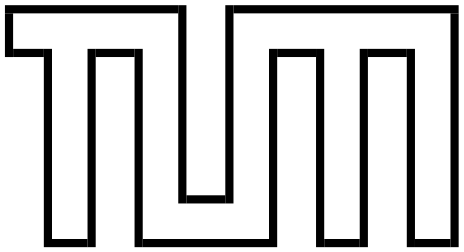


Technische Universität München

Faculty of Mathematics

Efficient Low-Resolution Approximation of Ultrasound Images

Master's Thesis

Johannes Nagler

March 26, 2012

Supervisor: Prof. Dr. Rupert Lasser

Advisor: Dr. Laurent Demaret
Dipl. Ing. Peter Amon

Declaration of Authorship

"I hereby declare that I have written the Master's Thesis on my own and have used no other than the stated sources and aids."

Zusammenfassung

Diese Arbeit untersucht effiziente Methoden, die die Bildgröße von JPEG komprimierten Ultraschall Bildern verkleinern. Zuerst wird das Thema der Arbeit von einem medizinischen Problemfall aus der Praxis her motiviert. Anschließend wird eine mathematische Formulierung dieses Problems gegeben. In Kapitel 2 werden die notwendigen mathematischen Grundlagen eingeführt. Aufgrund der Nachteile des Sampling Theorems von Shannon, gibt das nächste Kaptitel Einblicke in die moderne Sampling Theorie. Kapitel 4 beschäftigt sich mit B-Spline basierten Sampling Methoden, die wegen ihrer ausgezeichnetetn Approximationseigenschaften ausgewählt wurden. Zur Veranschaulichung wird zudem eine Verbindung zu dem Butterworth-Filter hergestellt. Das Runterskalieren von Bildern wird in Kapitel 5 behandelt und verschiedene Methoden werden vorgestellt und nach Bildqualität und Komplexität untersucht. Am Ende wird ein neues hybrides Verfahren vorgestellt. Die numerische Evaluierung in Kapitel 6 belegt die Überlegenheit des hybriden Verfahren gegenüber den anderen vorgestellten Verfahren.

Abstract

This thesis discusses the efficient downscaling of ultrasound images that are compressed by JPEG. First the topic is motivated by a real-world medical scenario and a mathematical formulation of the problem is developed. In Chapter 2, the mathematical foundations to consider this problem are provided. The next chapter introduces recent enhances of modern sampling theory due to the drawbacks of the classical Shannon sampling theorem. Sampling with B-splines is exclusively discussed in Chapter 4 due to their excellent approximation properties, and a connection to the Butterworth filter is outlined. Downscaling methods are shown in Chapter 5, where finally a new hybrid downscaling method is introduced. These downscaling methods are evaluated in Chapter 6, where the provided hybrid downscaling methods outperformed the others in terms of efficiency.

Contents

1. Introduction

In the present-day world, digital images play an important part in life. To reduce the amount of data, down sized versions of large images are provided in many use cases. Some examples are the thumbnails of directories or web sites for image galleries. While these thumbnails usually can be pre-computed, this thesis investigates methods that are able to generate low-resolution images on the fly. Efficient downscaling methods are examined and furthermore the quality of the low resolution approximation is considered. High quality is important to donwscale ultrasound images in health-care scenarios. Ideal efficient downscaling methods have low computational complexity, *i. e.,* they can be performed in real-time and guarantee a high image quality.

First, the topic is motivated by a medical real world scenario and its related problems. As medical images are compressed with the JPEG standard, a short introduction to the JPEG compression standard is given, followed by the downscaling problem formulated as a mathematical problem, concluding with an outline of the current state of the art. Chapter 2 introduces the mathematical background to consider the downscaling problem, with the Shannon sampling theorem as a central point. Based on Shannon's sampling theory, the concepts of the modern sampling theory are established in Chapter 3, while Chapter 4 reveals a useful relation between the sampling with B-Splines and the Butterworth filter. After these theoretical chapters several downscaling methods are discussed extensively and a new hybrid downscaling method is introduced that reduces complexity while preserving image quality. Finally, the downscaling methods are evaluated numerically in Chapter 6 and an outlook for future work is given in Chapter 7.

1.1. Motivation

The investigations of this thesis are motivated by a medical real-world scenario. In this scenario a doctor wants to observe several ultrasound streams at once to detect changes between them. The original sized streams are too large to fit on one single monitor side by side, hence the images have to be downscaled. Figure 1.1 shows an example with 4 downscaled ultrasound images in a 2x2 grid. A long waiting time is not acceptable in this scenario, therefore, a real-time algorithm is crucial. Specifically, medical applications require the guarantee of an image to be nearly without any loss in quality, because the loss of important information can be life-critical. The algorithm has to be efficient in the sense of image quality and computational complexity. Of course, the efficient solution of this problem can be applied to other problems as well. For example, the solution can be applied to a surveillance scenario, where several camera streams should be displayed simultaneously on one monitor for a security guard.

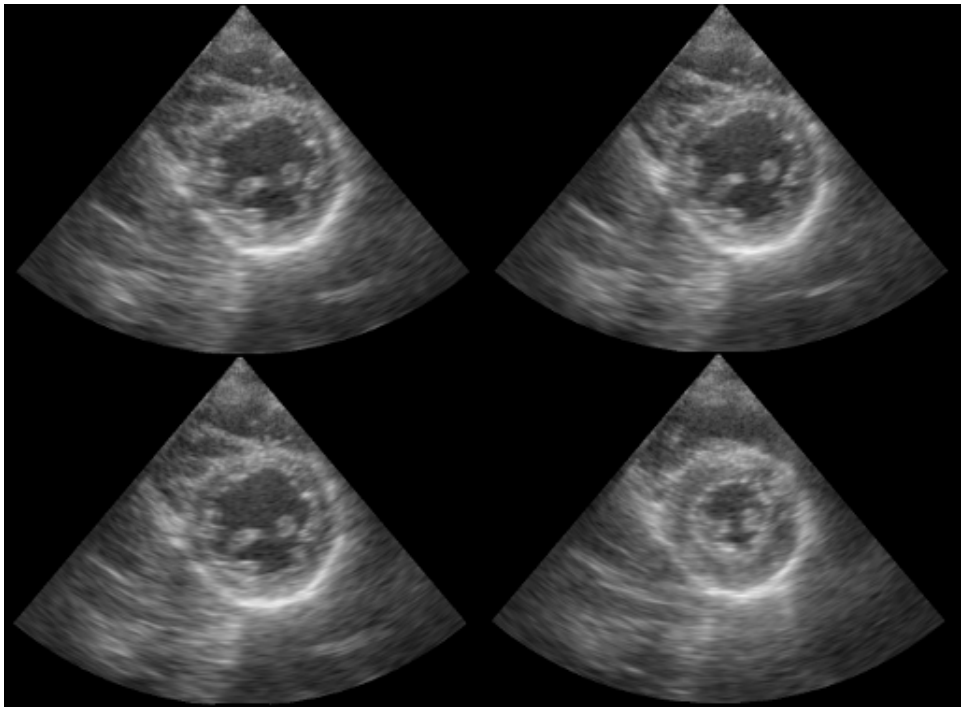


Figure 1.1. *Ultrasound images in a 2x2 grid*

The remainder of this section is devoted to the detailed description of the medical application considered in this thesis and its related problems. At the end, the importance of efficient algorithms is motivated once more.

In the present medical application, 16 streams of ultrasound images have to be displayed on a monitor in a 4x4 grid, as illustrated in Figure 1.2. The input and the output of the application is specified as:

Input: 16 grayscale image streams of size 1024x768 compressed with JPEG
Output: 1 single stream of size 1024x768 with all 16 images at a resolution of 256x192, displayed as a video stream with 30 frames per second.

To scale down a single image of size 1024x768 to 256x192, a down-scaling factor of 4 is used. Other scaling factors can be analogously addressed, as the algorithms described in this thesis are scalable, *i. e.,* independent of a specific scaling factor.

The ultrasound image streams in clinical networks are usually stored on a server. On request, the images are transmitted to a client for further processing. There are multiple options for solving the tasks of downscaling and displaying, and two of them are discussed in the following. One solution transmits 16 image streams to the client simultaneously and the client scales down and displays the image stream. This model is shown in Figure 1.3. It is not optimal due to the highly increased data transmission from the server to the client. In addition, it needs a high performance client system because of the time consuming downscaling task and its violating real-time requirement.

A more efficient solution can be achieved by minimizing the transmission costs. Therefore, the

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure 1.2. *The 4x4 grid displayed at the client.*

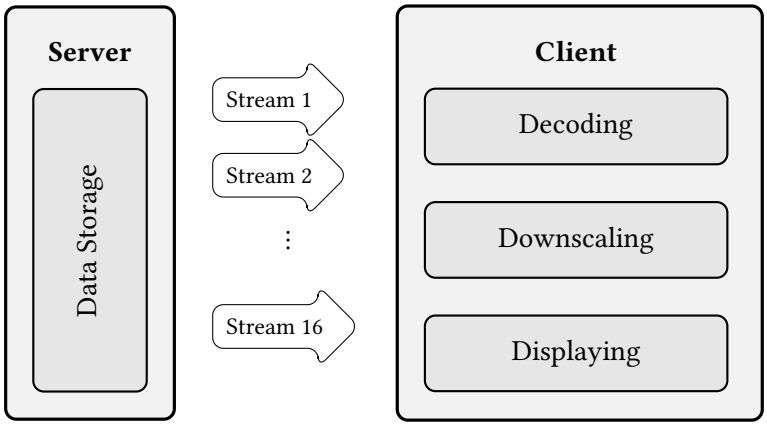


Figure 1.3. *Scheme of the suboptimal scenario model.*

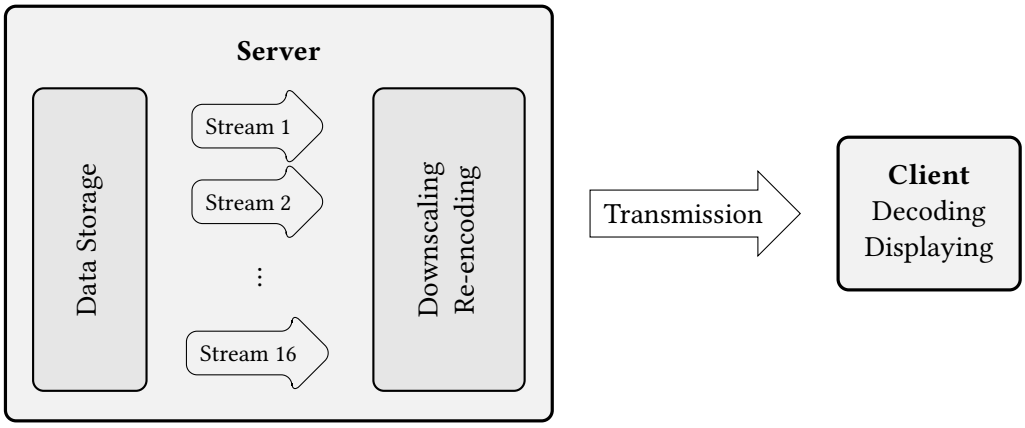


Figure 1.4. *Scheme of the chosen scenario model.*

images should be scaled down on the server side, and only one single compressed image stream is transmitted to the client. Performing the downscaling and re-encoding tasks on the server includes no additional hardware requirements, because servers are usually high end systems. Any recent PC or tablet is able to display a video stream in real-time. According to these considerations, this approach not only minimizes the transmission costs, it also addresses a larger range of clients, in contrary to the previously discussed model. The tasks on the server side are the

1. downscaling of the images, and the
2. re-encoding of the downscaled images,

as illustrated in Figure 1.4. To decode the compressed image stream and to display it, is in the hand of the client.

Even though this scenario model is designed in a proper way, there are still several issues to solve. First of all, there are lots of data to be transmitted from the server to the client. If the stream would contain only 5 seconds, the (downscaled) uncompressed data to transfer is around 1 GB, which would result in a rate greater than 200 MB/s. Today, usual networks have a throughput of 100 MB/s, therefore data compression is important to make this scenario possible. Second the downscaling process is significant for quality, but is also time consuming. There will be a trade off between quality gain and performance. Finding an acceptable compromise between those two factors is crucial. The re-encoding step afterwards should be quality preserving and fast. As the whole scenario is a streaming application, a good choice is to prefer a video codec over an image codec.

Remaining problems are the fast access of the scattered JPEG files on the file system and the handling of the client side. Both of them are not addressed in this thesis. The former problem can be solved technically with better hardware, e. g., RAID-systems or with the use of faster storage mediums like solid state disks. Solutions for the latter exist already that can receive, decode, and display a video stream.

The tasks of the server are discussed now. The server has to scale down 16 images simultaneously and re-encode them into a single video stream. The computational performance as

well as the achieved quality of the downscaling process are the most important aspects on the server side. To re-encode the data, a standardized method like JPEG or H.264/AVC is used. Consequently, the efficient downscaling of images is the focus of this thesis.

To further motivate the need of efficient algorithms, the problem is handled first in a naive approach with the available open source implementation of the Independent JPEG Group. Decoding and downscaling of a single big image of size 4096x3072, which corresponds to the 16 images of size 1024x768, is tested. A time measurement on a high performance PC [1] showed that the decoding and downscaling task requires a computation time around 1 second. But to challenge a frame rate of 30 frames per second, there is only time of 0.03 seconds left to process a single frame.

This example shows that further improvements are necessary to reduce the time of this task. One aspect is the parallelization of the downscaling and re-encoding process. Another is to find a method which has low complexity and good quality. Even though the tasks of the server can be parallelized, the speeding up gain is not sufficient to fulfill the real-time requirement. Once more, this demonstrates the need of efficient downscaling algorithms. Implementation details for the fast final server application using parallelization are described in Appendix A.

In general, downscaling methods can be separated in two classes, namely

1. downscaling in the pixel domain, and
2. downscaling in the DCT domain.

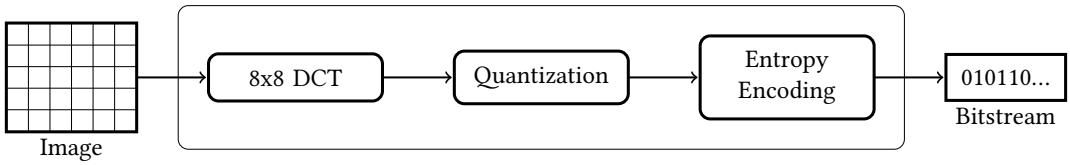
The pixel domain corresponds to the image pixel values, while the DCT domain denotes the frequency representation used by the JPEG compression, which is explained in the next section. Methods of the second approach result in faster algorithms, because these methods operate directly on the frequency representation, while the methods working in the pixel domain require the additional decoding step. Downscaling methods will be discussed exclusively in Chapter 5 and their properties will be outlined. As a compromise between performance and quality a new hybrid downscaling algorithm is introduced in Section 5.3.

1.2. The JPEG compression standard

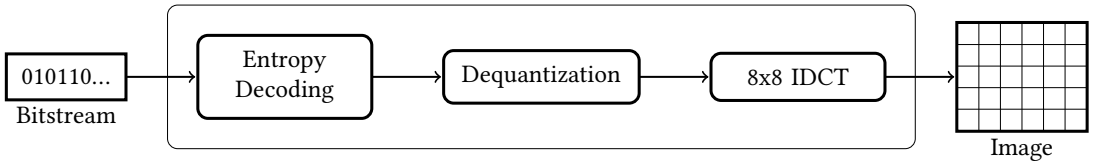
As mentioned in the last section, the input images are compressed with the JPEG standard [15]. This section only gives a short overview of the JPEG encoding and decoding scheme, while a complete overview can be found in Wallace [45].

The JPEG standard was established in 1992 by the Joint Photographic Experts Group (JPEG) as an international standard. It was the first standard for the compression of continuous-tone still images. Today, this standard is still the most used image compression method. JPEG also plays an important role for medical imaging. It was included in the DICOM standard, which specifies the file format for medical images. Thus, most of the medical images are compressed with the JPEG standard. This thesis considers the efficient downscaling of ultrasound images,

[1] CPU Intel Core i5 with 3.1 GHz, 8Gb RAM of memory, a 128 GB Intel SSD



(a) *Encoding*



(b) *Decoding*

Figure 1.5. *JPEG coding scheme.*

which are encoded with JPEG. Medical images are usually compressed at the most by a factor of 1:7 to preserve the image quality.

The most important technical aspects are described in the following. First of all, one has to separate the encoding and the decoding process. Encoding compresses an image and the result is a size minimized bitstream. The decoding process is the inverse of the encoding of an image, a compressed bitstream is processed to get the original image back. Both processes are illustrated in Figure 1.5.

The encoding process can be subdivided in the following core parts:

1. forward transformation with the 8x8 DCT,
2. quantization, and
3. entropy coding.

Before the forward transformation, the whole image is tiled into 8x8 blocks. Then, every block is transformed with the 8x8 discrete cosine transform (DCT), where the pixels are transformed to related frequencies. These frequencies are called coefficients, as they refer to the decomposition into the sinusoidal basis functions. The construction of the DCT basis is given in (1.1) and explained later in detail. The coefficients resulting from the transformation are then quantized, *i. e.,* frequencies are thresholded according to their contribution to the image. Usually this step loses information that cannot be reconstructed in the decoding process. Having the coefficients quantized, the corresponding representation is further compressed by an entropy encoder on the bit level. The JPEG standard uses the so called Huffman encoding, that tries to minimize the entropy of the bitstream. The same parts in reversed order are contained in the decoding process. First the bitstream is processed by the entropy decoder. The quantized coefficients are dequantized and finally transformed to pixels via the inverse 8x8 DCT.

In the scenario described in Section 1.1, the images are encoded with JPEG. To scale these down, the entropy decoding step as well as the dequantization step are necessary. Thus, there are only two possible ways left to perform the downscaling of the image. One can deal with

the DCT coefficients directly or transform them to pixel values. Of course, the additional transformation step is time consuming and methods that can resize an image by its DCT coefficients are worth considering. Such methods are discussed in Section 5.2.

In order to understand these methods in detail, let us recall the definition and some properties of the discrete cosine transform. The DCT is basically the real part of the discrete Fourier transform. Therefore, the DCT basis has almost the same properties. The N basis functions for a DCT of size N are given by the matrix $C_N \in \mathbb{R}^{N\times N}$ with entries

$$c_{kl}^N = \sqrt{\frac{2}{N}}\alpha_k \cos\left(\frac{(2l+1)k\pi}{2N}\right), \qquad \text{for } k,l \in \{0,\dots,N-1\}, \tag{1.1}$$

where

$$\alpha_k = \begin{cases} 1, & k = 0 \\ \frac{1}{\sqrt{2}}, & 1 \le k \le N-1 \end{cases}. \tag{1.2}$$

The basis functions are orthonormal and the matrix C_N is unitary, *i. e.,* $C_N \cdot C_N^{\mathrm{T}} = \mathrm{Id}$.

The two-dimensional DCT can be separated into first transforming the rows of the image followed by transforming the columns of the image. An 8x8 pixel block has the DCT transform

$$\hat{X} = C_8 X C_8^{\mathrm{T}}, \tag{1.3}$$

and can be inverted by

$$X = C_8^{\mathrm{T}} \hat{X} C_8. \tag{1.4}$$

The 64 two-dimensional basis functions are shown in Figure 1.6. Note that from the left to the right the horizontal frequency is increased, while from the top to the bottom the vertical frequency increases. The 8x8 DCT expresses an 8x8 block of pixels as a linear combination of those 64 basis functions. With increasing frequency the represented information gets more and more detailed. Often, the highest frequency (the bottom right corner) contains mostly information about noise. The quantization step will suppress exactly those detailed information over the more important low frequency information by the use of a weighted quantization matrix.

1.3. Mathematical model

This section transforms the technical model developed in Section 1.1 to a mathematical formulation. The goal is to provide a solution for the medical application, where 16 image streams have to be scaled down by a factor of 4. To scale down 16 images simultaneously will be reduced in the mathematical model to the problem of downscaling one single image. The resulting downscaling algorithm will be applied then to every single image, which can be parallelized very efficiently.

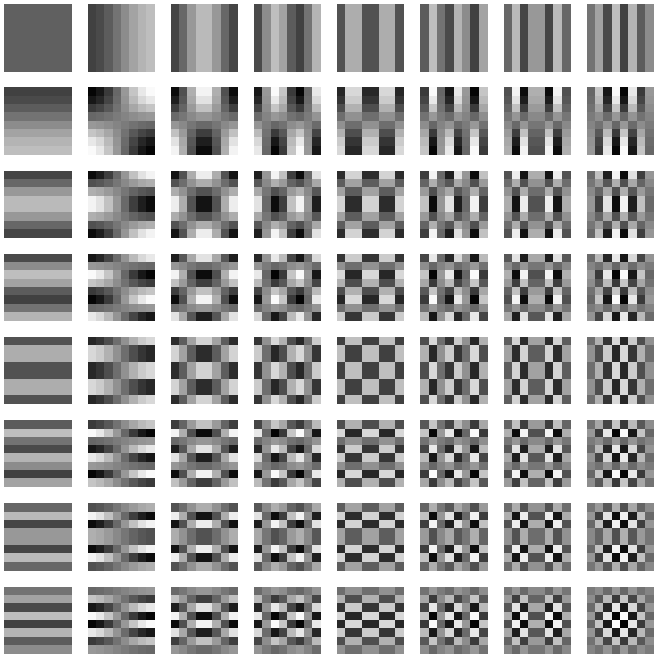


Figure 1.6. *The 64 basis function of the 2D-DCT.*

Digital images are always discrete copies of their corresponding real-world representations. Therefore, it is reasonable to model the downscaling task for the real-world representation of the image. The reconstruction of this continuous representation is crucial to model this image processing problem with a general mathematical framework. If a solution in this continuous framework exists, the downscaled image is provided by sampling of this solution.

According to these considerations, one can split the downscaling problem into two tasks. First, a continuous approximation of the discrete image has to be obtained. Having this continuous representation, the downscaling problem can be solved by dilating this function.

A discrete image of size $n \times m$ can be represented as a matrix $F \in \mathbb{R}^{n \times m}$, where the entries of the matrix $F_{i,j}$ represent the intensity value of the pixel i, j. The underlying continuous representation will be a two-dimensional function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, where the intensity values coincide at the discrete pixels, *i. e.,* $f(i, j) = F_{i,j}$, or they even differ, if the discrete image is e. g., corrupted by noise. Thus, the matrix F is the sampled version of the function f. The function space of the function f is unknown, thus the mathematical model has to provide an appropriate function space that corresponds to the discrete image. The literature often uses the space of square-integrable functions, $L^2(\mathbb{R}^2)$. As sampling is not well-defined for every function $f \in L^2(\mathbb{R})$, this thesis considers subspaces $V \subset L^2(\mathbb{R}^2)$ where the sampling operation is well-defined.

The approximation of the discrete image yields to an optimization problem. The best approximation $g \in V$ of the discrete image F in some space $V \in L^2(\mathbb{R}^2)$ is informally given by

$$g = \operatorname*{argmin}_{g \in V} \|g - F\|, \tag{1.5}$$

for some suited norm $\|\cdot\|$. Note that the the function g is continuous, while F is discrete. Therefore, the distance between g and F in the space V is not uniquely defined.

In the least-square sense, this problem yields to the interpolation problem

$$g = \underset{g \in V}{\operatorname{argmin}} P(g), \quad \text{such that } g(i,j) = F_{i,j}\ \forall i,j \in \mathbb{Z}, \tag{1.6}$$

or to the approximation problem

$$g = \underset{g \in V}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} |g(i,j) - F_{i,j}|^2 + \lambda P(g). \tag{1.7}$$

The former problem interpolates the discrete image exactly, *i. e.,* $g(i,j) = F_{i,j}$, while in the latter this is in general not the case. Both minimization problems use a penalty term $P(g) \geq 0$, that restricts the solution. For example, it penalizes the smoothness of the solution with $P(g) = \|D^m g\|^2$, $m \in \mathbb{N}$. The approximation problem uses a regularization parameter $\lambda > 0$ to adjust the penalty term. Note that the existence and the uniqueness of the solution depends mainly on the chosen function space V.

If the space $V \in L^2(\mathbb{R}^2)$ has a Hilbert basis, the approximated function $g \in V$ has a representation as a unique expansion of the basis functions. Function spaces of interest are spaces which are generated by integer translates of one single function. First, having only one function minimizes the computational complexity and second, the space V is translation invariant which is important for handling convolutions. Another reason is that translation invariant spaces have been studied quite well in the literature about wavelets, e. g., Mallat [22], and a solid mathematical foundation has been established.

Let $\varphi \in L^2(\mathbb{R}^2)$ and the span of its integer translates generate the closed subspace V. Then the discrete image F is approximated by an expansion

$$g(x) = \sum_{k \in \mathbb{Z}^2} c_k \varphi(x - k), \tag{1.8}$$

where the coefficients $(c_k)_{k \in \mathbb{Z}}$ minimize the approximation error of the continuous representation $f \in L^2(\mathbb{R})$ of F in V,

$$(c_k)_{k \in \mathbb{Z}^2} = \underset{(c_k) \in l^2(\mathbb{Z}^2)}{\operatorname{argmin}} \left\| f - \sum_{k \in \mathbb{Z}^2} c_k \varphi(\cdot - k) \right\|,$$

Downscaling the approximation g is done by dilation with a factor $\alpha > 1$:

$$g_\alpha(x,y) = g(\alpha x, \alpha y), \quad \text{for } x, y \in \mathbb{R}^2. \tag{1.9}$$

Sampling of g_α leads finally to the discrete solution. Note that g_α is in general not in V. But if sampling is well-defined for $g \in V$, so it is for g_α due to the unitary dilation operator, see Section 2.1.

Since the model is entirely described by the pixel values it has to be noted, that this can also be done in the DCT domain. Equation (1.9) can be expressed equivalently in the DCT domain as expansion of the transformed basis functions

$$\mathcal{C}g = \sum_{k \in \mathbb{Z}^2} c_k \mathcal{C}\varphi(x - k), \tag{1.10}$$

where $\mathcal{C}$ denotes the two dimensional DCT transform operator. The explicit discussion of downscaling methods are treated in Chapter 5.

Chapter 2 deals with the mathematical foundations required for the downscaling problem, but first an overview of the research of downscaling is given in the next section.

1.4. Related work

Section 1.1 mentioned that downscaling methods can be separated into downscaling in the pixel domain and downscaling in the DCT domain. The difference is that downscaling methods in the frequency domain operate on the specific 8x8 block tiling of JPEG, while the pixel based methods operate on the whole image. The former methods have less computational complexity, as they avoid the complete process of inversion of the DCT coefficients and the subsequent DCT of the down-scaled image. They scale down the image directly in the DCT domain. In contrast, pixel domain methods require the complete inverse/forward DCT, but can achieve high image qualities with the disadvantage of high computational costs.

1.4.1. DCT domain downscaling methods

In the literature of downscaling methods in the DCT domain there are two different approaches. One approach uses ideal filtering, *i. e.,* truncation and zero-padding of the DCT coefficients for downscaling [11, 12, 23, 26, 31], while the other approach adapts pixel domain downscaling methods to the DCT domain [21, 24]. Methods of the first type are using a rescaling of the DCT basis, as discussed by Martucci [23] for general block DCT's and for JPEG images explicitly by de Queiroz [11]. Based on this idea, Dugad and Ahuja [12] developed a fast algorithm for down-sampling and up-sampling by a factor of two and analyzed this algorithm in theory and in numerical experiments. A so called L/M-fold image resizing method with higher computational complexity was introduced by Park et al. [26], based on the correspondence of multiplication in the spatial domain and symmetric convolution in the DCT domain. The downscaling factor was also generalized to L/M, where L and M are integers. A fast version of this method was given later using fast DCT algorithms [27]. Salazar and Tran [31] generalized the method of Dugad and Ahuja [12] to any rational scale, as this method only considered downscaling by a factor of two. The drawbacks of all of these methods are due to the use of the ideal filter. On the one hand, they introduce ringing and aliasing effects, but on the other hand these methods have low computational complexity [12].

Other methods are using the fact that downscaling in the DCT domain can be separated into inverse DCT, downscaling, and finally the forward DCT. They combine this process into a

double sided matrix multiplication. In the work of Merhav and Bhaskaran [24], the pixel domain downscaling process was chosen as a linear interpolation scheme. The related double sided matrix multiplication was then factorized into sparse matrices to reduce computational complexity. A generalization of this idea, namely choosing any kind of sub-sampling scheme is discussed by Lee and Eleftheriadis [21]. The quality and the computational complexity of these methods depend only on the chosen pixel-domain downscaling method. Thus the most significant drawback is the increased complexity introduced by the usual non-sparse matrix multiplication.

However, all these DCT downscaling methods suffer from the fixed 8x8 block tiling of the image and thus may introduce blocking artifacts.

1.4.2. Pixel domain downscaling methods

Downscaling methods were first considered as re-sampling with a larger sampling step, according to Shannon's sampling theorem [34] from 1949. This directly corresponds to the so called sinc-interpolation method. To determine the best interpolation method, Parker et al. [28] considered in 1983 different interpolation functions which correspond to the nearest neighbor, the linear, and the cubic B-spline interpolation. The result showed the advantage of the cubic B-spline over the others, which approves the theoretical results of the approximation quality of B-splines examined by De Boor [8, 9]. Keys [16] introduced a fast cubic spline convolution method. In 1991, Unser et al. [42] considered fast B-spline transforms for image interpolation and further enhanced the theory and applications of discrete B-splines [43, 44]. Using B-splines to downscale images with arbitrary scaling factors is discussed in the sense of least square approximation, which led to orthogonal projection methods. The concept of downscaling is extended to pre-filtering, sub-sampling and post-filtering. Based on these considerations, Thevenaz et al. [37] introduced a generalized interpolation concept and compared several basis functions according to their approximation properties. This paper concludes that the approximation order is the most important factor for image quality, while the computational complexity is mostly affected by the support of the basis functions. The B-splines outperform almost every other method regarding the image quality and the computational complexity. Better in this sense are the so called MOMS basis functions discussed in Blu et al. [3], because they have maximal approximation order regarding their minimal support due to the lack of smoothness. Lee et al. [19] introduced a fast image resizing approach using oblique projections instead of orthogonal projections. Downscaling factors that are a power of two are extensively discussed in the wavelet literature [22]. A comprehensive overview of this history is given in [39] and a widespread discussion about the B-splines is shown in Unser [38].

In general, downscaling methods in the pixel domain have been well studied, where the resulting image quality corresponds to the approximation properties of the used basis functions. Therefore, high image qualities can be achieved due to higher computational costs compared to the DCT-based methods.

2. Mathematical Preliminaries

This chapter gives the foundation for the modern sampling theory, that will be discussed in the next chapter. First, the most important facts of the Fourier Analysis are reviewed. These facts are used frequently in this thesis, since most of the results rely on the Fourier transform. Then the theory of sampling of band-limited functions according to Shannon is introduced. Section 2.2 concludes with the result that band-limited functions can be perfectly reconstructed by their sampled values at a proper sampling rate. The next chapter introduces a generalized sampling approach by the use of a sequence of integer translates of a single basis function. This sequence is usually not orthonormal, which leads naturally to the concept of frames and Riesz bases. These concepts can be seen as an extension to orthonormal bases and will be considered in Section 2.3. Next, basis properties about B-splines are discussed. B-splines have excellent approximation properties and will be used in this thesis as the chosen basis function for the generalized sampling theory. It will be shown, that the integer translates form a Riesz basis. Finally, a computational problem related to the calculation of the expansion of a function in frames will be outlined, which is important for the numerical experiments.

2.1. Short review of Fourier analysis

In this section a short review of the most important statements of the Fourier analysis is given. To hold this section compact, the proofs are omitted, but can be looked up in Lasser [18], Mallat [22], and Ramanathan [29].

The *Fourier transform* for a function $f \in L^1(\mathbb{R})$ is defined as

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(x) \mathrm{e}^{-2\pi i x \cdot \omega} \mathrm{d}\omega, \qquad \text{for } \omega \in \mathbb{R}. \tag{2.1}$$

The function $\hat{f}$ is bounded and uniformly continuous on $\mathbb{R}$.

If further $\hat{f} \in L^1(\mathbb{R})$, the *inversion formula* holds:

$$f(x) = \int_{\mathbb{R}} \hat{f}(\omega) \mathrm{e}^{2\pi i \omega \cdot x} \mathrm{d}x, \qquad \text{for } x \in \mathbb{R}. \tag{2.2}$$

Because $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is dense in $L^2(\mathbb{R})$, the Fourier transform can be extended to the space of square-integrable function $L^2(\mathbb{R})$. The space $L^2(\mathbb{R})$ as Hilbert space with its inner product is usually preferred over the $L^1(\mathbb{R})$. This extension yields to the Fourier operator $\mathcal{F} : L^2(\mathbb{R}) \to L^2(\mathbb{R})$. The inverse operator is denoted by $\mathcal{F}^{-1}$. Note that in contrast to the

pointwise defined L^1 Fourier transform, the operator $\mathcal{F}$ for functions in $L^2(\mathbb{R})$ is only defined by the limit of a uniformly converging sequence in $L^1(\mathbb{R})$.

Some properties for $f, g \in L^2(\mathbb{R})$ are

(i) $\|\mathcal{F}f\|_2 = \|f\|_2$,

(ii) $\langle \mathcal{F}f, \mathcal{F}g \rangle = \langle f, g \rangle$, and

(iii) for $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ the equality $\mathcal{F}f(\omega) = \hat{f}(\omega)$ holds.

Now the *translation*, *dilation* and the *modulation* operator $T_a, D_\alpha, M_b : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ are introduced for $a, b \in \mathbb{R}$ and $\alpha > 0$ for easier notation:

$$\begin{aligned} T_a f(x) &= f(x-a), \\ D_\alpha f(x) &= f\left(\frac{x}{\alpha}\right), \\ M_b f(x) &= \mathrm{e}^{2\pi i b \cdot x} f(x). \end{aligned}$$

All these operators are unitary and their inverse is given by

$$T_a^{-1} = T_a^* = T_{-a}, \quad D_\alpha^{-1} = D_\alpha^* = D_{\frac{1}{\alpha}}, \quad M_b^{-1} = M_b^* = M_{-b}.$$

In combination with the Fourier transform the commutator relations of the operators

$$\mathcal{F}T_a = M_{-a}\mathcal{F}, \quad \mathcal{F}D_\alpha = D_{\frac{1}{\alpha}}\mathcal{F}, \quad \mathcal{F}M_b = T_b\mathcal{F}$$

hold.

The *convolution* of two functions $f, g \in L^1(\mathbb{R})$ is defined as

$$(f * g)(x) = \int_{\mathbb{R}} f(y) g(x-y) \mathrm{d}y = \int_{\mathbb{R}} f(y) T_y g(x) \mathrm{d}y. \tag{2.3}$$

The convolution is well defined and further $f * g \in L^1(\mathbb{R})$.

The *convolution theorem* states an interesting relation between the convolution and their Fourier transform:

$$\widehat{(f * g)}(\omega) = \hat{f}(\omega) \cdot \hat{g}(\omega). \tag{2.4}$$

Note that this equation also holds for function $f, g \in L^2(\mathbb{R})$ with the L^2-Fourier transform $\mathcal{F}$. With this observation the convolution can be seen as filtering the function f with a filter g where certain frequencies are omitted. A typical filter is the so called ideal filter defined by $\chi_{\left[-\frac{1}{2}, \frac{1}{2}\right]}(\omega)$, which removes all frequencies larger than $1/2$. The ideal filter is shown in Figure 2.1 with its Fourier transform, the $sinc$-function that is not in $L^1(\mathbb{R})$ but in ${}^2(\mathbb{R})$.

A function $f \in L^1(\mathbb{R})$ can be transformed into a corresponding 1-periodic function $f_{per} \in L^1(\mathbb{T})$ by

$$f_{per}(x) = \sum_{k\in\mathbb{Z}} f(x+k), \qquad \text{for almost all } x \in \mathbb{T}. \tag{2.5}$$

The function f_{per} is 1-periodic and thus $f_{per} \in L^1(\mathbb{T})$. For 1-periodic functions $f \in L^1(\mathbb{T})$, the spectrum of the Fourier transform is discrete. The k'th Fourier coefficient of f is defined by

$$\hat{f}(k) = \int_{\mathbb{T}} f(x)\mathrm{e}^{-2\pi i k\cdot x}\,\mathrm{d}x, \tag{2.6}$$

with inversion

$$f(x) = \sum_{k\in\mathbb{Z}} \hat{f}(k)\mathrm{e}^{2\pi i k\cdot x}. \tag{2.7}$$

As above, the following properties hold for $f, g \in L^1(\mathbb{T})$:

(i) $(f * g)(x)$ is well defined and $f * g \in L^1(\mathbb{T})$,

(ii) $\widehat{(f * g)}(k) = \hat{f}(k) \cdot \hat{g}(k)$.

The *Poisson summation formula* states that the inversion of the Fourier series, where the coefficients are the integer sampled values of $\hat{f}$, leads to the 1-periodic version of f according to (2.5). To be more specific, for $f \in L^1(\mathbb{R})$ sufficient smooth and such that $\hat{f} \in L^1(\mathbb{R})$ has sufficient fast decay

$$\sum_{k\in\mathbb{Z}} \hat{f}(k)\mathrm{e}^{2\pi i k\cdot x} = \sum_{k\in\mathbb{Z}} f(x+k) \tag{2.8}$$

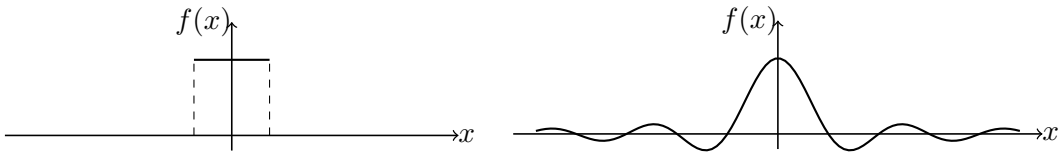
holds. Interpreting this result yields, that sampling in the frequency domain corresponds to periodization in the time domain and vice versa.

2.2. Shannon sampling theorem

In communication theory, the reduction of the amount of data to be sent over a network is crucial. In order to achieve data reduction, the continuous real-world signals are sampled and only the discretized signal values are sent. Therefore, one can ask for the minimal sampling rate at which continuous signals can be reconstructed without loss of information. This minimal rate was firstly considered by Nyquist in 1928. Later, in 1949 Shannon proofed this lower bound in his well known sampling theorem. This section will state this theorem and discuss related problems in the sense of sampling.

For some $\alpha > 0$ we consider the *Paley-Wiener* space

$$PW_\alpha = \left\{ f \in L^2(\mathbb{R}) \,:\, \operatorname{supp} \hat{f} \subset \left[-\frac{\alpha}{2}, \frac{\alpha}{2}\right] \right\}, \tag{2.9}$$



(a) $\chi_{\left[-\frac{1}{2},\frac{1}{2}\right]}$ **(b)** $\operatorname{sinc}(x)$

Figure 2.1. *The characteristic function and the* sinc*-function.*

which is the space of functions that are band-limited to $[-\alpha/2, \alpha/2]$. Note that sampling in this space is well-defind, as all functions $f \in PW_\alpha$ are continuous.

Sampling, *i. e.,* discretizing, a function $f \in PW_\alpha$, leads to periodization of its Fourier transform. This is a direct consequence of the inversion of the Fourier series and the Poisson summation formula (2.8):

$$\sum_{k\in\mathbb{Z}} f(k)\mathrm{e}^{2\pi i k\cdot\omega} = \sum_{k\in\mathbb{Z}} \hat{f}(\omega + k). \tag{2.10}$$

Now the main theorem of this section can be formulated according to Shannon [34]. Note that Kotelnikov (1933) and Whittaker (1935) proved similar results earlier in the context of interpolation theory.

Theorem 1 (Shannon sampling theorem) *Let $f \in PW_\alpha$ and $T > 0$ where $\rho := \frac{1}{T} \geq \alpha$. Then*

$$f(x) = \sum_{k\in\mathbb{Z}} f(kT)\operatorname{sinc}\left(\frac{x}{T} - k\right) = \sum_{k\in\mathbb{Z}} f\left(\frac{k}{\rho}\right)\operatorname{sinc}\left(\rho x - k\right). \tag{2.11}$$

The function $\operatorname{sinc}(x) \in L^2(\mathbb{R})$ *is defined as (see Figure 2.1)*

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0 \end{cases}. \tag{2.12}$$

The term ρ corresponds to the *sampling rate* of the function f and is often called the Nyquist rate according to its originator. The values $f(kT), k \in \mathbb{Z}$ are called *samples* or *sample values.*

This theorem states that for perfect reconstruction of f we only need the sample values $f(kT),\ k \in \mathbb{Z}$. If the function f is not bandlimited or the sampling rate ρ is too low, then the so called aliasing artifacts occur. This effect is illustrated in Figure 2.3 and an example in 2D is given in Figure 2.4.

To get a fundamental understanding of the aliasing effect, a sampling rate lower than the theorem states is assumed now. Therefore let $f \in PW_\alpha$ and $\rho < \alpha$. As in the Sampling theorem, a function $\tilde{f}$ is reconstructed by the interpolation formula

$$\tilde{f}(x) = \sum_{k\in\mathbb{Z}} f\left(\frac{k}{\rho}\right)\operatorname{sinc}\left(\rho x - k\right). \tag{2.13}$$

Clearly, the values of f and $\tilde{f}$ conincide at the sample values, But if the Fourier transforms of $\tilde{f}$ and f are compared, one can see that $\tilde{f}$ is not equal to f. First, note that the relation

$$\begin{aligned}\mathcal{F}\chi_{\left[-\frac{1}{2},\frac{1}{2}\right]}(\omega) &= \int_{\mathbb{R}} \chi_{\left[-\frac{1}{2},\frac{1}{2}\right]}(x)\mathrm{e}^{-2\pi i x\cdot\omega}\mathrm{d}x = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{e}^{-2\pi i x\cdot\omega}\mathrm{d}x = \frac{\mathrm{e}^{-\pi i\omega} - \mathrm{e}^{\pi i\omega}}{-2\pi i\omega} \\ &= \operatorname{sinc}(\omega) \end{aligned} \tag{2.14}$$

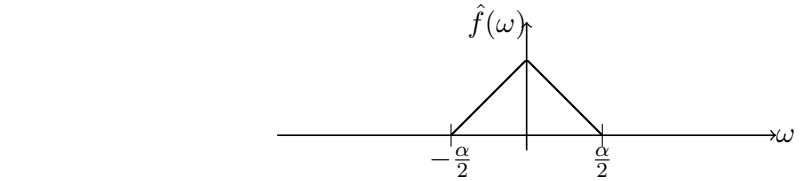
holds. Then the Fourier transform of $\tilde{f}$ yields to

$$\begin{aligned}\hat{\tilde{f}}(\omega) &= \int_{\mathbb{R}} \sum_{k\in\mathbb{Z}} f\left(\frac{k}{\rho}\right) \operatorname{sinc}(\rho x - k)\mathrm{e}^{-2\pi i\omega\cdot x}\mathrm{d}x \\ &= \sum_{k\in\mathbb{Z}} f\left(\frac{k}{\rho}\right) \int_{\mathbb{R}} \operatorname{sinc}(\rho x - k)\mathrm{e}^{-2\pi i\omega\cdot x}\mathrm{d}x \\ &= \sum_{k\in\mathbb{Z}} f\left(\frac{k}{\rho}\right) \mathrm{e}^{2\pi i\omega\cdot x}\chi_{\left[-\frac{\rho}{2},\frac{\rho}{2}\right]}(\omega) \\ &= \chi_{\left[-\frac{\rho}{2},\frac{\rho}{2}\right]}(\omega)\cdot\sum_{k\in\mathbb{Z}} \hat{f}\left(\omega+\rho k\right). \end{aligned}$$

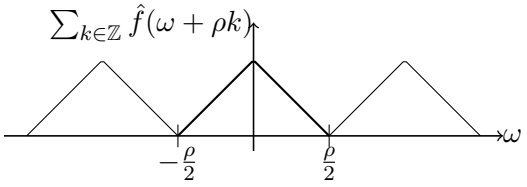
In the middle of the derivation the fact that $\mathcal{F}\operatorname{sinc}(\omega) = \chi_{\left[-\frac{1}{2},\frac{1}{2}\right]}(\omega)$ is used while in the last step the Poisson summation formula (2.8) is applied.

The last line shows that the ideal filter $\chi_{\left[-\frac{\rho}{2},\frac{\rho}{2}\right]}(\omega)$ cuts off the frequencies of the periodized version of $\hat{f}$ with period ρ. Hence the function $\mathcal{F}\tilde{f}$ and $\mathcal{F}f$ would be only equal if $f \in PW_\rho$. The last considerations are illustrated in Figure 2.2 and Figure 2.3. The former illustrates sampling with an appropriate sampling rate, while the latter demonstrates the aliasing effect.

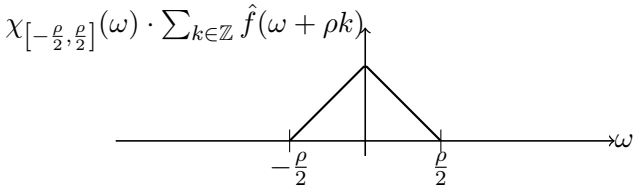
Note that the ideal filter represents the ideal anti-aliasing filter, because everything beyond the interval $\left[-\frac{\rho}{2}, \frac{\rho}{2}\right]$ is removed. A drawback of using this ideal filter is its discontinuity.



(a) *Fourier transform*

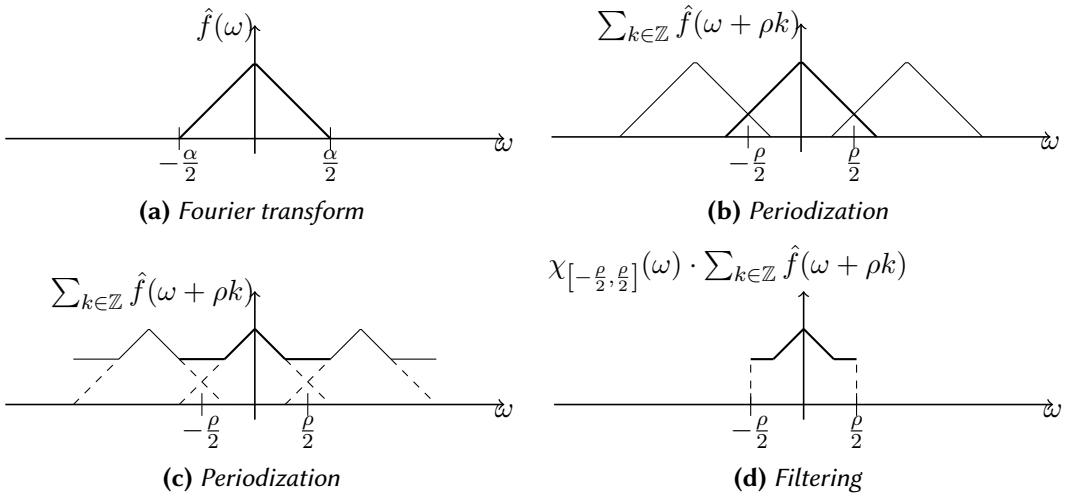


(b) *Periodization*



(c) *Filtering*

Figure 2.2. *Sampling theorem with appropriate sampling rate $\rho = \alpha$.*



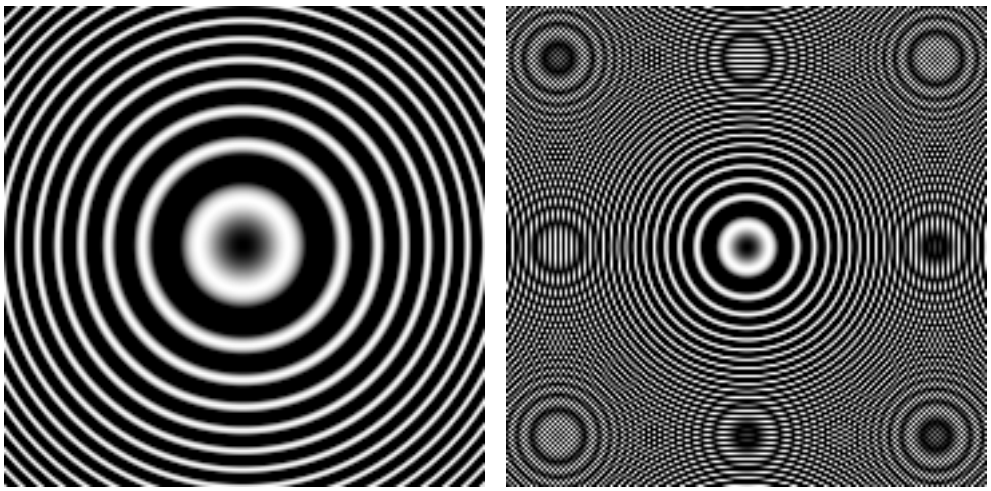
(a) *Fourier transform* **(b)** *Periodization*

(c) *Periodization* **(d)** *Filtering*

Figure 2.3. *Sampling theorem with a sampling rate $\rho < \alpha$.*

Therefore, the sinc-function as the Fourier transform of the ideal filter is not compactly supported. Also the sinc-function has a low decay asymptotically like $1/x$. At least it has to be mentioned that a real world device like the ideal filter does not exist, which underlines the meaning of ideal.

If a function is not band-limited it has to be pre-filtered with an anti-aliasing filter to apply the Shannon theorem. Shannon proposes the ideal filter for anti-aliasing. This pre-filtering step is an important step to consider, because functions occurring in the real world are usually not band-limited.



(a) *Correctly sampled image* **(b)** *Undersampled image*

Figure 2.4. *Aliasing effects in 2D.*

2.3. Frames and Riesz bases in $L^2(\mathbb{R})$

Orthonormal bases provide a well known framework to deal with in the Hilbert space $L^2(\mathbb{R})$, because every function can be represented as expansion of the elements of the orthonormal basis. The integer translates of the sinc-function, used in Shannon's theorem, form an orthonormal basis of the space of band-limited function. Usually, translates of other functions do not lead to an orthonormal basis. Frames can be seen as extension of this concept. For each function an expansion with frame elements exists. But in general, this expansion is not unique. The introduction of Riesz bases as special frames will fix this issue in Section 2.3.2. Finally, motivated by the preceding discussion, some properties for Riesz bases that are generated by integer translates of a single function will be developed.

This section gives only a compact overview about frame theory and more detailed information can be found in Christensen [6].

2.3.1. Frames

Definition 1 A sequence $\{f_k : k \in \mathbb{Z}\} \subset L^2(\mathbb{R})$ is called *complete* in $L^2(\mathbb{R})$, if

$$\overline{\text{span}}\,\{f_k : k \in \mathbb{Z}\} = L^2(\mathbb{R}). \tag{2.15}$$

Trivially, every sequence is complete for its closed span, $\overline{\text{span}}\,\{f_k : k \in \mathbb{Z}\}$, as direct consequence of its definition.

Definition 2 A *frame* for $L^2(\mathbb{R})$ is a sequence $\{f_k : k \in \mathbb{Z}\} \subset L^2(\mathbb{R})$, if there exist constants $0 < A \leq B$, such that

$$A\,\|f\|^2 \leq \sum_{k\in\mathbb{Z}} |\langle f, f_k\rangle|^2 \leq B\,\|f\|^2, \qquad \text{for all } f \in L^2(\mathbb{R}). \tag{2.16}$$

The constants A and B are called the lower and upper *frame bounds* respectively. The frame is an orthonormal basis iff $A = B = 1$.

In the following, the sequence $\{f_k : k \in \mathbb{Z}\}$ is assumed to be a frame. As stated, the interest lies on expressing a function as an expansion with frame elements. For that purpose one can examine the expansion

$$\sum_{k\in\mathbb{Z}} \langle f, f_k\rangle f_k.$$

Because of the existing upper frame bound this expansion is also well defined. To formalize this expansion, two important operators are introduced next.

Definition 3 The *synthesis operator* $U : l^2(\mathbb{Z}) \to L^2(\mathbb{R})$ is defined as

$$Uc = \sum_{k\in\mathbb{Z}} c_k f_k. \tag{2.17}$$

Its adjoint $U^* : L^2(\mathbb{R}) \to l^2(\mathbb{Z})$,

$$U^* f = (\langle f, f_k \rangle)_{k \in \mathbb{Z}}\,, \tag{2.18}$$

is called the *analysis operator.* Together they form the *frame operator* $S : L^2(\mathbb{R}) \to L^2(\mathbb{R})$,

$$Sf = UU^* f = \sum_{k=-\infty}^{\infty} \langle f, f_k \rangle f_k. \tag{2.19}$$

Theorem 2 *The frame operator S is bounded, invertible, self-adjoint, and positive. Furthermore the inverse frame operator generates the so called dual frame $\left\{S^{-1} f_k \,:\, k \in \mathbb{Z}\right\}$ with frame bounds B^{-1} and A^{-1}. The inverse frame operator has the same properties as the frame operator.*

As the inverse frame operator is a frame operator by itself, it has the same properties. It is bounded, invertible, self-adjoint, and positive.

Analog to the decomposition of a function to an expansion with uniquely defined coefficients and elements of a basis, one can express every function $f \in L^2(\mathbb{R})$ as a linear combination of the frame elements:

$$f = SS^{-1} f = \sum_{k \in \mathbb{Z}} \langle S^{-1} f, f_k \rangle f_k = \sum_{k \in \mathbb{Z}} \langle f, S^{-1} f_k \rangle f_k. \tag{2.20}$$

Not that in difference to a orthonormal basis, this representation is not unique.

Compared to bases, a frame is not a minimal system, *i. e.,* the elements are linearly dependent, and as such, contains in general redundant information. In the next section the concept of Riesz bases is introduced which can be seen as minimal frames.

2.3.2. Riesz bases

This section specializes the concept of the frames to those allowing for a unique decomposition into the frame elements.

Definition 4 A complete sequence $\{f_k \,:\, k \in \mathbb{Z}\}$ in $L^2(\mathbb{R})$ is a *Riesz basis* for $L^2(\mathbb{R})$, if there exist constants $0 < A \leq B$ such that for every sequence $c = (c_k)_{k \in \mathbb{Z}} \in l^2(\mathbb{Z})$ the following constraint holds:

$$A \left\|c\right\|^2_{l^2(\mathbb{Z})} \leq \left\| \sum_{k \in \mathbb{Z}} c_k f_k \right\|^2_{L^2(\mathbb{R})} \leq B \left\|c\right\|^2_{l^2(\mathbb{Z})}. \tag{2.21}$$

Every sequence that only fulfills (2.21) is a Riesz basis for its closed span and is called a *Riesz sequence.*

Clearly this is a stronger condition than required in (2.16) to be a frame. Indeed the next theorem states that every Riesz basis is a frame.

Theorem 3 *A Riesz basis* $\{f_k : k \in \mathbb{Z}\}$ *is a frame, i. e., there exist constants* $0 < A \leq B$*, such that*

$$A \|f\|^2 \leq \sum_{k\in\mathbb{Z}} |\langle f, f_k\rangle|^2 \leq B \|f\|^2, \qquad \text{for all } f \in L^2(\mathbb{R}).$$

Since every Riesz basis is a frame, the frame operator S allows a decomposition of a function $f \in L^2(\mathbb{R})$. Further it induced the dual frame basis $\left\{S^{-1} f_k : k \in \mathbb{Z}\right\}$.

The difference between a frame and a Riesz basis is characterized by its unique decomposition of the null function.

Theorem 4 *A frame* $\{f_k : k \in \mathbb{Z}\}$ *for* $L^2(\mathbb{R})$ *is a Riesz basis for* $L^2(\mathbb{R})$*, iff from the statement*

$$\sum_{k\in\mathbb{Z}} c_k f_k = 0, \qquad \text{for some } c = (c_k)_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$$

follows that $c_k = 0$ *for all* $k \in \mathbb{Z}$*.*

That means reversely that in a frame that is not a Riesz basis the null element has no unique representation, thus the frame elements are not linear independent.

Finally the decomposition of a function $f \in L^2(\mathbb{R})$ results into the unique representation for the Riesz basis elements f_k, $k \in \mathbb{Z}$:

$$f = SS^{-1} f = \sum_{k\in\mathbb{Z}} \langle S^{-1} f, f_k\rangle f_k = \sum_{k\in\mathbb{Z}} \langle f, S^{-1} f_k\rangle f_k. \tag{2.22}$$

2.3.3. Riesz bases of translates

The Shannon sampling theorem introduced in Section 2.2 motivated the use of translates of a function, where in this particular case the sinc-function is used. This section investigates the properties of Riesz bases that are generated by the integer translates of a certain function. In contrast to the last sections, some proofs are provided that demonstrate some typical calculations.

Let $\varphi \in L^2(\mathbb{R})$. The closed space $V_\varphi \subset L^2(\mathbb{R})$ is defined as the span of the integer translates of φ,

$$V_\varphi = \overline{\operatorname{span}}\left\{T_k\varphi : k \in \mathbb{Z}\right\} = \overline{\left\{\sum_{k\in\mathbb{Z}} c_k T_k\varphi : (c_k)_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})\right\}}. \tag{2.23}$$

This φ is called the *generator* of its generating space V_φ.

The Riesz basis property depends only on the function φ and can be furthermore characterized by its periodization in the Fourier domain.

Theorem 5 *The sequence $\{T_k\varphi : k \in \mathbb{Z}\}$ is a Riesz basis for V_φ, iff there exist constants $0 < A \leq B$ with*

$$A \leq \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega + k)|^2 \leq B. \tag{2.24}$$

The generated space V_φ is a proper subspace of the $L^2(\mathbb{R})$.

Proof. Let $c = (c_k)_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$.

$$\begin{aligned}
\left\| \sum_{k\in\mathbb{Z}} c_k T_k \varphi \right\| &= \left\| \left(\sum_{k=-\infty}^{\infty} c_k e_{-k} \right) \mathcal{F}\varphi \right\| = \int_{\mathbb{R}} \left| \sum_{k=-\infty}^{\infty} c_k e_{-k}(\omega) \right|^2 |\hat{\varphi}(\omega)|^2 \, \mathrm{d}\omega \\
&= \sum_{l=-\infty}^{\infty} \int_l^{l+1} \left| \sum_{k=-\infty}^{\infty} c_k e_{-k}(\omega) \right|^2 |\hat{\varphi}(\omega)|^2 \, \mathrm{d}\omega \\
&= \int_0^1 \sum_{l=-\infty}^{\infty} \left| \sum_{k=-\infty}^{\infty} c_k e_{-k}(\omega + l) \right|^2 |\hat{\varphi}(\omega + l)|^2 \, \mathrm{d}\omega \\
&= \int_0^1 \left| \sum_{k=-\infty}^{\infty} c_k e_{-k}(\omega) \right|^2 \cdot \sum_{l=-\infty}^{\infty} |\hat{\varphi}(\omega + l)|^2 \, \mathrm{d}\omega \\
&\leq B \int_0^1 \left| \sum_{k=-\infty}^{\infty} c_k e_{-k}(\omega) \right|^2 \mathrm{d}\omega \\
&= B \left\| c \right\|_{l^2(\mathbb{Z})}^2 .
\end{aligned}$$

The result about the lower bound follows in an analogous manner. □

Clearly the function $\sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega + k)|^2$ is 1-periodic. To simplify the notation, this periodic function is denoted by

$$\Phi(\omega) = \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega + k)|^2. \tag{2.25}$$

Lemma 1 *Let $\varphi \in L^2(\mathbb{R})$. Then the previously defined function Φ is 1-periodic and $\Phi \in L^1(\mathbb{T})$. It has the Fourier coefficients*

$$c_k = \langle \varphi, T - k\varphi \rangle. \tag{2.26}$$

Lemma 2 *If $\{T_k\varphi : k \in \mathbb{Z}\}$ is a Riesz basis for $V_\varphi = \overline{\operatorname{span}}\{T_k\varphi : k \in \mathbb{Z}\}$, then the following holds:*

(i) $f \in V_\varphi$, iff $\hat{f}(\omega) = \hat{h}(\omega) \cdot \hat{\varphi}(\omega)$ where $h \in L^2(\mathbb{T})$,

(ii) If $f = \sum_{k\in\mathbb{Z}} c_k T_k \varphi$, then $f(\omega) = \hat{h}(\omega) \cdot \hat{\varphi}(\omega)$ where $\hat{h}(\omega) = \sum_{k=-\infty}^{\infty} c_k e_{-k}(\omega)$.

Item (i) characterizes all functions in V_φ, while in (ii) a helpful version of the convolution theorem of a discrete convolution of a discrete and a continuous function is provided (compare with (2.4)). Both properties will be used in the following.

Now some important properties of the dual frame are developed. In the next theorem it will be shown, that the dual frame is also generated by the translates of a function depending on φ.

Theorem 6 *Let $\varphi \in L^2(\mathbb{R})$, $\{T_k\varphi : k \in \mathbb{Z}\}$ a Riesz basis for V_φ with frame operator S. Then the commutator relation*

$$ST_k = T_k \text{ and } S^{-1}T_k = T_kS^{-1}$$

holds. As a consequence the sequence of integer translates $\left\{T_kS^{-1}\varphi : k \in \mathbb{Z}\right\}$ constitutes the dual basis of the frame.

Proof. For $f \in L^2(\mathbb{R})$, it holds:

$$\begin{aligned} ST_kf &= \sum_{l=-\infty}^{\infty} \langle T_kf, T_l\varphi\rangle T_l\varphi = \sum_{l=-\infty}^{\infty} \langle f, T_{l-k}\varphi\rangle T_l\varphi \\ &= \sum_{l=-\infty}^{\infty} \langle f, T_l\varphi\rangle T_{l+k}\varphi \\ &= T_kSf. \end{aligned}$$

□

Because the dual basis is given by $\{T_kS^{-1}\varphi : k \in \mathbb{Z}\}$ a function $f \in V_\varphi$ can be uniquely decomposed as

$$f = SS^{-1}f = \sum_{k=-\infty}^{\infty} \langle S^{-1}f, T_k\varphi\rangle T_k\varphi = \sum_{k=-\infty}^{\infty} \langle f, T_kS^{-1}\varphi\rangle T_k\varphi.$$

Theorem 7 *Let $D = \{\omega \in \mathbb{R} : \Phi(\omega) = 0\}$. Then*

$$\hat{\phi}(\omega) = \begin{cases} \frac{\hat{\varphi}(\omega)}{\Phi(\omega)} & , \quad \omega \in D \\ 0 & , \quad \omega \notin D \end{cases}$$

defines the dual generator $S^{-1}\varphi$ to φ.

Proof. Let $\hat{h}(\omega) = \Phi(\omega)^{-1}$. Then for $0 < a \leq b$: $a < |\hat{h}(\omega)| \leq b$ and $h \in L^2(\mathbb{T})$. Hence, $\hat{\phi}(\omega) = \hat{h}(\omega) \cdot \hat{\varphi}(\omega)$ where $h \in L^2(\mathbb{T})$. According to Lemma 2, $\phi \in V_\varphi$.

Further we get

$$\begin{aligned}\mathcal{F}S\phi &= \sum_{k=-\infty}^{\infty} \langle \phi, T_k\varphi\rangle \mathcal{F}T_k\varphi \\ &= \sum_{k=-\infty}^{\infty} \langle \hat{\phi}, \mathcal{F}T_k\varphi\rangle \mathcal{F}T_k\varphi \\ &= \left(\sum_{k=-\infty}^{\infty} \langle \hat{\phi}, e_{-k}\hat{\varphi}\rangle \mathcal{F}e_{-k}\right)\hat{\varphi}.\end{aligned}$$

The inner product gets

$$\begin{aligned}\langle \hat{\phi}, e_{-k}\hat{\varphi}\rangle &= \int_{\mathbb{R}} \hat{\phi}(\omega)\overline{\hat{\varphi}(\omega)}\mathrm{e}^{2\pi i\omega k}\,\mathrm{d}\omega \\ &= \int_0^1 \sum_{n=-\infty}^{\infty} \hat{\phi}(\omega+n)\overline{\hat{\varphi}(\omega+n)}\mathrm{e}^{2\pi i\omega(k+n)}\,\mathrm{d}\omega \\ &= \int_0^1 \sum_{n=-\infty}^{\infty} \hat{\phi}(\omega+n)\overline{\hat{\varphi}(\omega+n)}\mathrm{e}^{2\pi i\omega(k+n)}\,\mathrm{d}\omega.\end{aligned}$$

The functions e_{-k}, Φ, and χ_D are 1-periodic, thus

$$\begin{aligned}\langle \hat{\phi}, e_{-k}\hat{\varphi}\rangle &= \int_0^1 \sum_{n=-\infty}^{\infty} \frac{|\hat{\varphi}(\omega+n)|^2}{\Phi(\omega+n)}\chi_D(\omega+n)\mathrm{e}^{2\pi i\omega(k+n)}\,\mathrm{d}\omega \\ &= \int_0^1 \Phi(\omega)^{-1}\chi_D(\omega)\mathrm{e}^{2\pi i\omega k}\sum_{n=-\infty}^{\infty} |\hat{\varphi}(\omega+n)|^2\,\mathrm{d}\omega \\ &= \int_0^1 \chi_D(\omega)\overline{e_{-k}}\,\mathrm{d}\omega,\end{aligned}$$

which is exactly the $-k$'th Fourier coefficient of χ_D. Therefore, the series yields to

$$\sum_{k=-\infty}^{\infty} \langle \hat{\phi}, e_{-k}\hat{\varphi}\rangle \mathcal{F}e_{-k} = \chi_D e.$$

As result we obtain $\mathcal{F}S\phi = \chi_D\hat{\varphi}$. By definition of the set D, it follows that $\chi_D(\omega) = 1 \Leftrightarrow \hat{\varphi}(\omega) \neq 0$. To sum up,

$$\mathcal{F}S\phi = \chi_D\hat{\varphi} = \hat{\varphi} \quad \Rightarrow S\phi = \varphi \quad \Rightarrow \phi = S^{-1}\varphi. \qquad \square$$

Theorem 8 *Let $\varphi \in L^2(\mathbb{R})$ generate the Riesz sequence $\{T_k\varphi : k \in \mathbb{Z}\}$ and $\tilde{\varphi}$ generate the dual frame, i. e., $\tilde{\varphi} = S^{-1}\varphi$. Then for all $f \in V_\varphi$*

$$f = \sum_{k\in\mathbb{Z}} \langle f, T_k\tilde{\varphi}\rangle T_k\varphi \tag{2.27}$$

and

$$\langle \varphi, T_k \tilde{\varphi} \rangle = \delta_{0,k}. \tag{2.28}$$

The sequences $\{T_k \varphi : k \in \mathbb{Z}\}$ *and* $\{T_k \tilde{\varphi} : k \in \mathbb{Z}\}$ *are said to be bi-orthogonal.*

Note that the dual frame elements $T_k \tilde{\varphi} \in V_\varphi$ for all $k \in \mathbb{Z}$ and therefore, the dual frame is also generated by integer translates.

2.4. B-Splines

In general, splines are continuous piecewise polynomial functions up to a certain degree $n \in \mathbb{N}$ which do have continuous derivatives up to order $n - 1$. If the knots of the splines are at the integer values, Schoenberg showed in 1946 that the integer translates of the B-splines form a basis for the space of splines of equivalent degree. Curry and Schoenberg generalized this result in 1966 to arbitrary knots [7, 8]. The B-Splines are compactly supported and have an interpolation order of n. Moreover, they have minimal support for a given interpolation order, as shown by Blu et al. [2], Boor and Lynch [4].

In this section, elementary properties and results about these basic splines are shown.

Definition 5 (B-Spline) The symmetric cardinal *B-Spline* β_n of order $n \in \mathbb{N}_0$ is inductively defined for $x \in \mathbb{R}$ as

$$\begin{aligned} \beta_0(x) &:= \chi_{\left[-\frac{1}{2}, \frac{1}{2}\right]}(x) \\ \beta_n(x) &= (\beta_{n-1} * \beta_0)(x), \qquad \text{for } n \in \mathbb{N}. \end{aligned}$$

Figure 2.5 shows the cardinal B-Splines of orders 0 to 3. With increasing order the spline gets smoother while the support gets larger.

To be more specific the characteristic properties of the B-Spline are shown next.

Theorem 9 (Basic properties) *Let* $n \in \mathbb{N}_0$*, then*

(i) $\operatorname{supp}(\beta_n) = \left[-\frac{n+1}{2}, \frac{n+1}{2}\right]$,

(ii) $\hat{\beta}_n(0) = \int_{\mathbb{R}} \beta_n(x) \mathrm{d}x = 1$,

(iii) for $n \geq 1$*, the partition of the unity holds, i. e.,* $\sum_{k \in \mathbb{Z}} \beta_n(x - k) = 1$ *for all* $x \in \mathbb{R}$*, and finally*

(iv) $\hat{\beta}_n(\omega) = \left(\frac{\sin(\pi\omega)}{\pi\omega}\right)^{n+1} = (\operatorname{sinc}(\omega))^{n+1}$.

The first property states that a B-Spline of order n is supported on an interval of length n. Secondly the L^2-energy of B-Spline of any order is always normalized, *i. e.,* equal to one. B-Splines form a partition of the unity, thus they are able to reproduce constants exactly. The last item characterizes the frequency behavior of the B-Spline where the fast decay with increasing degree has to be remarked.

According to Strang and Fix [36] having interpolation order of n can be examined by the derivatives of the Fourier transform.

Theorem 10 (Strang-Fix) *Let $\varphi \in L^2(\mathbb{R})$. Then the following conditions are equivalent:*

1. *φ has interpolation or approximation order n*
2. *It holds that*

$$\hat{\varphi}(0) = 1, \tag{2.29}$$

and for all $m \in \{0, 1, \dots, n\}$

$$D^m \hat{\varphi}(k) = 0 \quad \text{for all } k \in \mathbb{Z} \setminus \{0\} \tag{2.30}$$

3. *All monomials of degree up to n can be reproduced*

Property 3 of Theorem 10 implies the first condition for the characterization with the derivatives of the Fourier transform. Using the basic property (iv), for $k \in \mathbb{Z} \setminus \{0\}$ and $m \in \{0, 1, \dots, n\}$ the following equality holds

$$D^m \hat{\beta}_n(k) = \frac{(n+1)!}{(n+1-m)!} \left(\operatorname{sinc}(k)\right)^{n+1-m} \cdot R(k), \tag{2.31}$$

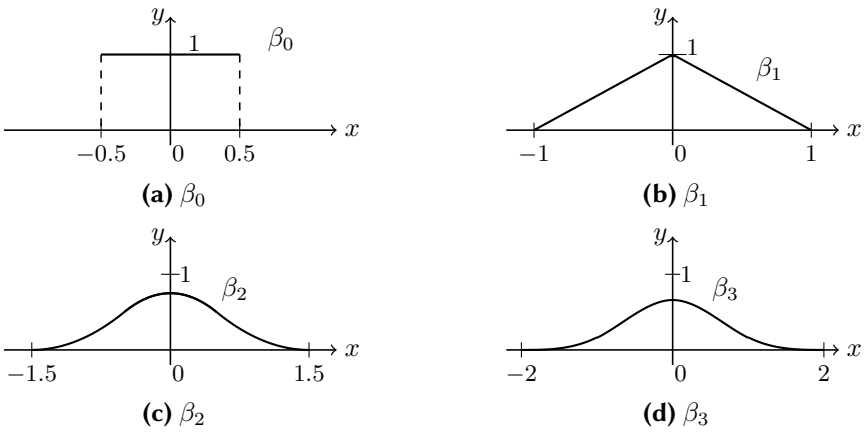
with some unspecified term $R(k)$. Because $\operatorname{sinc}(k) = \delta_{0,k}$ for integer numbers $k \in \mathbb{Z}$ it follows directly that $D^m \hat{\beta}_n(k) = 0$. Thus, the second Strang-Fix condition of Theorem 10 shows that B-Splines of degree n have interpolation order n. More specifically, they have minimal support for this property, see Boor and Lynch [4].

In the beginning it was mentioned that the B-Splines of degree n form a basis of the Splines with the same degree. The next theorem will link this statement with the results of the frame theory and guarantee the uniqueness of the B-Spline expansion.

Theorem 11 *For $n \in \mathbb{N}$, the translates of the B-spline of order n $\{T_k \beta_n \,:\, k \in \mathbb{Z}\}$ form a Riesz sequence and a Riesz basis for* span $\{T_k \beta_n \,:\, k \in \mathbb{Z}\}$.

2.5. Calculation of the frame coefficients

Looking back to Shannon's sampling theorem, it is known that the integer translates of the B-Spline of zero degree, $\beta_0(x) = \chi_{\left[-\frac{1}{2}, \frac{1}{2}\right]}(x)$, build an orthonormal basis for its generating space, as the shifted sinc-function build an orthonormal basis.



(a) β_0 **(b)** β_1 **(c)** β_2 **(d)** β_3

Figure 2.5. *Examples of the cardinal B-Spline.*

B-Splines of higher degrees only form a Riesz basis. According to Theorem 7, the dual frame basis can be calculated, if one can evaluate the series

$$\Phi(\omega) = \sum_{k\in\mathbb{Z}} \left|\hat{B}_n(\omega + k)\right|^2 . \tag{2.32}$$

The evaluation of these series for high orders is a difficult and time consuming task. In real world problems the dual frame basis is usually not explicitly required, the interest instead lies only on the analysis coefficients which can be examined by another way.

This section introduces one way to calculate these frame coefficients.

If the translates of φ build a Riesz basis then the coefficients are uniquely defined through the dual basis $\{T_k\tilde{\varphi} : k \in \mathbb{Z}\}$:

$$c_k = \langle f, T_k\tilde{\varphi}\rangle. \tag{2.33}$$

There is also a way to calculate the coefficients without the knowledge of the dual basis:

$$\begin{aligned} c_k &= \langle f, T_k\tilde{\varphi}\rangle = \langle f, \sum_{l\in\mathbb{Z}} \langle T_k\tilde{\varphi}, T_l\tilde{\varphi}\rangle T_l\varphi\rangle \\ &= \sum_{l\in\mathbb{Z}} \overline{\langle T_k\tilde{\varphi}, T_l\tilde{\varphi}\rangle}\langle f, T_l\varphi\rangle. \end{aligned}$$

Written with the frame operators, where U and $\tilde{U}$ denotes the analysis operator of the frame basis and its dual basis respectively.

$$c = \tilde{U}^*\tilde{U} \cdot U^* f. \tag{2.34}$$

Because the dual basis is characterized through the bi-orthogonality constrained $\langle T_k\varphi, T_l\tilde{\varphi}\rangle = \delta_{k,l}$ for $k, l \in \mathbb{Z}$, it holds that

$$U^*U \cdot \tilde{U}^*\tilde{U} = U^*(U\tilde{U}^*)\tilde{U} = U^*\tilde{U} = \mathrm{Id}. \tag{2.35}$$

By multiplying (2.34) with U^*U, the coefficients can be calculated as solution of

$$U^*Uc = U^*f, \tag{2.36}$$

where no knowledge about the dual basis is needed anymore. But instead, the inverse of the grammian U^*U is needed.

In the discrete setting the gram matrix is usually very sparse for integer translates. An efficient way to solve this is to use the conjugate gradient method ([35]). For large images, solving this linear system with conjugate gradient methods is still too slow. The next chapters will show an alternative calculation method, where the special characteristics of the gram matrix are used.

3. Introduction to Modern Sampling Theory

In Chapter 2, the tools of the classical sampling theory were introduced. This chapter continues with a brief insight into recent advances of modern sampling theory. In Unser [39], the reader finds a comprehensive overview of the progress in the sampling theory of in last decades.

This chapter extends the sampling theory based on Shannon from the space of band-limited functions to the space generated by the translates of a certain function. Especially, the space of splines generated by shifted B-splines is considered due to its approximation properties. As splines are in general not band-limited and cannot be reconstructed from the sample values with Shannon's sampling theorem, shown in the last chapter, sampling in the space of splines is a true generalization. Also, this chapter shows that the space of band-limited functions is equal to the space of splines of infinite degree.

Recall that the translates of a function $\varphi \in L^2(\mathbb{R})$ generate a closed subspace of $L^2(\mathbb{R})$

$$V_\varphi = \overline{\left\{ \sum_{k\in\mathbb{Z}} c_k T_k \varphi \,:\, (c_k)_{k\in\mathbb{Z}} \right\}}.$$

If there are $0 < A \leq B$ such that

$$A \leq \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega + k)|^2 \leq B,$$

the sequence $\{T_k\varphi \,:\, k \in \mathbb{Z}\}$ form a Riesz basis for the closed space V_φ, as shown in Section 2.3.3.

In this setting, the orthogonal projection of a general function $f \in L^2(\mathbb{R})$ into the space V_φ is given by the operator $P_\varphi : L^2(\mathbb{R}) \to V_\varphi$

$$P_\varphi f = \operatorname*{argmin}_{g\in V_\varphi} \|f - g\|_{L^2(\mathbb{R})} = \sum_{k\in\mathbb{Z}} \langle f, T_k\tilde{\varphi}\rangle T_k\varphi,$$

where the integer translates of $\tilde{\varphi}$ generate the bi-orthogonal dual basis as shown in Theorem 7.

3.1. Shannon sampling revisited

In this section, the already known sampling theory according to Shannon is viewed in the context of translates of a basis function. In particular, the integer translates of the function $\varphi(x) = \operatorname{sinc}(x)$ are considered.

One can show that these translates form an orthonormal basis:

$$\langle T_k \operatorname{sinc}, T_l \operatorname{sinc} \rangle = \operatorname{sinc}(k-l) = \delta_{k,l} \qquad \text{for } k, l \in \mathbb{Z}.$$

To simplify notation, the following is discussed only for functions which are band-limited to $[-1/2, 1/2]$. This corresponds to proper sampling rates of $\rho \geq 1$ and to a sampling step lower than one. Here the sampling rate of $\rho = 1$ is chosen, because sampling at the integers is required in a discrete setting. Different sampling rates are treated in an analog manner.

The Shannon sampling theorem (Section 2.2) states that every function $f \in PW_1$ can be represented as

$$f(x) = \sum_{k \in \mathbb{Z}} f(k) \operatorname{sinc}(x-k) = \sum_{k \in \mathbb{Z}} c_k \operatorname{sinc}(x-k),$$

for the coefficients $c_k = f(k)$, $k \in \mathbb{Z}$. As a consequence, the translates of sinc form an orthonormal basis of the space of band-limited functions PW_1, and the coefficients

$$c_k = \langle f, T_k \operatorname{sinc} \rangle = f(k)$$

are the projections to the translates. Not that if $f \notin PW_1$, then $c_k \neq f(k)$. The Fourier transform yields

$$\begin{aligned} \hat{c}_k &= \langle f, \widehat{T_k \operatorname{sinc}} \rangle = \widehat{f * \operatorname{sinc}} = \sum_{l \in \mathbb{Z}} f(\omega + l) \cdot \sum_{l \in \mathbb{Z}} \chi_{[-\frac{1}{2}, \frac{1}{2}]}(\omega + l) \\ &= \chi_{[-\frac{1}{2}, \frac{1}{2}]}(\omega) \cdot \sum_{l \in \mathbb{Z}} f(\omega + l), \end{aligned}$$

which is obviously different to $\hat{f}$, if $f \notin PW_1$. The samples are pre-filtered with the ideal filter.

With that knowledge, the orthogonal projection $P_{\operatorname{sinc}} : L^2(\mathbb{R}) \to PW_1$ is given by

$$P_{\operatorname{sinc}} f = \sum_{k \in \mathbb{Z}} \langle f, T_k \operatorname{sinc} \rangle T_k \operatorname{sinc},$$

and obviously $Pf = f$ holds for $f \in PW_1$. Interpreting this results yields to the fact that the Shannon sampling theorem minimizes the L^2-error for a function f that is not in PW_1

$$\min_{(c_k) \in l^2(\mathbb{Z})} \left\| f - \sum_{k \in \mathbb{Z}} c_k T_k \operatorname{sinc} \right\|_{L^2(\mathbb{R})}$$

with the coefficients $(c_k)_{k\in\mathbb{Z}} = (\langle f, T_k \operatorname{sinc}\rangle)_{k\in\mathbb{Z}}$. Furthermore, if $f \in PW_1$, the coefficients coincide with the sampled values of f.

The question arises if there are other spaces that are larger than the space of band-limited functions. Then more functions can be reconstructed from the samples if they are in this space. The space of piecewise polynomials is considered as this space and will be treated in the next section.

3.2. Generalized sampling

To generalize the Shannon sampling concept instead of the sinc-function general functions $\varphi \in L^2(\mathbb{R})$ are considered, that satisfy certain properties.

Let $\varphi \in L^2(\mathbb{R})$, such that the sequence of integer translates $\{T_k\varphi \,:\, k \in Zz\}$ form a Riesz basis for its generated space

$$V_\varphi = \operatorname{span}\{T_k\varphi \,:\, k \in \mathbb{Z}\}. \tag{3.1}$$

In the same manner as in the Shannon sampling theorem the orthogonal projection into this space can be considered. Let $f \in L^2(\mathbb{R})$

$$P_\varphi f(x) = \sum_{k\in\mathbb{Z}} \langle f, T_k\tilde{\varphi}\rangle \varphi(x-k) = \sum_{k\in\mathbb{Z}} c_k \varphi(x-k). \tag{3.2}$$

The difference to the Shannon theorem are the coefficients (c_k) which are different than the sampled values of f.

The coefficients c_k can be seen as a pre-filtered version of f with the filter $\tilde{\varphi}(-x)$. The following synthesis with the translates of φ can be seen as post-filtering. Instead of the ideal pre- and post-filter proposed by Shannon, the filters determined by the function φ are used.

For the discrete setting Thevenaz et al. [37] showed an interesting relation to interpolation. Let $(f_k) \in l^2(\mathbb{Z})$. The minimizer for the discrete minimization problem

$$\min_{c\in l^2(\mathbb{Z})} \sum_{n\in\mathbb{Z}} \left| f_n - \sum_{k\in\mathbb{Z}} c_k T_k \varphi(n) \right|^2 \tag{3.3}$$

interpolates the sequence exactly at the integer values, *i. e.,*

$$f_n = \sum_{k\in\mathbb{Z}} c_k T_k \varphi(n), \qquad \text{for all } n \in \mathbb{Z}. \tag{3.4}$$

This restriction can be re-written as a discrete convolution of the missing coefficients (c_k) and the integer values of φ. For a clear notation, the sequence $(\varphi_k)_{k\in\mathbb{Z}}$ denotes the integer

sequence generated by φ, *i. e.*, $\varphi_k = \varphi(k)$ for all $k \in \mathbb{Z}$. Then the minimization problem (3.3) yields to the deconvolution problem

$$f_n = \sum_{k\in\mathbb{Z}} c_k \varphi(n-k) = ((c_k)_{k\in\mathbb{Z}} * (\varphi_k)_{k\in\mathbb{Z}})(n).$$

The problem now consists of two tasks. First the coefficients c_k have to be calculated then the orthogonal projection can be represented as the series expansion with those coefficients and the integer translates of φ.

If one denotes with φ_k^{-1} the convolution inverse of the sequence $(\varphi_k)_{k\in\mathbb{Z}}$, *i. e.*,

$$((\varphi_k)^{-1} * (\varphi_k))(n) = \delta_{0,n}, \tag{3.5}$$

the calculation of the coefficients simplifies to

$$(c_n) = ((f_k) * (\varphi_k)^{-1})(n).$$

Combining the previous steps, the interpolated function $g \in L^2(\mathbb{R})$, where $g(n) = f_n$ for all $n \in \mathbb{Z}$, is given by

$$\begin{aligned} g(x) &= \sum_{k\in\mathbb{Z}} c_k \varphi(x-k) = \sum_{k\in\mathbb{Z}} (f * \varphi_k^{-1})\varphi(x-k) \\ &= \sum_{k\in\mathbb{Z}} f(k) \underbrace{\sum_{l\in\mathbb{Z}} \varphi_l^{-1} \varphi(x-k-l)}_{\varphi_{int}(x-k)} \qquad = \sum_{k\in\mathbb{Z}} f(k)\varphi_{int}(x-k). \end{aligned} \tag{3.6}$$

In brief, the function $\varphi_{int}(x-k)$ replaces the sinc-function and the property

$$\varphi_{int}(k-l) = \delta_{k,l}, \qquad \text{for } k,l \in \mathbb{Z} \tag{3.7}$$

holds. For a given sequence $(f_k)_{k\in\mathbb{Z}}$, the representation in (3.2) defines a projection into V_φ. With respect to the computation of this projection, the function φ_{int} should be compactly supported in contrast to the sinc function. As a consequence the series would turn into a finite sum.

3.3. Sampling with B-splines

The space of band-limited function PW_α is appropriate to handle in theory, but has too many drawbacks for real-world problems. The space of splines of a certain degree is significantly larger. An increasing order of the degree leads to smoother functions and the Paley-Wiener space PW_1 corresponds to the space of splines of infinite degree.

According to the theory of the last section, this section considers integer translates of a B-spline function. Therefore, let $\varphi(x) = \beta_n(x)$ for a given order n. A general function $f \in L^2(\mathbb{R})$ has to be approximated in the space V_{β_n} generated by the translates of B-Spline. To

clarify the notation the integer samples of the B-Spline are denoted with $b_k = \beta_n(k)$ also it should be kept in mind that the B-Splines are compactly supported. The convolution inverse $(b_k)^{-1}_{k\in\mathbb{Z}}$ of the sequence $(b_k)_{k\in\mathbb{Z}}$ is defined as in (3.5).

According to (3.2), the orthogonal projection of f to the space of splines of degree n has the B-Spline expansion

$$\begin{aligned} f(x) &= \sum_{k\in\mathbb{Z}} \langle f, T_k \tilde{\beta}_{m-1} \rangle \beta_{m-1}(x-k) \\ &= \sum_{k\in\mathbb{Z}} f(k) \underbrace{(\tilde{\beta}_{m-1}(-x) * \beta_{m-1}(x))(x-k)}_{\varphi_{int}(x-k)}, \end{aligned} \tag{3.8}$$

while the interpolating projection is defined by

$$\begin{aligned} f(x) &= \sum_{k\in\mathbb{Z}} c_k \beta_n(x-k) \\ &= \sum_{k\in\mathbb{Z}} f(k) \underbrace{\sum_{l\in\mathbb{Z}} b_l^{-1} \beta_n(x-k-l)}_{\varphi_{int}(x-k)}. \end{aligned} \tag{3.9}$$

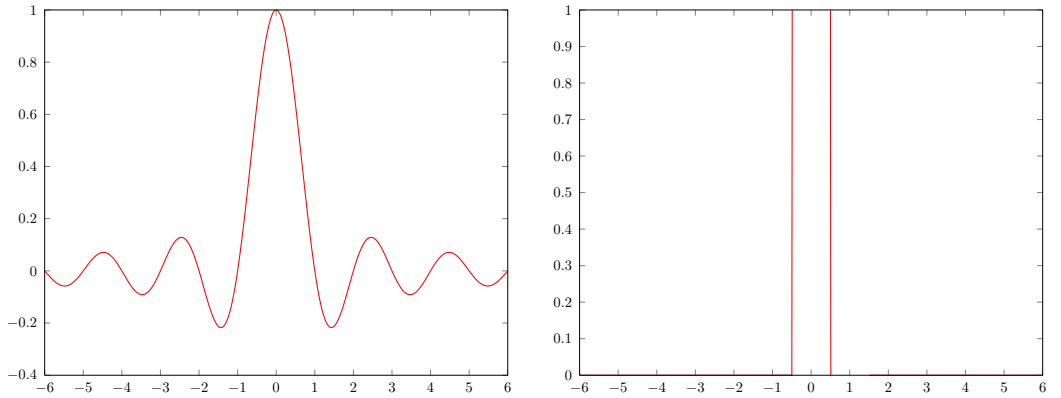
In the following the functions φ_{int} will be discussed and related to the sinc-function.

Figure 3.1 shows that the function φ_{int} looks very similar to the sinc-function. In common is the interpolation property, *i. e.,*

$$\varphi_{int}(k) = \sum_{l\in\mathbb{Z}} b_l^{-1} \beta_n(k-l) = \delta_{0,k} \quad \text{for all } k \in \mathbb{Z}, \tag{3.10}$$

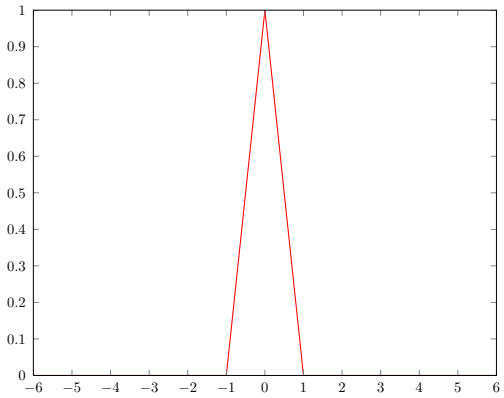
as a direct consequence of the definition of the inverse convolution. Furthermore Aldroubi et al. [1] showed that if the spline degree tends to infinity, the function φ_{int} converges to the sinc-function in the L^2-norm as well as the corresponding Fourier transform $\hat{\varphi}_{int}$ converges to the ideal filter $\chi_{\left[-\frac{1}{2},\frac{1}{2}\right]}$. This result clarifies, that the sampling theory based on Shannon is included as a limit case of the B-Spline interpolation.

Next the decay rate is compared. The slow decay rate of $\frac{1}{x}$ follows directly by the definition of the sinc-function. De Boor [10] showed that, in contrary, the function φ_{int} has an exponential fast decay rate. For all degrees the function φ_{int} is numerically compactly supported and hence can be computed very efficiently. The decay of several φ_{int}-functions are illustrated in Figure 3.1.

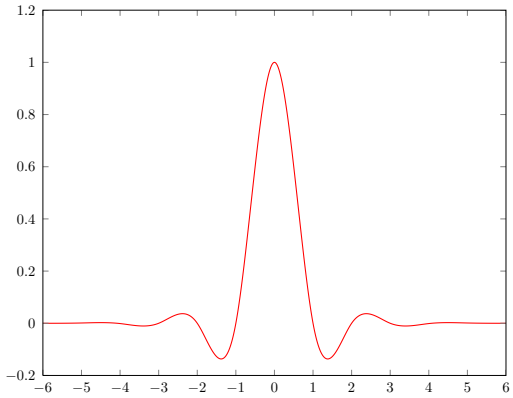


(a) sinc

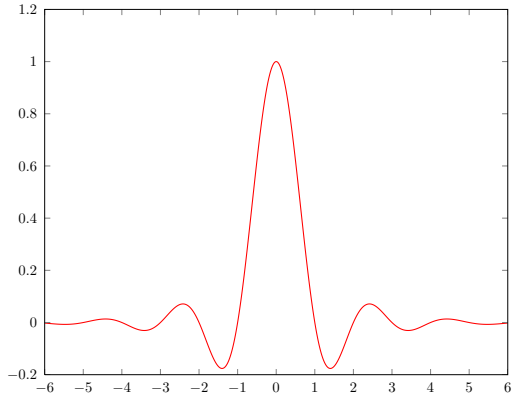
(b) φ_{int} *for* β_0



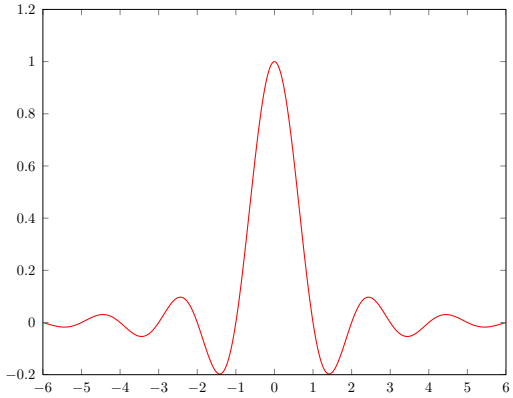
(c) φ_{int} *for* β_1



(d) φ_{int} *for* β_3



(e) φ_{int} *for* β_5



(f) φ_{int} *for* β_8

Figure 3.1. *The sinc-function and the φ_{int}-function for several B-splines. Compared to the infinite support of the sinc-function, the compact support of the φ_{int} function gets larger with increasing degree of the B-splines. Note that all functions values are zero at the integers except they have value one at the origin.*

4. Splines and the Butterworth filter

In the preceding chapter, sampling with B-splines was introduced by the projection to the space of splines. B-splines have minimal support and are maximally flat in respect to a given interpolation order. Furthermore, the B-spline interpolation function φ_{int}, defined in (3.9), converges with increasing degree of the B-spline to the sinc-function. Or equivalently, the Fourier transform $\hat{\varphi}_{int}$ converges to the ideal filter [1].

Since 1930, the so called Butterworth filter is well known in the literature of signal processing [5]. This filter is specified by a cutoff frequency and the order of the filter. It is designed to be maximally flat, *i. e.,* the derivatives up to twice the order of the filter are continuous. As the function $\hat{\varphi}_{int}$, the Butterworth filter converges to the ideal filter with increasing order. Additionally, the filter has no ripples and can be interpreted through the cutoff frequency analog to the ideal filter.

This chapter connects the theory of the splines with Butterworth filter, which leads to a new interpretation of B-spline approximation due to the recent work of Unser and Blu [41]. Besides, the relation of a regularized minimization problem and the splines is established. The intention is to get a better understanding of the projection to spline spaces and to motivate their usage in this thesis once more.

The first section naturally relates the mathematical model, introduced in Section 1.3, to the space of splines as the smoothest interpolating functions. Section 4.2 re-interprets the interpolation function φ_{int} with the use of the Butterworth filter.

4.1. Interpolation problem

For a given sequence $(f_k)_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$ one is interested in an interpolated function $f \in L^2(\mathbb{R})$, where $f(k) = f_k$ for $k \in \mathbb{Z}$ and f sufficient smooth. The smoothness is specified by the continuous derivatives up to a certain order $m \in \mathbb{N}$. To achieve a smooth solution, the highest order derivative is minimized in the L^2-norm. The function space, where all functions are differentiable up to order m, is the Sobolev space $W^m(\mathbb{R})$,

$$W^m(\mathbb{R}) := \left\{ f \in L^2(\mathbb{R}) \,:\, D^k f \in L^2(\mathbb{R}) \text{ for all } k \leq m \right\}. \tag{4.1}$$

Note that for all $m \in \mathbb{N}$, the space $W^m(\mathbb{R})$ is continuously embedded into the space of continuous functions. In two dimensions, this statement is only true for $m > 1$. As consequence, sampling in this space is well defined.

A formulation of previously discussed constraints leads to the minimization problem

$$\min_{g \in W^m} \|D^m g\|, \quad \text{such that } g(k) = f_k \; \forall k \in \mathbb{Z}. \tag{4.2}$$

The next theorem, first stated by Schoenberg [33] in 1973 and further enhanced by Unser and Blu [40], characterizes the solution of the minimization problem (4.2).

Theorem 12 *The solution $g \in W^m(\mathbb{R})$ of the minimization problem* (4.2) *is a spline of order $2m-1$. It can be written as expansion of the basis functions $\varphi(x) = (\beta_{m-1}(x) * \beta_{m-1}(-x))$:*

$$g(x) = \sum_{k \in \mathbb{Z}} (h * f)(k) \varphi(x-k), \tag{4.3}$$

where the coefficients of $(h_k)_{k\in\mathbb{Z}} \in l^1(\mathbb{Z})$ are defined by the Fourier series

$$H(\omega) = \sum_{k \in \mathbb{Z}} h_k \mathrm{e}^{2\pi i k \cdot \omega} = \frac{1}{\sum_{k\in\mathbb{Z}} \left| \hat{\beta}_{m-1}(\omega + k) \right|^2}. \tag{4.4}$$

Equation (4.3) can be further written as convolution of $(f_k)_{k\in\mathbb{Z}}$ and a function φ_{int},

$$\begin{aligned} g(x) &= \sum_{k\in\mathbb{Z}} (h*f)(k)\varphi(x-k) = \sum_{k\in\mathbb{Z}} \left[\sum_{l\in\mathbb{Z}} h_l f(k-l) \right] \varphi(x-k) \\ &= \sum_{k\in\mathbb{Z}} \left[\sum_{l\in\mathbb{Z}} h_l f(k) \right] \varphi(x-k-l) = \sum_{k\in\mathbb{Z}} f(k) \sum_{l\in\mathbb{Z}} h_l \varphi(x-k-l) \\ &= \sum_{k\in\mathbb{Z}} f(k) \varphi_{int}(x-k), \end{aligned} \tag{4.5}$$

where the function φ_{int} is defined as $\varphi_{int}(x) = \sum_{l\in\mathbb{Z}} h_l \varphi(x-l)$. Interpreting the convolution in the last line as filtering, this function acts like a filter on the integer values $(f_k)_{k\in\mathbb{Z}}$. It can be explicitly expressed as the Fourier transform

$$\hat{\varphi}_{int}(\omega) = H(\omega) \cdot \hat{\varphi}(\omega) = \frac{\hat{\beta}_{m-1}(\omega) \cdot \overline{\hat{\beta}_{m-1}(\omega)}}{\sum_{k\in\mathbb{Z}} \left| \hat{\beta}_{m-1}(\omega+k) \right|^2} = \frac{\hat{\beta}_{2m-1}(\omega)}{\sum_{k\in\mathbb{Z}} \left| \hat{\beta}_{m-1}(\omega+k) \right|^2}, \tag{4.6}$$

using the convolution theorem (2.4) and $\mathcal{F}(\beta_{m-1}(-x))(\omega) = \overline{\hat{\beta}_{m-1}(\omega)}$ due to the real-valued B-spline β_{m-1}. The middle representation consists of the B-spline $\hat{\beta}_{m-1}$ multiplied with its dual (bi-orthogonal) function as shown in Theorem 7. Note that the solution of the minimization problem (4.2) has a similar Fourier transform as the orthogonal projection of the sampled function f into the space of splines of degree $m-1$, namely

$$g(x) = \sum_{k\in\mathbb{Z}} \langle f, T_k \tilde{\beta}_{m-1} \rangle \beta_{m-1}(x-k), \tag{4.7}$$

where the bi-orthogonal dual B-spline basis is defined by Theorem 7 for $\varphi(x) = \beta_{m-1}(x)$.

To clarify this relation, the explicit calculation of the projection $g(x)$ yields to

$$\begin{aligned} g(x) &= \sum_{k\in\mathbb{Z}} \langle f, T_k \tilde{\beta}_{m-1}\rangle \beta_{m-1}(x-k) \\ &= \sum_{k\in\mathbb{Z}} \langle \tilde{\beta}_{m-1}, T_{-k} f\rangle \beta_{m-1}(x-k) \\ &= \sum_{k\in\mathbb{Z}} \left[\sum_{l\in\mathbb{Z}} \tilde{\beta}_{m-1}(l) f(k+l)\right] \beta_{m-1}(x-k) \\ &= \sum_{k\in\mathbb{Z}} \left[\sum_{l\in\mathbb{Z}} \tilde{\beta}_{m-1}(-l) f(k-l)\right] \beta_{m-1}(x-k) \\ &= \sum_{k\in\mathbb{Z}} f(k) \sum_{l\in\mathbb{Z}} \tilde{\beta}_{m-1}(-l)\beta_{m-1}(x-k-l) \\ &= \sum_{k\in\mathbb{Z}} f(k) \underbrace{(\tilde{\beta}_{m-1}(-x) * \beta_{m-1}(x))(x-k)}_{\varphi_{int}}. \end{aligned} \tag{4.8}$$

Using Theorem 7 and the fact that $\tilde{\beta}_{m-1}$ is a real function, one finally obtains the result related to (4.6):

$$\hat{\varphi}_{int}(\omega) = \overline{\hat{\tilde{\beta}}_{m-1}(\omega)} \cdot \hat{\beta}_{m-1}(\omega) = \frac{\hat{\beta}_{m-1}(\omega) \cdot \sum_{k\in\mathbb{Z}} \overline{\hat{\beta}_{m-1}(\omega+k)}}{\sum_{k\in\mathbb{Z}} \left|\hat{\beta}_{m-1}(\omega+k)\right|^2}. \tag{4.9}$$

Back to equation (4.6), the interpolated function can be obtained by calculating the convolution with the discrete values of f and the function φ_{int}.

4.2. Regularized approximation

If the interpolation problem is ill-posed where the data is noisy and the solution may not be continuous or even unique, one tries to regularize it. Since the smoothness of the solution is related to the energy of its derivatives one can use the highest order differentiation of the Sobolev space W^m as regularization term. According to this consideration, the interpolation problem turns into the approximation problem

$$\min_{g\in W^m} \sum_{k\in\mathbb{Z}} |f_k - f(k)|^2 + \lambda \left\|D^m f\right\|^2, \tag{4.10}$$

where $\lambda \geq 0$ is a regularization parameter. The first term corresponds to the interpolation problem, while the second term is a penalty term that ensures the smoothness of the solution. Setting the regularization parameter λ to zero leads to the interpolation problem treated in the last section. Larger values of λ act like a low-pass filter, where high frequencies are attenuated, and the solution gets smoother. As consequence, the solution may differ at the integer values from f_k.

Schoenberg [32] and Reinsch [30] showed that the solution of this problem is also a spline of degree $2m-1$. With the approximation property of the splines follows that every polynomial up to order $2m-1$ will be reproduced independently of the values of λ. The next theorem proved by Unser and Blu [41] characterizes the solution more clearly:

Theorem 13 *The solution of the regularized interpolation problem in* (4.10) *is a spline of degree* $2m-1$. *It is given by*

$$f(x) = \sum_{k\in\mathbb{Z}} (h * f)(k)\varphi(x-k) = \sum_{k\in\mathbb{Z}} f_k \varphi_{int}(x-k), \tag{4.11}$$

where the function $\varphi \in L^2(\mathbb{R})$ *is defined by* $\varphi(x) = (\beta_{m-1}(x) * \beta_{m-1}(-x))$. *The coefficients of* $h = (h_k)_{k\in\mathbb{Z}}$ *are specified through the Fourier series*

$$H(\omega) = \sum_{k\in\mathbb{Z}} h_k \mathrm{e}^{2\pi i k\cdot\omega} = \frac{1}{\sum_{k\in\mathbb{Z}} \left|\hat{\beta}_{2m-1}(\omega+k)\right|^2 + \lambda\left|1-\mathrm{e}^{-2\pi i\cdot\omega}\right|^{2m}}. \tag{4.12}$$

The function φ_{int} *can be represented as the spline expansion*

$$\varphi_{int}(x) = \sum_{k\in\mathbb{Z}} h_k \beta_{2m-1}(x-k). \tag{4.13}$$

The Fourier transform of φ_{int} shows some interesting properties. Applying the convolution theorem, shown in Lemma 2, to $\hat{\varphi}_{int}$ leads to

$$\hat{\varphi}_{int}(\omega) = H(\omega)\cdot\hat{\beta}_{2m-1}(\omega). \tag{4.14}$$

Replacing those terms shows that

$$\hat{\varphi}_{int}(\omega) = \frac{1}{\sum_{k\in\mathbb{Z}} \left|\frac{\pi\omega}{\pi\omega+\pi k}\right|^{2m} + 2\pi\lambda\left|\omega\right|^{2m}}. \tag{4.15}$$

Following the recent work of Unser and Blu [41] the next theorem shows that $\hat{\varphi}_{int}$ is an approximation of the well known Butterworth filter. The Butterworth filter B^m of order $m \in \mathbb{N}$ is defined as for $\omega_0 \in (0,1]$ as

$$B^m_{\omega_0}(\omega) = \frac{1}{1+\left(\frac{\omega}{\omega_0}\right)^{2m}}. \tag{4.16}$$

Theorem 14 *The Fourier transform* $\hat{\varphi}_{int}$ *of the smoothing spline estimator of order* $2m-1$ *and regularization parameter* $\lambda \geq 0$ *given in* (4.15) *approximates the Butterworth filter of order* m *with increasing* m *or* λ. *The relation between* λ *and the cut-off frequency* ω_0 *is given by*

$$\omega_0 = \left(\lambda + 2^{2m}\right)^{-\frac{1}{2m}} \leq 0.5, \tag{4.17}$$

$$\lambda = \omega_0^{-2m} - 2^{2m}. \tag{4.18}$$

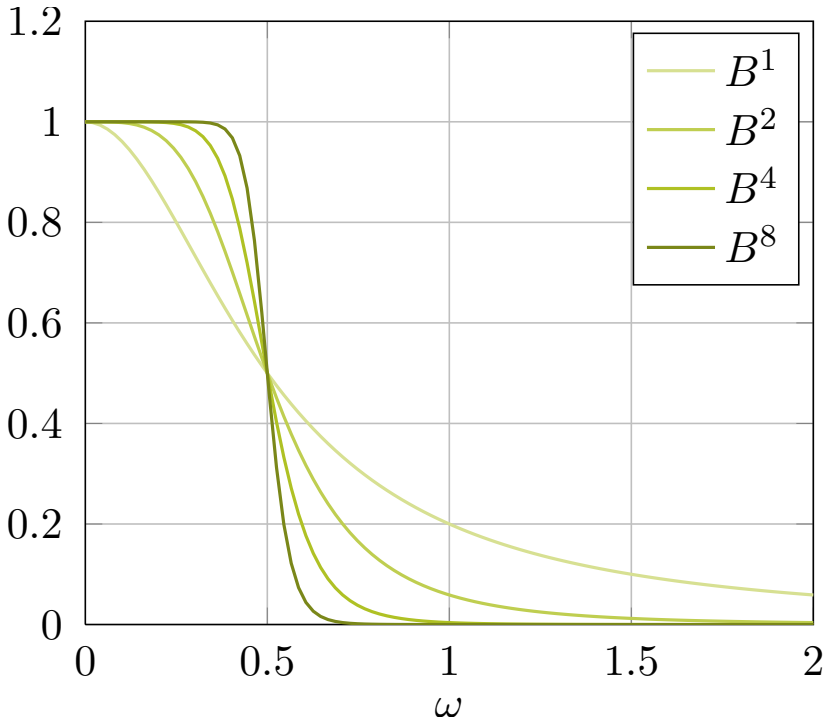


Figure 4.1. *Butterworth filter for orders* $n = 1, 2, 4, 8$.

Returning to the interpolation problem with $\lambda = 0$ the corresponding Butterworth filter gives an interesting interpretation of the spline interpolation. The solution of (4.2) is the sequence $(f_k)_{k\in\mathbb{Z}}$ filtered by the Butterworth filter with cut-off frequency 0.5.

This shows that the interpolant $f \in W^m$ is nearby band-limited in $[-1/2, 1/2]$ and thus can be reconstructed at the values $f(2k)$, $k \in \mathbb{Z}$ according to the generalized sampling theory. Contrary filtering a sequence with a Butterworth filter of a high order, e. g., $m = 8$ can be interpreted as solving the interpolation problem. High order Butterworth filter have a high interpolation order and as such can reproduce any polynomials up to order $2m - 1$. A strictly positive regularization parameter λ leads to a quasi-interpolant of order $2m - 1$.

5. Downscaling

This chapter concerns the link between the preceding theory about sampling and the task of downscaling.

The downscaling of an image with a fixed downscaling factor corresponds to sampling with a fixed sampling step. For the original sized discrete image, a sampling step of one can be assumed. According to Shannon, the underlying continuous function can only be reconstructed if the function is band-limited. As seen in the last chapters the band-limited requirement is weakened in the generalized sampling theory, depending on function space, that is generated by the integer translates of a suitable function. The closer the function space is to the underlying image model, the better is the approximation of the image in this space.

According to the mathematical model introduced in Section 1.3, the first step to do is to approximate a continuous representation of the discrete image in a function space. Afterwards, the image can be sub-sampled either at the integer values or at rational values depending on the alignment of the downscaled grid.

These considerations divide the downscaling process into two parts, namely the

1. approximation and the
2. sub-sampling

process. The approximation process is completely defined by the basis function for the chosen function space as seen in eq. (3.2) and its corresponding orthogonal projection in the least square sense. The approximation with B-splines was discussed in detail in the last chapter and it was shown in eq. (4.3) that it can be performed by digital filtering. Owing to the equivalent representation in the frequency domain, stated by the convolution theorem, the orthogonal projection to the space of splines can be computed either in the frequency domain or in the pixel domain. These two different approaches will be presented in the following.

In the medical scenario introduced in Section 1.1, the input images are encoded with JPEG (see Section **??**), where the underlying data consists of 8x8 blocks of DCT coefficients and corresponds to a localized frequency representation of the image. This block tiling, consisting of the DCT coefficients, is in the following called the *DCT domain of the image* while the pixel values of the image correspond to the *pixel domain*. Because the downscaling process should be fast and have a good quality as well, an efficient method should avoid the expensive inverse discrete cosine transform and downscale the image directly on the coefficients. The main drawback using these methods is the localization of the DCT to 8x8 blocks and the possible occurrence of block artifacts.

Problems to address, that are related with downscaling, are the aliasing effects and ringing effects mentioned in section 2.2. Also the details of the image have to be preserved as much

as possible. The aliasing effects can be minimized by smoothing, while the ringing effects depend heavily on the used filter function. For example, the use of the sinc-function introduces ringing effects because of its low decay rate and its infinite support. The details of the image correspond to high frequencies in the Fourier domain. Smoothing cancels higher frequencies, therefore there will be a trade-off between a detail-preserving method and a aliasing-minimizing method.

The next section reviews downscaling in the pixel domain first, followed by the discussion of downscaling methods in the DCT domain. This chapter finally introduces a new hybrid downscaling method in Section 5.3, that combines the advantages of both methods.

5.1. Pixel domain downscaling

The problem of downscaling is already examined during the last chapters for function in one dimension. In this section explicit downscaling schemes that are derived from the B-spline sampling method are discussed for two-dimensional images. The B-splines of degree 0 to 3 are shown in Figure 2.5 on page 26. As these functions are separable the two dimensional functions can be easily computed. The two-dimensional B-spline of degree m is defined for $x, y \in \mathbb{R}$ as the tensor product of the corresponding one-dimensional B-splines:

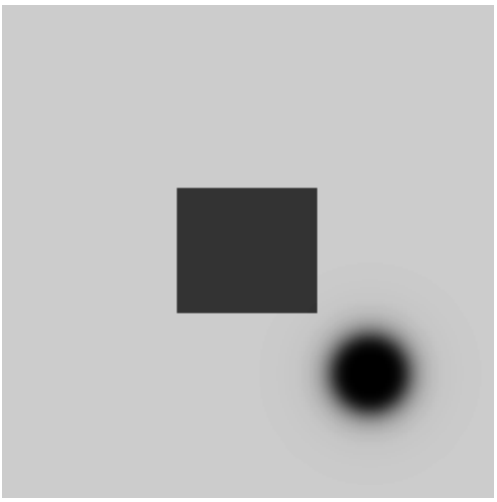
$$\beta_m(x, y) = \beta_m(x) \cdot \beta_m(y). \tag{5.1}$$

As consequence, the approximation of a two-dimensional discrete image reduces to first approximating the rows and subsequently approximating the columns of the image.

The B-spline of zero degree corresponds to the nearest neighbor interpolation, of degree one to the linear interpolation, of degree two to the quadratic interpolation, and finally the B-spline of degree three corresponds to cubic interpolation [28, 37]. These methods will be compared to the sinc-interpolation method recommended by Shannon. All these methods have been well studied in the literature.

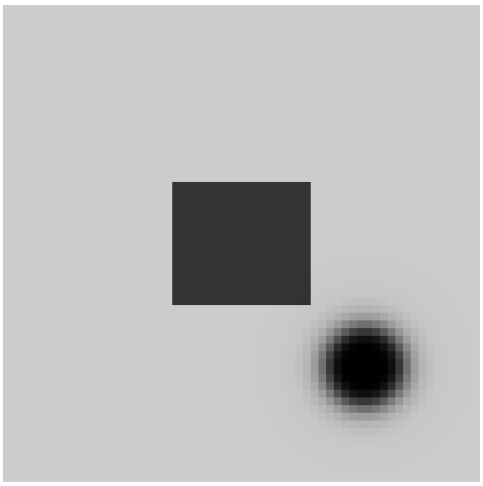
Figure 5.1 illustrates the interpolating projection of the top image into the spaces generated by the previously mentioned basis functions. The sinc-interpolation method has the same drawbacks as mentioned in Chapter 3, as it only approximates functions correctly if they are band-limited. The image 5.1b shows the heavily introduced ringing effects along the edges of the shown box. Although the smooth circle is well approximated. The nearest neighbor method shown in 5.1c exactly approximates the edges, but approximates the smooth circle as discontinuous step function. The linear interpolation (5.1d) leads to a continuous approximation due to the lack of smoothness. An often used compromise between smoothness and the approximation quality is the cubic B-spline approximation, shown in (5.1e), where the second order derivative is continuous as well. The last chapter gives a natural interpretation, as minimizing the second order derivative leads to a smooth function with minimized variation.

Comparing the sinc-approximation with the nearest neighbor approximation approves their characteristics of the theory. While the nearest neighbor method can only reconstruct all step functions, the sinc approximation method can only reconstruct all polynomials of infinite degree, *i. e.,* functions where all derivatives are continuous.



(a) *Original image*

(b) *Sinc*

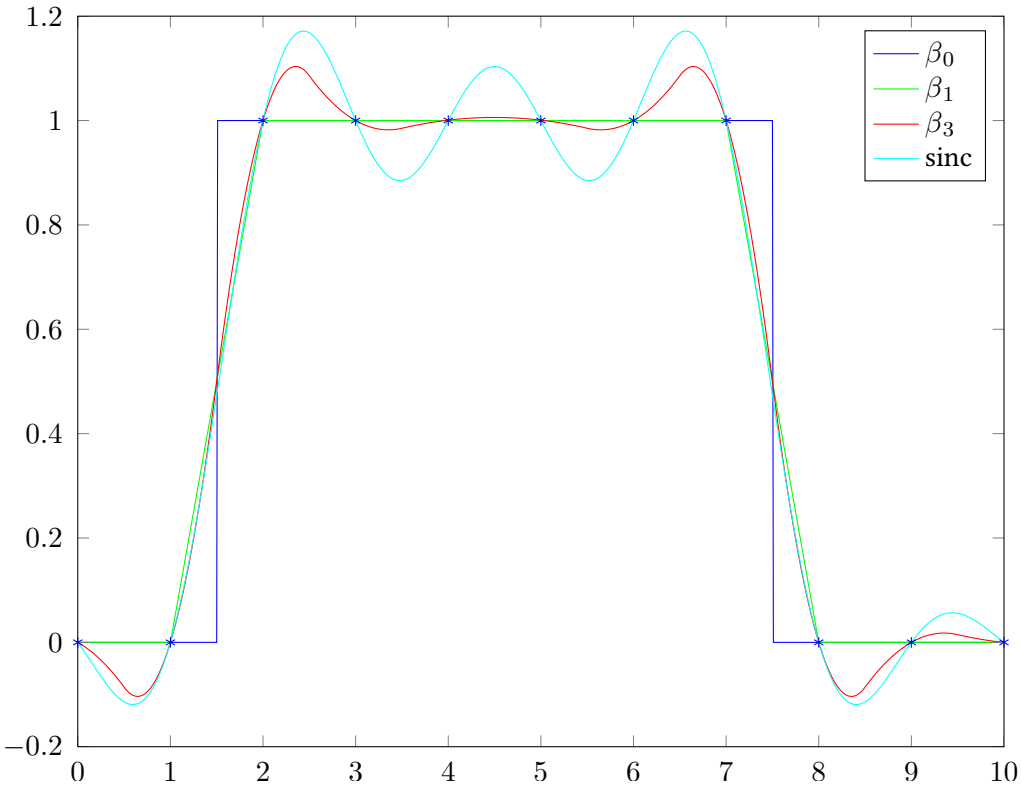


(c) *Nearest neighbor*

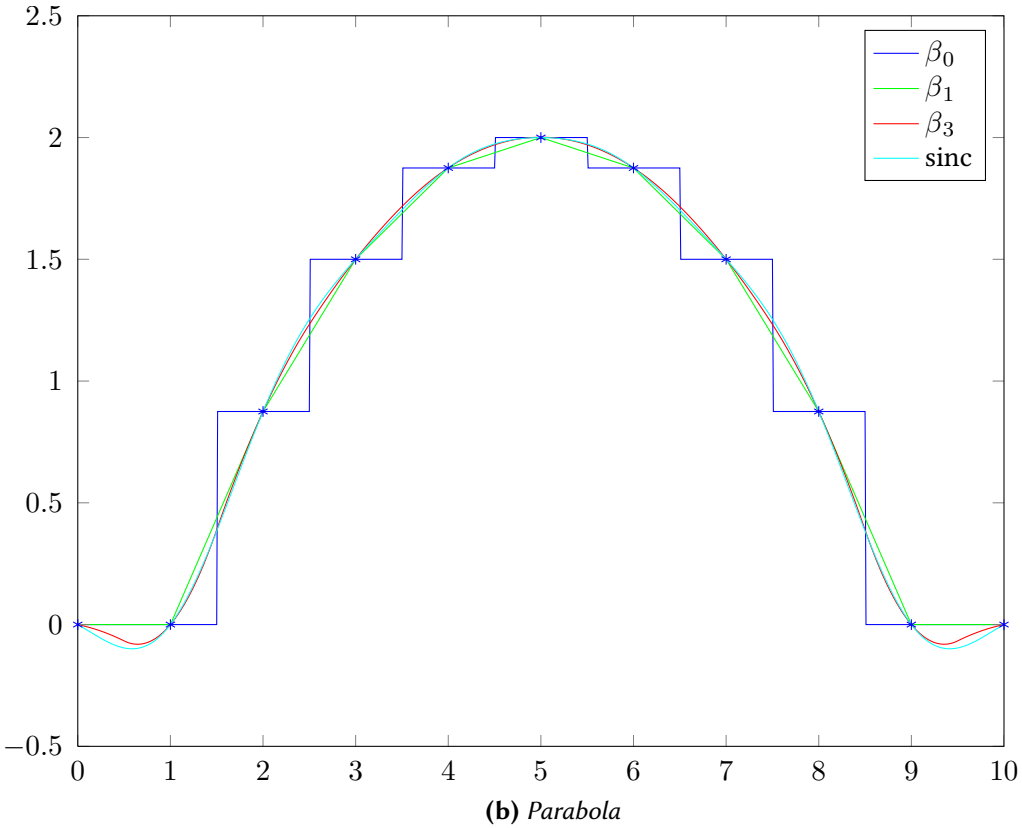
(d) *Linear B-spline*

(e) *Cubic B-spline*

Figure 5.1. *Image projection into different spaces.*

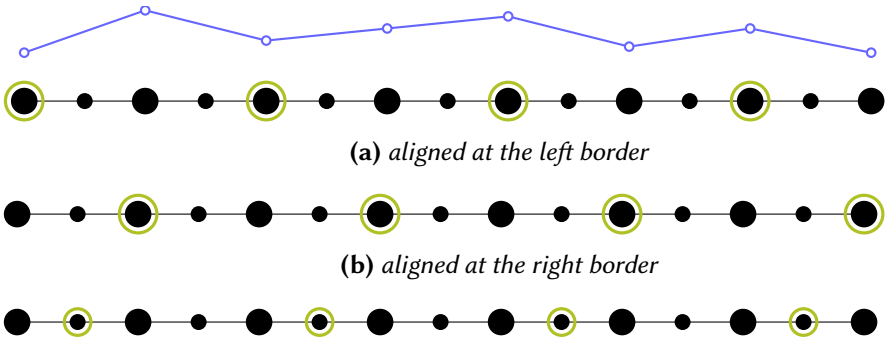


(a) *Rectangle*



(b) *Parabola*

Figure 5.2. *Crosscut of the 2D-Projections in 1D.*
Note that only the sinc*- and the β_3-approximations are smooth.*



(a) *aligned at the left border*

(b) *aligned at the right border*

(c) *aligned at the center*

Figure 5.3. *Options in choosing the coarse grid.*
The big black points are the integer values, while the small ones depict the interpolated floating values. The chosen sample values are the green circles. In this example, the function is reduced to half the size. On the top of the figure, 8 discrete values are interpolated with the linear B-spline. Using the centered grid leads exactly to the linear downscaling algorithm used in the literature.

After approximation the image is downscaled by sub-sampling. There are different options, how to choose the sample values. The interpolated integer values form an equally spaced grid. On this grid, the interpolated function is equal to the original discrete image. Sub-sampling corresponds to an equally spaced coarser grid, that has to be aligned. Figure 5.3 shows three options, how this grid can be aligned in one dimension. For image sub-sampling the first two options are inappropriate, as they choose the integer values as sample values. If these samples are used, the approximation step has no influence to the downscaled image. The grid shown at the bottom illustrates the grid aligned at the center of the line.

The quality of the downscaled image depends on the degree of the chosen B-spline basis. Now the computational complexity of the discussed methods are evaluated. The coefficients of the approximation can be obtained by convolution with the pre-filter $\tilde{\varphi}(-x)$, as shown in (3.2). Second, the approximation is calculated by post-filtering with $\varphi(x)$. Therefore, the complexity only depends on the size of the image and size of the support of the function φ. A B-spline of degree m is compactly supported on the interval $[-(m+1)/2, (m+1)/2]$. Sampling the B-spline at the integer values leads to m discrete values. Thus the size of the pre-filter is equal to the degree of the used B-spline. The computational complexity for filtering an image of size $N \times N$ with a filter of size $m \times m$ is $N^2 \cdot m^2$. The post-filtering process is computed the same way, thus the costs for a B-spline based downscaling method is $2 \cdot N^2 \cdot m^2$. To calculate the costs for the complete downscaling process considered in this thesis, the costs for the JPEG decoding process have to be added. Thevenaz et al. [37] compared several interpolation methods in the context of rotation of an image. One can say, that the linear B-spline interpolation is the most efficient one, and the cubic B-spline reaches the best quality which acceptable computational costs.

5.2. DCT domain downscaling

The basic idea behind the concept of DCT resizing is to calculate the downscaled image without separate JPEG decoding and re-encoding of the image. The special characteristic in the DCT domain is the tiled block structure. It is important to note, that downscaling an image in its Fourier domain results in equivalent methods as in the pixel domain. This follows directly from the last chapter, as the function φ_{int} is specified in the Fourier transform as well. In contrast, this section considers methods that deal with the block tiling used by present-day image and video compression standards like JPEG or H.264/ACV.

Beneficial for computational performance are block sizes that are a power of two, e. g., 8x8 and 16x16, because for these a fast DCT algorithm exists, similar to the fast Fourier transform.

The images of the medical scenario mentioned in Section 1.1 are compressed with JPEG, where the whole image is tiled into 8x8 blocks. Therefore, this section is mainly devoted to the discussion of this special case, but other block sizes can be considered analogous. The JPEG standard is briefly described in Section **??** on page **??**.

Every 8x8 block of the tiled image is transformed with the 2-dimensional DCT to a frequency decomposition. If $X \in \mathbb{R}^{8x8}$ denotes such an 8x8 block, the DCT-coefficients of this block are given as

$$\hat{X} = C_8 X C_8^{\mathrm{T}}, \tag{5.2}$$

and can be inverted by

$$X = C_8^{\mathrm{T}} \hat{X} C_8, \tag{5.3}$$

where C_m is the DCT transform matrix of size $m \times m$ defined in eq. (1.1) on page 7.

After this short review of the DCT used in JPEG, the downscaling problem is considered next. An interpretation of the Shannon theorem proposes a simple rescaling of the DCT basis, as rescaling of the DCT basis corresponds to ideal filtering. In the following this idea will be described in detail.

For a given quadratic image block X of size $B \times B$, the $n \times n$ submatrix $(X_{i,j})_{i=1,\ldots,n;\ j=1,\ldots,n}$ will be denoted with $X_{n\times n}$. As such, the submatrix of DCT coefficients $\hat{X}_{n\times n}$ corresponds to the lowest $n \times n$ frequencies. For $n < B$, a block downscaled to size $n \times n$ can be computed with the scaled inverse $n \times n$ DCT transform

$$\tilde{X} = \frac{n}{B} C_n^{\mathrm{T}} \hat{X}_{n\times n} C_n. \tag{5.4}$$

Clearly, $\tilde{X}$ is of size $n \times n$. The factor $\frac{n}{B}$ is necessary to scale the coefficients according to the smaller basis according to the definition in eq. (1.1). Not that (5.4) is equivalent to filter the pixel block with an ideal filter of size $n \times n$ followed by sampling with a sampling step of n of the filtered $B \times B$ block of pixels.

This concept can be generalized for JPEG images. Every 8x8 block of coefficients will be downscaled according to (5.4), which results in a downscaled image at $n/8 \times n/8$ times the

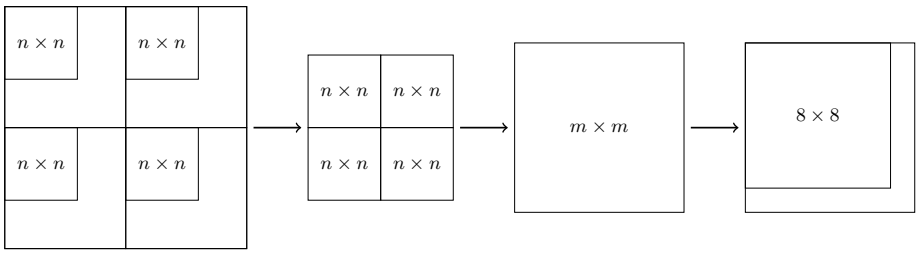


Figure 5.4. *Downscaling method for a downscale factor of n/m.*
First 4 adjacent blocks are downscaled to $n/8$ blocks. Then these 4 small blocks are transformed with a $m \times m$ forward DCT. Finally the $m \times m$ is truncated or zero-padded to an 8x8 block, depending if m is larger or smaller than 8 respectively.

original image size. Then the low resolution image can be again transformed to the DCT domain to obtain a downscaled JPEG image.

Using this concept, a fast algorithm to downsize an image to half the original size, was developed by Dugad and Ahuja [12]. In the paper of Salazar and Tran [31] the downscaling factor is extended to any rational numbers n/m, where $1 \leq n, m \leq 8$. Both papers explicitly include the final forward DCT transform to achieve a downscaled JPEG image.

In summary the resizing algorithm of Salazar and Tran [31] consists of two steps:

- apply a n-point inverse DCT to the 8-point inverse DCT to yield a resize factor of $n/8$
- apply a m-point forward DCT and retain only the lowest 8 coefficients to obtain a scaling factor of $8/m$.

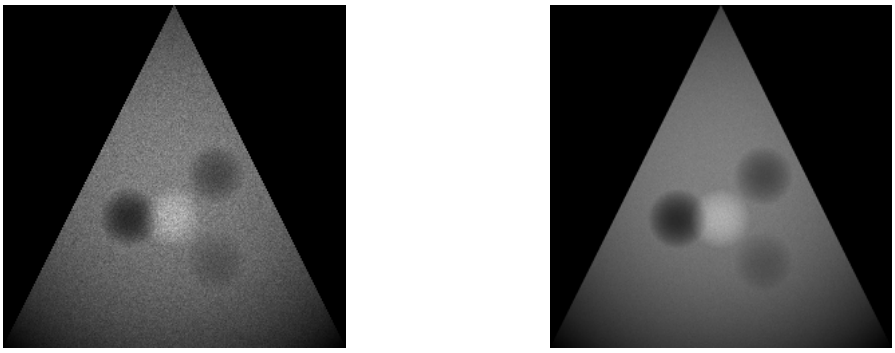
The image is now scaled by $(n/8) \cdot (8/m) = n/m$ and the result are 8x8 blocks of DCT-coefficients. This algorithm is illustrated in Figure 5.4.

The following example demonstrates the possibilities of this algorithm, because the representation of n/m is not unique for most of the scaling factors:

Example 1 A downscaled JPEG image by a factor of 0.5 can be achieved with an

1. inverse 3x3 transform and forward 6x6 transform and 8x8 zero padding,
2. inverse 4x4 transform and forward 8x8 transform,
3. inverse 5x5 transform and forward 10x10 transform and 8x8 truncation, or an
4. inverse 6x6 transform and forward 12x12 transform and 8x8 truncation.

The inverse transform, as first executed, influences the cut-off of the high frequencies. The more frequencies can be preserved in this step, the better the resulting image quality will be. Of course, the use of higher transformation sizes contains more multiplications and additions than using a small transformation size.



(a) *Reference Image* **(b)** *Downscaled in the DCT domain*

Figure 5.5. *Sub-sampling with a factor of 4 (2x2 block of the 8x8 DCT-coefficients).*

Because the strong smoothing of the image, e. g., shown in Figure 5.5, and the introduction of aliasing and ringing effects by the use of ideal filtering [12], this method is suboptimal for achieving a good quality, but optimal to reduce the computational complexity. Also block artifacts are noted in the image resizing experiments section in [31]. These results all conform, with the theory of Chapter 3, where these kinds of methods are seen as approximation in the space of band-limited functions and as such have the same drawbacks.

Other methods [21, 24] try to implement pixel domain based interpolation schemes in the DCT domain. These methods still suffer from the drawbacks of the block structure but achieve a better quality with increased computational complexity depending on the chosen interpolation method, due to the non-sparsity of the interpolation matrices. The ideal filter instead has a very sparse representation.

The computational complexity of computing a $B \times B$ two-dimensional DCT is $\mathcal{O}(B^2 \log B)$. The computational costs for the whole $N \times N$ image are $(N/B)^2 B^2 \log B = N^2 \log B$. The downscaling methods discussed in this section reduce the $B \times B$ DCT to a $n \times n$ DCT, where $1 \leq n \leq B$, which corresponds to a downscaling factor of n/B. In brief, the computational costs are reduced to $N^2 \log n$ for the complete JPEG decoding and downscaling process.

5.3. Hybrid downscaling

In the last sections downscaling methods in the pixel domain as well as in the DCT domain were discussed. Both methods are not efficient in the sense of image quality and the computational complexity. Methods operating in the pixel domain provide a good image quality due to high computational costs, while DCT domain methods achieve lower image qualities with low complexity.

This section combines the low complexity of the DCT downscaling methods with the good approximation quality of the downscaling methods in the pixel domain. The remainder of this section considers the scaling of an image of size $N \times N$ by some factor of $\alpha < 1$. As in the last section, the block size of the DCT tiling is denoted with B.

This approach is separated into two steps, first

1. downscaling in the DCT domain to n/B of the original size, where $1 \leq n \leq B$, and then
2. downscaling in the pixel domain to achieve the final image size.

Thus, the scaling factor α can be factorized to $\alpha = n/B \cdot \tilde{\alpha}$.

The first step discards all frequencies higher than n. For $n = B$, the pure pixel domain method is used. Therefore, the parameter n acts like a quality parameter, where lower values lead to lower image quality but decreas the computational complexity. The second step is completely determined by the approximation method and smoothes the blocking artifacts introduces by block based DCT downscaling method.

The computational costs for the hybrid method are $N^2 \cdot (\log(n) + 2(n/B)^2 m^2)$, where the pixel domain method uses the B-spline of degree m.

Figure 5.6 compares the previously discussed downscaling methods with the hybrid method for a downscaling factor of 4 and a cubic B-spline based approximation method. The complexity is reduced to more than a third of the costs of the pure pixel based method. Furthermore, the next chapter shows that this method achieves comparable results. In brief, the hybrid method clearly outperforms the other methods in terms of efficiency.

The last chapter allowed an interpretation for this hybrid downscaling method. It was shown, that the B-spline projection behaves like a low-pass filter with the same characteristics as the Butterworth filter. To simplify this discussion, a downscaling factor of 4 is assumed and futher $n = 2$ is chosen. In step half of all local frequencies are set to zero. The following B-spline projection further removes half of the global frequencies. The corresponding pure pixel domain based method, interpreted as Butterworth filter, rejects approximately a quarter of all global frequencies. Therefore, the DCT based method discards almost only those frequencies, that are nevertheless discarded in the second step.

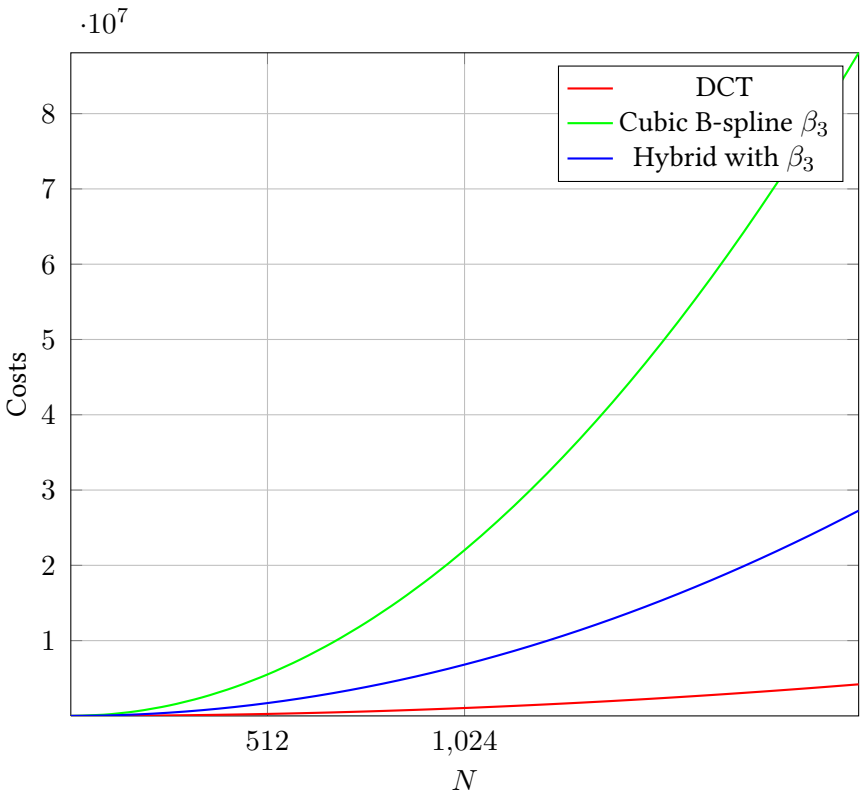


Figure 5.6. *Computational complexity of the downscaling methods.*

6. Numerical Experiments

This chapter evaluates the efficiency of the downscaling methods that are presented in the last chapter.

To compare the quality of images, a reference image is necessary. In most image processing problems, this image is equal to the original image (e. g., clockwise rotation, compression, denoising, ...). In the case of downscaling, no reference image is available. There are two reasonable methods to measure the quality of the downscaled image. One method uses an upscaling of the downscaled image to compare it with the original image. This method is not used in this thesis, as the upscaling process additionally influences the measurement and depends strongly on the used approximation method. Another method uses a reference image, that is created by a common high quality downscaling method. This thesis uses the downscaled images, that result from the projection to spline spaces. This is reasonable for this thesis, as the B-splines have excellent approximation properties and are often proposed for similar problems [20, 25]. Most often the cubic B-spline function is chosen due to its smoothness and high approximation order.

Next, the question arises, how the downscaled image is compared to the reference image. In the mathematical model of Section 1.3, the approximation of the image minimizes the L^2-error. Therefore, the L^2-error leads naturally to an objective quality measure. In the literature, the so called peak-signal-to-noise ratio (PSNR) is used to measure the image quality in dB. This measure combines the mean-square error in a logarithmic formulation. The PSNR of two discrete images $F, G \in \mathbb{N}^{M\times N}$ of size $M \times N$ with intensity value between 0 and 255 (8bit) is defined as

$$\mathrm{PSNR}(F,G) = 10 \cdot \log_{10}\left(\frac{255^2}{\mathrm{MSE}(F,G)}\right), \tag{6.1}$$

where the mean-square error of two images is calculated by

$$\mathrm{MSE}(F,G) = \frac{normF - G^2}{MN}\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|F(i,j) - G(i,j)|^2 . \tag{6.2}$$

The smaller the mean-square error is, the higher the PSNR-measure will get. With the PSNR-measure, the downscaling methods can be objectively compared.

As the pure pixel based downscaling methods are too slow to use in the medical scenario described in Section 1.1, they will serve here as reference images. In the following, the cubic B-spline will be considered as reference method. As required in the scenario introduced in Section 1.1, an image has to be scaled down to a quarter of the original size. Therefore, the hybrid method will reduce the size to a half of the original size with the DCT domain method,

Image	Method	PSNR	Time
Lena	DCT	42.72	1.04s
	Hybrid	61.43	1.06s
Ultrasound	DCT	53.12	4.21s
	Hybrid	74.73	4.33s
Ultrasound with speckle noise	DCT	49.08	4.19s
	Hybrid	63.56	4.23s

Table 6.1. *Comparison of downscaling methods for a DCT block size of 8.*

and further downscale this to the attainable size with the cubic B-spline downscaling method. The DCT domain algorithm uses the ideal filter described in Section 5.2.

The complexity of these methods was already compared in Section 5.3. It was pointed out, that the hybrid scheme significantly reduces the computational complexity. Next it will be shown that it almost achieves the same image quality.

Therefore, the quality of the downscaled images is measured and the results of the DCT based method are compared to the results of the hybrid downscaling scheme. Evaluated is the use of different block sizes of the DCT tiling, namely 8, 16, and 32. Three representing images are used for the evaluation. The classical Lena image is used often in the literature. The other two images are created synthetically. The first shows some smooth polynomials and the second is in addition corrupted by speckle noise, as in the case of real-world ultrasound images.

Table 6.1 shows the results for a DCT block size of 8, as used in the JPEG standard. The downscaled images for a block size of 8 are given in Figure 6.1, Figure 6.2, and Figure 6.3. The results for the other block sizes are stated in Table 6.2 and Table 6.3.

The results state clearly, that the hybrid scheme is much closer to the chosen reference method as the pure DCT based method for any block size. Besides, the quality of the DCT method decreases with increased block size. This can be interpreted by the introduced ringing effects. In contrast, the quality of the hybrid schemes takes profit of the increasing block size, as the localized frequency representation tends to a more global one.

Combined with the reduced complexity, the hybrid scheme outperforms the DCT method in the sense of efficiency. Furthermore, the hybrid method is scalable. It includes the DCT based method as special case, as well as the pure pixel domain downscaling method.

Image	Method	PSNR	Time
Lena	DCT	40.50	0.28s
	Hybrid	77.12	0.28s
Ultrasound	DCT	52.27	1.13s
	Hybrid	74.73	1.15s
Ultrasound with speckle noise	DCT	49.09	1.13s
	Hybrid	66.70	1.16s

Table 6.2. *Comparison of downscaling methods for a DCT block size of 16.*

Image	Method	PSNR	Time
Lena	DCT	39.22	0.08s
	Hybrid	79.41	0.08s
Ultrasound	DCT	51.36	0.30s
	Hybrid	74.73	0.33s
Ultrasound with speckle noise	DCT	48.79	0.32s
	Hybrid	69.67	0.33s

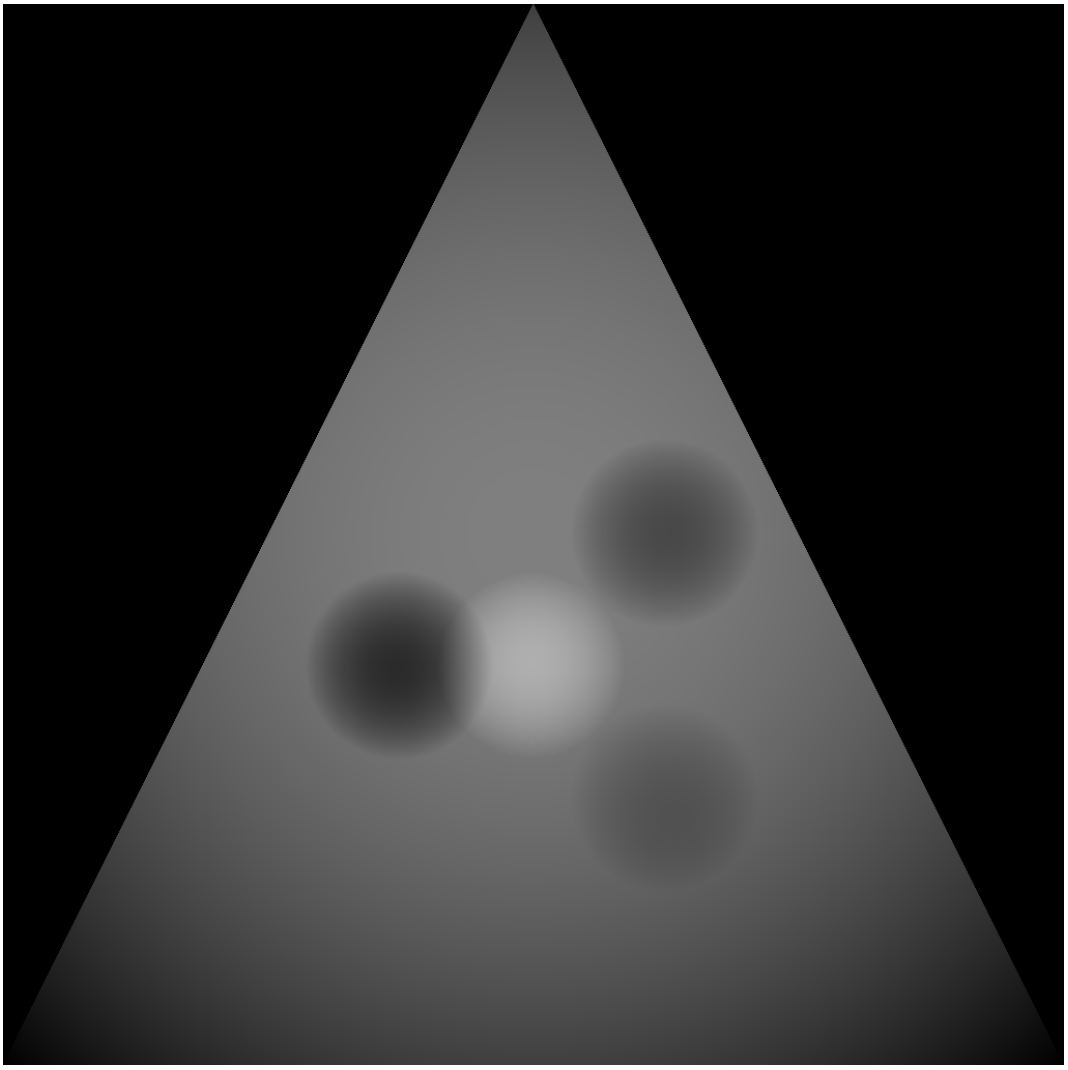
Table 6.3. *Comparison of downscaling methods for a DCT block size of 32.*

(a) *Original*

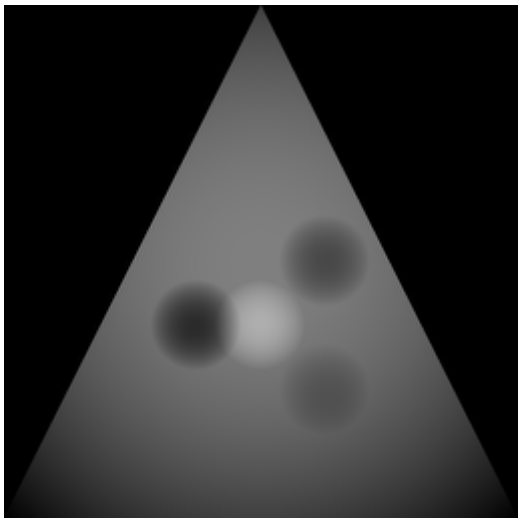
(b) *DCT*

(c) *Hybrid*

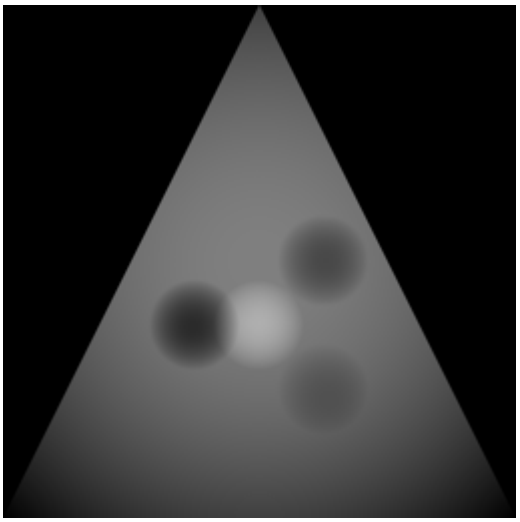
Figure 6.1. *Downscaling of the Lena image.*



(a) *Original*

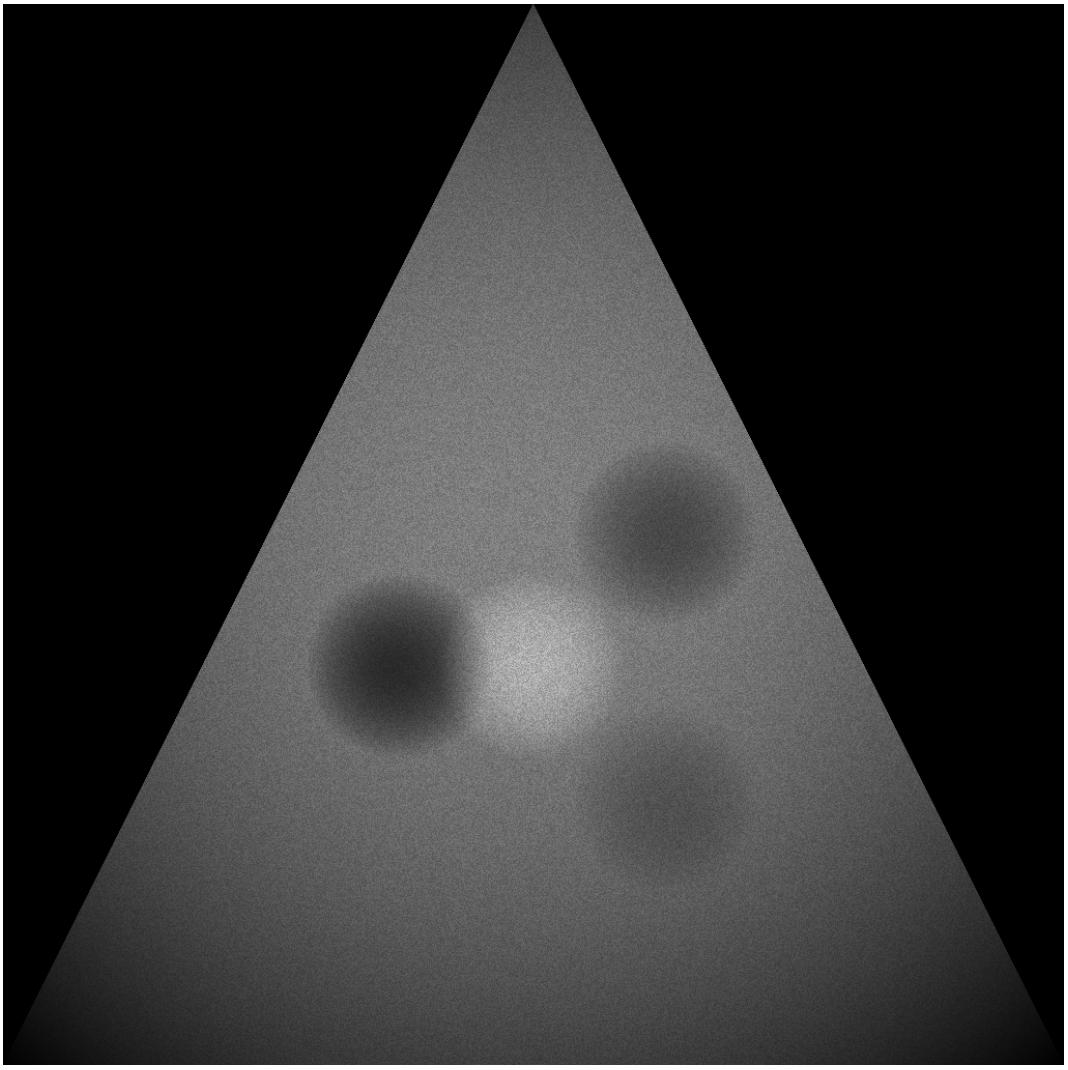


(b) *DCT*

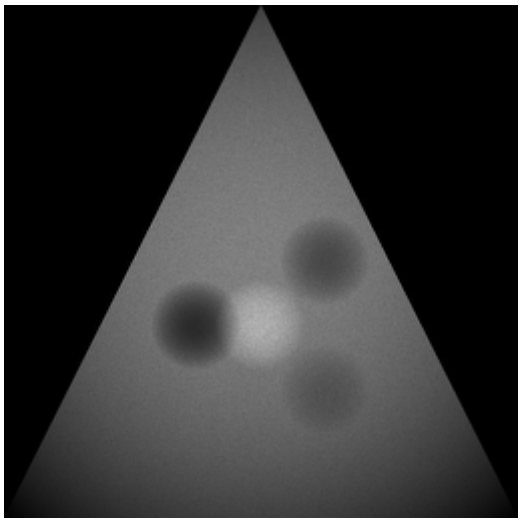


(c) *Hybrid*

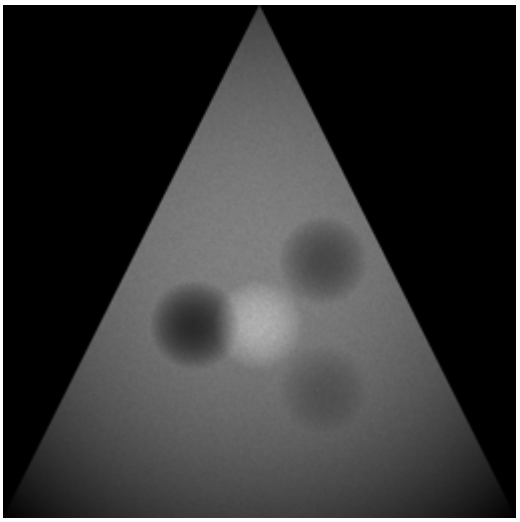
Figure 6.2. *Downscaling of the ultrasound image.*



(a) *Original*



(b) *DCT*



(c) *Hybrid*

Figure 6.3. *Downscaling of the ultrasound image with speckle noise.*

7. Summary

This thesis investigated efficient downscaling methods for compressed JPEG images and introduced a new hybrid method. The results of the last chapter showed, that while the complexity is reduced it almost reaches the same quality as the pure pixel based methods. To be clear, for every pixel based downscaling method the hybrid method achieves almost the same quality, but descreases the complexity. With this new hybrid method, the tasks of the server, discussed in Section 1.1, can be solved in an efficient manner. The design of the hybrid method is scalable in the sense of complexity and image quality, and contains a pure DCT domain downscaling method as well as the pure pixel domain downscaling. In addition, this approach is able to downscale images with arbitrary downscaling factors.

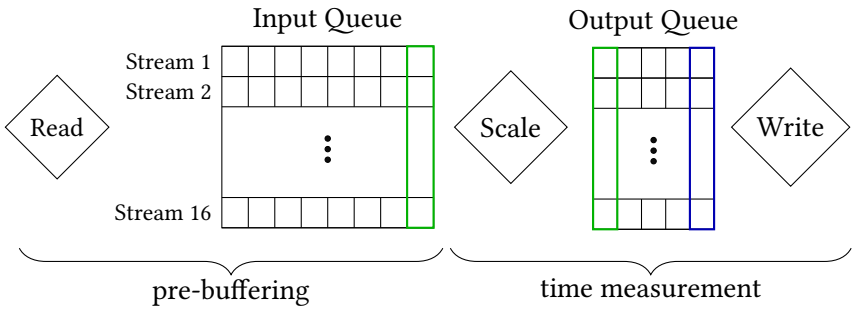
The mathematical tools to consider the downsizing problem in the pixel domain as well as in the DCT domain were provided in this thesis. The limits of the classical Shannon theorem were identified and recent enhances of modern sampling theory has been established. Interpolation and approximation methods related to B-splines have been discussed exclusively Chapter 4 due to their excellent approximation properties. For a better understanding of approximation with B-splines, an interesting connection to the Butterworth filter was shown. Methods that downscale an image were discussed in Chapter 5. On the one hand pixel domain downscaling methods can achieve high image qualities, and on the other hand DCT domain downscaling methods have low computational complexity. The hybrid method combines the advantages of both methods to a new efficient downscaling method.

Further improvements might consider adaptive block methods. Furthermore, the use of wavelet based compression methods like JPEG2000 can be examined in relation to the downscaling of images. To further improve the results for the ultrasound images in the medical scenario, the noise model of ultrasound images can be considered. The work of Dutt and Greenleaf [13], Hyeona Lim [14], Krissian et al. [17] modeles the noise by the Raleigh distribution and might serve as a good starting point for futher research.

Appendices

A. Implementation Details

- Mutlti-threaded application
- Number of thread of downscaling and encoding are scalable
- Extendable downscaling code (UML class diagram)
- point out bottlenecks



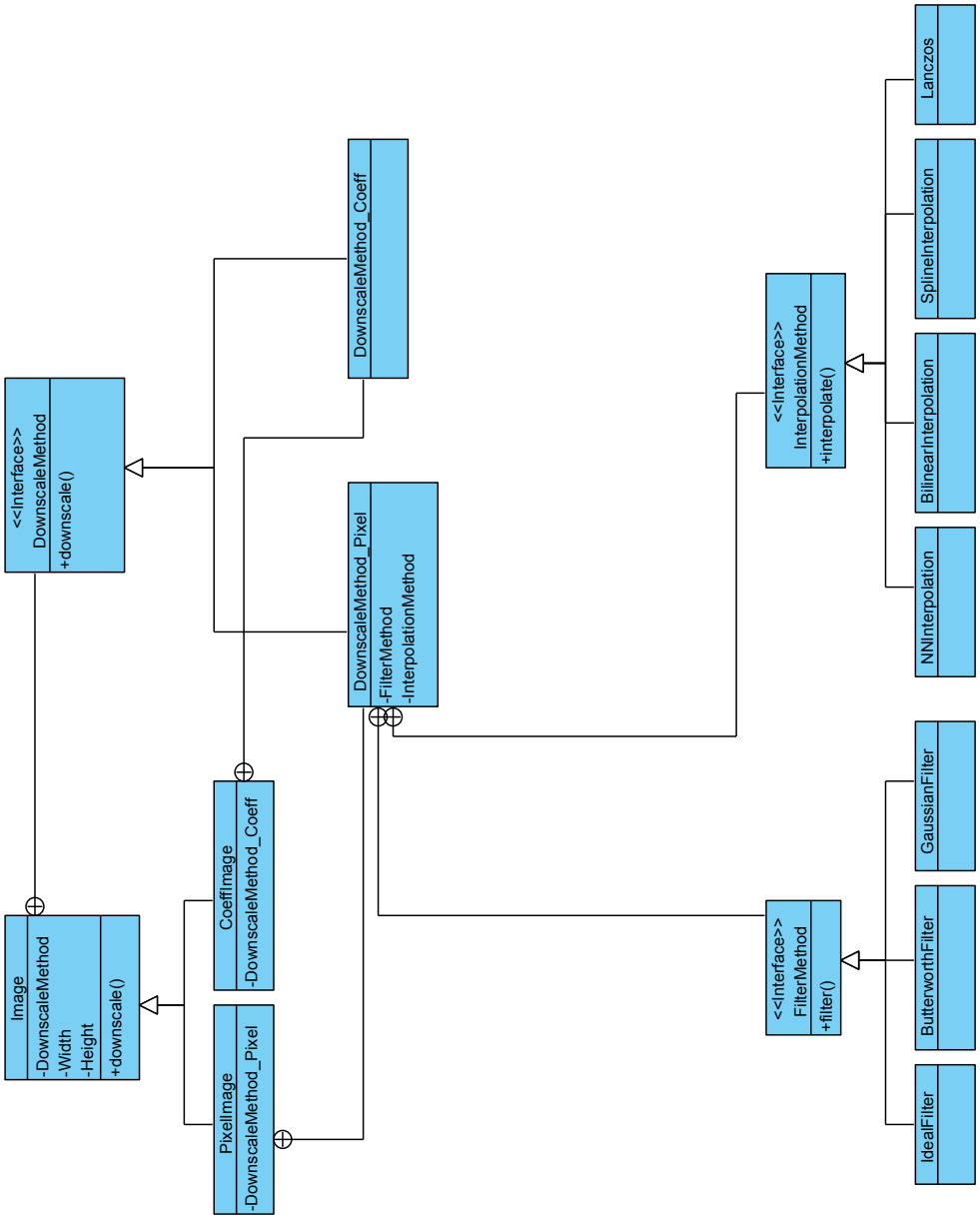


Figure A.1. *UML model of the downscaling.*

B. M-Code

```
%% approximation 1d: parabola
x = 0:0.01:10;
y = [0  -(1/8)*([0:8] - 4).^2 + 2  0];

hold on;
plot(x, bspline_projection(y, x, 0), 'b');
plot(x, bspline_projection(y, x, 1), 'g');
plot(x, bspline_projection(y, x, 2), 'r');
plot(x, sinc_projection(y, x), 'c');
plot(0:10, y, '*');
hold off
hleg1 = legend('\beta_0','\beta_1', '\beta_3', 'sinc');
matlab2tikz('../img/tikz/approximation1d_parabola.tikz');

%% approximation 1d: step
x = 0:0.01:10;
y = [0 0 ones(1, 6) 0 0 0];

hold on;
plot(x, bspline_projection(y, x, 0), 'b');
plot(x, bspline_projection(y, x, 1), 'g');
plot(x, bspline_projection(y, x, 2), 'r');
plot(x, sinc_projection(y, x), 'c')
plot(0:10, y, '*');
hold off
hleg1 = legend('\beta_0','\beta_1', '\beta_3', 'sinc');
matlab2tikz('../img/tikz/approximation1d_step.tikz');

%% sinc vs. phi_int

x = -6:0.01:6;

clf;
plot(x, phi_int(x, 0), 'k');
matlab2tikz('../img/tikz/sinc_vs_phi_0.tikz');

clf;
plot(x, phi_int(x, 1), 'k');
matlab2tikz('../img/tikz/sinc_vs_phi_1.tikz');

clf;
plot(x, phi_int(x, 3), 'k')
matlab2tikz('../img/tikz/sinc_vs_phi_3.tikz');

clf;
plot(x, sinc(x), 'k');
```

```
47 matlab2tikz('../img/tikz/sinc_vs_phi_sinc.tikz');
48
49 clf;
50 plot(x, phi_int(x, 5), 'k')
51 matlab2tikz('../img/tikz/sinc_vs_phi_5.tikz');
52
53 clf;
54 plot(x, phi_int(x, 8), 'k')
55 matlab2tikz('../img/tikz/sinc_vs_phi_8.tikz');
56
57 %% test 2d
58 img = im2double(imread('lena512.pgm'));
59 img = imresize(img, 1/8);
60 [M,N] = size(img);
61
62 x = 0:0.5:N-1;
63 y = 0:0.5:M-1;
64 tmp = zeros(M,length(x));
65 int = zeros(length(y),length(x));
66
67 for i = 1:M
68     tmp(i,:) = bspline_projection(img(i,:), x, 2)';
69 end
70 for i = 1:length(x)
71     int(:,i) = bspline_projection(tmp(:,i), y, 2);
72 end
73 imshow(int)
74 %for i = 1:M
75 %    int(:,i)bspline_projection(img(i,:), x, 3);
76 %end
77
78 %% approximation vs interpolation 1d: step
79 x = 0:0.1:7;
80 %y = [0 0 ones(1, 35) 0 0 0];
81 %y = ones(41,1);
82 %y = [-(1/8)*([-5:19] - 4).^2 + 2];
83
84 y = [zeros(1, 3) ones(1, 3) 2*ones(1, 2)];
85
86 %y = [1:11];
87
88 hold on;
89 %plot(x, bspline_projection(y, x, 0), 'b');
90 plot(x, bspline_projection(y, x, 1), 'g');
91 %plot(x, bspline_projection(y, x, 3), 'r');
92 plot(x, bspline_approximation(y, x, 0), 'g');
93 plot(x, bspline_approximation(y, x, 1), 'b');
94 plot(x, sinc_projection(y, x), 'c')
95 plot(0:7, y, '*');
96 hold off
97 %hleg1 = legend('\beta_0','\beta_1', '\beta_3', 'sinc');
98 %matlab2tikz('../img/tikz/approximation1d_step.tikz');
```

```
1 function y = bsplineN (x, n)
2
```

```
if n == 0
    y = zeros(size(x));
    idx = find(x>-1/2 & x<=1/2) ;
    y(idx) = 1;
elseif n == 1
    y = zeros(size(x));
    idx = find(x>=-1 & x<=1) ;
    y(idx) = 1 - abs(x(idx));
else
   a = (n+1)/2;
   y = (((x + a)/(n)).*bsplineN(x+0.5,n-1)) + (((a-x)/(n)).*bsplineN (x-0.5,n-1));
end
```

```
function y = bspline_dual(x, m)
    dim = size(x);
    x = x(:);
    N = length(x);
    t = linspace(-(2*m+2),(2*m+2), 2*(2*m+2)+1)';
    dual = real(ifft(1./fft(bsplineN(t,2*m+1)')))';
    phi = bsplineN(x(:,ones(1, 2*(2*m+2)+1)) - t(:,ones(size(x)))', m);
    y = dual(:,ones(1,N))' * circshift(phi, [0 1])';
    y = y(1,:);
    y = reshape(y, dim);
end
```

```
%%
function y = phi_int(x, m)
    % x values
    % m order of b-spline
    dim = size(x);
    x = x(:);
    N = length(x);
    t = linspace(-(m+1),(m+1), 2*(m+1)+1)';
    bspline = bsplineN(t,m)';
    bspline_dual = real(ifft(1./fft(bspline)))';
    phi = bsplineN(x(:,ones(1, 2*(m+1)+1)) - t(:,ones(size(x)))', m);

    y = bspline_dual(:,ones(1,N))' * circshift(phi, [0 1])';
    y = y(1,:);
    y = reshape(y, dim);
end
```

```
function H = bw(M, N, w0, n)
    [u, v] = meshgrid([0:(M-1)], [0:(N-1)]);

    D = u.^2 + v.^2;
    D = max(u,v).^2;

    % create butterworth filter with
    %   w0 cutoff frequency
    %   n  order
```

```
10      H = 1./(1.0 + (D./(w0^2)).^n);
11
12      %G = H.*F;
13      %g = abs(ifft2(ifftshift(G)));
14  end
```

List of Symbols

$\mathbb{R}$ The real numbers.

$\mathbb{N}$ The natural number: $1, 2, 3, \ldots$.

$\mathbb{Z}$ The integers.

$\mathbb{Q}$ The rational numbers.

$\mathbb{C}$ The complex numbers.

$D^m f$ The m'th order derivative of the function f.

$\mathcal{F}f$ The Fourier transform for $f \in L^2(\mathbb{R})$.

$\hat{f}$ The Fourier transform for $f \in L^1\mathbb{R}$.

χ_A The indicator function for the set A, *i. e.,* $\chi_A(x) = 1$ for all $x \in A$, and $\chi_A(x) = 0$ otherwise.

$\overline{A}$ The closure of the the set A.

span A The linear span of the the set A: $\operatorname{span} A = \left\{\sum_{k\in\mathbb{Z}} c_k a \,:\, a \in A,\ c = (c_k)_{k\in\mathbb{Z}} \in l^p(\mathbb{Z})\right\}$.

supp f The support of the function f: $\operatorname{supp} f = \overline{\{x \in \mathbb{R} \,:\, f(x) \neq 0\}}$.

$\delta_{k,l}$ The Kronecker delta, $\delta_{k,l} = 0$ if $k = l$ and $\delta_{k,l} = 1$ otherwise.

T_k The translation operator $(T_k f)(x) = f(x - k)$.

D_α The dilation operator $(D_\alpha f)(x) = f(x/\alpha)$.

M_b The modulation operator $(M_b f)(x) = \mathrm{e}^{2\pi i b\cdot x}$.

U The synthesis operator.

S The frame operator.

References

[1] A. Aldroubi, M. Unser, and M. Eden. Cardinal spline filters: Stability and convergence to the ideal sinc interpolator. *Signal Processing*, 28(2):127–138, 1992. (Cited on pages 32 and 34.)

[2] T. Blu, P. Thevenaz, and M. Unser. Minimum support interpolators with optimum approximation properties. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 242–245. IEEE, 1998. (Cited on page 24.)

[3] T. Blu, P. Thcvenaz, and M. Unser. Moms: maximal-order interpolation of minimal support. *IEEE Transactions on Image Processing*, 10(7):1069–1080, 2001. (Cited on page 11.)

[4] Carl De Boor and Robert E. Lynch. On splines and their minimum properties. *J. Math. Mech*, 15:953–969, 1966. (Cited on pages 24 and 25.)

[5] S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7:536–541, 1930. (Cited on page 34.)

[6] O. Christensen. *An introduction to frames and Riesz bases*. Applied and numerical harmonic analysis. Birkhäuser, 2003. (Cited on page 18.)

[7] H.B. Curry and I.J. Schoenberg. On pólya frequency functions iv: the fundamental spline functions and their limits. *Journal d'analyse mathématique*, 17(1):71–107, 1966. (Cited on page 24.)

[8] C. De Boor. Splines as linear combinations of b-splines. a survey. 1976. (Cited on pages 11 and 24.)

[9] C. De Boor. *B (asic)-spline basics*. Mathematics Research Center, University of Wisconsin-Madison, 1986. (Cited on page 11.)

[10] C. De Boor. *A practical guide to splines*, volume 27. Springer Verlag, 2001. (Cited on page 32.)

[11] R.L. de Queiroz. Processing jpeg-compressed images and documents. *Image Processing, IEEE Transactions on*, 7(12):1661–1672, 1998. (Cited on page 10.)

[12] R. Dugad and N. Ahuja. A fast scheme for image size change in the compressed domain. *IEEE Transactions on Circuits and Systems For Video Technology*, 11(4):461–474, 2001. (Cited on pages 10, 45, and 46.)

[13] V. Dutt and J.F. Greenleaf. Adaptive speckle reduction filter for log-compressed b-scan images. *Medical Imaging, IEEE Transactions on*, 15(6):802 –813, dec 1996. (Cited on page 55.)

[14] Thomas Neil Williams Hyeona Lim. A non-standard anisotropic diffusion for speckle noise removal, 2007. (Cited on page 55.)

[15] ITU. *Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines (ITU-T Recommendation T.81)*, 1992. (Cited on page 5.)

[16] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981. (Cited on page 11.)

[17] Karl Krissian, Kirby Vosburgh, Ron Kikinis, and Carl-Fredrik Westin. Anisotropic diffusion of ultrasound constrained by speckle noise model, 2004. (Cited on page 55.)

[18] R. Lasser. *Introduction to Fourier series*. Monographs and textbooks in pure and applied mathematics. M. Dekker, 1996. (Cited on page 12.)

[19] C. Lee, M. Eden, and M. Unser. High-quality image resizing using oblique projection operators. *Image Processing, IEEE Transactions on*, 7(5):679–692, 1998. (Cited on page 11.)

[20] C. Lee, M. Eden, and M. Unser. High-quality image resizing using oblique projection operators. *IEEE Transactions on Image Processing*, 7(5):679–692, May 1998. (Cited on page 49.)

[21] Jae-Beom Lee and A. Eleftheriadis. 2-d transform-domain resolution translation. *IEEE T Circ Syst Vid*, 10(5):704–714, 2000. (Cited on pages 10, 11, and 46.)

[22] S.G. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999. (Cited on pages 9, 11, and 12.)

[23] S. A. Martucci. Image resizing in the discrete cosine transform domain. In *Proc. Conf. Int Image Processing*, volume 2, pages 244–247, 1995. (Cited on page 10.)

[24] N. Merhav and V. Bhaskaran. Fast algorithms for dct-domain image downsampling and for inverse motion compensation. *IEEE T Circ Syst Vid*, 7(3):468–476, 1997. (Cited on pages 10, 11, and 46.)

[25] A. Muñoz Barrutia, T. Blu, and M. Unser. Least-squares image resizing using finite differences. *IEEE Transactions on Image Processing*, 10(9):1365–1378, September 2001. (Cited on page 49.)

[26] HyunWook Park, YoungSeo Park, and Seung-Kyun Oh. L/m-fold image resizing in block-dct domain using symmetric convolution. *IEEE T Image Process*, 12(9):1016–1034, 2003. (Cited on page 10.)

[27] Young Seo Park and Hyun Wook Park. Arbitrary-ratio image resizing using fast dct of composite length for dct-based transcoder. *IEEE T Image Process*, 15(2):494–500, 2006. (Cited on page 10.)

[28] J. Anthony Parker, Robert V. Kenyon, and Donald E. Troxel. Comparison of interpolating methods for image resampling. *IEEE T Med Imaging*, 2(1):31–39, 1983. (Cited on pages 11 and 40.)

[29] J. Ramanathan. *Methods of applied Fourier analysis*. Birkhauser, 1998. (Cited on page 12.)

[30] C.H. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 10(3):177–183, 1967. (Cited on page 37.)

[31] C. Salazar and T. D. Tran. A complexity scalable universal dct domain image resizing algorithm. *IEEE Transactions on Circuits and Systems For Video Technology*, 17(4):495–499, 2007. (Cited on pages 10, 45, and 46.)

[32] I.J. Schoenberg. Spline functions and the problem of graduation. *Proceedings of the National Academy of Sciences of the United States of America*, 52(4):947, 1964. (Cited on page 37.)

[33] I.J. Schoenberg. *Cardinal spline interpolation*. Number 12. Society for Industrial and Applied Mathematics, 1973. (Cited on page 35.)

[34] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1): 10–21, 1949. (Cited on pages 11 and 15.)

[35] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994. (Cited on page 27.)

[36] G. Strang and G. Fix. A fourier analysis of the finite element variational method. *Constructive aspects of functional analysis*, pages 796–830, 1971. (Cited on page 25.)

[37] P. Thevenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE T. Med. Imaging.*, 19(7): 739–758, 2000. (Cited on pages 11, 30, 40, and 43.)

[38] M. Unser. Splines: a perfect fit for signal and image processing. *IEEE Signal. Proc. Mag.*, 16(6):22–38, 1999. (Cited on page 11.)

[39] M. Unser. Sampling-50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000. (Cited on pages 11 and 28.)

[40] M. Unser and T. Blu. Generalized smoothing splines and the optimal discretization of the wiener filter. *Signal Processing, IEEE Transactions on*, 53(6):2146–2159, 2005. (Cited on page 35.)

[41] M. Unser and T. Blu. Self-similarity: Part I—Splines and operators. *IEEE Transactions on Signal Processing*, 55(4):1352–1363, April 2007. (Cited on pages 34 and 37.)

[42] M. Unser, A. Aldroubi, and M. Eden. Fast b-spline transforms for continuous image representation and interpolation. *IEEE T. Pattern. Anal.*, 13(3):277–285, 1991. (Cited on page 11.)

[43] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. i. theory. *IEEE T. Signal. Proces.*, 41(2):821–833, 1993. (Cited on page 11.)

[44] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. ii. efficiency design and applications. *IEEE T. Signal. Proces.*, 41(2):834–848, 1993. (Cited on page 11.)

[45] G.K. Wallace. The jpeg still picture compression standard. *Consumer Electronics, IEEE Transactions on*, 38(1):xviii–xxxiv, 1992. (Cited on page 5.)