Master Thesis

Describing Non-regular Sampling and Frequency Selective Reconstruction within the Compressed Sensing Framework

Veit Pirkl

Supervisor: Dr.-Ing. Jürgen Seiler

Erlangen, April 2017
Masterarbeit

für

Veit Pirkl

Beschreibung der nichtregelmäßigen Abtastung und frequenz-selektiven Rekonstruktion im Compressed Sensing Framework

Describing Non-regular Sampling and Frequency Selective Reconstruction within the Compressed Sensing Framework


Im Rahmen der Arbeit soll eine Literaturrecherche zur Bestimmung des Standes der Technik durchgeführt werden. Weiterhin wird auf saubere Dokumentation der Arbeit und des Quellcodes besonderer Wert gelegt.

Abgabe: 23.04.2017

(Prof. Dr.-Ing. A. Kaup)
Declaration

I confirm that I have written this thesis unaided and without using sources other than those listed and that this thesis has never been submitted to another examination authority and accepted as part of an examination achievement, neither in this form nor in a similar form. All content that was taken from a third party either verbatim or in substance has been acknowledged as such.

Place, Date

Signature
Abstract

Processing images and videos digitally requires them to be available in a sampled and quantized form. Concerning the sampling, usually a regular sampling pattern is utilized, which leads to a restriction of the spatial resolution to half of the sampling frequency. Alternatively, the framework of Compressed Sensing (CS) shows that by exploiting prior information on the existence of sparse representations for the signal to be measured, one can achieve a similar resolution with significantly less samples. Another possibility to capture images with a reduced number of samples is the process of non-regular sampling, where the image is sampled at a subset of positions of a regular grid in a non-regular fashion. One state-of-the-art algorithm for reconstructing the image to the full regular grid is the Frequency Selective Reconstruction (FSR). Even though FSR and CS were developed independently from each other, both approaches use the sparsity property of natural images to regularize the otherwise underdetermined problem. Therefore, an exchange of ideas between the two concepts seems valuable. In this work, similarities and differences between non-regular sampling, with specific focus on FSR, and CS are examined. The theoretical guarantees from CS seem to prove parts of the success of FSR. Subsequently, reconstruction algorithms, usually utilized in the CS framework, are analyzed for their applicability for resampling images to a full grid from only a non-regular subset of pixel positions. Without additional modifications, standard CS algorithms show insufficient results, but the incorporation of ideas from FSR allow for a significant improvement of reconstruction quality. Thus, novel modified CS algorithms are proposed in this work. The simulation results show improvements by these modified algorithms, compared to the unmodified versions, by at least 0.9 dB PSNR.
Kurzfassung


In dieser Arbeit werden die Gemeinsamkeiten und Unterschiede zwischen der nicht-regelmäßigen Abtastung, mit Fokus auf der FSR, und CS untersucht. Die theoretischen Aussagen aus CS können Teile des Erfolgs der FSR erklären. Des Weiteren wird die Verwendbarkeit von CS-Algorithmen für die Rekonstruktion unregelmäßig abgetasteter Bilder analysiert. Ohne weitere Veränderungen zeigen die standardmäßigen CS-Algorithmen ungenügende Ergebnisse, jedoch kann die Einbringung von Ideen aus der FSR zu einer signifikanten Steigerung der Rekonstruktionsqualität führen. Folglich werden in dieser Arbeit neue modifizierte CS-Algorithmen vorgestellt. Die Simulationsergebnisse zeigen eine Verbesserung durch die modifizierten Algorithmen, im Vergleich zu den unmodifizierten Varianten, um mindestens 0.9 dB PSNR.
Contents

1 Introduction

2 Basics of Compressed Sensing
   2.1 Description of the Basic Compressed Sensing Problem
   2.2 Compressed Sensing Reconstruction Algorithms
      2.2.1 Matching Pursuit
      2.2.2 Orthogonal Matching Pursuit
      2.2.3 Iterative Hard Thresholding k
      2.2.4 Iterative Hard Thresholding
      2.2.5 Iterative Soft Thresholding
      2.2.6 Subspace Pursuit
      2.2.7 Basis Pursuit Denoise
   2.3 Suitable Measurement Matrices for Compressed Sensing

3 Non-regular Sampling and Reconstruction
   3.1 Description of Non-Regular Sampling of Images
   3.2 Applications of Resampling from a Non-regular Grid
   3.3 Sparsity Prior and Related Resampling Algorithms
      3.3.1 Description of Resampling Algorithms Using a Sparsity Prior
      3.3.2 Description of the Frequency Selective Reconstruction

4 Connection between FSR and the CS Framework
   4.1 Describing Non-regular Sampling as a Linear System of Equations
   4.2 Connection between Non-regular Sampling and Compressed Sensing
4.3 Comparison of Frequency Selective Reconstruction (FSR) and Compressed Sensing (CS) reconstruction algorithms
48

5 Using CS Reconstruction Algorithms within the FSR
56
  5.1 Incorporation of Spatial Weighting and Frequency Prior into CS algorithms
     57
      5.1.1 Spatial Weighting
     58
      5.1.2 Frequency Prior
     61
  5.2 Implementation Details for Experimental Setup
     63
  5.3 Parameter Exploration on Natural Image Blocks
     67
  5.4 Evaluation on Artificial Strictly Sparse Blocks
     71
  5.5 Evaluation on Natural Image Blocks
     75

6 Comparison with other Available Resampling Algorithms
83

7 Conclusion and Outlook
92

A Reconstruction Quality for Artificial Sparse Blocks
95

B Visual Results for Sampling Rate 0.2
98

Mathematical Operations and Conventions
106

List of Abbreviations
108

List of Figures
110

List of Tables
114

Bibliography
115
Chapter 1

Introduction

When asking a professional in signal processing, how many samples are required to measure a signal sufficiently, so that it can be reconstructed perfectly, the answer is usually that one needs to sample with twice the highest frequency present in the signal. In the case, where only the information is given, that the signal is band-limited, this is of course the correct answer. But what if additional or different properties are known of the otherwise unknown signal?

A branch of signal processing dealing with this question is CS. There, the prior knowledge that the signal exhibits a sparse representation in some domain, i.e., it can be represented by only a small amount of linear combinations of basis functions, is analyzed. The CS framework [19, 28] states that sampling rates much lower than indicated by the Shannon-Nyquist theorem [79] can be achieved, if the signal is sparse enough. This is accomplished by taking a small amount of linear measurements of the signal, which is equivalent to correlating the signal with a range of measurement functions. Fortunately, many prevalent signals, like audio signals and natural or medical images, can be expressed – or at least approximated – by a sparse representation.
Separately to CS, the task of capturing image signals with a reduced number of samples is explored in literature, by sampling the image at non-regular positions and subsequently reconstructing it to a regular grid. A procedure well suited for this process is the FSR [73]. Even though CS and the FSR were developed independently from each other, both concepts use the same property of image signal to achieve a reconstruction of high quality, which is the existence of sparse representations.

Due to the similarities of the two approaches, a comparison and subsequent exchange of ideas, that were developed independently, seems valuable. Therefore, this work aims to describe non-regular sampling and FSR within the CS framework, by examining the similarities and differences between the concepts. Furthermore, the theoretical bounds on the reconstruction error, provided by CS, are applied to the problem of non-regular sampling. Finally, the reconstruction algorithms, originating from the CS framework, are utilized for the task of reconstructing non-regularly sampled images and their performance is analyzed. Ideas from the FSR are additionally incorporated into these CS algorithms to improve their reconstruction quality.

For this, at first the relevant basics of CS are introduced in Chapter 2. After describing the basic CS problem and conditions for its reconstruction, different CS reconstruction algorithms and their theoretical guarantees are presented. These algorithms are later analyzed for the reconstruction of non-regularly sampled images. The chapter is finished, by presenting suitable measurement systems, i.e. the measurement functions that are correlated with the signal. In Chapter 3 the concept of non-regular sampling of images and its reconstruction to a regular grid is described. There, different applications where the necessity of resampling algorithms is required are presented and an overview of available resampling algorithms is given, where the FSR is introduced in more detail. Subsequently, similarities and differences between CS and non-regular sampling, with additional focus on the FSR, are examined in Chapter 4 and the applicability of the theoretical guarantees of CS for non-regular sampling and the FSR is analyzed. Then, in Chapter 5, the applicability of the earlier presented CS reconstruction algorithms
for the task resampling images from non-regularly sampled positions is tested exper-
imentally on image blocks. Ideas from the FSR are incorporated for this, improving
the reconstruction quality of the CS reconstruction algorithms. This shows, that the
concepts developed with the FSR are generally applicable to CS algorithms. In Chap-
ter 6 the proposed modified CS reconstruction algorithms are integrated into the block
processing of the FSR, to reconstruct whole images and an experimental comparison
to other state-of-the-art resampling algorithms is presented. Although, the proposed
methods, can not achieve a higher reconstruction quality nor a lower execution time
than the original FSR, they can compete with other state-of-the-art resampling al-
gorithms. Lastly, Chapter 7 gives a concluding summarization of the results and an
outlook for potential future work is displayed.
Chapter 2

Basics of Compressed Sensing

About a decade ago, Compressed Sensing (CS) gained significant traction in different areas of electrical engineering, computer science and applied mathematics, starting from the seminal publications from Candès et al. [16] and Donoho [28]. It is a concept that challenges the conventional approach of sampling signals by suggesting that instead of assuming a band-limited property for the measured signal, the measured signal exhibits a sparse representation in some domain, i.e., it can be represented by only a small amount of linear combinations of basis functions. This allows for a sampling rate that can be significantly lower than specified by the classical Shannon-Nyquist theorem [79].

2.1 Description of the Basic Compressed Sensing Problem

The basics of Compressed Sensing deal with the efficient measurement and exact or approximate reconstruction of finite, discrete and sparse signals. As a measurement
2.1. DESCRIPTION OF THE BASIC COMPRESSED SENSING PROBLEM

\[
\begin{pmatrix}
0.3583 \\
-0.2268 \\
-0.0649 \\
0.0802 \\
0.1122 \\
-0.0503
\end{pmatrix}
= \begin{pmatrix}
0.10 & 0.59 & -0.02 & 0.03 & -0.52 & -0.04 & 0.36 & -0.13 & 0.30 & 0.30 \\
0.39 & -0.19 & 0.07 & 0.20 & 0.43 & -0.42 & 0.05 & 0.17 & 0.24 & 0.24 \\
0.28 & -0.06 & -0.07 & 0.19 & 0.12 & 0.54 & -0.29 & -0.65 & 0.05 & 0.05 \\
-0.22 & -0.22 & 0.01 & -0.47 & -0.31 & -0.29 & -0.57 & -0.15 & 0.30 & 0.30 \\
0.23 & 0.39 & 0.27 & 0.24 & -0.05 & -0.49 & -0.40 & -0.24 & -0.40 & -0.40 \\
-0.63 & 0.13 & -0.52 & 0.38 & 0.19 & -0.20 & -0.05 & -0.16 & -0.12 & -0.12
\end{pmatrix}
\begin{pmatrix}
0 \\
0.2334 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0.10 & 0.59 & -0.02 & 0.03 & -0.52 & -0.04 & 0.36 & -0.13 & 0.30 & 0.30 \\
0.39 & -0.19 & 0.07 & 0.20 & 0.43 & -0.42 & 0.05 & 0.17 & 0.24 & 0.24 \\
0.28 & -0.06 & -0.07 & 0.19 & 0.12 & 0.54 & -0.29 & -0.65 & 0.05 & 0.05 \\
-0.22 & -0.22 & 0.01 & -0.47 & -0.31 & -0.29 & -0.57 & -0.15 & 0.30 & 0.30 \\
0.23 & 0.39 & 0.27 & 0.24 & -0.05 & -0.49 & -0.40 & -0.24 & -0.40 & -0.40 \\
-0.63 & 0.13 & -0.52 & 0.38 & 0.19 & -0.20 & -0.05 & -0.16 & -0.12 & -0.12
\end{pmatrix}
\begin{pmatrix}
y \\
x
\end{pmatrix}
\]

**Figure 2.1:** Example of a linear measurement process of a 2-sparse vector.

system only linear systems are considered [16, 28]. Plainly spoken, we multiply a vector, of which we know that it is sparse, with a matrix and try to inverse the linear system. Without the prior knowledge of the sparsity, the linear system is under-determined. This is specified more detailed in the following.

At first only strictly sparse signals are considered. Given a vector \( x \) element of \( \Sigma_k \), where \( \Sigma_k \) is the set of all \( k \)-sparse vectors for a given dimensionality \( N \), i.e., \( \Sigma_k = \{ v \in \mathbb{R}^N \mid \|v\|_0 \leq k \} \), reconstruct \( x \) given only a measurement vector \( y = \Phi x \) (2.1) and the measurement matrix \( \Phi \). \( \Phi \) is element of \( \mathbb{R}^{M \times N} \), where \( M \ll N \), and thus \( y \) is element of \( \mathbb{R}^M \). \( \|v\|_0 \) denotes the \( l_0 \)-"norm" of a vector \( v \) and is defined as the number of non-zero elements of the vector. This results in an under-determined linear system of equations with sparsity as prior information about the vector to be recovered. Figure 2.1 shows an example of a measurement process. The measurement vector \( y \in \mathbb{R}^M \) is the result of a matrix vector multiplication of the \( k \)-sparse input vector \( x \in \Sigma_k \subseteq \mathbb{R}^N \), that has to be acquired, with the measurement matrix \( \Phi \in \mathbb{R}^{M \times N} \). As \( M \ll N \) the measurement vector consists of fewer samples than the \( k \)-sparse vector. While Chapter 2 is mostly restricted to real-valued signals, the concepts can be extended to complex-valued signals as well [38, 39].

CS now states that an exact recovery of \( x \), given \( y \) and \( \Phi \) is possible, if \( \Phi \) fulfills...
certain properties and a recovery process is given by

$$x = \arg \min_{x' \in \mathbb{R}^N} \|x'\|_0, \quad s.t. \quad y = \Phi x'.$$  

(2.2)

As this recovery algorithm has combinatorial computational complexity, it is not useful in practice and plenty of alternative reconstruction algorithms have been proposed and analyzed in the CS framework [31]. The goal is to find algorithms that can successfully recover the sparse vector $x$ for small numbers of measurements, while still being computationally feasible. Additionally, many real world problems consist of vectors which are not strictly sparse or contain added noise. Stable recovery algorithms that can find good approximations in these cases are also considered in the CS domain. Before introducing some of these reconstruction algorithms, a closer look at the matrix properties of $\Phi$, which are necessary for a perfect recovery or tight approximation, is given.

At first some conditions are presented, which guarantee that for each pair of different signals $x, x' \in \Sigma_k$ with $x \neq x'$ the measurement results in two different measurement vectors $\Phi x \neq \Phi x'$, leading to a unique recovery for each k-sparse vector. One property of a matrix $\Phi$ that is important in this context is the spark of a matrix. If the spark of a matrix, which is the smallest number of columns of the matrix that are linearly dependent, is greater than $2k$, a unique measurement vector for each sparse input vector is guaranteed [29]. As the computation of the spark of a general matrix is computationally expensive, it is preferable to use different properties than the spark to provide recovery guarantees. One of these is the coherence of a matrix. The coherence $\mu(\Phi)$ is the normalized largest absolute inner product between any two columns of $\Phi$:

$$\mu(\Phi) = \max_{0 \leq i \neq j \leq N-1} \frac{|\phi_i \phi_j^H|}{\|\phi_i\|_2 \|\phi_j\|_2},$$  

(2.3)

where $\phi_i$ denotes the $i$-th column of $\Phi$. The coherence of a matrix is bounded by
2.1. DESCRIPTION OF THE BASIC COMPRESSED SENSING PROBLEM

\( \mu(\Phi) \in [\sqrt{\frac{N-M}{M(N-1)}}, 1] \), with the lower bound being known as the Welch bound [81]. When \( M \ll N \), the Welch bound can be approximated by \( \frac{1}{\sqrt{M}} \). Now if

\[
 k < \frac{1}{2} \left( 1 + \frac{1}{\mu(\Phi)} \right),
\]

the uniqueness of the measurement vector for each sparse input vector is guaranteed [29]. Together with the Welch bound this results in an upper bound on the sparsity \( k \) which guarantees perfect recovery for

\[
 k \leq O(\sqrt{M}).
\]

These already presented matrix properties only guarantee uniqueness of the measurement vector for each strictly \( k \)-sparse signal. As we are interested in natural signals in this work, i.e., natural images, which do not have a strictly sparse representation in any domain or useful dictionary, a look at another important property is given. This property is known as the \((k, \delta)\)-restricted isometry property (RIP). It is fulfilled for the matrix \( \Phi \) if

\[
 (1 - \delta) \|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta) \|x\|_2^2,
\]

for all \( x \in \Sigma_k \) [31]. We denote the smallest \( \delta \), for which the \((k, \delta)\)-RIP for a given \( k \) is fulfilled as \( \delta_k \). The RIP is closely linked to the coherence and spark of a matrix. If \( \Phi \) has unit norm columns and coherence \( \mu(\Phi) \), then \( \Phi \) has the RIP with [11]

\[
 \delta_k \leq \mu(k - 1).
\]

Additionally the RIP provides guarantees for the stability of the recovery process in the presence of noise, i.e., the distance between the measurement vector of two different sparse signal should be proportional to the distance between the sparse signals itself. Thus, small noise leads to a recovered signal that has only a small distance to the
original signal. Another important aspect of the RIP are resulting guarantees for signals that are not strictly sparse but consist of some dominating or fast decaying coefficients. As will be presented in the next section, the RIP therefore allows to show the existence of efficient and robust algorithms for reconstructing sparse signals or signals which can be approximated with only a few coefficients in a specific basis from some small amount of linear measurements.

2.2 Compressed Sensing Reconstruction Algorithms

To invert the measurement process (2.1), efficient reconstruction algorithms with respect to runtime and recovery error have to be found. A large part of the CS framework deals with this problem and countless alternative approaches, originating from different areas of signal processing and optimization theory, have been proposed. In this work, only a small subset of these is utilized. The focus is on popular algorithms that have consistently been proven, in practice and partly in theory, to show good recovery results.

2.2.1 Matching Pursuit

The Matching Pursuit (MP) algorithm is an iterative greedy approach to reconstruct the sparse vector. It therefore belongs to the so called greedy class of CS reconstruction algorithms. At the beginning of the algorithm a residual vector $r$ is set to the measurement vector $y$. In each iteration the residual is correlated with each column (atoms) of the measurement matrix $\Phi$. The atom with the highest absolute correlation is selected and the correlation is added to the result vector $\hat{x}$ at the position $u$, indicated by $\hat{x}_u$, corresponding to the selected atom. The remaining coefficients of $\hat{x}$, indicated by $\hat{x}_{\neq u}$, are kept the same. Consequently, the residual needs to be updated by subtracting the projection of the previous residual onto the selected basis function. This is done until the residual gets small enough or a maximum number of iterations
is reached \[56\]. Without loss of generality, $\Phi$ is assumed to have unit norm columns for the rest of Section 2.2. Algorithm 1 shows the pseudo code of the Matching Pursuit, which stops after $t$ iterations. Here, $\phi_l$ denotes the $l$-th column of the matrix $\Phi$.

**Algorithm 1** Matching Pursuit

**Input:** Measurement Matrix $\Phi$, measurement vector $y$, number of iterations $t$

**Output:** Sparse model $\hat{x}$

1: Initialization: $r_0 := y$, $\hat{x}_0 := 0$, $i := 0$

2: while $i < t$ do

3: $u := \arg \max_{l \in \{0...N-1\}} \left| \phi_l^H r_i \right|$

4: $\hat{x}_{i+1}|_u := \hat{x}_i|_u + \phi_u^H r_i$

5: $\hat{x}_{i+1}|_{\Omega^C} := \hat{x}_i|_{\Omega^C}$

6: $r_{i+1} := r_i - \phi_u^H r_i \phi_u$

7: $i := i + 1$

8: end while

9: return $\hat{x} := \hat{x}_t$

Unfortunately, no theoretical recovery guarantees have been found in the literature for the Matching Pursuit, but it has shown good practical results, as will also be seen later in this work.

### 2.2.2 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) is an extension of the MP algorithm. Instead of subtracting the highest correlating atom from the residual, a vector base is created by adding the highest correlating vector to this base in each iteration. The result vector $\hat{x}$ is updated in each iteration by the projection of the measurement vector $y$ onto the vector space spanned by the created vector base. The residual is then updated accordingly as the difference between the current result vector and the measurement vector. In this case, a different atom is picked in each iteration, as the residual will always be orthogonal to previously picked atoms, even for non orthogonal atoms [65].

Algorithm 2 shows the pseudo code of the OMP. Here, $\Omega$ denotes the set containing the indices of the columns of $\Phi$, which constitute the current vector base. $\hat{x}_{i+1}|_{\Omega}$ indicates the restriction of $\hat{x}_{i+1}$ to the elements indexed by $\Omega$ and $\Omega^C$ is the complementary set of $\Omega$ with respect to all possible column indices.
CHAPTER 2. BASICS OF COMPRESSED SENSING

Algorithm 2 Orthogonal Matching Pursuit

Input: Measurement Matrix $\Phi$, measurement vector $y$, number of iterations $t$

Output: Sparse model $\hat{x}$

1: Initialization: $r_0 := y, \hat{x}_0 := 0, \Omega := \emptyset, i := 0$
2: while $i < t$ do
3: $u := \arg \max_{l \in \{0, \ldots, N-1\}} |\phi_l^H r_i|$
4: $\Omega := \Omega \cup \{u\}$
5: $\hat{x}_{i+1\Omega} := \Phi_{\Omega}^H y$
6: $\hat{x}_{i+1\Omega^C} := 0$
7: $r_{i+1} := y - \Phi \hat{x}_{i+1}$
8: $i := i + 1$
9: end while
10: return $\hat{x} := \hat{x}_t$

Recovery guarantees for OMP which is used to recover strictly sparse signals or strictly sparse signals with noise can be linked to the coherence and the RIP of the measurement matrix (see e.g., [30] and [84] for details). Promising guarantees for the recovery of approximately sparse signals, which are proven for some of the following algorithms, have not yet been shown for OMP [31].

2.2.3 Iterative Hard Thresholding $k$

Besides the greedy algorithms, there exists the class of iterative thresholding algorithms in the CS framework. One prominent instance is the Iterative Hard Thresholding Algorithm $k$ (IHT$_k$) from [8]. The algorithm can be simply stated in one line by

$$\hat{x}_{i+1} = H_k(\hat{x}_i + \Phi^H(y - \Phi \hat{x}_i)).$$

(2.8)

Here, $H_k(x)$ is a thresholding operator, setting all but the biggest - regarding their magnitude - $k$ elements in $x$ to zero. If there exists no such unique set, one can prioritize elements by an arbitrary predefined order. This calculation is again done for a fixed number of iterations $t$. The sparsity level $k$ is a free parameter, which should correspond to the sparsity of the vector to be recovered. IHT$_k$ tries to minimize the cost function $\|y - \Phi \hat{x}\|_2$ under the constraint that $\hat{x}$ is $k$-sparse using a majorization
minimization approach. If the norm of the measurement matrix \( \| \Phi \|_2 \) is smaller than one, the convergence of this algorithm to a local minimum of the cost function is guaranteed [7]. In this work, a slightly modified version, which is already suggested in [7] and expanded in [9] is used to counteract scaling of the measurement matrix \( \Phi \) which would result in a violation of the norm condition. The modified algorithm is described as:

\[
\hat{x}_{i+1} = H_k(\hat{x}_i + \kappa_{iht} \Phi^H(y - \Phi \hat{x}_i)).
\] (2.9)

In [9], \( \kappa_{iht} \) is calculated separately in each iteration. For the sake of simplicity, we set the step size \( \kappa_{iht} \) to a constant value but perform enough iterations to still converge to a solution.

Convergence, and additionally a powerful recovery guarantee, for the IHT\(_k\) algorithm have also been linked to the RIP of the measurement matrix \( \Phi \) [8]: Given a noisy observation \( y = \Phi x + e \), where \( x \) is an arbitrary vector to be reconstructed and \( e \) models a stochastic or deterministic observation noise, which has a known upper bound for its \( l_2\)-norm. Now, if \( \Phi \) has RIP with \( \delta_{2k} \leq 0.0222 \) [8], then, at iteration \( i \), IHT\(_k\) will recover an approximation \( x_i \) satisfying

\[
\| x - x_i \|_2 \leq 2^{-i} \| x_k \|_2 + 5\tilde{\epsilon}_k,
\] (2.10)

where

\[
\tilde{\epsilon}_k = \| x - x_k \|_2 + \frac{1}{\sqrt{s}} \| x - x_k \|_1 + \| e \|_2.
\] (2.11)

The vector \( x_k \) is defined as the best \( k \)-term approximation to the vector \( x \), i.e., \( x_k = \arg \min_{x' \in \Sigma_k} \| x - x' \|_2 \). Thus, with an increasing amount of iterations, the upper bound of the reconstruction error will converge to a multiple of an error term \( \tilde{\epsilon}_k \).

Moreover, to give a bound independent of the iteration count for comparison with
other algorithms, after at most

$$\tilde{i} = \left\lceil \log_2 \left( \frac{\|x_k\|_2}{\tilde{\epsilon}_k} \right) \right\rceil$$  \hspace{1cm} (2.12)

iterations, IHT\textsubscript{k} estimates $x$ bounded by

$$\|x - x_{\tilde{i}}\|_2 \leq 6\tilde{\epsilon}_k.$$  \hspace{1cm} (2.13)

The first part of the error term consists of a multiple of the $l_2$-distance between the signal $x$ to be reconstructed and its best $k$-term approximation $x_k$. Obviously, this dependence is optimal, up to a constant factor, as we cannot get a better approximation with only $k$ coefficients, even when the vector $x$ would be known. The second part, which represents the distance of the best $k$-term approximation with respect to the $l_1$-norm, can also be shown to be ideal up to a constant factor (see [8] IV. B.). Finally, the dependence on the norm of the noise vector $\|e\|_2$ is unavoidable, as even with a known support of $x_k$, the noise $e$ projected onto the subspace spanned by this support, has to be proportional to $\|e\|_2$. If an algorithm can give such a guarantee (2.13), it is said to hold to the uniform instance optimality, i.e., the recovery error is proportional to that of the best $k$-sparse approximation [31].

### 2.2.4 Iterative Hard Thresholding

Instead of hard thresholding the largest $k$ entries, as in IHT\textsubscript{k}, successful empirical and theoretical results have been found by using a thresholding constant in the hard thresholding operation. This results in the following description of the Iterative Hard Thresholding (IHT) algorithm:

$$\hat{x}_{i+1} = T_{\tau_i}(\hat{x}_i + \kappa_{\text{IHT}} \Phi^H(y - \Phi \hat{x}_i)), \hspace{1cm} (2.14)$$
2.2. COMPRESSED SENSING RECONSTRUCTION ALGORITHMS

where $T_{\tau_i}(x)$ is the hard thresholding function:

$$ T_{\tau_i}(x)_n = \begin{cases} x_n & \text{if } |x_n| \geq \tau_i \\ 0 & \text{otherwise} \end{cases} $$

(2.15)

Here, $x_n$ specifies the $n$-th coefficient of the vector $x$. For selecting the correct parameters $\kappa_{\text{IHT}}$ and $\tau_i$, ideas from [54] are used in this work. In this publication a concept for determining the thresholding parameter $\tau_i$ in each iteration is proposed and a parameter exploration is done. In Section 5.2 this is explained in more detail.

Successful applications of the IHT have been shown in particular by Starck and coauthors ([74], [35], [10]), using sparse approximations for the separation of the texture and cartoon layers and inpainting of images. In [45], a theoretical guarantee on the recovery of $x$ by IHT, using a constant threshold parameter $\tau$ for each iteration, is given for strictly sparse signals with constant entries utilizing the coherence of the measurement matrix $\Phi$. Reconstruction bounds for general signals, as presented for IHT$_k$ in Section 2.2.3, have not been found in the literature, when using a more general threshold parameter choice, i.e., not selecting the biggest $k$ entries in each iteration as in IHT$_k$.

2.2.5 Iterative Soft Thresholding

An additional iterative thresholding algorithm from the CS framework is the Iterative Soft Thresholding (IST), also known as Iterative Shrinkage-Thresholding algorithm (ISTA). It is essentially the IHT reconstruction algorithm with a soft thresholding function instead of the hard thresholding:

$$ \hat{x}_{i+1} = S_{\kappa_i}(\hat{x}_i + \kappa_{\text{IHT}} \Phi^H(y - \Phi \hat{x}_i)) $$

(2.16)
with

$$S_{s_i}(x)_n = \begin{cases} x|_n + s_i & \text{if } x|_n \leq -s_i \\ x|_n - s_i & \text{if } x|_n \geq s_i \\ 0 & \text{otherwise} \end{cases},$$

(2.17)

if $x \in \mathbb{R}^N$. When dealing with complex vectors, $x \in \mathbb{C}^N$, the thresholding is only applied on the magnitude, i.e., $S_{s}(r e^{j\omega}) = S_{s}(r)e^{j\omega}$. Here, $j$ corresponds to the imaginary unit. Again, like for IHT, the selection of the correct parameters $\kappa_{ist}$ and $s_i$ is inspired by [54] in this work. In contrast to IHT, the derivation of IST comes from the minimization of a relaxed cost function, the so called Basis Pursuit Denoise (BPDN):

$$\arg \min_{x' \in \mathbb{R}^N} \|y - \Phi x'\|_2 + \lambda \|x'\|_1,$$

(2.18)

where $\lambda$ is an arbitrary relaxation parameter. Here the $l_0$-“norm” is replaced by the convex $l_1$-“norm”. The derivation of the IST algorithm to solve problems of this form were introduced by different authors, with additional applications to inverse problems in imaging, like denoising or deconvolution [23, 33, 37, 63]. The convergence of the IST was analyzed in [23]. Successful extensions of the IST to provide a faster convergence were given in [5] and [3]. In this work, the basic version, as presented in (2.16) was used. As the IST with constant thresholding parameters $s_i$ converges to the minimum of the cost function under certain conditions [23, 42], it solves the BPDN problem, for which powerful recovery guarantees have been shown. These guarantees are stated in more detail in Section 2.2.7, where another algorithm to solve the BPDN problem is presented.

### 2.2.6 Subspace Pursuit

Another important iterative algorithm in the compressed sensing framework, which is also applied in this thesis, is the Subspace Pursuit (SP) algorithm [22]. It is a nearly identical version of the Compressive Sampling Matching Pursuit (CoSaMP) [62]. The
pseudo code of the SP algorithm with a hard iteration count $t$ can be seen in Algorithm 3. The function $\text{supp}(x)$ returns a set, containing all the indices of non-zero elements of the vector $x$. SP is a two stage algorithm that uses exact solutions of small linear systems combined with thresholding before and after this solution. In the first stage, the biggest $K$ coefficients in magnitude of the projected residual are calculated like in IHT$_k$, IHT and IST (which is the only difference to CoSaMP, as CoSaMP uses the biggest $2K$ coefficients instead of only $K$ in the first stage). The indices $T$ of these coefficients are merged to the support indices $\text{supp}(\hat{x}_i)$ of the current sparse signal estimate, giving the set $\Omega$. As in OMP, the linear system of equations, resulting from the restriction of the columns of the measurement matrix $\Phi$ to the set $\Omega$ is inverted in a least square sense. Effectively, the measurement vector $y$ is projected onto the space spanned by the columns of $\Phi$ with indices in $\Omega$. In stage two, the result is thresholded a second time for the largest $K$ coefficients, producing the current sparse model estimate $\hat{x}_{i+1}$. In contrast to the original proposal, this is done for $t$ iterations, whereas the algorithm in [22] stops when the residual energy increases. Like IHT$_k$, SP can give an elegant recovery error, regarding the reconstruction of general signals - not only sparse ones - if the measurement system $\Phi$ is sufficiently conditioned: Let $x \in \mathbb{R}^N$ and $y = \Phi x + e$, as in Section 2.2.3. Suppose that the sampling matrix satisfies the RIP with parameter $\delta_{6K} < 0.083$, then,

$$
\|x - \hat{x}\|_2 \leq c'_2 k \left( \|e\|_2 + \sqrt{\frac{1 + \delta_{6K}}{k} \|x - x_k\|_1} \right).
$$

The vector $x_k$ is again the best $k$-sparse approximation to $x$ and $c'_2 k = \frac{1 + \delta_{6k} + \delta_{6k}^2}{\delta_{6k}(1 - \delta_{6k})}$. In order to obtain the claim in (2.20), the input sparsity level $K$ of SP needs to be set to $K = 2k$ [22]. Therefore the uniform instance optimality is also shown for Subspace Pursuit. The reconstruction error $\|x - \hat{x}\|_2$ is bounded proportionally to the error of
the best $k$-sparse approximation $\|x - x_k\|_1$ and the norm of the error term $\|e\|_2$.

Algorithm 3 Subspace Pursuit

**Input:** Measurement Matrix $\Phi$, measurement vector $y$, sparsity level $K$, number of iterations $t$

**Output:** Sparse model $\hat{x}$

1: Initialization: $r_0 := y$, $\hat{x}_0 := 0$, $\Omega := \emptyset$, $i := 0$
2: while $i < t$ do
3: $e := \Phi^H r_i$ \{form projected residual signal estimate\}
4: $T := supp(H_K(e))$ \{threshold projected residual\}
5: $\Omega := T \cup supp(\hat{x}_i)$ \{merge supports\}
6: $b|_\Omega := \Phi|_\Omega^\dagger y$ \{form stage one signal estimate\}
7: $b|_{\Omega^c} := 0$ \{form stage one signal estimate\}
8: $x_{i+1} := H_K(b)$ \{threshold signal\}
9: $r_{i+1} := y - \Phi \hat{x}_{i+1}$ \{update residual\}
10: $i := i + 1$
11: end while
12: return $\hat{x} := \hat{x}_t$

### 2.2.7 Basis Pursuit Denoise

The Basis Pursuit Denoise (BPDN) problem was already introduced in (2.18):

$$\arg \min_{x' \in \mathbb{R}^N} \|y - \Phi x'\|_2 + \lambda \|x'\|_1.$$  \hspace{1cm} (2.21)

In the non-Lagrangian form - which is usually named LASSO - it is stated as follows:

$$\hat{x} = \arg \min_{x' \in \mathbb{R}^N} \|x'\|_1, \quad \text{s.t.} \quad \|\Phi x' - y\|_2 \leq \sigma,$$  \hspace{1cm} (2.22)

with $\sigma \geq 0$ being a regularization parameter, to be chosen appropriately. For a specific $\lambda$, LASSO (2.22) and BPDN (2.21) are equivalent. Besides utilizing Iterative Shrinkage-Thresholding algorithms to solve this optimization, the BPDN problem can be recast into a Second Order Cone Program (SOCP) and solved efficiently using known SOCP-solvers [17] or solved using a spectral gradient-projection approach [80]. When identifying an algorithm with BPDN from now on in this work, the solver from [80] is meant.
Regarding error bounds on the reconstructed signal $\hat{x}$, the BPDN was the first algorithm for which the uniform instance optimality could be shown. In [17], Candes et al. have proven it in the following form: Let $x \in \mathbb{R}^N$ and $y = \Phi x + e$, as previously described in Section 2.2.3 and Section 2.2.6. Now, if the measurement matrix $\Phi$ fulfills the RIP such that $\delta_{3k} + 3\delta_{4k} < 2$ or $\delta_{2k} < \sqrt{2} - 1$ [15], then the result $\hat{x}$ of the BPDN reconstruction algorithm obeys

$$\|x - \hat{x}\|_2 \leq C_{1,BPDN} \|e\|_2 + \frac{C_{2,BPDN}}{\sqrt{k}} \|x - x_k\|_2.$$  \hspace{1cm} (2.23)

For reasonable values of $\delta_{4k}$ the constants $C_{1,BPDN}$ and $C_{2,BPDN}$ are well behaved. Exemplary, if $\delta_{4k} = 1/5$, the constants are approximately $C_{1,BPDN} \approx 12.04$ and $C_{2,BPDN} \approx 8.77$ [17].

In literature, there exist plenty other reconstruction algorithms that try to inverse the measurement problem under a sparsity constraint, but this work concentrates only on the ones presented in the current section. After having introduced some of these stable reconstruction algorithms, partly with efficient recovery guarantees, the question, which measurement matrices are suitable for a robust reconstruction still remains open. Can we construct matrices that fulfill the RIP constraints gracefully with only a few rows compared to the number of its columns? This question will be tackled in the next section.

### 2.3 Suitable Measurement Matrices for Compressed Sensing

The guarantees, stated in the previous section for some of the algorithms, can be summarized according by the following. If the measurement matrix $\Phi$ fulfills the $(c \cdot k, \delta)$-RIP, i.e., $\delta_{c,k} < \delta$, then the recovery errors of the SP, IHT$_k$ and BPDN algorithms
are bounded by

$$\| \mathbf{x} - \hat{\mathbf{x}} \|_2 \leq C_1 \| \mathbf{x} - \mathbf{x}_k \|_2 + \frac{C_2}{\sqrt{k}} \| \mathbf{x} - \mathbf{x}_k \|_1 + C_3 \| \varepsilon \|_2 .$$  \hspace{1cm} (2.24)

The restriction on the parameters $c, \delta$ of the RIP and the constants $C_1, C_2, C_3$ are specific to each algorithm. To have a low bound on the reconstruction error, it is therefore necessary for the measured vector $\mathbf{x}$ to be sufficiently approximated by a strictly sparse vector with a low number of coefficients. Additionally it is important that the measurement matrix $\Phi$ has a small $\delta_k$ for a large $k$ even when its number of rows $M$, i.e., the number of measurements, is low compared to its number of columns $N$. Unfortunately no known deterministic matrix can provide a satisfying RIP constant with regard to its number of rows $M$ [31, 28, 78]. There exist deterministic matrices that can reach the coherence lower bound, resulting in perfect recovery when the number of measurements $M$ and the sparsity $k$ of the signal $\mathbf{x}$ is in the order of $M = \mathcal{O}(k^2)$ (2.5). Considering the connection between RIP and coherence in (2.7) and the Welch bound for the coherence ($\approx \frac{1}{\sqrt{N}}$), the RIP-constant is restricted by $\delta_k \leq \frac{(k-1)}{\sqrt{M}}$. Non-trivial RIP-constants, i.e., $\delta_k < 1$, can therefore only be given for $M \geq \mathcal{O}(k^2)$, as the denominator needs to increase at least linearly with $k$. The construction of deterministic measurement matrices obeying to a useful $(k, \delta)$-RIP has been shown in [25] if $k \leq \mathcal{O} \left( \frac{\sqrt{M \log(M)}}{\log(M)} \right)$. This results again in a required amount of measurements in the order of $M \geq \mathcal{O}(k^2)$. To give a better correspondence between the number of measurements and the sparsity of the signal, one needs to look at probabilistic sensing matrices and resulting probabilistic guarantees. Two important examples of these random matrices are the Gaussian matrices and partial Fourier matrices. The former is interesting, as it provides optimal RIP behavior [14] and the latter, as is can easily be stored in memory and applied to a vector in $\mathcal{O}(N \log(N))$ time. Furthermore the partial Fourier matrix is of special interest in this work, which will be expanded on in Section 4.2. Considering the Gaussian matrices, suppose that the entries of the $M \times N$ measurement matrix
2.3. SUITABLE MEASUREMENT MATRICES FOR COMPRESSED SENSING

\( \Phi_{\text{GAUSS}} \) are independently sampled from the normal distribution. Then, if

\[
M \geq \frac{C_{1,\text{GAUSS}} \cdot k \cdot \log(N/k)}{\epsilon_{\text{GAUSS}}^2},
\]

the Gaussian measurement matrix \( \Phi_{\text{GAUSS}} \) fulfills the RIP with constant \( \delta_k \leq \epsilon_{\text{GAUSS}} \), except with probability \( e^{-C_{2,\text{GAUSS}}} \) [62]. \( C_{1,\text{GAUSS}} \) and \( C_{2,\text{GAUSS}} \) represent again some well behaved constants. Sufficient RIP conditions can thus be achieved with high probability, when the number of rows \( M \) is in the order of \( O(k \cdot \log(N/k)) \), resulting in a linear relationship between the sparsity and the required amount of measurements. Together with the uniform instance optimality, CS allows, up to a constant, the measurement of the largest \( k \)-coefficients, i.e., the best \( k \)-sparse approximation, with only in the order of \( k \) measurements.

Partial Fourier matrices are defined as follows: The full Discrete Fourier Transform (DFT) matrix \( \mathbf{W} \) of size \( N \times N \) is the matrix, with entries,

\[
[W]_{a,b} = \frac{1}{\sqrt{N}} e^{-j2\pi ab/N}, \quad \text{for } a, b \in \{0 \ldots N - 1\}.
\]

Here \([W]_{a,b}\) denotes the matrix entry at position \((a, b)\) in \( \mathbf{W} \). The partial Fourier matrix \( \Phi_{\text{pf}} \) of size \( M \times N \) is then the probabilistic matrix, whose \( M \) rows are drawn uniformly at random from the rows of the full DFT matrix \( \mathbf{W} \). Illustratively the partial Fourier matrix, applied to a vector, can be described by seeing the vector coefficients as the Fourier transform of a spatial domain signal and just sampling the signal in the time domain at random positions. For the partial Fourier matrix exists a similar relationship to the RIP, as for the Gaussian matrix [62]. If \( \sqrt{M} \Phi_{\text{pf}} \) is the \( M \times N \) partial Fourier matrix and

\[
M \geq \frac{C_{1,\text{pf}} \cdot k \cdot (\log(N))^5 \cdot \log(\epsilon_{\text{pf}})}{\epsilon_{\text{pf}}^2},
\]

then \( \Phi_{\text{pf}} \) fulfills the RIP with constant \( \delta_k \leq \epsilon_{\text{pf}} \), except with probability \( N^{-1} \) [62]. \( C_{1,\text{pf}} \)
is an existent constant. Rudelson and Vershynin could improve this bound in [67] to
\[ M \geq \mathcal{O}(k \cdot (\log(N))^4). \] (2.28)
This establishes a slightly worse requirement than for the Gaussian matrix, but the scaling is still linear with respect to the sparsity level.

Interestingly, the process of sub-sampling \( M \) rows uniformly at random from an \( N \times N \) orthonormal matrix \( U \) and re-normalizing the columns to unit-norm has been analyzed generally. The resulting sub-sampled random matrix is then denoted \( \Phi_U \). Now the arguments from [67] state that a reasonable \((k, \delta)\)-RIP for \( \Phi_U \) can be achieved from
\[ M \geq \mathcal{O}(\mu_c^2 \cdot k \cdot (\log(N))^4) \] (2.29)
rows with overwhelming probability. The value of \( \mu_c \) is specific to the orthonormal matrix \( U \):
\[ \mu_c = \sqrt{N} \max_{a,b} |[U_{a,b}]|. \] (2.30)
Note, if \( U \) is the DFT matrix \( W \), then the result in (2.29) becomes the previously stated guarantee (2.28) for the partial Fourier matrix, as the complete DFT matrix has \( \mu_c = 1 \). Moreover the result holds for higher dimensional DFT matrices, like the 2D-DFT, due to their entries also having a constant amplitude. The 2D-DFT matrix can be expressed by bringing the two-dimensional basis functions, onto which the two-dimensional signal is projected, into a one dimensional vector form by concatenating each column together. The complex conjugate of these vectors are then used as rows of the 2D-DFT matrix. The two-dimensional signal, to be measured, is reduced to a vector in the same way, by concatenating its columns. The \( L^2 \) basis functions \( \psi_{a,b} \) \((a, b \in \{0 \ldots L - 1\})\) of the 2D-DFT of a signal with size \( L \times L \) can be written as
\[ \psi_{a,b}[m, n] = \frac{1}{L} e^{i2\pi am/L} e^{i2\pi bn/L}, \text{ for } m, n \in \{0 \ldots L - 1\}, \] (2.31)
which shows the constant amplitude of the coefficients of the matrix $U$, if it is a multi dimensional DFT-matrix. This concept can be expanded to an arbitrary amount of dimensions.

Considering the measurement of natural signals, often a so called sparsity matrix $\Psi \in \mathbb{C}^{N \times N}$ is necessary, because the signal $s$ to be measured is not sparse directly but exhibits a sparse representation with respect to some frame, i.e., $s = \Psi x$, where $x$ is sparse. This sparsity matrix is assumed to be orthonormal in the following. Sensing an image directly, by setting the spatial image $s$ to $x$, is obviously not feasible, as the spatial representation is not sparse. The sparsity matrix is then necessarily included in the measurement equation:

$$y = Q \Psi x,$$

(2.32)

to still preserve a sparsity prior on $x$. $Q$ is referred to as the spatial measurement matrix from now on and it is only a part of the whole measurement matrix $Q \Psi$. When setting $U = P \Psi$, with the expanded spatial measurement matrix $P \in \mathbb{R}^{N \times N}$ being another orthonormal matrix, the variable $\mu_c$ becomes

$$\mu_c = \sqrt{N} \max_{a,b \in \{1, \ldots, L^2\}} |p_a \Psi_b^T|,$$

(2.33)

where $p_a$ and $\Psi_b$ are the $a$-th and $b$-th rows of the matrices $P$ and $\Psi^T$ respectively. Here, the matrix $Q$ can be seen as the matrix, consisting of rows of $P$, selected uniformly at random. In this context, the value $\mu_c$ is called the mutual coherence between the matrices $P$ and $\Psi$ [19]. According to (2.29), the smaller the mutual coherence between the sparsity matrix and the expanded spatial measurement matrix $P$, the smaller is the number of measurements needed. This concept is also known as incoherent sampling [12]. In the case of partial Fourier matrices, $P$ is the identity matrix and $\Psi$ the DFT matrix. Note here, that one can take the DFT matrix $W$ or its inverse $W^H$ as the sparsity matrix $\Psi$ and achieve the same mutual coherence. This pair exhibits ideal mutual coherence, i.e., lowest possible mutual coherence $\mu_c = 1$, regardless of the DFT-
Another interesting aspect for suitable measurement matrices is shown in [18]. There, the problem of trying to approximate the sparse vector $x$ directly is replaced by just trying to have a good approximation in the spatial domain $\Psi x$. This allows efficient reconstruction guarantees for sparsity matrices that are not necessarily unitary and can even be an arbitrary frame with potentially more columns than rows. For natural image reconstruction problems, these guarantees could be useful, as the image signals are sparse with respect to certain frames, like the curvelet frame [13]. Unfortunately, the results in [18] do not show guarantees for simply sub-sampled frames, i.e., $\Psi$ is a frame and $Q$ is an identity matrix with some randomly missing rows in (2.32). Only guarantees of this form would be beneficial for the description task in this work, which is presented in Section 4.2.

As shown, CS allows for an efficient exploitation of a sparsity prior when measuring signals. The framework provides computationally feasible reconstruction algorithms, for which a subset can give near optimal theoretical recovery guarantees, when the measurement system is sufficiently conditioned. The important aspect to remember from this section for the following work, is that, when sensing nearly sparse signals with partial Fourier matrices, independently of the dimensionality, one can give efficient recovery guarantees for a set of reconstruction algorithms in the form of uniform instance optimality. Additionally, the sensing with partial Fourier matrices is equivalent to sub-sampling signals, which are nearly sparse in the Fourier domain, at random positions in the spatial domain. Sub-sampling signals randomly in the spatial domain is also known as non-regular sampling and is now introduced in the next chapter.

In summary it can be said that the FSR is related to the MP algorithm, but utilizes additional priors besides the sparsity information. There exist many approaches in the CS literature to incorporate further prior information into the reconstruction process. The additional modifications, i.e., orthogonality deficiency compensation, spatial
weighting and frequency prior, of FSR can not be directly found in CS literature and thus it might be beneficial to bring ideas from the FSR into existing CS reconstruction algorithms. Inversely, concepts like the smoothing with a Wiener filter as an intermediate step and usage of different CS reconstruction algorithms, like OMP, IHT\(_k\), IHT, IST, SP or BPDN as the basis reconstruction algorithm instead of MP can be included into FSR. As these algorithms often outperform MP and Section 4.2 indicated the applicability of CS algorithms for non-regular sampling, the idea to use different CS reconstruction algorithms inside the block reconstruction step of the FSR is analyzed experimentally in the next section. Furthermore, the concept of spatial weighting and frequency prior are incorporated into these CS procedures, showing potential use of the concepts, developed independently to the CS framework, of the FSR for other CS reconstruction tasks.
Chapter 3

Non-regular Sampling and its Reconstruction to a Regular Grid

Typically digital images are represented by pixels residing on a regular two-dimensional grid and the vast majority of image processing algorithms and image displaying applications often require the positioning of the pixels on a regular grid. Although most image capturing systems provide a regular representation directly, there occur many scenarios where this regular distribution of pixels is not given, but rather only a non-regular subset of pixel positions is available. As a consequence, a reconstruction algorithm, resampling the non-regularly sampled image to a regular grid, is needed to allow for further processing. Of course this problem can be expanded to different type of signals.

The main task of this work is to elaborate on the connections between Compressed Sensing, which has been introduced in the previous section, and the problem of reconstructing non-regularly sampled natural images to a regular grid with the specific algorithm from [73]. Therefore this chapter aims at describing the task of non-regular sampling and some possible reconstruction algorithms, that use the same prior information about the measured signals as utilized in CS. In the current chapter, this
3.1 Description of Non-Regular Sampling of Images

Non-regular sampling can be simply described by measuring only a subset of positions of a signal \( s[x_1, x_2] \), uniformly and independently at random, where \( x_1 \) and \( x_2 \) denote the spatial coordinates. This results in a non-regularly spaced signal \( s_{\text{nr}}[x_1, x_2] \), for which the positions that are not sampled are set to zero. The non-regular sampling generation process is thus given as

\[
s_{\text{nr}}[x_1, x_2] = s[x_1, x_2] \cdot b[x_1, x_2]. \tag{3.1}
\]

The sub-sampling mask \( b[x_1, x_2] \) is defined one for positions, that are sampled and zero otherwise. The probability of keeping a given sample, is denoted as the sub-sampling rate \( S \). Now, knowing only the non-regularly sampled signal \( s_{\text{nr}}[x_1, x_2] \) and the sub-sampling mask \( b[x_1, x_2] \), the task of the reconstruction process is to find — exactly or approximately — the full signal \( s[x_1, x_2] \). In Fig. 3.1 the non-regular sampling and its reconstruction is shown exemplary for a natural image. Of course, this task is inherently under-determined, and we require an additional prior on the original signal \( s[x_1, x_2] \) to inverse this problem.

Another way of describing the non-regular sampling is the vectorized form. In this thesis only mono channel images are primarily examined and thus the vector representation \( s \) of an image of size \( M_s \times N_s \) is defined by simply concatenating the columns of the image together:

\[
s_{x_1 + M_s \cdot x_2} = s[x_1, x_2] \iff s = \text{vectorize}(s[x_1, x_2]), \quad \text{for } x_1, x_2 \in \{0 \ldots M_s - 1\} \tag{3.2}
\]

where \( M_s \) is the number of rows of the image and \( s|_l \) addresses the \( l \)-th element of \( s \).
CHAPTER 3. NON-REGULAR SAMPLING AND ITS RECONSTRUCTION TO A REGULAR GRID

Figure 3.1: Example for the reconstruction of an image $s[x_1, x_2]$ from a subset of pixels $s_{nr}[x_1, x_2]$. (a): The pixels that are not available are set to zero/black. (b): Unavailable pixels are reconstructed utilizing prior information about images.

This correspondence between vectorized signal and non-vectorized signal is denoted, from now on, with the function $s = \text{vectorize}(s[x_1, x_2])$. Non-regular sampling can then be stated as the multiplication of a sub-sampling matrix $B$ with the vectorized image $s$, resulting in the available samples vector $s_{nr}$:

$$s_{nr} = Bs. \quad (3.3)$$

The sub-sampling matrix $B$ can be generated with the sub-sampling mask $b[x_1, x_2]$ by removing all rows containing only zeros from $\text{diag}(\text{vectorize}(b[x_1, x_2]))$. The expression $\text{diag}(b)$ is equivalent to the diagonal matrix with diagonal entries equal to $b$. Thus the rows of the matrix $B$ consist of randomly sub-sampled rows of the identity matrix and consequently the vector $s_{nr}$ contains only the available samples of positions, where the corresponding column in $B$ contains a one. Columns in $B$ that contain only zeros represent the samples that get lost during the sampling process. An example of a
sub-sampling matrix $B$ is shown in (3.4).

$$
B = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 
\end{pmatrix}
$$

(3.4)

3.2 Applications of Resampling from a Non-regular Grid

The problem of reconstructing images to a full grid from only a non-regular subset of pixels arises in many scenarios [73]. In light field acquisition systems, like the Optical Cluster Eye [59] or the Micro-Optical Artificial Compound Eyes [32] this is caused implicitly by the measurement system. Intentionally sampling images in a non-regular fashion can also be used to reduce the influence of aliasing, like introduced in [26] and argued for the use in acquiring seismic data in [44]. Another application is presented in [68] for increasing the spatial resolution of an imaging sensor, by covering three quarters of every sensor pixel non-regularly by an L-shaped mask of varying orientation. When reconstructing the missing three quarters of every pixel one can achieve an effective resolution four times that of the original underlying imaging sensor. Additionally, in some super-resolution algorithms [64] the final image is only available for a small number of pixel positions with respect to the high-resolution grid. To further process or display these sub-sampled images, it is usually necessary to reconstruct them to a regular grid. For this task, many algorithms have been proposed and some of the most important ones, using a specific prior information, are presented in the next section.
3.3 Sparsity Prior of Natural Images and Related Re-sampling Algorithms

Although there exist many reconstruction procedures, relying on different prior assumptions about the distribution of images, the focus in this thesis lies on algorithms utilizing a sparsity prior. A short overview of other reconstruction algorithms can be found in [73]. The concept of sparsity was already introduced in Chapter 2. A vector $x \in \mathbb{C}^P$ is declared sparse if most of its coefficients are equal to zero. Obviously natural images of size $N_s \times N_s$ are not sparse directly but usually exhibit a sparse representation over some frame (often called dictionary or sparsity matrix) $\Psi \in \mathbb{C}^{N^2_s \times P}$. A natural image $s$ can thus be expressed or at least approximated as a linear combination of a small amount of columns of the frame $\Psi$:

$$s = \Psi x,$$  \hspace{1cm} (3.5)

with $x$ being sparse. Popular analytic frames producing a sparse natural image representation are specific orthogonal bases including block-based discrete cosine transform, block-based discrete fourier transform, wavelets [55] or bandlets [49]. Useful frames consist of curvelets [75], contourlets [27] or the complex-valued dual-tree wavelet transform [47]. Concatenations of these dictionaries can additionally result in even more sparse representations for a broad range of natural images [43, 35]. Another class of dictionaries which have become popular in the recent years, are learned frames. The learned dictionary is constructed so that images in a training set have maximally sparse representations. When considering specific categories of images, these learned dictionaries can convincingly improve the performance of various image reconstruction tasks, relying on a sparsity prior [34, 85]. Exemplary Fig. 3.2 shows two decompositions of a natural image, one with a haar wavelet transform and another with the block-based discrete fourier transform. The concept of modeling natural images with sparse representations is fundamental and utilized in different areas of signal and image processing.
3.3. SPARSITY PRIOR AND RELATED RESAMPLING ALGORITHMS

like compression, denoising, inpainting, demosaicing, deconvolution, or super-resolution [83, 52, 36]. One application of these is the resampling of non-regularly sampled images to a regular grid, which can be seen as a special form of inpainting, where the goal is to find the best approximation to the original fully-sampled signal. Subsequently, a short overview of some proposed reconstruction algorithms, trying to solve this inverse problem, is given.

3.3.1 Description of Resampling Algorithms Using a Sparsity Prior

A popular approach is the reconstruction using a total variation minimization (TV) [20]. There, the image is assumed to be sparse in a gradient space. The basic idea is then to find a reconstruction that has a low total variation and is identical to the given samples. In [2] an efficient implementation of the TV minimization called Constrained Split Augmented Lagrangian Shrinkage Algorithm (CSALSA) is provided for the task of reconstructing an image from a subset of its pixels. Another algorithm, based on the sparsity property of image signals in the Wavelet domain, is the Wavelet Inpainting (WI) presented in [76]. Here, WI performs iterative hard thresholding, like presented in Section 2.2.4, with linearly decreasing threshold values to find a sparse model of the available samples in the Wavelet domain. As the sparse model reaches into the unavailable samples a reconstruction for the regular grid is found. The Morphological Component Analysis (MCA) approach from [35] also uses an iterative thresholding in a transformed domain, to find a sparse representation. Instead of hard thresholding, a soft thresholding function, as presented in Section 2.2.5, is used with constant step size \( \kappa_{\text{ist}} \) and linearly decreasing threshold value \( s_t \). Additionally a TV correction is applied after each thresholding step to generate a smooth image. The interesting aspect of the MCA is the use of a concatenated frame, consisting of a curvelet transform and a block-based Discrete Cosine Transform (DCT), to provide a convincing sparse representation for images containing cartoon parts and texture parts simultaneously. Another approach that can be used for resampling comes directly from the CS research area
CHAPTER 3. NON-REGULAR SAMPLING AND ITS RECONSTRUCTION TO A REGULAR GRID

Figure 3.2: Examples of sparse representations of a natural image from the Kodak database [40].
and is called block-based CS and smoothed projected Landweber reconstruction (BCS-SPL) [41]. It uses an iterative hard thresholding algorithm with specific thresholding and step size parameters and an additional interleaved smoothing update based on Wiener filtering [82] to find a sparse model for the whole image. Different sparsity domains are presented, including wavelets, complex dual-tree wavelets, contourlets and block-based DCT. In its proposed form, it is not considered for the resampling process, but rather for a classic CS reconstruction problem. With a small modification this algorithm can be reformulated to solve the task of reconstructing a regular grid from only a small subset of pixels, when the sub-sampling mask $B$ exhibits a block structure. An algorithm that applies the concept of learned dictionaries is shown in [53]. It is an extension of the single-scale KSVD algorithm from [34] and is denoted as MKSVD in this work. There, a learned dictionary is used on overlapping blocks with dictionary columns of different scales. Columns of smaller scales only represent part of the whole block and are zero-padded otherwise. The dictionary is initialized with an off-line learned dictionary and then adapted to the given samples of the non-regularly sampled image. MKSVD finds a fitting dictionary and a sparse reconstruction on this dictionary for the whole image by alternating between a sparse coding stage and a dictionary update. In the sparse coding stage, OMP (Section 2.2.2) is applied to find a sparse model that fits the available samples for each block. In the dictionary update, the atoms and sparse model coefficients are updated by the KSVD algorithm to better approximate the given samples in each block with only the few amount of atoms that were calculated in the previous step. In the end, a weighted average of the overlapping areas of the blocks is performed to reconstruct the image on the regular grid.

3.3.2 Description of the Frequency Selective Reconstruction

Finally the FSR algorithm, introduced in [73], showing state of the art results for the resampling task, uses the concept of sparse approximations in the block-based Fourier domain. As the main part of this work is to reveal connections between the CS framework and the FSR, this reconstruction procedure is now presented a little
more detailed than the other algorithms. It is an extension of the Frequency Selective Extrapolation (FSE) [58] which has been improved by several additions [70, 69, 71] and is used for various signal extrapolation tasks in the area of image and video signal processing.

The FSR is a block-based procedure, that splits the image into overlapping blocks and reconstructs them sequentially in a special processing order. The processing order is determined by the local sampling density of the sampling mask \( b(x_1, x_2) \). The sampling mask is therefore filtered with a 2D gaussian kernel \( d(x_1, x_2) \),

\[
\tilde{b}(x_1, x_2) = b(x_1, x_2) * d(x_1, x_2),
\]

and the result \( \tilde{b}(x_1, x_2) \) is summed up for each block. Blocks now get processed in decreasing order of the summation, by reconstruction them sequentially. This results in the advantage of being able to use the already extrapolated samples with higher confidence, due to the higher local sampling density, for the reconstruction of the following overlapping blocks. Within the sequential block reconstruction step, a sparse model of the block is calculated that fits the available samples, similar to the other resampling algorithms, by using a sparsity prior. A block of size \( N \times N \) at position \((p_0, p_1)\) is extracted from the whole image by

\[
f[m, n] = s[p_0 + m, p_1 + n], \quad \text{for} \ m,n \in 0 \ldots N - 1,
\]

the given non-regularly sampled block by

\[
f_{nr}[m, n] = s_{nr}[p_0 + m, p_1 + n], \quad \text{for} \ m,n \in 0 \ldots N - 1,
\]

and the sub-sampling mask block by

\[
q[m, n] = b[p_0 + m, p_1 + n], \quad \text{for} \ m,n \in 0 \ldots N - 1.
\]
The relationship between the full block $f[m, n]$ and the non-regularly sampled block is thus

$$f[m, n] = f[m, n] * q[m, n].$$  \hspace{1cm} (3.10)

To connect the FSR with the CS framework and comparing it to the different CS reconstruction algorithms, a vector notation of the FSR block reconstruction step is more suitable. However, this advantage is bought at the expense that this representation is not as clear and easy to understand as the formulation in [73], which uses the 2D representation. For a more detailed introduction to the FSR it is thus recommended to read the contribution from [73]. For the vectorized description, at first the inputs and outputs of the FSR are explained. The block $f[m, n]$ can be written in vectorized form by simply concatenating its columns using \texttt{vectorize()}, resulting in the fully sampled vector $f \in \mathbb{R}^{N^2}$. To write the available samples block $f_{nr}[m, n]$ in vector form, the columns are concatenated and the unavailable samples are removed, resulting in the available samples block vector $f_{nr} \in \mathbb{R}^{M}$. As unavailable samples are removed, $M$ is equal to $\sum_{\forall (m, n)} q[m, n]$. Again, the sub-sampling mask $q[m, n]$ — this time restricted to a block — can be represented with the block sub-sampling matrix $Q \in \mathbb{R}^{M \times N^2}$ by removing all rows, consisting of only zeros from \texttt{diag(vectorize(q[m, n]))}. Reconstructing this block can thus be described by inversing:

$$f_{nr} = Qf.$$  \hspace{1cm} (3.11)

The estimated model $g \in \mathbb{R}^{N^2}$ for the fully sampled block $f \in \mathbb{R}^{N^2}$ in the spatial domain that is calculated by the FSR is written from now on in dependence from its frequency domain coefficients $\hat{x}$, by

$$g = \Psi \hat{x}.$$  \hspace{1cm} (3.12)

Here $\Psi$ is the conjugate transpose of the 2D-DFT matrix, as presented around (2.31).
in Section 2.3, i.e.,

$$\Psi = (\psi_0, \psi_1, \ldots, \psi_{N^2-1})$$  \hspace{1cm} (3.13)

with the vectorized 2D-DFT basis functions

$$\psi_{a+N\cdot b} = \text{vectorize}(\psi_{a,b}[m,n]) \text{ for } a, b, m, n \in \{0 \ldots N-1\}.$$  \hspace{1cm} (3.14)

The 2D-DFT is therefore used as the dictionary/sparsity matrix in the FSR. The vector $\hat{x}$, consisting of estimated expansion coefficients, is to be determined in the block reconstruction by assuming that the original fully sampled block is sparse in the frequency domain, i.e., $f = \Psi \cdot x$, with $x \in \mathbb{R}^{N^2}$ being sparse. The complete FSR block reconstruction step is shown in Algorithm 4. As input we have the sub-sampling matrix $Q$, the available samples block vector $f_{nr}$ and some additional parameters that can be adjusted depending on the properties of the signal to be reconstructed. At first, we set a residual vector $r_{fsr,0}$ to the available samples block vector $f_{nr}$ and the estimated expansion coefficients $\hat{x}_0$ are initialized with zeros. The subscripts $i$ of the residual vector $r_{fsr,i}$ and the sparse model $\hat{x}_i$ indicates the state at iteration $i$. In lines 2–5 a spatial weighting matrix $W$ is calculated and normalized, that is later used to influence the impact of available samples in $f_{nr}$ on the model generation. To reduce the influence of already reconstructed pixels, but still use them for the reconstruction, reconstructed pixels are further weighted by the factor $\delta \in [0, 1]$. The normalization is not utilized in the original introduction of the FSR ([73]), but it does not change the output and simplifies the presentation of the algorithm. Subsequently in lines 6–9 a frequency weighting vector $w_f$ is generated, modeling a prior on the coefficients of the basis functions of the whole block $f$ [72]. This prior assures that low frequencies are preferred to high frequency components in the model, reflecting the distribution of the coefficients in natural images which is mainly due to the Optical Transfer Function of the diffraction limited imaging systems. High frequencies are suppressed, depending on the amount of available samples and their importance, as captured by the effective data variable $h$, which has shown significant improvement when compared to a constant
weighting independent to the amount of effective data [72]. Following the initialization, the estimation of the sparse model — \( \hat{x} \) in the frequency domain and \( g \) in the spatial domain — begins by executing greedy iterations similar to the MP algorithm (Algorithm 1). In each iteration the weighted correlation of the residual with all sub-sampled fourier basis functions \( Q\psi_l \) (\( \forall l \in \{0 \ldots N^2 - 1\} \)) is calculated and the basis function with the highest absolute correlation, weighted by the frequency prior, is selected as \( \psi_u \) (Line 11). When disregarding the frequency weighting, due to the spatial weighting with \( W \), the FSR picks the basis function, that could minimize the weighted residual energy \( E_w = (f_{nr} - Q\Psi\hat{x}_{i+1})^H W (f_{nr} - Q\Psi\hat{x}_{i+1}) \) the most, when added to the model. This basis function \( \psi_u \) is then added to the sparse model \( \hat{x}_i \) with a coefficient equal to \( \gamma(W^{\frac{1}{2}}Q\psi_u)^H(W^{\frac{1}{2}}r_{fsr,i}) \), i.e., the weighted correlation with the residual, multiplied by the orthogonality deficiency compensation factor \( \gamma \). The coefficients \( \hat{x}_i|_{u,c} \) corresponding to different basis functions are kept the same. Orthogonality deficiency compensation was introduced in [69] for the predecessor of the FSR and improves the algorithm by letting the quality of the model converge with increasing amount of iterations, instead of a declining quality if the iteration count \( t \) is set too high. At the end of each iteration, the residual needs to be updated by subtracting its weighted and orthogonality deficiency compensated projection onto the selected basis function from the previous residual. These iterations are applied \( t \) times and finally the estimated sparse expansion coefficients \( \hat{x} \) and the model in the spatial domain \( g = \Psi\hat{x} \) are returned. As the model \( g \), in its non-vectorized form \( g[m,n] \), extends into the unavailable samples, they can now be replaced in \( f_{nr}[m,n] \), resulting in the final reconstructed block. For this, only the real-valued part of the samples from the model is taken. Additionally a border of width \( \upsilon \) is disregarded in the final reconstructed block. The replaced pixels in \( f_{nr}[m,n] \), excluding the ones in the border, are then marked as reconstructed pixels from neighboring reconstruction and inserted into the full image \( s_{nr}[p_0 + m, p_1 + n] \) at the block position. These last two steps are not shown explicitly in Algorithm 4, as it is focused on the sparse model generation of the block. Following this, the next block from the determined processing order can be reconstructed, potentially using the reconstructed pixels from all previously reconstructed neighboring blocks for its model
CHAPTER 3. NON-REGULAR SAMPLING AND ITS RECONSTRUCTION TO A REGULAR GRID

generation. Due to the border of width $v$ each reconstructed block replaces separate pixels in the whole image, even though the blocks are overlapping. This is done until all blocks are reconstructed, finishing the FSR algorithm.

This chapter presented the problem of inversing non-regular sampling on a discrete grid — block-wise $f_{nr} = Qf$ or signal wide $s_{nr} = Bs$ — by using a sparsity prior. Natural images exhibit a sparse representation in a wide range of different dictionaries and reconstruction algorithms, utilizing this prior, have shown great success for different inverse image problems, including the resampling to a regular grid. A particular instance for resampling, providing state-of-the-art results, is the FSR, which will be linked to the CS framework in the next two chapters. The other algorithms (TV, WI, MCA, BCS-SPL and MKSVD) are again seen in Chapter 6, where a reconstruction quality comparison between all algorithms is done.
Algorithm 4 Frequency Selective Reconstruction, block reconstruction step

Input: Sub-sampling Matrix $Q$, available samples block vector $f_{nr}$, number of iterations $t$, decay factor $\hat{\rho}$, orthogonality deficiency compensation $\gamma$, weighting of already reconstructed areas $\delta$, frequency weighting factor $\beta$

Output: Sparse model $\hat{x}$ and $g$

Initialization:
1: $r_{fsr,0} := f_{nr}$, $\hat{x}_0 := 0$, $i := 0$

2: $\delta[m,n] := \left\{ \begin{array}{ll} 1 & \text{if pixel } f_{nr}[m,n] \text{ is not a reconstructed pixel from neighboring reconstruction} \\ \delta & \text{if pixel } f_{nr}[m,n] \text{ is a reconstructed pixel from neighboring reconstruction} \end{array} \right.$

3: $w[m,n] := \hat{\rho} \sqrt{\left( \frac{m-N}{2} \right)^2 + \left( \frac{n-N}{2} \right)^2} \cdot \delta[m,n]$

4: $w := \text{vectorize}(w[m,n])$

5: $W := \frac{\text{diag}(Qw)}{\|Qw\|_1} \cdot N^2$

6: $\delta := \text{vectorize}(\delta[m,n])$

7: $h := \frac{\sum_{l=0}^{N^2-1} (Qw)_l |(Qw)_l|}{\sum_{l=0}^{N^2-1} w_l}^{\frac{\log(h)}{\beta}}$

8: $w_f[a,b] := \left( 1 - \sqrt{2 \left( \frac{N}{N^2} - |a| \frac{N}{N^2} \right) + \frac{N}{N^2} - |b| \frac{N}{N^2} } \right)^{-\frac{\log(h)}{\beta}}$

9: $w_f := \frac{\text{vectorize}(w_f[a,b])}{\|\text{vectorize}(w_f[a,b])\|_1} \cdot N^2$

Iterations:
10: while $i < t$ do

11: $u := \text{arg max}_{l \in \{0,...,N^2-1\}} |(W^\frac{1}{2}Q\psi_l)^H(W^\frac{1}{2}r_{fsr,i})| \cdot w_f[l]$

12: $\hat{x}_{i+1}|_u := \hat{x}_i|_u + \gamma (W^\frac{1}{2}Q\psi_u)^H(W^\frac{1}{2}r_{fsr,i})$

13: $\hat{x}_{i+1}|_{u^C} := \hat{x}_i|_{u^C}$

14: $r_{fsr,i+1} := r_{fsr,i} - \gamma (W^\frac{1}{2}Q\psi_u)^H(W^\frac{1}{2}r_{fsr,i})Q\psi_u$

15: $i := i + 1$

16: end while

17: return $\hat{x} := \hat{x}_t$, $g := \Psi \hat{x}$
Chapter 4

Connection between Frequency Selective Reconstruction and the Compressed Sensing Framework

Most of the algorithms for resampling images from a non-regular subset of pixels to a regular grid were developed completely separate to the CS framework. Only BCS-SPL originates directly from the CS domain and WI is slightly connected to CS in the book from Starck [76]. Nevertheless, the algorithms show a strong connection to CS, as they try to solve a special form of the CS linear inverse problem by using the sparsity prior. Specifically the reconstruction algorithm FSR, as presented in Section 3.3.2, that uses a block-based discrete fourier transform to achieve a sparse representation of image signals, can be linked to the description of the classical CS problem and its reconstruction. However, to achieve additional reconstruction quality, further image priors are exploited within the FSR, introducing differences between the two concepts. This chapter tries to reveal the similarities and differences of FSR and the CS framework. Moreover, it gives insights on the applicability of the theoretical CS reconstruction guarantees for the task of non-regular sampling and the reconstruction to a regular grid.
4.1 Describing Non-regular Sampling as a Linear System of Equations

In Chapter 3, the process of sampling a signal at only a small subset of positions was already described with

$$s_{nr} = Bs,$$  \hspace{1cm} (4.1)

for the signal wide case and with

$$f_{nr} = Qf,$$  \hspace{1cm} (4.2)

for the block-wise case. All notations used here and in the following were introduced in Chapter 3 and Chapter 2 and are therefore not repeated in detail again. Due to the focus on the FSR algorithm, which works on image blocks, only the block-wise case is regarded in the following. By representing $f$ in a different domain, i.e., $f = \Psi x$, (4.2) becomes

$$f_{nr} = Q\Psi x,$$  \hspace{1cm} (4.3)

where $\Psi$ is a specific frame and $x$ is the corresponding coefficient vector. As $Q$ is element of $\mathbb{R}^{M \times N^2}$, with $M \ll N^2$, this is equal to the “compressed” linear measurement of $x$ with the measurement matrix

$$\Phi = Q\Psi,$$  \hspace{1cm} (4.4)

like in the CS framework. Here $Q$ can be seen as the spatial measurement matrix and $\Psi$ as the sparsity matrix. In Section 4.1, this correspondence is shown exemplary for a specific sub-sampling matrix $Q$, the transform matrix $\Psi$ being the 2D-DFT and a coefficient vector $x$, taken from a real image block. When assuming a sparsity prior on
the vector $\mathbf{x}$, i.e., it can be approximated by only a small number of coefficients, CS reconstruction algorithms and their guarantees seem to be applicable for the inversion of the otherwise under-determined system.
4.1. DESCRIBING NON-REGULAR SAMPLING AS A LINEAR SYSTEM OF EQUATIONS

\[ f_{nr} = Q \cdot \text{real}(\Psi^H) \cdot |x| \]

(a) separate sub-sampling matrix and conjugate transposed sparsity matrix

\[ f_{nr} = \text{real}(\Phi) \cdot |x| \]

(b) unified measurement matrix

Figure 4.1: Example for the correspondence between the sub-sampled image \( f_{nr} \) and the coefficients of the fully sampled image in the frequency domain \( x \), resulting in an under-determined system of linear equations (4.3). Black values correspond to zeros (or low negative values in \( \text{real}(\Psi^H) \) or \( \text{real}(\Phi) \)), scaling up to yellow values, indicating high coefficients. The vector \( x \) can be approximated with only a small number of coefficients, as it contains some dominant values.
4.2 Connection between Non-regular Sampling and Compressed Sensing

To consider the CS reconstruction algorithms useful for the task of reconstructing a non-regularly sampled image to a full grid, two aspects need to be fulfilled. First, the image needs to exhibit a sparse approximation with respect to a specific sparsity matrix $\Psi$. Secondly, the “measurement” system, consisting of $\Phi = Q\Psi$, needs to fulfill the properties described in Section 2.3.

Regarding the first point, there exist many transforms leading to a sparse representation of image signals. The different useful transforms were already stated in Section 3.3. Together with the second point, the selection of a suitable transform is additionally dependent on being mutually incoherent with the spatial measurement matrix $Q$. Recalling the important aspects presented in Chapter 2, of the matrices achieving a sparse representation for natural image signals, especially the DFT provides efficient guarantees on the recovery error with CS reconstruction algorithms in this setting. The reason for this is explained in the following. The sub-sampling matrix $Q$ is generated by selecting $M$ rows from the identity matrix uniformly at random. Therefore, the identity matrix corresponds to the expanded spatial measurement matrix $P$ (Section 2.3) in the case of non-regular sampling. Now, the DFT transform, regardless of its dimension, exhibits ideal mutual coherence (2.33) $- \mu_c = 1 -$ with respect to the identity matrix. Thus, the whole measurement matrix $\Phi$ can achieve useful RIP-constants $\delta_k$, when the number of rows $M$ in the random matrix $Q$ is in the order of only $M \geq O(k \cdot (\log(N))^4)$ (2.28). Due to the time-frequency duality of the Fourier transform, it does not matter whether one takes the DFT matrix or its inverse as the sparsity matrix. The sparsity of the coefficients $x$ does not change and neither does the mutual coherence. Using the DFT transform or its inverse as the sparsity matrix and the sub-sampling matrix $Q$ as the spatial measurement matrix is also equal to a “measurement” system with a partial Fourier matrix. Partial Fourier matrices were one of the first matrices,
which were regarded as suitable for the measurement process in CS [19], due to their small amount of required measurements, as stated above. Additionally the Fast Fourier Transform (FFT) allows application of the transform in $\mathcal{O}(N \log(N))$, while requiring low memory storage. As will be seen later, the possibility to apply the DFT or its inverse with low computational complexity can be exploited, providing fast CS and non-regular sampling reconstruction algorithms.

Together with the uniform instance optimality in (2.24), the utilization of the DFT as the sparsity matrix, leads to an efficient recovery error, when some of the CS reconstruction algorithms are used for inversing the non-regular sampling problem in (4.3). Because of the linear scaling of the required measurements $M$ for partial Fourier matrices with respect to $k$, one can nearly reconstruct the, in the order of $M$, highest Fourier coefficients $x$ of the fully sampled image block $f$, by applying IHT$_k$, SP or BPDN to (4.3). As the Fourier coefficients $x$ of an image block usually decay quickly, this best $\mathcal{O}(M)$-sparse approximation of $x$ is close to $x$, even if $\mathcal{O}(M)$ is small, and therefore these reconstruction procedures promise a low reconstruction error $\|x - \hat{x}\|_2$. Here $\hat{x}$ is again the output of the CS reconstruction algorithms. Of course, only in the order of $M$, i.e., $\mathcal{O}(M)$, coefficients can be nearly reconstructed and not necessarily exactly $M$. Additionally, the resulting approximation error of the best $\mathcal{O}(M)$-sparse approximation can again be scaled by a specific constant to result in the final reconstruction error. Assuming no quantization or measurement errors, to gain an explicit mean upper bound on the reconstruction error for a given $M$, one would first need to insert the concrete constants into the equation connecting RIP-constant and required number of measurements (e.g. (2.27)) and then solve for the maximum $k$ providing a RIP-constant $\delta_{2k} < \sqrt{2} - 1$, with high probability. Given this $k$, one needs to establish a probability distribution of the best $k$-sparse approximation errors for image blocks in the Fourier domain. This could be done by evaluating a large number of natural image blocks and estimating a distribution from the histogram of best $k$-sparse approximation errors. Now the expected value of the algorithm specific uniform instance optimality can be calculated by inserting the specific constants and using the established probability
On another note, the recovery guarantees concerning deterministic matrices can be analyzed, even though they can not give as strong guarantees as the probabilistic approach above. In [73] (III. A.) the effect of non-regular sampling on the frequency spectrum of the image \( f[m,n] \) is presented. The multiplication of the fully sampled block \( f[m,n] \) with the sub-sampling mask block \( q[m,n] \) in the spatial domain results in a 2D circular convolution (denoted by \( \odot \)) of the corresponding spectra in the frequency domain:

\[
f_{rn}[m,n] = f[m,n]q[m,n] \tag{4.5}
\]

\[
F_{rn}[a,b] = \frac{1}{N^2} F[a,b] \odot Q[a,b]. \tag{4.6}
\]

Here \( F_{rn}[a,b], F[a,b] \) and \( Q[a,b] \) indicate the 2D-DFT transformed signals of \( f_{rn}[m,n] \), \( f[m,n] \) and \( q[m,n] \), respectively, and the 2D-DFT \( \mathcal{F}_{a,b} \) of a signal \( x[m,n] \) is given by

\[
\mathcal{F}_{a,b}\{x[m,n]\} = X[a,b] = \sum_{\forall(m,n)} x[m,n] \psi_{a,b}^* \tag{4.7}
\]

Looking at the spectra of sub-sampling masks with non-regular structure, one can observe that it usually shows a dominant peak at frequency \((0,0)\), while the other frequencies are comparatively small, but often non-zero. This is in contrast to a regular sampling pattern, where dominant peaks exist at multiples of the sampling rate, while all other frequencies are zero, which would lead to the classical aliasing effect. In Fig. 4.2 a comparison between the absolute value of the DFT \( |Q[a,b]| \) of a regular and non-regular sub-sampling mask is shown. The authors in [73] argue, that the dominant frequencies in the fully sampled images can survive the non-regular sub-sampling process, as they only get small contributions (because of the small frequencies in \( Q[a,b] \) for \( a \neq 0 \) and \( b \neq 0 \)) from a low amount (because of the sparsity of \( F[a,b] \)) of other fre-
4.2. CONNECTION BETWEEN NON-REGULAR SAMPLING AND COMPRESSED SENSING

Thus the dominant frequencies in $F[a, b]$ remain dominant in $F_{nr}[a, b]$ and can therefore be extracted. In the CS framework this consideration can be interpreted with the concept of coherence (not to be mistaken with mutual coherence). The coherence $\mu(\Phi)$ of a matrix $\Phi$ was already defined in (2.3) as:

$$\mu(\Phi) = \max_{0 \leq i \neq j \leq N-1} \frac{\left| \phi_i \phi_j^H \right|}{\left\| \phi_i \right\|_2 \left\| \phi_j \right\|_2},$$

(4.8)

where $\phi_i$ denotes the $i$-th column of $\Phi$. In our case, $\Phi$ is a matrix consisting of a non-regular subset of rows from the conjugate transpose of the 2D-DFT matrix, i.e., it is a specific instance of a partial Fourier matrix. The contributions from different frequencies, present in the fully sampled signal, which occur due to the circular convolution with the Fourier transform $Q[a, b]$ of the sub-sampling mask, can be connected regarding the following. At first the columns of $\Phi$ can be presented in the non-vectorized form again. The resulting functions are simply the sub-sampled 2D-Fourier basis functions $\phi_{a,b}[m, n]q[m, n]$. Now, the coherence can be formulated as

$$\mu(\Phi) = \max_{a_1, b_1, a_2, b_2 \in \{0 \ldots N-1\}} \frac{N^2}{M} \sum_{m, n} \phi_{a_1, b_1}[m, n]q[m, n]\phi_{a_2, b_2}^*[m, n]q^*[m, n]$$

$$= \max_{a_1, b_1, a_2, b_2 \in \{0 \ldots N-1\}} \frac{N^2}{M} |\mathcal{F}_{a_2, b_2} \{ \phi_{a_1, b_1}[m, n]q[m, n] \}|$$

$$= \max_{a_1, b_1, a_2, b_2 \in \{0 \ldots N-1\}} \frac{1}{M} |\delta_{a_1, b_1}[a_2, b_2] \odot Q[a_2, b_2]|$$

$$= \max_{a_1, b_1, a_2, b_2 \in \{0 \ldots N-1\}, (a_1, b_1) \neq (a_2, b_2)} \frac{1}{M} |Q[(a_2 - a_1) \mod N, (b_2 - b_1) \mod N]|$$

$$= \max_{a, b \in \{0 \ldots N-1\}} \frac{1}{M} |Q[a, b]|.$$

(4.9)

Here, $\delta_{a_1, b_1}[a_2, b_2]$ is the delta impulse:

$$\delta_{a_1, b_1}[a_2, b_2] = \begin{cases} 1 & \text{if } a_1 = a_2 \land b_1 = b_2 \\ 0 & \text{otherwise.} \end{cases}$$

(4.10)
Thus, the coherence of the measurement system in deterministic non-regular sampling with a DFT as the sparsity matrix, is proportional to the maximum absolute value of the spectrum $Q[a, b]$ of the sub-sampling mask for frequencies unequal to $(0, 0)$. As the energy of the non-regular sampling mask (2.4) is spread out onto the whole frequency spectrum when disregarding frequency $(0, 0)$, the coherence is usually small. In CS a low coherence is a desirable property of the measurement matrix, as it provides a high possible sparsity $k$ (2.4) for which perfect recovery is achieved in the case of strictly sparse signal. Additionally, it guarantees a low upper bound on the RIP-constant $\delta_{km}$ shown in (2.7), which subsequently allows a reconstruction with uniform instance optimal recovery error for large $k$. This is in contrast to regular sampling, where the high coherence does not guarantee an effective reconstruction.

It has to be noted, that the theoretical bounds on the reconstruction error are only worst case bounds and thus the average reconstruction error across all image blocks could be much lower. Moreover, (2.24) is only shown for a small subset of reconstruction algorithms and, especially, not for the MP algorithm, which shows similarities to the examined FSR algorithm. Furthermore, the FSR has some modifications, separating it from the traditional CS approach by introducing the use of additional prior information. Due to these reasons, a more detailed analysis of concrete upper bounds on the reconstruction error is not examined further. Regardless of these limitations, as the involved constants are well behaved, the CS framework shows the existence of an efficient recovery of non-regularly sampled images and even provides algorithms for the task of resampling images to a regular grid when using the DFT as the sparsity matrix. If the CS algorithms, presented in Section 2.2, can compete with current state-of-the-art algorithms for non-regularly sampled image reconstruction, is analyzed in the next chapter. Moreover, as there is a strong connection between the problem of non-regular sampling and conventional CS sensing and reconstruction, ideas originating from the two frameworks could potentially be exchanged. For this the FSR is compared to existing CS approaches in the following section.
4.2. CONNECTION BETWEEN NON-REGULAR SAMPLING AND COMPRESSED SENSING

Figure 4.2: DFT coefficients of different sub-sampling masks with 50% subsampling density. Blue colors indicate high frequencies, while yellow colors correspond to low frequencies. The non-regular sub-sampling mask shows a noise like distribution for non-zero coefficients, while the regular sampling mask exhibits a peak at the regular sampling frequency. The non-regular sampling thus leads to a low coherence of the measurement system.
CHAPTER 4. CONNECTION BETWEEN FREQUENCY SELECTIVE RECONSTRUCTION AND THE COMPRESSED SENSING FRAMEWORK

4.3 Comparison of the FSR to existing CS reconstruction algorithms

To discover ideas from CS reconstruction algorithms for improving the FSR or vice versa, a comparison of different approaches developed within the CS framework and the FSR is given in the following. In Section 3.3.2 the FSR algorithm (Algorithm 4), for reconstructing images to a full grid from a non-regular subset of pixels, was introduced. There, it was already hinted that the algorithm shows similarities to the classical MP reconstruction procedure. To connect these two algorithms, the inputs need to be specified. As input, MP (Algorithm 1) takes the measurement matrix $\Phi$, which in the basic case of non-regular sampling is set to $\Phi = Q \Psi$. Additionally the measurement vector input $y$ is set to the non-regularly sampled block $y = f_{nr}$. To simplify the explanation, $\Phi$ is furthermore normalized by multiplying it with a specific factor, such that all columns have unit norm and the measurement vector $y$ is also multiplied by the same factor, so that the linear equation stays consistent. Without incorporating any weighting, which will be explained later, the measurement matrix $\Phi$ can be normalized by multiplying it with $\frac{N}{\sqrt{M}}$, resulting in the actual measurement matrix $\Phi_{\text{norm}}$. This normalized version is taken as input to the MP, together with the rescaled measurement vector. The inputs of the FSR are now regarded as $Q$ for the sub-sampling matrix and $f_{nr}$ as the available samples block vector. Comparing the iterations of the MP algorithms to the iterations of the FSR (lines 10 – 16 in Algorithm 4), they are exactly equivalent, when the spatial weighting $w[m,n]$ is all ones, the frequency weighting vector $w_f$ is all ones and the orthogonality deficiency compensation $\gamma$ is 1. This can be achieved by setting the input parameters of the FSR correspondingly, i.e., $\hat{\rho} = 1$, $\gamma = 1$, $\delta = 1$ and a high value for $\beta$. In the following a short explanation for this is given.

When setting the parameters of the FSR as described above, the columns of the measurement matrix $\phi_{\text{norm,l}}$ are equivalent to the sub-sampled and weighted functions
4.3. COMPARISON OF FSR AND CS RECONSTRUCTION ALGORITHMS

of the DFT:

\[ \phi_{\text{norm},l} = W^{\frac{1}{2}} Q \psi_l, \quad \text{for all } l \in \{0 \ldots N^2 - 1\}; \quad (4.11) \]

as the weighting \( W^{\frac{1}{2}} \) in the FSR is only a normalizing diagonal matrix:

\[ W^{\frac{1}{2}} = \text{diag}(\sqrt{\frac{N^2}{M}}) = \text{diag}(\frac{N}{\sqrt{M}}), \quad (4.12) \]

when \( \hat{\rho} = 1 \) and \( \delta = 1 \). Additionally the residuals \( r_i \) and \( r_{\text{fsr},i} \) in the two algorithms differ by just a multiplication with the normalization constant \( \frac{N}{\sqrt{M}} \):

\[ r_i = \frac{N}{\sqrt{M}} r_{\text{fsr},i}, \quad \text{for all } i. \quad (4.13) \]

Later the proof of this is shown in a more general form. When replacing \( \phi_l \) (which is \( \phi_{\text{norm},l} \) in this case) and \( r_i \) in MP (Algorithm 1) with the two just mentioned relations ((4.11) and (4.13)), the equivalence of MP and FSR can easily be seen for \( \hat{\rho} = 1, \gamma = 1, \delta = 1 \) and a high valued \( \beta \). Thus, the MP algorithm is a specific instance of the FSR algorithm and the FSR can be seen as a generalization of the simple MP algorithm. The extensions, provided by the FSR, are the spatial weighting, the orthogonality deficiency compensation and the frequency prior.

These additional extensions allow for an incorporation of prior information of natural images that are different to the sparsity prior. One of these is the spatial weighting, which, in combination with overlapping blocks, allows for an efficient weighting of the available samples for the model generation. Pixels closer to the reconstruction area are weighted stronger, due to the higher mutual information with the unknown samples to be reconstructed. When setting \( \hat{\rho} \neq 1 \), the spatial weighting influences the model generation and the basic MP is not equal to the FSR anymore. The FSR then selects the basis function and its coefficient which minimize the weighted residual energy

\[ E_w = (f_{\text{nr}} - Q \Psi \hat{x}_{i+1})^H W (f_{\text{nr}} - Q \Psi \hat{x}_{i+1}) \quad (4.14) \]
the most, when disregarding the frequency prior \( w_f = 1 \) and orthogonality deficiency compensation \( (\gamma = 1) \). In contrast, MP selects the basis function and its coefficient, which reduces the unweighted residual energy

\[
E = (f_{nr} - Q\Psi\hat{x}_{i+1})^H(f_{nr} - Q\Psi\hat{x}_{i+1})
\]

(4.15)

the most. This difference can be removed, by simply adjusting the inputs of the MP algorithm. With disabled frequency prior and orthogonality deficiency compensation \( \gamma = 1 \), FSR is equal to MP when taking the weighted measurement matrix

\[
\Phi_{\text{weight}} = W^{\frac{1}{2}}Q\Psi
\]

(4.16)

and weighted measurement vector

\[
y_{\text{weight}} = W^{\frac{1}{2}}f_{nr}
\]

(4.17)

as input to MP, even if \( W^{\frac{1}{2}} \) is not only a normalization matrix, but an arbitrary diagonal matrix. This can be seen with the following consideration. Similar to the case when using a normalizing weighting matrix, (4.16) states that the columns of the weighted measurement matrix \( \phi_{\text{weight},l} \) are equivalent to the weighted and sub-sampled DFT basis functions:

\[
\phi_{\text{weight},l} = W^{\frac{1}{2}}Q\psi_l, \quad \text{for all } l \in \{0 \ldots N^2 - 1 \}.
\]

(4.18)

Additionally, the residual in MP is equal to the weighted residual of the FSR, i.e.,

\[
r_i = W^{\frac{1}{2}}r_{\text{fsr},i}, \quad \text{for all } i \in \{0 \ldots t \}.
\]

(4.19)

Equation (4.19), and also (4.13) (as it is only a special instance with equal normalizing entries in \( W^{\frac{1}{2}} \)) can be shown by a simple mathematical induction:
4.3. COMPARISON OF FSR AND CS RECONSTRUCTION ALGORITHMS

Proof. In the base case, \( i = 0 \), the residuals are defined in the initialization of the two algorithms as:

\[
\begin{align*}
\mathbf{r}_{\text{fsr},0} &= \mathbf{f}_{\text{nr}}, \quad (4.20) \\
\mathbf{r}_0 &= \mathbf{y}_{\text{weight}} = W_{12} \mathbf{f}_{\text{nr}} = W_{12} \mathbf{r}_{\text{fsr},0}. \quad (4.21)
\end{align*}
\]

Thus (4.19) holds for the base case. Inserting the induction hypothesis (4.19) and correspondence (4.18) into line 6 of MP proofs the induction step:

\[
\begin{align*}
\mathbf{r}_{i+1} &\overset{\text{line 6 MP}}{=} \mathbf{r}_i - \phi_{\text{weight},u}^H \mathbf{r}_i \phi_{\text{weight},u} \quad (4.22) \\
&\overset{(4.19)}{=} W_{12} \mathbf{r}_{\text{fsr},i} - (W_{12} Q_{\psi_u}^H) W_{12} \mathbf{r}_{\text{fsr},i} (W_{12} Q_{\psi_u}) \quad (4.23) \\
&= W_{12} \mathbf{r}_{\text{fsr},i} - W_{12} (W_{12} Q_{\psi_u}^H) W_{12} \mathbf{r}_{\text{fsr},i} Q_{\psi_u} \quad (4.24) \\
&= W_{12} (\mathbf{r}_{\text{fsr},i} - (W_{12} Q_{\psi_u}) W_{12} \mathbf{r}_{\text{fsr},i} Q_{\psi_u}) \quad (4.25) \\
&\overset{\text{line 14 FSR}}{=} W_{12} \mathbf{r}_{\text{fsr},i+1} \quad (4.26)
\end{align*}
\]

Now, inserting (4.18) and (4.19) into the MP algorithm shows the equivalence between FSR and MP when using the weighted measurement matrix and weighted measurement vector as inputs to MP, while disregarding the frequency prior and orthogonality deficiency compensation. In Algorithm 5 the rewritten MP with these specific inputs is presented. Lines 3 – 5 in MP are now equal to lines 11 – 13 in the FSR. As the selected basis function index \( u \) and the update to the sparse model \( \hat{\mathbf{x}}_{i+1} \) is equivalent in both algorithms for all \( i \), the output sparse model \( \hat{\mathbf{x}} \) will be the same. The idea of using
a weighted measurement matrix and weighted measurement vector as inputs can be straightforwardly extended to other CS reconstruction algorithms. As the utilization of a spatial weighting in the FSR increases the reconstruction quality significantly [73], this concept could prove itself successful when incorporated into other CS reconstruction algorithms. In Chapter 5 an experimental analysis concerning this modification is given.

Algorithm 5 Matching Pursuit with weighted inputs

**Input:** Measurement Matrix \( \Phi_{\text{weight}} = W^\frac{1}{2}Q\Psi \), measurement vector \( y_{\text{weight}} = W^\frac{1}{2}f_{nr} \), number of iterations \( t \)

**Output:** Sparse model \( \hat{x} \)

1: Initialization: \( r_0 := y_{\text{weight}} = W^\frac{1}{2}r_{fsr,0}, \hat{x}_0 := 0, i := 0 \)

2: while \( i < t \) do

3: \( u := \arg \max_{l \in \{0...N-1\}} |\phi_{\text{weight},l}^H r_i| \) \((4.19)\) \((4.18)\)

4: \( \hat{x}_{i+1}|_u := \hat{x}_i|_u + \phi_u^H r_i = \hat{x}_i|_u + (W^\frac{1}{2}Q\psi_u)^H W^\frac{1}{2}r_{fsr,i} \)

5: \( \hat{x}_{i+1}|^c := \hat{x}_{i+1}|^c \)

6: \( r_{i+1} := r_i - \phi_u^H r_i \phi_u \) \((4.19)\)

7: \( i := i + 1 \)

8: end while

9: return \( \hat{x} := \hat{x}_t \)

The considerations above assume that no orthogonality deficiency compensation is used in FSR by setting \( \gamma = 1 \). As already stated, the orthogonality deficiency compensation, applied in line 12 of FSR, improves the algorithm by letting the quality of the model converge with increasing amount of iterations, instead of a declining quality if the iteration count \( t \) is set too high. Within FSR, the orthogonality deficiency compensation factor \( \gamma \) is usually set around 0.5, so that in each iteration only a part of the basis function that would maximally decrease the weighted residual energy is added to the model. This modification improves the quality drastically, as a high iteration count can be used as the sole stopping criterion, reaching a reconstruction quality that is equal to the one with optimal iteration count, when using no orthogonality deficiency compensation [69]. It is highly likely, that the MP, modified with orthogonality defi-
4.3. COMPARISON OF FSR AND CS RECONSTRUCTION ALGORITHMS

iciency compensation, can be applied for different CS reconstruction tasks and not only for non-regular sampling, when the sparsity level is not known in advance.

Another extension of the FSR compared to the basic MP algorithm is the frequency prior. The frequency prior in the form of $w_f$ incorporates information about the distribution of the sparse coefficients by preferring low frequencies within the greedy iterations. The idea of using prior knowledge about the distribution of the non-zero coefficients is also examined in CS related literature. In [46] a weighted $l_1$-minimization recovery algorithm is used to reconstruct signals where each coefficient of the sparse vector to be measured has a known independent probability of being non-zero. The probabilities are restricted to two categories, i.e., some coefficients have probability $P_1$ of being non-zero, while all other coefficients have probability $P_2$ of being non-zero. Simulations demonstrate the superiority of this concept compared to conventional $l_1$-optimization and theoretical guarantees on the reconstruction success are given [46]. The restriction to only two distinct probabilities and the $l_1$-based approach differs from the concept used in FSR, which uses a greedy approach with different weights for each projection coefficient. Another proposal is presented in [21], where a so called oblique projection is utilized in the OMP instead of the conventional projection columns of the measurement matrix $\Phi$. For the oblique projection, a different matrix $Y$ is used to select the next basis function and its coefficient, which allows to incorporate prior information about the distribution of the non-zero coefficients and the form of the measurement matrix. As the columns of $Y$ have unit norm in the presented approach and the OMP algorithm is used, this concept is different to the frequency weighting scheme with $w_f$ in the FSR. An additional concept from CS to exploit \textit{a priori} information about the distribution of the non-zeros in the unknown vector is the \textit{union of subspaces} model [50]. Here the signal $x$ is assumed to reside in a union of subspaces:

$$x \in \bigcup_{i=0}^{I-1} U_i, \quad (4.27)$$

where $U_i$, for $i \in \{0 \ldots I - 1\}$, are $k$-dimensional subspaces linked to a specific set of $k$
non-zero coefficients. Thus \( \mathbf{x} \) might not be in the complete \( k \)-sparse set \( \Sigma_k \), but rather only some combinations of non-zero positions are allowed. As the image prior used in FSR allows all combinations, but weights them differently, this concept can not directly be used in the FSR.

Finally, the idea of splitting the image into blocks is also examined in CS literature. In [41] and [61] the measurement matrix \( \Phi \) is applied to the whole image and is assumed to have a block structure, so that each image block can be sensed independently with the same block measurement matrix. Thus, in contrast to FSR, the sensing can be a general matrix, but it is restricted to be the same for each block. The reconstruction is then applied on the whole image using a smoothed variant of IHT, where a Wiener filtering step is interleaved into the hard thresholding iterations, incorporating a smoothness prior. In contrast, FSR processes overlapping blocks sequentially. There also exist CS reconstruction algorithms which utilize overlapping blocks, with the purpose to reduce edge artifacts. In [57] the image is split into equally sized overlapping blocks and each block is sensed and reconstructed independently. The final image is then obtained by simply averaging the overlapping areas. Contrary to this, FSR reuses already reconstructed samples of the image in subsequent block reconstructions, which is possible due to the special form of the sensing matrix in non-regular sampling. Moreover, FSR only uses the center of each reconstructed block for replacing the unknown values in the image block and the processing order is optimized in the FSR by making it adaptive to the local density of the sampled pixels.

In summary it can be said that the FSR is related to the MP algorithm, but utilizes additional priors besides the sparsity information. There exist many approaches in the CS literature to incorporate further prior information into the reconstruction process. The additional modifications, i.e., orthogonality deficiency compensation, spatial weighting and frequency prior, of FSR can not be directly found in CS literature and thus it might be beneficial to bring ideas from the FSR into existing CS reconstruction algorithms. Inversely, concepts like the smoothing with a Wiener filter as an interme-
4.3. COMPARISON OF FSR AND CS RECONSTRUCTION ALGORITHMS

diate step and usage of different CS reconstruction algorithms, like OMP, IHT\(_k\), IHT, IST, SP or BPDN as the basis reconstruction algorithm instead of MP can be included into FSR. As these algorithms often outperform MP and Section 4.2 indicated the applicability of CS algorithms for non-regular sampling, the idea to use different CS reconstruction algorithms inside the block reconstruction step of the FSR is analyzed experimentally in the next section. Furthermore, the concept of spatial weighting and frequency prior are incorporated into these CS procedures, showing potential use of the concepts, developed independently to the CS framework, of the FSR for other CS reconstruction tasks.
Chapter 5

Using Compressed Sensing Reconstruction Algorithms within the Frequency Selective Reconstruction

In Chapter 4 the suitability of classical CS reconstruction algorithms for reconstructing non-regularly sampled natural images was indicated. This is due to the possibly small RIP constants, even for large $k$, when using the DFT as a sparsity matrix and the additional sparse representation of image blocks in the DFT domain. Sparse modeling approaches, with algorithms similar to the CS reconstruction algorithms, already showed groundbreaking performances for the resampling task. In Section 3.3.1 and Section 3.3.2 these were already presented. MCA, WI, MKSVD and FSR use different base CS reconstruction algorithms with additional modifications and also distinct sparsity dictionaries to generate a sparse model fitting the available samples. As the CS reconstruction algorithms have different reconstruction performances among each other, depending on the amplitude distribution of the non-zero coefficients of the sparse vector and also the measurement matrix [54, 77, 6], an exploration, which CS algorithm is suited best for resampling non-regularly sampled images to a regular grid is analyzed.
5.1 Incorporation of Spatial Weighting and Frequency Prior into Compressed Sensing Algorithms

In this work. No exploration considering a partial Fourier matrix as measurement system and natural image blocks in the DFT domain as sparse vectors exists in literature yet. Unfortunately, it seems that the sparsity prior is not enough to achieve optimal results. FSR (with spatial weighting and frequency prior), MCA (with TV regularization) and also BCS-SPL (with interleaved Wiener filtering) use additional priors of the image block to improve reconstruction quality. Therefore the most important modifications of the FSR, namely the spatial weighting and also partly the frequency prior, are incorporated into the different CS reconstruction algorithms.

In this chapter, at first, the modifications, concerning the spatial weighting and the frequency prior are presented for all algorithms (Section 5.1). This is followed in Section 5.2 by a detailed description of the concrete implementation of the CS reconstruction algorithms used in this work, which mainly deals with the determination of the different parameters. As some parameters are still open and values from literature lead to unsatisfying results, a parameter exploration on a training set is done subsequently in Section 5.3. Thereafter, to show the correct implementation of the algorithms, an artificial test set, which exhibits a strict sparsity, is created and reconstructed with the determined parameters in Section 5.4. In the end of this chapter (Section 5.5), a reconstruction quality comparison on natural image blocks is given experimentally for all reconstruction algorithms, additionally showing the benefits of the modifications, like spatial weighting and the frequency prior.

5.1 Incorporation of Spatial Weighting and Frequency Prior into Compressed Sensing Algorithms

To improve the reconstruction performance of classic CS reconstruction algorithms, when applied to invert the non-regular sampling problem (4.3), two different weighting schemes, similar to the ones used in the FSR, are presented. One is the introduction of a spatial weighting function and the second one is the addition of a frequency prior.
into the thresholding based CS algorithms.

5.1.1 Spatial Weighting

As it was seen in Section 4.3, the spatial weighting function within the FSR leads to a greedy minimization of the weighted residual energy in each iteration, instead of the unweighted residual energy in MP. By simply changing the input to MP, the MP iterations became equal to the FSR iterations, when only using a spatial weighting and disabling all other features (orthogonality deficiency compensation and frequency prior) of the FSR. This leads to the generation of a sparse model \( \hat{x} \), which approximates the highly weighted samples in the measurement vector \( f_{nr} \) better than the lowly weighted samples. As only a small center part of the reconstructed block is used for replacement of unknown pixels in the sub-sampled image, pixels that are on the border of the block are not as valuable for the model generation and the weighting leads to significant quality increases. One could just omit the border pixels which are not used for replacement, but when setting the border width \( \nu = 0 \), the blocks are not overlapping anymore and no reusing of already reconstructed pixels is possible. Furthermore pixels on the outer area of the block are still not as significant as pixels in the center and no additional information from a larger area is taken into account.

The spatial weighting function, that is used from now in is equal to the one defined in lines 2 – 5 of the FSR [73]. It can be parameterized by \( \rho \) and \( \delta \). In this chapter \( \delta \) is ignored, as only the reconstruction from originally available samples is considered for each block reconstruction. Therefore it is specified for an image block of size \( N \times N \) and a sub-sampling matrix \( Q \) by:

\[
w[m, n] = \rho \sqrt{(m - \frac{N - 1}{2})^2 + (n - \frac{N - 1}{2})^2}, \forall m, n \in \{0 \ldots N - 1\} \tag{5.1}
\]

\[
w = \text{vectorize}(w[m, n]) \tag{5.2}
\]

\[
W = \frac{\text{diag}(Qw)}{\|Qw\|_1} \cdot N^2, \tag{5.3}
\]
5.1. INCORPORATION OF SPATIAL WEIGHTING AND FREQUENCY PRIOR INTO CS ALGORITHMS

Figure 5.1: The spatial weighting function $w[m, n]$, with $\hat{\rho} = 0.7$ for a block of size $32 \times 32$ used to control the influence of the coefficients in the measurement vector for the sparse model generation.

where $w[m, n]$ is the non-vectorized form (without sub-sampling) and $W$ the matrix form. Figure 5.1 shows the weighting function $w[m, n]$ in non-vectorized form, for $\hat{\rho} = 0.7$.

To incorporate this idea into the other CS reconstruction algorithms, the approach like it is applied to MP can be used directly. Instead of utilizing the normalized measurement matrix $\Phi_{\text{norm}}$ and rescaled measurement vector as inputs, the weighted measurement matrix $\Phi_{\text{weight}} = W^{\frac{1}{2}}Q\Psi$ and weighted measurement vector $y_{\text{weight}} = W^{\frac{1}{2}}f_{nr}$ are used. For all presented algorithms in Section 2.2 this leads to a sparse model generation, which tries to approximate the highly weighted samples better than the lowly weighted. In the following the effect of this additional weighting is shortly analyzed for each algorithm.

OMP Here the effect of the weighting is similar to the MP case. Instead of picking the column, which maximally reduces the residual energy, the one which maximally
reduces the weighted residual energy the most is added to the sparse model
basis. Thus, in the beginning of the iterations a model will be created, that tries
to approximate highly valued pixels more.

**Thresholding** The derivation of the three thresholding approaches IHT\(_k\), IHT and
IST are based on the minimization of the cost function \(\|y - \Phi \hat{x}\|_2\) under the con-
straint that \(\hat{x}\) is sparse. Considering IST, which uses a majorization-minimization
approach to minimize the BPDN problem, when using the weighted measurement
matrix \(\Phi_{\text{weight}}\) and weighted measurement vector \(y_{\text{weight}}\) as inputs, it tries to min-
imize a weighted form of the BPDN:

\[
\begin{align*}
\hat{x} &= \arg\min_{x' \in \mathbb{R}^N} \left\| W^{1/2} (y - \Phi x') \right\|_2 + \lambda \left\| x' \right\|_1 \\
\text{subject to} \quad \left\| W^{1/2} (\Phi x' - y) \right\|_2 &\leq \sigma,
\end{align*}
\]  

(5.4)

This leads again to the desirable property of preferring the approximation of
highly weighted samples more than of samples with small weights.

**SP** Here, in each iteration, the first stage now keeps the \(K\) basis functions that could
maximally reduce the weighted residual energy the most, instead of the residual
energy. It thus prefers basis functions, which especially approximate the highly
weighted samples.

**BPDN** The BPDN algorithm, which is the spectral gradient-projection method from
[80], also minimizes the BPDN/LASSO problem. Thus the application of the
weighted measurement matrix and weighted measurement vector lead to a mini-
mization with weighted constraint:

\[
\hat{x} = \arg\min_{x' \in \mathbb{R}^N} \left\| x' \right\|_1, \quad \text{s.t.} \quad \left\| W^{1/2} (\Phi x' - y) \right\|_2 \leq \sigma,
\]

(5.5)

resulting in a sparse model \(\hat{x}\), that especially fits highly weighted samples in the
measurement vector.
5.1. INCORPORATION OF SPATIAL WEIGHTING AND FREQUENCY PRIOR INTO CS ALGORITHMS

5.1.2 Frequency Prior

The frequency prior within the FSR allows that low frequencies are preferred to high frequency components in the model, reflecting the distribution of the coefficients in natural images. For this the frequency weighting vector $w_f$ is generated in lines 7 – 9, modeling a prior on the coefficients of the basis functions of the whole block $f$. Given the amount of effective data $h \in [0, 1]$ (which is a value indicating how many samples are available in the block and how high their weight is) and the frequency weighting factor $\beta$, the calculation of the frequency weighting vector and its non-vectorized form $w_f[a, b]$ for a block of size $N \times N$ are given by: \[73\]

$$w_f[a, b] = \left( 1 - \sqrt{2 \left( \frac{N}{2} - \frac{|a - N|}{N^2} + \frac{N}{2} - \frac{|b - N|}{N^2} \right)} \right)^{-\log(h)/\beta}. \tag{5.6}$$

$$w_f = \frac{\text{vectorize}(w_f[a, b])}{\|\text{vectorize}(w_f[a, b])\|_1} \cdot N^2. \tag{5.7}$$

In Fig. 5.2, a plot of the non-vectorized frequency weighting function is shown. High frequencies are nearly fully suppressed. In this work, the effect of the frequency weighting is only incorporated into the thresholding reconstruction algorithms, i.e., IHT$_k$, IHT, and IST. For this, the thresholding functions are replaced to preferably keep coefficients corresponding to highly weighted frequencies.

**IHT$_k$** Originally only the coefficients with the $k$ largest magnitudes are kept in each iteration. The novel approach, presented in this work, instead keeps the coefficients with the $k$ largest weighted magnitudes, where the weighting is done by multiplication with the frequency weighting vector $w_f$. Thus, the iterations are now changed to:

$$\hat{x}_{i+1} = H_{k, w_f}(\hat{x}_i + \kappa_{ihk} \Phi^H(y - \Phi \hat{x}_i)), \tag{5.8}$$

where $H_{k, w_f}(x)$ now only keeps coefficients in $x$ with index in $\text{supp}(H_k(\text{diag}(w_f) x))$.
Figure 5.2: The frequency weighting $w_f[a, b]$ with effective data $h = 0.5$ for a block of size $32 \times 32$, used to prefer low frequencies for the sparse model generation.

and sets the others to zero.

**IHT** Similarly, the IHT algorithm is adjusted, by modifying the iterations to

$$
\hat{x}_{i+1} = T_{\tau_i, w_f}(\hat{x}_i + \kappa_{\text{iht}} \Phi^H(y - \Phi \hat{x}_i)),
$$

(5.9)

where

$$
T_{\tau_i, w_f}(x)|_n = \begin{cases} 
  x|_n & \text{if } |x|_n \cdot w_f|_n \geq \tau_i \\
  0 & \text{otherwise}
\end{cases}.
$$

(5.10)

Therefore, in each thresholding step, only those coefficients with weighted magnitude above the given threshold are preserved.

**IST** The IST algorithm is modified nearly identically, by using the weighted coefficients to determine the thresholding area:

$$
\hat{x}_{i+1} = S_{\kappa_{\text{ist}}}(\hat{x}_i + \kappa_{\text{ist}} \Phi^H(y - \Phi \hat{x}_i)),
$$

(5.11)
where

\[
S_{n, w_f}(x)_n = \begin{cases} 
  x|_n + s_i & \text{if } x|_n \cdot w_f|_n \leq -s_i \\
  x|_n - s_i & \text{if } x|_n \cdot w_f|_n \geq s_i \\
  0 & \text{otherwise}
\end{cases}
\]  

(5.12)

The performances of the just presented modifications are now analyzed experimentally. For this, a more detailed explanation of the concrete implementations for all tested algorithms is given at first.

### 5.2 Implementation Details for Experimental Setup

To evaluate the reconstruction performance of different CS reconstruction algorithms and their modified versions, which were presented in Section 5.1, all algorithms were either implemented in MATLAB from literature or a library was used. Now, a specification of the concrete implementations, showing the utilized stopping criteria, input parameters and threshold choices, is given.

**MP/FSR (no weighting)/FSR** The used FSR block reconstruction implementation is from [73] with additional frequency weighting adaptation from [72]. It is exactly equivalent to the algorithm in Algorithm 4. When dealing with reconstruction of blocks, FSR denotes only the block reconstruction part of the full FSR algorithm from now on in this chapter, i.e., the reusing of previously processed blocks and the optimized processing order is left aside. With disabled orthogonality deficiency compensation, spatial weighting and frequency prior the FSR is equal to MP. As input parameters the number of iterations \( t \), the decay factor \( \hat{\rho} \), the orthogonality deficiency compensation factor \( \gamma \), and the frequency weighting factor \( \beta \) need to be specified. In all following tests the number of iterations \( t \) is set to 300. \( \beta \) is set to 2 if not stated otherwise. The number of iterations...
is picked higher than specified in the publication [73] to guarantee a high reconstruction quality and execution time is primarily neglected in this work. The frequency weighting factor $\beta = 2$ is taken from literature [73]. All other parameters ($\hat{\rho}$ and $\gamma$) are explored with a grid search in Section 5.3. When considering the MP algorithm, no parameters are left open, as the iteration count is also fixed to 300 in the following, which is the only parameter of MP. Additionally the FSR (no weighting) is examined (with disabled spatial weighting and disabled frequency weighting), where only the orthogonality deficiency compensation factor $\gamma$ needs to be trained.

**OMP/WOMP** The utilized implementation for OMP is from [4]. It is a fast implementation in MATLAB, replicating the algorithm in Algorithm 2. The modified version with additional spatial weighting, as introduced in Section 5.1.1, is called Weighted OMP (WOMP) from now. Input parameters of WOMP are the number of iterations $t$ and the decay factor $\hat{\rho}$ for the weighting function. Using the stopping criterion of a threshold value for the current residual energy instead of a fixed iteration count $t$ showed worse results. Executing WOMP is simply done by using the weighted measurement matrix and weighted measurement vector as inputs to OMP, like described in Section 5.1.1. When $\hat{\rho} = 1$ WOMP is equal to OMP. Both parameters ($t$ and $\hat{\rho}$) are trained in Section 5.3 using a grid search.

**IHT$_k$/WIHT$_k$** For the IHT$_k$ algorithm an own implementation was realized in MATLAB. Here, multiplications of the measurement matrix and its conjugate transpose are executed using the FFT, allowing for a significant speed up. The weighted version with both spatial weighting (Section 5.1.1) and frequency prior (Section 5.1.2) is named Weighted IHT$_k$ (WIHT$_k$). Inputs of WIHT$_k$ are the sparsity level $k$, the step size $\kappa_{\text{iht}}$, number of iterations $t$, $\hat{\rho}$ for the weighting function and $\beta$ for the frequency weighting. Again, executing WIHT$_k$ is simply done by using the weighted measurement matrix and weighted measurement vector as inputs to IHT$_k$, like described in Section 5.1.1. Additionally the frequency prior is incorporated, like described in Section 5.1.2. The parameters $k$, $\kappa_{\text{iht}}$ and $\hat{\rho}$ are
explored in Section 5.3 using a grid search and the number of iterations $t$ is set to 100, guaranteeing convergence of the algorithm. The frequency weighting factor $\beta$ is set to 2, when the frequency weighting is used. Otherwise it is set to a large value to disable the incorporation of the frequency prior. When $\hat{\rho} = 1$ and the frequency prior is not activated WIHT$_k$ is equal to IHT$_k$.

**IHT/WIHT** As for the IHT$_k$ algorithm, an own implementation, using the FFT, is used for the IHT algorithm. The weighted version with both spatial weighting (Section 5.1.1) and frequency prior (Section 5.1.2) is named Weighted IHT (WIHT). Inputs of WIHT are the sparsity level $k$, the step size $\kappa_{\text{iht}}$, number of iterations $t$, $\hat{\rho}$ for the weighting function and $\beta$ for the frequency weighting.

Using the IHT or WIHT, one needs to choose the parameters $\kappa_{\text{iht}}$ and $\tau_i$ appropriately. A heuristic for determining the thresholding parameter $\tau_i$ for each iteration $i$ is presented in [54]. There, it is assumed that the marginal histogram of the projected residual $\Phi^H r_i$ at positions where the measured vector $x$ is 0, is distributed according to a Gaussian distribution, with standard deviation $\sigma$. The standard deviation $\sigma$ is then robustly estimated using the median absolute deviation [66] and the threshold $\tau_i$ is set as a fixed multiple $\theta_{\text{iht}}$ of that standard deviation. In each iteration this implies a constant false alarm rate $\xi_{\text{iht}}$. The false alarm rate is the probability of mistakenly not setting a coefficient to 0, when it is 0 in the vector $x$ to be reconstructed. Given the false alarm rate $\xi_{\text{iht}}$, $\theta_{\text{iht}}$ and subsequently $\tau_i$ can be calculated. In [54] the parameters $\xi_{\text{iht}}$, $\theta_{\text{iht}}$ are trained using a specific set of measurement matrices $\Phi$ (e.g. random Gaussian matrices) and coefficient distributions for $x$ (e.g. strictly sparse signals with constant amplitude and random signs on the non-zero coefficients). Unfortunately the combination of partial Fourier matrices in two dimensions and a coefficient distribution of natural image blocks in the DFT domain is not analyzed. Therefore, both parameters, $\xi_{\text{iht}}$ and $\theta_{\text{iht}}$, are explored again in this work in Section 5.3 for this specific case. Other input parameters of the WIHT are the frequency weighting factor $\beta$, decay factor $\hat{\rho}$, number of iterations $t$, step size $\kappa_{\text{iht}}$. 

---

65
and false alarm rate $\xi_{\text{int}}$. Except for the frequency weighting factor $\beta$ (which is set to 2, if frequency weighting is activated) and number of iterations (which is set to 100) all other parameters are explored in Section 5.3.

**IST/WIST** The implementation of the IST and its corresponding weighted version Weighted IST (WIST), using both spatial weighting (Section 5.1.1) and frequency prior (Section 5.1.2), are equal to the implementation of IHT/WIHT, with the difference of a soft thresholding instead of the hard thresholding step. The step size and false alarm rate are denoted as $\kappa_{\text{ist}}$ and $\xi_{\text{ist}}$, respectively, for IST/WIST. The configuration of the input parameters and parameter exploration is also identical to IHT/WIHT.

**SP/WSP** The MATLAB implementation from [54] is used for SP. Again, the weighted version, this time only using the spatial weighting (Section 5.1.1), is denoted Weighted SP (WSP). Input parameters for WSP are the number of iterations $t$ and the decay factor $\hat{\rho}$ of the weighting function. The sparsity level $K$ is determined by

$$K = \left\lfloor M \cdot 0.044417 \left( \frac{M}{N^2} \right)^2 + 0.34142 \frac{M}{N^2} + 0.14844 \right\rfloor ,$$

which is the optimal value explored in [54]. The number of iterations $t$ is set to 25, showing already good convergence for different parameters of $\hat{\rho}$. The decay factor $\hat{\rho}$ is then explored in Section 5.3.

**BPDN/WBPDN** Regarding BPDN, the SPGL1 library [80], is used. For the weighted version, called Weighted BPDN (WBPDN), only the spatial weighting is utilized. Input parameters for WBPDN are therefore the input parameters of the standard BPDN, i.e., only the regularization parameter $\sigma$, and additionally the decay factor $\hat{\rho}$ for the spatial weighting function. Both parameters are explored in Section 5.3.

The final reconstructed block $\hat{f}[m,n]$ then results for all algorithms by taking the real part of the transformed/spatial sparse model output $g_{\text{real}} = \text{real}(\Psi \hat{x})$ and replac-
ing only the unknown pixels in $f_{nr}[m, n]$ with the corresponding reconstructed pixels. Furthermore values above 1 and below 0 are clipped to 1 and 0, respectively, as the image value range is restricted to $[0, 1]$. In other words: The real part of the transformed sparse model can also be presented in non-vectorized form by $g_{\text{real}}[m, n]$. Then

$$
\hat{f}[m, n] = \text{clip}(q[m, n]f_{nr}[m, n] + (1 - q[m, n])g_{\text{real}}[m, n]),
$$

where clip() does the $[0, 1]$ clipping described above.

### 5.3 Parameter Exploration on Natural Image Blocks

As all examined algorithms still contain some open parameters, a parameter exploration is done. The reconstruction quality for different parameter combinations is measured for each algorithm. The evaluation procedure, that is used to determine the quality of a given parameter combination is shown in Fig. 5.3 for a single block. At first, 10000 blocks of size $32 \times 32$, randomly picked from the first twelve images of the Kodak database [40], are independently randomly sub-sampled with the sub-sampling rates $S$ in $\{0.2, 0.4, 0.6, 0.8\}$, resulting in 40000 sub-sampled blocks. Subsequently, all blocks are reconstructed with the presented reconstruction algorithms and all different parameter combinations for each algorithm. The tried parameter values for each algorithm can be seen in Section 5.3. For a given algorithm all parameter combinations in the Cartesian product of the specified sets are explored. Table 5.1 now shows the best parameter combination for each algorithm and each sub-sampling rate. Additionally, to measure the benefits of the weighting modifications, the unweighted and weighted versions are explored separately. For the unweighted versions, the parameter $\hat{\rho}$ is not explored and furthermore the frequency prior is disabled, if it is implemented. Now, the Mean Squared Error (MSE) is calculated on each block between the centered $4 \times 4$ sub-block of the original block and the reconstructed block. The MSE value for each
reconstructed block is thus calculated by:

\[
\text{MSE} = \frac{\sum_{m=\frac{N}{2}}^{\frac{N}{2}+2} \sum_{n=\frac{N}{2}}^{\frac{N}{2}+2} (f[m,n] - \hat{f}[m,n])^2}{N^2},
\]  

(5.15)

and the resulting PSNR is then calculated with

\[
\text{PSNR} = 10 \log_{10} \left( \frac{B}{\sum_{i=0}^{B-1} \text{MSE}_i} \right),
\]  

(5.16)

where \( B \) is the number of blocks (here \( B = 10000 \)) used for the evaluation of each parameter combination and \( \text{MSE}_i \) is the MSE of the \( i \)-th block in the ordered set of reconstructed blocks. All 10000 MSE values are then combined to a single Peak Signal-to-Noise Ratio (PSNR) value, separately for each algorithm, parameter combination and sub-sampling rate. Note, that for each algorithm, parameter combination and sub-sampling rate, different 10000 reconstructed blocks are available and therefore a PSNR is calculated for each of these block sets.

In Table 5.1 the results of the parameter exploration can be seen. The best parameters, achieving the highest PSNR among the tested parameter combinations, are listed for each algorithm and sub-sampling density. Moreover, the PSNR of each best parameter combination is shown. As these are only results on a training set, these values are not analyzed further, except for the fact, that SP/WSP achieves a significantly worse reconstruction value, compared to all other algorithms. This is probably due to the calculation of the sparsity level \( K \) and the stopping criterion. The sparsity level calculation is from [54] and is not trained on the specific combination of natural image blocks in the DFT domain, measured with a partial Fourier matrix. Considering the stopping criterion, the algorithm stops after a fixed number of iterations, which could be improved by stopping, when the residual energy gets worse again, as a higher iteration count does not guarantee a lower residual energy. Because this would exceed the scope of this work, the SP/WSP algorithms are dropped for the evaluation on a test set in the following.
Table 5.1: Best parameters from the parameter exploration. Each column in the sub-tables “No weighting” and “W weighting” indicates the best value of a specific parameter for a different sampling rate $S$. “No weighting” uses the standard CS reconstruction algorithm or the FSR with only orthogonality deficiency compensation, but without the weighting modifications. Therefore $\hat{\rho}$ is ignored in the “No weighting” table. Moreover, the PSNR in dB for the best parameter combination can be seen for each algorithm, sub-sampling rate and weighting mode.
CHAPTER 5. USING COMPRESSED SENSING RECONSTRUCTION ALGORITHMS WITHIN THE FREQUENCY SELECTIVE RECONSTRUCTION

Figure 5.3: Evaluation procedure: Each block is randomly sampled and reconstructed by the presented algorithms with different parameters. Subsequently, the MSE is calculated for each block on the centered $4 \times 4$ sub-block. All MSE values are then combined to a single PSNR value, separately for each algorithm, parameter combination and sub-sampling rate.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tested Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSR</td>
<td>$\gamma$ {0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1} \rho {0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}</td>
</tr>
<tr>
<td>OMP/WOMP</td>
<td>$t$ {5, 10, 20, 30, 50, 60, 70, 80, 90, 100, 120, 150} \rho {0.6, 0.7, 0.8, 0.9, 1}</td>
</tr>
<tr>
<td>IHT$_k$/WIHT$_k$</td>
<td>$k$ {5, 10, 20, 30, 50, 70, 90, 120, 150, 200} $\kappa_{\text{ihht}}$ {0.001, 0.01, 0.03, 0.05, 0.07, 0.09, 0.12, 0.3, 0.5, 0.7, 0.9, 1} $\hat{\rho}$ {0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}</td>
</tr>
<tr>
<td>IHT/WIHT</td>
<td>$\xi_{\text{ihht}}$ {0.0005:0.003:0.06} $\kappa_{\text{ihht}}$ {0.05:0.1:1} $\hat{\rho}$ {0.75, 0.8, 0.85, 0.9, 0.95, 1}</td>
</tr>
<tr>
<td>IST/WIST</td>
<td>$\xi_{\text{ist}}$ {0.14:0.04:0.6} $\kappa_{\text{ist}}$ {0.05:0.1:1} $\hat{\rho}$ {0.72, 0.76, 0.80, 0.84, 0.88, 0.92, 0.96, 1}</td>
</tr>
<tr>
<td>SP/WSP</td>
<td>$\hat{\rho}$ {0.4:0.05:1}</td>
</tr>
<tr>
<td>BPDN/WBPDN</td>
<td>$\sigma$ {0.001 0.005 0.01 0.015 0.02 0.03 0.04 0.06} $\hat{\rho}$ {0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}</td>
</tr>
</tbody>
</table>

Table 5.2: For each algorithm and each sub-sampling density in all combinations of corresponding tested values are evaluated, i.e, a grid search is done with axis locations specified by the tested values column. The notation $[\text{start}:\text{step}\_\text{size}:\text{end}]$ indicates the interval from start to end, sampled at positions, beginning from start, with equal distance step\_size. Regarding the unweighted versions, $\hat{\rho}$ is not explored and the frequency prior, if implemented, is disabled.
5.4 Evaluation on Artificial Strictly Sparse Blocks

After training the parameters on a training set, the correctness of the implementation of the algorithms is now shown by presenting the reconstruction success for artificially created blocks. The generated blocks exhibit a strict sparsity with different sparsity levels $k$. For each of these sparsity levels 10000 blocks are created by setting $\frac{k}{2}$ randomly selected coefficients of the upper half of a $32 \times 32$ sized block to a uniformly distributed random value in $[0, 1]$. The final original block then results by taking the real part of the inverse 2D-DFT of this sparse block and additionally norming the block in the spatial domain by dividing by its maximum value. This way all frequencies get mirrored and almost all blocks exhibit a sparsity level of $k$ in the frequency domain, while being real in the spatial domain. Only the blocks, where the coefficient located at $(0,0)$ or the coefficient at $(16,16)$ is selected, have a sparsity of $k - 1$. Now, the created blocks are processed like in Fig. 5.3, except that only the best parameter combination, as determined in Section 5.3, is used for the reconstruction. Again, each of the blocks is sampled with the sub-sampling rates $S$ in $\{0.2, 0.4, 0.6, 0.8\}$ and subsequently reconstructed. The MSE is then calculated only for the centered $4 \times 4$ block and combined across the 10000 blocks for each sub-sampling rate and reconstruction algorithm to one PSNR value, like described in (5.15) and (5.16).

Figure 5.4 and Fig. 5.5 show the results of the artificial block reconstruction for the sub-sampling rates of 0.2 and 0.8, respectively. The graphs of for the sub-sampling rates 0.4 and 0.6 can be found in the appendix (Fig. A.1 and Fig. A.2). The graphs are clipped at 40 dB for clarity of presentation. It can be seen that for a low sparsity level, relative to the applied sub-sampling rate, each algorithm can reconstruct the blocks nearly perfectly (i.e. $PSNR > 40$ dB). Given a high sub-sampling rate, like 0.8, a perfect reconstruction is still possible for a sparsity level to number of available samples ratio $\frac{k}{S \cdot N^2}$ of up to 0.75. Therefore all algorithms work for their intended case of reconstruction sparse signals from only a small number of linear measurements. As all parameters were trained on a set with different characteristics, i.e., natural image
blocks, the results for the strictly sparse blocks can not be taken for the quality comparison of the different algorithms. Some algorithms might be more robust for the found optimal parameters with respect to the type of signals that are reconstructed. Nevertheless, it can be seen that the weighting modifications actually show worse results, which is because the priors exploited by the weighting is not present in the artificial blocks. Considering the reconstruction quality comparison of the different reconstruction algorithms, a test set, consisting of natural image blocks is evaluated in the next section.
5.4. EVALUATION ON ARTIFICIAL STRICTLY SPARSE BLOCKS

**Sub-sampling rate $S = 0.2$**

Figure 5.4: Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks with sparsity level $k$ and sub-sampling rate $S = 0.2$. The ordinate is clipped at 40 dB.
CHAPTER 5. USING COMPRESSED SENSING RECONSTRUCTION ALGORITHMS WITHIN THE FREQUENCY SELECTIVE RECONSTRUCTION

Sub-sampling rate $S = 0.8$

The ordinate is clipped at 40 dB

Figure 5.5: Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks with sparsity level $k$ and sub-sampling rate $S = 0.8$. The ordinate is clipped at 40 dB.
5.5 Evaluation on Natural Image Blocks

After verifying the reconstruction algorithms, the important results, regarding the reconstruction of natural image blocks are presented. The test set for natural image blocks consists of the 100000 blocks of size $32 \times 32$, randomly picked from the second twelve images of the Kodak database [40]. These blocks are then independently randomly sub-sampled, like in non-regular sampling, with the sub-sampling rates $S$ in \{0.2, 0.4, 0.6, 0.8\}, resulting in 400000 sub-sampled blocks. Again, all blocks are reconstructed with each algorithm and the corresponding trained optimal parameter combinations. Like in the previous section the reconstruction quality for each algorithm and each sub-sampling rate is established by calculation the PSNR across each set of reconstructed 100000 blocks. Here, again, only the centered $4 \times 4$ block is used for the MSE calculation in each block. The results are plotted in Fig. 5.6. In Fig. 5.6a the outcome for the algorithms, that do not use any weighting modification, is presented. These are the basic CS reconstruction algorithms, applied to invert the linear system present in non-regular sampling, and the FSR without spatial or frequency weighting, but with enabled orthogonality deficiency compensation (FSR (no weighting)). In Chapter 4 the applicability of basic CS reconstruction algorithms for the problem of inverting non-regular sampling is hinted, as the problem exhibits favorable properties, like a sparse vector $\mathbf{a}$, to be measured, and a measurement matrix $\mathbf{\Phi}$ with small RIP-constants $\delta_k$, even for large $k$.

For the unweighted algorithms, the best performance is shown by FSR (no weighting) and IST for low sampling densities and BPDN for higher sampling densities. When compared to the weighted versions in Fig. 5.6b, the unweighted algorithms can not compete. Across all sub-sampling rates an increase of at least 0.9 dB and up to 2.4 dB can be achieved for all algorithms. Especially the FSR can profit immensely from the weighting, outperforming all other algorithms. But also the other standard CS reconstruction algorithms can profit from the additional spatial weighting and — partly — the frequency weighting. Following from this, the basic CS reconstruction algorithms
CHAPTER 5. USING COMPRESSED SENSING RECONSTRUCTION ALGORITHMS WITHIN THE FREQUENCY SELECTIVE RECONSTRUCTION

Table 5.3: Execution times of the weighted algorithms for the reconstruction of a block of size $32 \times 32$ in seconds. The unweighted versions show a nearly equal respective execution time.

<table>
<thead>
<tr>
<th>Sub-sampling rate $S$</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSR</td>
<td>3.1 s</td>
<td>3.1 s</td>
<td>3.1 s</td>
<td>3.1 s</td>
</tr>
<tr>
<td>WOMP</td>
<td>72.1 s</td>
<td>111.9 s</td>
<td>182.3 s</td>
<td>261.3 s</td>
</tr>
<tr>
<td>WIHT$_k$</td>
<td>13.4 s</td>
<td>13.4 s</td>
<td>13.4 s</td>
<td>13.4 s</td>
</tr>
<tr>
<td>WIHT</td>
<td>7.4 s</td>
<td>7.4 s</td>
<td>7.4s</td>
<td>7.4 s</td>
</tr>
<tr>
<td>WIST</td>
<td>8.6 s</td>
<td>8.5 s</td>
<td>8.6s</td>
<td>8.6 s</td>
</tr>
<tr>
<td>WBPDN</td>
<td>174.8 s</td>
<td>455.5 s</td>
<td>584.2 s</td>
<td>635.7 s</td>
</tr>
</tbody>
</table>

are not really suitable for the reconstruction of non-regularly sampled images, but with some additional modifications, they show acceptable results. Nevertheless, the FSR still shows a significantly better reconstruction quality, especially for small sub-sampling rates. In Fig. 5.7 – Fig. 5.10 the reconstruction of 4 example blocks are shown visually for each algorithm. As only the small centered block is used for the final reconstruction of the whole image and also for the MSE calculation, a visual interpretation from this low sample size is difficult. But it can be noticed that the weighted versions actually show worse results on the whole image blocks, as they only try to approximate the middle of the block really well. To measure the execution times of the different algorithms, the weighted reconstruction algorithms are now executed on an i5-4690 CPU, running at 3.50 GHz, with MATLAB R2015b. Looking at the execution time in Table 5.3, the FSR furthermore achieves the lowest execution time, followed by the fast implementation of the Iterative Thresholding algorithms. For the OMP and BPDN, which achieve the worst performance, the reconstruction time is additionally dependent on the sub-sampling rate.

All in all, the experimental results of this section showed the benefits of the additional weighting modifications for every CS algorithms. Unfortunately the quality performance of the FSR could not be improved by exchanging the block reconstruction to a weighted CS reconstruction algorithm and neither could the execution time be improved. Nevertheless, the weighting modifications seem to be universally applicable
5.5. EVALUATION ON NATURAL IMAGE BLOCKS

Figure 5.6: Reconstruction quality of the presented algorithms for the reconstruction of natural blocks.
to CS reconstruction algorithms, showing the potential use for different CS reconstruction tasks. The question if the reconstruction quality of the FSR and the weighted CS reconstruction algorithms can compete with other relevant resampling algorithms on a whole image is examined in the next chapter. For this, the proposed weighted CS algorithms are used in the block extrapolation step of the FSR and the reconstruction is applied to a whole image.
5.5. EVALUATION ON NATURAL IMAGE BLOCKS

Figure 5.7: Visual results for the reconstruction of 4 example blocks. Only the centered block, marked in red, is used for the actual reconstruction and is considered for the calculation of the MSE. The sub-sampling rate is $S = 0.4$ for all blocks. First figure.
Figure 5.8: Visual results for the reconstruction of 4 example blocks. Only the centered block, marked in red, is used for the actual reconstruction and is considered for the calculation of the MSE. The sub-sampling rate is $S = 0.4$ for all blocks. Second figure.
Figure 5.9: Visual results for the reconstruction of 4 example blocks. Only the centered block, marked in red, is used for the actual reconstruction and is considered for the calculation of the MSE. The sub-sampling rate is $S = 0.4$ for all blocks. Third figure.
**Figure 5.10:** Visual results for the reconstruction of 4 example blocks. Only the centered block, marked in red, is used for the actual reconstruction and is considered for the calculation of the MSE. The sub-sampling rate is $S = 0.4$ for all blocks. Fourth figure.
Chapter 6

Comparison of the Reconstruction of Sub-Sampled Images with other available Resampling algorithms

Finally, the reconstruction algorithms proposed in the previous chapter are applied for the resampling of whole images. The preceding considerations only analyzed the reconstruction of small image patches. Now, the algorithms are incorporated in the block processing of the FSR. Instead of using the standard FSR reconstruction (Section 3.3.2), the implementations of the specific CS algorithms (Chapter 5) are used for the block reconstruction part of the FSR. Everything else, i.e., the optimized processing order and the reuse of already reconstructed neighboring pixels, is kept the same. In the following, these modified FSR versions are called “FSR + specific algorithm”, e.g., “FSR + WOMP”, while the original FSR with enabled frequency prior, orthogonality deficiency compensation and spatial weighting is simply called “FSR”. The weighting factor of previously reconstructed pixels $\delta$ is set to $0.5$ and the border width $\nu$ is set to $14$, with a block size of $32 \times 32$. 
Additionally, to analyze the reconstruction quality of the FSR and the modified versions in contrast to other popular reconstruction methods, the algorithms presented in Section 3.3.1 are also applied. Here, the implementations provided by the respective authors are utilized. For the parameters the provided specification is taken, if not mentioned otherwise in this work. Furthermore, a simple Linear Interpolation (LI), where the reconstruction of a missing sample is achieved by linearly interpolating between the three neighbor pixels of the Delaunay Triangulation [24] of the available samples, is used to show the raison d’être of more advanced techniques. Regarding Morphological Component Analysis (MCA) [35], a dictionary consisting of a combination of Curvelets with coarsest scale equal to 5 and a block-based DCT of size $32 \times 32$ is used. The number of iterations is fixed to 300 and the TV regularization is applied to both dictionaries with its parameter equal to 1.5. The threshold is decreased linearly, no additional stopping criterion is applied and the standard deviation for the noise is estimated using the median standard deviation. For the Constrained Split Augmented Lagrangian Shrinkage Algorithm (CSALSA) algorithm the implementation from [2] is used unchanged and for Wavelet Inpainting (WI) the implementation from [76] is utilized. To apply the block-based CS and smoothed projected Landweber reconstruction (BCS-SPL) algorithm [61] for the reconstruction of non-regularly sampled images, the sub-sampling matrix $Q$ needs to exhibit a specific block structure. Therefore, for all reconstruction algorithms the same sub-sampling matrix with block structure is used. The blocks consist of size $32 \times 32$ with equal sub-sampling pattern. Considering MKSVD [53], a C++ implementation from Julien Mairal [51] is used. The dictionary provided by the authors and named “L20m8n16GRAY” is taken as a starting point. The algorithm is applied 10 times for each image with parameters WEIGHT_CONSTANT = 2.5, PRUNE_RATIO = 1.0 and block size equal to 300. Therefore in each of the 10 steps a different dictionary, that was learned in the preceding applications is used. The specific parameters $J$ and $L$ are alternated in each of the ten application between $J = 5$ and $L = 10$ and $J = 0$ and $L = 25$.

All algorithms are single-thread implementations in MATLAB, except for MKSVD,
which is a C++ implementation and uses multi-threading.

For the evaluation on whole images, the second twelve images from the KODAK database of size $768 \times 512$ are sub-sampled with the sub-sampling rates $S \in \{0.2, 0.4, 0.6, 0.8\}$ and with a sub-sampling mask $b[x_1, x_2]$, that consists of the same $32 \times 32$ sized sub-sampling mask blocks. The sub-sampling mask that was used for sub-sampling rate $S = 0.4$ can be seen in Fig. 6.1 applied to an example image. Now, all images are reconstructed with the presented reconstruction algorithms. Details of four resampled images from the KODAK database can be seen in Fig. 6.3 and Fig. 6.4 for the different algorithms and sub-sampling rate 0.4. Figure 6.2 shows the average PSNR for each algorithm. In Appendix B all reconstructed images of the image “kodim15” is shown for the sub-sampling rate 0.2.

With respect to the reconstruction quality, the standard FSR can outperform all
other algorithms for all sampling densities. Only for low sampling densities the MKSVD algorithm can achieve similar performances, that are still a little bit worse. BCS-SPL can only achieve a reconstruction quality that is slightly better than the simple reconstruction from LI. It therefore seems to be insufficient for the task of resampling, when using the unmodified version, provided by the authors. Considering the newly introduced CS reconstruction algorithms for the block reconstruction step of the FSR, they can not compete with the original FSR, which was already foreseeable from the natural block evaluation. The CS reconstruction, which shows acceptable results and is the best among the novel algorithms is the “FSR + WIST” version. Although it is worse than FSR, it can outclass nearly all the other popular reconstruction algorithms, except MKSVD for all sampling densities and MCA for low sampling densities. “FSR + WIST” is followed by “FSR + WBPDN”, “FSR + WOMP” and “FSR + WIHT”, which achieve similar reconstruction results. The visual results in Fig. 6.3 and Fig. 6.4 match with the PSNR metric results and it can be seen, that the newly introduced algorithms show an acceptable reconstruction quality.

It has to be noted, that the parameter exploration was only done on image blocks and did not consider the re-using of previously reconstructed samples. Because of this, the reconstruction quality of the newly introduced algorithms might be better if trained again on whole image. E.g., The “FSR + WIHT” seems to have unstable parameters and therefore diverges for many blocks for the lowest sampling density. Here, the achieved quality is even worse than LI. Even though this might be mitigated by a re-training of the parameters, this shows the instability of the parameter combination for this CS-based algorithm. It does not show equal performances if the signal input changes slightly. In contrast to this, the original version of the FSR algorithm seems to be stable with respect to its parameter choice, making it even more attractive. This was also shown in [73].

Execution times for all algorithms are presented in Table 6.1. It is the average reconstruction time of a single image from the KODAK database, i.e., a greyscale image
Figure 6.2: Average PSNR for the reconstruction of the second twelve images from the KODAK database with different reconstruction algorithms and sub-sampling rates.
of size $768 \times 512$. The shortest execution time is of course accomplished by the three methods with the worst reconstruction quality, i.e., LI, followed by BCS-SPL, followed by CSALSA. The FSR, while achieving the best reconstruction quality, additionally shows an execution time, which is comparatively low. Popular algorithms, like MCA and MKSVD have reconstruction times that are an order of magnitude higher, even though MKSVD uses a multi-threaded implementation. The newly introduced algorithms, which use the weighted version of CS algorithms in the block reconstruction of the FSR, all have higher execution times, proportional to the execution times measured for the block extrapolation in Section 5.5.

Despite the fact, that the newly introduced weighted CS algorithms in the block reconstruction step of the FSR can not compete with the state-of-the-art original FSR, this section showed, that these new algorithms can still keep up with other popular reconstruction algorithms. This is only possible due to the introduced weighting schemes, which were inspired by the FSR. Given other types of signals or measurement matrices, where the concepts of overlapping blocks and spatial weighting could be applied, these algorithms might be suited better than the FSR. Unfortunately, for the task of resampling non-regularly sampled natural images to a regular grid the introduced CS algorithms within the FSR are not beneficial in the implemented form.
Table 6.1: Execution times for reconstructing a whole non-regularly sampled greyscale image of size 768 × 512 in seconds.

<table>
<thead>
<tr>
<th></th>
<th>Sub-sampling rate S</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>FSR</td>
<td>187 s</td>
<td>186 s</td>
<td>186 s</td>
<td>186 s</td>
</tr>
<tr>
<td>FSR + WOMP</td>
<td>804 s</td>
<td>1466 s</td>
<td>3094 s</td>
<td>6197 s</td>
</tr>
<tr>
<td>FSR + WIHT$_k$</td>
<td>334 s</td>
<td>333 s</td>
<td>333 s</td>
<td>331 s</td>
</tr>
<tr>
<td>FSR + WIHT</td>
<td>349 s</td>
<td>348 s</td>
<td>347 s</td>
<td>341 s</td>
</tr>
<tr>
<td>FSR + WIST</td>
<td>398 s</td>
<td>398 s</td>
<td>397 s</td>
<td>394 s</td>
</tr>
<tr>
<td>FSR + WBPDN</td>
<td>9027 s</td>
<td>24230 s</td>
<td>27740 s</td>
<td>24224 s</td>
</tr>
<tr>
<td>MCA [35]</td>
<td>2139 s</td>
<td>2129 s</td>
<td>2138 s</td>
<td>2123 s</td>
</tr>
<tr>
<td>CSALSA [2]</td>
<td>16.2 s</td>
<td>16.7 s</td>
<td>16.7 s</td>
<td>16.6 s</td>
</tr>
<tr>
<td>WI [76]</td>
<td>523 s</td>
<td>530 s</td>
<td>527 s</td>
<td>528 s</td>
</tr>
<tr>
<td>BCS-SPL [61]</td>
<td>3.4 s</td>
<td>3.5 s</td>
<td>3.2 s</td>
<td>3.3 s</td>
</tr>
<tr>
<td>LI</td>
<td>1.6 s</td>
<td>1.9 s</td>
<td>2.5 s</td>
<td>2.8 s</td>
</tr>
<tr>
<td>MKSVD [53]</td>
<td>7133 s</td>
<td>7110 s</td>
<td>7145 s</td>
<td>7018 s</td>
</tr>
</tbody>
</table>
Figure 6.3: Visual results for details of different test images from the KODAK image database with sub-sampling rate $S = 0.4$ and reconstruction with the different FSR + proposed algorithm combinations.
Figure 6.4: Visual results for details of different test images from the KODAK image data base with sub-sampling rate $S = 0.4$ and reconstruction with different state-of-the-art reconstruction algorithms.
Chapter 7

Conclusion and Outlook

In this work, interesting connections between CS and the FSR, or generally, the task of resampling images to a regular grid from only a subset of positions, could be shown. As natural images exhibit a sparse representation in a wide range of different dictionaries, many reconstruction algorithms, utilizing this sparsity prior, have shown success for resampling non-regularly sampled images to a regular grid. These algorithms were usually developed apart from the CS framework, so that an examination concerning the connection between these algorithms and CS seems promising. CS allows for an efficient exploitation of the sparsity prior when measuring signals with linear measurement systems. The framework provides computationally feasible reconstruction algorithms, for which some can give near optimal theoretical recovery guarantees, when the measurement system is sufficiently conditioned. When sensing nearly sparse signals with partial Fourier matrices, independently of the dimensionality, one can give efficient recovery guarantees for a set of reconstruction algorithms in the form of uniform instance optimality. The partial Fourier matrices show beneficial properties with respect to coherence and RIP-constants. Additionally, the sensing with partial Fourier matrices is equivalent to sub-sampling signals, which are nearly sparse in the Fourier domain, at random positions in the spatial domain.
The FSR can be compared to the MP algorithm, but incorporates some additional concepts, which are the orthogonality deficiency compensation, the spatial weighting and a frequency prior. The last two of these allow for the consideration of additional image priors, other than sparsity. As the FSR uses the DFT as a sparsity matrix, the theoretical guarantees for partial Fourier matrices seem to prove parts of the success of the FSR. When using a block-based DFT as the sparsity dictionary, the CS framework additionally shows the existence of an efficient recovery of non-regularly sampled images and even provides different algorithms for the task of resampling the image to a regular grid. It has to be noted, that the theoretical bounds on the reconstruction error are only worst case bounds and the uniform instance optimality is only shown for a small subset of reconstruction algorithms. Furthermore, the FSR separates itself from the traditional CS approach by introducing the use of additional prior information. Due to these reasons, a more detailed analysis of concrete theoretical upper bounds on the reconstruction error is not examined further.

Unfortunately, the basic CS reconstruction algorithms show sub-optimal experimental performances, both with regard to execution time and reconstruction quality, if simply applied to invert the non-regular sampling process. Thus, in this work ideas from the FSR algorithm are used by introducing two different weighting modifications for the basic CS algorithms to exploit additional prior information about natural images. These modifications improve the reconstruction quality for each algorithm tremendously, while retaining the execution time. Although the CS reconstruction algorithms with the weighting modifications still can not outperform the original FSR, the results confirm the general applicability of the weighting schemes for CS algorithms. One of the weighted CS algorithms, in combination with the block processing order and reusing of previously reconstructed samples of the FSR, can even compete against other state-of-the-art reconstruction algorithms, expect for the original FSR. But, if the type of signal is different to natural images or another measurement system is present, where the concepts of overlapping blocks and spatial weighting could be applied, these algorithms might outperform the FSR.
Additionally, there exists a continuously increasing variety of additional CS reconstruction algorithms, which could be tested within the block reconstruction step of the FSR and potentially show state-of-the-art performances for resampling images to a regular grid. Of course the exchange of ideas of the FSR and CS can be considered further. One interesting concept within the block reconstruction of the FSR is the orthogonality deficiency compensation, which can improve the reconstruction quality of the simple MP algorithm significantly. Concepts like the Wiener filtering step within BCS-SPL or other priors, developed within the CS framework and suitable for natural images, could be incorporated into the FSR. Finally, the significant advances in the recent years in the field of machine learning, lead to some promising results for learning CS reconstruction algorithms from many sparse vector and measurement vector pairs. In [1], an algorithm for inverting block based compressed sensing on images, like applied in BCS-SPL, is learned from a high amount of given input and output pairs and achieves an increased reconstruction quality, while providing a significantly lower execution time than BCS-SPL. In [48] and [60] a Convolutional Neural Net and a Stacked Denoising Autoencoder are successfully trained, respectively, to reconstruct compressively sensed images. Due to the strong connection between CS and non-regular sampling, these ideas could potentially provide a reconstruction algorithm with improved quality and remarkably low execution time. Perhaps ideas from the FSR, like a spatial weighting on the cost function during the training of the learning algorithm, can lead to additional improvements.
Appendix A

Reconstruction Quality for Artificial Sparse Blocks
CHAPTER A. RECONSTRUCTION QUALITY FOR ARTIFICIAL SPARSE BLOCKS

Sub-sampling rate $S = 0.2$

![Graph of Sub-sampling rate $S = 0.2$](image)

**Figure A.1:** Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks with sparsity level $k$ and sub-sampling rate $S = 0.4$. The ordinate is clipped at 40 dB.
Figure A.2: Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks with sparsity level $k$ and sub-sampling rate $S = 0.6$. The ordinate is clipped at 40 dB
Appendix B

Visual Results of Reconstructed Images for Sampling Rate 0.2

Figure B.1: Original image kodim15 from the KODAK database [40].
Figure B.2: Sampled image kodim15 with sub-sampling rate $S = 0.2$.

Figure B.3: Image kodim15 reconstructed with FSR.
Figure B.4: Image kodim15 reconstructed with FSR + WOMP.

Figure B.5: Image kodim15 reconstructed with FSR + WIHT_k.
Figure B.6: Image kodim15 reconstructed with FSR + WIHT.

Figure B.7: Image kodim15 reconstructed with FSR + WIST.
CHAPTER B. VISUAL RESULTS FOR SAMPLING RATE 0.2

Figure B.8: Image kodim15 reconstructed with FSR + WBPDN.

Figure B.9: Image kodim15 reconstructed with MCA.
Figure B.10: Image kodim15 reconstructed with CSALSA.

Figure B.11: Image kodim15 reconstructed with WI.
CHAPTER B. VISUAL RESULTS FOR SAMPLING RATE 0.2

**Figure B.12:** Image kodim15 reconstructed with BCS-SPL.

**Figure B.13:** Image kodim15 reconstructed with LI.
Figure B.14: Image kodim15 reconstructed with MKSVD.
Mathematical Operations and Conventions

Conventions

The following conventions are used:

- Vectors are written as lowercase variables in bold font.
- Matrices are written as uppercase variables in bold font.
- Images in the non-vectorized form and the spatial domain are written with $x[m,n]$, where the letter $x$ is replaced for the concrete image with another lowercase letter. The indices $m$ and $n$ denote the spatial coordinates.
- Images in the non-vectorized form and the frequency domain are written with $X[a,b]$, where the letter $X$ is replaced for the concrete image with another uppercase letter. The indices $a$ and $b$ denote the frequency indices.

Operations

$\| \cdot \|_0$ \quad $l_0$-“norm” of a vector

$\| \cdot \|_1$ \quad $l_1$-norm of a vector
\[ \| \cdot \|_2 \] \quad l_2\text{-norm of a vector}

\[ .^T \] \quad \text{Transpose operator}

\[ .^H \] \quad \text{Hermitian operator}

\[ .^* \] \quad \text{Conjugate Complex operator}

\[ .^\dagger \] \quad \text{Moore Penrose Pseudoinverse}

\[ \mathcal{O}(\cdot) \] \quad \text{Landau symbol for asymptotic upper bound}

\[ |\cdot|_u \] \quad u\text{-th element of a vector is } u \text{ is a natural number. If } u \text{ is a tuple, it is a vector, consisting of the elements of a vector at positions } u

\[ u^C \] \quad \text{Complementary set or tuple of } u. \text{ The reference set or tuple is given by context}

\[ \lceil \cdot \rceil \] \quad \text{Ceil operator}

\[ \lfloor \cdot \rfloor \] \quad \text{Floor operator}

\[ [\cdot]_{a,b} \] \quad \text{Denotes the matrix entry at position } (a,b)

\[ \text{vectorize}(\cdot) \] \quad \text{Vectorized image (3.2)}

\[ \text{diag}(b) \] \quad \text{Diagonal matrix with diagonal entries equal to the vector } b

\[ * \] \quad \text{2D Convolution with zero padding at the border}

\[ \oslash \] \quad \text{2D Circular convolution}

\[ \bullet \quad \circ \] \quad \text{Indicates DFT correspondence}

\[ \mathcal{F}_{a,b}\{ \cdot \} \] \quad \text{2D-DFT operator}

\[ \text{mod} \] \quad \text{Modulo operator}

\[ \text{real}(\cdot) \] \quad \text{Real parts of a vector}

\[ \text{clip}(\cdot) \] \quad \text{Clips the values to } [0, 1]
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>Compressed Sensing</td>
</tr>
<tr>
<td>OMP</td>
<td>Orthogonal Matching Pursuit</td>
</tr>
<tr>
<td>MP</td>
<td>Matching Pursuit</td>
</tr>
<tr>
<td>IHT&lt;sub&gt;k&lt;/sub&gt;</td>
<td>Iterative Hard Thresholding Algorithm k</td>
</tr>
<tr>
<td>RIP</td>
<td><em>restricted isometry property</em></td>
</tr>
<tr>
<td>IHT</td>
<td>Iterative Hard Thresholding</td>
</tr>
<tr>
<td>IST</td>
<td>Iterative Soft Thresholding</td>
</tr>
<tr>
<td>ISTA</td>
<td>Iterative Shrinkage-Thresholding algorithm</td>
</tr>
<tr>
<td>BPDN</td>
<td>Basis Pursuit Denoise</td>
</tr>
<tr>
<td>SP</td>
<td>Subspace Pursuit</td>
</tr>
<tr>
<td>CoSaMP</td>
<td>Compressive Sampling Matching Pursuit</td>
</tr>
<tr>
<td>SOCP</td>
<td>Second Order Cone Program</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>TV</td>
<td>total variation minimization</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>CSALSA</td>
<td>Constrained Split Augmented Lagrangian Shrinkage Algorithm</td>
</tr>
<tr>
<td>WI</td>
<td>Wavelet Inpainting</td>
</tr>
<tr>
<td>MCA</td>
<td>Morphological Component Analysis</td>
</tr>
<tr>
<td>BCS-SPL</td>
<td>block-based CS and smoothed projected Landweber reconstruction</td>
</tr>
<tr>
<td>FSR</td>
<td>Frequency Selective Reconstruction</td>
</tr>
<tr>
<td>FSE</td>
<td>Frequency Selective Extrapolation</td>
</tr>
<tr>
<td>WOMP</td>
<td>Weighted OMP</td>
</tr>
<tr>
<td>WIHT$_k$</td>
<td>Weighted IHT$_k$</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>WIHT</td>
<td>Weighted IHT</td>
</tr>
<tr>
<td>WIST</td>
<td>Weighted IST</td>
</tr>
<tr>
<td>WSP</td>
<td>Weighted SP</td>
</tr>
<tr>
<td>WBPDN</td>
<td>Weighted BPDN</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>LI</td>
<td>Linear Interpolation</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Example of a linear measurement process of a 2-sparse vector. . . . . 5

3.1 Example for the reconstruction of an image from a subset of pixels. . 26

3.2 Examples of sparse representations of a natural image from the Kodak
database [40]. ................................................................. 30

4.1 Example for the correspondence between the sub-sampled image $f_{nr}$ and
the coefficients of the fully sampled image in the frequency domain $x$. . 41

4.2 DFT coefficients of different sub-sampling masks with 50% subsampling
density. ................................................................. 47

5.1 The spatial weighting function used to control the influence of the coef-
ficients in the measurement vector for the sparse model generation. . . 59

5.2 The frequency weighting used to prefer low frequencies for the sparse
model generation. ................................................................. 62
5.3 Evaluation procedure: Each block is randomly sampled and reconstructed by the presented algorithms with different parameters. 70

5.4 Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks. 73

5.5 Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks. 74

5.6 Reconstruction quality of the presented algorithms for the reconstruction of natural blocks. 77

5.7 Visual results for the reconstruction of 4 example blocks. First Figure. 79

5.8 Visual results for the reconstruction of 4 example blocks. Second figure. 80

5.9 Visual results for the reconstruction of 4 example blocks. Third figure. 81

5.10 Visual results for the reconstruction of 4 example blocks. Fourth figure. 82

6.1 Sub-sampling mask used for all images and sub-sampling rate 0.4 applied to an example image from the KODAK database. 85

6.2 Average PSNR for the reconstruction of the second twelve images from the KODAK database with different reconstruction algorithms and sub-sampling rates. 87

6.3 Visual results for details of different test images. First figure. 90

6.4 Visual results for details of different test images. Second figure. 91
A.1 Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks. .......................................................... 96

A.2 Reconstruction quality of the presented algorithms for the reconstruction of artificial sparse blocks. .......................................................... 97

B.1 Original image kodim15 from the KODAK database [40]. ................. 98

B.2 Sampled image kodim15 with sub-sampling rate $S = 0.2$. ............... 99

B.3 Image kodim15 reconstructed with FSR. ........................................ 99

B.4 Image kodim15 reconstructed with FSR + WOMP. .......................... 100

B.5 Image kodim15 reconstructed with FSR + WIHT$_k$. ......................... 100

B.6 Image kodim15 reconstructed with FSR + WIHT. ........................... 101

B.7 Image kodim15 reconstructed with FSR + WIST. ........................... 101

B.8 Image kodim15 reconstructed with FSR + WBPDN. ......................... 102

B.9 Image kodim15 reconstructed with MCA. ....................................... 102

B.10 Image kodim15 reconstructed with CSALSA. ............................... 103

B.11 Image kodim15 reconstructed with WI. ........................................ 103

B.12 Image kodim15 reconstructed with BCS-SPL. .............................. 104
B.13 Image kodim15 reconstructed with LI. . . . . . . . . . . . . . . . . . . 104

B.14 Image kodim15 reconstructed with MKSVD. . . . . . . . . . . . . . . . 105
List of Tables

5.1 Best parameters from the parameter exploration. 69

5.2 Combinations for the parameter exploration. 70

5.3 Execution times of the weighted algorithms. 76

6.1 Execution times for reconstructing a whole non-regularly sampled greyscale image. 89
Bibliography


[57] I. Manotas Gutiérrez and H. Arguello Fuentes, “Overlapped block-based compres-
sive sensing imaging on mobile handset devices,” Revista Facultad de Ingeniería

[58] K. Meisinger and A. Kaup, “Minimizing a weighted error criterion for spatial error
concealment of missing image data,” in Image Processing, 2004. ICIP’04. 2004

[59] J. Meyer, A. Brückner, R. Leitel, P. Dammberg, A. Bräuer, and A. Tünnermann,
17506–17519, 2011.

tured signal recovery,” in Communication, Control, and Computing (Allerton),

[61] S. Mun and J. E. Fowler, “Block compressed sensing of images using directional
transforms,” in 16th IEEE International Conference on Image Processing (ICIP).

and inaccurate samples,” Applied and Computational Harmonic Analysis, vol. 26,


a technical overview,” IEEE signal processing magazine, vol. 20, no. 3, pp. 21–36,
2003.

Recursive function approximation with applications to wavelet decomposition,” in
BIBLIOGRAPHY


