



universität  
wien

# DIPLOMARBEIT

Titel der Diplomarbeit

“Foundations of Gabor Analysis  
for Image Processing”

Verfasser

Stephan Paukner

angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, im November 2007

Studienkennzahl lt. Studienblatt: A 405

Matrikelnummer: 9702122

Studienrichtung lt. Studienblatt: Mathematik (Stzw)

Betreuer: Univ.-Prof. Dr. Hans G. Feichtinger



# Acknowledgements

I wish to thank my supervisor Prof. Dr. Hans G. Feichtinger for guiding me through the domain of my topic with patience, lots of useful suggestions, references and support. Thanks go to his team, the Numerical Harmonic Analysis Group (NuHAG), as well, especially to Dr. Norbert Kaiblinger, Anna Grybos and Dr. Monika Dörfler for providing me with hints to literature and MATLAB code. I also wish to thank Dr. Peter L. Søndergaard for some fruitful discussions.

I am infinitely grateful to my family, the Püspöks and the Paukners, for their unconditional love and support, continuous motivation and neverending patience through the years of my studies and occupation. I wouldn't have come that far without you always believing in me. Special thanks go to my wonderful girlfriend Doris for being a never-ebbing font of deep love, understanding, patience, and everything that's not related to mathematics.

Thanks to my fellows from LDAG for sharing a great time with me in the auditorium and for providing me with help and support for how to successfully write a Master's thesis.

I am also grateful for being supported by the European Social Fund by a six-month final degree scholarship.



# Contents

<b>Abstract</b>	<b>xi</b>
<b>1 Basics of Time-Frequency Analysis</b>	<b>1</b>
1.1 Notation and Spaces	1
1.2 The Fourier Transform	3
1.3 Fundamental Operations	4
1.4 The Short-Time Fourier Transform	6
1.5 The Gaussian Function	9
1.6 Tools from Linear Algebra	11
1.7 Tensor Products	12
<b>2 Frames in Hilbert Spaces</b>	<b>15</b>
2.1 From Bases to Frames	15
2.2 Gabor Frames in $L^2(\mathbb{R}^d)$	21
2.3 Gabor Frames in $\ell^2(\mathbb{Z})$	25
<b>3 Finite Discrete Gabor Analysis</b>	<b>27</b>
3.1 Finite Discrete Periodic Signals	27
3.2 Frames and Gabor Frames in $\mathbb{C}^L$	28
3.3 The Structure of the Gabor Frame Matrix	30
3.3.1 The Walnut Representation	31
3.3.2 The Janssen Representation	32
3.3.3 Factorizations of the Gabor Matrices	34
3.4 The Dual Window on Non-Separable Sampling Sets	36
<b>4 Fourier Analysis of Discrete Images</b>	<b>41</b>
4.1 Digital Representation of Images	41
4.1.1 The Nature of Images	41
4.1.2 Digital Images and Color	42
4.1.3 RGB Images	43
4.2 Understanding 2D Frequencies	45
4.3 Frequency Behavior of Natural Images	46
4.3.1 Experiment: Coefficient Filtering by Masking	49
4.4 STFT of Discrete Images	53
4.5 2D Window Functions	58
<b>5 Image Representation by Gabor Expansion</b>	<b>63</b>
5.1 2D Gabor Expansions	64

5.2	Separable Atoms on Fully Separable Lattices . . . . .	66
5.2.1	Efficient Gabor Expansion by Sampled STFT . . . . .	70
5.2.2	Visualizing a Sampled STFT of an Image . . . . .	73
5.2.3	Experiment: Gabor Coefficient Thresholding . . . . .	77
5.3	Separable Atoms on Partially Non-Separable Lattices . . . . .	78
5.4	Non-Separable Atoms on Fully Separable Lattices . . . . .	83
5.4.1	2D Gabor Expansions by 1D Algorithms . . . . .	84
5.4.2	Matching the Atom to the Signal . . . . .	87
5.5	General 2D Atoms on General 4D Lattices . . . . .	87
5.5.1	Data Reduction by Downsampling . . . . .	90
5.6	Discussion and Outlook . . . . .	93

## **Appendix 97**

### **A MATLAB Code 97**

A.1	nsgauss.m – Non-Separable 2D Gaussian . . . . .	97
A.2	acf2.m – 2D Autocorrelation Function . . . . .	98
A.3	stft2sep.m – STFT: Separable 2D Atom, Separable Lattice . . . . .	98
A.4	istft2sep.m – Inverse STFT in Separable Setting . . . . .	99
A.5	gabfc.m – Gabor Matrix, Modulation Priority . . . . .	100
A.6	gabtf.m – Gabor Matrix, Translation Priority . . . . .	100
A.7	stft2q.m – STFT: General 2D Atom, Quincunx-Like Lattice . . . . .	101
A.8	istft2q.m – Inverse STFT in General Setting . . . . .	102

### **Bibliography 105**

# List of Figures

1.1	Two signals and their (short-time) Fourier transforms . . . . .	7
3.1	A lattice and its adjoint lattice . . . . .	33
3.2	Sampling subgroups with redundancy $\frac{4}{3}$ and their building blocks . . .	36
3.3	Dual windows on general subgroups with zero imaginary part . . . . .	37
3.4	Dual windows on general subgroups with non-zero imaginary part . .	38
4.1	Red, green and blue color channel of an RGB image . . . . .	44
4.2	Real parts of $e^{2\pi i(x,\omega)}$ for $\omega = (0.5, 1)^T$ and $\omega = (1, -2)^T$ . . . . .	46
4.3	The well-known Lena image and its 2D-FFT spectrum . . . . .	47
4.4	A zebra image and its 2D-FFT spectrum . . . . .	48
4.5	Image signals are periodic over their borders . . . . .	49
4.6	Reconstruction of Lena from masked Fourier coefficients . . . . .	50
4.7	Reconstruction of zebra from masked Fourier coefficients . . . . .	51
4.8	A 2D Gaussian function as tensor product of two 1D Gaussians . . . .	53
4.9	Localized Fourier transforms of zebra . . . . .	56
4.10	Convolutions of zebra with modulated 2D Gaussians . . . . .	57
4.11	Common windows for 1D signal analysis . . . . .	58
4.12	Separable 2D windows might be concentrated along the axes . . . . .	59
4.13	A non-separable 2D window and its Fourier transform . . . . .	60
4.14	Frequency-shifts of a non-separable window . . . . .	61
5.1	2D separable window and its dual on a fully separable lattice . . . . .	71
5.2	Discrete 2D Gabor transform of zebra, modulation priority . . . . .	75
5.3	Discrete 2D Gabor transform of zebra, translation priority . . . . .	76
5.4	Thresholding by separable atom on fully separable lattice . . . . .	77
5.5	Separable 2D dual windows on partially non-separable lattices . . . . .	79
5.6	Thresholding by separable atom on a product-quincunx lattice . . . . .	80
5.7	Separable 2D dual window with non-zero imaginary part . . . . .	82
5.8	Partial reconstruction by dual with non-zero imaginary part . . . . .	83
5.9	A non-separable window and some duals on fully separable lattices . .	84
5.10	Obtaining a dual 2D atom by the dual of its 1D extracted atom . . . . .	85
5.11	Thresholding by non-separable window on a fully separable lattice . .	86
5.12	2D non-separable dual atoms on 4D quincunx-like lattices . . . . .	88
5.13	Thresholding by non-separable atom on 4D quincunx-like lattice . . . .	89
5.14	Comparison of image downsampling methods . . . . .	90
5.15	Dual 2D atoms by downsampling . . . . .	91





# List of Tables

5.1	Results of Experiment 5.2.1 . . . . .	73
5.2	Results of Experiment 5.2.3 . . . . .	78
5.3	Results of thresholding on product-quincunx lattice . . . . .	81
5.4	Results of thresholding by non-separable atom on separable lattice . .	87
5.5	Results of thresholding by general atom on 4D quincunx-like lattice .	89
5.6	Accuracy of downsampled dual Gaussian windows . . . . .	92



# Abstract

Signal processing plays an important role in modern society: Its applications span a comprehensive domain from automotive to entertainment industry, from medical diagnostic technology to communication services. Signals are electrical information that represents acoustical, optical or other data, like brain waves (EEG) or geophysical data, whose values exhibit a certain variability in location and time. These electrical data are to be edited, filtered, amplified, denoised, interpolated, transmitted or processed in some other way.

Fourier analysis quickly established as a standard method in signal analysis. Mathematically, signals are represented as discrete periodic functions that are composed of pure oscillations with different frequencies and amplitudes, and the Fourier transform unveils in what amount what frequencies are contained in a signal. But time information gets lost thereby, what led to the idea of a *windowed* or *short-time Fourier transform* (STFT), where only short time-intervals in the signal undergo a Fourier transform, cut out by a smooth window function that is subsequently shifted over the signal. However, the time-frequency information that the STFT provides is highly redundant, and a reduction is sought while preserving the complete time-frequency behavior of a signal.

Digital images can be handled just like digital sound signals, they're representable as a sum of pure two-dimensional (2D) oscillations. Mathematically, there is no distinction at all, as the theory is developed in general function spaces. But it can be confusing in practise to talk about the time-frequency analysis of images, as the image signal does not evolve in one-dimensional time, but on a two-dimensional plane. Furthermore, the time-frequency plane becomes a four-dimensional position-frequency space, what makes it difficult to produce descriptive graphs of windowed 2D Fourier transforms of images. Nevertheless, mathematically the terms time and frequency are handled in arbitrary dimensions.

The problems that emerge in signal analysis nourish a standard task in mathematical analysis, namely that of describing arbitrary functions by the help of a set of simple functions that possess well-known and easy-to-handle analytical properties. Fourier analysis fulfills this by expanding a signal into a sum of elementary oscillations. The STFT equivalence is to use a sufficient set of time-frequency shifts of a single window function instead. The concern about when the emerging analysis and synthesis operations are reasonable is handled by the theory of frames, a generalization of the concept of bases. Gabor analysis unites these mathematical approaches: It yields conditions for a set of time-frequency shifted window functions to be a frame for the signal space and provides an understanding of signals by expanding them into a sum of those elementary shifted and modulated atoms.

This thesis wants to examine how the concepts of Gabor analysis apply to the case of digital images and what new problems emerge compared to one-dimensional signals. The following paragraphs summarize how this work is organized.

Chapter 1 is a summary of the foundations of time-frequency analysis. The Fourier transform and STFT are introduced in general function spaces that possess an inner product and are thus equipped with some kind of geometry. It makes sense to only consider functions of finite energy, making the space of square-integrable functions,  $L^2(\mathbb{R}^d)$ , the candidate of choice.

Chapter 2 introduces the concept of frames in general (separable) Hilbert spaces. The special case of Gabor frames is again treated in  $L^2(\mathbb{R}^d)$ . It is explained that Gabor expansions consider two window functions: The analyzing prototype and its reconstructing dual that is dependent on the chosen subgroup of the time-frequency plane. Whereas signal processing previously was of the time-continuous analog type, it has changed to a finite time-discrete model today due to the high availability of digital computers. The chapter ends with the necessary step to a time-discrete signal model.

Chapter 3 is dedicated to finite discrete Gabor frames for a finite discrete periodic signal model, eventually enabling computational implementations. A finite sequence of real or complex values can be represented as a vector, where the dimensionality of the emerging signal space should not be confused with the one-dimensionality of the vector shape. Things become related to terms of linear algebra, and the matrix representation of frames is introduced. Finally, dual Gabor windows on general sampling subgroups are shown that possess a significant non-zero imaginary part.

Chapter 4 takes the step from 1D signals to 2D images. It explains how images are represented in digital computers. The chapter provides an understanding of 2D elementary oscillations and the idea of 2D frequencies. The 2D Fourier transform unveils to what extent low frequencies contribute to the homogeneous areas in natural images and how higher frequencies are responsible for contours, edges and sharpness. Then the author provides a way of how to visualize the 4D position-frequency behavior of an image by presenting a collection of 2D images. The chapter ends with a treatment of 2D windows for the 2D STFT of images.

Chapter 5 finally comes to Gabor expansions of images. It turns out that the possible non-separability of 2D windows intervenes with the various depths of the non-separability of 4D sampling subgroups. The following sections reflect the order of increasing difficulty for computational implementations, going from twofold separability (of both 2D atom and 4D lattice) to true non-separability (of both atom and lattice). All cases are accompanied by numerical experiments with Gabor coefficient thresholding. Separable atoms allow for the consideration of tensor products of two 1D frames. The case of fully separable lattices allows for an efficient Gabor expansion by using the sampled 1D STFT. The involvement of non-separable subgroups leads to 2D duals with non-zero imaginary parts as well. Under certain conditions general 2D atoms on 4D grids can be mapped to a 1D case. Finally, an approach for obtaining quicker 2D Gabor expansions is provided by signal downsampling.

“Serious” image processing is not treated in this thesis, as implementations that

involve Gabor systems had to be compared with approved existing methods. This work thus ends by providing references to literature that consider image processing methods by Gabor expansion, and lists some questions for further research.

The published numerical computations were performed on a PC using a single 1.6 GHz CPU and 1 GB RAM. The experiments were implemented in MATLAB 7 and Octave 2.9 on the Debian GNU/Linux operating system ('lenny', Kernel 2.6) and involved functions from the NuHAG MATLAB toolboxes. All figures were produced with MATLAB or The GIMP. The typesetting was done in L<sup>A</sup>T<sub>E</sub>X.

This thesis is available in the web at

<http://paukner.cc/math/ga4ip/>

and probably also via <http://nuhag.eu/>.



# Chapter 1

## Basics of Time-Frequency Analysis

In this chapter we summarize the most important terms and results of time-frequency analysis. Mathematically, this topic is developed on general function spaces, whereas for practical purposes this has to be adapted to a finite-dimensional time-discrete model. Whenever it appears reasonable, functions are referred to as signals even in the infinitesimal context.

### 1.1 Notation and Spaces

Cartesian products of the standard number sets are denoted by  $\mathbb{R}^d$ ,  $\mathbb{C}^d$ , etc., with  $d$  reflecting the dimension of a signal. The symbol  $I$  is used for a countable index set, and its cardinality is given by  $|I|$ . For  $n \in \mathbb{N}$  we abbreviate  $\langle n \rangle := \{0, \dots, n-1\}$ . We visually distinguish the complex unit  $i \in \mathbb{C}$  from an arbitrary index  $i \in I$ . The complex conjugate of  $z \in \mathbb{C}$  is written as  $\bar{z}$ . The conjugate transpose of a matrix  $A$  is  $A^* := \bar{A}^T$ . The space of all  $m \times n$  matrices is  $\mathbb{C}^{m \times n}$ , where the matrix lines are numbered  $0, \dots, m-1$  and the columns  $0, \dots, n-1$ .

During the text, symbols like  $X, Y$  are used for Banach spaces and  $\mathcal{H}, \mathcal{K}$  for Hilbert spaces. A sequence of elements  $f_i \in X$ ,  $i \in I$ , is denoted by  $\{f_i\}_{i \in I}$  and is to be distinguished from an unordered collection of the elements.

The finite linear span of a subset  $A \subseteq X$  is denoted by  $\text{span } A$  and its closure in the norm-topology of  $X$  is written as  $\overline{\text{span}} A$ .

$\mathcal{L}(X, Y)$  is the space of bounded linear operators  $T: X \rightarrow Y$ . The operator norm is given as

$$\|T\|_{\mathcal{L}(X, Y)} := \sup_{u \neq 0} \frac{\|Tu\|_Y}{\|u\|_X} = \sup_{\|u\|_X=1} \|Tu\|_Y = \sup_{\|u\|_X \leq 1} \|Tu\|_Y.$$

For notational convenience, the subscripts of the respective norms might be dropped.

Regarding the notation of operations on functions, we shortly write  $U Tf(x)$  for  $(U(Tf))(x)$ .

The sequence spaces  $\ell^p(I)$ ,  $1 \leq p \leq \infty$ , are Banach spaces of complex sequences  $c = \{c_i\}_{i \in I}$  for which in the case  $p < \infty$  the norm

$$\|c\|_p := \left( \sum_{i \in I} |c_i|^p \right)^{1/p}$$

is finite; for  $p = \infty$  the supreme-norm

$$\|c\|_\infty := \sup_{i \in I} |c_i|$$

is finite. For  $p = 2$  and  $I = \mathbb{Z}^d$  one gets the Hilbert space  $\ell^2(\mathbb{Z}^d)$  with the inner product

$$\langle u, v \rangle_2 := \sum_{k \in \mathbb{Z}^d} u_k \overline{v_k}.$$

The inner product in  $\mathbb{C}^d$  is written as  $x \cdot y := \sum_{i=1}^d x_i \overline{y_i}$  and we abbreviate  $x^2 := x \cdot x$ . The Euclidean norm is  $|x| := \sqrt{x \cdot x}$ .

The Lebesgue spaces  $L^p(\mathbb{R}^d)$  (or shortly  $L^p$ ),  $1 \leq p \leq \infty$ , are Banach spaces of measurable functions  $f: \mathbb{R}^d \rightarrow \mathbb{C}$  for which in the case  $p < \infty$  the norm

$$\|f\|_{L^p} := \left( \int_{\mathbb{R}^d} |f(x)|^p dx \right)^{1/p}$$

is finite; for  $p = \infty$  the essential supreme

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

is finite.

Elements of  $L^p$  are actually classes of functions, identifying functions which only differ on zero-sets with respect to the Lebesgue measure. Continuous functions  $f \in L^\infty(\mathbb{R}^d)$  yield  $\|f\|_{L^\infty} = \|f\|_\infty$ .

Again, the case  $p = 2$  provides a Hilbert space,  $L^2(\mathbb{R}^d)$ , equipped with the inner product

$$\langle f, g \rangle_{L^2} := \int_{\mathbb{R}^d} f(x) \overline{g(x)} dx.$$

For notational convenience, we might again drop the subscripts whenever it is clear what inner product is currently in use.

$C_0(\mathbb{R}^d)$  is the space of continuous functions on  $\mathbb{R}^d$  vanishing at infinity. The space of test functions  $C_c^\infty(\mathbb{R}^d)$  consists of all infinitely differentiable functions with compact support. It lies dense in  $L^p(\mathbb{R}^d)$  for  $p < \infty$ . The Schwartz class  $\mathcal{S}(\mathbb{R}^d)$  consists of all infinitely differentiable functions which are rapidly decreasing at infinity, i.e.,

$$\lim_{|x| \rightarrow \infty} x^\alpha D^\beta f(x) = 0 \quad \forall \alpha, \beta \in \mathbb{N}_0^d \quad \forall f \in \mathcal{S}(\mathbb{R}^d),$$

where we use the multi-index notation  $x^\alpha := x_1^{\alpha_1} \cdots x_d^{\alpha_d}$  and

$$D^\beta f(x) := \frac{\partial^{\beta_1} \cdots \partial^{\beta_d}}{\partial x_1^{\beta_1} \cdots \partial x_d^{\beta_d}} f(x_1, \dots, x_d).$$

The absolute value of  $\alpha \in \mathbb{N}_0^d$  is  $|\alpha| := \sum_{i=1}^d \alpha_i$ .

As  $\mathcal{S}(\mathbb{R}^d) \subset L^p(\mathbb{R}^d)$  for  $1 \leq p < \infty$  and  $\mathcal{S}(\mathbb{R}^d)$  contains  $C_c^\infty(\mathbb{R}^d)$ , it is also dense in  $L^p(\mathbb{R}^d)$ .



## 1.2 The Fourier Transform

**1.2.1 Definition** For  $f \in L^1(\mathbb{R}^d)$ , the Fourier transform is defined as

$$\hat{f}(\omega) := (\mathcal{F}f)(\omega) := \int_{\mathbb{R}^d} f(x) e^{-2\pi i x \cdot \omega} dx . \quad (1.1)$$

The Fourier transform is the most important tool in signal analysis. It can be interpreted as an inner product of a function with an exponential function  $t \mapsto e^{i\omega t}$  and therefore as a projection of a given function onto a circle function with certain frequency. The value of the integral shows how much they have in common, i.e., in what amount the given signal  $f$  contains the pure frequency  $\omega$ . The function  $\hat{f}(\omega)$  now describes the frequency behavior of the signal  $f(x)$ .

**1.2.2 Lemma of Riemann–Lebesgue** *If  $f \in L^1(\mathbb{R}^d)$ , then  $\hat{f}$  is uniformly continuous and  $\lim_{|\omega| \rightarrow \infty} |\hat{f}(\omega)| = 0$ .*

The Fourier transform is even essentially bounded for  $L^1$ -functions, so we have a mapping

$$\mathcal{F}: L^1(\mathbb{R}^d) \rightarrow C_0(\mathbb{R}^d) ,$$

and as  $\|\mathcal{F}f\|_{L^\infty} \leq \|f\|_{L^1}$  we have  $\|\mathcal{F}\| \leq 1$ .

The relevance of the Schwartz class now becomes clear when we state that the Fourier transform yields a continuous bijection from  $\mathcal{S}(\mathbb{R}^d)$  to  $\mathcal{S}(\mathbb{R}^d)$ , what is not the case for  $L^1(\mathbb{R}^d)$ . This is because the Fourier transform has the important property to turn derivations into multiplications, i.e., to turn analytic operations into algebraic ones:

$$\mathcal{F}(D^\alpha f) = (2\pi i)^{|\alpha|} X^\alpha(\mathcal{F}f) \quad (1.2)$$

and

$$D^\alpha(\mathcal{F}f) = (-2\pi i)^{|\alpha|} \mathcal{F}(X^\alpha f) , \quad (1.3)$$

where  $X^\alpha$  is the multiplication operator  $(X^\alpha f)(x) := x^\alpha f(x)$ . It follows from the definition that  $\mathcal{S}(\mathbb{R}^d)$  is stable under derivations and multiplications, i.e.,

$$X^\alpha f \in \mathcal{S}(\mathbb{R}^d) \quad \text{and} \quad D^\alpha f \in \mathcal{S}(\mathbb{R}^d) \quad \forall \alpha \in \mathbb{N}_0^d \quad \forall f \in \mathcal{S}(\mathbb{R}^d) .$$

Using the reflection operator  $(\mathcal{I}f)(x) := f(-x)$  one can show that  $\mathcal{F}^2 = \mathcal{I}$  and so  $\mathcal{F}^4 = \text{Id}_{\mathcal{S}(\mathbb{R}^d)}$ . This yields

$$\mathcal{F}^{-1} = \mathcal{I}\mathcal{F} \quad (1.4)$$

and we can give an inversion formula explicitly:

**1.2.3 Theorem (Inversion Formula)** *The Fourier transform is a bijection from  $\mathcal{S}(\mathbb{R}^d)$  to  $\mathcal{S}(\mathbb{R}^d)$  and the inverse operator is given by*

$$(\mathcal{F}^{-1}f)(x) = \int_{\mathbb{R}^d} f(\omega) e^{2\pi i x \cdot \omega} d\omega \quad \forall x \in \mathbb{R}^d . \quad (1.5)$$

Furthermore,  $\mathcal{F}$  is unitary, i.e.,

$$\langle \mathcal{F}f, \mathcal{F}g \rangle_{L^2} = \langle f, g \rangle_{L^2} \quad \forall f, g \in \mathcal{S}(\mathbb{R}^d) .$$

Proof. See [Wer97, V.2.8]. □

We therefore have

$$\|\mathcal{F}f\|_{L^2} = \|f\|_{L^2} \quad \forall f \in \mathcal{S}(\mathbb{R}^d)$$

if we see  $\mathcal{S}(\mathbb{R}^d)$  as a dense subspace of  $L^2(\mathbb{R}^d)$ . Now, as we have  $\mathcal{F}$  well defined, bijective and even  $\|\cdot\|_{L^2}$ -isometrical on a dense subspace of  $L^2(\mathbb{R}^d)$ , it can be extended to an isometric operator on all of  $L^2(\mathbb{R}^d)$ . We will use the same symbol  $\mathcal{F}$  although the Fourier transform on  $L^2(\mathbb{R}^d)$  is not defined by (1.1) anymore if  $f \in L^2 \setminus L^1(\mathbb{R}^d)$ ;  $\mathcal{F}f$  is then no function, but an equivalence class of functions.

**1.2.4 Theorem of Plancherel** *If  $f \in L^1 \cap L^2(\mathbb{R}^d)$ , then*

$$\|f\|_{L^2} = \|\mathcal{F}f\|_{L^2}. \quad (1.6)$$

*As a consequence  $\mathcal{F}$  extends in a unique way to a unitary operator on  $L^2(\mathbb{R}^d)$  and satisfies Parseval's formula*

$$\langle f, g \rangle_{L^2} = \langle \mathcal{F}f, \mathcal{F}g \rangle_{L^2} \quad \forall f, g \in L^2(\mathbb{R}^d). \quad (1.7)$$

In signal analysis the isometry of the Fourier transform has the interpretation that it preserves the energy of a signal.

The Fourier transform can also be seen as a mapping between more general  $L^p$ -spaces:

**1.2.5 Theorem of Hausdorff–Young** *Let  $1 \leq p \leq 2$  and  $q$  be the conjugate exponent such that  $\frac{1}{p} + \frac{1}{q} = 1$ . Then  $\mathcal{F}: L^p(\mathbb{R}^d) \rightarrow L^q(\mathbb{R}^d)$  and  $\|\mathcal{F}f\|_{L^q} \leq \|f\|_{L^p}$ .*

Again, the Fourier transform is only defined by (1.1) if  $f \in L^1 \cap L^p(\mathbb{R}^d)$ .

For more details on the role of the Schwartz class for the Fourier transform see [Wer97, V].

## 1.3 Fundamental Operations

### Translation and Modulation

**1.3.1 Definition** For  $x, \omega \in \mathbb{R}^d$  we define the translation operator  $T_x$  by

$$(T_x f)(t) := f(t - x) \quad (1.8)$$

and the modulation operator  $M_\omega$  by

$$(M_\omega f)(t) := e^{2\pi i \omega \cdot t} f(t). \quad (1.9)$$

It is clear that  $T_x^{-1} = T_{-x}$  and  $M_\omega^{-1} = M_{-\omega}$ . The operator  $T_x$  is also called a time shift, and  $M_\omega$  a frequency shift. Operators of the form  $T_x M_\omega$  or  $M_\omega T_x$  are called time-frequency shifts (TF-shifts). They satisfy the commutation relations

$$T_x M_\omega = e^{-2\pi i x \cdot \omega} M_\omega T_x, \quad (1.10)$$

what follows from a simple computation. These commutation relations tell that compared to the translation of a modulated function, a modulation of a translated function yields an additional phase shift. It corresponds to a rotation of the function values in the complex plane, as  $|e^{i\theta}| \equiv 1$ .

Time-frequency shifts are isometries on  $L^p$  for all  $1 \leq p \leq \infty$ , i.e.,

$$\|T_x M_\omega f\|_{L^p} = \|f\|_{L^p} .$$

The interplay of TF-shifts with the Fourier transform is as follows:

$$\widehat{T_x f} = M_{-x} \hat{f} \quad \text{or} \quad \mathcal{F} T_x = M_{-x} \mathcal{F} \quad (1.11)$$

and

$$\widehat{M_\omega f} = T_\omega \hat{f} \quad \text{or} \quad \mathcal{F} M_\omega = T_\omega \mathcal{F} . \quad (1.12)$$

Note that these are actually discrete versions of (1.2) and (1.3). Equation (1.12) makes clear why modulations are called frequency shifts, because modulations become translations on the Fourier transform side.

In combination, we get the important property

$$\widehat{T_x M_\omega f} = M_{-x} T_\omega \hat{f} = e^{-2\pi i x \cdot \omega} T_\omega M_{-x} \hat{f} .$$

## Convolution, Involution and Reflection

**1.3.2 Definition** The *convolution* of two functions  $f, g \in L^1(\mathbb{R}^d)$  is the function  $f * g$  defined by

$$(f * g)(x) := \int_{\mathbb{R}^d} f(y) g(x - y) dy . \quad (1.13)$$

It satisfies

$$\|f * g\|_{L^1} \leq \|f\|_{L^1} \|g\|_{L^1}$$

and

$$\widehat{f * g} = \hat{f} \cdot \hat{g} .$$

In practise,  $f * g$  can be interpreted as  $f$  being “smeared” by  $g$  and vice versa. This can be used to smoothen a function by convolving it with a narrow bump function.

**1.3.3 Definition** The *involution* of a function is defined by

$$f^*(x) := \overline{f(-x)} \quad (1.14)$$

and the *reflection operator*  $\mathcal{I}$  by

$$(\mathcal{I}f)(x) := f(-x) . \quad (1.15)$$

It follows then that

$$\widehat{f^*} = \bar{\hat{f}} \quad \text{and} \quad \widehat{\mathcal{I}f} = \mathcal{I}\hat{f} .$$

With this notation one can often write the convolution operation as an inner product

$$(f * g)(x) = \langle f, T_x g^* \rangle_{L^2}$$

whenever both sides are defined. In addition, we have

$$(f * g)^* = g^* * f^* .$$

## 1.4 The Short-Time Fourier Transform

As described in Section 1.2 the Fourier transform provides information of the overall frequency behavior of a given signal. This is useful for signals that don't vary during the time, e.g., for analyzing the spectrum of a violin tone. However, for dynamic signals such as music, there is no information about what frequencies occur at what time when using the Fourier transform. The idea to overcome this is to split a given signal into short time-intervals on which it can be considered periodic, and look at the Fourier transforms of those pieces. Since sharp cut-offs introduce discontinuities in the signal and therefore unwanted noise in its frequency spectrum, a smooth window function is usually taken instead.

**1.4.1 Definition** Fix a window function  $g \in L^2(\mathbb{R}^d) \setminus \{0\}$ . The short-time Fourier transform (STFT), also called (continuous) Gabor transform of a function  $f \in L^2(\mathbb{R}^d)$  with respect to  $g$  is defined as

$$(\mathcal{V}_g f)(x, \omega) := \int_{\mathbb{R}^d} f(t) \overline{g(t-x)} e^{-2\pi i t \cdot \omega} dt \quad \text{for } x, \omega \in \mathbb{R}^d. \quad (1.16)$$

Note that the restriction to  $L^2(\mathbb{R}^d)$  is somehow artificial, as a (pointwise) product  $f \cdot g$  is  $L^1$ -integrable whenever  $f \in L^p(\mathbb{R}^d)$  and  $g \in L^q(\mathbb{R}^d)$  as follows by Hölder's inequality. But for  $f, g \in L^2(\mathbb{R}^d)$  the STFT  $\mathcal{V}_g f$  is uniformly continuous on  $\mathbb{R}^{2d}$  and can be written as

$$(\mathcal{V}_g f)(x, \omega) = \widehat{f \cdot T_x \bar{g}}(\omega) \quad (1.17)$$

$$= \langle f, M_\omega T_x g \rangle_{L^2} \quad (1.18)$$

$$= e^{-2\pi i x \cdot \omega} (f * M_\omega g^*)(x). \quad (1.19)$$

By the inner product representation (1.18) the STFT can be extended to a greater class of functions or distributions where the integral (1.16) is no longer defined.

The STFT as a function in  $x$  and  $\omega$  seems to provide the possibility to obtain information about the occurrence of arbitrary frequencies  $\omega$  at arbitrary locations  $x$  as desired. This is done by projecting a signal  $f$  onto corresponding time-frequency shifted versions of a single envelope function  $g$ . However, there is a fundamental observation that destroys this dream: the *uncertainty principle*, telling that even the “nicest” function  $g$  cannot be well-concentrated in both time and frequency, i.e., essential support  $\text{supp } g \subseteq \mathbb{R}^d$  and essential spectrum  $\text{supp } \hat{g} \subseteq \mathbb{R}^d$  will always occupy a minimal area on the time-frequency plane or phase space  $\mathbb{R}^d \times \mathbb{R}^d$ . We will give the following version here, which is sometimes referred to as the Heisenberg–Pauli–Weyl inequality.

**1.4.2 Theorem (Uncertainty principle)** *If  $f \in L^2(\mathbb{R})$  and  $a, b \in \mathbb{R}$  are arbitrary, then*

$$\left( \int_{\mathbb{R}} (x-a)^2 |f(x)|^2 dx \right)^{1/2} \left( \int_{\mathbb{R}} (\omega-b)^2 |\hat{f}(\omega)|^2 d\omega \right)^{1/2} \geq \frac{1}{4\pi} \|f\|_{L^2}^2. \quad (1.20)$$

*Equality holds if and only if  $f$  is a multiple of  $T_a M_b \varphi_c(x) = e^{2\pi i b(x-a)} e^{-\pi(x-a)^2/c}$  for some  $a, b \in \mathbb{R}$  and  $c > 0$ .*

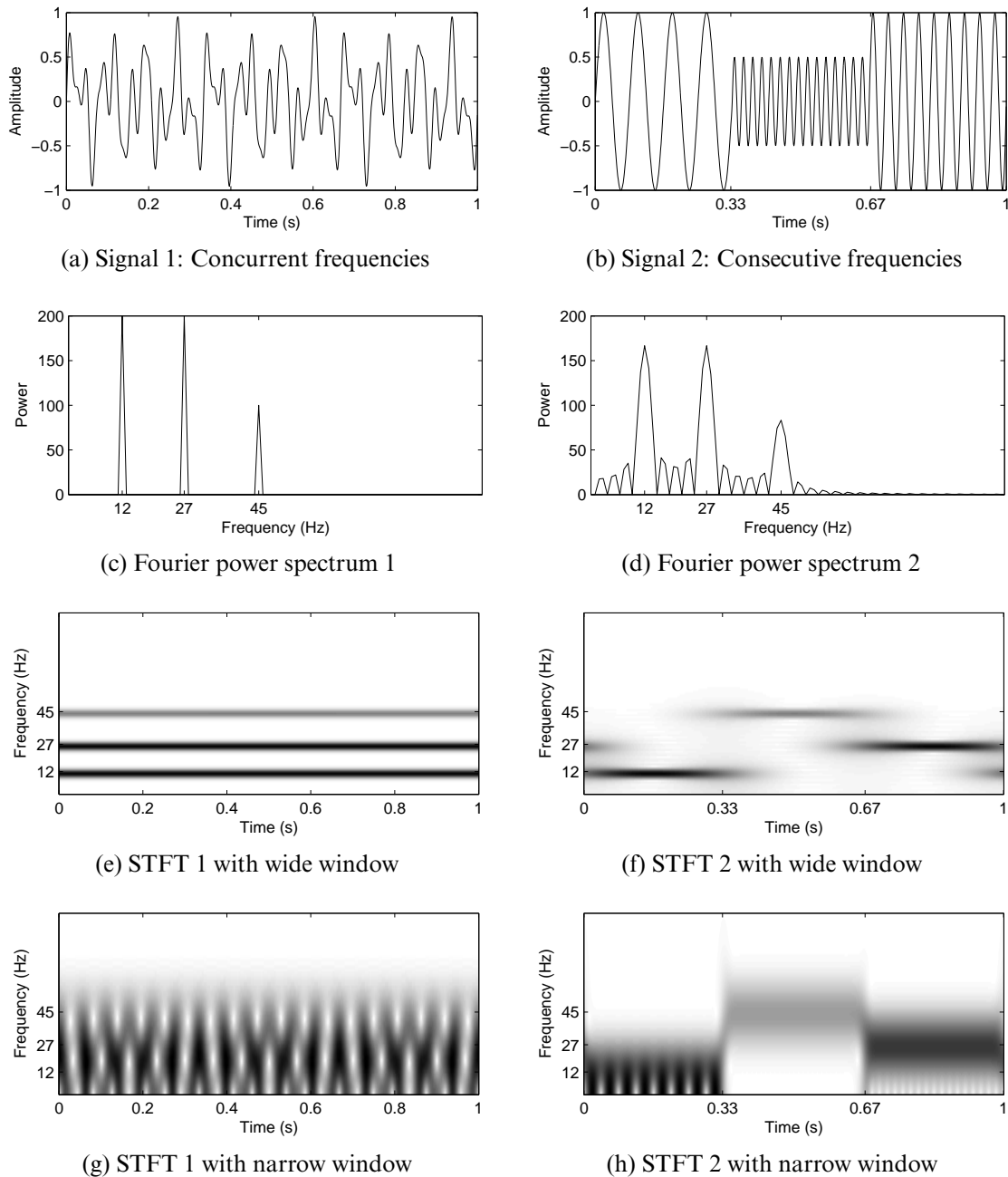


FIGURE 1.1: Two signals and their (short-time) Fourier transforms. Despite the visible difference between (a) and (b), their Fourier transforms (c) and (d) look similar (up to a discontinuity in the second signal) and indicate their same frequency content. However, for (a) the time-frequency behavior is not clear at first sight. STFT (e) of the first signal incorporates a wide window and yields a high frequency resolution, telling that the spectrum is time-invariant. STFT (f) uses the same wide window and shows that the frequencies in the second signal are time-variant, but time-resolution is low and the frequency lines blend horizontally. Using a narrow window, STFT (h) indicates the time-points where frequencies change, but frequency-resolution becomes bad, yielding broad frequency bands. For the same narrow window, STFT (g) is incapable of detecting the low frequencies in the first signal.

Proof. See [Grö01, 2.2].  $\square$

The standard form of the uncertainty principle involves an  $f \in L^2(\mathbb{R}^d)$  with  $\|f\|_{L^2} = 1$ , the minimum over all  $a \in \mathbb{R}$  for the first term in (1.20), written as  $\Delta_f x$ , and the minimum over all  $b \in \mathbb{R}$  for the second term, written as  $\Delta_f \omega$ . Using these standard deviations, the uncertainty principle reads as

$$\Delta_f x \Delta_f \omega \geq \frac{1}{4\pi}.$$

This theorem shows that the STFT has limitations in its time-frequency resolution capability: Low frequencies can hardly be located with narrow windows, and similarly, short pulses remain invisible for wide windows. The choice of the analyzing window is therefore crucial and leads to the build-up of dedicated window classes for TF-analysis in general Hilbert spaces.

Just like the Fourier transform, the STFT is another kind of time-frequency representation of a signal. This again raises the question of how to reconstruct the signal from its time-frequency representation. To approach this we need the result of the orthogonality relations of the STFT as an equivalence to Parseval's formula (1.7) for the Fourier transform:

**1.4.3 Theorem (Orthogonality relations for STFT)** *Let  $f_1, f_2, g_1, g_2 \in L^2(\mathbb{R}^d)$ . Then  $\mathcal{V}_{g_j} f_j \in L^2(\mathbb{R}^{2d})$  for  $j \in \{1, 2\}$ , and*

$$\langle \mathcal{V}_{g_1} f_1, \mathcal{V}_{g_2} f_2 \rangle_{L^2(\mathbb{R}^{2d})} = \langle f_1, f_2 \rangle_{L^2} \overline{\langle g_1, g_2 \rangle_{L^2}}.$$

Proof. See [Grö01, 3.2.1].  $\square$

**1.4.4 Corollary** *If  $f, g \in L^2(\mathbb{R}^d)$ , then*

$$\|\mathcal{V}_g f\|_{L^2(\mathbb{R}^{2d})} = \|f\|_{L^2} \|g\|_{L^2}.$$

*In the case of  $\|g\|_{L^2} = 1$  we have*

$$\|f\|_{L^2} = \|\mathcal{V}_g f\|_{L^2(\mathbb{R}^{2d})} \quad \forall f \in L^2(\mathbb{R}^d) \quad (1.21)$$

*and therefore the STFT as an isometry from  $L^2(\mathbb{R}^d)$  into  $L^2(\mathbb{R}^{2d})$ .*

Formula (1.21) shows that the STFT, too, preserves the energy of a signal; it corresponds to (1.6) which shows the same property for the Fourier transform. Therefore,  $f$  is completely determined by  $\mathcal{V}_g f$  and the inversion is given by a vector-valued integral:

**1.4.5 Corollary (Inversion formula for the STFT)** *Be  $g, \gamma \in L^2(\mathbb{R}^d)$  and  $\langle g, \gamma \rangle \neq 0$ . Then*

$$f = \frac{1}{\langle \gamma, g \rangle_{L^2}} \iint_{\mathbb{R}^{2d}} \mathcal{V}_g f(x, \omega) M_\omega T_x \gamma \, d\omega \, dx \quad \forall f \in L^2(\mathbb{R}^d). \quad (1.22)$$

Proof. See [Grö01, 3.2.3].  $\square$

Formula (1.22) is similar to the inversion of the Fourier transform, with the difference that the elementary functions  $e^{2\pi i x \cdot \omega}$  occurring in  $f(x) = \int \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega$  are not  $L^2$ -integrable in contrary to the elementary functions  $M_\omega T_x \gamma$ . As  $\langle g, g \rangle_{L^2} = \|g\|_{L^2}^2 \neq 0$  for all allowed choices for  $g$ , one might as well take the same window for reconstruction that has been originally used in the STFT.

The time-frequency analysis of signals is usually done by three subsequent steps:

1. *Analysis*: A signal is transformed into a joint time-frequency representation, like the STFT.
2. *Processing*: The obtained signal representation is then manipulated in a certain way, e.g. by restriction to a set where something interesting seems to happen or where the values are above a given threshold.
3. *Synthesis*: The processed representation is then inverted to create a new signal which corresponds to the respectively processed original signal.

A function is completely represented by its STFT, but in a highly redundant way: Elementary functions are incorporated as infinitesimal TF-shifts of a single window function. To overcome the uncertainty principle, the analyzing window  $g$  and its Fourier transform  $\hat{g}$  should be chosen to both decay rapidly, but even if they have compact support, their TF-shifted versions largely overlap. This fact yields the problem of high computational effort in practice. Besides that, a computational implementation can only be obtained by a discretization of both the functions and the STFT. Therefore, only sampled versions of the STFT are possible and only certain locations and frequencies are used for analyzing a given signal. The very challenge is to find the appropriate steps in time and frequency and to obtain good time-frequency resolution in spite of the uncertainty principle.

A common task in analysis is the approximation or representation of arbitrary functions with the help of “special” functions of similar structure which are well-understood and possess easy-to-handle analytical properties. But besides that, there is the question about how a set of elementary functions has to be structured in a more general way such that the analysis and synthesis operations are “reasonable”; in Chapter 2 we want to be more precise on this and introduce the more general concept of frames.

## 1.5 The Gaussian Function

We have seen in Theorem 1.4.2 that there exists a certain function that minimizes the uncertainty principle: The Gaussian function. It is therefore an ideal candidate for being used as window function in the STFT to obtain best time-frequency resolution. Besides that minimization property, it is (up to constants) invariant under the Fourier transform and therefore an eigenfunction of this transform.

**1.5.1 Definition** The non-normalized Gaussian function of width  $a > 0$  is defined as

$$\varphi_a: \mathbb{R}^d \rightarrow \mathbb{R}, \quad \varphi_a(x) := e^{-\pi x^2/a}.$$

Note that  $x^2 = x \cdot x$ . We abbreviate  $\varphi := \varphi_1$ .

Some authors define the Gaussian using a complex parameter  $c > 0$  and splitting into the real and complex part of  $\frac{1}{c}$ , i.e.,  $\varphi_c(x) = e^{-\pi a x^2} e^{-\pi i b x^2}$  what corresponds to a Gaussian  $\varphi_{1/a}$  multiplied by a chirp function.

Some sources like [Wer97, V.2.6] or [Grö01, 1.5.1] prove the Fourier transform invariance of the Gaussian by using the fact that initial value problems for linear common differential equations have unique solutions, but here we want to show a different approach and restrict to the case of  $x \in \mathbb{R}$ .

**1.5.2 Theorem** The function  $\varphi(x) = e^{-\pi x^2}$  satisfies  $\hat{\varphi} = \varphi$ , i.e.,

$$(\mathcal{F}\varphi)(\omega) = e^{-\pi\omega^2}.$$

**Proof.** [Zim98] The Fourier transform of  $\varphi(x)$  with  $x \in \mathbb{R}$  is given as

$$\begin{aligned} \hat{\varphi}(\omega) &= \int_{\mathbb{R}} e^{-\pi x^2} e^{-2\pi i \omega x} dx \\ &= \int_{\mathbb{R}} e^{-\pi(x+i\omega)^2} e^{-\pi\omega^2} dx \\ &= e^{-\pi\omega^2} \int_{\mathbb{R}} e^{-\pi(x+i\omega)^2} dx. \end{aligned} \tag{1.23}$$

Writing  $h(x, \omega) := e^{-\pi(x+i\omega)^2}$ , we have  $\frac{\partial h}{\partial \omega}(x, \omega) = -2\pi i(x+i\omega)e^{-\pi(x+i\omega)^2}$  which is uniformly integrable with respect to  $x$  over bounded ranges of  $\omega$ . Therefore, we may switch the order of differentiation and integration in the differentiation of the parameter integral  $H(\omega) := \int_{\mathbb{R}} h(x, \omega) dx$  and get

$$\begin{aligned} \frac{dH}{d\omega}(\omega) &= \int_{\mathbb{R}} \frac{\partial h}{\partial \omega}(x, \omega) dx \\ &= \int_{\mathbb{R}} -2\pi i(x+i\omega)e^{-\pi(x+i\omega)^2} dx \\ &= \left[ i e^{-\pi(x+i\omega)^2} \right]_{x=-\infty}^{\infty} \\ &= 0 \quad \forall \omega \in \mathbb{R}. \end{aligned}$$

This shows that  $H(\omega)$  is constant and thus  $H(\omega) \equiv H(0)$ . Its value is given by

$$\begin{aligned} H(0)^2 &= \int_{\mathbb{R}} e^{-\pi x^2} dx \int_{\mathbb{R}} e^{-\pi y^2} dy \\ &= \iint_{\mathbb{R} \times \mathbb{R}} e^{-\pi(x^2+y^2)} dx dy \end{aligned}$$



$$\begin{aligned}
&= \iint_{\mathbb{R}^+ \times [0, 2\pi]} e^{-\pi r^2} r \, dr \, d\theta \\
&= \int_0^\infty 2\pi r e^{-\pi r^2} \, dr \\
&= \left[ -e^{-\pi r^2} \right]_{r=0}^\infty = 1.
\end{aligned}$$

As  $H(0) \geq 0$  since  $h(x, 0) \geq 0$ , we thus have  $H(0) = 1$  and (1.23) yields  $\hat{\varphi}(\omega) = e^{-\pi\omega^2}$  as asserted.  $\square$

## 1.6 Tools from Linear Algebra

### Matrix Norms and Inner Products

For numerical calculations we will need an inner product of matrices and a matrix norm. The Frobenius norm is quickly to compute, as it incorporates matrices as vectors.

**1.6.1 Definition** The Frobenius norm of  $A \in \mathbb{C}^{m \times n}$  is defined as

$$\|A\|_F := \sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |a_{ij}|^2}.$$

It can be expressed as

$$\|A\|_F = \sqrt{\operatorname{tr}(AA^*)} = \sqrt{\operatorname{tr}(A^*A)},$$

where  $\operatorname{tr} B$  is the trace of  $B \in \mathbb{C}^{k \times \ell}$  given as the sum of its diagonal entries, i.e.,  $\operatorname{tr} B := \sum_{i=0}^{\min\{k, \ell\}-1} b_{ii}$ .

Observe that this norm is the same as the  $\ell^2$ -norm of the matrix if it is seen as an  $mn$ -dimensional vector. One can show that  $\|A\| \leq \|A\|_F$ . The Frobenius norm is not induced by a vector norm, but by the inner product

$$\langle A, B \rangle_F := \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} a_{ij} \overline{b_{ij}} = \operatorname{tr}(AB^*).$$

### Singular Value Decomposition and Pseudoinverse

The singular value decomposition (SVD) and the pseudoinverse of a matrix are important concepts in linear algebra which will reappear in the theory of frames in Chapter 2.

**1.6.2 Theorem** Be  $A \in \mathbb{C}^{m \times n}$  with rank  $r \geq 1$ . Then there exist unitary matrices  $U \in \mathbb{C}^{m \times m}$ ,  $V \in \mathbb{C}^{n \times n}$ , a matrix  $\Sigma \in \mathbb{C}^{m \times n}$  and unique values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  such that

$$\Sigma = \begin{pmatrix} \operatorname{diag}(\sigma_1, \dots, \sigma_r) & 0_{r \times (n-r)} \\ 0_{(m-r) \times r} & 0_{(m-r) \times (n-r)} \end{pmatrix}$$

with  $0_{k \times \ell} := 0 \in \mathbb{C}^{k \times \ell}$  and

$$A = U \Sigma V^* . \quad (1.24)$$

**Proof.** See [Chr03, 1.5.4].  $\square$

Formula (1.24) is called the **singular value decomposition (SVD)** of the matrix  $A$ , and the numbers  $\sigma_1, \dots, \sigma_r$  are the **singular values** for  $A$ . They are the square roots of the positive eigenvalues for  $A^*A$ . The involved unitary matrices  $U$  and  $V$  are not necessarily unique.

**1.6.3 Definition** For  $A \in \mathbb{C}^{m \times n}$  with an SVD as in Theorem 1.6.2, the **pseudoinverse** or **Moore–Penrose inverse** of  $A$  is given by

$$A^\dagger := V \Sigma^\dagger U^* \in \mathbb{C}^{n \times m}$$

where

$$\Sigma^\dagger := \begin{pmatrix} \text{diag}(\sigma_1^{-1}, \dots, \sigma_r^{-1}) & 0_{r \times (m-r)} \\ 0_{(n-r) \times r} & 0_{(n-r) \times (m-r)} \end{pmatrix} \in \mathbb{C}^{n \times m} .$$

If  $A$  is invertible, then  $A^\dagger = A^{-1}$  and  $\Sigma^\dagger = \Sigma^{-1}$ . The importance of the pseudoinverse lies in the solution of an important minimization problem:

**1.6.4 Theorem** Let  $A \in \mathbb{C}^{m \times n}$ . For  $y \in \text{ran } A$ , the equation  $Ax = y$  has a unique solution of minimal norm, namely  $x = A^\dagger y$ .

**Proof.** See [Chr03, 1.5.2].  $\square$

## 1.7 Tensor Products

As we'll look at tensor products of Hilbert spaces later, we repeat a few facts from multilinear algebra.

**1.7.1 Definition** For finitely many vector spaces  $V_i$ ,  $i \in I$ , and a vector space  $W$ , all over the field  $K$ , a mapping  $f: \prod_{i \in I} V_i \rightarrow W$  is called **( $K$ -)multilinear**, if  $f$  is ( $K$ -)linear in each component  $V_i$ ,  $i \in I$ . The space of all multilinear functions is  $\text{Mult}_K(V_i, i \in I; W)$ , and we write  $\text{Mult}(V_i, i \in I) := \text{Mult}_K(V_i, i \in I; K)$  for the multilinear forms on  $\prod_{i \in I} V_i$ . The **tensor product** of  $V_i$ ,  $i \in I$ , is now defined as  $\bigotimes_{i \in I} V_i := \text{Mult}(V_i, i \in I)^*$ .

If we choose  $I = \{1, \dots, n\}$  for simplicity and  $f \in \text{Mult}(V_1, \dots, V_n)$ , i.e., a multilinear form  $f: V_1 \times \dots \times V_n \rightarrow K$ , the universal multilinear mapping

$$\tau: V_1 \times \dots \times V_n \rightarrow V_1 \otimes \dots \otimes V_n$$

is defined by

$$\tau(x_1, \dots, x_n)(f) := f(x_1, \dots, x_n) ,$$

and we abbreviate  $x_1 \otimes \dots \otimes x_n := \tau(x_1, \dots, x_n)$ . In short, we define **linear functionals** of multilinear forms by the multilinear forms themselves. Note that tensor products

are associative and Abelian. The universal property of the tensor product is that any multilinear function  $f: \prod_{i \in I} V_i \rightarrow W$  is uniquely determined by a linear mapping  $F: \bigotimes_{i \in I} V_i \rightarrow W$  such that  $f = F \circ \tau$ , see e.g. [SW01, 6.A.1].

$V_1 \otimes \cdots \otimes V_n$  is created by the separable tensors  $x_1 \otimes \cdots \otimes x_n$  for  $x_i \in V_i$ . Every tensor is a sum of separable tensors due to

$$a(x_1 \otimes \cdots \otimes x_i \otimes \cdots \otimes x_n) = x_1 \otimes \cdots \otimes ax_i \otimes \cdots \otimes x_n,$$

and especially

$$x_1 \otimes \cdots \otimes (x_i + x'_i) \otimes \cdots \otimes x_n = (x_1 \otimes \cdots \otimes x_i \otimes \cdots \otimes x_n) + (x_1 \otimes \cdots \otimes x'_i \otimes \cdots \otimes x_n).$$

We notice that not every tensor  $x \in V_1 \otimes \cdots \otimes V_n$  is separable in general, i.e., we generally have  $x \neq x_1 \otimes \cdots \otimes x_n$  for any  $x_i \in V_i$ , because e.g.

$$x \otimes y + x' \otimes y' \neq (x + x') \otimes (y + y') = x \otimes y + x \otimes y' + x' \otimes y + x' \otimes y'$$

for  $x, x' \in V$  and  $y, y' \in W$ .

**1.7.2 Theorem** Let  $V_i, i \in I$ , be vector spaces over  $K$  with bases  $v_{r_i}^{(i)}, r_i \in R_i$ , then

$$\bigotimes_{i \in I} v_{r_i}^{(i)}, \quad (r_i)_{i \in I} \in \prod_{i \in I} R_i,$$

is a basis of the tensor product  $\bigotimes_{i \in I} V_i$ .

*Proof.* See [SW01, 6.A.2]. □

**1.7.3 Corollary** If the vector spaces  $V_i, i \in I$ , are finite-dimensional, then  $\bigotimes_{i \in I} V_i$  is finite-dimensional as well, and  $\dim \bigotimes_{i \in I} V_i = \prod_{i \in I} \dim V_i$ .

Compare that in contrary  $\dim \prod_{i \in I} V_i = \sum_{i \in I} \dim V_i$ . So if  $\{v_1, \dots, v_m\}$  is a basis for  $V$  and  $\{w_1, \dots, w_n\}$  a basis for  $W$ , then  $\{v_i \otimes w_j\}_{i,j}$  for  $i \in \{1, \dots, m\}$  and  $j \in \{1, \dots, n\}$  is a basis of  $mn$  elements for  $V \otimes W$ .

A bilinear form  $\Phi: V \times W \rightarrow K$  is simply an element of  $(V \otimes W)^*$ , what is isomorphic to  $V^* \otimes W^*$  if both spaces are finite-dimensional, and so  $\Phi$  can be called a tensor as well. To be able to identify a sesquilinear form  $\Phi: V \times W \rightarrow \mathbb{C}$  as a tensor, we have to replace  $W$  by its anti vector space  $\bar{W}$ , where the scalar multiplication is given as  $(a, x) \mapsto \bar{a}x$ . Particularly, a sesquilinear form on  $V$  is a tensor in  $V^* \otimes \bar{V}^*$ . A scalar product on the vector space  $V$  over  $\mathbb{K}$  is a metrical tensor.

For finitely many vector spaces  $V_i$  and  $W_i$  over  $K$  and bilinear forms  $\Phi_i: V_i \times W_i \rightarrow K, i \in I$ , the mapping

$$\bigotimes_{i \in I} \Phi_i: \bigotimes_{i \in I} (V_i \otimes W_i) \longrightarrow K$$

is a linear form. Thus, the bilinear mapping

$$\Phi: \left( \bigotimes_{i \in I} V_i \right) \times \left( \bigotimes_{i \in I} W_i \right) \longrightarrow K \quad \text{with} \quad \left( \bigotimes_{i \in I} x_i, \bigotimes_{i \in I} y_i \right) \longmapsto \prod_{i \in I} \Phi_i(x_i, y_i)$$

is well defined and identified with  $\bigotimes_{i \in I} \Phi_i$ . In this sense, a family of sesquilinear forms  $\Phi_i: V_i \times W_i \rightarrow \mathbb{C}$  defines a sesquilinear mapping  $\bigotimes_{i \in I} \Phi_i$ . Particularly, scalar products  $\Phi_i: V_i \times V_i \rightarrow \mathbb{K}$  yield a scalar product  $\bigotimes_{i \in I} \Phi_i$ , and

$$\left\| \bigotimes_{i \in I} x_i \right\| = \prod_{i \in I} \|x_i\|, \quad x_i \in V_i, i \in I.$$

**1.7.4 Theorem** *If  $v_{r_i}^{(i)}$ ,  $r_i \in R_i$ , are Hilbert bases (complete orthonormal systems) of  $V_i$  over  $\mathbb{K}$ ,  $i \in I$ , then their tensor product*

$$\bigotimes_{i \in I} v_{r_i}^{(i)}, \quad (r_i)_{i \in I} \in \prod_{i \in I} R_i,$$

*is a Hilbert basis of  $\bigotimes_{i \in I} V_i$ .*

**Proof.** [SW01, 6.A.15] We restrict to the case  $V \otimes W$  and have to show that all separable tensors  $x \otimes y \in V \otimes W$  are in the closed linear span of the linear combinations of  $\{v_{r_1}^{(1)} \otimes v_{r_2}^{(2)}\}_{r_1, r_2}$ . This follows from the estimate

$$\begin{aligned} \|x \otimes x' - y \otimes y'\| &= \|x \otimes x' - y \otimes x' + y \otimes x' - y \otimes y'\| \\ &= \|(x - y) \otimes x' + y \otimes (x' - y')\| \\ &\leq \|(x - y) \otimes x'\| + \|y \otimes (x' - y')\| \\ &= \|x - y\| \|x'\| + \|y\| \|x' - y'\|. \quad \square \end{aligned}$$

A tensor product of Hilbert spaces also carries a scalar product due to the above considerations, but it is only a Hilbert space itself if all factors except for at most one are finite-dimensional. So, the completion of  $\bigotimes_{i \in I} \mathcal{H}_i$  for Hilbert spaces  $\mathcal{H}_i$ ,  $i \in I$ , is usually of interest, and the complete Hilbert space tensor product is denoted by

$$\widehat{\bigotimes_{i \in I} \mathcal{H}_i}.$$

Theorem 1.7.4 reveals a Hilbert basis for the complete tensor product if Hilbert bases for its factors are given. If  $\mathcal{H}_i$ ,  $i \in I$ , are separable Hilbert spaces, then  $\widehat{\bigotimes_{i \in I} \mathcal{H}_i}$  is separable as well.

# Chapter 2

## Frames in Hilbert Spaces

### 2.1 From Bases to Frames

A classical problem in analysis is to try to split a signal  $f$  into a convergent series of elementary functions  $g_i$  with equal structure. These should have well-known and easy-to-handle analytical properties and should simplify the understanding of the signal  $f$ . One way to express  $f$  as a linear combination  $\sum_{i \in I} c_i g_i$  is using the concept of bases.

**2.1.1 Definition** Two sequences  $\{f_i\}_{i \in I}$  in a Banach space  $X$  and  $\{g_i\}_{i \in I}$  in the dual  $X^*$  are biorthogonal if

$$g_j(f_i) = \delta_{i,j} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

The notion of a (Schauder) basis for  $X$  allows to have unique expansions of the form

$$f = \sum_{i \in I} c_i(f) g_i \quad \forall f \in X$$

where  $\{c_i\}_{i \in I}$  are linear functionals in  $X^*$  and biorthogonality is given by  $c_j(g_i) = \delta_{i,j}$ . For Hilbert spaces, biorthogonality can be expressed by the respective inner product due to Riesz' Representation Theorem. In the case of an orthonormal basis  $\{e_i\}_{i \in I}$  in a separable Hilbert space  $\mathcal{H}$ , the coefficients  $\{c_i(f)\}_{i \in I}$  are given by  $\{\langle f, e_i \rangle_{\mathcal{H}}\}_{i \in I}$  and the expansion takes the form [Wer97, V.4.9]

$$f = \sum_{i \in I} \langle f, e_i \rangle_{\mathcal{H}} e_i \quad \forall f \in \mathcal{H}.$$

This shows that an orthonormal basis is biorthogonal to itself. Given an arbitrary basis  $\{e_i\}_{i \in I}$  for  $\mathcal{H}$ , one can show [Chr03, 3.3.2] that there exists a unique biorthogonal basis  $\{g_i\}_{i \in I}$  in  $\mathcal{H}$  for which

$$f = \sum_{i \in I} \langle f, g_i \rangle_{\mathcal{H}} e_i \quad \forall f \in \mathcal{H}. \tag{2.1}$$

Besides the question of how to actually find that biorthogonal sequence, it is not really convenient in practise to use a basis as set of expansion elements: Although

an orthonormal basis is biorthogonal to itself, the Gram-Schmidt orthogonalization routine is numerically fairly unstable. Orthogonality and the basis property alone are already rather restricting. In addition, a perturbation of single basis elements immediately results in a perturbation of the sum. And in the case of the loss of a single basis element, one even cannot span the whole space anymore with the remaining elements.

The sought generalization of bases leads to the notion of frames. We explore their structure by first considering the analysis and synthesis operation that we already came across in Section 1.4.

**2.1.2 Definition** Given a sequence  $\{g_i\}_{i \in I}$  in  $\mathcal{H}$ , the analysis operator or coefficient operator  $C$  is given by

$$Cf := \{\langle f, g_i \rangle_{\mathcal{H}}\}_{i \in I}$$

and the synthesis operator or reconstruction operator  $D$  is defined for a finite sequence  $c = \{c_i\}_{i \in I}$  as

$$Dc := \sum_{i \in I} c_i g_i \in \mathcal{H}.$$

The frame operator  $S$  is defined as

$$Sf := DCf = \sum_{i \in I} \langle f, g_i \rangle_{\mathcal{H}} g_i. \quad (2.2)$$

In this general setting of the analysis operation, we don't have any structural knowledge about the coefficient sequence  $\{\langle f, g_i \rangle_{\mathcal{H}}\}_{i \in I}$ . For a reasonable and stable reconstruction of  $f$ , we need the analysis operation to be continuous. Also, we ask about conditions for convergence of the sum  $\sum_{i \in I} c_i g_i$  if the sequence  $\{c_i\}_{i \in I}$  is not finite.

**2.1.3 Definition** A sequence  $\{g_i\}_{i \in I}$  in  $\mathcal{H}$  is called a Bessel sequence if there exists a constant  $B > 0$  such that

$$\|Cf\|_2^2 = \sum_{i \in I} |\langle f, g_i \rangle_{\mathcal{H}}|^2 \leq B \|f\|_{\mathcal{H}}^2 \quad \forall f \in \mathcal{H}. \quad (2.3)$$

Every number  $B$  that satisfies (2.3) is called a Bessel bound for  $\{g_i\}_{i \in I}$ .

Technically, it would be enough to state that  $\{\langle f, g_i \rangle_{\mathcal{H}}\} \in \ell^2(I)$ , but the knowledge of a bounding constant  $B$  is stronger than just knowing that the sum does converge. It is enough to have the Bessel property on a dense subset of  $\mathcal{H}$  [Chr03, 3.2.6].

**2.1.4 Theorem** Let  $\{g_i\}_{i \in I}$  be a sequence in  $\mathcal{H}$ . Then  $\{g_i\}_{i \in I}$  is a Bessel sequence with Bessel bound  $B$  if and only if

$$D: \{c_i\}_{i \in I} \rightarrow \sum_{i \in I} c_i g_i$$

is a well-defined bounded operator  $\ell^2(I) \rightarrow \mathcal{H}$  with  $\|D\| \leq \sqrt{B}$ .

Proof. See [Chr03, 3.2.3].  $\square$

As a consequence, the series  $\sum_{i \in I} c_i g_i$  converges unconditionally for all  $\{c_i\}_{i \in I} \in \ell^2(I)$ , and so does the series defining the frame operator  $S$  for all  $f \in \mathcal{H}$ .

We have an interesting relation between the analysis and synthesis operation: The operators  $C$  and  $D$  are adjoint to each other, i.e.,  $C = D^*$ , and for Bessel sequences the analysis operator becomes the bounded linear operator

$$D^*: \mathcal{H} \rightarrow \ell^2(I), \quad D^* f = \{\langle f, g_i \rangle_{\mathcal{H}}\}_{i \in I}.$$

The frame operator is then the self-adjoint operator  $S = C^*C = DD^*$ . This justifies calling the synthesis operator  $D$  the pre-frame operator.

However, all this does not yet tell us how to obtain the expansion coefficients  $\{c_i\}_{i \in I}$  in the expansion  $f = \sum_{i \in I} c_i g_i$ . As we can see from (2.2), we'd need the frame operator to be invertible. For this, we need the analysis operator to be positive and thus injective—that's where we arrive at the definition of a frame.

**2.1.5 Definition** A sequence  $\{g_i\}_{i \in I}$  in a (separable) Hilbert space  $\mathcal{H}$  is called a **frame** if there exist positive constants  $A, B > 0$  such that

$$A\|f\|_{\mathcal{H}}^2 \leq \sum_{i \in I} |\langle f, g_i \rangle_{\mathcal{H}}|^2 \leq B\|f\|_{\mathcal{H}}^2 \quad \forall f \in \mathcal{H}. \quad (2.4)$$

Any constants  $A$  and  $B$  satisfying (2.4) are called **frame bounds**. A frame is called **tight** if  $A = B$ .

The frame inequality (2.4) is some kind of “approximate Plancherel formula”. As  $\|D^* f\|_2^2 = \sum_{i \in I} |\langle f, g_i \rangle_{\mathcal{H}}|^2$ , the left hand side of (2.4) shows that  $D^*$  is positive with  $\|D^*\| \geq \sqrt{A}$ . Observe that an orthonormal basis is a tight frame with  $A = B = 1$  and that a frame is a Bessel sequence. A frame is **complete** in  $\mathcal{H}$ , meaning that  $\overline{\text{span}}\{g_i\}_{i \in I} = \mathcal{H}$ . For frames, the analysis operator has closed range in  $\ell^2(I)$ .

Because we have

$$\langle Sf, f \rangle_{\mathcal{H}} = \left\langle \sum_{i \in I} \langle f, g_i \rangle_{\mathcal{H}} g_i, f \right\rangle_{\mathcal{H}} = \sum_{i \in I} |\langle f, g_i \rangle_{\mathcal{H}}|^2,$$

a reformulation of (2.4) yields

$$A\|f\|_{\mathcal{H}}^2 \leq \langle Sf, f \rangle_{\mathcal{H}} \leq B\|f\|_{\mathcal{H}}^2,$$

showing that  $S$  is bounded positive and thus invertible with inverse  $S^{-1}$ . Consequently, every  $f \in \mathcal{H}$  has an expansion of the form

$$f = S^{-1}Sf = \sum_{i \in I} \langle f, g_i \rangle_{\mathcal{H}} S^{-1}g_i \quad (2.5)$$

and

$$f = SS^{-1}f = \sum_{i \in I} \langle S^{-1}f, g_i \rangle_{\mathcal{H}} g_i = \sum_{i \in I} \langle f, S^{-1}g_i \rangle_{\mathcal{H}} g_i, \quad (2.6)$$

where we used the self-adjointness of  $S$  and therefore of  $S^{-1}$ . Writing  $T \leq U$  whenever  $\langle Tf, f \rangle_{\mathcal{H}} \leq \langle Uf, f \rangle_{\mathcal{H}}$  for all  $f \in \mathcal{H}$ , we get that

$$A \operatorname{Id}_{\mathcal{H}} \leq S \leq B \operatorname{Id}_{\mathcal{H}} . \quad (2.7)$$

Multiplying (2.7) with  $S^{-1}$  shows that

$$\frac{1}{B} \operatorname{Id}_{\mathcal{H}} \leq S^{-1} \leq \frac{1}{A} \operatorname{Id}_{\mathcal{H}} ,$$

yielding that  $\{S^{-1}g_i\}_{i \in I}$  is again a frame with lower frame bound  $1/B$  and upper frame bound  $1/A$ .

Thus, with (2.5) and (2.6) we already have found a way to obtain a biorthogonal sequence for reconstruction. If we think of (2.1), we know at least one dual frame  $\{\tilde{g}_i\}_{i \in I}$  for a given frame  $\{g_i\}_{i \in I}$  such that the non-orthogonal expansion

$$f = \sum_{i \in I} \langle f, g_i \rangle \tilde{g}_i = \sum_{i \in I} \langle f, \tilde{g}_i \rangle g_i$$

holds for all  $f \in \mathcal{H}$ . The frame  $\{S^{-1}g_i\}_{i \in I}$  we obtained above is called the *canonical dual frame* to  $\{g_i\}_{i \in I}$ . Among all possible dual frames, this is the one where the expansion coefficients have minimal  $\ell^2$ -norm:

**2.1.6 Proposition** *Be  $\{g_i\}_{i \in I}$  a frame for  $\mathcal{H}$  and  $f \in \mathcal{H}$ . Let  $a = \{\langle f, S^{-1}g_i \rangle_{\mathcal{H}}\}_{i \in I}$  be the canonical frame expansion coefficients such that  $f = \sum_{i \in I} a_i g_i$ . If  $f$  has another representation  $f = \sum_{i \in I} c_i g_i$  for some coefficients  $c \in \ell^2(I)$ , then*

$$\|c\|_2^2 = \|a\|_2^2 + \|c - a\|_2^2 .$$

*Proof.* See [Grö01, 5.1.4] or [Chr03, 1.1.5]. □

So, in contrary to a basis, the expansion coefficients for a frame are not unique in general. The minimality in the  $\ell^2$ -norm is not always the ideal property for finding coefficients, there might be other criteria in certain cases.

If we consider Theorem 1.6.4, we obtain an explicit way to compute the canonical frame coefficients by using the pseudoinverse of the pre-frame operator:

**2.1.7 Theorem** *Let  $\{g_i\}_{i \in I}$  be a frame for  $\mathcal{H}$  with synthesis operator  $D$  and frame operator  $S$ . Then*

$$D^\dagger f = \{\langle f, S^{-1}g_i \rangle_{\mathcal{H}}\}_{i \in I} . \quad (2.8)$$

In operator terms, (2.8) says that

$$D^\dagger = D^* S^{-1} = D^* (DD^*)^{-1} ,$$

what is valid more generally for the pseudoinverse of any surjective operator  $D$ . This also emphasizes the difference between  $D^\dagger$  and the analysis operator  $D^*$ : For reconstruction, one needs to *compute* either  $S^{-1}$  or directly  $D^\dagger$ .

Similar to Bessel sequences, the explicit knowledge of frame bounds  $A$  and  $B$  again provides some kind of “quality criteria”. The lowest upper frame bound and highest



lower frame bound are called the optimal frame bounds. One gets  $B_{\text{opt}} = \sup_{\|f\|=1} \langle Sf, f \rangle = \|S\|$  and similarly  $A_{\text{opt}} = 1/\|S^{-1}\|$ . Since  $S$  is a self-adjoint, positive and bounded linear operator, these values are equal to its largest and smallest eigenvalues. The ratio between the largest and smallest eigenvalue of an operator or matrix is called its condition number. Given a frame, the condition number therefore equals  $B_{\text{opt}}/A_{\text{opt}} = \|S\| \|S^{-1}\|$ . The condition number is a measure for the speed of convergence of numerical algorithms. If only bad estimates for the optimal frame bounds are known and  $B$  is much larger than  $A$ , one gets a high condition number, resulting in a slow numerical convergence. Therefore, some of the various known acceleration methods from numerical analysis are applied in practice to speed up convergence.

The importance of tight frames lies within the fact that the frame operator is then just a multiple of the identity, i.e.,  $S = A \text{Id}_{\mathcal{H}}$ , what follows from (2.7). The same is valid for its inverse  $S^{-1}$ , and so the canonical dual frame is just  $\{\frac{1}{A}g_i\}_{i \in I}$ , yielding that the dual frames equal the original frame up to a constant, avoiding the complicated structure of the frame operator as well as its inversion. Another important thing is that in general the duals might not be of the same structure as the original frame due to the application of  $S^{-1}$ ; e.g., a dual of a frame in  $L^2$  might not be in  $L^2$  anymore. Tight frames therefore ensure that the duals keep this structure.

However, for arbitrary frames there is a way of getting a tight frame: Because both  $S$  and  $S^{-1}$  are self-adjoint and positive, their square-roots  $S^{1/2}$  and  $S^{-1/2}$  exist and are again self-adjoint and positive. As  $S^{-1/2} S S^{-1/2} = \text{Id}_{\mathcal{H}}$ , we get that

$$f = S^{-1/2} S (S^{-1/2} f) = \sum_{i \in I} \langle f, S^{-1/2} g_i \rangle_{\mathcal{H}} S^{-1/2} g_i,$$

yielding that  $\{S^{-1/2} g_i\}_{i \in I}$  is the canonical tight frame with frame bounds  $A = B = 1$ . As the elements aren't normalized in general, these need not be an orthonormal basis.

The question when the expansion coefficients are uniquely determined has its own interest. This is the case of a Riesz basis where the synthesis operator is bounded invertible:

**2.1.8 Definition** A family  $\{g_i\}_{i \in I}$  in a Hilbert space  $\mathcal{H}$  is a Riesz basis for  $\mathcal{H}$  if there exist constants  $A, B > 0$  such that the inequalities

$$A \|c\|_2^2 \leq \|Dc\|_{\mathcal{H}}^2 \leq B \|c\|_2^2$$

or, in more detail,

$$A \sum_{i \in I} |c_i|^2 \leq \left\| \sum_{i \in I} c_i g_i \right\|_{\mathcal{H}}^2 \leq B \sum_{i \in I} |c_i|^2$$

hold for all finite sequences  $c = \{c_i\}_{i \in I}$ .

There are some equivalent conditions for a frame to be a Riesz basis. We give a list in the following statement.

**2.1.9 Proposition** Suppose  $\{g_i\}_{i \in I}$  is a frame for  $\mathcal{H}$ , then the following conditions are equivalent:

- (a)  $\{g_i\}_{i \in I}$  is a Riesz basis for  $\mathcal{H}$ .
- (b)  $\{g_i\}_{i \in I}$  is a basis for  $\mathcal{H}$ .
- (c) The operators  $C$  and  $D$  are bijections.
- (d) The coefficients  $c \in \ell^2(I)$  in the series expansion (2.6) are unique.
- (e) The analysis operator  $C$  has closed and dense range in  $\ell^2(I)$ .
- (f)  $\{g_i\}_{i \in I}$  is the image of an orthonormal basis  $\{e_i\}_{i \in I}$  for  $\mathcal{H}$  under a bounded bijective operator  $U: \mathcal{H} \rightarrow \mathcal{H}$ , i.e.,  $g_i = Ue_i \forall i \in I$ .
- (g)  $\{g_i\}_{i \in I}$  is an exact frame for  $\mathcal{H}$ , i.e., it is no longer a frame if a single element is removed.

**Proof.** See [Grö01, 5.1.5] and [Chr03, 6.1.1]. □

It follows that the unique dual Riesz basis is given by  $\{S^{-1}g_i\}_{i \in I}$ . Some sources use condition (f) of Proposition 2.1.9 as an alternative definition for Riesz bases.

We already mentioned that a frame is complete in  $\mathcal{H}$ . If we compare this to the fact that a frame could be inexact, i.e. it could still be a complete sequence although elements could be removed, this also means that there might be more elements in a frame than needed to span all of  $\mathcal{H}$ . In this case a sequence is said to be *overcomplete*. In some sense we expect the elements to be dependent on each other, in analogy to linear dependence. In the question about the uniqueness of the expansion coefficients, the need for  $\{g_i\}_{i \in I}$  to be independent in some sense raises the question about the analogy to linear (in-)dependence of elements in infinite-dimensional Hilbert spaces. The following concepts of independence exist in infinite-dimensional Banach spaces:

**2.1.10 Definition** Be  $\{g_i\}_{i=1}^{\infty}$  a sequence in  $X$ . One says that

- (a)  $\{g_i\}_{i=1}^{\infty}$  is linearly independent if every finite subsequence of  $\{g_i\}_{i=1}^{\infty}$  is linearly independent (in the usual way).
- (b) If whenever  $\exists \{c_i\}_{i=1}^{\infty} \sum_{i=1}^{\infty} c_i g_i = 0$  implies that  $c_i = 0 \forall i \in \mathbb{N}$ , then  $\{g_i\}_{i \in I}$  is called  $\omega$ -independent.
- (c)  $\{g_i\}_{i=1}^{\infty}$  is minimal if  $g_j \notin \overline{\text{span}\{g_i\}_{i \neq j}} \forall j \in \mathbb{N}$ , i.e., if every single element is not in the span of the others.

The relation between these different flavors of independence is that minimality implies  $\omega$ -independence, what in turn implies linear independence [Chr03, 3.1.3]. As a final result, we state that  $\{g_i\}_{i \in I}$  being a Riesz basis is equivalent to  $\{g_i\}_{i \in I}$  being  $\omega$ -independent [Chr03, 6.1.1]. So, linear independence alone is not enough.

Considering the situation in the finite-dimensional setting, one could expect that whenever a sequence  $\{g_i\}_{i \in I}$  is complete in  $\mathcal{H}$ , this would imply that all  $f \in \mathcal{H}$  have an expansion  $f = \sum_{i \in I} c_i g_i$  for some coefficients  $\{c_i\}_{i \in I}$ . However, for an infinite-dimensional Hilbert space this is generally not true. A counter-example is the

sequence  $\{e_i + e_{i+1}\}_{i \in I}$  built from an orthonormal basis  $\{e_i\}_{i \in I}$  for  $\mathcal{H}$ , cf. [Chr03, 5.4.6].

Luckily, all this is no concern for the finite-dimensional discrete case of applied signal analysis.

## 2.2 Gabor Frames in $L^2(\mathbb{R}^d)$

If we go back to Section 1.4 and remember formula (1.18), the STFT analyzes a function  $f \in L^2(\mathbb{R}^d)$  into coefficients  $\langle f, M_\omega T_x g \rangle_{L^2}$  using modulations and translations of a single window function  $g \in L^2(\mathbb{R}^d) \setminus \{0\}$ . One problem we noticed was that these TF-shifts are infinitesimal and overlap largely, making the STFT a highly redundant time-frequency representation. The idea is to replace this infinitesimality by discrete choices of time-positions  $x$  and frequencies  $\omega$  such that this redundancy is decreased while leaving enough information in the coefficients about the time-frequency behavior of  $f$ . This is the very essence of Gabor analysis: It is sought to expand functions in  $L^2(\mathbb{R}^d)$  into an absolutely convergent series of modulations and translations of a window function  $g$ , i.e., we are looking for necessary and sufficient conditions for

$$\{g_{x,\omega}\}_{(x,\omega) \in \Lambda} := \{M_\omega T_x g\}_{(x,\omega) \in \Lambda}$$

to be a frame for a certain discrete subset  $\Lambda \subseteq \mathbb{R}^d \times \mathbb{R}^d$ . The question rises about how this sampling set  $\Lambda$  should be structured. We will notice that it is very convenient to have this set closed under the addition operation, urging  $\Lambda$  to be a subgroup of the time-frequency plane, i.e.,  $\Lambda \trianglelefteq \mathbb{R}^d \times \mathbb{R}^d$ . Dennis Gabor<sup>1</sup> suggested 1946 in his *Theory of Communication* [Gab46] to use fixed step-sizes  $\alpha, \beta > 0$  for time and frequency and use the set  $\{\alpha k\}_{k \in \mathbb{Z}^d}$  for the time-positions and  $\{\beta n\}_{n \in \mathbb{Z}^d}$  for the frequencies, yielding the functions

$$g_{k,n}(x) := M_{\beta n} T_{\alpha k} g(x) = e^{2\pi i \beta n \cdot x} g(x - \alpha k)$$

as analyzing elements. This is the approach that is usually presented in the literature, although there is also a more general group-theoretical setting possible where  $\Lambda$  is an arbitrary (discrete) subgroup. This subgroup is also called a *time-frequency lattice*, although it doesn't have to be of such a "rectangular" shape in general.

**2.2.1 Definition** A lattice  $\Lambda \subseteq \mathbb{R}^d$  is a (discrete) subgroup of  $\mathbb{R}^d$  of the form  $\Lambda = \mathfrak{A}\mathbb{Z}^d$ , where  $\mathfrak{A}$  is an invertible  $d \times d$ -matrix over  $\mathbb{R}$  that is not necessarily unique. Lattices in  $\mathbb{R}^{2d}$  can be described as

$$\Lambda = \{(x, y) \in \mathbb{R}^{2d} \mid (x, y) = (Ak + B\ell, Ck + D\ell), (k, \ell) \in \mathbb{Z}^{2d}\}$$

with  $A, B, C, D \in \mathbb{C}^{d \times d}$  and

$$\mathfrak{A} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

A lattice  $\Lambda = \alpha\mathbb{Z}^d \times \beta\mathbb{Z}^d \trianglelefteq \mathbb{R}^{2d}$  for  $\alpha, \beta > 0$  is called a separable or product lattice or a grid.

---

<sup>1</sup>Actually *Dénes Gábor*

In the following, our lattice will be of the separable type for fixed lattice parameters  $\alpha, \beta > 0$ . We will discuss other lattices in Chapter 3.

**2.2.2 Definition** For a non-zero window function  $g \in L^2(\mathbb{R}^d)$  and lattice parameters  $\alpha, \beta > 0$ , the set of time-frequency shifts

$$\mathcal{G}(g, \alpha, \beta) := \{M_{\beta n} T_{\alpha k} g\}_{k, n \in \mathbb{Z}^d}$$

is called a **Gabor system**. If  $\mathcal{G}(g, \alpha, \beta)$  is a frame for  $L^2(\mathbb{R}^d)$ , it is called a **Gabor frame** or **Weyl–Heisenberg frame**. The associated frame operator is the **Gabor frame operator** and takes the form

$$\begin{aligned} Sf &= \sum_{k, n \in \mathbb{Z}^d} \sum_{k, n \in \mathbb{Z}^d} \langle f, M_{\beta n} T_{\alpha k} g \rangle_{L^2} M_{\beta n} T_{\alpha k} g \\ &= \sum_{k, n \in \mathbb{Z}^d} \sum_{k, n \in \mathbb{Z}^d} \mathcal{V}_g f(\alpha k, \beta n) M_{\beta n} T_{\alpha k} g \end{aligned} \quad (2.9)$$

for all  $f \in L^2(\mathbb{R}^d)$ . The window  $g$  is also called the **Gabor atom**.

If we remember the commutation relations (1.10) that modulation and translation do not commute, we luckily get as a result to (2.9) that the order of translation and modulation is *not* important since the phase factor in  $T_x M_\omega = e^{-2\pi i x \cdot \omega} M_\omega T_x$  occurs both linearly and conjugate-linearly in (2.9) because the inner product is sesquilinear, and the factors thus cancel. So, the frame expansion could as well be written as

$$Sf = \sum_{k, n \in \mathbb{Z}^d} \sum_{k, n \in \mathbb{Z}^d} \langle f, T_{\alpha k} M_{\beta n} g \rangle_{L^2} T_{\alpha k} M_{\beta n} g \quad (2.10)$$

and  $\mathcal{G}(g, \alpha, \beta) = \{T_{\alpha k} M_{\beta n} g\}_{k, n \in \mathbb{Z}^d}$ . This justifies using a joint notation for TF-shifts, what we will be doing later.

Now we ask how a dual frame could look like such that we can expand  $f \in L^2(\mathbb{R}^d)$  with respect to that Gabor frame. In the previous section we showed that  $\{S^{-1} g_{k, n}\}_{k, n \in \mathbb{Z}^d}$  yields the canonical dual frame. So we'd have to compute  $S^{-1}$  and apply it to *all* modulated and translated versions of the Gabor atom  $g$ . Luckily, there is an observation that reduces this computational cost: The frame operator commutes with TF-shifts, what follows from the following computation. For arbitrary fixed indices  $\ell, m \in \mathbb{Z}^d$ , we get

$$\begin{aligned} (M_{\beta m} T_{\alpha \ell})^{-1} S M_{\beta m} T_{\alpha \ell} f &= \sum_{k, n \in \mathbb{Z}^d} \sum_{k, n \in \mathbb{Z}^d} \langle M_{\beta m} T_{\alpha \ell} f, M_{\beta n} T_{\alpha k} g \rangle_{L^2} (M_{\beta m} T_{\alpha \ell})^{-1} M_{\beta n} T_{\alpha k} g \\ &= \sum_{k, n \in \mathbb{Z}^d} \sum_{k, n \in \mathbb{Z}^d} \langle f, M_{\beta(n-m)} T_{\alpha(k-\ell)} g \rangle_{L^2} M_{\beta(n-m)} T_{\alpha(k-\ell)} g \end{aligned} \quad (2.11)$$

after cancelling the occurring commutation factors  $e^{\pm 2\pi i \alpha \beta \ell \cdot (n-m)}$ . Comparing (2.11) with (2.9) and considering  $k - \ell$  and  $n - m$  as new indices, we finally get

$$(M_{\beta m} T_{\alpha \ell})^{-1} S M_{\beta m} T_{\alpha \ell} f = Sf$$

what corresponds to

$$SM_{\beta m}T_{\alpha \ell} = M_{\beta m}T_{\alpha \ell}S, \quad (2.12)$$

the commutation of  $S$  with the time-frequency shifts determined by  $\Lambda = \alpha\mathbb{Z}^d \times \beta\mathbb{Z}^d$ .

Consequently, also  $S^{-1}$  and  $S^{-1/2}$  commute with time-frequency shifts, what can be seen by applying them to the left and right of (2.12). We get very important benefits from that fact, explaining why the case of Gabor frames is so interesting.

**2.2.3 Theorem** *If the given Gabor system  $\mathcal{G}(g, \alpha, \beta)$  is a frame for  $L^2(\mathbb{R}^d)$ , then all of the following hold:*

- (a) *There exists a dual window  $\gamma \in L^2(\mathbb{R}^d)$  such that the dual frame is given by the Gabor frame  $\mathcal{G}(\gamma, \alpha, \beta)$ .*
- (b) *Every  $f \in L^2(\mathbb{R}^d)$  has an expansion of the form*

$$\begin{aligned} f &= D_\gamma C_g f = \sum_{k,n \in \mathbb{Z}^d} \langle f, M_{\beta n} T_{\alpha k} g \rangle_{L^2} M_{\beta n} T_{\alpha k} \gamma \\ &= D_g C_\gamma f = \sum_{k,n \in \mathbb{Z}^d} \langle f, M_{\beta n} T_{\alpha k} \gamma \rangle_{L^2} M_{\beta n} T_{\alpha k} g \end{aligned} \quad (2.13)$$

*with unconditional convergence in  $L^2(\mathbb{R}^d)$ .*

- (c) *The canonical dual frame is given by the Gabor frame  $\{M_{\beta n} T_{\alpha k} S^{-1} g\}_{k,n \in \mathbb{Z}^d}$  built from the canonical dual window  $\gamma^\circ := S^{-1} g$ .*
- (d) *The canonical tight frame is given by the Gabor frame  $\{M_{\beta n} T_{\alpha k} S^{-1/2} g\}_{k,n \in \mathbb{Z}^d}$  built from the canonical tight window  $S^{-1/2} g$ .*
- (e) *The inverse frame operator  $S^{-1}$  is just the frame operator for the Gabor system  $\mathcal{G}(\gamma^\circ, \alpha, \beta)$  and*

$$S^{-1} f = \sum_{k,n \in \mathbb{Z}^d} \langle f, M_{\beta n} T_{\alpha k} \gamma^\circ \rangle_{L^2} M_{\beta n} T_{\alpha k} \gamma^\circ. \quad (2.14)$$

Notice the affinity between (2.13) and the inversion of the STFT (1.22).

So, instead of calculating the “multi-infinite” family  $\{S^{-1} M_{\beta n} T_{\alpha k} g\}_{k,n \in \mathbb{Z}^d}$ , it is enough to compute  $S^{-1} g$  once and use the same set of TF-shifts to obtain the desired frame expansion.

And here’s the reason why one wants to have a subgroup  $\Lambda \trianglelefteq \mathbb{R}^d \times \mathbb{R}^d$  as sampling subset of the TF-plane: The series (2.11) could not be seen as the expansion of the frame operator if the TF-shifts  $M_{\beta(n-m)} T_{\alpha(k-l)} g$  were not part of the Gabor frame. So, as soon as  $\Lambda$  is a subgroup and  $\mathcal{G}(g, \Lambda) := \{M_\omega T_x g\}_{(x,\omega) \in \Lambda}$  is a frame, the frame operator commutes with the corresponding TF-shifts and the dual frame is built from a single dual window using the same set of TF-shifts.

However,  $\mathcal{G}(g, \Lambda)$  might still be a frame for an arbitrary countable subset  $\Lambda \subseteq \mathbb{R}^d \times \mathbb{R}^d$  and there still is a dual frame according to the general theory, but both

the elements and the frame operator lack structure, and the inversion of the frame operator is more difficult and costlier than in (2.14).

Thinking of an orthonormal basis, it is a tight frame with bounds  $A = B = 1$  and condition number  $B/A = 1$ , granting it a very stable expansion property. A non-tight frame might have larger condition numbers and therefore not such a good stability of the frame expansion, but still provide some stability by its overcompleteness. However, one could try to reduce the computational effort by removing some of the elements in  $\mathcal{G}(g, \Lambda)$  if the frame stays complete, but there is a good reason not to try this: By removing some TF-shifts, the subgroup structure of  $\Lambda$  will be lost, the Gabor frame operator no longer commutes with TF-shifts and the dual frame will be more difficult to find.

So far we only know that the canonical dual Gabor frame is a dual frame built from a single window, but have no idea yet whether there are other dual frames that have the Gabor structure. Indeed, one can show [Grö01, 7.6.1] that all dual windows  $\gamma$  are within an affine subspace of  $L^2(\mathbb{R}^d)$ , namely  $\gamma \in \gamma^\circ + \mathcal{K}^\perp$ , where  $\mathcal{K}$  is the closed linear span of  $\mathcal{G}(g, \frac{1}{\beta}, \frac{1}{\alpha})$  and therefore

$$\mathcal{K}^\perp = \{h \in L^2(\mathbb{R}^d) : \langle h, M_{n/\alpha} T_{k/\beta} g \rangle_{L^2} = 0 \quad \forall k, n \in \mathbb{Z}^d\}. \quad (2.15)$$

Hence we have  $\gamma = \gamma^\circ + h$  for a certain  $h \in \mathcal{K}^\perp$ , and as  $\gamma^\circ \in \mathcal{K}$ , the canonical dual window possesses the smallest  $L^2$ -norm among all dual windows and is most similar to the original window  $g$ . However, there might be reasons not to choose the canonical dual window, but one of the others in  $\gamma^\circ + \mathcal{K}^\perp$ , if, for example, one wants the dual window to have a smaller essential support, or if the window should be as smooth as possible. The variety of dual windows is determined by  $\dim \mathcal{K}^\perp$ .

Using a separable lattice with parameters  $\alpha, \beta > 0$  has some advantages over using general discrete subgroups. First, the Gabor frame operator possesses a special structure and has some numerically efficient representations, involving relations to the used lattice parameters. The so-called *adjoint lattice*  $\Lambda^\circ = \frac{1}{\beta} \mathbb{Z}^d \times \frac{1}{\alpha} \mathbb{Z}^d$  occurring in (2.15) plays an important role for this, and some results will be mentioned in Section 3.3. Second, some conditions for a Gabor system to be a frame or not also rely on the lattice parameters. Finally, for certain Gabor atoms  $g$  there are known conditions on  $\alpha$  and  $\beta$  to make  $\mathcal{G}(g, \alpha, \beta)$  a frame.

**2.2.4 Theorem** *Be  $g \in L^2(\mathbb{R}^d) \setminus \{0\}$  and  $\alpha, \beta > 0$ . If  $\mathcal{G}(g, \alpha, \beta)$  is a frame, then all of the following hold:*

- (a)  $\alpha\beta \leq 1$ .
- (b)  $\mathcal{G}(g, \alpha, \beta)$  is a Riesz basis if and only if  $\alpha\beta = 1$ .

*Proof.* See [Grö01, 7.5.1+2] or [Chr03, 8.3.1]. □

So whenever  $\alpha\beta > 1$  one can for no window manage to build a Gabor frame. Unfortunately, having  $\alpha\beta \leq 1$  is still not sufficient. Sufficient conditions are presented e.g. in [Chr03, 8.4].

However, for certain classes of windows things get easier. A special result is known for the Gaussian function that we introduced in Section 1.5.

**2.2.5 Theorem** *Be  $\varphi(x) := 2^{d/4}e^{-\pi x^2}$  the normalized Gaussian such that  $\|\varphi\|_{L^2} = 1$ . Then  $\mathcal{G}(\varphi, \alpha, \beta)$  is a frame for  $L^2(\mathbb{R}^d)$  if and only if  $\alpha\beta < 1$ .*

It is customary in signal analysis to call the case

- $\alpha\beta < 1$  *oversampling*,
- $\alpha\beta = 1$  *critical sampling*, and
- $\alpha\beta > 1$  *undersampling*.

In the case of the Gaussian window, oversampling guarantees an excellent time-frequency localization. But for Gabor frame theory in Hilbert spaces it is quite delicate to find appropriate windows for given  $\alpha\beta \leq 1$ . One of the reasons is due to the uncertainty principle 1.4.2, disallowing functions to have small essential support on the time-frequency plane. The version commonly used in time-frequency analysis is the Balian–Low Theorem, here presented in a negated form; we again write  $Xf$  for the function  $x \mapsto xf(x)$ .

**2.2.6 Theorem (Balian–Low)** *Be  $g \in L^2(\mathbb{R}^d)$  a non-zero window and  $\alpha, \beta > 0$  with  $\alpha\beta = 1$ . If*

$$\|Xg\|_{L^2}\|X\hat{g}\|_{L^2} < \infty,$$

*then  $\mathcal{G}(g, \alpha, \beta)$  cannot constitute a frame or a Riesz basis.*

Therefore, dedicated window classes are introduced for the purpose of Gabor analysis in this general setting, with conditions that combine applicable local and global properties of window functions. Among these window classes are the Wiener space ([Grö01, 6.1], [Chr03, 8.5]) or the modulation spaces ([FZ98], [Grö01, 11+12]).

In the signal processing literature, especially when it comes to image processing, there’s sometimes the term of “Gabor functions” used synonymously for modulated Gaussian functions, because that was the window Gabor originally used in [Gab46]. But Gabor analysis is in no way restricted to using the Gaussian function as analyzing prototype, although it has some advantages that we already mentioned. Another justification is that modulated Gaussians can be used to describe biological vision, see the references in Section 5.6.

## 2.3 Gabor Frames in $\ell^2(\mathbb{Z})$

If we think of signals such as sound, we think of them as continuous waves. Indeed, the technology for signal processing originally was of the continuous-time analog type before digital computers came into our everyday life. Nowadays digital signal processing is used almost exclusively, forcing us to change our function model to a time-discrete one. In both the time-continuous and time-discrete case, it makes sense to consider functions of finite energy only and to equip them with an inner product. It is therefore natural to switch from  $L^2(\mathbb{R})$  to  $\ell^2(\mathbb{Z})$ , where functions are reduced to *samples* of their function values, i.e., we could have  $f(j) := F(\alpha j)$  for  $F \in L^2(\mathbb{R})$  and

$$f: \mathbb{Z} \rightarrow \mathbb{C}, \quad \|f\|_2^2 = \sum_{j=-\infty}^{\infty} |f(j)|^2 < \infty.$$

But in general the sequence elements don't have to be samples of a continuous function, i.e. discrete signals  $f \in \ell^2(\mathbb{Z})$  as such are simply *not defined* between the integer positions.

Besides the consideration of certain conditions in the case of sampling, cf. [Grö01, 10.2], Gabor frame theory in  $\ell^2(\mathbb{Z})$  is very similar to that in  $L^2(\mathbb{R})$  and will therefore only be discussed briefly in this section. The main differences concern the time shifts and frequency shifts. Time shifts are given as multiples of integer translates, because one cannot shift by “less than one position” or by fractions of positions, i.e.,

$$T_k f(j) = f(j - k) \quad (2.16)$$

for  $k \in \mathbb{Z}$  and  $f \in \ell^2(\mathbb{Z})$ . A shift parameter  $\alpha > 0$  for Gabor frames in  $\ell^2(\mathbb{Z})$  can only be given as  $\alpha = N \in \mathbb{N}$ .

Almost the same is valid for the modulations, where the modulation parameter  $\beta > 0$  is given as  $1/L$  for fixed  $L \in \mathbb{N}$  and therefore

$$M_{\ell/L} f(j) = e^{2\pi i j \ell/L} f(j) \quad (2.17)$$

for  $\ell \in \mathbb{Z}$ . It turns out that the modulations are now periodic with period  $L$ , i.e.

$$M_{\frac{\ell+nL}{L}} = M_{\frac{\ell}{L}+n} = M_{\frac{\ell}{L}} \quad \forall n \in \mathbb{Z},$$

yielding that actually only the modulations  $M_0, \dots, M_{\frac{L-1}{L}}$  are to be considered, because otherwise a sequence  $\{M_{\ell/L} g\}_{\ell \in \mathbb{Z}}$  could never be a Bessel sequence for  $g \neq 0$ .

The discrete Gabor system generated by the atomic sequence  $g \in \ell^2(\mathbb{Z})$  and shift parameters  $N$  and  $1/L$  is now the family of sequences  $\{g_{k,\ell}\}_{k \in \mathbb{Z}, \ell \in \langle L \rangle}$  where the sequences  $g_{k,\ell} \in \ell^2(\mathbb{Z})$  are defined by

$$g_{k,\ell}(j) := M_{\ell/L} T_{kN} g(j) = e^{2\pi i j \ell/L} g(j - kN).$$

Like previously, if the system satisfies the frame inequality, it is a Gabor frame for  $\ell^2(\mathbb{Z})$ , and the dual frame is again a Gabor frame built from a dual window  $\gamma \in \ell^2(\mathbb{Z})$ . The frame expansion for any  $f \in \ell^2(\mathbb{Z})$  now takes the form

$$f = \sum_{k=-\infty}^{\infty} \sum_{\ell=0}^{L-1} \langle f, M_{\ell/L} T_{kN} \gamma \rangle_2 M_{\ell/L} T_{kN} g.$$

Many results and conditions for Gabor systems in  $\ell^2(\mathbb{Z})$  can *mutatis mutandis* be taken over from  $L^2(\mathbb{R})$ , e.g., a necessary condition for the mentioned Gabor system to be a frame for  $\ell^2(\mathbb{Z})$  is that  $\alpha\beta = \frac{N}{L} \leq 1$ ; and if the Gabor system is a frame, it is a Riesz basis if and only if  $N = L$ .

The step from  $L^2(\mathbb{R})$  to  $\ell^2(\mathbb{Z})$  is the first one when trying to apply the results so far achieved for computational implementations. However, infinite sequences or infinite sums will not make sense when one is interested to achieve results in finite time. In reality, only finitely many elements can be considered, only vectors of finite length can be handled and only finite sums can be computed in finite time. Therefore, we will now take the next step to signals of finite length.



# Chapter 3

## Finite Discrete Gabor Analysis

### 3.1 Finite Discrete Periodic Signals

In the last section of the previous chapter we were creating a time-discrete signal model. We mentioned that for implementing signal analysis on digital computers only *finite* sequences  $f \in \ell^2(\mathbb{Z})$  can be considered. If the length of that sequence is given as  $L \in \mathbb{N}$ , we can interpret this as if  $f \in \mathbb{C}^L$  and write  $f = (f(0), \dots, f(L-1))$ , meaning that it's no restriction to assume the domain of  $f$  to be  $\langle L \rangle \subseteq \mathbb{Z}$ . This makes it possible to connect with the results of sequences in  $\ell^2(\mathbb{Z})$ .

Whereas the discrete modulation  $M_{\ell/L}$  defined in (2.17) can still be applied to the finite case, the translation  $T_k$  defined in (2.16) does not make sense for  $\mathbb{C}^L$  unless  $j - k \in \langle L \rangle$ . The natural solution lies in a periodic extension of the finite sequences such that

$$f(j + nL) := f(j) \quad \forall n \in \mathbb{Z}, j \in \langle L \rangle.$$

Therefore the domain where  $f$  is defined is not just the set  $\langle L \rangle$ , but rather the finite Abelian group  $\mathcal{G} = \mathbb{Z}_L := \mathbb{Z}/L\mathbb{Z}$  which is closed under the addition operation. The finite sequences therefore live in  $\ell^2(\mathbb{Z}_L) \cong \mathbb{C}^L$ , and we use both spaces synonymously.

It also becomes possible to carry the `discrete convolution` of  $f, g \in \ell^2(\mathbb{Z})$ , defined as

$$(f * g)(j) := \sum_{k \in \mathbb{Z}} f(k) g(j - k),$$

over to functions  $f, g \in \ell^2(\mathbb{Z}_L)$ , because  $g(j - k)$  is now always defined.

The `discrete Fourier transform` (DFT) of  $f \in \mathbb{C}^L$  is defined as

$$\hat{f}(j) := (\mathcal{F}f)(j) := \sum_{k=0}^{L-1} f(k) e^{-2\pi i j k / L}, \quad j \in \mathbb{Z}_L, \quad (3.1)$$

what is up to a constant a unitary mapping  $\mathbb{C}^L \rightarrow \mathbb{C}^L$ . The inverse is given as

$$(\mathcal{F}^{-1}f)(j) := \frac{1}{L} \sum_{k=0}^{L-1} f(k) e^{2\pi i j k / L}, \quad j \in \mathbb{Z}_L. \quad (3.2)$$

To define the DFT as the unitary mapping  $\mathbb{C}^L \rightarrow \mathbb{C}^L$ , it has to be equipped with the prefactor  $\frac{1}{\sqrt{L}}$ . But this is not consistent with the implementation in MATLAB or

Octave, where it behaves like defined above. A well-known and very efficient implementation of the DFT is the *Fast Fourier Transform* (FFT).

From now on we abbreviate  $M_\ell := M_{\ell/L}$  for modulations in  $\mathbb{C}^L$ . The discrete STFT of  $f \in \mathbb{C}^L$  with respect to the discrete window  $g \in \mathbb{C}^L$  is given as

$$(\mathcal{V}_g f)(k, \ell) = \langle f, M_\ell T_k g \rangle_{\mathbb{C}^L} .$$

The actions of time- and frequency shifts are in more detail given as

$$T_k f = T_k(f(0), \dots, f(L-1)) = (f(k), \dots, \underbrace{f(L-1)}_{L-k}, \underbrace{f(0)}_{L-k+1}, \dots, f(k-1))$$

and

$$\begin{aligned} M_\ell f &= M_\ell(f(0), \dots, f(L-1)) = \\ &= (f(0), e^{2\pi i \ell/L} f(1), e^{2\pi i 2\ell/L} f(2), \dots, e^{2\pi i (L-1)\ell/L} f(L-1)) . \end{aligned}$$

The actions of the TF-shifts can be described as matrices that operate on the vector  $f = (f(0), \dots, f(L-1))^T$ . The time-shift matrix  $T_k$  is given as the permutation matrix with ones on the (periodized)  $k$ -th subdiagonal, whereas the modulation matrix has its exponential entries positioned at the main diagonal. It is therefore again clear that the composition of TF-shifts does not commute. In fact, one gets

$$T_k M_\ell = e^{2\pi i k \ell/L} M_\ell T_k , \quad k, \ell \in \mathbb{Z}_L$$

after a simple computation. To get a joint notation for TF-shifts, we write

$$\pi(\lambda) := \pi(k, \ell) := M_\ell T_k \quad \text{with} \quad \lambda = (k, \ell) \in \mathbb{Z}_L \times \mathbb{Z}_L ,$$

where  $\mathbb{Z}_L \times \mathbb{Z}_L$  is the discrete time-frequency plane<sup>1</sup>. From the commutation relations one easily derives for  $\lambda = (r, m)$  and  $\mu = (s, n)$  that

$$\pi(\lambda) \pi(\mu) = \pi(\lambda + \mu) e^{2\pi i r n/L} \tag{3.3}$$

$$= \pi(\mu) \pi(\lambda) e^{2\pi i (r n - s m)/L} . \tag{3.4}$$

## 3.2 Frames and Gabor Frames in $\mathbb{C}^L$

The definitions and results from the previous chapter can be carried over to the finite discrete case without major problems. One thing that emerges is that things become more related to terms of linear algebra. The conditions for the finite sequence  $\{g_0, \dots, g_{N-1}\}$  of elements  $g_j \in \mathbb{C}^L$  to be a frame for the finite-dimensional Hilbert space  $\mathbb{C}^L$  are that there exist  $A, B > 0$  such that

$$A \sum_{k=0}^{L-1} |f(k)|^2 \leq \sum_{j=0}^{N-1} |\langle f, g_j \rangle_{\mathbb{C}^L}|^2 \leq B \sum_{k=0}^{L-1} |f(k)|^2 \quad \forall f \in \mathbb{C}^L$$

<sup>1</sup>The TF-plane is often written as  $\mathbb{Z}_L \times \widehat{\mathbb{Z}_L}$  due to the group theoretical syntax  $\mathcal{G} \times \widehat{\mathcal{G}}$ , where  $\widehat{\mathcal{G}}$  is the so-called *character group* of  $\mathcal{G}$ , cf. [Grö98].

or

$$A\|f\|_2^2 \leq \|Cf\|_2^2 \leq B\|f\|_2^2 \quad \forall f \in \mathbb{C}^L,$$

where  $C$  is the analysis operator. It is obvious that the sequence  $\{g_j\}_{j \in \langle N \rangle}$  has to span all of  $\mathbb{C}^L$ , i.e.,  $\overline{\text{span}}\{g_j\}_{j \in \langle N \rangle} = \mathbb{C}^L$ , because otherwise there would be a non-zero  $f \in \overline{\text{span}}\{g_j\}_{j \in \langle N \rangle}^\perp$  such that  $\langle f, g_j \rangle = 0$  for all  $j \in \langle N \rangle$ , what contradicts the frame inequality. Thus we must have  $N \geq L$  elements to get a frame in the Hilbert space with dimension  $L$ . The same is valid the other way round: Every set with more (or equal) than  $L$  elements that spans  $\mathbb{C}^L$  is a frame for  $\mathbb{C}^L$ .

The action of the linear analysis operator  $C$  on the vector  $f$  is given as the vector  $Cf = (\langle f, g_j \rangle)_{j \in \langle N \rangle}$ , indicating that its  $j$ -th entry is

$$(Cf)_j = \langle f, g_j \rangle = \sum_{k=0}^{L-1} f(k) \overline{g_j(k)} = g_j^* f,$$

where  $g^* = \bar{g}^T$ . Therefore the matrix form of  $C \in \mathbb{C}^{N \times L}$  is

$$C = \begin{pmatrix} g_0^* \\ \vdots \\ g_{N-1}^* \end{pmatrix} = \begin{pmatrix} \overline{g_0(0)} & \cdots & \overline{g_0(L-1)} \\ \vdots & \vdots & \vdots \\ \overline{g_{N-1}(0)} & \cdots & \overline{g_{N-1}(L-1)} \end{pmatrix}.$$

By this structure we get the equivalence that a family  $\{g_j\}_{j \in \langle N \rangle}$  is a frame if and only if the corresponding analysis operator  $C$  has full rank, and every matrix with full rank uniquely represents a frame. Therefore we can write  $C$  synonymously for the corresponding frame.

The frame operator  $S = C^*C$  becomes an  $L \times L$ -matrix that also has full rank, and it is therefore invertible. Its condition number equals the ratio between its largest and smallest singular value, what is equal to the ratio  $B/A$ .

If we translate the discrete frame expansion

$$f = C^*c = (g_0, \dots, g_{N-1}) \begin{pmatrix} c(0) \\ \vdots \\ c(N-1) \end{pmatrix} = \begin{pmatrix} \sum_{j=0}^{N-1} c(j) g_j(0) \\ \vdots \\ \sum_{j=0}^{N-1} c(j) g_j(L-1) \end{pmatrix}$$

for a given  $f \in \mathbb{C}^L$  and sought  $c \in \mathbb{C}^N$  into terms of linear algebra, we notice that we're seeking  $N$  unknowns  $c(0), \dots, c(N-1)$  by using  $L \leq N$  equations, what shows that the solution cannot be unique in general. Considering that

$$f = SS^{-1}f = C^*C(C^*C)^{-1}f,$$

we see that one solution for  $c$  could be given as

$$c = C(C^*C)^{-1}f = (C^*)^\dagger f$$

by computing the pseudoinverse of the synthesis operator  $C^*$ . This also provides the matrix form of the canonical dual frame that is given by

$$(S^{-1}g_0, \dots, S^{-1}g_{N-1})^* = (S^{-1}C^*)^* = CS^{-1} = (C^*)^\dagger.$$

Now that we have seen what frames in  $\mathbb{C}^L$  look like, we can proceed to the special case of Gabor frames. These are given as a sequence of TF-shifts of a single window function  $g \in \mathbb{C}^L$ , i.e., a Gabor frame for  $\mathbb{C}^L$  is a sequence  $\{g_\lambda\}_{\lambda \in \Lambda} := \{\pi(\lambda)g\}_{\lambda \in \Lambda}$  for a certain discrete subset  $\Lambda \subseteq \mathbb{Z}_L \times \mathbb{Z}_L$ . We write  $C_g$  for the Gabor analysis operator to indicate the dependence on  $g$  and use it synonymously for the Gabor frame itself. It is clear that it is necessary to have  $N \geq L$  elements to span all of  $\mathbb{C}^L$ , but this is of course not sufficient for validating a frame. The ratio between  $N$  and  $L$  is also called the *redundancy* of the frame,

$$\text{red}_C := \frac{N}{L},$$

indicating how much more elements are given to span all of  $\mathbb{C}^L$ .

An irregular Gabor frame might be given for arbitrary  $\Lambda \subseteq \mathbb{Z}_L \times \mathbb{Z}_L$ , but as soon as we have a subgroup  $\Lambda \trianglelefteq \mathbb{Z}_L \times \mathbb{Z}_L$ , the Gabor frame operator  $S_g = C_g^* C_g$  commutes with all TF-shifts  $\pi(\lambda)$  for  $\lambda \in \Lambda$ , what can be shown in a similar way like we did in Section 2.2. Therefore the dual frame is once again a Gabor frame, built by the same TF-shifts of a single dual window  $\gamma \in \mathbb{C}^L$ . The canonical dual frame consists of elements

$$S_g^{-1} \pi(\lambda) g = \pi(\lambda) S_g^{-1} g = \pi(\lambda) \gamma^\circ,$$

and the computation of the canonical dual window reduces to finding a solution for the linear equation

$$S_g \gamma^\circ = g. \quad (3.5)$$

Therefore, the discrete Gabor expansion of an  $f \in \mathbb{C}^L$  is given as

$$f = \sum_{\lambda \in \Lambda} \langle f, \pi(\lambda) g \rangle_{\mathbb{C}^L} \pi(\lambda) \gamma^\circ = \sum_{\lambda \in \Lambda} \langle f, \pi(\lambda) \gamma^\circ \rangle_{\mathbb{C}^L} \pi(\lambda) g,$$

where the Gabor coefficients reside in  $\ell^2(\Lambda) \cong \mathbb{C}^N$ .

A special case for a lattice is a so-called separable lattice  $\Lambda = \alpha \mathbb{Z}_L \times \beta \mathbb{Z}_L$  with  $\alpha, \beta \in \mathbb{N}$  being divisors of  $L$ . The elements of such a Gabor frame take the form

$$M_{\beta \ell} T_{\alpha k} g(j) = e^{2\pi i \beta \ell j / L} g(j - \alpha k)$$

with  $k \in \langle \frac{L}{\alpha} \rangle$  and  $\ell \in \langle \frac{L}{\beta} \rangle$ . The number of elements is  $N = \frac{L}{\alpha} \cdot \frac{L}{\beta} = \frac{L^2}{\alpha\beta}$ , and it is necessary to have  $\frac{L^2}{\alpha\beta} \geq L$  elements to have a frame. The oversampled case is therefore given for  $\alpha\beta < L$ , and the undersampled case for  $\alpha\beta > L$ . Critical sampling is given for  $\alpha\beta = L$ .

### 3.3 The Structure of the Gabor Frame Matrix

For any kind of lattice, the discrete (Gabor) frame expansion reads as

$$f = C_g^* c$$

for a certain  $c \in \mathbb{C}^N$ . A solution is given by

$$c = (C_g^*)^\dagger f,$$

what might take some numerical effort to compute. Luckily, for Gabor frames and  $\gamma^\circ = S_g^{-1}g$  we get

$$\pi(\lambda)\gamma^\circ = \pi(\lambda)S_g^{-1}g = S_g^{-1}\pi(\lambda)g ,$$

what reads in matrix notation as

$$C_{\gamma^\circ} = S_g^{-1}C_g = (C_g^\dagger)^* .$$

Hence, for Gabor frames it follows that

$$C_g^\dagger = C_{\gamma^\circ}^*$$

and

$$S_g^{-1} = S_{\gamma^\circ} ,$$

indicating that it is enough to find  $\gamma^\circ$  by solving the linear equation (3.5).

This equation and the above derivations show that the frame matrix  $S_g$  is one of the most important objects in Gabor analysis. Studying this operator not only provides a deeper understanding of its structure, but also yields methods for fast numerical computations of the dual window.

### 3.3.1 The Walnut Representation

In the case of a separable lattice  $\Lambda = \alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$ , the Gabor frame matrix has a special structure. As the frame matrix is for any frame given as

$$S_g = C_g^* C_g = \sum_{\lambda \in \Lambda} g_\lambda g_\lambda^* \in \mathbb{C}^{L \times L} ,$$

its  $(u, v)$ -th entry is

$$(S_g)_{u,v} = \sum_{\lambda \in \Lambda} g_\lambda(u) \overline{g_\lambda(v)} , \quad u, v \in \langle L \rangle .$$

For a separable lattice, this reads as

$$\begin{aligned} (S_g)_{u,v} &= \sum_{k=0}^{\frac{L}{\alpha}-1} \sum_{\ell=0}^{\frac{L}{\beta}-1} M_{\beta\ell} T_{\alpha k} g(u) \overline{M_{\beta\ell} T_{\alpha k} g(v)} \\ &= \sum_{k=0}^{\frac{L}{\alpha}-1} \sum_{\ell=0}^{\frac{L}{\beta}-1} e^{2\pi i \beta \ell u / L} g(u - \alpha k) e^{-2\pi i \beta \ell v / L} \overline{g(v - \alpha k)} \\ &= \sum_{k=0}^{\frac{L}{\alpha}-1} \sum_{\ell=0}^{\frac{L}{\beta}-1} e^{2\pi i \beta \ell (u-v) / L} g(u - \alpha k) \overline{g(v - \alpha k)} . \end{aligned}$$

By looking at the inner sum we can see that generally

$$\sum_{\ell=0}^{M-1} e^{2\pi i \ell j / M} = \begin{cases} M & \text{if } j \equiv 0 \pmod{M} \\ 0 & \text{else ,} \end{cases}$$

and so the  $(u, v)$ -th element of the frame matrix  $S_g$  is given by

$$(S_g)_{u,v} = \begin{cases} \frac{L}{\beta} \sum_{k=0}^{\frac{L}{\alpha}-1} g(u - \alpha k) \overline{g(v - \alpha k)} & \text{if } |u - v| \equiv 0 \pmod{\frac{L}{\beta}} \\ 0 & \text{otherwise.} \end{cases} \quad (3.6)$$

Equation (3.6) is the so-called Walnut representation of the Gabor frame operator on a separable lattice, indicating that  $S_g$  is a sparse matrix where

- (1) Only every  $\frac{L}{\beta}$ -th subdiagonal is non-zero,
- (2) The entries are  $\frac{L}{\alpha}$ -periodic within every subdiagonal,
- (3) According to [Str97],  $S_g$  is a block circulant matrix of the form

$$S_g = \begin{pmatrix} A_0 & A_1 & \cdots & A_{\frac{L}{\alpha}-1} \\ A_{\frac{L}{\alpha}-1} & A_0 & \cdots & A_{\frac{L}{\alpha}-2} \\ \vdots & \vdots & \ddots & \vdots \\ A_1 & A_2 & \cdots & A_0 \end{pmatrix},$$

where the blocks  $A_j$  are non-circulant  $\alpha \times \alpha$ -matrices with entries

$$(A_j)_{u,v} = (S_g)_{u+j\alpha, v+j\alpha}$$

due to (3.6), with  $j \in \langle \frac{L}{\alpha} \rangle$  and  $u, v \in \langle \alpha \rangle$ .

### 3.3.2 The Janssen Representation

For general lattices there is another kind of representation of the Gabor frame matrix, derived from the fact that the set of time-frequency matrices  $\{\pi(\lambda)\}_{\lambda \in \mathbb{Z}_L \times \mathbb{Z}_L}$  is an orthogonal system for the matrix algebra  $\mathbb{C}^{L \times L}$  with respect to the Frobenius inner product. The system becomes orthonormal for

$$\langle A, B \rangle_f := \frac{1}{L} \langle A, B \rangle_F = \frac{1}{L} \sum_{i,j=0}^{L-1} a_{ij} \overline{b_{ij}}.$$

Therefore every matrix  $A \in (\mathbb{C}^{L \times L}, \langle -, - \rangle_f)$  has a *unique* expansion

$$A = \sum_{\lambda \in \mathbb{Z}_L \times \mathbb{Z}_L} c_\lambda \pi(\lambda) \quad (3.7)$$

with respect to the orthonormal system  $\{\pi(\lambda)\}_{\lambda \in \mathbb{Z}_L \times \mathbb{Z}_L}$ . This expansion is also called a time-frequency representation and has a generalization to the continuous case, cf. [FLW07].

The Gabor frame matrix  $S_g$  is at first sight “yet another matrix” in  $\mathbb{C}^{L \times L}$  with some time-frequency representation like in (3.7), but the introduced decomposition has an essential simplification due to a special structure of  $S_g$  that depends on another time-frequency sampling lattice.

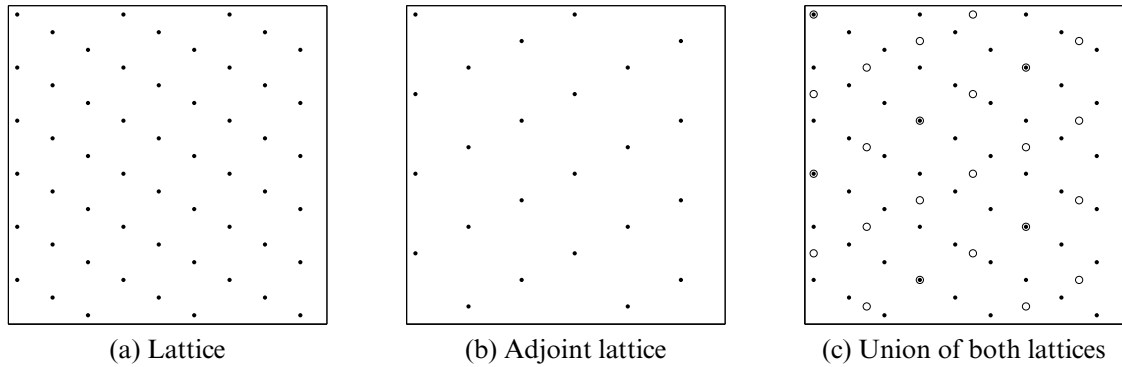


FIGURE 3.1: A lattice and its adjoint lattice

**3.3.1 Definition** The adjoint lattice  $\Lambda^\circ$  to the lattice  $\Lambda \subseteq \mathbb{Z}_L \times \mathbb{Z}_L$  consists of those elements  $\lambda^\circ \in \mathbb{Z}_L \times \mathbb{Z}_L$  for which

$$\pi(\lambda) \pi(\lambda^\circ) = \pi(\lambda^\circ) \pi(\lambda) \quad (3.8)$$

holds for all  $\lambda \in \Lambda$ .

According to (3.4), if  $\lambda = (k, \ell)$  and  $\lambda^\circ = (k^\circ, \ell^\circ)$ , then we must have

$$e^{2\pi i(k\ell^\circ - k^\circ\ell)/L} = 1 \quad (3.9)$$

for all  $(k, \ell) \in \Lambda$  and all  $(k^\circ, \ell^\circ) \in \Lambda^\circ$ . In other words, the time-frequency shifts on the adjoint lattice commute with those on the original lattice.

The syntax  $\lambda^\circ$  might be a bit misleading, as one could read that there is a single  $\lambda^\circ$  for each  $\lambda$  such that the commutation (3.8) holds. In fact, if  $\pi(\mu)$  commutes with  $\pi(\lambda')$  for a certain  $\lambda' \in \Lambda$ , then it also commutes with all  $\pi(\lambda)$  for  $\lambda \in \Lambda \setminus \{\lambda'\}$ .

Note that  $|\Lambda||\Lambda^\circ| = L^2$ . In the case of a separable lattice  $\Lambda = \alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$ , the adjoint lattice is just  $\Lambda^\circ = \frac{L}{\beta}\mathbb{Z}_L \times \frac{L}{\alpha}\mathbb{Z}_L$ .

Coming back to the Gabor frame matrix, we saw that it commutes with all TF-shift matrices  $\pi(\lambda)$  with  $\lambda \in \Lambda$  as well. As it can be uniquely expanded into a sum of TF-matrices like in (3.7), it immediately becomes clear that only those coefficients  $c_\lambda$  can be non-zero that apply to  $\pi(\lambda^\circ)$  for  $\lambda^\circ \in \Lambda^\circ$ . The time-frequency representation of  $S_g$  therefore is

$$S_g = \sum_{\lambda^\circ \in \Lambda^\circ} c_{\lambda^\circ} \pi(\lambda^\circ),$$

what is known as the Janssen representation of  $S_g$ . It can be shown [FLW07] that the Janssen coefficients  $c_{\lambda^\circ}$  are given as

$$c_{\lambda^\circ} = \langle S_g, \pi(\lambda^\circ) \rangle_f = \frac{N}{L} \langle g, \pi(\lambda^\circ)g \rangle_{\mathcal{C}^L} = \frac{N}{L} (\mathcal{V}_g g)(\lambda^\circ).$$

This relation allows for efficient numerical computation of the Janssen coefficients by using the standard FFT for the STFT.

The benefits from the knowledge of the structure of the Gabor matrix is that it paves the way for efficient numerical solutions of the linear equation (3.5) and thus for efficient computation of the dual Gabor system for reconstruction, what we will discuss in the following.

### 3.3.3 Factorizations of the Gabor Matrices

If we think back to Section 1.6 about the pseudoinverse of matrices, we saw that when the unitary factorization of a matrix  $A = U\Sigma V^*$  was found, the computation of the pseudoinverse  $A^\dagger = V\Sigma^\dagger U^*$  reduces to the calculation of the pseudoinverse of a diagonal matrix  $\Sigma$ , what is done with much more simplicity. As the calculation of the dual Gabor system involves  $S_g^{-1}$  or  $C_g^\dagger$ , we seek unitary factorizations of these matrices to either make  $S_g$  easier invertible or make the obtainment of the pseudoinverse of  $C_g$  easier possible.

Normal matrices  $A$ , i.e. such where  $AA^* = A^*A$ , are diagonalizable into  $\Delta = U^*AU$ , but it might still need some effort to actually find  $U$  and compute the matrix product  $U^*AU$ . However, if the diagonality condition on  $\Delta$  can be relaxed, one might be able to find applicable unitary matrices  $U$  more easily, compute  $U^*AU$  more easily, and  $\Delta = U^*AU$  could at least be block diagonal, such that the (pseudo-) inversion reduces to (pseudo-) inverting the blocks separately [GvL96]. We will now summarize why the matrices occurring in discrete Gabor expansions meet these directives, and start with the definition of some special matrices.

We already know that  $\text{diag}(d) = \text{diag}(d_0, \dots, d_{L-1})$  is the  $L \times L$ -matrix containing  $d \in \mathbb{C}^L$  in its main diagonal. If  $D_0, \dots, D_{L-1}$  are  $p \times p$ -matrices, we use the same notation to describe the block diagonal matrix

$$\text{diag}(D_0, \dots, D_{L-1}) := \begin{pmatrix} D_0 & 0 & \cdots & 0 \\ 0 & D_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_{L-1} \end{pmatrix} \in \mathbb{C}^{pL \times pL}$$

with  $0 \in \mathbb{C}^{p \times p}$ .

**3.3.2 Definition** A permutation matrix is derived from the identity matrix  $I_L$  by a reordering of its columns. The Modulo  $p$  perfect shuffle permutation  $P_{p,L}$  with  $L = pq$  is a certain kind of permutation matrix that reorders the rows of an  $L \times L$ -matrix (or of a vector in  $\mathbb{C}^L$ ) in the following way:

$$P_{p,L}: \{0, 1, \dots, L-1\} \mapsto \{0, q, 2q, \dots, (p-1)q, 1, q+1, \dots, (q-1)p-1, pq-1\}.$$

**3.3.3 Definition** The Fourier matrix  $F_L \in \mathbb{C}^{L \times L}$  of order  $L$  implements the unitary DFT on  $\mathbb{C}^L$  and has the symmetric form

$$F_L := \sqrt{\frac{1}{L}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{L-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \cdots & w^{(L-1)(L-1)} \end{pmatrix} \quad (3.10)$$

with  $w := e^{-2\pi i/L}$ .



**3.3.4 Definition** The Kronecker product  $A \otimes B$  of  $A \in \mathbb{C}^{p \times q}$  and  $B \in \mathbb{C}^{m \times n}$  is the  $p \times q$  block matrix

$$A \otimes B := \begin{pmatrix} a_{0,0}B & \cdots & a_{0,q-1}B \\ \vdots & & \vdots \\ a_{p-1,0}B & \cdots & a_{p-1,q-1}B \end{pmatrix} \in \mathbb{C}^{pm \times qn}$$

with blocks of size  $m \times n$ .

Note that  $I_p \otimes B = \text{diag}(\underbrace{B, \dots, B}_{p \text{ times}}) \in \mathbb{C}^{pm \times pn}$  is block diagonal.

These matrices are now the ingredients of important factorizations of the Gabor frame matrix in the case of a product lattice  $\Lambda = \alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$ :

**3.3.5 Theorem**  $S_g$  can be unitarily factorized into the block diagonal matrix

$$\Delta_{S_g} = P_{L/\beta, L}^* S_g P_{L/\beta, L} = \text{diag}(B_0, \dots, B_{L/\beta-1}),$$

where  $B_k$  is the  $\beta \times \beta$ -matrix with entries

$$(B_k)_{u,v} = (S_g)_{k+uL/\beta, k+vL/\beta}.$$

*Proof.* See [Str97, 8.3.1]. □

**3.3.6 Theorem**  $S_g$  can be unitarily factorized into a block diagonal matrix by

$$(F_{L/\alpha} \otimes I_\alpha)^* S_g (F_{L/\alpha} \otimes I_\alpha).$$

*Proof.* See [Str97, 8.3.3]. □

The block diagonalization of Theorem 3.3.5 can be derived from a block diagonalization of the analysis matrix  $C_g$  found by Prinz in [Pri96]:

**3.3.7 Proposition** The matrix  $C_g$  can be factorized into a block diagonal matrix

$$\Delta_{C_g} = P_{L/\beta, L}^* C_g (I_{L/\alpha} \otimes F_{L/\beta}^*) P_{L/\alpha, L^2/(\alpha\beta)}^* = \text{diag}(W_0, \dots, W_{L/\beta-1})$$

with rectangular  $\beta \times \frac{L}{\alpha}$ -blocks  $W_k$  with entries

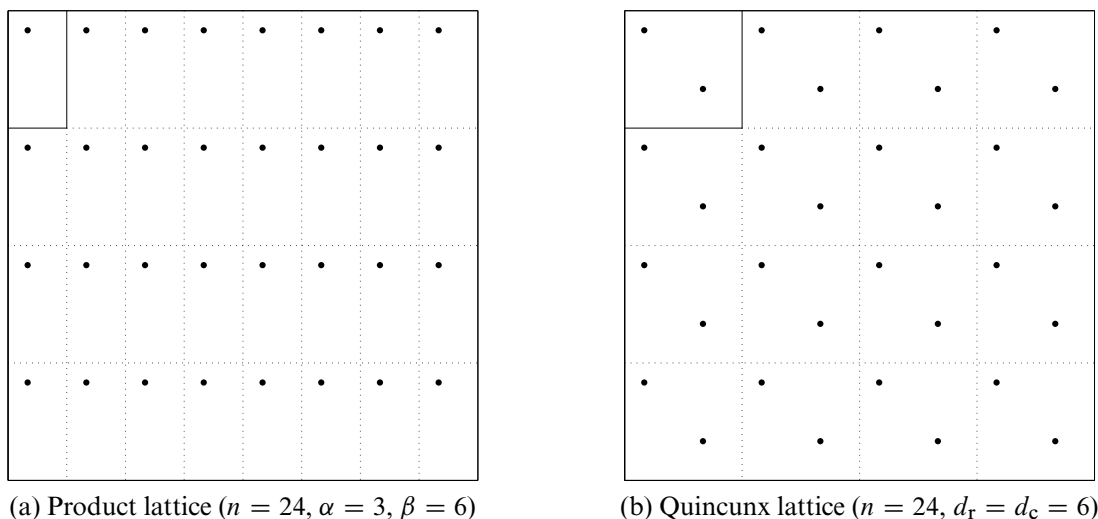
$$(W_k)_{u,v} = \sqrt{\frac{L}{\beta}} g(k + u\frac{L}{\beta} - v\alpha).$$

A benefit of this structure is that the blocks  $W_k$  can be obtained by simple permutations of the entries of  $g$ . Finally, it can be shown that the derived factorizations relate as

$$\Delta_{S_g} = \Delta_{C_g} \Delta_{C_g}^*.$$

$C_g$  and thus  $\Delta_{C_g}$  are not invertible in the case of oversampling, but due to  $\Delta_{C_g}^\dagger = \Delta_{C_g}^* (\Delta_{C_g} \Delta_{C_g}^*)^{-1}$ , the dual Gabor system is obtained by the inversion of  $\Delta_{S_g}$ .

These methods for quick computation of the discrete Gabor transform were recently extended by Søndergaard in [Søn07] to multiple signals  $f \in \mathbb{C}^{L \times N}$ , where  $N$  is the number of signals.

FIGURE 3.2: Sampling subgroups with redundancy  $\frac{4}{3}$  and their building blocks

### 3.4 The Dual Window on Non-Separable Sampling Sets

There is the question of how the dual Gabor window can be calculated efficiently for arbitrary sampling subgroups  $\Lambda \trianglelefteq \mathbb{Z}_L \times \mathbb{Z}_L$  besides pseudo-inverting the corresponding Gabor analysis operator. Prinz proposed a method in [Pri96] by reducing the case of non-separable subgroups to the known case of a separable subgroup, what makes it possible to apply the factorization methods that we mentioned in the previous section.

Why should non-separable subgroups be of interest? An important thing that emerges is that a non-separable sampling set might yield some better properties of the dual window, like a better localization compared to the dual on a separable lattice  $\Lambda = \alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$ , even if a lower redundancy is used. For certain non-separable cases, however, the raise of truly complex-valued duals might not be favored for (partial) reconstruction. In this section we will show examples for both cases.

General sampling subgroups can be represented by the smallest row and column distance ( $d_r, d_c$ ) and by the sampling points  $(k, \ell) \neq (0, 0)$  that satisfy  $k < d_r$  and  $\ell < d_c$ , cf. [Pri96]. It turns out that any sampling matrix is a block matrix determined by a single  $d_r \times d_c$ -block. A product lattice for instance has four points at the corners of the block with size  $\alpha \times \beta$ , cf. subfigure 3.2a. A special case of a non-separable subgroup is the so-called *quincunx* lattice, where the corresponding block is square and the points are located like those of the number 5 on a dice; cf. subfigure 3.2b. It looks like a union of a product lattice plus the lattice shifted by half of the distance between the points.

A direct approach for describing Gabor expansions on a quincunx lattice is given by Bastiaans and van Leest e.g. in [BvL98a] and [BvL98b]. However, the bookkeeping of various indices and prefactors takes quite some effort that is avoided in the language of group theory. By the way, it is not possible to keep the number of sampling points and thus the redundancy when switching from a square product lattice

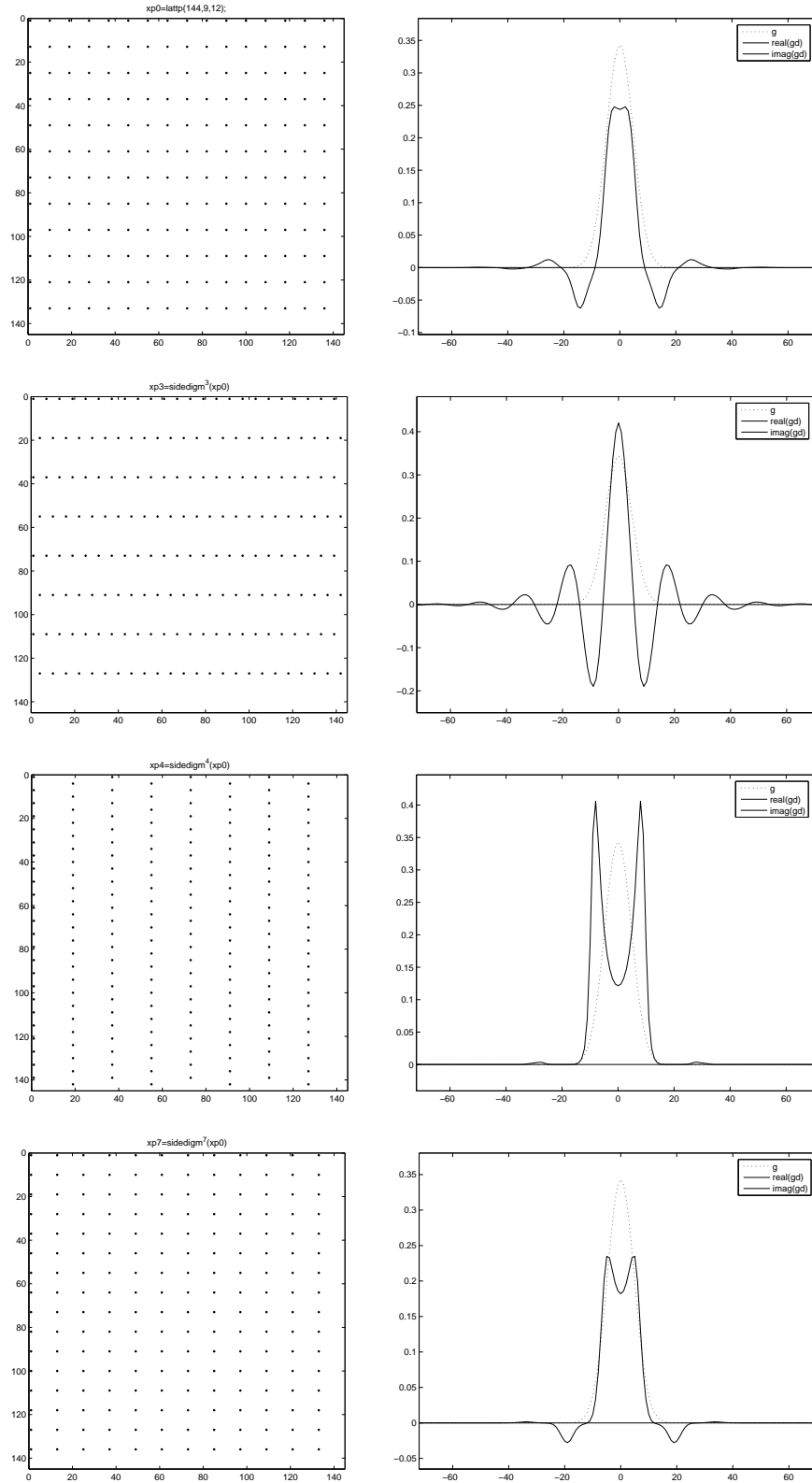


FIGURE 3.3: Dual windows on general subgroups with zero imaginary part. The signal length is  $L = 144$ , and the initial product lattice is built by  $\alpha = 9$  and  $\beta = 12$ . The other lattices are built by extracting the subdiagonals of the lattice matrices subsequently and thus share the same redundancy of  $\frac{4}{3}$ .

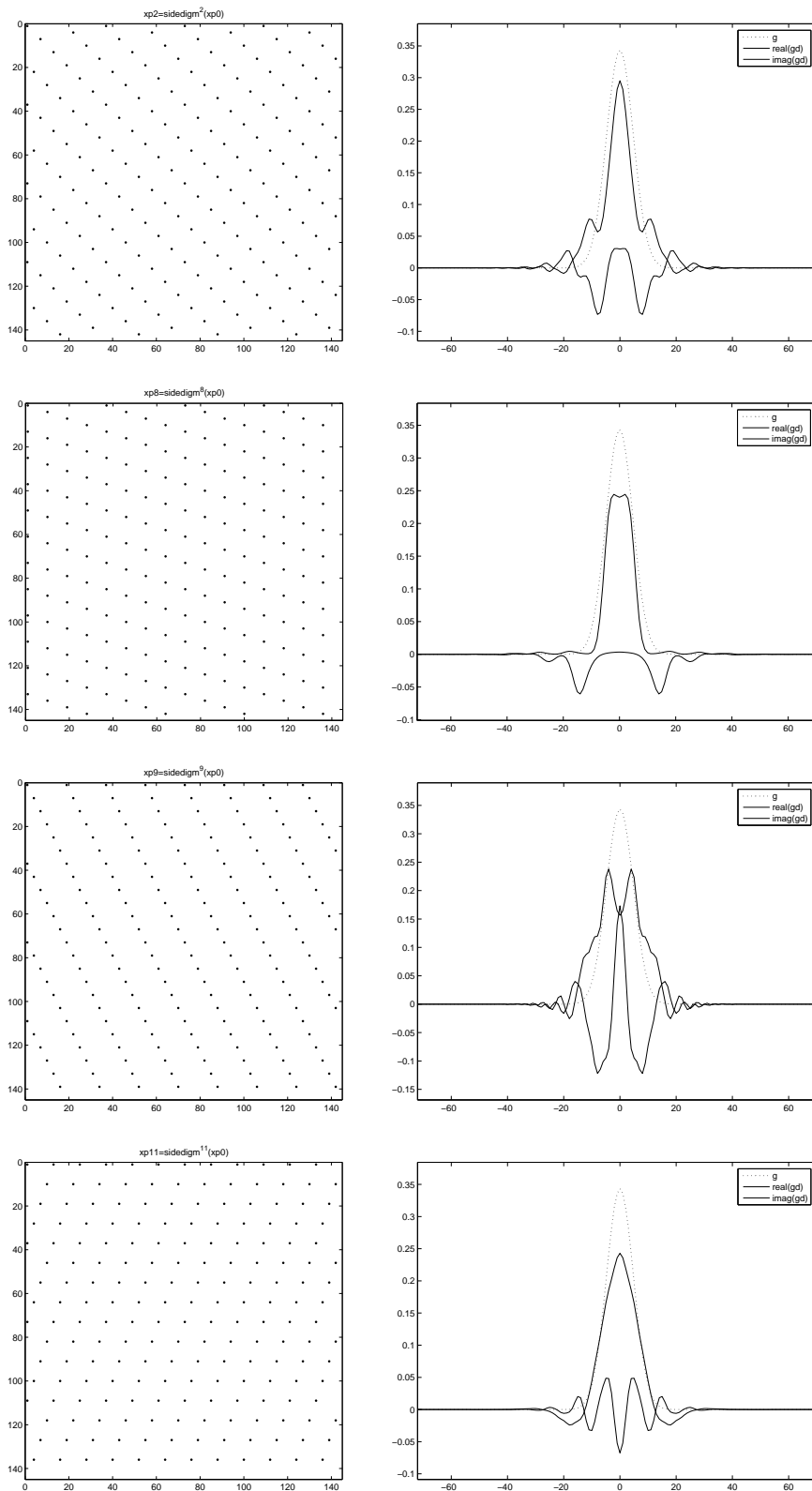


FIGURE 3.4: Dual windows on general subgroups with non-zero imaginary part. The lattices are built the same way as in Figure 3.3.

with  $\frac{L^2}{\alpha^2}$  points to a quincunx lattice with  $2n^2$  points, as this would force

$$\frac{L^2}{\alpha^2} = 2n^2 \Rightarrow \sqrt{2} = \frac{L}{\alpha n} \in \mathbb{Q} ,$$

what can never be achieved.

The upper two subfigures in Figure 3.3 show a product lattice and the corresponding dual window for the Gaussian of length  $L = 144$ . In MATLAB the lattice is stored as a 0-1-matrix with ones at the position of the sampling points and is constructed by the function `lattice.m` developed by NuHAG. One way of getting to a non-separable subgroup with same redundancy is extracting the diagonals of the lattice matrix and using it as new sampling matrix. This extraction is implemented by `sidedigm.m`. Applying this method subsequently yields various separable or non-separable subgroups with the same number of points and thus with same redundancy as the initial product lattice. Unluckily, one cannot derive all possible subgroups of given redundancy this way. The remaining three subfigures in Figure 3.3 show examples of sampling lattices where the dual remains purely real-valued.

However, on other (non-separable) lattices with same redundancy the dual might get a significant imaginary part as shown in Figure 3.4. The second lattice in that figure is an example for a case where the real part of the dual is very well localized, but its non-zero imaginary part eventually equips the whole dual with the same overall localization as the duals on the other lattices with same number of sampling points. If a partial Gabor reconstruction is done on such a subgroup where the dual is truly complex, one ends up with a truly complex signal as well, what might not be preferred for comparison with an initially real-valued signal. In the case of a full Gabor reconstruction the imaginary part of the signal should of course turn numerically zero.

Finite-dimensional discrete Gabor analysis is usually described for one-dimensional signals, i.e., signal vectors in  $\mathbb{C}^L$ , just as it was done in this chapter. As this thesis wants to show Gabor expansions of image signals, some things emerge that are not present in the case of one-dimensional signals. Therefore, we will now explore the basics of what image signals are and how classical Fourier analysis of these is done in the next chapter. Chapter 5 finally extends the analysis of images to Gabor expansions.



# Chapter 4

## Fourier Analysis of Discrete Images

### 4.1 Digital Representation of Images

In this section we want to give a brief summary of how image signals are represented in digital computers, what their value range is and how they are shaped.

Time-variant signals such as sound or brainwaves (electroencephalograms, EEG) are one-dimensional (1D) signals because they only evolve in the time dimension. When such signals are to be represented in digital computers, they can only be considered as a discrete and finite sequence of real values. The length of this sequence is called the *signal length*. Mathematically, such a signal of length  $L$  is considered as vector in  $\mathbb{C}^L$  although *real* signals might only be real-valued. One should not confuse the  $L$ -dimensionality of the signal space with the one-dimensionality of the signal as such; a 1D signal should be seen as a synonym for a signal vector. Although the sequence values are ordered corresponding to their chronological occurrence, the actual time span in which they subsequently occur is not represented in the vector as such. It has to be clear from the signal processing context how many subsequent values or *samples* occur within a certain time span. This is called the *sampling rate* of the signal.

The value range of 1D signals allows arbitrary finite real or complex values, as far as they can be represented as numerical values. Sine waves for example simply oscillate between  $-1$  and  $1$ .

#### 4.1.1 The Nature of Images

For the case of the representation of natural images as digital signals, some more things have to be considered. One way to try to understand natural images is to define them as projections of three-dimensional (3D) natural structures onto a two-dimensional (2D) plane by means of electromagnetic rays such as light or x-rays, or particle rays such as electrons. The rays are sent out in parallel, e.g. by the sun or an artificial source, are reflected or filtered by the structures and strike on a planar sensor such as the animal or human retina or an artificial sensor that is sensitive to the corresponding type of rays. Artificial sensors are built to imitate the behavior of the retina. However, if the surfaces emit rays by themselves, these could be detected directly, if e.g. the surface of the sun or the heat behavior of a different surface are to be studied. The detected intensity of the rays is then coded into corresponding low

and high (numerical) values. In the case of biological vision, a low intensity of rays is interpreted in the brain as darkness or shadows, and high intensities are seen bright.

As long as all this is time-invariant, an image does not evolve in (one-dimensional) time, but on a two-dimensional plane. In the case of time-variance of these images, the signal is to be considered three-dimensional, as a finite collection of single images, each representing a certain discrete time-position.

Throughout this thesis, only the time-invariant case of 2D signals is considered, what may correspond to a single temporal sample of a time-variant image signal.

### 4.1.2 Digital Images and Color

The discretized versions of natural images can still be considered as vectors of finite length if it is clear what values correspond to what locations on the plane. However, many applications have the target to make these digital representations again visible to the human eye. One-dimensional signals are most of the time visualized as two-dimensional graphs, with time evolving horizontally and the signal values vertically. 2D signals could therefore be visualized as (projections of) three-dimensional graphs, but this does not correspond to the behavior of the human or animal retina. Graphically, the detected low intensities, coded as low numerical values, should be made visible as “dark” values, and detected high intensities should again appear “bright”. Therefore, instead of presenting a graph of the values over a tilted plane, the values should be represented directly on a planar surface as tones of black, gray and white spots at the corresponding positions on the plane.

However, these intensities do not consider color. Even if the ray source consisted of electro-magnetic waves of different wavelength, so far we can only construct black-and-white images, what might be enough for heat images or images of surface structure, but it is not clear how certain intensity values should be mapped back to electro-magnetic waves of different wavelength, or, in other words, what color the human eye would have seen at certain spots. The only possibility so far would be using a different *colormap*, where certain intensities are not just represented as shades between black and white. For instance, infrared images (heat images) could be presented with false colors to the human eye, where low temperatures (low numerical values) are visualized as tones of blue, and high temperatures (high numerical values) as tones of red. The values “in between” could again be represented by other colors to make the middle tones better visible to the human eye.

As the human eye cannot see infrared light or x-rays, it is even more important to visualize image values in a reasonable way. If it comes to natural images where colors should be representable, one immediately steps over the two ways of creating colors: The *additive* or the *subtractive* way. Additive color creation happens by the direct emission of certain wavelengths of light, whereas subtractive color happens by filtering certain wavelengths out of light that already contains many different wavelengths. The concept of additive color is not a property of light itself, but a result of how the human eye detects color. The additive reproduction process usually uses varying intensities of red, green and blue light, what is enough to produce a reasonable subset of the possible colors in human vision. The subtractive color system is



implemented through pigments, dyes, inks and other substances by filtering and reflection. Here, the basic colors are cyan (bluish), magenta (reddish), and yellow. In printing processes that usually happen onto white surfaces, black is added for a better representation of correspondingly dark values.

These two color models are also implemented in digital computers. Additive color creation happens at monitors or screens by active light emission, whereas colors are created the subtractive way in print-outs. The corresponding color models are commonly named by abbreviating the incorporated basic colors and are therefore referred to as the RGB or CMYK color models, where the K stands for black. It is important to note that the created color spaces are disjoint, e.g. there are RGB colors that cannot be represented as CMYK colors and vice versa. In addition, the exact mapping between RGB and CMYK may vary with the used hardware implementations on both sides.

### 4.1.3 RGB Images

Our first restriction is that to the RGB color model. RGB color images are represented in digital computers by considering the three color channels as three different intensity images. Each of the three images represents the intensities of red, green and blue light, respectively, by using corresponding low and high numerical values. As it is not possible to do such a split after the recording of an image with a light sensor, sensors either have to be covered by corresponding color filters, or three different sensors have to be used to distinguish the recorded light intensities into the three color layers of an image.

So far we have not defined where “low” and “high” numerical values may range. In digital computers, numerical values are stored as binary values with a certain fixed digit length. The more *bits* (binary digits) are allowed, the higher decimal values can be represented. It is common to use a width of 8 bits for each color channel, as this allows for coding 256 integer values between 0 and 255 for each of the channels, what makes it possible to code  $256 \times 256 \times 256$  and therefore more than 16.7 million different color values, what is a good compromise between computational simplicity and the demands of the human eye. But the true reason for using 8-bit values is that today’s standard hardware is limited in its capability of either detecting or showing many different color shades.

Figure 4.1 visualizes the red, green and blue color channels of an RGB image as three different black-and-white images. One can see how the nose of the pictured animal shows intensities mostly in the red channel, whereas the area around the nose has its intensities in the green and blue channel. This results in a red colored nose in the actual color image, and the surrounding area appears in a light blue. The eyes are also intense in the red channel, and a little amount of green gives them their orange color in the resulting image.

In this thesis we restrict to only a single color layer and therefore to simple black-and-white images with values between 0 and 255. This of course will ignore phenomena that do not just happen within a single color channel. Therefore, disturbances that incorporate all three color layers at certain areas on the two-dimensional plane

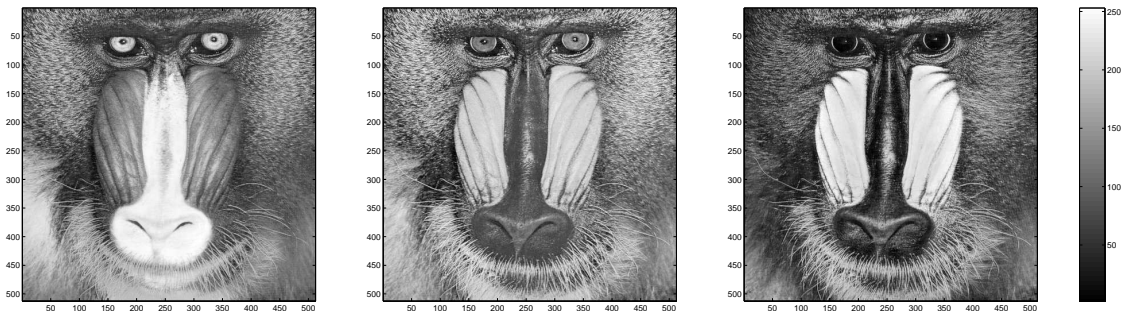


FIGURE 4.1: Red, green and blue color channel of an RGB image

are not considered here.

One already notices a little discrepancy between the visualization of low or high numerical values. Low ray intensities appear dark or black to the human eye and correspond to numerical values close or equal to zero. The opposite is a high presence of light, seen as bright color and represented as high numerical values close or equal to 255. However, for usual prints on white paper it's just the other way round: Dark colors are created by a high presence of black ink or toner, and bright or white colors are given by the absence of ink and the white color of the paper. Therefore, when visualizing low or high values, one must not mix this up with presence or absence of ink. As prints of digital images are shown in this thesis, we should keep in mind that dark colors represent low values and bright colors represent high values. If however most of the values in a signal are low and if it appears reasonable to not show an almost black image, the color map might be inverted and zero-values correspond to the white color of the paper. It should be clear from the context or from the graphics what color map is in use. Additionally, if significant behavior of an image mainly happens in the lower or the higher value range, the color map might be changed to a non-linear one such that the diversity in the image is better visible to the human eye. Also, if for whatever reason the image values exceed the allowed range after numerical processings, the values are scaled back into their allowed range for presenting prints. We equip figures of images with a corresponding color bar where possible or reasonable.

We mentioned that such an image can still be given as a vector, but it is not automatically clear what values appear at what position. To avoid an additional storage of coordinates, numerical computer software such as MATLAB or Octave simply shape a digital image into a matrix form. This way the *pixels* (picture elements) get their coordinates simply by their location in the number matrix. This case of quadratic pixels is the easiest to implement, for both software or hardware engineers. As for 1D signals, it should be clear from the type of application how many subsequent pixels represent a certain “real” length, if it is important at all.

However, for describing the evaluation of linear operations mathematically, this still has to be done by applying operator matrices on signal vectors.

## 4.2 Understanding 2D Frequencies

Fourier series have a very natural relation to acoustics. They define a periodic function as a series of exponential functions, and from acoustics one might remember that tones and sound are composed of pure oscillations that overlay without disturbance. The speed of oscillation of a complex exponential function can therefore be interpreted as such a pure frequency. The Fourier transform of a sound signal is a perfect way to show what frequencies occur in the sound.

We defined the Fourier transform in a general way, what allows it to be used for image signals. An image is as well analyzed into certain oscillating building blocks, but the idea of a two-dimensional (2D) frequency doesn't seem very natural. Whereas one might have some understanding that frequencies can be heard, there might be some uncertainty whether one has ever *seen* a 2D frequency.

In this section we want to take a closer look at the exponentials  $e^{2\pi i \mathbf{x} \cdot \boldsymbol{\omega}}$  for  $\mathbf{x}, \boldsymbol{\omega} \in \mathbb{R}^2$  or even in  $\mathbb{R}^d$ , as these are the building blocks of the continuous Fourier transform. Considering that  $\mathbf{x} \cdot \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{C}^d}$ , it is not immediately clear how  $e^{i\langle \mathbf{x}, \boldsymbol{\omega} \rangle}$  behaves as a function in multidimensional  $\mathbf{x}$  and  $\boldsymbol{\omega}$ , whereas it is easier to understand for  $t, \omega \in \mathbb{R}$ : Here, the inner product is just the product  $\omega t \in \mathbb{R}$ , and seeing  $t$  as time,  $e^{i\omega t}$  rotates quicker on the complex unit circle the larger  $|\omega|$  gets, and naming  $\omega$  the frequency has a very natural justification. The graph of this exponential is a helix in  $\mathbb{R} \times \mathbb{C}$ .

The first increment in dimension leads to a non-trivial form of the inner product, and we first try to understand  $u(\mathbf{x}, \mathbf{y}) := \langle \mathbf{x}, \mathbf{y} \rangle$  as a (linear) function in  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ . Unfortunately,  $u$  is now a mapping  $\mathbb{R}^4 \rightarrow \mathbb{R}$ , what is rather difficult to visualize. To be able to get a better picture, we first fix  $\mathbf{y} = \mathbf{a} \in \mathbb{R}^2$  and look at the function  $u_{\mathbf{a}}(\mathbf{x}) := \langle \mathbf{x}, \mathbf{a} \rangle$  what is now a mapping  $\mathbb{R}^2 \rightarrow \mathbb{R}$  and therefore easy to visualize. We know that the inner product  $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{k=1}^d x_k y_k$  is a linear function for  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ , and in the 2D case it just corresponds to a plane equation  $z = a_1 x_1 + a_2 x_2$ . That plane cuts the zero-plane  $z = 0$  in a straight line that is dependent on the orientation of the fixed  $\mathbf{y} = \mathbf{a}$ : The zero-line evolves where the vectors  $\mathbf{x}$  are orthogonal to the fixed vector  $\mathbf{y}$ , i.e.,  $\langle \mathbf{x}, \mathbf{y} \rangle = 0 \Leftrightarrow \mathbf{x} \perp \mathbf{y}$ . This zero-line will now change its angle on the zero-plane while varying the fixed vector  $\mathbf{y}$ . This gives a picture how it behaves as a whole, and it provides the key for understanding what happens when the inner product is put into the exponent of the complex exponential function.

As  $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{R}$  for  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ , the exponential  $e^{i\langle \mathbf{x}, \mathbf{y} \rangle}$  still has its values on the complex unit circle. Fixing a vector  $\boldsymbol{\omega} \in \mathbb{R}^2$  and varying  $\mathbf{x} \in \mathbb{R}^2$ ,  $e^{i\langle \mathbf{x}, \boldsymbol{\omega} \rangle}$  rotates fastest on the unit circle if  $\mathbf{x}$  “goes parallel” to  $\boldsymbol{\omega}$ , and it is constant when  $\mathbf{x}$  “goes perpendicular” to  $\boldsymbol{\omega}$ . Keeping  $\mathbf{x}$  parallel to  $\boldsymbol{\omega}$ , the speed of the oscillation increases with increasing length of  $|\boldsymbol{\omega}|$ , as the plane  $\mathbf{x} \mapsto \langle \mathbf{x}, \boldsymbol{\omega} \rangle$  becomes steeper. For real signals, it makes sense to look at the real part of the exponential only, and just as in the 1D case, the 2D oscillation also describes a wave, a plane wave with oscillations between  $-1$  and  $1$  when going on  $\mathbf{x}$  parallel to  $\boldsymbol{\omega}$ , and with constant value when going perpendicular to  $\boldsymbol{\omega}$ .

Seeing the 2D elementary circle functions as building blocks for images, they appear as line patterns, where the “density” of the pattern grows and shrinks with  $|\boldsymbol{\omega}|$ .

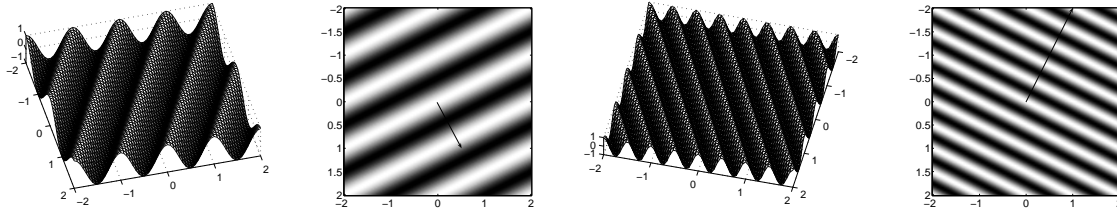


FIGURE 4.2: Real parts of  $e^{2\pi i(x,\omega)}$  for  $\omega = (0.5, 1)^T$  and  $\omega = (1, -2)^T$ . The plane waves are shown as 3D graphs and as black-and-white images, where the vectors  $\omega$  are indicated by the respective arrows.

Compared to the 1D case, there is now a second degree of freedom—the orientation. Instead of seeing  $\omega_1$  and  $\omega_2$  as the components of a 2D frequency, one might prefer to see  $|\omega|$  as the parameter for the pattern density and define an orientation  $\theta \in (-\pi, \pi]$  for  $|\omega| \neq 0$  similar to the argument of complex numbers by

$$\theta := \begin{cases} \arccos \frac{\omega_1}{|\omega|} & \omega_2 \geq 0 \\ -\arccos \frac{\omega_1}{|\omega|} & \omega_2 < 0. \end{cases}$$

The 2D Fourier transform of a function in  $x \in \mathbb{R}^2$  is a function in  $\omega \in \mathbb{R}^2$ , considering all possible pattern densities and all their orientations, and analyzing the amount of the corresponding 2D frequency among the variations of the function values. When we talk about low or high 2D frequencies, we mean low and high values of  $|\omega|$  and include all orientations  $\theta$ .

Considering that  $e^{2\pi i x \cdot \omega} = e^{2\pi i(x_1\omega_1 + \dots + x_d\omega_d)} = e^{2\pi i x_1\omega_1} \dots e^{2\pi i x_d\omega_d}$  for  $x, \omega \in \mathbb{R}^d$  and writing  $(f \otimes g)(x, y) := f(x)g(y)$ , a  $d$ -dimensional frequency function can be described as a tensor product of  $d$  one-dimensional oscillations.

The term “frequency” as “oscillations per time” is not valid anymore for signals with more than one dimension. Images don’t have a temporal domain, but two spatial domains, making a 2D frequency a description for “oscillations per area”. Instead of talking about the time-frequency analysis of images, one could rather use terms like *position-frequency analysis* or *space-wavenumber analysis*. Mathematically, time-frequency analysis happens in arbitrary dimensions anyway.

### 4.3 Frequency Behavior of Natural Images

As we observed that images can as well be described as being composed of pure planar oscillations, we now want to have a look at what the Fourier transform of natural images tells us, what patterns are dominant and how image signals respond to a filtering or thresholding of the Fourier coefficients.

In signal processing, signals are considered as finite periodic sequences, meaning that for  $f = (f(0), \dots, f(L-1)) \in \mathbb{C}^L$  with signal length  $L \in \mathbb{N}$  one has  $f(j) = f(j+nL)$  for all  $n \in \mathbb{Z}$  and  $j \in \langle L \rangle := \{0, \dots, L-1\}$ . The domain of  $f$  is therefore given as  $\mathbb{Z}_L := \mathbb{Z}/L\mathbb{Z}$  rather than just as the set  $\langle L \rangle$ , what ensures the structure of a finite Abelian group  $\mathcal{G}$  under the addition operation. The signal space can be

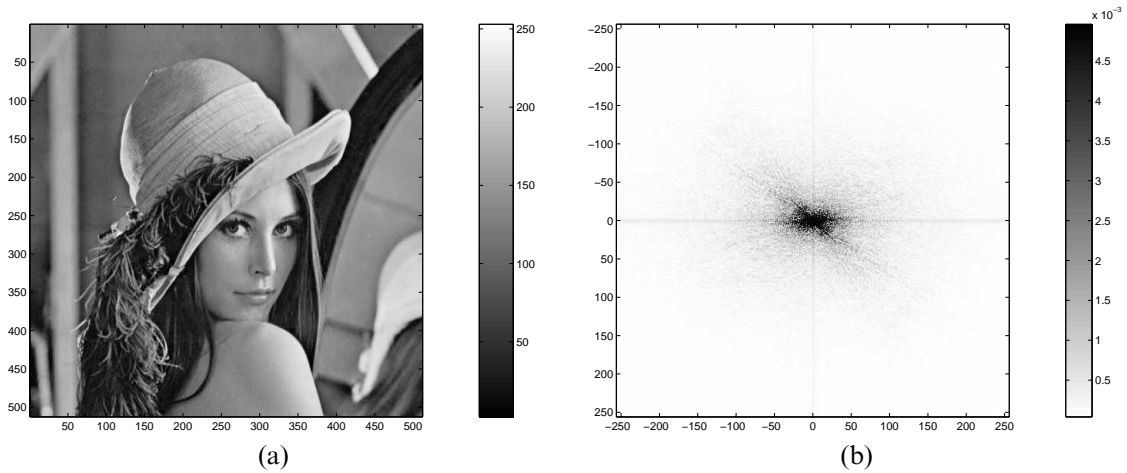


FIGURE 4.3: The well-known Lena image and its 2D-FFT spectrum

described as  $\ell^2(\mathcal{G}) = \ell^2(\mathbb{Z}_L) \cong \mathbb{C}^L$ . Image signals incorporate a second signal length, and the image space can be described as

$$\ell^2(\mathcal{G}) = \ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}) \cong \mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2} \cong \mathbb{C}^{L_1 L_2} .$$

$L_1$  and  $L_2$  are the height and width<sup>1</sup> of an image, and the group structure of  $\mathcal{G} = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  yields a two-dimensional periodicity, where the image plane becomes some sort of discrete torus. Just as for 1D signals, discontinuities might appear at the signal borders when performing time-frequency analysis, because values at opposite borders become neighbors.

Given an image  $f = (f(u, v))_{u,v} \in \mathbb{C}^{L_1 \times L_2}$ , its 2D discrete Fourier Transform (DFT) is given as

$$\hat{f}(j, k) = (\mathcal{F}f)(j, k) = \sum_{u=0}^{L_1-1} \sum_{v=0}^{L_2-1} f(u, v) e^{-2\pi i(uj/L_1 + vk/L_2)} \quad (4.1)$$

for  $(j, k) \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ .

We will look at some examples. Figure 4.3 shows the well-known Lena image and its 2D Fast Fourier Transform (FFT), an efficient numerical implementation of the 2D DFT. The coefficients are shown in their absolute values, resulting in an image of the Fourier power spectrum. Image domain and frequency domain have the same size of  $512 \times 512$  pixels. As the image signal is real-valued, the image of the Fourier power spectrum shows a symmetry, similar to the symmetry that is known for the Fourier transform of real-valued 1D signals. The colormap in Subfigure 4.3b has been scaled to visualize the lower values among the Fourier coefficients. This is because the lowest frequencies, located at the center of 4.3b, are very dominant such that only the very center would appear as a single black pixel when using a linear colormap. The

<sup>1</sup>Numerical software such as MATLAB or Octave name the vertical dimension first, just like for  $L_1 \times L_2$ -matrices. Image processing software usually does it the other way round and talks about an  $L_2 \times L_1$ -image.

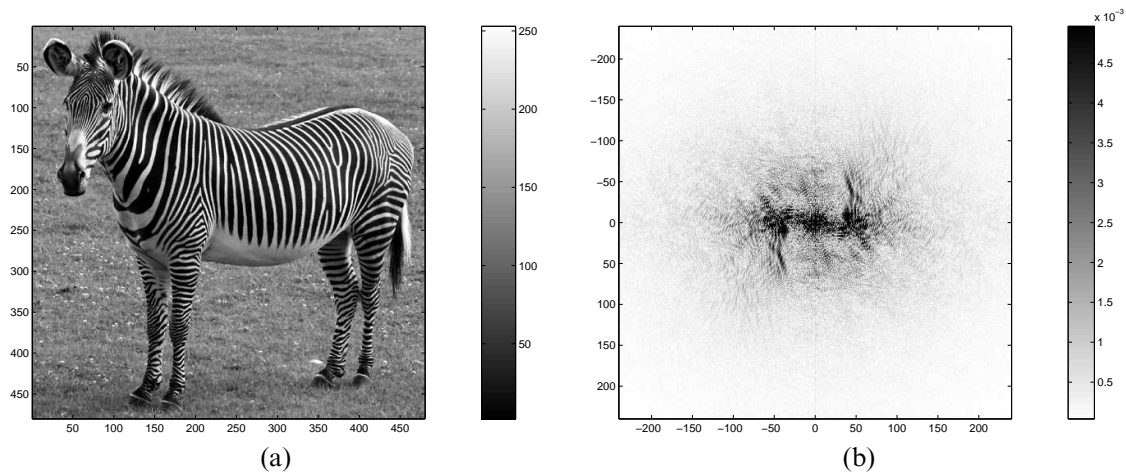


FIGURE 4.4: A zebra image and its 2D-FFT spectrum

horizontal and vertical line going through the center of the FFT-image are due to disturbances at the borders of the Lena image, where a quick (local) change from dark to bright means a jump from low to high values. This discontinuity yields a wide range of frequencies in the Fourier coefficients, just as it is known for 1D signals, and this wide frequency range is indicated by the horizontal and vertical line going from the center to the borders. Figure 4.5 shows what is meant by 2D periodicity, and the border discontinuities of the Lena image are clearly visible.

Besides those horizontal and vertical discontinuities, Subfigure 4.3b also seems to indicate a dominant frequency range from “northwest” to “southeast” of the center. One should remember that the line patterns of a pure 2D oscillation are perpendicular to the corresponding 2D frequency vector. Therefore, the observed direction indicates that there must be local jumps from dark to bright in the direction from the upper *right* to the lower *left* of the Lena image 4.3a. Indeed, we might presume them as parts of the mirror and Lena’s hat.

Another test image we will frequently use is that of a Grevy’s zebra, sized  $480 \times 480$  pixels. The visual appearance of the pure 2D oscillations as black-and-white line patterns in images suggests the usage of a zebra test image, as one intuitively guesses that such an image contains many lone oscillations at various positions. Figure 4.4 shows the zebra image and its 2D-FFT. We don’t see those border discontinuities like in the 2D-FFT of the Lena image, what Subfigure 4.5b confirms. Nevertheless, some other frequencies seem to be dominant compared to the Lena image. The spots to the left and right of the center of 4.4b correspond to vertical line patterns of medium frequency, and we guess them as the zebra patterns at the body of the animal. Just as mentioned in Section 1.4, the Fourier transform lacks information of where these frequencies actually occur. This will eventually lead us to looking at the STFT of an image signal. The term “short-time” is again misleading for the 2D case, and one might prefer to talk about a *localized Fourier transform* or simply a *windowed Fourier transform*.

As a different approach to get a better idea of where the indicated frequencies actually occur in the analyzed image, one could do a corresponding masking of the

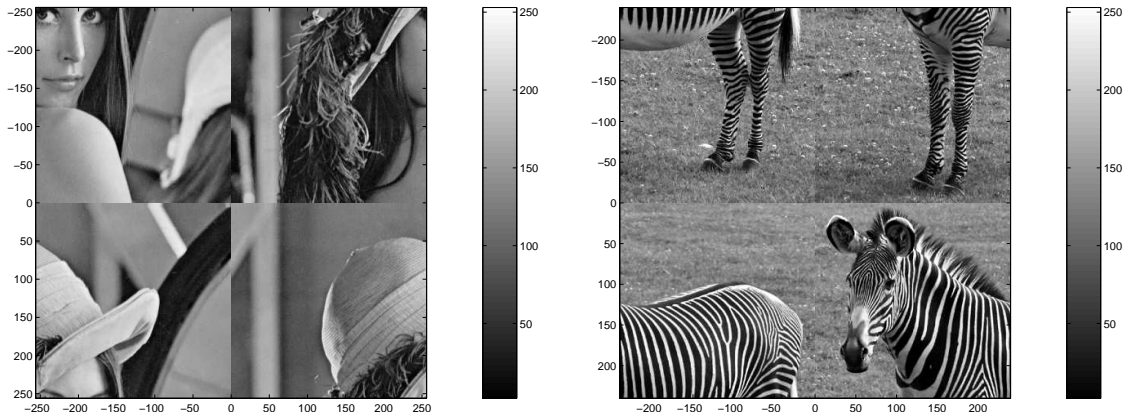


FIGURE 4.5: Image signals are periodic over their borders

Fourier coefficients and compute the inverse Fourier transform. Uninteresting frequencies can be suppressed by putting their Fourier coefficient values to zero, the inverse Fourier transform will therefore only consider the untouched coefficients and the resulting image signal will only be composed of the interesting frequencies. We want to do an experiment on that, as we notice that our two test images seem to be heavily dominant in the lowest frequencies<sup>2</sup>. Indeed, subfigures 4.3b and 4.4b only scale the lowest 0.5% of the Fourier coefficients into the colormap from white (lowest) to black (highest). The upper 99.5% also appear black and only consist of a few pixels around the centers.

### 4.3.1 Experiment: Coefficient Filtering by Masking

We want to ask what pure oscillations give what contributions to the complete image. To what extent do low frequencies contribute so much to the image? What properties do the medium or high frequencies add to the image? We will answer these questions by applying different masks to the domain of Fourier coefficients such that only the lower, medium or higher frequencies will survive. A circular partition around the center of the Fourier domain allows to cover all orientations  $\theta$  of frequencies within a certain value range of  $|\omega|$ . By doing the inverse DFT we assess the visual appearance of the reconstruction and measure the differences to the original image.

Applying a mask to the Fourier coefficient space corresponds to applying a window function to the Fourier transform that trims the coefficient values to an area of interest. The 2D-FFT of an  $L_1 \times L_2$  image yields a matrix of same size, and a mask could therefore be implemented as pointwise multiplication by another matrix of same size, namely by a 0-1-matrix with ones at locations where the Fourier coefficients should survive, and with zeroes where they should be put to zero.

The four images on the left of Figure 4.6 show four such disjoint maskings on the Fourier space of the Lena image, and the four images on the right show the corresponding reconstructions by the inverse 2D-FFT. The masks have circular boundaries that correspond to 5%, 20%, 50% and 100% of the maximum radius around the

<sup>2</sup>Do not mix up low frequencies  $|\omega|$  with low coefficient values  $|\hat{f}(\omega)|$ .

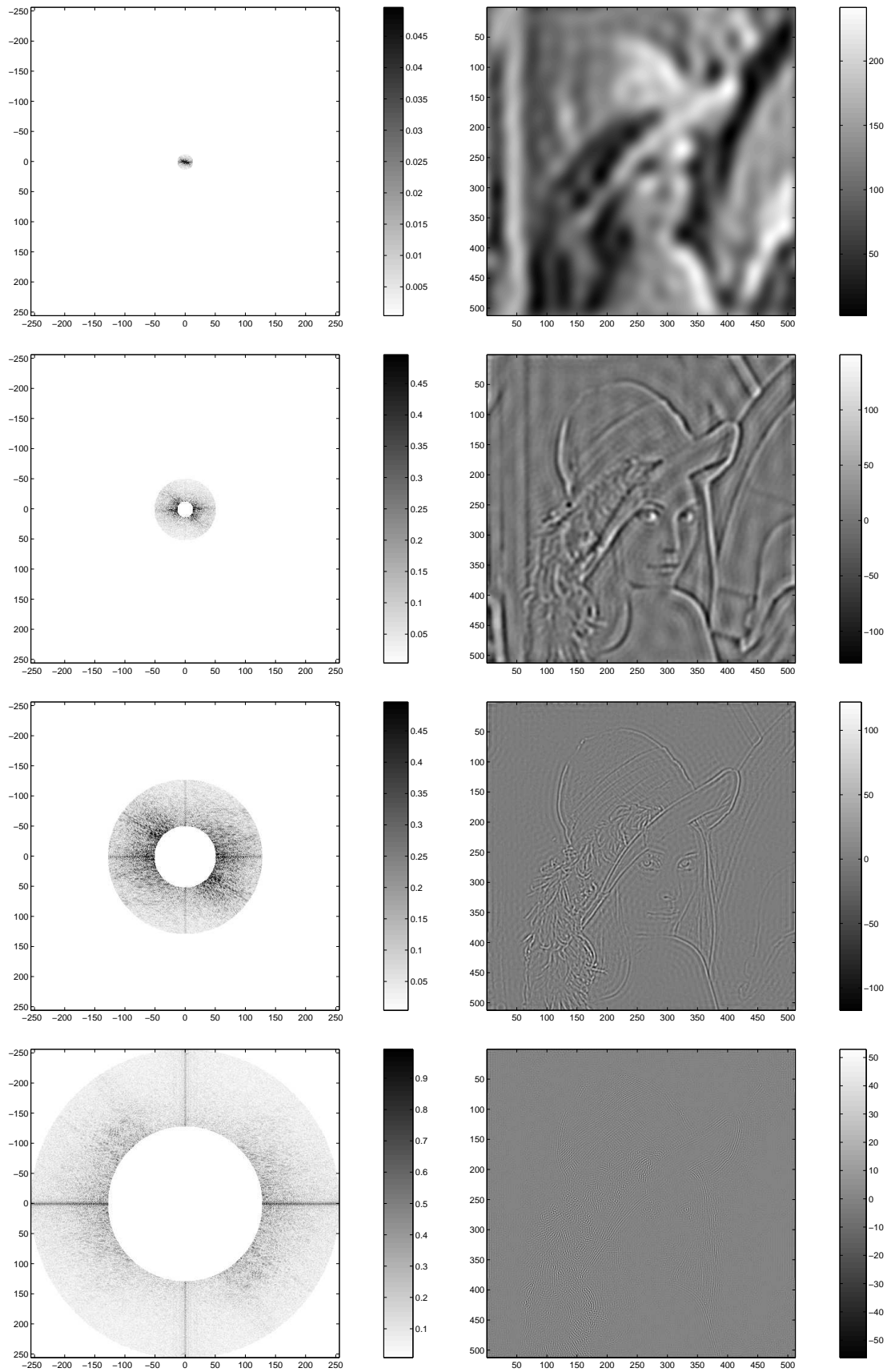


FIGURE 4.6: Reconstruction of Lena from masked Fourier coefficients



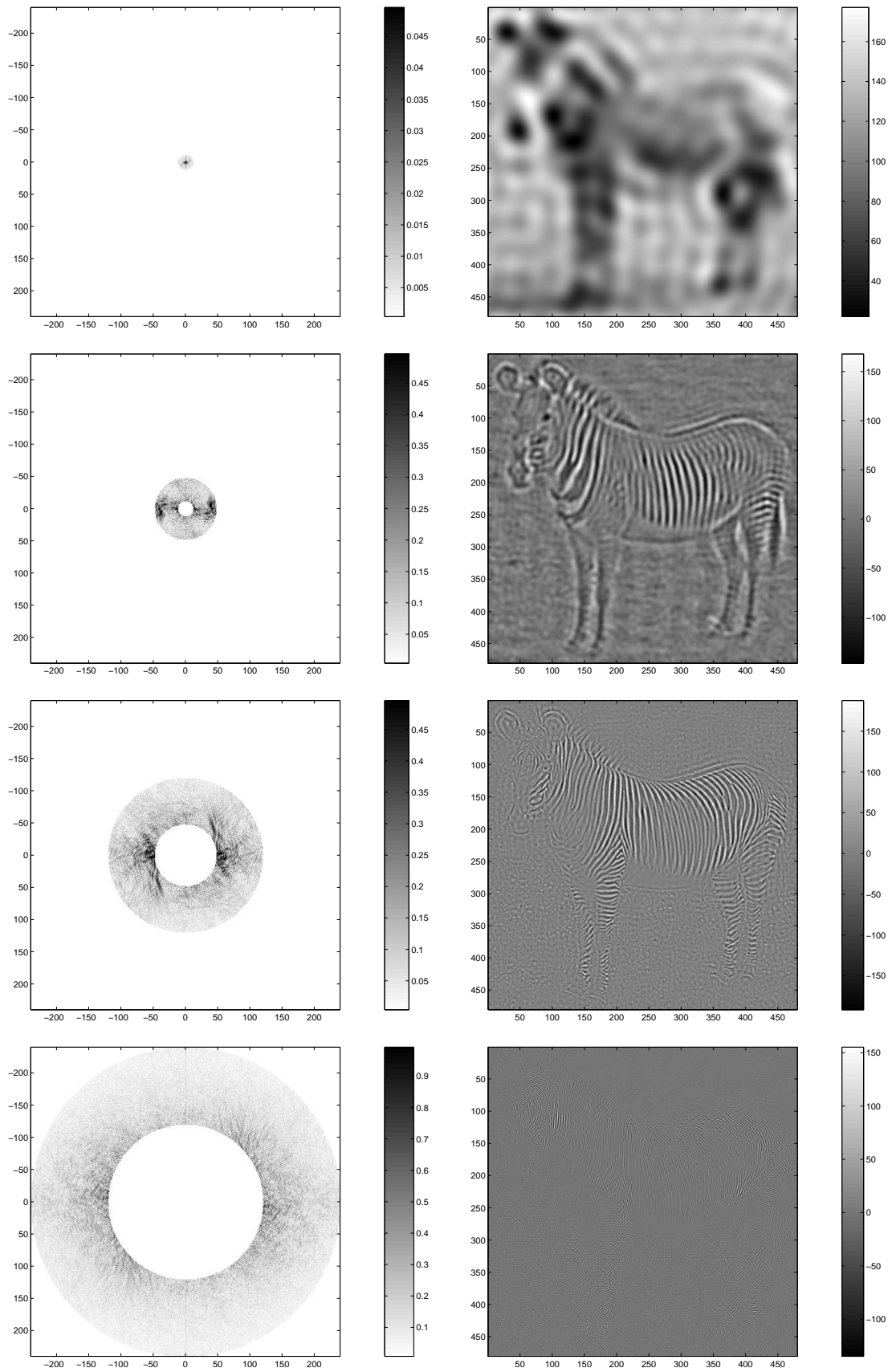


FIGURE 4.7: Reconstruction of zebra from masked Fourier coefficients

zero point. In the case of Lena, these radii amount to 13, 51, 128 and 256 pixels. Figure 4.7 shows the same procedure for the zebra image, where the radii of the mask boundaries amount to 12, 48, 120 and 240 pixels.

The upper two images in both Figure 4.6 and 4.7 show a restriction of the coefficients to the innermost circular area, covering all coefficients with frequencies of length lower than 5%. This implements a low-pass filter. It seems that the low frequencies are responsible for producing homogeneous areas on the image. The reconstruction shows the white, gray and black areas that are present in the original, but it lacks contour, sharpness and clear borders. So far, the difference between the reconstructed Lena and the original amounts to 0.2053, and that between the reconstructed zebra and the original to 0.3847.

The next two images in both Figure 4.6 and 4.7 show the inner ring on the Fourier domain and the corresponding reconstructed image. The mask covers the next bunch of higher frequencies with a length between 5% and 20% of the half image width. This ring implements a band-pass filter. It seems that these medium-ranged frequencies are responsible for producing raw borders and small areas. These images seem to purely add the boundaries of the areas we detected above, as the visible gray areas only correspond to zero values and therefore don't add up much to the complete image. The Lena image doesn't seem to possess coarse texture, whereas for the zebra we detected the broader line patterns on the neck and body of the animal. Adding these rings to the previous masks, the difference between the original Lena and its reconstruction reduces to 0.0979, and that between the zebra and its reconstruction to 0.2990.

The next outer rings on the Fourier domains cover even higher frequencies that reflect the boundaries and contours of the original images. Lena doesn't appear to have finer texture in the image, and for the zebra we detected the narrower line patterns on the body and legs of the animal. Adding these rings to the previous masks, the error for the Lena image becomes 0.0441, and that of the zebra 0.1317.

The outermost rings on the Fourier domains finally contain the frequencies responsible for constructing the clear sharpness and hard edges in the final image. Including these rings eventually reduces the error for the Lena image to 0.0115, and that of the zebra to 0.0248. This sequence of differences indicates that the Lena image is more concentrated in the lower frequencies than the zebra.

As a conclusion, we found out that the lowest frequencies are responsible for creating the areas, and they already determine the image to a high degree. Frequencies in the middle range construct boundaries, coarse details and coarse texture, and are responsible for adding some contour<sup>3</sup>. Higher frequencies are dominant where fine details, fine texture or edges occur. The highest frequencies are needed to add the last bit to the sharpness and hard edges of the image, but their contribution is only low.

This makes it possible to compare natural images to natural sounds such as music, and raises some ideas for doing windowed 2D Fourier transforms by looking at the 1D case: For music, the lower frequencies create melody, and for a good frequency discrimination one needs a narrow window on the frequency domain. But due to the

---

<sup>3</sup>The term *contrast* is reserved for describing the difference between the lowest and highest values.

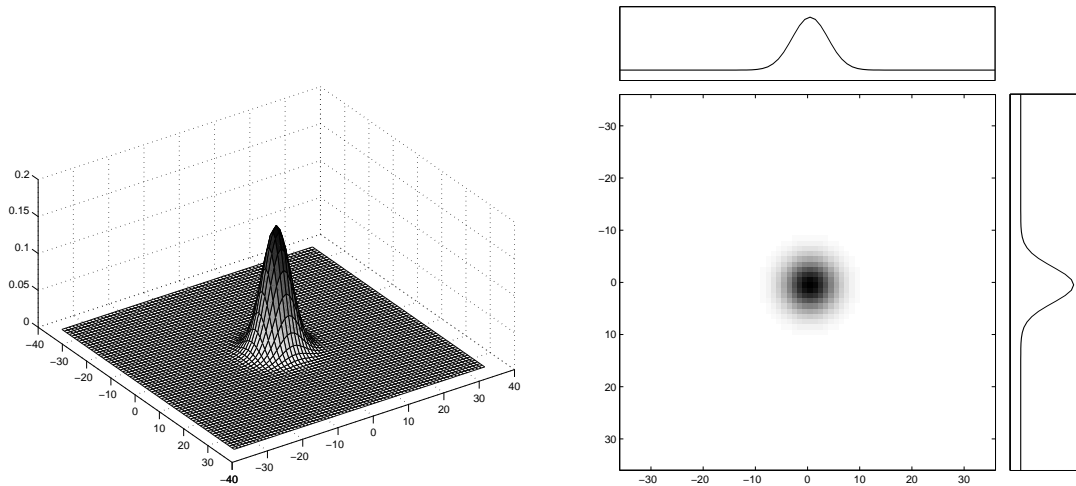


FIGURE 4.8: A 2D Gaussian function as tensor product of two 1D Gaussians. The plots could show the localization of a 1D Gaussian in the time-frequency space, or a 2D Gaussian in the image space, or its 2D Fourier transform in the Fourier space.

uncertainty principle these will have a wider support on the time domain, resulting in a low capability of temporal resolution. Luckily, higher temporal resolution is not needed for the lower frequencies, but the picture changes for the higher frequencies in music. These carry the transient or even percussive part and need a good temporal resolution, demanding a window with narrow support on the time domain. This will result in a wide window on the frequency domain, making a good distinction of frequencies hardly possible. Luckily, there's not a need for doing this. It's not of that interest what exact frequencies transient or percussive sounds contain, a coarse knowledge is enough as the main interest lies in the temporal behavior.

The same approach could now be taken for analyzing natural images: It seems to be important to distinguish low frequencies, as they highly determine the image, demanding a good resolution and therefore a narrow window on the 2D Fourier domain, resulting in a window with wide support on the image domain and therefore only unsharp localization on the image itself. However, the higher frequencies are responsible for creating boundaries and contour, and one needs a good local resolution for these. Frequency resolution will therefore be bad due to the resulting wide support on the Fourier domain, but a knowledge of the exact frequency values is not needed for describing contours.

If one is mainly interested in texture analysis, then the medium-ranged frequencies might be of importance. For texture classification, a better frequency resolution is reasonable and the localization might not be that important.

## 4.4 STFT of Discrete Images

Looking back to Figure 1.1 we were able to visualize the time-frequency behavior of 1D signals with the help of the STFT. As it is a function in time and frequency, its domain is two-dimensional, and a plot could be provided as a 3D plot or an image

plot. The STFTs in Figure 1.1 incorporated different windows and therefore show different aspects of the time-frequency behavior of the signals. Using the Gaussian as analyzing window minimizes the time-frequency fuzziness.

Now, if the signals are 2D themselves, their Fourier domain is 2D as well. The combined position-frequency<sup>4</sup> space is

$$\mathcal{G} \times \widehat{\mathcal{G}} = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2} \times \widehat{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}$$

and possesses four dimensions. This makes it rather difficult to visualize the four-dimensional (4D) position-frequency behavior of an image. It's already hard to visualize values that evolve in three dimensions, like the temperature behavior or fluid dynamics of the solar surface. In such cases, the values are usually represented as colors in a certain colormap, and the 3D space might be pervaded by slice planes that can be plotted more easily. But for the 4D position-frequency space of images, there seems to be no reasonable approach but showing the two 2D planes—position and frequency—separately.

Our image space  $\ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}) \cong \mathbb{C}^{L_1 \times L_2}$  is equipped with the inner product

$$\langle \mathbf{f}, \mathbf{g} \rangle := \sum_{u=0}^{L_1-1} \sum_{v=0}^{L_2-1} f(u, v) \overline{g(u, v)}$$

with associated norm  $\|\mathbf{f}\|^2 := \langle \mathbf{f}, \mathbf{f} \rangle$ . These equal the Frobenius inner product and norm if an image is seen as  $L_1 \times L_2$ -matrix. The translation operator is the cyclic position shift

$$(T_{\mathbf{k}} \mathbf{f})(u, v) = f(u - k_1 \bmod L_1, v - k_2 \bmod L_2)$$

with  $\mathbf{k} = (k_1, k_2) \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ . The modulation operator acts as

$$(M_{\boldsymbol{\ell}} \mathbf{f})(u, v) = f(u, v) e^{2\pi i(u\ell_1/L_1 + v\ell_2/L_2)}$$

with  $\boldsymbol{\ell} = (\ell_1, \ell_2) \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ , and we have the relation  $M_{\boldsymbol{\ell}} = \mathcal{F}^{-1} T_{\boldsymbol{\ell}} \mathcal{F}$  where  $\mathcal{F}$  is the 2D DFT of (4.1). The STFT of  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$  with respect to the window  $\mathbf{g} \in \mathbb{C}^{L_1 \times L_2}$  is now

$$(\mathcal{V}_{\mathbf{g}} \mathbf{f})(\mathbf{k}, \boldsymbol{\ell}) = \langle \mathbf{f}, M_{\boldsymbol{\ell}} T_{\mathbf{k}} \mathbf{g} \rangle$$

with  $(\mathbf{k}, \boldsymbol{\ell}) \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2} \times \widehat{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}$ . We'll write  $(\mathbf{x}, \boldsymbol{\omega})$  instead of  $(\mathbf{k}, \boldsymbol{\ell})$  to emphasize their meaning as position and frequency.

We want to keep in mind that the parameter  $\mathbf{x}$  in the STFT  $\mathcal{V}_{\mathbf{g}} \mathbf{f}(\mathbf{x}, \boldsymbol{\omega})$  is only a shift parameter for the window  $\mathbf{g}$ . If we consider the STFT as some subsequent procedure, we're slowly moving the window over the signal and compute the Fourier transform for each cutout, just as it is given by equation (1.17). Like in our approach to understand 2D frequencies, we could fix a certain shift  $\mathbf{x} = \mathbf{x}_0$  and look at the function  $\boldsymbol{\omega} \mapsto \mathcal{V}_{\mathbf{g}} \mathbf{f}(\mathbf{x}_0, \boldsymbol{\omega})$ , what provides the complete picture on the 2D Fourier

<sup>4</sup>In signal analysis, the notation  $\mathcal{G}$  vs.  $\widehat{\mathcal{G}}$  can be ignored and just indicates whether we're on the position or the frequency domain.

domain for that certain window position  $T_{\mathbf{x}_0} \mathbf{g}$ . Varying the fixed shifts  $\mathbf{x}$  will now change the look of the STFT as a function in  $\boldsymbol{\omega}$ .

The second possibility would be to fix the modulation parameter  $\boldsymbol{\omega} = \boldsymbol{\omega}_0$  and consider the STFT as a function  $\mathbf{x} \mapsto \mathcal{V}_{\mathbf{g}} \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_0)$ , providing a complete picture for the 2D signal space with respect to that single frequency. Varying  $\boldsymbol{\omega}$  will change the behavior of the STFT as a function in  $\mathbf{x}$ . If we look back at equation (1.19), we see that this corresponds to a convolution of the image with a modulated window function and a pointwise multiplication with the pure oscillation  $\mathbf{x} \mapsto e^{-2\pi i(\mathbf{x}, \boldsymbol{\omega})}$ .

By these approaches, the STFT of an image can be visualized as a bunch of single images, either showing the complete Fourier domain for certain shift positions of the 2D window, or showing the complete image domain for certain modulations. To get a full picture, the single Fourier images can be stacked to form a block image, where every Fourier image is placed according to the respectively considered shift position. The huge image would represent the position domain, and each block represents the whole frequency domain for a certain position of the window. The second approach would be the other way round: The convolution images are stacked to form a huge block image representing the frequency domain, and every block shows the complete position domain for the corresponding modulation.

Rather than really trying to construct these huge block images, we show single samples of the STFT of our zebra image. We mentioned previously that the STFT is too redundant anyway, so we try to construct such a picture later when we're doing Gabor analysis of images. In the case of the zebra, a  $480 \times 480$ -image, the full STFT image would have a size of  $(480 \times 480) \times (480 \times 480)$  what contains more than 53 billion entries.

What we haven't considered yet is the choice of an appropriate 2D window function. For a first approach we want to compute the STFT using the Gaussian window like we defined it in Definition 1.5.1. As it is an exponential function, the  $d$ -dimensional Gaussian  $\varphi(\mathbf{x}) = e^{-\pi \mathbf{x}^2} = e^{-\pi x_1^2} \dots e^{-\pi x_d^2}$  can be expressed as a tensor product  $\varphi(x_1, \dots, x_d) = \varphi(x_1) \otimes \dots \otimes \varphi(x_d)$  of  $d$  one-dimensional Gaussians, similar to our discussion of  $d$ -dimensional oscillations. We will discuss 2D windows in more detail in the next section.

Figure 4.9 shows an example for the first possibility of the two mentioned approaches: The three images on the left show different shifted 2D Gaussians applied to the zebra image. The cutouts almost look like modulated Gaussians, but the visible line patterns are those of the body of the animal. We could therefore expect that their 2D-FFTs almost look like shifted 2D Gaussians in the Fourier domain. The three images on the right show the corresponding 2D-FFTs of the images on the left. Indeed, we see two symmetrically positioned spots in the Fourier images, similar to shifted Gaussians, and the positions correspond to the orientations of the line patterns in the image domain. The symmetry is due to the real-valued signal. But additionally to the shifted spots we see an unshifted Gaussian in the center due to the fact that the line patterns in the original images did not simply oscillate between  $-1$  and  $1$ .

Figure 4.10 implements an example for the second possible approach: We see different 2D convolutions of the zebra image with modulated 2D Gaussians. Instead of simply showing shifted Gaussians in the Fourier image, it's more interesting to

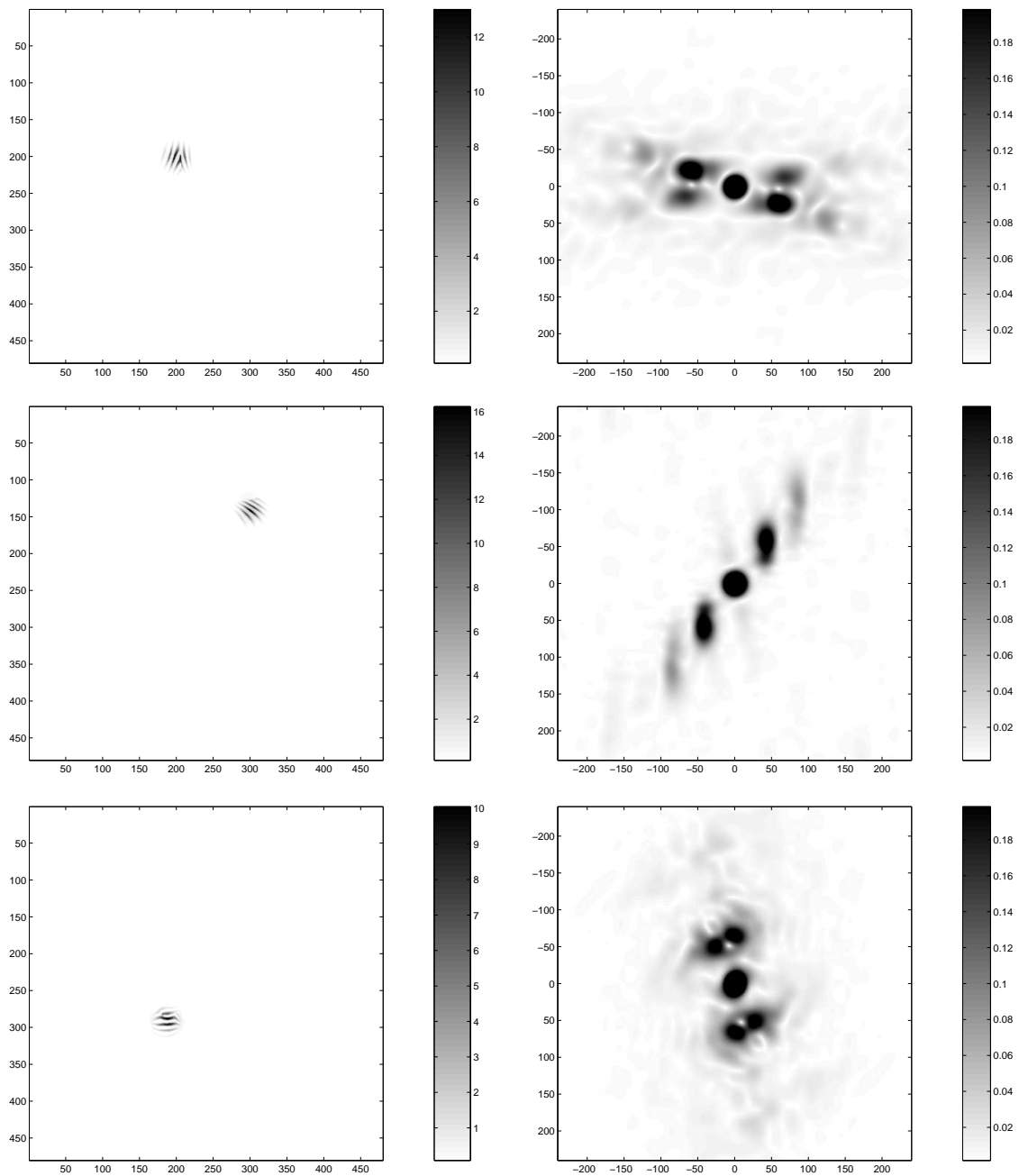


FIGURE 4.9: Localized Fourier transforms of zebra. The images on the left show the zebra windowed by Gaussians, using an inverted colormap. The images on the right show the corresponding 2D-FFTs; the Gaussians in the centers occur because the values in the line patterns do not simply oscillate between  $-1$  and  $1$ .

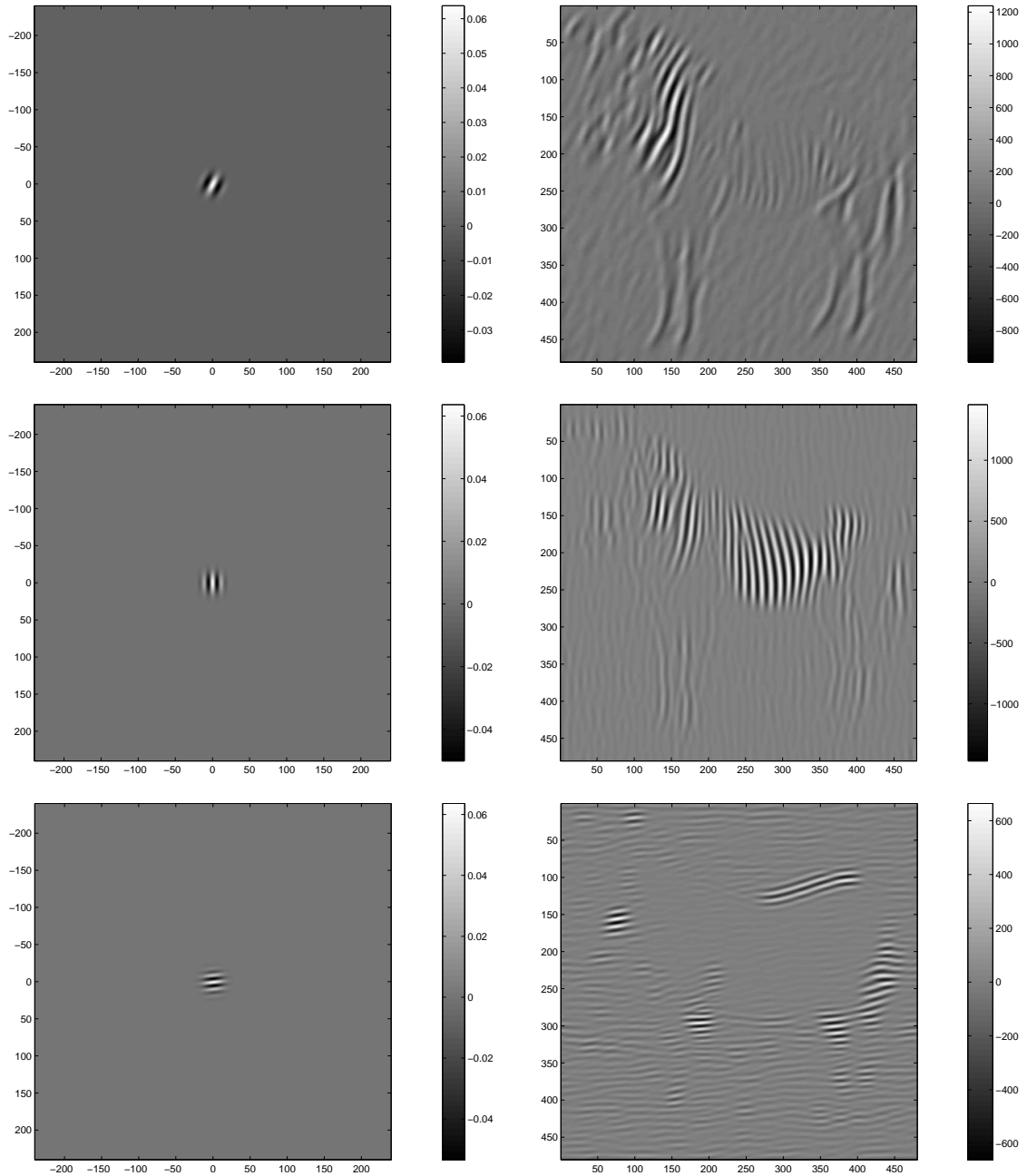


FIGURE 4.10: Convolutions of zebra with modulated 2D Gaussians. Rather than simply showing shifted Gaussians on the Fourier domain, it's more interesting to see the modulations that occur in the image domain due to the shifts. The images on the right show the inverse FFTs after windowing, yielding convolutions of the zebra with the corresponding modulated Gaussians on the left.

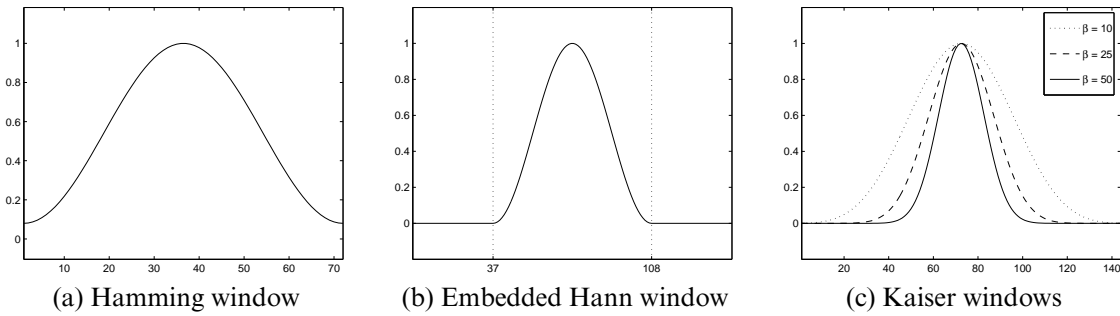


FIGURE 4.11: Common windows for 1D signal analysis. The Hamming window cannot be zero-extended, but the Hann window can. The Kaiser window allows a scaling parameter.

show the emerging modulations of the window in the image domain, visualized by the images on the left. The images on the right show the mentioned convolutions, indicating where and in what amount the zebra image responds to different pattern densities and orientations; the oscillation  $x \mapsto e^{-2\pi i(x, \omega)}$  of (1.19) was not applied.

Before we are looking at how to implement Gabor expansions of images, we have to look at how 2D windows can be constructed, as additional properties for 2D windows emerge that are not given in the 1D case.

## 4.5 2D Window Functions

In the previous section we computed (parts of) the STFT of an image by using the 2D Gaussian function as a window. It is given as a tensor product of two 1D Gaussians, and this provides one way to construct 2D windows by considering 1D windows. Therefore we will give a brief summary of commonly used 1D window functions and how one can construct 2D windows by taking tensor products.

**4.5.1 Definition** The Hamming window  $w^{\text{Hm}}$  of length  $L$  is defined as

$$w^{\text{Hm}}(j) := 0.53836 - 0.46164 \cos\left(\frac{2\pi j}{L-1}\right), \quad j \in \langle L \rangle.$$

The Hann<sup>5</sup> window  $w^{\text{Hn}}$  of length  $L$  is defined as

$$w^{\text{Hn}}(j) := \frac{1}{2} \left( 1 - \cos\left(\frac{2\pi j}{L-1}\right) \right), \quad j \in \langle L \rangle.$$

These two discrete-time windows look very similar, reaching their maximum at the center  $j = \lfloor \frac{L}{2} \rfloor$  and declining to the borders, with the difference that the Hamming window does not reach zero at the borders. The only possibility to decrease their essential support is to zero-embed them, what will therefore only work with the Hann

<sup>5</sup>Named after J. v. Hann and sometimes confusingly enough referred to as *Hanning* window, from “to Hann a signal”.



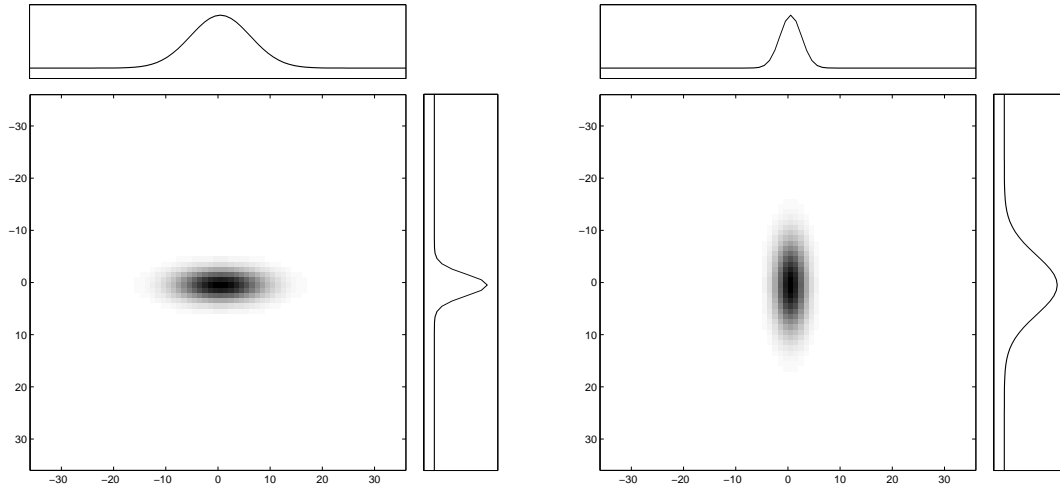


FIGURE 4.12: Separable 2D windows might be concentrated along the axes. The shown windows are tensor products of Gaussians that have been stretched by a factor of  $\frac{5}{3}$  or shrunk by  $\frac{3}{5}$ , respectively.

window. To zero-embed means to construct the window for some length  $L_w < L$  with  $L - L_w = 2n$  and define the actual window  $w$  of length  $L$  as

$$w(j) := \begin{cases} w^{\text{Hn}}(j - n) & j - n \in \langle L_w \rangle \\ 0 & \text{else.} \end{cases}$$

See subfigures 4.11a and 4.11b for graphs of a Hamming and a zero-embedded Hann window.

To define another commonly used window, we first need to mention Bessel functions.

**4.5.2 Definition** The Bessel function of the first kind of order  $\alpha \in \mathbb{Z}$ , denoted by  $J_\alpha(x)$ , is defined by its Taylor series expansion around  $x = 0$  as

$$J_\alpha(x) := \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m + \alpha + 1)} \left(\frac{x}{2}\right)^{2m+\alpha}.$$

**4.5.3 Definition** The Kaiser window  $w_\beta^K$  of length  $L$  and with shape parameter  $\beta \in \mathbb{R}$  is defined as

$$w_\beta^K(j) := \frac{J_0\left(\pi\beta\sqrt{1 - \left(\frac{2j}{L-1} - 1\right)^2}\right)}{J_0(\pi\beta)}, \quad j \in \langle L \rangle.$$

By construction, this window also peaks at the center  $j = \lfloor \frac{L}{2} \rfloor$  and decays exponentially to the borders. The window becomes narrower for increasing  $|\beta|$ . See subfigure 4.11c for graphs of Kaiser windows with different shape parameters.

As it is obvious that the previously defined windows look more or less like stretched or shrunk Gaussians, we will mostly restrict to Gaussian-like functions for our calculations, as these minimize the time-frequency localization.

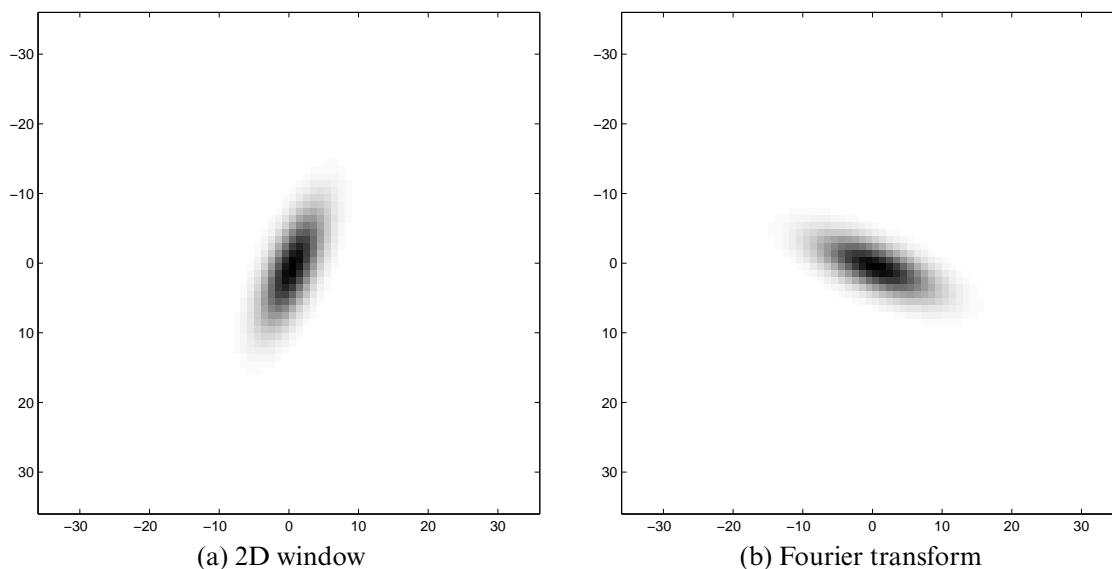


FIGURE 4.13: A non-separable 2D window and its Fourier transform. The window is a 2D Gaussian, stretched horizontally by  $\frac{5}{3}$  and shrunk vertically by  $\frac{3}{5}$ , then rotated by  $\frac{3}{8}\pi$ . Preserving the uncertainty principle, its Fourier transform shows increased localization in the direction where the window had narrow support and decreased localization where it had wide support.

Using these windows now makes it possible to define 2D windows by taking the tensor product of two 1D windows. I.e., to get a window  $w \in \ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2})$  one takes a window  $w_1 \in \ell^2(\mathbb{Z}_{L_1})$  of length  $L_1$  and a window  $w_2 \in \ell^2(\mathbb{Z}_{L_2})$  of length  $L_2$  and defines

$$w(j, k) := (w_1 \otimes w_2)(j, k) := w_1(j) w_2(k), \quad j \in \mathbb{Z}_{L_1}, k \in \mathbb{Z}_{L_2}.$$

Figure 4.12 shows what such separable windows could look like. Each of the two incorporated windows evolves along one of the two dimensions of their tensor product, and due to their appearance as symmetrically stretched or shrunk Gaussian-like functions, their tensor product always appears symmetric to the axes. The separable 2D window is narrow in that dimension that incorporated a narrow window, and it has wide support where a wide window was used. Corresponding to this, if the Fourier transform of such a 2D window is computed, it satisfies the uncertainty principle and will show a 2D window on the Fourier domain that has wide support in that dimension that used a narrow window, and it shows narrow support where a wide window evolved. If the 2D window was wide in both dimensions, its Fourier transform has narrow extension in both dimensions, and vice versa.

If one takes two Gaussians of length  $L_1$  and  $L_2$  with  $L_1 \neq L_2$ , their tensor product appears in elliptic shape, but is of course still invariant under the Fourier transform on  $\ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2})$ .

Using this procedure makes it possible to construct 2D windows that have different time-frequency resolutions for the horizontal or vertical dimension *only*. However, this might be rather limiting, as images are usually not separable in that sense. To

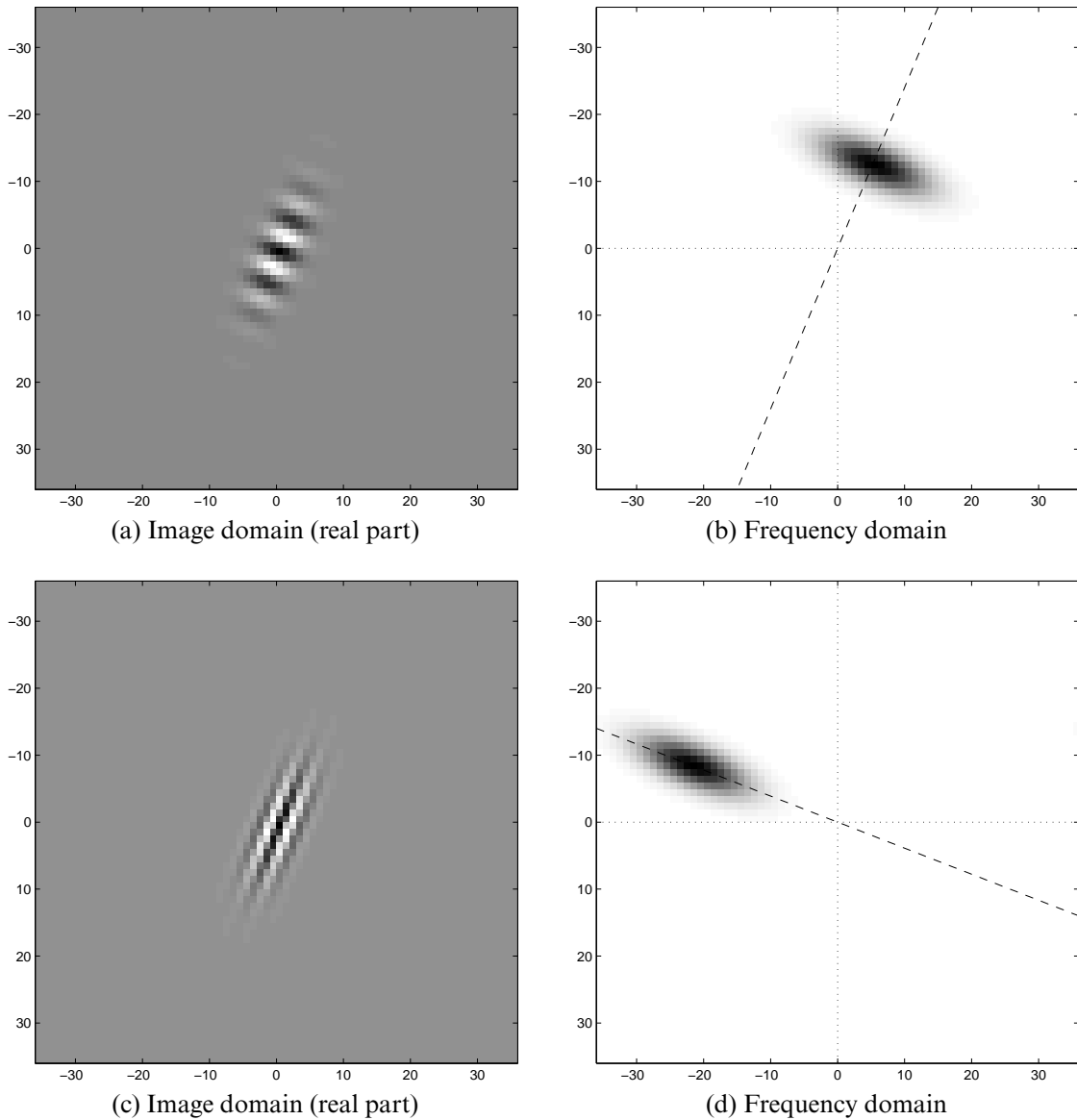


FIGURE 4.14: Frequency-shifts of a non-separable window. If the window has a higher frequency resolution in a certain direction (b), it loses the ability for good localization of the corresponding oscillations in the image domain (a). On the other hand, if frequency resolution is not important for a certain orientation (d), the window is capable of detecting correspondingly oriented edges (c).

optimally adapt a 2D window to a class of images, the need for a non-separable 2D window might arise, meaning that it can no longer be separated into its two dimensions and therefore cannot be composed as a tensor product of two 1D functions, but has to be defined as a whole on the image domain. Where separable windows allow easier computational procedures, this advantage will get lost for non-separable windows, what we will later examine in more detail.

Figure 4.13 shows an example for a non-separable 2D window. It is a Gaussian-like function defined on the image domain, but shows a stretch in a direction that is not oriented along one of the image dimensions. Additionally it has been shrunken by the same factor in the direction perpendicular to the first one, ensuring that it covers the same area on the image domain like the unmodified 2D Gaussian, shown in Figure 4.8. Because of that, its Fourier transform will not cover an overall smaller area in the Fourier domain than in the image domain. But due to the uncertainty principle, its extension will increase into that dimension where the window was narrow in the image domain, and it decreases where the window was stretched.

The shape of 2D windows can therefore be adapted to the dominating pattern orientations in the image signal. Choosing a certain stretch direction of the window decreases its position resolution in that direction and drops at the same time more frequencies of a certain orientation in the Fourier domain, enabling it to have a better frequency resolution for oscillations that evolve along the stretch direction of the window. On the opposite the perpendicular line patterns are not that much involved. Figure 4.14 shows how the window gets modulated in the image domain when selecting a frequency region in the direction that is best resolved by the window. The line patterns are then oriented in a way that makes it impossible to localize them in a well manner in the image domain. If for the other case frequency orientations are selected that cannot be well resolved, one gets the possibility of a good line pattern localization in the image domain.

Now that we have studied the STFT of images and the behavior of 2D window functions, we want to examine how Gabor expansions of images can be obtained. It will turn out that the case of separable windows yields a very handy computation of both the 2D Gabor transform and the dual 2D Gabor windows. And the possibility for separable or non-separable sampling subgroups will show an interplay with the (non-)separability of the possible windows.

# Chapter 5

## Image Representation by Gabor Expansion

We saw that Gabor analysis is a certain approach of doing localized Fourier analysis, where the main design freedom is the choice of (a) the time-frequency lattice and (b) the analysis prototype. The type of sampling lattice can be distinguished into a separable or non-separable case, where the first one can be described by the choice of lattice constants  $\alpha, \beta > 0$ . In the case of multidimensional signals we indicated in the previous chapter that another flavor of (non-)separability comes up, namely that of the analysis atom itself. However, in the language of group theory this distinction does not emerge at all, but for the ease of numerical implementations these terms become relevant.

It turns out that in the twofold-separable case, i.e. where the  $d$ -dimensional analysis window is a tensor product of  $d$  one-dimensional functions  $\mathbf{g} = g_1 \otimes \cdots \otimes g_d$  and the sampling lattice  $\Lambda$  is a product  $\Lambda = \prod_{i=1}^d \alpha_i \mathbb{Z}_{L_i} \times \prod_{i=1}^d \beta_i \mathbb{Z}_{L_i}$ , the dual Gabor window  $\boldsymbol{\gamma}$  is given as a product  $\boldsymbol{\gamma} = \gamma_1 \otimes \cdots \otimes \gamma_d$  as well, where the computation reduces to finding the 1D duals  $\gamma_i$  of the 1D atoms  $g_i$  with respect to the corresponding 2D time-frequency lattices  $\Lambda_i = \alpha_i \mathbb{Z}_{L_i} \times \beta_i \mathbb{Z}_{L_i}$ .

But as multidimensional signals such as 2D images are generally not separable, both the analysis window and the sampling lattice might have to be matched to the class of signals and might thus be non-separable. These cases are well-described in the language of group theory, cf. [FSC95], [FKPS96], [Grö98]. And as indicated in [Str97], the factorizations of the Gabor frame matrix mentioned in Section 3.3 can still be obtained, i.e., extending Theorem 3.3.5 to two dimensions means that  $S_g$  can be factorized into a block diagonal matrix where the blocks themselves are again block diagonal. Thus, the questions about how to obtain the dual Gabor frame or the synthesis prototype is answered by the theory.

Our aim here is to show how the results can be applied to the case of image signals. Gabor expansions of finite discrete 2D signals aren't quite different to those of finite discrete 1D signals. In a more general notation, there isn't a difference at all. But for taking the step to numerical implementations, the need for a more explicit notation raises.

## 5.1 2D Gabor Expansions

As already indicated in the previous chapter, images of size  $L_1 \times L_2$  evolve on the additive Abelian group  $\mathcal{G} = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ , and the image space  $\ell^2(\mathcal{G})$  is isomorphic to  $\mathbb{C}^{L_1 L_2}$ . The position-frequency space (PF-space) is

$$\mathcal{G} \times \widehat{\mathcal{G}} = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2} \times \widehat{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}.$$

In signal analysis, the notation  $\mathcal{G}$  vs.  $\widehat{\mathcal{G}}$  just indicates whether we're on the position or the frequency domain.

A Gabor system  $\mathcal{G}(\mathbf{g}, \Lambda)$  consists of TF-shifts  $M_{\ell} T_{\mathbf{k}} \mathbf{g}$  of a window  $\mathbf{g} \in \mathbb{C}^{L_1 \times L_2}$ , where  $(\mathbf{k}, \ell)$  are elements of a sampling subgroup  $\Lambda \trianglelefteq \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2} \times \widehat{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}$ . The Gabor coefficients of the image  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$  are defined as

$$c_{\mathbf{k}, \ell} := \langle \mathbf{f}, M_{\ell} T_{\mathbf{k}} \mathbf{g} \rangle_{\mathbb{F}}, \quad (\mathbf{k}, \ell) \in \Lambda.$$

The Gabor system generates a frame if there are constants  $0 < A \leq B < \infty$  such that

$$A \|\mathbf{f}\|_{\mathbb{F}}^2 \leq \sum_{(\mathbf{k}, \ell) \in \Lambda} |\langle \mathbf{f}, M_{\ell} T_{\mathbf{k}} \mathbf{g} \rangle_{\mathbb{F}}|^2 \leq B \|\mathbf{f}\|_{\mathbb{F}}^2 \quad \forall \mathbf{f} \in \mathbb{C}^{L_1 \times L_2}.$$

In the discrete case it's easy to check whether a Gabor system is a frame or not: It is a frame iff  $\overline{\text{span}}\{M_{\ell} T_{\mathbf{k}} \mathbf{g}\}_{(\mathbf{k}, \ell) \in \Lambda} = \mathbb{C}^{L_1 \times L_2}$ . A necessary condition is that we have more elements in  $\Lambda$  than the dimension of the signal space, and therefore we need  $L_1 L_2 \leq |\Lambda| \leq (L_1 L_2)^2$ . The redundancy of the Gabor frame is

$$\text{red}_{\Lambda} := \frac{|\Lambda|}{L_1 L_2} \geq 1.$$

The Gabor frame operator

$$S_{\mathbf{g}} \mathbf{f} := \sum_{(\mathbf{k}, \ell) \in \Lambda} \langle \mathbf{f}, M_{\ell} T_{\mathbf{k}} \mathbf{g} \rangle_{\mathbb{F}} M_{\ell} T_{\mathbf{k}} \mathbf{g}$$

commutes with TF-shifts determined by  $\Lambda$ , and its minimum and maximum eigenvalues equal the frame bounds  $A$  and  $B$ .

Just as mentioned previously, the dual Gabor frame incorporates the same TF-shifts, but applied to a dual window  $\boldsymbol{\gamma} \in \mathbb{C}^{L_1 \times L_2}$  such that the expansion

$$\mathbf{f} = \sum_{(\mathbf{k}, \ell) \in \Lambda} \langle \mathbf{f}, M_{\ell} T_{\mathbf{k}} \mathbf{g} \rangle_{\mathbb{F}} M_{\ell} T_{\mathbf{k}} \boldsymbol{\gamma} = \sum_{(\mathbf{k}, \ell) \in \Lambda} \langle \mathbf{f}, M_{\ell} T_{\mathbf{k}} \boldsymbol{\gamma} \rangle_{\mathbb{F}} M_{\ell} T_{\mathbf{k}} \mathbf{g}$$

holds for all  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$ . The existence of that dual is guaranteed by the theory of frames, and the calculation of the dual Gabor frame is done by the methods mentioned in Chapter 3 as well.

It turns out that the effort for obtaining Gabor expansions of images depends on the structure of the given 4D sampling lattice. A (fully) separable position-frequency lattice (PF-lattice) could be described by parameters  $\alpha_1, \alpha_2, \beta_1, \beta_2 > 0$  such that  $\alpha_i$

and  $\beta_i$  are divisors of  $L_i$  and indicate the position and frequency shift parameters, respectively. The set  $\Lambda$  itself is given as

$$\Lambda = \{(\mathbf{k}, \boldsymbol{\ell}) = (k_1, k_2, \ell_1, \ell_2) = (\alpha_1 u_1, \alpha_2 u_2, \beta_1 v_1, \beta_2 v_2) \mid u_i \in \langle \frac{L_i}{\alpha_i} \rangle, v_i \in \langle \frac{L_i}{\beta_i} \rangle\}.$$

It could therefore be described as a product  $\Lambda = \Lambda_1 \times \Lambda_2$  with  $\Lambda_i = \alpha_i \mathbb{Z}_{L_i} \times \beta_i \widehat{\mathbb{Z}_{L_i}}$ .

But what does a general 4D lattice look like? What's the structure of its adjoint lattice? One has to notice that for multidimensional signals the separability of a lattice could have various "depths": If it is a product  $\Lambda = \Lambda_1 \times \Lambda_2$  in the case of 2D signals, this doesn't necessarily mean that the sublattices  $\Lambda_i$  are separable themselves like  $\Lambda = \alpha_1 \mathbb{Z}_{L_1} \times \beta_1 \widehat{\mathbb{Z}_{L_1}} \times \alpha_2 \mathbb{Z}_{L_2} \times \beta_2 \widehat{\mathbb{Z}_{L_2}}$  as described above. Indeed, one of those sublattices could be non-separable, e.g.  $\Lambda_1 \trianglelefteq \mathbb{Z}_{L_1} \times \widehat{\mathbb{Z}_{L_1}}$ . Or furthermore, both  $\Lambda_1$  or  $\Lambda_2$  could be non-separable 2D lattices in their respective 2D TF-planes, but the 4D lattice is still a product  $\Lambda_1 \times \Lambda_2$ . So, one flavor of separability is that of looking at two independent signal spaces  $\mathbb{C}^{L_1}$  and  $\mathbb{C}^{L_2}$  on which the corresponding lattices  $\Lambda_1$  and  $\Lambda_2$  might still be arbitrary subgroups of the respective TF-planes.

However, there is another possibility for describing  $\Lambda$  as a product of subgroups, namely by separating the position domain  $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  from the frequency domain  $\widehat{\mathbb{Z}_{L_1}} \times \widehat{\mathbb{Z}_{L_2}}$ . I.e., a 4D PF-lattice could be a product  $\Lambda = \Omega \times \widehat{\Omega}$  with  $\Omega \trianglelefteq \mathcal{G} = \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  and  $\widehat{\Omega} \trianglelefteq \widehat{\mathcal{G}} = \widehat{\mathbb{Z}_{L_1}} \times \widehat{\mathbb{Z}_{L_2}}$ . These subgroups could both be arbitrary, or one or both could be separable, e.g.  $\Lambda = \alpha_1 \mathbb{Z}_{L_1} \times \alpha_2 \mathbb{Z}_{L_2} \times \widehat{\Omega}$  for non-separable  $\widehat{\Omega}$ .

And finally, there could of course be truly non-separable subgroups  $\Lambda \trianglelefteq \mathcal{G} \times \widehat{\mathcal{G}}$ . In arbitrary dimensions, this mixture of separability and non-separability increases correspondingly.

As already mentioned, there is a second notion of (non-)separability for multidimensional signals, namely that of the Gabor window, what adds another degree of complexity to multidimensional Gabor expansions. It seems that the possible combinations of both notions produce an order corresponding to increasing difficulty for computational implementations of 2D Gabor expansions. We include the fundamental cases for 1D signals in this list, as they are being reverted to:

- (1) 1D window on a separable (2D) TF-lattice (traditionally),
- (2) 1D window on a non-separable TF-lattice (covered in Chapter 3),
- (3) 2D separable window on a fully separable (4D) PF-lattice (reverts to case (1)),
- (4) 2D separable window on a partially non-separable PF-lattice (cases (1) or (2)),
- (5) 2D non-separable window on a fully separable PF-lattice,
- (6) 2D non-separable window on a partially or truly non-separable PF-lattice.

The following sections in this chapter reflect this ordered list, and we will show how to compute Gabor expansions for each of the mentioned cases. The traditional cases (1) and (2) have been covered in the previous chapters, and we start with case (3) of twofold separability.

## 5.2 Separable Atoms on Fully Separable Lattices

In this section we want to show why the case of a 2D separable window  $g = g_1 \otimes g_2$  and a fully separable PF-lattice

$$\Lambda = \Lambda_1 \times \Lambda_2 = \alpha_1 \mathbb{Z}_{L_1} \times \beta_1 \widehat{\mathbb{Z}_{L_1}} \times \alpha_2 \mathbb{Z}_{L_2} \times \beta_2 \widehat{\mathbb{Z}_{L_2}}$$

allows for efficient Gabor expansions. There is the question whether this case can be reduced to finding a dual 1D window  $\gamma_1$  for the 1D window  $g_1$  on the TF-lattice  $\Lambda_1 \trianglelefteq \mathbb{Z}_{L_1} \times \widehat{\mathbb{Z}_{L_1}}$  and a dual 1D window  $\gamma_2$  for the 1D window  $g_2$  on the TF-lattice  $\Lambda_2 \trianglelefteq \mathbb{Z}_{L_2} \times \widehat{\mathbb{Z}_{L_2}}$  and obtaining the dual 2D window  $\boldsymbol{\gamma}$  for  $g$  on the lattice  $\Lambda$  simply by  $\boldsymbol{\gamma} := \gamma_1 \otimes \gamma_2$ . Rather than trying to find a 2D Gabor system and trying to do such a separation, we want to see whether the product of two given 1D Gabor frames for two given signal spaces  $\mathbb{C}^{L_1}$  and  $\mathbb{C}^{L_2}$  could be a frame for the product space  $\mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2}$ . The following Lemma shows that this is possible in a general way.

**5.2.1 Lemma** *Let  $\mathcal{H} = \mathcal{H}_1 \widehat{\otimes} \mathcal{H}_2$  be the tensor product of two Hilbert spaces  $\mathcal{H}_1$  and  $\mathcal{H}_2$  that is a Hilbert space with respect to the inner product*

$$\langle e_1 \otimes e_2, f_1 \otimes f_2 \rangle_{\mathcal{H}} := \langle e_1, f_1 \rangle_{\mathcal{H}_1} \langle e_2, f_2 \rangle_{\mathcal{H}_2} \quad \forall e_i, f_i \in \mathcal{H}_i.$$

*Let  $\{e_m\}_{m \in \mathbb{Z}} \subseteq \mathcal{H}_1$  be a frame for  $\mathcal{H}_1$  and  $\{f_n\}_{n \in \mathbb{Z}} \subseteq \mathcal{H}_2$  a frame for  $\mathcal{H}_2$ . Then the collection  $\{e_m \otimes f_n\}_{m,n \in \mathbb{Z}}$  is a frame for  $\mathcal{H}_1 \otimes \mathcal{H}_2$ .*

**Proof.** [FG94, 8.18] Let  $S_1$  and  $S_2$  be the frame operators of the two frames, i.e.,  $S_1 g = \sum_{m \in \mathbb{Z}} \langle g, e_m \rangle_{\mathcal{H}_1} e_m$  and  $S_2 h = \sum_{n \in \mathbb{Z}} \langle h, f_n \rangle_{\mathcal{H}_2} f_n$  for  $g \in \mathcal{H}_1$  and  $h \in \mathcal{H}_2$ . If we denote the frame operator on  $\mathcal{H}_1 \otimes \mathcal{H}_2$  with respect to the system  $\{e_m \otimes f_n\}_{m,n \in \mathbb{Z}}$  by  $S$  and apply it to  $g \otimes h \in \mathcal{H}_1 \otimes \mathcal{H}_2$ , we get

$$\begin{aligned} S(g \otimes h) &= \sum_{m,n \in \mathbb{Z}} \langle g \otimes h, e_m \otimes f_n \rangle_{\mathcal{H}} e_m \otimes f_n \\ &= \sum_{m,n \in \mathbb{Z}} \langle g, e_m \rangle_{\mathcal{H}_1} \langle h, f_n \rangle_{\mathcal{H}_2} e_m \otimes f_n \\ &= \left( \sum_{m \in \mathbb{Z}} \langle g, e_m \rangle_{\mathcal{H}_1} e_m \right) \otimes \left( \sum_{n \in \mathbb{Z}} \langle h, f_n \rangle_{\mathcal{H}_2} f_n \right) = S_1 g \otimes S_2 h. \end{aligned}$$

Thus we can define  $S(g \otimes h) = (S_1 \otimes S_2)(g \otimes h) := S_1 g \otimes S_2 h$ , and the frame operator is invertible by  $S^{-1} = S_1^{-1} \otimes S_2^{-1}$ . This implies that the corresponding system is indeed a frame with bounds  $A_1 A_2$  and  $B_1 B_2$  if  $A_i$  and  $B_i$  are the respective frame bounds in  $\mathcal{H}_i$ .  $\square$

**5.2.2 Corollary** *If  $\mathcal{H}_1 = \mathbb{C}^{L_1}$  and  $\mathcal{H}_2 = \mathbb{C}^{L_2}$  in Lemma 5.2.1 and  $\{e_m\}_{m \in \{N_1\}} \subseteq \mathbb{C}^{L_1}$  and  $\{f_n\}_{n \in \{N_2\}} \subseteq \mathbb{C}^{L_2}$  are frames for  $\mathbb{C}^{L_1}$  and  $\mathbb{C}^{L_2}$ , respectively, then the sequence  $\{e_m \otimes f_n\}_{(m,n) \in \{N_1\} \times \{N_2\}}$  is a frame for  $\mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2}$ , where  $(g \otimes h)(j, k) := g(j) h(k)$  for  $g \in \mathbb{C}^{L_1}$  and  $h \in \mathbb{C}^{L_2}$ . The joint redundancy is  $\frac{N_1 N_2}{L_1 L_2} \geq 1$ .*



Proof. We only have to show that the tensor product of elements in the Hilbert spaces really corresponds to the tensor product of vectors. For  $f_i, g_i \in \mathbb{C}^{L_i}$  we have

$$\begin{aligned}
 \langle f_1, g_1 \rangle_{\mathbb{C}^{L_1}} \langle f_2, g_2 \rangle_{\mathbb{C}^{L_2}} &= \left( \sum_{j=0}^{L_1-1} f_1(j) \overline{g_1(j)} \right) \left( \sum_{k=0}^{L_2-1} f_2(k) \overline{g_2(k)} \right) \\
 &= \sum_{j=0}^{L_1-1} \sum_{k=0}^{L_2-1} f_1(j) f_2(k) \overline{g_1(j) g_2(k)} \\
 &= \sum_{j=0}^{L_1-1} \sum_{k=0}^{L_2-1} (f_1 \otimes f_2)(j, k) \overline{(g_1 \otimes g_2)(j, k)} \\
 &= \langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle_{\mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2}}.
 \end{aligned}$$

The redundancy value is clear due to the number of elements in the frame, what is  $N_1 N_2$ , and the dimension of  $\mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2} \cong \mathbb{C}^{L_1 L_2}$ .  $\square$

These results hold for arbitrary frames, and therefore for Gabor frames as well. As our image space is such a tensor product, we could define 2D Gabor windows  $\mathbf{g} \in \mathbb{C}^{L_1 \times L_2}$  by  $\mathbf{g} = g_1 \otimes g_2$  for  $g_i \in \mathbb{C}^{L_i}$ . As we're looking at the case where  $\Lambda = \Lambda_1 \times \Lambda_2$ , we take two Gabor frames  $\{g_{k_i, \ell_i}^{(i)}\}_{(k_i, \ell_i) \in \Lambda_i} := \{M_{\ell_i} T_{k_i} g_i\}_{(k_i, \ell_i) \in \Lambda_i} \subseteq \mathbb{C}^{L_i}$  with frame operators  $S_i$  and use the set of products  $\{g_{k_1, \ell_1}^{(1)} \otimes g_{k_2, \ell_2}^{(2)}\}_{(\mathbf{k}, \boldsymbol{\ell}) \in \Lambda} \subseteq \mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2}$  as frame for the image space with frame operator  $S_1 \otimes S_2$ .

The question remains whether this product of Gabor frames is really a *Gabor* frame for the image space, and not just any kind of frame. We have to show that the product of 1D TF-shifts form 2D position-frequency shifts (PF-shifts) of the 2D window  $\mathbf{g}$ . Trivially, this is the case for position shifts, as 1D shifts modulo  $L_1$  and  $L_2$  become corresponding 2D shifts on the image domain  $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ . And for the modulations we had already mentioned in the previous chapter that 2D modulations can be expressed as tensor products of 1D modulations. As a roundup, we really have

$$M_{\ell_1} T_{k_1} g_1 \otimes M_{\ell_2} T_{k_2} g_2 = M_{(\ell_1, \ell_2)} T_{(k_1, k_2)} (g_1 \otimes g_2) \quad \forall (k_1, k_2), (\ell_1, \ell_2) \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$$

as building blocks for our 2D Gabor frame.

The canonical dual of  $\mathbf{g}$  with respect to that frame is now given as  $\boldsymbol{\gamma}^\circ = S^{-1} \mathbf{g} = S_1^{-1} g_1 \otimes S_2^{-1} g_2 = \boldsymbol{\gamma}_1^\circ \otimes \boldsymbol{\gamma}_2^\circ$ . The calculation of 1D dual windows on separable TF-lattices had been well-implemented in MATLAB by NuHAG. But we didn't yet have an eye on how to efficiently obtain the Gabor coefficients of an image  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$  by

$$c_{\mathbf{k}, \boldsymbol{\ell}} := \langle \mathbf{f}, M_{\boldsymbol{\ell}} T_{\mathbf{k}} \mathbf{g} \rangle_{\mathbb{F}}.$$

What does the Gabor matrix  $C_{\mathbf{g}}$  look like if it is to be applied to an image  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$  that could be considered as an  $L_1 \times L_2$ -matrix? Sure,  $\mathbf{f}$  must be seen as a vector in  $\mathbb{C}^{L_1 L_2}$  and  $C_{\mathbf{g}}$  as an  $N_1 N_2 \times L_1 L_2$ -matrix if the number of elements in the 2D frame is  $N_1 N_2$  and the coefficient vector is  $c \in \mathbb{C}^{N_1 N_2}$ . In general,  $\mathbf{f}$  cannot be assumed to be separable, thus the only thing that seems to ease our computation is the structure

$$c_{\mathbf{k}, \boldsymbol{\ell}} = \langle \mathbf{f}, M_{\ell_1} T_{k_1} g_1 \otimes M_{\ell_2} T_{k_2} g_2 \rangle_{\mathbb{F}}.$$

If we think of the 1D case with some  $f \in \mathbb{C}^L$  and a general frame  $\{g_j\}_{j \in \langle N \rangle} \subseteq \mathbb{C}^L$ , the coefficients are obtained by

$$c = Cf = (\langle f, g_j \rangle)_{j \in \langle N \rangle} = (c_j)_{j \in \langle N \rangle},$$

and for Gabor frames,  $c = (c_{k,\ell})_{(k,\ell) \in \Lambda}$  with  $\Lambda \trianglelefteq \mathbb{Z}_L \times \widehat{\mathbb{Z}}_L$  is actually a coefficient matrix in  $\mathbb{C}^{L \times L}$  with  $|\Lambda| = N \leq L^2$  non-zero entries. But due to simply stacking the vectors  $\{g_{k,\ell}\}_{(k,\ell) \in \Lambda} = \{g_j\}_{j \in \langle N \rangle} \subseteq \mathbb{C}^L$  in the coefficient matrix

$$C = \begin{pmatrix} g_0^* \\ \vdots \\ g_{N-1}^* \end{pmatrix} \in \mathbb{C}^{N \times L}, \quad (5.1)$$

one just gets a “flat”  $c \in \mathbb{C}^N$ . In our 2D case, the Gabor coefficient even consists of entries  $c_{\mathbf{k},\ell} = c_{k_1,k_2,\ell_1,\ell_2}$ . We also want to take the approach by using general frames  $\{g_m\}_{m \in \langle N_1 \rangle} \subseteq \mathbb{C}^{L_1}$  and  $\{h_n\}_{n \in \langle N_2 \rangle} \subseteq \mathbb{C}^{L_2}$ , and look at the product frame  $\{g_m \otimes h_n\}_{m,n}$  for  $\mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2}$ . We also reduce the coefficient  $c = (c(m,n))_{m,n} \in \mathbb{C}^{N_1 N_2}$  to a vector of form

$$c = (c(0,0), c(0,1), \dots, c(0, N_2 - 1), c(1,0), \dots, c(1, N_2 - 1), \dots, c(N_1 - 1, 0), \dots, c(N_1 - 1, N_2 - 1))^T$$

such that we can try to find the corresponding coefficient matrix  $C \in \mathbb{C}^{N_1 N_2 \times L_1 L_2}$  that can be applied to  $f \in \mathbb{C}^{L_1 L_2}$ , where

$$f = (f(0,0), \dots, f(0, L_2 - 1), f(1,0), \dots, f(L_1 - 1, L_2 - 1))^T. \quad (5.2)$$

Now we can look at the  $(m,n)$ -th, or rather,  $(mN_2 + n)$ -th entry of the coefficient:

$$\begin{aligned} (Cf)_{m,n} &= c(m,n) = \langle f, g_m \otimes h_n \rangle_{\mathbb{C}^{L_1 L_2}} \\ &= \sum_{u=0}^{L_1-1} \sum_{v=0}^{L_2-1} f(u,v) \overline{(g_m \otimes h_n)(u,v)} \\ &= \sum_{u=0}^{L_1-1} \sum_{v=0}^{L_2-1} f(u,v) \overline{g_m(u) h_n(v)}. \end{aligned} \quad (5.3)$$

Now that we are able to split the indices  $u$  and  $v$  for the frame elements, we can consider the order in (5.2) and get

$$(Cf)_{m,n} = \overline{(g_m(0) h_n^*)} \overline{(g_m(1) h_n^*)} \cdots \overline{(g_m(L_1 - 1) h_n^*)} f = (C)_{m,n} f,$$

where  $(C)_{m,n}$  is the  $(m,n)$ -th or  $(mN_2 + n)$ -th line of  $C$  and contains  $L_1 L_2$  entries. The line vectors  $\{h_n^*\}_{n \in \langle N_2 \rangle}$  form the frame matrix  $C_2 \in \mathbb{C}^{N_2 \times L_2}$  like in (5.1). If we look at the range of  $N_2$  lines  $\{(m,0), \dots, (m, N_2 - 1)\}$ , we are able to express the corresponding segment of  $C$  as

$$(C)_{m;n \in \langle N_2 \rangle} = \overline{(g_m(0) C_2} \overline{(g_m(1) C_2} \cdots \overline{(g_m(L_1 - 1) C_2)},$$

what finally gives us the result that the frame matrix of the product frame is the Kronecker product

$$C = C_1 \otimes C_2 \in \mathbb{C}^{N_1 N_2 \times L_1 L_2}$$

for  $C_i \in \mathbb{C}^{N_i \times L_i}$ .

Thus, one possibility for obtaining the Gabor coefficients of a product frame is calculating the product  $C_1 \otimes C_2$  and apply it to the image  $f \in \mathbb{C}^{L_1 L_2}$ . If we're taking our zebra test image of size  $480 \times 480$ , that means we'd have two frame matrices of size  $N_1 \times 480$  and  $N_2 \times 480$ , and the *minimal* redundancy of 1 makes the frame matrix of the product frame a  $(480 \times 480) \times (480 \times 480)$ -matrix, what contains more than 53 billion entries. If we use double precision for numerical computations, every entry takes 8 bytes, and so we'd end up with a huge matrix allocating *at least* more than 395 Gigabyte, what is way too much for today's computers. Contrary to this, storing  $C_1$  and  $C_2$  alone would just take about 3.5 Megabyte for minimal redundancy. And we're not even thinking of doing a pseudoinversion of the huge matrix, whereas it is enough to pseudoinvert the matrices  $C_i$ , as we'll see below.

Nevertheless, we want to see whether we can compute  $c = (C_1 \otimes C_2) f$  in a cheaper way by applying the frame matrices  $C_i$  without computing their Kronecker product. As images aren't stored as vectors  $f \in \mathbb{C}^{L_1 L_2}$  but rather as matrices  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$  in numerical software like MATLAB or Octave, we could try to get the coefficient  $c = (c(m, n))_{m,n} \in \mathbb{C}^{N_1 \times N_2}$  more directly.

**5.2.3 Proposition** *Given two frames  $\{g_m\}_{m \in \langle N_1 \rangle} \subseteq \mathbb{C}^{L_1}$  and  $\{h_n\}_{n \in \langle N_2 \rangle} \subseteq \mathbb{C}^{L_2}$  with frame matrices  $C_i \in \mathbb{C}^{N_i \times L_i}$ , then the frame coefficient  $c \in \mathbb{C}^{N_1 \times N_2}$  for the image  $f \in \mathbb{C}^{L_1 \times L_2}$  with respect to the product frame  $\{g_m \otimes h_n\}_{(m,n) \in \langle N_1 \rangle \times \langle N_2 \rangle}$  is given as*

$$\begin{aligned} c &= C_1 \mathbf{f} C_2^T = \\ &= \begin{pmatrix} \overline{g_0(0)} & \cdots & \overline{g_0(L_1-1)} \\ \vdots & & \vdots \\ \overline{g_{N_1-1}(0)} & \cdots & \overline{g_{N_1-1}(L_1-1)} \end{pmatrix} \begin{pmatrix} f(0,0) & \cdots & f(0,L_2-1) \\ \vdots & & \vdots \\ f(L_1-1,0) & \cdots & f(L_1-1,L_2-1) \end{pmatrix} \begin{pmatrix} \overline{h_0(0)} & \cdots & \overline{h_{N_2-1}(0)} \\ \vdots & & \vdots \\ \overline{h_0(L_2-1)} & \cdots & \overline{h_{N_2-1}(L_2-1)} \end{pmatrix}. \end{aligned} \quad (5.4)$$

*Proof.*  $C_1 \mathbf{f}$  is a matrix in  $\mathbb{C}^{N_1 \times L_2}$ , and its  $m$ -th line contains the  $L_2$  entries

$$(C_1 \mathbf{f})_m = \left( \sum_{u=0}^{L_1-1} f(u,0) \overline{g_m(u)} \quad \cdots \quad \sum_{u=0}^{L_1-1} f(u,L_2-1) \overline{g_m(u)} \right).$$

If it is applied to the  $n$ -th column of  $C_2^T \in \mathbb{C}^{L_2 \times N_2}$ , we get the coefficient entry

$$\begin{aligned} c(m,n) &= (C_1 \mathbf{f})_m (C_2)_n^T \\ &= \sum_{v=0}^{L_2-1} \sum_{u=0}^{L_1-1} f(u,v) \overline{g_m(u) h_n(v)} \\ &= \langle \mathbf{f}, g_m \otimes h_n \rangle_{\mathbb{F}} \end{aligned}$$

due to equation (5.3), yielding that the coefficient matrix really contains the desired entries.  $\square$

Note that similar thoughts reveal the fact that the 2D-DFT (4.1) of an image  $f \in \mathbb{C}^{L_1 \times L_2}$  can be obtained by the matrix multiplication

$$\mathcal{F}f = F_{L_1} f F_{L_2} \in \mathbb{C}^{L_1 \times L_2}, \quad (5.5)$$

where  $F_{L_i} \in \mathbb{C}^{L_i \times L_i}$  are the (symmetric) Fourier matrices (3.10) of order  $L_i$ .

We saw in Chapter 3 that if the synthesis operation is to be done by  $f = C^*c$  for given  $f \in \mathbb{C}^L$  and a frame  $C \in \mathbb{C}^{N \times L}$ , one solution is obtained by  $c = (C^*)^\dagger f$  with a right-inverse for  $C^*$  such that  $I_L = SS^{-1} = C^*C(C^*C)^{-1} = C^*(C^*)^\dagger$ , making the pseudoinverse of the synthesis operator the matching analysis operator.  $C^*(C^*)^\dagger$  is the orthogonal projection onto the range of the desired synthesis operator. One notices that due to  $(C^*)^\dagger = (C^\dagger)^*$  we already have  $I_L = (C^\dagger C)^* = C^\dagger C$ , the orthogonal projection onto  $\text{ran } C^\dagger$  [Chr03, 1.5.1]. Thus, the role of the operators can be interchanged, meaning that  $C^\dagger$  is the matching synthesis operator for the analysis operator  $C$ .

If we again interpret signals  $f \in \mathbb{C}^{L_1} \otimes \mathbb{C}^{L_2}$  as  $f \in \mathbb{C}^{L_1 L_2}$  and take a product frame  $\{g_m \otimes h_n\}_{m,n}$  with analysis operator  $C_1 \otimes C_2$ , we get  $I_{L_1 L_2} = C^\dagger(C_1 \otimes C_2)$  and  $I_{L_1 L_2} = I_{L_1} \otimes I_{L_2} = (C_1^\dagger C_1) \otimes (C_2^\dagger C_2)$ , yielding that the matching synthesis operator is  $C^\dagger = C_1^\dagger \otimes C_2^\dagger$ . Due to Proposition 5.2.3, we can thus reconstruct  $f \in \mathbb{C}^{L_1 \times L_2}$  by

$$f = (C_1^\dagger C_1) f (C_2^\dagger C_2)^\top = C_1^\dagger c (C_2^\dagger)^\top \quad (5.6)$$

because  $c = C_1 f C_2^\top$  is in the range of the corresponding analysis operator.

These results were derived for products of general frames and therefore also hold for products of Gabor frames. Given two Gabor frames  $\{M_{l_i} T_{k_i} g_i\}_{(k_i, l_i) \in \Lambda_i} \subseteq \mathbb{C}^{L_i}$  on subgroups  $\Lambda_i \trianglelefteq \mathbb{Z}_{L_i} \times \widehat{\mathbb{Z}_{L_i}}$  and with analysis operators  $C_{g_i}$ , we get their synthesis operators by  $C_{g_i}^\dagger = C_{\gamma_i^\circ}^*$  with  $\gamma_i^\circ := S_{g_i}^{-1} g_i$ . The product of those two frames is the Gabor frame  $\{M_\ell T_k \mathbf{g}\}_{(k, \ell) \in \Lambda_1 \times \Lambda_2}$  consisting of PF-shifts of the window  $\mathbf{g} = g_1 \otimes g_2 \in \mathbb{C}^{L_1 \times L_2}$  on the lattice  $\Lambda = \Lambda_1 \times \Lambda_2$ . The dual window to  $\mathbf{g}$  is given by  $\boldsymbol{\gamma}^\circ := \gamma_1^\circ \otimes \gamma_2^\circ$ . Due to (5.4) and (5.6), the 2D Gabor analysis operation for the image  $f \in \mathbb{C}^{L_1 \times L_2}$  is obtained by

$$c = C_{g_1} f C_{g_2}^\top \quad (5.7)$$

and a possible reconstructing synthesis operation by

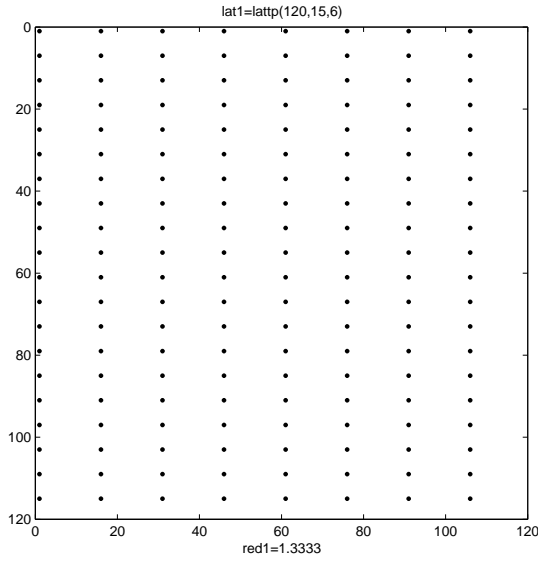
$$f = C_{\gamma_1^\circ}^* c (C_{\gamma_2^\circ}^*)^\top = \overline{C_{\gamma_1^\circ}^\top} c \overline{C_{\gamma_2^\circ}^\top}, \quad (5.8)$$

yielding that it is enough to obtain the two duals  $\gamma_i^\circ$ .

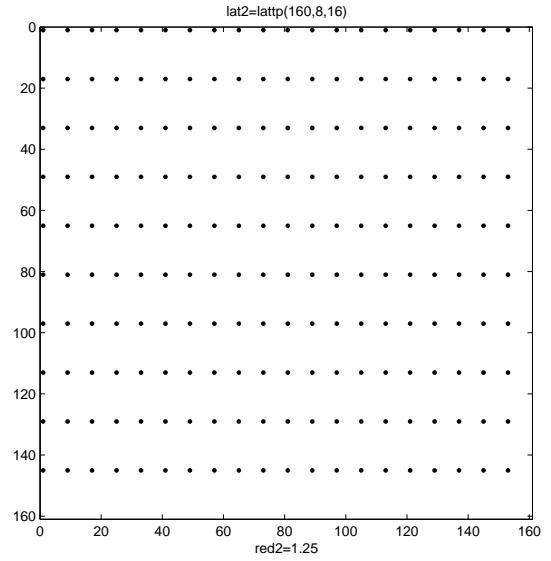
Figure 5.1 shows the construction and look of the separable dual 2D window of a 2D Gaussian window on a fully separable PF-lattice.

### 5.2.1 Efficient Gabor Expansion by Sampled STFT

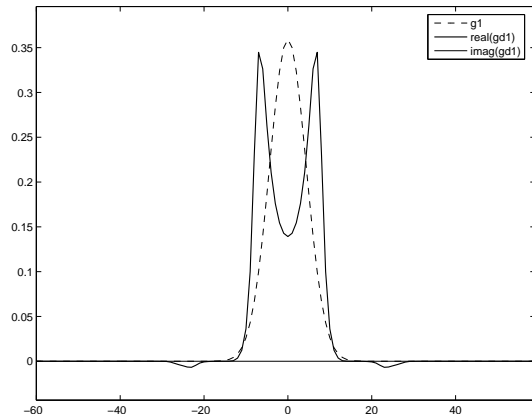
In the case of a separable 2D atom and a fully separable PF-lattice we can make use of the fast 1D STFT implementation by NuHAG to obtain the Gabor analysis coefficient  $c = C_{g_1} f C_{g_2}^\top$  and the Gabor reconstruction  $f = C_{\gamma_1^\circ}^* c (C_{\gamma_2^\circ}^*)^\top$  for a given



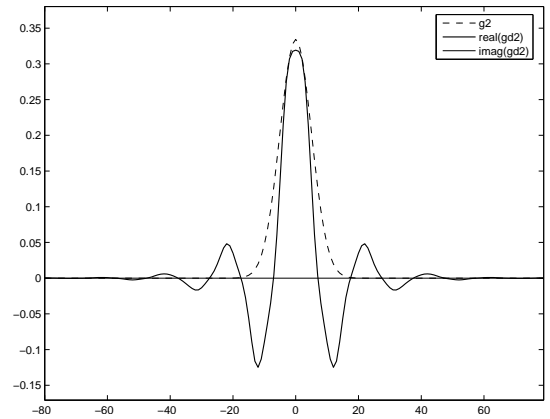
(a) Lattice  $\Lambda_1 = 15\mathbb{Z}_{120} \times 6\mathbb{Z}_{120}$



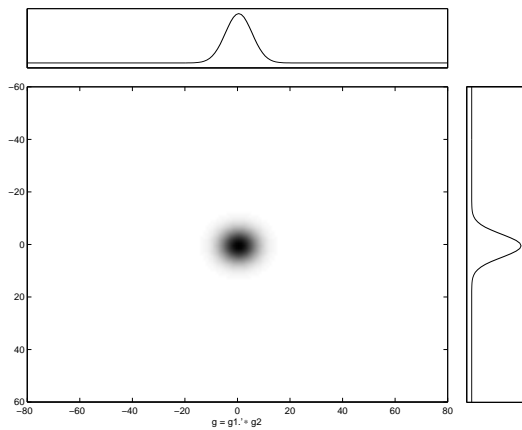
(b) Lattice  $\Lambda_2 = 8\mathbb{Z}_{160} \times 16\mathbb{Z}_{160}$



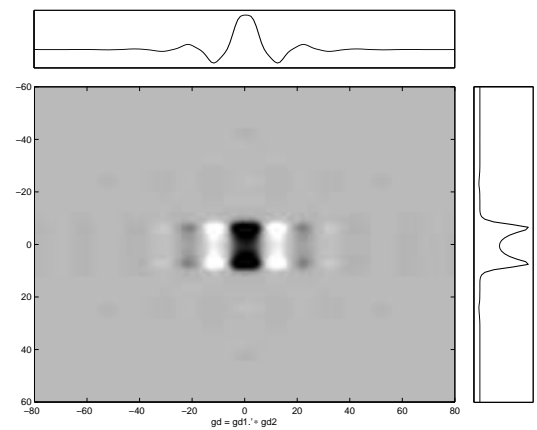
(c) Gaussian  $g_1$  and its dual  $\gamma_1^\circ$  on  $\Lambda_1$



(d) Gaussian  $g_2$  and its dual  $\gamma_2^\circ$  on  $\Lambda_2$



(e) 2D window  $g = g_1 \otimes g_2$



(f) 2D dual window  $\gamma^\circ = \gamma_1^\circ \otimes \gamma_2^\circ$  on  $\Lambda_1 \times \Lambda_2$

FIGURE 5.1: 2D separable window and its dual on a fully separable lattice

image  $\mathbf{f} \in \mathbb{C}^{L_1 \times L_2}$ . These matrix multiplications from the left and right could still be rather expensive, so we want to see to what extent the Gabor coefficient  $\mathbf{c}$  is obtained by a finite number of sampled 1D STFT operations, where the sampling points are determined by shift parameters  $\alpha_1, \alpha_2$  and modulation parameters  $\beta_1, \beta_2$ .

If we remember the 1D case, the Gabor frame  $C_g$  for  $\mathbb{C}^L$  by a window  $g \in \mathbb{C}^L$  involves a separable lattice  $\Lambda = \alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$  with  $|\Lambda| = N = \frac{L^2}{\alpha\beta}$ , and for arbitrary  $f \in \mathbb{C}^L$  we have

$$(C_g f)_{k,\ell} = c_{k,\ell} = \langle f, M_{\beta\ell} T_{\alpha k} g \rangle_{\mathbb{C}^L} = \sum_{u=0}^{L-1} f(u) \overline{M_{\beta\ell} T_{\alpha k} g(u)} = \mathcal{V}_g f(\alpha k, \beta \ell)$$

for  $k \in \langle \frac{L}{\alpha} \rangle$  and  $\ell \in \langle \frac{L}{\beta} \rangle$ , what is merged to a flat vector of length  $N$  if the frame is seen as a matrix  $C_g \in \mathbb{C}^{N \times L}$ . In the 2D case, if we consider  $\mathbf{f} = (f_0, \dots, f_{L_2-1})$  with  $f_j := (f(0, j), \dots, f(L_1 - 1, j))^T$ , then  $b_j = C_{g_1} f_j$  acts as the Gabor analysis operation for all  $f_j \in \mathbb{C}^{L_1}$  with coefficients  $b_j \in \mathbb{C}^{N_1}$  for all  $j \in \langle L_2 \rangle$ . The operation  $\mathbf{b} = C_{g_1} \mathbf{f}$  collects these in a matrix  $\mathbf{b} = (b_0, \dots, b_{L_2-1})$ . If we express its  $k$ -th line as a line vector  $q_k^T := (\mathbf{b})_k = (b_0(k), \dots, b_{L_2-1}(k))$ , we get

$$C_{g_1} \mathbf{f} = \mathbf{b} = \mathbf{q}^T = \begin{pmatrix} q_0^T \\ \vdots \\ q_{N_1-1}^T \end{pmatrix} \in \mathbb{C}^{N_1 \times L_2}.$$

The complete 2D Gabor analysis operation is now  $\mathbf{c} = \mathbf{q}^T C_{g_2}^T = (C_{g_2} \mathbf{q})^T$ , and this is just the Gabor analysis operation of the vectors  $q_k \in \mathbb{C}^{L_2}$  for  $k \in \langle N_1 \rangle$  with respect to the Gabor frame  $C_{g_2}$ .

All in all, the 2D Gabor analysis operation in the twofold-separable case can be obtained by first computing  $L_2$  1D STFT-operations of output length  $N_1$  using the parameters  $\alpha_1, \beta_1$  followed by  $N_1$  1D STFT-operations of output length  $N_2$  using the parameters  $\alpha_2, \beta_2$ . Of course, everything could be wound up from the other side: Instead of considering  $(C_{g_1} \mathbf{f}) C_{g_2}^T$ , it could be done as  $C_{g_1} (\mathbf{f} C_{g_2}^T)$ , where one ends up with  $L_1$  operations with output length  $N_2$  followed by  $N_2$  operations with output length  $N_1$ . These two possibilities shall be referred to as the *Left-STFT* and *Right-STFT* implementations, respectively.

In the case of square images that allow for a single Gabor matrix  $C_g$ , the implementation that makes use of NuHAG's `stft.m` turned out to be slightly faster for the second approach, maybe due to fewer reshaping operations. However, if  $L_1 < L_2$  ("landscape" format), there turned out to be a benefit when the image was transposed to the "portrait" format and the roles of the Gabor frames were exchanged, i.e., the equivalent implementation of  $\mathbf{c}^T = C_{g_2} \mathbf{f}^T C_{g_1}^T$  was faster. Some results are given in Table 5.1.

Computing time (s $\pm$ 0.1s)						
$L_1$	$L_2$	$\text{red}_{\Lambda_1 \times \Lambda_2}$	$C_{g_1} f C_{g_2}^T$	L-STFT	R-STFT	$C_{g_2} f^T C_{g_1}^T$
300	300	100	17.9	15.8	14.8	—
1500	1500	2.25	100.7	37.0	36.4	—
300	1500	15	86.0	39.4	12.6	20.2
1500	300	15	(20.2)	13.1	38.3	(86.0)

TABLE 5.1: Results of Experiment 5.2.1

As the reconstruction (2D Gabor expansion) is just a multiplication of the dual Gabor matrices  $C_{\gamma_i}^*$  from the left and right of  $c$ , this task can be seen as a sequence of 1D Gabor expansions and can thus be obtained by a sequence of inverse 1D STFT-operations as well, implemented by NuHAG as `istft.m` for product lattices. There are again two ways: The first one is to do  $N_1$  inverse operations with output length  $L_2$  using the parameters  $\alpha_2, \beta_2$  followed by  $N_2$  operations with output length  $L_1$  using  $\alpha_1, \beta_1$ . The second way exchanges  $L_i$  and  $N_i$  correspondingly.

### 5.2.2 Visualizing a Sampled STFT of an Image

In the previous chapter we had the idea to visualize the full STFT of an image as a large block image, where either each block fully represents the frequency domain and the position of the blocks the position domain, or vice versa. As such an image would become rather huge, we prefer to visualize only a sampled STFT instead. In the case of a separable atom, this can be realized by obtaining the discrete 2D Gabor transform by (5.7), where the two involved Gabor matrices  $C_{g_i}$  consider a special order of their Gabor frame elements  $M_{\ell_i} T_{k_i} g_i$ .

For a Gabor frame  $\{M_{\ell} T_k g\}_{(k,\ell) \in \Lambda} \subseteq \mathbb{C}^L$  given by a 1D window  $g \in \mathbb{C}^L$  on a separable lattice  $\Lambda = \alpha \mathbb{Z}_L \times \beta \mathbb{Z}_L$  with  $N = |\Lambda| = \frac{L^2}{\alpha\beta}$  elements, we want to say that the Gabor frame elements are ordered with *modulation priority* if the Gabor frame matrix  $C_g \in \mathbb{C}^{N \times L}$  is of the form

$$C_g = \begin{pmatrix} M_0 T_0 g^* \\ M_{\beta} T_0 g^* \\ \vdots \\ M_{L/\beta-1} T_0 g^* \\ M_0 T_1 g^* \\ \vdots \\ M_{L/\beta-1} T_1 g^* \\ \vdots \\ M_{L/\beta-1} T_{L/\alpha-1} g^* \end{pmatrix}$$

and that they are ordered with *translation priority* if the Gabor frame matrix is

$$\tilde{C}_g = \begin{pmatrix} M_0 T_0 g^* \\ M_0 T_\alpha g^* \\ \vdots \\ M_0 T_{L/\alpha-1} g^* \\ M_1 T_0 g^* \\ \vdots \\ M_1 T_{L/\alpha-1} g^* \\ \vdots \\ M_{L/\beta-1} T_{L/\alpha-1} g^* \end{pmatrix}.$$

It is clear that we have  $\tilde{C}_g = P C_g$  for a certain permutation matrix  $P \in \mathbb{C}^{N \times N}$ .

If we now take an image  $f \in \mathbb{C}^{L_1 \times L_2}$  and two Gabor frames  $\{M_{\ell_i} T_{k_i} g_i\}_{(k_i, \ell_i) \in \Lambda_i} \subseteq \mathbb{C}^{L_i}$  on separable lattices  $\Lambda_i = \alpha_i \mathbb{Z}_{L_i} \times \beta_i \mathbb{Z}_{L_i}$ , we can take their product Gabor frame for  $\mathbb{C}^{L_1 \times L_2}$  and obtain the mentioned two possibilities for an STFT block image by either considering the frame matrices  $C_{g_i}$  or  $\tilde{C}_{g_i}$ . The matrices  $C_{g_i}$  are ordered by modulation priority, and if  $c = C_{g_1} f C_{g_2}^T$ , then  $c$  consists of  $\frac{L_1}{\beta_1} \times \frac{L_2}{\beta_2}$ -blocks

$$X_{k_1, k_2} := (\langle f, M_{(\ell_1, \ell_2)} T_{(k_1, k_2)} g \rangle)_{\ell_1, \ell_2}$$

such that

$$c = \begin{pmatrix} X_{0,0} & \cdots & X_{0, L_2/\alpha_2-1} \\ \vdots & \cdots & \vdots \\ X_{L_1/\alpha_1-1, 0} & \cdots & X_{L_1/\alpha_1-1, L_2/\alpha_2-1} \end{pmatrix}.$$

The blocks  $X_{k_1, k_2}$  equal the part  $(\mathcal{V}_g f(k_1, k_2, \ell_1, \ell_2))_{\ell_1, \ell_2}$  of the sampled STFT and thus contain the whole (sampled) set of frequency shifts for a certain position shift of the window  $g = g_1 \otimes g_2$ . The (sampled) frequency domain is therefore spanned in each of the blocks  $X_{k_1, k_2}$ , and their positions in  $c$  span the (sampled) position domain. Each  $X_{k_1, k_2}$  could be seen as a sampled ‘‘Fourier image’’ of the discrete Fourier transform  $\widehat{f \cdot T_{(k_1, k_2)} g}$ .

In the other case, where we’ve got  $\tilde{c} = \tilde{C}_{g_1} f \tilde{C}_{g_2}^T$ , the Gabor coefficient consists of  $\frac{L_1}{\alpha_1} \times \frac{L_2}{\alpha_2}$ -blocks

$$Y_{\ell_1, \ell_2} := (\langle f, M_{(\ell_1, \ell_2)} T_{(k_1, k_2)} g \rangle)_{k_1, k_2}$$

such that

$$\tilde{c} = \begin{pmatrix} Y_{0,0} & \cdots & Y_{0, L_2/\beta_2-1} \\ \vdots & \cdots & \cdots \\ Y_{L_1/\beta_1-1, 0} & \cdots & Y_{L_1/\beta_1-1, L_2/\beta_2-1} \end{pmatrix}.$$

Here, the blocks  $Y_{\ell_1, \ell_2}$  equal the part  $(\mathcal{V}_g f(k_1, k_2, \ell_1, \ell_2))_{k_1, k_2}$  of the sampled STFT and contain the corresponding set of position shifts for a certain frequency-shift of  $g$ . The position domain is spanned in each of the blocks  $Y_{\ell_1, \ell_2}$ , and their positions in  $\tilde{c}$  span the frequency domain. Each  $Y_{\ell_1, \ell_2}$  could now be seen as a (sampled) ‘‘convolution image’’  $f * M_{(\ell_1, \ell_2)} g^*$  that has been pointwise multiplied with the discrete function  $(k_1, k_2) \mapsto e^{-2\pi i(k_1 \ell_1 / L_1 + k_2 \ell_2 / L_2)}$ , according to equation (1.19).



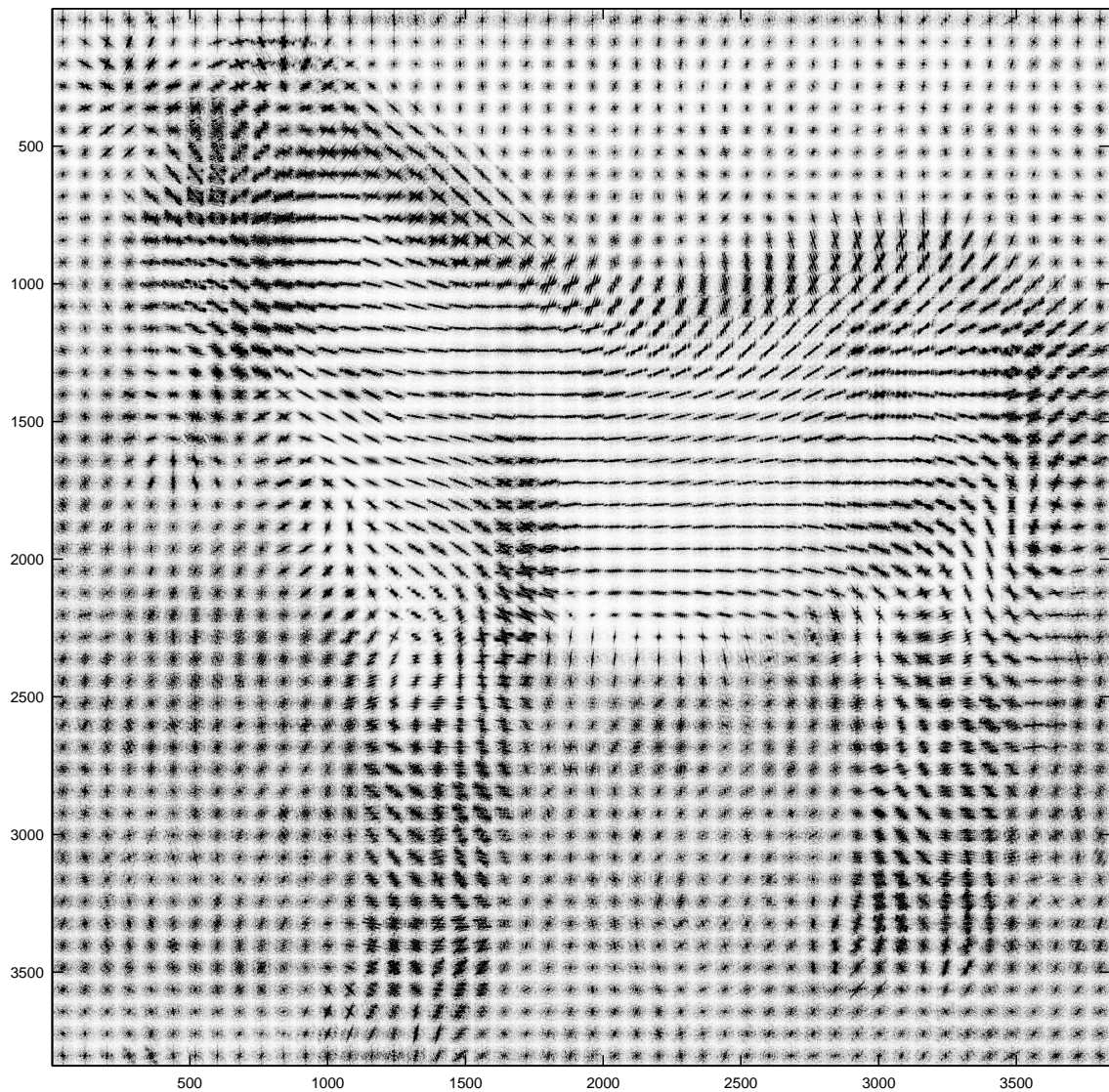


FIGURE 5.2: Discrete 2D Gabor transform of zebra, modulation priority. The picture shows the absolute values of  $\mathbf{c} = C_g \mathbf{f} C_g^T$ , where  $g$  is the 1D Gaussian of length 480 and  $C_g$  is the Gabor matrix for the lattice  $\Lambda = 10 \mathbb{Z}_{480} \times 6 \mathbb{Z}_{480}$ , whose entries were ordered with modulation priority.

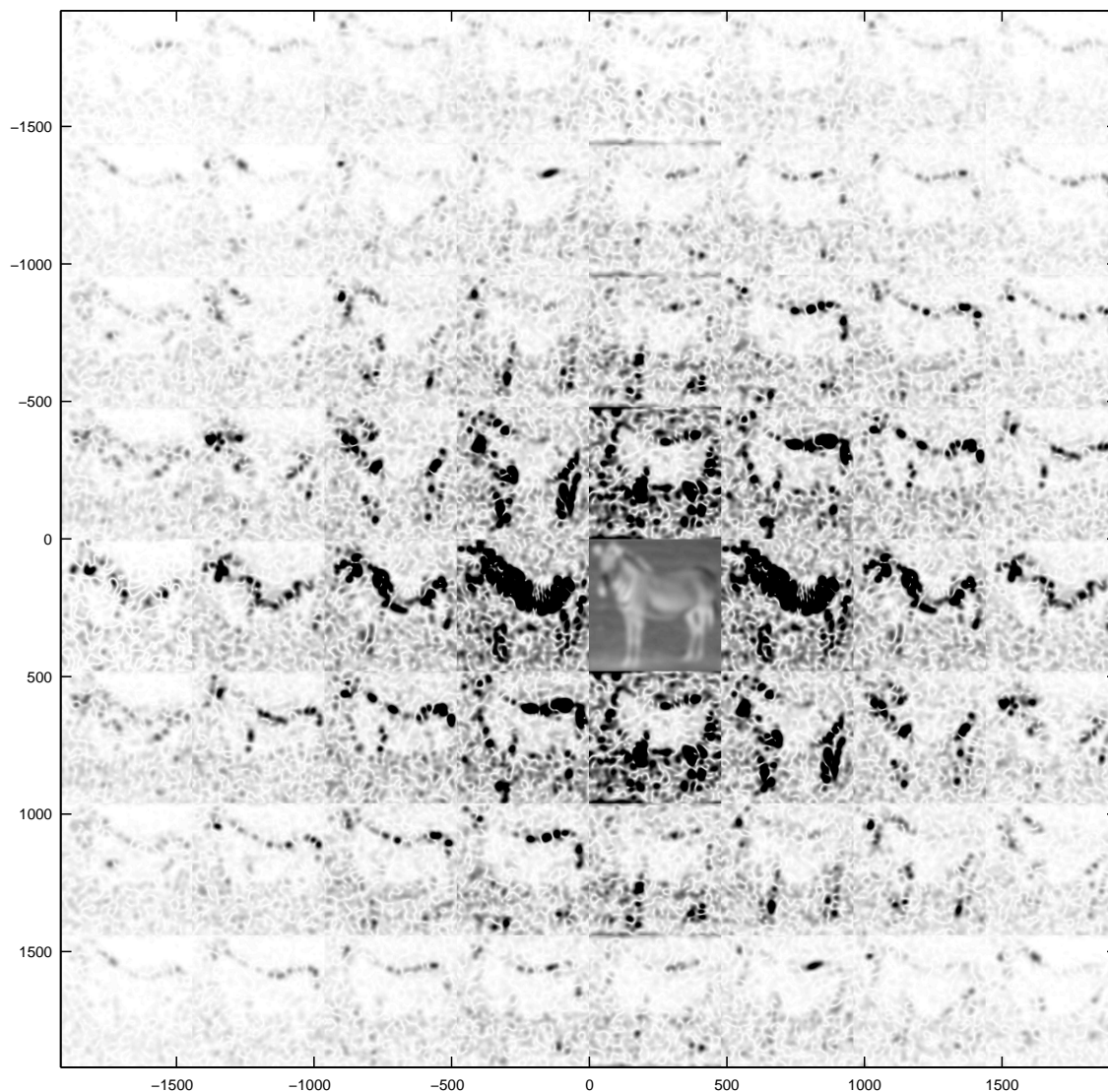


FIGURE 5.3: Discrete 2D Gabor transform of zebra, translation priority. The picture shows the absolute values of  $\tilde{c} = \tilde{C}_g f \tilde{C}_g^T$ , where  $g$  is the 1D Gaussian of length 480 and  $\tilde{C}_g$  is the Gabor matrix for the lattice  $\Lambda = \mathbb{Z}_{480} \times 60 \mathbb{Z}_{480}$ , whose entries were ordered with translation priority. The Gaussian blurred image in the middle has been scaled into the colormap individually.

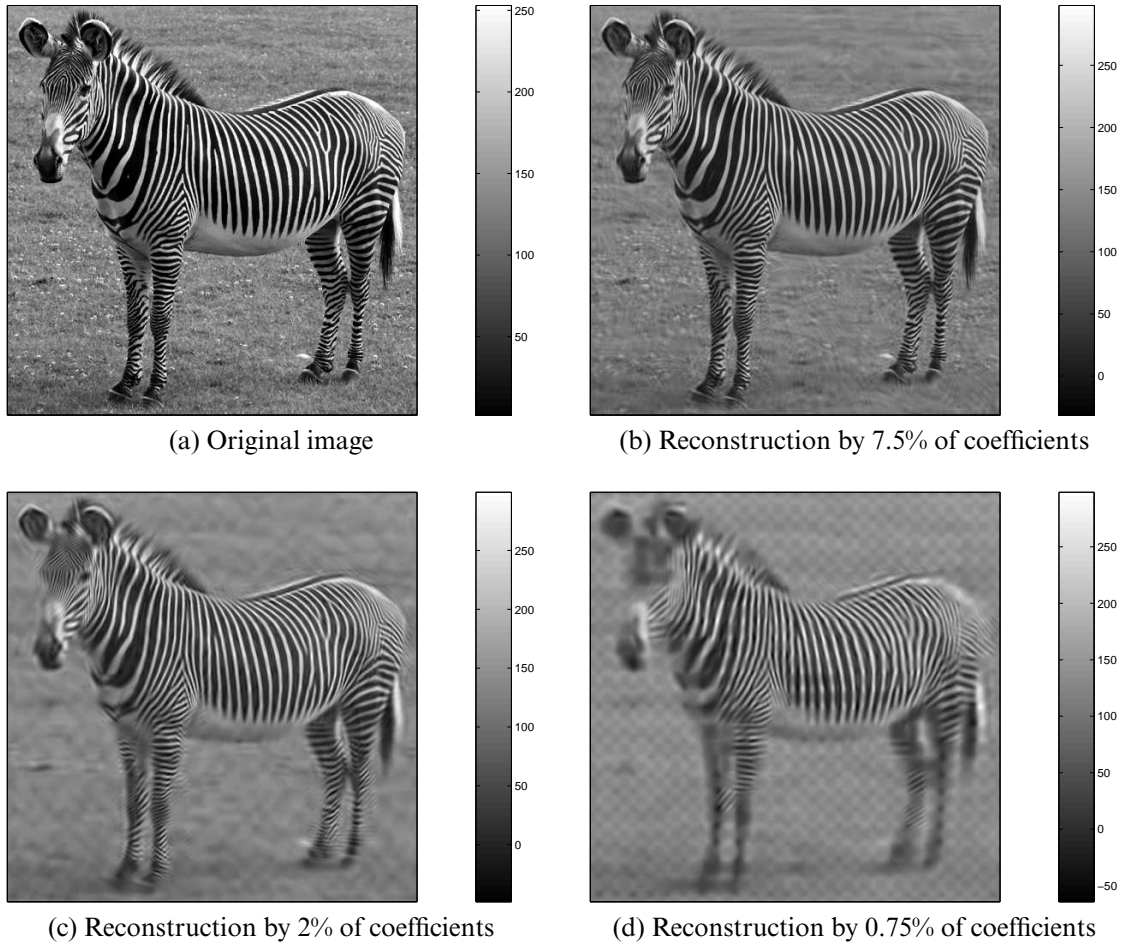


FIGURE 5.4: Thresholding by separable atom on fully separable lattice

Figures 5.2 and 5.3 show examples for both cases using the zebra test image. As it is a square image, we can take  $g_1 = g_2$  and thus  $C_{g_1} = C_{g_2}$ . The first figure composes the Gabor transform coefficient matrix as blocks of Fourier images. Clearly, the overall image reflects the shape of the zebra. The “pixels” of that image contain “Fourier jets” that are orthogonal to the edges at the corresponding position in the original zebra image. Thus, the “jets” are oriented horizontally where e.g. the body of the animal shows vertical line patterns. The second figure clearly shows blocks of zebra images that have been convolved with modulated Gaussians. The absolute values show the peaks as black spots within the respective image blocks.

### 5.2.3 Experiment: Gabor Coefficient Thresholding

Here we want to explore how the reconstructed image responds to a filtering of its Gabor coefficients. This is a leering at the task of image compression or denoising. We make use of the zebra test image again, so we’re able to take one Gabor frame for  $\mathbb{C}^{480}$  and use its tensor product as Gabor frame for the  $480 \times 480$  image space. Our experiment involves a Gaussian of length 480 as analysis atom  $g$  and a Gabor frame matrix  $C_g$  on the separable lattice  $\Lambda = 20 \mathbb{Z}_{480} \times 16 \mathbb{Z}_{480}$  with  $N = |\Lambda| = 720$  sam-

$\theta$	0	45.3789	140.2533	393.0024
$ \Phi $	518400	38880	10368	3888
$\ \tilde{f} - f\ _F$	$7.7683 \times 10^{-14}$	0.0998	0.1669	0.2462

TABLE 5.2: Results of Experiment 5.2.3

pling points, yielding a redundancy of 1.5 for the 1D frame and thus a redundancy of 2.25 for the 2D frame on the fully separable 4D lattice  $\Lambda \times \Lambda$ . The Gabor coefficient of the square image  $f$  is obtained by  $c = C_g f C_g^T$ . Thresholding means to put those (absolute) coefficient values to zero that are below a given threshold  $\theta > 0$ . We want to sort the absolute coefficient values  $|c(m, n)|$  decreasingly and only use the largest 7.5%, 2% and 0.75% for reconstruction. The minimum value of the remaining coefficients determines the used threshold. A filter could be given as a 0-1-matrix  $\Phi \in \mathbb{C}^{N \times N}$  such that the pointwise multiplication  $\Phi \cdot c$  yields the desired filtered coefficient matrix.  $\Phi$  is given as

$$\Phi(m, n) := \begin{cases} 0 & |c(m, n)| \leq \theta \\ 1 & \text{else.} \end{cases}$$

If  $\gamma^\circ := (C_g^* C_g)^{-1} g$  is the canonical dual window to  $g$  on the lattice  $\Lambda$ , the reconstruction of the square image  $f$  is obtained via  $f = C_{\gamma^\circ}^* c (C_{\gamma^\circ}^*)^T$ . The calculation of the dual is obtained by efficient MATLAB code developed by NuHAG, and it's therefore much cheaper to get  $C_{\gamma^\circ}^*$  than to compute  $C_g^\dagger$ , what is equal to the former.

After the filter matrix  $\Phi$  has been obtained, the filtered reconstruction is given as

$$\tilde{f} = C_{\gamma^\circ}^* (\Phi \cdot c) (C_{\gamma^\circ}^*)^T,$$

and the (norm) difference between  $\tilde{f}$  and the original  $f$  can be measured. To enable a better comparison with other atoms or lattices, this difference is calculated for the (Frobenius-)normalized images only.

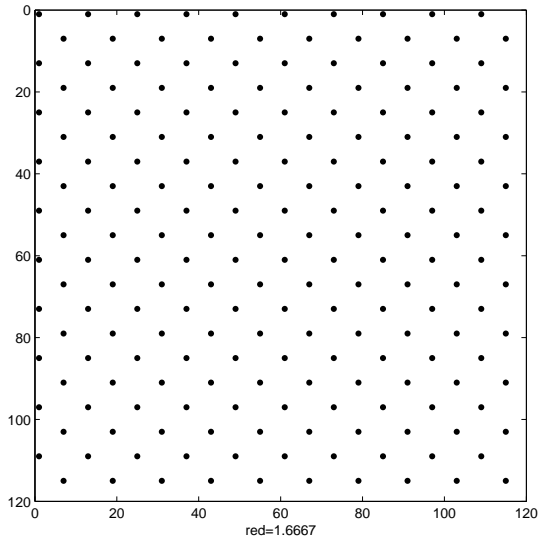
Figure 5.4 shows the resulting reconstructed images that used all, 7.5%, 2% and only 0.75% of the largest coefficients. Subfigure 5.4a shows the reconstruction for  $\theta = 0$  and is therefore equal to the original. The remaining images indicate how the filtering creates blur in the reconstructions. With increasing size of  $\theta$ , only the most prominent patterns survive, and the distribution of the elements  $T_k \gamma^\circ$  of the dual frame on the image domain becomes visible. The obtained numerical values are given in Table 5.2.

### 5.3 Separable Atoms on Partially Non-Separable Lattices

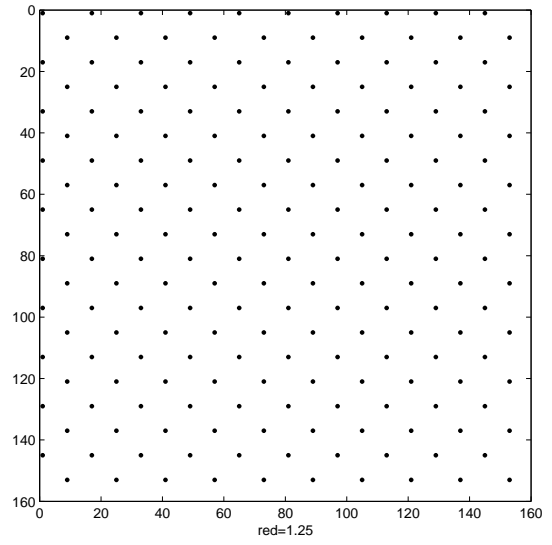
In this section we still want to consider separable Gabor atoms  $g = g_1 \otimes g_2 \in \mathbb{C}^{L_1 \times L_2}$  on lattices

$$\Lambda = \Lambda_1 \times \Lambda_2 \trianglelefteq \mathbb{Z}_{L_1} \times \widehat{\mathbb{Z}_{L_1}} \times \mathbb{Z}_{L_2} \times \widehat{\mathbb{Z}_{L_2}}$$

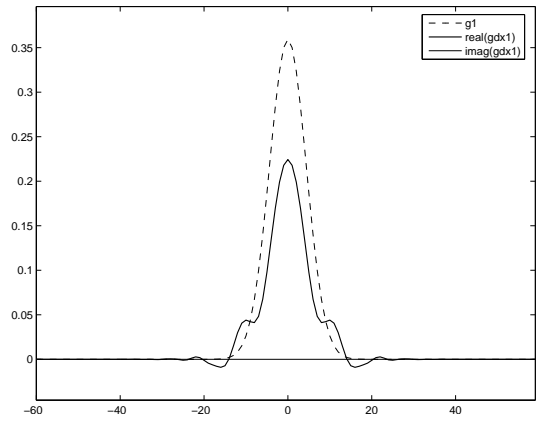
where the lattices  $\Lambda_i$  might or might not be separable subgroups  $\Lambda_i \trianglelefteq \mathbb{Z}_{L_i} \times \widehat{\mathbb{Z}_{L_i}}$ . To be more precise, at least one of those subgroups should be non-separable to speak



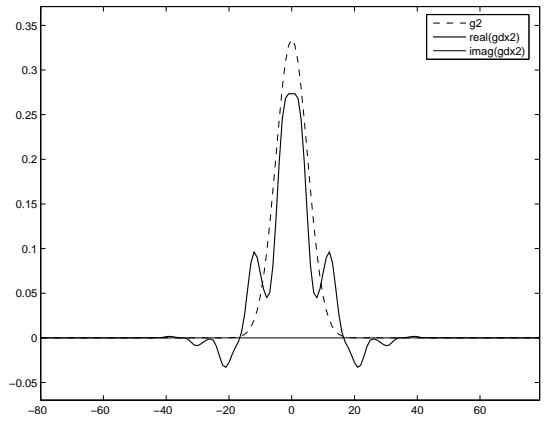
(a) Quincunx lattice  $\Lambda_1^\times$  ( $d_r = d_c = 12$ )



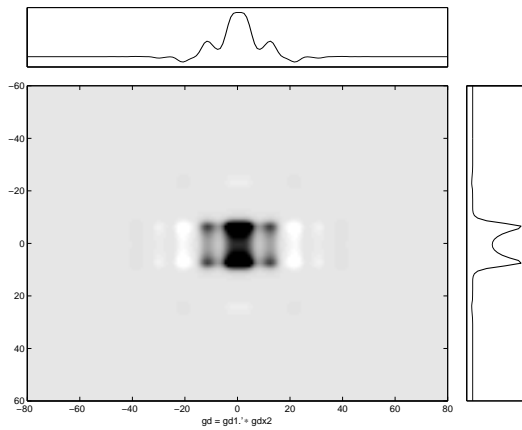
(b) Quincunx lattice  $\Lambda_2^\times$  ( $d_r = d_c = 16$ )



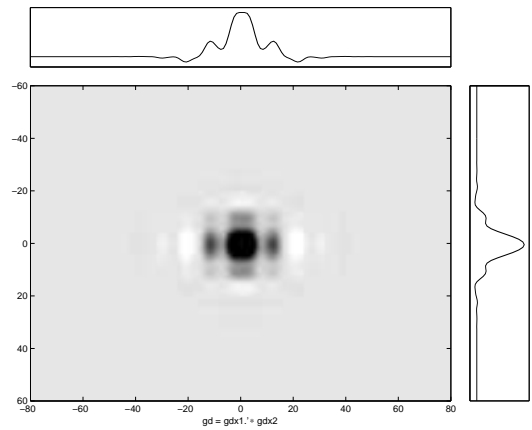
(c) Dual  $\gamma_{1,x}^\circ$  of Gaussian  $g_1$  on the quincunx  $\Lambda_1^\times$



(d) Dual  $\gamma_{2,x}^\circ$  of Gaussian  $g_2$  on the quincunx  $\Lambda_2^\times$



(e) 2D dual window  $\gamma_{1,x}^\circ \otimes \gamma_{2,x}^\circ$  on  $\Lambda_1 \times \Lambda_2^\times$



(f) 2D dual window  $\gamma_{1,x}^\circ \otimes \gamma_{2,x}^\circ$  on  $\Lambda_1^\times \times \Lambda_2^\times$

FIGURE 5.5: Separable 2D dual windows on partially non-separable lattices. The dual  $\gamma_1^\circ$  on the separable lattice  $\Lambda_1$  is that of Figure 5.1.

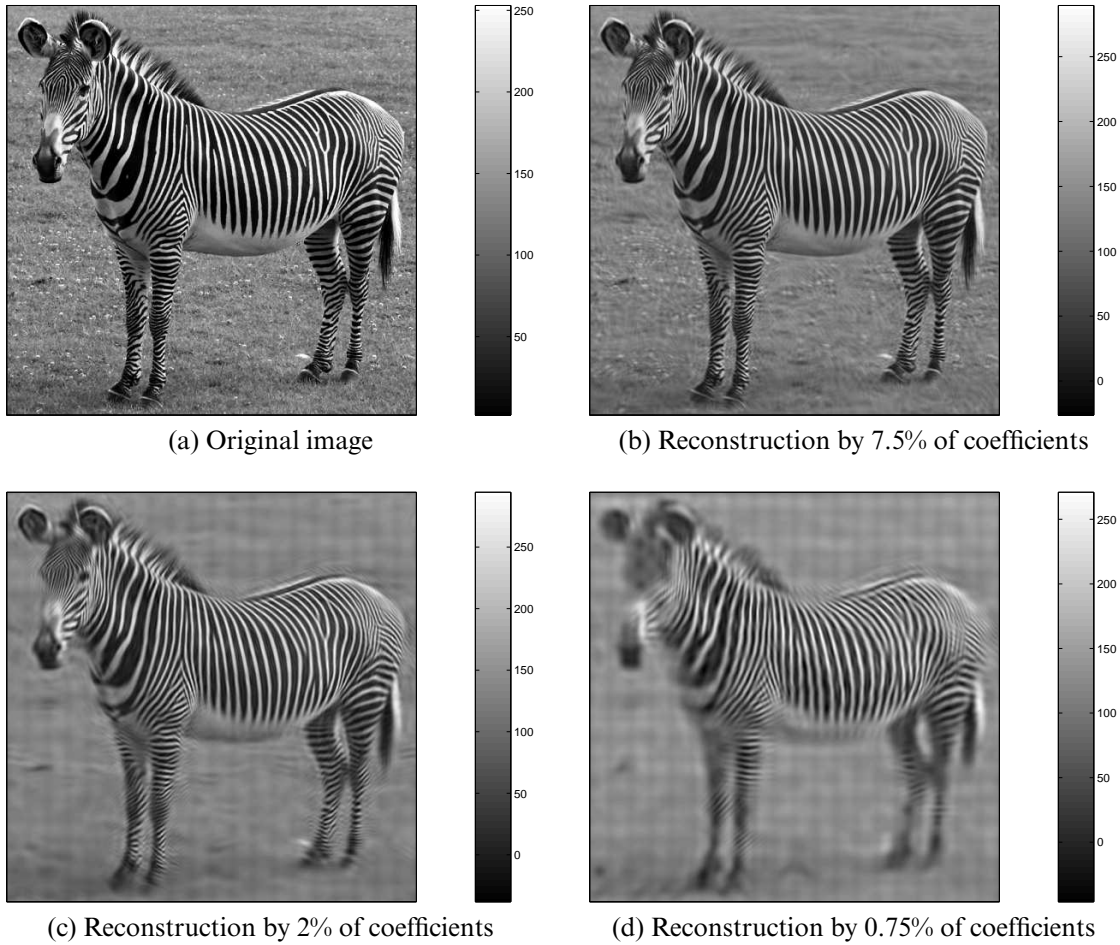


FIGURE 5.6: Thresholding by separable atom on a product-quincunx lattice

of a case of a partially non-separable lattice, as it cannot be fully separated into four 1D subgroups of  $\mathbb{Z}_{L_i}$ . However, for the described case we can carry over many of the results from the previous section, because they hold for general frames. E.g., the tensor product of two 1D Gabor frames  $\{M_{\ell_i} T_{k_i} g_i\}_{(k_i, \ell_i) \in \Lambda_i} \subseteq \mathbb{C}^{L_i}$  on non-separable lattices  $\Lambda_i$  is still a Gabor frame for the image space  $\mathbb{C}^{L_1 \times L_2}$ . The Gabor analysis and synthesis operation is still given by multiplying the two Gabor matrices from the left and right, (5.7) and (5.8), and the (canonical) dual window is still the tensor product of the 1D duals. The other possibility of partial non-separability, where  $\Lambda = \Omega \times \hat{\Omega}$  with non-separable  $\Omega \trianglelefteq \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  and  $\hat{\Omega} \trianglelefteq \widehat{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}$ , does not allow such a split into two 1D Gabor frames.

Figure 5.5 shows how lattices, 1D windows and their 1D duals can be chosen independently for each dimension. The 2D dual in Subfigure 5.5e is composed of a separable lattice for  $L_1$  and a quincunx lattice for  $L_2$ . The 2D dual in Subfigure 5.5f uses quincunx lattices for both dimensions.

Figure 5.6 shows how the reconstruction of the zebra image responds to a thresholding of its Gabor coefficients on a product of two equal quincunx lattices that are determined by  $d_r = d_c = 24$ . The thresholds use the same subset of coefficients as in Experiment 5.2.3, but the redundancy is slightly higher at 2.78 instead of 2.25. The

$\theta$	0	45.0879	132.5164	398.1295
$ \Phi $	640000	48000	12800	4800
$\ \tilde{f} - f\ _F$	$2.3910 \times 10^{-13}$	0.0968	0.1602	0.2410

TABLE 5.3: Results of thresholding on product-quincunx lattice

distribution of the 2D duals is not that evident anymore in the reconstructed images. However, the results could still be beat on fully separable lattices by using an even higher redundancy. Table 5.3 contains the obtained numerical values.

What also still works is visualizing the sampled STFT of an image as a block image. If one of the involved 2D lattices is non-separable, there's not the same set of modulations used for all of the corresponding positions, but subsequently a different subset of modulations. Maybe the non-separable lattice is even structured such that only a single modulation is used at every position—a valid case. But the corresponding Gabor matrix can still sort the TF-shifted elements prioritized by modulation or translation, and the Gabor coefficient  $\mathbf{c} = C_{g_1} \mathbf{f} C_{g_2}^T$  of the image  $\mathbf{f}$  still gets a block structure. However, the blocks might be of small width or height where only few sampling points were involved.

One thing that ceases to work is the implementation of the 2D Gabor analysis and synthesis by the help of the sampled 1D STFT, as the sampling points are not distributed in grid shape anymore and thus cannot be described by shift parameters  $\alpha_1, \alpha_2$  and modulation parameters  $\beta_1, \beta_2$ . There could of course be the idea to see (2D) non-separable lattices as a finite union of shifts of a single grid. E.g., a quincunx lattice with  $d = d_r = d_c$  could be seen as a grid  $d\mathbb{Z}_L \times d\mathbb{Z}_L$  united with the same grid shifted by  $(\frac{d}{2}, \frac{d}{2})$ . But the `stft.m` implementation had to be changed to support this shift of the involved grid. If this is implemented, the complete Gabor coefficient can be obtained as the union of the single analysis operations that incorporate the shifted grids. The same holds for the reconstruction. The calculation of the dual window, of course, has to consider the whole union of sampling points.

One important thing is different to the case of fully separable lattices: The (1D) duals on general lattices might have significant non-zero imaginary parts, as indicated in Section 3.4. Thus, if the 4D PF-lattice involves at least one non-separable 2D lattice such that the 1D dual has a non-zero imaginary part, the tensor product of that dual with the second dual gets a non-zero imaginary part as well. The values of the emerging 2D dual could therefore be seen as evolving in four dimensions if  $\mathbb{C}^2$  is interpreted as  $\mathbb{R}^4$ . Figure 5.7 indicates that although the real part of the dual might be rather similar to the original atom, its imaginary part lacks a “nice” behavior. This also means that partial Gabor reconstructions that involve such dual windows earn a significant non-zero imaginary part as well. Figure 5.8 indicates what such a case does to a partial reconstruction of the zebra image. The colorbar to the right of the visualized imaginary part of the obtained image indicates to what extent its color values are far from being zero.

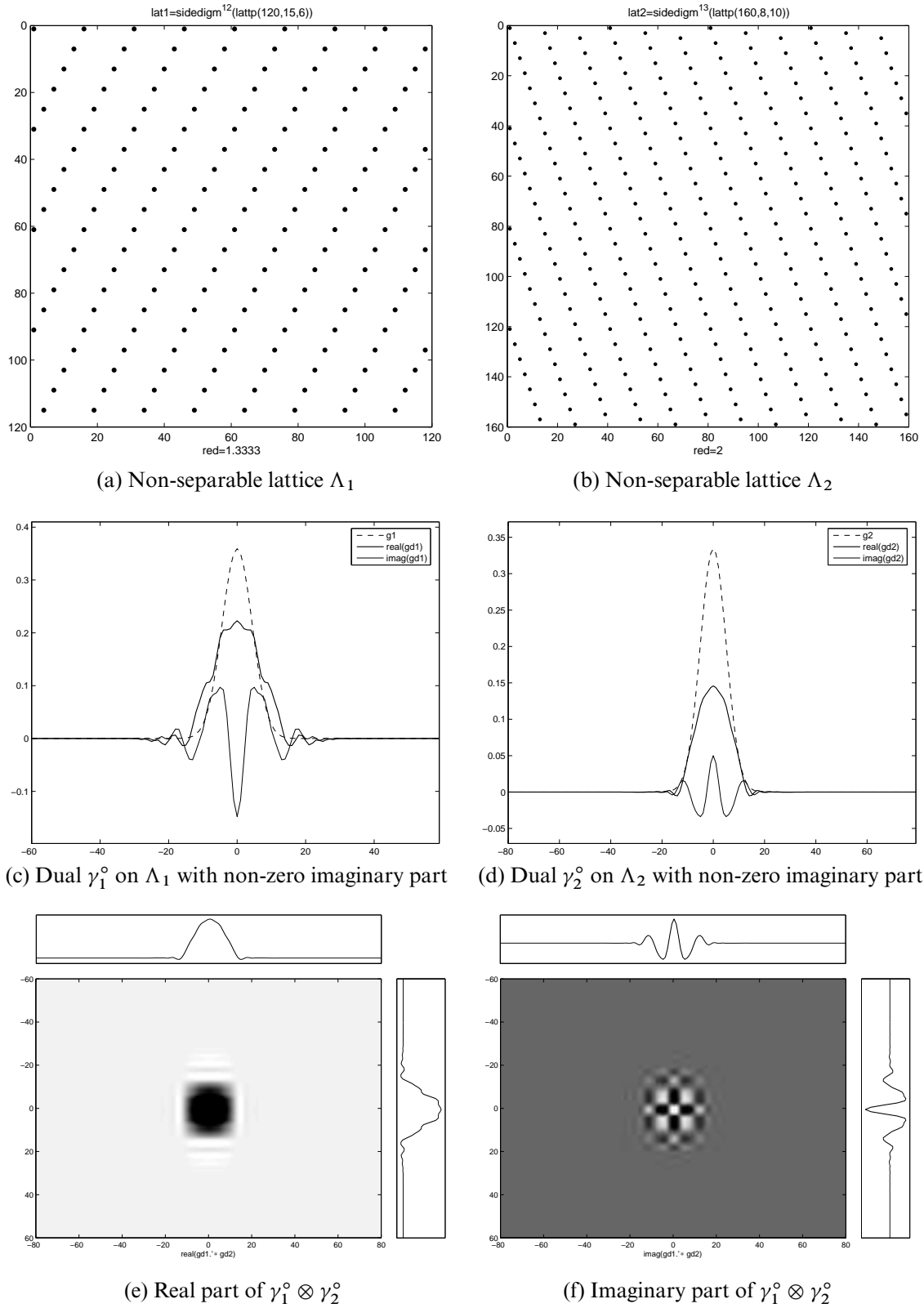


FIGURE 5.7: Separable 2D dual window with non-zero imaginary part



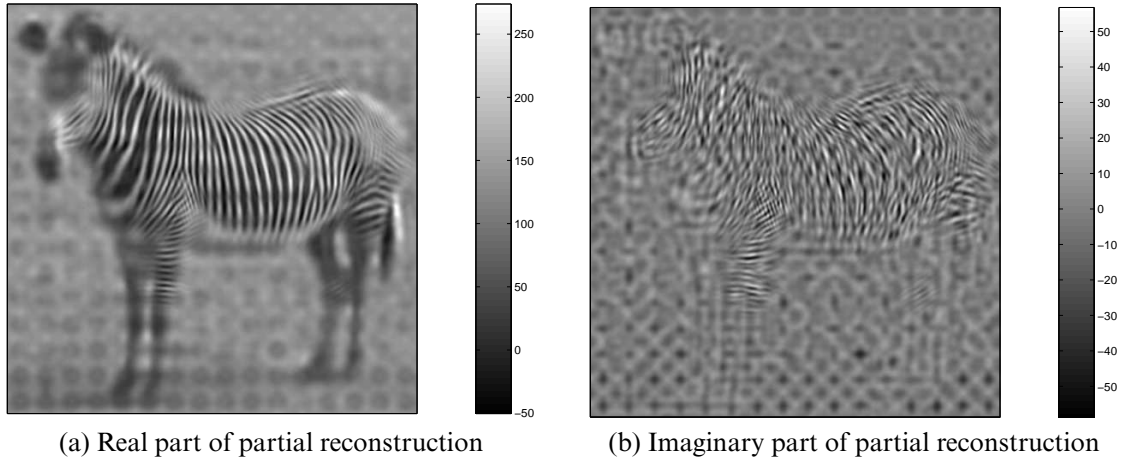


FIGURE 5.8: Partial reconstruction by dual with non-zero imaginary part

## 5.4 Non-Separable Atoms on Fully Separable Lattices

Non-separable windows are those which can only be defined considering the complete image domain  $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$ , and not  $\mathbb{Z}_{L_1}$  and  $\mathbb{Z}_{L_2}$  separately. These cannot be described as a tensor product  $g_1 \otimes g_2$  with  $g_i \in \mathbb{C}^{L_i}$  anymore, but only generally as  $\mathbf{g} \in \mathbb{C}^{L_1 \times L_2}$ . With this case we lose the ability to consider two (1D) frames independently for each dimension and we cannot apply two frame matrices independently to an image. It appears that we have to stick to the known factorizations of Gabor matrices on (fully) separable lattices with parameters  $\alpha_i, \beta_i$ , like denoted in Section 3.3, and we thus cannot make use of the equidistantly sampled 1D STFT. However, under certain conditions this case can be completely referred to a 1D case, as we'll see below.

Figure 5.9 indicates an important thing about the redundancy. Sure, a redundancy of  $\frac{|\Lambda|}{L_1 L_2} \geq 1$  is only a necessary condition, but it seems to be important to consider the redundancy in each dimension. The involved window is a 2D Gaussian window  $\mathbf{g} \in \mathbb{C}^{120 \times 160}$ , stretched vertically by  $\frac{4}{3}$ , shrunk horizontally by  $\frac{3}{4}$ , then rotated (counter-clockwise) by  $\frac{3}{8}\pi$ . Subfigure 5.9d shows its dual on a fully separable 4D PF-lattice with overall redundancy 6.4. It was computed by Prinz' implementation `ppdw2.m` that makes use of the Gabor matrix factorizations. But although the redundancy value gives the impression to be safe, it hides the fact that the involved lattice is actually  $\Lambda = 10 \mathbb{Z}_{120} \times 12 \mathbb{Z}_{120} \times 5 \mathbb{Z}_{160} \times 5 \mathbb{Z}_{160}$ , yielding the redundancy as  $\frac{120}{10 \cdot 12} \cdot \frac{160}{5 \cdot 5} = 1 \cdot 6.4$ . This shows that the vertical redundancy is critical, and the dual has a bad localization in the vertical dimension. It is therefore necessary to make sure that the redundancy is reasonably distributed among the dimensions. In this sense, fully separable 4D lattices can always be considered as a product of two 2D TF-lattices with independent redundancies, no matter what structure the 2D window possesses.

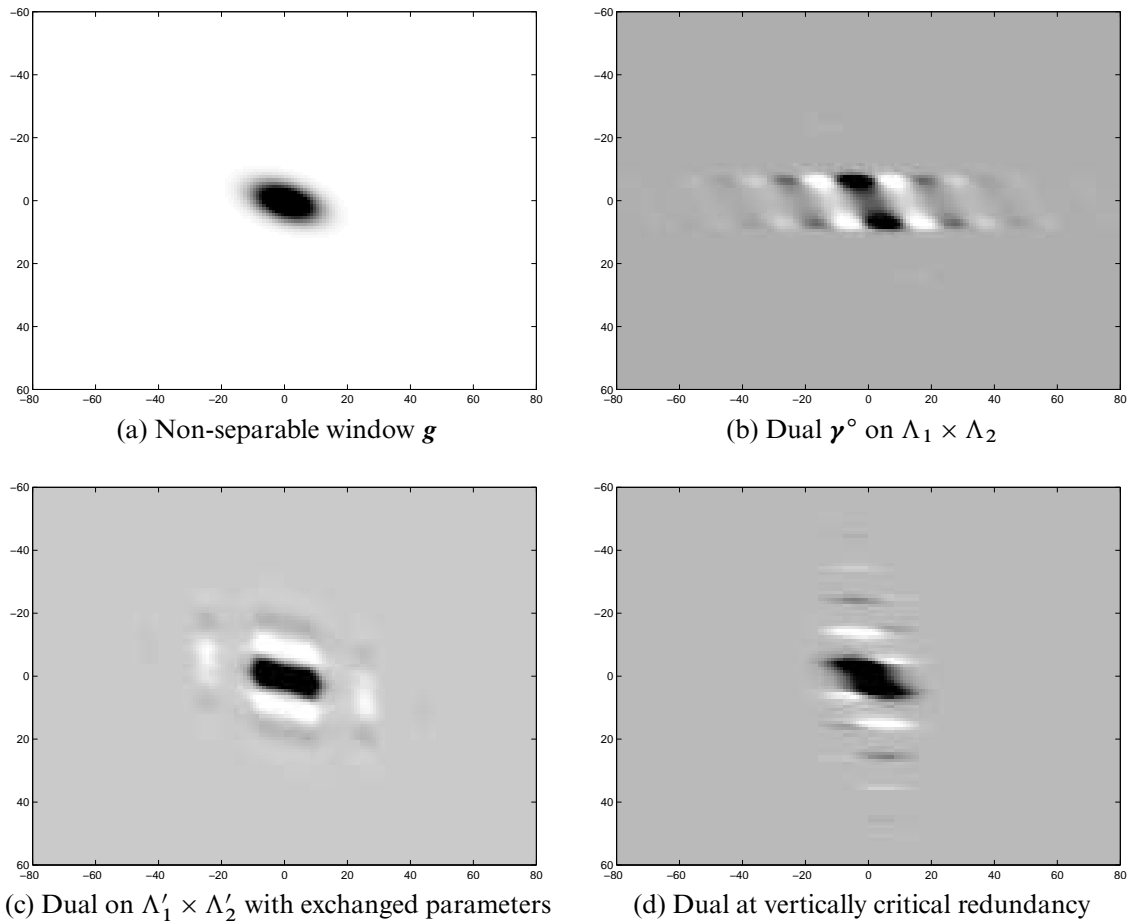


FIGURE 5.9: A non-separable window and some duals on fully separable lattices. The lattices  $\Lambda_i$  are that of Figure 5.1. The lattices  $\Lambda'_i$  exchange  $\alpha_i$  with  $\beta_i$ . The last lattice has vertical redundancy 1 and horizontal redundancy 6.4.

### 5.4.1 2D Gabor Expansions by 1D Algorithms

Feichtinger and Kaiblinger published a result in [FK97] that whenever  $L_1$  and  $L_2$  are relatively prime, i.e.,  $\gcd(L_1, L_2) = 1$ , the calculation of a 2D dual to a 2D window  $g \in \mathbb{C}^{L_1 \times L_2}$  can be obtained by calculating the 1D dual of a matching 1D window  $\kappa(g) \in \mathbb{C}^{L_1 L_2}$ . The key lies within a certain isomorphism of finite groups: If  $L_1$  and  $L_2$  are relatively prime, then  $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  is isomorphic to  $\mathbb{Z}_{L_1 L_2}$ . This means that any image of size  $L_1 \times L_2$  can then be transformed to a vector of length  $L = L_1 L_2$  while preserving the group-law, because  $\ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}) \cong \ell^2(\mathbb{Z}_L)$ , whose isomorphism we denote by  $\kappa$ . The mapping is determined in a way that  $\begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  corresponds to  $1 \in \mathbb{Z}_L$ , even more, we have  $\begin{pmatrix} j \\ k \end{pmatrix} \mapsto j$  for  $j \in \{\min\{L_1, L_2\}\}$ . More explicitly, the isomorphism between the domains, denoted by  $\iota$ , is given by

$$\iota: \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2} \rightarrow \mathbb{Z}_L, \quad \begin{pmatrix} j \\ k \end{pmatrix} \mapsto aL_1k + bL_2j \pmod{L},$$

where  $a, b \in \mathbb{Z}$  are chosen such that

$$aL_1 + bL_2 \equiv 1 \pmod{L}.$$

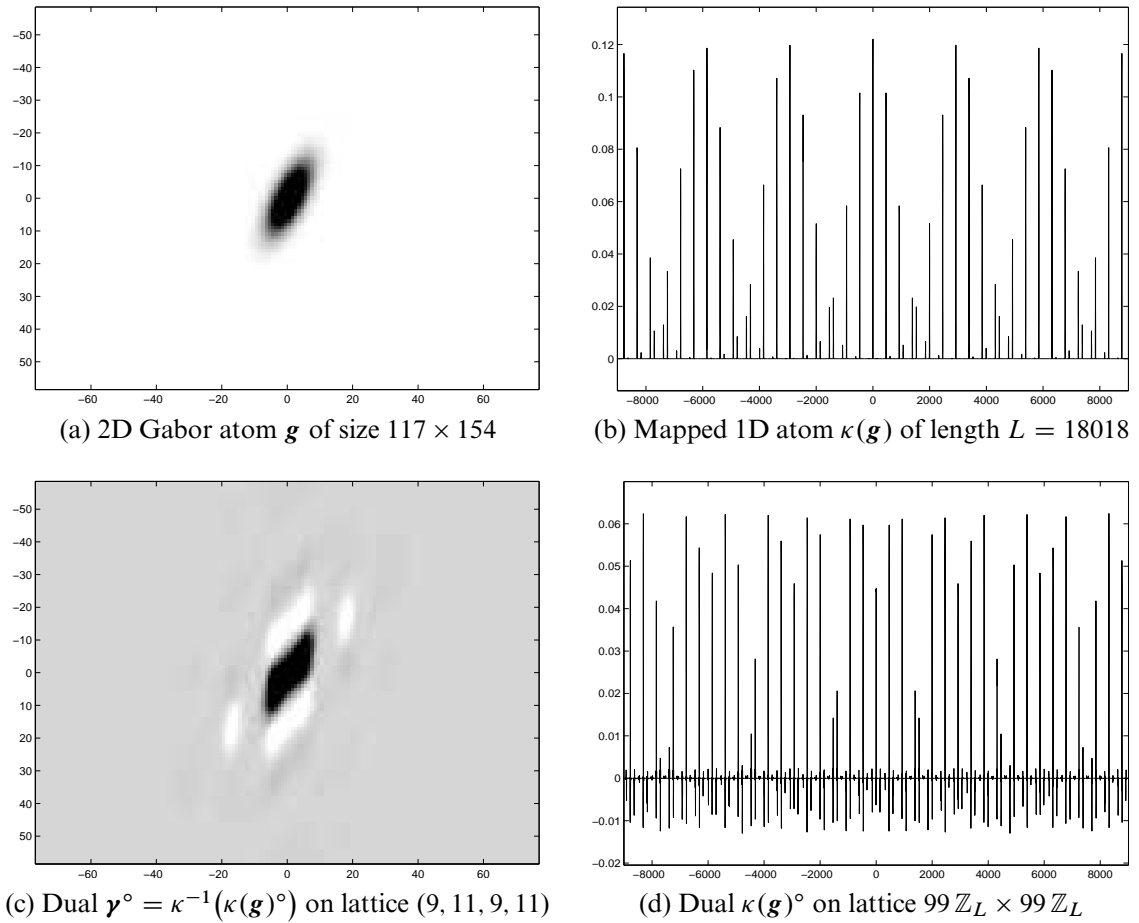


FIGURE 5.10: Obtaining a dual 2D atom by the dual of its 1D extracted atom

Taking care of preserving the group-law one needs not compute  $a$  and  $b$  explicitly.

In other words, by moving on the diagonal of the image domain, i.e. along the points  $\begin{pmatrix} j \bmod L_1 \\ j \bmod L_2 \end{pmatrix}$ , one is able to span the complete image domain and return to the starting point without touching any point twice if  $L_1$  and  $L_2$  are relatively prime. Shifts on the image domain  $\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  correspond to shifts on  $\mathbb{Z}_L$ , as all points get moved to a new position by the same amount. Similarly, any plane wave (pure 2D oscillation) in  $\ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2})$  becomes a pure 1D oscillation in  $\ell^2(\mathbb{Z}_L)$ . A 4D PF-lattice determined by 2D shift parameters  $(\alpha_1, \alpha_2)$  and 2D modulation parameters  $(\beta_1, \beta_2)$  becomes a 2D TF-lattice with parameters  $\alpha = \alpha_1\alpha_2$  and  $\beta = \beta_1\beta_2$ .

All in all, we've got two benefits: The 2D dual window to  $\mathbf{g} \in \ell^2(\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2})$  on the lattice determined by  $(\alpha_1, \alpha_2, \beta_1, \beta_2)$  can be obtained by mapping the window  $\mathbf{g}$  to its vector  $\kappa(\mathbf{g}) \in \ell^2(\mathbb{Z}_L)$ , calculating its dual  $\kappa(\mathbf{g})^\circ$  on the lattice  $\alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$ , and inverting the isomorphism to obtain  $\boldsymbol{\gamma}^\circ = \kappa^{-1}(\kappa(\mathbf{g})^\circ)$ . Figure 5.10 visualizes this procedure. And even more, we can make use of the efficient 1D STFT routine to compute the Gabor analysis and synthesis operations using that isomorphism.

The urge for the width and height of an image to be relatively prime to benefit from that isomorphism is not a major restriction in practice. Considering large images and a good localization of the analyzing prototype, the well-localized windowed image

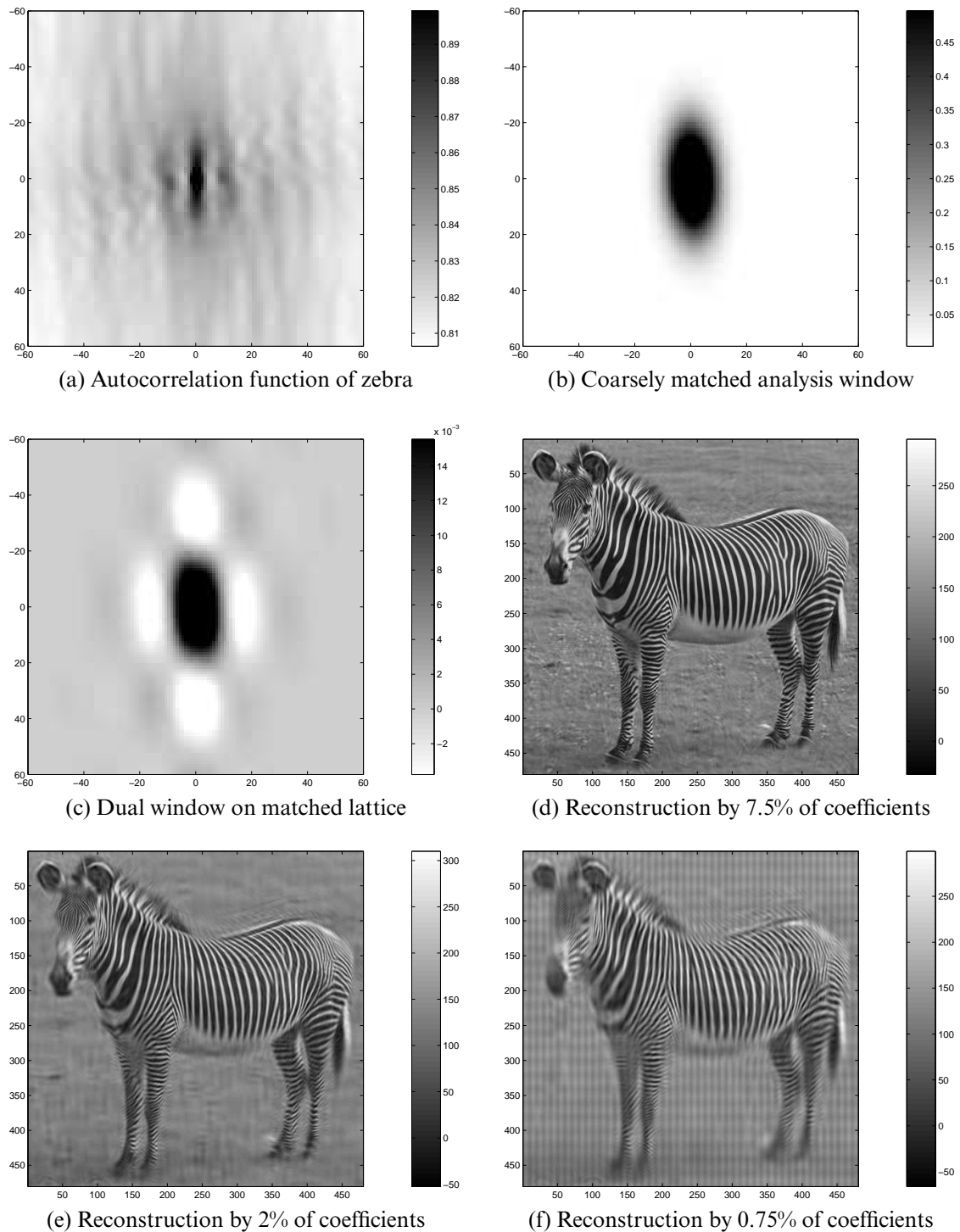


FIGURE 5.11: Thresholding by non-separable window on a fully separable lattice. The window is a 2D Gaussian, dilated by  $\frac{2}{3}$  and rotated by 0.05. The lattice is defined by  $(\alpha_1, \alpha_2, \beta_1, \beta_2) = (20, 10, 16, 32)$  with redundancy 2.25.

$\theta$	0	45.8784	138.1565	277.0584
$ \Phi $	518400	38880	10368	3888
$\ \tilde{f} - f\ _{\mathbb{F}}$	$9.4573 \times 10^{-15}$	0.1010	0.1678	0.2307

TABLE 5.4: Results of thresholding by non-separable atom on separable lattice

cutouts allow for a partition of the image into smaller parts that keep the format requirements.

### 5.4.2 Matching the Atom to the Signal

Having the freedom to adapt the analysis atom to a given image or class of images raises the question about how this could be achieved. In the publication [FKP96] of Feichtinger, Prinz and Kozek an approach was mentioned to adapt the analyzing prototype to an autocorrelation estimate of the image. The (cyclic) autocorrelation estimate for an image  $f \in \mathbb{C}^{L_1 \times L_2}$  is given as

$$\tilde{r}_f(k_1, k_2) := \langle f, T_{(k_1, k_2)} f \rangle_{\mathbb{F}}, \quad (k_1, k_2) \in \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}.$$

Subfigure 5.11a shows the (cropped) autocorrelation function of the zebra image. It indicates a large vertical correlation width like we'd expect it from the dominating line patterns on the body of the animal. The first step is now to match the analysis atom to that estimate, and be it only coarsely. Subfigure 5.11b shows an example for the occurring non-separable window.

However, adapting the window is not enough. If an atom is stretched or shrunken, the lattice should be stretched or shrunken as well to avoid a decreased localization of the dual atom. In our experiment we try to keep the dual well-localized by adapting the shift parameters as well. Subfigure 5.11c shows the dual of the mentioned atom on the lattice determined by  $(\alpha_1, \alpha_2, \beta_1, \beta_2) = (20, 10, 16, 32)$ , keeping the redundancy at 2.25.

Using this atom, lattice and dual we want to repeat the thresholding experiment. The thresholds are obtained by the same way as in Experiment 5.2.3. The corresponding filtered reconstructions are shown in Figure 5.11 and the emerging numerical values in Table 5.4. Compared to the first experiment with a separable atom, i.e., Figure 5.4 on page 77 and Table 5.2 on page 78, the results aren't really better for the matched non-separable atom, except in the case of the highest threshold, where the (norm) difference between the reconstruction and the original is a little less. Subfigure 5.11f seems to reveal a few more details, although the distribution of the duals on the image domain is prominent.

## 5.5 General 2D Atoms on General 4D Lattices

Without considering PF-lattices that are composed as a product of two non-separable 2D TF-lattices, we want to deal with the idea of general atoms on general 4D lattices

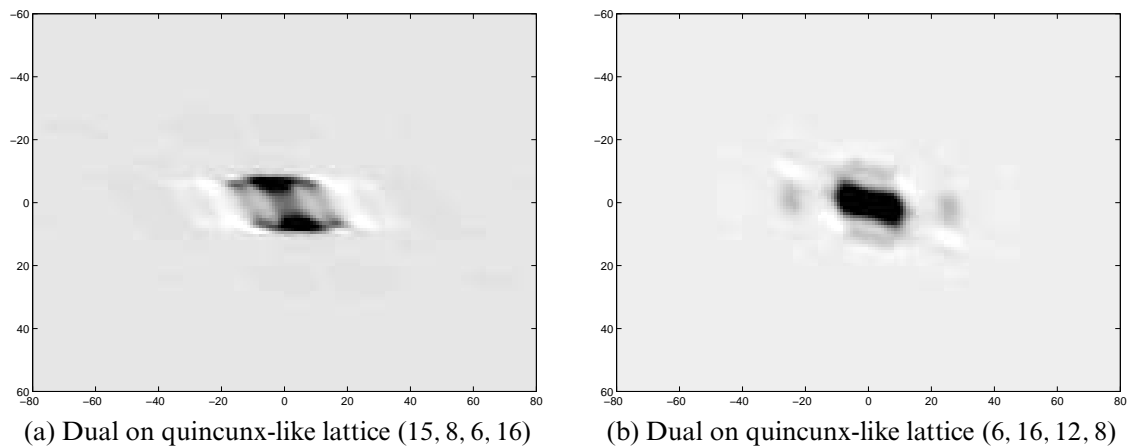


FIGURE 5.12: 2D non-separable dual atoms on 4D quincunx-like lattices. The lattice in (a) uses the same parameters as that of Figure 5.1, but the sampling points on the frequency domain are shifted by  $(\frac{\beta_1}{2}, \frac{\beta_2}{2}) = (3, 8)$  alternately.

in this section, because there seems to be no major advantage if the lattice is partially separable. There is an implementation by Prinz to compute the dual for a certain 4D quincunx-like case, what we demonstrate here. The function `ppdw2qx.m` computes the dual for a lattice that is defined by shift parameters  $\alpha_1, \alpha_2, \beta_1, \beta_2$  where  $\beta_1$  and  $\beta_2$  are even numbers. With these even frequency shift parameters it is possible to form a quincunx-like lattice by shifting the frequency sampling points by half of their distance at every second time-position. This is illustrated in Figure 3.2 on page 36 for a 2D TF-lattice.

If we think of traditional non-separable 2D TF-lattices, we only have a certain subset (actually a sub-subgroup) of frequency-shifts involved at every time-position, or vice versa. If the same set of frequencies were used at every position, we're back to the case of a separable lattice. A non-separable 4D lattice could start with a grid on the image domain, but every point on that grid incorporates a different subset of modulations. By using even frequency shift parameters, we're able to build a grid determined by  $(\beta_1, \beta_2)$  on the frequency domain and include the  $(\frac{\beta_1}{2}, \frac{\beta_2}{2})$ -shifted version of that grid to form a partition of a quincunx-like subgroup on  $\overline{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}$ . These grids are now taken alternately through the position sampling points. This is the case that Prinz implemented in `ppdw2qx.m`, where the grid on the image domain is determined by  $(\alpha_1, \alpha_2)$  and the grid on the frequency domain by  $(\beta_1, \beta_2)$  that is shifted by  $(\frac{\beta_1}{2}, \frac{\beta_2}{2})$  alternately. Figure 5.12 shows two examples for resulting dual windows on lattices of the mentioned type. They indicate a better localization than those on the grids that use the same shift parameters.

We also want to repeat the thresholding experiment on the zebra image by using a lattice of the above kind. However, the computation of duals and of the Gabor analysis and synthesis operations of images is still quite cumbersome for general 2D atoms on general PF-lattices. Although efficient Gabor matrix factorizations are known, there are currently no computationally quick implementations of the mentioned operations. Plans to include efficient code for general lattices into Søndergaard's *Linear*

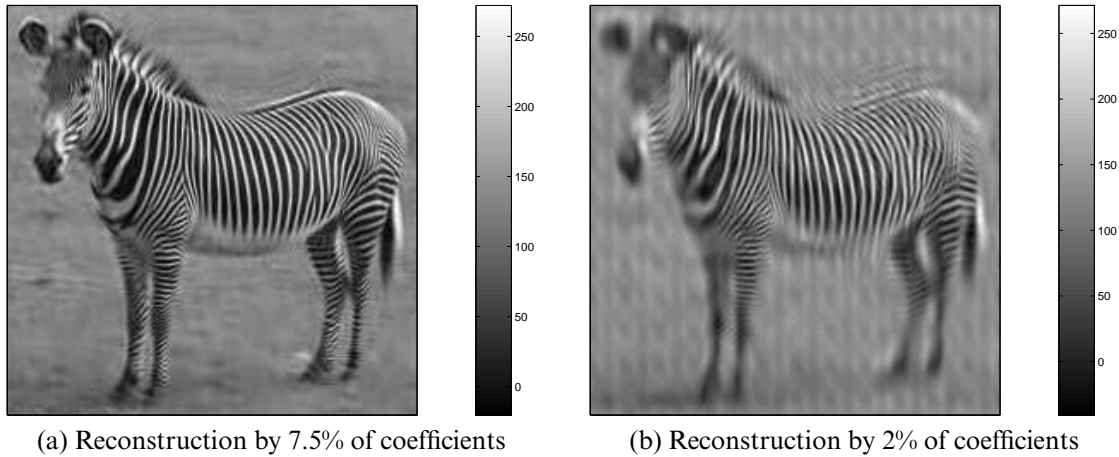


FIGURE 5.13: Thresholding by non-separable atom on 4D quincunx-like lattice

$\theta$	0	187.7358	57.4567
$ \Phi $	129600	2592	9720
$\ \tilde{f} - f\ _{\mathbb{F}}$	$3.4950 \times 10^{-14}$	0.2148	0.1267

TABLE 5.5: Results of thresholding by general atom on 4D quincunx-like lattice

*Time-Frequency Analysis Toolbox*<sup>1</sup> (LTFAT) have been delayed. The calculation of a dual and the analysis and synthesis operations had to be done for a downsampled  $240 \times 240$ -version of the zebra image, what still took an impressively long time. Results of the thresholding experiment are shown in Figure 5.13 and Table 5.5. The quincunx-like 4D lattice was determined by the same shift parameters as in Experiment 5.2.3.

The present computational implementations are only efficient if they're fed with reasonably small amounts of data. There seem to be two approaches to handle larger images by present code:

- (1) Split an image into smaller, maybe overlapping tiles and do Gabor analysis and synthesis of those tiles of decreased size, or
- (2) Downsample image and atom, compute the dual of the downsampled atom by using equivalent lattice parameters, do Gabor analysis and synthesis, and sample the result up to the original size.

The first approach raises the question of how to get rid of border phenomena at the tile borders that could appear after processing, because the DFT considers each tile as being periodic at its borders. These artifacts might make it difficult to stitch the processed tiles together to a complete image. This could be handled by choosing the tiles in an overlapping manner and blend the processed image tiles at their borders in a certain way. This approach is not handled further in this thesis, we'll rather take a closer look at the idea of downsampling.

<sup>1</sup><http://lftat.sourceforge.net/>

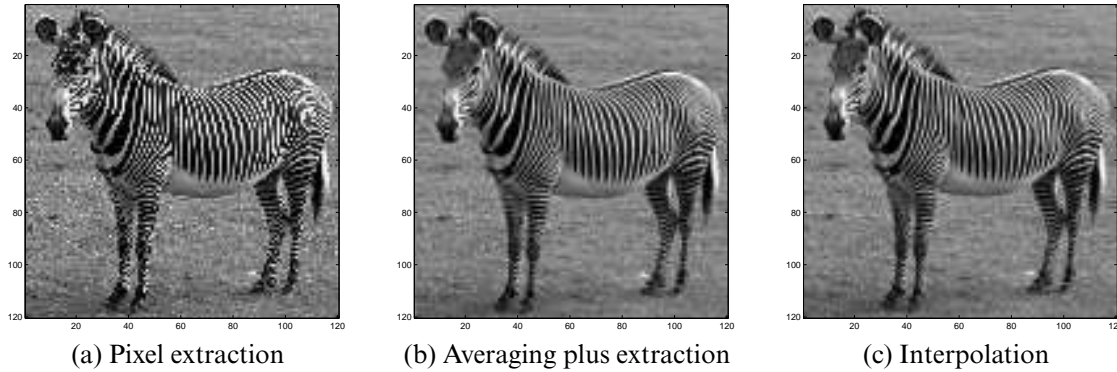


FIGURE 5.14: Comparison of image downsampling methods

### 5.5.1 Data Reduction by Downsampling

In this subsection we briefly demonstrate an efficient signal downsampling mechanism that makes use of the FFT, and show that the dual of a downsampled window is numerically equal to the dual of the original window if the lattice parameters can be scaled in the same manner. For simplicity, we express the idea for 1D signals first.

The challenge is to scale the samples of a signal  $f \in \mathbb{C}^L$  of length  $L$  reasonably into a new vector of length  $M = \frac{m}{n}L \in \mathbb{N}$  for applicable  $m, n \in \mathbb{N}$ , no matter if  $M$  is smaller or larger than  $L$ . Note that  $m = M$  and  $n = L$  are always valid choices. If  $M$  is a fraction of  $L$ , i.e.  $M = \frac{L}{n}$ , there could be the idea to extract every  $n$ -th entry of the signal to obtain  $(f(0), f(n), \dots, f(L-n)) \in \mathbb{C}^M$ . But a single sample  $f(j)$  does not represent the behavior in the segment  $(f(j - \lfloor \frac{n}{2} \rfloor), \dots, f(j + \lceil \frac{n}{2} \rceil))$ . The signal should therefore be averaged such that the samples at the positions  $\{0, n, \dots, L-n\}$  contain the average of the signal in their neighborhood of length  $n$ . This is achieved by convolving the signal with a window that has an essential support of length  $n$ , prior to extracting the samples. To grant each position the same weight, the window should be a box function with ones at  $\{0, \dots, n-1\}$ . Figure 5.14 shows examples for scaling the zebra image down to size  $120 \times 120$ . Subfigure 5.14a is the result of pure pixel extraction. Compared to subfigure 5.14b, which combines averaging with pixel extraction, the result is smoother in the latter case. The third example is a downscaling obtained by current image processing software.

The remaining problem is that of sampling a signal of length  $L$  up to length  $mL$ , because if this is achieved, the downsampling to size  $M = \frac{mL}{n}$  can be done by the above method. Having  $f = (f(0), \dots, f(L-1))$ , we want to get an upsampled version  $F = (F(0), \dots, F(mL-1))$  where  $F(mj) = f(j)$  for  $j \in \langle L \rangle$ . How do we get the samples in between as an interpolation of the known values? We want to show how this can be done using the DFT. The main idea is to let the DFT tell us what oscillations of length  $mL$  are contained if we only know each  $m$ -th sample. Recovering the signal from its DFT is obtained by

$$f(j) = \frac{1}{L} \sum_{k=0}^{L-1} \hat{f}(k) e^{2\pi i j k / L}, \quad j \in \mathbb{Z}_L.$$



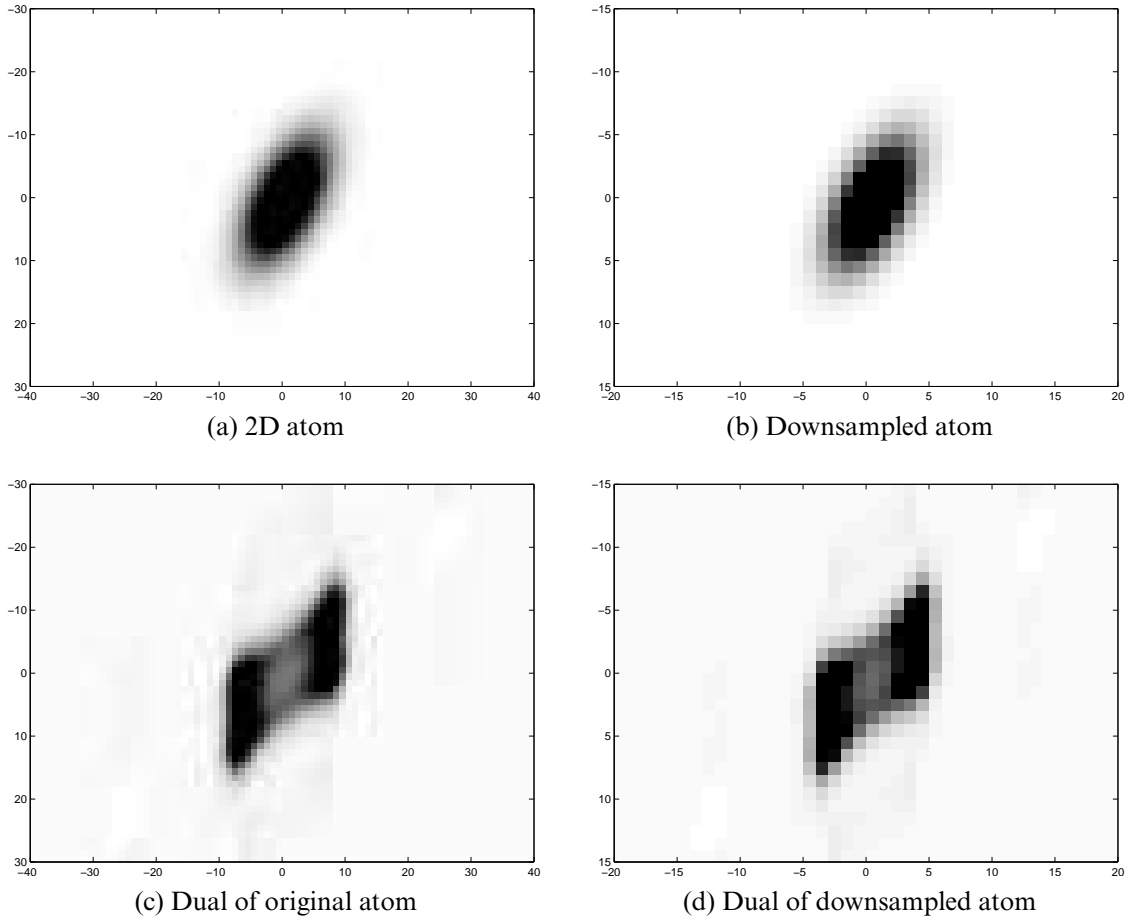


FIGURE 5.15: Dual 2D atoms by downsampling

Thus, we are seeking a sequence  $F$  such that

$$F(mj) = \frac{1}{L} \sum_{k=0}^{L-1} \hat{f}(k) e^{2\pi i mjk/(mL)}, \quad j \in \langle L \rangle.$$

This summation can be extended up to the index  $mL - 1$  by defining  $\hat{f}(k) := 0$  for  $L \leq k \leq mL - 1$ , and we can now consider

$$F(j) = \frac{1}{L} \sum_{k=0}^{mL-1} \hat{f}(k) e^{2\pi i jk/(mL)} = \frac{1}{mL} \sum_{k=0}^{mL-1} \hat{F}(k) e^{2\pi i jk/(mL)}, \quad j \in \mathbb{Z}_{mL}.$$

This means that we can derive the desired function  $F \in \mathbb{C}^{mL}$  simply by defining its DFT as

$$\begin{aligned} \hat{F} &= (\hat{F}(0), \dots, \hat{F}(L-1), \hat{F}(L), \dots, \hat{F}(mL-1)) \\ &:= (m\hat{f}(0), \dots, m\hat{f}(L-1), 0, \dots, 0). \end{aligned}$$

But this is not the only solution for the condition  $F(mj) = f(j)$  for  $j \in \langle L \rangle$ . In fact, it is fulfilled for all shifts  $T_{\ell L} \hat{F}$  for  $\ell \in \langle m \rangle$ , as the sequence  $\hat{f}$  is periodic with

$L$	$\frac{m}{n}$	$M$	$\alpha$	$\frac{m}{n}\alpha$	$\beta$	$\ \gamma_M^\circ - \eta^\circ\ _F$
144	$\frac{3}{4}$	108	12	9	9	$1.1136 \times 10^{-4}$
144	$\frac{2}{3}$	96	12	8	8	$1.1023 \times 10^{-4}$
240	$\frac{2}{3}$	160	15	10	10	$2.9901 \times 10^{-6}$

TABLE 5.6: Accuracy of downsampled dual Gaussian windows

period  $L$ . One has to take care to also map the negative frequencies the correct way from  $\mathbb{C}^L$  to  $\mathbb{C}^{mL}$ , and so the desired solution is given by

$$\hat{F} := (m\hat{f}(0), \dots, m\hat{f}(\lceil \frac{L}{2} \rceil - 1), 0, \dots, 0, m\hat{f}(-\lfloor \frac{L}{2} \rfloor), \dots, m\hat{f}(-1)) .$$

The approach of doing data reduction for Gabor analysis by downsampling could have various depths:

- Sample a signal in  $\mathbb{C}^L$  down to  $\mathbb{C}^M$  once and do all calculations by a lattice in  $\mathbb{C}^M$ , or
- Find the Gabor parameters (atom and lattice) already in  $\mathbb{C}^L$  and transform these settings and the signal to  $\mathbb{C}^M$ , where analysis and synthesis is done, or
- Only map the Gabor parameters from  $\mathbb{C}^L$  to  $\mathbb{C}^M$ , calculate the dual of the downsampled atom, scale it back to length  $L$  and do Gabor analysis of the original signal with the original atom and synthesis with the approximated dual atom.

Obviously, the second and third possibilities raise the question to what extent a sampling lattice can be scaled down as well. We want to see to what extent the dual of a downsampled atom corresponds to the downsampled version of the original dual window. This means, we try to sample a given atom  $g \in \mathbb{C}^L$  down to  $h \in \mathbb{C}^M$  with  $M < L$ , compute its dual  $\eta^\circ \in \mathbb{C}^M$  with respect to a certain lattice  $\Lambda_M \trianglelefteq \mathbb{Z}_M \times \widehat{\mathbb{Z}}_M$  and see whether it is similar to the original dual  $\gamma^\circ \in \mathbb{C}^L$  with respect to a lattice  $\Lambda \trianglelefteq \mathbb{Z}_L \times \widehat{\mathbb{Z}}_L$  after sampling it down to  $\gamma_M^\circ \in \mathbb{C}^M$ . Considering that  $M = \frac{m}{n}L$ , we notice that we have to keep the redundancy as

$$\text{red}_\Lambda = \frac{N}{L} = \frac{\frac{m}{n}N}{\frac{m}{n}L} = \text{red}_{\Lambda_M} .$$

For a product lattice  $\Lambda = \alpha\mathbb{Z}_L \times \beta\mathbb{Z}_L$  we get  $|\Lambda| = N = \frac{L^2}{\alpha\beta}$ , and thus an equivalent lattice in  $\mathbb{Z}_M \times \widehat{\mathbb{Z}}_M$  would be determined by

$$|\Lambda_M| = \frac{m}{n}N = \frac{M^2}{\alpha_M\beta_M} = \frac{\frac{m^2}{n^2}L^2}{\alpha_M\beta_M} = \frac{\frac{m}{n}L^2}{\alpha\beta} .$$

Therefore we either have  $(\alpha_M, \beta_M) = (\frac{m}{n}\alpha, \beta)$  or  $(\alpha_M, \beta_M) = (\alpha, \frac{m}{n}\beta)$ . But it is immediately clear that we may only consider the first case where the time-shift parameter is scaled, because we're scaling the signal domain down and map the oscillations  $j \mapsto e^{2\pi ijk/L}$  in  $\mathbb{C}^L$  to  $j \mapsto e^{2\pi ijk/M}$  in  $\mathbb{C}^M$ . This also reveals some important restrictions on the choice of valid lattice parameters:

- (1)  $\alpha, \beta$  must be divisors of  $L$  such that
- (2)  $\frac{m}{n}\alpha \in \mathbb{N}$ ,
- (3)  $\frac{m}{n}\alpha$  is a divisor of  $M$  and
- (4)  $\beta$  is a divisor of  $M$  as well.

For constructing a quincunx-like lattice, there's another restriction on either  $\beta$  or both  $\alpha$  and  $\frac{m}{n}\alpha$  to be even. This might make it rather difficult to find matching parameters for a fixed  $M$ , but as the choice of  $M$  is another degree of freedom, the problem isn't that severe.

Figure 5.15 shows that the dual of a 2D downsampled atom looks similar to the dual of the original 2D atom. Table 5.6 shows some results of the accuracy of this approach by the example of 1D Gaussians.

## 5.6 Discussion and Outlook

The aim of this thesis was to examine the path that leads from general Gabor analysis to implementations of two-dimensional finite discrete Gabor expansions of images. We have shown how to understand the 2D elementary oscillations of the 2D Fourier transform as building blocks of images and indicated the visualization problems for the emerging four dimensions of the STFT. The possible non-separability of 4D position-frequency lattices has various depths and intervenes with the non-separability of 2D atoms when it comes to numerical implementations. We were also able to demonstrate known ways to make the analysis and synthesis operations faster.

What we haven't considered here is the accomplishment and implementation of "serious" image processing tasks such as compression, denoising and deblurring by means of the Gabor transform. If reasonable image processing had to be done, one has to get familiar with existing and approved methods and compare their performance with implementations that involve Gabor systems. It is the goal of this final section to provide references to literature and publications that already introduce implementations of image processing tasks, but also to show up in what directions the research could continue.

Discrete Gabor analysis for image processing was addressed e.g. in [Dau88] and [Li94]. The task of image compression by Gabor expansion was treated by Ebrahimi and Kunt in [EK91], where a wavelet-like setting of atoms and TF-shifts was chosen. Cristobal and Navarro used Gabor expansion techniques in [CN94] for localized contrast enhancement to remove clouds from satellite images.

Modulated 2D Gaussian functions ("Gabor filters") were found to be suited for the description of biological vision, modelling the behavior of simple cells in the visual cortex. Publications by Petkov include the tasks of contour detection, texture classification and object recognition, e.g. in [GPK02] or [GPW03]. Another source for object recognition is e.g. [BAJW98].

A topic that this thesis didn't touch so far was the concept of *multi-window* Gabor frames that consist of finitely many different windows  $g_1, \dots, g_r \in L^2(\mathbb{R}^d)$  and possibly different lattices  $\Lambda_1, \dots, \Lambda_r \subseteq \mathbb{Z}^{2d}$ . If the union  $\bigcup_{j=1}^r \mathcal{G}(g_j, \Lambda_j)$  of the (possibly incomplete) Gabor systems  $\mathcal{G}(g_j, \Lambda_j)$  is a frame for  $L^2(\mathbb{R}^d)$  with frame operator

$$Sf = \sum_{j=1}^r \sum_{\lambda_j \in \Lambda_j} \langle f, \pi(\lambda_j)g_j \rangle_{L^2} \pi(\lambda_j)g_j = \sum_{j=1}^r S_{g_j} f ,$$

it could provide some better performance than a frame that only involves a single window. The Fourier analysis of images unveiled that natural images are highly determined by the area-producing oscillations of low frequency, whereas higher frequencies contribute to the edges, contours and sharpness. This suggests the consideration of multi-window Gabor frames for image processing, using 2D windows with good frequency resolution in the lower frequency region and windows with good spatial resolution for the higher frequencies in an image. Publications on this topic are e.g. [ZZP98] or [Li99].

Questions for continued research could be:

- How could multi-2D-window Gabor expansions be performed with general windows on general 4D lattices?

[JT07] proposed algorithms for selecting the best-matching atoms for a signal from a given multi-window dictionary.

- How could such an adaptive multi-window Gabor expansion be applied to the 2D case of images?

In the case of separable 2D atoms that allow for separate Gabor systems for each of the two dimensions, the performance of the 2D frame can be measured by computing the condition numbers of the two independent frame matrices. But we indicated that the matrix representation of a 2D frame might become rather huge.

- How could the performance of general 2D frames be measured without building a huge frame matrix?

We mentioned in Section 1.7 that a tensor  $x \in V \otimes W$  doesn't have to be of a separable type  $x = x_1 \otimes x_2$  with  $x_1 \in V$  and  $x_2 \in W$ , but rather be representable as a sum of separable tensors, i.e.,  $x = \sum_{j=1}^n x_j^{(1)} \otimes x_j^{(2)}$ .

- Does one get a benefit from finding a representation  $\mathbf{g} = \sum_{j=1}^n g_j^{(1)} \otimes g_j^{(2)}$  with  $g_j^{(i)} \in \mathbb{C}^{L_i}$  for a window  $\mathbf{g} \in \mathbb{C}^{L_1 \times L_2}$ ?

The list of questions could continue like:

- How is the dual 2D window efficiently calculated for a lattice  $\Lambda = \Omega \times \hat{\Omega}$  with non-separable subgroups  $\Omega \subseteq \mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}$  and  $\hat{\Omega} \subseteq \widehat{\mathbb{Z}_{L_1} \times \mathbb{Z}_{L_2}}$ ?

- Can the Gabor analysis and synthesis operations be made significantly quicker when splitting a general lattice into a union of shifts of a single grid? Could this approach also ease the computation of the dual window?
- How is a Gabor expansion obtained by splitting an image into tiles of smaller size?
- What is the exact mathematical background that the correct downsampling of atom and lattice keep the behavior of the dual window?
- What other methods could be proposed to make the computation of 2D Gabor expansions quicker?

This collection of references and questions does in no way claim to be complete.



# Appendix A

## MATLAB Code

Here the author provides the functions that were involved to produce the numerical experiments and printed figures in this thesis. They were also tested to work with Octave 2.9 and build upon code from the NuHAG MATLAB toolboxes<sup>1</sup>. The functions by the author, A.1–A.6, are published under the GNU General Public License, version 2. `stft2q.m` and `istft2q.m` are derived from code by P. Prinz.

### A.1 `nsgauss.m` – Non-Separable 2D Gaussian

```
1 function g=nsgauss(p,q,vdil,hdil,rot)
2 % Computes a non-separable (dilated + rotated) 2D Gaussian
3 % Usage: g = nsgauss(p, q, vdil, hdil, rot);
4 % Input: p,q .... size of g
5 %         vdil ... vertical dilation factor (before rotation)
6 %         hdil ... horizontal dilation factor (before rotation)
7 %         rot .... rotation angle, e.g. pi/4
8 % Example:
9 % norm(nsgauss(p,q,1,1,0) - gaussnk(p)'*gaussnk(q)) == eps
10
11 % Version 0.2-20070525
12 % by Stephan Paukner <stephan+math at paukner dot cc>
13 % Licensed under the GNU General Public License v2
14
15 D=[1/vdil 0; 0 1/hdil]; %dilation matrix
16 R=[cos(rot) -sin(rot); sin(rot) cos(rot)]'; %rotation matrix
17
18 sp=sqrt(p); sq=sqrt(q);
19 g=zeros(1,p*q);
20 for jp=-3:3
21     for jq=-3:3
22         [x y]=meshgrid( (0:p-1)/sp+jp*sp, (0:q-1)/sq+jq*sq );
23         v=D*R*[x(:)'; y(:)'];
24         g=g+exp(-pi*(v(1,:).^2 + v(2,:).^2));
25     end
26 end
```

---

<sup>1</sup><http://www.univie.ac.at/nuhag-php/mmodule/>

```

27 g=reshape(g,q,p)';
28 g=g/norm(g,'fro');

```

## A.2 acf2.m – 2D Autocorrelation Function

```

1 function ac=acf2(img);
2 % Implementation of the ACF of an image
3 % Usage: ac=acf2(img);
4 % Input: img ... image as matrix of size [p q]
5 % Output: ac .... ACF of size [p q]
6
7 [p,q]=size(img);
8 ac=zeros(p,q);
9 %im=img(:);
10 %ii=1;
11 for ll=1:q
12     disp([num2str(ll) '/' num2str(q)])
13     for kk=1:p
14         %ac(ii)=im.'*rot(im,ii-1); %too slow
15         %ii=ii+1;
16         A=img.*rotrc(img,kk-1,ll-1);
17         ac(kk,ll)=sum(A(:));
18     end
19 end

```

## A.3 stft2sep.m – STFT: Separable 2D Atom, Separable Lattice

```

1 function c=stft2sep(img,g1,g2,a1,a2,b1,b2)
2 % Computes the Gabor analysis coefficient for an image in the
3 % case of a separable 2D atom g1.*g2 and a fully separable
4 % 4D lattice, determined by the shift parameters a1,a2,b1,b2,
5 % by using stft.m.
6 %
7 % Usage: c = stft2sep(img, g1,g2, a1,a2, b1,b2);
8 %         c = stft2sep(img,g,a,b); % If img is square
9 %
10 % Parameters: img ..... Image of size [L1 L2]
11 %             g1,g2 ... Atoms with length(g1)==L1 and length(g2)==L2
12 %             a1,a2 ... Position shift parameters where a1|L1 and a2|L2
13 %             b1,b2 ... Frequency shift parameters where b1|L1 and b2|L2
14 %
15 % Example: c == gabbasp(g1,a1,b1) * img * gabbasp(g2,a2,b2).';
16 %

```



```

17 if nargin~=4 & nargin~=7
18     error('Invalid_number_of_arguments!');
19 elseif nargin==4
20     b1=a1; b2=a1;    %freq shift is at a1
21     a1=g2; a2=g2;    %pos shift is at g2
22     g2=g1;          %atom is at g1
23 end
24
25 [L1 L2]=size(img);
26 if length(g1)~=L1 | length(g2)~=L2
27     error('The_lengths_of_the_atoms_have_to_match_the_image_size!');
28 end
29 tst=L1/a1*L1/b1*L2/a2*L2/b2;
30 if tst~=round(tst)
31     error('One_of_the_shift_parameters_does_not_divide_one_of_the_lengths!');
32 end
33
34 N2=L2^2/a2/b2;
35 b=zeros(L1,N2);
36 for kk=1:L1    %lines in image
37     b(kk,:)=reshape(stft(img(kk,:),g2,a2,b2), 1, N2);
38 end
39 N1=L1^2/a1/b1;
40 c=zeros(N1,N2);
41 for kk=1:N2
42     c(:,kk)=reshape(stft(b(:,kk)'.',g1,a1,b1), 1, N1)';
43 end

```

## A.4 istft2sep.m – Inverse STFT in Separable Setting

```

1 function rimg=istft2sep(c,gd1,gd2,a1,a2,b1,b2)
2 % Inverse operation of c = stft2sep(img, g1,g2, a1,a2, b1,b2)
3 %           or c = gabbasp(g1,a1,b1)*img*gabbasp(g2,a2,b2)'.
4 %
5 % Usage: rimg = istft2sep(c, gd1,gd2, a1,a2, b1,b2)
6 %           rimg = istft2sep(c, gd,a,b)    % If image was square
7 % (The shift parameters are needed for correct input to istft)
8 %
9 % Parameters: c ..... The 2D Gabor coefficient
10 %              gd1,gd2 ... Dual atoms to g1 and g2
11 %              a1,a2 ..... 2D shift parameters
12 %              b1,b2 ..... 2D modulation parameters
13 %
14 % Output:      rimg ..... The reconstructed image

```

```

15 %
16 if nargin~=4 & nargin~=7
17     error('Invalid_number_of_arguments!');
18 elseif nargin==4
19     b1=a1; b2=a1;      %freq shift is at a1
20     a1=gd2; a2=gd2;  %pos shift is at gd2
21     gd2=gd1;        %atom is at gd1
22 end
23
24 [N1 N2]=size(c);
25 L1=sqrt(N1*a1*b1); L2=sqrt(N2*a2*b2); % image size
26 ac1=L1/a1; bc1=L1/b1; ac2=L2/a2; bc2=L2/b2; %number of channels
27 dummy=[];
28 b=zeros(L1,N2); ring=zeros(L1,L2);
29
30 for kk=1:N2 % in N1, out L1
31     b(:,kk)=istft(reshape(c(:,kk)',bc1,ac1),dummy,gd1);
32 end
33 for kk=1:L1 % in N2, out L2
34     ring(kk,:)=istft(reshape(b(kk,:),bc2,ac2),dummy,gd2);
35 end

```

## A.5 gabfc.m – Gabor Matrix, Modulation Priority

```

1 function Gc=gabfc(G,fgap)
2 % Reorder Gabor matrix to center the frequency shifts
3 % (0 to middle).
4 % Gc=gabfc(G,fgap) where G=gabbasp(g,tgap,fgap)
5 %
6 [N L]=size(G);
7 tgap=L^2/N/fgap;
8 Gc=zeros(N,L);
9 % L/tgap blocks with L/fgap modulations, these must be centered
10 x=1:L/fgap;
11 xc=fftshift(x);
12 for t=0:L/tgap-1
13     Gc(xc+t*L/fgap,:)=G(x+t*L/fgap,:);
14 end

```

## A.6 gabt f.m – Gabor Matrix, Translation Priority

```

1 function H=gabtf(G,tgap)
2 % Reorder Gabor matrix to first contain the time shifts,
3 % then freq.
4 % H=gabtf(G,tgap) where G=gabbasp(g,tgap,fgap)

```

```

5 %
6 [N L]=size(G);
7 fgap=L^2/N/tgap;
8 H=zeros(N,L);
9 x=1:L/fgap:N; % Span lines of G with steps of
10 % the number of f-channels
11 for t=0:L/fgap-1 %How many blocks with shifts of a certain mod?
12     H((1:L/tgap)+t*L/tgap,:)=G(x+t,:);
13 end

```

## A.7 stft2q.m – STFT: General 2D Atom, Quincunx-Like Lattice

```

1 function ST = stft2q(X,G,a1,a2,b1,b2);
2 % STFT of image with general 2D atom on quincunx-like
3 % 4D lattice, derived from Prinz' gabbas2q.m
4 %
5 % Usage: ST = stft2q(X,G,a1,a2,b1,b2)
6 %
7 % Input: X ..... Image of size [L1 L2]
8 %        G ..... Window of size [L1 L2]
9 %        a1,a2,b1,b2 .. Lattice parameters !!with EVEN b1,b2!!
10 %
11 % Output: ST ..... STFT coefficients
12 %
13 X=X(:).';
14 [m,n] = size(G);
15 red=(n*m)/(a1*a2*b1*b2)
16 count = 0;
17
18 ST = zeros(1,n*m*red);
19 disp(['Size:_' num2str(n*m*red)])
20
21 for k1 = 0 : a1 : m-a1
22     disp(['_' num2str(k1) '/' num2str(m-a1)])
23     for k2 = 0 : a2 : n-a2
24         disp([num2str(k2) '/' num2str(n-a2)]) %verbose
25         if rem(k1/a1,2) % if k1/a1 is odd then TRUE
26             a1q = b1/2;
27         else
28             a1q = 0;
29         end
30         if rem(k2/a2,2) % if k2/a2 is odd then TRUE
31             a2q = b2/2;
32         else
33             a2q = 0;

```

```

34     end
35     grf = fft2(rotrc(G,k1,k2)); % for the modulation
36     for k4 = 0 : b2 : n - b2
37         for k3 = 0 : b1 : m - b1
38             g=ifft2(rotrc(grf,k3+a1q,k4+a2q)); %modulation
39             count = count+1;
40             ST(count) = X*g(:);
41         end
42     end
43 end
44 end

```

## A.8 `istft2q.m` – Inverse STFT in General Setting

```

1 function X = istft2q(ST,GD,a1,a2,b1,b2);
2 % Inverse operation of stft2q.m, derived from Prinz' gabbas2q.m
3 % Usage: X = istft2q(ST,GD,a1,a2,b1,b2)
4 %
5 % Input: ST ..... Result of stft2q(X,G,a1,a2,b1,b2)
6 %       GD ..... 2D dual atom to G
7 %       a1,a2,b1,b2 .. Lattice parameters !!with EVEN b1,b2!!
8 %
9 % Output: X ..... Reconstructed image
10 %
11
12 %The dual is already conjugated:
13 GD=GD'; GD=GD.';
14 [m,n] = size(GD);
15 red=(n*m)/(a1*a2*b1*b2)
16 count = 0;
17
18 X = zeros(n*m,1);
19 %disp(['Size: ' num2str(n*m*red)])
20
21 for k1 = 0 : a1 : m-a1
22     disp([num2str(k1) '/' num2str(m-a1)])
23     for k2 = 0 : a2 : n-a2
24         if rem(k1/a1 ,2) % if k1/a1 is odd then TRUE
25             a1q = b1/2;
26         else
27             a1q = 0;
28         end
29         if rem(k2/a2 ,2) % if k2/a2 is odd then TRUE
30             a2q = b2/2;
31         else
32             a2q = 0;

```

```
33     end
34     gdrf = fft2(rottrc(GD,k1,k2)); % for the modulation
35     for k4 = 0 : b2 : n - b2
36         for k3 = 0 : b1 : m - b1
37             gd=ifft2(rottrc(gdrf,k3+a1q,k4+a2q)); %modulation
38             count = count+1;
39             gd=gd(:)';
40             X = X + ST(count)*gd(:);
41         end
42     end
43 end
44 end
45 X=reshape(X,m,n);
```



# Bibliography

- [BAJW98] J. Ben-Arie and Z. Wang. Gabor kernels for affine-invariant object recognition. In H.G. Feichtinger and T. Strohmer, editors, *Gabor analysis and algorithms: Theory and applications*. 1998.
- [BvL98a] M.J. Bastiaans and A.J. van Leest. From the rectangular to the quincunx Gabor lattice via fractional Fourier transformation. *IEEE Signal Proc. Letters*, 5(8):203–205, 1998.
- [BvL98b] M.J. Bastiaans and A.J. van Leest. Product forms in Gabor analysis for a quincunx-type sampling geometry. In J.P. Veen, editor, *Proc. CSSP-98, ProRISC/IEEE Workshop on Circuits, Systems and Signal Processing, Mierlo, Netherlands, 16–17 November 1998*, pages 23–26, Utrecht, Netherlands, 1998. STW, Technology Foundation.
- [Chr03] O. Christensen. *An Introduction to Frames and Riesz Bases*. Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, 2003.
- [CN94] G. Cristobal and R. Navarro. Space and frequency variant image enhancement based on a Gabor representation. *Pattern Recognition Letters*, 15(3):273–277, 1994.
- [Dau88] J.G. Daugman. Complete discrete 2-D Gabor transform by neural networks for image analysis and compression. *IEEE Trans. ASSP*, 36(7):1169–1179, 1988.
- [EK91] T. Ebrahimi and M. Kunt. Image compression by Gabor expansion. *Opt. Eng.*, 30(7):873–880, July 1991.
- [FG94] H.G. Feichtinger and K. Gröchenig. Theory and practice of irregular sampling. In J.J. Benedetto and M. Frazier, editors, *Wavelets: mathematics and applications*, Studies in Advanced Mathematics, pages 305–363, Boca Raton, FL, 1994. CRC Press.
- [FK97] H.G. Feichtinger and N. Kaiblinger. 2D-Gabor analysis based on 1D algorithms. In *Proc. OEAGM-97 (Hallstatt, Austria)*, 1997.
- [FKP96] H.G. Feichtinger, W. Kozek, and P. Prinz. Gabor systems with good TF-localization and applications to image processing. In *Proc. ICIP-96*, volume 1, pages 249–252. IEEE, 1996.

- [FKPS96] 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.
- [FLW07] H.G. Feichtinger, F. Luef, and T. Werther. A Guided Tour from Linear Algebra to the Foundations of Gabor Analysis. In *Gabor and Wavelet Frames*, volume 10 of *IMS Lecture Notes Series*. 2007.
- [FSC95] H.G. Feichtinger, T. Strohmer, and O. Christensen. A group-theoretical approach to Gabor analysis. *Opt. Eng.*, 34:1697–1704, 1995.
- [FZ98] H.G. Feichtinger and G. Zimmermann. A Banach space of test functions for Gabor analysis. In H.G. Feichtinger and T. Strohmer, editors, *Gabor analysis and algorithms: Theory and Applications*, Applied and Numerical Harmonic Analysis, pages 123–170, Boston, MA, 1998. Birkhäuser Boston.
- [Gab46] D. Gabor. Theory of communication. *J. IEE*, 93(26):429–457, 1946.
- [GPK02] S.E. Grigorescu, N. Petkov and P. Kruizinga. Comparison of texture features based on Gabor filters. *IEEE Trans. on Image Processing*, 11(10):1160–1167, 2002.
- [GPW03] C. Grigorescu, N. Petkov and M.A. Westenberg. Contour detection based on nonclassical receptive field inhibition. *IEEE Trans. on Image Processing*, 12(7):729–739, 2003.
- [Grö98] K. Gröchenig. Aspects of Gabor analysis on locally compact abelian groups. In H.G. Feichtinger and T. Strohmer, editors, *Gabor analysis and algorithms: Theory and Applications*, pages 211–231. Birkhäuser Boston, Boston, MA, 1998.
- [Grö01] K. Gröchenig. *Foundations of Time-Frequency Analysis*. Appl. Numer. Harmon. Anal. Birkhäuser Boston, Boston, MA, 2001.
- [GvL96] G. Golub and C.F. van Loan. *Matrix computations. 3rd ed.* The Johns Hopkins University Press, Baltimore, MD, 3rd ed. edition, 1996.
- [JT07] F. Jaillet and B. Torresani. Time-frequency jigsaw puzzle: adaptive multiwindow and multilayered Gabor expansions, 2007.
- [Li94] S. Li. Nonseparable 2D-discrete Gabor expansions for image representation and compression. In *Proc. IEEE ICIP-94*, Austin, 1994.
- [Li99] S. Li. Discrete multi-Gabor expansions. *IEEE Trans. Inf. Theory*, 45(6):1954–1967, 1999.
- [Pri96] P. Prinz. Calculating the dual Gabor window for general sampling sets. *IEEE Trans. Signal Process.*, 44(8):2078–2082, 1996.



- [Søn07] P.L. Søndergaard. *Finite Discrete Gabor Analysis*. PhD thesis, Technical University of Denmark, 2007.
- [Str97] T. Strohmer. Numerical algorithms for discrete Gabor expansions. In H.G. Feichtinger and T. Strohmer, editors, *Gabor Analysis and Algorithms: Theory and Applications*, pages 267–294. Birkhäuser Boston, Boston, 1997.
- [SW01] U. Storch and H. Wiebe. *Lehrbuch der Mathematik, Band 4: Analysis auf Mannigfaltigkeiten – Funktionentheorie – Funktionalanalysis*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 2001.
- [Wer97] D. Werner. *Funktionalanalysis. 2., überarb. Aufl.* Springer-Verlag, Berlin, 1997.
- [Zim98] G. Zimmermann. Eigenfunctions of the Fourier Transform. Technical report, 1998.
- [ZZP98] Y.Y. Zeevi, M. Zibulski, and M. Porat. Multi-window Gabor schemes in signal and image representations. In H.G. Feichtinger and T. Strohmer, editors, *Gabor analysis and algorithms: Theory and Applications*, Appl. Numer. Harmon. Anal., pages 381–407. Birkhäuser Boston, Boston, MA, 1998.



# Curriculum Vitae

Name: Stephan PAUKNER  
Born: November 22nd, 1977  
Nationality: Austria  
E-mail: [stephan\(AT\)paukner.cc](mailto:stephan(AT)paukner.cc)

1988–1996 Federal grammar school (Bundesrealgymnasium Waidhofen/Th.),  
school leaving examination (Matura) with distinction  
1996–1997 Military service at the Austrian Federal Armed Forces  
1997–2000 Study of Physics at the University of Vienna (unfinished)  
1997–2007 Study of Mathematics at the University of Vienna  
2003–2007 Software developer at SOL4 IT-Consulting GmbH, Vienna