Osnabrück University
Institute for Conmputer Science - Research Group Distributed Systems

Quality-Aware Compressed Sensing for Distributed Precision Agriculture Systems

Dissertation
zur Erlangung des Doktorgrades (Dr. rer. nat.)
des Fachbereichs Mathematik/Informatik
der Universität Osnabrück
vorgelegt von

Thomas Hänel

2022

Advisor: Prof. Dr. Nils Aschenbruck

IEEE copyright message

Contents

Preface

I wrote this thesis as part of my work as research assistant in the Distributed Systems research group at Osnabrück University. Therefore, I would first and foremost like to thank Nils for building up this fantastic research group and advising me on my path to this thesis. I would also like to extend my gratitude to all my colleagues and the student workers in the research group for creating an amazing working atmosphere, interesting discussions, and becoming dear friends with whom I also shared quite some free time, e.g., playing board games and Bouldering.

During my work in the Distributed Systems group, I mainly worked on two research projects, one funded by the company HARTING, therefore I would like to express my gratitude to HARTING and its employees for the pleasant and productive cooperation that lead to multiple of my research publications. The second research project is CARPE MEMORIAM, funded by the BMBF (FKZ 01IS19054B), in which much of the research that became part of this thesis took place. Therefore, I would like to extend my gratitude to our project partners from the company m2xpert and the Remote Sensing research group, especially Thom, for the long and fruitful discussions.

I would also like to thank Mike Wakin for letting me benefit from his extensive knowledge on Compressed Sensing by pointing me at some of the core research papers, that I based my research on.

Crucial, yet often overlooked, support in research comes from the anonymous reviewers - thank you for many constructive suggestions!

Real-world data played a crucial role in this thesis. Therefore, I would like to thank the company CLAAS for providing us with the combine harvester yield data used for the evaluation of CFE, Martin and Deutsches GeoForschungsZentrum (GFZ) for the airborne data, and Bastian for the ground-based data used in the trace-based evaluations of M2HSN, and Maren and Manuel for the cooperation out on the field, testing and evaluating the prototype, and acquiring the comparison data. Special thanks also go to Freddy for letting us use his field for the measurements.

Last but not least, I would like to thank my family and friends for always being there for me during the exciting years of this thesis project.

Abstract

Around 2006 the signal processing community was thrilled when the concept of Compressed Sensing was brought forward. While according to the long-established Nyquist-Shannon theorem, the sampling rate for signals must be at least twice as high as the highest relevant frequency in the signal, with Compressed Sensing lower average sampling rates become suitable. As a more general toolkit, Compressed Sensing allows for merging sensing and compression in a single step. In other words, only a tiny amount of data needs to be sensed but a huge amount of information can be reconstructed from this data. While Compressed Sensing already has been successfully applied in some areas, mainly in medical scanning where it helps to reduce the exposure to radiation, adaptation to new application areas is relatively slow.

We identify two causes that slow down the adaption and contribute to overcoming these obstacles: firstly, the adaption of Compressed Sensing requires interdisciplinary thinking even more than for many other new technologies: Compressed Sensing itself comes from the field of signal processing, the application adds a second field. The automatized processing of the data always touches the field of computer science and lastly, designing Compressed Sensing solutions often requires a modification or redesign of measurement hardware, touching the fields of electronics and sometimes mechanics. We address this issue by supplying a more structured approach for designing Compressed Sensing solutions. Secondly, Compressed Sensing usually performs a lossy compression on real world data. Not knowing the quality of the solution limits its usability. We address this issue by supplying a list of potential metrics for assessing the quality of the solution and evaluate their performance for various datasets.

Along the way, we develop two Compressed Sensing solutions in the application area of Precision Agriculture: The first is Compressive Field Estimate (CFE), a method for improving the remote estimate of a scalar field based on limited data supplied by a moving probe such as a combine harvester. The second is Multi- to Hyperspectral Sensor Network (M2HSN), a wireless sensor network that records light spectra at mediocre spectral resolution and allows for increasing the spectral resolution. For the M2HSN, we discuss different designs of the sensor nodes and different approaches for increasing the resolution. Those are simulatively evaluated on different datasets and in a real-world prototype.

1. Introduction

Despite being one of the oldest technologies of mankind, agriculture is still continuously improved with innovative new technologies. In the last two decades the concept of precision farming (Auernhammer 2001; Zhang et al. 2002) became popular. The core idea of precision farming is optimizing the cultivation of land at a high spatial resolution. Instead of adding the same amount of, e.g., fertilizer or irrigation on a whole field, for every part of the field, the exact amount is chosen. This not only helps to make the process more economical but also more sustainable.

A crucial first step for taking such actions is obtaining accurate measurements of the situation. In the last decades, especially remote sensing with satellites made major progress in this direction (Mulla 2013). More recently, the spatial resolution was increased by using Unmanned Aerial Vehicles (UAVs) for remote sensing (Maes et al. 2019).

These approaches have some drawbacks: due to weather constraints and regulations, they are not suitable for continuous monitoring at high temporal resolution. For such measurements, ground based solutions are better-suited. These can even be used for some more direct measurements of vegetation data such as yield rather than indirectly deriving such data. The high temporal resolution is critical for early detection of anomalies that reduce the yield unless taking countermeasures timely. Such anomalies are, e.g., weed growth, pathogens, and drought stress (Maes et al. 2019). Furthermore, it can help to improve yield prediction which is useful, e.g., for predicting trading prices (Rembold et al. 2013). Continuous ground-based measurements can also be helpful for calibration of satellite data.

However, there are major challenges in collecting data on the ground. Due to the number of measurement devices required for high spatial resolutions, it is critical to keep the devices as simple and cheap as possible. For long-term continuous measurements, they need to be optimized for low power consumption. Another major challenge is, that in rural areas, wireless communication architecture is often weak, offering only slow and unreliable connections.

Compressed Sensing is a technology perfectly suited for these kinds of challenges. It combines the common steps of sensing data and compressing data in a single process. It helps by reducing the measurement effort and thereby simplifying measurement devices. At the same time, the

amount of data which needs to be transmitted is reduced, lowering power consumption of data transfer and making it more suitable for weak wireless connections.

Nevertheless, Compressed Sensing only gets adapted slowly in different application areas. We identify mainly two issues causing this:

1. Building Compressed Sensing (CS) solutions is rather difficult due to the multidisciplinary challenge touching electronics, mechanics, computer science, mathematics, and the field of the application, here agriculture.

2. CS reconstructions are not lossless for real-world data but an information on the quality of the reconstruction is not generated by ordinary CS.

We address the first issue by describing a more structured approach to build CS based solutions in section 2. We use this structured approach to build two solutions for ground-based sensing in agriculture: Compressive Field Estimate (CFE), which we have previously published in (Hänel et al. 2017a) and the Multi- to Hyperspectral Sensor Network (M2HSN), which we have first proposed in (Hänel et al. 2019) and improved in (Hänel et al. 2021b). The content of these three publications is the core of this thesis.

In the CFE, the focus is mainly on weak wireless connections. We accelerated the approximation of scalar fields such as a yield map based on a limited number of measurement samples taken by a harvester during harvest. In the M2HSN, the focus is mainly on developing simple measurement devices. We conceptualized, built and evaluated both simulatively and experimentally a Wireless Sensor Network (WSN) consisting of multispectral sensors. From the sensor readings, we derived hyperspectral data.

In both solutions, some special challenges arose that we researched in greater detail. In the context of CFE, we especially discuss how to select measurement samples for transmitting when the data rate is insufficient for sending all data while keeping the data suitable for CS. In the context of the M2HSN, we compare different designs that can handle varying amounts of a priori knowledge and evaluate different approaches for the upsampling of the spectral resolution.

In both solutions, one question came up frequently: is it possible to estimate the quality of the result? While there are some approaches for determining this, we did not find a large-scale comparison of multiple approaches. Especially, more trivial estimates were not included in such comparisons. Therefore, we contribute exactly this comparison for both CFE and M2HSN data and extract observations that seem applicable to other problems as well.

The attached core research questions can be phrased as follows:

1. To what degree can Compressed Sensing results be improved by selecting sample positions smartly in sparse sampling settings?

2. To what degree can hyperspectral data be derived from a multispectral wireless sensor network?

3. How well can the quality of a Compressed Sensing Reconstruction be estimated?

In this thesis, we contribute to answering these questions.

The remaining thesis is structured as follows: in chapter 2, we give an overview of CS, formulating a more structured approach for developing CS based solutions. In chapter 3, we describe vegetation spectra, which play a crucial role for the M2HSN and we describe related work and put our work in context. In chapter 4, we conceptualize the two approaches CFE and M2HSN. In chapter 5, we compile a list of approaches for estimating the reconstruction quality. In chapter 6, we optimize the configurations of the approaches which we then evaluate in detail in chapter 7 and discuss in chapter 8.

2. Compressed Sensing (CS)

Being the core technology of the approaches considered in this thesis, we will summarize the basic principle of Compressed Sensing (CS) in this chapter, mostly sticking to the terminology that has been commonly used since the survey written by Candes et al. (2008). Compressed sensing was originally described by Donoho (2006). It is also known as Compressive Sensing or Compressive Sampling. While the terms Compressed Sensing and Compressive Sensing are used mostly synonymously, Compressive Sampling is mainly used in cases where sampling occurs over time.

2.1. Overview

The basic process of CS is visualized in figure 2.1. Similar to traditional compression, one may differentiate a compression and a decompression step.

2.1.1. Compression / Measurement Process

For the compression step in CS, a vector of measurements f with length n is compressed into a shorter vector y of length m by matrix multiplication:

$$y = \Phi \cdot f \tag{2.1}$$

where Φ is called the measurement matrix. Note that, in order to actually compress while measuring, it is crucial to avoid both measuring f completely and explicitly calculating formula (2.1).

2.1.2. Decompression

The compressed vector y needs to be decompressed. As equation (2.1) is under-determined, when trying to gain f from Φ and y, a special trick is needed. The core idea is integrating the knowledge that f is sparse in some known domain such as the frequency space. The transformation may again be expressed as a matrix multiplication:

$$f = \Psi \cdot x \text{ and } x = \Psi^{-1} \cdot f \tag{2.2}$$

where x denotes the transformed vector, Ψ the transformation matrix and Ψ^{-1} the inverse matrix of Ψ. Inserting (2.2) in (2.1), we gain:

$$y = \Phi \cdot \Psi \cdot x \tag{2.3}$$

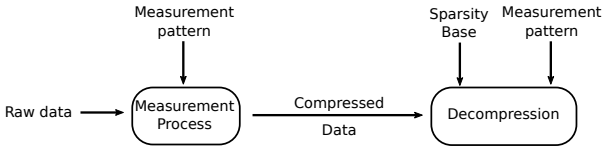


Figure 2.1.: The compressed sensing process in an abstract form.

Now, the sparsity requirement needs to be incorporated. This is expressed by minimizing the ℓ_0-norm which equals the number of non-zero values of a vector and is denoted by $||\cdot||_0$. Which leads us to the decompression operation:

$$x_{estimate} = \underset{x}{\operatorname{argmin}} ||x||_0 \text{ with } y = \Phi \cdot \Psi \cdot x \tag{2.4}$$

As this is a non-convex problem, which usually cannot be efficiently solved, it is common to instead solve the equation:

$$x_{estimate} = \underset{x}{\operatorname{argmin}} ||x||_1 \text{ with } y = \Phi \cdot \Psi \cdot x \tag{2.5}$$

with the ℓ_1-Norm:

$$||x||_1 = \sum_{i=0}^{n} |x_i| \tag{2.6}$$

This is possible because vectors with minimal ℓ_1-Norm have a minimal ℓ_0 as well with overwhelming probability. Equation (2.5) may, e.g., be solved using linear programming. However, many solvers have been used or specifically developed for CS. Some of these solvers will be discussed in section 2.4. In handling noise, equations (2.4) and (2.5) are often modified as section 2.4 will show. Note that decompression is rather expensive in terms of computational effort in comparison to compression. Therefore, it is recommendable to perform decompression at a sink with significant computational resources or using cloud computing.

2.1.3. Matrix design

One task in order to use this compression and decompression scheme is the proper design of the matrices Φ and Ψ. Whereas the transformations behind Ψ are often well known - Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), or Discrete Wavelet Transform (DWT) are used frequently - the corresponding matrix representations are somewhat less popular due to the comparably slow calculation. Therefore, we would like to highlight the possibility of transforming the unit basis vectors of the Cartesian coordinate system to normal space and using the resulting vectors as columns of Ψ (Jensen et al. 2001, pp. 37–39).

In section 2.3, multiple transform matrices will be discussed. The design of Φ is more specific to CS. A key requirement is incoherence between Φ and Ψ. Although, newer findings by Candes et al. (2011) indicated that it is not strictly required. The most astonishing finding here is that dense random matrices for Φ of various designs meet the requirement independent of the transformation Ψ. This means that the sparsity promoting basis does not necessarily need to be known while compressing the data. Furthermore, it has been shown that the sparse Φ matrices mentioned in section 2.1.1, which may be constructed randomly just like the dense matrices, often suffice. The following section 2.2 provides more details on the measurement process.

2.2. Measurement Matrices and the Measurement Process

The key trick for compression is finding a proper way for gaining y by implicitly calculating equation (2.1). We describe three examples on how this may be accomplished:

Firstly, the most famous example for this is the single pixel camera, where the matrix calculation is done implicitly by measuring the total intensity of light reflected by an array of switchable mirrors with a single light sensor (Takhar et al. 2006). Secondly, Analog-to-Information Converters (AICs) may be used where in a first step, analog sensor readings are processed according to equation (2.1) in an analog circuit and in the second step, the resulting values are read by an Analog-to-Digital Converter (ADC) at a reduced rate (Kirolos et al. 2006). Thirdly, a more trivial example is a sparse matrix Φ with a single non-zero value per row. In that case, the measurement process degrades to sampling less frequently and at random points of time. This third case is the one where sub-Nyquist sampling may be achieved in a direct way. The success of these sparse measurement matrices was shown by (Rudelson et al. 2006) for DFT and Gaussian transforms. Taking such random samples still leaves some design decisions: firstly, the pattern of the samples needs to be chosen, which plays a crucial role for multiple topics of this thesis, especially for CFE in section 4.1. Secondly, of course the number of samples must be selected which we discuss here from a practical perspective.

2.2.1. Choice of the Compression Ratio

Considering sparse samples, a typical CS decompression task is the reconstruction of some physical signal $x(t)$ in a given timespan T from M measurements within this timespan. For simplification, here we assume

only the time domain and only sparse sampling matrices. However, the same holds true in a non-time domain (e.g. spectra) and similar problems occur with dense sampling matrices (e.g. gaussian matrices).

A problem, which plays a role in this thesis, arises from the fact, that $x(t)$ won't be handled as a continuous function but rather as a time-discrete vector x of N values at equidistant points in time. So during system design it does not only need to be decided how many measurements M are taken but also how many values N shall be available after the reconstruction. In the non-CS case, only N has to be chosen. Apart from technical limitation, the only limitation comes from the Shannon-Nyquist theorem that requires the sampling rate to be twice as high as the highest relevant frequency in the signal:

$$\nu_{OS} > 2 \cdot \nu_{sig} \tag{2.7}$$

Which allows us to determine N:

$$N > 2 \cdot \nu_{sig} \cdot T \tag{2.8}$$

The important part here is that there is no fundamental upper limit for N.

For CS, the following condition has been found for M (Candes et al. 2008; Donoho 2006):

$$M \geq C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot log(N) \tag{2.9}$$

So instead of depending on the maximum frequency of the signal, the number of frequencies S and the number of reconstructed values N must be considered. Therefore, CS does suffer from choosing N too high for a given signal and number of measurements M. For this reason, when N is chosen too high, even a simple interpolation can outperform CS. Evaluation results in such cases can be misleading because the compression ratio M/N seems extremely good due to the high value of N and the reconstructed signal also looks pretty good.

Idea of CS-Interpolation Hybrid

In many cases, N can indeed be chosen arbitrarily. In these cases, the maximal frequency may be estimated or extracted from existing data. N should then be chosen only slightly above the lower bound set by the Nyquist-Shannon theorem (see equation (2.8)). In some cases, a specific N may be required in the application. Here, one may consider introducing an interim sampling frequency ν_I or number of samples U. Then, the decompression generates a vector of U values, which is afterwards upsampled to N values via some interpolation method, e.g. linear interpolation. This scenario is also useful for better visualizing the problem.

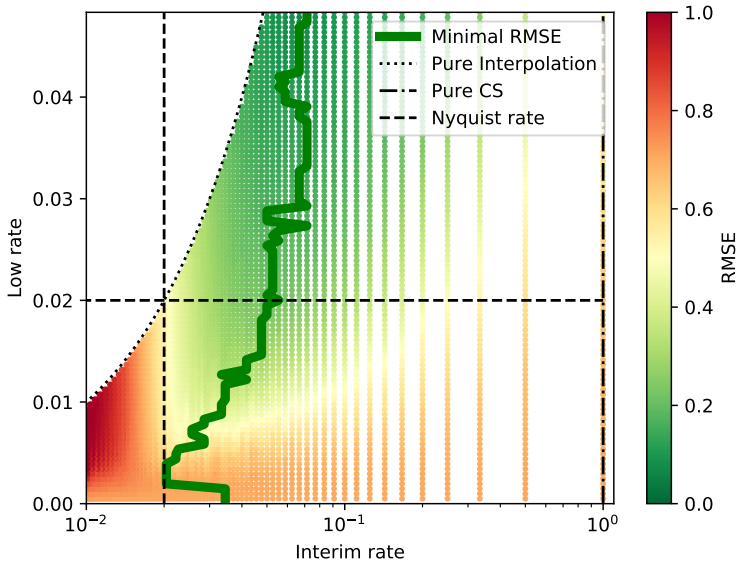


Figure 2.2.: Influence of interim rate and measurement (low) rate on reconstruction quality.

Numerical Evaluation

Rather than just setting the U as suggested we varied it in a simple experiment to demonstrate this. We used a sine wave with a fixed frequency of 0.01 Hz and a fixed target sampling rate of 1 Hz. So it is drastically oversampled. N is fixed to 2048 values. We varied the number of samples M chosen randomly from the N original values. We decompressed this vector via CS using a DFT matrix and the solver SL0 (see section 2.4.2). For the decompression we again use different vector lengths U. Then a vector of N values is generated by upsampling with linear interpolation. The difference between the resulting vector and the original vector is calculated by means of the Root Mean Square Error (RMSE).

The result is shown in figure 2.2. Instead of giving the raw values of M and U, the interim rate and the average measurement rate (here: low rate) are used on the axes. At the right side of the plot, the interim rate is equal to the original rate. This is the case of just using CS to estimate the original vector. On the left, along the border of the evaluation region, indicated by a dotted line, the measurement rate is equal to the interim rate which corresponds to just using linear interpolation. Note, that the Root Mean Square (RMS) of the sine wave is approximately 0.707.

Only when the RMSE is significantly below this value, the reconstruction becomes better than guessing, e.g., the mean. This is reflected by the colors. So, the green region indicates the region of the parameter space where a suitable result can be obtained. Taking a look at the results along the line of pure interpolation results, the RMSE increases significantly when the measurement rate drops below the Nyquist rate shown as a dashed line. This is just as expected from the theorem.

On the right side, the results are rather bad which demonstrates the oversampling problem. However, the results here are still better than linear interpolation when the sampling rate is low. The interim rate which gives the lowest RMSE for a given measurement rate can be found along the solid green line. It clearly should be chosen higher than the Nyquist rate, and it should be chosen higher for higher measurement rates. It also shows that the RMSE becomes worse when the interim rate is chosen too high.

Overall, this shows that it is crucial to avoid oversampling during reconstruction by carefully choosing not only M but also N and thereby the compression ratio. Especially if sensing above the Nyquist rate, CS might even lead to worse results than ordinary sampling if this is not obeyed. Understanding this dependency is crucial for comprehending some of the evaluation results.

2.3. Transform Matrices

Finding a proper transform domain is critical for designing a successful CS solutions. In the beginning, mainly commonly used transforms for signal processing were considered. Some of these are listed in sections 2.3.1–2.3.4. Later on, the construction of transforms became more creative, adapting CS to more kinds of data. There are even approaches for CS with unknown transform, called blind compressed sensing, e.g., by Gleichman et al. (2011). Several transforms are considered in this thesis, ranging from the simple addition of signals in different transform bases (section 2.3.5) over the combinations of different transforms per dimension (section 2.3.7) and modeling similarity between signals (section 2.3.8) to learned transform bases (section 2.3.9).

2.3.1. Discrete Fourier Transform (DFT)

The DFT may be one of the best known sparsity promoting transforms. This short summary is based on the explanation by Burger et al. (2016, pp. 469–479). The DFT is used in a wide range of fields, especially in signal processing. It makes use of the fact that periodic signals can be represented as a weighted sum of base periodic signals of different

frequencies. For the DFT, the complex roots serve as periodic signals:

$$x[n] = \sum_{u=0}^{N-1} f[u] \cdot e^{i \cdot 2 \cdot \pi \cdot n \cdot u / N} \text{ with } 0 \leq n < N \tag{2.10}$$

This can be expressed in matrix form $x = \Psi \cdot f$ with the following matrix elements:

$$\psi_{n,u} = e^{i \cdot 2 \cdot \pi \cdot n \cdot u / N} \tag{2.11}$$

DFT and Fourier transform have the advantage that they are elegant and well-understood in mathematics which makes them suitable for proofs such as the one by Rudelson et al. (2006). Very efficient implementations of the DFT, named Fast Fourier Transform (FFT), such as the one by Cooley et al. (1965), are widely available in libraries. One drawback of the DFT is that the transformed vector is complex even for purely real valued input data. This can be problematic for practical applications. E.g., in classic compression one may transform input data and then only transmit or store the most significant entries of the transformed vector. As these entries are complex valued, typically more bits are required for storing or transmitting the values. This does not affect CS because real valued measurements are transmitted. The transformations which require complex numbers are only performed during decompression. However, some solvers or their implementations do not support complex numbers.

2.3.2. Discrete Cosine Transform (DCT)

The DCT (Burger et al. 2016, pp. 503–509) and similarly the less used discrete sine transform offer a solution to this problem, using the cosine or sine as purely real-valued periodic base functions. The transform is defined as follows:

$$\begin{aligned} x[n] = & \sqrt{1/N} \cdot f[0] \cdot \cos\left(\pi \frac{u \cdot (2 \cdot n + 1)}{2N}\right) + \\ & + \sqrt{2/N} \cdot \sum_{u=1}^{N-1} f[u] \cdot \cos\left(\pi \frac{u \cdot (2 \cdot n + 1)}{2N}\right) \text{ for } 0 \leq n < N \end{aligned} \tag{2.12}$$

This can be expressed in matrix form $x = \Psi \cdot f$ with the following matrix elements:

$$\psi_{n,u} = \begin{cases} \sqrt{1/N} \cdot \cos\left(\pi \frac{u \cdot (2 \cdot n + 1)}{2N}\right) & \text{for } u = 0 \\ \sqrt{2/N} \cdot \cos\left(\pi \frac{u \cdot (2 \cdot n + 1)}{2N}\right) & \text{for } u > 0 \end{cases} \tag{2.13}$$

The DCT is very popular for classic compression tasks. It is, e.g., a core component of Joint Photographic Experts Group (JPEG) image compression.

2.3.3. Discrete Wavelet Transform (DWT)

A major drawback of DFT and DCT is that the transformed vector only expresses frequency information and no time information. An early approach for circumventing this is using a windowed version of the DFT. However, a resolution trade-off always needs to be made - it is impossible to measure the frequency of very slow components and the time of short-term components at once. (Kaiser 2011, p. 60)

The DWT offers a sophisticated solution to this problem and is also being applied for compression, e.g., in JPEG-2000. There are multiple ways of introducing wavelets. As the building of different transforms is particularly useful for CS, the approach via the lifting principle seems most appropriate, which has also been used in the book by Jensen et al. (2001) which served as the main source for this section.

In this approach, two operations are used, *prediction* and *update*. These operations are applied to pairs of neighboring elements of the input vector. These steps differ depending on the wavelet family. In the case of *Haar* wavelets, the *prediction* is that the values remain the same and the difference from this *prediction* is calculated. The *update* calculates the mean of the two elements.

Applying these operations to a vector of N elements generates $N/2$ elements that result from the *prediction* and $N/2$ elements that result from the *update*. Note, that elements resulting from the *update* contain a down-sampled or low-pass filtered approximate version of the original vector, they are also called approximation coefficients. Similarly, the elements resulting from the *prediction* contain local details and are often called detail coefficients. These terms are used, e.g., in the Python library *PyWavelets*[1] which is the library used for all wavelet-related calculations in this thesis.

The vector consisting of these coefficients is called the single level DWT result. The same operations can be performed on the approximation coefficients again, leading to $N/4$ approximation coefficients and $N/4 + N/2$ detail coefficients. This can be repeated until only one approximation coefficient remains, that contains the mean of the original vector.

The vector resulting in the end is the result of the maximum level DWT. Unless otherwise specified, we refer to this transformation when mentioning the DWT.

Note, that the vector at each level can be expressed as a linear combination of the vector elements of any of the previous levels. Therefore, it is a linear transform. It can be reversed and expressed in matrix form. We refrain from supplying formulas for the matrix forms of all the matrix families. However, we would like to highlight that the matrix can be obtained by applying the transformation to the vectors of an identity matrix as demonstrated by Jensen et al. (2001, pp. 37–39).

[1] `https://pywavelets.readthedocs.io`, visited on 2021-11-15

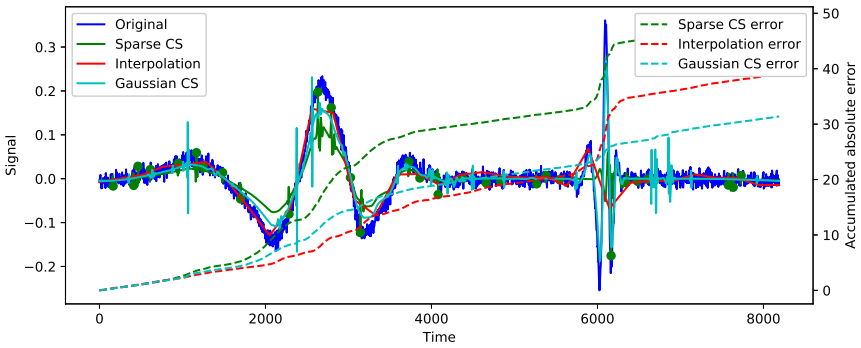


Figure 2.3.: Comparison of linear interpolation and CS for an exemplary wavelet signal.

A note on DWT in combination with sparse sampling matrices

For temporal sampling, operators or matrices that only require few samples at random points in time as described, e.g., by Charbiwala et al. (2009) is of particular interest because it usually requires less special hardware than dense measurement matrices. The feasibility of using sampling at random instances has been proven by Rudelson et al. (2006) for the DFT.

We will show at a small example, that the combination of certain DWTs with this sparse random sampling might actually pose a problem. The example is shown in figure 2.3 consisting of a signal which is 2-sparse in Daubechies-4 wavelet space. A CS reconstruction based on sparse random sampling and a linear interpolation based on roughly the same number of samples are shown. The corresponding accumulated absolute errors are shown as well. The error clearly increases steeper for the CS-reconstruction. For the longer, lower frequency wavelet on the left, linear interpolation is already sufficient making CS superfluous. At the shorter, higher frequency wavelet on the right, there are just not enough samples for successful reconstruction. The same problem does not occur for high frequency continuous oscillations when using DCT or DFT as there are more samples due to the spreading across the whole time. Therefore, the big advantage of DWT, the combination of frequency and time information is somewhat lessened when performing CS with sparse sampling matrices. Note, that when using one of the non-sparse random matrices, e.g., a Gaussian matrix, the reconstruction of the short wavelet is pretty good.

Being restricted to dense matrices for actual CS, imposes a limitation to more specialized hardware solutions for the measurement process such as the single-pixel camera proposed by Takhar et al. (2006) with an ar-

ray of mirrors with adjustable reflectivity or Compressive Wireless Sensing (CWS) proposed by Bajwa et al. (2006) which uses constructive interference of radio waves.

2.3.4. Difference Matrix[2]

For piece-wise constant vectors, a matrix that describes f as the element-wise difference of x or inversely x as the cumulative sum of f similar to the *Horz-diff* matrix (Quer et al. 2009) may be sufficient:

$$\Psi_D = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & \dots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \tag{2.14}$$

Note, that this may also be expanded to the difference of the difference similar to the second derivative for functions $\Psi_{D2} = \Psi_D \Psi_D$ or even to any higher degree $\Psi_{DN} = \Psi_D^N$.

2.3.5. Building Custom Transforms

While the typical introductions to CS may raise the suspicion that only few real-world signals are actually sparse under one of the listed transforms, transforms can be built quite flexibly as a simple example will show. Previous works that hint at this are, e.g., the one on Compressive Data Gathering (CDG) (Luo et al. 2009) which handled anomalous readings similarly to the example considered here and Kronecker Compressive Sensing (KCS) (Duarte et al. 2012) and Distributed Compressive Sensing (DCS) (Baron et al. 2009) which also heavily use the possibility of constructing special transforms. Learned transforms, such as via K Singular Value Decomposition (K-SVD) (Aharon et al. 2006) are also an extreme example for the freedom in building transforms.

A main aspect to keep in mind is that Ψ is not limited to a quadratic form. Or in other words, the vector in sparse domain does not need to have the same length as the one in the ordinary domain. When a signal is known to be the sum of two components ($x = x_A + x_B$) and there are sparse representations f_A and f_B of each component ($x_A = \Psi_A f_A$ and $x_B = \Psi_B f_B$) a solution for x may be found using the combined transform matrix and sparse representation:

$$\Psi = \begin{bmatrix} \Psi_A & \Psi_B \end{bmatrix} \text{ and } f = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{2.15}$$

[2]The content of section 2.3.4 has been previously published (Hänel et al. 2019).

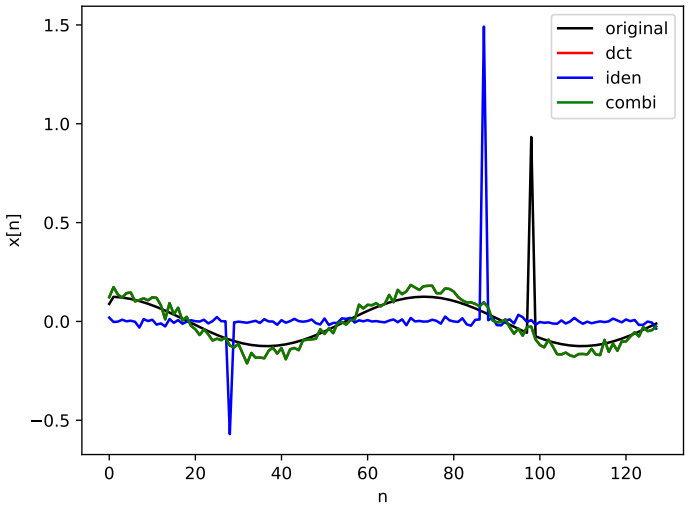


Figure 2.4.: The signal and its reconstructions with $M = 11$. *dct* exactly covers *combi* here.

Note the notation for constructing larger matrices and vectors from stacking smaller matrices or vectors vertically or horizontally. This notation will be used extensively within this section. The resulting Ψ can be used just like any other transform matrix in equations (2.4) or (2.5).

Example

Now we consider an example of a sine signal and an abnormal sample. An appropriate Ψ then consists of a DCT matrix and an identity matrix. We have $N = 128$ and vary M. The measurement matrix is a Gaussian. The solver is Smoothed ℓ_0 (SL0). The original and the reconstructions for different values of M are shown in figure 2.4 and 2.5.

The figures do not only contain the reconstruction assuming the combined Ψ (*combi*) but also those assuming a pure DCT matrix (*dct*) and a pure identity matrix (*iden*). The RMSE was calculated for more values of M as shown in figure 2.6.

Generally, the RMSE falls with increasing M - the more data, the better the reconstruction. The pure identity matrix makes a good reconstruction impossible. Sometimes the abnormal reading is captured quite well as in figure 2.5. However, it is impossible to reconstruct the sine wave - the other peaks in the reconstruction are not helpful.

For DCT and the combined Φ, the solutions are more interesting. At $M = 11$ and higher, the reconstruction improves significantly as the step in figure 2.6 indicates. Figure 2.4 shows the reason for this step: From here on, the sine wave gets reconstructed quite well. A second step occurs

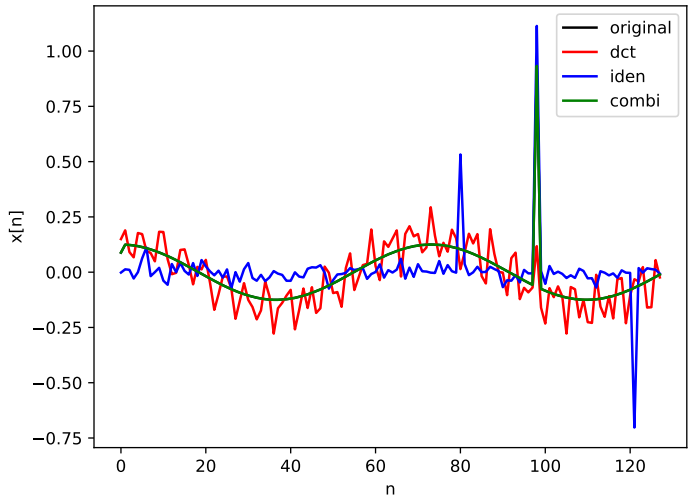


Figure 2.5.: The signal and its reconstructions with $M = 28$. *combi* exactly covers *original* here.

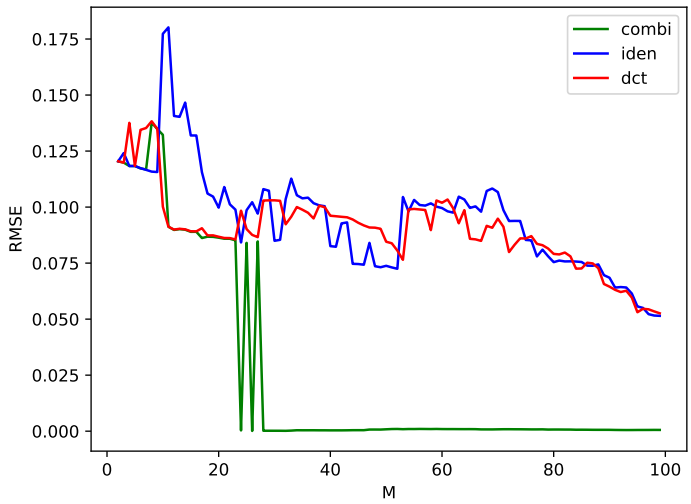


Figure 2.6.: The resulting RMSE at different values of M

at approximately $M = 28$: the combined solution becomes almost perfect while the DCT solution becomes worse. Figure 2.5 reveals the reason: the combined solution now captures the abnormal reading as well. The DCT solution suffers from over-fitting: it attempts to reconstruct the abnormal reading but thereby worsens the remaining solution.

Discussion

The major observation here is that the combined Ψ improved the reconstruction at a very low cost as it is about as good as the DCT reconstruction at low M and significantly better at high M. Just the calculation time increases which has not been evaluated here. With this relatively simple custom matrix already showing such a drastic improvement, it becomes clear that building custom transform matrices is a key ingredient for designing successful CS solutions.

2.3.6. Wavelet Packet and the Best Basis

While the DWT already offers many different transforms, due to the different families and the different levels, even more transforms can be generated when switching to the wavelet packet (Jensen et al. 2001, pp. 87–97). While for the normal DWT, only the approximation coefficients were processed when adding the new layers. The same can be done for the detail coefficients, leading to the full wavelet packet decomposition of the signal if repeated up to the maximal level.

In a next step, one may vary the number of levels for different parts of the input vector. The choice is rather arbitrary. However, for a given metric, e.g., the reconstruction quality or achievable compression ratio, a best suitable choice may be determined. This is called the best basis. It has been used for CS by Peyré (2010) and Bi et al. (2015). These approaches have not been used in this thesis because the overhead in selecting the basis does not seem justified.

2.3.7. Kronecker Compressive Sensing (KCS)

The term Kronecker Compressive Sensing (KCS) has been formed by Duarte et al. (2012), who investigated this approach in detail. The core idea of KCS is the ability to form transform and measurement matrices for data with any number of dimensions. In this thesis, we focus solely on the construction of transform matrices based on KCS, not the construction of measurement matrices. The transform matrices for the individual dimensions $\Psi_1, \Psi_2, \ldots, \Psi_D$ are combined in a single transform matrix Ψ, making use of the Kronecker product $\otimes$:

$$\Psi = \Psi_1 \otimes \Psi_2 \otimes \cdots \otimes \Psi_D \tag{2.16}$$

This allows for any combination of transform matrices to create a particularly well-suited sparsifying base for multi-dimensional signals. Note, that this also offers a very compact formulation for the construction of two-dimensional DFT, DCT, DWT, etc. matrices based on the corresponding one-dimensional matrices. The Kronecker product creates a matrix consisting of the second matrix multiplied with each element of the first matrix:

$$C = A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \ddots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix} \tag{2.17}$$

So, if A is of size $m \times n$ and B of size $o \times p$, then C is of size $(m \cdot o) \times (n \cdot p)$.

2.3.8. DCS-JSM-1[3]

Distributed Compressive Sensing (DCS) (Baron et al. 2009) encompasses three different models for expressing the similarity between vectors measured at different locations. These models are called Joint Sparsity Models (JSMs). JSM-1 assumes a sparse common component and a sparse additive innovation per location, JSM-2 assumes a shared support but different coefficients, and JSM-3 assumes a non-sparse common component and a sparse additive innovation. Caione et al. (2014) point out, that DCS is usually better suited for realistic WSN applications than KCS.

Among the three sparsity models, JSM-1 seems most suitable for spectra as they are piece-wise constant and the same holds true for the differences. Furthermore, JSM-2 and JSM-3 require customized solvers, whereas JSM-1 is compatible with any solver suitable for Compresed Sensing. For those reasons, we limit our considerations to JSM-1 from here on. In JSM-1, the data for each position is modeled as

$$\begin{aligned} x_j &= x_{\Delta,j} + x_c \\ x_j &= \Psi_j f_j + \Psi_c f_c \end{aligned} \tag{2.18}$$

where $x_{\Delta,j}$ denotes the innovation at position j, x_c the common part, Ψ_j the transform matrix for position j, and Ψ_c the transform matrix for the common part. The second row was obtained by inserting equation (2.2).

For a total of J sensors, vectors combining all positions may be defined:

$$\begin{aligned} x &= (x_1^T, x_2^T, \dots, x_{J-1}^T, x_J^T)^T \\ y &= (y_1^T, y_2^T, \dots, x_{J-1}^T, y_J^T)^T \\ f &= (f_c^T, f_1^T, f_2^T, \dots, f_{J-1}^T, f_J^T)^T \end{aligned} \tag{2.19}$$

[3]The content of section 2.3.8 has been previously published (Hänel et al. 2019).

with x being of length $J \cdot N$, y of length $J \cdot M$, and f of length $(J+1) \cdot N$. This allows formulating a joined Ψ of size $(J \cdot N) \times ((J+1) \cdot N)$:

$$\Psi = \begin{pmatrix} \Psi_c & \Psi_1 & 0 & \dots & 0 \\ \Psi_c & 0 & \Psi_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Psi_c & 0 & 0 & \dots & \Psi_J \end{pmatrix} \tag{2.20}$$

With these vectors, a combined Φ with a size of $(M \cdot J) \times (J \cdot N)$ may be defined:

$$\Phi = \begin{pmatrix} \Phi_1 & 0 & \dots & 0 \\ 0 & \Phi_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Phi_J \end{pmatrix} \tag{2.21}$$

with Φ_j being the measurement matrix of the j-th position.

2.3.9. K Singular Value Decomposition (K-SVD)[4]

The transforms considered so far in this section all require some kind of expert knowledge on the data. A transform needs to be selected that transforms to a sparse representation of the data. When example data is available, one possible approach for finding a suitable transform is just trying multiple and selecting the best. But for sufficiently large example data sets, another approach may be chosen which consists of training a transform matrix. One example approach for training a transform matrix is K-SVD (Aharon et al. 2006). Such a transform matrix is also called an overcomplete dictionary. Given the time of the publication, Aharon et al. (2006) did not explicitly present the transform matrix as a part of the CS toolkit. A transform matrix obtained with K-SVD was used explicitly as part of the CS toolkit, e.g., by Cilia et al. (2019).

The learning process of K-SVD consists of two steps which are repeated until the solution converges. In the first step, a sparse solution for the training data based on the current dictionary is determined using a compressed sensing solver, typically Orthogonal Matching Pursuit (OMP) such as in the Python implementation[5] used in this thesis. In the second step, the dictionary is improved by replacing elements of the dictionary using Singular Value Decomposition (SVD).

[4] The content of section 2.3.9 has partially been previously published (Hänel et al. 2021b).

[5] `https://github.com/nel215/ksvd`, visited on 2022-05-19

2.4. Solvers

Clearly, a major challenge lies in calculating solutions for the compressed sensing problems, i.e. performing the decompression. The algorithms for this tasks are called solvers. Intensive research work has been spent on new solvers that provide improved results, provide them faster, or concentrate on niche applications. SL0 is the mostly used solver in this thesis, simply because it turned out superior whenever we compared solvers empirically for the problems considered in this thesis. However, especially for reconstruction quality estimation, we also consider some other solvers, namely ℓ_1 magic, LASSO, BCS, and OMP, listed here as well. A more complete survey on solvers was given by Crespo Marques et al. (2019).

2.4.1. ℓ_1 magic

The ℓ_1 magic packet (Candès et al. 2005) has become a reference for CS problems because it was the first solver packet that became widely popular. It addresses multiple specialized CS-related problems. Thereof we only consider the variant *Min-ℓ_1 with quadratic constraints* because it addresses the conventional CS problem. Specifically, it solves the following problem:

$$f_e = \underset{f}{\operatorname{argmin}} ||f||_{\ell_1} \text{ subject to } ||\Phi \cdot \Psi \cdot f - y||_{\ell_2} \leq \epsilon \tag{2.22}$$

It handles noise by allowing a discrepancy between the simulated measurement for the solution and the actual measurement limited by ϵ.

Due to the calculation of the ℓ_2 norm, this problem cannot be addressed as a linear program. It can however be addressed as a Second Order Cone Program (SOCP). Candès et al. (2005) solved it with a log barrier approach. The penalization of the log barrier is increased iteratively and a solution is found by a Newton search in each iteration. The solver usually requires three parameters: the ϵ from equation (2.22), a value called μ that determines the rate for increasing the barrier, and the tolerance for the Newton search. The number of iterations in the Newton search is limited to 50 because it is guaranteed to require few steps. The number of iterations of the log barrier is calculated from the other parameters. This feature is deactivated for the parametrization. Instead, the number of iterations is varied, introducing a fourth parameter.

2.4.2. SL0

The solver Smoothed ℓ_0 (SL0) has been proposed by Mohimani et al. (2009). The core idea is to start with the ℓ_2 solution and iteratively approach a sparse solution. For this purpose, they introduce the following

Listing 2.1: Pseudocode of the algorithm SL0

```
#Input: Φ, Ψ, y
#Parameters: σ_min, α, μ_0, L
#Initialization:
A = Φ · Ψ
f = A^{-1} · y
σ = 2 · max(|f|)

while σ > σ_min:
    for i in 1...L:
        #Gradient descent:
        δ = f · exp(-|f|^2 / σ^2)
        f = f - μ_0 · δ
        #Fulfilling measurement again:
        f = f - A^{-1} · (A · f - y)
    σ = σ · α
return f
```

Gaussian-based metric:

$$||f||_\sigma = N - \sum_{n=1}^{N} \exp\left(\frac{-|f_n|^2}{\sigma^2}\right) \tag{2.23}$$

For $\sigma \to 0$, this approaches the ℓ_0 norm, similarly to how the Dirac delta function can be approximated by a Gaussian. Iteratively, two steps are performed: In the first step, the gradient for optimizing the norm is calculated and the solution is modified in that direction. In the second step, the solution is modified to match the measurements again. In the next iteration, the steps are repeated. In some iterations, σ is reduced, approaching the ℓ_0 norm. The algorithm stops, when σ has been reduced to a pre-defined value. The pseudo-code for the solver is given in listing 2.1. This includes some modifications in comparison to the original paper as it is based on the actual implementation[6] by the authors.

There is a total of four input parameters: the desired σ_{min} which determines how close the ℓ_0 shall be approximated. A smaller value generally leads to better results at prolonged calculation time. The rate at which σ is reduced is given by α. Higher values lead to better results at prolonged calculation time. The μ_0 determines the step size in the gradient descent. The L sets the number of iterations without decreasing σ.

[6] `http://ee.sharif.ir/~SLzero/`, visited on 2022-05-19

The major advantages of SL0 are its simplicity, its speed, and that it can handle complex number and is therefore suitable for DFT matrices. A major disadvantage is the relatively high number of parameters.

2.4.3. LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) problem formulation by Tibshirani (1996) is actually much older than CS. However, it can produce solid results especially for noisy data. The main idea here is using a single metric that combines the ℓ_1 norm of the solution and the difference from the measurements via the ℓ_2 norm. The problem to solve has the following form:

$$f_e = \underset{f}{\operatorname{argmin}} \left(\frac{1}{2 \cdot N} ||\Phi \cdot \Psi \cdot f - y||_{\ell_2}^2 + \lambda \cdot ||f||_{\ell_1} \right) \tag{2.24}$$

Only if λ is chosen properly, both norms are adequately considered.

Strictly speaking, LASSO is not a solver but rather a problem formulation. Here, we consider the solution via coordinate descent, which is common in machine learning libraries (Pedregosa et al. 2011). Besides λ, there are two parameters: the maximum number of iterations and a tolerance value, that determines when improvements become insignificant.

The main advantage of LASSO is the wide availability in libraries due to its age and it being used in machine learning. The main disadvantages are that it is relatively slow and the results are often worse than those of the other algorithms.

2.4.4. Bayesian Compressive Sensing (BCS)

BCS has been proposed by Ji et al. (2008). Being a Bayesian approach, it does not only return an estimate for f but also a covariance matrix which estimates the accuracy of the estimate. The algorithm is shown in listing 2.2. As the stopping condition has not been clearly stated in the original paper, multiple stopping conditions have been introduced to avoid infinite loops. The choice of τ is relatively simple. It should be close to the maximal allowed value of the data-type in use. It stops the loop when an element of α becomes too high. This is necessary in actual implementations because α increases indefinitely and may increase too fast to purely rely on the main stopping condition.

The main advantage of BCS is that it supplies the user with the covariance matrix and that it requires nearly no parameter optimization. A disadvantage is that it may be outperformed by a different, well-parameterized solver.

Listing 2.2: Pseudocode of the algorithm BCS

#Input: Φ, Ψ, y
#Parameters: α_i, τ
#Initialization:
$M = length(y)$
$N = width(\Psi)$
$\alpha = (\alpha_i \dots \alpha_i)^T \in \mathbb{R}^N$
$\alpha_0 = \alpha_i$
$\Sigma_{old} = \begin{bmatrix} \tau & \dots & \tau \\ \vdots & \ddots & \vdots \\ \tau & \dots & \tau \end{bmatrix}$ with size $N \times N$
$\Sigma = \Sigma_{old}/2$
$A = \Psi \cdot \Phi$

while $\max(\alpha) < \tau$ and $\max(\Sigma) < \max(\Sigma_{old})$:
    #Step 1:
    $\mu_{old} = \mu$
    $\Sigma_{old} = \Sigma$
    $\Sigma = \left(\alpha_0 \cdot A^T \cdot A + \text{diag}(\alpha)\right)^{-1}$
    $\mu = \alpha_0 \cdot \Sigma \cdot (A^T \cdot y)$
    #Step 2:
    $\gamma_n = 1 - \alpha_n \cdot \sigma_{nn}$
    $\alpha_n = \gamma_n / \mu_n^2$
    $\alpha_0 = \frac{M - \sum_{n=1}^{N} \gamma_n}{||y - A \cdot \mu)^2||_{\ell_2}^2}$

return μ_{old}, Σ_{old}

2.4.5. OMP

Just like LASSO, OMP dates back to the era before CS. The term has first been used by Pati et al. (1993). It has become widely used in CS, e.g., in JSM-3 of DCS (Baron et al. 2009) and in K-SVD (Aharon et al. 2006). Due to its age, similarly to LASSO, it has found its place in libraries such as scikit-learn. It is closely related to a normal Matching Pursuit and consists of an iterative approach. OMP iteratively adds elements from a dictionary (the matrix $\Phi \cdot \Psi$) in order to create the observed signal. In an iteration, OMP first determines an unused element of the dictionary, that correlates strongly with the error in recreating the signal. This element is then used to further improve the recreation. This is repeated until the remaining error is sufficiently small.

According to the description by Pati et al. (1993), there are two ad-

justable parameters: a parameter α that adjusts the threshold for how strong the new elements have to be correlated with the remaining error and the abort condition, i.e., the maximal remaining error δ. The implementation in scikit-learn does not know the parameter α. Instead, there is a target sparsity which can replace the δ as a stopping condition.

2.5. Conclusion and Recipe

In this chapter, an overview of CS was given, which allows for the formulation of a recipe for designing CS solutions.

1. The first and often most challenging step is designing a measurement process which actually helps by reducing the measurement effort.
2. The second step is finding a suitable transform, this is the step most crucial for designing an effective solution, because this is where the most variations are possible as discussed in section 2.3.
3. The third step of choosing a solver can be kept relatively simple as discussed in section 2.4.
4. The fourth step is finding a suitable parametrization: the compression ratio needs to be chosen as discussed in section 2.2, the choice of the transform needs to be refined, and the solver needs to be parameterized.

In sections 4 and 6, we will apply this recipe, designing multiple CS-based solutions for applications in agriculture.

3. Background and Related Work

Before engineering compressed sensing based solutions for precision agriculture in chapter 4, this chapter gives an overview of related approaches and some background information. Section 3.1 discusses the relevance of vegetation spectra. In section 3.2, an overview of approaches for estimating scalar fields from moving sensors is given which is related to the first of the solutions, CFE. In section 3.3, an overview for making use of CS in WSNs is given, putting the choices for the second solution, the M2HSN, in context. Previous approaches for the main task in the M2HSN, the increase of the spectral resolution, are explained in section 3.4. Lastly, existing approaches for estimating reconstruction quality in CS are listed in section 3.5.

3.1. Spectral Information in Vegetation

In this section, we give a short overview of reflectance spectra of vegetation and the information which may be derived from the spectra. Understanding the significance of these spectra is crucial for the main approach developed in this thesis, the M2HSN.

3.1.1. Biological background

The spectra of plants in and around visible light are dominated by the components which take part in photosynthesis. The core components and their contribution to the spectra were, e.g., compiled by Blackburn (2006) and are summarized here briefly. Photosynthesis absorbs visible light and uses the energy for the growth of the plant. Interacting with visible light, this also dictates the color of the plant. The components which influence the color, are called pigments. The most important pigments for photosynthesis are the group of Chlorophylls, especially Chlorophyll a and Chlorophyll b.

Both types mainly absorb red and blue light. Green being less absorbed makes the plants appear green. In the Near Infra Red (NIR), the absorbance of Chlorophyll a and b is much weaker, leading to a steep decrease of absorbance towards wavelengths above 700 nm, called the red edge (Mulla 2013).

While the spectra absorbance peak in the blue is rather narrow for the Chlorophylls, the next group of pigments contributing to photosynthesis,

the Carotenoids, absorb light in a wider range, further stretching to the greens. There are two subtypes, the Carotenes and the Xanthophylls. The most important and best known representative of the Carotenes is Beta Carotene. The spectral range of the light which is used by photosynthesis stems from the combined absorption ranges. It is also called Photosynthetically Active Radiation (PAR). There are also further, less dominant and less researched pigments such as the Anthocyanins.

3.1.2. Vegetation Indices[7]

The spectrum of plants contains plenty of information on their status. A basic example is assessing how green a plant is, which the human eye is capable of. In the remote sensing community, this was expressed more formally with vegetation indices such as the Visible-Band Difference Vegetation Index (VDVI), the Normalized Green-Red Difference Index (NGRDI), and the Normalized Green-Blue Difference Index (NGBDI) (Du et al. 2017) which are based on a red, a green, and a blue band or a subset thereof. However, reflection of vegetation is much higher in the NIR. Therefore, the Normalized Difference Vegetation Index (NDVI) (Tucker 1979) which uses a red band and a near infrared band became widely used. It measures the red edge and has also been used in the calculation of another vegetation index, the Normalized Difference Red Edge Index (NDRE)(Barnes et al. 2000).

However, the exact choice of the bands used for the calculation of these vegetation indices is not defined, depends on the sensors, and has a significant impact on vegetation indices which was shown for the NDVI (Huang et al. 2013). Gaining higher resolution spectra allows for choosing the exact bands for the calculation of indices later on. An online database (Henrich et al. 2009) lists more than 200 vegetation indices based on many different bands. While low resolution spectra only contain small subsets of bands for small subsets of these indices, all indices which lie in the spectral range of a high resolution spectrum are calculable from it. While the individual indices typically require relatively few bands, for the determination of the inflection point of the red edge, multiple bands at high resolution around the red edge are required (Horler et al. 1983).

3.1.3. Classification [8]

Besides the expansion to a greater selection of vegetation indices, another application that benefits from high resolution spectra is the classification of plant species (Hennessy et al. 2020).

[7] The content of section 3.1.2 has been previously published (Hänel et al. 2021b).
[8] The content of section 3.1.3 has been previously published (Hänel et al. 2021b).

3.1.4. Regression Models

Besides finding different types of the soils or vegetation via classification, a typical task is the derivation of actual physical properties of a plant, such as the Leaf Area Index (LAI), the chlorophyll content, the humidity of the plant, and especially the expected yield in t/ha.

A very common approach is predicting the yield from vegetation indices such as the NDVI with a simple linear regression model (Panek et al. 2021; Mirasi et al. 2021). Especially, when using more bands, it is also possible to use regression models such as Partial Least Squares Regression (PLSR) for estimating physical parameters directly from the bands (Siegmann et al. 2015).

3.1.5. Conclusion

In this section, the composition of plant spectra was discussed, showing that high resolution spectra are desirable for two reasons: firstly, more information may be derived from them. Secondly, lower resolution spectra may be derived from high resolutions spectrum, allowing for the re-use of trained models and the comparability with metrics acquired by different sensors of lower spectral resolution.

3.2. Field Estimation / Interpolation [9]

In the beginning of CS research, the exploitation of sub-Nyquist sampling made time series an obvious field of research. But researchers soon expanded their considerations to the measurement or sensing of two-dimensional scalar fields. Whereas these scalar fields are usually camera images (Takhar et al. 2006) or medical imagery (Lustig et al. 2008), environmental monitoring with static wireless sensor networks attracted some interest as well as discussed in section 3.3. However, the CS-based measurement of an environmental scalar field with moving devices has only been rarely considered and usually a rather high number of devices, which are homogeneously distributed, is considered (Yu et al. 2010; Lin et al. 2010). In contrast to this we pay attention to a – to the best of our knowledge – previously not examined aspect: When exploring a new area with few devices or even just one, the available information is focused only on a small, slowly increasing subset of the area. Therefore, our proposed approach CFE performs not only a compression but also a forecast for the still unexplored area. Alternatively, in conjunction with the sparse sampling mentioned in section 2.2, the compression may instead be seen as an interpolation, similarly to the considerations by Yu et al. (2010) and consequently the forecast as an extrapolation.

[9] The content of section 3.2 has been previously published (Hänel et al. 2017a).

In the robotics community, scalar field estimation has been explored as well (e.g., (Jin et al. 2007)). However, they heavily rely on navigating a robot for maximal information gain. In real-world applications this is often undesirable as movement is predetermined by the main task whereas sensing is just a side task.

3.3. CS in WSNs

WSNs offer a wide range of possibilities for exploiting CS. We provide a classification of the applications of CS to WSNs in figure 3.1. As sensor nodes in WSNs are equipped with sensors and ADCs, special sensing hardware such as Analog-to-Information Converters (AICs) may be used. Furthermore, new Medium Access Control (MAC)-schemes were developed that exploit CS and are applied to WSNs (Mao et al. 2010; Qaseem et al. 2009). We argue that these applications are not specific for WSNs and focus on methods that apply CS directly to the collected data for compression.

These methods can be classified in two classes. Some exploit temporal sparsity whereas the majority of methods exploits spatial sparsity. The class of the spatial methods may be further differentiated into those that aggregate data while routing, those concentrating purely on the localization of events, and those which simply take less measurements. As the majority of methods uses aggregation schemes, we further differentiate between dense and sparse aggregation schemes.

The classes are explained in the remainder of this section.

3.3.1. Temporal

Temporal CS in WSNs exactly coincides with the original idea of CS - sampling is performed at a lower rate thereby reducing both the sampled data as well as the transmitted data. In contrast to the remaining classes, here it is often required to actually sample below the Nyquist-Shannon rate. However, due to the limited resources in wireless sensor nodes, some special requirements arise and research has been spent on meeting these requirements.

3.3.2. Spatial

With WSNs being distributed systems, spatial CS gained significantly more attention in research than temporal CS. We further differentiate using the following subclasses:

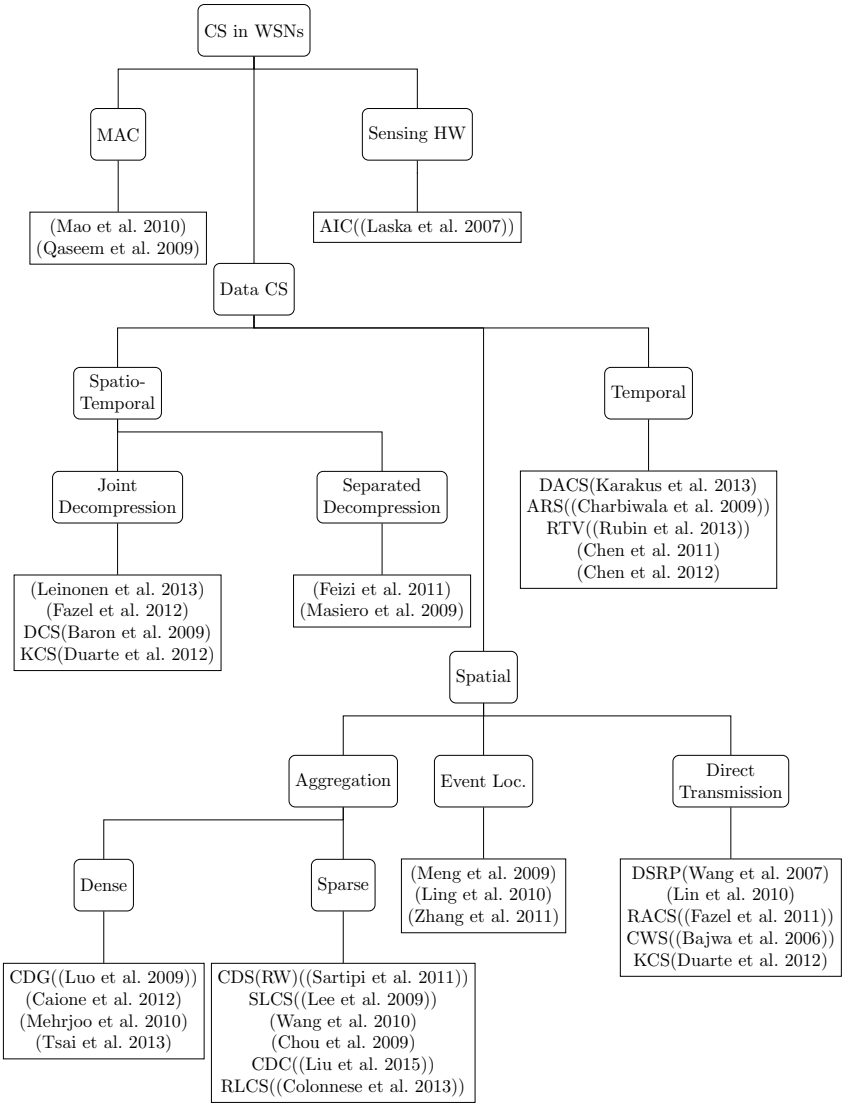


Figure 3.1.: Classification of the application of CS in WSNs.

Aggregation

The algorithms of the most common subclass use aggregation schemes. All those aggregation schemes rely on a common principle. The elements of f are the sensor readings at n different nodes. The calculation of a single element in y, i.e. the dot product of f and a row of Φ is called a projection (Haupt et al. 2008):

$$y_i = \sum_{j=0}^{n} \varphi_{ij} f_j \tag{3.1}$$

where φ_{ij} is an element of Φ. A single projection is calculated by aggregating the sum while routing through the network. Different kinds of routes are possible and common. Either a simple route may be used. A projection may also be sparse, then only a fraction of the network is covered by the route or tree. Note, that for one projection each node only needs to transmit at most one value due to the summation (Luo et al. 2009).

Due to the fact that this is the class of methods that has gained most attention in research, we further differentiate approaches with dense sampling, which are mostly using trees that cover the whole network, and sparse approaches, which use mostly short routes.

Dense Although the general idea of using aggregation for CS was known before as mentioned by Haupt et al. (2008), the first publications with in-depth descriptions of the approach followed in the year 2009. One of the most popular approaches that use dense sampling is CDG (Luo et al. 2009). A well-balanced tree is formed that covers the whole network with the sink as the root node of the tree. In order to gain a snapshot of the sensor readings in the whole network, each node only needs to transmit m instead of up to n sensor readings.

Sparse Instead of performing a full aggregation, random routes through parts of the networks may be chosen, offering spatially sparse measurements. E.g., Spatially Localized Compressive Sensing (SLCS) (Lee et al. 2009) aims for a compromise between sparse sampling and performing dense measurements across the whole network. The key idea is partitioning the network in clusters and calculating projections within these clusters. This lowers the number of hops required for the aggregation.

Event Localization

The event localization approaches are quite different from the other approaches in the way that they do not aim at recovering a rawly measured

field of environmental values but instead at locating rare events within a given area. These events are localized to small areas and thereby sparsity is already given.

The Ψ matrix contains coefficients to model the propagation of the observable signal from the event sources to the sensors. Examples for the event are acoustic or radio wave sources. The Φ matrix is a simple selection matrix which picks the rows of Ψ corresponding to the active sensors.

Direct Transmission

Rather simple approaches are based on just sending measurements directly to the sink. Some special mechanisms can be used in single hop environments. E.g., Fazel et al. (2011) researched transmitting at random times, analyzing collision probabilities and the impact on reconstruction probability. This approach was named Random Access Compressed Sensing (RACS). CWS by Bajwa et al. (2006) creates projections analogously by using coherent transmissions. Sparsely sampling scalar fields, assuming a two-dimensional transform also falls in this category. In this case, any way of transport can be used for the samples. One approach for building such two-dimensional transforms is KCS which was researched by Duarte et al. (2012). Direct Transmission is the application class mainly considered in this thesis with M2HSN falling into this class, despite using a transform from the spatio-temporal class.

3.3.3. Spatio-Temporal

Actual Spatio-Temporal CS has been considered less frequently, although Leinonen et al. (2013) showed that it can be beneficial. Some approaches split the usage of spatial and temporal redundancies in separate steps (Feizi et al. 2011; Masiero et al. 2009). Others decompress the spatio-temporal data jointly in a single step. Transforms which are suitable for this case are, e.g., described in DCS by Baron et al. (2009) and three-dimensional matrices built with KCS(Duarte et al. 2012).

3.4. Increasing spectral resolution[10]

Due to the high costs of hyperspectral sensors, in section 4.2, we instead suggest using sensors with lower spectral resolution for WSNs. Related research mainly focused on images, not single measurements. Creating hyperspectral images from images of lower spectral resolution is uncommon. However, it is possible to perform an image fusion of a panchromatic

[10]The content of section 3.4 has been partially published previously by Hänel et al. (2019) except for section 3.4.2.

image with high spatial resolution and a hyper- or multispectral image with low spatial resolution, which is called pan-sharpening. In the past, researchers mainly focused on multispectral images (Ehlers et al. 2010; Vivone et al. 2015). However, pan-sharpening of hyperspectral images gained more attraction in the last decades spawned by the launch of hyperspectral satellites such as EO-1. Pan-sharpening can also be used for fusing multi- and hyperspectral data when a panchromatic image is first created from the multispectral data.

Two methods that take a slightly different approach are Universal Pattern Decomposition Method (UPDM) (Liu et al. 2009) and Spectral Resolution Enhancement Method (SREM) (Sun et al. 2015) which fuse multi- and hyperspectral data directly without first creating a panchromatic image. Thus, they increase the spectral resolution of multispectral images by incorporating hyperspectral data from a subset of pixels where it is available. SREM still requires a hyperspectral image but generates a hyperspectral image of higher spatial resolution in a larger area than the one of the original hyperspectral image. UPDM was instead used to simulate hyperspectral images based on multispectral images. It only requires few high-resolution spectra to be known a-priori and may, therefore, be adapted for the envisioned M2HSN.

3.4.1. Universal Pattern Decomposition Method (UPDM)

Universal Pattern Decomposition Method (UPDM) (Liu et al. 2009) represents each spectrum as a linear combination of some base spectra such as soil, water, and vegetation:

$$x_j = c_{s,j}x_s + c_{w,j}x_w + c_{v,j}x_v \tag{3.2}$$

with the spectrum x_j at position j, the coefficients $c_{s,j}$ of soil, $c_{w,j}$ of water, and $c_{v,j}$ of vegetation which state the fractions of these components and the corresponding base spectra x_s, x_w, and x_v. This may also be expressed in matrix form:

$$x_j = Xc_j \text{ with } X = \begin{pmatrix} x_s & x_w & x_v \end{pmatrix} \text{ and } c_j = \begin{pmatrix} c_{s,j} \\ c_{w,j} \\ c_{v,j} \end{pmatrix} \tag{3.3}$$

with the matrix of base spectra X and the fraction vector c_j. At a lower spectral resolution, the coefficients remain the same but the spectra change. Equation (3.3) then changes to:

$$y_j = Yc_j \tag{3.4}$$

with the low-resolution spectrum y_j and the matrix of low-resolution base spectra Y.

The matrix of base spectra need to be known in its multispectral version Y and in its hyperspectral version X. Then, the vector of coefficients c_j is gained from the low-resolution spectra using the pseudoinverse Y^{-1} of Y:

$$c_j = Y^{-1} y_j \tag{3.5}$$

Lastly, the coefficients are used to gain the hyperspectral representation x_j at the given location by inserting c_j from equation (3.5) into equation (3.3):

$$x_j = X Y^{-1} y_j \tag{3.6}$$

As only the base spectra, albeit in two representations, are required, the a priori knowledge put into the process is reduced in comparison to pan-sharpening approaches which require hyperspectral images.

3.4.2. K-SVD for spectra[11]

One other research group attempted to apply K-SVD to calculate hyperspectral data and impressively demonstrated its success on RGB images from DSLR cameras (Arad et al. 2016; Arad et al. 2017). The photos considered were those typically taken with such cameras, e.g., landscape and architecture. Here we use a very similar approach. However, the kind of data we consider, i.e. ground-measurements and remote sensing data, is very different from those examples. The spectral range is wider, including the near infra-red which is of particular interest in vegetation applications and the number of bands is not fixed to three. We specifically focus our research on how the approach performs with this kind of data as it is one of the prime application areas for hyperspectral data.

3.5. Quality metrics

Some previous research has been performed on quality metrics before, these mainly focused on methods based on those known from statistics such as cross-validation (Ward 2009), jackknife, and bootstrap (Tygert et al. 2018). These are closely related, as they always use a subset of the measurements and calculate CS solutions for these. They differ in how they select the subset and whether they compare the difference of simulated and actual measurement or whether they measure the convergence of the solution. A second group of approaches encompass solvers that output a quality metric besides the result. A popular representative of these solvers is BCS from section 2.4.4 with its covariance matrix. A qualitative comparison for real-world data was done for jackknife and bootstrap by

[11] The content of section 3.4.2 has been previously published (Hänel et al. 2021b).

Tygert et al. (2018). However, a large-scale quantitative comparison on real-world data of existing approaches and more trivial metrics has not been performed to the best of our knowledge.

3.6. Conclusion

In this chapter, we have outlined the necessity of deriving vegetation information. We provided a systematic classification of CS methods in WSNs and used it for putting the solutions developed in this thesis into context. Furthermore, approaches were summarized that are used in the solutions engineered in sections 4 and 5 or related to these solutions.

4. Engineering Compressed Sensing Solutions for Precision Agriculture

In this chapter, the first step of the recipe for the creation of CS based solutions from chapter 2 is applied for building the concept of CFE in section 4.1 and the concept of M2HSN in section 4.2. For the M2HSN, two variants are presented, the Heterogeneous M2HSN (HeM2HSN) in section 4.3 and the Homogeneous M2HSN (HoM2HSN) in section 4.4. After concentrating on the concept and the algorithms, section 4.5 presents the design decisions for the actual protoype of the M2HSN.

4.1. Compressive Field Estimate (CFE)[12]

With Compressive Field Estimate (CFE), we address the scenario of a harvester that is harvesting a crop field and measures various information with some of this information being spatially distributed scalar fields. Some remote operator is interested in gaining the whole scalar field. Classic approaches to transmit the data encompass transmitting measured values in order as fast as possible or collecting the data and transmitting it in the end in compressed form. The former approach is usually preferable as it allows for a live preview.

However, imagine the case that the time τ to transmit a sample is longer than the time δ between measuring two consecutive samples. Then, the transmitted values quickly lack behind as the data rate is too low. Not even recent technologies such as 5th-generation cellular network (5G) and Long Range Wireless Area Network (LoRaWAN) aim to combine high data rates with good coverage in rural areas: 5G concentrates on higher data rates in small and dense cells (Gupta et al. 2015) which is more suited for urban areas whereas LoRaWAN (LoRa-Alliance 2015) concentrates on offering long ranges suitable for rural areas but it offers low data rates. We expect this problem to persist in the future as improving public land mobile networks infrastructure in rural areas will remain unattractive for wireless providers due to low population densities. Furthermore,

[12]The content of section 4.1 has been previously published (Hänel et al. 2017a).

harvesters acquire more and more information at increasing rates, so that only a small portion of the available bandwidth may be used for each scalar field.

4.1.1. Procedure

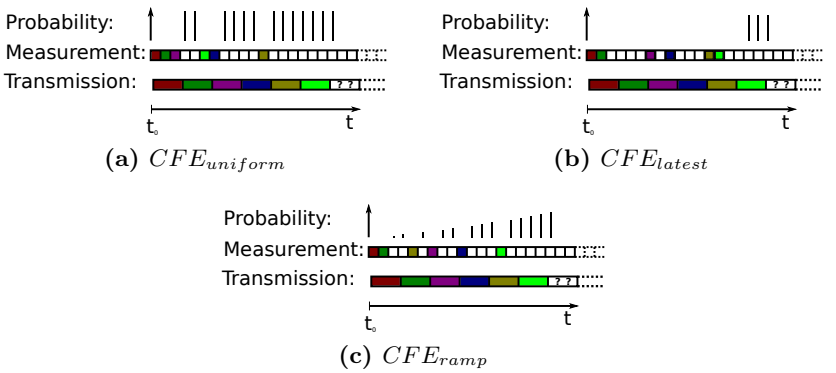
We propose CFE as a new approach that builds upon the toolkit of CS to send and recover data in a way that allows for a faster recovery of the whole scalar field. The core principle is to not simply send samples in order but rather selecting samples for transmitting randomly from the values measured so far. To put it more formally, the harvester begins measuring at t_0, taking a measurement every δ seconds. The sample measured at time t is called x_t. The corresponding coordinates (either geographical or indexes in a grid) are called i_t and j_t . At time T_{now} the harvester has all tuples $(x_{t'}, i_{t'}, j_{t'})$ with $t' \leq t$ in its storage. It sends tuples every τ seconds randomly selected from the available samples. So that at time t there are $(t-t_0)/\delta$ samples in the harvesters storage and $(t-t_0)/\tau$ samples in the storage of the remote evaluation center.

This randomization is the only change required at the harvester. The compressed sensing toolkit is applied at the remote evaluation center. Instead of a time series representation, it uses a spatial representation. For now, let us assume that some preprocessing assures that we get at most one measurement per location and the locations form a two-dimensional grid. With every received tuple we can fill in one cell in the grid: $x_{i,j}$ with $0 \leq i < I$ and $0 \leq j < J$. The total number of grid cells is $N = I \cdot J$. For CS, a vector representation is more desirable, therefore, we introduce a new index $l = i \cdot J + j$ that allows addressing all cells of the grid. The vector of all x_l is called x.

The task of the remote evaluation center is recovering x from the measured x_l. The vector of all measured x_l at time t is called y_t and has length $M_t = (t - t_0)/\tau$. Note that the extraction of y_t from the unknown vector x may be represented with a matrix product: $y_t = \Phi_t x$ where Φ_t is an $M_t \times N$-matrix in which all $\phi_{l',l}$ are set to one where l denotes a received packet in x and l' the same element in y_t. All other elements of Φ are set to zero. So with every received measurement, a new row is appended to the matrix, which results in an identity matrix where random rows have been removed. With this measurement matrix and the vector of received measurements y_t, the remote evaluation center has a common CS-problem to solve (compare equation (2.4)):

$$f_{estimate,t} = \underset{f}{\operatorname{argmin}} ||f||_0 \text{ with } (y_t - \bar{y}_t) \approx \Phi_t \cdot \Psi \cdot f \tag{4.1}$$

Where Ψ now denotes the matrix of the two-dimensional transform which may be constructed from the corresponding matrix of the one-dimensional transform using the Kronecker product (Duarte et al. 2012) as described



(a) $CFE_{uniform}$ **(b)** CFE_{latest}

(c) CFE_{ramp}

Figure 4.1.: Examples and probability densities for different sampling strategies (Hänel et al. (2017a), © 2017 IEEE).

in section 2.3.7. The mean of the measurements $\bar{y}_t$ has been removed as solvers struggle with the reconstruction of the offset. From $f_{estimate,t}$ an estimate for x may be calculated:

$$x_{estimate,t} = \Psi \cdot f_{estimate,t} + \bar{y}_t \tag{4.2}$$

Note that the compression ratio R decreases over time:

$$R(t) = \frac{F}{M_t} = \frac{\tau \cdot F}{(t - t_0)} \tag{4.3}$$

Where F denotes the amount of cells in the grid that are actually part of the field. This naturally leads to an increase of reconstruction quality. At $t_e = \tau \cdot F + t_0$ all samples have been received. Therefore, the compression ratio has decreased to $R(t_e) = 1$ and the reconstruction is perfect as the constraint in equation (4.1) allows only a single solution. However, the normal way would be to abort the transmission earlier, once a sufficient compression ratio has been reached and thereby a lossy compression has been performed.

4.1.2. Random sampling strategy

So far, we have omitted one question on the harvester side: which strategy is used for the selection of values to transmit? At $t_p = F \cdot \delta + t_0$ the harvester has explored the whole area. From here on it may select values from its storage uniformly that have not been transmitted yet. Before t_p, an adequate strategy is required. There are two straight forward approaches: In the first approach (Fig. 4.1a), samples are selected uniformly

from the un-transmitted values in storage at all times. Therefore, we call it $CFE_{uniform}$. In the second approach (Fig. 4.1b), the sample is selected only from samples that have been measured since the last transmission, i.e., all $x_{t'}$ with $t - \tau < t' \leq t$. We call this approach CFE_{latest}.

$CFE_{uniform}$ may lead to poor reconstruction quality as more samples will be selected from the beginning of the measurement leading to a non-uniform distribution over the whole field. A potential problem that may arise when using CFE_{latest} is that the selected values may be a checkerboard-like pattern, only distorted by some jitter. Such a checkerboard pattern is periodic and, therefore, coherent with common transform matrices such as DCT and DFT which leads to poor reconstruction quality (Candes et al. 2008). In order to further explore this trade-off we try a third strategy (Fig. 4.1c) with a probability density that is increasing linearly over the un-transmitted samples in a timely order. Due to the form of the probability density function we call this approach CFE_{ramp}. In figure 4.1c the ramp seems distorted due to the samples which have been transmitted already. Showing only the un-transmitted samples, the ramp would be perfectly linear. The extension to other monotonically increasing probability density functions may be a promising improvement for future work.

4.1.3. Remarks

We would like to make some remarks about the benefits of our approach that go beyond what is covered in the following evaluations: The modification of transmissions only affects the order and not the content of the measurements, so each single transmission keeps a meaning on its own. Building the measurement matrix from the received samples has two key benefits. Firstly, no additional information, such as the seed for generating the measurement matrix (Yu et al. 2010), needs to be exchanged and secondly it does not need to be known which data has been sent originally. The latter can be exploited in multiple ways: If only an approximate field is required, transmissions may be simply stopped earlier. Dealing with lost packets is particularly easy as pointed out by Fazel et al. (2011) and others: If a lossy reconstruction suffices, no explicit retransmit is required - transmission goes on until sufficient packets have been received. If a full reconstruction is desired, some form of Automatic Repeat reQuest (ARQ) mechanism may be used. As duplicate packets or a wrong order of packets do not influence the reconstruction, CFE is very insensitive towards the ARQ system in use. For the estimate just like for the lossy reconstruction, packet loss only leads to a more slowly increasing reconstruction quality.

A special case arises when the harvester crosses areas that have been explored before or stops for some time: as no relevant new values are measured, CFE_{latest} has to select values uniformly instead. The other strategies may proceed as before. This situation presents a relaxation of

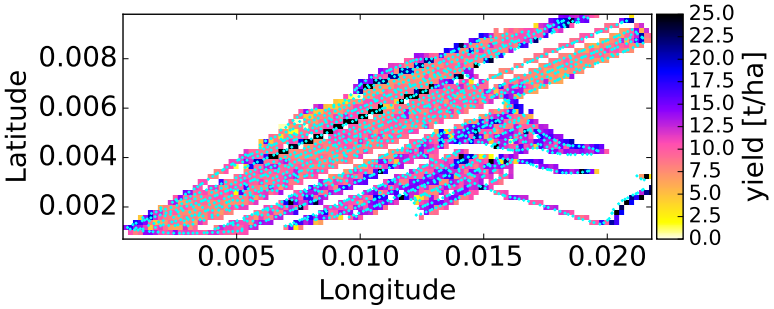


Figure 4.2.: The location of samples and the resulting grid distribution for the crop yield (Hänel et al. (2017a), © 2017 IEEE).

the problem as the gap between measurement rate and transmission rate is reduced. Therefore, in this thesis, we focus on the worst case scenario with no stops.

4.1.4. Preprocessing

Before we can actually evaluate CFE's performance, some preprocessing is required. The evaluations in the sections 6.2, 7.1, and 7.6 are based on traces from a harvester working on a field in an area of approximately 1.7 km by 0.5 km. The harvester collected telemetry data every five seconds. We concentrate on data that describes spatial distributions across the field, e.g., the yield and the humidity of the crop. As most spatial compression algorithms concentrate on data in a grid, we build such a grid in a preprocessing step as follows: First an empty grid is initialized with a fixed cell size. Then, we iterate the trace of samples in the order of recording. For each sample, the corresponding grid cell is determined. If the grid cell is empty and the value of the sample is not zero, the value of the sample is assigned to the grid cell. Otherwise, the sample is ignored. Our motivation for this approach rather than using the sum or mean of the values is the following line of thought: If the value is zero, either the harvester is currently not harvesting or the area has already been cleared. In both cases, the values do not reflect the original spatial distribution. We only consider the first value as when harvesting in a grid cell, the spatial distribution is disturbed, making later measurements incorrect.

While creating the grid, a new trace is generated that only contains samples contributing to the grid and replaces the original trace in the following evaluations. This trace contains some unnatural movement as there are jumps when the harvester crosses already explored areas. However, a more natural trace makes the evaluation less expressive as a lot of

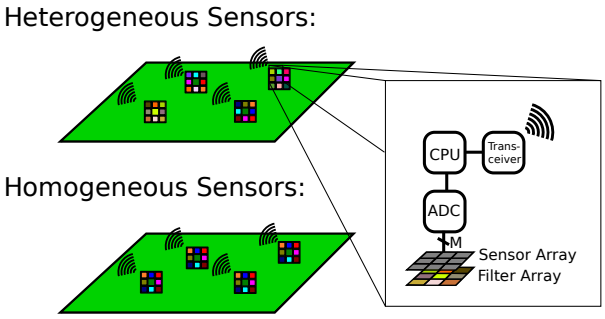


Figure 4.3.: Sensor node and network architecture of an M2HSN.

data can be transmitted during breaks and while crossing already explored areas. Thereby, the generated trace represents the worst case scenario as discussed in section 4.1.3.

When using the approach in reality, this preprocessing may be either performed on the harvester or in a remote evaluation center. The only prerequisite is the desired grid size which may be set based on the field shape calculated in previous years, e.g., using the algorithm from Lauer et al. (2014).

Figure 4.2 shows the locations of the original measurements and the resulting grid. The coordinates have been anonymized by removing an offset in order to preserve the privacy of the owner. Note that we have chosen the grid cell size as a trade-off being coarse enough to avoid gaps in the distribution and fine enough to not waste too many of the original samples. The displayed distribution is the one of the crop yield, i.e. the amount of crop that was collected per area in tons per hectare (t/ha).

4.2. Multi- to Hyperspectral Sensor Network (M2HSN)[13]

An M2HSN consists of multiple WSN sensor nodes which are equipped with multispectral sensors. The multispectral data of the whole WSN is collected to then estimate hyperspectral data from it. This estimation process can be based, e.g., on DCS, UPDM, or K-SVD. Before discussing the estimation process, we first describe the sensor nodes and the network.

4.2.1. Sensor Nodes

First, sensor nodes have to be prepared, e.g., as shown in figure 4.3 on the right. Each node has a multispectral sensor. Albeit not being the only possible architecture, in this thesis we assume the multispectral sensor to consist of M light sensors (e.g., photodiodes). The light sensors are connected to an ADC which supplies a Central Processing Unit (CPU) with digital readings of each sensor, that are then transmitted to a fusion center. Each light sensor is covered with a filter. The choice of the filters depends on the requirement for the hyperspectral resolution. From this requirement, one may deduce the number of bands N to be included in the output data generated by the fusion center. This equals the number of bands which would be acquired by conventional hyperspectral aerial or satellite-based remote sensing. For each sensor node, each of the M light sensors is covered with a filter picked from the N different filter types. Different processes for picking the filters are discussed in section 4.2.3.

In order to measure the reflectance, it makes sense to put the sensor in a slightly elevated position, such as on a post, pointing downwards. The angle of view of the light sensors should be sufficiently wide to ensure averaging over a sufficiently wide area. Neither should the angle be too wide as light from the sides may disturb the measurement. If the angle of view of the sensors is too narrow or to wide, a diffuser or an aperture may be added.

4.2.2. Sensor Network

An example network of $J = 4$ sensor nodes is shown in the upper left part of figure 4.3. In a real network, the number of sensor nodes J depends on the size of the area that shall be monitored, the required spatial resolution, and the required estimation quality of the hyperspectral data.

Each sensor node is prepared as described in the previous section 4.2.1. The sensor nodes can be placed in the given area using different patterns: They may be placed at areas of interest, on a grid, or randomly. The grid pattern allows for generating images similar to those of a satellite.

Each sensor node sends its readings of each light sensor to a fusion center at a frequency depending on the required temporal resolution. With the amount of data being lower in comparison to hyperspectral data, the transmission time is also lower, leading to reduced power consumption and, thereby, a prolonged lifetime of the sensor node. The frequency of measurements will usually be on a scale of hours, making synchronization challenges, bandwidth requirements, and the calculation time for the estimation of the high-resolution spectrum negligible. This allows us to describe the data acquisition as taking a snapshot of the whole area. Each

[13] The content of section 4.2 has been partially previously published by Hänel et al. (2019) except for section 4.2.3.

snapshot may be recorded independently from snapshots at other times. The data collection for one snapshot can be done using any communication pattern: it may happen in a multi- or single-hop pattern or even via the internet.

4.2.3. Choice of Filters[14]

A key aspect for this thesis is the choice of the filters in the sensor nodes. There are two fundamental approaches for choosing the filters: these may be chosen heterogeneously with varying filter sets at different positions or homogeneously with the same set of filters in every node. The two variants are shown in figure 4.3. The main benefit of the heterogeneous approach is that it is capable of gaining information in additional parts of a spectrum.

For the homogeneous approach, no customization of the hardware is required with respect to the location, the mass fabrication of such sensors will be more feasible. These are in fact similar to sensors already being cheaply available nowadays, such as the sensors integrated in the SparkFun Triad (see section 4.5.1). A major advantage for this homogeneous approach is the possibility of finding a particularly well-suited set of filters.

M2HSNs with these approaches are also called Heterogeneous M2HSNs (HeM2HSNs) and Homogeneous M2HSNs (HoM2HSNs) in this thesis.

4.2.4. Algorithms for the Spectral Reconstruction

For making the M2HSN work, the main task in processing is the estimation of the hyperspectral data. If the resources of the fusion center are insufficient, this estimation may also be outsourced to a high performance computing cluster. As the general idea of an M2HSN is new to the best of our knowledge, there are no existing algorithms to compare against. It seems unreasonable to develop a completely new algorithm instead of relying on previous research results in signal processing and remote sensing. Therefore we modify existing algorithms to be used for an M2HSN, yielding four different approaches for the estimation: (1) interpolation, (2) UPDM, (3) DCS, and (4) K-SVD. The interpolation is independent for each location and consists of a simple linear interpolation between the band readings. It is merely included as a trivial solution for comparison.

The dependencies of the algorithms differ as some of them exclusively work with a HeM2HSN or a HoM2HSN. Table 4.1 lists the compatibility. More information on the algorithms are given in the following sections 4.3 and 4.4.

[14]The content of section 4.2.3 has partially been previously published (Hänel et al. 2021b).

Table 4.1.: Different variants for realizing the M2HSN

	UPDM	UPDM-BBS	K-SVD	KSVD-BBS	DCS
HoM2HSN	✓	✓	✓	✓	✗
HeM2HSN	✓	✗	✓	✗	✓

4.3. Heterogeneous M2HSN (HeM2HSN)[15]

In the Heterogeneous M2HSN (HeM2HSN), the data processing based on DCS uses JSM-1. Ψ is generated using equation (2.20) with $\Psi_c = \Psi_1 = \Psi_2 = ... = \Psi_J = \Psi_D$ with Ψ_D being the difference matrix from equation (2.14). An explanation for using the difference matrix is given in section 6.3.2. The Φ_j constructing Φ according to equation (2.21) are directly deduced from the filters used in sensor node j according to $\Phi_j = \Phi'_j\Theta$. Each Φ'_j consists of the M rows of an $N \times N$ identity matrix corresponding to the M out of N filters used on the sensor node. The $N \times N_{Spec}$ sized Θ contains the transmission spectra of the filters. From here on we assume ideal narrow-band filters by setting Θ to the identity matrix and $\Phi_j = \Phi'_j$ respectively. y contains the concatenated band readings from all sensors. The combined Φ, Ψ, and y are used in equation (2.4):

$$f_{estimate} = \underset{f'}{\operatorname{argmin}} ||f'||_0 \text{ with } y \approx \Phi\Psi f' \tag{4.4}$$

We use the solver SL0 (see section 2.4) which offers a good trade-off between reconstruction quality and speed in our experience.

The result $f_{estimate}$ may be inserted into equation (2.2) to get the concatenation of all high-resolution spectra $x_{estimate}$:

$$x_{estimate} = \Psi f_{estimate} \tag{4.5}$$

4.4. Homogeneous M2HSN (HoM2HSN)[16]

In contrast to the HeM2HSN, the HoM2HSN does not gather any information on the bands not included in the band set. The data must instead be obtained from a different source. One may argue, that this is a problem of the approach, because obtaining hyperspectral data is not fully

[15]The content of section 4.3 has been previously published (Hänel et al. 2019).
[16]The content of section 4.4 has partially been previously published by Hänel et al. (2021b) except for subsection 4.4.2.

avoided. Therefore, we evaluate the feasibility of supplying this information by learning it from remote sensing data. The remote sensing data for training needs to be hyperspectral; however, no hyperspectral measurements are required on the ground. As the M2HSN is mainly intended as an addition to remote sensing, the less frequently obtained remote sensing data can be used as training data for the HoM2HSN. It is therefore relatively easy to obtain.

4.4.1. K Singular Value Decomposition (K-SVD)

One algorithm for learning the information is K-SVD (see section 2.3.9). K-SVD is used similar to the approach by Arad et al. (2016) explained in section 3.4.2: a high resolution dataset is used to train a transform base for compressed sensing. The resulting transform matrix consists of some artificial spectra that are well suited for sparse representations of the spectra in the training set. The resulting transform matrix is then used for the recovery of the high resolution spectrum in an ordinary compressed sensing approach. In this step, we use SL0.

4.4.2. Universal Pattern Decomposition Method (UPDM)[17]

UPDM is applied by measuring the base spectra in advance and then independently applying the algorithm to the multispectral data. It may be used for random bands as well, when the matrix of base spectra Y is known for each combination of filters. As each combination should ideally be unique, this will usually result in one matrix per node, i.e., replacing Y in equations (3.4) and (3.5) with Y_j.

4.4.3. Band Selection

As K-SVD, in contrast to DCS, does not require heterogeneous band selection, we used a simple selection of best bands. Unlike Arad et al. (2017), we did not use a genetic algorithm for two reasons: firstly, a genetic algorithm would introduce many adjustable parameters, making the solution too much dependent on finding good parameter sets for each dataset. For our scenario, we consider this an unjustified overhead considering the small increase in performance found by Arad et al. (2017). Secondly, we will show that it is more important to discard bad band sets than trying to find a near-optimal selection - this is also in line with the original idea of making random samples in Compressed Sensing. Discarding these is much easier: we simply try a few random band sets on a different dataset and se-

[17]The content of section 4.4.2 has been previously published (Hänel et al. 2019).

Table 4.2.: Multispectral sensors considered for the M2HSN.

Name	#Bands	Wavelengths[nm]	Price [€]
SparkFun Triad	18	410–940	≈70
Hammatsu Minispec.	e.g. ≈ 136	e.g. 320–1000	not listed
Ocean Insight Flame	e.g. ≈ 488	e.g.350–1000	e.g. ≈ 3000
Pixelteq Pixelsensor	8 of 16	425–850	250?
X-Rite i1Studio	36	380–730	400

lect one of the good ones. This approach can also be used for UPDM. The variants using the band selection are called *UPDM-BBS* and *DCS-BBS* in the evaluations.

4.5. Prototype for the M2HSN

A core goal of this thesis project was building an actual prototype for the M2HSN. This required several design decisions: A sensor node platform had to be chosen, containing power supply, processing, and communication capabilities. But the most important choice are the sensors. These can be split into two main categories: The first category are ready-to-use multispectral sensors which offer cheap solutions and less development effort. However, these are not suitable for the HeM2HSN, because the wavelengths cannot be chosen arbitrarily. The second category of sensors are simple light sensors, that need to be equipped with filters.

4.5.1. Sensors

Several multispectral sensors were found during research. These are compiled in table 4.2. While the spectrometer series by Hammatsu and Ocean Insight offer compelling wavelength ranges and spectral resolution, these have not been further considered mostly because the prices exceed what is acceptable for WSN nodes. The Pixelteq Pixelsensor is particularly intriguing: because of its configurability, it seems usable even for the HeM2HSN.

However, even if considering all available filters, there are still less bands available than with the SparkFun Triad. As the SparkFun Triad also offers a slightly wider wavelength range and a lower price, it was chosen as one of the sensors used in the prototype.

For the evaluation of the HeM2HSN, none of the ready-to-use multispectral sensors seems suitable. Therefore, a self-built multispectral sensor was developed. It consists of multiple light sensors equipped with exchangeable filter. Note, that the design is similar to the cheaper ones of the

Table 4.3.: Lightsensors considered for the prototype.

Name	WL[nm]	Measurement range	Output	TC	Dyn. range
OPT3001	440–670	$10mlx - 83klx$	Digital	✓	$8.3 \cdot 10^6$
VEML7700 ALS	470–640	$7mlx - 120klx$	Digital	✓	$1.7 \cdot 10^7$
VEML7700 W	400–1000	unspecified	Digital	✓	unspecified
VEML6035 ALS	460–690	$4mlx - 6.7klx$	Digital	✓	$1.7 \cdot 10^6$
VEML6035 W	400–1000	unspecified	Digital	✓	unspecified
OPT3002	300–1000	$(1.2nW - 10mW)/cm^2$	Digital	✓	$8.3 \cdot 10^6$
OPT101 (+ ADS115)	400–1100	depends on circuit	Analog		$1.6 \cdot 10^6$
TSL237	300–1050	$(43pW - 435\mu W)/cm^2$	Freq.	✓	$5.0 \cdot 10^6$
S1087-01 (+ TelosB)	320–1100	$(1.5\mu W - 6mW)/cm^2$	Analog		$4.1 \cdot 10^3$

ready-to-use multispectral sensors. The price of the filters made this option more expensive than the ready-to-use multispectral sensor but it is still cheaper than the more expensive ones of the ready-to-use multispectral sensors. It is used for the evaluation of the HeM2HSN. However, the vision of an integrated sensor like the existing ones remains: miniaturized and at a lower price tag. This second design should only be seen as an interim step for validating the HeM2HSN approach.

For this second design option, a suitable light sensor had to be found. Multiple such sensors were considered, those are listed in table 4.3. The OPT3001, the S1087-01, and the TSL237 are included because we encountered them in popular measurement platforms, namely the Sensortag CC2650STK by Texas Instruments, which we used in (Hänel et al. 2021a), the Telos B sensor node by Crossbow and UC Berkeley, which was presented by Polastre et al. (2005) and widely used in WSN research, and the Sky Quality Meter by Unihedron, which was researched, e.g., by Cinzano (2005) and is widely used by astronomers. The OPT3001 is representative of a wide variety of sensors, which are not usable because of the limitation to wavelengths of the human eye. These are typically called Ambient Light Sensors (ALSs).

Besides the wavelength range, the measurement range played a crucial role for the selection of the sensors. The band filters only let a fraction of the light pass through, leading to very low levels of power. The narrow viewing angle required for the filters further reduces the power. This holds true especially in the infrared region, because in this region, the sensitivity of the sensors is low and the infrared intensity in the sunlight is low. However, it is desirable to use the same sensor not only for the reflected light but also directly for the incoming sunlight without requiring a neutral density filter. In this case, the power level is very high. Furthermore, the illumination varies significantly in the course of a day. A wider measurement range makes a longer part of the day usable.

This made the OPT3002 a perfect choice for the prototype with its dynamic range of $8.3 \cdot 10^6 : 1$. It adjusts the gain setting automatically which also makes it easy to use.

While the TSL237 comes close, its sensitivity is shifted overall to lower power levels which lead to its exclusion. The VEML7700 is an interesting alternative for future evaluations because of its even higher dynamic range and the combination of two sensors in a single packet: one with visible light response function (here called VEML7700 ALS) and one with a wider response function (here called VEML7700 W). A sensor named VEML6030 offers mostly identical specification but a different form factor. Unfortunately, detailed specifications are only offered for the visible light sensor. Measurements indicated, that resolution at low light is better with the OPT3002 in a side-by-side comparison making it better suited for usage behind narrowband filters. The VEML6035 offers similar specifications to the VEML7700, albeit shifted to lower light levels, but it was released too late to be considered for the first prototype. The OPT101 may be adjustable to an even wider range with an ADC that offers higher resolution and range of gains, however this was not pursued as it would require significantly more development effort.

4.5.2. Filters

For the configuration of sensor nodes actually deployed, some additional considerations played a role. While the results of the scalability evaluation discussed in section 7.2.4 lead to a suggestion of 40 sensor nodes, for the prototypical experimental evaluations we limited the number to five sensor nodes with exchangeable filters due to funding constraints. Furthermore, we wanted to make sure, that we can use the measurement results for processing via both DCS and K-SVD. To generate filter sets suitable for both approaches, we optimized the filter sets for K-SVD as described in section 4.4. We then simulated using the J best filter sets for a HeM2HSN. The results are shown in figure 4.4 for $J = 5$ for $J = 64$ as *DCS-KSVDCHANNELSETS*. $J = 64$ was chosen because it is significantly higher than necessary so it approximates the best achievable result.

The K-SVD results with the same band sets (*KSVD-BBS*) is shown for comparison as well as the DCS result for completely random band sets (*DCS-RANDOM*). For $J = 5$, the selection of the band sets even helps to improve the reconstruction quality, making these filter sets a surprisingly good compromise. Although, DCS cannot compete with K-SVD in this case. For $J = 64$, the limitation to the optimized band sets shows its impact on the extraction of the common signal. The result is slightly worse in this case. However, the result is still better than the K-SVD result in this case.

The datasets used in this simulation were *Air_MA* for training K-SVD, *Air_MB* for the band set evaluation and *Ground_Full* for the evaluation.

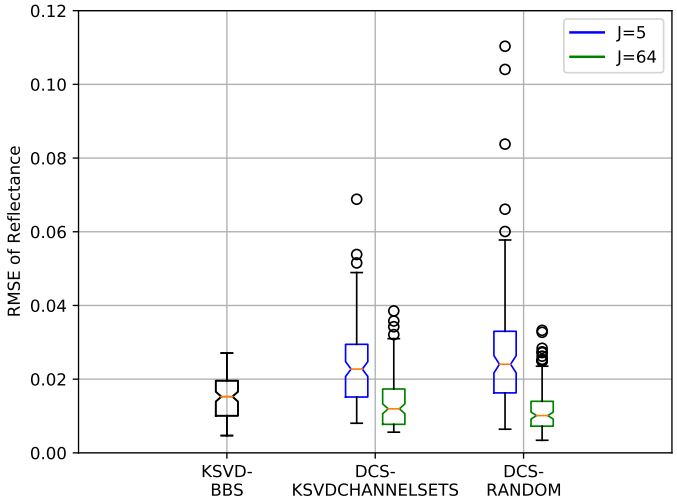


Figure 4.4.: Results for K-SVD band sets in the HeM2HSN.

Table 4.4.: Filter sets selected for the prototypes. Wavelengths in nm.

Set 1	Sim.	579.9	690.5	732.5	746.6	803.4	959.8	1031.7
	Real	580	690	730	750	800	960	1040
Set 2	Sim.	566.0	593.8	649.3	760.7	846.0	9959.8	1050.7
	Real	570	590	650	770	850	960	1050
Set 3	Sim.	552.1	579.9	649.3	746.6	846.0	959.8	1031.7
	Real	550	580	650	750	850	960	1020
Set 4	Sim.	538.5	552.1	649.3	746.6	817.6	974.0	1050.7
	Real	540	550	650	750	820	970	1050
Set 5	Sim.	498.4	538.5	593.8	760.7	846.0	974.0	1069.6
	Real	500	540	590	770	850	970	1064

For more information on these datasets, we refer to section 6.4.1. The number of atoms is 8 as it turned out superior in the evaluation. The number of bands is $M = 7$ with an 8th sensor being used as sky sensor.

The resulting filter sets are listed in table 4.4. Note, that the wavelengths of the filters actually built into the prototypes slightly vary as listed due to availability.

4.5.3. System

As the mainboard for the system, a LILLYGO TTGO T-Beam V1.1 was chosen, that combines LoRa, a GPS, an ESP-32, and a rechargeable battery (type 18650) on a single board. The ESP-32 offers sufficient GPIO ports, as well as deep sleep support and additional wireless technologies, namely Bluetooth Low Energy (BLE) and Wireless Local Area Network (WLAN). Note, that previous versions offer less power management functions making them less suitable for long-term deployments. All components were connected with custom designed Printed Circuit Boards (PCBs) that help directing the light sensors in the correct directions. Cases for rudimentary protection from stray light and weather conditions were produced via Fused Deposition Modeling (FDM) from Polylactic Acid (PLA) and PolyEthylene Terephthalate Glycol (PETG). The case contains light tubes for restricting the field of view of the light sensors directed to the ground. The resulting field of view is $(10.5 \pm 2)^\circ$ for the nodes with configurable bands and $(14.5 \pm 2)^\circ$ for the nodes with fixed bands. The choice of the field of view was constrained by factors such as the size of the filters, the angle-dependency of the filters, and the distance of the light sensors on the PCB. However, the core trade-off consists of choosing a sufficiently wide angle for averaging across a sufficient area and having the sensors sufficiently illuminated and a sufficiently narrow angle for blocking stray light. In the case of the nodes with configurable bands, a tray containing one bandpass filter per band is placed in the path of the light tubes. For the sky sensor, instead a diffuser was chosen that has been found to have consistent transmission properties across a wide range of wavelengths (Bauer et al. 2016).

Figure 4.5 shows impressions of the sensor nodes. The total costs of the two sensor node types are listed in table 4.5. Especially the node design with fixed wavelengths already qualifies as a low-cost solution. When producing larger numbers of the sensor nodes, joining everything on a single PCB will allow reducing weight, size, assembly time, and costs even further.

4.5.4. Communication

Communication in the WSN takes place in two steps, the first is via LoRa to a border router, the second is via Public Land Mobile Network (PLMN). The sufficient range of LoRa, the redundancy of consecutive measurements, and the uni-directional nature of the data transfer allowed us to choose a very simplistic and efficient communication pattern that lost some popularity with the birth of the Internet of Things (IoT): The sensors sleep most of the time. They wake up periodically for taking measurements and sending data without waiting for any incoming packets. In comparison to

Figure 4.5.: Photos of top and bottom of the connection PCB, the filter tray, and a fully assembled sensor node.

Table 4.5.: Resulting material costs of one sensor node in Euro.

Common	Connector PCB	1	2.91
	Main Board	1	34.93
	Battery	1	8.14
	Bubble Level	1	1.49
	M3 rods	4	2.15
	Filament	76g	1.52
	OPT 3002	1	7.37
	other (glue, solder, etc.)		< 5.00
	Sub-total		63.51
Fixed	Triad Spectro	1	72.44
	Filament	84g	1.68
	Total		137.63
Configurable	I2C Multiplexer	1	2.00
	OPT 3002	7	51.57
	Filters	7	396.20
	Filament	138g	2.76
	Total		516.04

RACS by Fazel et al. (2011), the transmission times are not chosen explicitly at random, because the probability of collisions was sufficiently low and no CS for the time domain was planned.

The benefits of this approach is that the sleep time can be maximized because no additional wake time is required for forwarding packets, receiving acknowledgments, or synchronization of the clocks. Due to its simplicity, the approach is also very robust against software errors and failure states. However, it is not robust against packet loss.

Measurements are taken once every five minutes containing the node's identifier, the temperature, the voltage of the battery, the brightness values from the spectrometers, and the brightness from the sky sensor. Every 6 hours, instead of a normal measurement, a Global Navigation Satellite System (GNSS) fix is acquired - this simplifies the deployment process because no manual GNSS measurements are required. As a side-effect, the time to fix is highly variable, which helps to further avoid LoRa collisions.

For LoRa, a coding rate of 4/5, bandwidth of 250 kHz, center frequency of 868 MHz, and a spreading factor of 7 are used. This is sufficient for small areas and reduces battery consumption with short transmission times of approximately 150 ms for packets of 200 bytes. No packet exceeded this length. Assuming this worst case transmission time, the probability for a collision of a packet in a network of ten nodes is $1-(\frac{300\,\mathrm{s}-2\cdot 0.15\,\mathrm{s}}{300\,\mathrm{s}})^9 \approx 0.009$, a loss of this magnitude is negligible in comparison to other packet loss causes.

4.5.5. The test network

A prototypical test network was built near Osnabrück on a winter wheat field. The last weeks before the harvest were monitored from June 15th, 2021 till August 3rd, 2021. For comparison, in-situ measurements of the spectra at high resolution with a spectroradiometer of type HR-1024i by Spectra Vista Corporation (SVC), and multispectral images by an Unmanned Aerial Vehicle (UAV) of type DJI Phantom 4 Multispectral (P4) were taken. Figure 4.6 shows sensor node positions on an NDVI map generated from a UAV image taken on June 18th, 2021. A total of 10 sensor nodes were placed, among them 5 of the 7-band model and 5 of the 18-band model. The positions were chosen such that they are evenly distributed on the field for both sensor types. At the center position, one of each type was used.

4.5.6. Data Processing

The sensor readings x_{raw} are processed by first dividing the readings for the individual bands by the reading of the sky sensor x_{sky}:

$$x_{amc} = x_{raw}/x_{sky} \tag{4.6}$$

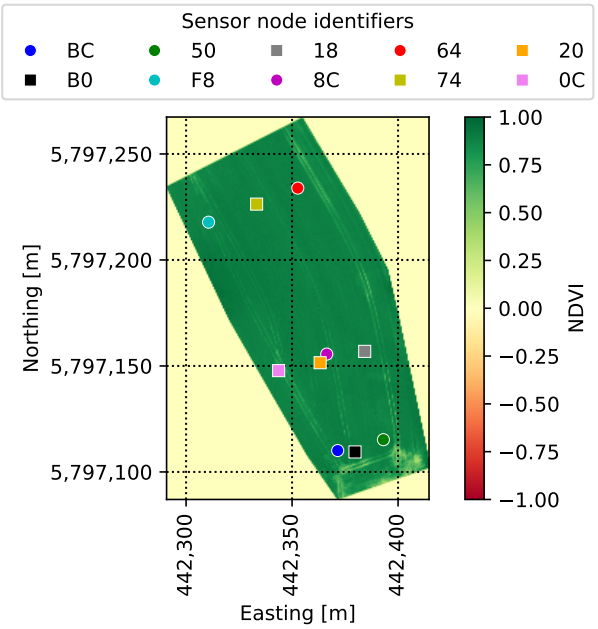


Figure 4.6.: Positions of the sensor nodes on an NDVI map. Circles indicate 18-band sensors and squares indicate 7-band sensors. The Coordinates are in UTM zone 32U.

On June 18th, we held white reference surfaces below the sensors and took multiple measurements. The resulting values $x_{amc,whiteref}$ are used for conversion into reflectance values:

$$x_{refl} = x_{amc}/x_{amc,whiteref} \tag{4.7}$$

A dark current compensation did not turn out to be necessary because the sensors already read 0 in darkness. Unless otherwise mentioned, we calculated the median spectrum of a day over all measurements taken between 10am UTC and 2pm UTC. Solar noon was approximately at 11:30am. The limitation to noon reduces the effect of shadows on the reflection values.

4.6. Conclusion

In this chapter, the concepts of CFE and M2HSN were presented. For the M2HSN, two different approaches were differentiated, the HoM2HSN and the HeM2HSN and two CS based approaches for processing the data were introduced. Furthermore, the design of an actual prototypical M2HSN was

described. Before parametrizing the approaches for optimal performance in chapter 6, chapter 5 gives an explanation on how the quality of the reconstructions may be estimated.

5. Adding Quality Estimates to Compressed Sensing

A major issue of CS based systems is that the data generated by them is only an approximation due to the lossy decompression for realistic data. This issue can be drastically reduced when the approximation error is known. Therefore, this chapter lists approaches for estimating the error. In section 5.1 metrics for quantifying the approximation error are introduced. These are used for most of the evaluations in chapter 7. Thereafter, in section 5.2 a concept for developing metrics that estimate the approximation error based on information available during decompression is presented, culminating in a list of such metrics in section 5.3. Lastly, in preparation for the evaluation of these metrics, section 5.4 discusses metrics for judging the success of the estimation.

5.1. Error Metrics

In order to make an estimate on the reconstruction quality, we first need a way to measure the reconstruction quality. This is easy whenever the uncompressed data is known besides the decompressed or reconstructed data. In the simulations we have this data available, but in real-world applications it is not available because it never gets measured. Only the compressed data and the reconstructed data are known then, not the uncompressed data. The error metrics we use are well-established in literature. The Root Mean Square Error (RMSE) measures the difference between two vectors:

$$RMSE = \sqrt{\sum_{n=1}^{N} (x_{n,est} - x_n)^2 / N} \tag{5.1}$$

A nice property of the RMSE is that it has the same unit as the original values making it intuitively interpretable.

One drawback of the RMSE is its inability to reflect an error of the same magnitude usually being worse when the values in x are generally lower.

Therefore, usually the Signal-to-Noise Ratio (SNR) or the Noise-to-Signal Ratio (NSR) is used. Here we choose the NSR to keep consistency with the RMSE in the way that larger values are worse than smaller values:

$$NSR = \frac{RMSE^2}{x^2/N} = \frac{\sum\limits_n (x_n - x_{n,est})^2}{\sum\limits_n (x_n)^2} \tag{5.2}$$

Instead of measuring the quality of the whole vector, in some cases, the absolute error of individual elements in the vector needs to be assessed instead:

$$\epsilon_{n,abserr} = |x_n - x_{n,est}| \tag{5.3}$$

5.2. The Quality Metrics

Estimating the reconstruction quality in Compressed Sensing is challenging because far less data is collected than traditionally assumed necessary. For the reconstruction quality, even more information needs to be squeezed from this tiny amount of data. One way of gaining such information is the incorporation of the calculation in the solver (Ji et al. 2008). However, the major drawback of this is that many solvers have been developed, some usable for all kinds of data, some specialized on specific kinds of data. This development is ongoing, and quality estimation would have to be incorporated in every single one of these solvers which may not always be possible. A generic approach that uses information independent of the solver seems more tempting. In the quest for finding suitable metrics we use two strategies: In the first strategy we use that the prerequisites for Compressed Sensing are known. We cannot check how well the prerequisites have been fulfilled by the original data because we do not know the original. However, we have an estimate of the original data and can check if the estimate fulfills the prerequisites. The two core requirements are that the data needs to be sparse and that the measurement needs to be suitable for the transformation, and hence, the data. In the second strategy for finding such metrics, we research solvers. These often use iterative approaches which also rely on metrics for finding out whether a sufficient solution has been found. These metrics may in some cases be usable independent of the solvers.

Two main types of quality metrics will be considered in this thesis: on the one hand there are metrics which assign a quality value to each element of a reconstructed vector, from here on called In-Signal Metric (ISM), on the other hand there are metrics which assign a quality value to a whole vector, from here on called Per-Signal Metric (PSM).

Using a suitable operation, such as median, a PSM may be derived from an ISM. ISMs mainly help experts assessing the data and which parts to

rely on or ignore. It may also be used in algorithms that rely on the reconstructed data for improved decision-making. The PSMs have the potential to become useful for even more purposes, if being well-calibrated: They may be used to find a better reconstruction if being used to compare the results from different solvers or different solver parameter sets. Similarly, the PSMs can be used to select the best transform when it is not known. When it is possible to acquire more samples iteratively, PSMs may be used to keep sampling until sufficient data has been collected (Chou et al. 2009).

5.3. The List of Quality Metrics

These considerations let us build a list of metrics given in tables 5.1 and 5.2. Recalling our first strategy, one core requirement for CS is sparsity in the data. This is expressed by the ℓ_0 norm. As we are dealing with real-world, noisy data, this metric is inconvenient, a threshold is needed for differentiating between zero and non-zero values. In the CS community, it is well known that ℓ_p norms with $p \leq 1$ are similarly well suited as a measure of sparsity. Therefore, we also consider some metrics in this range, namely $\ell_{0.25}$, $\ell_{0.5}$, and ℓ_1. The second core requirement for CS is incoherence between the measurement matrix and the transform matrix. According to Candes et al. (2008), this is expressed as:

$$\mu(\Phi, \Psi) = \sqrt{N} \max_{1 \leq k,j \leq N} |\langle \phi_k | \psi_j \rangle| \tag{5.4}$$

While this is usually defined for the whole base, in an actual measurement, only a subset of M vectors out of the ϕ_k is actually applied. Using only these for the calculation of μ may lead to a decreased value. However, we quickly rejected this as a quality metric because it only differs significantly from the actual μ in extremely rare cases. Furthermore, the requirement of incoherence is somewhat outdated (Candes et al. 2011). The third and most trivial core requirement for CS is that M is sufficiently high. Therefore, we include the compression ratio N/M as another quality metric, albeit it is only usable in some scenarios, often N and M will be constant for all results that need to be compared.

Recalling our second strategy for finding results, we take a look at metrics used in solvers. Besides minimizing the ℓ_p norm with $0 \leq p \leq 1$ for f, the solution must closely match the measurements. In LASSO, this is measured with the ℓ_2 norm for the difference of the reconstruction and the measurements. We call this RMSME from here on. In some solvers, this metric is not usable, as they only generate solutions that fit the measurements perfectly. Another observation for the solvers is, that they sometimes generate solutions which are obviously wrong for human observers, because they are in the wrong value range. With the ℓ_p metrics,

Table 5.1.: List of Per-Signal Metrics (PSMs)

Name	Description	Requirements
l0norm_psm	ℓ_0 norm of f, requires threshold parameter	
l025norm_psm	$\ell_{0.25}$ norm of f	
l05norm_psm	$\ell_{0.5}$ norm of f	
l1norm_psm	ℓ_1 norm of f	
l2norm_psm	ℓ_2 norm of f	
l1overl2_psm	ℓ_1/ℓ_2 for f	
l05normnd_psm	$\ell_{0.5}$ norm of x	
l1normnd_psm	ℓ_1 norm of x	
l2normnd_psm	ℓ_2 norm of x	
rmsme_psm	Root Mean Square Measurement Error (RMSME)	Suitable solver (e.g. LASSO)
biggestgap_psm	Biggest gap in vector between 2 consecutive samples	Suitable measurement (spike)
gapdiff_psm	Difference between the biggest gap and the smallest gap	Suitable measurement (spike)
emptybinsfraction_psm	Number of filled bin, when splitting vector in M bins	Suitable measurement (spike)
compressionratio_psm	The compression ratio N/M	Variation of M
coherence_psm	Coherence calculated between measurement and transform	
diff2common_psm	RMSE between common signal and individual signal	DCS-JSM-1
angulardiff2common_psm	Angle between common signal and individual signal	DCS-JSM-1
variance_psm	mean of variance_ism	Suitable solver (BCS)
crossval1o_psm	RMSME from leave-one-out cross-validation	

we already have metrics that may help to identify these cases. However, as they are chosen to promote sparsity, which is not relevant for this consideration, we also include ℓ_2 for the sparse space reconstruction f and several ℓ_p metrics for the normal space reconstruction x. The solver BCS plays a special role, because it has the covariance matrix Σ as an additional output. A quality metric can be derived by extracting the diagonal elements of the transformed covariance matrix (Ji et al. 2008):

$$\gamma_{variance,ISM} = \text{diagonal}\left(\Psi \cdot \Sigma \cdot \Psi^T\right) \tag{5.5}$$

Table 5.2.: List of In-Signal Metrics (ISMs)

Name	Description	Requirements
samdist_ism	Distance to closest sample	Suitable measurement (spike)
samdist2d_ism	Distance to closest sample in two-dimensional array	Suitable measurement (spike), suitable data (two-dimensional)
diff2common_ism	Difference per element	DCS-JSM-1
variance_ism	Variance as determined by BCS	Suitable solver (BCS)

This is a vector with the ISM version of the metric. The PSM version is the mean of the ISM version.

Besides these strategies, we collected some more metrics: Especially in the case of simple sparse measurement matrices, intuitively, solutions are expected to be better close to the samples. This leads us to the distance to the closest sample, which is a discrete value as it specifies the number of vector elements to the closest one. This metric is an ISM. For the derivation of a PSM, we do not simply use the median or average as it is closely related to the compression ratio. Instead, we concentrate on the heterogeneity of the sample distribution. We try the following metrics for this purpose: the size of the biggest gap, the difference between the size of the biggest and the smallest gap. As a third metric for this purpose, we cut the vector in M equally sized bins and count the bins that contain no sample. We then calculate the fraction of empty bins.

We also consider JSM-1 of DCS. As it models multiple individual signals as the sum of a common signal and an innovative term specific for each individual, we compare each individual signal with the common signal. The vector of absolute differences for each element of the two signal vectors is the ISM version. The length of this vector and the angle between the two signal vectors are two PSM versions.

Lastly, we also use a cross-validation metric similar to the one introduced by Ward (2009): From the measurement vector y, we derive M vectors y_m of length $M-1$ by leaving each sample out once. The left out sample is $p_m = y[m]$.

We build the corresponding Ψ_m matrices, and calculate a CS solution in sparse space $f_{est,m}$ for each of these vectors. For this, we calculate a measurement result with $y_{est,m} = \Phi \cdot \Psi \cdot f_{est,m}$. From this vector we select the element corresponding to the previously left out sample: $p_{est,m} =$

$y_{est,m}[m]$. Now, we can calculate a cross-validation metric as follows:

$$\gamma_{crossvall1o,PSM} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(p_{est,m} - p_m)^2} \tag{5.6}$$

Note that the calculation overhead is significantly higher than for the other metrics, because $M + 1$ CS problems need to be solved instead of just a single one.

5.4. Evaluation Metrics

For assessing the suitability of the quality metrics, we require evaluation metrics to measure how well a quality metric correlates with the actual quality of the reconstruction, measured by one of the error metrics. A common measure for the correlation of two metrics is the coefficient of determination R^2. An optimal quality metric would have an R^2 of one. Achieving this is highly unlikely, but metrics with a lower R^2 will often be helpful as well. One issue of R^2 in the context of quality metrics is that it strongly penalizes non-linear dependencies. However, for many applications it suffices to make a correct decision based on the comparison of two reconstructions. Therefore, monotonicity in the dependency would be perfectly sufficient in these cases. For rating this, we introduce a new metric called Good Pairs (GP), that is related to the inversion number, used for rating how well an array is sorted. For the calculation of GP, every possible pair of the J values in the case of PSMs or the N values in the case of ISMs is evaluated. There are 9 possible relations between the pairs:

$$R_1 = \{\epsilon_i < \epsilon_j \text{ and } \gamma_i < \gamma_j\} \tag{5.7a}$$
$$R_2 = \{\epsilon_i < \epsilon_j \text{ and } \gamma_i = \gamma_j\} \tag{5.7b}$$
$$R_3 = \{\epsilon_i < \epsilon_j \text{ and } \gamma_i > \gamma_j\} \tag{5.7c}$$
$$R_4 = \{\epsilon_i = \epsilon_j \text{ and } \gamma_i < \gamma_j\} \tag{5.7d}$$
$$R_5 = \{\epsilon_i = \epsilon_j \text{ and } \gamma_i = \gamma_j\} \tag{5.7e}$$
$$R_6 = \{\epsilon_i = \epsilon_j \text{ and } \gamma_i > \gamma_j\} \tag{5.7f}$$
$$R_7 = \{\epsilon_i > \epsilon_j \text{ and } \gamma_i < \gamma_j\} \tag{5.7g}$$
$$R_8 = \{\epsilon_i > \epsilon_j \text{ and } \gamma_i = \gamma_j\} \tag{5.7h}$$
$$R_9 = \{\epsilon_i > \epsilon_j \text{ and } \gamma_i > \gamma_j\} \tag{5.7i}$$

Note, that relations (5.7d)-(5.7f) occur with rather low probabilities ($P(R_4)$, $P(R_5)$, $P(R_6) \ll 1$), because the error metrics are rarely exactly identical. A similar statement can be made for relations (5.7b) and (5.7h). However, some of the quality metrics tend to assume few discrete values

making the relations significantly more frequent in these cases. For continuous metrics $P(R_2), P(R_8) \ll 1$ is expected. For random continuous metrics, independent of the actual, continuous error, the combinations of relations have equal probabilities $P(R_1) = P(R_3) = P(R_7) = P(R_9)$ and with $\sum_{i=1}^{9} P(R_i) = 1$ and $P(R_2), P(R_4), P(R_5), P(R_6), P(R_8) \ll 1$), it must be $P(R_1) = P(R_3) = P(R_7) = P(R_9) = 0.25 - \epsilon$ with $0 \leq \epsilon \ll 1$. The metric GP_+ contains the number of pairs fulfilling relation (5.7a), (5.7e), or (5.7i) divided by the total number of pairs. For random continuous metrics, a value of $GP_+ \approx 0.5$ is expected. Significantly higher values indicate a suitable metric. In some cases, the relation is inverted: a low value of the quality metric indicates a high error and a high value indicates a low error. Therefore, a second metric, GP_-, is introduced, which instead considers relations (5.7c), (5.7e), and (5.7g) as good. For continuous metrics one can assume $GP_+ + GP_- \approx 1$ whereas for discrete metrics we typically get $GP_+ + GP_- < 1$ because equal values occur regularly. Note that GP_- can also be interpreted as GP_+ for the metric $-\gamma$, making all the following discussions for GP_+ applicable to GP_- as well.

A remaining question is how much higher than 0.5 a GP_+ value must be for it to be significant. For continuous metrics, the significance of a GP_+ value can be determined similarly to the p value for R^2. For a given GP_+ value, the GP_+ p value is designed to indicate the probability of a random metric generating a higher GP_+ value. The following calculation only makes an approximation in the sense that it assumes each value to be unique. For the calculation of the GP_+ p value, we consider a vector of quality metrics γ. Among all permutations of this vector, we count the percentage of the permutations that have less inversions than the given GP_+ value.

The number of permutations for a given number of inversions k in a vector of size n is given by the Mahonian number $M_{n,k}$ which is defined recursively (MacMahon 1913):

$$M_{n,k} = \begin{cases} 0 & \text{if } k < 0 \\ 0 & \text{if } k > \frac{(n^2-n)}{2} \\ 1 & \text{if } k = 0 \\ M_{n-1,k} + M_{n,k-1} - M_{n-1,k-n} & \textit{otherwise} \end{cases} \tag{5.8}$$

The probability for k inversions in γ is $p(k) = M_{n,k}/n!$. Which allows us to determine the probability that a random metric generates not more than $K = (1 - GP_+) \cdot (n^2 - n)/2$ inversions:

$$p(k \leq K) = \sum_{k=0}^{K} M_{n,k}/n! \tag{5.9}$$

If this is sufficiently low, the value for GP_+ can be considered a relevant result. For discrete metrics (and non-unique continuous values), the GP_+

tends to lower values because equal values for γ are treated as inversions as well (see relations (5.7b) and (5.7h)). This makes the $p(k \leq K)$ from equation (5.9) an upper limit in these cases. The relevance is still given, values with a higher $p(k \leq K)$ might be relevant as well. We refrain from finding an adjusted value because the discretization is highly metric-dependent and finding an upper limit seems sufficient.

5.5. Conclusion

In this chapter the error metrics used in the upcoming parametrization and evaluation chapters 6 & 7 have been formally introduced. The challenge of estimating this error based on the information was pointed out and approaches for the estimation were listed. Metrics for grading these approaches have been introduced that will be used in sections 7.5-7.7 to determine the quality of the approaches.

6. Parametrization and Configuration

The CS based solutions CFE, HeM2HSN, and HoM2HSN developed in chapter 4 need to be parametrized and configured as laid out by the recipe in chapter 2. Before getting to these specific solutions, the steps of selecting and parameterizing CS solvers are first discussed for a more general example in section 6.1, greatly simplifying this step for the following parametrizations and configurations of CFE, HeM2HSN, and HoM2HSN in sections 6.2-6.4.

6.1. Solver parametrization and why it does (not) matter

In an attempt to find good parameters for each of the solvers, we use a simple experiment. We generate an example signal with $N = 256$ samples with a sample rate of 1 Hz. The signal is a sine wave with a frequency of 0.2 Hz and an amplitude of 2 with an additive Gaussian noise with a standard deviation of 0.1. Out of the $N = 256$ samples, $M = 32$ samples are chosen randomly as the measurement vector y. As pointed out in section 2.2, this means assuming a sparse measurement matrix consisting of M rows of an identity matrix. As transform matrix, a DCT matrix is assumed. We refrain from varying the signal or the measurement of it to keep calculation time handleable. With each of the solvers, we solve this CS problem and vary the parameters of the solver. More details on the parameters and solvers were given in section 2.4. In the parameter variation, we perform a global optimization using a brute force approach on a grid. The goal is finding parameter sets that minimize the RMSE between the reconstructed signal and the original signal.

ℓ_1 magic

The parameter variation result for ℓ_1 magic is shown in figure 6.1. For all parameters, the common logarithm is shown. Within each subplot, μ and the tolerance of the Newton search are varied. Between the plots, the number of iterations is increased from left to right and ϵ is increased from top to bottom. Cyan color indicates errors which exceed the scale. Red color indicates that the calculation has been aborted because it took more

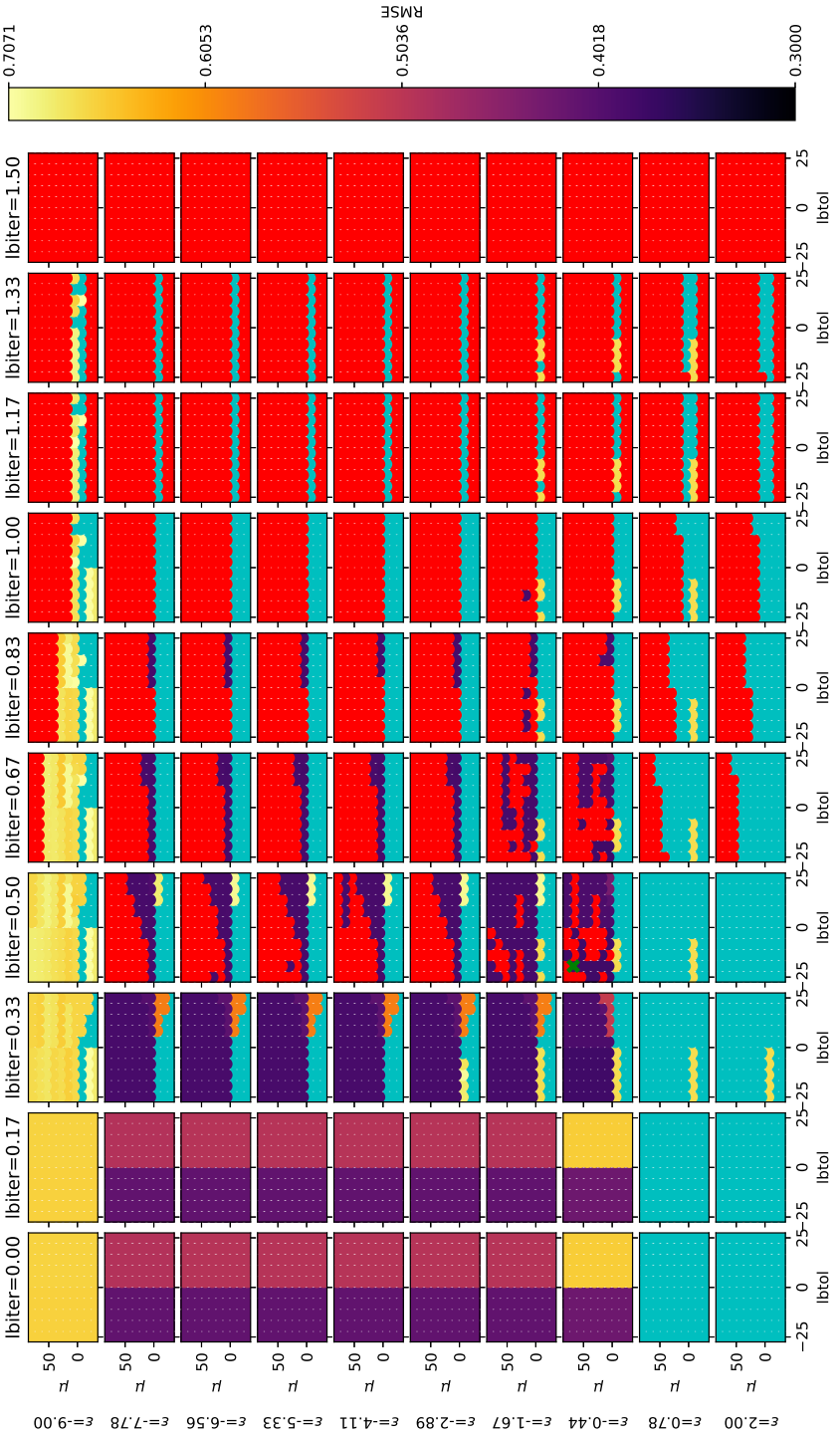


Figure 6.1.: Results for parameter variation of ℓ_1 magic.

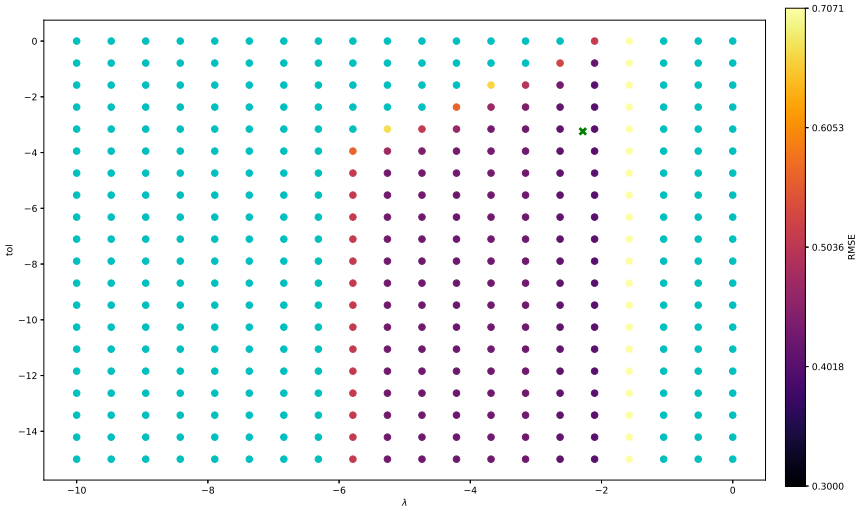


Figure 6.2.: Results for parameter variation of LASSO.

than 10 seconds. In the first and the last rows, errors are always high, showing that the epsilon has to be chosen appropriately, although the sensitivity is very low with valid values stretching over 8 magnitudes. With the lower bound being arbitrary, the upper boundary is more interesting. It is close to the actually achievable error. As a rule of thumb, it should be chosen slightly below the expected achievable error. The number of iterations must not be too low as indicated by the relatively high errors in the first two columns. A higher number of iterations mainly increases calculation time, leading to many aborted calculations. Further improvements in dependence of the number of iterations do not seem significant. The μ must be sufficiently high to actually increase the barrier, i.e. above 1. However, the exact value seems to be rather unimportant. The tolerance of the Newton search again has a rather low influence. Only a high value, especially in combination with a low μ, low number of iterations and high ϵ leads to high errors. Similar to the ϵ, the threshold is close to the actually achievable error.

LASSO

For LASSO, only λ and the tolerance (tol) were varied. The maximum number of iterations was kept constant. The result is shown in figure 6.2. The axes show the common logarithm of the parameters. The color shows the RMSE. The figure shows that the RMSE has a clear minimum region in λ direction. It must neither be chosen too low, nor too high. Decreasing the tolerance widens the minimum region, making the λ selection more

forgiving. Overall, this shows that the λ has to be chosen relatively well, whereas the other parameters merely determine how much calculation time one is willing to spend.

SL0

Finding appropriate parameters for SL0 is particularly hard because there are four parameters which have to be considered and actually have an influence on the performance. Figure 6.3 shows the result of the parameter scan. The colors indicate the RMSE of the reconstruction. Within a single figure, σ_{min} and α have been varied, the parameters of the ℓ_0 approximation. Note, that α is chosen linearly along the allowed range with $\alpha \in]0, 1[$ while σ_{min} decreases exponentially from 1 closer to 0. Between the individual figures, the gradient descent has been varied with the number of variations L increasing exponentially from 1 to 100 and the step size factor being varied linearly: $\mu_0 \in [0.1, 4.0]$. For σ_{min} and for L, the common logarithm is shown instead of the actual value. Note, that the calculation time increases within a subplot from top left to bottom right and across the subplots from top to bottom.

First of all, comparing the subplots, there is an area of subplots containing low RMSE values around the diagonal from the upper right to the bottom left of the figure. This shows that the gradient descent has to be properly parametrized. Either many small steps or few large steps have to be made. Setting both high as in the lower right corner or both low as in the upper left corner results in bad performance. Generally, it is better to do many steps instead of large steps. In the figure, this is indicated by a widening area of low RMSE values when moving along the diagonal from the upper right to the lower left.

Within most of the individual subplots, a strikingly similar pattern may be observed: the lowest RMSE values are again observed along the diagonal from the upper right to the lower left. This indicates that σ should either be decreased slowly to a high value or fast to a low value. Going too fast to a high value, there are too few steps to reach a good solution. Going too slow to a low value, gives the gradient descent too many opportunities to optimize without obeying sparsity. Overall, when roughly obeying these rules, chances are good for ending up within the area of low RMSE values as it stretches decently far along each parameter. When being in this area, chances are good for obtaining results that outperform the other solvers.

BCS

For BCS, only α can be varied. Choosing different values seems rather arbitrary, therefore all elements of α are set to the same value. The result of the parameter variation is shown in figure 6.4. The x-axis (*alphas*) shows

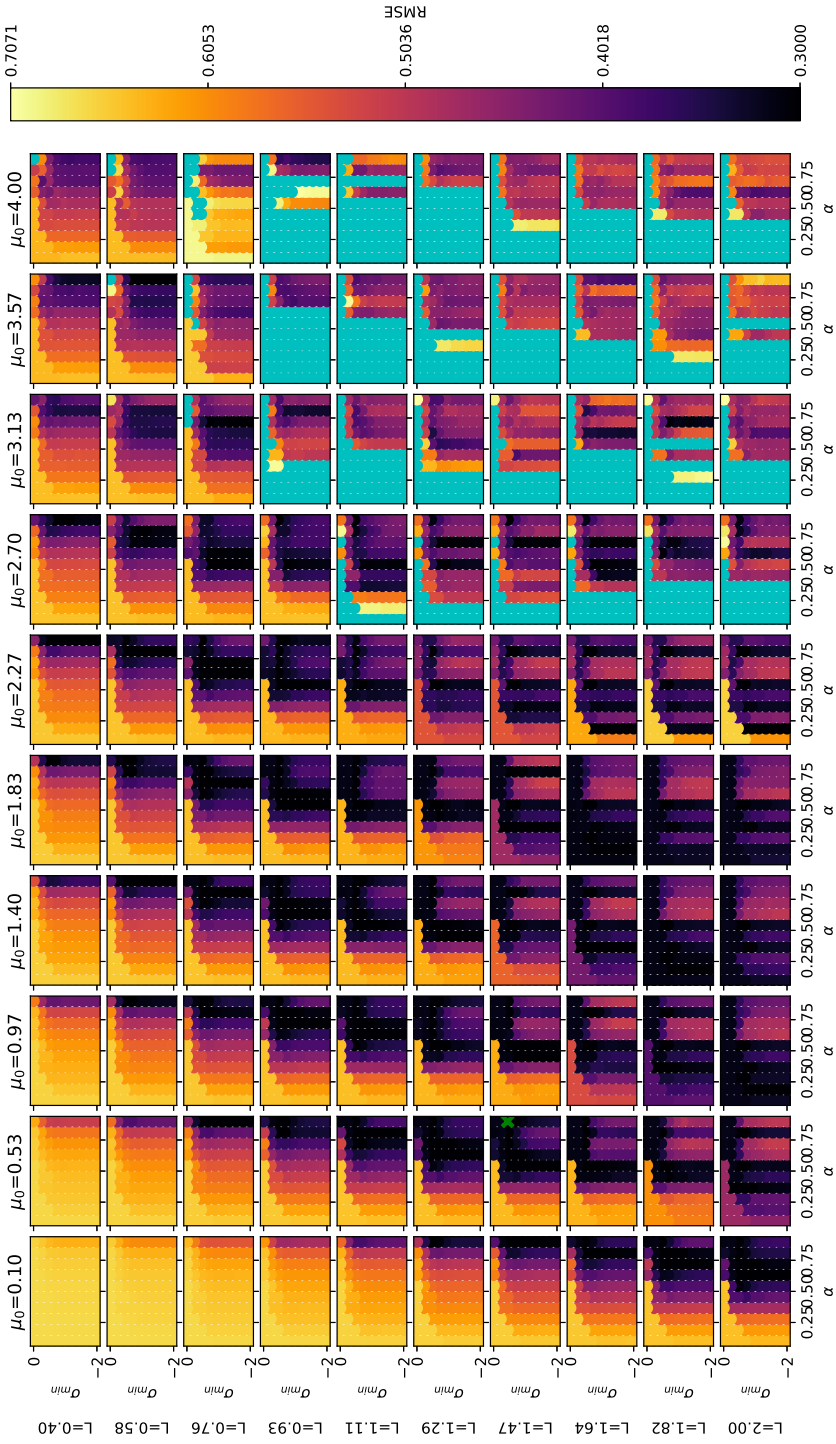


Figure 6.3.: Results for parameter variation of SL0.

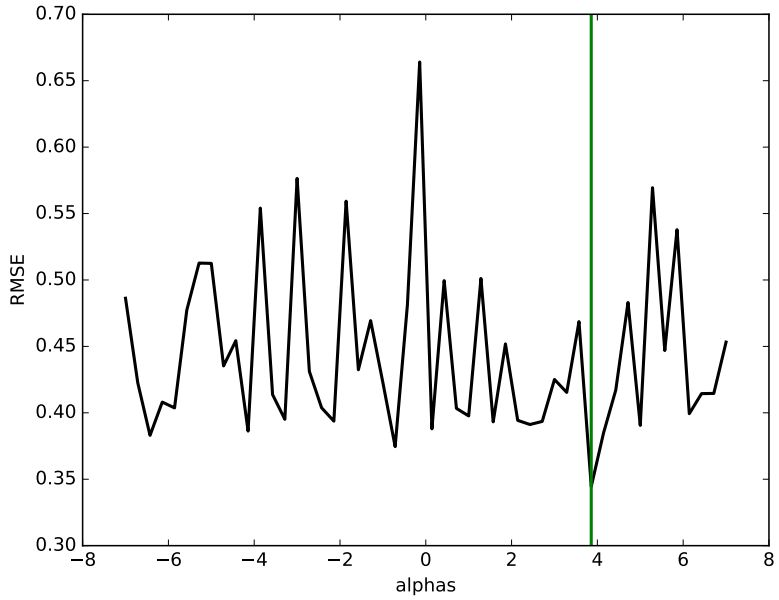


Figure 6.4.: Results for parameter variation of BCS.

the common logarithm of α_i. The RMSE varies mostly between 0.4 and 0.5. However the variation seems chaotic. There is no clear dependency from α_i. This means, BCS requires no parametrization, which facilitates its usage.

OMP

The result for OMP is shown in figure 6.5 for variation of the tolerated error term. The best result is achieved between values of approximately 4 and 10. Outside of this interval, the error is significantly higher. This area needs to be hit, albeit the exact value within the interval is not of importance. Note, that it drastically differs in this point from SL0, LASSO, and ℓ_1 magic in that the tolerance must not be set too low.

Figure 6.6 shows the alternative stopping condition, the sparsity goal. Here again, the value needs to be hit exactly, disabling one of the core ideas of CS, i.e. that sparsity does not need to be known.

Nevertheless, finding the optimum is relatively easy in both cases with sparsity goal being an integer and the tolerance having a well-defined range of good values. Comparing both variants, the optimal value is the same - apparently, the same number of iterations was performed due to the two stopping conditions.

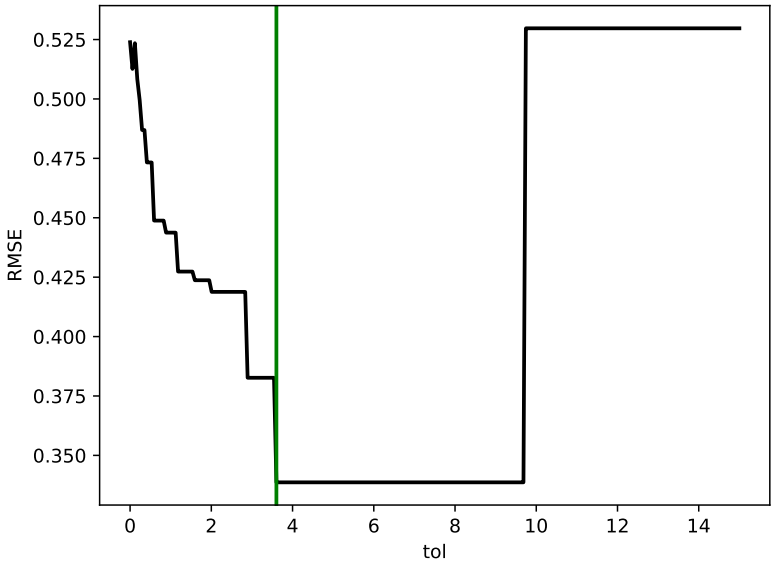


Figure 6.5.: Results for error tolerance variation of OMP.

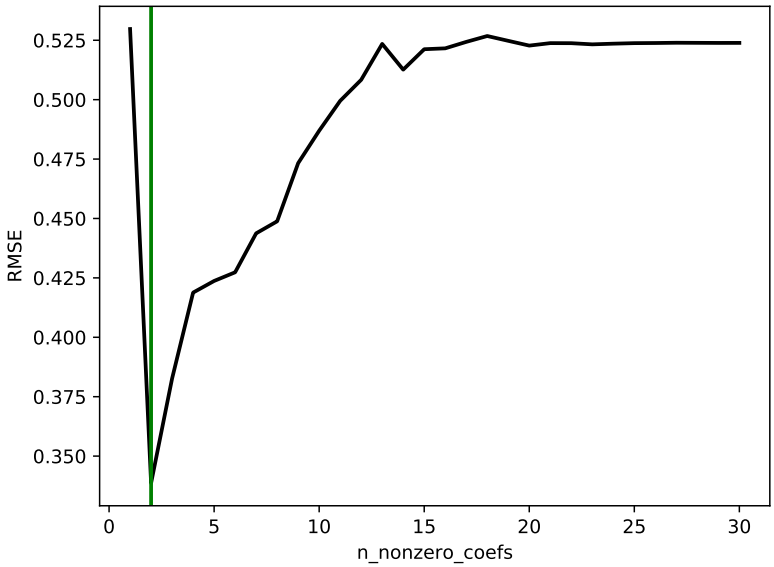


Figure 6.6.: Results for sparsity goal variation of OMP.

Conclusion

Overall, these example parameter scans helped to find simple rules for finding good solver parameters for all solvers that greatly simplify this step in the remaining sections of this chapter. As SL0 quickly turned out to be the best solver overall, this parametrization and configuration step was kept to a minimum, only considering some of the solvers.

6.2. CFE

In this section, we choose a transform and a solver for the CFE based on the sample yield data of a crop field, acquired by a combine harvester. For simplification, we perform these evaluations directly on the whole data instead of using the actual sample process of CFE. It does include an offset removal and re-addition. We refer to this as the static case from here on.

6.2.1. Transform selection[18]

Having designed the concept of CFE in section 4.1, the next step according to the CS recipe from section 2 is finding the best suited transform. One approach for this is measuring the sparsity directly based on existing data. For this purpose, we applied multiple common transformations to the grid data, namely DCT, DFT, and the Haar transform as a representative for DWTs. Furthermore, we considered the untransformed version. As we are focusing on spatial distributions, we used the two-dimensional versions of the transforms, created with KCS.

Figure 6.7 shows the resulting coefficients in descending order. For comparability, they were scaled so that the maximal value is one. In the untransformed case (*Spike*) the magnitude of the highest approximately 1800 values decreases slowly. Then there is a sudden drop from 0.17 to 0 with all remaining values being zero. The zero values are simply those outside the measurement area whereas the non-zero values are those within the measurement area of size $f = 1782$. This border is depicted as a dashed line. The coefficients in Haar domain decrease about as slowly as the untransformed values. The sudden drop is missing, instead the trend is held across all values. The coefficients in frequency domain display a significantly steeper decrease. Only 97 values have a magnitude larger than 0.05 in DFT. For DCT there are only 25 values. After all, especially the two frequency space transformations DFT and DCT are promising candidates to be used in compressed sensing.

To further restrict our selection of Ψ, we compressed and decompressed the grid data using the compressed sensing scheme with all combinations of the transform matrices Identity (*Spike*), Haar, DCT, and DFT and the

[18]The content of section 6.2.1 has been previously published (Hänel et al. 2017a).

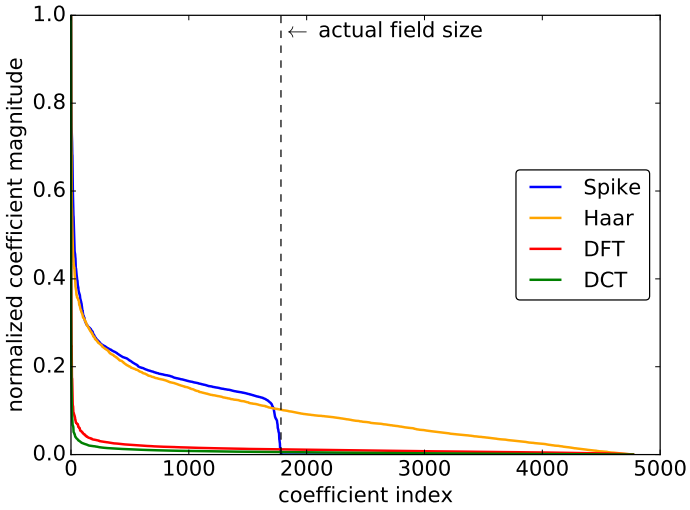
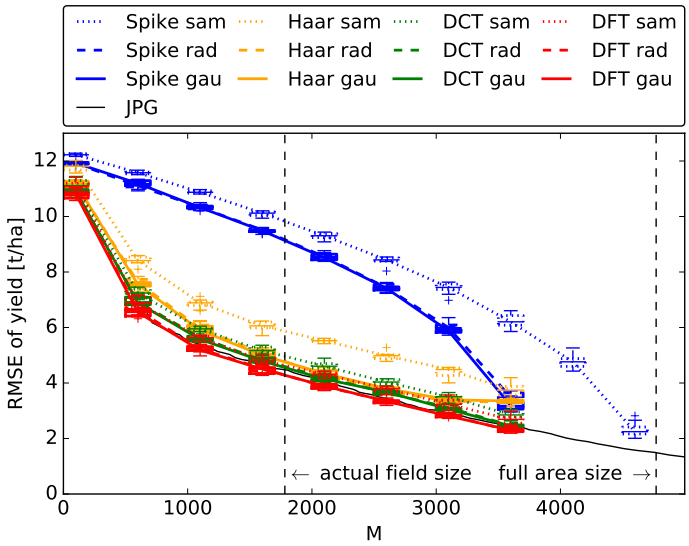


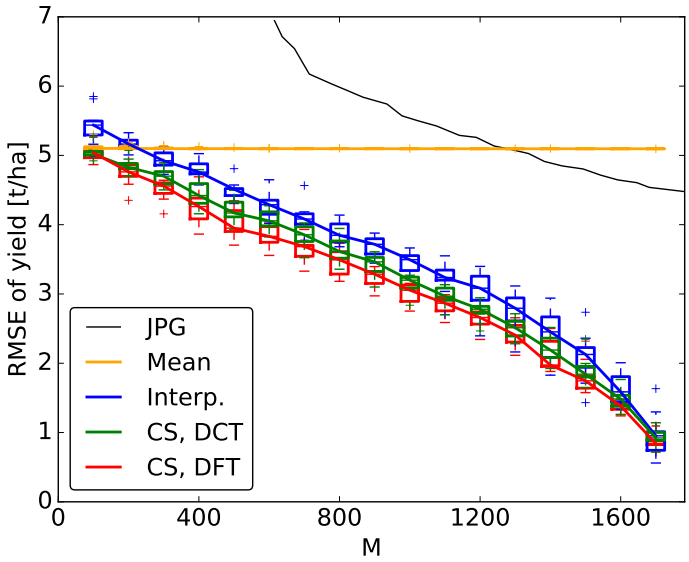
Figure 6.7.: Coefficients in various domains in descending order (Hänel et al. (2017a), © 2017 IEEE).

measurement matrices Identity (*sam*), Rademacher (*rad*), and Gaussian (*gau*). Implicitly, this also contains a test of the third requirement of compressed sensing, the incoherence. CFE being the first CS-based solution considered in this thesis, we included dense measurement matrices to assess the impact of using sparse sampling matrices discussed in sections 2.2 and 2.3.3 for a real-world example. We used the solver SL0 mostly because it supports complex values and thereby DFT. As a metric for the quality of the reconstruction, we use the RMSE between the original and the reconstructed field. This evaluation is limited to the area where we actually have measurements. The results are shown in figure 6.8a where the boxplots are based on random measurement matrices generated from ten different seeds. For comparison, the reconstruction quality of JPEG is shown as well, being one of the most established compression algorithms for two-dimensional fields. For JPEG, we scaled the measured values to the range of a byte. Due to this byte-wise representation, the file size in byte is a metric that is approximately equivalent to M. Therefore, we use it as M in the evaluation.

The RMSE of the compressed sensing results with DCT, DFT, and Haar transform is about as low as the one for JPEG and shows the same continuous decrease in dependence of M. In a stark contrast, the RMSE for the untransformed case (*Spike*) is significantly higher and decreases slower up to an M-value of approximately 3000. Then, the decrease becomes steeper. This change may be attributed to the fact that from here on M is sufficiently higher than the approximately 1700 non-zero values



(a) Reconstruction quality for various pairs of Ψ and Φ.



(b) Samples extracted solely from the relevant area.

Figure 6.8.: Examining the suitability of the data for CS and CFE (Hänel et al. (2017a), © 2017 IEEE).

due to the limited measurement area. Whereas there is no significant difference in the RMSE for the Gaussian and Rademacher measurement matrices, the results with random sample matrices fall behind especially in the case of Haar tranform and the untransformed case. For DCT and DFT the difference is rather small due to the incoherence with random sampling which was proven by Rudelson et al. (2006). The overall disappointing result in comparison with JPEG may be attributed to the waste of samples for reconstructing the values outside the actual measurement area.

However, random samples open up an elegant way for circumventing the problem by just using samples within the measurement area. The result is depicted in figure 6.8b. Note that the task performed by compressed sensing now resembles an interpolation more than a typical compression. Therefore, we included a linear interpolation between the random values for comparison as well as the most trivial interpolation which is just assuming the mean of the samples for the whole field. For comparability with figure 6.8a, we again included the result with JPEG compression. As we are using random samples for the measurement, we limited our evaluations to the transforms DCT and DFT.

Both, the interpolation techniques as well as compressed sensing, yield drastically better results than JPEG because JPEG still suffers from wasting data on the area outside of the field as well as having some more overhead. In contrast to all other techniques, the mean has a constant RMSE of approximately 5 t/ha and barely shows any improvements with an increasing number of samples. Linear interpolation and CS use the mean as default value and improve upon it. Therefore, they become better than the mean assumption even with a small number of samples. CS with DCT shows results that are slightly better than with linear interpolation. Using DFT, the difference gets even clearer, significantly less samples are required for the same reconstruction quality. Note, that this is already a significant result for certain applications: measurements of the mass of plants that may be harvested can be taken at some random locations on the field in order to get an estimate for the yield distribution.

6.2.2. Solver comparison

Having found DFT and DCT as the most promising transforms, the next step according to the CS recipe is selecting a solver. As in section 6.2.1, we consider DFT and DCT with solver SL0. Now, we add the solvers BCS and LASSO in the evaluation. We use them only with DCT as they cannot handle complex numbers. The result is shown in figure 6.9. LASSO performs about as good as SL0 with DCT. BCS performs much worse for unknown reasons on this data. It performs worse than linear interpolation.

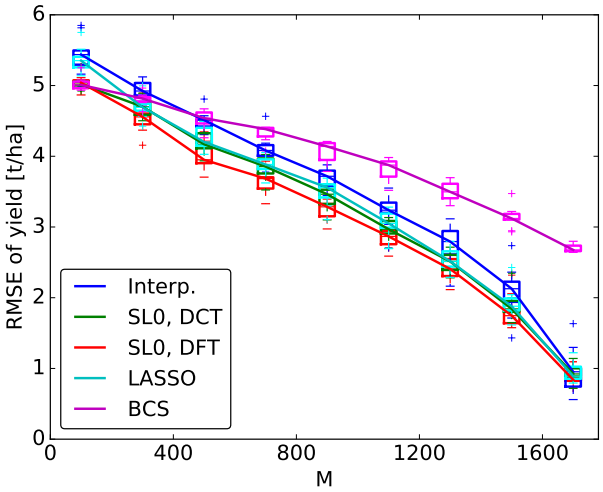


Figure 6.9.: Performance of various algorithms on yield data.

SL0 in combination with DFT delivers slightly but consistently the best results. This makes SL0 the solver of choice, mostly because in contrast to LASSO it works with DFT.

6.3. HeM2HSN

For the HeM2HSN, we compare transforms and solvers analogously to CFE. But first we take a look at the datasets used for the evaluations.

6.3.1. Datasets[19]

For the trace-based evaluation of the HeM2HSN, three datasets as listed in table 6.1 will be used, the first one has been measured directly on the ground in wheat fields with a spectroradiometer of type HR-1024i by Spectra Vista Corporation (SVC) and will give an insight of how spectra typically look when acquired directly in-situ. The second dataset has been acquired using a hyperspectral camera on an airplane and can be used for evaluations at a larger scale. The third dataset has been acquired by satellite. The main advantage of this third dataset is its availability which facilitates reproduction of the results.[20]

[19] The content of section 6.3.1 has been previously published (Hänel et al. 2019).
[20] The satellite data is data available from the U.S. Geological Survey:
`https://earthexplorer.usgs.gov/`,
Entity ID: EO1H1960232013159110KF_SG1_01

Table 6.1.: Parameters of the datasets used in the evaluation.

Source	Area [km^2]	Pixel size	# Spectra	N	Wavelengths [nm]
Ground	3.5×3.5	$< 1\,\mathrm{m}^2$	124	48	400.4 - 1099.2
Airplane	6×2.7	$9\,\mathrm{m}^2$	1.8×10^6	49	406.1 - 1107.5
Satellite	10×100	$900\,\mathrm{m}^2$	9.4×10^5	92	396.3 - 1104.2

6.3.2. Transform comparison

In finding a transform base for the spectra in an earlier Bachelor Thesis by Tessmer (2016), it was shown that the established transforms do not generate proper reconstructions at suitable compression ratios. Considering the extremely low spectral sampling rates necessary for the HeM2HSN, a different transform is required. The core idea here is to model the spectra at different locations jointly, which is achieved with JSM-1 of DCS (see section 2.3.8). Experiments with the other JSMs were discarded early on, as they did not generate suitable results with established transforms.

Generally, Joint Sparsity Model 1 (JSM-1) requires one transform base for the common spectrum and one for each spectrum that is used for expressing the difference between this spectrum and the common spectrum. Clearly, using different transforms for the individual spectra seems rather arbitrary as more knowledge on the different locations needs to be incorporated.

As the evaluation for CFE in section 6.2 showed that the transform base with the highest sparsity is not automatically the one best suited for CS, we skip the measurement of the sparsity. Instead, we directly evaluate the transform bases using the RMSE between original data and reconstruction.

Figure 6.10 shows the resulting values, solved with SL0 for a simple sampling matrix, picking $M = 8$ samples. The transform for the common part is varied along the y-axis, the individual part is varied along the x-axis. Most of the transform bases considered here are from the different DWT families. Apart from these, DCT, DFT, an identity matrix, i.e. a spike base, and difference matrices of first and second order are included. All these matrices were introduced in sections 2.3.1-2.3.4.

The most important observation in this parameter scan is that the choice of the individual transform seems most critical as indicated by the vertical dark green line with lower RMSE values. It indicates the lowest RMSE values only being achievable with the difference matrices, especially the first order for the individual term. With different values for M and different measurement matrices, the same effect can be observed. This is not shown here. The success of the difference matrix makes sense - adding an offset in some wavelength ranges seems like an intuitively correct way of modifying the common spectrum to model the individual spectra.

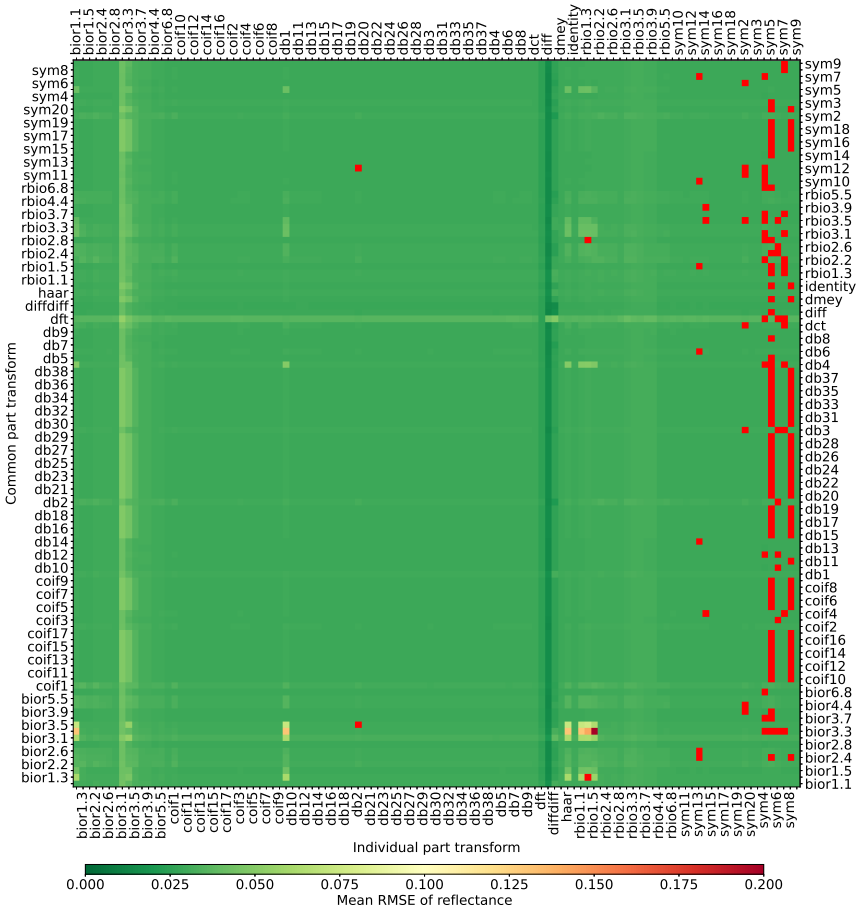


Figure 6.10.: Variation of Ψ_c and Ψ_i for a sampling measurement matrix and $M = 8$.

For the choice of the transform base for the common part, we created a line plot in figure 6.10 as the values are difficult to differentiate in the heat maps. The transform for the innovation terms is the difference matrix here. The five transforms with the lowest RMSE are highlighted. Among them there are two representatives of the Daubechies wavelet family, two representatives of the Symlet family, and the first order difference matrix. For simplicity, we choose the first order difference matrix for both the innovation transform and the common transform.

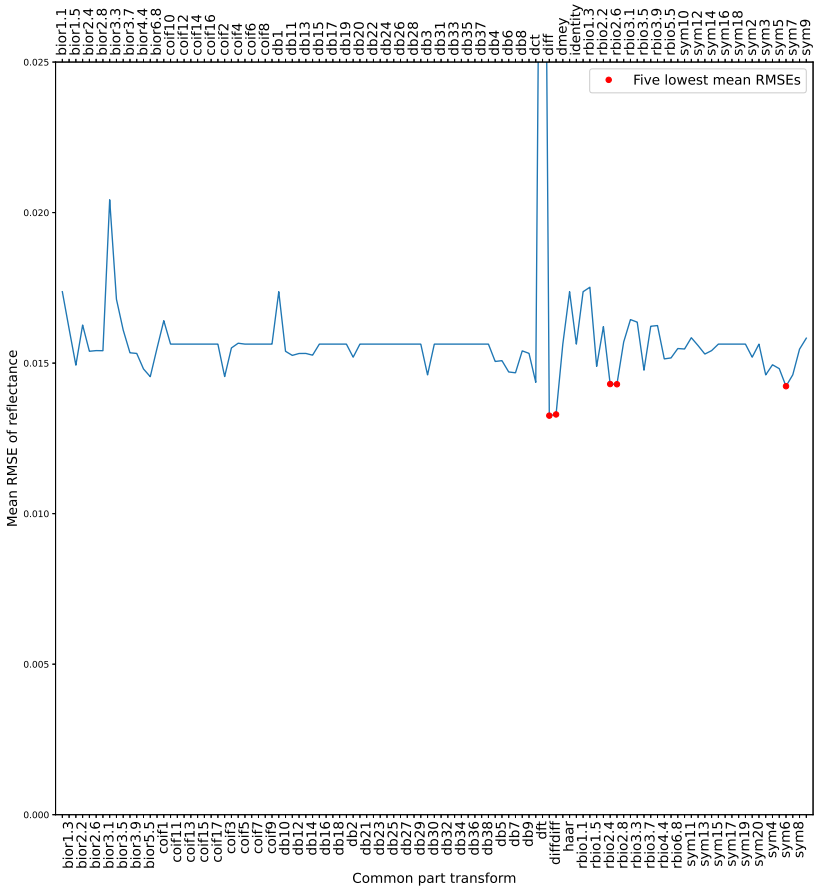


Figure 6.11.: Comparison of DCS JSM-1 performance for different matrix combinations on ground spectra.

6.3.3. Solver comparison

For the solver comparison we use one of the datasets as an example, the satellite dataset. The result is shown in figure 6.12. All solvers generate good reconstructions. The difference between the solvers is significant but smaller than the variation between sensor nodes as the boxplots indicate. SL0 (called *DCS* here for consistency with the other figures) shows the best performance, BCS comes in second, and LASSO performs slightly worse.

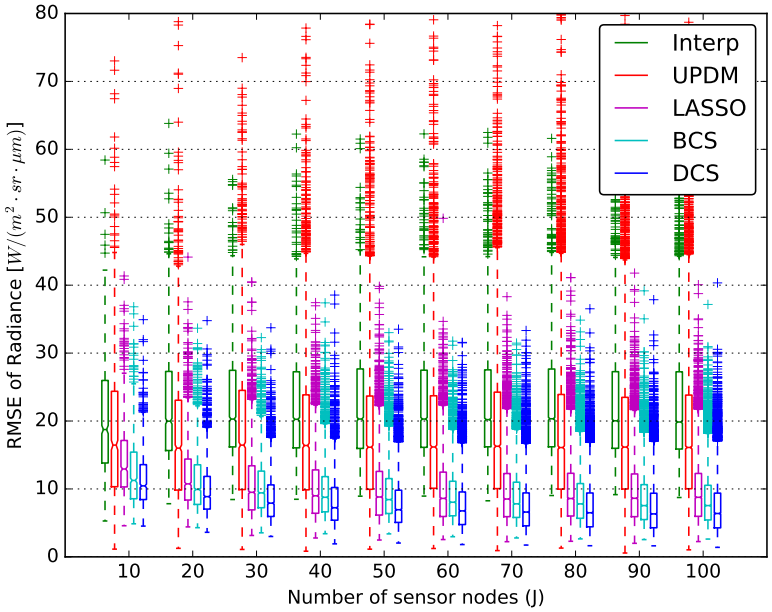


Figure 6.12.: Performance of various algorithms on satellite based M2HSN data.

6.4. HoM2HSN [21]

For the HoM2HSN, we deviate from the strict scheme used for HeM2HSN and CFE: as SL0 always turned out superior, we refrain from testing other solvers. For the transform, the choice has already been made for K-SVD, however, it needs to be parameterized.

6.4.1. Datasets

In this work, we consider two of the datasets considered for the HoM2HSN listed in section 6.3.1: the one with spectra acquired with hand-held sensors on the ground, containing only spectra from agricultural fields. The second is the one taken by an airplane, containing an area mostly consisting of agricultural fields. For the following evaluations we work with these sub-datasets:

Air_Full This is the complete air-based dataset.

[21] The content of section 6.4 has been previously published (Hänel et al. 2021b).

Air_MA, Air_MB These are two sub-datasets acquired by cutting *Air_Full* in two parts of equal size. The *M* stands for *Mixed* as they contain different kinds of land-use.

Air_M1, Air_M2, Air_M3 These are three sub-datasets acquired by cutting *Air_Full* in three parts of equal size. The *M* stands for *Mixed* as they contain different kinds of land-use.

Air_V1, Air_V2 These are two sub-datasets acquired by cutting squares out of *Air_Full*, which solely contain vegetation.

Ground_Full This is the complete ground-based dataset.

Ground_V1, Ground_V2 These two sub-datasets are obtained by cutting *Ground_Full* in two parts of equal size.

These datasets will be used as training datasets, as datasets for the selection of best bands, and for the evaluation of the algorithms. Note, that the air-based dataset and the ground-based dataset originally contained different bands. In order to make them inter-operable, we limited the range of wavelengths to those included in both sensors, from 406 nm to 1100 nm. We then re-sampled the ground-based dataset selecting the same bands as in the air-based dataset. This direction minimizes the resampling error due to the higher spectral resolution of the ground-based dataset.

6.4.2. Parametrization

In this section, the goal is to develop an understanding for how K-SVD manages to cope with the data and to find an appropriate parametrization. In a first step, we determine the size of the dictionary and the appropriate value for the sparsity target. We use a similar value range to the plot by Arad et al. (2016) in order to allow for a comparison. The result is shown in figure 6.13. The sparsity target has been varied from 5 to 50 and the dictionary size from 100 to 400. The training has been performed on *Air_M1*, the best bands have been selected using *Air_M2* and the evaluation has been performed on *Air_M3*. The color shows the median RMSE which we determined by first calculating the RMSE for the spectrum of each pixel in the evaluation area. We then determined the median across all pixels. Lastly, we repeated the process 20 times with different seeds for the band selection and obtained the median of the 20 medians. Note, that we defer a closer investigation of the distribution of the RMSEs to section 7. Curiously, the result differs drastically from the

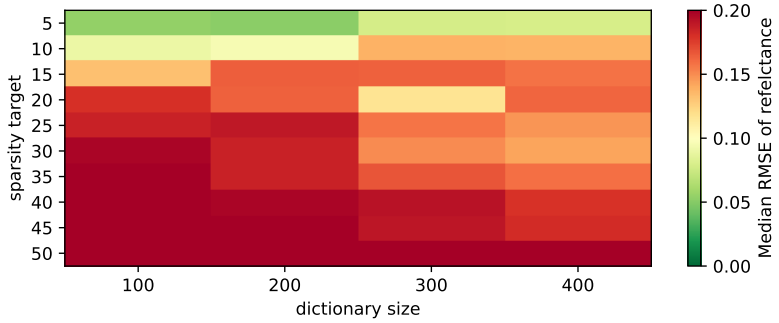


Figure 6.13.: Impact of different dictionary sizes and sparsity targets on reconstruction quality.

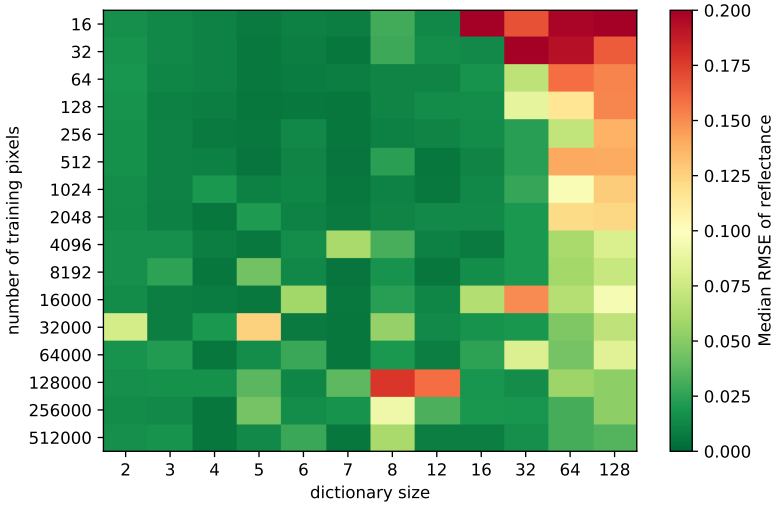


Figure 6.14.: Impact of different dictionary sizes and sizes of the training data on reconstruction quality.

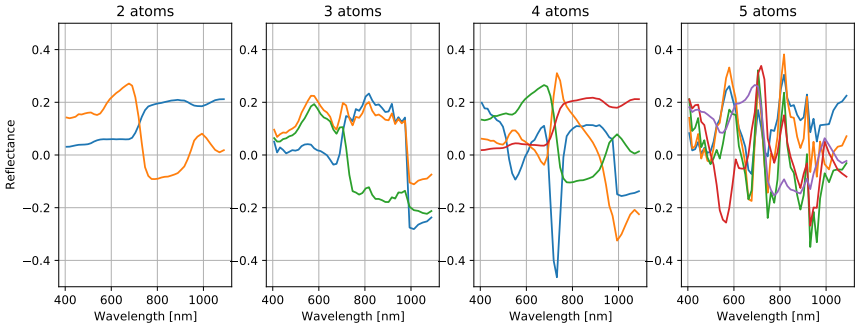
one in (Arad et al. 2016): the error increases when increasing the sparsity target. Increasing the dictionary size only increases reconstruction for higher sparsity targets. However, as the previous effect is stronger, therefore, best results are achieved with small dictionary and low sparsity target. We attribute this to the spectra varying less across pixels, because of the comparably low resolution in remote sensing which leads to averaging out special spectra and because of the relatively homogeneous agricultural environment.

Having found that a very small dictionary suffices already, such a small dictionary should be trainable from a smaller training dataset. Therefore, we evaluated the training dataset size in the next step by just selecting

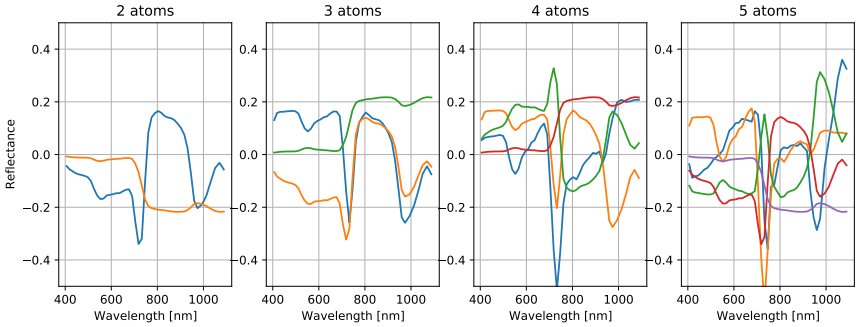
a limited number of pixels at random from the training dataset. This is shown in figure 6.14. It again displays the median RMSE of the reflectance. Instead of the sparsity target, we now vary the number of training pixels. The sparsity target is set to 10 percent of the number of bands in the hyperspectral version of the spectrum. The 10 percent rule is the default of the K-SVD implementation in use. The resulting sparsity target is 4, which is close to the optimum in figure 6.13. Furthermore, the sparsity target is limited to the dictionary size as it's impossible to choose more atoms than available. From figure 6.14 it becomes clear, that surprisingly, a small number of training pixels already suffices for a reflectance RMSE of less than approximately 0.03. Increasing the number of training pixels, mainly helps increasing the result with larger dictionaries. However, as seen already in figure 6.13, a large dictionary leads to a lower reconstruction quality. For dictionary sizes below approximately 16 the results are quite good. Dictionary sizes of 2 and 8 are slightly worse. At the value of 2, we attribute this to the dictionary simply being too small. At the value of 8, we found out that this happens due to SL0 performing poorly when the dictionary size equals the number of bands. The effect does not occur with OMP - but we still stick with SL0 because of an overall better reconstruction quality. Curiously, the reconstruction quality becomes more variable with increasing training set size. We attribute this to an increasing chance of having anomaly pixels in the training set. Just a few of these suffice to create a transform base that tries to cover the anomaly pixels as well. For smaller training sets, this can happen as well in rare cases and will have an even worse effect. However, these cases are not reflected in this evaluation plot as they get rejected when calculating the median.

A nice advantage of such a small dictionary is that it can be visualized for a qualitative investigation. Some samples are shown in figure 6.15. Each plot contains the elements of a trained transform, also called atoms, as line plots. The number of atoms is increased from left to right by retraining with a different dictionary size. The atoms have different colors merely for visualization, the order is arbitrary. Clearly, most atoms are dominated by the red edge and adding more atoms mainly helps refining the representation of the red edge. A comparison of 6.15a and 6.15b shows that this effect may be observed for both datasets. Note that the dimension of this basis is comparable to the one assumed in UPDM with its three base spectra. Due to small number of atoms, which are usually all present in the solution, even using an ℓ_2 -solver becomes viable, turning the approach into a more simplistic approach. However, we still use SL0 which starts with the ℓ_2-solution anyway and can thereby be considered a more general solver. Note, that this model of the spectra, being a linear combination of few base spectra is very similar to the model in UPDM.

In a last step before the main evaluations, we consider the role of the band selection. Rather than developing a sophisticated algorithm and tun-



(a) Trained on Air_Full.



(b) Trained on Ground_Full.

Figure 6.15.: Atoms trained at different dictionary sizes and on different training sets.

ing its parameters as in (Arad et al. 2017), we concentrate on evaluating on how well a band optimization on one dataset may be transferable to another. The result is shown in figure 6.16: the dictionary was trained on *Air_MA* and evaluated for 20 random band sets on the datasets *Air_MB* and *Ground_Full*. The figure shows the resulting RMSE for all band sets sorted according to the RMSE. Each band set is denoted by a different color and same band sets are connected with straight lines to visualize how the order of set quality correlates between datasets. Firstly, as found in (Arad et al. 2017), there are few very badly performing band sets. The remaining bands show similar performance. Now, taking the corresponding position of the sets between datasets into consideration, the high RMSE sets are rejected quite effectively by choosing some of the low RMSE filter sets. However, in the plateau, there are many non-parallel lines, indicating that choosing one of the low RMSE sets is likely to be far less optimal in the other dataset. Therefore, we refrain from using a more sophisti-

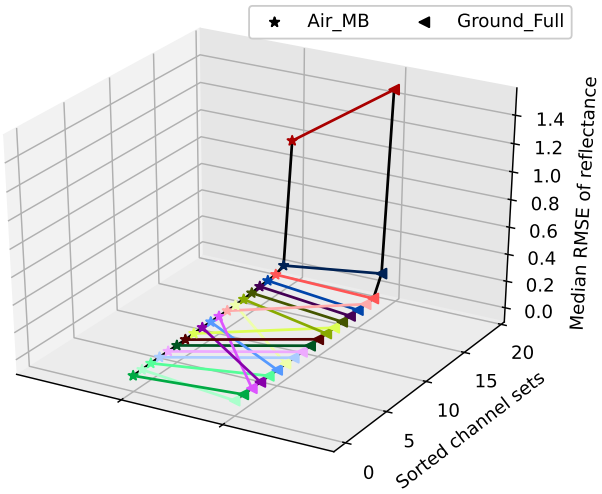


Figure 6.16.: Applicability of band selection between datasets. Training was performed on dataset *Air_MA*.

cated algorithm and just select some of the good bands in the following evaluations, as this brings a major part of the improvements with far less effort.

Overall, we have found that the training size has relatively low influence on the result, while the sparsity target and even the dictionary size should be chosen surprisingly low. More precise values will be further investigated in section 7.3.

6.5. Conclusion

In this chapter, we have selected transforms and solvers for CFE, HeM2HSN, and HoM2HSN. We also parametrized these as far as required. These settings will be used for the evaluations in the next chapter 7. Furthermore, the datasets used in the evaluations have been presented.

7. Evaluation

In this chapter, in sections 7.1–7.3, the performance of the approaches described in chapter 4 with the configurations from chapter 6 will be evaluated. For the M2HSN, an additional evaluation of a real-world continuous measurement is supplied in section 7.4. The quality metrics from chapter 5 are evaluated for synthetic data in section 7.5, for CFE in section 7.6, and for the HeM2HSN in section 7.7.

7.1. Trace-based Evaluation of CFE [22]

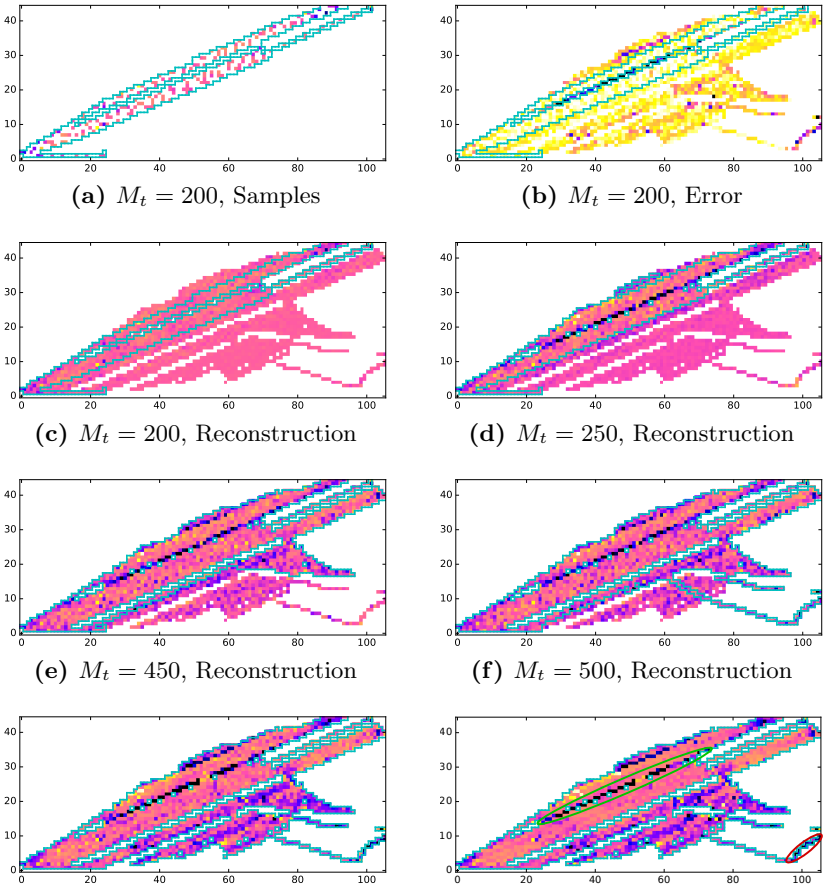
Having shown the achievable performance when sampling from the whole field area, the static case, in section 6.2.1, we now consider the case of transmitting samples while driving and, therefore, not having all values available using CFE as proposed in section 4.1. This approach will be called CFE or the dynamic case from here on. As discussed in section 4.1, we applied the three different strategies for selecting samples for transmission in the dynamic case. The limited bandwidth is simulated by being able to transmit only one sample every $\tau = 15\,\mathrm{s}$ while measuring a sample every $\delta = 5\,\mathrm{s}$, which are values typically used for low- and high-rate sampling.

7.1.1. Qualitative evaluation

In order to gain an intuition for CFE's performance, we first compare its performance at different points of time and thereby a different number of received samples. For now, we limit our considerations to the sampling strategy CFE_{latest}, we introduced in section 4.1.2. The samples that have been transmitted as well as the results after reconstructing the field solving equations (4.1) and (4.2) are shown in figure 7.1. Note that in favor of a more compact visualization, we have omitted the color bar and changed to integer indexes for the grid cells rather than longitude and latitude. This information may be obtained from figure 4.2.

Figures 7.1a–7.1c show the situation after having received the first $M_t = 200$ samples. Figure 7.1a shows the samples that have been received. The borderline of the area which has been explored by the harvester at this time

[22] The content of section 7.1 has been previously published (Hänel et al. 2017a).



(a) $M_t = 200$, Samples

(b) $M_t = 200$, Error

(c) $M_t = 200$, Reconstruction

(d) $M_t = 250$, Reconstruction

(e) $M_t = 450$, Reconstruction

(f) $M_t = 500$, Reconstruction

(g) $M_t = 1200$, Reconstruction

(h) $M_t = 1780$, Reconstruction

Figure 7.1.: Development of the reconstruction quality during harvest (Hänel et al. (2017a), © 2017 IEEE).

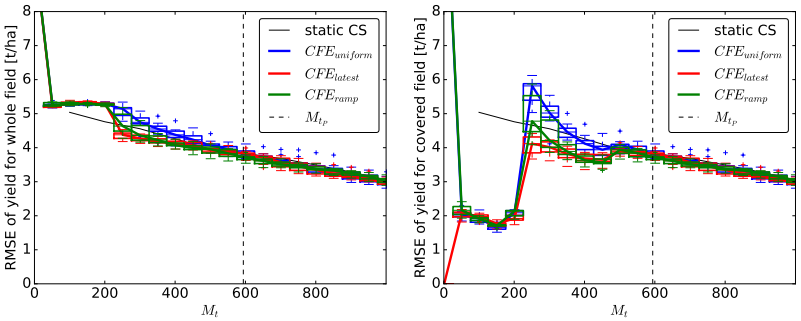
is shown for comparison. The samples display a homogeneous distribution within the explored area due to the sampling strategy CFE_{latest}. In figure 7.1c, the reconstruction result is shown.

Note, that we have set all cells outside the field to 0 as this area has no relevance and would be rather distracting. In the areas far from the explored area, the values default to the mean due to its temporary removal in equations (4.1) and (4.2) and display relatively high deviations (Fig. 7.1b). In the explored area, the distribution already has some similarity to the original distribution (see figure 4.2 or 7.1h for comparison).

In the upper left of the field there is a row with high deviations. The original distribution reveals that it coincides with a row of higher yields highlighted by a green ellipse in figure 7.1h. Shortly after, when $M_t = 250$ readings have been received (Fig. 7.1d), this row has been crossed by the harvester and seems to be well captured in the reconstruction (Fig. 7.1d) but there are still some errors along this row. After $M_t = 450$ readings (Fig. 7.1e), diagonal stripes which coincide with the movement pattern of the harvester become apparent in the reconstruction. This is actually a step in the right direction when comparing with the real distribution in figure 7.1h which also contains such stripes. This structure was forced by the assumed DFT which demonstrates an explanation for DCT's and DFT's superiority determined in section 6.2.1. After $M_t = 500$ (Fig. 7.1f), nearly the whole area has been explored by the harvester. Afterwards (Fig. 7.1g), the reconstruction gradually improves by filling the gaps with more values, getting very close to the original (Fig. 7.1h) and ultimately identical to it after transmitting all values.

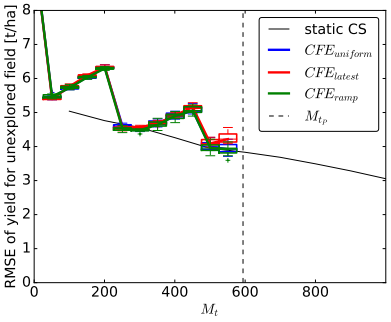
7.1.2. CFE as field estimate

Although the distributions in figure 7.1 provide a more detailed understanding, this is not suitable for a broader evaluation. Instead, we again measure the quality of the field estimate using RMSE as in section 6.2.1, this time for the dynamic case. Therefore, the M_t axis now also serves as a time axis. The results are shown in figure 7.2a. The behavior may be roughly separated in two phases: in the first phase, the harvester did not explore the whole field. Here the reconstruction is a mixture of compression and forecast. The change between phases occurs at $t_P = t_0 + F \cdot \delta$ or after receiving $M_{t_P} = t_P/\tau = 594$ samples. Initially, all dynamic sampling strategies perform worse than the static case. The RMSE decreases with increasing M_t for the dynamic as well as the static case. However, the decrease is less regular but overall steeper in the dynamic case. Even before t_P, all strategies achieve RMSEs about as low as in the static case. This is actually a remarkable result as it shows that the influence of the restricted sampling is so insignificant for the whole field that it disappears even before having covered the whole field. The results also help us finding the appropriate sampling strategy: Mostly, the reconstruction quality is not significantly different. However, in the phases where it is different, the result is somewhat surprising considering the trade-off discussion from section 4.1.2: CFE_{latest} is a clear winner. It continuously shows better results than the other strategies with $CFE_{uniform}$ coming in last and CFE_{ramp} in between the two. We attribute the superiority of CFE_{latest} to the fact that the achieved homogeneous distribution of samples dominates the reconstruction quality whereas the resulting mea-



(a) Estimate of whole field.

(b) Estimate of explored part of field.



(c) Forecast for unexplored part.

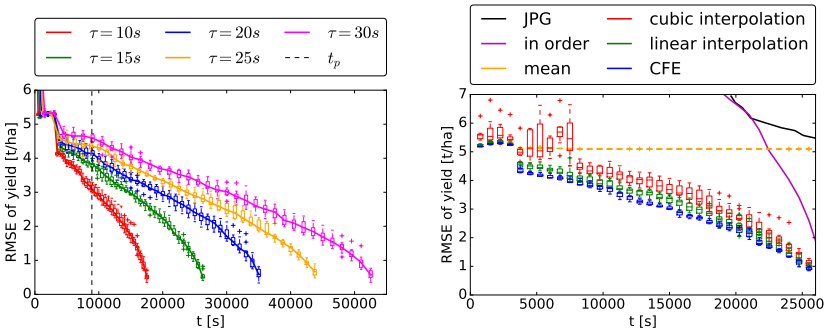
Figure 7.2.: Reconstruction quality in the dynamic case with different sampling strategies and metrics (Hänel et al. (2017a), © 2017 IEEE).

surement pattern does not become less incoherent with the transformation base because sufficient randomization is induced by the route of the harvester as well as the jitter.

The first phase requires some more investigation in order to separate the influence of compression and forecast.

7.1.3. CFE as compression

First, we evaluate the influence of the compression area by calculating the RMSE only for the part of the area that has been explored by the harvester already. The result is shown in figure 7.2b. As expected, in the beginning, the overall RMSE is mostly lower than for the whole field



(a) Comparison of bandwidths.

(b) Comparison with alternatives.

Figure 7.3.: Reconstruction quality over time (Hänel et al. (2017a), © 2017 IEEE).

estimate (Fig. 7.2a). Note that from M_{t_P} onwards, the values are the same as in figure 7.2a because the area that contributes to the RMSE calculation becomes the same in phase two.

The ranking of the sampling strategies is the same as for the whole field estimate. At the beginning when only few values from a highly localized area have been transmitted, the RMSE is highly volatile. Whereas this may be partially attributed to the small sample size of grid cells, the course is qualitatively the same for all random seeds and sampling strategies, indicating that it is mainly influenced by the real data of the field and the exploration path. When comparing with figure 7.1, e.g., the steep increase between $M_t = 200$ and $M_t = 250$ may be explained: The row of high yield values in the upper left area of the field has been harvested in between these points of time as a comparison of figures 7.1c and 7.1d reveals. In several cells, reconstruction errors were made around this row, generating the steep increase. The difference between the strategies which is observable in figure 7.2a as well, is most significant during and after this increase.

7.1.4. CFE as forecast

In the next step, we take a look at the reconstruction quality in the unexplored area. This is shown in figure 7.2c. Naturally, this evaluation can only be performed up to t_P as there is no unexplored area left afterwards. The RMSE is significantly higher, indicating that the forecast is not very good. The difference between the strategies coincides with the previous observations. Most promising are the drastic improvements of reconstruction quality between $M_t = 200$ and $M_t = 250$ as well as the

one between $M_t = 400$ and $M_t = 450$. Comparing with the compression performance, the first increase clearly coincides with the steep decrease of reconstruction quality in figure 7.2b, indicating that a difficult to reconstruct area has been shifted from unknown to known space, the row of high yields discussed for the compression performance. The same occurs between $M_t = 400$ and $M_t = 450$: The difficult area here has been the narrow path in the lower right corner. Figure 7.1f reveals that the values have been drastically underestimated in comparison with those in the real distribution which is marked by a red ellipse in figure 7.1h. In the beginning and in between, the reconstruction quality slowly increases, as the unexplored area decreases and, therefore, the relatively small difficult areas contribute more to the RMSE.

7.1.5. Timeliness of results in comparison

In the previous sections, we mainly used the number of transmitted values M_t in order to compare algorithms, which is a perfectly suited metric for compression performance. However, the availability of data depends on the time and, therefore, we evaluate reconstruction quality over time. This allows us to evaluate the influence of the bandwidth. Thereby, we can assess the situation of having only a limited bandwidth available which we assume to be the usual case. Note that we mainly aim for making good use of a given limited bandwidth rather than trying to save bandwidth. However, one may instead put a lower limit to enforce bandwidth savings. We varied τ, which is directly linked to the available bandwidth b with $b = \gamma/\tau$ where γ is the size of a transmitted tuple. The resulting RMSE over time is shown in figure 7.3a. The decrease of the RMSE is stretched further over time the lower the bandwidth. An additional effect from the availability of the information for transmission is not observable and thereby negligible.

Taking the timeliness into consideration also allows to expand our considerations to more algorithms. The result is shown in figure 7.3b: The most basic approach is sending data in the order of measurement, however this leads to a very slow decrease of the RMSE. In order to perform a JPEG-compression, first all data needs to be known. Therefore, it cannot be done before t_p. Assuming that compression can be performed nearly instantly, the time of the completed reception as well as the reception quality depend on the selected quality. Therefore, the line for JPEG in figure 7.3b has a slightly different meaning: for all other approaches, the reconstruction quality decreases over time. In contrast, for JPEG, just one point on the line is chosen via the quality level.

The approaches that send current data on the go yield better reconstruction quality earlier. Even the rather simple approach of simply predicting the mean of the collected samples for the whole area easily outperforms *in order* and *JPEG* for a long time. The remaining approaches are based

on the data transmission scheme of CFE. As it turned out to be the best, we limit our considerations to the strategy CFE_{latest} here. Like in the static case, we choose interpolation techniques for comparison. During interpolation, we again default to the mean of the measured samples outside the convex hull of samples. We have used the two common interpolation methods *linear* and *cubic*. Whereas *cubic* tends to overshoot and, therefore, offers highly volatile reconstruction quality especially in the beginning, *linear* interpolation leads to relatively good reconstruction qualities at all times. CFE is superior for a long time range. Only at the beginning and in the end, linear interpolation can keep up.

7.2. Trace-based Evaluation of HeM2HSN[23]

Prior to spending effort on building an actual WSN, it is desirable to get an estimate on the potential of such a system by means of trace-based simulations which we supply in this section.

7.2.1. In-situ Data

We first evaluate the approach on the ground-based dataset described in section 6.3.1. We consider a total of 124 wheat spectra acquired at different times and locations from an area of approximately $3.5\,\text{km} \times 3.5\,\text{km}$, giving a good overview on how wheat spectra vary. The spectrometer acquires data between 350 nm and 2500 nm. Thereof, we consider the 482 bands with center wavelengths from 400.4 nm to 1099.2 nm to match the range typical of silicon-based photodiodes (400 nm to 1100 nm) as close as possible. These are light sensors suitable for low-cost sensors. For an increased wavelength range, significantly more expensive sensors are required. We investigate one sample reflectance spectrum. It is shown as a black curve in figure 7.4. The blue, green, and red vertical lines show the wavelengths of the corresponding colors in visible light and are included for orientation. The spectrum is dominated by a step near the color red and the border of visible and infrared light. It is called red edge and serves as an important indicator for the condition of plants (Mulla 2013). In visible light below 700 nm, the reflectance is lower and in infrared light above 700 nm the reflectance is higher. In both ranges, the variations are smaller but still contain features important for gaining information on the plants. No significant variations occur for very small wavelength changes. This implies that the 482 bands in the relevant spectral range are significantly more than necessary.

As we want to concentrate on how far we can reduce the resolution, we first reduce it by throwing away redundant values so we can concentrate

[23]The content of section 7.2 has been previously published (Hänel et al. 2019).

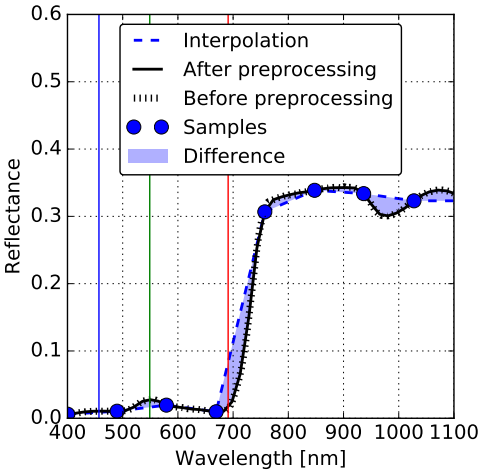


Figure 7.4.: Preprocessing of a high-resolution spectrum and qualitative evaluation of its reconstruction from a low-resolution spectrum using interpolation.

on the harder part of resolution reduction. This addresses the issue discussed in section 2.2.1 of trying to reconstruct too many values. After this preprocessing step, we keep $N = 48$ bands. The bands are selected such that the center wavelengths are closest to equidistant wavelengths. This was not given in the original data as it is common to make bands wider at high wavelengths because energy density is lower in this range in the spectrum of the sun. This is not necessary in a stationary M2HSN as the light sensors can instead be configured individually to collect photons for a longer time at long wavelengths. Both the spectra before and after preprocessing are shown in figure 7.4 as a black dotted curve and as a black solid curve. They barely differ, showing that the resolution reduction kept all the relevant information. The resolution after preprocessing is what we call the hyperspectral resolution from here on.

For the multispectral version, we begin the evaluation with $M = 8$ approximately equidistant wavelengths. Eight bands are a convenient choice because many ADCs offer eight channels. Using linear interpolation, the samples and the result are shown for the example spectrum in figure 7.4 in blue. The differences to the spectrum at the hyperspectral resolution are highlighted in a lighter blue. Some extrema got cut off, making the resulting spectra unsuitable for many tasks.

For a qualitative comparison, we perform DCS on all $J = 124$ spectra which gives the results shown in figure 7.5. For the following evaluations we use the RMSE between the original spectrum and the calculated spec-

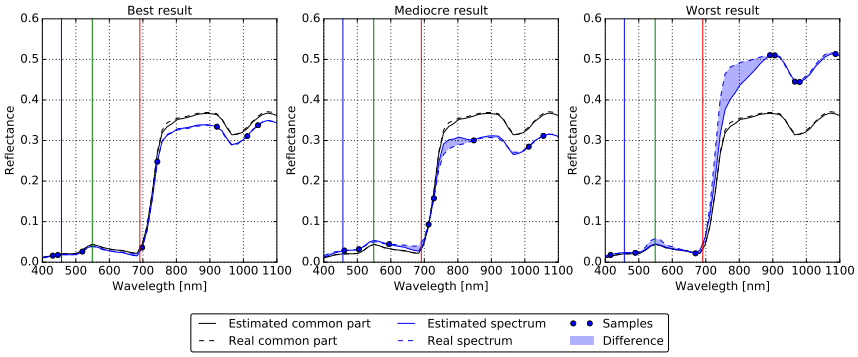


Figure 7.5.: Qualitative comparison of the hyperspectral estimate using DCS in comparison with the original data for the minimal, median, and maximal RMSE.

trum. Such a metric is better suited than a comparison of resulting soil or plant metrics as it evaluates the spectrum in all bands and making all bands available is the main purpose of the approach. The bands sensed by the sensors are now selected randomly and again shown with blue markers. The best spectrum, the median spectrum, and the worst spectrum as indicated by the RMSE are shown as solid blue curves to give an overview of the effects. The original spectrum (dashed blue), the mean spectrum or common part (dashed black), and its estimate (solid black), which is a by-product of JSM-1, calculated as $x_c = \Psi_D f_c$, are included for comparison. The best result as well as the common part include all relevant features of the spectrum, deviations are negligible. The mediocre spectrum is just as good across most wavelengths, only the range near the red edge shows some deviations at approximately 750 nm. In the worst spectrum, the reflectance in the range between 700 nm and 900 nm has been significantly underestimated. We attribute this to the lack of samples in this range and thereby defaulting to values closer to the common part. Additionally, there are some smaller differences. This indicates that the data may at some sensor nodes get insufficiently reconstructed in parts of the spectrum. However, these deviations may be identified during evaluation of experts or in an automated process by finding discrepancies over time, space, or wavelength:

If the spectrum behaves un-normally for a short time, this may indicate a problem with the estimate. If it differs drastically from its spatially neighboring sensor nodes, this may be an indicator as well. Many plant properties can be extracted from features at multiple wavelengths. If the features at different wavelengths indicate contradicting information on a plant property, this is an indicator for a bad estimation as well. We will revisit this topic in section 7.7.

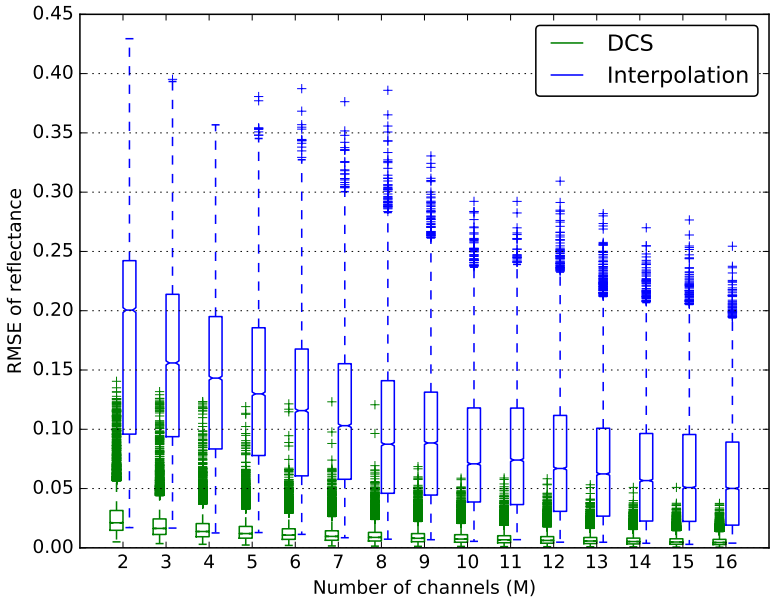


Figure 7.6.: Quantitative comparison of DCS and linear interpolation at different multispectral resolutions for ground-based measurements.

For a more thorough comparison of linear interpolation and DCS, we have repeated the calculation for different values of M as well as 100 different seeds for the spectrum selection. The results for both DCS and interpolation are shown in figure 7.6. Each box of the boxplot contains the RMSEs of all combinations of spectra and seeds, a total of 12,400 values per box. The RMSE decreases with increasing M for both variants. The RMSE of DCS is significantly lower than for the interpolation, demonstrating the benefit of incorporating multiple spectra to get a better estimate. As it was mainly designed for remote sensing data, we defer the evaluation of UPDM to the next datasets.

7.2.2. Aircraft-Based Remote Sensing Data

The second dataset has been acquired by airplane in a rural area in the east of Germany with a hyperspectral camera (HySpex VNIR-1800 by Norsk Elektro Optikk). Figure 7.7 shows an image generated from the hyperspectral data. The area of $6\,\mathrm{km} \times 2.7\,\mathrm{km}$ contains agricultural fields with and without plants, forests, and small villages, giving a representative mix found in rural areas. There is also a small lake covered by the blue marker, making all base spectra available which are necessary for UPDM. The coordinates refer to Universal Transverse Mercator (UTM) zone 32N.

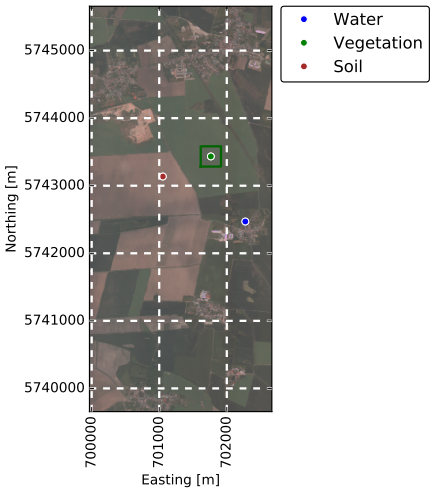
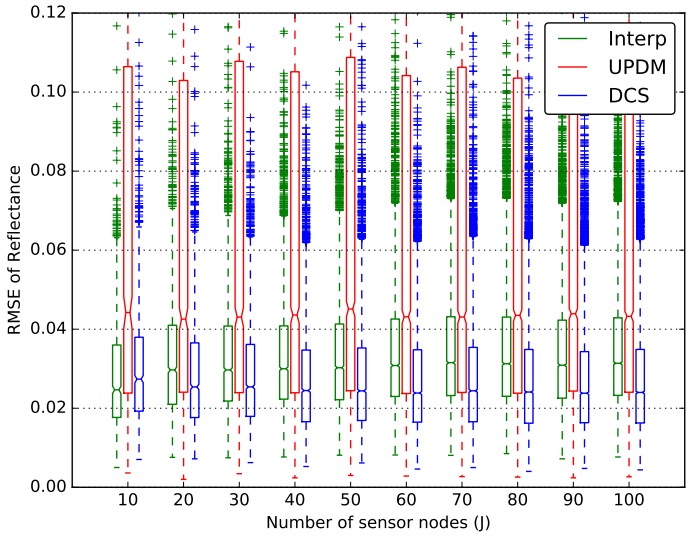


Figure 7.7.: Image from hyperspectral remote sensing data used for larger scale evaluation. The locations for UPDM base spectra are indicated by points. The green square indicates the area used for vegetation-only evaluation.

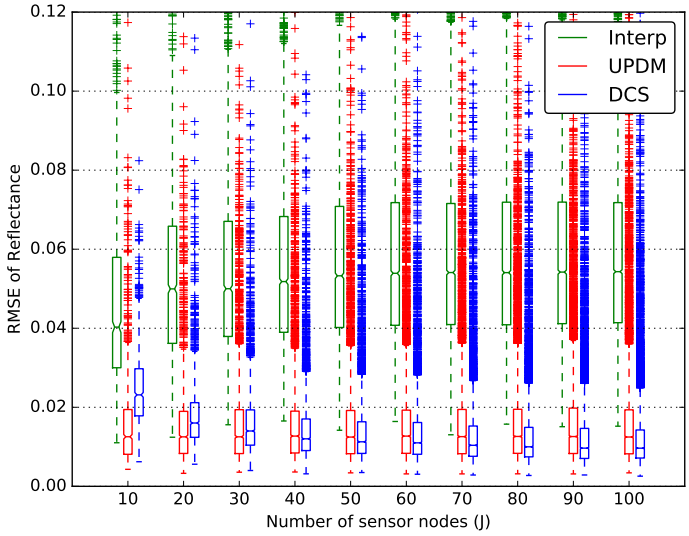
The image has a spatial resolution of 3 meters per pixel and a total of 122 spectral bands between 406.1 nm and 2482.2 nm. As for the in-situ data, we only use the $N = 49$ bands up to 1107.5 nm to match the range of silicon-based photodiodes. Both the spectral range and the spectral resolution approximately match those of the in-situ data. We fix M to 8 and vary the number of sensor nodes J. For each value of J, we pick J pixels randomly from the map, serving as sensor node locations, and repeat this selection of pixels with $S = 100$ different seeds.

We calculated the estimates for the high-resolution spectrum using all three algorithms. For interpolation and DCS, the procedure is the same as in section 7.2.1. For UPDM, we extracted the base spectra of soil, vegetation, and water directly from the map at the locations indicated in figure 7.7.

The resulting RMSEs are shown in figure 7.8a. Each box of the boxplot contains the RMSE for all positions at all seeds, a total of $J \cdot S$ values per box. For UPDM, the RMSE is highest and varies the most. This may be attributed to the problem that the spectra at some locations are not well-representable with the base spectra. The interpolation and DCS results are significantly better. The DCS results are the best for approximately $J \geq 20$. This shows clearly that DCS profits from incorporating



(a) Data selected from the whole map.



(b) Data selected from the square containing solely vegetation.

Figure 7.8.: Comparative evaluation of the high-resolution spectrum reconstruction using interpolation, UPDM, and DCS on aircraft-based remote sensing data.

measurements from multiple locations. In contrast to DCS, the RMSE for interpolation and UPDM is independent from J. For approximately $J \geq 40$ the DCS results show few further improvements.

As WSNs will often be deployed in more homogeneous areas, such as an agricultural field with only one type of crops, we limited the evaluation to such an area in the next step. The area of 300 m × 300 m is indicated by a green square in figure 7.7 and the result is shown in 7.8b.

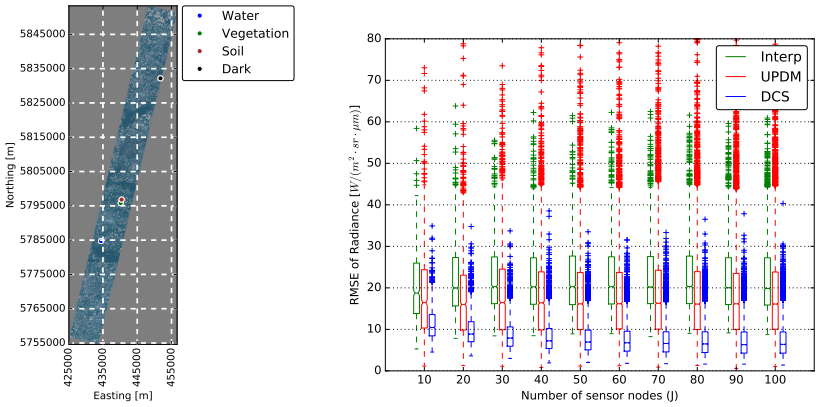
The interpolation performs much worse because vegetation spectra show more variations and are thereby more difficult to approximate with an interpolation than other spectra such as soil or water. UPDM performs much better as the spectra are well-describable by the base spectra and it delivers the best result for a small number of sensor nodes. It is outperformed by DCS for approximately $J \geq 40$. DCS's overall quality is better than for the whole area (see Fig. 7.8a) as the spectra are more similar, making the common part of JSM-1 a more appropriate assumption.

7.2.3. Satellite-Based Remote Sensing Data

The third dataset has been acquired by the EO-1 satellite with the hyperspectral camera Hyperion. Although having quality issues in comparison to the previous datasets due to an incomplete calibration of the bands and some bands not being operative, we included this dataset for assessing the approaches when dealing with faulty data and at an even larger scale. Furthermore, the data is freely available[24], allowing for a reproduction of the results.

The area has a size of approximately 10 km × 100 km and contains a mixture of urban areas, forests, wetlands and mostly agricultural areas. An image generated from the hyperspectral data is given in figure 7.9a. It has a spatial resolution of 30 m and a total of 242 bands in the range from 355.6 nm to 2577 nm. We processed the data as for the previous datasets and select the $N = 92$ bands between 396.3 nm and 1104.2 nm for the evaluation. For atmospheric correction, dark object subtraction assuming the dark object marked in the map has been applied, this approach was explained, e.g., by Song et al. (2001) and by Chavez Jr (1989). The evaluation follows the same pattern for the airborne data with a fixed number of $M = 8$ bands per sensor node and a varying number of sensor positions J picked at random with $S = 100$ different seeds per value of J. The resulting boxplots are shown in figure 7.9b. It shows the RMSE of the radiance instead of the reflectance due to the incomplete calibration. For interpolation and DCS, all outliers are included in the figure. For UPDM, several outliers are significantly higher. The median of the RMSE for the interpolation is the highest. UPDM yields medium RMSEs while DCS

[24]The satellite data is data available from the U.S. Geological Survey:
`https://earthexplorer.usgs.gov/`,
Entity ID: EO1H1960232013159110KF_SG1_01



(a) Map with locations of base spectra and dark object marked.

(b) Evaluation results using interpolation, UPDM, and DCS.

Figure 7.9.: Map generated from EO-1 Hyperion data and comparative evaluation of the high-resolution spectra.

achieves the lowest RMSE and again benefits from an increased number of sensors. UPDM's mediocre performance in comparison to the good performance in the vegetation-only evaluation on the airborne data and the bad performance on the mixed evaluation on the airborne data may be explained by the relatively high percentage of area covered by vegetation as shown in figure 7.9a.

Besides the median RMSE, the spread of the RMSE is important, too. It indicates how reliable a low RMSE can be achieved. For UPDM, the spread is very high and the above mentioned outliers clearly indicate, that the error is significantly higher than with interpolation, albeit only in a few pixels. This may be attributed to some pixels not being describable as a composition of the base spectra. Overall, DCS generates the lowest errors at high reliability while UPDM is highly dependent on the choice of base spectra matching the whole area to be evaluated.

7.2.4. Discussion

The evaluation results allow for a more thorough discussion of design choices when realizing an M2HSN. First, we can make a more informed statement on the scale of the network. In both remote sensing datasets using more than approximately 40 sensors nodes did not generate further RMSE improvements which opens up a way to improve the scalability of the calculation: The sensor network may be partitioned in groups of approximately 40 sensor nodes and DCS may then be applied to each group

of nodes separately at low loss of estimation quality. With respect to the calculation time, which is higher for DCS, such a partitioning helps reducing the overall calculation time for large networks.

Note that the improvement up to 40 sensor nodes shall not be used to draw a conclusion such as: more sensor nodes have to be deployed to obtain an improved spectral reconstruction. Instead, the number of sensor nodes will be dictated by the required spatial resolution. If more than 40 nodes are necessary for the desired spatial resolution, the spectrum reconstruction will be good as well with DCS. For a smaller number of nodes, UPDM is a better choice, if the base spectra are well-chosen. If that is not possible, more spectral bands or even an actual hyperspectral sensor on each sensor node may be necessary.

Besides the number of nodes, the viewing angle of the sensor and thereby the surface area sensed by this sensor needs to be chosen as discussed in section 4.2.1. The evaluations showed that the approach works for a great range of sensed surface areas with $900\,\mathrm{m}^2$ in the satellite data, $9\,\mathrm{m}^2$ in the airborne data, and less than $1\,\mathrm{m}^2$ in the in-situ data, introducing no stricter limits on the choice of the viewing angle.

The number of spectral bands enables an estimate for the communication requirements. Eight bands turned out to be a good compromise. Assuming 16 bit ADCs, this totals at 16 Bytes per reading. Timestamps are not necessary - due to the soft timing requirements, the reception timestamp at the fusion center will suffice. Information about location and selection of bands are also not required because they can be stored at the fusion center.

Nevertheless, out of the stored information, the fusion center has to select the part corresponding to the transmitting node. Therefore, an identifier needs to be added to the transmitted data, raising the total payload of a message to 17 or 18 bytes. However, such an identifier is already included in many protocols (e.g. MAC-address, IP-address, host name).

After all, the tiny amount of data allows for the use of communication technologies that provide high energy efficiency at the cost of low data rates such as IEEE 802.15.4 based protocols which are very commonly used in WSNs with small operating systems such as Contiki [25] and TinyOS[26] that support data collection via multiple hops at a fusion center. An alternative for larger areas is LoRa which is an emerging technology that is particularly well-suited for agricultural applications due to its long range. The $J \cdot M$ sensor readings of one snapshot are collected and processed at the fusion center.

While the ranges may suffice for a single agricultural field, on huge areas, especially for calibrating satellite data, cellular networks might be an

[25] `http://www.contiki-os.org/`
[26] `http://www.tinyos.net/`

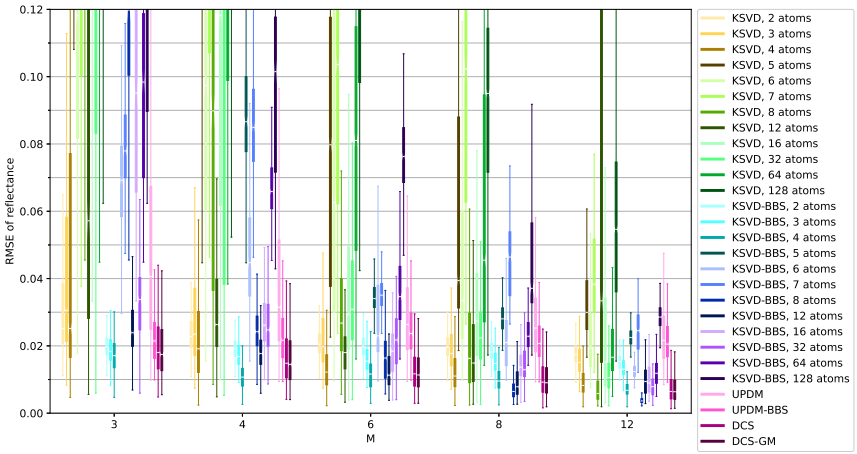


Figure 7.10.: The evaluation results for all approaches on training set *Air_MA*, band selection dataset *Air_MB*, and evaluation dataset *Ground_Full*.

option in areas with sufficient coverage; many carriers already offer specialized plans for machine-to-machine communication which are affordable when sending small amounts of data from many devices.

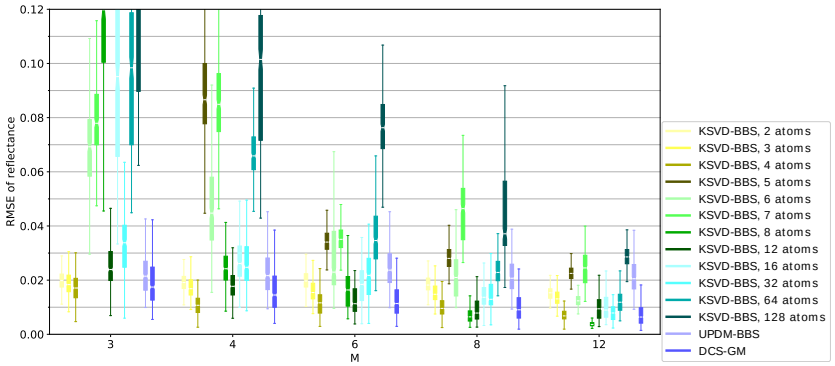
7.3. Trace-based Evaluation of HoM2HSN[27]

In this section, we compare the performance of the HoM2HSN based on K-SVD against the other approaches in order to find out which is the best choice under what circumstances. Other than the preliminary evaluations in section 6.4, here, training is always performed on a dataset using a different sensor than the dataset used for evaluation in order to better reflect the real-world situation. The selection of best bands is also performed on one of the sub-datasets using the same sensor as the training sub-dataset because the selection of best bands belongs to the training phase.

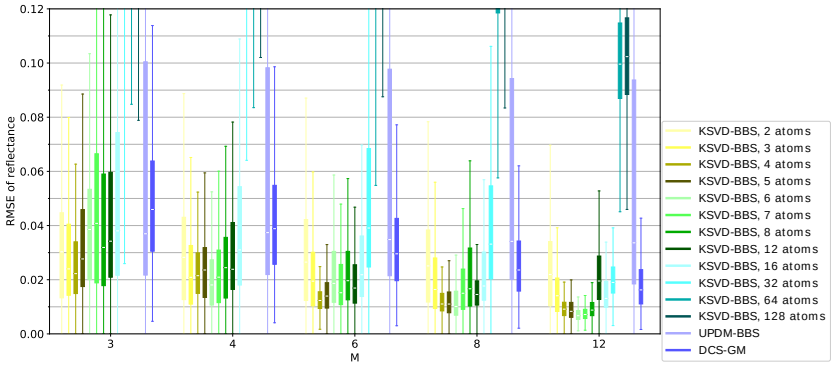
In each dataset combination, we compare the resulting RMSE for all pixels in the dataset with 20 replications for band selection and, in the case of DCS, for the groups of pixels evaluated together. For K-SVD we keep varying the dictionary size in order to further investigate which size is appropriate.

The results are shown in figures 7.10, 7.11, and 7.12 for a varying number of bands M as box plots. Note, that we refrained from including outliers

[27]The content of section 7.3 has been previously published (Hänel et al. 2021b).



(a) Training: *Air_MA*, Band Selection: *Air_MB*, Evaluation: *Ground_Full*



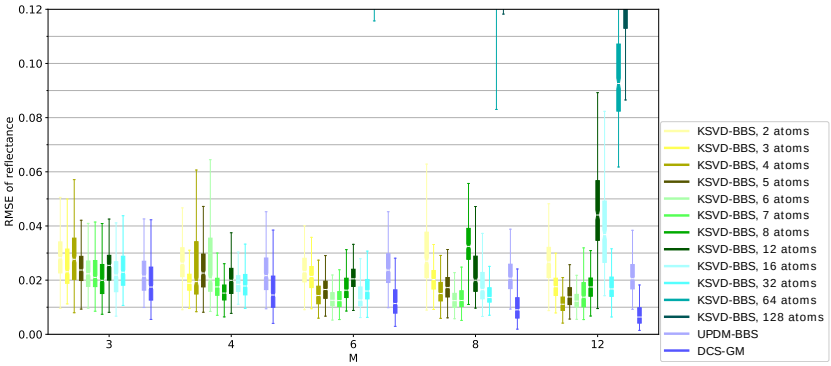
(b) Training: *Ground_V1*, Band Selection: *Ground_V2*, Evaluation: *Air_Full*

Figure 7.11.: The evaluation results for combinations of mixed datasets and datasets containing only vegetation.

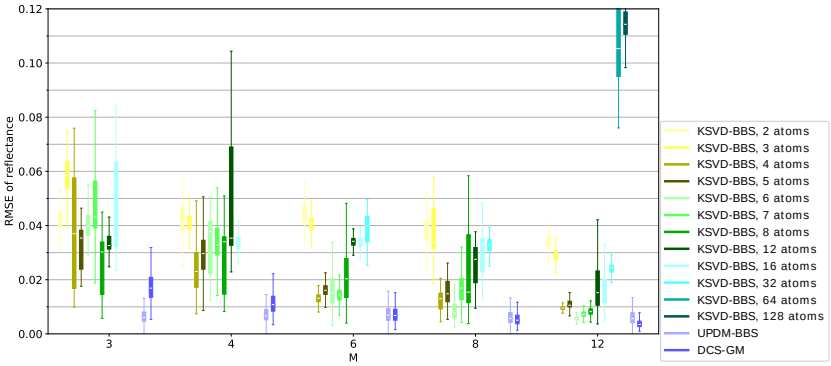
in the graphics as they were highly distracting due to the sheer number as a result of the huge sample size. For some parameters, the boxes lie partially or completely outside the plotting range.

7.3.1. Quantitative Evaluation on Ground-based Data

The first setting shown in figure 7.10 is the main use case followed in the thesis: training on a remote sensing image with diverse environment and using it on measurements from the ground. We compared a total of six different approaches: *K-SVD*, *KSVD-BBS*, *UPDM*, *UPDM-BBS*, *DCS*, and *DCS-GM*. *KSVD-BBS* refers to K-SVD including best band selection, herein, the three best band sets according to the band selection sub-dataset are kept. *UPDM-BBS* refers to UPDM including best band



(a) Training: *Air_V1*, Band Selection: *Air_V2*, Evaluation: *Ground_Full*

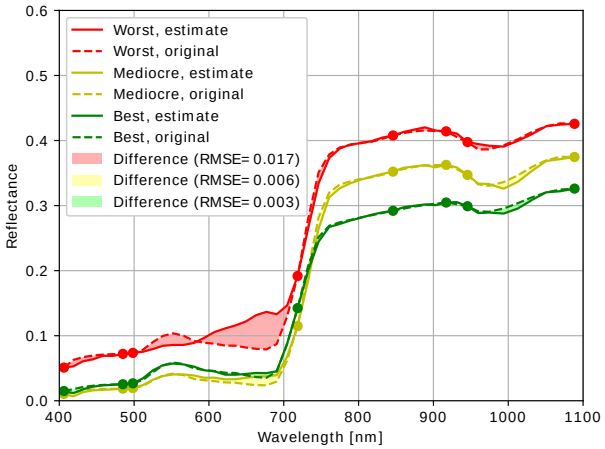


(b) Training: *Ground_V1*, Band Selection: *Ground_V2*, Evaluation: *Air_V1*

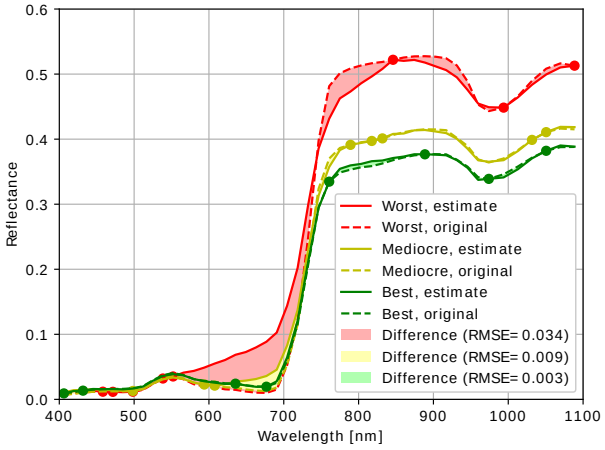
Figure 7.12.: The evaluation results for datasets containing only vegetation.

selection, again, the three best band sets are kept. *DCS-GM* refers to DCS with mixing of the groups by calculating the median of all spectra calculated for one pixel. As the 20 replications are generated by combining 5 group selections with 4 band set selections, 4 median spectra of 5 spectra each are constructed per parameter set and pixel.

K-SVD and *KSVD-BBS* were performed with different number of atoms as shown in figure 7.10. The main observations here are that an increasing number of bands, naturally, leads to reduced RMSE for all approaches. The selection of best bands in *KSVD-BBS* leads to a significant improvement of the *K-SVD* results. Besides lowering the median, the spread of values is also much lower. The selection of best bands in *UPDM-BBS* leads to a decent improvement, especially at small numbers of bands. The



(a) *KSVD-BBS*



(b) *DCS-GM*

Figure 7.13.: Selected reconstructed spectra from *Ground_Full*.

group mixing in *DCS-GM* also leads to a slight but reliable reduction of the RMSE in comparison to pure *DCS*, proving the benefit of this improvement. These improvements were very similar for all dataset combinations. Therefore, we refrained from including the un-improved versions in figures 7.11 and 7.12 to make it more comprehensible and ease the comparison of the dataset combinations.

The first plot 7.11a shown in figure 7.11 is the same as in figure 7.10 in a more compact version, included here for easing the comparison. In this dataset combination, *DCS-GM* reliably delivers good results and outperforms *UPDM-BBS*. *KSVD-BBS* beats *DCS-GM* at certain dictionary

sizes. Interestingly, for less than 6 bands, it performs best with approximately 4 atoms, while for more than 6 bands it performs best with 8 atoms. For all numbers of bands, *KSVD-BBS* is able to outperform *DCS-GM* at the best fitting number of atoms. However, even when selecting the number of atoms only slightly wrong, *DCS-GM* tends to be better.

7.3.2. Qualitative Evaluation on Ground-based Data

In order to provide a better understanding of the results, we picked the spectra with lowest, highest, and median RMSE from one of the simulations with 8 bands and, in the case of *KSVD-BBS*, 8 atoms, for both *KSVD-BBS* and *DCS-GM* shown in figure 7.13. In comparison to figure 7.5, the three plots are merged here. The results slightly vary for *DCS(-GM)* due to differences in preprocessing, leading to different band selections and due to using the group mixing modification. For *KSVD-BBS*, the band set which performed best in the band selection dataset (*Air_MB*) was chosen. For *DCS-GM*, the band sets were picked randomly as no such indicator is available. In both approaches for both the best case as well as the median, the differences of the estimate in comparison to the original spectra are very low, confirming the quantitative findings qualitatively. This allows an estimate of the impact on vegetation indices: these are usually built by comparing the reflectance at different wavelengths. For a vegetation index, that requires the reflectances at wavelengths which have not been directly measured, the reconstruction results clearly deliver better reflectance values for these wavelengths than simply using the closest measured values or an interpolation between the closest measured values. Hence, the result of the vegetation index will also be improved.

For the cases considered here, the errors with *KSVD-BBS* and *DCS-GM* are qualitatively similar, not allowing the derivation of a general statement on the cause of the difference between the two approaches. However, the differences are much clearer in the worst case spectrum. In both approaches, errors in the worst case spectrum become worst in spectral ranges with few bands. In the case of *DCS-GM* (Fig. 7.13b), these ranges are wider due to the many different band sets. One of the particularly uneven band distributions generates the worst case seen here with 5 bands clustered in the range between 450 nm and 570 nm. In the case of *KSVD-BBS* (Fig. 7.13a), the band selection is not as uneven because such band sets get rejected. While the relatively large gap with no bands between 500 nm between 700 nm causes the problems in the worst case, it cannot be identified as a fundamental problem because it only affects the worst cases. In the median and best case, this gap has little to no impact on the reconstruction quality. Note, that the three representatives in both figures 7.13a and 7.13b were selected from the same set of pixels. Therefore, the reflectance being overall higher in 7.13b is completely coincidental.

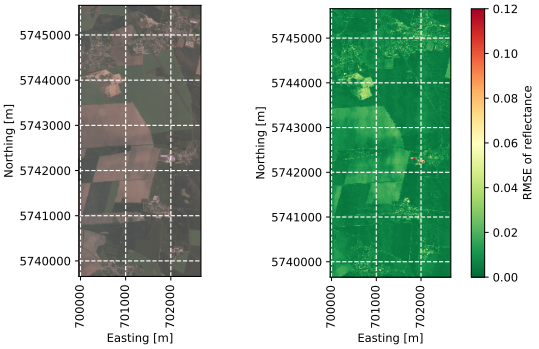
7.3.3. Quantitative Evaluation on Air-based Data

In figures 7.12a and 7.12b, we investigated the performance when training and evaluating on datasets containing solely vegetation data. In figure 7.12a, training was done on the remote sensing dataset and evaluation on the ground dataset. In figure 7.12b, it was the other way around. In both cases, *DCS-GM* is ahead as it performs particularly well with homogeneous data. Interestingly, in figure 7.12a, *KSVD-BBS* is superior over *UPDM-BBS* despite suffering from the same restriction with both the training for *KSVD-BBS* and the base spectra for *UPDM-BBS* being based on the other dataset. In figure 7.12b, *UPDM-BBS* performs almost as good as *DCS-GM* for high band numbers and superior for small band numbers as mentioned in section 7.2.2 which may be explained by very similar vegetation spectra in dataset *Air_V1* that are also very similar to the base spectra in use. In figure 7.12b, *KSVD-BBS* performs slightly better than in 7.12a at least for higher number of bands and optimal number of atoms. Together with figure 7.11a this shows a trend of *KSVD-BBS* benefiting from more diversified training data. However, in figure 7.12b, *DCS-GM* and *UPDM-BBS* benefit more from the similar data, rendering *KSVD-BBS* inferior in this case. The similarity also leads to very low variation of the results with all approaches.

Figure 7.11b is included mainly for completion of the dataset combinations. Its sense is limited as the learning dataset is far less diverse than the evaluation dataset. However, *KSVD-BBS* still works surprisingly well in this case in comparison to *DCS-GM* and *UPDM-BBS* which both fall behind in this case because of lower similarity between pixels in the case of *DCS-GM* and less vegetation pixels in the case of *UPDM-BBS*. With *Air_Full* being a typical remote sensing dataset, this shows that *KSVD-BBS* may also be a promising approach for estimating hyperspectral remote sensing images from based on multispectral remote sensing images aside from its application in M2HSNs.

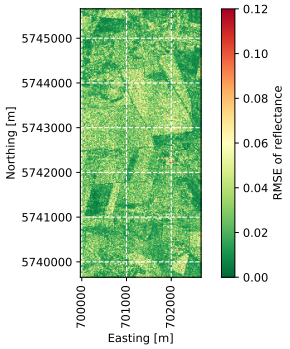
7.3.4. Qualitative Evaluation on Air-based Data

For a qualitative evaluation of this case, we assess the reconstruction quality across the area in figure 7.14. Figure 7.14a is an RGB image including the corresponding bands from the original dataset. Figure 7.14b and 7.14c show the RMSE per pixel for one of the best band sets according to training in *KSVD-BBS* and a group mixing selected randomly for *DCS-GM*. In figure 7.14b, *KSVD-BBS* generates low RMSE values especially in vegetation areas. Curiously, the trained base is also suitable for the bare soil areas although no such samples were included in the training base. The quality suffers in the villages. If planning to use the approach on remote sensing data, clearly a more diversified training dataset is required. Figure 7.14c shows why the RMSE values are higher with *DCS-GM*: the quality



(a) Map with normal coloring.

(b) Resulting RMSE with *KSVD-BBS*.



(c) Resulting RMSE with *DCS-GM*.

Figure 7.14.: Investigation of reconstruction quality on the area of *Air_Full*. The coordinates are in UTM zone 32U.

is only about as good as with *KSVD-BBS* in some of the vegetation areas but the main effect affecting the RMSE is the extreme variation across the whole area. We attribute this to mixture of two effects: the first one is the one observed in figure 7.13b, that in some pixels an uneven bands selection leads to bad reconstructions. The second effect is that the groupings often include spectra of differing kinds reducing the overall reconstruction quality of the group. The latter effect also serves as the main explanation for why the reconstruction quality is significantly better in all the other scenarios with less variation across the spectra.

7.3.5. Conclusion

Overall, we have found that the HoM2HSN due to the option of preselecting bands greatly helps improving the spectral upsampling with K-SVD and UPDM. In large-scale M2HSNs with very similar data, a HeM2HSN with DCS turned out superior, in small-scale M2HSNs with very similar data, a HoM2HSN with UPDM turned out superior. In M2HSNs with variable data, a HoM2HSN with K-SVD turned out superior.

7.4. Real-world Evaluation of the M2HSN

The previous trace-based evaluations in sections 7.2 and 7.3 gave a good impression of the accuracy of the spectral upsampling and the extraction of spatial differences. However, for the evaluation of the differences over time and the measurement quality of low-cost sensors, an M2HSN prototype was deployed on a wheat field as explained in section 4.5. In section 7.4.1, we first investigate the measurement of the low-cost sensors in comparison with other instrument. In section 7.4.2, we take a look at the measurement over time.

7.4.1. Spectra Comparison

For the comparison of the measured spectra, we select a single day. On June 18th in the forenoon, we measured the spectra with multiple instruments: the WSN, the SVC, and the P4. From the WSN, we extracted the values between 8am UTC and 10am UTC and formed median spectra for each position. With the SVC, we took spectra next to the WSN nodes. Note, that the SVC does not have a sky sensor. White reference surface measurements were instead taken shortly before or after the measurements. From the P4 flight an orthomosaic was created with a pixel resolution of 1.8 cm. For squares of 100×100 pixels or $1.78\,\mathrm{m} \times 1.78\,\mathrm{m}$, the median spectra were calculated. Note, that the median in combination with the relatively large area helps ignoring the sensor nodes and their shadows, which are also visible in the orthomosaic.

The resulting spectra are shown in figure 7.15. For P4 and the 7-band sensor nodes, a line plot for the spectra was added that shows the result obtained with K-SVD trained on *Air_M1* for the WSN and trained on *Ground_Full* for the P4, meaning that the training data stems from a different year and location.

The original samples for these spectra are shown as markers. For the color codes, please refer to figure 4.6. Note, that the figure placement corresponds with the sensor node positions on the field. The difference between the spectra from the WSN and the SVC is indicated by colored areas. The 18-band spectrometers (cyan, red, magenta, blue, and green)

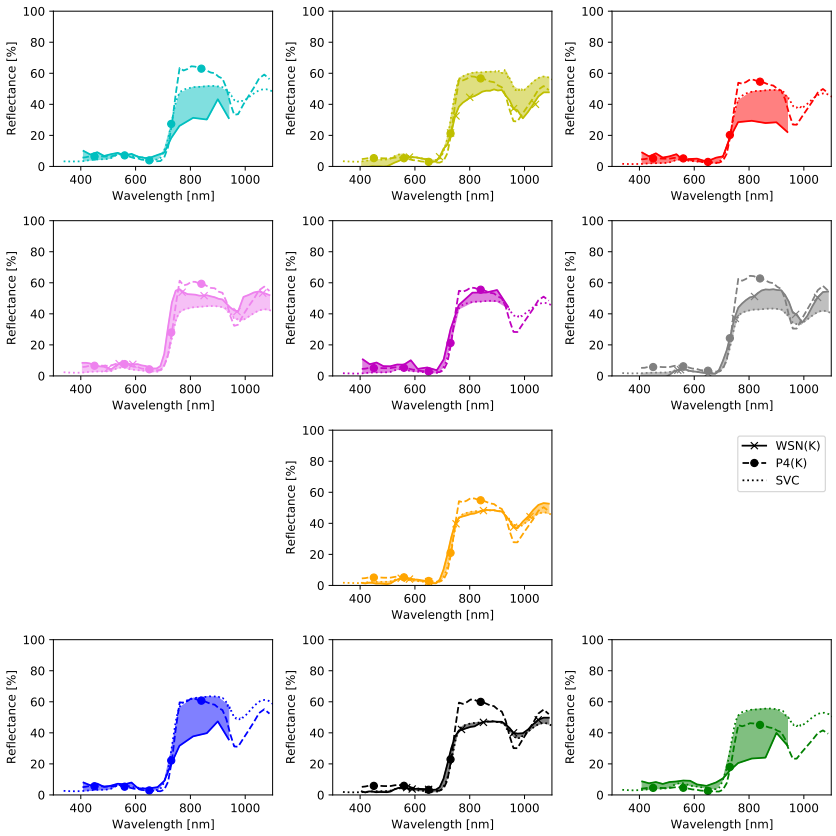


Figure 7.15.: Measured spectra on June 18th with different instruments at all sensor node positions.

often underestimate the reflectance at higher wavelengths. For the 7-band spectrometers (yellow, violet, grey, orange, and black), no obvious pattern may be observed, sometimes the WSN is closest to the SVC, sometimes the WSN to the P4, sometimes the SVC to the P4.

Measurements with the SVC for spots closely together, not shown here, offered an explanation for this: at the used fields of view, this is simply the expected variation.

For a quantitative comparison, the RMSEs are compiled in figure 7.16. Here, we always compare the spectra of two approaches. For the calculation of the RMSE, we selected the closest matching band of the higher resolution spectrum to each band of the lower resolution spectrum. Then, the RMSE was calculated for these pairs. Here, WSNK and P4K again refer to high-resolution spectra calculated from the WSN and P4 spectra.

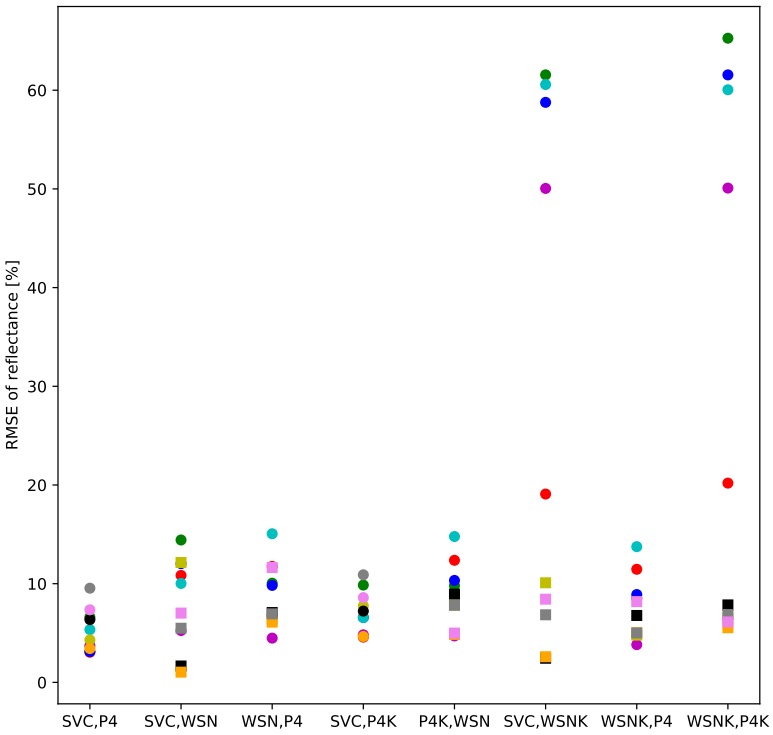


Figure 7.16.: Difference between the spectra acquired with different sensors and processing approaches.

The results show, that SVC and P4 readings agree quite well with RMSEs of 3 to 10 percent points. The average of the similarity of the WSN results and the SVC results is of a similar scale. However, the variation is higher with RMSEs varying between 1 percent point and 15 percent points. Increasing the resolution with K-SVD, increased the RMSE by a few percent points in the case of the P4. This is the expected behavior because perfect reconstructions are not expected. However, it is noteworthy that the additional error is far smaller than the one stemming from the initial samples. For the WSN, K-SVD has a very interesting impact: For the 18-band spectrometers, the spectra have extremely high RMSEs, which stem mainly from the long wavelengths, where no samples have been acquired. In comparison to the P4K, the high number of bands with partially false readings lead to faulty over-fitting. For the 7-band spectrometers, the RMSE remains in the same magnitude, but the spread is reduced. The resulting RMSEs are roughly in the same value range as for the P4 - this is actually a remarkable result, showing that the high resolution spectra are about as similar to the SVC as the low resolution

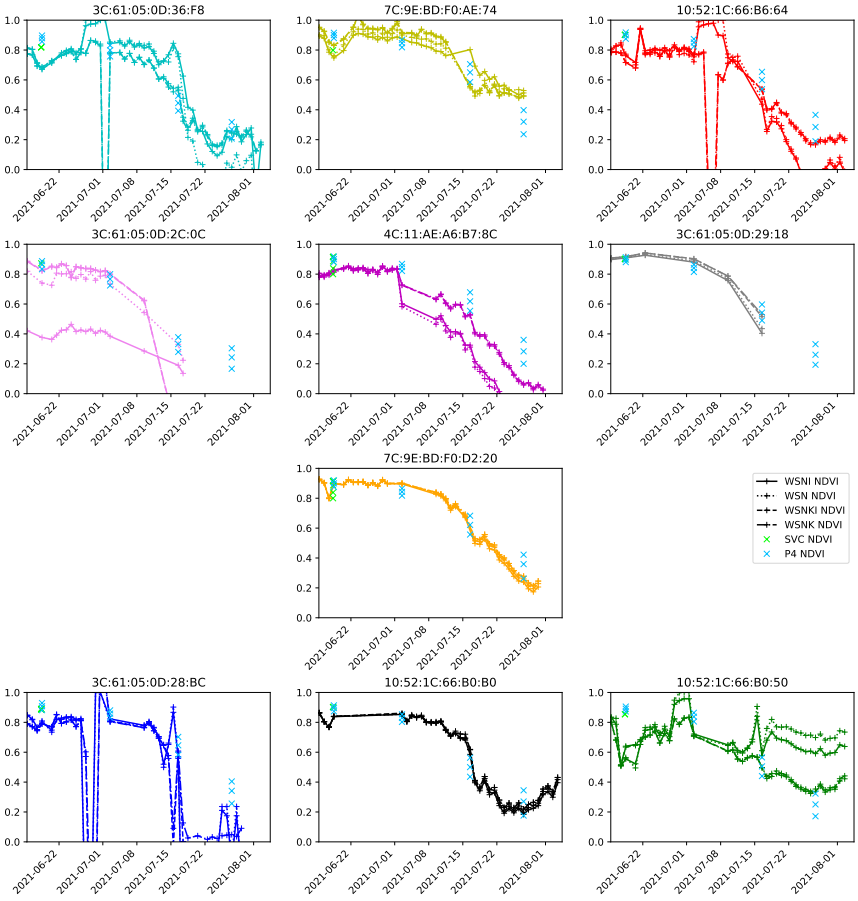


Figure 7.17.: Development of the NDVI during the test deployment.

spectra of the P4. Overall, the main finding here is, that the experiment confirmed the findings from the simulations: increasing the spectrum resolution is working well. The errors introduced by the procedure are small in comparison to the errors of the initial low-resolution measurements. Sometimes, it even helps reducing the errors.

7.4.2. NDVI Development

For the long-term evaluations, the spectra did not prove practical, because comparison spectra were only acquired once and visualizations with many spectra are difficult to analyze. Instead we consider one vegetation index derived from the data. We choose the most commonly used index,

the NDVI. This also gives an impression of the performance of derived parameters. The development of the NDVI at each position is shown in figure 7.17. For June 18th, the SVC measurements are included with one NDVI value per spectrum. For the central position, the values show more variation because the position of the SVC has been slightly varied. For the P4, measurements from multiple dates are included. Per date, the NDVI was calculated for each pixel in the measurement area and three markers indicate the first, second, and third quartiles of these values. For the NDVI calculation, the wavelengths of the Red and NIR P4 bands were used in the SVC and WSN data as well.

For the WSN, as before, the corresponding reflectance values were determined with different approaches: just using the closest available value (WSN), using an interpolation between the two closest values (WSNI), using the closest values in a high-resolution spectrum created via K-SVD (WSNK), and using interpolation between the two closest values of this high-resolution spectrum (WSNKI). Similar to the median spectra, the median NDVI in the 4-hour noon window per day is shown.

For all positions, a decline of the NDVI becomes visible, which is the expected behaviour at this time of the year: the crops are drying and loosing chlorophyll, leading to a yellowish color, and the difference between reflectance in the near infrared and the visible range is reduced. A similar decay from approximately 0.8 to approximately 0.2 was, e.g., observed by Vannoppen et al. (2020) for wheat in Latvia.

The plots contain gaps and anomaly values because of problems such as sensor nodes dropping on the ground after heavy rain, crashes of the sensor nodes computing cores, water getting into the cases, and dirt in the light tubes. Nevertheless, the decline is very similar for the P4 data and the WSN data, especially for WSNK and WSNKI.

For a more quantitative evaluation, a correlation plot of the NDVI is given in figure 7.18. For WNSKI and WSNK, the resulting R^2 values with 0.75 and 0.77 are higher in comparison to 0.58 and 0.61 for WSNI and WSN. This shows, that increasing the spectral resolution with K-SVD can improve the calculation of the NDVI when comparing NDVI values derived from multispectral sensors that sense in non-matching bands.

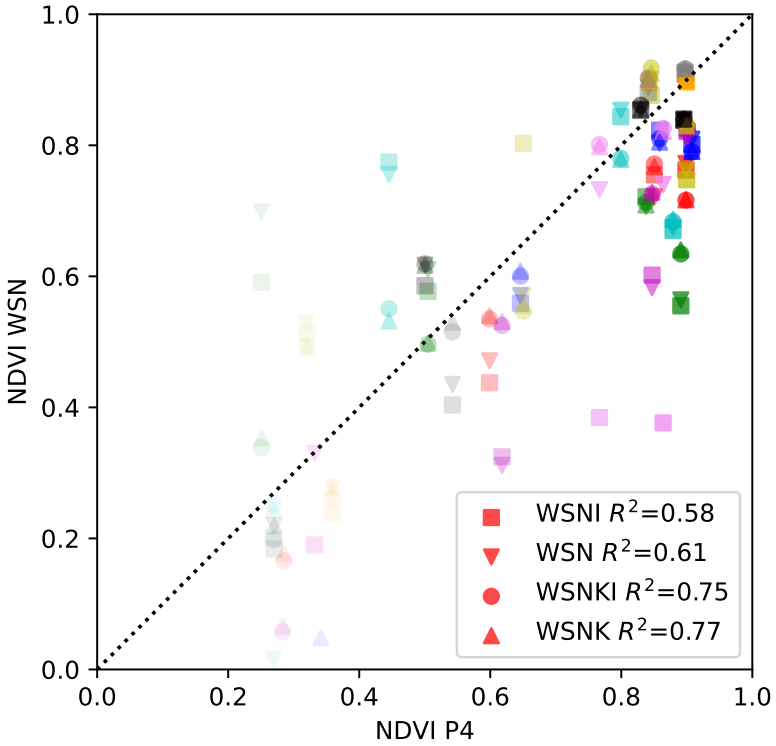


Figure 7.18.: Comparison of P4 and WSN NDVI measurements. Higher transparency indicates samples at later dates. The colors indicate the sensor positions.

7.5. Quality Metrics under Parameter Variation

In this first evaluation of the quality metrics, synthetic data is considered. The main advantage of synthetic data is that parameters can be varied in a controlled manner for researching their influence. Furthermore, replication of results is particularly easy for synthetic data.

Being the most common example, we concentrate on a sine wave:

$$x_n = \hat{x} \cdot \sin(2 \cdot \pi \cdot \nu_{signal} \cdot n + \varphi) + \delta_n \tag{7.1}$$

For the evaluations, while measuring the impact of most variables, some are seen as fix: The length of vector x is $N = 512$. The δ_n values, representing noise, are drawn from a Gaussian distribution with mean of 0 and standard deviation of 0.2. Having just a single frequency, the sparsity is 1.

The number of samples M, the frequency of the sine ν_{signal}, the amplitude $\hat{x}$, and the phase φ are varied. Herein, M and ν_{signal} are not

varied directly. Instead, we vary $\tau = \frac{\nu_{sample}}{\nu_{signal}}$ with the average sampling rate ν_{sample} of the elements in y. This may be easier interpreted in terms of the Nyquist rate: according to pre-CS signal processing theory τ must be greater than 2. This being a CS-based example, we just use values up to 2. Similarly, we introduce $\zeta = \frac{\nu_{original}}{\nu_{signal}}$ which allows the same interpretability for the reconstructed signal and must actually be greater than 2 to make the signal usable in further applications. Note, that $\nu_{original}$, the sampling rate of x, is implicitly fixed to 1 in equation (7.1). From these values, ν_{signal} can be obtained as $\nu_{signal} = \frac{\nu_{original}}{\zeta}$ and M as $M = \lfloor N \cdot \frac{\nu_{sample}}{\nu_{original}} \rfloor = \lfloor N \cdot \frac{\tau}{\zeta} \rfloor$

Ψ is a DCT transform matrix - although DFT may deliver superior results - because we want to repeat the evaluation with various solvers and not all of the solvers support complex numbers. Φ is a sparse sampling matrix with M ones, because those offer the most intriguing applications and allow more quality metrics.

7.5.1. Varying all parameters at once

For the following evaluations, we build J=100 signals x while varying the four variables (τ, ζ, $\hat{x}$, and φ) and the noise. The parameters are chosen in the following intervals: $\tau = [0.5, 2.0]$, $\zeta = [4.0, 16.0]$, $\hat{x} = [0.5, 2.0]$, and $\varphi = [0, \pi]$. We then draw M random samples from each of the 100 signals x, creating 100 compressed vectors y and build matching Φ matrices. For each of the 100 compressed vectors, we reconstruct the signal x_{est} from y, Ψ, and Φ, using a suitable solver, for now, BCS. x_{est} is modified by re-inserting the measurements from y. For most solvers, this is not even necessary because the values match. However, we want to avoid these samples having an influence on the result, because this simple improvement of the reconstruction quality is always possible with the simple sub-sampling Φ. From Ψ, Φ, x_{est}, y, and in the case of BCS, the covariance matrix, we then calculate all the quality metrics. We also calculate the actual reconstruction error from x and x_{est}. Note, that we compare against the noisy original vector, not a noise free original vector. This better reflects the approach of handling real world data, where it's usually impossible to fully remove noise from the data. Naturally, higher amplitudes correspond with high RMSEs which would lead to high correlation between the norms and the RMSEs. In order to compensate for this, we use the NSR as error metric for the PSMs.

For the ISMs, we just use the absolute error because none of the ISMs is norm-like. We then calculate R^2, GP_+, GP_-, and the corresponding p-values for each of the quality metrics. For the ISMs, pairs are built for combinations within a signal, but all pairs are merged to calculate GP_+, and GP_-. R^2 is calculated per signal and the median of the resulting values is shown. The only one of the quality metrics requiring external

parameters is *l0norm_psm* with the threshold for determining zero values set to 1. The resulting correlation plots for each of the quality metrics are shown in figure 7.19. The corresponding values may be found in the appendix in table A.1. Firstly, the norms that are not promoting sparsity in the sparse domain, i.e., *l2norm_psm* and the norms for the normal domain, show a clear correlation with high norms coinciding with low NSRs and vice versa. This can be explained with BCS showing high NSRs when being too careful, choosing small values overall. Taking the other norms into consideration, this group of signals with high NSRs seems relatively dense as indicated by the high *l05norm_psm* and *l025norm_psm*. Apparently, BCS tends to slightly increase all elements instead of going for a sparse result in these cases. This is not reflected in the *l0norm_psm* because too many of the values still fall below the threshold. However, in the group with small NSRs, no clear correlation with the sparsity promoting norms is visible. Overall, the correlation is too small for the sparsity promoting norms being usable as quality metrics as originally intended. The same groups are also identifiable in *variance_psm*, displaying high values at high NSR and 0 at low NSR. The other metric from related work, *crossvall1o_psm* works similarly well. The parameters evaluating the sample distribution (*biggestgap_psm* and *gapdiff_psm*) except for *emptybinsfraction_psm* also show similarly good performance, but the more trivial *compressionratio_psm* works even better.

coherence_psm and *rmsme_psm* are not usable and, therefore, excluded in the plots. For the combination of DCT and samples, Ψ includes a high number of high values in Ψ due to the sine shape: even for small sample sizes, the calculated coherence will equal the actual coherence and, therefore, is equal for all Φ, rendering the *coherence_psm* useless. *rmsme_psm* is useless for BCS and SL0 because they match the measured samples perfectly - in this example, it is also not usable for LASSO, because the measured values are re-inserted in the solution.

With the exclusion of the original samples in the evaluation of the ISMs, neither *variance_ism* nor *samdist_ism* showed significant correlation with the error per sample.

7.5.2. Influence of Parameters and Solvers

In the next step, we want to investigate the influence of the parameters more closely. As this would require several hundred correlation plots as those in figure 7.19, we choose a more compact presentation here, albeit losing some information. To explain the figures 7.21-7.23, we first take a step back. We fix all parameters and just vary noise and sample selection, again 100 times. We repeat this for 81 different parameter combinations, resulting from three different values for each of the four parameters (τ, ζ, $\hat{x}$, and φ), equally spaced in the same intervals as before. Figure 7.20 shows the $GP\_$ value for each of the 81 parameter combinations for metric

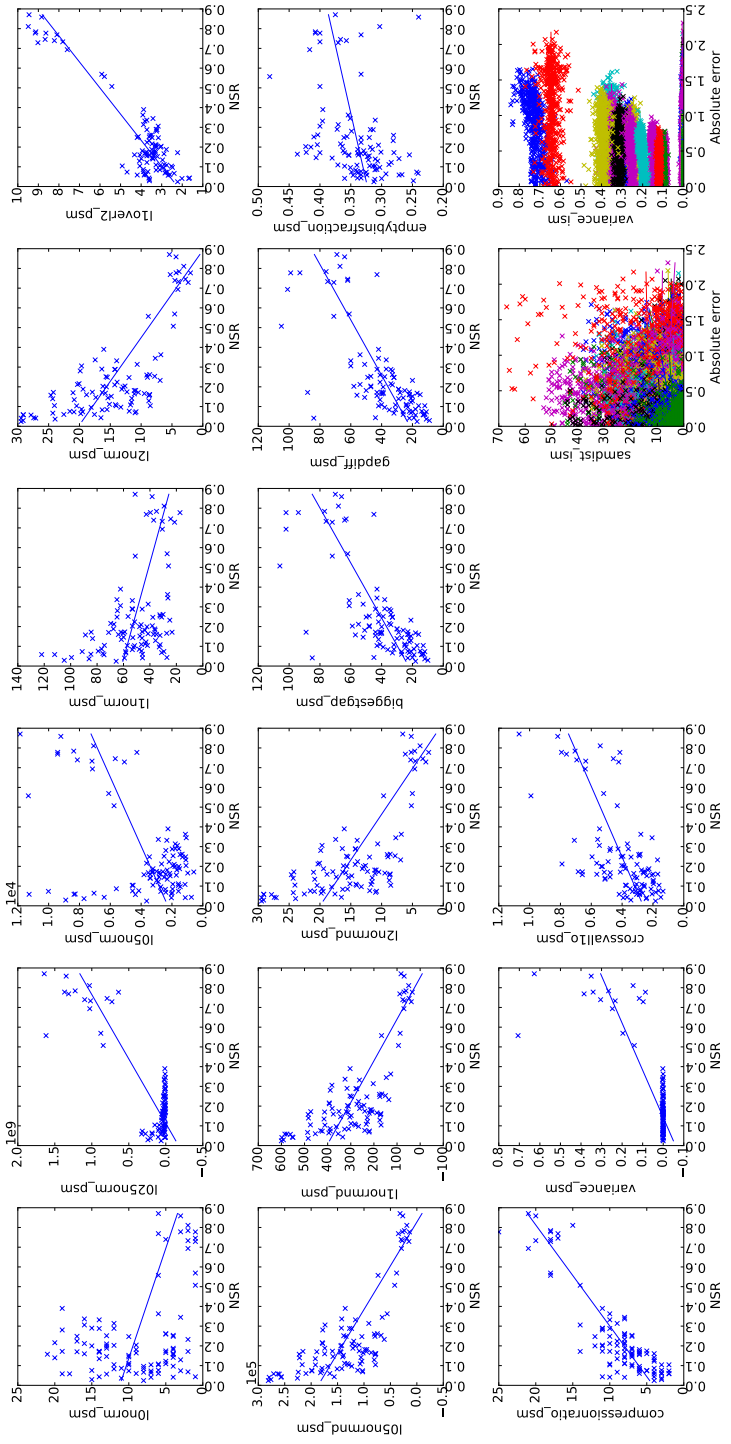


Figure 7.19.: Resulting correlation plots for BCS on synthetic data with all parameters varied.

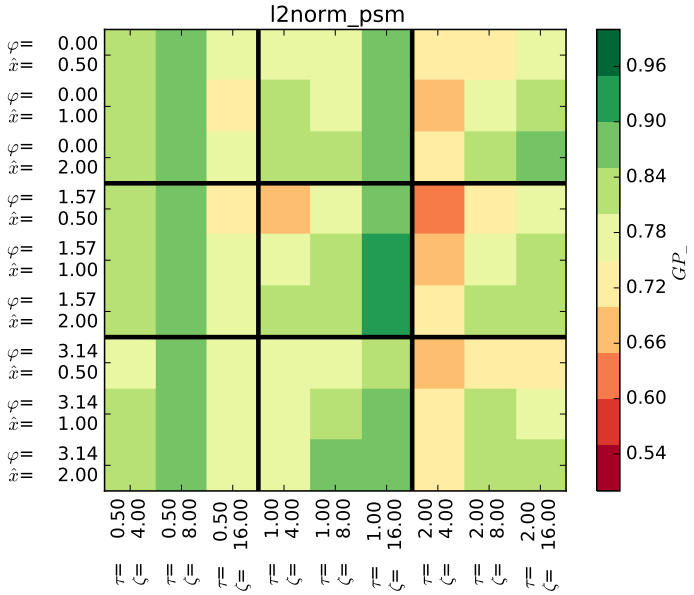


Figure 7.20.: Resulting GP_{-} values for metric *l2norm_psm* in 81 different parameter sets for a sine wave, solved with BCS.

l2norm_psm as a heatmap. It shows that the observation from figure 7.19, that high *l2norm_psm* values correspond with small errors and vice versa, does not only hold true for reconstruction issues caused by variation of parameters but also for reconstruction issues caused by noise and sample selection for all points in the parameter space. This is indicated by the GP_{-} values being significantly higher than 0.5 in all cells of the heatmap.

Now we use the same visualization principle, albeit omitting the labels in order to generate a more compact visualization, for all quality metrics in the first row ("none") of figure 7.21. We now make just a single parameter variable, again in the same interval as before. This results in similar grids, albeit with just 27 values, shown in the central 4 rows in figure 7.21. The varied parameter is indicated on the left. The last row ("all") shows the result from figure 7.19 in the compact format, consisting only of a single value because all parameters are variable. Some of the grids show the GP_{+} value while some show the GP_{-}, this is indicated by the $+$ or $-$ in the center of each parameter grid. It is always the value, that has the higher median for the whole grid. This step was necessary because the directions in which the metrics work invert under some circumstances.

Taking a look at all the metrics, the performance is quite similar when not varying the parameters and when varying solely the phase, this is the expected behavior because changing the phase means just shifting

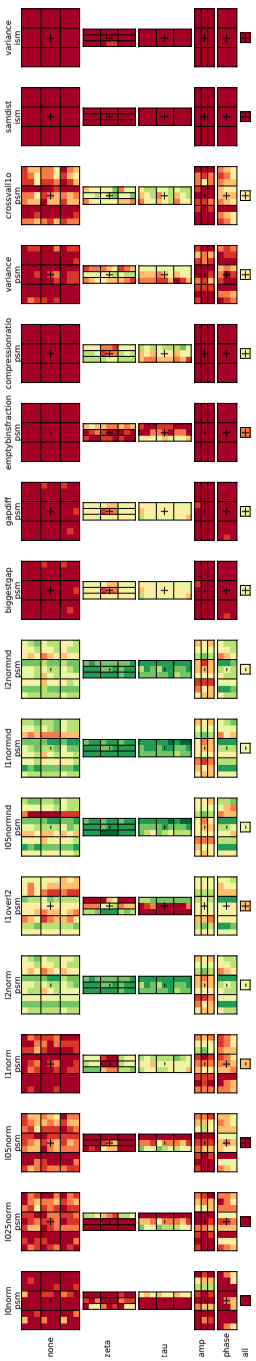


Figure 7.21.: Resulting good pairs values for all quality metrics when varying different parameters of the synthetic data for different solver BCS (axes labels may be found in figure 7.20).

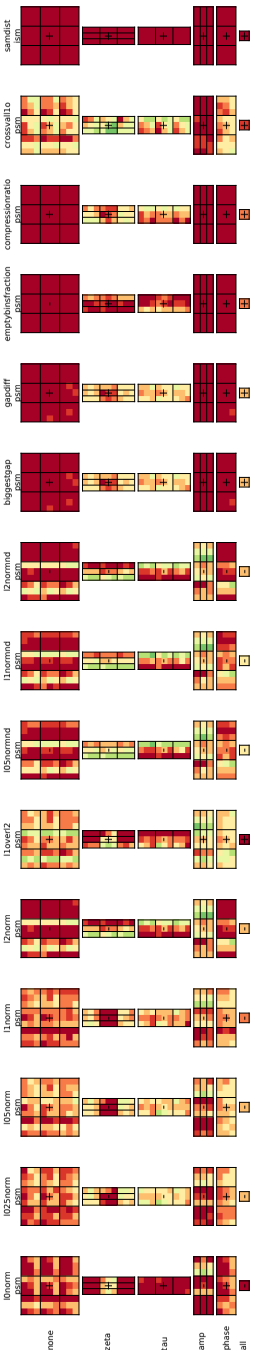


Figure 7.22.: Resulting good pairs values for all quality metrics when varying different parameters of the synthetic data for different solver SL0 (axes labels may be found in figure 7.20).

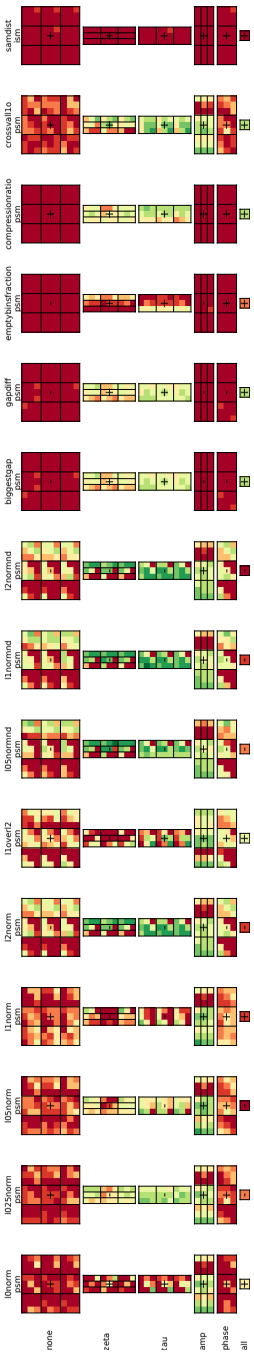


Figure 7.23.: Resulting good pairs values for all quality metrics when varying different parameters of the synthetic data for different solver LASSO (axes labels may be found in figure 7.20).

the signal. Taking a look at the norms that do not promote sparsity, it again becomes clear that by choosing the NSR over the RMSE, the extremely good performance under amplitude variation was successfully avoided: they even perform worse for amplitude variation than in the other cases. This again reassures that the main effect visible here is indeed the underestimation made by BCS. When the amplitudes change as well, this effect becomes more difficult to differentiate, hence the decreased performance under amplitude variation. However, even in this case, the norms that do not promote sparsity perform better than the metric from related work, *crossvall1o_psm* despite the calculation of the norms being significantly more simple and fast.

In the sparsity promoting norms, especially in *l1norm_psm*, similar effects to those of the non sparsity promoting metrics are visible, merely less pronounced, overshadowing the sparsity detection. Only in the cases with no variation and solely phase variation, the sparsity grading effect becomes visible as indicated by the GP_+ instead of the GP_- being superior, albeit only very lightly. The sampling pattern based metrics all are just usable when τ and ζ are varied - this shows, that they just help by grading the overall sampling density and not the inequalities, rendering *compressionratio_psm* a better metric for this purpose. The ISMs turn out unsuccessful in all cases. *variance_psm* has a similar performance to the metrics evaluating the sampling pattern but it shows slighly better performance with varying amplitude.

We now take a look at the other solvers. For SL0, shown in figure 7.22, the differences in performance are similar. The slight success indicated by the GP_- metric for the non-sparsity promoting metrics stems again from the solver tending to choose smaller values when delivering worse results. BCS and SL0 behave quite similar in that regard. However, the GP_- value is significantly lower for SL0 and less consistent across all cases. We attribute this to an overall better performance of SL0, making the cases of underestimation less likely to occur. For the metrics rating the sampling pattern, the results are also similar to BCS but overall, the GP_+ is a bit lower which we again attribute to less failures overall.

For LASSO, shown in figure 7.23, the situation changes more drastically. When varying everything, the norms generally perform significantly worse, in both terms of GP_+ and GP_-. Taking a look at the other rows, the difference stems from the amplitude variation where GP_+ is dominating in contrast to all other solvers and the τ and ζ variation. The inversion of the metric cancels out the successes. The explanation for this may be found in the non-variation cases. There, the norms show lower GP_- values for high amplitudes which is caused by a different type of failure where not only an underestimation happens but rather a complete failure with false frequencies. LASSO's sensitivity towards the amplitude can be explained by the non-matching factor in LASSO weighing the ℓ_1 versus ℓ_2. Note, that this becomes more striking towards lower τ and higher ζ which is the

situation where CS becomes more likely to fail. The sampling rate based metrics perform better for LASSO which we attribute to a generally lower performance of LASSO.

Overall, we have made the following observations: Using gaps between samples for the determination of reconstruction quality fails, it is only suitable for telling that the data is better at the samples. Otherwise, the only information on the quality shown by the metrics derived from the sampling pattern is simply the compression ratio which can be used more easily directly. The sparsity of the result is only a very weak indicator for the quality, it only becomes slightly usable when it is not overshadowed by other effects.

Surprisingly good indicators are the non-sparsity promoting norms. However, they tend to change direction in which they work because they work indirectly by helping to identify failure cases of the solvers. A general trend seen from comparing between the solvers is that the estimate of the quality of the reconstruction becomes better when the reconstruction itself becomes worse.

7.6. Quality Estimate for CFE

In this section, we consider a real world dataset containing yield data from a harvester, evaluated in section 7.1. We first consider the iterative case in section 7.6.1 and then the static case from section 6.2 in section 7.6.2.

7.6.1. Iterative Case

As a short recap, samples are recorded periodically and whenever the data rate limitation allows a new transmission, a sample for transmission gets selected at random from all the samples which have been recorded since the last transmission. The scalar field of yield is reconstructed from these samples, assuming two-dimensional DFT or DCT, modeled with KCS. The resulting error over the number of collected samples is shown in figure 7.24.

In addition to the evaluation in section 7.1, the results with BCS and LASSO are included. However, they are outperformed by SL0. BCS fails altogether just as in the static case in section 6.2.2. The ℓ_1 norm is shown as an example metric. Its behavior differs from the initial expectation. Instead of decreasing over time like the RMSE, it actually increases with a similar behavior as the RMSE, only mirrored. However, it is not sufficiently accurate to correctly reflect the differences between the solvers.

The corresponding correlation plots may be found in figure 7.25 for SL0. These plots slightly differ from the previous correlation plots as values obtained from different seeds per value of M are included grouped by color. The regressions have been performed per value of M as well. The

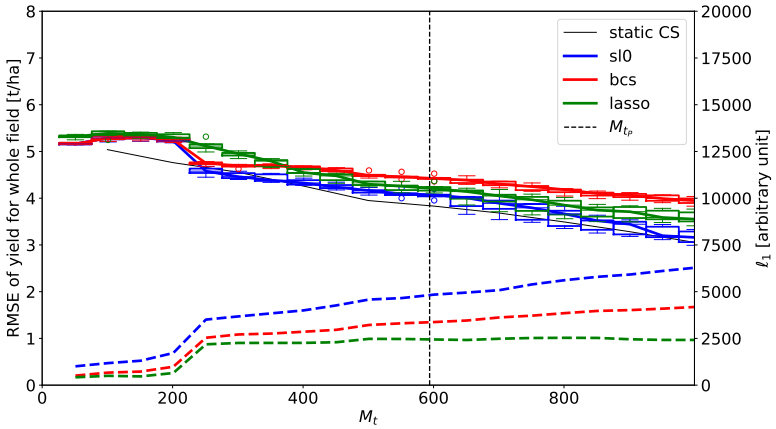


Figure 7.24.: RMSE development (lines) and ℓ_1 development (dashed) of iteratively collected samples in CFE dataset.

overall correlation is even stronger as indicated by the data being close to a diagonal line for most metrics. The dominating effect here seems to be: the more data available, the better the reconstruction. This is achieved by having more structure in the data which also leads to increasing values of the norms, causing the reverse dependency. This effect dominates in the iterative CFE data. We attribute this to the CFE data being less sparse altogether. This is also reflected by the relatively low achievable compression ratio and the performance of interpolation techniques being only slightly worse. As already observed for the synthetic dataset, under M variation, the compression ratio already serves as a very good indicator, not justifying the overhead for the other metrics. These findings also hold true for LASSO, the correlation may be found in the appendix in figure A.1.

7.6.2. Static Case

In order to treat the case where the compression ratio cannot be used, we took a closer look at the behavior with fixed M. Therefore, we switch to the non-iterative or static version evaluated in section 6.2 which is relevant when taking samples manually on the field.

In the following evaluations, we excluded the original samples because estimating the perfect accuracy in these cases is trivial. The resulting GP values for the ISMs and PSMs are shown in figures 7.26 for different values of M, different solvers, and in the case of SL0, both DCT and DFT. The selection of samples has been performed 50 times. Independent of the solver and the transform matrix, the norms show a solid correlation,

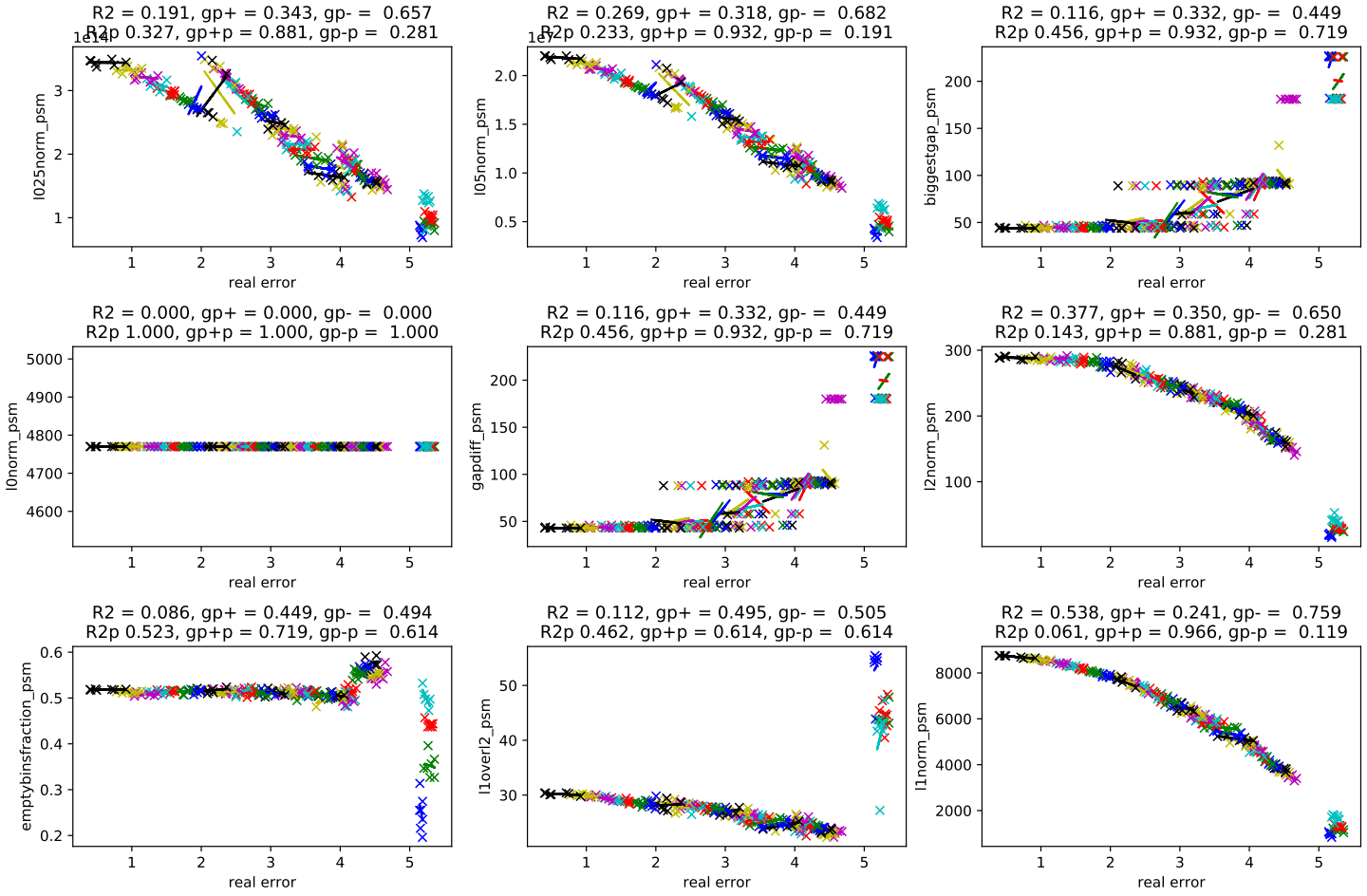


Figure 7.25.: Correlation plots for PSM metrics on iterative CFE data using solver SL0.

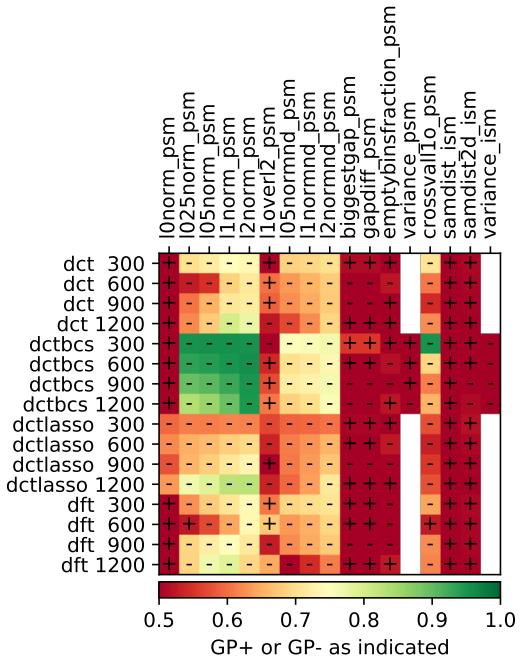


Figure 7.26.: Resulting GP value for all quality metrics.

except for ℓ_0, while the gap-based metrics fail. Neither the distance-based ISMs nor the variance show significant correlation with the reconstruction quality.

For the norms, the inverse correlation is dominant, as indicated by GP_- being higher than GP_+. This is consistent with the relation observed for the iterative case. Note, that all norms in both normal and frequency space show good correlations. The higher degree norms perform slightly better.

For a more in-depth investigation of the cause, we take a closer look at some example results. This is done for $M = 300$ because the differences are more obvious in that case. For the investigation, we just consider ℓ_1 with DFT and SL0. The correlation plot is shown in figure 7.27. We select four samples as indicated in the figure. The corresponding reconstructions and reconstruction errors are shown in figure 7.28. The selection encompasses two samples where the ℓ_1 serves as a good estimator - one with small error (green) and one with significant error (blue). The other two samples are a significant underestimation of the error (red) and a significant overestimation of the error (yellow). The green and the blue sample show the behavior described above: higher ℓ_1 indicates that the solver was braver which led to a better reconstruction. The red sample shows one of the expected cases where structure was introduced in the recon-

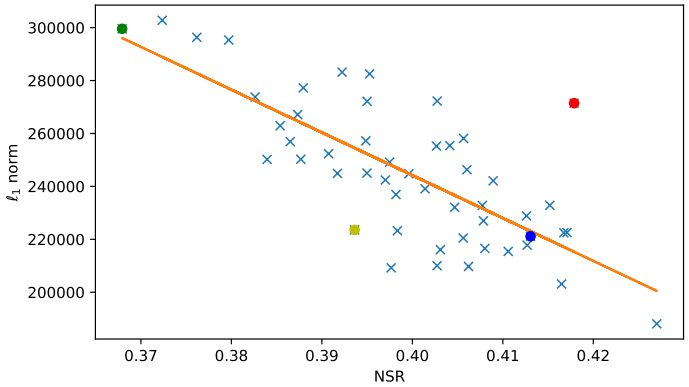


Figure 7.27.: Correlation plot of ℓ_1 with highlighted examples for qualitative investigation.

struction but the structure is simply wrong. The yellow sample is more curious: ℓ_1 is low, the reconstruction still shows a lot of structures and the reconstruction is actually quite good. Here, we are actually closer to the originally expected behavior where the solver succeeds by generating a sparse reconstruction. However, the other effect is dominating, which also serves as an explanation for the other norms working as well - they also increase with more structure.

The performance of the cross validation is similar to the one of the norms. Interestingly, it works the wrong way, as indicated by the GP_- being higher than the GP_+. This holds true not only for the NSR shown here but also for the RMSE, not shown here. Instead of working the way intended, the correlation merely stems from working like a norm.

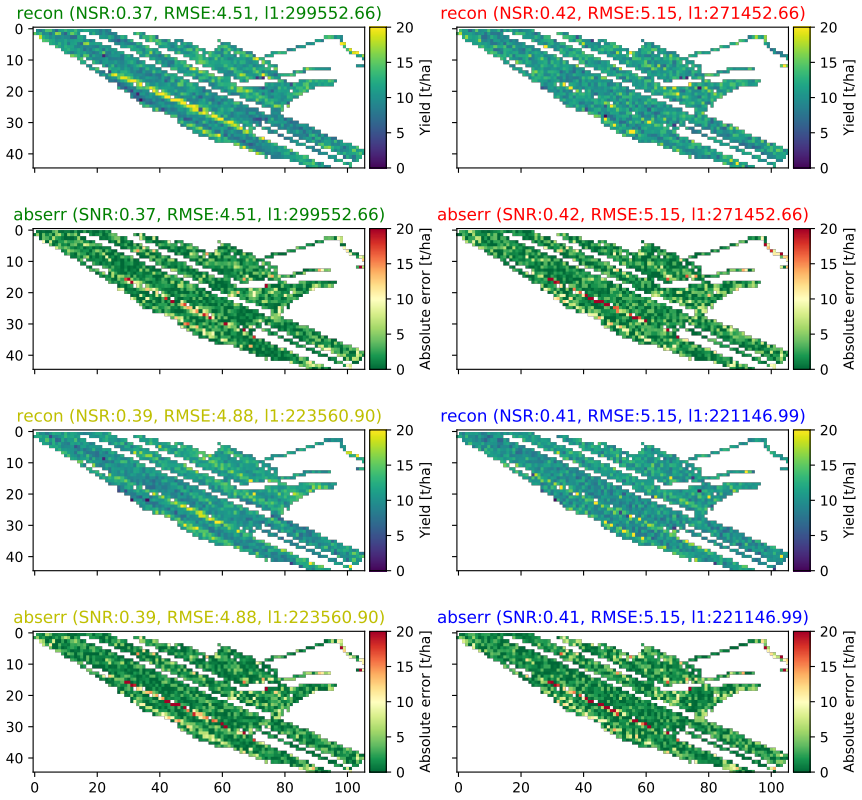


Figure 7.28.: Reconstruction result and reconstruction error across the field for the seeds selected in figure 7.27.

7.7. Quality Estimate for HeM2HSN

As a second real-world data set, we use plant spectra acquired on the ground for further evaluating the quality metrics for CS and DCS. The RMSE evaluations for this CS application were given in section 7.3. The band preprocessing of the spectra is the same as in section 6.4.1, i.e., dataset *Ground_Full*, not the preprocessing from 6.3.1, which leads to tiny differences in the evaluation results.

Figure 7.29 shows three exemplary spectra as reconstruction and original. The three examples are the spectrum with the minimal error, the one with the median error, and the one with maximal error, just like in figure 7.5. The NSR and an exemplary PSM are given in the top for each spectrum. These are the results acquired with SL0. For these three sample spectra, the ranking is perfectly reflected by the *angulardiff2common_psm*.

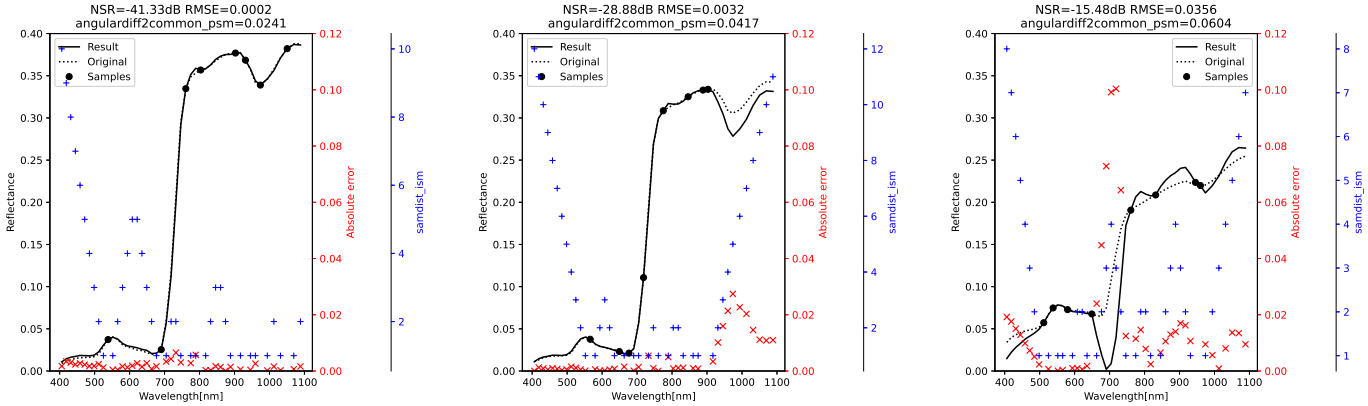


Figure 7.29.: Best, median, and worst case reconstruction of ground spectra according to reconstruction error with exemplary error metrics.

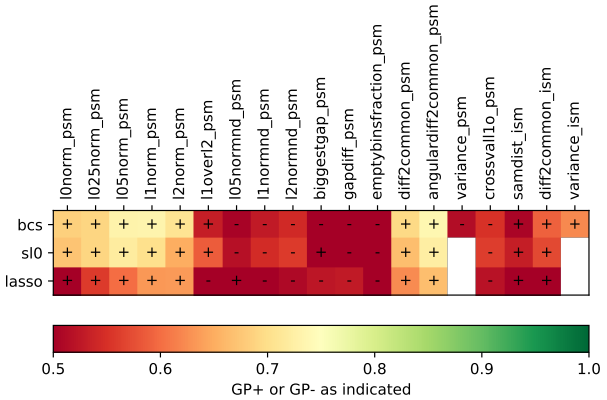


Figure 7.30.: Resulting good pairs values for all solvers and metrics on M2HSN data.

The absolute error at each band is shown with red "x" signs. The corresponding *samdist_ism* is indicated with blue "+" signs. In the median error and maximum error spectra, *samdist_ism* helps to identify the erroneous regions over 1000 nm. In the maximum error spectrum, additionally the erroneous regions around 700 nm, around 900 nm, and below 500 nm are identifiable based on *samdist_ism*. However, there are also cases where the metric is misleading, mainly at wavelengths below 500 nm both for the minimal and median error spectra.

For a more quantitative evaluation, figure 7.30 shows the the resulting GP values for all metrics and solvers, similar to the previous visualizations. It again shows the higher one of the good pairs metrics, specifying with a plus or a minus which one it is. The full correlation plots and meta metric values may be found in the appendix in figures A.2, A.3, and A.4. For ℓ_0, a threshold of 0.001 was used. We first concentrate on the results with BCS. Among the PSMs, significant GP_+ values are mainly observed for all the ℓ_p metrics in sparse domain and metrics that compare the signal with the common signal. Note, that the sparse version of the signal also consists of the differences to the original signal. Therefore, its ℓ_p metrics work similarly to the metrics comparing with the common signal.

The ℓ_p metrics for the normal domain show only a very weak correlation with the reconstruction quality. We attribute this to the failure cases of the solver not being well reflected in the norms of the normal domain. Failure cases here usually have localized deviations from the original signal, those may be over or below the actual value and thereby do not systematically influence the norms. As for the CFE, the cross validation metric behaves again mostly like the ℓ_p metrics for the normal domain.

The ISMs show slightly higher GP_+ values here than in the synthetic example. For *samdist_ism*, this may not be so obvious with a GP_+ value

0.507. However, in comparison with the much lower GP_{-} of 0.322 it can be considered about as good as the other ISMs. Especially for *samdist_ism*, this slight advantage in comparison to the synthetic data may be attributed to the transform base. The difference matrix does not smear the information of a single sample across the whole signal as the DCT does. Therefore, samples in the vicinity help to improve the signal locally. However, while this is working for the ISMs, the indication is not sufficient to make the corresponding PSMs successful, except for the difference to the common signal where the PSMs are even more successful. We attribute the reduced performance of *diff2common_ism* in comparison to the corresponding PSMs to the easy displacement of large parts of the common part through the difference matrix for the innovation term. Thereby, the differences to the common signal will greatly vary within the signal anyways for both good and bad reconstructions. For the other solvers, the findings are very similar. For LASSO, both the norms in sparse domain and the ISMs perform worse.

7.8. Conclusion

In this evaluation chapter, we have shown that the CS based solutions developed in this thesis, CFE and M2HSN are actually beneficial. For the M2HSN, we have also provided a selection guide for different solutions based on the findings in the evaluations. The prototype showed that the measurement hardware and its placement offer more potential for improvements than the spectral upsampling. However, it already showed results competing with professional systems and an improvement from the spectral upsampling.

For the quality metrics, we have found that the approaches which intuitively seem promising based on the sample positions and the compressed sensing requirements showed no evidence of actually being suitable. Quality differences within a signal were only estimated correctly for the HeM2HSN data because of the difference transform. For the quality difference between signals, whenever the sample size varies, the compression ratio is a very good indicator which is hard to beat. For constant sample sizes, the most promising approach is identifying cases where solvers fail. This is surprisingly easy with a higher order ℓ_p norm of f or x for multiple solvers. The most suitable of these metrics and its correlation with the reconstruction quality is dependent on solver, transform, and data and must be found for every application. The metrics from related work, namely the variance and the cross validation result did not show good performance for the comparison of solutions. However, in comparison to the other approaches, they estimate the magnitude of the actual RMSE well, making them useful for error bars.

8. Conclusion and Future Work

In section 1, we have identified two main issues of building Compressed Sensing based solutions for real world applications. The first issue is the complexity of building solutions due to their interdisciplinary nature. We have contributed solutions to this issue by supplying a more structured approach to engineering CS-based solutions in section 2 and by building two such solutions for precision agriculture, the CFE and the M2HSN. The second issue is the lack of information on the quality of reconstructions. We have contributed to the solution of this issue by providing a detailed comparison of approaches for quantifying this information for the data in both CFE and M2HSN.

Along the way, we have found the following answers to the research questions from section 1: question 1 was on the choice of sparse samples. For CFE in section 7.1, we have found that a homogeneous distribution of samples is usually the best choice even though the randomness is limited due to constraints on the random sampling. When similar data is measured repeatedly, a specific selection can be helpful, as seen for the HoM2HSN in section 7.3. This improvement was even transferable to the HeM2HSN as seen in section 4.5.2, albeit only for small networks. For larger networks, it imposes too much of a constraint on the random sampling. Despite these findings, the homogeneity of sampling positions did not turn out useful for the assessment of the reconstruction quality in sections 7.5–7.7 except for in-signal comparison for the HeM2HSN.

Question 2 was on the quality of the hyperspectral data of the HeM2HSN. Quantitative results for this were found in sections 7.2 and 7.3. In the prototype evaluation in section 7.4, the upsampling error turned out smaller than typical measurement errors. For NDVI, it was also demonstrated that the up- and down-scaling can indeed help to make data from different multispectral sensors comparable.

Question 3 was on the estimate of the reconstruction quality. Here, we have found that this is only possible to a very limited extent. Literature approaches such as BCS and cross validation are suitable for estimating the magnitude of the error but not for comparing the quality of different results for similar data. For this purpose, simple metrics for identifying solver failure cases are more effective.

In our future work, we would like to investigate an improvement of the HeM2HSN by building clusters of similar solutions, some of the preliminary work in this direction has been provided in this thesis with the group-building and the quality metrics. We are also interested in further

miniaturizing the sensors. One long-term vision known as smart dust is having simple bio-degradable sensors that may be distributed on the field just like fertilizer or seeds. If this becomes reality, we have described one option for realizing parts of it. Of course, it is impossible to foresee whether this kind of sensor network will actually be chosen in the future. Another alternative for continuous monitoring may be swarms of autonomous drones which use multispectral cameras. If this technology gets chosen instead, the contributions from this thesis can help as well, CFE can be applied to fill spatial gaps and the spectral up-scaling algorithms of the M2HSN may render cheaper cameras sufficient.

In the nearer future, we are planning to make a larger-scale deployment to further investigate the potential of measuring spatial differences accurately. Besides the impact on the NDVI, we are also working on evaluating the impact on further derived metrics. We would also like to expand these considerations to application areas beyond agriculture, e.g., to surveying the use of artificial light at night in the course of a night.

In this thesis, narrowband filters were used to sample continuous spectra. When observing spectra of gasses instead of solids, this becomes harder because these spectra consist of discrete lines. Generally, the spectral up-scaling algorithm can work for these spectra as well when using different filters than simple narrowband filters. Similar to the selection of filters, it may even become interesting to produce filters with transmission spectra optimized for estimating certain classes of spectra. This may even develop into a new approach of building spectrometers for many application areas. However, a major limitation in using filters which are not narrowband is that the light source spectrum also needs to be considered more explicitly in the calculations which makes calibration harder, albeit it may be applicable in situations with controlled illumination.

Going back to the issues that initially sparked this research, we are planning to further facilitate the use of CS and its quality estimate by providing the tools developed during this thesis project.

A. Appendix

Table A.1.: Resulting metrics for the synthetic dataset using BCS and varying all parameters.

Metric	R^2	R^2p	GP_+	GP_+p	GP_-	GP_-p
l0norm_psm	0.128	0.000	0.418	0.993	0.534	0.163
l025norm_psm	0.720	0.000	0.493	0.588	0.507	0.414
l05norm_psm	0.221	0.000	0.503	0.461	0.497	0.541
l1norm_psm	0.192	0.000	0.344	1.000	0.656	0.000
l2norm_psm	0.501	0.000	0.249	1.000	0.751	0.000
l1overl2_psm	0.833	0.000	0.697	0.000	0.303	1.000
l05normnd_psm	0.541	0.000	0.231	1.000	0.769	0.000
l1normnd_psm	0.525	0.000	0.238	1.000	0.762	0.000
l2normnd_psm	0.475	0.000	0.252	1.000	0.748	0.000
biggestgap_psm	0.539	0.000	0.767	0.000	0.220	1.000
gapdiff_psm	0.535	0.000	0.766	0.000	0.220	1.000
emptybinsfraction_psm	0.120	0.000	0.643	0.000	0.347	1.000
compressionratio_psm	0.776	0.000	0.755	0.000	0.173	1.000
variance_psm	0.580	0.000	0.714	0.000	0.286	1.000
crossvall1o_psm	0.430	0.000	0.703	0.000	0.297	1.000
samdist_ism	0.011	0.365	0.439	1.000	0.424	1.000
variance_ism	0.015	0.260	0.522	0.074	0.478	0.926

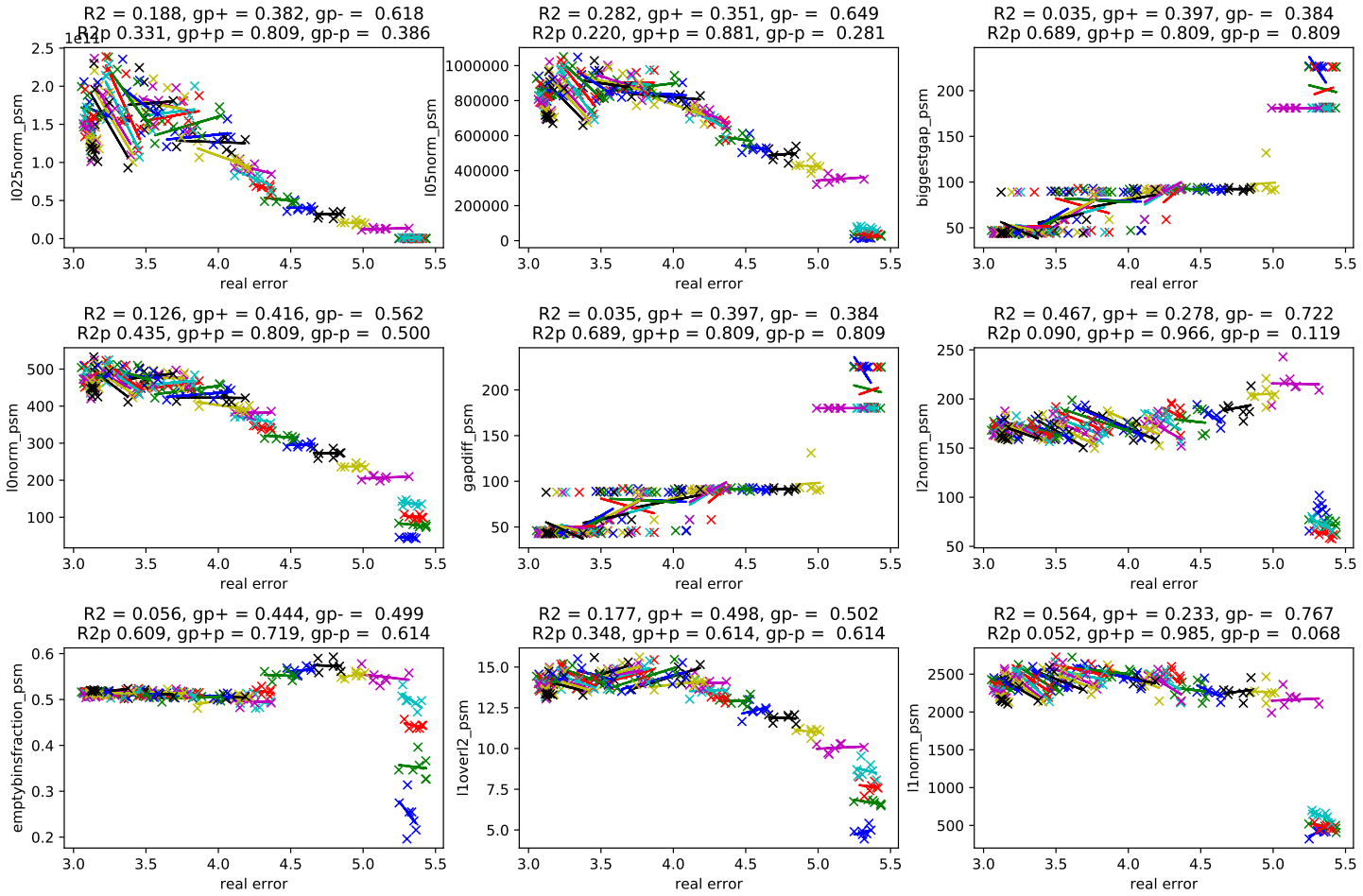


Figure A.1.: Correlation plots for PSM metrics on iterative CFE data using solver LASSO.

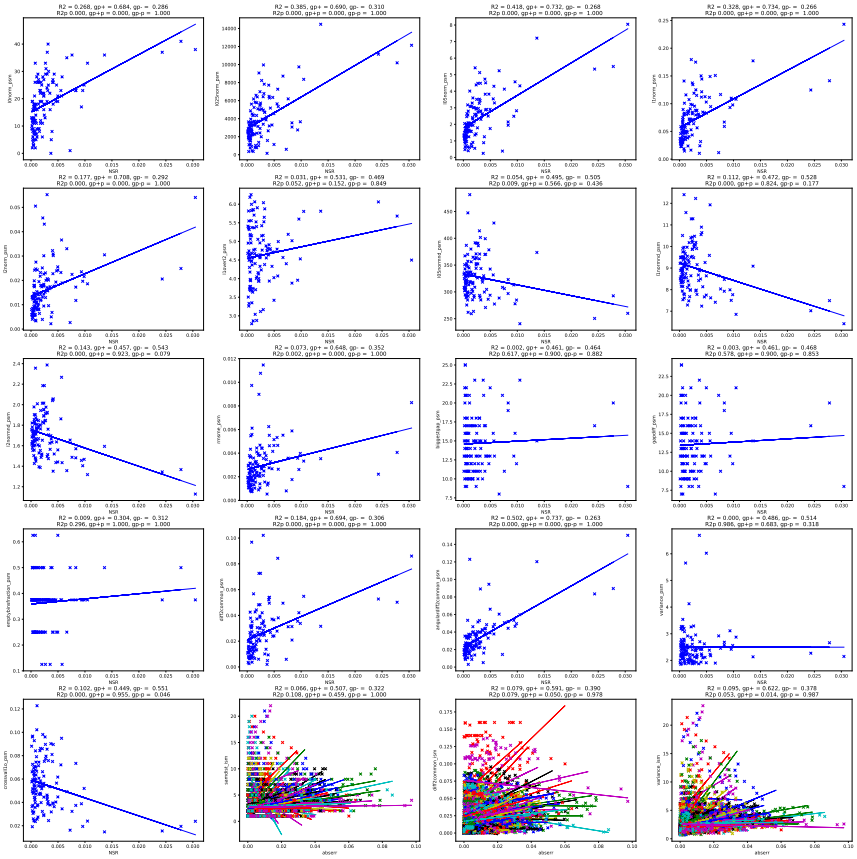


Figure A.2.: Resulting correlation plots for solver BCS on M2HSN data.

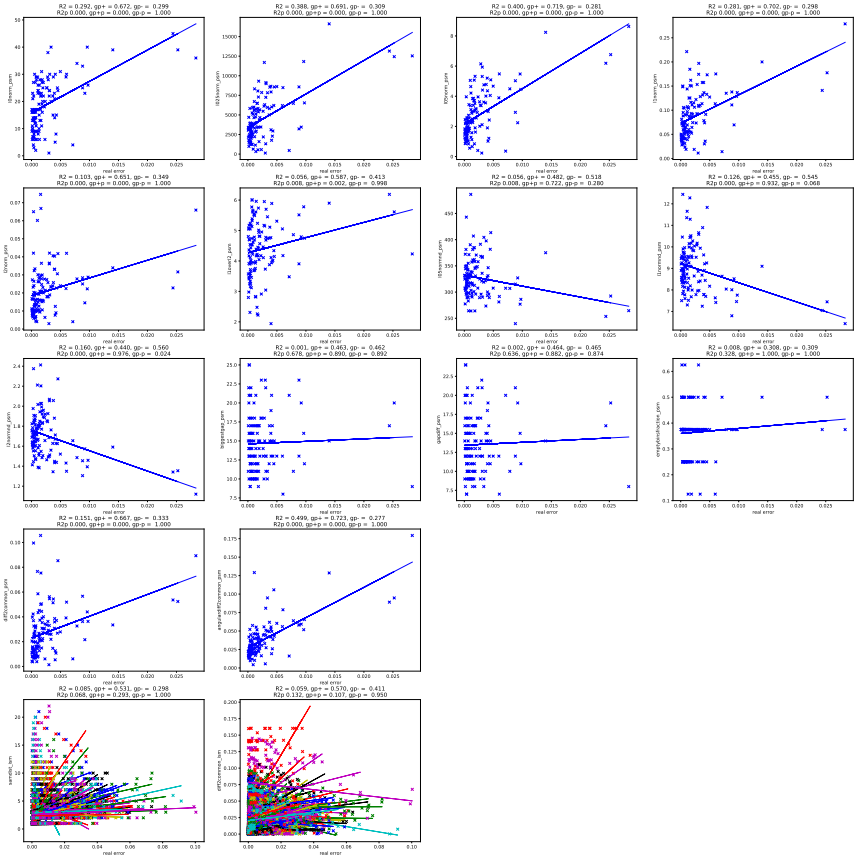


Figure A.3.: Resulting correlation plots for solver SL0 on M2HSN data.

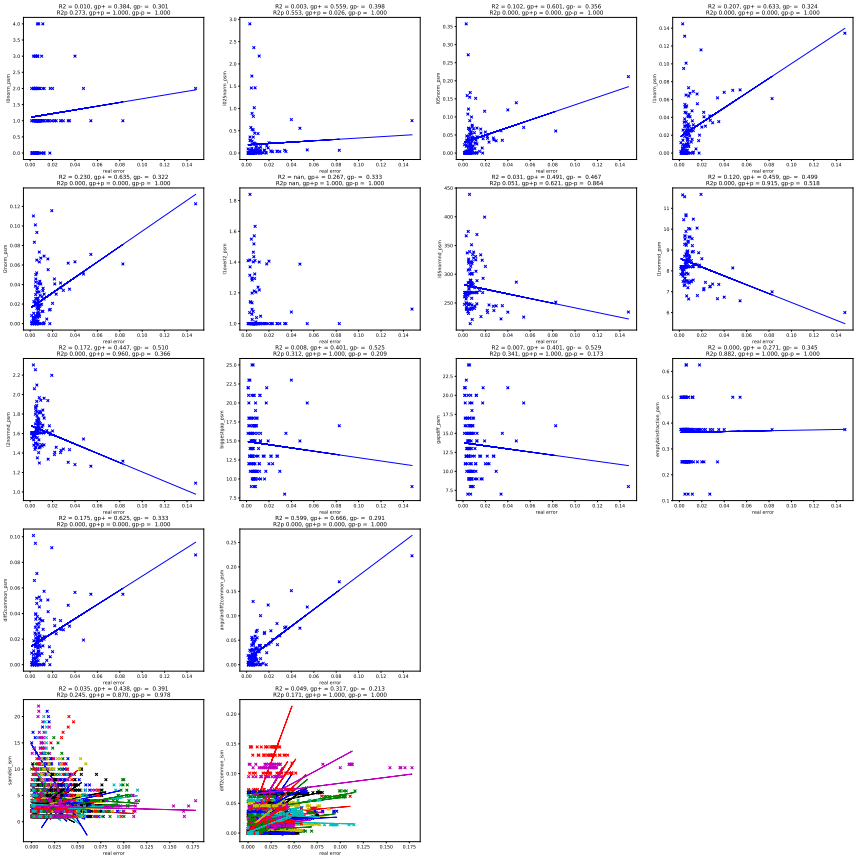


Figure A.4.: Resulting correlation plots for solver LASSO on M2HSN data.

Previous Partial Dissertation Publications

Hänel, T. and Aschenbruck, N. (2017a). "Accelerating Yield Mapping at Low Data Rates Using Compressive Field Estimate". In: *Proc. of the 42nd IEEE Conference on Local Computer Networks (LCN)*, pp. 374–382.
https://doi.org/10.1109/LCN.2017.56.

Hänel, T., Jarmer, T., and Aschenbruck, N. (2019). "Using distributed compressed sensing to derive continuous hyperspectral imaging from a wireless sensor network". In: *Computers and Electronics in Agriculture* Vol. 166. No. 104974.
https://doi.org/10.1016/j.compag.2019.104974.

Hänel, T., Jarmer, T., and Aschenbruck, N. (2021b). "Learning a Transform Base for the Multi- to Hyperspectral Sensor Network with K-SVD". In: *MDPI Sensors* Vol. 21. No. 21.
https://doi.org/10.3390/s21217296.

Other (Co-) Authored Publications

Rosemann, N., Brockmann, W., and Hänel, R. T. (2011). "Tackling uncertainties in self-optimizing systems by strategy blending". In: *Proc. of the IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 185–192.
https://doi.org/10.1109/EAIS.2011.5945920.

Hänel, T., Bothe, A., and Aschenbruck, N. (2015a). "RaLaNS — A ray launching based propagation loss model for ns-3". In: *Proc. of the Int. Conf. and Workshops on Networked Systems (NetSys)*, pp. 1–7.
https://doi.org/10.1109/NetSys.2015.7089069.

Hänel, T., Schwamborn, M., Bothe, A., and Aschenbruck, N. (2015b). "On the map accuracy required for network simulations based on ray launching". In: *Proc. of the 16th Int. IEEE Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–8.
https://doi.org/10.1109/WoWMoM.2015.7158158.

Hänel, T., Krampe, F., Gericke, C., and Aschenbruck, N. (2016). "On the Potential of Data-Based Time Synchronization in Wireless Sensor Networks for Condition Monitoring". In: *Proc. of the Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, pp. 216–224.
https://doi.org/10.1109/DCOSS.2016.11.

Hänel, T., Bothe, A., Helmke, R., Gericke, C., and Aschenbruck, N. (2017b). "Adjustable security for RFID-equipped IoT devices". In: *Proc. of the Int. IEEE Conference on RFID Technology Application (RFID-TA)*, pp. 208–213.
https://doi.org/10.1109/RFID-TA.2017.8098883.

Tessmer, A., Hänel, T., and Aschenbruck, N. (2020). "On the Impact of Geographic Restrictions on Contact Metrics in Opportunistic Networks". In: *Proc. of the 45th IEEE LCN Symposium on Emerging Topics in Networking*, pp. 131–140.
https://doi.org/10.1109/LCNSymposium50271.2020.9363269.

Hänel, T., Brüggemann, L., Loske, F., and Aschenbruck, N. (2021a). "Long-Term Wireless Sensor Network Deployments in Industry and Office Scenarios". In: *Proc. of the 22nd Int. IEEE Symposium on a*

World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 109–118.
https://doi.org/10.1109/WoWMoM51794.2021.00024.

Mast, J., Hänel, T., and Aschenbruck, N. (2021). "Enhancing Adaptive Frequency Hopping for Bluetooth Low Energy". In: *Proc. of the 46th IEEE Conf. on Local Computer Networks (LCN)*, pp. 447–454.
https://doi.org/10.1109/LCN52139.2021.9524988.

References

Aharon, M., Elad, M., and Bruckstein, A. (2006). "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation". In: *IEEE Transactions on Signal Processing* Vol. 54. No. 11, pp. 4311–4322.
https://doi.org/10.1109/TSP.2006.881199.

Arad, B. and Ben-Shahar, O. (2016). "Sparse recovery of hyperspectral signal from natural RGB images". In: *Proc. of the European Conf. on Computer Vision (ECCV)*, pp. 19–34.
https://doi.org/10.1007/978-3-319-46478-7_2.

Arad, B. and Ben-Shahar, O. (2017). "Filter Selection for Hyperspectral Estimation". In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.
https://doi.org/10.1109/ICCV.2017.342.

Auernhammer, H. (2001). "Precision farming — the environmental challenge". In: *Computers and Electronics in Agriculture* Vol. 30. No. 1–3, pp. 31–43.
https://doi.org/10.1016/S0168-1699(00)00153-8.

Bajwa, W., Haupt, J., Sayeed, A., and Nowak, R. (2006). "Compressive Wireless Sensing". In: *Proc. of the 5th Int. Conf. on Information Processing in Sensor Networks (IPSN)*. ACM, pp. 134–142.
https://doi.org/10.1145/1127777.1127801.

Barnes, E., Clarke, T., Richards, S., Colaizzi, P., Haberland, J., Kostrzewski, M., Waller, P., Choi, C., Riley, E., Thompson, T., et al. (2000). "Coincident detection of crop water stress, nitrogen status and canopy density using ground based multispectral data". In: *Proc. of the Fifth International Conf. on Precision Agriculture, Bloomington, MN, USA*.
http://handle.nal.usda.gov/10113/4190.

Baron, D., Duarte, M. F., Wakin, M. B., Sarvotham, S., and Baraniuk, R. G. (2009). "Distributed Compressive Sensing". In: *CoRR* Vol. abs/0901.3403.
https://arxiv.org/abs/0901.3403.

Bauer, J., Siegmann, B., Jarmer, T., and Aschenbruck, N. (2016). "On the potential of Wireless Sensor Networks for the in-situ assessment

of crop leaf area index". In: *Computers and Electronics in Agriculture* Vol. 128, pp. 149–159.
https://doi.org/10.1016/j.compag.2016.08.019.

Bi, D., Xie, Y., Li, X., and Zheng, Y. (2015). "A Sparsity Basis Selection Method for Compressed Sensing". In: *IEEE Signal Processing Letters* Vol. 22. No. 10, pp. 1738–1742.
https://doi.org/10.1109/LSP.2015.2429748.

Blackburn, G. A. (Sept. 2006). "Hyperspectral remote sensing of plant pigments". In: *Journal of Experimental Botany* Vol. 58. No. 4, pp. 855–867.
https://doi.org/10.1093/jxb/erl123.

Burger, W. and Burge, M. J. (2016). *Digital Image Processing*. Springer London.
https://doi.org/10.1007%2F978-1-4471-6684-9.

Caione, C., Brunelli, D., and Benini, L. (2012). "Distributed Compressive Sampling for Lifetime Optimization in Dense Wireless Sensor Networks". In: *IEEE Trans. on Industrial Informatics* Vol. 8. No. 1, pp. 30–40.
https://doi.org/10.1109/TII.2011.2173500.

Caione, C., Brunelli, D., and Benini, L. (2014). "Compressive Sensing Optimization for Signal Ensembles in WSNs". In: *IEEE Trans. on Industrial Informatics* Vol. 10. No. 1, pp. 382–392.
https://doi.org/10.1109/TII.2013.2266097.

Candes, E. and Wakin, M. (2008). "An Introduction To Compressive Sampling". In: *IEEE Signal Processing Magazine* Vol. 25. No. 2, pp. 21–30.
https://doi.org/10.1109/MSP.2007.914731.

Candes, E. J., Eldar, Y. C., Needell, D., and Randall, P. (2011). "Compressed sensing with coherent and redundant dictionaries". In: *Applied and Computational Harmonic Analysis* Vol. 31. No. 1, pp. 59–73.
https://doi.org/10.1016/j.acha.2010.10.002.

Candès, E. and Romberg, J. (2005). *ℓ_1-MAGIC: Recovery of Sparse Signals via Convex Programming*. Manual.
https://statweb.stanford.edu/~candes/software/l1magic/downloads/l1magic.pdf.

Charbiwala, Z., Kim, Y., Zahedi, S., Friedman, J., and Srivastava, M. B. (2009). "Energy Efficient Sampling for Event Detection in Wireless

Sensor Networks". In: *Proc. of the ACM/IEEE Int. Symposium on Low Power Electronics and Design (ISLPED)*, pp. 419–424. https://doi.org/10.1145/1594233.1594339.

Chavez Jr, P. (1989). "Radiometric calibration of Landsat Thematic Mapper multispectral images". In: *Photogrammetric engineering and remote sensing* Vol. 55. No. 9, pp. 1285–1294. https://www.asprs.org/wp-content/uploads/pers/1989journal/sep/1989_sep_1285-1294.pdf.

Chen, W. and Wassell, I. (2012). "Energy-efficient signal acquisition in wireless sensor networks: a compressive sensing framework". In: *IET Wireless Sensor Systems* Vol. 2 (1), pp. 1–8. https://doi.org/10.1049/iet-wss.2011.0009.

Chen, W. and Wassell, I. (2011). "Energy efficient signal acquisition via compressive sensing in wireless sensor networks". In: *Proc. of the 6th Int. Symposium on Wireless and Pervasive Computing (ISWPC)*, pp. 1–6. https://doi.org/10.1109/ISWPC.2011.5751335.

Chou, C. T., Rana, R., and Hu, W. (2009). "Energy efficient information collection in wireless sensor networks using adaptive compressive sensing". In: *Proc. of the 34th IEEE Conf. on Local Computer Networks (LCN)*, pp. 443–450. https://doi.org/10.1109/LCN.2009.5355162.

Cilia, M. and Yamamoto, K. (2019). "Intelligent IoT Sensing System Based on Compressive Sensing with Adaptively Learned Dictionary". In: *Proc. of the 38th IEEE International Performance Computing and Communications Conference (IPCCC)*. IEEE, pp. 1–8. https://doi.org/10.1109/IPCCC47392.2019.8958723.

Cinzano, P. (2005). *Night Sky Photometry with Sky Quality Meter*. Tech. rep. Istituto di Scienza e Tecnologia dell'Inquinamento Luminoso (ISTIL).

Colonnese, S., Cusani, R., Rinauro, S., Ruggiero, G., and Scarano, G. (2013). "Efficient compressive sampling of spatially sparse fields in wireless sensor networks". In: *EURASIP Journal on Advances in Signal Processing* Vol. 2013. No. 1. https://doi.org/10.1186/1687-6180-2013-136.

Cooley, J. W. and Tukey, J. W. (1965). "An algorithm for the machine calculation of complex Fourier series". In: *Mathematics of computation* Vol. 19. No. 90, pp. 297–301.

Crespo Marques, E., Maciel, N., Naviner, L., Cai, H., and Yang, J. (2019). "A Review of Sparse Recovery Algorithms". In: *IEEE Access* Vol. 7, pp. 1300–1322.
https://doi.org/10.1109/ACCESS.2018.2886471.

Donoho, D. L. (2006). "Compressed sensing". In: *IEEE Trans. on Information Theory* Vol. 52. No. 4, pp. 1289–1306.
https://doi.org/10.1109/TIT.2006.871582.

Du, M. and Noguchi, N. (2017). "Monitoring of Wheat Growth Status and Mapping of Wheat Yield's within-Field Spatial Variations Using Color Images Acquired from UAV-camera System". In: *Remote Sensing* Vol. 9. No. 3.
https://doi.org/10.3390/rs9030289.

Duarte, M. and Baraniuk, R. (2012). "Kronecker Compressive Sensing". In: *IEEE Trans. on Image Processing* Vol. 21. No. 2, pp. 494–504.
https://doi.org/10.1109/TIP.2011.2165289.

Ehlers, M., Klonus, S., Åstrand, P. J., and Rosso, P. (2010). "Multi-sensor image fusion for pansharpening in remote sensing". In: *Int. Journal of Image and Data Fusion* Vol. 1. No. 1, pp. 25–45.
https://doi.org/10.1080/19479830903561985.

Fazel, F., Fazel, M., and Stojanovic, M. (2011). "Random Access Compressed Sensing for Energy-Efficient Underwater Sensor Networks". In: *IEEE Journal on Selected Areas in Communications* Vol. 29. No. 8, pp. 1660–1670.
https://doi.org/10.1109/JSAC.2011.110915.

Fazel, F., Fazel, M., and Stojanovic, M. (2012). "Random access sensor networks: Field reconstruction from incomplete data". In: *Information Theory and Applications Workshop (ITA), 2012*, pp. 300–305.
https://doi.org/10.1109/ITA.2012.6181785.

Feizi, S. and Médard, M. (2011). "A Power Efficient Sensing/Communication Scheme: Joint Source-Channel-Network Coding by Using Compressive Sensing". In: *Proc. of the 49th Ann. Allerton Conf.*
https://arxiv.org/abs/1110.0428.

Gleichman, S. and Eldar, Y. C. (2011). "Blind Compressed Sensing". In: *IEEE Transactions on Information Theory* Vol. 57. No. 10, pp. 6958–6975.
https://doi.org/10.1109/TIT.2011.2165821.

Gupta, A. and Jha, R. K. (2015). "A Survey of 5G Network: Architecture and Emerging Technologies". In: *IEEE Access* Vol. 3, pp. 1206–1232.
https://doi.org/10.1109/ACCESS.2015.2461602.

Hänel, T. and Aschenbruck, N. (2017a). "Accelerating Yield Mapping at Low Data Rates Using Compressive Field Estimate". In: *Proc. of the 42nd IEEE Conference on Local Computer Networks (LCN)*, pp. 374–382.
https://doi.org/10.1109/LCN.2017.56.

Hänel, T., Bothe, A., and Aschenbruck, N. (2015a). "RaLaNS — A ray launching based propagation loss model for ns-3". In: *Proc. of the Int. Conf. and Workshops on Networked Systems (NetSys)*, pp. 1–7.
https://doi.org/10.1109/NetSys.2015.7089069.

Hänel, T., Bothe, A., Helmke, R., Gericke, C., and Aschenbruck, N. (2017b). "Adjustable security for RFID-equipped IoT devices". In: *Proc. of the Int. IEEE Conference on RFID Technology Application (RFID-TA)*, pp. 208–213.
https://doi.org/10.1109/RFID-TA.2017.8098883.

Hänel, T., Krampe, F., Gericke, C., and Aschenbruck, N. (2016). "On the Potential of Data-Based Time Synchronization in Wireless Sensor Networks for Condition Monitoring". In: *Proc. of the Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, pp. 216–224.
https://doi.org/10.1109/DCOSS.2016.11.

Hänel, T., Schwamborn, M., Bothe, A., and Aschenbruck, N. (2015b). "On the map accuracy required for network simulations based on ray launching". In: *Proc. of the 16th Int. IEEE Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–8.
https://doi.org/10.1109/WoWMoM.2015.7158158.

Hänel, T., Brüggemann, L., Loske, F., and Aschenbruck, N. (2021a). "Long-Term Wireless Sensor Network Deployments in Industry and Office Scenarios". In: *Proc. of the 22nd Int. IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 109–118.
https://doi.org/10.1109/WoWMoM51794.2021.00024.

Hänel, T., Jarmer, T., and Aschenbruck, N. (2019). "Using distributed compressed sensing to derive continuous hyperspectral imaging from a wireless sensor network". In: *Computers and Electronics in Agriculture* Vol. 166. No. 104974.
https://doi.org/10.1016/j.compag.2019.104974.

Hänel, T., Jarmer, T., and Aschenbruck, N. (2021b). "Learning a Transform Base for the Multi- to Hyperspectral Sensor Network with K-SVD". In: *MDPI Sensors* Vol. 21. No. 21.

https://doi.org/10.3390/s21217296.
Published under CC-BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

Haupt, J., Bajwa, W., Rabbat, M., and Nowak, R. (2008). "Compressed Sensing for Networked Data". In: *IEEE Signal Processing Magazine* Vol. 25. No. 2, pp. 92–101.
https://doi.org/10.1109/MSP.2007.914732.

Hennessy, A., Clarke, K., and Lewis, M. (2020). "Hyperspectral Classification of Plants: A Review of Waveband Selection Generalisability". In: *Remote Sensing* Vol. 12. No. 1, p. 113.
https://doi.org/10.3390/rs12010113.

Henrich, V., Götze, C., Jung, A., Sandow, C., Thürkow, D., and Cornelia, G. (2009). "Development of an online indices database: Motivation, concept and implementation". In: *Proc. of the 6th EARSeL Imaging Spectroscopy SIG Workshop Innovative Tool for Scientific and Commercial Environment Applications, Tel Aviv, Israel*.

Horler, D. N. H., Dockray, M., and Barber, J. (1983). "The red edge of plant leaf reflectance". In: *International Journal of Remote Sensing* Vol. 4. No. 2, pp. 273–288.
https://doi.org/10.1080/01431168308948546.

Huang, W., Huang, J., Wang, X., Wang, F., and Shi, J. (2013). "Comparability of red/near-infrared reflectance and NDVI based on the spectral response function between MODIS and 30 other satellite sensors using rice canopy spectra". In: *Sensors* Vol. 13. No. 12, pp. 16023–16050.
https://doi.org/10.3390/s131216023.

Jensen, A. and Cour-Harbor, A. la (2001). *Ripples in Mathematics, The Discrete Wavelet Transform*. 1st edition. Springer.
https://doi.org/10.1007/978-3-642-56702-5.

Ji, S., Xue, Y., and Carin, L. (2008). "Bayesian Compressive Sensing". In: *IEEE Trans. on Signal Processing* Vol. 56. No. 6, pp. 2346–2356.
https://doi.org/10.1109/TSP.2007.914345.

Jin, Z. and Bertozzi, A. L. (2007). "Environmental boundary tracking and estimation using multiple autonomous vehicles". In: *Proc. of the 46th IEEE Conf. on Decision and Control (CDC)*, pp. 4918–4923.
https://doi.org/10.3390/CDC.2007.4434857.

Kaiser, G. (2011). *A Friendly Guide to Wavelets*. Reprint of the 1994 Edition. Birkhäuser.
http://dx.doi.org/10.1007/978-0-8176-8111-1.

Karakus, C., Gurbuz, A. C., and Tavli, B. (2013). "Analysis of Energy Efficiency of Compressive Sensing in Wireless Sensor Networks". In: *IEEE Sensors Journal* Vol. 13. No. 5, pp. 1999–2008. https://doi.org/10.1109/JSEN.2013.2244036.

Kirolos, S., Laska, J., Wakin, M., Duarte, M., Baron, D., Ragheb, T., Massoud, Y., and Baraniuk, R. (2006). "Analog-to-Information Conversion via Random Demodulation". In: *Proc. of the IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, pp. 71–74. https://doi.org/10.1109/DCAS.2006.321036.

Laska, J. N., Kirolos, S., Duarte, M. F., Ragheb, T. S., Baraniuk, R. G., and Massoud, Y. (2007). "Theory and Implementation of an Analog-to-Information Converter using Random Demodulation". In: *Proc. of the Int. Symposium on Circuits and Systems (ISCAS)*, pp. 1959–1962. https://doi.org/10.1109/ISCAS.2007.378360.

Lauer, J., Richter, L., Ellersiek, T., and Zipf, A. (2014). "TeleAgro+: Analysis Framework for Agricultural Telematics Data". In: *Proc. of the 7th ACM SIGSPATIAL Int. Workshop on Computational Transportation Science (IWCTS)*. Dallas/Fort Worth, Texas: ACM, pp. 47–53. https://doi.acm.org/10.1145/2674918.2674925.

Lee, S., Pattem, S., Sathiamoorthy, M., Krishnamachari, B., and Ortega, A. (2009). "Spatially-Localized Compressed Sensing and Routing in Multi-hop Sensor Networks". In: *Proc. of the 3rd Int. Conf. on GeoSensor Networks (GSN)*, pp. 11–20. https://doi.org/10.1007/978-3-642-02903-5_2.

Leinonen, M., Codreanu, M., and Juntti, M. (2013). "Distributed correlated data gathering in wireless sensor networks via compressed sensing". In: *Proc. of the Asilomar Conf. on Signals, Systems and Computers*, pp. 418–422. htttp://doi.org/10.1109/ACSSC.2013.6810310.

Lin, M., Luo, C., Liu, F., and Wu, F. (2010). "Compressive Data Persistence in Large-Scale Wireless Sensor Networks". In: *Proc. of the Global Telecommunications Conf. (GLOBECOM)*. https://doi.org/10.1109/GLOCOM.2010.5684035.

Ling, Q. and Tian, Z. (2010). "Decentralized Sparse Signal Recovery for Compressive Sleeping Wireless Sensor Networks". In: *IEEE Trans. on Signal Processing* Vol. 58. No. 7, pp. 3816–3827. https://doi.org/10.1109/TSP.2010.2047721.

Liu, B., Zhang, L., Zhang, X., Zhang, B., and Tong, Q. (2009). "Simulation of EO-1 Hyperion Data from ALI Multispectral Data Based

on the Spectral Reconstruction Approach". In: *Sensors* Vol. 9. No. 4, pp. 3090–3108.
https://doi.org/10.3390/s90403090.

Liu, X. Y., Zhu, Y., Kong, L., Liu, C., Gu, Y., Vasilakos, A. V., and Wu, M. Y. (2015). "CDC: Compressive Data Collection for Wireless Sensor Networks". In: *IEEE Trans. on Parallel and Distributed Systems* Vol. 26. No. 8, pp. 2188–2197.
https://doi.org/10.1109/TPDS.2014.2345257.

LoRa-Alliance (2015). *LoRaWan — What is it?* White Paper. LoRa-Alliance.
www.lora-alliance.org.

Luo, C., Wu, F., Sun, J., and Chen, C. W. (2009). "Compressive Data Gathering for Large-scale Wireless Sensor Networks". In: *Proc. of the 15th Ann. Int. Conf. on Mobile Computing and Networking (MobiCom)*. ACM, pp. 145–156.
https://doi.org/10.1145/1614320.1614337.

Lustig, M., Donoho, D. L., Santos, J. M., and Pauly, J. M. (2008). "Compressed Sensing MRI". In: *IEEE Signal Processing Magazine* Vol. 25. No. 2, pp. 72–82.
https://doi.org/10.1109/MSP.2007.914728.

MacMahon, P. A. (1913). "The Indices of Permutations and the Derivation Therefrom of Functions of a Single Variable Associated with the Permutations of any Assemblage of Objects". In: *American Journal of Mathematics* Vol. 35. No. 3, pp. 281–322.
https://doi.org/10.2307/2370312.

Maes, W. H. and Steppe, K. (2019). "Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture". In: *Trends in Plant Science* Vol. 24. No. 2, pp. 152–164.
https://doi.org/10.1016/j.tplants.2018.11.007.

Mao, R. and Li, H. (2010). "A novel multiple access scheme via compressed sensing with random data traffic". In: *Journal of Communications and Networks* Vol. 12. No. 4, pp. 308–316.
https://doi.org/10.1109/JCN.2010.6388467.

Masiero, R., Quer, G., Munaretto, D., Rossi, M., Widmer, J., and Zorzi, M. (2009). "Data Acquisition through Joint Compressive Sensing and Principal Component Analysis". In: *Proc. of the Global Telecommunications Conf. (GLOBECOM)*, pp. 1–6.
https://doi.org/10.1109/GLOCOM.2009.5425458.

Mast, J., Hänel, T., and Aschenbruck, N. (2021). "Enhancing Adaptive Frequency Hopping for Bluetooth Low Energy". In: *Proc. of the 46th IEEE Conf. on Local Computer Networks (LCN)*, pp. 447–454. https://doi.org/10.1109/LCN52139.2021.9524988.

Mehrjoo, S., Shanbehzadeh, J., and Pedram, M. (2010). "A novel intelligent energy-efficient delay-aware routing in WSN, based on compressive sensing". In: *Proc. of the 5th Int. Symposium on Telecommunications (IST)*, pp. 415–420. https://doi.org/10.1109/ISTEL.2010.5734062.

Meng, J., Li, H., and Han, Z. (2009). "Sparse event detection in wireless sensor networks using compressive sensing". In: *43rd Ann. Conf. on Information Sciences and Systems (CISS 2009)*, pp. 181–185. https://doi.org/10.1109/CISS.2009.5054713.

Mirasi, A., Mahmoudi, A., Navid, H., Kamran, K. V., and Asoodar, M. A. (2021). "Evaluation of sum-NDVI values to estimate wheat grain yields using multi-temporal Landsat OLI data". In: vol. 36. 12. Taylor and Francis, pp. 1309–1324. https://doi.org/10.1080/10106049.2019.1641561.

Mohimani, H., Babaie-Zadeh, M., and Jutten, C. (2009). "A Fast Approach for Overcomplete Sparse Decomposition Based on Smoothed ℓ^0 Norm". In: *IEEE Trans. on Signal Processing* Vol. 57. No. 1, pp. 289–301. https://doi.org/10.1109/TSP.2008.2007606.

Mulla, D. J. (2013). "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps". In: *Biosystems Engineering* Vol. 114. No. 4. Special Issue: Sensing Technologies for Sustainable Agriculture, pp. 358–371. https://doi.org/10.1016/j.biosystemseng.2012.08.009.

Panek, E. and Gozdowski, D. (2021). "Relationship between MODIS Derived NDVI and Yield of Cereals for Selected European Countries". In: *Agronomy* Vol. 11. No. 2. https://doi.org/10.3390/agronomy11020340.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition". In: *Proc. of 27th Asilomar Conf. on Signals, Systems and Computers*, pp. 40–44. https://doi.org/10.1109/ACSSC.1993.342465.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and

Duchesnay, E. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* Vol. 12, pp. 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html.

Peyré, G. (2010). "Best Basis Compressed Sensing". In: *IEEE Trans. on Signal Processing* Vol. 58. No. 5, pp. 2613–2622. https://doi.org/10.1109/TSP.2010.2042490.

Polastre, J., Szewczyk, R., and Culler, D. (2005). "Telos: enabling ultra-low power wireless research". In: *Proc. of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 364–369. https://doi.org/10.1109/IPSN.2005.1440950.

Qaseem, S. T., Al-Naffouri, T. Y., and Al-Murad, T. M. (2009). "Compressive sensing based opportunistic protocol for exploiting multiuser diversity in wireless networks". In: *Proc. of the 20th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1447–1451. https://doi.org/10.1109/PIMRC.2009.5450184.

Quer, G., Masiero, R., Munaretto, D., Rossi, M., Widmer, J., and Zorzi, M. (2009). "On the interplay between routing and signal representation for Compressive Sensing in wireless sensor networks". In: *Proc. of the Information Theory and Applications Workshop*, pp. 206–215. https://doi.org/10.1109/ITA.2009.5044947.

Rembold, F., Atzberger, C., Savin, I., and Rojas, O. (2013). "Using low resolution satellite imagery for yield prediction and yield anomaly detection". In: *Remote Sensing* Vol. 5. No. 4, pp. 1704–1733. https://doi.org/10.3390/rs5041704.

Rosemann, N., Brockmann, W., and Hänel, R. T. (2011). "Tackling uncertainties in self-optimizing systems by strategy blending". In: *Proc. of the IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 185–192. https://doi.org/10.1109/EAIS.2011.5945920.

Rubin, M. and Camp, T. (2013). "On-mote compressive sampling to reduce power consumption for wireless sensors". In: *Proc. of the 10th Ann. IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 291–299. https://doi.org/10.1109/SAHCN.2013.6644998.

Rudelson, M. and Vershynin, R. (2006). "Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements". In: *Proc. of the 40th*

Ann. Conf. on Information Sciences and Systems (CISS), pp. 207–212.
https://doi.org/10.1109/CISS.2006.286463.

Sartipi, M. and Fletcher, R. (2011). "Energy-Efficient Data Acquisition in Wireless Sensor Networks Using Compressed Sensing". In: *Proc. of the Data Compression Conf. (DCC)*, pp. 223–232.
https://doi.org/10.1109/DCC.2011.29.

Siegmann, B. and Jarmer, T. (2015). "Comparison of different regression models and validation techniques for the assessment of wheat leaf area index from hyperspectral data". In: *International Journal of Remote Sensing* Vol. 36. No. 18, pp. 4519–4534.
https://doi.org/10.1080/01431161.2015.1084438.

Song, C., Woodcock, C. E., Seto, K. C., Lenney, M. P., and Macomber, S. A. (2001). "Classification and Change Detection Using Landsat TM Data: When and How to Correct Atmospheric Effects?" In: *Remote Sensing of Environment* Vol. 75. No. 2, pp. 230–244.
https://doi.org/10.1016/S0034-4257(00)00169-3.

Sun, X., Zhang, L., Yang, H., Wu, T., Cen, Y., and Guo, Y. (2015). "Enhancement of Spectral Resolution for Remotely Sensed Multispectral Image". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* Vol. 8. No. 5, pp. 2198–2211.
https://doi.org/10.1109/JSTARS.2014.2356512.

Takhar, D., Laska, J. N., Wakin, M. B., Duarte, M. F., Baron, D., Sarvotham, S., Kelly, K. F., and Baraniuk, R. G. (2006). "A new compressive imaging camera architecture using optical-domain compression". In: *Proc. of the SPIE Conf. on Computational Imaging IV*. Vol. 6065.
https://doi.org/10.1117/12.659602.

Tessmer, A. (2016). "Evaluation von Compressive Sensing in simulierten drahtlosen Sensornetzen zur Erfassung ausgewählter Pflanzenparameter basierend auf Lichtspektren". Bachelor Thesis. Osnabrück University.

Tessmer, A., Hänel, T., and Aschenbruck, N. (2020). "On the Impact of Geographic Restrictions on Contact Metrics in Opportunistic Networks". In: *Proc. of the 45th IEEE LCN Symposium on Emerging Topics in Networking*, pp. 131–140.
https://doi.org/10.1109/LCNSymposium50271.2020.9363269.

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* Vol. 58. No. 1, pp. 267–288.
http://www.jstor.org/stable/2346178.

Tsai, T.-Y., Lan, W.-C., Liu, C., and Sun, M.-T. (2013). "Distributed Compressive Data Aggregation in Large-Scale Wireless Sensor Networks". In: *Journal of Advances in Computer Networks* Vol. 1. No. 4, pp. 295–300.
https://doi.org/10.7763/JACN.2013.V1.59.

Tucker, C. J. (1979). "Red and photographic infrared linear combinations for monitoring vegetation". In: *Remote Sensing of Environment* Vol. 8. No. 2, pp. 127–150.
https://doi.org/10.1016/0034-4257(79)90013-0.

Tygert, M., Ward, R., and Zbontar, J. (2018). "Compressed sensing with a jackknife and a bootstrap". In:
https://arxiv.org/abs/1809.06959.

Vannoppen, A., Gobin, A., Kotova, L., Top, S., De Cruz, L., Vīksna, A., Aniskevich, S., Bobylev, L., Buntemeyer, L., Caluwaerts, S., De Troch, R., Gnatiuk, N., Hamdi, R., Reca Remedio, A., Sakalli, A., Van De Vyver, H., Van Schaeybroeck, B., and Termonia, P. (2020). "Wheat Yield Estimation from NDVI and Regional Climate Models in Latvia". In: *Remote Sensing* Vol. 12. No. 14.
https://doi.org/10.3390/rs12142206.

Vivone, G., Alparone, L., Chanussot, J., Mura, M. D., Garzelli, A., Licciardi, G. A., Restaino, R., and Wald, L. (2015). "A Critical Comparison Among Pansharpening Algorithms". In: *IEEE Trans. on Geoscience and Remote Sensing* Vol. 53. No. 5, pp. 2565–2586.
https://doi.org/10.1109/TGRS.2014.2361734.

Wang, W., Garofalakis, M., and Ramchandran, K. (2007). "Distributed Sparse Random Projections for Refinable Approximation". In: *Proc. of the 6th Int. Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 331–339.
https://doi.org/10.1109/IPSN.2007.4379693.

Wang, X., Zhao, Z., Xia, Y., and Zhang, H. (2010). "Compressed sensing based random routing for multi-hop wireless sensor networks". In: *Int. Symposium on Communications and Information Technologies (ISCIT)*, pp. 220–225.
https://doi.org/10.1109/ISCIT.2010.5664840.

Ward, R. (2009). "Compressed Sensing With Cross Validation". In: *IEEE Transactions on Information Theory* Vol. 55. No. 12, pp. 5773–5782.
https://doi.org/10.1109/TIT.2009.2032712.

Yu, X., Zhao, H., Zhang, L., Wu, S., Krishnamachari, B., and Li, V. O. K. (2010). "Cooperative Sensing and Compression in Vehicular Sensor

Networks for Urban Monitoring". In: *Proc. of the IEEE Int. Conf. on Communications (ICC)*.
https://doi.org/10.1109/ICC.2010.5502562.

Zhang, B., Cheng, X., Zhang, N., Cui, Y., Li, Y., and Liang, Q. (2011). "Sparse target counting and localization in sensor networks based on compressive sensing". In: *Proc. of IEEE INFOCOM*, pp. 2255–2263.
https://doi.org/10.1109/INFCOM.2011.5935041.

Zhang, N., Wang, M., and Wang, N. (2002). "Precision agriculture—a worldwide overview". In: *Computers and Electronics in Agriculture* Vol. 36. No. 2–3, pp. 113–132.
https://doi.org/10.1016/S0168-1699(02)00096-0.

List of Figures

Acronyms

5G	5th-generation cellular network
ADC	Analog-to-Digital Converter
AIC	Analog-to-Information Converter
ALS	Ambient Light Sensor
ARS	Additive Random Sampling
ARQ	Automatic Repeat reQuest
BCS	Bayesian Compressive Sensing
BLE	Bluetooth Low Energy
CDC	Compressive Data Collection
CDG	Compressive Data Gathering
CDS(RW)	Compressive Distributed Sensing using Random Walk
CFE	Compressive Field Estimate
CPU	Central Processing Unit
CS	Compressed Sensing
CWS	Compressive Wireless Sensing
DACS	Data Acquisition and Compressive Sensing
DCS	Distributed Compressive Sensing
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSRP	Distributed Sparse Random Projections
DWT	Discrete Wavelet Transform
FDM	Fused Deposition Modeling
FFT	Fast Fourier Transform
GNSS	Global Navigation Satellite System
GP	Good Pairs
HeM2HSN	Heterogeneous M2HSN
HoM2HSN	Homogeneous M2HSN
IoT	Internet of Things
ISM	In-Signal Metric
JPEG	Joint Photographic Experts Group
JSM	Joint Sparsity Model
JSM-1	Joint Sparsity Model 1
K-SVD	K Singular Value Decomposition
KCS	Kronecker Compressive Sensing
LAI	Leaf Area Index
LASSO	Least Absolute Shrinkage and Selection Operator
LoRaWAN	Long Range Wireless Area Network
M2HSN	Multi- to Hyperspectral Sensor Network
MAC	Medium Access Control
NDRE	Normalized Difference Red Edge Index
NDVI	Normalized Difference Vegetation Index

NGBDI	Normalized Green-Blue Difference Index
NGRDI	Normalized Green-Red Difference Index
NIR	Near Infra Red
NSR	Noise-to-Signal Ratio
OMP	Orthogonal Matching Pursuit
P4	DJI Phantom 4 Multispectral
PAR	Photosynthetically Active Radiation
PCB	Printed Circuit Board
PETG	PolyEthylene Terephthalate Glycol
PLA	Polylactic Acid
PLMN	Public Land Mobile Network
PLSR	Partial Least Squares Regression
PSM	Per-Signal Metric
RACS	Random Access Compressed Sensing
RLCS	Radon-like Compressive Sampling
RMS	Root Mean Square
RMSE	Root Mean Square Error
RMSME	Root Mean Square Measurement Error
RTV	Randomized Time Vector
SL0	Smoothed ℓ_0
SLCS	Spatially Localized Compressive Sensing
SNR	Signal-to-Noise Ratio
SOCP	Second Order Cone Program
SREM	Spectral Resolution Enhancement Method
SVC	HR-1024i by Spectra Vista Corporation
SVD	Singular Value Decomposition
UAV	Unmanned Aerial Vehicle
UPDM	Universal Pattern Decomposition Method
UTM	Universal Transverse Mercator
VDVI	Visible-Band Difference Vegetation Index
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network