wrapper only supports the threshold parameter *thre* of the onset detection algorithm. To obtain optimal results, the remaining parameters need to be fine tuned as well. An experienced user should use `onsetdet`, `nsgsclwin` and `nsgt_real` on separately instead. Also see the Onset How-To included in the toolbox.

References: [10], [51]

INVONSETNSGT - Onset-based nonstationary Gabor synthesis

Usage

```
fr = invonsetnsgt(c,g,shift,M,Ls)
fr = invonsetnsgt(c,g,shift,M)
```

Input parameters

c	Cell array of transform coefficients
g	Cell array of analysis windows
shift	Vector of time shifts
M	Number of time channels
Ls	Original signal length

Output parameters

fr	Reconstructed signal

Description This is a wrapper function for the inverse scaleframe nonstationary Gabor transform with onset detection based adaptation. It basically just forwards the input to `nsdual` and `nsigt_real`.

For more information see `onsetnsgt`.

References: [10]

A.6.2 Wavelet transform

WVLTTRANS - Wavelet frame transform

Usage

```
[c,g,shift,M,Ls,fb,tgtfl] = wvlttrans(f,fmin,sr,bins,bw,tgtfl)
[c,g,shift,M,Ls,fb,tgtfl] = wvlttrans(f,fmin,sr,bins,bw)
[c,g,shift,M,Ls,fb,tgtfl] = wvlttrans(f,fmin,sr,bins)
[c,g,shift,M,Ls,fb,tgtfl] = wvlttrans(f,fmin,sr)
[c,g,shift,M,Ls,fb,tgtfl] = wvlttrans(f,fmin)
[c,g,shift,M,Ls,fb] = wvlttrans(...)
```

```
[c,g,shift,M,Ls] = wvlttrans(...)
[c,g,shift,M] = wvlttrans(...)
c = wvlttrans(...)
```

Input parameters

f	Input signal
fmin	Desired minimum center frequency (in Hz)
sr	Sampling rate of f (in Hz)
bins	Desired number of bins per octave
bw	Desired bandwidth in the first frequency band (in Hz)
winfun	String containing the desired window function name

Output parameters

c	Cell array of Wavelet coefficients
g	Cell array of Fourier transforms of the analysis Wavelets
shift	Vector of frequency shifts
M	Number of time steps
Ls	Original signal length
fb	Frame bounds (vector)
tgtfl	Tightflag (1 if frame is tight)

Description This is a wrapper function for the painless Wavelet transform via nonstationary Gabor filterbank. Given a signal *f* and minimum frequency *fmin*, a tight system with 4 scales per octave is constructed using logarithmically sampled Hann windows with 3/4 overlap. The additional parameters *sr*, *bins*, *bw* and *winfun* can be specified to individually construct different Wavelet systems.

To construct systems with specific overlap factors $(n-1)/n$, choose $bw = 2^{n/(2bins)} - 2^{-n/(2bins)}$.

In addition to the Wavelet coefficients *c*, also the analysis system *g*, *shift*, *M* can be returned, as can the length *Ls* of the input signal *f*, the frame bounds of the system *g*, *shift*, *M* and a flag indicating if a tight frame was used. These parameters are necessary to perform reconstruction with the inverse Wavelet transform wrapper `invwvlttrans`.

References: [10]

INVWVLTTRANS - Wavelet frame synthesis

Usage

```
fr = invwvlttrans(c,g,shift,M,tgtfl,fb,Ls)
fr = invwvlttrans(c,g,shift,M,tgtfl,fb)
fr = invwvlttrans(c,g,shift,M,tgtfl,Ls)
fr = invwvlttrans(c,g,shift,M,tgtfl)
fr = invwvlttrans(c,g,shift,M,Ls)
fr = invwvlttrans(c,g,shift,M)
```

Input parameters

c	Cell array of Wavelet coefficients
g	Cell array of Fourier transforms of the analysis Wavelets
shift	Vector of frequency shifts
M	Number of time steps
Ls	Original signal length
fb	Frame bounds (vector)
tgtfl	Tightflag (1 if frame is tight)

Output parameters

fr	Reconstructed signal

Description This is a wrapper function for the inverse painless Wavelet transform via nonstationary Gabor filterbank. Given the cell array *c* and the painless Wavelet frame *g*, *shift*, *M*, the corresponding dual frame is computed and the corresponding inverse Wavelet transform is performed.

If the original signal length *Ls* is specified, the synthesized signal will be truncated to length *Ls*. If the parameters *tgtfl* and *fb* are given, the system *g*, *shift*, *M* is assumed to be a tight frame and synthesis is performed using the original system.

References: [10]

A.6.3 SliCQ transform

SLICQ - Sliced constant-Q/variable-Q transform

Usage

```
[c,g,shift,M,Ls,sl_len,tr_area]
= slicq(f,fmin,fmax,bins,sl_len,tr_area,sr,M,min_win,Qvar)
= slicq(f,fmin,fmax,bins,sl_len,tr_area,sr,M,min_win)
= slicq(f,fmin,fmax,bins,sl_len,tr_area,sr,M)
= slicq(f,fmin,fmax,bins,sl_len,tr_area,sr)
= slicq(f,fmin,fmax,bins,sl_len,tr_area)
= slicq(f,fmin,fmax,bins,sl_len)
= slicq(f,fmin,fmax,bins)
= slicq(f,fmin,fmax)
= slicq(f,fmin)
c = slicq(...)
```

Input parameters

f Input signal

fmin Desired minimum frequency (in Hz)

fmax Desired maximum frequency (in Hz)

bins Bins per octave (constant or vector (for VQ))

sl_len Desired slice length (in samples)

tr_area Transition area length (in samples, $\leq sl\_len/2$)

sr Sampling rate (in Hz)

M Desired number of time steps per slice, if set to 0, a channel vector will be computed (M must be a multiple of 4 or will be set to $4 * ceil(M/4)$)

min_win Minimum filter bandwidth (default 16 samples)

Qvar Factor varying the bandwidth. $Qvar = X$ leads to a Q-factor of Q/X

Output parameters

c	Cell array of coefficients
g	Cell array of analysis filters
shift	Vector of frequency shifts of filters
M	Number of time steps per slice (vector or constant)
Ls	Original signal length
sl_len	Slice length
tr_area	Transition area length

Description This is a wrapper function for the sliced constant-Q nonstationary Gabor transform of the signal *f*. The signal is smoothly sliced into half-overlap segments of length *sl_len* weighted by a Tukey window with transition areas of length *tr_area* and total length of `sl_len/2 + tr_area`.

Subsequently, a constant-Q nonstationary Gabor transform with essential frequency range *fmin* to *fmax* and *bins* bins per octave will be applied to each segment using `nsgcqwin` with modified Blackman-Harris windows and `nsgtf`.

The additional parameters are an optional fixed number of time steps *M* per slice in each frequency channel and a bandwidth variation factor *Qvar*. Setting the minimum support *min_win* of the filters used helps in preserving shape and localization of low frequency filters, but may lead to a varying Q-factor in that frequency range.

See the help of `nsgcqwin` for more information on the constant-Q nonstationary Gabor transform.

References: [164], [92]

ISLICQ - Sliced constant-Q/variable-Q synthesis

Usage

```
[fr,gd] = islicq(c,g,shift,M,Ls,sl_len,tr_area)
fr = islicq(c,g,shift,M,Ls,sl_len,tr_area)
```

Input parameters

c	Cell array of coefficients
g	Cell array of analysis filters
shift	Vector of frequency shifts of filters
M	Number of time steps (vector/scalar)
Ls	Original signal length
sl_len	Slice length
tr_area	Transition area length

Output parameters

fr	Reconstructed signal
gd	Cell array of synthesis windows

Description This is a wrapper function for the inverse sliced constant-Q nonstationary Gabor transform. Given an array of coefficients *c* corresponding to a sliced nonstationary Gabor system *g*, *shift*, *M* with slice length *sl_len*, this function computes the corresponding synthesis operation.

That is, `nsdual` is used to compute the canonical dual frame of the system *g*, *shift*, *M* if possible. Afterwards, a sliced signal is synthesized using this dual system and the function `nsigtf`. Finally the synthesized signal is unsliced by the helper function `unslicing`.

If the output of `slicq` is used as input for this function, and the system *g*, *shift*, *M* used is a painless frame, then the originally analyzed function is reconstructed perfectly.

See the help of `nsgcqwin` for more information on the constant-Q nonstationary Gabor transform.

References: [164], [92]

A.7 NSGToolbox - Plotting tools

A.7.1 Nonstationary Gabor spectrogram

PLOTNSGT - Plot nonstationary Gabor coefficients

Usage

```
plotnsgt(c,shift,sr,varargin)
plotnsgt(c,shift,sr)
plotnsgt(c,shift)
```

Input parameters

c Array of coefficients.

shift Vector of time shifts

sr Sampling rate in Hz (default 1 Hz)

varargin Optional input parameters (see table below)

Description Given a coefficient array *c* and the time shift vector *shift*, this function plots the dB-scaled nonstationary Gabor spectrogram corresponding to *c*. To capture the correct position of the coefficients in the time frequency plane, the columns of the spectrogram (coefficients corresponding to the same time position) are stretched accordingly.

If additionally, the sampling rate *sr* is provided, time and frequency axes will be labeled properly.

If the coefficients were obtained using `nsgt_real`, the *realsig* switch should be used, otherwise only half the desired frequency range will be displayed. The shown frequency range can be controlled with the *cutout* parameter (default: 2) and the dynamic range of the spectrogram can be adjusted with *dynrange*.

Optional input arguments arguments can be supplied like this:

```
plotnsgt(c,shift,sr,'dynrange',dynrange)
```

The arguments must be character strings followed by an argument:

'dynrange',dynrange Colorscale dynamic range in dB (default 60 dB)

'cutout',cutout Desired part of the spectrogram, e.g. choice of '2' shows frequencies up to Nyquist ('X' shows the 'number_of_bins/X' lowest frequency bins)

'realsig',realsig Input coefficients are taken from a representation for real-valued signals

PLOTNSGTF - Plot nonstationary Gabor filterbank coefficients

Usage

```
plotnsgtf(c,shift,sr,fmin,fmax,bins,cutout,dynrange)
plotnsgtf(c,shift,sr,fmin,fmax,bins,cutout)
plotnsgtf(c,shift,sr,fmin,fmax,bins)
plotnsgtf(c,shift,sr,cutout,dynrange)
plotnsgtf(c,shift,sr,cutout)
plotnsgtf(c,shift,sr)
plotnsgtf(c,shift)
plotnsgtf(c)
```

Input parameters

c Array of coefficients.

shift Vector of frequency shifts

sr signal sample rate in Hz (default 1 Hz)

fmin Minimum frequency used in the transform

fmax Maximum frequency used in the transform

bins Bins per octave (in constant or vector form)

cutout Desired part of the spectrogram, e.g. choice of `2` shows frequencies up to Nyquist (`X` shows the *number_of_bins/X* lowest frequency bins)

dynrange Colorscale dynamic range in dB (default 60 dB)

Description Given a coefficient array *c* and the frequency shift vector *shift*, this function plots the dB-scaled nonstationary Gabor filterbank spectrogram corresponding to *c*. The vector *shift* and sampling rate *sr* are used to determine the correct time axis labels. The frequency axis is by default labeled by bin number.

For constant-Q nonstationary Gabor filterbanks, labeling with the actual center frequencies is supported, requiring the filterbank parameters *fmin*, *fmax* and *bins* as additional input.

The shown frequency range can be controlled with the *cutout* parameter (default: 2) and the dynamic range of the spectrogram can be adjusted with *dynrange*.

A.7.2 Dictionary plotting

PLOT_WINS - Plot nonstationary Gabor windows/filters

Usage

```
plot_wins(g,shift)
plot_wins(g,shift,normalize)
```

Input parameters

g	Cell array of windows/filters
shift	Vector of time/frequency shifts
normalize	Re-normalize the windows to have approximately uniform height

Description This helper function plots the distribution of the windows/filters of a nonstationary Gabor system/filterbank along the time/frequency axis. The shape of the windows/filters is determined from the cell array *g* and their position on the respective axis from the position vector *shift*.

A.7.3 Advanced spectrograms

PLOTSLICQ - PLOTNSGTF wrapper for sliced transforms (sliCQ)

Usage

```
plotslicq(c,shift,sr,fmin,fmax,bins,cutout,dynrange)
plotslicq(c,shift,sr,fmin,fmax,bins,cutout)
plotslicq(c,shift,sr,fmin,fmax,bins)
plotslicq(c,shift,sr,cutout,dynrange)
plotslicq(c,shift,sr,cutout)
plotslicq(c,shift,sr)
plotslicq(c,shift)
```

Input parameters

c	Array of coefficients.
shift	Vector of frequency shifts of windows

sr signal sample rate in Hz (default 1 Hz)

fmin Minimum frequency used in the transform

fmax Maximum frequency used in the transform

bins Bins per octave (in constant or vector form)

cutout Desired part of the spectrogram, e.g. choice of `2` shows frequencies up to Nyquist (`X` shows the *number_of_bins*/*X* lowest frequency bins)

dynrange Colorscale dynamic range in dB (default 60 dB)

Description This is a wrapper function for `plotnsgtf` that rearranges the coefficients of a sliced nonstationary Gabor filterbank, in particular those of a sliced constant-Q nonstationary Gabor filterbank, to resemble a full length transform.

For an explanation of the parameters, please refer to the help of `plotnsgtf`.

References: [92]

A.8 NSGToolbox - Window functions

A.8.1 Window function generator

WINFUNS - Window function generator

Usage

```
g = winfuns(name,x)
g = winfuns(name,N,L)
g = winfuns(name,N)
```

Input parameters

name String containing the window name

x Vector of sampling positions

N Window support (in samples)

L Output length (in samples)

Output parameters

g Output window

Description This function serves to compute a variety of standard and some more exotic window functions. Most of the functions used are detailed and discussed in classical papers (see references below), but several are included for special purposes in the toolbox only.

Given a character string *name* containing the name of the desired window function, the function offers 2 modes of operation. If the second input parameter is a vector x of sampling values, then the specified function is evaluated at the given points. If a window length N and optionally a signal length L are supplied, a symmetric, whole-point centered window with a support of N samples is produced and, given L, zero-extended to length L.

The following windows are available:

'hann' von Hann window. Forms a PU. The Hann window has a mainlobe with of 8/N, a PSL of -31.5 dB and decay rate of 18 dB/Octave.

'cos' Cosine window. This is the square root of the Hanning window. The cosine window has a mainlobe width of 6/N, a PSL of -22.3 dB and decay rate of 12 dB/Octave.

'rec' Rectangular window. The rectangular window has a mainlobe width of 4/N, a PSL of -13.3 dB and decay rate of 6 dB/Octave. Forms a PU. Alias: `'square'`

'tri' Triangular window.

'hamming' Hamming window. Forms a PU that sums to 1.08 instead of 1.0 as usual. The Hamming window has a mainlobe width of 8/N, a PSL of -42.7 dB and decay rate of 6 dB/Octave.

'blackman' Blackman window. The Blackman window has a mainlobe width of 12/N, a PSL of -58.1 dB and decay rate of 18 dB/Octave.

'blackharr' Blackman-Harris window. The Blackman-Harris window has a mainlobe width of 16/N, a PSL of -92.04 dB and decay rate of 6 dB/Octave.

'modblackharr' Modified Blackman-Harris window. This slightly modified version of the Blackman-Harris window has a mainlobe width of 16/N, a PSL of -90.24 dB and decay rate of 18 dB/Octave.

'nuttall' Nuttall window. The Nuttall window has a mainlobe width of 16/N, a PSL of -93.32 dB and decay rate of 18 dB/Octave.

'nuttall10' 2-term Nuttall window with 1 continuous derivative. Alias: `'hann'`.

'nuttall01' 2-term Nuttall window with 0 continuous derivatives. Alias: `'hamming'`.

'nuttall20' 3-term Nuttall window with 3 continuous derivatives. The window has a mainlobe width of 12/N, a PSL of -46.74 dB and decay rate of 30 dB/Octave.

'nuttall11' 3-term Nuttall window with 1 continuous derivative. The window has a mainlobe width of 12/N, a PSL of -64.19 dB and decay rate of 18 dB/Octave.

'nuttall02' 3-term Nuttall window with 0 continuous derivatives. The window has a mainlobe width of 12/N, a PSL of -71.48 dB and decay rate of 6 dB/Octave.

'nuttall30' 4-term Nuttall window with 5 continuous derivatives. The window has a mainlobe width of 16/N, a PSL of -60.95 dB and decay rate of 42 dB/Octave.

'nuttall21' 4-term Nuttall window with 3 continuous derivatives. The window has a mainlobe width of 16/N, a PSL of -82.60 dB and decay rate of 30 dB/Octave.

'nuttall12' 4-term Nuttall window with 1 continuous derivatives. Alias: `'nuttall'`.

'nuttall03' 4-term Nuttall window with 0 continuous derivatives. The window has a mainlobe width of 16/N, a PSL of -98.17 dB and decay rate of 6 dB/Octave.

'gauss' Truncated, stretched Gaussian: exp(-18*x^2) restricted to the interval]-.5,.5[.

'wp2inp' Warped Wavelet uncertainty equalizer (see WP 2 of the EU funded project UnlocX). This function is included as a test function for the Wavelet transform implementation and serves no other purpose in this toolbox.

References: [87], [123], [170]

A.9 NSGToolbox - Helper functions

A.9.1 Helper functions for the Onset-based transform

ONSETDET - Onset detection wrapper

Usage

```
[pos,V0]
= onsetdet(f,win_length,thre,range,multi,shift,showplot)
= onsetdet(f,win_length,thre,range,multi,shift)
= onsetdet(f,win_length,thre,range,multi)
= onsetdet(f,win_length,thre,range)
= onsetdet(f,win_length,thre)
= onsetdet(f,win_length)
pos = onsetdet(...)
```

Input parameters

f Signal to be analyzed (single channel only)

win_length Window length for the STFT analysis (in samples)

thre Peak-picking threshold

range Area of interest for the choice of local maxima

multi Area of interest multiplication factor for the peak-picking

shift Readjustment of the peaks (in $shift*win\_length/16$)

showplot Plot the results (0/1)

Output parameters

pos	Onset sequence
V0	Regular discrete Gabor transform of f

Description This routine produces a sequence of onsets using a straightforward realization of a spectral flux based onset detection process as described, e.g. by Dixon (see reference).

The spectral flux onset detection function is computed with a 16 times redundant Gabor transform using a Hann window, implemented in `specflux`.

Local maxima of the onset detection function are chosen as onset if larger than the local mean by at least the threshold parameter *thre*. This choice is performed by `peakpick`, a simple peakpicking algorithm.

A time slice is considered a local maximum if its spectral flux value is larger than those of the surrounding slices on an area of $+-range$. The local mean is computed as the mean value of the spectral flux function on an area corresponding to $-multi * range$ to $+range$ of the current position.

References: [10], [51]

SPECFLUX - Spectral flux onset detection function

Usage

```
[SF,V0] = specflux(f,win_length,tgap)
SF = specflux(f,win_length,tgap)
```

Input parameters

f	Input signal
win_length	Desired window length for the STFT
tgap	Time step for the STFT

Output parameters

SF	Spectral flux of *f*
V0	STFT coefficients of *f*

Description This is a helper function for `onsetdet` and not meant to be used individually.

Computes the spectral flux onset-detection function of f with a Hann window of length *win_ length*. The STFT is taken with time shift parameter *tgap* and *win_ length* frequency channels.

Externals: COMP_DGT_FB (LTFAT routine, included in NSGToolbox V0.1.0 and higher)

References: [51]

PEAKPICK - Peakpicking routine

Usage

```
peaks = peakpick(SF,thre,range,multi)
```

Input parameters

SF	Onset detection function
thre	Threshold value
range	Relevance area for local maximum
multi	Asymmetric extension factor for relevance area

Output parameters

peaks	Significant maxima of *SF*

Description This is a helper function for `onsetdet` and not meant to be used individually.

For an onset detection function *SF*, the routine picks only those local maxima that are larger than the local mean over an area of the form

by more than the threshold given by *thre*.

References: [52]

A.9.2 Helper functions for the sliCQ wrapper

SLICING - Cut a signal into uniform slices with half-overlap

Usage

```
f_sliced = slicing(f,sl_len,tr_area,Ls)
f_sliced = slicing(f,sl_len,tr_area)
```

Input parameters

f Signal to be sliced

sl_len Slice length (in samples, must be even)

tr_area Length of each transition area (in samples, optional, default is *ceil*(*sl_len*/16))

Ls Length of *f* (optional)

Output parameters

f_sliced Matrix containing the slices of f as columns

sl_len Possibly corrected slice length (in samples, even)

tr_area Possibly corrected transition area length (in samples, even)

Description This function cuts a signal into compactly supported pieces of length *sl_ len* using a uniform partition of unity composed of Tukey windows with plateau area *sl_len*/2 − *tr_area* and transition areas of length *tr_ area*. The resulting signal slices are stored in the columns of the output.

References: [92]

UNSLICING - Reconstruct full signal from uniform half-overlap slices

Usage

```
fr = unslicing(f_sliced,sl_len,tr_area,slices)
```

Input parameters

f_sliced Matrix containing signal slices as columns

sl_len Slice length (in samples)

tr_area “Length of each transition area (in samples, optional, default is *ceil*(*sl_len*/16))

slices Number of slices in *f_ sliced* (optional)

Output parameters

fr Signal resulting from the overlap-add procedure

Description This function performs a windowed overlap-add procedure on the columns of the input matrix *f_ sliced*. The signal is assumed to have been cut with a Tukey window of plateau length $sl\_len/2 - tr\_area$ and transition areas of length *tr_ area*. The window used for the overlap-add process will be a smooth dual window to this Tukey window in the sense that unaltered input created by the routine **slicing** will recreate the original input signal.

References: [92]

A.10 NSGToolbox - Examples

A.10.1 Nonstationary Gabor examples

DEMO_NSGT - Onset Detection type Nonstationary Gabor transform usage demo

This script sets up a nonstationary Gabor frame with the specified parameters, computes windows and corresponding canonical dual windows and a test signal, and plots the windows and the energy of the coefficients.

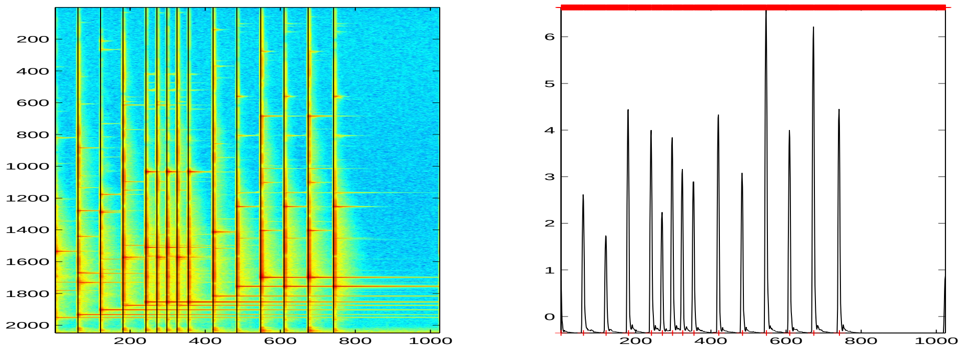


Figure A.1: Onset detection results: This figure shows a regular spectrogram with marked onsets and the spectral flux function with marked onsets.

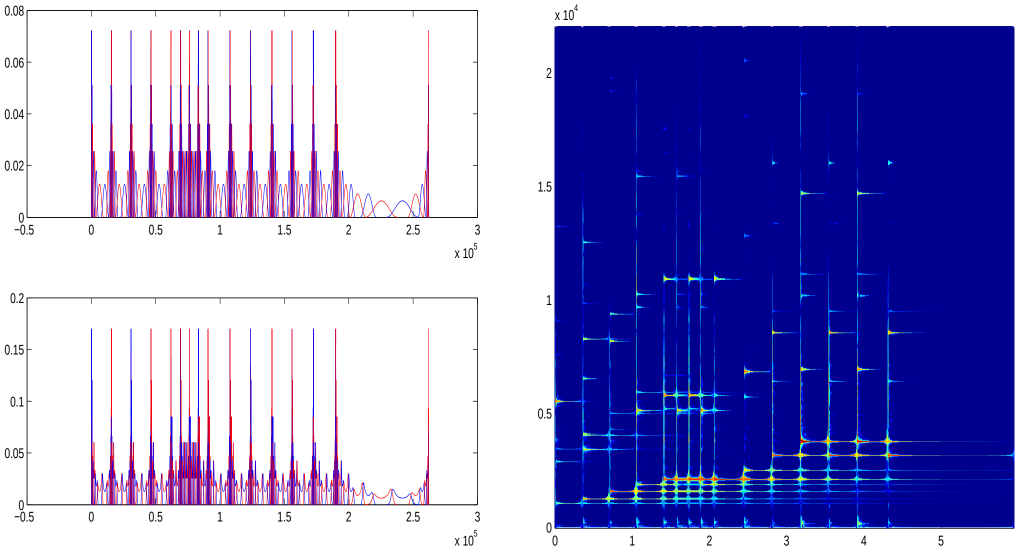


Figure A.2: (left) windows + dual windows: This figure shows the window functions used and the corresponding canonical dual windows. (right) spectrogram (absolute value of coefficients in dB): This figure shows a (color coded) image of the nsgt coefficient modulus.

Output

```
Relative error of reconstruction (should be close to zero.):
3.294496e-16
```

DEMO_NSGTF - Nonstationary Gabor filterbank usage demo

This script sets up different nonstationary Gabor filterbank frames with the specified parameters, computes filters and corresponding canonical dual filters as well as the transform and reconstruction of a test signal, and plots the filters and the energy of the coefficients.

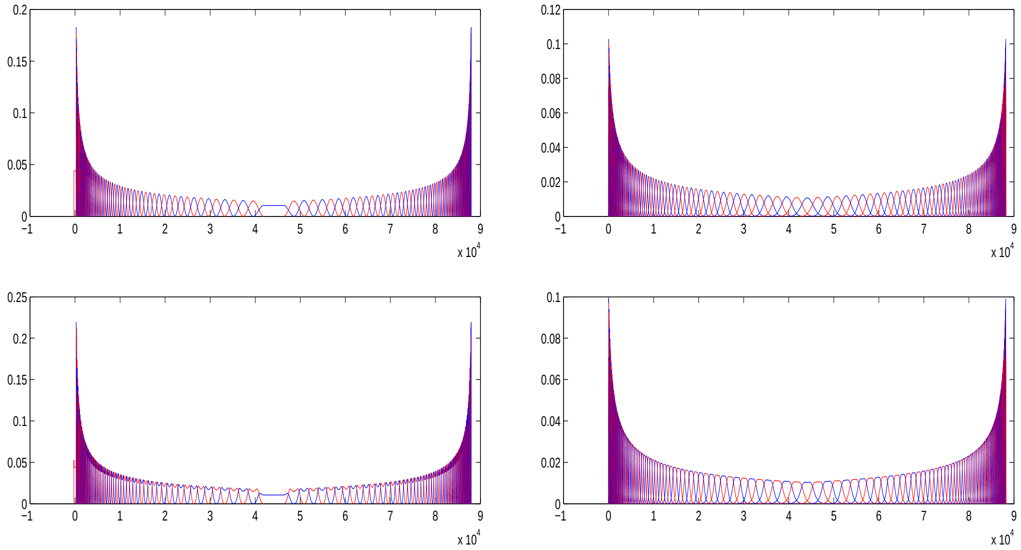


Figure A.3: (left) filters + dual filters (constant-Q): This figure shows the filter functions used in the constant-Q filterbank and the corresponding canonical dual filters. (right) filters + dual filters (ERBlet): This figure shows the filter functions used in the ERBlet filterbank and the corresponding canonical dual filters.

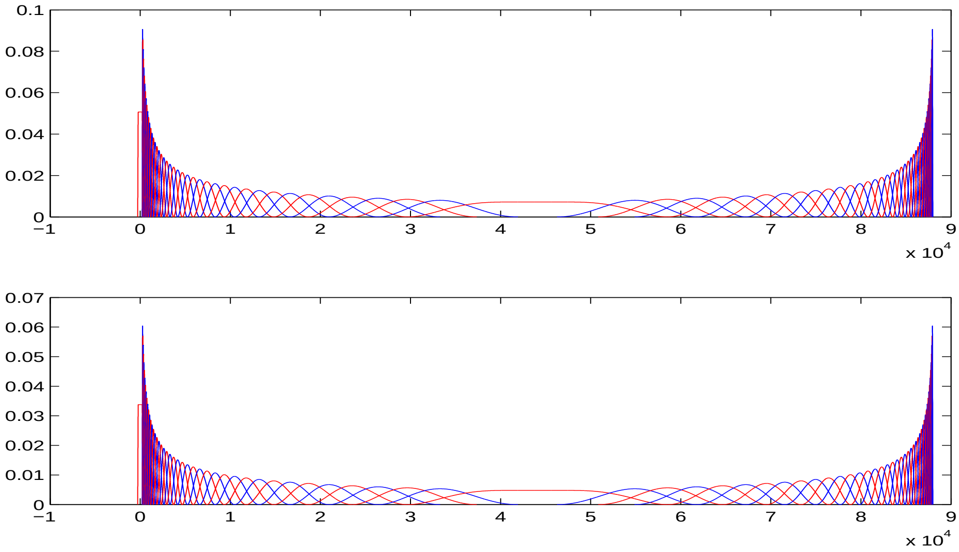


Figure A.4: filters + dual filters (Wavelet): This figure shows the filter functions used in the Wavelet filterbank and the corresponding canonical dual filters.

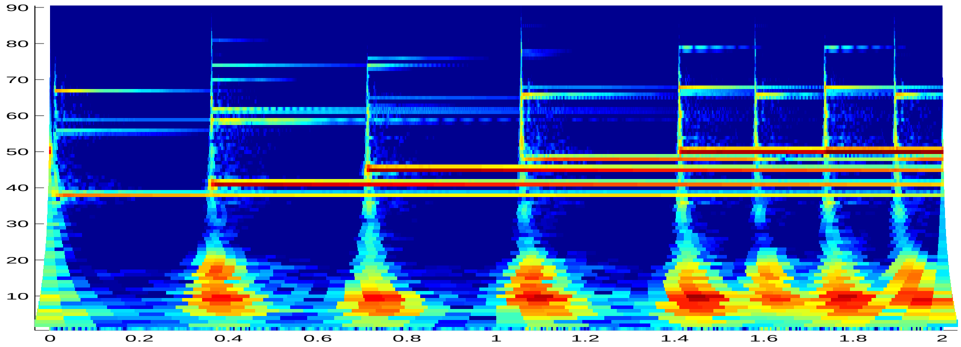


Figure A.5: constant-Q spectrogram (absolute value of coefficients in dB): This figure shows a (color coded) image of the constant-Q coefficient modulus.

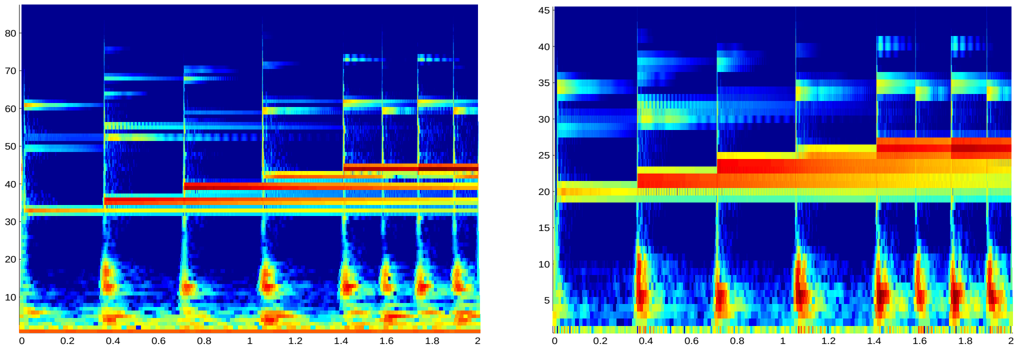


Figure A.6: (left) ERBlet spectrogram and (right) Wavelet spectrogram (absolute value of coefficients in dB): This figure shows a (color coded) image of the coefficient modulus.

Output

```
Relative error of constant-Q reconstruction:
6.435922e-16
Relative error of ERBlet reconstruction:
5.597245e-16
Relative error of Wavelet reconstruction:
5.137871e-16
```

DEMO_SLICQ - Sliced constant-Q usage/comparison demo

This script sets up nonstationary Gabor filterbank frames with the specified parameters, computes filters and corresponding canonical dual filters as well as the transform and reconstruction of a test signal, and compares the respective analysis/synthesis filters and spectrograms.

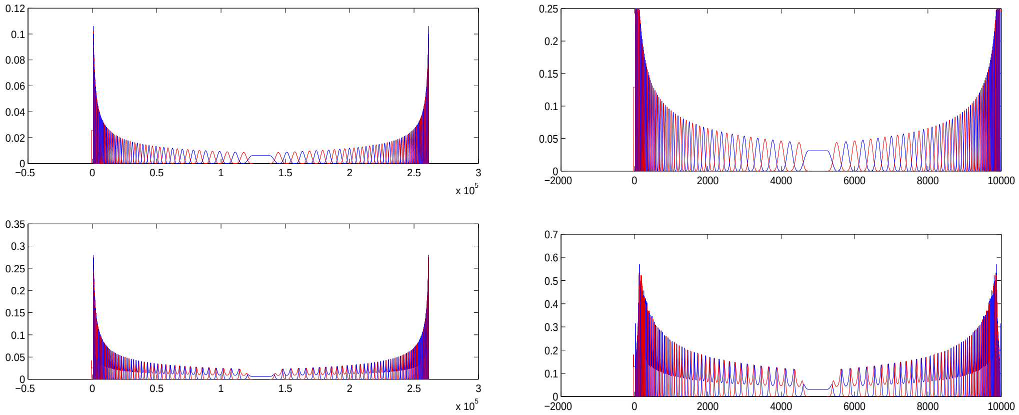


Figure A.7: filters + dual filters (left - constant-Q, right - sliCQ): This figure shows the filter functions used in the constant-Q and sliCQ filterbanks and the corresponding canonical dual filters.

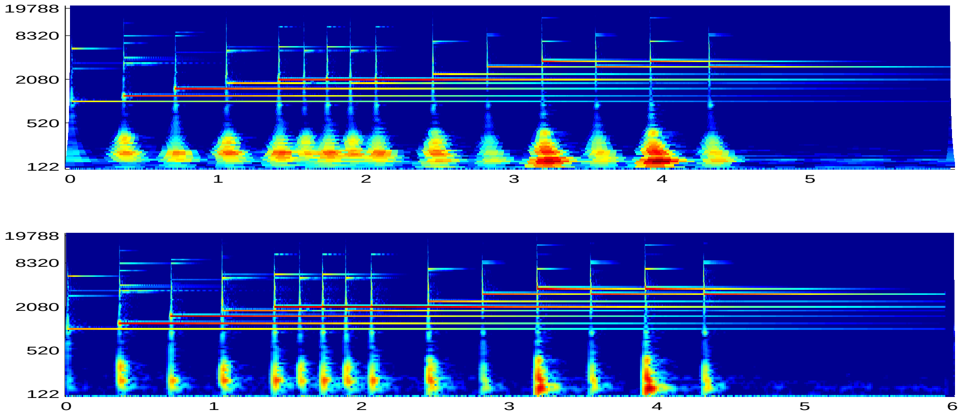


Figure A.8: constant-Q/sliCQ spectrogram (absolute value of coefficients in dB): This figure shows (color coded) images of the constant-Q and sliced constant-Q coefficient modulus.

Output

```
Relative error of constant-Q reconstruction:
7.646168e-16
Relative error of sliCQ reconstruction:
3.246836e-16
```

A.11 Nonseparable Gabor systems in LTFAT

A.11.1 Gabor systems

DGT - Discrete Gabor transform

Usage

```
c=dgt(f,g,a,M);
c=dgt(f,g,a,M,L);
c=dgt(f,g,a,M,'lt',lt);
[c,Ls]=dgt(...);
```

Input parameters

f	Input data.
g	Window function.
a	Length of time shift.
M	Number of channels.
L	Length of transform to do.
lt	Lattice type (for nonseparable lattices).

Output parameters

c	$M \times N$ array of coefficients.
Ls	Length of input signal.

Description `dgt(f,g,a,M)` computes the Gabor coefficients (also known as a windowed Fourier transform) of the input signal f with respect to the window g and parameters a and M. The output is a vector/matrix in a rectangular layout.

The length of the transform will be the smallest multiple of a and M that is larger than the signal. f will be zero-extended to the length of the transform. If f is a matrix, the transformation is applied to each column. The length of the transform done can be obtained by `L=size(c,2)*a;`

The window g may be a vector of numerical values, a text string or a cell array. See the help of `gabwin` for more details.

`dgt(f,g,a,M,L)` computes the Gabor coefficients as above, but does a transform of length L. f will be cut or zero-extended to length L before the transform is done.

`[c,Ls]=dgt(f,g,a,M)` or `[c,Ls]=dgt(f,g,a,M,L)` additionally returns the length of the input signal *f*. This is handy for reconstruction:

```
[c,Ls]=dgt(f,g,a,M);
fr=idgt(c,gd,a,Ls);
```

will reconstruct the signal f no matter what the length of f is, provided that gd is a dual window of g.

`[c,Ls,g]=dgt(...)` additionally outputs the window used in the transform. This is useful if the window was generated from a description in a string or cell array.

The Discrete Gabor Transform is defined as follows: Consider a window g and a one-dimensional signal f of length L and define $N = L/a$. The output from `c=dgt(f,g,a,M)` is then given by:

$$c\,(m+1,n+1) = \sum_{l=0}^{L-1} f(l+1)\overline{g(l-an+1)}e^{-2\pi i l m/M}$$

where $m = 0,\ldots,M-1$ and $n = 0,\ldots,N-1$ and $l-an$ is computed modulo L.

Nonseparable lattices: `dgt(f,g,a,M,'lt',lt)` computes the DGT for a non-separable lattice given by the time-shift a, number of channels M and lattice type lt. Please see the help of matrix2latticetype for a precise description of the parameter lt.

The nonseparable discrete Gabor transform is defined as follows: Consider a window g and a one-dimensional signal f of length L and define $N = L/a$. The output from `c=dgt(f,g,a,M,L,lt)` is then given by:

$$c\,(m+1,n+1) = \sum_{l=0}^{L-1} f(l+1)\overline{g(l-an+1)}e^{-2\pi i l(m+w(n))/M}$$

where $m = 0,\ldots,M-1$ and $n = 0,\ldots,N-1$ and $l-an$ are computed modulo L. The additional offset w is given by $w(n) = \text{mod}(n \cdot lt_1, lt_2)/lt_2$ in the formula above.

Additional parameters: `dgt` takes the following flags at the end of the line of input arguments:

'freqinv' Compute a DGT using a frequency-invariant phase. This is the default convention described above.

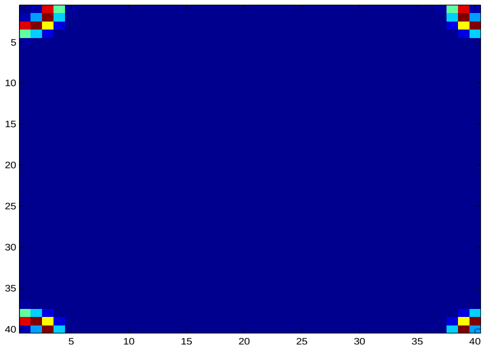
'timeinv' Compute a DGT using a time-invariant phase. This convention is typically used in filter bank algorithms.

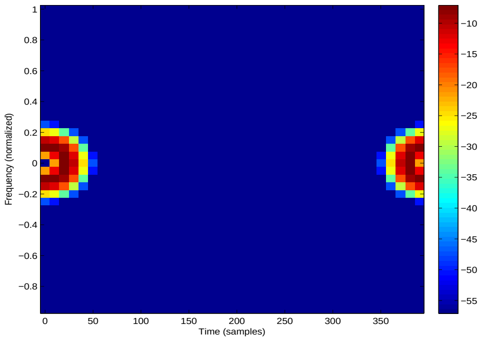
Examples: In the following example we create a Hermite function, which is a complex-valued function with a circular spectrogram, and visualize the coefficients using both `imagesc` and `plotdgt`:

```
a=10;
M=40;
L=a*M;
h=pherm(L,4); % 4th order hermite function.
c=dgt(h,'gauss',a,M);

% Simple plot: The squared modulus of the coefficients on
% a linear scale
figure(1);
imagesc(abs(c).^2);

% Better plot: zero-frequency is displayed in the middle,
% and the coefficients are show on a logarithmic scale.
figure(2);
plotdgt(c,a,'dynrange',50);
```





References: [71], [84]

IDGT - Inverse discrete Gabor transform

Usage

```
f=idgt(c,g,a);
f=idgt(c,g,a,Ls);
f=idgt(c,g,a,Ls,lt);
```

Input parameters

c	Array of coefficients.
g	Window function.
a	Length of time shift.
Ls	Length of signal.
lt	Lattice type (for non-separable lattices)

Output parameters

f	Signal.

Description `idgt(c,g,a)` computes the Gabor expansion of the input coefficients c with respect to the window g and time shift a. The number of channels is deduced from the size of the coefficients c.

`idgt(c,g,a,Ls)` does as above but cuts or extends f to length Ls.

`[f,g]=idgt(...)` additionally outputs the window used in the transform. This is useful if the window was generated from a description in a string or cell array.

For perfect reconstruction, the window used must be a dual window of the one used to generate the coefficients.

The window g may be a vector of numerical values, a text string or a cell array. See the help of `gabwin` for more details.

If g is a row vector, then the output will also be a row vector. If c is 3-dimensional, then `idgt` will return a matrix consisting of one column vector for each of the TF-planes in c.

Assume that `f=idgt(c,g,a,L)` for an array c of size $M \times N$. Then the following holds for $k = 0, \ldots, L-1$:

$$f(l+1) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} c(m+1, n+1) e^{2\pi i m l/M} g(l - an + 1)$$

Non-separable lattices: `idgt(c,g,a,'lt',lt)` computes the Gabor expansion of the input coefficients c with respect to the window g, time shift a and lattice type lt. Please see the help of matrix2latticetype for a precise description of the parameter lt.

Assume that `f=dgt(c,g,a,L,lt)` for an array c of size $M \times N$. Then the following holds for $k = 0, \ldots, L-1$:

$$f(l+1) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} c(m+1, n+1) e^{2\pi i m l / M} g(l - an + 1)$$

Additional parameters: idgt takes the following flags at the end of the line of input arguments:

'freqinv'	Compute an IDGT using a frequency-invariant phase. This is the default convention described above.
'timeinv'	Compute an IDGT using a time-invariant phase. This convention is typically used in filter bank algorithms.

Examples: The following example demostrates the basic pricinples for getting perfect reconstruction (short version):

```
f=greasy;             % test signal
a=32;                 % time shift
M=64;                 % frequency shift
ga={'blackman',128}; % analysis window

[c,Ls]=dgt(f,ga,a,M);    % analysis

% ... do interesting stuff to c at this point ...

r=idgt(c,{'dual',ga},a,Ls); % synthesis

norm(f-r)           % test
```

This code produces the following output:

```
ans =
   6.3564e-15
```

The following example does the same as the previous one, with an explicit construction of the analysis and synthesis windows:

```
f=greasy;      % test signal
a=32;          % time shift
M=64;          % frequency shift
Ls=length(f); % signal length

% Length of transform to do
```

```
L=dgtlength(Ls,a,M);

% Analysis and synthesis window
ga=firwin('blackman',128);
gs=gabdual(ga,a,M,L);

c=dgt(f,ga,a,M); % analysis

% ... do interesting stuff to c at this point ...

r=idgt(c,gs,a,Ls);  % synthesis

norm(f-r)  % test
```

This code produces the following output:

```
ans =
   6.3404e-15
```

DGTLENGTH - DGT length from signal

Usage

```
L=dgtlength(Ls,a,M);
L=dgtlength(Ls,a,M,lt);
```

Description `dgtlength(Ls,a,M)` returns the length of a Gabor system that is long enough to expand a signal of length *Ls*. Please see the help on dgt for an explanation of the parameters *a* and *M*.

If the returned length is longer than the signal length, the signal will be zero-padded by dgt.

A valid transform length must be divisable by both *a* and *M*. This means that the minumal admissable transform length is

```
Lsmallest = lcm(a,M);
```

and all valid transform lengths are multipla of *Lsmallest*

Nonseparable lattices: `dgtlength(Ls,a,M,lt)` does as above for a nonseparable lattice with lattice-type *lt*. For non-separable lattices, there is the additinal requirement on the transform length, that the structure of the lattice must be periodic. This gives a minimal transform length of

```
Lsmallest = lcm(a,M)*lt(2);
```

A.11.2 Reconstructing windows

GABDUAL - Canonical dual window of Gabor frame

Usage

```
gd=gabdual(g,a,M);
gd=gabdual(g,a,M,L);
gd=gabdual(g,a,M,'lt',lt);
```

Input parameters

g	Gabor window.
a	Length of time shift.
M	Number of channels.
L	Length of window. (optional)
lt	Lattice type (for non-separable lattices).

Output parameters

gd	Canonical dual window.

Description `gabdual(g,a,M)` computes the canonical dual window of the discrete Gabor frame with window g and parameters a, M.

The window g may be a vector of numerical values, a text string or a cell array. See the help of `gabwin` for more details.

If the length of g is equal to M, then the input window is assumed to be an FIR window. In this case, the canonical dual window also has length of M. Otherwise the smallest possible transform length is chosen as the window length.

`gabdual(g,a,M,L)` returns a window that is the dual window for a system of length L. Unless the dual window is a FIR window, the dual window will have length L.

`gabdual(g,a,M,'lt',lt)` does the same for a non-separable lattice specified by lt. Please see the help of matrix2latticetype for a precise description of the parameter lt.

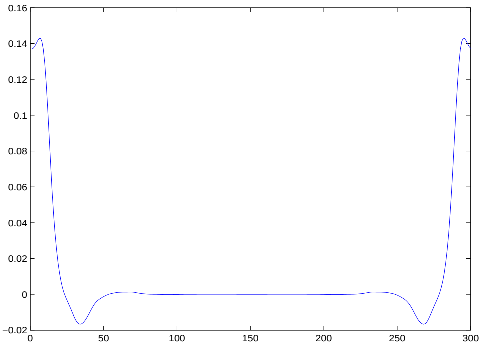
If $a > M$ then the dual window of the Gabor Riesz sequence with window g and parameters a and M will be calculated.

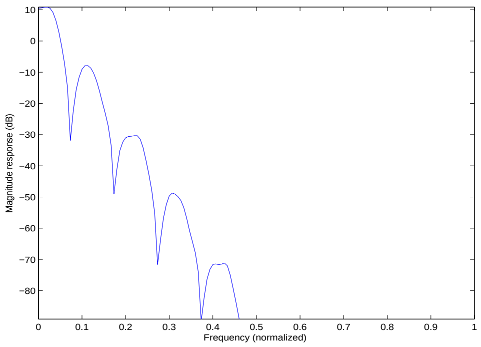
Examples: The following example shows the canonical dual window of the Gaussian window:

```
a=20;
M=30;
L=300;
g=pgauss(L,a*M/L);
gd=gabdual(g,a,M);

% Simple plot in the time-domain
figure(1);
plot(gd);

% Frequency domain
figure(2);
magresp(gd,’dynrange’,100);
```





GABTIGHT - Canonical tight window of Gabor frame

Usage

```
gt=gabtight(a,M,L);
gt=gabtight(g,a,M);
gt=gabtight(g,a,M,L);
gd=gabtight(g,a,M,’lt’,lt);
```

Input parameters

g	Gabor window.
a	Length of time shift.
M	Number of modulations.

L	Length of window. (optional)
lt	Lattice type (for non-separable lattices).

Output parameters

gt	Canonical tight window, column vector.

Description `gabtight(a,M,L)` computes a nice tight window of length L for a lattice with parameters a, M. The window is not an FIR window, meaning that it will only generate a tight system if the system length is equal to L.

`gabtight(g,a,M)` computes the canonical tight window of the Gabor frame with window g and parameters a, M.

The window g may be a vector of numerical values, a text string or a cell array. See the help of `gabwin` for more details.

If the length of g is equal to M, then the input window is assumed to be a FIR window. In this case, the canonical dual window also has length of M. Otherwise the smallest possible transform length is chosen as the window length.

`gabtight(g,a,M,L)` returns a window that is tight for a system of length L. Unless the input window g is a FIR window, the returned tight window will have length L.

`gabtight(g,a,M,'lt',lt)` does the same for a non-separable lattice specified by *lt*. Please see the help of matrix2latticetype for a precise description of the parameter *lt*.

If $a > M$ then an orthonormal window of the Gabor Riesz sequence with window g and parameters a and M will be calculated.

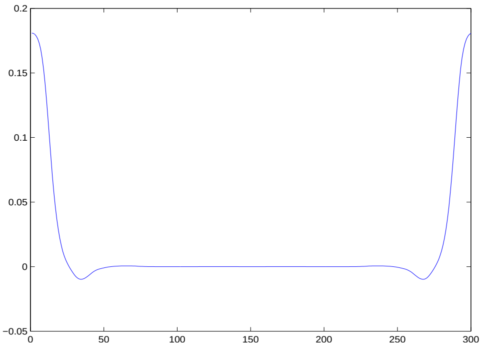
Examples: The following example shows the canonical tight window of the Gaussian window. This is calculated by default by gabtight if no window is specified:

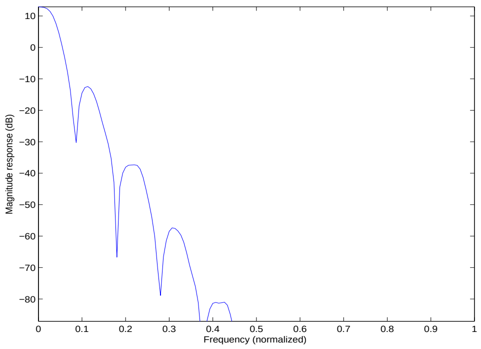
```
a=20;
M=30;
L=300;
gt=gabtight(a,M,L);

% Simple plot in the time-domain
figure(1);
plot(gt);

% Frequency domain
```

```
figure(2);
magresp(gt,'dynrange',100);
```





A.11.3 Support for nonseparable lattices

MATRIX2LATTICETYPE - Convert matrix form to standard lattice description

Usage

```
[a,M,lt] = matrix2latticetype(L,V);
```

Description `[a,M,lt]=matrix2latticetype(L,V)` converts a 2×2 integer matrix description into the standard description of a lattice using the a, M and lt. The conversion is *only* valid for the specified transform length L.

The lattice type lt is a 1×2 vector $[lt_1, lt_2]$ denoting an irreducible fraction lt_1/lt_2. This fraction describes the distance in frequency (counted in frequency channels) that each coefficient is offset when moving in time by the time-shift of a. Some examples: `lt=[0 1]` defines a square lattice, `lt=[1 2]` defines the quinqunx (almost hexagonal) lattice, `lt=[1 3]` describes a lattice with a $1/3$ frequency offset for each time shift and so forth.

An example:

```
[a,M,lt] = matrix2latticetype(120,[10 0; 5 10])
```

This code produces the following output:

```
a =
    10
M =
    12
lt =
     1     2
```

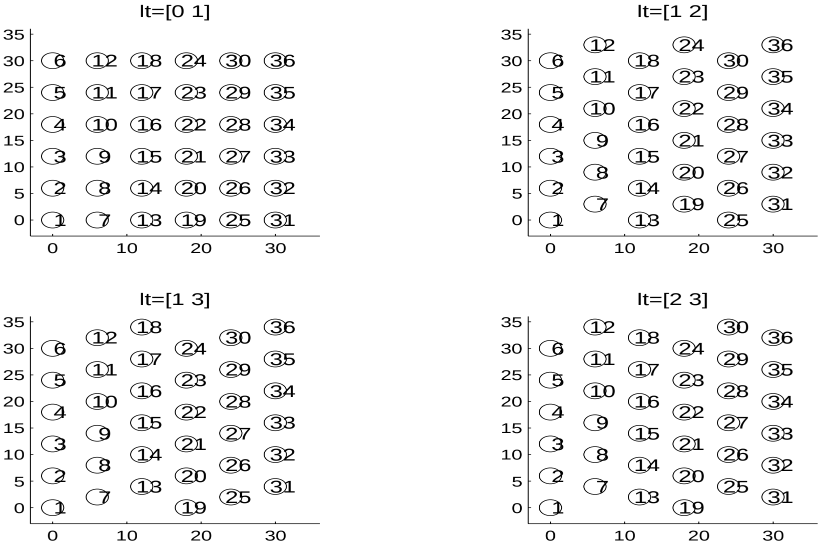
Coefficient layout: The following code generates plots which show the coefficient layout and enumeration of the first 4 lattices in the time-frequecy plane:

```
a=6;
M=6;
L=36;
b=L/M;
N=L/a;
cw=3;
ftz=12;

[x,y]=meshgrid(a*(0:N-1),b*(0:M-1));

lt1=[0 1 1 2];
lt2=[1 2 3 3];

for fignum=1:4
  subplot(2,2,fignum);
  z=y;
  if lt2(fignum)>0
    z=z+mod(lt1(fignum)*x/lt2(fignum),b);
  end;
  for ii=1:M*N
    text(x(ii)-cw/4,z(ii),sprintf('%2.0i',ii),'Fontsize',ftz);
    rectangle('Curvature',[1 1], 'Position', ...
     [x(ii)-cw/2,z(ii)-cw/2,cw,cw]);
  end;
  axis([-cw L -cw L]);
  axis('square');
  title(sprintf('lt=[%i %i]',lt1(fignum),lt2(fignum)), ...
   'Fontsize',ftz);
end;
```



lt=[0 1]

lt=[1 2]

lt=[1 3]

lt=[2 3]

LATTICETYPE2MATRIX - Convert lattice description to matrix form

Usage

```
V=latticetype2matrix(L,a,M,lt);
```

Description `V=latticetype2matrix(L,a,M,lt)` converts a standard description of a lattice using the a, M and lt parameters into a 2×2 integer matrix description. The conversion is *only* valid for the specified transform length L.

The output will be in lower triangular Hemite normal form.

For more information, see `http://en.wikipedia.org/wiki/Hermite_normal_form`.

An example:

```
V = latticetype2matrix(120,10,12,[1 2])
```

This code produces the following output:

```
V =
    10     0
     5    10
```

SHEARFIND - Shears for transformation of a general lattice to separable

Usage

```
[s0,s1,br] = shearfind(L,a,M,lt);
```

Description `[s0,s1,br]=shearfind(L,a,M,lt)` computes three numbers, the first two represent a frequency and time shear respectively. With the returned choices of s_0 and s_1 one can transform an initial lattice given by a, M and lt into a separable (rectangular) lattice given by

$$a_r = \frac{aL}{b_r M}, \quad M_r = \frac{L}{b_r}.$$

If s_0 is non-zero, the transformation from general to separable lattice requires a frequency-side shear. Similarly, if s_1 is non-zero, a time-side shear is required.

Bibliography

[1] "AMToolbox - The Auditory Modelling Toolbox," http://amtoolbox.sourceforge.net/.

[2] "LTFAT - The Large Time-Frequency Analysis Toolbox," http://ltfat.sourceforge.net/.

[3] "The on-line encyclopedia of integer sequences," http://oeis.org.

[4] M. D. Abolhassani and Y. Salimpour, "A human auditory tuning curves matched wavelet function," in *Engineering in Medicine and Biology Society (EMBS 2008). 30th Annual International Conference of the IEEE*, Vancouver, Canada, August 20–24 2008, pp. 2956–2959.

[5] L. Abreu and M. Dörfler, "An Inverse Problem for Localization Operators," *Preprint, arXiv: 1202.5841*, 2012.

[6] J. Allen and L. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[7] P. Auscher, "Remarks on the local Fourier bases," in *Wavelets: Mathematics and Applications*, ser. Stud. Adv. Math., J. Benedetto and M. Frazier, Eds. Boca Raton: CRC Press, 1994, pp. 203–218.

[8] L. Auslander, I. Gertner, and R. Tolimieri, "The discrete Zak transform application to time-frequency analysis and synthesis of nonstationary signals," vol. 39, no. 4, pp. 825–835, 1991.

[9] R. Balan, J. G. Christensen, I. A. Krishtal, K. A. Okoudjou, and J. L. Romero, "Multi-window Gabor frames in amalgam spaces," *Preprint, arXiv: 1108.6108*.

[10] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. A. Velasco, "Theory, implementation and applications of nonstationary Gabor Frames," *J. Comput. Appl. Math.*, vol. 236, no. 6, pp. 1481–1496, 2011.

[11] P. Balazs, H. G. Feichtinger, M. Hampejs, and G. Kracher, "Double preconditioning for the Gabor frame operator," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. Volume 3, 14-19 May 2006*, May 2006, pp. III–408 – III–411.

[12] M. J. Bastiaans, "A sampling theorem for the complex spectrogram and Gabor's expansion of a signal in Gaussian elementary signals," *Opt. Eng.*, vol. 20, no. 4, pp. 594–598, July/August 1981.

[13] M. J. Bastiaans and M. C. W. Geilen, "On the discrete Gabor transform and the discrete Zak transform," *Sig. Proc.*, vol. 49, no. 3, pp. 151–166, 1996.

[14] M. J. Bastiaans and A. J. van Leest, "From the rectangular to the quincunx Gabor lattice via fractional Fourier transformation," *IEEE Signal Proc. Letters*, vol. 5, no. 8, pp. 203–205, 1998.

[15] ——, "Modified zak transform for the quincunx-type gabor lattice," in *Time-Frequency and Time-Scale Analysis, 1998. Proceedings of the IEEE-SP International Symposium on.* IEEE, 1998, pp. 173–176.

[16] ——, "Gabor's signal expansion and the Gabor transform based on a non-orthogonal sampling geometry," in *Signal Processing and its Applications, Sixth International, Symposium on. 2001*, vol. 1. IEEE, 2001, pp. 162–163.

[17] ——, "Gabor's signal expansion for a non-orthogonal sampling geometry," in *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*, B. Boashash, Ed. Oxford, UK: Elsevier, 2003, pp. 252–259.

[18] K. Bauer, D. Sen, and P. Zvengrowski, "A generalized Goursat lemma," *Preprint, arXiv: 11009.0024v2.*

[19] F. Baumgarte, "Improved audio coding using a psychoacoustic model based on a cochlear filter bank," *IEEE Speech Audio Proc.*, vol. 10, no. 7, pp. 495–503, October 2002.

[20] J. J. Benedetto, C. Heil, and D. F. Walnut, "Gabor systems and the Balian-Low theorem," in *Gabor Analysis and Algorithms: Theory and Applications*, ser. Appl. Numer. Harmon. Anal. Boston, MA: Birkhäuser Boston, 1998, pp. 85–122.

[21] K. Bittner, "Wilson bases on the interval," in *Advances in Gabor Analysis*, ser. Appl. Numer. Harmon. Anal., H. Feichtinger and T. Strohmer, Eds. Boston, MA: Birkhäuser Boston, 2003, pp. 197–221.

[22] R. Boas, "Poisson's summation formula in l^2." *J. London Math. Soc.*, vol. 21, pp. 102–105, 1946.

[23] H. Bölcskei, F. Hlawatsch, and H. G. Feichtinger, "Frame-theoretic analysis of oversampled filter banks," *IEEE Trans. Sig. Proc.*, vol. 46, no. 12, pp. 3256–3268, 1998.

[24] H. Bölcskei and A. J. E. M. Janssen, "Gabor frames, unimodularity, and window decay," *J. Fourier Anal. Appl.*, vol. 6, no. 3, pp. 255–276, 2000.

[25] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *J. Acoust. Soc. Am.*, vol. 92, no. 5, pp. 2698–2701, 1992.

[26] J. Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991.

[27] W. Calhoun, "Counting the subgroups of some finite groups," *Amer. Math. Monthly*, vol. 94, pp. 54–59, 1987.

[28] P. G. Casazza, "The art of frame theory." *Taiwanese J. Math.*, vol. 4, no. 2, pp. 129–201, 2000.

[29] P. G. Casazza and O. Christensen, "Gabor frames over irregular lattices," *Adv. Comput. Math.*, vol. 18, no. 2-4, pp. 329–344, 2003.

[30] P. G. Casazza and M. Fickus, "Fourier transforms of finite chirps." *EURASIP J. Adv. Sig. Proc.*, pp. 1–7, 2006.

[31] O. Christensen, *An Introduction to Frames and Riesz Bases.*, ser. Appl. and Numer. Harmon. Anal. Boston: Birkhäuser, 2003.

[32] ——, "Pairs of dual Gabor frame generators with compact support and desired frequency localization." *Appl. Comput. Harmon. Anal.*, vol. 20, no. 3, pp. 403–410, 2006.

[33] ——, *Frames and Bases. An Introductory Course.* Applied and Numerical Harmonic Analysis. Basel Birkhäuser, 2008.

[34] O. Christensen, H. G. Feichtinger, and S. Paukner, *Gabor Analysis for Imaging.* Springer Berlin, 2011, vol. 3, pp. 1271–1307.

[35] O. Christensen, H. Kim, and R. Y. Kim, "Gabor windows supported on [-1, 1] and dual windows with small support," *Adv. Comput. Math.*, vol. 36, no. 4, pp. 525–545, 2012.

[36] ——, “Gabor windows supported on $[-1, 1]$ and compactly supported dual windows.” *Appl. Comput. Harmon. Anal.*, vol. 28, no. 1, pp. 89 – 103, January 2010.

[37] O. Christensen and R. Y. Kim, “On dual Gabor frame pairs generated by polynomials,” *J. Fourier Anal. Appl.*, vol. 16, no. 1, pp. 1–16, 2010.

[38] O. Christensen and E. Osgooei, “On frame properties for Fourier-like systems,” *J. Approx. Theory*, vol. 172, p. 47âĂŞ57, 2013.

[39] O. Christensen and W. Sun, “Explicitly given pairs of dual frames with compactly supported generators and applications to irregular B-splines,” *J. Approx. Theory*, vol. 151, no. 2, pp. 155–163, 2008.

[40] C. K. Chui and X. Shi, “Inequalities of Littlewood-Paley type for frames and wavelets.” *SIAM J. Math. Anal.*, vol. 24, no. 1, pp. 263–277, 1993.

[41] ——, “Wavelets of Wilson type with arbitrary shapes.” *Appl. Comput. Harmon. Anal.*, vol. 8, no. 1, pp. 1–23, 2000.

[42] J. B. Conway, *A Course in Functional Analysis. 2nd ed.* New York: Springer, 1990.

[43] J. Cooley and J. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Math. Comput*, vol. 19, no. 90, pp. 297–301, 1965.

[44] M. Cranitch, M. Cychowski, and D. FitzGerald, “Towards an Inverse Constant Q Transform,” in *Audio Engineering Society Convention 120*, 2006.

[45] G. Călugăreanu, “The total number of subgroups of a finite abelian group,” *Sci. Math. Jpn.*, vol. 60, pp. 157–167, 2004.

[46] Z. Cvetković and J. D. Johnston, “Nonuniform oversampled filter banks for audio signal processing,” *IEEE Trans. Speech Audio Proc.*, vol. 11, no. 5, pp. 393–399, September 2003.

[47] I. Daubechies, *Ten lectures on wavelets.*, ser. CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics, 1992, vol. 61.

[48] I. Daubechies, A. Grossmann, and Y. Meyer, “Painless nonorthogonal expansions,” *J. Math. Phys.*, vol. 27, no. 5, pp. 1271–1283, May 1986.

[49] I. Daubechies, S. Jaffard, and J. L. Journé, “A simple Wilson orthonormal basis with exponential decay,” *SIAM J. Math. Anal.*, vol. 22, pp. 554–573, 1991.

[50] F. de Mesmay, Y. Voronenko, and M. Püschel, "Offline library adaptation using automatically generated heuristics," in *International Parallel and Distributed Processing Symposium (IPDPS)*, 2010.

[51] Z. Ditzian, "Approximation on Banach spaces of functions on the sphere," *J. Approx. Theory*, vol. 140, no. 1, pp. 31–45, 2006.

[52] S. Dixon, "Onset Detection Revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, September 2006, pp. 133–137.

[53] M. Dolson, "The phase vocoder: a tutorial," *Computer Musical Journal*, vol. 10, no. 4, pp. 11–27, 1986.

[54] M. Dörfler, "Quilted Gabor frames - A new concept for adaptive time-frequency representation," *Advances in Applied Mathematics*, vol. 47, no. 4, pp. 668 – 687, Oct. 2011.

[55] M. Dörfler and E. Matusiak, "Nonstationary Gabor Frames - Existence and Construction," *Preprint, arXiv: 1112.5262*, 2011.

[56] ——, "Nonstationary Gabor Frames - Approximately Dual Frames and Reconstruction Errors." *Preprint, arXiv:1301.1802*, 2012.

[57] R. J. Duffin and A. C. Schaeffer, "A class of nonharmonic Fourier series." *Trans. Amer. Math. Soc.*, vol. 72, pp. 341–366, 1952.

[58] M. Ehler, "On multivariate compactly supported bi-frames," *J. Fourier Anal. Appl.*, vol. 13, pp. 511–532, 2007.

[59] M. Ehler and B. Han, "Wavelet bi-frames with few generators from multivariate refinable functions," *Appl. Comput. Harmon. Anal.*, vol. 25, pp. 407–414, 2008.

[60] G. Evangelista, M. Dörfler, and E. Matusiak, "Phase Vocoders With Arbitrary Frequency Band Selection," *Proceedings of the 9th Sound and Music Computing Conference, July 11-14th 2012 Kopenhagen*, 2012.

[61] H. G. Feichtinger, "On a new Segal algebra," *Monatsh. Math.*, vol. 92, pp. 269–289, 1981.

[62] ——, "Banach convolution algebras of Wiener type," in *Proc. Conf. on Functions, Series, Operators, Budapest 1980*, Eds. B. Sz.-Nagy and J. Szabados. ed., ser. Colloq. Math. Soc. Janos Bolyai. Amsterdam: North-Holland, 1983, vol. 35, pp. 509–524.

[63] ——, “Modulation spaces on locally compact Abelian groups,” Tech. Rep., January 1983.

[64] H. G. Feichtinger and K. Gröchenig, “Gabor wavelets and the Heisenberg group: Gabor expansions and short time Fourier transform from the group theoretical point of view,” in *Wavelets :a tutorial in theory and applications*, ser. Wavelet Anal. Appl., C. K. Chui, Ed. Boston: Academic Press, 1992, vol. 2, pp. 359–397.

[65] H. G. Feichtinger, K. Gröchenig, and D. F. Walnut, “Wilson bases and modulation spaces,” *Math. Nachr.*, vol. 155, pp. 7–17, 1992.

[66] H. G. Feichtinger, M. Hazewinkel, N. Kaiblinger, E. Matusiak, and M. Neuhauser, “Metaplectic operators on C^n,” *Quart. J. Math. Oxford Ser.*, vol. 59, no. 1, pp. 15–28, 2008.

[67] H. G. Feichtinger and N. Kaiblinger, “2D-Gabor analysis based on 1D algorithms,” in *Proc. OEAGM-97 (Hallstatt, Austria)*, 1997.

[68] ——, “Varying the time-frequency lattice of Gabor frames,” *Trans. Amer. Math. Soc.*, vol. 356, no. 5, pp. 2001–2023, 2004.

[69] H. G. Feichtinger, N. Kaiblinger, and P. Prinz, “A POCS approach to Gabor analysis,” in *DIP-97 (Vienna, Austria)*, ser. SPIE, vol. 3346, October 1997, pp. 18–29.

[70] H. G. Feichtinger, W. Kozek, P. Prinz, and T. Strohmer, “On multidimensional non-separable Gabor expansions,” in *Proc. SPIE: Wavelet Applications in Signal and Image Processing IV*, August 1996.

[71] H. G. Feichtinger and T. Strohmer, *Gabor Analysis and Algorithms. Theory and Applications.* Boston: Birkhäuser, 1998.

[72] H. G. Feichtinger and W. Sun, “Sufficient conditions for irregular Gabor frames,” *Adv. Comput. Math.*, vol. 26, no. 4, pp. 403–430, April 2007.

[73] H. G. Feichtinger and G. Zimmermann, “A Banach space of test functions for Gabor analysis,” in *Gabor Analysis and Algorithms: Theory and Applications*, ser. Applied and Numerical Harmonic Analysis, H. G. Feichtinger and T. Strohmer, Eds. Boston, MA: Birkhäuser Boston, 1998, pp. 123–170.

[74] C. Feldbauer, G. Kubin, and W. B. Kleijn, “Anthropomorphic coding of speech and audio: A model inversion approach,” *EURASIP J. Adv. Sig. Proc.*, vol. 2005, no. 9, pp. 1334–1349, 2005.

[75] G. B. Folland, *Fourier Analysis and its Applications.* CA: Wadsworth and Brooks, 1992.

[76] J. J. F. Fournier and J. Stewart, "Amalgams of L^p and ℓ^q," *Bull. Amer. Math. Soc., New Ser.*, vol. 13, pp. 1–21, 1985.

[77] M. Frigo and S. Johnson, "The design and implementation of FFTW 3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[78] D. Gabor, "Theory of communication," *J. IEE*, vol. 93, no. 26, pp. 429–457, 1946.

[79] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hear. Res.*, vol. 47, pp. 103–138, 1990.

[80] M. Grady, "A group theoretic approach to a famous partition formula," *Amer. Math. Monthly*, vol. 112, pp. 645–651, 2005.

[81] L. Grafakos, *Classical Fourier Analysis (Second Edition).* Springer, 2008.

[82] K. Gröchenig, "Acceleration of the frame algorithm," *IEEE Trans. Sig. Proc.*, vol. 41, pp. 3331–3340, 1993.

[83] ——, "Irregular sampling of wavelet and short-time Fourier transforms," *Constr. Approx.*, vol. 9, pp. 283–297, 1993.

[84] ——, *Foundations of Time-Frequency Analysis*, ser. Appl. Numer. Harmon. Anal. Boston, MA: Birkhäuser Boston, 2001.

[85] K. Gröchenig and J. Stöckler, "Gabor frames and totally positive functions," *Duke Math. J.*, vol. 162, no. 6, pp. 1003–1031, 2013.

[86] G. A. Hagedorn, "Raising and lowering operators for semiclassical wave packets," *Ann. Phys. (8)*, vol. 269, no. 1, pp. 77–104, 1998.

[87] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51 – 83, January 1978.

[88] H. Helms, "Fast Fourier transform method of computing difference equations and simulating filters," *IEEE Trans. Audio Electroac.*, vol. 15, no. 2, pp. 85–90, 1967.

[89] E. Hernández, D. Labate, and G. Weiss, "A unified characterization of reproducing systems generated by a finite family. II." *J. Geom. Anal.*, vol. 12, no. 4, pp. 615–662, 2002.

[90] V. Hohmann, "Frequency analysis and synthesis using a gammatone filterbank," *Acta Acust. united Ac.*, vol. 88, no. 3, pp. 433–442, 2002.

[91] N. Holighaus, "Structure of nonstationary gabor frames and their dual systems," *Preprint, arXiv: 1306.5037*, 2013.

[92] N. Holighaus, M. Dörfler, G. Velasco, and T. Grill, "A framework for invertible, real-time constant-Q transforms," *IEEE Trans. Audio Speech Language Proc.*, vol. 21, no. 4, pp. 775–785, April 2013.

[93] F. Holland, "Harmonic analysis on amalgams of L^p and ℓ^q." *J. London Math. Soc.*, vol. 10, pp. 295–305, 1975.

[94] T. Irino and R. D. Patterson, "A dynamic compressive gammachirp auditory filterbank," *IEEE Trans. Audio Speech Language Proc.*, vol. 14, no. 6, pp. 2222–2232, November 2006.

[95] F. Jaillet, "Représentation et traitement temps-fréquence des signaux audionumériques pour des applications de design sonore," Ph.D. dissertation, Université de la Méditerranée - Aix-Marseille II, 2005.

[96] F. Jaillet, P. Balazs, M. Dörfler, and N. Engelputzeder, "Nonstationary Gabor Frames," in *Proceedings of SAMPTA'09, Marseille, May 18-22*, 2009.

[97] F. Jaillet and B. Torrésani, "Time-frequency jigsaw puzzle: adaptive multiwindow and multilayered Gabor expansions," *Int. J. Wavelets Multiresolut. Inf. Proc.*, vol. 2, pp. 293–316, 2007.

[98] A. J. E. M. Janssen, "Duality and biorthogonality for Weyl-Heisenberg frames," *J. Fourier Anal. Appl.*, vol. 1, no. 4, pp. 403–436, 1995.

[99] ——, "The duality condition for Weyl-Heisenberg frames." in *Gabor Analysis and Algorithms: Theory and Applications*, H. G. Feichtinger and T. Strohmer, Eds., 1998, pp. 33–84, 453–488.

[100] A. J. E. M. Janssen and P. L. Søndergaard, "Iterative algorithms to approximate canonical Gabor windows: Computational aspects," *J. Fourier Anal. Appl.*, vol. 13, no. 2, pp. 211–241, 2007.

[101] S. Johnson and M. Frigo, "A Modified Split-Radix FFT With Fewer Arithmetic Operations," vol. 55, no. 1, p. 111.

[102] N. Kaiblinger, "Metaplectic representation, eigenfunctions of phase space shifts, and Gelfand-Shilov spaces for LCA groups," Ph.D. dissertation, Dept. Mathematics, Univ. Vienna, 1999.

[103] N. Kaiblinger and M. Neuhauser, "Metaplectic operators for finite abelian groups and R^d," *Indag. Math.*, vol. 20, no. 2, pp. 233–246, 2009.

[104] Y. Katznelson, *An Introduction to Harmonic Analysis. 3rd Corr. ed.* Cambridge University Press, 2004.

[105] G. Kutyniok and T. Strohmer, "Wilson bases for general time-frequency lattices," *SIAM J. Math. Anal.*, vol. 37, no. 3, pp. 685–711, 2005.

[106] J. Laroche and M. Dolson, "Phase-vocoder: about this phasiness business," in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics 1997*, October 1997.

[107] ——, "Improved phase vocoder time-scale modification of audio," *IEEE Trans. Speech Audio Proc.*, vol. 7, no. 3, pp. 323–332, 1999.

[108] R. S. Laugesen, "Gabor dual spline windows," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 2, pp. 180 – 194, September 2009.

[109] J. Li, T. Nguyen, and S. Tantaratana, "A simple design method for near-perfect-reconstruction nonuniform filter banks," *IEEE Trans. Sig. Proc.*, vol. 45, no. 8, 1997.

[110] J. Lim, "Neumann series expansion of the inverse of a frame operator," *Commun. Korean Math. Soc.*, vol. 13, no. 4, pp. 791–800, 1998.

[111] M. Liuni, A. Röbel, E. Matusiak, M. Romito, and X. Rodet, "Automatic adaptation of the time-frequency resolution for sound analysis and re-synthesis," *IEEE Trans. Audio, Speech, Language Proc.*, vol. 21, no. 5, pp. 959 –970, May 2013.

[112] M. Liuni, A. Röbel, M. Romito, and X. Rodet, "Rényi information measures for spectral change detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011*, May 2011, pp. 3824 – 3827.

[113] E. A. Lopez-Poveda and R. Meddis, "A human nonlinear filterbank," *J. Acoust. Soc. Am.*, vol. 110, no. 6, pp. 3107–3118, December 2001.

[114] F. Low, "Complete sets of wave packets," in *A Passion for Physics - Essays in Honor of Geoffrey Chew*, C. DeTar, J. Finkelstein, and C. Tan, Eds. Singapore: World Scientific, 1985, pp. 17–22.

[115] S. Mallat, *A Wavelet Tour of Signal Processing.*, 2nd ed. San Diego, CA: Academic Press, 1999.

[116] ——, *A Wavelet Tour of Signal Processing: The Sparse Way.* Academic Press, 2009.

[117] H. Malvar, *Signal Processing with Lapped Transforms.* Boston, MA: Artech House. xvi, 1992.

[118] R. Marks, *Handbook of Fourier Analysis and its Applications.* Oxford University Press, 2009.

[119] R. Meddis, W. Lecluyse, N. R. Clark, T. Jürgens, C. M. Tan, M. R. Panda, and G. J. Brown, "A computer model of the auditory periphery and its application to the study of hearing," in *Proceedings of the 16th International Symposium on Hearing (ISH 2012)*, Cambridge, UK, July, 23–27 2012.

[120] B. C. J. Moore, *An introduction to the psychology of hearing*, 6th ed. Emerald Group Publishing, 2012.

[121] K. Nayebi, T. Barnwell, III, and M. Smith, "Nonuniform filter banks: a reconstruction and design theory," *IEEE Trans. Sig. Proc.*, vol. 41, no. 3, 1993.

[122] T. Necciari, P. Balazs, N. Holighaus, and P. Søndergaard, "The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction," May 2013, proceedings of the 38th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013).

[123] A. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Trans. Acoust. Speech Signal Proc.*, vol. 29, no. 1, pp. 84–91, 1981.

[124] J. J. O'Donovan and D. J. Furlong, "Perceptually motivated time-frequency analysis," *J. Acoust. Soc. Am.*, vol. 117, no. 1, pp. 250–262, January 2005.

[125] S. Paukner, "Foundations of Gabor Analysis for Image Processing," Master's thesis, 2007.

[126] J. Petrillo, "Counting subgroups in a direct product of finite cyclic groups," *College Math J.*, vol. 42, pp. 215–222, 2011.

[127] P. Philippe, F. M. de Saint-Martin, and M. Lever, "Wavelet packet filterbanks for low time delay audio coding," *IEEE Trans Speech Audio Proc.*, vol. 7, no. 3, pp. 310–322, May 1999.

[128] R. Pichevar, H. Najaf-Zadeh, L. Thibault, and H. Lahdili, "Auditory-inspired sparse representation of audio signals," *Speech Commun.*, vol. 53, no. 5, pp. 643–657, 2011.

[129] M. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform," vol. 24, no. 3, pp. 243–248, 1976.

[130] P. Prinz, "Calculating the dual Gabor window for general sampling sets," *IEEE Trans. Sig. Proc.*, vol. 44, no. 8, pp. 2078–2082, 1996.

[131] K. Ramchandran, Z. Xiong, C. Herley, and M. Orchard, "Flexible Tree-structured Signal Expansions Using Time-varying Wavelet Packets," *IEEE Trans. Sig. Proc.*, vol. 45, pp. 233–245, 1997.

[132] J. Roe, *Lectures on Coarse Geometry*, ser. University Lecture Series. Providence, RI: American Mathematical Society, 2003, vol. 31.

[133] A. Ron and Z. Shen, "Frames and stable bases for shift-invariant subspaces of $L^2(\mathbb{R}^d)$." *Canad. J. Math.*, vol. 47, no. 5, pp. 1051–1094, 1995.

[134] ——, "Affine systems in $l_2(\mathbb{R}^d)$. II: Dual systems." *J. Fourier Anal. Appl.*, vol. 3, no. 5, pp. 618–637, 1997.

[135] ——, "Affine systems in $L^2(\mathbb{R}^d)$: The analysis of the analysis operator." *J. Funct. Anal.*, vol. 148, no. 2, pp. 408–447, 1997.

[136] ——, "Weyl-Heisenberg frames and Riesz bases in $L_2(\mathbb{R}^d)$." *Duke Math. J.*, vol. 89, no. 2, pp. 237–282, 1997.

[137] ——, "Generalized shift-invariant systems." *Constr. Approx.*, vol. 22, pp. 1–45, 2005.

[138] R. Schafer and L. Rabiner, "Design and Simulation of a Speech Analysis-Synthesis System based on Short-Time Fourier Analysis," *IEEE Trans. Audio Electroac.*, vol. 21, no. 3, pp. 165–174, 1973.

[139] R. Schmidt, *Subgroup Lattices of Groups*, ser. de Gruyter Expositions in Mathematics. Berlin: de Gruyter, 1994, vol. 14.

[140] C. Schörkhuber, A. Klapuri, and A. Sontacchi, "Pitch shifting of audio signals using the constant-Q transform," in *15th International Conference on Digital Audio Effects (DAFx-12)*, September 2012.

[141] C. Schörkhuber and A. Klapuri, "Constant-Q toolbox for music processing," in *Proceedings of the 7th Sound and Music Computing Conference (SMC), 2010*, 2010.

[142] I. Selesnick and I. Bayram, "Frequency-domain design of overcomplete rational-dilation wavelet transforms," *IEEE Trans. Sig. Proc.*, vol. 57, no. 8, pp. 2957–2972, 2009.

[143] H. Shao, W. Jin, and S. Qian, "Order tracking by discrete Gabor expansion," *IEEE Trans. Inst. Meas.*, vol. 52, pp. 754–761, 2003.

[144] J. Smith, "Audio FFT filter banks," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, September 2009.

[145] P. Søndergaard, B. Torrésani, and P. Balazs, "The Linear Time Frequency Analysis Toolbox," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 10, no. 4, p. 1250032, 2011.

[146] P. L. Søndergaard, "Finite Discrete Gabor Analysis," Ph.D. dissertation, Technical University of Denmark, 2007.

[147] P. L. Søndergaard, "Efficient Algorithms for the Discrete Gabor Transform with a long FIR window," *J. Fourier Anal. Appl.*, vol. 18, no. 3, pp. 456–470, 2012.

[148] T. Stockham Jr, "High-speed convolution and correlation," in *Proceedings of the Spring joint computer conference,April 26-28, 1966.* ACM, 1966, pp. 229–233.

[149] S. Strahl and A. Mertins, "Analysis and design of gammatone signal models," *J. Acoust. Soc. Am.*, vol. 126, no. 5, pp. 2379–2389, November 2009.

[150] T. Strohmer, "Numerical algorithms for discrete Gabor expansions," in *Gabor Analysis and Algorithms: Theory and Applications*, H. G. Feichtinger and T. Strohmer, Eds. Boston: Birkhäuser Boston, 1998, pp. 267–294.

[151] W. Sun and X. Zhou, "Irregular Gabor frames and their stability," *Proc. Amer. Math. Soc.*, vol. 131, no. 9, pp. 2883–2893 (electronic), 2003.

[152] ——, "Irregular wavelet/Gabor frames." *Appl. Comput. Harmon. Anal.*, vol. 13, no. 1, pp. 63–76, 2002.

[153] M. Suzuki, "On the lattice of subgroups of finite groups," *Trans. Amer. Math. Soc.*, vol. 70, pp. 345–371, 1951.

[154] L. Toth, M. Hampejs, N. Holighaus, and C. Wiesmeyr, "On the subgroups of the group $Z_m \times Z_n$," *Preprint, arXiv: 1211.1797.*

[155] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra.* SIAM, 1997.

[156] M. Tărnăuceanu, "An arithmetic method of counting the subgroups of a finite abelian group," *Bull. Math. Soc. Sci. Math. Roumanie (N.S.)*, vol. 53(101).

[157] ——, “A new method of proving some classical theorems of abelian groups,” *Southeast Asian Bull. Math.*, vol. 31, pp. 1191–1203, 2007.

[158] A. J. van Leest, “Non-separable Gabor schemes. Their Design and Implementation,” Ph.D. dissertation, Tech. Univ. Eindhoven, 2001.

[159] A. J. van Leest and M. J. Bastiaans, “Gabor’s discrete signal expansion and the discrete Gabor transform on a non-separable lattice,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’00)*, vol. 1, 2000, pp. 101–104.

[160] ——, “Gabor’s signal expansion and the Gabor transform on a non-separable time-frequency lattice,” *J. Franklin Inst.*, vol. 337, no. 4, pp. 291–301, 2000.

[161] ——, “Implementations of non-separable Gabor schemes,” in *Proc. Eusipco 2004 ,12th European Signal Processing Conference, Vienna, Austria*, 2004, pp. 1565–1568.

[162] N. H. van Schijndel, T. Houtgast, and J. M. Festen, “Intensity discrimination of Gaussian-windowed tones: Indications for the shape of the auditory frequency-time window,” *J. Acoust. Soc. Am.*, vol. 105, no. 6, pp. 3425–3435, June 1999.

[163] P. Vandewalle, J. Kovačević, and M. Vetterli, “Reproducible research in signal processing,” *IEEE Signal Process. Mag.*, vol. 26, pp. 37–47, 2009.

[164] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, “Constructing an invertible constant-Q transform with nonstationary Gabor frames,” *Proceedings of DAFX11, Paris*, 2011.

[165] D. F. Walnut, “Continuity properties of the Gabor frame operator.” *J. Math. Anal. Appl.*, vol. 165, no. 2, pp. 479–504, 1992.

[166] A. Weil, “Sur certains groupes d’opérateurs unitaires,” *Acta Math.*, vol. 111, pp. 143–211, 1964.

[167] E. Wesfreid and M. V. Wickerhauser, “Adapted local trigonometric transforms and speech processing,” *IEEE Trans. Sig. Proc.*, vol. 41, no. 12, pp. 3596–3600, Dec 1993.

[168] R. C. Whaley and A. Petitet, “Minimizing development and maintenance costs in supporting persistently optimized BLAS,” *Software: Practice and Experience*, vol. 35, no. 2, pp. 101–121, February 2005.

[169] C. Wiesmeyr, N. Holighaus, and P. Søndergaard, “Efficient algorithms for the discrete gabor transform on a nonseparable lattice,” *to appear in IEEE Trans. Sig. Proc.*, 2013.

[170] Wikipedia, “Window function - wikipedia article,” http://en.wikipedia.org/wiki/Window_function.

[171] P. J. Wolfe, D. Rudoy, and B. Prabahan, “Superposition frames for adaptive time-frequency analysis and fast reconstruction,” *IEEE Trans. Sig. Proc.*, vol. 58, pp. 2581–2596, 2010.

[172] J. Youngberg and S. Boll, “Constant-Q signal analysis and synthesis,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’78)*, vol. 3, 1978, pp. 375–378.

[173] Y. Y. Zeevi and M. Zibulski, “Oversampling in the Gabor scheme,” *IEEE Trans. Sig. Proc.*, vol. 41, no. 8, pp. 2679–2687, 1993.

[174] Y. Y. Zeevi, M. Zibulski, and M. Porat, “Multi-window Gabor schemes in signal and image representations,” in *Gabor Analysis and Algorithms: Theory and Applications*, ser. Appl. Numer. Harmon. Anal., H. G. Feichtinger and T. Strohmer, Eds. Boston, MA: Birkhäuser Boston, 1998, pp. 381–407.

[175] M. Zibulski and Y. Y. Zeevi, “Signal- and image-component separation by a multi-window Gabor-type scheme,” in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on,*, vol. 2. Vienna , Austria: IEEE, aug 1996, pp. 835 –839.

[176] ——, “Analysis of multiwindow Gabor-type schemes by frame methods,” *Appl. Comput. Harmon. Anal.*, vol. 4, no. 2, pp. 188–221, 1997.

[177] ——, “Discrete multiwindow Gabor-type transforms.” *IEEE Trans. Sig. Proc.*, vol. 45, no. 6, pp. 1428–1442, 1997.

[178] ——, “The generalized Gabor scheme and its application in signal and image representation,” in *Signal and Image Representation in Combined Spaces*, ser. Wavelet Anal. Appl. San Diego, CA: Academic Press, 1998, vol. 7, pp. 121–164.

[179] Y. Zou, “Gaussian binomials and the number of sublattices,” *Acta Cryst.*, vol. 62, pp. 409–410, 2006.

Curriculum Vitæ

Dipl.-Math. Nicki Holighaus

Education

1994–2003 Wilhelm-von-Oranien Schule Dillenburg, Germany (Gymnasium)

2004–2010 Diploma studies at Justus-Liebig Universtität Gießen, Germany

2010–now Doctoral studies at Universität Wien

Work experience

2008–2010 Tutor at Justus-Liebig Universtität Gießen, Germany

2011–2012 Tutor at Universität Wien

2010–2012 Research assistant at Universität Wien

2012–now Research assistant at the Acoustics Research Institute, ÖAW, Vienna

Other

2003–2004 Civil service at Wohnheim der Lebenshilfe Dillenburg, Simmersbach, Germany

2004–2009 Side job at Wohnheim der Lebenshilfe Dillenburg, Simmersbach, Germany

Scientific work

Publications:

T. Necciari, P. Balazs, N. Holighaus, and P. Søndergaard, “The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction,” 2013, Proceedings the 38th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)

N. Holighaus, M. Dörfler, G. Velasco, and T. Grill, "A framework for invertible, real-time constant-Q transforms," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, pp. 775-785, April 2013

N. Holighaus, C. Wiesmeyr and P.L. Søndergaard, "Efficient Algorithms for the Discrete Gabor Transform on a nonseparable lattice," to appear in IEEE Transactions on Signal Processing, 2013

G. Velasco, N. Holighaus, M. Dörfler and T. Grill, "Constructing an invertible constant-Q transform with nonstationary Gabor frames", Proceedings of the 14th International Conference on Digital Audio Effects (DAFx 11), Paris, France, 2011.

P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. A. Velasco, "Theory, implementation and applications of nonstationary Gabor Frames", J. Comput. Appl. Math., 236(6):1481-1496, 2011.

A. Holzapfel, G. Velasco, N. Holighaus, M. Dörfler, and A. Flexer, "Advantages of nonstationary Gabor transforms in beat tracking", Proceedings of MIRUM11., November 2011.

C. Wiesmeyr, N. Holighaus and G. Velasco, "A continuous time-frequency representation via warping", Poster, Asian Mathematical Conference (AMC) 2013, Busan, June 2013.

N. Holighaus, "Zeit-Frequenz-Analyse mit Methoden der Gabor Analysis". Diploma thesis. Justus-Liebig Universität Giessen, 2010

Preprints:

L. Toth, M. Hampejs, N. Holighaus, and C. Wiesmeyr, "On the subgroups of the group $\mathbb{Z}_m \times \mathbb{Z}_n$," Preprint, submitted, arXiv: 1211.1797

N. Holighaus, "Structure of nonstationary Gabor frames and their dual systems," Preprint, submitted, arXiv: 1306:5037