From the Institute of Neuro- and Bioinformatics
of the University of Lübeck
Director: Prof. Dr. rer. nat. Thomas Martinetz

# On the Learning of Orthogonal Dictionaries for Sparse Coding and the Adaptive Hierarchical Sensing of Sparse and Compressible Signals

Dissertation
for Fulfillment of Requirements for the Doctoral Degree
of the University of Lübeck
from the Department of Computer Sciences/Engineering

Submitted by
Henry Schütze
from Finsterwalde

Lübeck 2017

First referee: Prof. Dr. rer. nat. Thomas Martinetz
Second referee: Prof. Dr.-Ing. Alfred Mertins
Chairman: Prof. Dr. rer. nat. Karsten Keller
Date of oral examination: July 23rd, 2018
Approved for printing: July 25th, 2018

# Abstract

Sparsifying signal transforms play a fundamental role in various engineering disciplines. They allow to represent signals less redundantly by exploiting regular structures. Their scope of application is versatile and covers feature extraction, lossy compression and signal restoration, to name a few. Early pioneering developments of sparsifying signal transforms have been devoted to static orthogonal transform schemes. Later on, the learning of overcomplete dictionaries became popular as these sparsifying transforms can be tailored to the data. While numerous learning algorithms exist for that overcomplete setting, only few different learning strategies have been proposed for the orthogonal one, although orthogonality of the dictionary bears appealing advantages.

With the first part of this thesis we contribute to the collection of orthogonal dictionary learning methods. We propose two novel online learning methods that challenge the existing state-of-the-art batch learning strategies as they can achieve sparser representations. By Orthogonal Sparse Coding (OSC) we propose a stochastic descent approach which sequentially updates the dictionary atoms based on a fusion of a Hebbian learning rule and an iterative Gram-Schmidt orthogonalization scheme. By Geodesic Flow Orthogonal Sparse Coding (GF-OSC) we propose a stochastic gradient descent approach that is based on the geodesic flow optimization framework by Plumbley. The gradient of the cost function is derived in the space of free dictionary parameters and leads to a rotational update rule for the dictionary.

We compare the ability of different learning methods to recover an orthogonal reference dictionary from synthetic sparse data and show that OSC and GF-OSC master the recovery task for challenging scenarios for which the other methods fail, such as low sparsity levels or the presence of noise. We analyze the dictionaries that emerge from learning on real training data sets and show that those learned by OSC and GF-OSC achieve superior encoding performance, particularly for lower sparsity levels. Two applications of orthogonal dictionary learning by means of OSC are demonstrated. An image denoising experiment reveals that the use of an orthogonal dictionary learned by OSC leads to image restoration qualities comparable to the orthogonal dictionary learned by a baseline approach and an overcomplete dictionary learned by K-SVD. We also show that an orthogonal dictionary learned by OSC can be used for image compression and that the resulting rate-distortion performance can be improved relative to the JPEG baseline codec, particularly for low bit rates.

Nowadays, the sparse encodability of natural signals by sparsifying transforms is also exploited by contemporary acquisition paradigms, such as Compressed Sensing (CS), to capture only the crucial information of a signal by merely few linear measurements.

With the second part of this thesis we contribute to the collection of such alternative sampling techniques. We propose Adaptive Hierarchical Sensing (AHS) for sampling sparse or compressible signals by a number of linear measurements which corresponds to the measurement complexity of CS. AHS is an adaptive approach that selects sensing vectors during the sampling process based on simple decision rules and depending on previously observed measurements of the signal. Prior to sampling, the user chooses a suitable sparsifying transform in which the signal of interest is assumed to have a sparse or compressible representation. The transform determines the collection of sensing vectors. AHS gradually refines initially coarse measurements towards significant signal coefficients in the transform domain based on a sensing tree which provides a natural hierarchy of sensing vectors. AHS eventually captures significant signal coefficients and does not require a recovery stage based on inverse optimization. We formulate two AHS variants: $\tau$-AHS, a variant based on absolute comparisons of the measurements with a threshold, and $K$-AHS, a variant based on relative comparisons of the measurements.

On standard benchmark images, we demonstrate that $K$-AHS achieves lower reconstruction errors than $\tau$-AHS and, for the relevant scenario of few measurements, also lower reconstruction errors than CS. We present a learning strategy that optimizes, based on training data, the composition of sensing vectors and show, exemplarily for natural image patches, that it improves sensing performance and leads to meaningful spatial structures of the sensing vectors. Furthermore, we investigate the sensing performance of $K$-AHS mathematically from a deterministic and a probabilistic perspective. A sufficient condition is proven which guarantees to deterministically sample the $k$ most significant signal coefficients. The condition is applied to particular signal models in order to derive sufficient conditions depending on the model parameters. The analytical findings are supported by simulations with synthetic signals and real world images.

# Zusammenfassung

Sparsifizierende Signaltransformationen spielen in verschiedenen Ingenieursdisziplinen mittlerweile eine wichtige Rolle. Sie erlauben, Signale weniger redundant zu repräsentieren, indem sie reguläre Strukturen ausnutzen. Ihr Anwendungsbereich ist vielseitig und deckt Merkmalsextraktion, verlustbehaftete Kompression und Signalaufbereitung ab, um nur einige zu nennen. Frühe bahnbrechende Entwicklungen sparsifizierender Signaltransformationen, widmeten sich statischen orthogonalen Transformationen. Später wurde das Lernen übervollständiger Wörterbücher populär, da sparsifizierende Transformationen damit auf Daten zugeschnitten werden können. Während zahlreiche Lernalgorithmen für den übervollständigen Fall existieren, wurden nur wenige unterschiedliche Lernstrategien für den orthogonalen Fall vorgeschlagen, obwohl Orthogonalität des Wörterbuchs viele Vorzüge mit sich bringt.

Mit dem ersten Teil der Arbeit leisten wir einen Beitrag zum Gebiet der Lernmethoden für orthogonale Wörterbücher. Wir schlagen zwei neue online Lernmethoden vor, die existierende state-of-the-art Batch-Lernstrategien herausfordern, da sie spärlichere Kodierungen erzielen können. Mit Orthogonal Sparse Coding (OSC) schlagen wir ein stochastisches Abstiegsverfahren vor, das die Atome des Wörterbuchs sequentiell anpasst, beruhend auf einer Zusammenführung einer Hebbschen Lernregel und einem iterativen Gram-Schmidt Orthogonalisierungsschema. Mit Geodesic Flow Orthogonal Sparse Coding (GF-OSC) schlagen wir ein stochastisches Gradientenabstiegsverfahren vor, basierend auf der Optimierung mittels geodätischem Fluss von Plumbley. Der Gradient der Kostenfunktion wird im Raum der freien Wörterbuchparameter bestimmt und liefert eine rotierende Anpassungsregel für das Wörterbuch.

Wir vergleichen die Fähigkeit verschiedener Lernmethoden, ein orthogonales Referenzwörterbuch von synthetischen spärlichen Daten wiederherzustellen und zeigen, dass OSC und GF-OSC die Aufgabe in schwierigen Situationen meistern, bei denen die anderen Methoden versagen, wie beispielweise bei geringem Spärlichkeitsgrad oder bei der Anwesenheit von Rauschen. Wir analysieren die Wörterbücher, die sich beim Lernen auf reellen Trainingsdatensätzen herausbilden und zeigen, dass die von OSC und GF-OSC gelernten Wörterbücher eine bessere Kodierungsleistung erzielen, im Speziellen bei geringeren Spärlichkeitsgraden. Zwei Anwendungen des Lernens orthogonaler Wörterbücher durch OSC werden aufgezeigt. Ein Experiment zur Bildentrauschung zeigt, dass die Verwendung eines durch OSC gelernten orthogonalen Wörterbuchs zu vergleichbarer Wiederherstellungsgüte führt wie ein orthogonales Wörterbuch, das durch ein Standardverfahren gelernt bzw. wie ein übervollständiges Wörterbuchs, das durch K-SVD gelernt wurde. Wir zeigen auch, dass ein orthogonales Wörterbuch, das durch OSC gelernt wird, zur Bildkompression verwendet werden kann und dass die resultierende Rate-Distortion Güte relativ zum JPEG Standard verbessert werden kann, im Speziellen für geringe Bitraten.

Heutzutage wird die spärliche Kodierbarkeit von Signalen durch sparsifizierende Transformationen auch von modernen Akquisitionsparadigmen wie z.B. Compressed Sensing (CS) ausgenutzt, um die ausschlaggebende Information eines Signals mit lediglich wenigen linearen Messungen einzusammeln.

Mit dem zweiten Teil dieser Arbeit leisten wir einen Beitrag zum Gebiet solcher alternativen Samplingtechniken. Wir schlagen Adaptive Hierarchical Sensing (AHS) vor, um spärliche oder komprimierbare Signale mit einer Anzahl linearer Messungen zu erfassen, die der Messkomplexität von CS entspricht. AHS ist ein adaptiver Ansatz, der Sensingvektoren während des Samplingprozesses basierend auf einfachen Entscheidungsregeln bzgl. zuvor beobachteter Messungen des Signals auswählt. Vor dem Sampling wählt der Nutzer eine geeignete sparsifizierende Transformation aus, in der das Signal mutmaßlich eine spärliche oder komprimierbare Repräsentation hat. Die Transformation determiniert den Satz an Sensingvektoren. AHS verfeinert sukzessive anfänglich grobe Messungen hin zu signifikanten Signalkoeffizienten der Transformationsdomäne mittels eines Sensingbaums, der eine natürliche Hierarchie der Sensingvektoren repräsentiert. AHS erfasst letztlich signifikante Signalkoeffizienten und benötigt keine Wiederherstellungsstufe, die auf inverser Optimierung beruht. Wir formulieren zwei AHS Varianten: $\tau$-AHS, eine Variante basierend auf absoluten Vergleichen der Messungen mit einem Schwellwert, und $K$-AHS, eine Variante basierend auf relativen Vergleichen der Messungen.

Wir demonstrieren anhand von Benchmarkbildern, dass $K$-AHS geringere Rekonstruktionsfehler als $\tau$-AHS und CS erreicht, im Besonderen für das relevante Szenario von wenigen Messungen. Wir präsentieren eine Lernstrategie, die ausgehend von Trainingsdaten die Zusammensetzung der Sensingvektoren optimiert und zeigen beispielhaft für natürliche Bildausschnitte, dass die Sensingleistung dadurch gesteigert wird und zu sinnvollen räumlichen Strukturen bei den Sensingvektoren führt. Weiterhin untersuchen wir die Sensingleistung von $K$-AHS mathematisch von einer deterministischen und einer probabilistischen Perspektive. Wir können eine hinreichende Bedingung beweisen, die die Erfassung der $k$ signifikantesten Signalkoeffizienten garantiert. Die Bedingung wird für bestimmte Signalmodelle angewendet, um hinreichende Bedingungen für die Modellparameter abzuleiten. Die analytischen Ergebnisse werden durch Simulationen mit synthetischen Signalen und realen Bildern gestützt.

# Contents

# List of Figures

# Publications

- Schütze, H., Barth, E., and Martinetz, T. (2017). Adaptive Hierarchical Sensing for the Efficient Sampling of Sparse and Compressible Signals. in preparation

- Schütze, H., Barth, E., and Martinetz, T. (2016). Learning Efficient Data Representations with Orthogonal Sparse Coding. *IEEE Transactions on Computational Imaging*, 2(3):177–189

- Schütze, H., Barth, E., and Martinetz, T. (2015). Learning orthogonal sparse representations by using geodesic flow optimization. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8

- Schütze, H., Barth, E., and Martinetz, T. (2014). An adaptive hierarchical sensing scheme for sparse signals. In Rogowitz, B. E., Pappas, T. N., and de Ridder, H., editors, *Human Vision and Electronic Imaging XIX*, volume 9014 of *Proc. of SPIE Electronic Imaging*, pages 15:1–8

- Schütze, H., Barth, E., and Martinetz, T. (2013). Learning orthogonal bases for k-sparse representations. In Hammer, B., Martinetz, T., and Villmann, T., editors, *Workshop New Challenges in Neural Computation 2013*, volume 02/2013 of *Machine Learning Reports*, pages 119–120

- Schütze, H., Martinetz, T., Anders, S., and Madany Mamlouk, A. (2012). A Multivariate Approach to Estimate Complexity of FMRI Time Series. In Villa, A. E., Duch, W., Érdi, P., Masulli, F., and Palm, G., editors, *22nd International Conference on Artificial Neural Networks and Machine Learning*, volume 7553 of *Lecture Notes in Computer Science*, pages 540–547. Springer

# 1 Introduction

## 1.1 Orthogonal Dictionary Learning for Sparse Coding

Many higher level machine learning tasks, as for instance object recognition, rely on a suitable feature representation of raw input data such that initially hidden structural properties become accessible. The underlying objective is to extract and exploit relevant information from the data in order to solve the task or to improve performance on it [Bengio et al., 2013].

Many unsupervised machine learning problems impose a basic generative model, a linear feature model, on a given set of observed data [Roweis and Ghahramani, 1999, Oja, 2002]. In general, such a linear feature model can be formulated as a matrix factorization problem, by which the set of observed data instances, strung in a data matrix $\mathbf{X}$, is assumed to be (approximately) generated by a matrix product $\mathbf{WS}$, where $\mathbf{W}$ is a matrix representing some linear transform which maps the feature space to the input space and $\mathbf{S}$ is a matrix representing the data within the feature space. Commonly, $\mathbf{W}$ and $\mathbf{S}$ are unknown and shall be identified such that an objective function is optimized. The objective function embodies the desired criterion of the representation in a formal mathematical sense.

A well-known example of a linear feature model is given by the Principal Component Analysis (PCA) which asks for a low-dimensional decorrelated representation $\mathbf{S}$ of the data matrix, and an orthonormal matrix $\mathbf{W}$ spanning a low-dimensional subspace, such that the approximation $\mathbf{X} \approx \mathbf{WS}$ has minimal error [Pearson, 1901, Hotelling, 1933].

Likewise, Sparse Coding imposes a linear feature model on observed data [Olshausen and Field, 1996a, Rubinstein et al., 2010], and spans an important subclass of unsupervised machine learning problems [Lee et al., 2007]. The principal learning task for Sparse Coding can be phrased as follows: generate for a given set of observed signals a dictionary, a suitable collection of atomic signals, such that each observed signal can be well approximated by an individual sparse linear combination of atomic signals. In other words, find $\mathbf{W}$ and $\mathbf{S}$ such that, column-wise, only few entries of $\mathbf{S}$ are distinct from zero and such that $\mathbf{X} \approx \mathbf{WS}$ has minimal error.

Sparse Coding has a connection to the neurobiological processing of the brain, particularly to the encoding and processing of sensory inputs [Olshausen and Field, 2004] such as vision [Olshausen and Field, 1996a, Olshausen and Field, 1997], audition [Hromádka et al., 2008, Willmore and King, 2009], touch [Crochet et al., 2011], and olfaction [Ito et al., 2008, Lin et al., 2014], and also to memory formation [Kanerva, 1988, Palm, 2013]. Early work on Sparse Coding proposed that the goal of visual coding is to faithfully represent the visual input with minimal neural activity in order to save energy and computational resources. This principle is called efficient-coding hypothesis and goes back to Barlow [Barlow, 1961] and is based on earlier work of Mach [Mach, 1886] and MacKay [MacKay, 1956]. It has been later extended in several ways and related to the statistics of natural images [Field, 1994, Zetzsche et al., 1993, Olshausen and Field, 1996a]. Olshausen and Field have shown that learning a coding strategy that maximizes sparsity is sufficient to let atomic signals emerge that have receptive field properties of simple cells in the primary visual cortex [Olshausen and Field, 1996b, Olshausen and Field, 1996a].

Sparse Coding has various applications in the area of image processing and computer vision. For instance, dictionaries learned on image patches can be used for lossy compression, i.e. to store good approximate versions of uncompressed images at a much lower bit rate [Bryt and Elad, 2008, Skretting and Engan, 2011, Pati et al., 2015]. Furthermore, Sparse Coding can be used to restore corrupted images, i.e. to remove noise [Elad and Aharon, 2006, Sundaresan and Porikli, 2012], to fill in intensity values for missing pixels [Mairal et al., 2008a, Mairal et al., 2008b], or to revert the convolution of an image with a known filter (deblurring) [Yang et al., 2014, Xiang et al., 2015]. Sparse Coding approaches have also been used in many pattern recognition applications, for instance to classify images [Labusch et al., 2008, Mairal et al., 2009, Qin et al., 2016, Bao et al., 2016].

The vast majority of existing methods to learn dictionaries for Sparse Coding covers primarily the non-orthogonal overcomplete setting, in which the number of atomic signals is much larger than the data dimensionality. The overcomplete setting has been focused on, in order to capture invariances, to achieve robustness in the presence of noise, flexibility to fit the data, and coding efficiency [Rubinstein et al., 2010, Lewicki and Sejnowski, 2000, Elad, 2010].

In the first part of this thesis, we propose and investigate novel methods to learn complete orthogonal dictionaries for Sparse Coding [Schütze et al., 2013, Schütze et al., 2015, Schütze et al., 2016]. The question arises: What is the motivation to contribute to the orthogonal dictionary variant of the Sparse Coding problem?

First, learning a dictionary for Sparse Coding induces in most cases an alternating update scheme of two nested subproblems [Rubinstein et al., 2010, Elad, 2010]. In the orthogonal setting, there is a distinct advantage: both subproblems can be solved fast and optimally. One subproblem, finding the optimal sparse representation of a signal subject to a given dictionary, is particularly important as it might be solved frequently after learning is finished. The efficient and optimal solvability of both subproblems entails a fast alternating batch learning approach that has been independently developed for different models [Lesage et al., 2005, Sezer et al., 2008, Bao et al., 2013, Cai et al., 2014]. Unfortunately, it suffers from suboptimal solutions in conditions that are quite relevant in practice, which leaves room for improvements and motivates the development of new algorithmic approaches.

Second, many natural signals, e.g. natural images can be sparsely encoded by orthogonal linear transforms. This fact has been exploited in the area of image compression to build efficient codecs such as the JPEG standard [Pennebaker and Mitchell, 1992]. By using an adequate analytic orthogonal transform, e.g. the Discrete Cosine Transform (DCT) [Ahmed et al., 1974], many transform coefficients are close to zero and do not need not be encoded. If the (image) data originates from a different domain with unique statistical properties, but is nonetheless sparsely encodable by an orthogonal transform, a suitable analytic one might be unknown. A learning method copes with this issue as it is adaptive and generates the transform tailored to the statistics of the data.

Third, principal applications of Sparse Coding can be solved by orthogonal dictionaries as well [Sezer et al., 2008, Bao et al., 2013, Cai et al., 2014, Sezer et al., 2015, Bao et al., 2015, Rusu et al., 2016]. Furthermore, there are scenarios which require that the sparsifying transform (the dictionary) is invertible, e.g. for particular reconstruction approaches in the area of Compressed Sensing. Orthogonality is highly convenient as the dictionary serves simultaneously as synthesis and analysis transform.

Last but not least, approaches to solve the orthogonal dictionary learning problem are not exhausted. It has not attained as much attention as the counterpart dealing with overcomplete dictionaries. Consequently, only few conceptually different approaches have been proposed to solve the problem. Currently, it is still an active topic of research with recent contributions such as [Bao et al., 2015, Rusu et al., 2016, Rusu and Thompson, 2017].

## 1.2 Adaptive Hierarchical Sensing

During the last decade Compressed Sensing has rapidly emerged. It is now established as a sophisticated sampling technique in various engineering disciplines [Eldar and Kutyniok, 2012]. Many digital acquisition devices, for instance digital cameras, first fully sample the analog signal of interest and subsequently perform lossy compression to get rid of the vast amount of redundant information collected in the first stage. Compressed Sensing, on the contrary, is a much more efficient approach as it embeds the data compression step into the sampling stage [Takhar et al., 2006]. Given the signal is sparse or compressible in some transform domain, the total number of required Compressed Sensing measurements is much lower than the Nyquist-Shannon sampling theorem demands in the case of classical sampling [Candès et al., 2006, Donoho, 2006].

Fortunately, the sparseness assumption holds for many types of natural signals. Classical sampling of a signal, e.g. capturing a visual scene by a digital camera, can be seen as making linear measurements in terms of inner products of the signal with canonical basis vectors. With Compressed Sensing, inner products of the signal are instead measured sequentially with alternative sensing vectors. These sensing vectors can be composed of random entries, or can be randomly selected basis vectors of some transform basis. Given the small collection of linear measurements, the sparse representation of the signal is recovered by solving an inverse optimization problem. In essence, such an optimization reduces to the problem of finding a sparse solution to an underdetermined system of linear equations, and is thus related to Sparse Coding, particularly to sparse recovery problems [Kutyniok, 2012].

Compressed Sensing has found versatile applications. For radar imaging systems, Compressed Sensing is used to improve hardware designs and to increase resolution [Baraniuk and Steeghs, 2007, Herman and Strohmer, 2009, Potter et al., 2010, Ender, 2010]. In the area of Magnetic Resonance Imaging (MRI), image acquisition is done in the Fourier domain, which allows to apply Compressed Sensing to improve the image quality while reducing the number of collected measurements [Lustig et al., 2008, Gamper et al., 2008, Jung et al., 2009]. Compressed sensing has also found applications in the area of seismic imaging to improve acquisition of seismic data [Herrmann and Hennenfent, 2008, Hennenfent and Herrmann, 2008] Furthermore, single pixel imaging has been realized based on Compressed Sensing, which contributed considerably to its popularity. A single photo detector can be used, in combination with some spatial light modulator, to capture images in fairly high resolution [Takhar et al., 2006, Wakin et al., 2006a, Wakin et al., 2006b, Welsh et al., 2013, Sun et al., 2013].

Commonly, Compressed Sensing measurements are collected non-adaptively, i.e. with the beginning of the acquisition process all sensing vectors are entirely determined [Donoho, 2006, Candes and Tao, 2006, Kutyniok, 2012]. They are sequentially processed during the sampling process independent from previously received sensing values. Due to the independence of the sensing vectors from the signal, non-adaptive sampling has been advocated as it prevents any computational overhead for computing the sensing vectors during the acquisition process.

In the second part of this thesis, we present and analyze novel adaptive approaches to the Compressed Sensing problem, where sensing vectors are selected dependent on values of previously observed measurements. In general, previously proposed adaptive Compressed Sensing schemes can lead to more accurate reconstructions, for instance in the presence of noise [Castro et al., 2008, Ji et al., 2008, Seeger, 2008, Seeger and Nickisch, 2008]. Furthermore, some adaptive approaches, e.g. [Deutsch et al., 2009], as well as the ones proposed in this thesis, do not rely on solving an optimization problem to reconstruct the signal, but identify relevant signal coefficients directly in the sparse transform domain [Schütze et al., 2014, Schütze et al., 2017].

## 1.3 Thesis Organization

The thesis is organized in two major parts.

The first part covers the topic orthogonal dictionary learning for Sparse Coding. After introducing the basic terminology, the learning problem is characterized algebraically and two principal sparse models are introduced together with their solutions to the sparse approximation problem. A literature review gives an overview of previous approaches to the problem. Subsequently, the Canonical Approach (CA), the Orthogonal Sparse Coding (OSC) as well as the Geodesic Flow Orthogonal Sparse Coding (GF-OSC) algorithms are presented. Subsequent sections cover various numerical experiments for methodical comparisons. On synthetic data, the superiority of OSC and GF-OSC is demonstrated at dictionary recovery tasks. Orthogonal dictionaries learned on real world image data are analyzed and their sparse encoding performance is assessed. Finally, applications are demonstrated in form of image compression and image denoising experiments.

The second part covers the topic Adaptive Hierarchical Sensing (AHS). First, we introduce the sensing problem formally, together with common approaches and requirements to reconstruct a signal from a small set of linear measurements. Prior to the detailed presentation of two AHS algorithms, we explain the central structural component of AHS, the sensing tree. Subsequently, $\tau$-AHS and $K$-AHS are presented, their sampling complexity is analyzed, and it is outlined how the sparse signal representation is obtained. We analyze mathematically situations in which AHS can miss important portions of a signal and prove a sufficient deterministic success condition. The performance of AHS is evaluated for synthetic signals as well as for natural images. A comparison of the imaging results with a conventional Compressed Sensing scheme is provided. We show that AHS sensing performance can be increased if the structure of the sensing tree is learned from training data. Furthermore, it is shown that, throughout the sensing procedure, AHS automatically intensifies the sensing at salient locations of the scene.

Finally, the developed methods and results presented in this thesis are concluded. A discussion weighs out value and limitations of the proposed methods and experiments and outlines possible advancements that remain open.

# 2 Orthogonal Dictionary Learning for Sparse Coding

This chapter is organized as follows. In Section 2.1 basic terms and definitions are introduced to make the reader familiar with the terminology for diving into the orthogonal sparse coding world. Section 2.3 and Section 2.4 introduce two principal models for learning orthogonal dictionaries for sparse coding: the constrained $K$-sparse model and the unconstrained regularized sparse model. For each model it is shown how to efficiently perform, for a given dictionary, optimal updates of sparse coefficients. A literature review in Section 2.5 gives an overview which algorithmic approaches have been proposed so far to solve the learning problem of interest. Section 2.6 presents the Canonical Approach (CA) which is a natural modification of a base line method from the unconstrained model to the constrained model. In Section 2.7, the new online learning algorithm Orthogonal Sparse Coding (OSC) is proposed to solve the constrained model using a Hebbian learning rule and Gram-Schmidt orthogonalization. In Section 2.8, a further new online learning algorithm, Geodesic Flow Orthogonal Sparse Coding (GF-OSC), is proposed to address the same model using a gradient descent approach based on geodesic flow optimization. Section 2.9 provides a performance comparison of several methods for the task to recover a generating orthogonal dictionary from synthetic sparse data. Section 2.10 and Section 2.11 present and analyze the orthogonal dictionaries that emerged from learning on natural image data and on image data of handwritten digits. Section 2.12 presents image compression and image denoising applications for dictionaries learned by the proposed methods.

## 2.1 Terminology and Formal Definitions

**Definition 1** (Data Sample)**.** In the following, we consider a data sample as a real $N$-element column vector and denote it by $\mathbf{x} \in \mathbb{R}^N$.

**Definition 2** (Data Set)**.** A data set is a collection of multiple, say $L$, data samples which are stored column-wise in a matrix $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L) \in \mathbb{R}^{N \times L}$.

**Definition 3** (Dictionary Atom)**.** A dictionary atom is an $N$-element column vector $\mathbf{u} \in \mathbb{R}^N$ with unit Euclidean length $\|\mathbf{u}\|_2 = 1$.

**Definition 4** (Orthogonal Dictionary)**.** A dictionary is a collection of multiple, say $M$, dictionary atoms which are stored column-wise in a matrix $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_M) \in \mathbb{R}^{N \times M}$. The dictionary is called orthogonal if for all $M(M-1)/2$ pairs of distinct dictionary atoms $\mathbf{u}_i$, $\mathbf{u}_j$ the inner product $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$, or in other words $\mathbf{U}^T\mathbf{U} = \mathbf{I}_M$. The dictionary can alternatively be termed orthonormal (rather than orthogonal) as the dictionary atoms have unit length. An orthogonal dictionary $\mathbf{U}$ is called undercomplete if $M < N$, and complete if $M = N$. In the latter case, $\mathbf{U}$ is an orthonormal basis (ONB). In the remaining part of this chapter, the complete setting can be assumed, if not explicitly stated otherwise.

**Definition 5** (Sparse Representation, Sparse Approximation)**.** Given an orthogonal dictionary $\mathbf{U}$, a data sample $\mathbf{x} \in \mathbb{R}^N$ is said to have a sparse representation by a coefficient vector $\hat{\mathbf{a}} \in \mathbb{R}^M$, if most of its entries – the coefficients – are zero or close to zero, and the data sample is well approximated as follows: $\mathbf{x} \approx \hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{a}}$. We call $\hat{\mathbf{x}}$ the sparse approximation of $\mathbf{x}$ subject to $\mathbf{U}$ and $\hat{\mathbf{a}}$.

**Definition 6** (Residual/Error of a Sparse Approximation)**.** The residual $\mathbf{r}$ of a sparse approximation $\hat{\mathbf{x}}$ of $\mathbf{x}$ is defined by $\mathbf{r} := \mathbf{x} - \hat{\mathbf{x}}$. Its squared Euclidean norm $\|\mathbf{r}\|_2^2$ – the residual norm – measures the approximation error.

**Definition 7** (Sparsity Measures)**.** To measure the sparsity of a vector $\mathbf{a}$ we use the $\ell_0$-norm $\|\cdot\|_0 : \mathbb{R}^N \to \{0, ..., N\}$ to obtain the size of its support, i.e. its number of non-zero coefficients

$$\|\mathbf{a}\|_0 = |\{j \mid a_j \neq 0\}| = \sum_{j=1}^{N} \mathbf{1}_{\mathbb{R}\setminus\{0\}}(a_j)\,. \tag{2.1}$$

The smaller $\|\mathbf{a}\|_0$, the higher is the sparsity of $\mathbf{a}$. Note that the $\ell_0$-norm is not a true norm as is does not satisfy the property of homogeneity. Alternatively, the $\ell_1$-norm $\|\cdot\|_1 : \mathbb{R}^N \to \mathbb{R}$ can be used as a convex relaxation to measure the sparsity of a vector

$$\|\mathbf{a}\|_1 = \sum_{j=1}^{N} |a_j|\,. \tag{2.2}$$

Generalizations of $\|\cdot\|_0$ and $\|\cdot\|_1$ for matrices are obtained by taking the indices over all matrix elements.

**Definition 8** (Overlap of two Vectors)**.** The (normalized) overlap of two vectors $\mathbf{v}$ and $\mathbf{w}$ is defined by

$$\operatorname{ovlp}(\mathbf{v}, \mathbf{w}) = \frac{|\mathbf{v}^T\mathbf{w}|}{\|\mathbf{v}\|_2 \, \|\mathbf{w}\|_2}\,, \tag{2.3}$$

and is equivalent to the magnitude of the cosine of the (aligned) angle that is embraced by $\mathbf{v}$ and $\mathbf{w}$. Note that $\operatorname{ovlp}(\mathbf{v}, \mathbf{w}) \in [0, 1]$ is invariant to a sign switch of $\mathbf{v}$ or $\mathbf{w}$ due to the absolute value taken in the numerator.

**Definition 9** (Mutual Coherence of a Dictionary)**.** The mutual coherence of a dictionary is the maximal overlap among all $M(M-1)/2$ pairs of distinct dictionary atoms.

**Definition 10** (Orthogonal Group)**.** The orthogonal group $\mathrm{O}(N)$ defines the set containing all ONBs spanning the $\mathbb{R}^N$:

$$\mathrm{O}(N) = \left\{ \mathbf{U} \mid \mathbf{U} \in \mathbb{R}^{N \times N}, \mathbf{U}^T\mathbf{U} = \mathbf{I}_N \right\} . \tag{2.4}$$

The orthogonal group consists of two disconnected subgroups: $\mathrm{SO}(N)$ and $\overline{\mathrm{SO}(N)} := \mathrm{O}(N) \setminus \mathrm{SO}(N)$. The $\mathrm{SO}(N)$ is called the special orthogonal group and contains all ONBs $\mathbf{U}$ with $\det(\mathbf{U}) = +1$.

## 2.2 Characterization of the Learning Problem

Learning a dictionary for sparse coding can be grasped as the task to find a suitable set of low dimensional linear subspaces to encode a given data set. The dictionary with its atoms is a collection of direction vectors from which small subsets can be taken to span lower-dimensional subspaces that contain the given data samples up to a small error. Suppose a data sample can be represented by a $K$-sparse linear combination of dictionary atoms, i.e. by the product of the dictionary and a $K$-sparse coefficient vector, then these coefficients correspond to the coordinates of the sample in the $K$-dimensional subspace that is spanned by the atoms which correspond to the indices of the non-zero coefficients.

### 2.2.1 Orthogonal and Overcomplete Dictionaries

Learning overcomplete dictionaries allows to arbitrarily increase the collection of atoms to a size larger than the dimensionality of the signal space which in turn increases the number of possible subspaces that can be used for encoding. Disjoint[1] linear subspaces composed from an overcomplete dictionary are mutually non-orthogonal which enables, in general, a better adaptation to the training data set and can represent a wider range of signal phenomena [Rubinstein et al., 2010, Lewicki and Sejnowski, 2000, Elad, 2010]. However, without further conditions on the dictionary it becomes difficult to compute the optimal sparse data representations, i.e. the optimal sparse coefficient vectors and their support. For general overcomplete dictionaries, this problem is NP-hard [Davis et al., 1997]. Sparse recovery algorithms like Basis Pursuit [Chen et al., 1998] or Orthogonal Matching Pursuit [Pati et al., 1993] can find optimal coefficients only if the dictionary satisfies particular conditions such as upper bounds of the mutual coherence [Donoho and Elad, 2003] or the restricted isometry property [Candes and Tao, 2005]. These properties require that dictionary atoms are not too similar and might be interpreted as a relaxation of orthogonality. However, unlike orthogonality, it is difficult to

[1] except the shared zero element $\mathbf{0}$

implement such properties as constraints in dictionary learning algorithms. Orthogonal dictionaries, on the other hand, are mathematically simple and also maximally incoherent. Disjoint linear subspaces composed from an orthogonal dictionary are mutually orthogonal with the implication that optimal sparse coefficients of a data sample can be efficiently computed from its dense representation. Moreover, an orthogonal dictionary can be easily inverted. It serves simultaneously as synthesis and as analysis operator.

### 2.2.2 Interpretation As a Special Blind Source Separation Problem

The orthogonal dictionary learning problem for sparse coding can be casted to a special blind source separation (BSS) problem [Mishali and Eldar, 2009, Dobigeon and Tourneret, 2010], where $N$ sensors record different linear mixtures of $M \leq N$ sparse source signals. One sample is acquired per discrete time index. The sparseness condition implies that only few sources are active for each time index. The recorded, i.e. observed, signals are given by the rows of the data matrix $\mathbf{X}$, the source signals are given by the rows of the coefficient matrix $\mathbf{A}$, the mixture coefficients for the individual source signals are given by the atoms of the dictionary $\mathbf{U}$. In this special setting one additionally assumes that columns of $\mathbf{U}$, which contain the mixture coefficients, are mutually orthonormal. Solving this blind source separation problem is ill-posed, meaning that $\mathbf{X}$ is given, whereas both $\mathbf{A}$ and $\mathbf{U}$ are unknown and have to be estimated.

### 2.2.3 Alternating Optimization

The sparse coding literature provides a considerable number of algorithms to learn sparse representations for a given data set. Generally, a joint optimization problem has to be solved which takes two terms into account. On the one hand, the approximation error of the training data set, which is commonly measured by the residual norm, shall be minimized. On the other hand, the sparsity of the data representation, which is commonly measured by the $\ell_0$-norm or $\ell_1$-norm, shall be maximized. Note that maximizing the sparsity of the data representation is equivalent to minimizing one of the aforementioned norms. Hence, two "forces" drive the optimization process in general. To jointly optimize the sparse coefficients and the dictionary is difficult [Rubinstein et al., 2010, Elad, 2010]. Therefore, an update scheme which alternately optimizes two kinds of subproblems is used to handle the nested optimization of the sparse model. One subproblem addresses the update of the sparse coefficients while the dictionary is fixed. The second subproblem addresses the update of the dictionary while the sparse coefficients are fixed.

When learning orthogonal dictionaries for sparse coding, as opposed to learning overcomplete ones, the first subproblem can be solved efficiently and optimally. Its solution depends on the sparse model. In the following two sections we introduce the two primary sparse models that occur in the literature: the constrained $K$-sparse model and the unconstrained regularized sparse model. For each model we give the corresponding optimal solutions to the first subproblem. Indeed, the second subproblem can be solved fast and optimally as well. However, performing alternating updates using the optimal solutions of both subproblems does not necessarily yield an optimal solution to the joint optimization problem, particularly if the sparsity is not very high, noise is present or a good initial dictionary is unknown. Therefore, we provide alternative strategies to solve the second subproblem, which can yield superior solutions to the joint optimization problem.

## 2.3 Constrained $K$-Sparse Model

In the following, we introduce the constrained $K$-sparse model to find for a given training data set $\mathbf{X}$ an optimal ONB $\mathbf{U}^*$ in which the training data samples $\mathbf{x}_i$ are optimally approximated by $K$-sparse coefficient vectors $\mathbf{a}_i^*$. Sparsity level $K \in \{1, \dots, N\}$ is a user parameter to control the sparsity of the representations.

The learning methods presented in this chapter are predominantly based on this model.

### 2.3.1 General Cost Function

Suppose a given data matrix $\mathbf{X}$ is represented by some (arbitrary) coefficient matrix $\mathbf{A}$ (of the same size) subject to an ONB $\mathbf{U}$. The cost function

$$\begin{aligned} E_{\mathbf{X}}(\mathbf{U}, \mathbf{A}) &= \|\mathbf{X} - \mathbf{U}\mathbf{A}\|_{\mathrm{F}}^2 && (2.5) \\ &= \sum_{i=1}^{L} \sum_{j=1}^{N} \left( \mathbf{X}_{j,i} - (\mathbf{U}\mathbf{A})_{j,i} \right)^2 && (2.6) \\ &= \sum_{i=1}^{L} \|\mathbf{x}_i - \mathbf{U}\mathbf{a}_i\|_2^2 && (2.7) \end{aligned}$$

assesses the inaccuracy of the representation by measuring the squared residual norm, i.e. the squared error between data matrix $\mathbf{X}$ and its approximation given by $\mathbf{U}\mathbf{A}$. Up to the constant factor $\frac{1}{L}$, the cost function is equivalent to the mean squared error (MSE) of the approximated samples.

### 2.3.2 Joint Optimization Problem

The joint optimization problem of the constrained $K$-sparse model is given by minimizing the cost function $E_{\mathbf{X}}(\mathbf{U}, \mathbf{A})$, as given by (2.5), regarding its two arguments with the constraint that the columns of $\mathbf{A}$ are $K$-sparse:

$$\mathrm{P}_{(2.8)}: \quad \underset{\mathbf{U} \in \mathcal{O}(N),\, \mathbf{A} \in \mathbb{R}^{N \times L}}{\arg\min} \|\mathbf{X} - \mathbf{U}\mathbf{A}\|_{\mathrm{F}}^2 \text{ , s.t. } \|\mathbf{a}_i\|_0 \le K \text{ for } i = 1, \dots, L\,. \quad (2.8)$$

### 2.3.3 Optimal Coefficient Update

Suppose ONB $\mathbf{U}$ is given, then $\mathrm{P}_{(2.8)}$ reduces to the batch learning variant of the $K$-sparse approximation problem:

$$\mathrm{P}_{(2.9)}: \qquad \underset{\mathbf{A}\in\mathbb{R}^{N\times L}}{\arg\min} \|\mathbf{X}-\mathbf{U}\mathbf{A}\|_{\mathrm{F}}^2 \ , \text{ s.t. } \|\mathbf{a}_i\|_0 \leq K \text{ for } i=1,\ldots,L\,. \tag{2.9}$$

Its solution gives the optimal (column-wise) $K$-sparse coefficient matrix $\mathbf{A}^*$ to approximate $\mathbf{X}$ in ONB $\mathbf{U}$ with minimal error and is denoted by $\mathcal{S}_K\left(\mathbf{U}^T\mathbf{X}\right)$. Solving $\mathrm{P}_{(2.9)}$ is equivalent to solve $L$ independent optimization problems of the form

$$\mathrm{P}_{(2.10)}: \qquad \underset{\mathbf{a}\in\mathbb{R}^{N}}{\arg\min} \|\mathbf{x}-\mathbf{U}\mathbf{a}\|_2^2 \ , \text{ s.t. } \|\mathbf{a}\|_0 \leq K\,, \tag{2.10}$$

one for each training data sample $\mathbf{x}_i$. $\mathrm{P}_{(2.10)}$ is the online learning variant of the $K$-sparse approximation problem. Its minimizer gives the optimal $K$-sparse coefficient vector to represent a single data sample $\mathbf{x}$ in ONB $\mathbf{U}$, and can be efficiently computed as follows:

*Remark.* For given data sample $\mathbf{x}$, ONB $\mathbf{U}$ and sparsity level $K$, let $h_1,\ldots,h_N$ be a sequence such that $\left(\mathbf{u}_{h_1}^T\mathbf{x}\right)^2 \geq \cdots \geq \left(\mathbf{u}_{h_N}^T\mathbf{x}\right)^2$. The $K$-sparse coefficient vector $\mathbf{a}^* = \mathcal{S}_K\left(\mathbf{U}^T\mathbf{x}\right)$, with entries

$$a_{h_k}^* = \begin{cases} \mathbf{u}_{h_k}^T\mathbf{x} & \text{if } k \leq K \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

is a global minimizer of $\mathrm{P}_{(2.10)}$.

*Proof.* First, let $\mathbf{a}$ be in the feasible set of $\mathrm{P}_{(2.10)}$, i.e. $\mathbf{a}$ is an arbitrary $K$-sparse coefficient vector. Assume $S \subseteq \{1,\ldots,N\}$ is the support of $\mathbf{a}$, where $|S| \leq K$. We have

$$\begin{aligned} \|\mathbf{x}-\mathbf{U}\mathbf{a}\|_2^2 &= \left\|\mathbf{x}-\sum_{j\in S} a_j\mathbf{u}_j\right\|_2^2 & (2.12)\\ &= \|\mathbf{x}\|_2^2 - 2\sum_{j\in S} a_j\mathbf{u}_j^T\mathbf{x} + \sum_{j\in S} a_j^2 & (2.13) \end{aligned}$$

Taking the partial derivative $\frac{\partial}{\partial a_j}$ of (2.13) and setting to zero, yields $a_j = \mathbf{u}_j^T\mathbf{x}$ for $j \in S$, and $0=0$ for $j \notin S$. Hence, any stationary point $\mathbf{a}^*$ of $\mathrm{P}_{(2.10)}$ requires non-zero coefficients of the form $a_j^* = \mathbf{u}_j^T \mathbf{x}$. Taking this into account, yields

$$\begin{aligned} \|\mathbf{x} - \mathbf{U}\mathbf{a}^*\|_2^2 &= \left\| \mathbf{x} - \sum_{j \in S} \left(\mathbf{u}_j^T \mathbf{x}\right) \mathbf{u}_j \right\|_2^2 && (2.14) \\ &= \|\mathbf{x}\|_2^2 - \sum_{j \in S} \left(\mathbf{u}_j^T \mathbf{x}\right)^2 . && (2.15) \end{aligned}$$

Hence, $S^* = \{h_1, \dots, h_K\}$ is optimal as no other support $S$ with $|S| \leq K$ can further decrease (2.15), which yields (2.11) as the global minimizer. Furthermore, the solution $\mathbf{a}^*$ is unique iff $\left(\mathbf{u}_{h_K}^T \mathbf{x}\right)^2 > \left(\mathbf{u}_{h_{K+1}}^T \mathbf{x}\right)^2$. □

In other words, an optimal solution $\mathbf{a}^*$ to the $K$-sparse approximation problem $\mathrm{P}_{(2.10)}$ can be efficiently determined by first computing the dense representation $\mathbf{a} = \mathbf{U}^T \mathbf{x}$, then retaining the $K$ entries $a_{h_1}, \dots, a_{h_K}$ with largest magnitude (e.g. via partial sorting), and setting the other $N - K$ entries $a_{h_{K+1}}, \dots, a_{h_N}$ to zero.

Another, sometimes more useful way to write the minimizer (2.11) of the $K$-sparse approximation problem (2.10) is given by

$$\mathbf{a}^* = \mathcal{S}_K(\mathbf{U}^T \mathbf{x}) = \mathbf{D}\mathbf{U}^T \mathbf{x} , \qquad (2.16)$$

where $\mathbf{D}$ is a diagonal matrix with $K$ entries equal to 1 and otherwise entries equal to 0. The locations of 1-entries on the diagonal correspond to the indices $h_1, \dots, h_K$ which select the $K$ largest squared projections $\left(\mathbf{u}_{h_1}^T \mathbf{x}\right)^2, \dots, \left(\mathbf{u}_{h_K}^T \mathbf{x}\right)^2$.

### 2.3.4 $K$-Sparse Approximation Error

Given a data matrix $\mathbf{X}$ and a sparsity level $K$, the cost function measuring the (optimal) $K$-sparse approximation error as a function of an ONB $\mathbf{U}$, is given by

$$\begin{aligned} E_{\mathbf{X},K}(\mathbf{U}) &= \left\| \mathbf{X} - \mathbf{U}\mathcal{S}_K(\mathbf{U}^T \mathbf{X}) \right\|_\mathrm{F}^2 && (2.17) \\ &= \|\mathbf{X}\|_\mathrm{F}^2 - \sum_{i=1}^{L} \mathbf{x}_i^T \mathbf{U}\mathbf{D}_i \mathbf{U}^T \mathbf{x}_i , && (2.18) \end{aligned}$$

where $\mathcal{S}_K(\mathbf{U}^T \mathbf{X})$ is the solution $\mathbf{A}^*$ to $\mathrm{P}_{(2.9)}$.

By (2.17) the subproblem of finding the optimal $K$-sparse representation of $\mathbf{X}$ is merged to a certain extent into the cost function. In the broader sense, solving $\mathrm{P}_{(2.8)}$ is equivalent to minimizing (2.17). When we assess the sparse encoding performance of an ONB $\mathbf{U}$, we evaluate $E_{\mathbf{X},K}(\mathbf{U})$ and call it the (total) costs of encoding $\mathbf{X}$ by its optimal $K$-sparse representation subject to $\mathbf{U}$.

The single sample (online) variant of the $K$-sparse approximation error is given by

$$\begin{aligned} E_{\mathbf{x},K}(\mathbf{U}) &= \left\|\mathbf{x} - \mathbf{U}\mathcal{S}_K(\mathbf{U}^T\mathbf{x})\right\|_2^2 && (2.19)\\ &= \|\mathbf{x}\|_2^2 - \sum_{k=1}^{K}(\mathbf{u}_{h_k}^T\mathbf{x})^2 && (2.20)\\ &= \|\mathbf{x}\|_2^2 - \mathbf{x}^T\mathbf{U}\mathbf{D}\mathbf{U}^T\mathbf{x}\,. && (2.21)\end{aligned}$$

## 2.4 Unconstrained Regularized Sparse Model

In the following, we introduce the unconstrained regularized sparse model. The key difference to the constrained $K$-sparse model is, that the sparsity inducing term is not tied as a side condition, but instead imposed as a regularization term on the objective function. Hence, the two driving forces of the optimization task, approximation error and sparsity of the representation, are linearly combined. Although the learning methods presented in this chapter are focussed on the constrained $K$-sparse model, the unconstrained regularized sparse model is relevant as well, as it has been addressed by other authors [Lesage et al., 2005, Bao et al., 2013, Bao et al., 2015, Sezer et al., 2008].

We emphasize that the methods proposed in Section 2.7 and Section 2.8 can be easily adapted for the unconstrained model by simply interchanging the coefficient update module. The coefficient update subproblem can be solved fast and optimally for both models, cf. Section 2.3.3 and Section 2.4.3 below.

### 2.4.1 Cost Function

The cost function of the unconstrained regularized sparse model is given by:

$$E_{\mathbf{X},\lambda}(\mathbf{U},\mathbf{A}) = \|\mathbf{X} - \mathbf{U}\mathbf{A}\|_\mathrm{F}^2 + \lambda\,\|\mathbf{A}\|_p\ , \qquad (2.22)$$

where $p \in \{0, 1\}$. Thus, the sparsity of the representation is measured by the matrix variant of either the $\ell_0$-norm or the $\ell_1$-norm. The sparsity term is weighted by a global regularization coefficient $\lambda$ in order to balance the sparsity of the representation relative to the approximation error.

### 2.4.2 Joint Optimization Problem

The joint optimization problem of the unconstrained regularized model is given by minimizing cost function $E_{\mathbf{X},\lambda}(\mathbf{U},\mathbf{A})$, as given by (2.22), regarding its two arguments.

$$\mathrm{P}_{(2.23)}: \qquad \underset{\mathbf{U}\in\mathcal{O}(N),\,\mathbf{A}\in\mathbb{R}^{N\times L}}{\arg\min}\ \|\mathbf{X} - \mathbf{U}\mathbf{A}\|_\mathrm{F}^2 + \lambda\,\|\mathbf{A}\|_p\ . \qquad (2.23)$$

### 2.4.3 Optimal Coefficient Update

Suppose ONB $\mathbf{U}$ is given, then $\mathrm{P}_{(2.23)}$ reduces to the batch variant of the regularized sparse approximation problem:

$$\mathrm{P}_{(2.24)} : \qquad \underset{\mathbf{A}\in\mathbb{R}^{N\times L}}{\arg\min} \left\|\mathbf{X} - \mathbf{U}\mathbf{A}\right\|_{\mathrm{F}}^2 + \lambda \left\|\mathbf{A}\right\|_p . \tag{2.24}$$

The optimal sparse coefficient matrix $\mathbf{A}^*$, which minimizes $\mathrm{P}_{(2.24)}$, is obtained by applying an element-wise sparsification operator $\mathcal{S}_{p,\lambda}(\cdot)$ to threshold the dense coefficient matrix $\mathbf{U}^T\mathbf{X}$. Dependent on the sparsity measure, i.e. dependent on $p$, the global thresholding operation is either hard or soft.

In the case $p = 0$, the minimizer of $\mathrm{P}_{(2.24)}$ is given by $\mathbf{A}^*$ with entries resulting from hard thresholding

$$A^*_{j,i} = \mathcal{S}_{0,\lambda}(\mathbf{u}_j^T\mathbf{x}_i) = \begin{cases} \mathbf{u}_j^T\mathbf{x}_i & \text{if } |\mathbf{u}_j^T\mathbf{x}_i| \geq \sqrt{\lambda} \\ 0 & \text{if } |\mathbf{u}_j^T\mathbf{x}_i| < \sqrt{\lambda} \end{cases} \tag{2.25}$$

[Bao et al., 2013, Sezer et al., 2008, Cai et al., 2014, Bao et al., 2015].

In the case $p = 1$, the minimizer of $\mathrm{P}_{(2.24)}$ is given by $\mathbf{A}^*$ with entries resulting from soft thresholding

$$A^*_{j,i} = \mathcal{S}_{1,\lambda}(\mathbf{u}_j^T\mathbf{x}_i) = \begin{cases} \mathbf{u}_j^T\mathbf{x}_i - \lambda/2 & \text{if } \mathbf{u}_j^T\mathbf{x}_i > \lambda/2 \\ 0 & \text{if } |\mathbf{u}_j^T\mathbf{x}_i| \leq \lambda/2 \\ \mathbf{u}_j^T\mathbf{x}_i + \lambda/2 & \text{if } \mathbf{u}_j^T\mathbf{x}_i < -\lambda/2 \end{cases} \tag{2.26}$$

[Lesage et al., 2005].

## 2.5 Literature Review

Some authors approached the problem of learning orthogonal dictionaries for sparse coding before.

Coifman et al. proposed the Wavelet Packet Transform [Coifman et al., 1990], which is an early attempt to enhance orthogonal transforms with a certain degree of adaptivity to the represented signal. For a given signal, it allows to select a basis from a large collection of dyadic time frequency atoms derived from a specific pair of mother wavelet and scaling function.

Mishali and Eldar addressed the constrained $K$-sparse problem, where the number of non-zero coefficients for each sample is modeled to be exactly $K$ rather than bounded by $K$ [Mishali and Eldar, 2009]. They proposed a method with two separate successive stages. The first stage aims to estimate the support pattern of the sparse coefficient matrix by inference exclusively based on data matrix $\mathbf{X}$. Locations of zero and non-zero coefficients are iteratively deduced by applying a small set of heuristic rules, such as $\mathbf{x}_i^T\mathbf{x}_j = 0 \Rightarrow \mathbf{x}_i$ and $\mathbf{x}_j$ have disjoint support. The resulting support pattern matrix $\mathbf{Z}$ estimated by this first stage is fixed and passed to the second stage, where the following alternating update scheme is conducted. $(i)$ The dense coefficient matrix is created via $\mathbf{A} = \mathbf{U}^T\mathbf{X}$. Subsequently, coefficients predicted to be zero, according to the support pattern estimate $\mathbf{Z}$, are set to zero leading to a sparse coefficient matrix $\mathbf{A}$. $(ii)$ The ONB $\mathbf{U}$ is updated by solving the Orthogonal Procrustes Problem (OPP) [Schönemann, 1966] as described in Section 2.6.1 below, using $\mathbf{A}$ resulting from step $(i)$. In [Mishali and Eldar, 2009] only low-dimensional synthetic data sets were investigated, and merely two quite high sparsity levels ($K \in \{2, 3\}$) were considered. The authors point out that the support recovery stage can be inaccurate. We can confirm this observation and found on synthetic data that this becomes an issue for the subsequent stage as ONB recovery capabilities are severely impaired if the sparsity level is lowered. Another issue with their first stage is the rigid requirement that the given data has an exactly $K$-sparse representation which does not tolerate small amplitude noise, and is therefore not applicable to real word data. For this reason, we can evaluate this approach only with noiseless synthetic data (see Section 2.9 below).

Lesage et al. proposed overcomplete dictionary learning for sparse coding, where the dictionary is a union of ONBs [Lesage et al., 2005]. The authors addressed the unconstrained regularized sparse model, as described in Section 2.4 ($p = 1$), and propose a customary alternating optimization scheme consisting of coefficient update and dictionary update. For dictionaries composed of unions of ONBs, the coefficient update problem can be relieved by implementing the well-known Basis Pursuit (BP) algorithm [Chen et al., 1998] more efficiently using Block Coordinate Relaxation. Their approach is evolved starting from the case where the dictionary is a single ONB, which justifies its consideration here. In the single ONB setting, the BP based coefficient update reduces to soft thresholding as described in Section 2.4.3. Moreover, ONB $\mathbf{U}$ is updated by solving the OPP, as described in Section 2.6.1 below, using $\mathbf{A}^*$ resulting from the soft thresholding step.

Sezer et al. proposed similarly an alternating optimization scheme to learn data-driven a set of multiple ONBs for sparse coding [Sezer et al., 2008, Sezer et al., 2015]. Their iterative method consists of three alternating stages: In the first stage, each data sample is assigned to the individual ONB that provides the lowest cost function value. The two subsequent stages, coefficient update stage and dictionary update stage, are then sequentially applied to the single ONBs using the correspondingly assigned data subset. The authors addressed the unconstrained regularized sparse model, as described in Section 2.4 ($p = 0$). For this model, the optimal coefficient update is given by hard thresholding as described in Section 2.4.3. Each individual ONB $\mathbf{U}_l$ is updated by solving the OPP, as described in Section 2.6.1 below, using $\mathbf{A}_l^*$ resulting from the hard thresholding step. Sezer et al. applied their method to natural image patches and observed ONBs emerging with selectivity to particular spatial directions prior to appropriate initializations. In image compression experiments, their method attained superior rate-distortion compared to the DCT.

Bao et al. proposed an alternating batch algorithm to learn an ONB to sparsely encode image patches [Bao et al., 2013]. The authors addressed the unconstrained regularized sparse model, as described in Section 2.4 ($p = 0$). Their proposed method is equivalent to [Sezer et al., 2008, Sezer et al., 2015] (if only one ONB would be learned rather than multiple ones) and to the tight frame learning approach proposed in [Cai et al., 2014] (the correspondence is pointed out more evidently in [Bao et al., 2015]). The main difference in [Bao et al., 2013] is the option that, in advance, a static subset of ONB atoms can be reserved which is not updated during learning. The remaining atoms of the ONB are learned subject to the orthogonal complement of the fixed ones in the same alternating iterative scheme consisting of hard thresholding and solving the OPP. The primary application in [Bao et al., 2013] was to learn ONBs on patches of corrupted images with the objective to solve image restoration problems.

Dobigeon and Tourneret proposed the hierarchical Bayesian model BOCA for learning undercomplete orthogonal dictionaries for sparse coding [Dobigeon and Tourneret, 2010]. BOCA relies on selecting suitable prior distributions for the unknown model parameters and hyperparameters. The authors model the sparse coefficients by a Bernoulli-Gaussian process and the dictionary by a uniform distribution on the Stiefel manifold. To estimate the hyperparameters, a second level of hierarchy is introduced in the Bayesian model. The joint posterior distribution of the unknown model parameters is approximated from samples generated by a Markov chain Monte Carlo (MCMC) method. The MCMC scheme is a partially collapsed Gibbs sampler.

Gribonval and Schnass considered the joint $\ell_1$-norm minimization problem with respect to the ONB and the coefficient matrix [Gribonval and Schnass, 2008]. Their main results are identifiability conditions that guarantee local convergence to the generating ONB. They showed that the Bernoulli-Gaussian model satisfies these conditions with high probability, provided that enough samples are given. However, an explicit algorithm is not proposed and the convergence relies on a good initialization.

Rusu et al. proposed an orthogonal dictionary learning method for sparse coding, where the ONB is composed by a product of few Householder reflectors [Rusu et al., 2016]. The main advantage of the proposed approach is its low computational complexity in terms of applying and manipulating the dictionary which implies a fast learning process. The number of reflectors balances the trade-off between computational complexity and accuracy of the sparse representation. Note that the fewer reflectors are used the more is the search space of candidate ONBs limited to subsets of $\mathcal{O}(N)$. The authors apply their approach to natural image data and investigate sparse approximation performance as well as image denoising capabilities. Merely very high sparsity levels, $K \in \{4, 6\}$, are considered for $8 \times 8$ image patches. This approach based on

Householder reflectors seems to yield inferior encoding performance compared to the Canonical Approach, which is introduced in the following section. Most recently, Rusu et al. proposed an alternative learning approach that is based on generalized Givens rotation [Rusu and Thompson, 2017].

## 2.6 Canonical Approach (CA)

The Canonical Approach (CA) is a batch learning procedure to minimize $\mathrm{P}_{(2.8)}$, the joint optimization problem of the constrained $K$-sparse model [Schütze et al., 2015, Schütze et al., 2016]. To each ONB update, all training data samples contribute simultaneously to reduce cost function (2.17). CA performs alternating minimization, i.e. the sparse coefficient matrix $\mathbf{A}$ is updated while ONB $\mathbf{U}$ is fixed and conversely, $\mathbf{U}$ is updated while $\mathbf{A}$ is fixed.

CA is related to orthogonal dictionary learning approaches which were previously proposed by other authors who addressed the unconstrained regularized sparse model [Lesage et al., 2005, Sezer et al., 2008, Bao et al., 2013, Cai et al., 2014, Sezer et al., 2015, Bao et al., 2015]. CA is the natural modification to the constrained $K$-sparse model [Schütze et al., 2016]. Particularly the coefficient update stage is different from the global thresholding, cf. Section 2.3.3 and Section 2.4.3. The same model modification, as given by CA, was later independently used by Rusu et al. (who credited [Lesage et al., 2005]) for the sake of baseline comparisons [Rusu et al., 2016].

For either model, the two subproblems, updating sparse coefficient matrix $\mathbf{A}$ and updating ONB $\mathbf{U}$, can be solved fast and optimally. However, using these globally optimal solutions to the subproblems in an alternating scheme does not guarantee to always minimize the joint problem $\mathrm{P}_{(2.8)}$ globally. This will become apparent in our numerical experiments presented in Section 2.9 and Section 2.10.

### 2.6.1 Dictionary Update

CA updates ONB $\mathbf{U}$ by solving the Orthogonal Procrustes Problem.

#### Orthogonal Procrustes Problem

The Orthogonal Procrustes Problem (OPP) is a matrix nearness problem in linear algebra which seeks, given two equal sized matrices $\mathbf{A}, \mathbf{X} \in \mathbb{R}^{N \times L}$, for an ONB $\mathbf{U} \in \mathrm{O}(N)$ which maps $\mathbf{A}$ in minimal distance to $\mathbf{X}$ in terms of the Frobenius metric:

$$\mathrm{P}_{(2.27)} : \qquad \underset{\mathbf{U} \in \mathrm{O}(N)}{\arg\min} \, \|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 \, . \tag{2.27}$$

The OPP has a unique closed form solution which is given by

$$\mathbf{U}^* = \mathbf{V}\mathbf{W}^T \, , \tag{2.28}$$

where $\mathbf{V}$ and $\mathbf{W}^T$ are the outer matrices of the singular value decomposition (SVD) of $\mathbf{X}\mathbf{A}^T = \mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T$ [Schönemann, 1966]. If for a given data set $\mathbf{X}$ the optimal sparse coefficients $\mathbf{A}$ subject to an unknown ONB $\mathbf{U}^*$ were known, then solving (2.27) would derive $\mathbf{U}^*$. See Section 2.5 for the previous usage of (2.28) in the context of orthogonal dictionary learning for sparse coding.

### 2.6.2 Complete Learning Algorithm

Algorithm 1 lists CA in pseudo code.

**Algorithm 1** Canonical Approach (CA)

```
Input: Training data set $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L) \in \mathbb{R}^{N \times L}$
    Total number of batch ONB updates $t_{\max}$
    Sparsity level $K$
    Initial ONB $\mathbf{U}_{(0)}$ (optional)
Output: ONB $\mathbf{U}$ minimizing $\mathrm{P}_{(2.8)}$
Initialize ONB $\mathbf{U}_{(0)}$ randomly if not supplied
for all $t = 1, ..., t_{\max}$ do
    Update the sparse coefficient matrix $\mathbf{A}_{(t)} \leftarrow \mathcal{S}_K \left(\mathbf{U}_{(t-1)}^T \mathbf{X}\right)$
    Compute the SVD $\mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T$ of $\mathbf{X}\mathbf{A}_{(t)}^T$
    Update the ONB $\mathbf{U}_{(t)} \leftarrow \mathbf{V}\mathbf{W}^T$
end for
$\mathbf{U} \leftarrow \mathbf{U}_{(t_{\max})}$
```

### 2.6.3 Computational Complexity

A great benefit of CA is its low computational complexity which enables a fast implementation. To update the sparse coefficients, first, the dense coefficient matrix $\mathbf{U}_{(t-1)}^T\mathbf{X}$ of the training data set $\mathbf{X}$ is computed subject to ONB $\mathbf{U}_{(t-1)}$, which requires $\mathcal{O}(LN^2)$ floating point operations (flops). Subsequently, the $N - K$ least important coefficients of each column are set to zero, which requires additionally $\mathcal{O}(LN)$ flops using a partial sorting algorithm [Chambers, 1971]. Updating the dictionary by solving the OPP requires to compute the product $\mathbf{X}\mathbf{A}_{(t)}^T$ ($\mathcal{O}(LN^2)$ flops), its SVD $\mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T$ as well as the product $\mathbf{U}_{(t)} = \mathbf{V}\mathbf{W}^T$ (both $\mathcal{O}(N^3)$ flops). Since commonly $L \gg N$, the dominating term for the computational complexity of a CA learning epoch, i.e. one batch update, is $\mathcal{O}(LN^2)$. Although CA is a batch learning procedure, since sample size $L$ contributes linearly to the complexity, one can interpret the corresponding "per sample complexity" as $\mathcal{O}(N^2)$ flops for comparisons with the following online learning methods.

## 2.7 Orthogonal Sparse Coding (OSC)

Orthogonal Sparse Coding (OSC) is an online learning procedure to minimize $\mathrm{P}_{(2.8)}$, the joint optimization problem of the constrained $K$-sparse model [Schütze et al., 2013, Schütze et al., 2016]. Each ONB update is done subject to a single data sample which is randomly selected from the training data set. OSC reduces the cost function (2.17) via stochastic descent using alternating minimization, i.e. updating the sparse coefficient vector $\mathbf{a}$ of the selected sample while ONB $\mathbf{U}$ is fixed and conversely, updating $\mathbf{U}$ while $\mathbf{a}$ is fixed.

### 2.7.1 Dictionary Update

OSC updates the atoms of ONB $\mathbf{U}$ sequentially using a Hebbian learning rule and Gram-Schmidt orthogonalization.

When a new training data sample $\mathbf{x}$ is drawn from $\mathbf{X}$, its dense representation $\mathbf{a} = \mathbf{U}^T\mathbf{x}$ is computed subject to the current, temporary fixed $\mathbf{U}$. Subsequently, sorting the squared entries of $\mathbf{a}$ yields an index sequence $h_1, \ldots, h_N$ such that $(\mathbf{u}_{h_1}^T\mathbf{x})^2 \geq \cdots \geq (\mathbf{u}_{h_N}^T\mathbf{x})^2$. Recall that the $K$-sparse approximation error of $\mathbf{U}$ subject to $\mathbf{x}$, i.e. the contribution to cost function (2.17), is given by $E_{\mathbf{x},K}(\mathbf{U}) = \|\mathbf{x}\|_2^2 - \sum_{k=1}^{K}(\mathbf{u}_{h_k}^T\mathbf{x})^2$. Consequently, the costs for $\mathbf{x}$ are reduced if $\mathbf{U}$ is modified such that the sum of the $K$ largest squared coefficients of $\mathbf{x}$ is increased. Loosely speaking, this requires to update $\mathbf{U}$ such that the sample energy $\|\mathbf{x}\|_2^2$ is more focussed on $\mathbf{u}_{h_1}, \ldots, \mathbf{u}_{h_K}$, the $K$ atoms which are most relevant to encode sample $\mathbf{x}$.

The index sequence $h_1, \ldots, h_N$ defines furthermore the order in which OSC updates the atoms, starting with $\mathbf{u}_{h_1}$ which contributes most to (2.21). Before an atom $\mathbf{u}_{h_k}$, $k \in \{1, \ldots, K\}$ is updated by the Hebbian learning rule, it is orthogonalized with respect to $\mathrm{span}(\{\mathbf{u}_{h_1}, ..., \mathbf{u}_{h_{k-1}}\})$, the span of atoms that were already updated in the current learning step, using Gram-Schmidt:

$$\mathbf{u}_{h_k} \leftarrow \mathbf{u}_{h_k} - \left(\mathbf{u}_{h_k}^T\mathbf{u}_{h_l}\right)\mathbf{u}_{h_l}\ , \ l = 1, \ldots, k-1\,. \tag{2.29}$$

This scheme of iterative Gram-Schmidt steps ensures that $\mathbf{U}$ remains an ONB when the update is finished. Subsequently, the orthogonalized atom $\mathbf{u}_{h_k}$ is updated via gradient descent subject to residual vector $\mathbf{x}_{\mathrm{res}}$, the original training sample $\mathbf{x}$ which is likewise orthogonalized to $\mathrm{span}(\{\mathbf{u}_{h_1}, ..., \mathbf{u}_{h_{k-1}}\})$, such that the cost contribution $-(\mathbf{u}_{h_k}^T\mathbf{x}_{\mathrm{res}})^2$ in (2.21) decreases. This leads to the Hebbian update rule

$$\begin{aligned} y &\leftarrow \mathbf{u}_{h_k}^T\mathbf{x}_{\mathrm{res}} && (2.30)\\ \Delta\mathbf{u}_{h_k} &\propto \frac{\partial}{\partial\mathbf{u}_{h_k}}\left(\mathbf{u}_{h_k}^T\mathbf{x}_{\mathrm{res}}\right)^2 && (2.31)\\ \Delta\mathbf{u}_{h_k} &\leftarrow \varepsilon_t \cdot y \cdot \mathbf{x}_{\mathrm{res}}\ , && (2.32) \end{aligned}$$

where $\varepsilon_t$ is the learning rate for the current learning step $t$, which cools down from $\varepsilon_{\text{init}}$ to $\varepsilon_{\text{final}}$ with increasing number of ONB updates. The updated atom $\mathbf{u}_{h_k}$ is normalized to unit Euclidean length.

Subsequently, $\mathbf{x}_{\text{res}}$ is orthogonalized with respect to the new $\mathbf{u}_{h_k}$ using Gram-Schmid:

$$\mathbf{x}_{\text{res}} \leftarrow \mathbf{x}_{\text{res}} - \left(\mathbf{x}_{\text{res}}^T \mathbf{u}_{h_k}\right) \mathbf{u}_{h_k} \, , \tag{2.33}$$

thus becoming the residual vector for the update of $\mathbf{u}_{h_{k+1}}$.

How many atoms are updated with learning rule (2.32) depends on the sparsity parameter $K$. Note, however, that due to the required orthogonality, all atoms are modified even if only $K$ were updated by (2.32). A learning step is complete when the last atom $\mathbf{u}_{h_N}$ has been normalized.

### 2.7.2 Complete Learning Algorithm

Algorithm 2 lists OSC in pseudo code.

### 2.7.3 Universality for Unknown Sparsity Levels

We have observed that OSC does not rely on receiving the "right" or an optimal value for user parameter $K$. When setting user parameter $K$ to $N$, OSC is able to learn a universal ONB $\tilde{\mathbf{U}}$ from the training data set $\mathbf{X}$ such that $\tilde{\mathbf{U}}$ minimizes the $K$-sparse approximation error $E_{\mathbf{X},K}(\tilde{\mathbf{U}})$ for many different sparsity levels $K$. More precisely, given a particular value $K_0$, the cost function value $E_{\mathbf{X},K_0}(\tilde{\mathbf{U}})$ is as small as $E_{\mathbf{X},K_0}(\mathbf{U})$, where $\mathbf{U}$ is learned by OSC using the matching sparsity level as user parameter, i.e. $K = K_0$. That the ONB learned by OSC for $K = N$ is universal, has been observed for several data sets such as synthetic data sets, where the ground-truth for $K$ is known, as well as for natural data, where a suitable $K$ might be unknown (see Section 2.9, Section 2.10 and Section 2.11). In the following, we distinguish the corresponding OSC variants by $K$-OSC and $N$-OSC.

### 2.7.4 Stochastic Descent

$N$-OSC is an online learning algorithm that updates sparse coding ONB $\mathbf{U}$ for each presented training data sample $\mathbf{x}$. The following theorem assures that such an $N$-OSC learning step increases the sparsity of the representation of $\mathbf{x}$. This can be proven for small learning rates $\varepsilon$, but seems to be valid for large $\varepsilon$ as well according to our numerical experiments.

**Theorem 1.** *Given an ONB* $\mathbf{U}$. *If learning rate* $\varepsilon > 0$ *is small enough, applying an* $N$*-OSC learning step to an arbitrary non-zero* $\mathbf{x}$ *yields a new ONB* $\mathbf{U}'$ *such that for the sequences* $(\mathbf{u}_{h_1}^T\mathbf{x})^2 \geq (\mathbf{u}_{h_2}^T\mathbf{x})^2 \geq ... \geq (\mathbf{u}_{h_N}^T\mathbf{x})^2$ *and* $(\mathbf{u}'^T_{h_1}\mathbf{x})^2 \geq (\mathbf{u}'^T_{h_2}\mathbf{x})^2 \geq ... \geq (\mathbf{u}'^T_{h_N}\mathbf{x})^2$

**Algorithm 2** Orthogonal Sparse Coding (OSC)

```
Input: Training data set $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L) \in \mathbb{R}^{N \times L}$
    Total number of online ONB updates $t_{\max}$
    Initial and final learning rate $\varepsilon_{\text{init}} \geq \varepsilon_{\text{final}}$
    Sparsity level $K$ (optional, default $K = N$, see Section 2.7.3)
    Initial ONB $\mathbf{U}_{(0)}$ (optional)
Output: ONB $\mathbf{U}$ minimizing $\mathrm{P}_{(2.8)}$
Initialize ONB $\mathbf{U}_{(0)}$ randomly if not supplied
for all $t = 0, ..., t_{\max}$ do
    Set the learning rate for the current learning step $\varepsilon_t \leftarrow \varepsilon_{\text{init}} (\varepsilon_{\text{final}}/\varepsilon_{\text{init}})^{t/t_{\max}}$
    Select data sample $\mathbf{x}$ from $\mathbf{X}$ uniformly at random, and set $\mathbf{x}_{\text{res}} \leftarrow \mathbf{x}$
    Determine sequence $h_1, ..., h_N$ such that $\left(\mathbf{u}_{h_1}^T \mathbf{x}\right)^2 \geq ... \geq \left(\mathbf{u}_{h_N}^T \mathbf{x}\right)^2$
    Update $\mathbf{U}_{(t)}$ to $\mathbf{U}_{(t+1)}$ as follows:
    for all $k = 1, ..., N$ do
        for all $l = 1, ..., (k-1)$ do
            Orthogonalize the current atom $\mathbf{u}_{h_k}$ subject to the previously updated atom
              $\mathbf{u}_{h_l}$ by a Gram-Schmidt step
                $$\mathbf{u}_{h_k} \leftarrow \mathbf{u}_{h_k} - \left(\mathbf{u}_{h_k}^T \mathbf{u}_{h_l}\right) \mathbf{u}_{h_l}$$
        end for
        if $k \leq K$ then
            Apply the Hebbian learning rule to the current atom $\mathbf{u}_{h_k}$
                $$\mathbf{u}_{h_k} \leftarrow \mathbf{u}_{h_k} + \varepsilon_t \cdot \left(\mathbf{u}_{h_k}^T \mathbf{x}_{\text{res}}\right) \mathbf{x}_{\text{res}}$$
        end if
        Normalize $\mathbf{u}_{h_k}$ to unit Euclidean length
        Orthogonalize residual $\mathbf{x}_{\text{res}}$ subject to the current atom $\mathbf{u}_{h_k}$ by a Gram-Schmidt
        step
                $$\mathbf{x}_{\text{res}} \leftarrow \mathbf{x}_{\text{res}} - \left(\mathbf{u}_{h_k}^T \mathbf{x}_{\text{res}}\right) \mathbf{u}_{h_k}$$
    end for
end for
$\mathbf{U} \leftarrow \mathbf{U}_{(t_{\max})}$
```

*the ordering*

$$\frac{(\mathbf{u}'^T_{h_{k+1}} \mathbf{x})^2}{(\mathbf{u}'^T_{h_k} \mathbf{x})^2} \leq \frac{(\mathbf{u}^T_{h_{k+1}} \mathbf{x})^2}{(\mathbf{u}^T_{h_k} \mathbf{x})^2} \tag{2.34}$$

*holds for all* $k = 1, ..., N-1$.

*Proof.* In the following, we use the assumption that $\varepsilon$ is small. We develop all expressions up to first order in $\varepsilon$ and treat terms of order $\varepsilon^2$ and higher as vanishing.

Without any loss of generality, we assume $(\mathbf{u}_1^T \mathbf{x})^2 \geq (\mathbf{u}_2^T \mathbf{x})^2 \geq ... \geq (\mathbf{u}_N^T \mathbf{x})^2$ which defines the order of basis vector updates. Each $\mathbf{u}_k$, except for $\mathbf{u}_1$, is updated in two steps. First, the Gram-Schmidt orthogonalization

$$\mathbf{v}_k = \mathbf{u}_k - \sum_{l=1}^{k-1} (\mathbf{u}_l'^T \mathbf{u}_k)\mathbf{u}_l' \,, \tag{2.35}$$

followed by the normalized Hebbian main update

$$\mathbf{u}_k' = \frac{\mathbf{v}_k + \varepsilon(\mathbf{v}_k^T \mathbf{x}_k)\mathbf{x}_k}{\left\|\mathbf{v}_k + \varepsilon(\mathbf{v}_k^T \mathbf{x}_k)\mathbf{x}_k\right\|_2} \,, \tag{2.36}$$

where

$$\mathbf{x}_k = \mathbf{x} - \sum_{l=1}^{k-1} (\mathbf{u}_l'^T \mathbf{x})\mathbf{u}_l' \,. \tag{2.37}$$

Atom $\mathbf{u}_1$ is only updated by (2.36) due to (2.35). In that sense $\mathbf{v}_1 = \mathbf{u}_1$ and $\mathbf{x}_1 = \mathbf{x}$ due to (2.37).

We will show by induction that

$$\mathbf{v}_k = \mathbf{u}_k - \varepsilon(\mathbf{u}_k^T \mathbf{x}) \sum_{l=1}^{k-1} (\mathbf{u}_l^T \mathbf{x})\mathbf{u}_l + \mathcal{O}(\varepsilon^2) \,. \tag{2.38}$$

Note that by (2.38) it holds $\|\mathbf{v}_k\|_2 \approx 1 + \mathcal{O}(\varepsilon^2)$. Hence, the Taylor expansion of update step (2.36) up to first order in $\varepsilon$ is

$$\mathbf{u}_k' = \mathbf{v}_k + \varepsilon(\mathbf{v}_k^T \mathbf{x}_k)(\mathbf{x}_k - (\mathbf{v}_k^T \mathbf{x}_k)\mathbf{v}_k) + \mathcal{O}(\varepsilon^2) \,. \tag{2.39}$$

Note that (2.36) is a Oja learning rule, i.e. a Hebbian learning rule with a normalization constraint. We apply the same expansion as in Section 4 of [Oja, 1982].

Furthermore, since $\mathbf{v}_k^T \mathbf{x}_k = \mathbf{v}_k^T \mathbf{x}$ and with (2.38) we have $\mathbf{v}_k^T \mathbf{x}_k = \mathbf{u}_k^T \mathbf{x} + \mathcal{O}(\varepsilon)$ as well as $(\mathbf{v}_k^T \mathbf{x}_k)\mathbf{v}_k = (\mathbf{u}_k^T \mathbf{x})\mathbf{u}_k + \mathcal{O}(\varepsilon)$. Hence, (2.39) can be restated as

$$\mathbf{u}_k' = \mathbf{v}_k + \varepsilon \left(\mathbf{u}_k^T \mathbf{x}\right) \left(\mathbf{x}_k - \left(\mathbf{u}_k^T \mathbf{x}\right) \mathbf{u}_k\right) + \mathcal{O}(\varepsilon^2) \,. \tag{2.40}$$

We will now show (2.38) by induction.

*Initial Step* $k = 1$. According to (2.35), we have by definition $\mathbf{v}_1 = \mathbf{u}_1$ which satisfies (2.38).

*Induction Step* $(k-1) \to k$. According to (2.35), we have by definition

$$\mathbf{v}_k \;\; = \;\; \mathbf{u}_k - \sum_{l=1}^{k-1} (\mathbf{u}_l'^T \mathbf{u}_k)\mathbf{u}_l' \,.$$

Due to induction hypothesis (2.38), $\mathbf{u}_l'$ can be restated according to (2.40). In addition to (2.38) we will use $\mathbf{u}_l^T \mathbf{u}_k = 0$ and $\mathbf{v}_l^T \mathbf{u}_k = \mathcal{O}(\varepsilon^2)$ as well as $\mathbf{u}_k^T \mathbf{x}_l = \mathbf{u}_k^T \mathbf{x} + \mathcal{O}(\varepsilon)$.

$$\begin{aligned}
\mathbf{v}_k &= \mathbf{u}_k - \sum_{l=1}^{k-1} \left[ (\mathbf{v}_l + \varepsilon(\mathbf{u}_l^T \mathbf{x})(\mathbf{x}_l - (\mathbf{u}_l^T \mathbf{x})\mathbf{u}_l))^T \mathbf{u}_k \right] \mathbf{u}'_l + \mathcal{O}(\varepsilon^2) \\
&= \mathbf{u}_k - \varepsilon(\mathbf{u}_k^T \mathbf{x}) \sum_{l=1}^{k-1} (\mathbf{u}_l^T \mathbf{x}) \mathbf{u}'_l + \mathcal{O}(\varepsilon^2) \\
&= \mathbf{u}_k - \varepsilon(\mathbf{u}_k^T \mathbf{x}) \sum_{l=1}^{k-1} (\mathbf{u}_l^T \mathbf{x})(\mathbf{v}_l + \varepsilon(\mathbf{u}_l^T \mathbf{x})(\mathbf{x}_l - (\mathbf{u}_l^T \mathbf{x})\mathbf{u}_l)) + \mathcal{O}(\varepsilon^2) \\
&= \mathbf{u}_k - \varepsilon(\mathbf{u}_k^T \mathbf{x}) \sum_{l=1}^{k-1} (\mathbf{u}_l^T \mathbf{x}) \mathbf{u}_l + \mathcal{O}(\varepsilon^2)
\end{aligned}$$

The induction is complete.

Combining (2.40) and (2.38) gives us up to first order in $\varepsilon$

$$\mathbf{u}'_k = \mathbf{u}_k + \varepsilon(\mathbf{u}_k^T \mathbf{x}) \left( \mathbf{x}_k - \sum_{l=1}^{k} (\mathbf{u}_l^T \mathbf{x}) \mathbf{u}_l \right) .$$

Hence, for small $\varepsilon$ and with (2.37) we obtain

$$\begin{aligned}
\frac{(\mathbf{u'}_{k+1}^T \mathbf{x})^2}{(\mathbf{u'}_k^T \mathbf{x})^2} &= \frac{(\mathbf{u}_{k+1}^T \mathbf{x})^2}{(\mathbf{u}_k^T \mathbf{x})^2} \frac{\left(1 + \varepsilon \left[ \mathbf{x}_{k+1}^T \mathbf{x} - \sum_{l=1}^{k+1} (\mathbf{u}_l^T \mathbf{x})^2 \right] \right)^2}{\left(1 + \varepsilon \left[ \mathbf{x}_k^T \mathbf{x} - \sum_{l=1}^{k} (\mathbf{u}_l^T \mathbf{x})^2 \right] \right)^2} \\
&= \frac{(\mathbf{u}_{k+1}^T \mathbf{x})^2}{(\mathbf{u}_k^T \mathbf{x})^2} \frac{\left(1 + \varepsilon \left[ ||\mathbf{x}||^2 - \sum_{l=1}^{k} (\mathbf{u}_l'^T \mathbf{x})^2 - \sum_{l=1}^{k+1} (\mathbf{u}_l^T \mathbf{x})^2 \right] \right)^2}{\left(1 + \varepsilon \left[ ||\mathbf{x}||^2 - \sum_{l=1}^{k-1} (\mathbf{u}_l'^T \mathbf{x})^2 - \sum_{l=1}^{k} (\mathbf{u}_l^T \mathbf{x})^2 \right] \right)^2} \\
&\leq \frac{(\mathbf{u}_{k+1}^T \mathbf{x})^2}{(\mathbf{u}_k^T \mathbf{x})^2} ,
\end{aligned}$$

since the square bracket in the nominator is smaller than the square bracket in the denominator. □

Theorem 1 states that an $N$-OSC update decreases the magnitude of each coefficient relative to its predecessor in the sequence of sorted coefficients. This means, that after the learning step the squared coefficients obey a stronger decay.

Figure 2.1 illustrates the squared coefficients of an image patch $\mathbf{x}$ in an ONB $\mathbf{U}_{(t)}$ as well as the squared coefficients of $\mathbf{x}$ in $\mathbf{U}_{(t+1)}$, i.e. after an update of $\mathbf{U}_{(t)}$ by $N$-OSC. Note that both curves integrate to the same value as the energy of $\mathbf{x}$ is preserved under any orthonormal transformation, i.e. $\|\mathbf{x}\|_2^2 = \left\|\mathbf{U}^T \mathbf{x}\right\|_2^2$. It can be seen that after the $N$-OSC update, more energy is distributed over less coefficients and that the magnitude of the less encoding relevant atoms shrinks. Thus, the sparsity of the representation of $\mathbf{x}$ is increased.

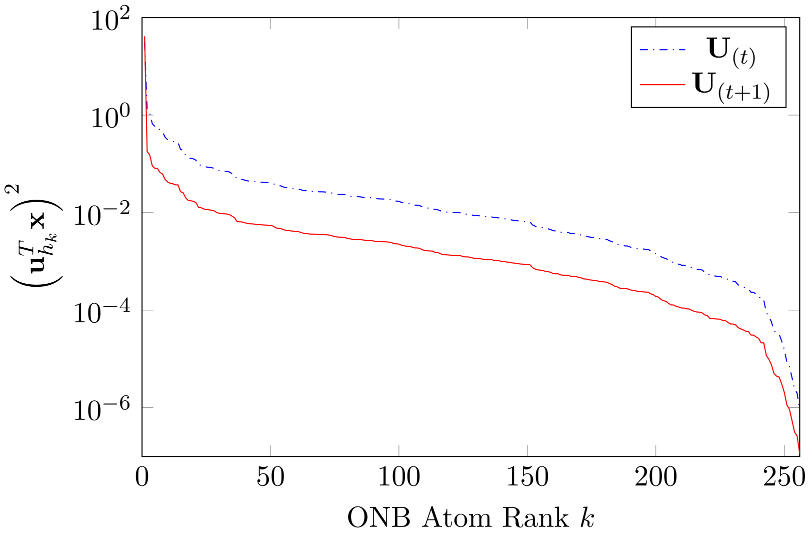


Figure 2.1: Squared coefficients of a natural image patch $\mathbf{x}$ ($N = 256$) in an ONB $\mathbf{U}_{(t)}$ and in ONB $\mathbf{U}_{(t+1)}$ due to an $N$-OSC update.

From Theorem 1 follows directly

**Corollary 1.** *Given an ONB* $\mathbf{U}$. *Applying an* $N$*-OSC learning step subject to an arbitrary* $\mathbf{x}$ *leads to an* $\mathbf{U}'$ *such that for each* $K = 1, ..., N$

$$\begin{aligned} & -\sum_{k=1}^{K}(\mathbf{u}'^T_{h_k}\mathbf{x})^2 && \leq && -\sum_{k=1}^{K}(\mathbf{u}^T_{h_k}\mathbf{x})^2 \\ \Leftrightarrow \quad & \|\mathbf{x}\|_2^2 - \sum_{k=1}^{K}(\mathbf{u}'^T_{h_k}\mathbf{x})^2 && \leq && \|\mathbf{x}\|_2^2 - \sum_{k=1}^{K}(\mathbf{u}^T_{h_k}\mathbf{x})^2 \\ \Leftrightarrow \quad & E_{\mathbf{x},K}(\mathbf{U}') && \leq && E_{\mathbf{x},K}(\mathbf{U})\,. \end{aligned} \tag{2.41}$$

This means that an $N$-OSC learning step reduces the costs (2.21) that the presented sample contributes to the total costs (2.17).

Unfortunately, this result does not imply that $N$-OSC minimizes cost function (2.17) for the entire training data set $\mathbf{X}$. However, in general such a global descent cannot be proven for online learning algorithms as each update blinds out the costs of all the other training data samples.

Theorem 1 and Corollary 1 are useful to realize that $N$-OSC performs a stochastic descent of cost function (2.17) similar to a stochastic gradient descent. $N$-OSC converges to an ONB $\mathbf{U}$ which yields on average small costs for each training data sample due to the online learning scheme and the cooling learning rate.

From an experiment with natural image patches (see Section 2.10), Figure 2.2 illustrates for a complete $N$-OSC learning phase the $K$-sparse approximation error (2.17) subject to ONB $\mathbf{U}_{(t)}$ as a function of the number of ONB updates. It can be seen that OSC performs a stochastic descent of the cost function (2.17).

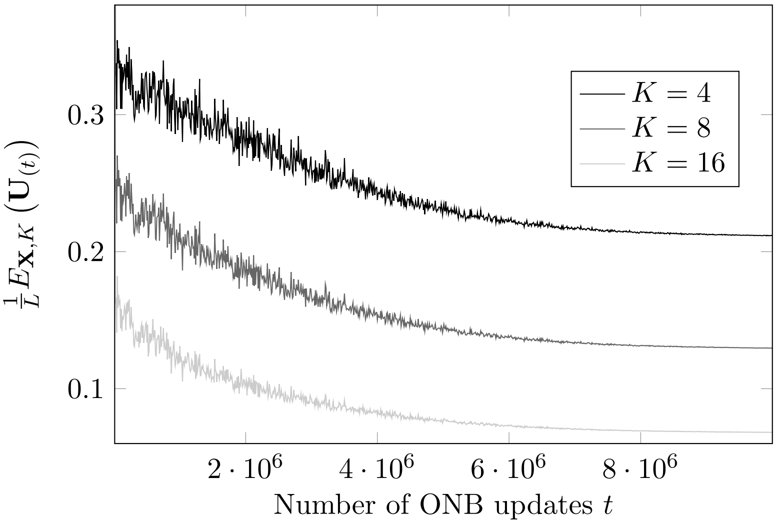


Figure 2.2: $K$-sparse approximation error $\frac{1}{L}E_{\mathbf{X},K}(\mathbf{U}_{(t)})$ as a function of $t$, the number of ONB updates by $N$-OSC for a learning phase on the NSSiVS data set.

### 2.7.5 Computational Complexity

Drawing a training data sample $\mathbf{x}$, setting residual vector $\mathbf{x}_{\text{res}}$ (line 4) and sorting the coefficients (line 5) requires $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ flops, respectively. The loop in lines 7-17 iterates over all $N$ atoms $\mathbf{u}_{h_k}$. The Gram-Schmidt steps for each $\mathbf{u}_{h_k}$ (lines 8-10) have a complexity of at most $\mathcal{O}(N^2)$. A single Hebbian update of a $\mathbf{u}_{h_k}$ (line 12), the length normalization of $\mathbf{u}_{h_k}$ (line 14), and the update of $\mathbf{x}_{\text{res}}$ (line 15) require $\mathcal{O}(N)$ flops. Altogether, the dominating term of the computational complexity of an ONB update by OSC is $\mathcal{O}(N^3)$.

## 2.8 Geodesic Flow Orthogonal Sparse Coding (GF-OSC)

Geodesic Flow Orthogonal Sparse Coding (GF-OSC) is an online learning procedure to minimize $\mathrm{P}_{(2.8)}$, the joint optimization problem of the constrained $K$-sparse model [Schütze et al., 2015]. Analogous to OSC, each ONB update is done subject to a randomly selected sample $\mathbf{x}$ from the training data set. GF-OSC reduces the cost function (2.17) via stochastic gradient descent using alternating minimization, i.e. updating the sparse coefficient vector $\mathbf{a}$ of the selected sample while the ONB $\mathbf{U}$ is fixed and conversely, updating $\mathbf{U}$ while $\mathbf{a}$ is fixed.

### 2.8.1 Dictionary Update

GF-OSC updates $\mathbf{U}$ rotationally via a multiplication with another ONB $\Delta\mathbf{U}$. The update is equivalent to a gradient descent step within the $\frac{N(N-1)}{2}$-dimensional space of free ONB parameters and is derived from the geodesic flow optimization framework.

#### Geodesic Flow Optimization Framework

In general, minimizing a scalar-valued cost function with respect to a square $N \times N$ matrix is an optimization problem in an $N^2$-dimensional search space. If, in addition, an orthonormality constraint is incorporated, the search space can be considerably reduced because any orthonormal $N \times N$ matrix has merely $\frac{N(N-1)}{2}$ degrees of freedom. For this kind of optimization problems, Plumbley proposed the geodesic flow framework [Plumbley, 2004] which exploits the reduced search space. Suppose the corresponding cost function is differentiable, then the geodesic flow approach allows to derive its gradient within the reduced space of free parameters, and therefore gradient based optimization techniques can be deployed to minimize the cost function.

The geodesic flow approach is restricted to the subgroup $\mathrm{SO}(N)$ as it is not possible to go smoothly from $\mathrm{SO}(N)$ to $\overline{\mathrm{SO}(N)}$ or vice versa. $\mathrm{SO}(N)$ forms a Lie group with an associated Lie algebra given by the set of skew-symmetric matrices, $\mathfrak{so}(N) = \{\mathbf{B} \in \mathbb{R}^{N\times N} \mid \mathbf{B}^T = -\mathbf{B}\}$ and the Lie bracket given by the matrix commutator $[\mathbf{Q},\mathbf{R}] = \mathbf{QR} - \mathbf{RQ}$. Since $\mathrm{SO}(N)$ is a matrix Lie group, the matrix exponential $\exp(\mathbf{B}) = \sum_{n=0}^{\infty} \frac{\mathbf{B}^n}{n!}$ provides a surjective mapping from $\mathfrak{so}(N)$ to $\mathrm{SO}(N)$ and we have $\mathbf{U}\mathbf{U}^T = \exp(\mathbf{B})\left(\exp(\mathbf{B})\right)^T = \exp(\mathbf{B})\exp(-\mathbf{B}) = \mathbf{I}_N$. Let $E_{\mathbf{x},K} : \mathbb{R}^{N\times N} \to \mathbb{R}$ be the differentiable cost function that is to be optimized under the orthogonality constraint. By using the gradient $\nabla_{\mathbf{U}} E_{\mathbf{x},K}$, the gradient of $E_{\mathbf{x},K}$ with respect to the Lie algebra $\mathfrak{so}(N)$ is derived as follows:

$$\nabla_{\mathbf{B}} E_{\mathbf{x},K} = \left(\nabla_{\mathbf{U}} E_{\mathbf{x},K}\right)\mathbf{U}^T - \mathbf{U}\left(\nabla_{\mathbf{U}} E_{\mathbf{x},K}\right)^T . \tag{2.42}$$

The geodesic flow approach starts with some initial $\mathbf{U}_{(0)}$ and optimizes $\mathbf{U}_{(t)}$ sequentially according to the iteration variable $t = 1, ..., t_{\max}$. For the most recent $\mathbf{U}_{(t-1)}$ an adaptation within $\mathfrak{so}(N)$ into the steepest descent direction $\Delta\mathbf{B} = -\varepsilon\nabla_{\mathbf{B}} E_{\mathbf{x},K}$ is determined by (2.42), where $\varepsilon$ is a sufficiently small learning rate. This adaptation within $\mathfrak{so}(N)$ is mapped to $\mathrm{SO}(N)$ by the matrix exponential, i.e., $\Delta\mathbf{U} = \exp(\Delta\mathbf{B})$. Subsequently, the adaptation within $\mathrm{SO}(N)$ is applied rotationally to $\mathbf{U}_{(t-1)}$, thus providing the new orthogonal matrix $\mathbf{U}_{(t)} = (\Delta\mathbf{U})\,\mathbf{U}_{(t-1)}$. This iterative scheme enables the minimization of a scalar-valued cost function subject to the $\mathrm{SO}(N)$ and is based on a gradient descent in $\mathfrak{so}(N)$, which is the space of the underlying degrees of freedom. Each gradient descent step yields naturally a new ONB $\mathbf{U}_{(t)}$. As a consequence, reimposing the orthogonality constraint separately is dispensable.

### Derivation of the Update Rule

Suppose $\mathbf{x}$ is the current training data sample randomly selected from $\mathbf{X}$ and sparsity level $K$ is given. In order to derive the ONB update rule using the geodesic flow framework, we first derive the gradient of the cost function $E_{\mathbf{x},K}(\mathbf{U})$ given by (2.21) which measures the $K$-sparse approximation error of $\mathbf{x}$ subject to $\mathbf{U}$:

$$\begin{aligned} \nabla_{\mathbf{U}} E_{\mathbf{x},K} &= \frac{\partial}{\partial \mathbf{U}} \left( \|\mathbf{x}\|_2^2 - \mathbf{x}^T \mathbf{U}\mathbf{D}\mathbf{U}^T \mathbf{x} \right) && (2.43) \\ &= -\frac{\partial}{\partial \mathbf{U}} \mathbf{x}^T \mathbf{U}\mathbf{D}\mathbf{U}^T \mathbf{x} && (2.44) \\ &= -2\mathbf{x}\mathbf{x}^T \mathbf{U}\mathbf{D} \, . && (2.45) \end{aligned}$$

Subsequently, we insert (2.45) into (2.42) to obtain the desired gradient $\nabla_{\mathbf{B}} E_{\mathbf{x},K}$ of the cost function (2.21) with respect to the Lie algebra $\mathfrak{so}(N)$. Note that the derived $\nabla_{\mathbf{B}} E_{\mathbf{x},K}$ is the key ingredient of our GF-OSC algorithm and that it can be simplified as follows:

$$\nabla_{\mathbf{B}} E_{\mathbf{x},K} \propto \hat{\mathbf{x}}\mathbf{x}^T - \mathbf{x}\hat{\mathbf{x}}^T, \qquad (2.46)$$

where $\hat{\mathbf{x}} = \mathbf{U}\mathbf{a}^* = \mathbf{U}\mathcal{S}_K(\mathbf{U}^T\mathbf{x}) = \mathbf{U}\mathbf{D}\mathbf{U}^T\mathbf{x}$ is the optimal $K$-term approximation of the sample $\mathbf{x}$ subject to $\mathbf{U}$.

## 2.8.2 Complete Learning Algorithm

Algorithm 3 lists GF-OSC in pseudo code.

**Algorithm 3** GF-OSC

```
Input: Training data set $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L) \in \mathbb{R}^{N \times L}$
    Total number of online ONB updates $t_{\max}$
    Sparsity level $K$
    Initial ONB $\mathbf{U}_{(0)}$ (optional)
Output: ONB $\mathbf{U} \in \mathrm{SO}(N)$ minimizing $\mathrm{P}_{(2.8)}$
Initialize ONB $\mathbf{U}_{(0)} \in \mathrm{SO}(N)$ randomly if not supplied
for all $t = 1, ..., t_{\max}$ do
    Select a sample $\mathbf{x}$ from $\mathbf{X}$ randomly
    Compute $\hat{\mathbf{x}}$, the optimal $K$-term approximation of $\mathbf{x}$ subject to $\mathbf{U}_{(t-1)}$
    Compute $\nabla_{\mathbf{B}} E_{\mathbf{x},K}$ according to (2.46)
    Select a suitable learning rate $\varepsilon_t$
    $\Delta\mathbf{B} \leftarrow -\varepsilon_t \nabla_{\mathbf{B}} E_{\mathbf{x},K}$
    $\Delta\mathbf{U} \leftarrow \exp(\Delta\mathbf{B})$
    $\mathbf{U}_{(t)} \leftarrow (\Delta\mathbf{U})\,\mathbf{U}_{(t-1)}$
end for
$\mathbf{U} \leftarrow \mathbf{U}_{(t_{\max})}$
```

### Selecting Learning Rate $\varepsilon_t$

To update $\mathbf{U}$ by GF-OSC, different strategies can be chosen for selecting learning rate $\varepsilon_t$. It seems natural to apply a dynamic learning rate which cools down from a large initial value $\varepsilon_{\text{init}}$ to a small final value $\varepsilon_{\text{final}}$ over the number of applied ONB updates. We propose, similar to the learning rate of OSC, an exponential decay of the form

$$\varepsilon_t \leftarrow \varepsilon_{\text{init}} \left( \frac{\varepsilon_{\text{final}}}{\varepsilon_{\text{init}}} \right)^{\frac{t}{t_{\max}}} . \tag{2.47}$$

In principle, any suitable technique for selecting the step length during a gradient descent can be used. We observed, for instance, that the convergence of GF-OSC can be improved for synthetic noise free data if the learning rate $\varepsilon_t$ is adaptively calculated via backtracking line search based on the Armijo-Goldstein condition [Armijo, 1966]. For natural image patches, however, a cooling learning rate yields better results. Note that sophisticated line search techniques increase the computational load as each additional evaluation of the cost function requires a mapping from $\mathfrak{so}(N)$ to $\mathrm{SO}(N)$.

### 2.8.3 Computational Complexity

Selecting training data sample $\mathbf{x}$ (line 3), computing its optimal $K$-sparse approximation $\hat{\mathbf{x}}$ subject to $\mathbf{U}_{(t-1)}$ (line 4) via partial sorting [Chambers, 1971], calculating $\nabla_{\mathbf{B}} E_{\mathbf{x},K}$ (line 5) and $\Delta\mathbf{B}$ (line 7) requires $\mathcal{O}(N^2)$ flops. Applying the matrix exponential on $\Delta\mathbf{B}$ to get $\Delta\mathbf{U}$ (line 8) and performing the rotational update $(\Delta\mathbf{U})\,\mathbf{U}_{(t-1)}$ requires $\mathcal{O}(N^3)$ flops [Moler and Loan, 2003]. Thus, the total complexity of a single GF-OSC update is $\mathcal{O}(N^3) + \mathcal{O}(\text{select } \varepsilon_t)$ flops.

### 2.8.4 Remarks on a GF-OSC Batch Update Rule

Formulating a GF-OSC batch update rule is straight forward. The gradient $\nabla_{\mathbf{B}} E_{\mathbf{X},K}$ of the batch cost function (2.17) is given by summing (2.46) over all the training data samples $\mathbf{x}_i$:

$$\nabla_{\mathbf{B}} E_{\mathbf{X},K} \quad \propto \quad \sum_{i=1}^{L} \hat{\mathbf{x}}_i \mathbf{x}_i^T - \mathbf{x}_i \hat{\mathbf{x}}_i^T . \tag{2.48}$$

Fixing the sparse coefficient matrix $\mathbf{A}^* = \mathcal{S}_K(\mathbf{U}^T\mathbf{X})$, expanding (2.17) directly using the definition of the Frobenius norm and the Frobenius inner product as well as deploying $\frac{\partial}{\partial \mathbf{U}} \mathrm{tr}(\mathbf{X}^T\mathbf{U}\mathbf{A}^*) = -2\mathbf{X}\mathbf{A}^{*T}$ yields, analogous to (2.46), more compactly

$$\nabla_{\mathbf{B}} E_{\mathbf{X},K} \quad \propto \quad \hat{\mathbf{X}}\mathbf{X}^T - \mathbf{X}\hat{\mathbf{X}}^T . \tag{2.49}$$

We experimented with the batch variant of GF-OSC as well as a corresponding mini batch variant on synthetic noiseless data. We observed that these variants were inferior to the proposed online variant of GF-OSC.

## 2.9 ONB Recovery from Synthetic Data

In this section we investigate the performance of the proposed learning methods at the task to recover a reference ONB from synthetic data sets obeying the constrained $K$-sparse model. We study both a noiseless scenario, where data samples are strictly $K$-sparse, as well as a noisy scenario, where $K$-sparse data samples are contaminated by additive isotropic Gaussian noise. To analyze ONB recovery performance, we measure the similarity between a learned ONB with the reference ONB which was used to generate the synthetic data sets. Furthermore, we will investigate the cost descent in terms of the number of learning epochs.

An inevitable degree of ambiguity is inherent to the problem. Since ONB $\mathbf{U}$ and coefficient matrix $\mathbf{A}$ are unknown in the problem formulation $\mathrm{P}_{(2.8)}$, their estimation cannot be unique regarding the atom order or the atom signs. Any permutation or sign switch, simultaneously applied to both the columns of $\mathbf{U}$ as well as the rows of $\mathbf{A}$, yields the same data matrix $\mathbf{X} = \mathbf{U}\mathbf{A}$. Thus, a correspondence matching is necessary to align the atoms of the estimated ONB with the reference ONB, which enables an automated performance analysis (see Section 2.9.3 below).

We compare

- K-SVD [Aharon et al., 2006]
- OCA [Mishali and Eldar, 2009]
- CA [Schütze et al., 2015, Schütze et al., 2016] (see Algorithm 1 and also [Rusu and Thompson, 2017, Lesage et al., 2005])
- OSC [Schütze et al., 2013, Schütze et al., 2016] (see Algorithm 2)
- GF-OSC [Schütze et al., 2015] (see Algorithm 3)

All these learning methods comply with the ONB learning task $\mathrm{P}_{(2.8)}$ of the constrained $K$-sparse model, except for K-SVD which is an algorithm for learning non-orthogonal (commonly overcomplete) dictionaries and does therefore not exploit the orthogonality condition. A comparison with K-SVD is nonetheless justifiable for two reasons. First, K-SVD does not rely by design on $M > N$, hence, a complete dictionary can be learned instead of an overcomplete one. Second, orthogonality is a good-natured scenario for K-SVD, because the mutual coherence of the reference dictionary is minimal. To each synthetic data set, OSC is applied in two variants. In one case, user parameter $K$ is set to the true generating sparsity level, which is called $K$-OSC in the following. In the other case, the true sparsity level is assumed to be unknown and user parameter $K$ is set to $N$, which is called $N$-OSC in the following.

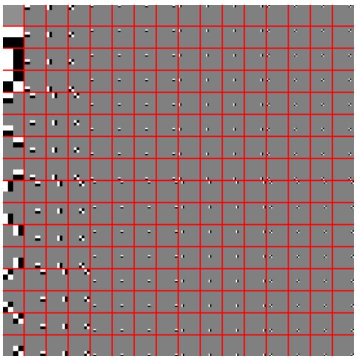
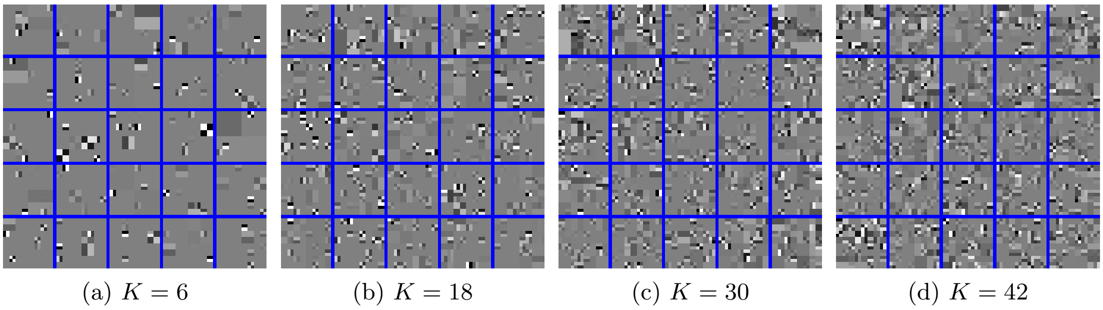


Figure 2.3: The orthogonal non-standard 2D Haar wavelet basis ($N = 256$), the reference ONB for the recovery experiments. Each basis vector is visualized as a $16 \times 16$ patch. For display purposes, the entries of each basis patch (except the DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

### 2.9.1 Noiseless Synthetic $K$-Sparse Data

For signal dimensionality $N = 256$ and sample size $L = 1000$, synthetic data sets were generated for various sparsity levels $K \in \{2, 6, ..., 58, 62\}$. Each data sample was generated to have a strictly $K$-sparse representation in the reference ONB. We selected the orthogonal non-standard 2D Haar wavelet basis as the ground truth. The attribute "2D" refers to the fact that the original domain of the data has two spatial dimensions, i.e. dictionary atoms and synthetic data samples can be reshaped and visualized as $16 \times 16$ patches. The reference ONB is illustrated in Figure 2.3.

First, for each sample to be synthesized the support pattern of its coefficient vector was generated, i.e., the $K$ locations of non-zero coefficients were selected uniformly at random. Second, the $K$ non-zero coefficients were drawn randomly from a standard Gaussian distribution. The data samples were synthesized by multiplying the $K$-sparse coefficient vectors with the reference ONB. Figure 2.4 illustrates exemplarily synthetic



(a) $K = 6$

(b) $K = 18$

(c) $K = 30$

(d) $K = 42$

Figure 2.4: Synthetic data samples being $K$-sparse in the non-standard 2D Haar wavelet basis ($N = 256$), noiseless scenario. Locations of non-zero coefficients are uniformly distributed, values of non-zero coefficients are standard Gaussian distributed. Each sample is visualized as a $16 \times 16$ patch. For display purposes, the entries of each patch are shifted to have zero mean and are subsequently scaled to unit supremum norm.

data samples of different sparsity levels for the noiseless scenario. For each sparsity level we created 10 different data sets to be able to measure deviations of recovery results over multiple runs. In total, 160 data sets were generated for the ONB recovery experiment. Furthermore, we generated for each data set an initial random ONB $\mathbf{U}_{(0)}$, such that each learning method starts the optimization from the same initial point.

### 2.9.2 Noisy Synthetic $K$-Sparse Data

To investigate the robustness of the learning methods at the ONB recovery task, we contaminated the 160 data sets, generated as described in Section 2.9.1, by 5 dB additive isotropic Gaussian noise. To this end, we computed for each data set $\mathbf{X}$ the average spatial variance $\sigma_{\mathbf{X}}^2$. Subsequently, a noise matrix $\mathbf{G} \in \mathbb{R}^{N \times L}$ was generated with entries randomly drawn i.i.d. from the Gaussian distribution $\mathcal{N}(0, \sigma_{\mathbf{G}})$, where $\sigma_{\mathbf{G}} = \sqrt{\sigma_{\mathbf{X}}^2 \cdot 10^{-\frac{1}{2}}}$. The noisy data set was obtained by adding $\mathbf{G}$ to $\mathbf{X}$. Figure 2.5 illustrates exemplarily synthetic data samples of different sparsity levels for the noisy scenario.

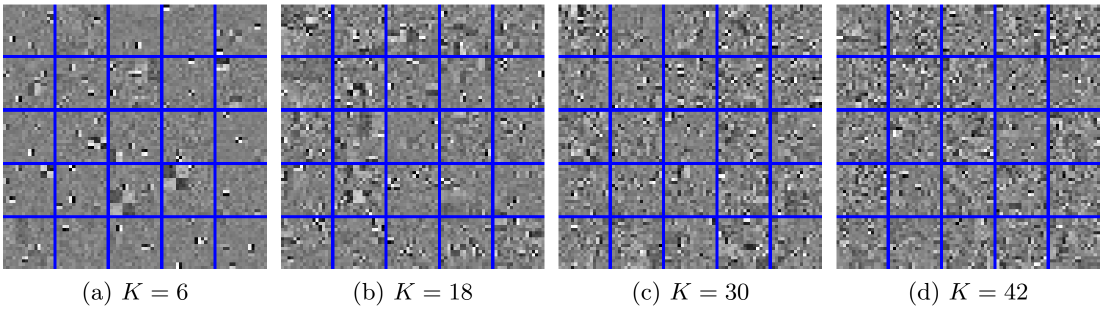


Figure 2.5: Synthetic data samples being $K$-sparse in the non-standard 2D Haar wavelet basis ($N = 256$), noisy scenario (5 dB additive Gaussian noise). Locations of non-zero coefficients are uniformly distributed, values of non-zero coefficients are standard Gaussian distributed. Each sample is visualized as a $16 \times 16$ patch. For display purposes, the entries of each patch are shifted to have zero mean and are subsequently scaled to unit supremum norm.

### 2.9.3 Performance Measures

Prior to measuring the ONB recovery performance for a synthetic data set, we apply a correspondence matching to align the atoms of the estimated ONB with the atoms of the reference ONB. We determine the best matching pairs of atoms between the two dictionaries using the following greedy strategy. First, all overlaps of all $N^2$ possible atom pairs are sorted in descending order. Subsequently, the $N$ best matching pairs were assigned according to that sequence such that each atom is assigned exactly once. Thus, the atoms of the learned ONB and the atoms of the reference ONB obey a one-to-one assignment. The overlaps of the $N$ matched atom pairs (matched overlaps) reflect the similarity between the learned and the reference ONB in terms of the cosine of angles. The reference ONB is perfectly recovered iff all matched overlaps are equal to one. For a single learned ONB, the most detailed assessment of recovery performance can be made by inspecting the distribution of the $N$ matched overlaps by a histogram. Alternatively, a scalar-valued recovery performance measure is given by computing either the mean matched overlap (MMO), or by counting the relative number of matched overlaps exceeding a particular threshold (recovery rate). Note that the described procedure also allows to measure the ONB recovery performance for a non-orthogonal dictionary, e.g. resulting from the K-SVD algorithm, as long as it has the same size as the reference ONB.

### 2.9.4 Choices of User Parameter Values

For each learning method (except for the $N$-OSC variant), user parameter $K$ was set to the true generating sparsity level, and 1000 learning epochs were allowed to be iterated. However, learning was terminated prematurely as soon as the smallest matched overlap exceeded 0.99. For K-SVD, we set the number of atoms $M = N$ to learn a complete dictionary. For OSC, the learning rate decreases exponentially from an initial value $\varepsilon_{\text{init}}$ to a final value $\varepsilon_{\text{final}}$. As we could not exclude a priori a dependence of OSC on appropriately chosen values for $\varepsilon_{\text{init}}$ and $\varepsilon_{\text{final}}$, we examined OSC convergence empirically for several combinations on one independently generated noiseless data set with sparsity level $K = 18$. The investigated set of learning rate combinations spanned a wide range of scales: $(\varepsilon_{\text{init}}, \varepsilon_{\text{final}}) \in \{10^1, 10^0, 10^{-1}, 10^{-2}\} \times \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. We observed different slopes of the cost descent. The final costs were similarly small for most combinations, given $\varepsilon_{\text{final}}$ was sufficiently small (see Figure 2.6). We selected the combination $\varepsilon_{\text{init}} = 10^{-1}$ and $\varepsilon_{\text{final}} = 10^{-4}$ as it yields the steepest initial slope with almost the smallest final cost value among the tested combinations, while the decay still is spread over all learning epochs without saturations. This OSC learning parameter combination was adopted for all the other synthetic data sets, too. OSC was applied to each synthetic data set in two variants. In one variant, user parameter $K$ is set to the true generating sparsity level ($K$-OSC). In the other variant, the true sparsity level is assumed to be unknown and user parameter $K$ is set to $N$ ($N$-OSC). The GF-OSC learning rates were selected via backtracking line search based on the Armijo-Goldstein condition, where $c = \tau = \frac{1}{2}$ as proposed in [Armijo, 1966], and $\alpha = 5$. We did not validate the latter parameter as it appeared to be a conservative upper bound for the learning rate. The OSC and GF-OSC parameters were used unaltered also for the noisy synthetic data sets.

### 2.9.5 Results for the Noiseless Scenario

For the rather high sparsity level $K = 18$ ($\approx 7\%$ non-zero coefficients) and the rather low sparsity level $K = 42$ ($\approx 16.4\%$ non-zero coefficients) we provide a detailed empirical

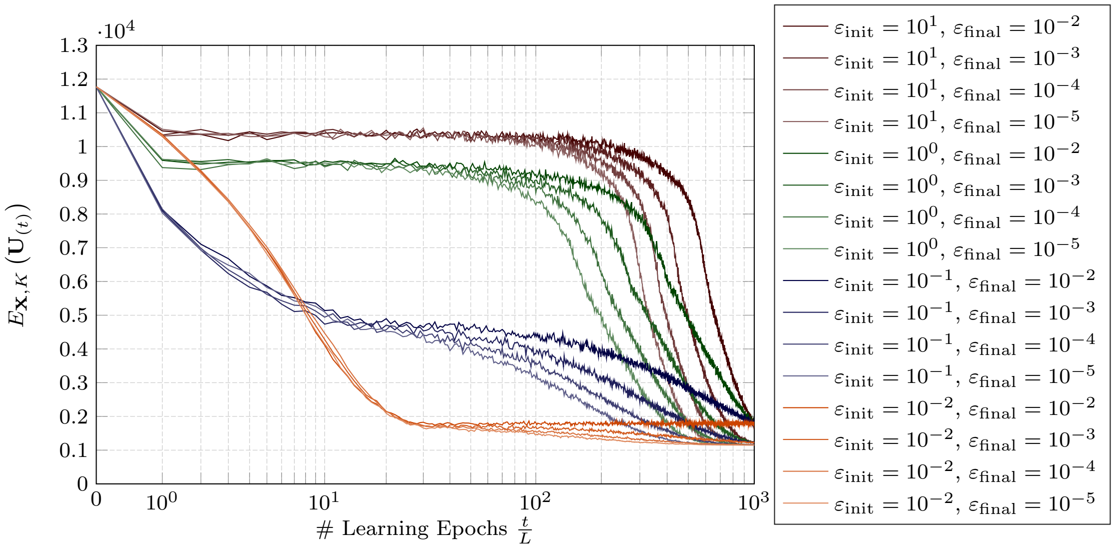


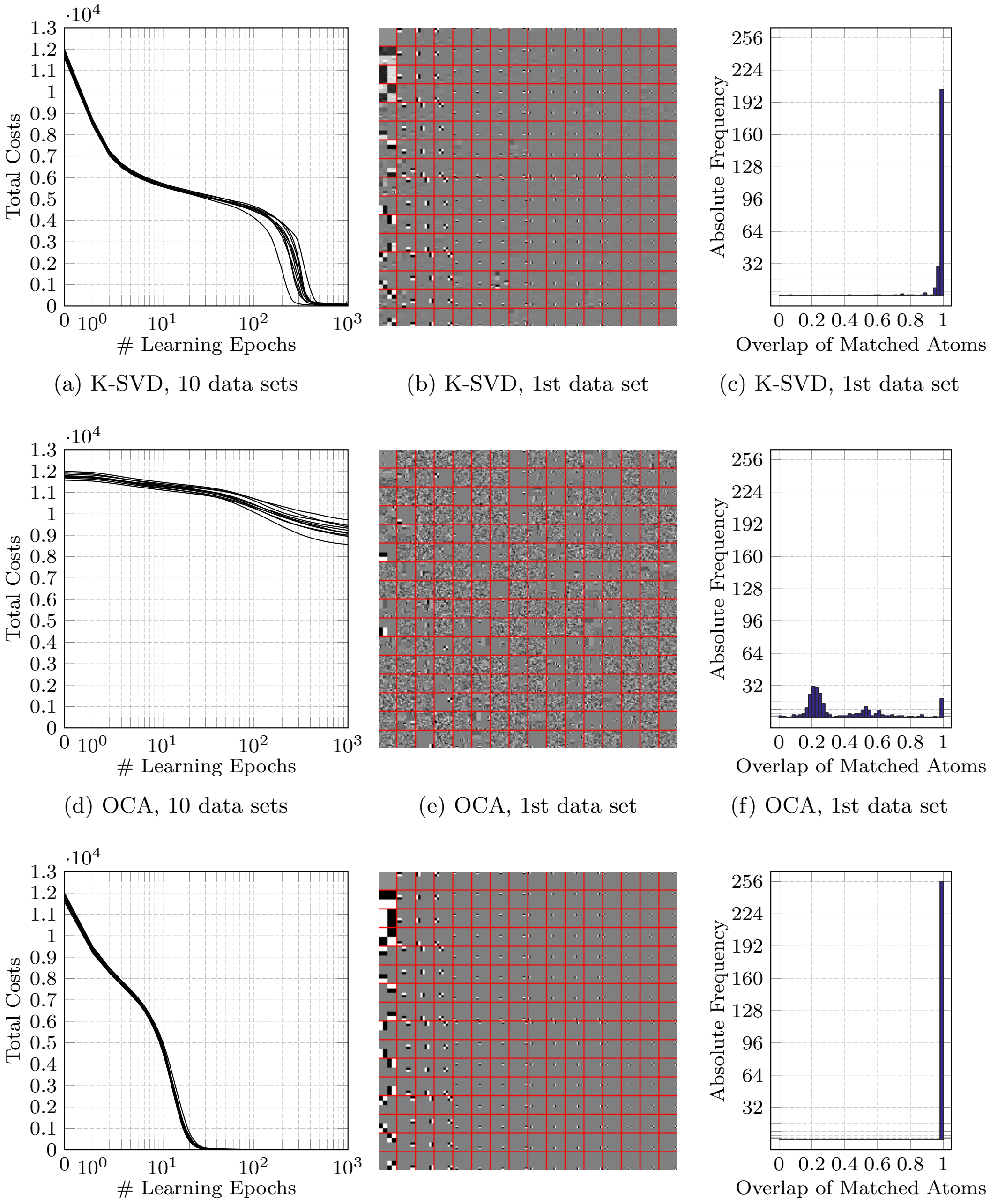
Figure 2.6: Validation of initial and final learning rate combinations $(\varepsilon_{\text{init}}, \varepsilon_{\text{final}}) \in \{10^1, 10^0, 10^{-1}, 10^{-2}\} \times \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for a $K$-OSC learning phase on a synthetic data set with sparsity level $K = 18$. The $K$-sparse approximation error $E_{\mathbf{X},K}(\mathbf{U}_{(t)})$ is plotted as a function of $t/L$, the number of (completed) learning epochs.

ONB recovery performance analysis. Respectively, Figure 2.7 – Figure 2.10 illustrate for each learning method (rows) three characteristics of the optimization process (columns). First, the convergence is shown empirically in terms of the $K$-sparse approximation error (the costs) as a function of the number of learning epochs (left column). Each of the ten black curves corresponds to one of the ten data sets with the same sparsity level. For the noiseless scenario, recovering the reference ONB perfectly implies a cost function value of zero. Second, for the first of the ten data sets per $K$, the atoms of the learned dictionary are visualized as patches (middle column). To facilitate comparisons with the reference ONB, the learned atoms are shown in the same order as in Figure 2.3 due to the correspondence matching. Third, for the same dictionary, the distribution of the $N$ matched overlaps is illustrated (right column). Each histogram consists of 50 bins covering the interval $[0, 1]$. Thus, each bin has a width of 0.02. The more accurate the reference ONB is recovered, the more concentrated is the distribution of matched overlaps on the right hand side. If all matched overlaps are contained in the most right bin, the reference ONB is (close to) perfectly recovered since the smallest matched overlap then is at least 0.98.

### Detailed Recovery Performance Analysis, $K = 18$ ($\approx 7\%$ Non-Zero Coefficients)

K-SVD reduces the costs to the global minimum, where a major leap from an intermediate cost level to the final cost level occurs between learning epochs 100 and 400 (see Figure 2.7a). As the K-SVD dictionary is not orthogonal, Optimized Orthogonal

Matching Pursuit (OOMP) [Rebollo-Neira and Lowe, 2002] is used to derive the $K$-



(a) K-SVD, 10 data sets

(b) K-SVD, 1st data set

(c) K-SVD, 1st data set

(d) OCA, 10 data sets

(e) OCA, 1st data set

(f) OCA, 1st data set

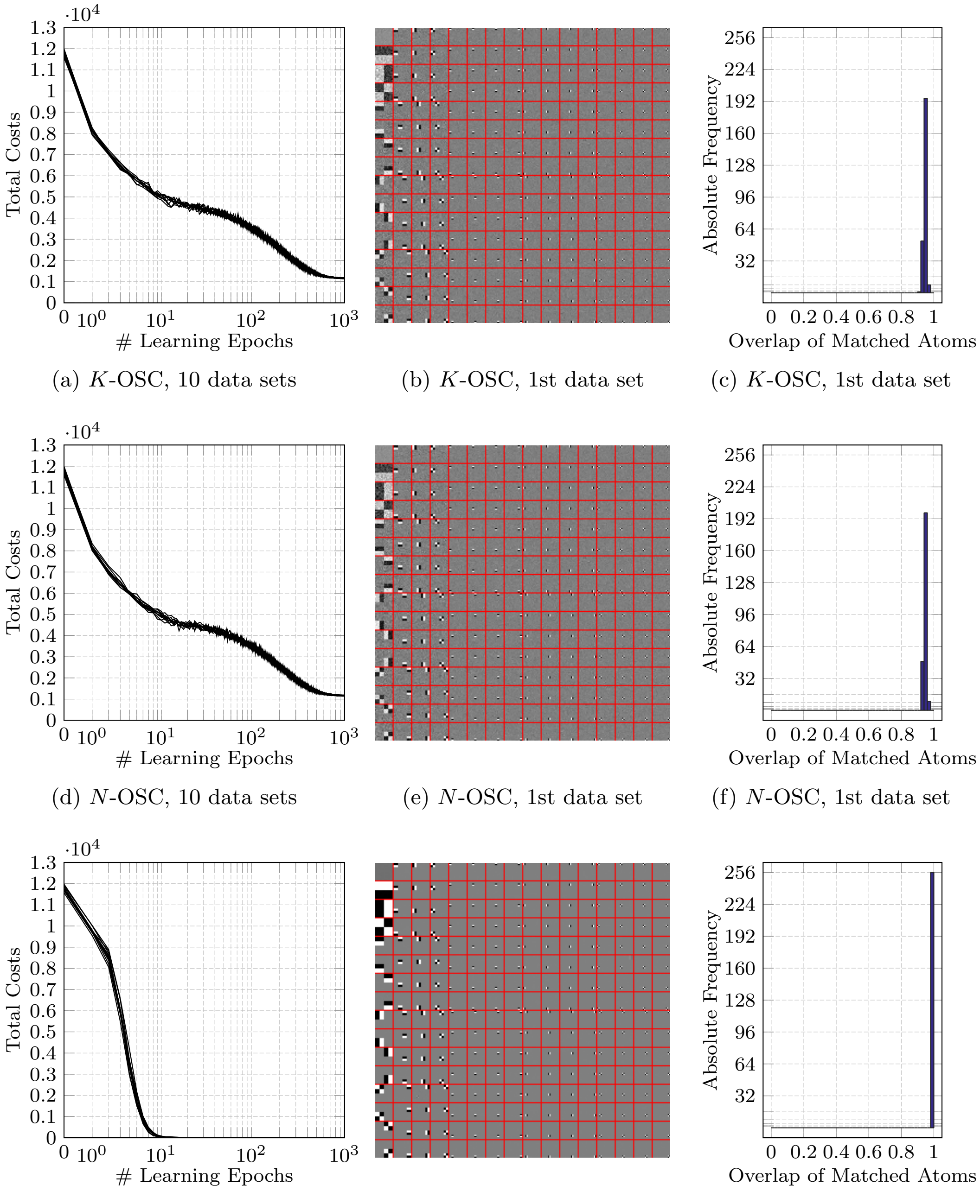
(g) CA, 10 data sets

(h) CA, 1st data set

(i) CA, 1st data set

Figure 2.7: Detailed ONB recovery performance analysis for the noiseless synthetic $K$-sparse data of sparsity level $K = 18$ ($\approx 7\%$ non-zero coefficients). The results obtained by K-SVD, OCA and CA are shown row-wise. In the left column, the $K$-sparse approximation error is plotted for the 10 data sets as a function of the number of learning epochs. In the middle column, the dictionary learned on the 1st of the 10 data sets is visualized according to the correspondence matching with the reference ONB. In the right column, a histogram shows the resulting distribution of matched overlaps.

sparse approximations of the training data samples. The learned K-SVD dictionary,



(a) $K$-OSC, 10 data sets

(b) $K$-OSC, 1st data set

(c) $K$-OSC, 1st data set

(d) $N$-OSC, 10 data sets

(e) $N$-OSC, 1st data set

(f) $N$-OSC, 1st data set

(g) GF-$K$-OSC, 10 data sets

(h) GF-$K$-OSC, 1st data set

(i) GF-$K$-OSC, 1st data set

Figure 2.8: Detailed ONB recovery performance analysis for the noiseless synthetic $K$-sparse data of sparsity level $K = 18$ ($\approx 7\%$ non-zero coefficients). The results obtained by $K$-OSC, $N$-OSC and GF-OSC are shown row-wise. In the left column, the $K$-sparse approximation error is plotted for the 10 data sets as a function of the number of learning epochs. In the middle column, the dictionary learned on the 1st of the 10 data sets is visualized according to the correspondence matching with the reference ONB. In the right column, a histogram shows the resulting distribution of matched overlaps.

although it is not orthogonal, resembles the reference ONB. However, most of the K-SVD atoms reveal, beside the dominating appearance of one reference atom, additional shadows of further reference atoms as well. This indicates that mixtures of reference atoms are learned by K-SVD. Despite this non-perfect recovery of the reference ONB, the training data samples can be well approximated by $K$-sparse combinations from the K-SVD dictionary as the final costs are close to zero. The bulk of matched overlaps is larger than 0.94, but a few are quite small (less than 0.7).

OCA decreases the costs only by a small amount. Even after one thousand learning epochs, the cost function value of the dictionary is far away from the global minimum. OCA, particularly its support estimation stage, appears to require much higher sparsity levels to succeed. Only a few atoms of the reference ONB are accurately recovered by OCA. The remaining atoms seem to be either mixtures of multiple reference atoms or look entirely noisy, which might indicate that they remain at their initial random state and are not sufficiently involved during the learning phase. The matched overlaps are distributed over the entire interval $[0, 1]$. Only 25 of the 256 reference atoms seem to be accurately recovered with matched overlaps in the interval $[0.8, 1]$. A large bulk of matched atoms is centered at 0.2, which corresponds to the atom patches showing random structure.

CA accomplishes a steep monotonic descent of the cost function, and achieves costs at the global minimum after only 12 learning epochs. The ONB estimated by CA looks indeed identical to the reference ONB. All atom patches are free of noise-like patterns or shadows from additional reference atoms. The matched overlaps fall entirely into the most right bin, and thus are all in $[0.98, 1]$. Hence, the ONB recovery result is highly accurate.

The costs during OSC learning have a clear descending trend, although the cost descent is not exclusively monotonic on a fine scale. An online update is specific only to a single training data sample. Hence, short sequences of such online updates do not necessarily improve the costs for the entire training data set. Nevertheless, a considerable cost reduction is achieved on the long run. OSC does not reach the global minimum of the cost function. The learned ONB, however, distinctly resembles the reference ONB without any shadow patterns from multiple reference atoms. However, minor inhomogeneous patterns are visible on the learned atom patches, which are likely responsible for the final cost residuals. The distribution of matched overlaps is noticeably localized and centered at 0.94. All overlaps are contained in the interval $[0.9, 0.98]$. Remarkably, these results are obtained for both OSC variants, $K$-OSC and $N$-OSC.

GF-OSC globally minimizes the costs in about ten learning epochs. Despite its online update scheme, the total cost descent is monotonic and steep, indicating that short update sequences reduce the costs for the entire training data set. the ONB learned by GF-OSC looks indeed identical to the reference ONB. The corresponding matched overlaps fall into the most right bin, and thus are all in $[0.98, 1]$.

### Detailed Recovery Performance Analysis, $K = 42$ ($\approx 16.4\%$ Non-Zero Coefficients)

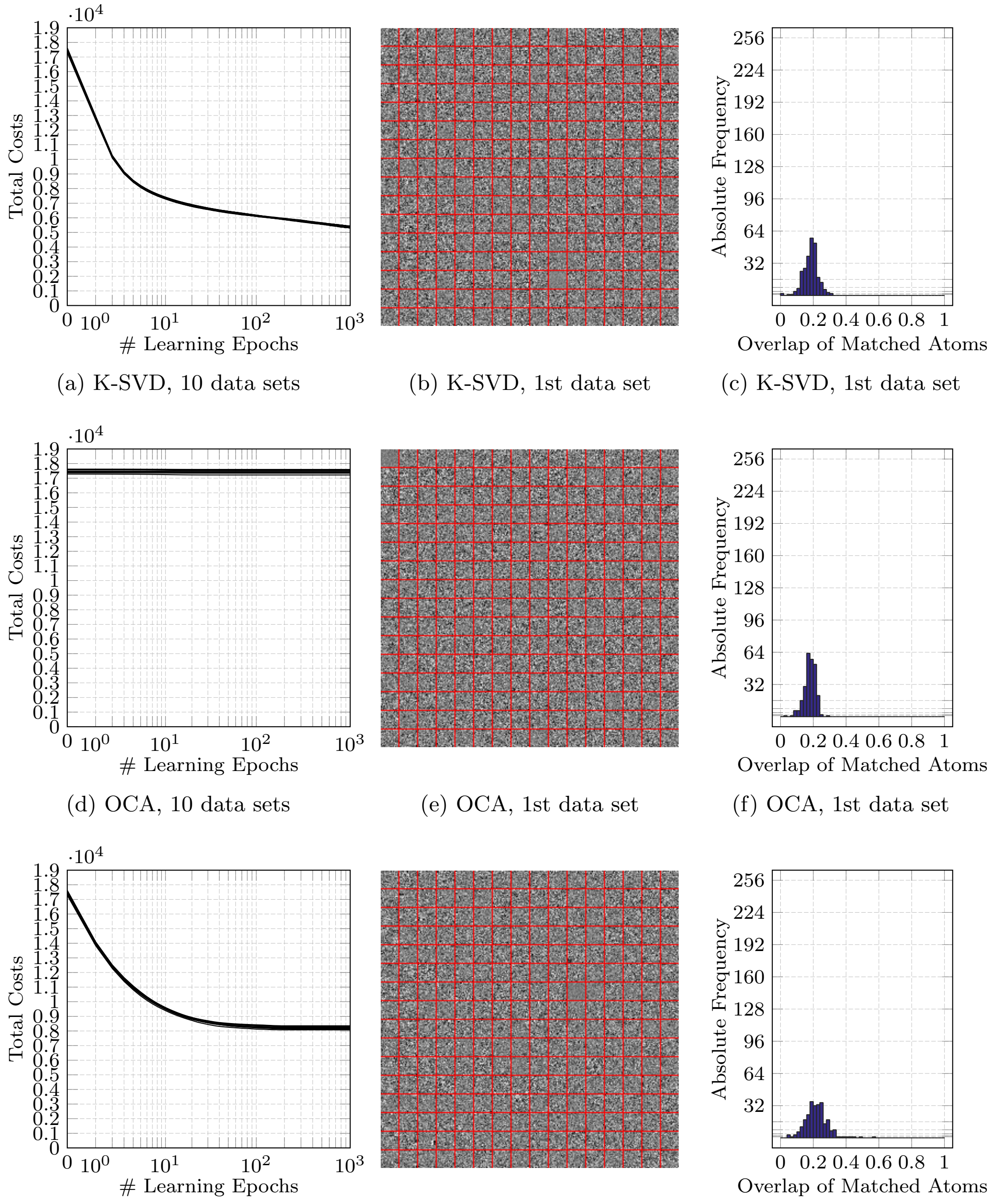
For the challenging recovery setting, where the sparsity level is rather low ($K = 42$), the ONB recovery is impaired for some of the methods. In this setting, K-SVD, OCA, and even CA reduce the costs only poorly. While K-SVD and CA at least halve the costs compared to the initial cost value, OCA persists at the initial cost level. Furthermore, the atom patches of the corresponding dictionaries look entirely noisy with nearly no identifiable structure from the reference ONB. In line with this observation is the distribution of matched overlaps around the value 0.2.

The poor performance by OCA is caused by the inferior outcome of its support estimation stage. Due to the low sparsity of the data, the support is mistakenly estimated to be maximally dense rather than $K$-sparse, which results in a dense coefficient matrix as well. Thus, a zero approximation error is achieved due to the dense coefficients, which prevents any change of the ONB by solving the OPP. Consequently, coefficients and ONB remain at their initial state and cause the costs to be constant over the whole learning phase.

The deficient recovery performance by CA is much more surprising since both subproblems, the dictionary update and the coefficient update, are solved optimally. The reason might be that the initial ONB is generated randomly and does not provide a suitable starting point. While CA achieves optimal performance for high sparsity levels, more challenging settings with less sparse data could imply more local minima of the cost function, in which CA gets stuck easier. For low sparsity levels, CA might rely on a sufficiently good initial ONB.

OSC performs remarkably well for the challenging recovery setting. Although OSC recovery performance was not perfect at the easy setting ($K = 18$), it has not changed noticeably at the difficult one ($K = 42$). However, it takes more learning epochs for OSC to initiate the descent of the costs. For the first 100 learning epochs, the costs remain coarsely at a rather high level. The primary amount of cost reduction takes place during the subsequent part of the learning phase. The final residual costs are higher for $K = 42$ than for $K = 18$. Interestingly, the recovered atom patches look quite similar in both settings. Accordingly, the distribution of matched overlaps is similar, but localized in the interval $[0.88, 0.96]$. Note that the mean matched overlap is slightly decreased. Again, the $K$-OSC variant and the $N$-OSC variant yield equivalent results. The results indicate that OSC recovers the underlying ONB robustly in terms of the sparsity level.

In the challenging recovery setting, GF-OSC is the only learning method that reduces the costs to the global minimum. In contrast to the easy recovery setting, the cost descent by GF-OSC is not monotonic on a fine scale. During the first 50 learning epochs, the costs remain at a rather high level. Subsequently, a major leap to the zero level occurs between learning epoch 80 and 150. The recovered ONB looks identical to the reference ONB up to two atom patches showing a mutual shadow. Accordingly,



(a) K-SVD, 10 data sets

(b) K-SVD, 1st data set

(c) K-SVD, 1st data set

(d) OCA, 10 data sets

(e) OCA, 1st data set

(f) OCA, 1st data set

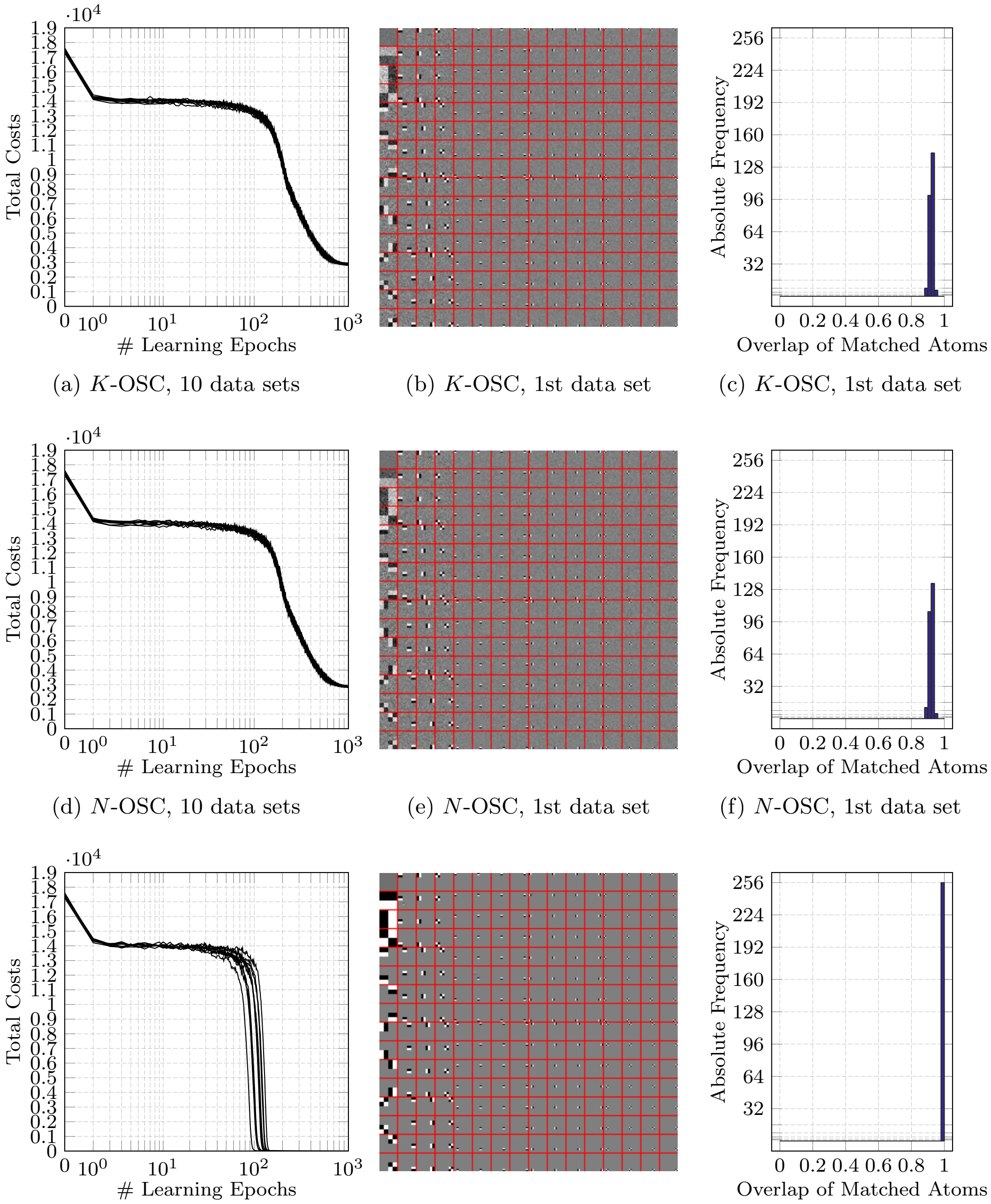
(g) CA, 10 data sets

(h) CA, 1st data set

(i) CA, 1st data set

Figure 2.9: Detailed ONB recovery performance analysis for the noiseless synthetic $K$-sparse data of sparsity level $K = 42$ ($\approx 16.4\%$ non-zero coefficients). The results obtained by K-SVD, OCA and CA are shown row-wise. In the left column, the $K$-sparse approximation error is plotted for the 10 data sets as a function of the number of learning epochs. In the middle column, the dictionary learned on the 1st of the 10 data sets is visualized according to the correspondence matching with the reference ONB. In the right column, a histogram shows the resulting distribution of matched overlaps.

the mutual overlaps are entirely contained in the most right bin, confirming that the



(a) $K$-OSC, 10 data sets (b) $K$-OSC, 1st data set (c) $K$-OSC, 1st data set

(d) $N$-OSC, 10 data sets (e) $N$-OSC, 1st data set (f) $N$-OSC, 1st data set

(g) GF-$K$-OSC, 10 data sets (h) GF-$K$-OSC, 1st data set (i) GF-$K$-OSC, 1st data set

Figure 2.10: Detailed ONB recovery performance analysis for the noiseless synthetic $K$-sparse data of sparsity level $K = 42$ ($\approx 16.4\%$ non-zero coefficients). The results obtained by $K$-OSC, $N$-OSC and GF-OSC are shown row-wise. In the left column, the $K$-sparse approximation error is plotted for the 10 data sets as a function of the number of learning epochs. In the middle column, the dictionary learned on the 1st of the 10 data sets is visualized according to the correspondence matching with the reference ONB. In the right column, a histogram shows the resulting distribution of matched overlaps.

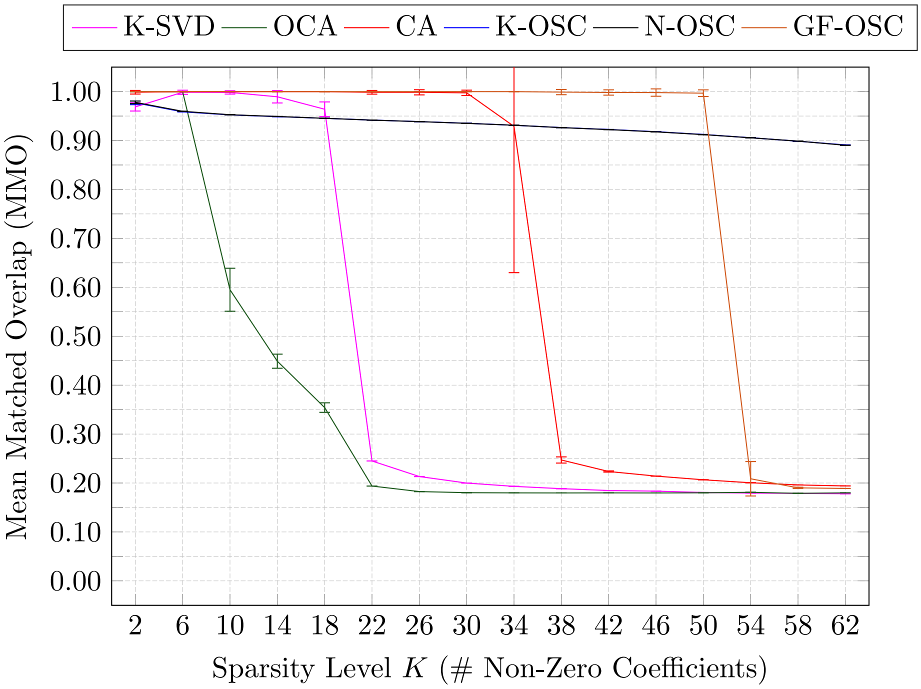


Figure 2.11: ONB recovery performance for the noiseless synthetic $K$-sparse data. The MMO is plotted as a function of sparsity level $K$. The error bar plot shows average and standard deviation of the MMO over the 10 data sets given for each $K$. Each data set contained $L = 1000$ training samples being $K$-sparse in the non-standard 2D Haar wavelet basis ($N = 256$). For each method, the total number of learning epochs was limited to 1000.

ONB recovery task is accurately solved.

### Condensed Recovery Performance Analysis for $K \in \{2, 6, \dots, 62\}$

Figure 2.11 provides for each sparsity level $K \in \{2, 6, \dots, 62\}$, for which synthetic data sets were generated, a condensed overview of the ONB recovery performance. For K-SVD, OCA, CA, $K$-OSC, $N$-OSC and GF-OSC, the mean matched overlap (MMO) after 1000 learning epochs is plotted as a function of sparsity level $K$. The plotted curves illustrate the average MMO over the 10 data sets that are given for each $K$, and error bars respectively indicate the standard deviation of the MMO. Except for OSC, each method has a sparsity limit in the investigated range of sparsity levels, which we define as the smallest value $K \in \{2, 6, \dots, 62\}$ for which the average MMO drops immediately from a high level to a minimal baseline level.

K-SVD attains for $K \in \{2, 6, \dots, 18\}$ a high average MMO of at least 0.96 with a standard deviation that it less than 0.014. For $K \in \{6, 10\}$, the ONB recovery performance has a maximum with an average MMO of at least 0.998. The sparsity limit of K-SVD is given by $K = 22$.

OCA achieves accurate ONB recovery with an average MMO of at least 0.999, but only if the sparsity level is very high, i.e. for $K \in \{2, 6\}$. The average MMO decreases significantly for $K \in \{10, 14, \dots, 22\}$, and is at the minimum level for $K \geq 22$.

CA attains for $K \in \{2, 6, \dots, 30\}$ very high ONB recovery performance with an average MMO of at least 0.997 and standard deviations less than 0.002. Sparsity level $K = 34$ indicates a tipping point, where the reference ONB is for 1 of the 10 data sets not recovered correctly, which explains the large standard deviation of 0.214. The sparsity limit of CA is given by $K = 38$.

OSC shows an average MMO which is high but not as close to the maximum as that of, e.g. CA or GF-OSC. But on the other hand, there is no sparsity limit within the investigated range of sparsity levels. Instead, the average MMO decreases slightly with a small linear slope from 0.98 to 0.89 for $K \in \{2, 6, \dots, 62\}$. The results obtained by $K$-OSC and $N$-OSC are not identical but very similar, their differences vanish. This comes as a surprise, as for $N$-OSC the true sparsity level of the data does apparently not need to be known, which is a great benefit in comparison with all the other methods. Furthermore, the standard deviation of the MMO vanishes for both OSC variants, regardless of $K$. For any $K \in \{2, 6, \dots, 62\}$, the standard deviation of the MMO is very small, less than 0.0017.

GF-OSC achieves for $K \in \{2, 6, \dots, 50\}$ very high ONB recovery performance with an average MMO of at least 0.996. The MMO standard deviation is less than $7 \cdot 10^{-5}$ for $K \in \{2, 6, \dots, 34\}$, and less than 0.002 for $K \in \{34, 38, \dots, 50\}$. The sparsity limit of GF-OSC is given by $K = 54$.

### 2.9.6 Results for the Noisy Scenario

In the noisy scenario we, narrow our analysis of the ONB recovery experiment down to the condensed recovery performance analysis.

#### Condensed Recovery Performance Analysis for $K \in \{2, 6, \dots, 62\}$

In complete analogy to Figure 2.11, Figure 2.12 provides a condensed overview of the ONB recovery performance when 5 dB isotropic Gaussian noise is added to the synthetic data. Note that OCA had to be excluded in the noisy scenario as the support estimation stage relies on strictly $K$-sparse data [Mishali and Eldar, 2009]. Altogether, the additive noise impairs the maximal MMO the methods can obtain. Furthermore, the method-specific sparsity limits, the value of $K$ at which the average MMO breaks down, are reduced by the noise. Hence, in the presence of noise all methods require higher sparsity of the data in order to successfully recover the reference ONB.

K-SVD attains a relatively high average MMO, which increases from 0.88 to 0.91 for $K \in \{2, 6, 10\}$. The MMO standard deviation for these values of $K$ are less than 0.015. Compared to the noiseless scenario, the sparsity limit reduces from $K = 22$ to $K = 14$.

CA obtains for $K \in \{2, 6, \ldots, 18\}$ the highest average MMO on a rather constant level within $[0.96, 0.97]$. In this range, the standard deviation of the MMO is also small, less than 0.0022. Compared to the noiseless scenario, the sparsity limit reduces from $K = 38$ to $K = 22$.

On the noisy data, OSC learned with the same learning rates as for the noiseless data, i.e. $\varepsilon_{\text{init}} = 10^{-1}, \varepsilon_{\text{final}} = 10^{-4}$. In contrast to the noiseless scenario, OSC reveals a sparsity limit when noise is present. Unlike the other methods, the drop of the average MMO does not occur abruptly but rather continuously. For $K = 2, 6, \ldots, 38$, the average MMO decays from 0.95 to 0.83 with a small linear slope. For $K = 38$, the average MMO has a knee, where the slope of the MMO decay increases considerably. Up to that point, i.e. for any $K \in \{2, 6, \ldots, 34\}$, the standard deviation of the MMO is small, less than $3 \cdot 10^{-3}$. Both OSC variants $K$-OSC and $N$-OSC yield very similar results.

On the noisy data, GF-OSC learned with the same learning rates as for the noiseless data, i.e. $c = \tau = \frac{1}{2}$ and $\alpha = 5$. GF-OSC attains the smallest average MMO compared to the other methods. The ONB recovery performance is considerably degraded. This comes as a surprise as GF-OSC is more than competitive in the noiseless scenario. The average MMO increases from 0.64 to 0.85 for $K \in \{2, 6, 10\}$ with standard deviation less than 0.038. As for K-SVD, the sparsity limit of GF-OSC is given by $K = 34$.

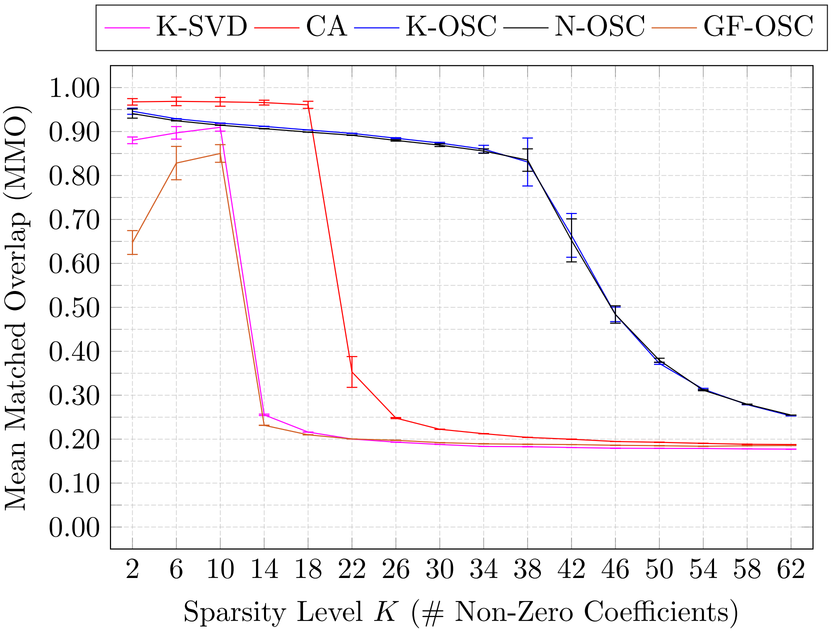


Figure 2.12: ONB recovery performance for the noisy synthetic $K$-sparse data. The MMO is plotted as a function of sparsity level $K$. The error bar plot shows average and standard deviation of the MMO over the 10 data sets given for each $K$. Each data set contained $L = 1000$ training samples being $K$-sparse in the non-standard 2D Haar wavelet basis ($N = 256$). For each method, the total number of learning epochs was limited to 1000.

## 2.10 Sparse Coding ONBs Learned from Natural Image Patches

We applied our orthogonal dictionary learning methods to learn sparse coding ONBs from a real world data set containing natural image patches. In this section we investigate the resulting ONBs and analyze their sparse encoding performance on test data.

### 2.10.1 The NSSiVS Data Set

We extracted image patches from the first set of the Nature Scene Collection [Geisler and Perry, 2011] containing images of nature scenes without man made objects or people. The uncompressed RGB images have a size of $2844 \times 4284$ pixels. The color channels are linearly scaled, each with a depth of 16 bits per pixel (bpp). As the images looked unnatural due to the linear scaling of the color channels, we converted them to a logarithmic scale to achieve a more natural appearance. Accordingly, to each color channel we applied pixel-wise the operation $\log_2(\cdot + 1)$ and divided subsequently by 16 to map the intensity values into the double precision floating point range $[0, 1]$. Subsequently, the color images were converted to grayscale images. From the entire set of 308 images, we randomly selected 250 training images. From each training image, we extracted 400 patches of size $16 \times 16$ pixels at random positions. These $10^5$ image patches were exclusively used for training. Analogously, a test data set with 23200 patches was generated from the remaining 58 images. Data preprocessing comprised the sample-wise subtraction of the DC component and of the sample mean vector.

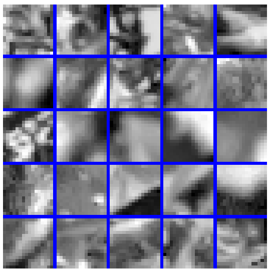


Figure 2.13: Data samples of the NSSiVS data set containing natural image patches. The entries of each sample are shifted to have zero mean and are subsequently scaled to unit supremum norm.

Figure 2.13 shows exemplarily samples of the NSSiVS data sets before DC component and mean vector are subtracted.

### 2.10.2 $K$-Sparse Approximation Performance for the NSSiVS Data Set

Figure 2.14 illustrates for the NSSiVS test data set, the $K$-sparse approximation performance of different bases as measured by the average signal-to-noise-ratio (SNR). On the one hand, we include ONBs of static transforms, i.e. of the 2D DCT and of the non-standard 2D Haar basis, into the comparison as they are known to provide sparse representations of natural image data [Ahmed et al., 1974, Pennebaker and Mitchell, 1992, Talukder and Harada, 2007]. On the other hand, we include bases derived from learning on the NSSiVS training data set. We include non-orthogonal complete dictionaries learned by K-SVD as well as ONBs resulting from PCA and from the orthogonal dictionary learning methods CA, $K$-OSC, $N$-OSC and GF-OSC. The sparse coding methods, i.e. K-SVD, CA, OSC and GF-OSC learned for 100 epochs, which corresponds to $t_{\max} = 10^7$ online ONB updates, starting with an initial random ONB $\mathbf{U}_{(0)}$. GF-OSC was applied with a cooling learning rate as it yields better results. The initial and final learning rates used by OSC were set to $\varepsilon_{\text{init}} = 10$ and $\varepsilon_{\text{final}} = 10^{-2}$, those used by GF-OSC were set to $\varepsilon_{\text{init}} = 1$ and $\varepsilon_{\text{final}} = 10^{-3}$ as a result of a parameter validation on an independently generated validation data set. With PCA, the $K$-sparse approximations are derived based on the $K$ leading PCs. With K-SVD, the $K$-sparse approximations are obtained using Batch OMP [Rubinstein et al., 2008]. Note that

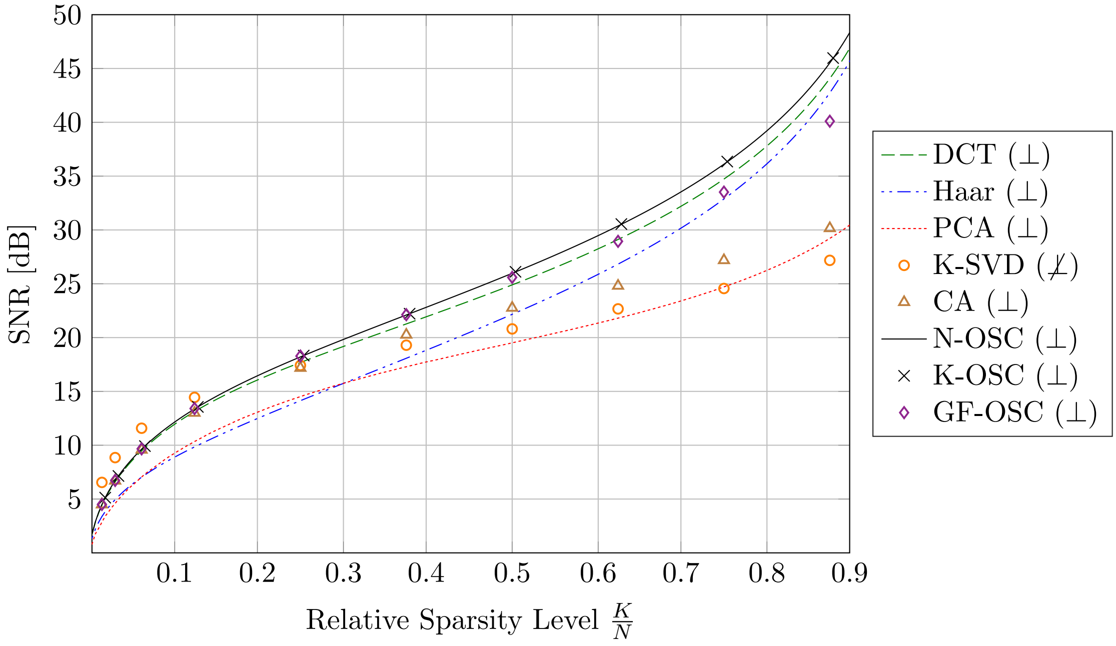


Figure 2.14: Average $K$-sparse approximation performance of the NSSiVS test data set containing natural image patches ($N = 256$). The SNR is plotted as a function of the relative sparsity level $K/N$.

OCA could not be included in the comparison, because its support recovery stage requires that the data are strictly $K$-sparse.

For the 2D DCT, the non-standard 2D Haar basis, the PCA and $N$-OSC only one single ONB is respectively available, independent from sparsity level $K$. For each of these methods, we computed the $K$-sparse approximation performance densely subject to $K = 1, 2, \ldots, 230$. The corresponding results are illustrated in Figure 2.14 by one curve for each ONB. K-SVD, CA, $K$-OSC, and GF-OSC, on the other hand, learn the sparse coding basis dependent on $K$. Thus, the $K$-sparse approximation performance is computed subject to bases specifically learned for individual sparsity levels in the range $K \in \{4, 8, 16, 32\} \cup \{64, 96, \ldots, 224\}$. The corresponding results are illustrated in Figure 2.14 by markers.

First of all, note that the single ONB learned by $N$-OSC yields the same $K$-sparse approximation performance as the ONBs which were learned by $K$-OSC specifically for the individual sparsity levels $K$. Apparently, the universality of $N$-OSC is not only limited to artificial data, but also holds for natural image data as well. For $K \in \{4, 8, 16, 32\}$, the non-orthogonal dictionaries learned by K-SVD achieve a higher SNR than the orthogonal ones. For $K \in \{4, 8, 16\}$, the ONB of the DCT, and the ONBs learned by CA, OSC, and GF-OSC have nearly equal approximation performance. For $K = 32$, the ONBs learned by OSC and GF-OSC are slightly superior compared to the ONBs of DCT and CA. For $K \in \{64, 96\}$, the sparse coding ONBs learned by OSC and GF-OSC attain the highest $K$-sparse approximation performance outperforming K-SVD, and increasing the SNR difference to DCT and CA. For even lower sparsity levels $K \in \{128, 160, 192, 224\}$ the ONBs by OSC show exclusively the highest $K$-sparse approximation performance with increasing difference to GF-OSC and still distinct difference to the remaining bases.

We investigated the stability of $N$-OSC by fixing the initial and final learning rates and by applying $N$-OSC in multiple runs. In each of the 20 runs, a different initial random ONB and a different random sequence of training sample was used. On the test data set, we computed the $K$-sparse approximation performance for each resulting $N$-OSC basis. We found, that the standard deviation of the $K$-term approximation performance was less than 0.081 dB for any $K \leq 254$, and even less than 0.017 dB for any $K \leq 230$, which indicates that the sparse encoding performance of an ONB learned by $N$-OSC is stable.

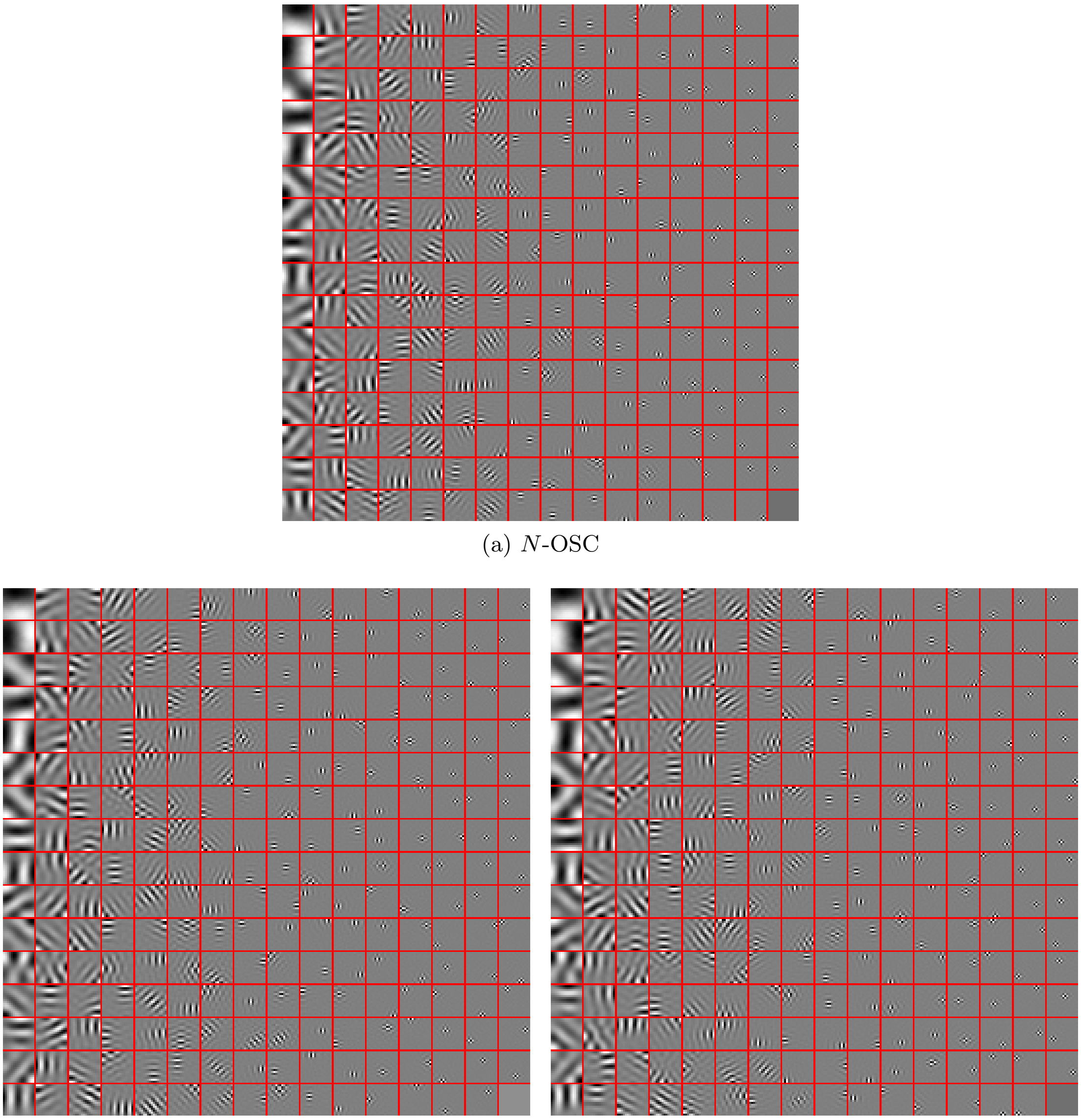
In the following subsection some of the learned sparse coding ONBs resulting from CA, OSC and GF-OSC are illustrated and discussed.

### 2.10.3 CA, OSC and GF-OSC on the NSSiVS Data Set

Figure 2.15 - Figure 2.17 illustrate the sparse coding ONBs learned by OSC, CA and GF-OSC on the NSSiVS training data set which contains natural image patches. The visualized ONBs were also used for the $K$-sparse approximation performance analysis

provided in Section 2.10.2. The ONBs of CA, $K$-OSC and GF-OSC are shown for the high sparsity level $K = 8$ and the low sparsity level $K = 64$. The atoms are sorted in decreasing order of encoding relevance (column-major order) as ranked by the average signal energy of the dense coefficients of the training data samples.

$N$-OSC learns a sparse coding ONB whose atoms have a remarkable structure. The atom patches have regular grating patterns of particular frequencies, orientations, and



(a) $N$-OSC

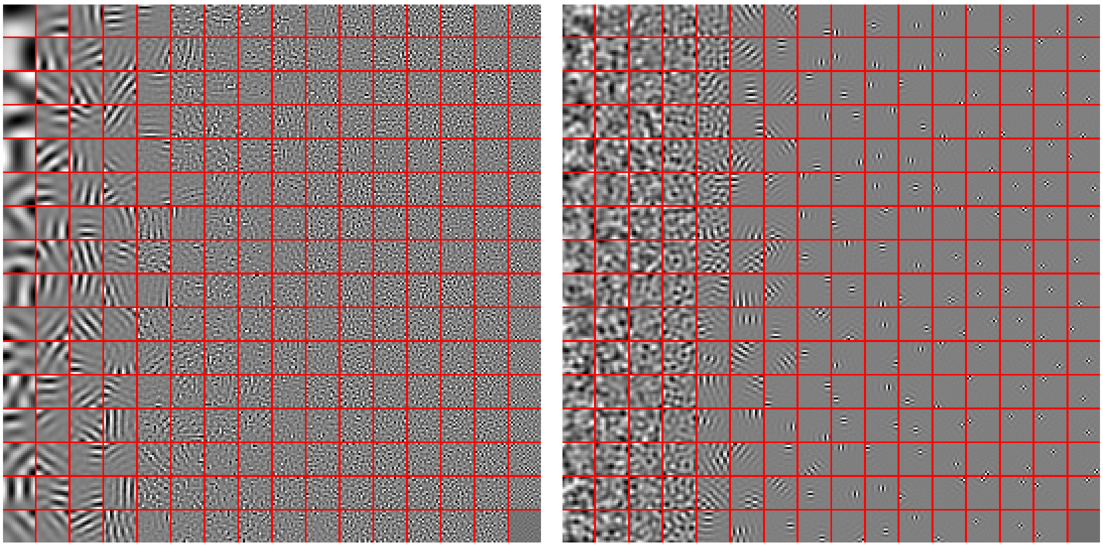
(b) $K$-OSC, $K = 8$

(c) $K$-OSC, $K = 64$

Figure 2.15: Sparse coding ONB learned by $N$-OSC and $K$-OSC on the NSSiVS training data set containing natural image patches ($N = 256$). The training phase included 100 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

spatial localizations. Furthermore, the atoms appear to be organized over multiple scales and some seem to be a hybrid of regular non-local sinusoidal gratings which are known from the 2D DCT basis, on the one hand, and spatially localized edge-detectors of different scales known from the 2D DWT, on the other hand. These analytic orthonormal bases with their sophisticated transform schemes were manually designed to sparsely encode natural image data. Hence, it makes sense that a learning method with the objective to maximize sparsity evolves atoms with similar properties purely driven from natural image data. Due to the relevance sorting, the following correlation can be observed: the larger the encoding relevance of an atom patch, the lower is its frequency and the larger is its spatial support. Three main orientations can be found among the atom patches: horizontal (0°), vertical (90°), and diagonal (45°) orientations. Among the atoms of intermediate to high frequency and localization a small set of basis functions is repeatedly found in different oriented and shifted versions.

Figure 2.15b and 2.15c show, for $K = 8$ and $K = 64$, the ONB learned by $K$-OSC on the NSSiVS training data set. Note that the sparse coding ONB learned by $K$-OSC is by visual inspection very similar to the sparse coding ONB learned by $N$-OSC. Apparently, OSC is consistent in terms of learning on the same data set similar sparse coding ONBs for quite various values of $K$, including $K = N$. In that sense, $N$-OSC is robust as it learns a universal sparse coding ONB analogous to the ONB



(a) CA, $K = 8$

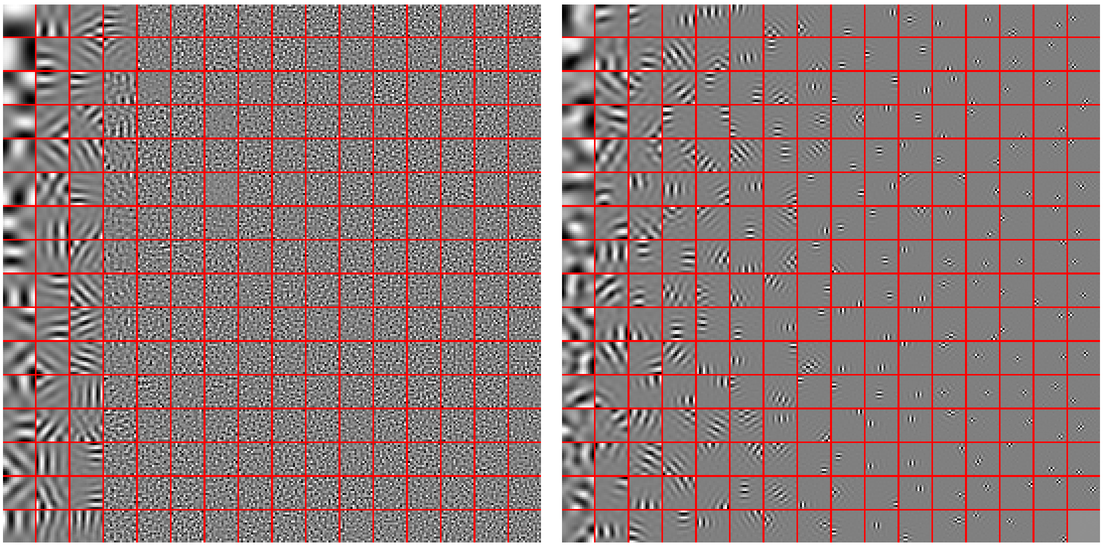
(b) CA, $K = 64$

Figure 2.16: Sparse coding ONB learned by CA on the NSSiVS training data set containing natural image patches ($N = 256$). The training phase included 100 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

recovery experiments. This is a great benefit, particularly for real world data sets for which an ideal sparsity level is a priori unknown, or in applications where the sparsity level cannot be assumed to be constant. The similarity of the learned ONBs is in line with the similar $K$-term approximation performance between $K$-OSC and $N$-OSC, cf. Figure 2.14.

According to Figure 2.14, CA has inferior $K$-sparse approximation performance compared to OSC. A priori one might expect that CA achieves a sparser data representation as the two subproblems of the alternating iteration are optimally solved. Figure 2.16 allows an explanation. It shows that CA learns atoms which are in principle similar to those learned by OSC. In contrast to OSC, however, CA does not manage to evolve the full repertoire of atom features at once. Instead CA atoms show subsets of those features, depending on parameter $K$. This might explain the inferior $K$-sparse approximation performance.

For small values of $K$, atoms with low to intermediate localization and frequency emerge, similar to OSC. However, highly localized atoms of high frequencies do not emerge. Instead, CA develops atoms that show a non-local structure resembling random noise. Since these atoms are among the least encoding relevant ones, they might be rarely involved in the learning process and thus remain to a large extent at their initial random state. For large values of $K$, atoms with intermediate to high localiza-



(a) GF-OSC, $K = 8$

(b) GF-OSC, $K = 64$

Figure 2.17: Sparse coding ONB learned by GF-OSC on the NSSiVS training data set containing natural image patches ($N = 256$). The training phase included 100 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

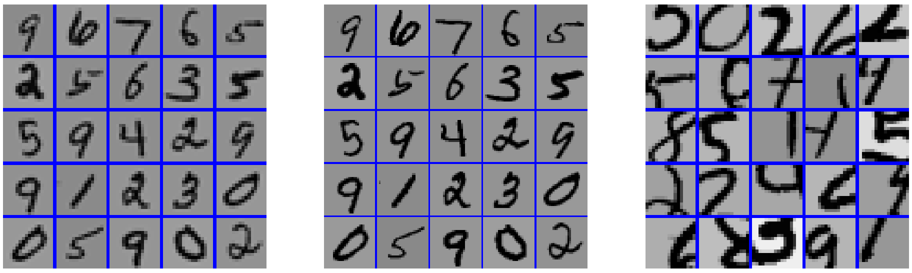
tion and frequency emerge similar to OSC, whereas the most encoding relevant atoms reveal an irregular and indifferent structure, distinct to the localized monodirectional low frequency gratings that OSC develops. Extensively increasing the number of CA learning epochs marginally expands the repertoire of atom features for some $K$, but does not prevent the issue.

Figure 2.17 shows that GF-OSC learns an ONB of atoms which have in principle a similar structure as the atoms learned by OSC and CA. Similar to CA, the repertoire of emerging features depends on user parameter $K$. For small values of $K$, the most encoding relevant atoms develop monodirectional gratings of low to intermediate frequency and localization. The less encoding relevant features, on the other hand, show merely a random noise structure. With increasing $K$ the atoms with low to intermediate frequency and localization become increasingly irregular and indifferent, whereas atoms with intermediate to high frequency and localization emerge. Figure 2.17b shows the ONB learned for $K = 64$, where a tipping point of the phenomenon can be observed. It can be seen that the more encoding relevant features already lost a small degree of regularity in comparison to Figure 2.17a. For $K = 64$, the two types of atom features seem to be slightly more balanced by GF-OSC than by CA. However, further increasing user parameter $K$ leads to a similar situation as depicted in Figure 2.16b for CA.

## 2.11 Sparse Coding ONBs Learned from Image Data of Handwritten Digits

According to the results presented in Section 2.10.2, a learned sparse coding ONB can provide sparser encodings than the ONB of a static transform. On natural image data, differences between the learned ONBs and the ONBs of a static transform are comparatively small, as such transforms were manually optimized to sparsely encode natural image data. Other types of image data, for which specialized analytic transforms are not available, should allow larger improvements relative to general transforms. The orthogonal dictionary learning methods we propose are adaptive and should extract relevant structures from the data, which are a priori inaccessible, and should exploit them in order to obtain superior sparse coding ONBs.

We applied our orthogonal dictionary learning methods to learn sparse coding ONBs from real world data sets containing image data of handwritten digits. In this section we investigate the resulting ONBs and analyze their sparse encoding performance on test data.



(a) MNIST variant 1

(b) MNIST variant 2

(c) MNIST variant 3

Figure 2.18: Data samples for the three variants of the MNIST data set. The entries of each sample are shifted to have zero mean and are subsequently scaled to unit supremum norm.

### 2.11.1 Three Variants of the MNIST Data Set

The MNIST data set [LeCun et al., 1998] contains a training set of $6 \cdot 10^4$ as well as a test set of $10^4$ grayscale images of size $28 \times 28$ with a gray-level depth of 8 bit. Each image shows a centered handwritten digit composed of black pen strokes on a white background. We used 3 different variants of the MNIST data:

- **MNIST variant 1**: downscaled MNIST images of size $16 \times 16$
- **MNIST variant 2**: MNIST images of original size $28 \times 28$
- **MNIST variant 3**: patches of size $16 \times 16$ extracted at random positions from the original MNIST images

To compose the data sets for all variants, we first transformed the original MNIST images to the double precision floating point range $[0, 1]$. For MNIST variant 1, we rescaled each image to size $16 \times 16$ using bicubic interpolation. We did that rescaling in order to reduce the data dimensionality and to relieve the computational load for the learning methods as well as to enable comparisons with the non-standard 2D Haar wavelet basis. For MNIST variant 3, we selected $10^4$ MNIST training images uniformly at random and extracted for each selected image 5 patches of size $16 \times 16$ at interior random positions. The training data set of MNIST variant 3 contained $5 \cdot 10^4$ samples in total. For each MNIST variant, we subtracted the DC component from the resulting data samples as well as, subsequently, the mean vector of the training data set.

Figure 2.18 shows exemplarily samples of the MNIST data sets before DC component and mean vector are subtracted.

### 2.11.2 $K$-Sparse Approximation Performance for the MNIST Variant 1 Data Set (Downscaled MNIST Images)

Analogous to Section 2.10.2, we studied the average $K$-sparse approximation performance of several bases with respect to the test set of MNIST variant 1, i.e. downscaled MNIST test images of size $16 \times 16$, in terms of the average signal-to-noise-ratio

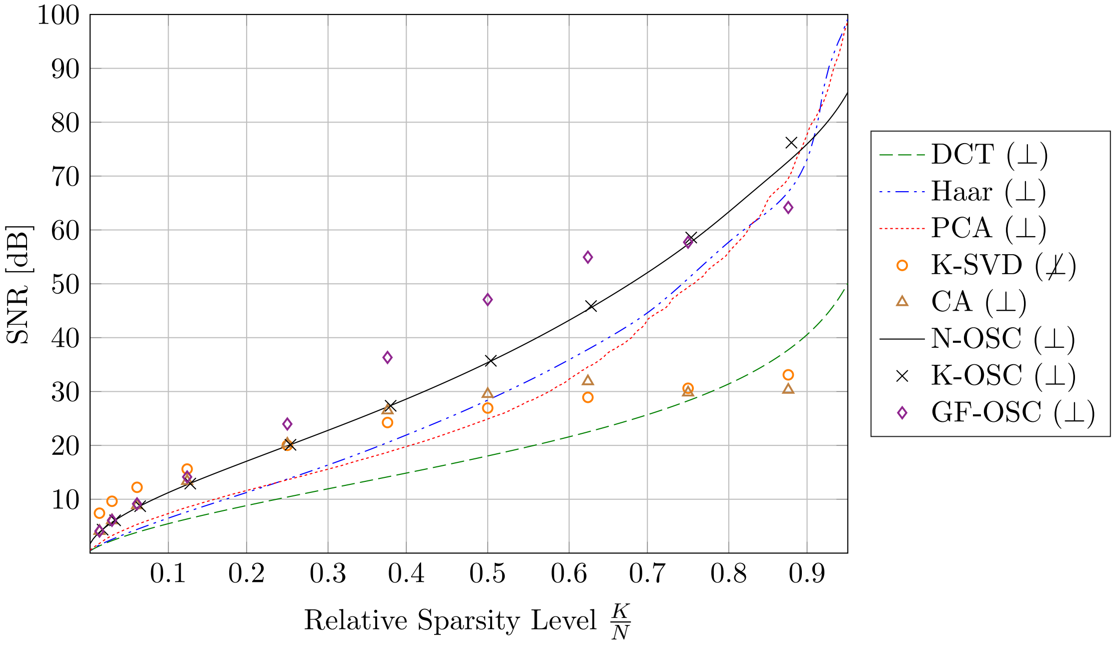


Figure 2.19: Average $K$-sparse approximation performance of the MNIST variant 1 test data set containing images of (downscaled) handwritten digits ($N = 256$). The SNR is plotted as a function of the relative sparsity level $K/N$.

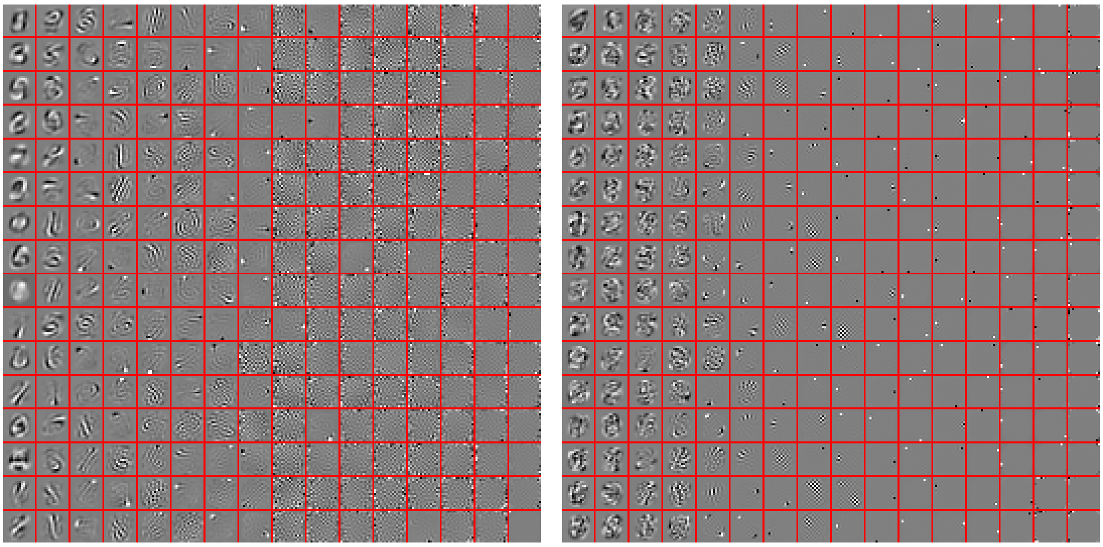
(SNR). Figure 2.19 illustrates the corresponding results. The sparse coding methods, i.e. K-SVD, CA, OSC and GF-OSC learned for 600 epochs, which corresponds to $t_{\max} = 3.6 \cdot 10^7$ online ONB updates, starting with an initial random ONB $\mathbf{U}_{(0)}$. Similar to the experiment based on the NSSiVS data set, GF-OSC was applied with a cooling learning rate as it yields better results. The initial and final learning rates used by OSC were set to $\varepsilon_{\text{init}} = 2.8$ and $\varepsilon_{\text{final}} = 2.8 \cdot 10^{-3}$, those used by GF-OSC were set to $\varepsilon_{\text{init}} = 0.1$ and $\varepsilon_{\text{final}} = 10^{-5}$.

The ONBs of the static transforms, i.e. 2D DCT and the 2D Haar basis, and the PCA show for $K \leq 96$ inferior $K$-sparse approximation performance compared to any dictionary learned by K-SVD, CA, OSC and GF-OSC. However, the ONB of the Haar transform and the PCA attain for lower sparsity levels a higher approximation performance than the dictionaries learned by K-SVD and CA. For high sparsity levels $K \in \{4, 8, 16, 32\}$ the non-orthogonal dictionary learned by K-SVD yields a higher $K$-sparse approximation performance than any learned or static ONB. For these sparsity levels the ONBs learned by CA, OSC, and GF-OSC encode comparably well with a SNR difference of approximately $2 - 3$ dB to the K-SVD dictionary. For $K = 32$, the $K$-sparse approximation performance by GF-OSC is the second best and is 0.77 dB higher than by CA and 1.23 dB higher than by OSC. That difference in sparse encoding performance increases for lower sparsity levels. For $K \in \{64, 96, 128, 160\}$ the ONBs learned by GF-OSC attain distinctly higher $K$-sparse approximation performance than the ONBs learned by K-SVD, OSC and CA. The SNR difference between GF-OSC and OSC is for these $K$ between 3.8 dB and 11.3 dB. For $K \geq 96$ the ONBs learned by OSC attain the second best $K$-sparse approximation performance. Furthermore, the $K$-sparse approximation performance of CA saturates for $K \geq 128$ and is no longer competitive as the SNR difference relative to OSC and GF-OSC decreases significantly. Note that the ONBs learned by $K$-OSC and $N$-OSC yield again a similar encoding performance for almost all investigated values for $K$.

In the following subsection some of the learned sparse coding ONBs resulting from CA, OSC and GF-OSC are illustrated and discussed.

### 2.11.3 CA, OSC and GF-OSC on the MNIST Variant 1 Data Set

Figure 2.20 - 2.22 illustrate the sparse coding ONBs learned by CA, OSC and GF-OSC on the MNIST variant 1 data set which contains downscaled MNIST images of size $16 \times 16$. The visualized ONBs were also used for the $K$-sparse approximation performance analysis provided in Section 2.11.2. The ONBs of CA, $K$-OSC and GF-OSC are shown for the high sparsity level $K = 8$ and the low sparsity level $K = 64$. The atoms are sorted in decreasing order of encoding relevance (column-major order) as ranked by the average signal energy of the dense coefficients of the training data



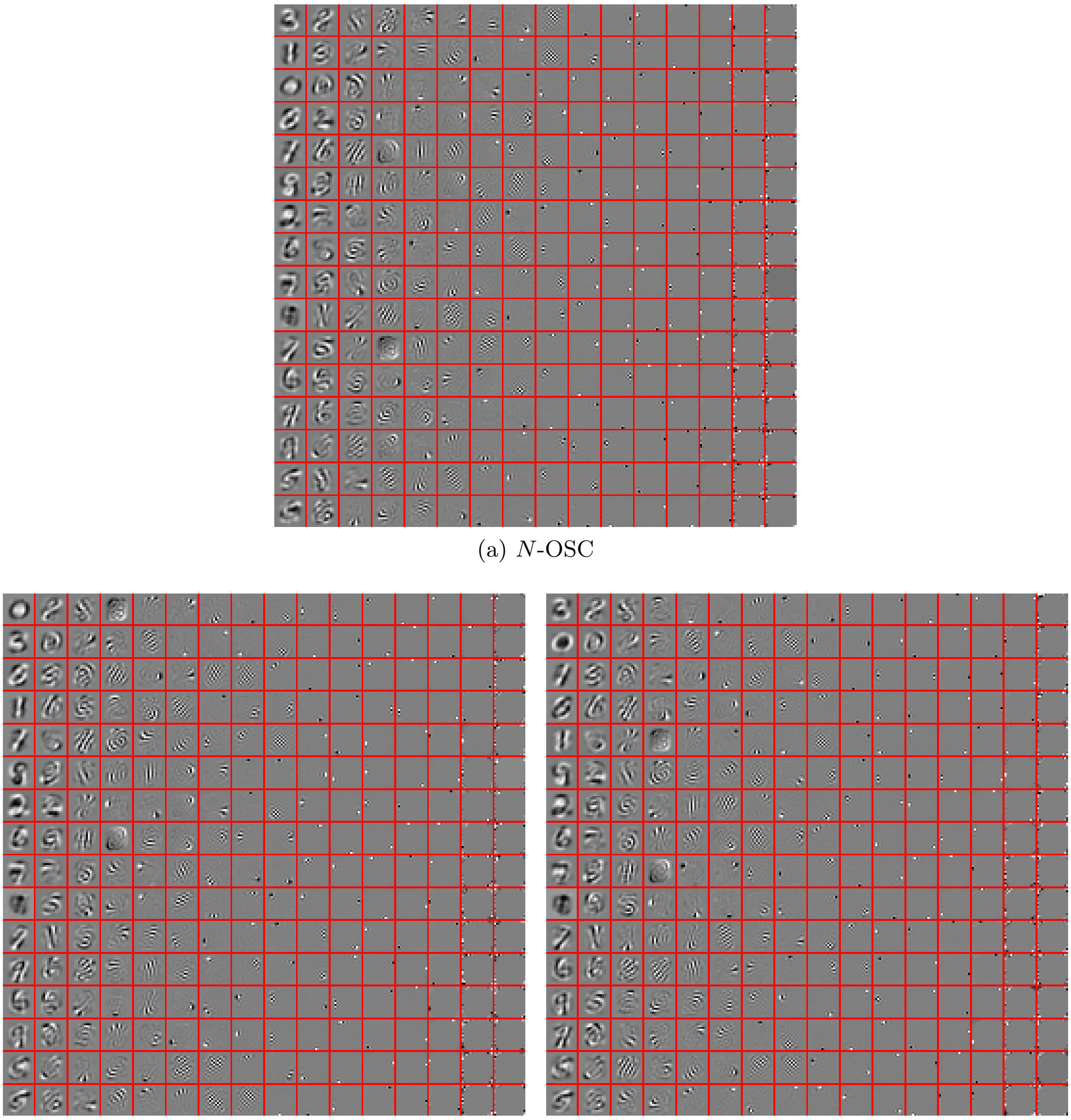
(a) CA, $K = 8$

(b) CA, $K = 64$

Figure 2.20: Sparse coding ONB learned by CA on the MNIST variant 1 training data set containing (downscaled) images of handwritten digits ($N = 256$). The training phase included 600 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

samples.

As observed previously for synthetic data sets and natural image data sets, the ONBs learned by $K$-OSC and $N$-OSC are comparatively similar. For CA and GF-OSC, the repertoire of atom features depends on the sparsity parameter $K$, whereas OSC does not seem to depend noticeably on that parameter. For small values of

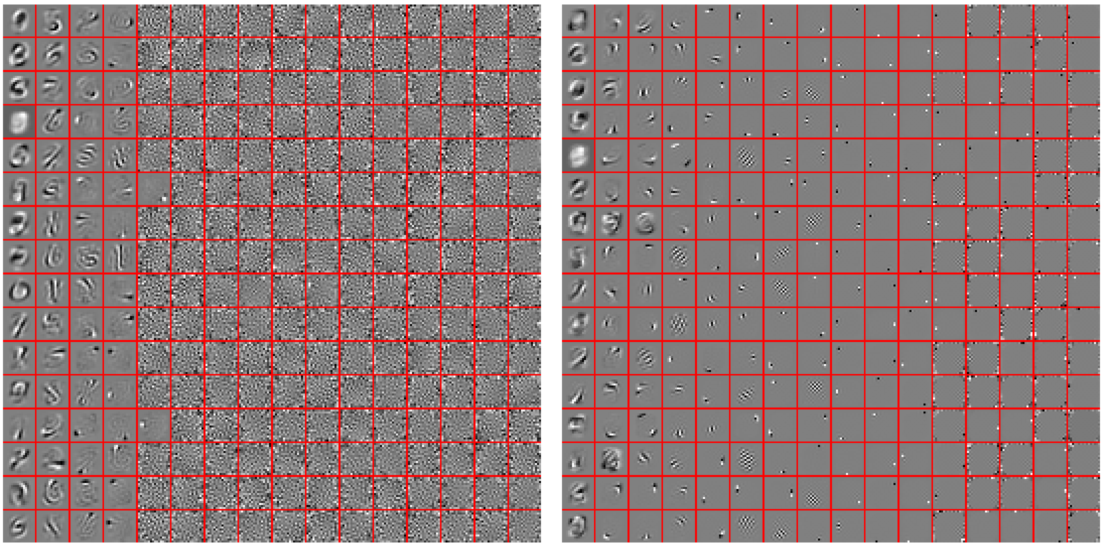


(a) $N$-OSC

(b) $K$-OSC, $K = 8$

(c) $K$-OSC, $K = 64$

Figure 2.21: Sparse coding ONB learned by $N$-OSC and $K$-OSC on the MNIST variant 1 training data set containing (downscaled) images of handwritten digits ($N = 256$). The training phase included 600 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.



(a) GF-OSC, $K = 8$

(b) GF-OSC, $K = 64$

Figure 2.22: Sparse coding ONB learned by GF-OSC on the MNIST variant 1 training data set containing (downscaled) images of handwritten digits ($N = 256$). The training phase included 600 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

$K$, CA and GF-OSC learn sparse coding ONBs which contain a significant number of atoms that look rather unstructured and noisy, seemingly due to residues of the random initialization which might persist as these atoms are barely involved in the learning process. For large values of $K$, the most encoding relevant atoms learned by CA contain rather large blobs in the center without any interpretable structure.

Overall, the methods learn comparable types of features. On the one hand, atoms emerge that resemble prototypes of particular digits or digit combinations which reflects the fact that during the learning an adaptation of the sparse coding ONB to the image content, i.e. the handwritten digits takes place. For CA and GF-OSC, these atoms emerge as the most encoding relevant atoms, but only if the parameter $K$ is set to a small value, i.e. for high sparsity levels. OSC learns these atom features as well, but independent of $K$. Another kind of atom features, which frequently appears, seems to induce a sensitivity to various stroke patterns and arch shapes which are localized and also typical for the handwritten digits. For CA and GF-OSC, such atoms are learned also for small values of $K$ and appear among the more (but not necessarily the most) encoding relevant atoms. A sparse coding ONB learned by OSC does also develop such stroke-like features. However, they sometimes tend to be accompanied by some additional grating structure. A large set of atoms resembles edge-detectors of rather high but different degrees of localization. For CA that kind of atoms develop for low sparsity levels as intermediate to less encoding relevant atoms. In contrast, GF-OSC learns for large values of $K$ almost exclusively highly localized edge-detectors with a high encoding relevance, which might explain the superior encoding performance for the large values of $K$ compared to CA and OSC. Furthermore, a few atoms show grating patterns in the center with intermediate localization and mixed directions. Commonly, the least encoding relevant atoms are similar to canonical basis vectors (only one non-zero pixel) or show an indifferent noise-like structure which is concentrated at the image border.

### 2.11.4 OSC on the MNIST Variant 2 Data Set

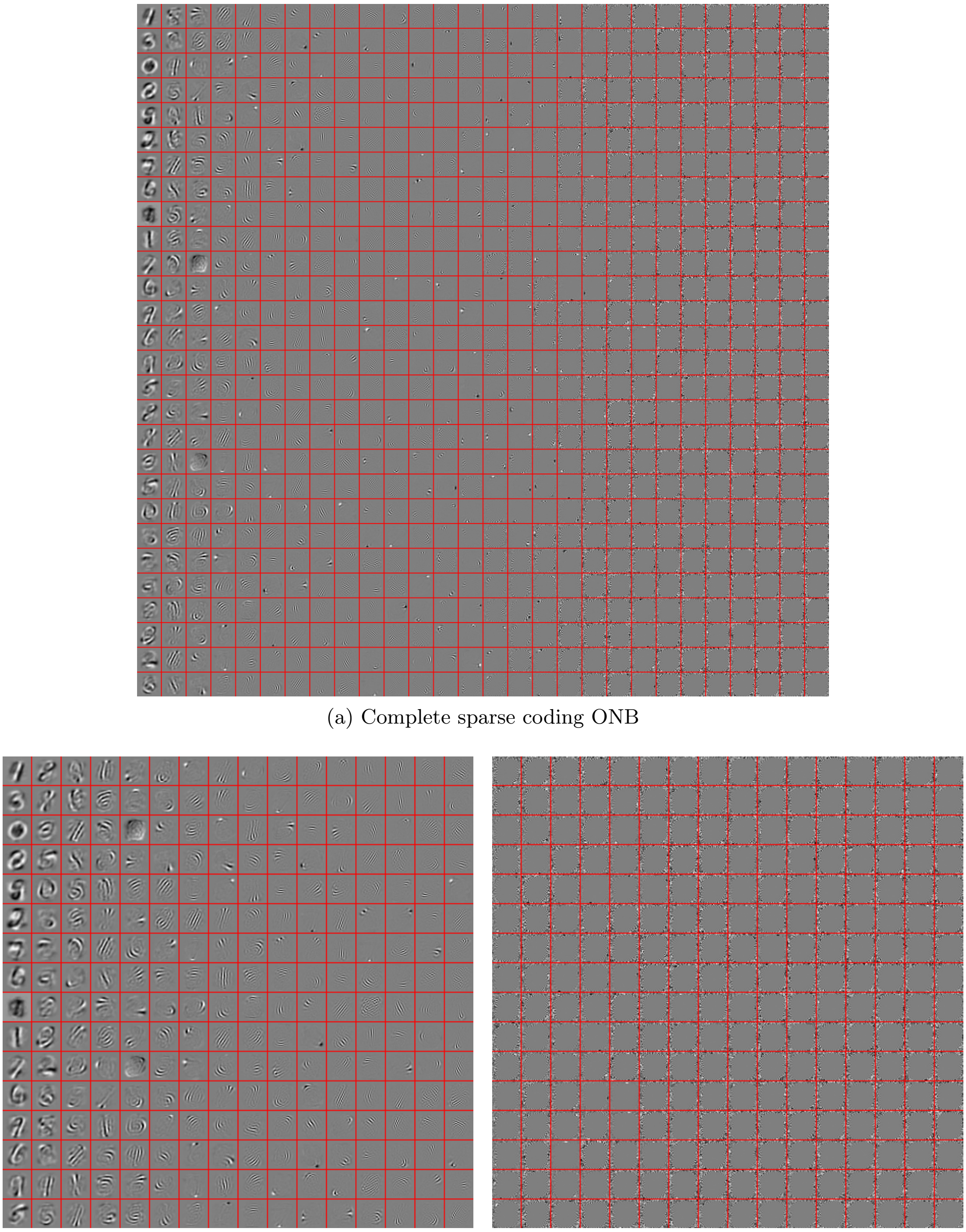
Figure 2.23 illustrates a sparse coding ONB learned by $N$-OSC on the MNIST variant 2 data set which contains MNIST images of original size $28 \times 28$. Figure 2.23a show the complete ONB with all $N = 784$ atoms. Figure 2.23b depicts the 256 most encoding relevant atoms and Figure 2.23c the least encoding relevant ones. The atoms are sorted in decreasing order of encoding relevance (column-major order) as ranked by the average signal energy of the dense coefficients of the training data samples. Starting with an initial random ONB, $N$-OSC performed $t_{\max} = 2 \cdot 10^6$ updates of the ONB, which corresponds to $\approx 16.7$ learning epochs. The number of learning epochs was selected rather small in order to relieve the computational load as the computational complexity of OSC grows polynomially by data dimensionality $N$. The initial and final learning rate were set to $\varepsilon_{\text{init}} = 10^{-1}$ and $\varepsilon_{\text{final}} = 10^{-3}$.

It can be seen that for $N = 784$ (MNIST variant 2) similar atom features emerge as for $N = 256$ (MNIST variant 1). In the high dimensional scenario, however, the described repertoire of atom features appears to be more diverse among the most encoding relevant atoms. On the other hand, many of the least encoding relevant atoms seem to be unstructured as they show noise-like patterns at the image border and are otherwise zero, which might be due to an insufficient number of ONB updates.

### 2.11.5 OSC on the MNIST Variant 3 Data Set

Figure 2.24 illustrates a sparse coding ONB learned by $N$-OSC on the MNIST variant 3 data set which contains patches of MNIST images of size $16 \times 16$. The atoms are sorted in decreasing order of encoding relevance (column-major order) as ranked by the average signal energy of the dense coefficients of the training data samples. Starting with an initial random ONB, $N$-OSC performed $t_{\max} = 2 \cdot 10^7$ updates of the ONB, which corresponds to 400 learning epochs. The initial and final learning rate were set to $\varepsilon_{\text{init}} = 1.5 \cdot 10^{-1}$ and $\varepsilon_{\text{final}} = 1.5 \cdot 10^{-3}$.

Similar to the ONBs learned on natural image patches (cf. Section 2.10.3), $N$-OSC learns as well a sparse coding ONB with atoms revealing consistent structural properties. The atom features resemble contour detectors and grating structures of



(a) Complete sparse coding ONB

(b) 256 most encoding relevant atoms

(c) 256 least encoding relevant atoms

Figure 2.23: Sparse coding ONB learned by $N$-OSC on the MNIST variant 2 training data set containing images of handwritten digits ($N = 784$). The training phase included $\approx 16.7$ learning epochs. Each atom is visualized as a $28 \times 28$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

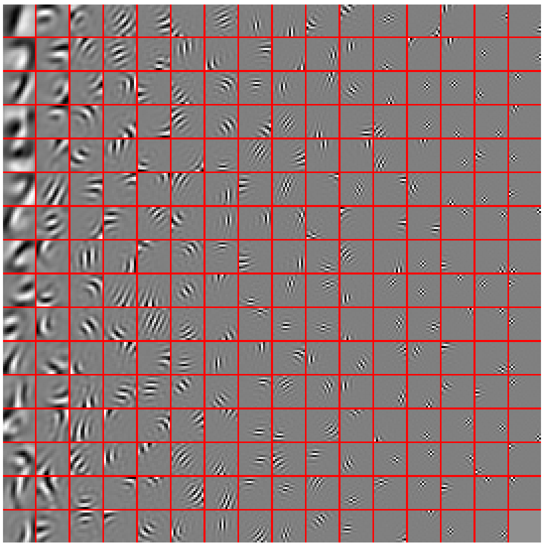


Figure 2.24: Sparse coding ONB learned by $N$-OSC on the MNIST variant 3 training data set containing patches of handwritten digits ($N = 256$). The training phase included 400 learning epochs. Each atom is visualized as a $16 \times 16$ patch. The atom patches are sorted in decreasing order of encoding relevance (column-major order). For display purposes, the entries of each atom patch (except the estimated DC component) are shifted to have zero mean and are subsequently scaled to unit supremum norm.

different frequencies, orientations and localizations. With decreasing encoding relevance of the atoms the frequency and the spatial localization increases. A relatively small set of basis functions can be observed occurring repeatedly in differently oriented and shifted versions. Interestingly, these basis functions do not always have a single straight direction but are curved for some of the atoms. This is reasonable because the strokes of many digits are also curved and the sparse coding ONB adapts to that property of the data.

## 2.12 Applications

To study applications for orthogonal dictionary learning methods, we made image compression and image denoising experiments.

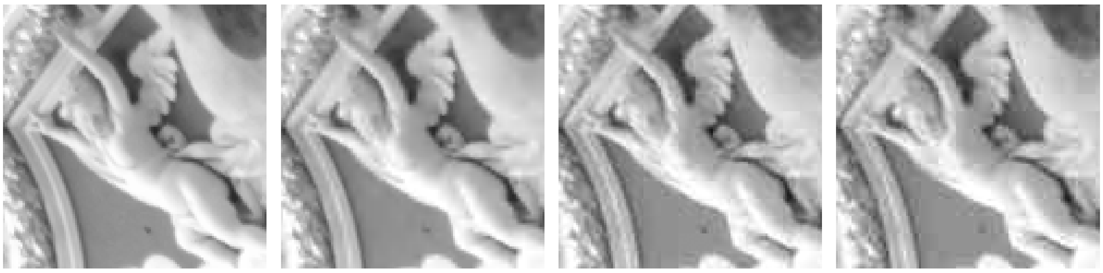
### 2.12.1 Image Compression

We conducted image compression experiments with an OSC codec that is based on the JPEG baseline codec [Pennebaker and Mitchell, 1992], for which the 8×8 DCT was replaced by an 8×8 $N$-OSC basis which was learned on the NSSiVS data set as described in Section 2.10.3. We used gray level images (8 bit gray level depth) of a freely available benchmark data set [Becker et al., 2014] containing 15 large uncompressed images. We divided the data set into two parts, i.e. 9 training images and 6 test images.

A JPEG encoder processes the quantized AC coefficients of each image tile in a zig-zag pattern, which defines an ordering of the DCT basis functions by decreasing encoding relevance. It was required to generate analogously an ordering for the 8×8 $N$-OSC basis. We determined an equivalent ordering by measuring the encoding relevance of each atom. This was done by counting how often, on average over all training data samples, each $N$-OSC basis vector occurred among the $N/2$ largest squared coefficients. This yielded a ranking histogram, which was not only used to derive the ordering but also to generate the AC quantization table for the quality level 50 (on a scale from 1 to 100). The obtained ranking histogram was rescaled to have the same minimal and maximal quantization value as the JPEG luminance quantization table for the AC coefficients at that quality level. Quantization values for any other quality level are derived from this table as defined in the JPEG baseline standard. We also applied a zero run-length encoding of the AC coefficients and generated a Huffman code from patches, which were extracted from the 9 training images[2].

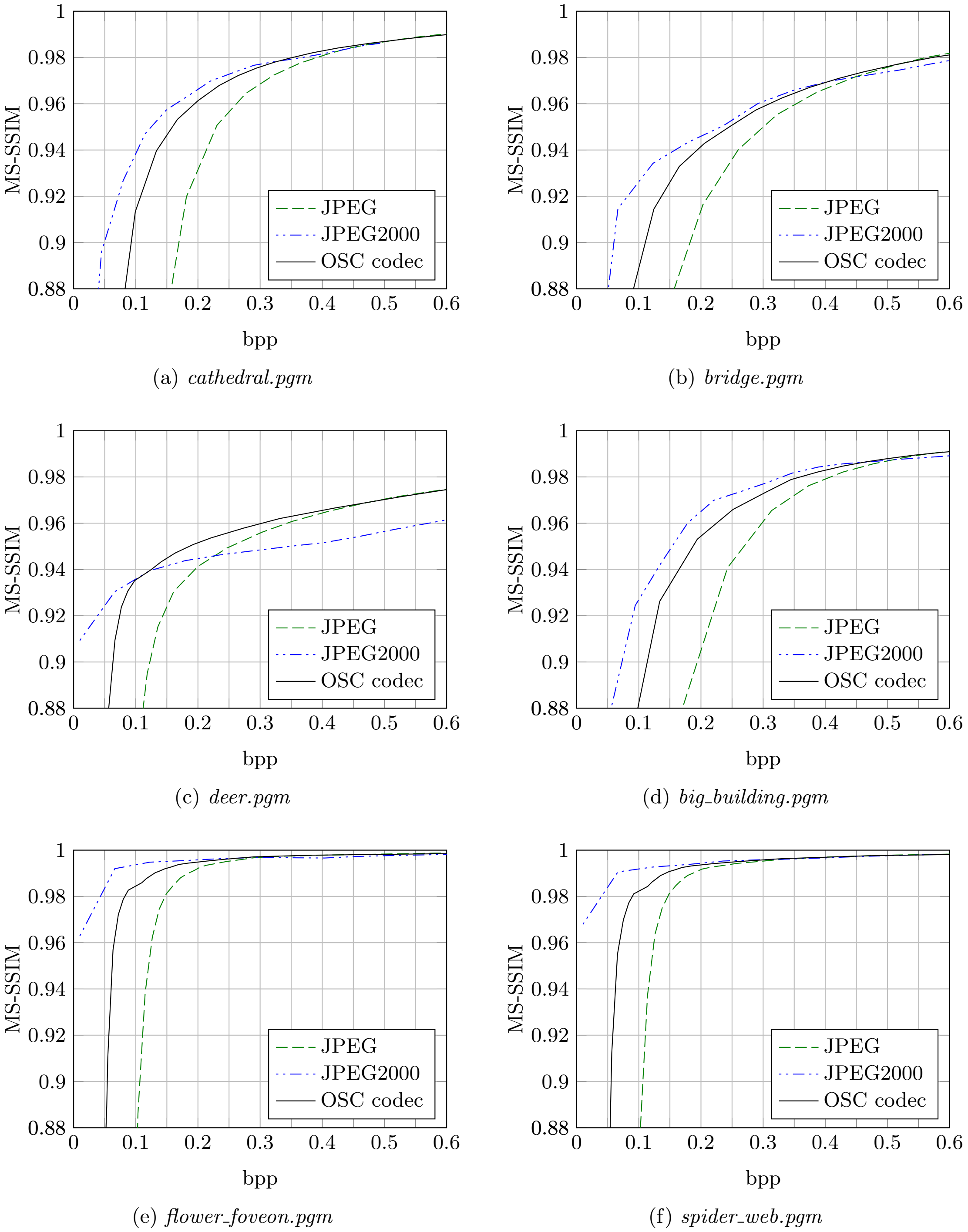
For each test image we computed a rate distortion plot by varying the quality level. To measure reconstruction performance, we computed the Multi-Scale Structural Similarity Index (MS-SSIM) [Wang et al., 2003] which is plotted as a function of the bitrate (bpp), i.e., the file size of the compressed image (in bits) divided by the number of pixels. The comparison to the JPEG codec was done by applying the command line tool `pnmtojpeg`, which uses the Independent JPEG Group's JPEG library. The used parameters were: `-grayscale`, `-dct=float`, and `-quality=n`. For the sake of a broader assessment, we also provide rate distortion curves for the JPEG2000 codec, although it is not based on an ONB but on biorthogonal Cohen-Daubechies-Feauveau (CDF) wavelets and operates in terms of a full image transform instead of processing disjoint image patches. We used the Open JPEG command line tool `image_to_j2k` with default parameters.

[2] Note that the 9 training images of the benchmark data set [Becker et al., 2014] were only used for the Huffman code and not for learning the $N$-OSC basis



(a) Original crop (b) JPEG2000 (c) $N$-OSC (d) JPEG

Figure 2.25: An image region of size 120 × 120 extracted from different compressed versions of test image *cathedral.pgm*. The compressed images were obtained by the JPEG2000 codec, the OSC codec, and the JPEG baseline codec for a compression rate of 0.29 bpp.



(a) *cathedral.pgm*

(b) *bridge.pgm*

(c) *deer.pgm*

(d) *big_building.pgm*

(e) *flower_foveon.pgm*

(f) *spider_web.pgm*

Figure 2.26: Rate distortion analysis for different test images.

Usually, we observed that the MS-SSIM for all three codecs converges to a common rate distortion curve as the bitrate exceeds an image dependent value around 0.5 bpp. Figure 2.26 shows that for low bitrates a higher compression performance is obtained with the OSC codec than with the JPEG baseline standard. The JPEG2000 codec is, nevertheless, superior at low bitrates. Note, however, that it benefits from the multi-

| $\sigma$ | 2 | 5 | 10 | 15 | 20 | 25 | 50 |
|---|---|---|---|---|---|---|---|
| Image | `cameraman` (512 × 512) | | | | | | |
| K-SVD | 46.07 | 40.31 | 36.52 | **33.89** | **31.75** | **30.00** | **24.37** |
| CA | **46.36** | 40.64 | **36.57** | 33.72 | 31.46 | 29.74 | 23.68 |
| $N$-OSC | 46.32 | **40.69** | 36.56 | 33.71 | 31.42 | 29.61 | 23.77 |
| Image | `house` (512 × 512) | | | | | | |
| K-SVD | 48.07 | **42.83** | **38.33** | **36.12** | **34.66** | **33.31** | **27.18** |
| CA | **48.31** | 42.62 | 38.09 | 35.93 | 34.34 | 32.84 | 25.88 |
| $N$-OSC | 48.30 | 42.41 | 37.95 | 35.74 | 34.11 | 32.58 | 26.58 |
| Image | `lena` (512 × 512) | | | | | | |
| K-SVD | 42.25 | 37.50 | 34.50 | 32.76 | **31.35** | **30.13** | **25.07** |
| CA | **43.40** | 38.23 | 34.88 | 32.75 | 31.17 | 29.83 | 24.29 |
| $N$-OSC | 43.25 | **38.24** | **35.00** | **32.88** | 31.28 | 29.88 | 24.57 |
| Image | `peppers` (512 × 512) | | | | | | |
| K-SVD | 40.84 | 35.23 | 33.03 | **31.77** | **30.59** | **29.48** | **24.83** |
| CA | **42.68** | **36.57** | 33.32 | 31.66 | 30.35 | 29.17 | 23.88 |
| $N$-OSC | 42.53 | 36.48 | **33.38** | 31.75 | 30.39 | 29.16 | 24.20 |
| Image | `barboon` (512 × 512) | | | | | | |
| K-SVD | 44.06 | 37.23 | 32.49 | 29.89 | 28.02 | **26.55** | **21.64** |
| CA | 44.28 | 37.77 | **32.90** | **30.05** | **28.03** | 26.48 | 21.37 |
| $N$-OSC | **44.46** | **37.88** | 32.86 | 29.98 | 27.94 | 26.39 | 21.46 |
| Image | `pirate` (512 × 512) | | | | | | |
| K-SVD | 41.31 | 36.15 | 32.23 | 30.31 | **28.94** | **27.79** | **23.62** |
| CA | **43.15** | 37.17 | **32.97** | **30.59** | 28.92 | 27.62 | 23.10 |
| $N$-OSC | 43.08 | **37.18** | 32.96 | 30.56 | 28.87 | 27.56 | 23.25 |
| Image | `barbara` (512 × 512) | | | | | | |
| K-SVD | 38.74 | 35.64 | 32.11 | 29.74 | 27.94 | 26.50 | 22.02 |
| CA | **43.24** | 37.55 | 33.34 | 30.90 | 29.06 | 27.55 | 21.91 |
| $N$-OSC | 43.20 | **37.64** | **33.54** | **31.07** | **29.17** | **27.63** | **22.17** |
| Image | `boat` (512 × 512) | | | | | | |
| K-SVD | 40.74 | 35.52 | 32.13 | 30.19 | 28.81 | **27.61** | **23.27** |
| CA | **42.92** | **36.81** | 33.00 | 30.72 | 28.96 | 27.60 | 22.80 |
| $N$-OSC | 42.78 | 36.76 | **33.01** | **30.73** | **28.98** | 27.60 | 22.93 |
| Image | `fingerprint` (512 × 512) | | | | | | |
| K-SVD | 42.57 | 35.61 | 31.78 | **29.56** | **27.90** | **26.51** | **20.11** |
| CA | **42.79** | 36.29 | 31.93 | 29.49 | 27.76 | 26.32 | 19.29 |
| $N$-OSC | 42.72 | **36.30** | **31.96** | 29.48 | 27.68 | 26.18 | 19.78 |

Table 2.1: Image denoising results based on sparse approximations of overlapping image patches of size 16 × 16. The denoising performance, as measured by the PSNR (dB) between original image and denoised estimate, was averaged over 5 runs.

scale representation of images in the wavelet domain. Both, the JPEG and OSC codecs are patch based and suffer from blocking artifacts at very low bitrates. This might be one reason for their inferior compression performance compared to JPEG2000.

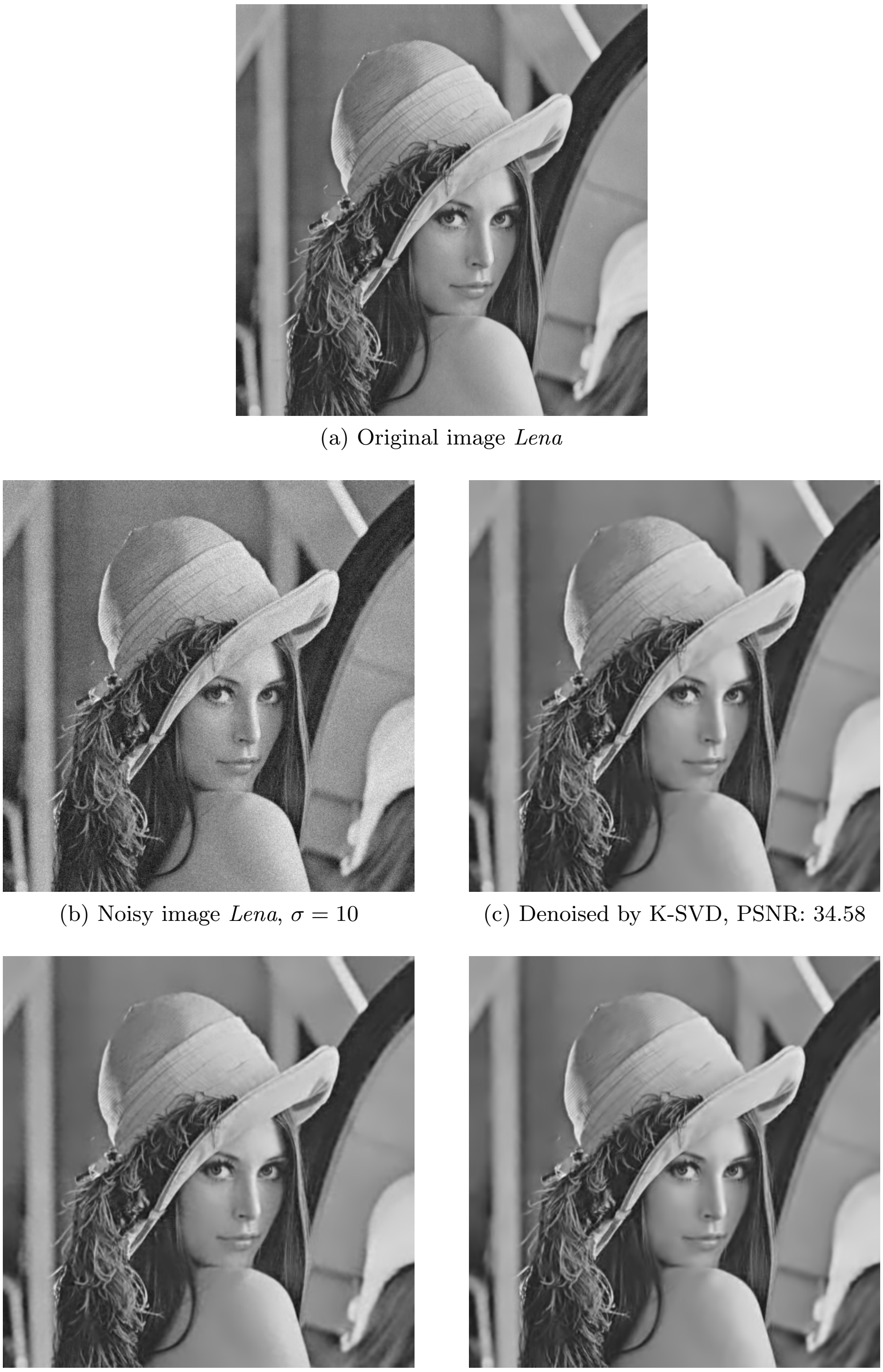
### 2.12.2 Image Denoising

We made experiments to assess the applicability of $N$-OSC for image denoising and followed the denoising framework proposed in [Elad and Aharon, 2006]. We used sparse coding ONBs as well as non-orthogonal overcomplete sparse coding dictionaries learned on the NSSiVS data set containing natural image patches (see Section 2.10.1). First, we distorted 9 standard test images of size 512 × 512 with additive isotropic Gaussian noise using standard deviations $\sigma \in \{2, 5, 10, 15, 20, 25, 50\}$. The noisy images were subsequently clipped to the range [0, 255]. From each noisy image, we extracted overlapping patches of size 16 × 16 from all possible locations such that no patch exceeds the image border. Subsequently, the patches were sparsely approximated using OMP. The objective function minimized by OMP does not only take the reconstruction error of all the extracted image patches into account but also incorporates the reconstruction error of the whole image, which is reobtained by fusing the approximated patches, in form of a regularization term. Note that for orthogonal dictionaries a sparse approximation computed by OMP is equivalent to (2.11) and thus optimal. The denoised image estimate is constructed by fusing the sparsely approximated patches. The gray value of each denoised image pixel is averaged from all its overlapping patches. For the entire image denoising procedure we used the `ompdenoise2.m` function of the KSVD-Box v13 in combination with OMPBox v10 [Rubinstein et al., 2008] with parameters as proposed in [Elad and Aharon, 2006].

We compared image denoising performance between dictionaries learned by K-SVD, CA and $N$-OSC for 100 epochs on the NSSiVS training data set. For CA, we report results obtained with user parameter $K^* = 28$, because it yields the best results for the investigated parameters $K \in \{20, 24, 28, 32\}$. For K-SVD, we report results obtained with user parameter[3] combination $(K^*, M^*) = (16, 1024)$, because it yielded the best results of the investigated parameter combinations $(K, M) \in \{4, 8, 12, 16\} \times \{512, 1024\}$. The denoising performance as measured by the PSNR (dB) is listed in Table 2.1 and was averaged over 5 runs. For each combination of image and noise level, the dictionary providing the highest PSNR is highlighted in bold face. The experimental results show that K-SVD, CA, and $N$-OSC dictionaries perform comparably well. The optimal dictionary for the denoising task depends on the chosen combination of image and noise level. For the small noise levels $\sigma \in \{2, 5, 10\}$ the ONBs learned by CA and $N$-OSC achieve superior denoising performance compared to the overcomplete dictionary learned by K-SVD. For the large noise levels $\sigma \in \{25, 50\}$ the situation reverses. However, for K-SVD and CA, the denoising performance also depends on the user parameters, i.e. codebook size (only K-SVD) and sparsity level $K$, while $N$-OSC does not require the optimization of these parameters.

Figure 2.27 shows the denoised test image *Lena* which is obtained by using an overcomplete sparse coding dictionary learned by K-SVD in comparison to using sparse coding ONBs learned by CA and $N$-OSC for the noise level $\sigma = 10$. In the depicted run, the denoised image derived by the $N$-OSC ONB yields a PSNR of 35.0 dB which is approximately 0.4 dB higher than the denoised image derived by the $K$-SVD dictionary and approximately 0.1 dB higher than the reconstruction derived by the CA ONB.

[3] Recall that $M$ is the user parameter for the dictionary size, i.e. the number of atoms.



(a) Original image *Lena*

(b) Noisy image *Lena*, $\sigma = 10$

(c) Denoised by K-SVD, PSNR: 34.58

(d) Denoised by CA, PSNR: 34.90

(e) Denoised by $N$-OSC, PSNR: 35.00

Figure 2.27: Comparison of image denoising results between a non-orthogonal overcomplete dictionary learned by K-SVD and ONBs learned by CA and $N$-OSC for the test image *Lena*.

# 3 Adaptive Hierarchical Sensing

This chapter is organized as follows. In Section 3.1 we formally introduce the sensing problem to sample a signal by collecting only few linear measurements. We briefly distinguish between conventional non-adaptive Compressed Sensing (CS) and adaptive hierarchical sensing (AHS), the central sensing approach proposed in this chapter. Section 3.2 gives an overview of related literature which also proposes adaptive sensing schemes. In Section 3.3 we introduce CS, outline how a signal is recovered from the collected set of non-adaptive linear measurements and cite exemplarily a condition for obtaining reconstruction guarantees. Section 3.4 summarizes the key principle of AHS. The sensing tree, the essential data structure supplying the collection of sensing vectors for AHS, is introduced in Section 3.5. In Section 3.6 we emphasize the simplicity of the direct signal recovery that comes naturally with AHS. We furthermore investigate if the AHS signal recovery performance can be improved by exploiting particular measurements more efficiently. Section 3.7 presents the $\tau$-AHS algorithm and analyzes the number of measurements in the case of $k$-sparse signals. Likewise, Section 3.8 presents and analyzes the $K$-AHS algorithm. In Section 3.9 we study a greedy approach to optimize the sensing tree structure for improving AHS performance based on a training data set that represents a signal population of interest. The usefulness of this structuring approach is demonstrated for natural image patches and different sparse coding bases. Section 3.10 is dedicated to the theoretical analysis of $K$-AHS in terms of recovering the most significant signal coefficients. A sufficient optimality condition for $K$-AHS is derived, which enables to infer sufficient conditions on the parameters of different signal models. A theoretical analysis of $K$-AHS is also given from a probabilistic perspective. Upper bounds are derived for the probability that $K$-AHS fails to collect the $K$ largest coefficients. In Section 3.11, we validate the $K$-AHS performance for the signal models by numeric simulations. In Section 3.12, we apply $K$-AHS for sensing real-world images and compare the results to $\tau$-AHS and a standard CS approach.

## 3.1 Sensing Problem

### 3.1.1 Signal Sparsity/Compressibility Assumption

Suppose $\mathbf{x} \in \mathbb{R}^N$ is an unknown signal that we intend to sample. The main precondition for both Compressed Sensing (CS) and Adaptive Hierarchical Sensing (AHS) is, that $\mathbf{x}$ has a sparse or compressible representation $\mathbf{a}$ in a linear transform basis $\mathbf{\Psi} = (\psi_1, \ldots, \psi_N) \in \mathbb{R}^{N \times N}$. In other words, the representation $\mathbf{a} = \mathbf{\Psi x}$ itself is unknown, but we suppose that it has only few entries that are substantially distinct from zero. Furthermore, let $\overline{\mathbf{\Psi}}^T = \mathbf{\Psi}^{-1} \in \mathbb{R}^{N \times N}$ be the corresponding inverse transform basis which transforms the representation $\mathbf{a}$ back to the original signal $\mathbf{x} = \overline{\mathbf{\Psi}}^T \mathbf{a}$. We call $\mathbf{\Psi}$ analysis transform (analysis basis) and $\overline{\mathbf{\Psi}}^T$ synthesis transform (synthesis basis).[1] For instance, $\mathbf{\Psi}$ can be an ONB (with the implication $\overline{\mathbf{\Psi}} = \mathbf{\Psi}$) such as the Discrete Cosine Transform (DCT) or a Daubechies wavelet basis. Alternatively, the pair $\mathbf{\Psi}$ and $\overline{\mathbf{\Psi}}$ can be a biorthonormal basis such as a Cohen-Daubechies-Feauveau wavelet basis.

### 3.1.2 Collecting $\mathcal{O}(K \log \frac{N}{K})$ Linear Measurements

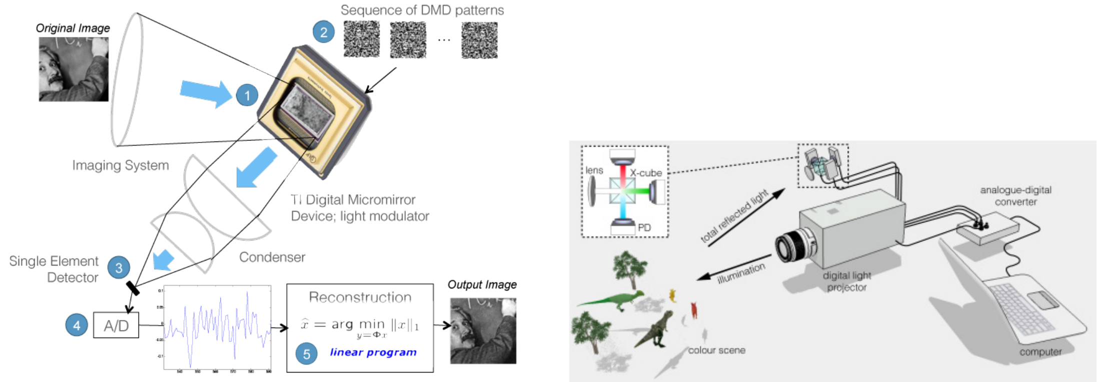
The goal of CS and AHS is to collect only few, i.e. less than $N$ linear measurements of the unknown signal $\mathbf{x}$, particularly less than $N$, and to recover the sparse representation of the signal from that collection. Such a linear measurement is defined as the inner product of $\mathbf{x}$ with a selectable sensing vector $\varphi$, i.e. it has the form $y = \langle \mathbf{x}, \varphi \rangle$, and is in practice realized by particular sensing hardware (see e.g. Section 3.1.3). The desired number of measurements, $M$, is ideally of order $\mathcal{O}(K \log \frac{N}{K})$, where $K$ is the number of significant coefficients (e.g. the number of non-zero coefficients of a sparse signal). This class of measurement bounds has been shown to be near-optimal in the context of CS in order to accurately reconstruct the signal [Candes and Tao, 2006, Candès et al., 2006]. For most CS applications, single measurements are collected sequentially over time: $y_t = \langle \mathbf{x}, \varphi_t \rangle$, where $t = 1, \ldots, M$ are discrete time steps.

### 3.1.3 Examples of Imaging Hardware

CS and AHS imaging is applicable whenever the involved sensing hardware enables to compute linear measurements in form of inner products between the scene of interest and controllable sensing vectors. In the following, we give two examples of hardware setups that have been used for CS imaging and could also be used for AHS imaging implementing a minor modification.

In 2006, the architecture of a one pixel camera implementing the CS principle has been proposed [Takhar et al., 2006, Wakin et al., 2006a, Wakin et al., 2006b]. Instead

[1] The completeness assumption regarding $\mathbf{\Psi}$ and $\overline{\mathbf{\Psi}}$ does primarily occur in the CS literature and is made here as well. However, there are also theoretical CS results showing that $\ell_1$-analysis optimization is capable of dealing with overcomplete dictionaries $\mathbf{\Psi}$, see e.g. [Candès et al., 2010].



(a) Imaging based on light modulation by a digital micromirror. The image is taken from [Herman et al., 2015]

(b) Imaging based on active illumination via structured light emitted by a digital light projector. The image is taken from [Welsh et al., 2013].

Figure 3.1: Two examples of hardware setups for implementing compressive imaging with a single photo detector.

of a high-definition image sensor that measures the light intensity for each pixel, such a camera consists of only one single photo detector and a controllable array of micro mirrors (DMD). With such an architecture a measurement is collected as follows: the light reflected from the scene is projected through a lens onto the DMD which is controlled externally and causes a micro mirror-wise modulation of the light according to a control pattern. The control pattern corresponds to the sensing vector. The modulated light is reflected by the DMD, bundled by a second lens and integrated by the single photo detector to a scalar measurement value. In order to sample an image, multiple measurements need to be taken sequentially using distinct control patterns. However, the total number of measurements that is required for accurately reconstructing the image is commonly much lower than the dimensionality of the "discretized scene", i.e. the number of micro mirrors.

Alternatively, CS based single pixel imaging can be realized by emitting structured light patterns into the scene and integrating the light reflected from the scene by a single photo detector [Welsh et al., 2013]. An advantage is that this approach does not require optical lenses. However, the fact that active illumination is the light modulating component of this approach limits its applicability in real world scenarios to a certain extend. This active illumination approach using structured light has also been used to realize 3D imaging based on two photo detectors [Sun et al., 2013].

Figure 3.1 illustrates the two described CS imaging hardware setups. For AHS imaging a simple additional feedback of the collected measurements, i.e. the A/D converter output, to the processing unit is required such that measurements can be compared for selecting new sensing vectors (i.e. control patterns).

### 3.1.4 Brief Distinction Between CS and AHS

AHS will be proposed as an alternative adaptive approach to solve sensing problems suited to CS. However, there are a few distinctions to CS which are outlined in the following.

CS makes non-adaptive measurements using sensing vectors which obey some random structure and which are independent from the signal and from the measurements collected throughout the sampling process. In a separate reconstruction stage, the signal is recovered from the collected measurements by inverse optimization, for which a suitable transform pair $\mathbf{\Psi}$ and $\overline{\mathbf{\Psi}}$ is selected. CS obeys the principle *sample first, reconstruct later*.

In contrast to CS, AHS requires to select $\mathbf{\Psi}$ prior to sampling as it uses sensing vectors which are linear combinations of analysis basis vectors. Based on simple decision rules, upcoming AHS sensing vectors are selected depending on relative comparisons of previously observed measurements. Thus, a feedback about collected measurements is required. Fortunately, AHS directly tracks down relevant transform coefficients of the signal in the analysis basis and does not require a reconstruction stage based on inverse optimization.

## 3.2 Literature Review

Adaptive sensing schemes have been proposed before and can be quite diverse in nature. The most of them have in common that sensing vectors are selected or generated based on information about the signal which is gathered by previous measurements throughout the sampling process. The objective is to optimize the gain of new information.

Coulter at al. proposed the neural network model Adaptive Compressed Sensing (ACS), which is a sparse coding neural network with a synaptic learning scheme that is embedded into the compressed sensing framework. Motivated by neurobiological findings, encoding and weight adaptation stages of their ACS network have limited access to the original data. They showed that with these networks smooth and biologically realistic receptive fields, also known from sparse coding models, emerge despite the fact that the sensory input is subsampled and mixed by the feedforward connectivity [Coulter et al., 2010].

Burciu et al. proposed Hierarchical Manifold Sensing (HMS), an adaptive hierarchical sensing scheme to solve classification tasks for images that are distributed on a non-linear manifold [Burciu et al., 2016]. By hierarchically decomposing the training data into partitions using PCA and k-means clustering, HMS infers the class of an input image based on only few linear measurements. Their approach, however, has limitations as it requires to have instances in the training set which are similar to the unknown signal that is to be classified.

Deutsch et al. proposed Adaptive Direct Sampling (ADS) to directly sample relevant wavelet coefficients of an image in a selective hierarchical manner [Deutsch et al., 2009]. The set of potential sensing vectors matches with the wavelet basis. First, ADS samples all transform coefficients in all sub-bands within a limited number of the coarsest levels. Subsequently, a heuristic based on the Lipschitz exponent is applied to iteratively decide at which image locations and for which sub-bands the coefficients of the next finer scale will be sampled or omitted. Their approach, however, is limited to the wavelet domain.

Aldroubi et al. proposed an adaptive compressed sampling approach to sample sparse signals based on a Huffman tree [Akram Aldroubi and Zarringhalam, 2011]. The Huffman tree is derived from probabilities assigned to sets of non-zero locations, which reflect statistics of the signal population. In a way, such a Huffman tree is related to the sensing tree that is used by K-AHS (see Section 3.5 below) as it is traversed during sampling and each visited node corresponds to a linear measurement of the signal with a sensing vector that yields the sum of a subset of signal components. On average, their method has a sampling complexity of $k \log N + 2k$ measurements to find $k$ non-zero locations. In contrast to $K$-AHS, their sampling scheme traverses the Huffman tree multiple times (one run for each non-zero component), and requires furthermore to recalculate sensing vectors after each run, depending on already identified non-zero locations. However, the authors do not address the issue that unfavorable constellations of significant coefficients can cancel each other. Furthermore, their method was not tested on real world signals.

A further class of adaptive sensing approaches is inspired by group testing or experimental design.

Group testing, in general, is a strategy to identify few elements with particular properties in a large set by performing tests on subsets, rather than on individual elements [Du and Hwang, 2000]. Applications are given, for instance, by the false coin problem or by medical screening problems, where the task is to effectively identify a small amount of infected people in a large population by conducting as few tests as possible on pooled samples [Dorfman, 1943, Hwang, 1972]. In [Iwen and Tewfik, 2012] an adaptive group testing approach is presented for sensing sparse signals by collecting as few noisy measurements as possible. The considered measurement model deals with Gaussian background noise, also known as “clutter noise” in radar applications. Two algorithms are presented. One is designed to detect the only non-zero component of a 1-sparse signal using a binary search procedure. The second algorithm is designed to detect all non-zero components of a $k$-sparse signal by performing a partitioning that isolates each significant signal component with high probability and applies the first algorithm to the obtained subsets. Theoretical bounds for the required number of measurements are developed and it is shown that the adaptive sensing requires asymptotically less samples than non-adaptive sensing.

Experimental design is an information theoretic framework addressing the problem of optimally designing a sequence of experiments in order to gain knowledge about the true state of the world. The outcome of each experiment can reduce the experimenters uncertainty about the state by providing new bits of information. The experimenters objective is to exploit the information of previous experiments and design the subsequent experiment in a way that maximizes the expected information gain [DeGroot, 1962, Lindley, 1956]. Bayesian adaptive sensing is inspired by this concept and designed for sensing an unknown sparse signal using a sequence of random sensing matrices. These sensing matrices are drawn from a probability distribution that is gradually adjusted throughout the sensing process, i.e. shaped by the distribution of observed measurements such that the expected information gain is maximized. This is in contrast to non-adaptive CS, where only a single sensing matrix is used whose entries are drawn i.i.d. from a symmetric distribution. The information obtained from previous measurements is utilized to adequately place probability mass on the columns of the sensing matrix such that sensing energy is focussed onto locations for which it is likely that significant signal coefficients are contained and away from locations for which it is unlikely. For each sensing step the sensing matrix is drawn from the updated distribution that maximizes the Kullback-Leibler divergence between the posterior distribution of the signal given the measurements and the prior distribution of the signal [Haupt and Nowak, 2012]. Bayesian Adaptive Sensing can outperform non-adaptive CS in noisy settings in terms of the reconstruction error relative to the number of measurements [Castro et al., 2008, Ji et al., 2008, Seeger, 2008, Seeger and Nickisch, 2008].

Another adaptive sensing approach based on Bayesian adaptive sensing is given by distilled sensing (DS) [Haupt et al., 2011]. It is a quasi-Bayesian approach as it merely approximates the focusing described above, and thus allows a theoretical analysis that is otherwise difficult due to inherent statistical dependencies [Haupt and Nowak, 2012]. In [Haupt et al., 2011], a component-wise measurement-model with additive Gaussian noise is considered for the DS framework. For each component-wise measurement, a portion of a globally given measurement budget is taken to modulate the variance of the additive noise. DS aims to identify the few non-zero components of the signal by iteratively reducing the set of candidate non-zero locations (starting with the full set of locations) via non-negative thresholding of the noisy measurements. As for each measurement the global measurement budget is divided by the number of retained candidate non-zero locations, each reduction increases sensing accuracy. However, the DS approach proposed in [Haupt et al., 2011] is not compressive as it performs $\mathcal{O}(N)$ measurements [Haupt and Nowak, 2012].

A corresponding extension, Compressive Distilled Sensing (CDS), is proposed in [Haupt et al., 2009]. Similar to DS, a set of candidate non-zero locations of the signal is iteratively reduced. In each iteration multiple measurements of the signal are collected by an undercomplete sensing matrix. The measurement model includes additive standard Gaussian noise. The entries in the columns which correspond to candidate non-zero locations are drawn i.i.d. from a zero-mean Gaussian distribution whose variance increases as the number of candidate non-zero locations decreases. The remaining entries are set to zero which results in discarding the corresponding locations. In each iteration an estimate of the signal is computed by multiplying the transposed of the sensing matrix with the vector of measurements that was derived by the sensing matrix. Subsequently non-negative thresholding is performed on the signal estimate in order to reduce the set of candidate non-zero locations.

Note that the theoretical analysis of DS and CDS is done for the limiting case $N \to \infty$. It is furthermore concentrated on the support recovery of signals whose non-zero components are equal (the amplitude is assumed to be some function of $N$).

Malloy and Nowak proposed Compressive Adaptive Sense and Search (CASS) [Malloy and Nowak, 2014]. It is conceptually equivalent to our $K$-AHS algorithm. Both methods were independently developed. They are based on the idea to iteratively bisection the signal and to collect each measurement of the signal due to a sensing vector that essentially sums up signal components within a partition. Measurements resulting from equally sized partitions are compared. A fix number of the most promising partitions yielding the largest measurement magnitudes are further bisected whereas the least promising partitions are discarded. The theoretical analysis in [Malloy and Nowak, 2014] is focussed on sensing $k$-sparse signals while additive Gaussian measurement noise is present and the total sensing energy is constrained in form of a measurement budget. The authors provide sufficient conditions in terms of lower bounds on the number of measurements (which are of order $\mathcal{O}(k \log N)$) as well as lower bounds on the amplitudes of the non-zero components. Two cases are differentiated regarding these bounds: ($i$) recovery of the full support of the signal dependent on a minimum target recovery probability given the signal is non-negative and ($ii$) average recovery of a fix fraction of the support given the signal contains both positive and negative entries. The conditions depend on the signal dimensionality, the sparsity level, the measurement budget and, depending on the case, on either a minimum target recovery probability or a fraction of the support to recover.

The impact of adaptivity for sensing is judged controversially by the sensing community. On the one hand, works, proposing adaptive sensing schemes, demonstrate improvements over non-adaptive CS, particularly when measurement noise is present. On the other hand, contrary findings have been made indicating that an adaptive sensing strategy cannot be substantially better than a non-adaptive one, no matter how sophisticatedly the collected measurements are exploited [Arias-Castro et al., 2013].

## 3.3 Non-Adaptive Compressed Sensing Principle

### 3.3.1 Sensing Matrix

Standard CS measurements are non-adaptive, which means that a sensing vector $\varphi_t$ does not depend on $y_1, \ldots, y_{t-1}$. Thus, collecting all the measurements can be concisely written as a matrix-vector product $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is the sensing matrix consisting row-wise of $M$ sensing vectors. Assume that $\mathbf{x}$ is sparse in some synthesis ONB $\mathbf{\Psi}$. By using the synthesis transform basis, one can also write $\mathbf{y} = \mathbf{\Phi \Psi a} = \mathbf{\Theta a}$.

Scientific contributions in the area of Compressed Sensing are manifold. It has been intensively studied for which conditions sparse or compressible signals can be successfully recovered from linear measurements. Moreover, several signal recovery algorithms have been proposed. Here, we give a brief insight into the CS framework by selecting a few important aspects.

### 3.3.2 Signal Recovery

Recovering a sparse or compressible signal $\mathbf{x}$ from the linear measurements $\mathbf{y} = \mathbf{\Phi x}$ implies to solve an optimization problem of the form

$$\mathrm{P}_{(3.1)}: \qquad \mathbf{a}^* = \arg\min_{\mathbf{a}} \|\mathbf{a}\|_p \ , \text{ s.t. } \mathbf{a} \in \mathcal{B}_{\mathbf{\Theta}}(\mathbf{y}) \, , \tag{3.1}$$

where $\mathcal{B}_{\mathbf{\Theta}}(\mathbf{y}) = \{\mathbf{a} : \mathbf{\Theta a} = \mathbf{y}\}$ (noiseless recovery) or $\mathcal{B}_{\mathbf{\Theta}}(\mathbf{y}) = \{\mathbf{a} : \|\mathbf{\Theta a} - \mathbf{y}\|_2 \leq \varepsilon\}$ (noisy recovery) and $p \in \{0, 1\}$. For $p = 0$, the objective function is non-convex and difficult to solve exactly. In fact, the optimization problem is in general NP-hard [Boche et al., 2015]. Commonly, the $\ell_0$ minimization problem ($p = 0$) is approached by greedy algorithms, see e.g. [Blumensath et al., 2012, Boche et al., 2015, Blanchard and Tanner, 2015] for a survey. A popular alternative strategy is to solve $\mathrm{P}_{(3.1)}$ for $p = 1$. This $\ell_1$ minimization approach relaxes the non-convex problem to a convex one such that tractable optimization algorithms can be deployed to find optimal solutions. Indeed, conditions have been elaborated which guarantee $\ell_0/\ell_1$ equivalence, which means that a solution to $\mathrm{P}_{(3.1)}$ for $p = 1$ coincides with a solution for $p = 0$ [Donoho, 2005, Candès et al., 2006, Donoho and Elad, 2003].

### 3.3.3 Restricted Isometry Property

One central property which is frequently used for theoretical CS results is given by:

**Definition 11** (Restricted Isometry Property)**.** A matrix $\mathbf{\Theta}$ satisfies the restricted isometry property (RIP) of order $k$ if there exists a $0 \leq \delta_k \leq 1$ such that

$$(1 - \delta_k) \|\mathbf{a}\|_2^2 \leq \|\mathbf{\Theta a}\|_2^2 \leq (1 + \delta_k) \|\mathbf{a}\|_2^2 \tag{3.2}$$

holds for all $\mathbf{a} : \|\mathbf{a}\|_0 \leq k$.

The RIP states that matrix $\boldsymbol{\Theta}$ approximately preserves the distances between any pair of $k$-sparse vectors [Davenport et al., 2012].

### 3.3.4 Example of a RIP Based Recovery Result

To give a prototypical example of a RIP based signal recovery result with $\ell_0/\ell_1$ equivalence, we cite the following theorem:

**Theorem 2** (Theorem 1.1 of [Candes, 2008])**.** *Suppose that* $\boldsymbol{\Theta}$ *satisfies the RIP of order* $2k$ *with* $\delta_{2k} < \sqrt{2} - 1$ *and we obtain measurements of the form* $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x} = \boldsymbol{\Theta}\mathbf{a}$*, then the solution* $\mathbf{a}^*$ *to* $\mathrm{P}_{(3.1)}$ *for* $p = 1$ *and* $\mathcal{B}_{\boldsymbol{\Theta}}(\mathbf{y}) = \{\mathbf{a} : \boldsymbol{\Theta}\mathbf{a} = \mathbf{y}\}$ *obeys*

$$\|\mathbf{a}^* - \mathbf{a}\|_2 \leq C_0 \frac{\|\mathbf{a} - \mathcal{S}_k(\mathbf{a})\|_1}{\sqrt{k}}, \tag{3.3}$$

*where* $\mathcal{S}_k(\mathbf{a})$ *is the optimal* $k$*-sparse approximation*[2] *of* $\mathbf{a}$*. In particular, if* $\mathbf{a}$ *is* $k$*-sparse, the recovery is exact.*

Theorem 2 bounds the approximation error of the recovered signal due to an $\ell_1$ minimization for the noiseless scenario. In [Candes, 2008], Theorem 1.2 states a similar condition for the noisy scenario, where $\mathcal{B}_{\boldsymbol{\Theta}}(\mathbf{y}) = \{\mathbf{a} : \|\boldsymbol{\Theta}\mathbf{a} - \mathbf{y}\|_2 \leq \varepsilon\}$.

Finally, we outline that the number of measurements required for a successful recovery of a sparse signal, i.e. the required number of rows for the sensing matrix, can be very small if the sensing matrix consists of randomized entries. For instance, a sensing matrix composed of $M = \mathcal{O}(k \log(N/k)/\delta_{2k}^2)$ rows, drawn from a sub-Gaussian distribution, will satisfy the RIP of order $2k$ with probability at least $1 - 2\exp(-C_1\delta_{2k}^2 M)$ [Davenport et al., 2012].

In practice, CS measurements are made essentially independent from the analysis domain. Thus, the collected measurements $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$ can be stored and a suitable transform $\boldsymbol{\Psi}$ does not need to be selected until it comes to recovering the signal.

## 3.4 Adaptive Hierarchical Sensing Principle

For AHS, the transform pair $\boldsymbol{\Psi}, \overline{\boldsymbol{\Psi}}$ is selected prior to sampling. This is required, because the sensing vectors are composed of analysis basis vectors. AHS partially traverses a so called sensing tree (see Section 3.5) and collects for each visited node, one linear measurement of the signal with a node-specific sensing vector. During the traversal of the sensing tree, decisions based on previously observed measurements, are made whether to descend or not to descend subtrees rooted at individual nodes.

[2] The optimal $k$-sparse approximation of $\mathbf{a}$ is obtained by setting all but the $k$ largest absolute entries of $\mathbf{a}$ to zero.

The sensing scheme is adaptive as a sensing operation, except for the initial level, is only made if large measurements are collected at corresponding ancestor nodes. Furthermore, AHS operates hierarchically. The transition from high level nodes (near the root) to low level nodes (near the leaves) corresponds to a gradual refinement of initially coarse measurements towards a set of significant signal coefficients in the analysis transform basis $\mathbf{\Psi}$.

## 3.5 Sensing Tree

Suppose signal dimensionality $N$ is a power of 2. The key data structure underlying AHS is a so called sensing tree. It is a perfect binary tree of height $\log_2 N$ with $2N-1$ nodes. Each node of the tree is indexed by a tuple $(l, n)$ and is associated with a sensing vector $\varphi_{l,n}$, where $l = 0, \dots, \log_2 N$ is the index of the tree level (starting at the bottom level), and $n = 1, \dots, N2^{-l}$ is the index of the node within level $l$.

### Sensing Vector Composition for Signal Dimensionalities $N = 2^{N_0}$

The sensing vectors of the bottom level correspond to elements of analysis basis $\mathbf{\Psi} = (\psi_1, \dots, \psi_N)$ in which $\mathbf{x}$ is assumed to have a sparse representation, i.e.

$$\varphi_{0,n} = \psi_n, \quad n = 1, \dots, N\,. \tag{3.4}$$

In a bottom-up manner, the sensing vector of each internal node is the sum of sensing vectors assigned to its two direct descendant nodes, i.e. for any $l \in \{1, ..., \log_2 N\}$

$$\varphi_{l,n} = \varphi_{l-1,2n-1} + \varphi_{l-1,2n}, \quad n = 1, \dots, N2^{-l}\,. \tag{3.5}$$

Note that by construction, $\varphi_{l,n}$ can also be written directly as the sum of a subset of basis vectors from $\mathbf{\Psi}$:

$$\varphi_{l,n} \quad = \sum_{i=(n-1)2^l+1}^{n2^l} \psi_i\,. \tag{3.6}$$

The set of analysis basis vectors that forms $\varphi_{l,n}$ corresponds to the leaves of the subtree with root node $(l, n)$. Figure 3.2 illustrates the sensing tree schematically.

For each node $(l, n)$ that is visited, one linear measurement is collected by the sensing operation $\langle \mathbf{x}, \varphi_{l,n} \rangle$, i.e. by the inner product between the unknown signal $\mathbf{x}$ and the node specific sensing vector $\varphi_{l,n}$. Note that due to (3.6) and the bilinearity of the inner product for real vector spaces, a sensing operation implicitly computes the sum

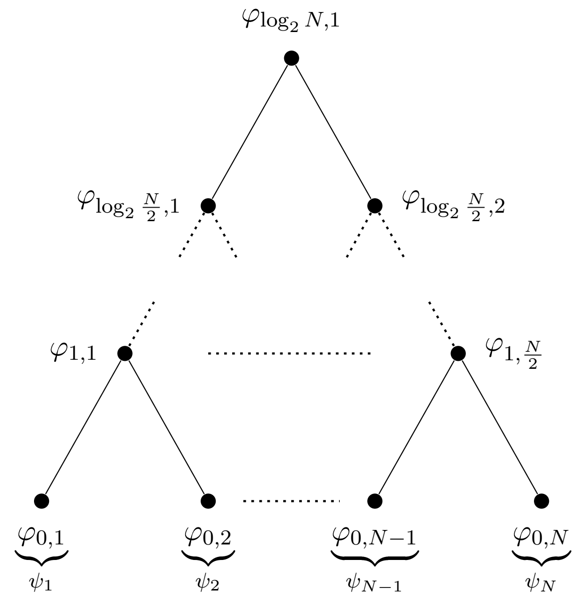


Figure 3.2: Schematic illustration of the AHS sensing tree. To each node $(l, n)$ a sensing vector $\varphi_{l,n}$ is assigned. The first index $l \in \{0, ..., \log_2 N\}$ indicates the tree level starting with $l = 0$ at the bottom level. The second index $n \in \{1, ..., N2^{-l}\}$ is the node index for level $l$. Each analysis basis vector is assigned to exactly one leaf node.

of signal coefficients in the sparse transform domain $\mathbf{\Psi}$, i.e.

$$\langle \mathbf{x}, \varphi_{l,n} \rangle \quad = \quad \sum_{i=(n-1)2^l+1}^{n2^l} a_i \, . \tag{3.7}$$

### Sensing Vector Composition for Arbitrary Signal Dimensionalities $N \neq 2^{N_0}$

To handle a signal dimensionality $N$ which is not a power of 2, we propose to expand the analysis basis $\mathbf{\Psi}$ by $\tilde{N} - N$ additional zero vectors, where $\tilde{N} = 2^{\lceil \log_2 N \rceil}$. The expanded analysis basis $\tilde{\mathbf{\Psi}} \in \mathbb{R}^{N \times \tilde{N}}$ is then given by

$$\tilde{\mathbf{\Psi}} = \left( \mathbf{\Psi}, \mathbf{0}_{N \times (\tilde{N}-N)} \right) \, .$$

The size of the sensing tree, i.e. the number of nodes will be increased due to the additional zero vectors. However, the vast majority of additional nodes will be automatically discarded very early during sensing as they provide merely zero measurements. For the reconstruction, only the $N$ original dimensions will be used. The artificial $\tilde{N} - N$ components of $\hat{\mathbf{a}}$ will be zero and can be discarded such that the original synthesis transform $\overline{\mathbf{\Psi}}$ is used for the reconstruction as described in Section 3.6.

## 3.6 Signal Recovery

AHS does not require a sophisticated recovery procedure for the sparse representation of the signal. At the bottom level of the sensing tree, AHS directly measures some entries of **a**, namely one coefficient for each visited leaf node. These coefficients are used to build $\hat{\mathbf{a}} \in \mathbb{R}^N$ as the estimation of **a**. The remaining entries of $\hat{\mathbf{a}}$, which correspond to unvisited leaf nodes, are set to zero. Let $H \subseteq \{1, ..., N\}$ be the index set of all visited leaf nodes. Then,

$$\hat{a}_h = \begin{cases} a_h = \langle \mathbf{x}, \psi_h \rangle \,, & \text{if } h \in H \\ 0, & \text{otherwise}\,. \end{cases} \tag{3.8}$$

Finally, the approximated signal is obtained by $\hat{\mathbf{x}} = \overline{\mathbf{\Psi}}^T \hat{\mathbf{a}}$. Note that AHS differs in an important point from CS as no inverse optimization problem, such as $\mathrm{P}_{(3.1)}$, has to be solved to obtain $\hat{\mathbf{a}}$.

### 3.6.1 Exploitation of Internal Measurements

AHS measurements collected at internal nodes are utilized solely for the decisions to descend or to omit the subtree of a node. For the reconstruction, only the measurements of the leaf nodes are taken into account. Intuitively, measurements of internal nodes should contain (at least a small amount of) additional information about the signal. It is thus natural to investigate to which extent the signal recovery can be improved by use of these internal measurements. We study two modifications of the direct signal recovery as described in Section 3.6, which we denote *AHS Modification A* and *AHS Modification B*, respectively.

**AHS Modification A** A straight forward approach to improve the standard AHS reconstruction (3.8) by the additional use of internal measurements is to evenly divide the observed measurement of each non-winner node, whose rooted subtree is omitted, over all its corresponding leaf components. Thus, the estimate of **a** does not necessarily have zero entries for these unvisited leaf nodes. We show in the following that, if $\mathbf{\Psi}$ is an ONB, the reconstructed signal has a smaller approximation error.

*Remark.* Suppose $y = \sum_{j \in \mathcal{J}} a_j$ is an insignificant measurement from an internal node whose rooted subtree is not further processed. Let $\mathcal{J} \subseteq \{1, \ldots, N\}$ be the corresponding set of discarded leaf nodes. Suppose $\hat{\mathbf{a}}$ is the standard AHS estimate of **a** according to (3.8) and $\tilde{\mathbf{a}}$ is a modified AHS estimate with the same entries as $\hat{\mathbf{a}}$ for $j \in \overline{\mathcal{J}}$, but different entries for $j \in \mathcal{J}$. If $\mathbf{\Psi}$ is an ONB, then setting for $j \in \mathcal{J}$

$$\tilde{a}_j = \frac{y}{|\mathcal{J}|} \quad \text{as opposed to} \quad \hat{a}_j = 0 \quad \text{(as for standard AHS)} \tag{3.9}$$

yields a better approximation of signal **x**, i.e. $\left\|\mathbf{\Psi}^T \tilde{\mathbf{a}} - \mathbf{x}\right\|_2^2 \leq \left\|\mathbf{\Psi}^T \hat{\mathbf{a}} - \mathbf{x}\right\|_2^2$.

*Proof.* Due to orthonormality of $\boldsymbol{\Psi}$, we have $\left\|\boldsymbol{\Psi}^T\tilde{\mathbf{a}}-\mathbf{x}\right\|_2^2 \leq \left\|\boldsymbol{\Psi}^T\hat{\mathbf{a}}-\mathbf{x}\right\|_2^2 \Leftrightarrow \|\tilde{\mathbf{a}}-\mathbf{a}\|_2^2 \leq \|\hat{\mathbf{a}}-\mathbf{a}\|_2^2$. Thus, it is sufficient to exclusively consider the coefficients. It is even sufficient to consider only the reduced coefficient vectors subject to index set $\mathcal{J}$, i.e. $\tilde{\mathbf{a}}_{\mathcal{J}}$ and $\hat{\mathbf{a}}_{\mathcal{J}}$, as they are otherwise equal. Hence,

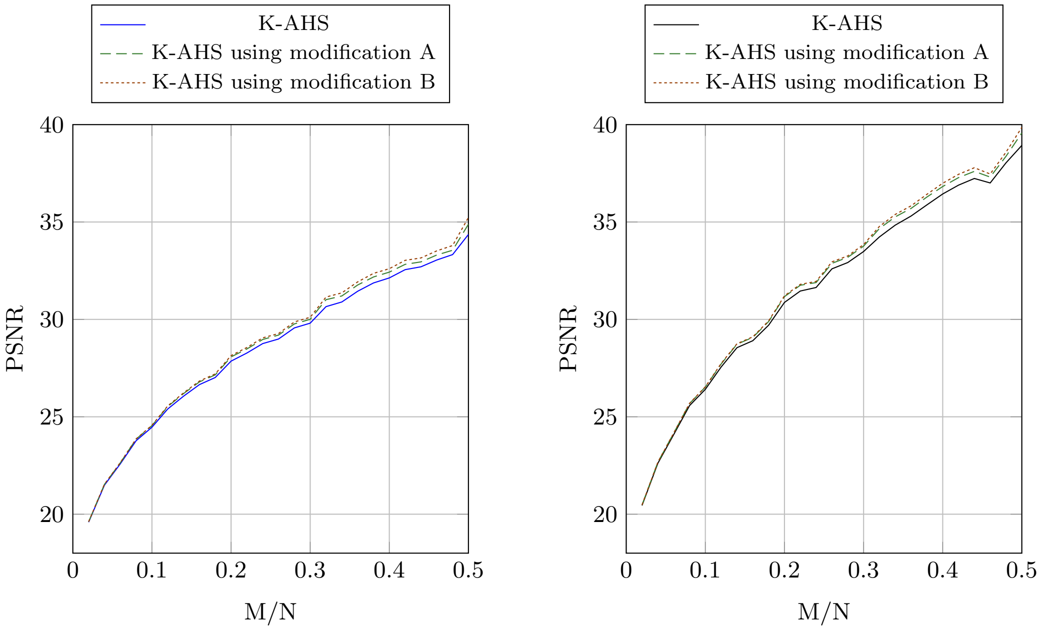
$$\begin{aligned}
\|\tilde{\mathbf{a}}_{\mathcal{J}}-\mathbf{a}_{\mathcal{J}}\|_2^2 &\overset{!}{\leq} \|\mathbf{a}_{\mathcal{J}}\|_2^2 && (3.10)\\
\Leftrightarrow \sum_{j\in\mathcal{J}}\left(\frac{y}{|\mathcal{J}|}-a_j\right)^2 &\overset{!}{\leq} \sum_{j\in\mathcal{J}} a_j^2 && (3.11)\\
\Leftrightarrow \sum_{j\in\mathcal{J}}\left(\frac{y^2}{|\mathcal{J}|^2}-2\frac{y}{|\mathcal{J}|}a_j+a_j^2\right) &\overset{!}{\leq} \sum_{j\in\mathcal{J}} a_j^2 && (3.12)\\
\Leftrightarrow \sum_{j\in\mathcal{J}}\frac{y^2}{|\mathcal{J}|^2} &\overset{!}{\leq} 2\frac{y}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} a_j && (3.13)\\
\Leftrightarrow \frac{y^2}{|\mathcal{J}|} &\leq 2\frac{y^2}{|\mathcal{J}|}\,. && (3.14)
\end{aligned}$$

Note that $y \neq 0 \Rightarrow \|\tilde{\mathbf{a}}-\mathbf{a}\|_2^2 < \|\hat{\mathbf{a}}-\mathbf{a}\|_2^2$. □

Observe that this remark takes only one omitted subtree, represented by $\mathcal{J}$, into account. However, the argument can be simultaneously applied to all omitted subtrees $\mathcal{J}_1, \mathcal{J}_2, \ldots$ because the corresponding coefficient subsets are disjoint.

**AHS Modification B** A second approach to improve the standard AHS reconstruction (3.8) by the additional use of internal measurements would be to solve an inverse optimization problem such as $\mathrm{P}_{(3.1)}$ for $p = 1$. The sensing matrix $\boldsymbol{\Theta}$ would consist row-wise of all the sensing vectors $\varphi_{l,n}$ corresponding to visited leaf nodes ($l = 0$) as well as internal nodes whose subtrees are omitted and $\mathbf{y}$ would contain the correspondingly observed measurements. Of course, this approach undermines the AHS principle of directly sensing signal coefficients in the analysis domain which is a primary selling point of AHS. However, by this modification we can more intuitively grasp what can ideally get out of the (usually discarded) non-winner measurements.

Figure 3.3 illustrates a comparison of the sensing performance between *AHS modification A*, *AHS modification B* and the standard AHS reconstruction (3.8) as obtained by $K$-AHS (a variant of AHS introduced in Section 3.8) for the test image *Cameraman* (see Figure 3.13). It can be seen that the PSNR increases slightly due to the modifications. Although the gain of approximation performance is not overwhelming, the PSNR difference increases with the number of measurements. The PSNR by *AHS Modification B* is the highest. However, the PSNR difference between *AHS Modification B* and *AHS Modification A* is smaller than between *AHS Modification A* and the standard reconstruction (3.8). Interestingly, the $K$-AHS approximation performance due to *AHS Modification A* is also improved for the biorthogonal CDF97 basis which



(a) Haar basis

(b) CDF97 basis

Figure 3.3: $K$-AHS sensing performance comparison for test image *Cameraman* ($N = 2^{18}$) between the standard AHS reconstruction (3.8) and the two AHS modifications A and B as described in Section 3.6.1. The PSNR is plotted as a function of the relative number of measurements.

is not covered by the above remark.

## 3.7 $\tau$-AHS Algorithm

In the following we present the $\tau$-AHS algorithm [Schütze et al., 2014], which selectively traverses the sensing tree in a pre-order fashion. For each visited node the magnitude of the collected measurement is compared to a threshold $\tau$. If the magnitude of the measurement exceeds the threshold, the subtree is further traversed, i.e. the direct descendant nodes are going to be visited. Otherwise, the subtree of that node is omitted and remains unvisited. If a subtree is omitted or a leaf node is reached, the next candidate node is processed. To account for different partition sizes, threshold $\tau$ is adjusted depending on the level index of the node, i.e. $\tau = \sqrt{2^l}\tau_0$, where $\tau_0$ represents a canonical threshold, a user parameter. This parameter allows to control implicitly how many signal coefficients are going to be identified. The $\tau$-AHS pseudo code is listed in Algorithm 4. Note that the use of the stack data structure $s$ results in a selective pre-order traversal. Basically, any other tree traversal scheme can be used alternatively.

An adequate choice of $\tau_0$ is crucial for detecting relevant signal coefficients. A too small value $\tau_0$ can result in a large number of measurements and a non-sparse estimate $\hat{\mathbf{a}}$, e.g. if the representation $\mathbf{a}$ is not strictly $k$-sparse but compressible. A too large value $\tau_0$, on the other hand, implies a strict omission policy and can result in the trivial estimate $\hat{\mathbf{a}} = \mathbf{0}_N$ which is obtained if no leaf node is reached.

The initial tree level $L$ is an optional user parameter. Its default value $L = \log_2 N$ causes $\tau$-AHS to collect the first measurement at the root node. For smaller values $L$, all nodes of the corresponding initial level have to be processed.

**Algorithm 4** Adaptive Hierarchical Sensing ($\tau$-AHS)

```
Input: Sensing tree $\{\varphi_{l,n}\}$,
    Canonical threshold $\tau_0 \geq 0$,
    Initial sensing tree level $L \in \{0, \dots, \log_2 N\}$ (optional, default: $L = \log_2 N$)
Output: Approximation $\hat{\mathbf{a}}$ of $\mathbf{a}$
Initialize $\hat{\mathbf{a}} \leftarrow \mathbf{0}_N$
Initialize empty stack $s$
for all nodes of level $L$: $n = 1, \dots, N2^{-L}$ do
    Collect measurement $y \leftarrow \langle \varphi_{L,n}, \mathbf{x} \rangle$
    Push tuple $(L, n, y)$ to stack $s$
end for
while stack $s$ is not empty do
    Pop tuple $(l, n, y)$ from stack $s$
    if $l = 0$ then
        $\hat{a}_n \leftarrow y$
    else
        Set threshold according to size of subtree: $\tau \leftarrow \sqrt{2^l}\tau_0$
        if $|y| > \tau$ then
            Collect measurement for left child node: $y \leftarrow \langle \varphi_{l-1,2n-1}, \mathbf{x} \rangle$
            Push tuple $(l-1, 2n-1, y)$ to stack $s$
            Collect measurement for right child node: $y \leftarrow \langle \varphi_{l-1,2n}, \mathbf{x} \rangle$
            Push tuple $(l-1, 2n, y)$ to stack $s$
        end if
    end if
end while
```

### 3.7.1 Measurement Bound for $k$-Sparse Signals

In the noiseless scenario, where signal representation $\mathbf{a}$ is $k$-sparse, the canonical threshold should be set to $\tau_0 = 0$. Furthermore, suppose there is no subset of non-zero coefficients that sums up to zero, which holds almost surely if the non-zero coefficients stem from a continuous probability distribution. In this particular setting the signal can be perfectly sensed by $\tau$-AHS and we are able to determine upper and lower bounds on the number of measurements.

For $k = 1$ and $L = \log_2 N$, $\tau$-AHS requires $2\log_2 N + 1$ measurements in order to track down the only non-zero coefficient. For $k > 1$ we have two limiting cases which yield the lower bound and the upper bound respectively.

#### Lower Bound

The lowest number of $\tau$-AHS measurements arises, if all $k$ non-zero coefficients are most tightly clustered within one subtree. Such a subtree has at least $k$ leaves, and within this subtree, at least $2k - 1$ nodes need to be visited. In addition we have to visit the nodes on the way from the root of the AHS tree to the root of the subtree. Note that the latter need to be counted twice, because one additional measurement per node is required in order to decide to omit all other subtrees. Hence, the *lower bound* on the required number of measurements for perfectly sensing a $k$-sparse signal is $2\log_2(N/k) + 2k - 1$.

#### Upper Bound

The highest number of $\tau$-AHS measurements arises, if the $k$ non-zero coefficients are maximally scattered over the leaves. This leads to $k$ disjoint subtrees of equal size, each carrying exactly one non-zero coefficient. The number of leaves of each of these $k$ subtrees is at most $N/k$. Consequently, the number of measurements within each subtree is at most $2\log_2(N/k) + 1$. Starting from the root of the AHS tree, $k - 1$ measurements are required to reach the roots of these subtrees. Hence, the *upper bound* on the number of measurements is $2k\log_2(N/k) + 2k - 1$.

## 3.8 $K$-AHS Algorithm

In the following we present $K$-AHS [Schütze et al., 2017], which selectively traverses the sensing tree level by level based on relative comparisons of measurements collected in a level. The $K$-AHS algorithm has a user parameter $K$. This parameter allows to explicitly control how many signal coefficients are going to be identified. Furthermore, it determines how many nodes $K$-AHS takes into consideration when it transitions from one level to the next. The direct descendants of the nodes corresponding to the $K$ largest[3] measurements are visited in the next iteration. Thus, there are $2K$ sensing operations for the new level from which again the nodes coinciding with the $K$ largest measurements are further processed. This iterative scheme is continued until $2K$ leaf nodes are reached. The $K$-AHS pseudo code is listed in Algorithm 5. The idea is that, by this procedure, the $K$ largest entries of $\mathbf{a} = \mathbf{\Psi}\mathbf{x}$ are collected. For instance, if $\mathbf{a}$ has at most $K$ non-zero entries (without any subset summing up to exactly zero, e.g., when drawn from a continuous probability distribution), then the signal is completely sensed and can be perfectly reconstructed. In Section 3.10 we investigate further models of compressive signals.

### 3.8.1 Initial Sensing Tree Level

In order to avoid unnecessary sensing operations, $K$-AHS does not start with the first measurement at the root node of the sensing tree but at a suitable initial level $L$. This initial level has to be sensed completely in order to identify the $K$ nodes providing the largest measurements for processing the next level. At each subsequent level $l < L$,

[3] In the following, for $K$-AHS measurements the relation *larger* and *smaller* is exclusively meant in terms of their magnitude.

**Algorithm 5** Adaptive Hierarchical Sensing ($K$-AHS)

```
Input: Sensing tree $\{\varphi_{l,n}\}$,
    Target sparsity level $K < \frac{N}{4}$
    Initial sensing tree level $L \in \{0, \ldots, \log_2 N\}$ (optional, default: $L = \log_2 N - \lfloor \log_2 K \rfloor - 2$)
Output: Approximation $\hat{\mathbf{a}}$ of $\mathbf{a}$, where $\|\hat{\mathbf{a}}\|_0 \leq 2K$
Set $\mathcal{Y}_L \leftarrow \emptyset$
for $n = 1 \ldots, N2^{-L}$ (all nodes in level $L$) do
    $y_{L,n} \leftarrow \langle \mathbf{x}, \varphi_{L,n} \rangle$
    $\mathcal{Y}_L \leftarrow \mathcal{Y}_L \cup \{(n, y_{L,n})\}$
end for
for $l = L - 1, ..., 0$ (all subsequent levels) do
    Let $n_1, ..., n_K$ be the subscripts of the $K$ largest measurements in $\mathcal{Y}_{l+1}$
    Set $\mathcal{Y}_l \leftarrow \emptyset$
    for $j = 1, ..., K$ (the $K$ largest measurements of level $l + 1$) do
        collect the measurements of the two child nodes of $(l + 1, n_j)$
            $$\begin{aligned} y_{l,2n_j-1} &\leftarrow \left\langle \mathbf{x}, \varphi_{l,2n_j-1} \right\rangle \\ y_{l,2n_j} &\leftarrow \left\langle \mathbf{x}, \varphi_{l,2n_j} \right\rangle \\ \mathcal{Y}_l &\leftarrow \mathcal{Y}_l \cup \left\{ (2n_j - 1, y_{l,2n_j-1}), (2n_j, y_{l,2n_j}) \right\} \end{aligned}$$
    end for
end for
for $n = 1, \ldots, N$ (all signal coefficient indices) do
    Set
        $$\hat{a}_n \leftarrow \begin{cases} y_{0,n} & \text{if contained in } \mathcal{Y}_0 \\ 0 & \text{otherwise} \end{cases}$$
end for
```

only $2K$ measurements are collected. Regarding the total number of measurements the optimal initial tree level depends on the user parameter $K$ and is given by

$$L = \log_2 N - \lfloor \log_2 K \rfloor - 2 \,. \tag{3.15}$$

$L$ is the highest level $l \in \{0, \ldots, \log_2 N/4\}$ that contains more than $2K$ nodes. For example, for $K = 1$ we start with the level $L = \log_2 N/4$ which contains 4 nodes. For $N/4 \leq K \leq N/2$ we obtain $L = 0$ and $N$ measurements, a trivial scenario where each coefficient is sensed individually. This shows that K-AHS makes sense only for small values of $K$, i.e., sparse signals.

### 3.8.2 Measurement Bound

$K$-AHS has a sampling complexity of the same order as Compressed Sensing.

**Theorem 3.** *Let $\mathbf{x} \in \mathbb{R}^N$ and $1 \leq K < N/4$. For $M$, the total number of $K$-AHS measurements, the following bound holds*

$$M \leq 2K \log_2 \frac{N}{K} \,. \tag{3.16}$$

*Proof.* According to Algorithm 5, $K$-AHS entirely processes the initial level $L$ of the sensing tree, which results in $N2^{-L}$ measurements. There are $L$ subsequent levels, each adds $2K$ measurements. Hence,

$$M = N2^{-L} + 2KL\,. \tag{3.17}$$

Plugging (3.15) into (3.17) yields

$$\begin{aligned} M &= 2^{\lfloor \log_2 K \rfloor + 2} + 2K(\log_2 N - \lfloor \log_2 K \rfloor - 2) && (3.18)\\ &\leq 2^{\log_2 K + 2} + 2K(\log_2 N - \log_2 K - 2) && (3.19)\\ &\leq 2K \log_2 \frac{N}{K}\,. \end{aligned}$$

For (3.19), we have used the inequality

$$2^{\lfloor \log_2 K \rfloor + 2} - 2K \lfloor \log_2 K \rfloor \leq 2^{\log_2 K + 2} - 2K \log_2 K\,.$$

□

Equality in (3.16) holds if $K \in \{1, 2, 4, 8, \dots\}$.

## 3.9 Optimizing Sensing Tree Structure

### 3.9.1 Introducing Weights

According to (3.5), the sensing vector of an internal node of the sensing tree is constructed by the sum of the sensing vectors assigned to its direct descendant nodes. It might be useful to generalize (3.5) such that the direct sum becomes a weighted sum:

$$\varphi_{l,n} = \alpha_{l,n}\, \varphi_{l-1,2n-1} + \beta_{l,n}\, \varphi_{l-1,2n}\,, \tag{3.20}$$

where $\alpha_{l,n}$ and $\beta_{l,n}$ are real non-zero weights. Suppose we intend to sense signals from a class with particular statistical properties. It might be possible to optimize these weights in order to improve AHS performance. For instance, if the measurements provided by two sibling nodes $\varphi_{l,2n-1}$ and $\varphi_{l,2n}$ are strongly anti-correlated, it would be advantageous to choose weights $\alpha_{l,n}$ and $\beta_{l,n}$ with opposite signs as this increases the magnitude of the observable measurement. In this setting a sensing vector $\varphi_{l,n}$ can be written as a weighted sum of analysis basis vectors, analogous to (3.6):

$$\varphi_{l,n} = \sum_{i=(n-1)2^l+1}^{n2^l} \left( \prod_{l'=0}^{l} \alpha_{l', \left\lfloor \frac{i}{2^{l'}} \right\rfloor} \right) \psi_i\,. \tag{3.21}$$

### 3.9.2 Reordering Analysis Basis

Furthermore, a suitable order of analysis basis vectors can improve the AHS performance. For instance, if the most significant coefficients (with equal signs) are sibling nodes at the bottom level, then measurements can be saved to find them ($\tau$-AHS) or, if the number of measurements is fix ($K$-AHS), other significant coefficients can be revealed. Hence, it is desirable to cluster analysis basis vectors, which are likely to yield co-occurring coefficients of large magnitude within small subtrees.

### 3.9.3 Greedy Approach to Optimize Sensing Tree Structure

Suppose we intend to sample signals of a particular signal class using AHS and we have given a training data set representing that population. What is a suitable strategy to build up the sensing tree? It is natural to construct the sensing tree in a bottom up fashion. Thus, to compose level $l = 1$, we look at first for the pair $\psi_i$, $\psi_j$ of analysis basis vectors such that its optimally weighted linear combination yields the sensing vector $\alpha_i\psi_i + \beta_j\psi_j$ that maximizes the expectation of a large measurement magnitude. We repeat this step for the remaining nodes of level $l = 1$ subject to the residual analysis basis vectors which were not yet selected. Subsequently, we continue with the next level $l = 2$ but now subject to the collection of the recently composed sensing vectors for level $l - 1$. We proceed analogously for the remaining levels until the sensing vector of the root node is combined.

Suppose $\mathcal{P}_{l-1}$ is the set of available sensing vectors of level $l - 1$ from which the next optimal pair can be selected to compose a new sensing vector for level $l$. We start with $l = 1$ and $\mathcal{P}_0 = \{\psi_1, \ldots, \psi_N\}$. The optimization problem we have to solve is given by

$$\mathrm{P}_{(3.22)}: \qquad \underset{\substack{\varphi_i,\varphi_j \in \mathcal{P}_{l-1},\\ \varphi_i \neq \varphi_j}}{\arg\max} \; \max_{\substack{\alpha,\beta:\\ \alpha^2+\beta^2=1}} \; \mathbb{E}\left[\langle \mathbf{x}, \alpha\varphi_i + \beta\varphi_j\rangle^2\right], \tag{3.22}$$

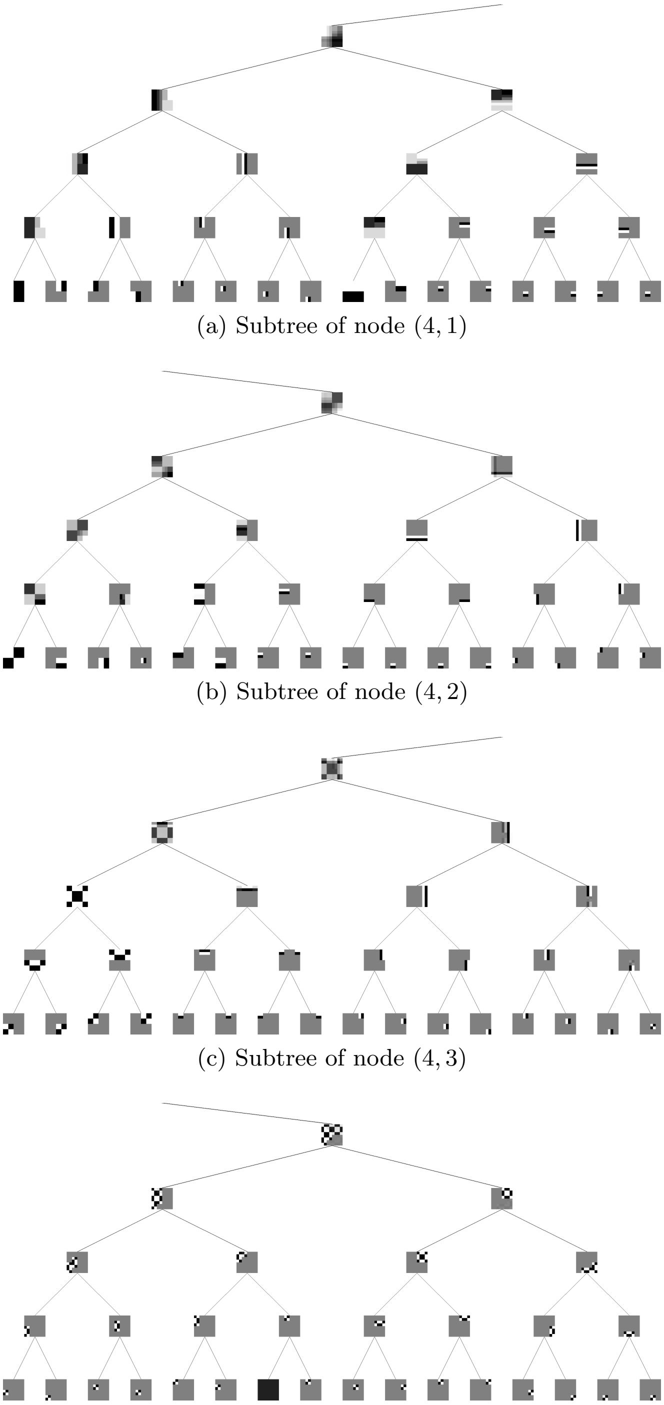
where the expectation value is taken subject to the training data set $\mathbf{X}$. Let $\alpha^*\varphi_{i^*} + \beta^*\varphi_{j^*}$ be the solution to $\mathrm{P}_{(3.22)}$. This optimally combined new sensing vector is added to $\mathcal{P}_l$ and $\varphi_{i^*}$, $\varphi_{j^*}$ are removed from $\mathcal{P}_{l-1}$. This procedure is continued until $\mathcal{P}_{l-1}$ is empty. The energy of the weighting coefficients is constrained by $\alpha^2 + \beta^2 = 1$ to ensure that all sensing vectors of one level have the same energy.

Given a fix pair of indices $i, j$, the inner subproblem of $\mathrm{P}_{(3.22)}$ can be rewritten as follows

$$\max_{\alpha,\beta} \; \mathbb{E}\left[\langle \mathbf{x}, \alpha\varphi_i + \beta\varphi_j\rangle^2\right] \qquad \text{s.t.} \qquad \alpha^2 + \beta^2 = 1 \tag{3.23}$$

$$\Leftrightarrow \max_{\alpha,\beta} \; (\alpha,\beta) \underbrace{\begin{pmatrix} c_i & c_{i,j} \\ c_{i,j} & c_j \end{pmatrix}}_{Q} (\alpha,\beta)^T \qquad \text{s.t.} \qquad \alpha^2 + \beta^2 = 1, \tag{3.24}$$

where $c_i = \mathbb{E}\left[\left(\mathbf{x}^T\varphi_i\right)^2\right]$, $c_j = \mathbb{E}\left[\left(\mathbf{x}^T\varphi_j\right)^2\right]$ and $c_{i,j} = \mathbb{E}\left[\left(\mathbf{x}^T\varphi_i\right)\left(\mathbf{x}^T\varphi_j\right)\right]$. This optimization problem has a quadratic form as well as a unit energy constraint on the variables. Hence, the solution is given by the eigenvector of the largest eigenvalue of matrix $Q$.



(a) Subtree of node $(4, 1)$

(b) Subtree of node $(4, 2)$

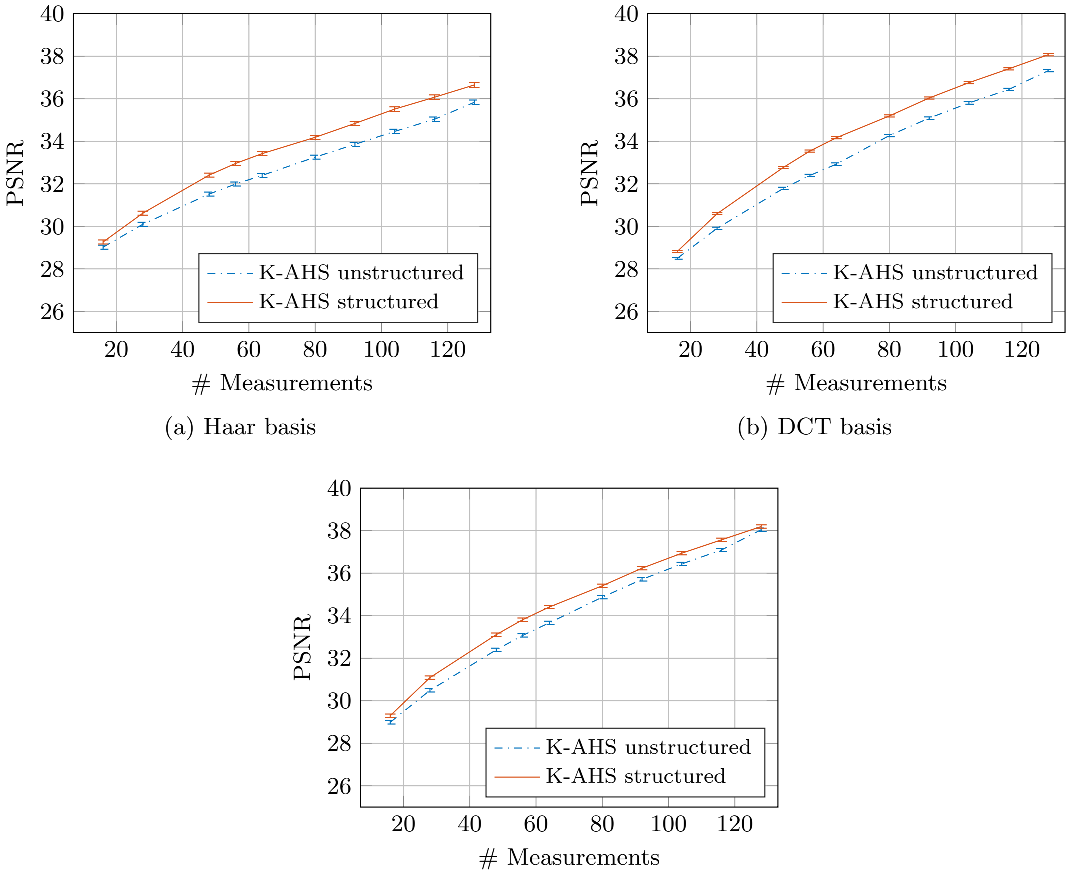
(c) Subtree of node $(4, 3)$

(d) Subtree of node $(4, 4)$

Figure 3.4: Sensing vectors of an AHS sensing tree with optimized structure as described in Section 3.9.3. The structure is learned for the non-standard 2D Haar wavelet basis ($N = 256$) and based on the NSSiVS training data set containing natural image patches (see Section 2.10.1).

### 3.9.4 Optimized Sensing Tree Structure for Natural Image Patches

Consider the problem of sensing natural image patches by AHS subject to a particular basis, e.g. the 2D Haar wavelet basis. Figure 3.4 illustrates which sensing vectors emerge by applying the proposed structural optimization procedure for this task. The training data set $\mathbf{X}$ was obtained as described in Section 2.10.1, where $N = 64$ and $L = 10^5$. Due to limited space for picturing the whole structured sensing tree, Figure 3.4 illustrates all subtrees of level $l = 4$. Observe that the minimization of $\mathrm{P}_{(3.22)}$ indeed clusters basis functions of the same orientation selectivity in common subtrees, see e.g. the disjoint subtrees of $\varphi_{3,n}$, $n \in \{1, 2, 6, 7, 8\}$ and $\varphi_{2,n}$, $n \in \{6, 7, 8, 9, 10\}$. Moreover, notice that the localized basis functions, that have adjacent support (same frequency and orientation), are combined to assemble sensing vectors which form extended edge detectors: $\varphi_{2,n}$, $n \in \{2, 4, 7, 8, 10, 11, 12\}$ or regular grid-like structures: $\varphi_{4,4}$. Interestingly, the proposed procedure chooses weights with opposite signs which yields more



(a) Haar basis

(b) DCT basis

(c) $N$-OSC basis

Figure 3.5: $K$-AHS sensing performance comparison between a random sensing tree structure and an optimized sensing tree structure for the NSSiVS test data set containing natural image patches ($N = 256$). The PSNR is plotted as a function of the absolute number of measurements. Error bars indicate the standard error.

regular structures, see e.g. $\varphi_{2,9}$, $\varphi_{2,10}$ or $\varphi_{4,4}$. While the magnitudes of the weights are rather equal when those regular structures are composed, in some cases one weight has a significant higher magnitude than the other, which thus enhances the sensing vector structure of one child node whereas the other is muted for the combination, see e.g. $\varphi_{2,3}$.

To demonstrate that minimizing $\mathrm{P}_{(3.22)}$ indeed yields a tree structure that improves AHS performance, we conducted sensing experiments with $K$-AHS on natural image patches subject to three different sparse coding ONBs. Adaptive sensing with an optimized AHS tree structure is compared to a random AHS tree structure, for which unit weights are used and the leaf nodes are randomly shuffled. We used 8 bit gray level variants (with centered intensity values) of the training and test data sets described in Section 2.10.1. The patches of the training data set were used to optimize the sensing tree structure and the patches of the test data set were sensed by $K$-AHS subject to the structured and randomized tree. Figure 3.5 illustrates the average PSNR of the sensed image patches from the NSSiVS test data ($N = 256$, $L \approx 2.3 \cdot 10^4$). For each of the three sparse coding bases (Haar, DCT, $N$-OSC), the optimized sensing tree structure yields consistently a higher sensing performance compared to the random sensing tree. The improvements of the PSNR are $0.24 - 1.05$ dB for the Haar basis, $0.32 - 1.24$ dB for the DCT basis, and $0.14 - 0.74$ dB for the $N$-OSC basis.

## 3.10 Analyzing $K$-AHS Sensing Performance

In this section we investigate the sensing performance of $K$-AHS theoretically.

When a signal $\mathbf{x}$ is sensed by $K$-AHS, the best result one can expect is that the $K$ largest entries of the signal representation $\mathbf{a}$ are collected. Whether this optimal result is achieved depends on the constellation of the coefficients within the individual partitions. More precisely, there are situations in which the magnitude of a measurement is small although the corresponding partition contains significant coefficients. Due to e.g. unfavorable sign constellations such significant coefficient can cancel each other and could get lost. In the following we study $K$-AHS from a theoretical perspective in order to provide an insight when it performs reliably.

We consider the problem from two different perspectives, i.e. a deterministic perspective, for which a success scenario is guaranteed, and a probabilistic perspective, for which a success scenario arises with a high probability. We do not restrict the support of the sparse signal representation $\mathbf{a}$. We assume that the magnitudes of the signal coefficients have some sufficiently strong decay, which can be grasped as dealing with compressible signals.

In the deterministic consideration, we derive a sufficient optimality condition, i.e. a condition which guarantees that $K$-AHS finds the $k$ signal coefficients with the largest magnitudes. Locations and signs of the signal coefficients can be arbitrary in this setting. We introduce a few signal models which define some particular decay of the coefficient magnitudes dependent on a model parameter. Our sufficient condition is applied to these signal models, which allows to deduce sufficient optimality conditions for $K$-AHS depending on the model parameter. In the probabilistic setting, we relax the deterministic condition and take the assignment of coefficients into account. The probability of a fail scenario is bounded by the probability that a critical number of significant coefficients is exceeded for any initial partition. In this setting, we additionally assume that the locations of the signal coefficients are uniformly distributed.

We assume a noise free measurement model, which is equivalent to being able to increase the measurement budget such that any noise level can be compensated.

### 3.10.1 Sufficient Condition for $K$-AHS to Succeed Collecting the $k$ Largest Coefficients

Here, we elaborate a sufficient condition which guarantees that $K$-AHS finds the $k$ largest coefficients of the unknown signal $\mathbf{x}$ in analysis basis $\mathbf{\Psi}$, where $k \leq K$.

Recall that due to (3.7), a sensing operation $\langle \mathbf{x}, \varphi_{l,n} \rangle$ computes the sum of a partition of $\mathbf{a}$. Similarly, any sensing operation $\left\langle \mathbf{x}, \varphi_{l,n'} \right\rangle$ at any other node $(l, n')$ of the same level calculates a sum of another disjoint partition of $\mathbf{a}$. For any tree level $l$, the size of such a partition (number of summands) is $2^l$. Merely the $K$ nodes with the largest measurements (the largest sums) are further processed. Consequently, the magnitude of measurements, which include large coefficients, should not become too small. In particular, large coefficients should not cancel each other within a sum.

Suppose $\mathcal{K} = \{a_{h_1}, \dots, a_{h_k}\}$ is the set of the $k$ largest coefficients we want to collect. We call them significant coefficients, in the following. We define $u$ as the smallest absolute value that can possibly occur by summing up any non-empty subset of these significant coefficients:

$$u = \min_{\mathcal{A} \in 2^{\mathcal{K}},\, |\mathcal{A}|>0} \left| \sum_{a_n \in \mathcal{A}} a_n \right| . \tag{3.25}$$

The following theorem states a sufficient optimality condition for $K$-AHS in terms of collecting all $k$ significant coefficients.

**Theorem 4.** *Let $k \leq K$, $L$ the initial tree level, $\Pi = 2^L$ the initial partition size (the number coefficients summed up by a measurement in level $L$), and*

$$r = \sum_{n=k+1}^{2\Pi-1} |a_{h_n}| . \tag{3.26}$$

*$K$-AHS will collect all significant coefficients $a_n \in \mathcal{K}$, if*

$$u > r . \tag{3.27}$$

*Proof.* (*reductio ad absurdum*) If not all significant coefficients are found by $K$-AHS, then there is at least one measurement containing significant coefficients, which is smaller or equal than a measurement containing only non-significant coefficients. Let $\mathcal{A}$ be the coefficient set of such a measurement containing significant coefficients ($\mathcal{A} \cap \mathcal{K} \neq \emptyset$), and $\mathcal{B}$ be the set of coefficients of the measurement containing only non-significant coefficients ($\mathcal{B} \cap \mathcal{K} = \emptyset$). Then the following inequality

$$\left| \sum_{a_n \in \mathcal{A}} a_n \right| \leq \left| \sum_{a_n \in \mathcal{B}} a_n \right| \tag{3.28}$$

holds. This can be written as

$$\left| \sum_{a_n \in \mathcal{A} \cap \mathcal{K}} a_n + \sum_{a_n \in \mathcal{A} \setminus (\mathcal{A} \cap \mathcal{K})} a_n \right| \leq \left| \sum_{a_n \in \mathcal{B}} a_n \right| , \tag{3.29}$$

from which follows

$$\begin{align} \left| \sum_{a_n \in \mathcal{A} \cap \mathcal{K}} a_n \right| &\leq \left| \sum_{a_n \in \mathcal{B}} a_n \right| + \left| \sum_{a_n \in \mathcal{A} \setminus (\mathcal{A} \cap \mathcal{K})} a_n \right| \tag{3.30} \\ &\leq \sum_{a_n \in \mathcal{B}} |a_n| + \sum_{a_n \in \mathcal{A} \setminus (\mathcal{A} \cap \mathcal{K})} |a_n| \tag{3.31} \\ &\leq r \, . \tag{3.32} \end{align}$$

Since $u$ is smaller or equal than the left hand side, this contradicts (3.27). □

With Theorem 4 we can analyze the $K$-AHS sensing quality for the following signal models.

### 3.10.2 Signal Models

The following signal models characterize the decay of signal coefficients. Let $h_1, ..., h_N$ be a sequence of indices which sorts the entries of $\mathbf{a}$ in descending order of their magnitudes, i.e., $|a_{h_1}| \geq |a_{h_2}| \geq ... \geq |a_{h_N}|$. Each signal model assumes certain properties regarding $|a_{h_n}|$, $n = 1, \ldots, N$.

#### $k$-Sparse Model

A $k$-sparse signal, denoted by $\|\mathbf{x}\|_0 = k$, has the property

$$|a_{h_n}| \begin{cases} > 0, & \text{if } n \leq k \\ = 0, & \text{otherwise} \, . \end{cases} \tag{3.33}$$

Commonly, the number of non-zero coefficients is very small compared to the signal dimensionality, i.e. $k \ll N$. We furthermore assume that the $k$ non-zero coefficients come from a continuous probability distribution, e.g. $a_{h_n} \sim \mathcal{N}(0,1)$, $n = 1, \ldots, k$.

#### Exponential Model

The decay of the coefficient magnitudes can be modeled by an exponential law

$$|a_{h_n}| = Rq^{-n+1}\,, \tag{3.34}$$

where base $q > 1$ is the model parameter and $R > 0$ is a scaling constant.

#### Power Law Model

Similar to [Candes and Tao, 2006], the decay of the coefficient magnitudes can be modeled by a power law

$$|a_{h_n}| = Rn^{-\alpha}\,, \tag{3.35}$$

where exponent $\alpha > 1$ is the model parameter and $R > 0$ is a scaling constant. It has been shown that many natural signal classes are consistent with this model [Candes and Tao, 2006, DeVore, 1998, Donoho et al., 1998, Mallat, 2008].

### 3.10.3 Sufficient Optimality Condition for $K$-AHS Depending on the Parameter of the Signal Models

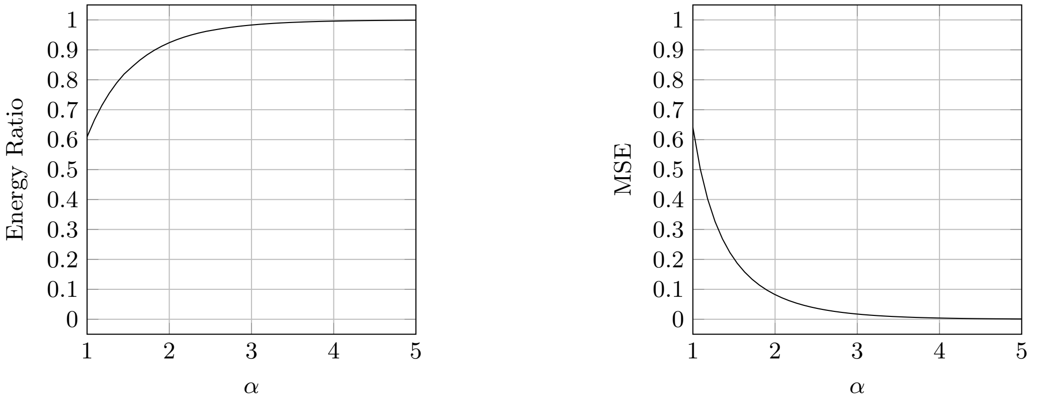
#### Application of Theorem 4 to the $k$-Sparse Model

For signals obeying the $k$-sparse model, where the $k$ non-zero coefficients are drawn from a continuous probability distribution (see Eq. (3.33)), condition (3.27) of Theorem 4 holds almost surely for any $k \leq K$, since $r = 0$ and $u > 0$ with overwhelming probability.

#### Application of Theorem 4 to the Exponential Model

For signals obeying the exponential model (see Eq. (3.34)), condition (3.27) of Theorem 4 holds for any $k \leq K$, if model parameter $q \geq 2$. For the left hand side of (3.27), we have $u \geq Rq^{-k}$. For the right hand side of (3.27), we have

$$\begin{aligned} r = \sum_{n=k+1}^{2\Pi-1} |a_{h_n}| \quad &< \quad \sum_{n=k+1}^{\infty} |a_{h_n}| = Rq^{-k}\frac{1}{q-1} \\ &< \quad Rq^{-k} \leq u\,. \end{aligned} \tag{3.36}$$



(a) Energy ratio between optimal 1-term approximation of $\mathbf{x}$ and the full signal $\mathbf{x}$ depending on $\alpha$.

(b) Mean squared error (MSE) between optimal 1-term approximation of $\mathbf{x}$ and the full signal $\mathbf{x}$ depending on $\alpha$.

Figure 3.6: Relevance of the most significant coefficient $a_{h_1}$ for signals obeying the power law model (see Section 3.10.2).

### Application of Theorem 4 to the Power Law Model

For signals obeying the power law model (see Eq. (3.35)), Theorem 4 cannot be applied unrestrictedly for arbitrary $k \leq K$. Nevertheless, it allows to state conditions on model parameter $\alpha$ for the case $k = 1$, meaning that the detection of $a_{h_1}$, the most significant coefficient, is guaranteed. This can be useful since the bulk of the signal energy lies in this coefficient. Figure 3.6a illustrates, for signals of the power law model, the energy ratio between optimal 1-term approximation of the signal and the complete signal as a function of model parameter $\alpha$. An increase of $\alpha$ considerably concentrates the signal energy on $a_{h_1}$ such that this coefficient contributes nearly exclusively to the entire energy of the signal. A similar illustration is provided by Figure 3.6b in terms of the mean squared error (MSE).

**Independence of Initial Partition Size** For $k = 1$, condition (3.27) of Theorem 4 holds if $\alpha > \alpha^*$, with $\alpha^*$ being defined by

$$\sum_{n=2}^{\infty} n^{-\alpha^*} = \zeta(\alpha^*) - 1 = 1\,, \tag{3.37}$$

where $\zeta(\cdot)$ denotes the Riemann zeta function. The value of $\alpha^*$ is about 1.73. Since $k = 1$, we have $u = R$ and furthermore $r < R$, due to (3.37). Hence, if $\alpha > \alpha^*$, we can guarantee, according to Figure 3.6a, that $K$-AHS captures more than 88% of the signal energy. Note that this finding does not depend on the initial partition size $\Pi$ and the initial sensing tree level $L$.

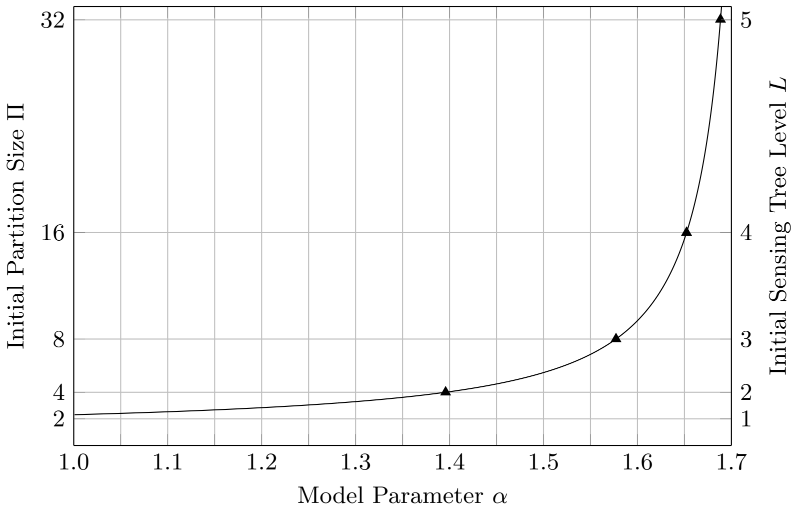


Figure 3.7: Sufficient upper bound (3.38) of the initial partition size Π and the initial sensing tree level $L$ as a function of model parameter $\alpha$ for detecting the most prominent signal coefficient $a_{h_1}$ for signals of the power law model. The triangular markers indicate values of $\alpha$ for which the integer valued partition size Π has to be modified.

**Dependence on Initial Partition Size** By taking the initial partition size Π into account, the detection of $a_{h_1}$ can be guaranteed for even smaller values of $\alpha$. By increasing user parameter $K$, the initial sensing tree level $L$ is decreased by which the initial partitions are getting smaller. This allows for a small improvement. (Note that, on the other hand, the number of measurements is increased.) By using integral approximations of the partial sum (3.26), we obtain

$$\begin{aligned} r &= \sum_{n=2}^{2\Pi-1} n^{-\alpha} \\ &\leq 2^{-\alpha} + \int_{\frac{5}{2}}^{2\Pi-\frac{1}{2}} x^{-\alpha}\mathrm{d}x \\ &\leq 2^{-\alpha} + \frac{1}{1-\alpha}\left(\left(2\Pi - \frac{1}{2}\right)^{1-\alpha} - \left(\frac{5}{2}\right)^{1-\alpha}\right) \end{aligned} \tag{3.38}$$

If we choose $L$ such that we start with a value Π for which the right hand side of (3.38) is smaller than 1, then the most significant coefficient $a_{h_1}$ is definitely captured by $K$-AHS. Figure 3.7 plots the sufficient upper bound of Π, according to (3.38), as a function of $\alpha$.

### 3.10.4 Probabilistic Bounds for $K$-AHS to Fail Collecting the $k$ Largest Coefficients

In the following, we are interested when (3.27) holds with high probability. As we know from our proof of Theorem 4, significant coefficients can get lost, if there are non-empty subsets $\mathcal{C} \subseteq \mathcal{K}$ such that

$$\left| \sum_{a_n \in \mathcal{C}} a_n \right| \leq r \,, \tag{3.39}$$

where $r$ is given as in (3.26). Let

$$s = \min_{\mathcal{C} \in 2^{\mathcal{K}}, |\mathcal{C}|>0} |\mathcal{C}| \text{ s.t. } \left| \sum_{a_n \in \mathcal{C}} a_n \right| \leq r \tag{3.40}$$

be the minimal number of significant coefficients that is necessary to occur within a measurement such that (3.39) is satisfied.

In other words, whenever $\leq s-1$ significant coefficients are contained in each initial partition, $K$-AHS will succeed in terms of collecting all significant coefficients $\mathcal{K}$. What is the probability for such a situation? The answer follows from the following combinatorial problem: in how many ways can $k$ distinct objects (significant coefficients) be distributed into $B$ distinct bins (initial partitions) such that at most $s-1$ objects fall into each bin, where $1 < s \leq k < B$. This number of restricted distributions relative to $B^k$, the number of all unrestricted distributions, is the desired probability, which we denote in the following by $p_{k,B,s-1}$. However, $p_{k,B,s-1}$ cannot be written in closed form except by a formula which involves a generating function, see e.g. [Flajolet and Sedgewick, 2009, Heubach and Mansour, 2009]:

$$p_{k,B,s-1} = \frac{1}{B^k} \sum_{\substack{\lambda_1+\lambda_2+\cdots+\lambda_B=k \\ 0\leq\lambda_i\leq s-1}} \frac{k!}{\lambda_1!\lambda_2!\cdots\lambda_B!} \,. \tag{3.41}$$

Calculating $p_{k,B,s-1}$ requires to iterate over all weak integer compositions of the number $k$ into $B$ parts with restricted part size $0 \leq \lambda_i \leq s-1$, e.g. by a procedure proposed in [Page, 2013]. Each of these compositions $(\lambda_1, ..., \lambda_B)$ corresponds to a good-natured distribution of significant coefficients into the $B$ measurements and thereby to a safe situation for $K$-AHS. Each summand of the sum can be grasped as a multinomial coefficient that counts all permutations of $\mathcal{K}$ satisfying the absolute frequencies given by $\lambda_1, ..., \lambda_B$, where reordering within each measurement is compensated by the factorials in the denominator.

Let $\overline{p}_{k,B,s-1} = 1 - p_{k,B,s-1}$ be the complementary probability, i.e. the probability that in at least one partition of the initial sensing tree level $> s-1$ significant coefficients are contained. Note that $\overline{p}_{k,B,s-1}$ is an upper bound of the probability that $K$-AHS

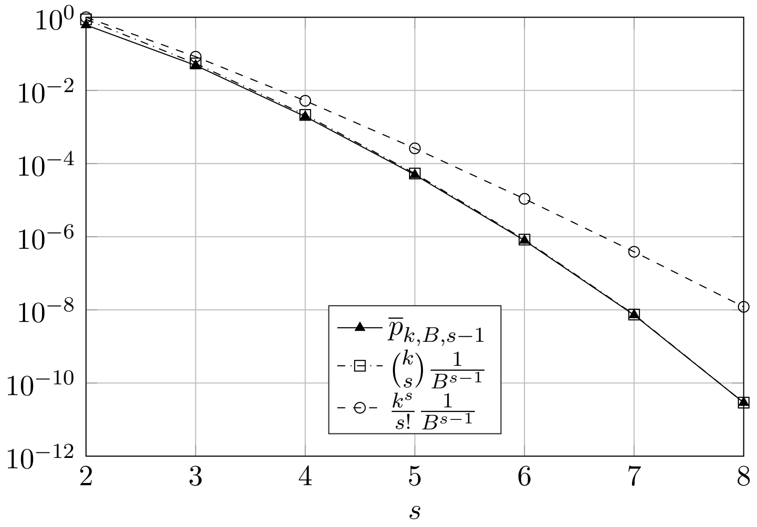


Figure 3.8: Upper bounds of the probability that $K$-AHS fails to detect the whole set of significant coefficients $\mathcal{K} = \{a_{h_1}, ..., a_{h_k}\}$ as predicted by Theorem 5. The bounds are plotted as a function of $s$, the size of the smallest subset of significant coefficients satisfying (3.39), and are illustrated for $k = 8$ and $B = 32$ (meaning $K$ is chosen such that $L = 5$).

fails in terms of missing significant coefficients from $\mathcal{K}$.

By the following theorem we provide two upper bounds of $\overline{p}_{k,B,s-1}$ and thus of the $K$-AHS fail probability. The bounds have a closed form and it can be seen that they rapidly decrease as $s$ increases.

**Theorem 5.** *Let $B = N2^{-L}$ be the number of $K$-AHS measurements of the initial sensing tree level $L$ and $\mathcal{K} = \{a_{h_1}, ..., a_{h_k}\}$ the set of significant signal coefficients. Given that the locations of the signal coefficients are uniformly distributed, then the probability that $K$-AHS fails (in terms of not capturing all coefficients $\mathcal{K}$) is bounded as follows*

$$p_{\text{fail}} \leq \binom{k}{s} \frac{1}{B^{s-1}} \leq \frac{k^s}{s!} \frac{1}{B^{s-1}} , \tag{3.42}$$

*where $s$ is given by (3.40) and $1 < s \leq k < B$.*

*Proof.* There are $k = |\mathcal{K}|$ significant coefficients. Each appears in one of the $B$ initial measurements. The $K$-AHS fail probability $p_{\text{fail}}$ is bounded from above by

$$p_{\text{fail}} \leq \overline{p}_{k,B,s-1} = 1 - p_{k,B,s-1} ,$$

where $p_{k,B,s-1}$ is defined in (3.41).

$\overline{p}_{k,B,s-1}$ is bounded as follows

$$\overline{p}_{k,B,s-1} \leq \frac{1}{B^k} B \binom{k}{s} B^{k-s} , \tag{3.43}$$

where $B\binom{k}{s}B^{k-s}$ is the absolute number of ways to select one of the $B$ partitions, select $s$ of the $k$ significant coefficients, insert these $s$ selected coefficients into that selected partition and distribute the remaining $k-s$ significant coefficients arbitrarily (without restrictions) into the $B$ partitions. This is an upper bound since (3.43) counts a small number of constellations more than once. The normalizing factor $1/B^k$ represents the number of ways to arbitrarily distribute the $k$ significant coefficients (without restrictions) into the $B$ partitions.

The proof is complete with

$$\binom{k}{s} = \frac{k!}{s!\,(k-s)!} = \frac{k(k-1)\cdots(k-s+1)}{s!} \leq \frac{k^s}{s!}\,.$$

□

Note that by $\overline{p}_{k,B,s-1}$ the concrete "unsuccessful" subsets $\mathcal{C}$, which satisfy (3.39), are not taken into account, only the cardinality of the smallest one. From this point of view the bound is pessimistic and the true probability is even smaller.

Figure 3.8 illustrates that the bound of the $K$-AHS fail probability has a strong decay even in moderate scenarios.

## 3.11 $K$-AHS Results for Synthetic Signals

We simulated sensing using $K$-AHS for synthetic signals of the models introduced in Section 3.10.2. To complement our theoretical findings of Section 3.10, we empirically assess the performance of $K$-AHS to detect significant coefficients depending on the model parameters. For each tested parameter value we generated $10^5$ signals of dimensionality $N = 1024$. First, the magnitudes of coefficients were computed as given by the model. Second, locations and signs of the coefficients were assigned uniformly at random. Subsequently, we applied $K$-AHS to each signal by setting the user parameter to $K = 4$, and calculated the empirical detection probability for the principal coefficient ranks, i.e. $a_{h_1}, \ldots, a_{h16}$. Ideally, the empirical probability for each coefficient $a_{h_1}, ..., a_{h_K}$ is equal or close to 1. Figure 3.9 to 3.11 show, subject to the three different signal models, the empirical detection probability of the 16 most significant coefficients.

Figure 3.9 illustrates $K$-AHS simulation results for the $k$-sparse signal model (see Eq. (3.33)). While the number of non-zero coefficients is given by model parameter $k$, their values were drawn from a standard Gaussian distribution. The values of the model parameter that we investigated were $k \in \{2, 4, 8\}$. In the cases $k = 2$ and $k = 4$ all non-zero coefficients were identified correctly. This is in accordance with our theoretical finding in 3.10.3 which predicts perfect recovery, almost surely, if $K \geq k$. In the case $k = 8$, we have the situation $K < k$ and the empirical detection probability is decreased. However, it is still above 0.8 for each $a_{h_1}, ..., a_{h_K}$ despite the fact that the number of non-zero coefficients of the signal is considerably underestimated.

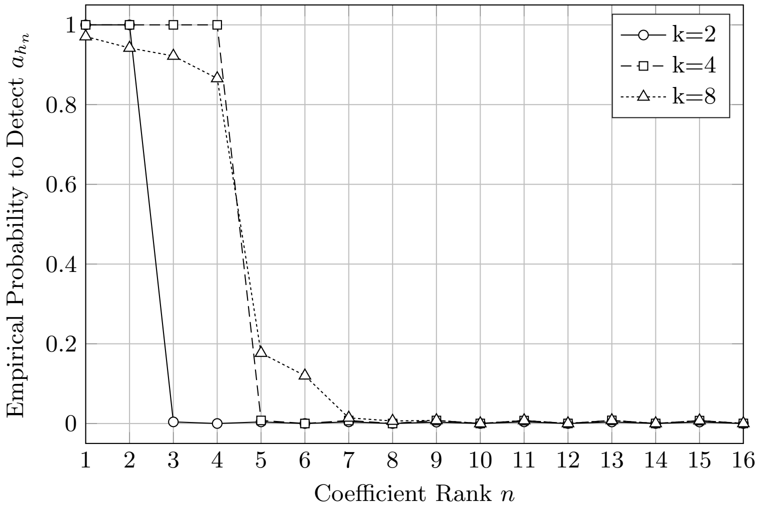


Figure 3.9: Empirical probability to detect significant coefficients of signals obeying the $k$-sparse model when sensed by $K$-AHS. For model parameter $k \in \{2, 4, 8\}$, $10^5$ signals of dimensionality $N = 1024$ were generated. The detection probability of the 16 most significant coefficients is plotted as a function of their rank. $K$-AHS was applied with user parameter $K = 4$. As long as $K \geq k$, all $k$ non-zero coefficients are identified correctly.

Figure 3.10 illustrates $K$-AHS simulation results for the exponential model (see Eq. (3.34)). The values of the model parameter that we investigated were $q \in \{1.2, 1.6, 2\}$. It can be seen that an increase of base $q$ (steeper decay of coefficients) leads to an

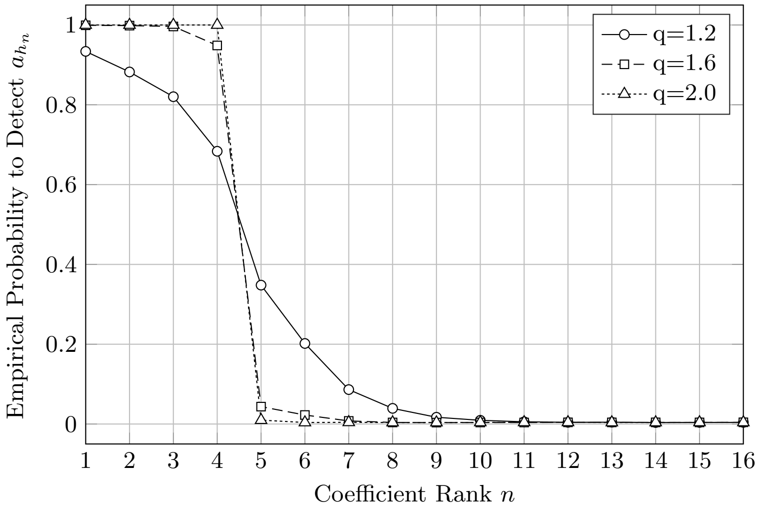


Figure 3.10: Empirical probability to detect significant coefficients of signals obeying the exponential model when sensed by $K$-AHS. For model parameter $q \in \{1.2, 1.6, 2\}$, $10^5$ signals of dimensionality $N = 1024$ were generated. The detection probability of the 16 most significant coefficients is plotted as a function of their rank. $K$-AHS was applied with user parameter $K = 4$. As soon as $q \geq 2$, the $K$ most significant coefficients are identified correctly.

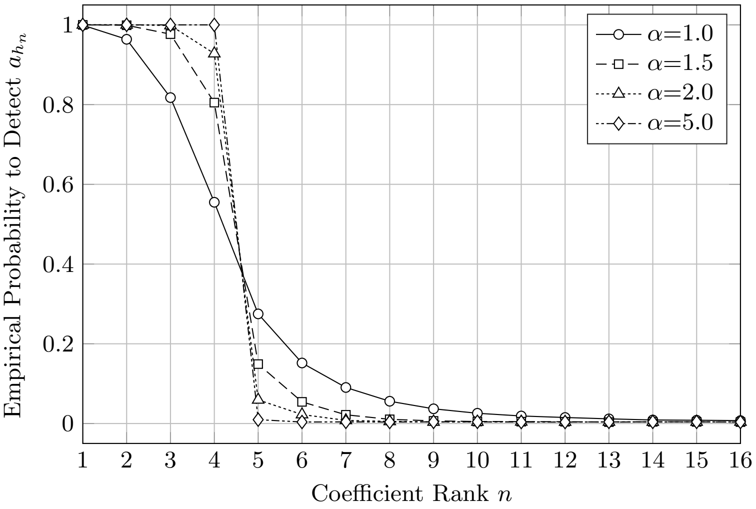


Figure 3.11: Empirical probability to detect significant coefficients of signals obeying the power law model when sensed by $K$-AHS. For model parameter $\alpha \in \{1, 1.5, 2, 5\}$, $10^5$ signals of dimensionality $N = 1024$ were generated. The detection probability of the 16 most significant coefficients is plotted as a function of their rank. $K$-AHS was applied with user parameter $K = 4$. The most significant coefficient $a_{h_1}$ is almost always detected, even if $\alpha < \alpha^*$.

increase of the empirical detection probability for $a_{h_1}, ..., a_{h_K}$. All of the $K$ most prominent coefficients are identified correctly in the scenario $q = 2$, which is predicted by our theoretical finding in 3.10.3.

Figure 3.11 illustrates $K$-AHS simulation results for the power law model (see Eq. (3.35)). As for the exponential model, a larger parameter value $\alpha$ results in a steeper decay of coefficients and increases the detection probability for significant coefficients. As opposed to the exponential model, a single threshold of model parameter $\alpha$ does not guarantee the detection of the most prominent coefficients for all values of $K$. On the other hand, the signal energy rapidly focuses on $a_{h_1}$ as $\alpha$ increases, see Figure 3.6a. Therefore, we additionally illustrate for the power decay model the relative signal energy obtained by $K$-AHS dependent on $K$. Figure 3.12 shows that, for various values of $\alpha$, the reconstruction performance in terms of captured signal energy increases as $K$ is set to higher values.

## 3.12 AHS Results for Natural Images

We made experiments simulating compressive imaging with $\tau$-AHS, $K$-AHS and $\ell_1$-based CS on standard test images (*Cameraman*, *Lena*, *Pirate*) with a size of $512 \times 512$ pixels and a gray level depth of 8 bit (see Figure 3.13). For the individual test images, we assessed the reconstruction performance by the peak-signal-to-noise ratio (PSNR) as a function of the relative number of measurements $M/N$.

We selected the user parameters such that $M$, the number of measurements, takes

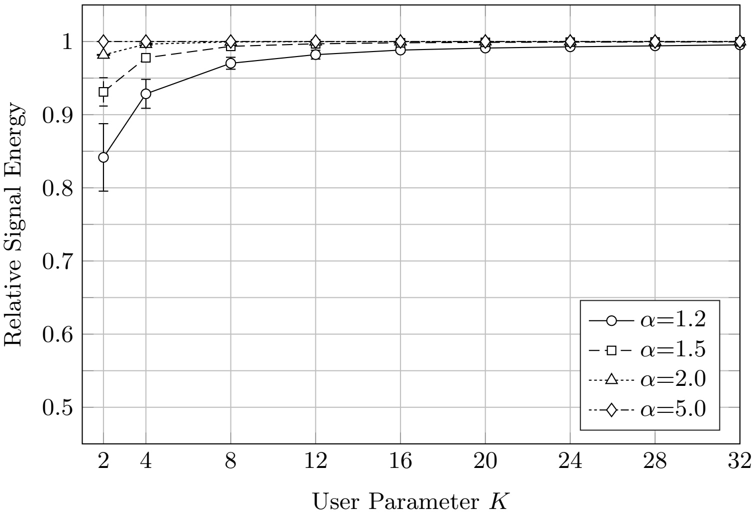


Figure 3.12: Relative signal energy captured from signals obeying the power law model when sensed by $K$-AHS. For model parameter $\alpha \in \{1, 1.5, 2, 5\}$, $10^5$ signals of dimensionality $N = 1024$ were generated. The signal energy is plotted as a function of user parameter $K$.

(a) Test image *Cameraman*.

(b) Test image *Lena*.

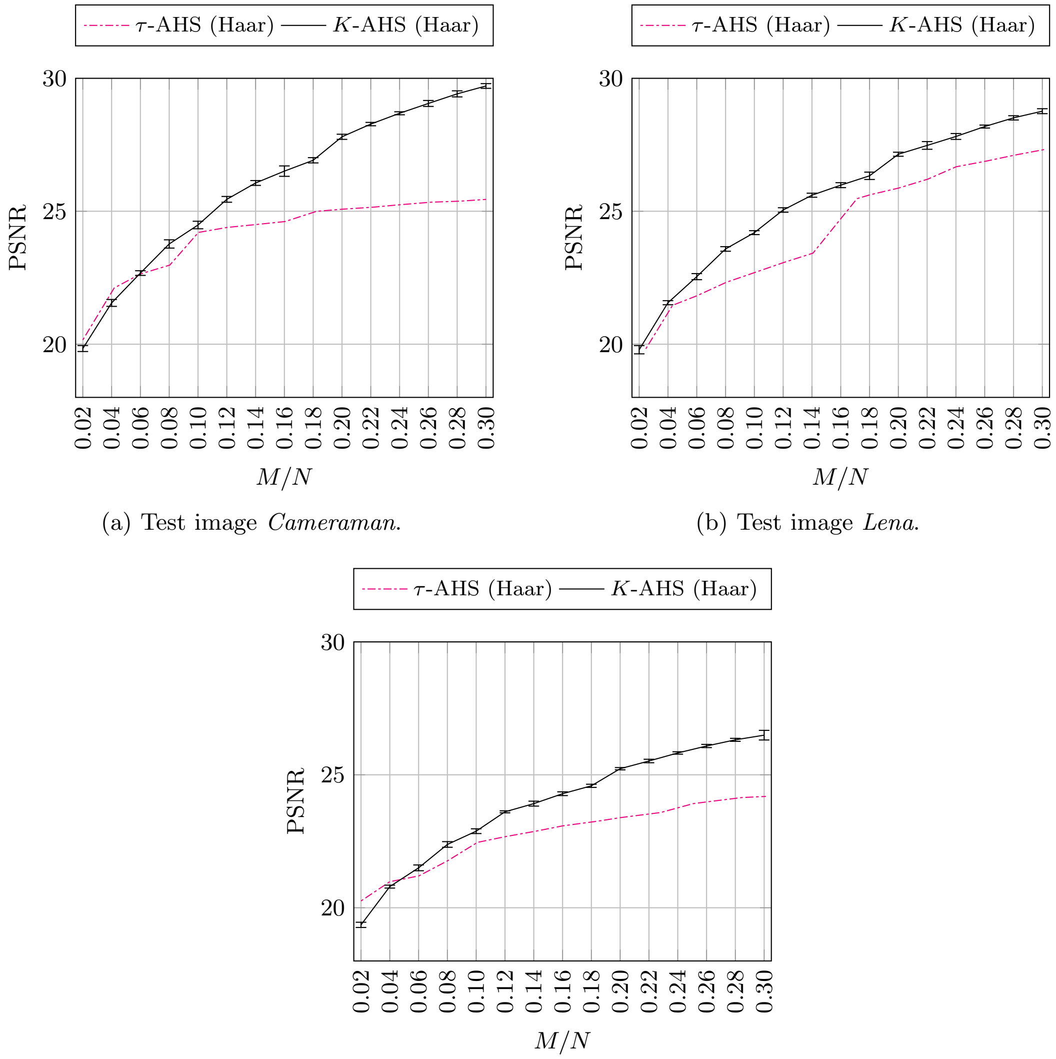
(c) Test image *Pirate*.

Figure 3.13: Original test images used for the compressive imaging experiments (size $512 \times 512$, 8 bit gray level depth).

values $0.02N, 0.04N, \ldots, 0.3N$. For CS, we deployed sensing matrices with the corresponding number of rows. For $\tau$-AHS, we gradually changed the canonical threshold $\tau_0$. For $K$-AHS, we rearranged Eq. (3.18).

### 3.12.1 Comparison $\tau$-AHS and $K$-AHS

Firstly, Figure 3.14 illustrates the difference between $\tau$-AHS and $K$-AHS in terms of reconstruction performance as measured by the peak-signal-to-noise ratio (PSNR) for the three test images *Cameraman*, *Lena* and *Pirate*. The non-standard 2D Haar wavelet basis was selected as analysis basis $\mathbf{\Psi}$. For all test images and any $M > 0.04N$, $K$-AHS yields a higher PSNR than $\tau$-AHS. We observed that the $\tau$-AHS performance is rather sensitive to the order of the $\psi_i$ and report the sensing performance results subject to the canonical order of the Haar basis (cf. Figure 2.3 for $N = 256$, column-major order) as it yields superior results compared to a random permutation. A shortcoming of $\tau$-AHS is that a fix canonical threshold $\tau_0$ can lead to different $M$, given two different permutations. The reason is that $\tau$-AHS decisions for descents and omissions of subtrees are based on absolute measurements as opposed to $K$-AHS for which such decisions are based on relative comparisons of the measurements in a level. $K$-AHS is more stable. A fix combination of the parameters $K$ and $L$ yields always the same $M$. Furthermore, $K$-AHS is robust in terms of reconstruction performance for different random permutations. In Figure 3.14, we report the mean and the standard deviation (indicated by the error bars) of the PSNR over 10 trials with different random permutations of the



(a) Test image *Cameraman*.

(b) Test image *Lena*.

(c) Test image *Pirate*.

Figure 3.14: Image sensing performance comparison between $\tau$-AHS and $K$-AHS for three test images. The PSNR is plotted as a function of the relative number of measurements.

$\psi_i$.

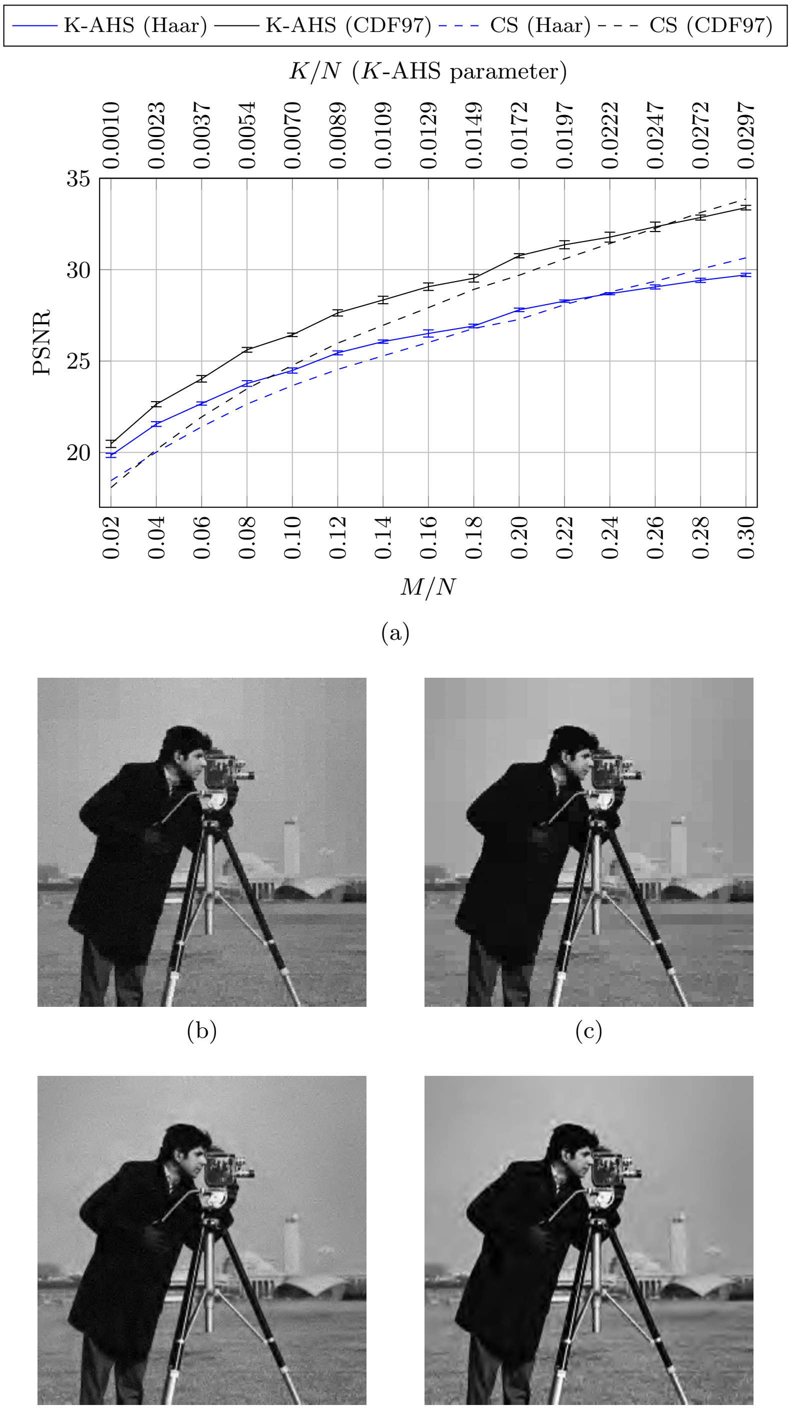
### 3.12.2 Comparison $K$-AHS and CS

We compare furthermore $K$-AHS and CS. We investigate two different sparse transforms $\mathbf{\Psi}$: ($i$) the orthogonal non-standard 2D Haar wavelet basis and ($ii$) the biorthogonal Cohen-Daubechies-Feauveau 9/7 (CDF97) wavelet basis, which is part of the JPEG 2000 standard [Taubman and Marcellin, 2013].

The CS results were obtained by an $\ell_1$-norm minimization approach, i.e. by solving $\text{P}_{(3.1)}$ for $p = 1$ subject to the same sparse coding transforms and the same values of $M$. The CS measurements of the images were collected by sensing vectors that were randomly drawn (without replacement) from the real valued noiselet transform. The random noiselet measurement ensemble was chosen in favor of CS due to its low coherence to the Haar basis [Tuma and Hurley, 2009] and to the CDF97 basis [Pereira et al., 2014]. A low coherence between measurement ensemble $\mathbf{\Phi}$ and sparse transform $\mathbf{\Psi}$ assures that $\ell_1$-norm minimization recovers the original signal accurately [Candes and Romberg, 2007]. In our classical CS experiments we addressed the following optimization problem In order to solve $\text{P}_{(3.1)}$, we used the NESTA [Becker et al., 2011] Matlab package, an $\ell_1$-recovery toolbox suited for solving large-scale compressed sensing reconstruction problems. NESTA is a cutting-edge first-order optimization procedure that exploits ideas from Nesterov [Nesterov, 2005] such as accelerated descent methods and smoothing techniques.

Figure 3.15, 3.16 and 3.17 illustrate $K$-AHS and CS results for the test images *Cameraman*, *Lena*, and *Pirate*.

For each image, Figure 3.15(a), 3.16(a), and 3.17(a) illustrate the rate distortion analysis showing reconstruction performance as measured by the peak signal-to-noise ratio (PSNR) as a function of the (relative) number of collected measurements. Each curve corresponds to one of the four compressive imaging variants described above. Both approaches, $K$-AHS and CS, achieve consistently higher PSNR with the CDF97 wavelet basis than with the Haar wavelet basis. This can be explained by the fact that natural images have generally sparser representations by smooth CDF97 basis functions than by ternary, discontinuous Haar basis functions. For measurements up to 25% of the number of dimensions $N$ (usually $M \ll N$), the PSNR of $K$-AHS reconstructions is higher than the PSNR of CS reconstructions for both Haar and CDF97 wavelets. That difference is larger with the CDF97 basis than with the Haar basis. The reason might be that noiselets and Haar wavelets have minimal mutual coherence [Tuma and Hurley, 2009] as opposed to the combination of noiselets and CDF97 wavelets for which the mutual coherence is small but not minimal [Pereira et al., 2014]. Therefore, it is more difficult for $K$-AHS to achieve higher reconstruction accuracy than CS. The larger the number of collected measurements, the smaller the PSNR difference between $K$-AHS and CS. For really large numbers of measurements, where $M \not\ll N$, CS reconstructions



(a)

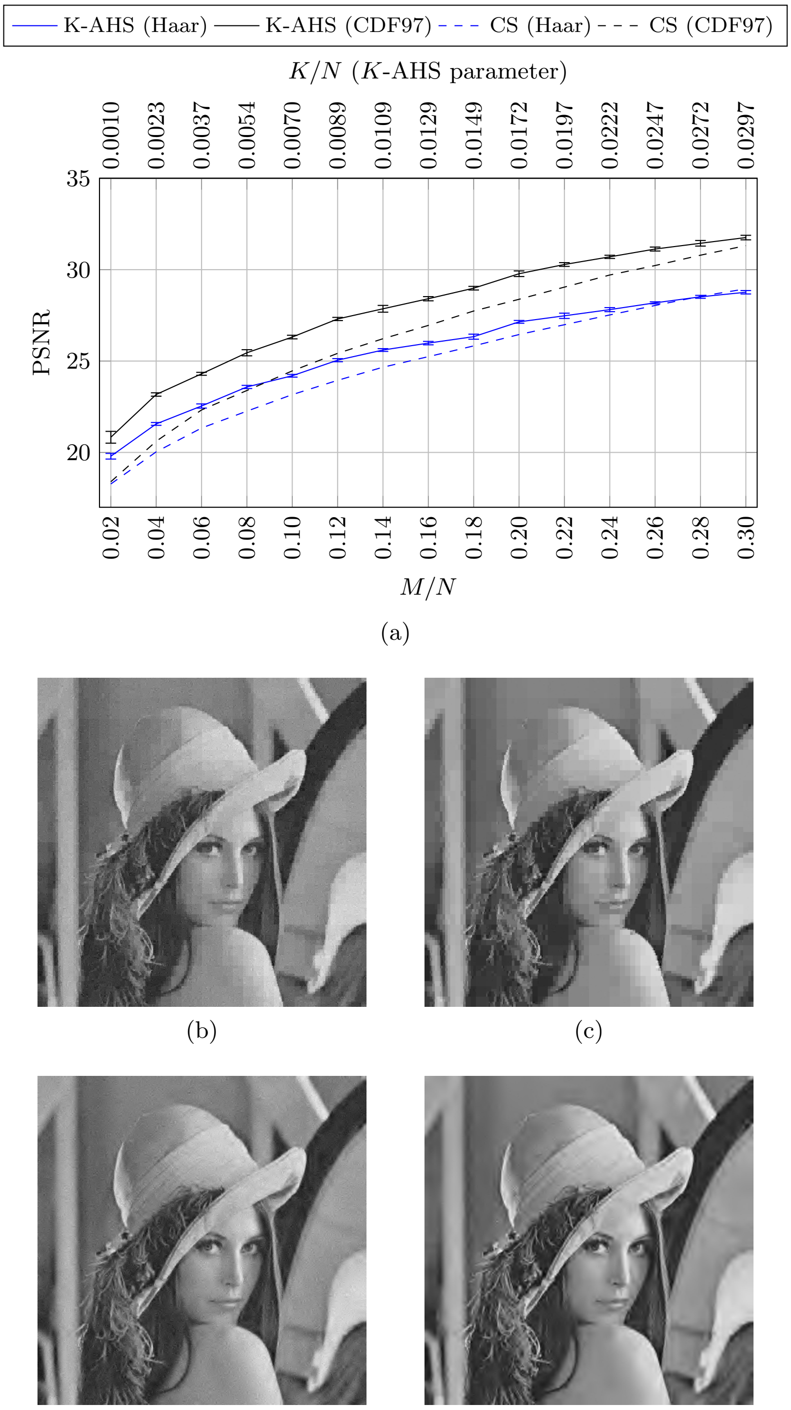
(b)

(c)

(d)

(e)

Figure 3.15: Image sensing performance comparison between $K$-AHS and $\ell_1$-based CS for test image *Cameraman* ($N = 2^{18}$). (a) The PSNR dependent on the relative number of measurements. (b) CS reconstruction from $M = 0.2N$ random noiselet measurements, Haar basis, PSNR: 27.27. (c) $K$-AHS reconstruction from $M = 0.2N$ adaptive measurements, Haar basis, PSNR: 27.86. (d) CS reconstruction from $M = 0.2N$ random noiselet measurements, CDF97 basis, PSNR: 29.69. (e) $K$-AHS reconstruction from $M = 0.2N$ adaptive measurements, CDF97 basis, PSNR: 30.85. For visualization, reconstructed images were clipped to [0, 255].



(a)

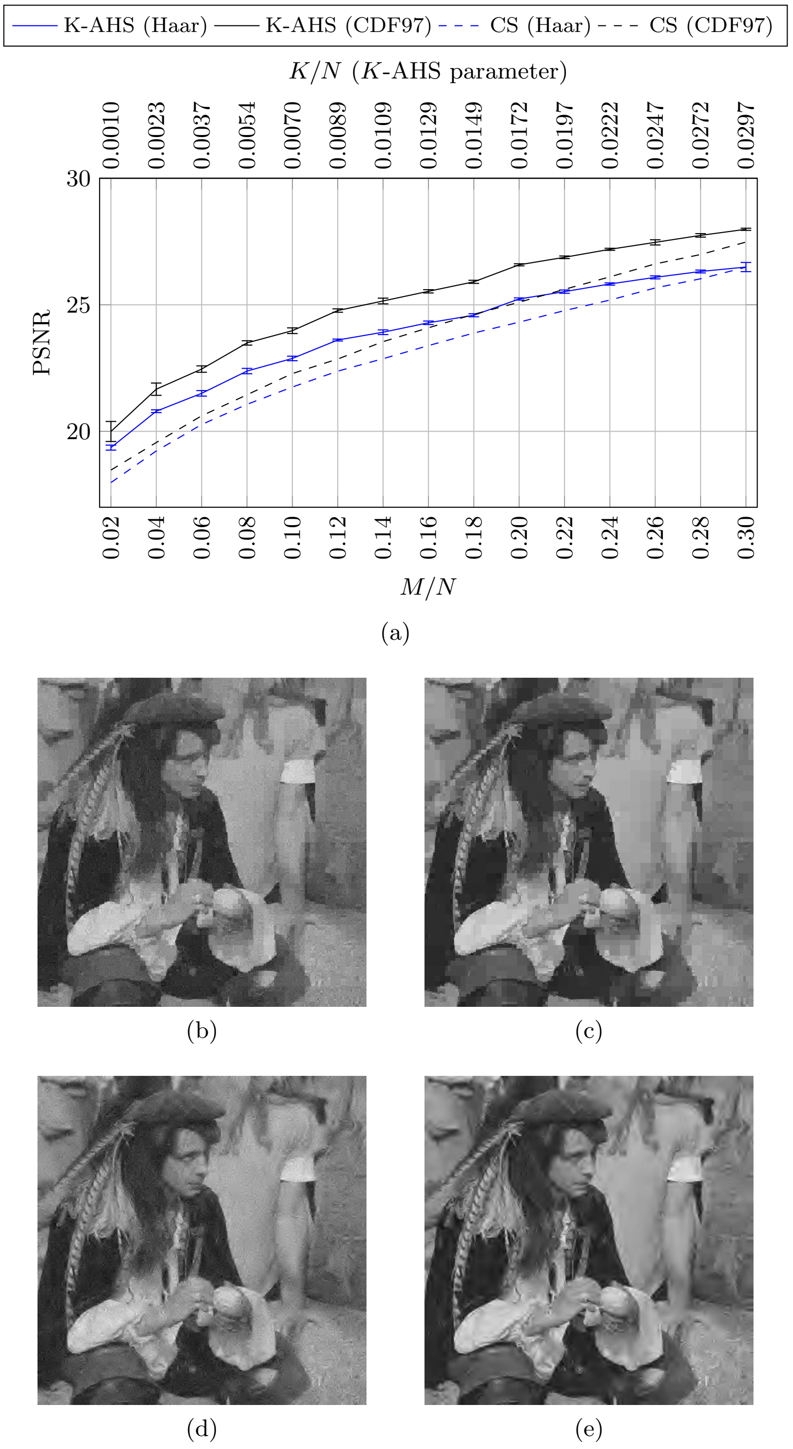
(b)

(c)

(d)

(e)

Figure 3.16: Image sensing performance comparison between $K$-AHS and $\ell_1$-based CS for test image *Lena* ($N = 2^{18}$). (a) The PSNR dependent on the relative number of measurements. (b) CS reconstruction from $M = 0.2N$ random noiselet measurements, Haar basis, PSNR: 26.45. (c) $K$-AHS reconstruction from $M = 0.2N$ adaptive measurements, Haar basis, PSNR: 27.15. (d) CS reconstruction from $M = 0.2N$ random noiselet measurements, CDF97 basis, PSNR: 28.38. (e) $K$-AHS reconstruction from $M = 0.2N$ adaptive measurements, CDF97 basis, PSNR: 29.71. For visualization, reconstructed images were clipped to [0, 255].



(a)

(b)

(c)

(d)

(e)

Figure 3.17: Image sensing performance comparison between $K$-AHS and $\ell_1$-based CS for test image *Pirate* ($N = 2^{18}$). (a) The PSNR dependent on the relative number of measurements. (b) CS reconstruction from $M = 0.2N$ random noiselet measurements, Haar basis, PSNR: 24.31. (c) $K$-AHS reconstruction from $M = 0.2N$ adaptive measurements, Haar basis, PSNR: 25.26. (d) CS reconstruction from $M = 0.2N$ random noiselet measurements, CDF97 basis, PSNR: 25.10. (e) $K$-AHS reconstruction from $M = 0.2N$ adaptive measurements, CDF97 basis, PSNR: 26.57. For visualization, reconstructed images were clipped to [0, 255].

have higher PSNR than $K$-AHS reconstructions.

For each image, Figure 3.15(b)-(c), 3.16(b)-(c), and 3.17(b)-(c) illustrate CS and K-AHS reconstructions from $M = 0.2N$ measurements using the Haar wavelet domain. Each reconstructed image shows blocking artifacts, due to the discontinuity of the Haar wavelet basis. While both approaches restore edges and contours satisfactory, CS seems slightly more accurate at image regions containing slight edges. On the other hand, CS reconstructions suffer considerably from high frequency noise which is evenly distributed over the entire image and perhaps causes the inferior PSNR. $K$-AHS shows at some image regions slightly coarser block structures than CS but recovers overall homogeneous image regions more accurately. Furthermore, $K$-AHS does not suffer from high frequency noise.

For each image, Figure 3.15(d)-(e), 3.16(d)-(e), and 3.17(d)-(e) illustrate CS and K-AHS reconstructions from $M = 0.2N$ measurements using the CDF97 wavelet domain. In accordance with the rate distortion analysis, the images reconstructed in the CDF97 wavelet domain look, for both approaches, visually more pleasant than the images reconstructed in the Haar wavelet domain. Some contours of the $K$-AHS reconstructions show minor ringing artifacts whereas image regions with constant luminance and small luminance variation are more accurately recovered compared to CS. Again, images reconstructed by CS suffer from evenly distributed high frequency noise.

The results presented in Section 3.12.2 indicate that $K$-AHS successfully collects significant coefficients of natural images. In order to deepen the intuition to what extent the top $K$ coefficients captured by $K$-AHS deviate from the $K$ largest coefficients, we provide a corresponding comparison for one of the test images. For the image

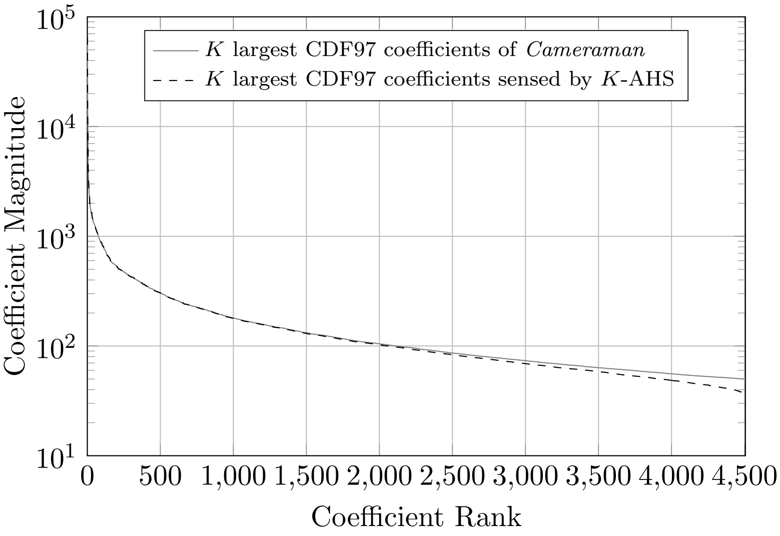


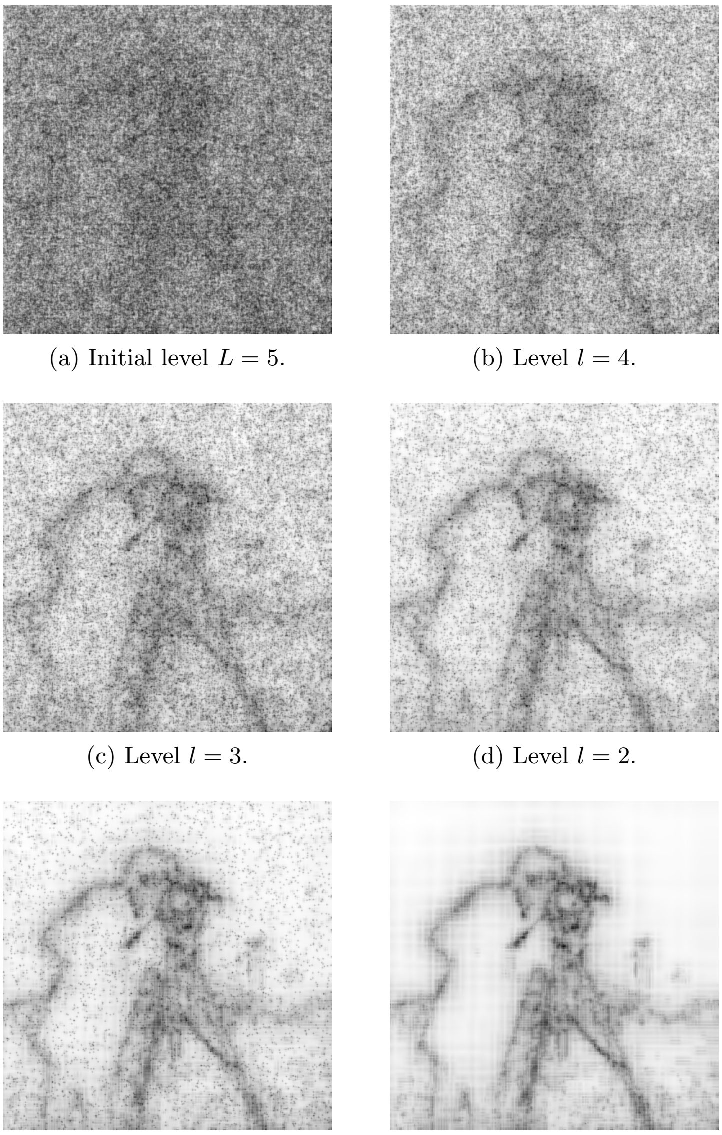
Figure 3.18: Comparison between the $K$ largest CDF97 wavelet coefficients of test image *Cameraman* ($N = 2^{18}$), and the $K$ largest CDF97 wavelet coefficients found by $K$-AHS ($K = 4506 \Rightarrow M \approx 0.2N$).

*Cameraman*, Figure 3.18 illustrates the magnitude of the $K$ largest coefficients in the CDF97 wavelet domain, as well as the $K$ largest coefficients that are sensed by $K$-AHS, where $K = 4506$ $(M = 0.2N)$. The direct comparison shows that $K$-AHS collects a considerable amount of the most significant CDF97 coefficients of the image. The length, $q$, of the largest gapless sequence of successfully identified coefficient magnitudes $|a_{h_1}| \geq \cdots \geq |a_{h_q}|$ varies depending on the order of the wavelet basis vectors in the bottom level of the sensing tree. Over 1000 runs with different random permutations of the basis vectors, the average of the largest gapless sequence length is $\overline{q} = 454.90$ (with a standard deviation of 188.02). Although not all of the $K$ largest coefficients are identified, it is apparent that those coefficients found, have only a comparably small deviation from the optimal ones. For the run illustrated in Figure 3.18, the difference of the magnitudes between the $K$-th largest image coefficient and the $K$-th largest coefficient found by $K$-AHS is less than 12.8.

### Spatial Sensing Maps

The spatial regions at which $K$-AHS senses can be visualized if the analysis basis vectors are localized, like in the case of a wavelet domain. For each sensing tree level that is processed by $K$-AHS, we identified the sensing vectors which provided the $K$ largest measurements, and replaced each entry by its absolute value. Subsequently, we calculated the sum of these $K$ rectified "winner sensing vectors" to obtain a spatial sensing map. This spatial sensing map indicates which image regions are sensed to which extent by the $K$ "winner sensing vectors" of the corresponding level. Since the $K$ winner determine in particular, by which branches of the sensing tree the sensing proceeds, they also determine, which regions shall be refined. Figure 3.19 shows a sequence of spatial sensing maps from the initial level to the bottom level of the sensing tree while the image *Cameraman* is sampled by $K$-AHS setting $K = 2^{12} - 1 = 4095$. The value of $K$ is chosen such that six spatial sensing maps are obtained.

At the initial level $L$, the spatial sensing map is diffuse, i.e. image regions which are sampled by the "winner sensing vectors" of that level are evenly distributed. Note that the image content is barely perceptible from the first and the second sensing map. However, as $K$-AHS descends to lower levels of the tree, the spatial sensing maps reveal more and more image structures. From the spatial map of the bottom level, the image content, in this case the depicted cameraman with the tripod, is well recognizable. Apparently, regions at which $K$-AHS focuses the sensing are successively refined and lead to salient regions of the image such as distinct contours, edges and corners.



(a) Initial level $L = 5$.

(b) Level $l = 4$.

(c) Level $l = 3$.

(d) Level $l = 2$.

(e) Level $l = 1$.

(f) Bottom level $l = 0$.

Figure 3.19: Spatial sensing maps obtained as the test image *Cameraman* ($N = 2^{18}$) is sensed by $K$-AHS in the CDF97 wavelet domain ($K = 2^{12} - 1$). The spatial sensing maps indicate, for each level of the sensing tree, how intensively each region of the image is sensed by the $K$ "winner sensing vectors", i.e. where the sensing load is focused (see Section 3.12.2). Each spatial sensing map is normalized. White regions indicate minimal sensing activity, whereas black regions indicate maximal sensing activity.

# 4 Conclusion and Discussion

In this chapter, we conclude the contributions presented in this thesis. We summarize the main results and discuss properties of the proposed algorithms. Moreover, it is outlined how future research might advance the reported findings and which further experiments could yield new insights or might answer open questions.

## 4.1 Orthogonal Dictionary Learning

We have addressed the problem of learning a complete orthogonal dictionary, i.e. an ONB, from training data to accomplish sparse data encodings based on the constrained $K$-sparse model. We introduced CA, the modification of a base line approach to match with that model, as well as OSC and GF-OSC, two novel online learning methods which yield sparse coding ONBs that outperform static orthogonal transforms and dictionaries from alternative batch learning approaches in terms of higher encoding accuracy for a wide range of representation sparsity.

For the task to recover a reference ONB from synthetic sparse data, GF-OSC demonstrates for the noiseless setting superb recovery performance up to quite low levels of sparsity. However, when noise contaminates the data, GF-OSC fails to recover the ONB. In contrast, OSC recovers the ground truth from noisy samples much more robustly. Admittedly, OSC leaves on the synthetic data sets minor residuals in terms of the mean matched overlap (MMO), the applied recovery error measure. The cause of these small residuals remains yet unexplained. On the other hand, OSC converges remarkably close to the reference ONB in cases in which all the other methods completely fail, i.e. for very low sparsity or when noise is present. This is one facet of stability that OSC entails.

Moreover, a further and very useful stability property of OSC has revealed in the course of our experiments: the true or optimal sparsity level does not need to be known. By setting the user parameter for the target sparsity level to $K = N$, OSC learns a sparse coding ONB which is as good as if $K$ were set to the true or optimal value. According to our experiments, this property holds consistently for all applied data sets including synthetic and real data sets.

On real world image data our proposed orthogonal sparse coding methods OSC and

GF-OSC learn atoms that are highly adapted to the data and resemble a multifarious set of features. A selectivity of the dictionary atoms subject to particular frequencies, orientations and localizations emerges for natural image patches. For images of handwritten digits, atoms emerge which contain prototypical digit combinations or localized “stroke detectors”. OSC consistently acquires the entire repertoire of those features (independent of the chosen user sparsity parameter), whereas other learning methods resemble different subsets of those features depending on the choice of the user sparsity parameter. Although for both types of data, a complete non-orthogonal dictionary achieves slightly sparser encodings for the highest sparsity levels, an ONB learned by OSC (in the case of natural image patches) or GF-OSC (in the case of handwritten digits) achieves the sparsest encodings for the remaining range of sparsity levels, compared to static transforms (2D DCT, non-standard 2D Haar DWT), PCA, K-SVD and the baseline approach CA.

Unfortunately, the improvement of the cost function minimization by OSC and GF-OSC entails an increased computational load. Both online learning methods perform one dictionary update for each presented training data sample, which facilitates to overcome local minima. Such an update requires $\mathcal{O}(N^3)$ operations whereas the equivalent “per sample complexity” of the base line batch learning method CA is $\mathcal{O}(N^2)$. On the other hand, the numbers of learning steps required by OSC and GF-OSC can be considerably reduced if the initial and final learning rate are adequately chosen. Our numerical experiments have been carried out on a computer cluster with heterogeneous architectures and partially active processing load and are based on implementations by different programming languages, which prevents a balanced runtime comparison. Future studies should address a fair comparison of the absolute time consumption for the individual learning methods.

Another practice-oriented aspect of future research could be the parallelization of OSC and GF-OSC. For instance, a GF-OSC learning step involves several matrix multiplications, which should allow considerable accelerations. While such parallel operations are realized by multithreaded implementations, one could consider to achieve further runtime reductions by GPU-based implementations. A parallelization of OSC does not seem to be straight-forward but should be investigated.

A disadvantage of online learning schemes is that a monotonic descent on the cost function or local convergence is commonly difficult to prove because each update blinds out all but one training data sample. Stochastic gradient descent, as implemented by GF-OSC, is a strategy which is largely accepted in the area of machine learning and which assures local convergence on expectation, given the step sizes are sufficiently small. However, OSC is not a pure gradient based method due to the entangled orthogonalization steps. Therefore, we have proven that the non-trivial ONB update by $N$-OSC reduces the costs for the presented sample, given the learning rate is sufficiently small. It that sense, one can deduce that $N$-OSC performs a stochastic descent.

We have formulated the dictionary learning methods for the constrained $K$-sparse model. We studied the encoding performance of the learned dictionaries in terms of the associated cost function of that model. Redesigning our learning methods to comply with the unconstrained regularized model is straight forward and has been outlined in Section 2.4. Based on a spot-check, we can confirm that equivalent performance properties hold when the learning methods are modified for the alternative unconstrained model. However, a thorough experimental comparison is not provided in this thesis and remains as a future task.

In the scope of this thesis, we have focussed on the complete setting, i.e. the ONB learning task. However, there are scenarios (e.g. high data dimensionality) for which it might be reasonable to learn instead an undercomplete orthogonal sparse coding dictionary consisting of $M < N$ atoms. Note that particular methods such as BOCA [Dobigeon and Tourneret, 2010] are specifically defined for this setting. It would be interesting to adapt our methods accordingly for the sake of a comparison. At least for CA and OSC a corresponding extension is straight forward. In the case of CA, the OPP $\mathrm{P}_{(2.27)}$ has to be solved subject to a non-square matrix, which implies to replace line 5 of Algorithm 1, i.e. $\mathbf{U}_{(t)} \leftarrow \mathbf{V}\mathbf{W}^T$ by $\mathbf{U}_{(t)} \leftarrow \mathbf{V}\mathbf{I}_{N\times M}\mathbf{W}^T$, where $\mathbf{I}_{N\times M} \in \{0,1\}^{N\times M}$ is a non-square diagonal matrix with entries 1 on the diagonal. In the case of OSC, the number of iterations of the loop starting in line 7 of Algorithm 2 has to be adapted to the dictionary size, i.e. $N$ is replaced by $M$. In the case of GF-OSC, a corresponding modification is not equally straight-forward. Analogous to the Geodesic-Flow framework, a differentiable parameterization of undercomplete orthogonal dictionaries, i.e. orthonormal $M$-frames in $\mathbb{R}^N$, would be required. The Stiefel manifold $\mathrm{St}_{M,N}$ is a manifold object containing all such $M$-frames and can be seen as the counterpart of the orthogonal group $\mathrm{O}(N)$ which contains all ONBs in $\mathbb{R}^N$. Developing a learning algorithm could be based, e.g. on the Cayley transform, and might be supported by approaches proposed in [Fraikin et al., 2007, Edelman et al., 1999].

With our ONB recovery experiments, we studied carefully how the data sparsity level influences the ability of the individual methods to recover the reference ONB, while the data dimensionality and the sample size was fix. Similar to [Lesage et al., 2005], a way to further stress our findings would be to extend the experimental setup by systematically varying the fixed parameters $N$, $L$ as well as the noise level in order to investigate their role regarding a successful recovery.

Most recently, new methods [Rusu et al., 2016, Rusu and Thompson, 2017] have been proposed to solve the learning task we have considered here. While the authors of [Rusu and Thompson, 2017] are aware of our work [Schütze et al., 2016], a corresponding experimental comparison with OSC, and also with GF-OSC, is yet missing and should be done in the future in order to complement the big picture of orthogonal dictionary learning algorithms.

## 4.2 Adaptive Hierarchical Sensing

Furthermore, we have addressed the problem of sampling an unknown signal, which has an unknown sparse or compressible representation in a known transform basis, by collecting only a small number of adaptive linear measurements. We introduced the sensing scheme adaptive hierarchical sensing (AHS) as an alternative to non-adaptive Compressed Sensing (CS). The main differences are that for AHS (*i*) the analysis transform basis is chosen prior to sensing because the sensing vectors depend on the transform, (*ii*) the sensing vectors are adaptively selected by computationally performant decision rules (applied to previously collected measurements), and (*iii*) the sparse signal representation is obtained without solving an inverse optimization problem.

We proposed two sensing algorithms: $\tau$-AHS and $K$-AHS. $\tau$-AHS selects the sensing vectors based on absolute comparisons of the measurements with an adaptive threshold, whereas $K$-AHS makes that selection based on relative comparisons. For strictly $k$-sparse signals, $\tau$-AHS collects at most $2k(log_2 N/k + 1)$ measurements. For other signals, the number of $\tau$-AHS measurements and the sparsity of the resulting signal representation depends on the choice of a canonical threshold parameter $\tau_0$. $K$-AHS, on the other hand, entails a much better control regarding the number of measurements. The user parameter $K$ determines the target sparsity of the signal representation. An additional parameter controls the number of measurements. Its default value leads for any signal to at most $2K \log_2 N/K$ measurements.

Experiments, in which sensing is simulated based on natural test images, revealed that $K$-AHS outperforms $\tau$-AHS and that $K$-AHS is highly competitive with a common non-adaptive CS approach (based on $\ell_1$-norm minimization) in terms of reconstruction accuracy as measured by the peak-signal-to-noise ratio (PSNR). Particularly for the relevant case of small numbers of measurements, $K$-AHS achieves even higher PSNRs than the standard CS approach using a noiselet measurement ensemble that is highly incoherent to the wavelet transform domains we have considered. The superiority of $K$-AHS in this scenario is noticeable by the fact that the resulting images do not suffer from high-frequency noise as opposed to the images recovered by CS.

Our image sensing experiments are designed with the intention to make the comparison between $K$-AHS and CS as fair as possible, i.e. images are assumed to be compressible in the sparse transform domain and the locations of the coefficients are assumed to be uniformly distributed. Note, however, that CS based on $\ell_1$-norm minimization is not necessarily the optimal recovery strategy for compressively sensing large-scale image data. Perhaps, better compressive imaging results could be obtained by more sophisticated recovery approaches, which additionally exploit e.g. (*i*) spatial smoothness properties during the optimization procedure (e.g. minimizing the TV-norm of the reconstructed image [Pant et al., 2013] rather than the $\ell_1$-norm of the sparse image representation) or (*ii*) exploit structural properties of the sparse model such as the wavelet tree, see e.g. [Baraniuk et al., 2010]. Future research could investigate the possibility of AHS extensions that similarly exploit such domain specific signal properties to allow for a corresponding comparison.

$K$-AHS could be easily implemented using existing CS (imaging) hardware. Only minor modifications are necessary to realize a feedback of the acquired measurements, meaning that the processing unit which controls the spatial patterns of the sensing vectors requires memory access to essentially the $2K$ most recent measurements. Every $2K$-th sensing operation a partial sorting of the $2K$ most recent measurements is required, which implies only few additional computations. For $\tau$-AHS the situation is equally simple. Complementary experiments should investigate to what extent the AHS performance of our simulations can be obtained with real sensing hardware. Practice-oriented aspects of sensing, such as measurement noise, might be a relevant factor and its role should be assessed.

The collection of potential AHS sensing vectors can be entirely precomputed and stored in the memory. Thus, each sensing vector requested by AHS does not have to be computed during the sensing process but can be instantly loaded on demand. This kind of precaching saves computational resources and time and is an advantage compared to other adaptive sensing approaches. On the other hand, precaching consumes memory of the order $\mathcal{O}(N^2)$, which can be limiting in certain applications. If memory of that size is unavailable, then each sensing vector requested by AHS can be alternatively computed on demand, which takes the computational time of one analysis transform of an auxiliary vector containing the weights. If the analysis basis is a fast transform (e.g. the Discrete Cosine Transform (DCT) or the Fast Wavelet Transform (FWT)) the computational demand would be equivalent to state-of-the art CS realizations with measurement ensembles based on fast transforms (e.g. the Fast Fourier Transform (FFT) or the Fast Noiselet Transform (FNT)).

Based on a heuristic for increasing the sensing performance, we proposed an approach to learn the weights of the linear combinations of the AHS sensing vectors and their placement within the sensing tree. We demonstrated for natural image patches that the proposed learning scheme not only leads to meaningful spatial structures of the sensing vectors but that it indeed increases sensing accuracy for signals of a test data set. It would be interesting to investigate by future experiments if this approach works similarly well for higher dimensional image data, i.e. mid-scale or large-scale images. In a future study, one could try to derive a structuring approach which combines the learning of the weights with the learning of the analysis transform basis.

We analyzed the sensing quality of $K$-AHS theoretically and studied scenarios in which the $k$ ($k \leq K$) most significant coefficients of a (not necessarily strictly sparse) signal are detected. We derived a sufficient optimality condition which deterministically guarantees sensing success in that regard. The premise of that condition is that the smallest magnitude among all subset sums of the $k$ significant coefficients is sufficiently large compared to the sum of magnitudes of a particular number of the principal non-significant coefficients. We extended the accessibility of our result and applied the condition to three signal models and obtained sufficient conditions depending on the model parameters. Sensing experiments with synthetic signals randomly generated according to these models confirmed our theoretical predictions. The applicability of the condition is limited for the power law model as we can use it merely for the case $k = 1$, i.e. for detecting the most significant coefficient. The empirical AHS performance for that model is however satisfactory.

For our probabilistic analysis, the deterministic condition is relaxed by considering the cardinality of problematic subsets of the most significant coefficients and by deriving a combinatorial bound. We assumed that the locations of the significant coefficients are uniformly distributed. Uniformity is assumed for technical reasons, and can not be necessarily applied in particular cases, e.g. an optimized sensing tree structure. Note also that both the deterministic and the probabilistic analyses assume that all sensing vectors of the same sensing tree level are combined by the same constant weight. An extension of the theoretical analysis for heterogeneous weights seems challenging and remains as a future task.

# Bibliography

[Aharon et al., 2006] Aharon, M., Elad, M., and Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322.

[Ahmed et al., 1974] Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93.

[Akram Aldroubi and Zarringhalam, 2011] Akram Aldroubi, H. W. and Zarringhalam, K. (2011). Sequential adaptive compressed sampling via huffman codes. In *Sampling Theory in Signal and Image Processing*, volume 10, pages 231–254.

[Arias-Castro et al., 2013] Arias-Castro, E., Candes, E. J., and Davenport, M. A. (2013). On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 59(1):472–481.

[Armijo, 1966] Armijo, L. (1966). Minimization of functions having lipschitz continuous first partial derivatives. *Pacific J. Math.*, 16(1):1–3.

[Bao et al., 2013] Bao, C., Cai, J.-F., and Ji, H. (2013). Fast sparsity-based orthogonal dictionary learning for image restoration. In *The IEEE International Conference on Computer Vision (ICCV)*.

[Bao et al., 2016] Bao, C., Ji, H., Quan, Y., and Shen, Z. (2016). Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1356–1369.

[Bao et al., 2015] Bao, C., Ji, H., and Shen, Z. (2015). Convergence analysis for iterative data-driven tight frame construction scheme. *Applied and Computational Harmonic Analysis*, 38(3):510–523.

[Baraniuk and Steeghs, 2007] Baraniuk, R. and Steeghs, P. (2007). Compressive radar imaging. In *2007 IEEE Radar Conference*, pages 128–133.

[Baraniuk et al., 2010] Baraniuk, R. G., Cevher, V., Duarte, M. F., and Hegde, C. (2010). Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001.

[Barlow, 1961] Barlow, H. B. (1961). Possible principles underlying the transformation of sensory messages. *Sensory Communication*, pages 217–234.

[Becker et al., 2014] Becker, A., Richter, T., and Fröhling, N. (2014). Image Compression | Benchmark.

[Becker et al., 2011] Becker, S., Bobin, J., and Candès, E. J. (2011). Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sciences*, 4(1):1–39.

[Bengio et al., 2013] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

[Blanchard and Tanner, 2015] Blanchard, J. D. and Tanner, J. (2015). Performance comparisons of greedy algorithms in compressed sensing. *Numerical Lin. Alg. with Applic.*, 22(2):254–282.

[Blumensath et al., 2012] Blumensath, T., Davies, M. E., and Rilling, G. (2012). *Greedy algorithms for compressed sensing*, pages 348–393. Cambridge University Press.

[Boche et al., 2015] Boche, H., Calderbank, R., Kutyniok, G., and Vybíral, J. (2015). *A Survey of Compressed Sensing*, pages 1–39. Springer International Publishing, Cham.

[Bryt and Elad, 2008] Bryt, O. and Elad, M. (2008). Compression of facial images using the k-svd algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270 – 282.

[Burciu et al., 2016] Burciu, I., Martinetz, T., and Barth, E. (2016). Hierarchical Manifold Sensing with foveation and adaptive partitioning of the dataset. *Journal of Imaging Science and Technology*, 60(2):20402:1–10.

[Cai et al., 2014] Cai, J.-F., Ji, H., Shen, Z., and Ye, G.-B. (2014). Data-driven tight frame construction and image denoising. *Applied and Computational Harmonic Analysis*, 37(1):89 – 105.

[Candes, 2008] Candes, E. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589–592.

[Candes and Romberg, 2007] Candes, E. and Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969.

[Candès et al., 2006] Candès, E. J., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509.

[Candès et al., 2006] Candès, E. J., Romberg, J. K., and Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223.

[Candes and Tao, 2005] Candes, E. J. and Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215.

[Candes and Tao, 2006] Candes, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theor.*, 52(12):5406–5425.

[Candès et al., 2010] Candès, E. J., Eldar, Y. C., and Needell, D. (2010). Compressed sensing with coherent and redundant dictionaries. *CoRR*, abs/1005.2613.

[Castro et al., 2008] Castro, R. M., Haupt, J., Nowak, R., and Raz, G. M. (2008). Finding needles in noisy haystacks. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5133–5136.

[Chambers, 1971] Chambers, J. M. (1971). Algorithm 410: Partial sorting. *Commun. ACM*, 14(5):357–358.

[Chen et al., 1998] Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61.

[Coifman et al., 1990] Coifman, R. R., Meyer, Y., and Wickerhauser, V. (1990). Wavelet analysis and signal processing. In Auslander, L., Kailath, T., and Mitter, S. K., editors, *Signal Processing, Part I: Signal Processing Theory*, pages 59–68. Springer-Verlag, New York, NY.

[Coulter et al., 2010] Coulter, W., Hillar, C., Isley, G., and Sommer, F. (2010). Adaptive compressed sensing: A new class of self-organizing coding models for neuroscience. In *IEEE International Conference on Acoustics Speech and Signal Processing*, volume 5370, pages 5494–5497.

[Crochet et al., 2011] Crochet, S., Poulet, J., Kremer, Y., and Petersen, C. (2011). Synaptic mechanisms underlying sparse coding of active touch. *Neuron*, 69(6):1160 – 1175.

[Davenport et al., 2012] Davenport, M. A., Duarte, M. F., Eldar, Y. C., and Kutyniok, G. (2012). *Introduction to Compressed Sensing*, pages 1–64. Cambridge University Press.

[Davis et al., 1997] Davis, G., Mallat, S., and Avellaneda, M. (1997). Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98.

[DeGroot, 1962] DeGroot, M. H. (1962). Uncertainty, information, and sequential experiments. *Ann. Math. Statist.*, 33(2):404–419.

[Deutsch et al., 2009] Deutsch, S., Averbuch, A., and Dekel, S. (2009). Adaptive compressed image sensing based on wavelet modeling and direct sampling. In *International Conference on Sampling Theory and Applications (SAMPTA'09).* Tel Aviv University.

[DeVore, 1998] DeVore, R. A. (1998). Nonlinear approximation. *ACTA NUMERICA*, 7:51–150.

[Dobigeon and Tourneret, 2010] Dobigeon, N. and Tourneret, J.-Y. (2010). Bayesian orthogonal component analysis for sparse representation. *IEEE Trans. Signal Process.*, 58(5):2675–2685.

[Donoho, 2005] Donoho, D. L. (2005). Neighborly polytopes and sparse solutions of underdetermined linear equations. Technical report.

[Donoho, 2006] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

[Donoho and Elad, 2003] Donoho, D. L. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202.

[Donoho et al., 1998] Donoho, D. L., Vetterli, M., DeVore, R. A., and Daubechies, I. (1998). Data compression and harmonic analysis. *IEEE Trans. Information Theory*, 44(6):2435–2476.

[Dorfman, 1943] Dorfman, R. (1943). The detection of defective members of large populations. *Ann. Math. Statist.*, 14(4):436–440.

[Du and Hwang, 2000] Du, D. and Hwang, F. (2000). *Combinatorial Group Testing and Its Applications.* Applied Mathematics. World Scientific.

[Edelman et al., 1999] Edelman, A., Arias, T. A., and Smith, S. T. (1999). The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353.

[Elad, 2010] Elad, M. (2010). *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing.* Springer Publishing Company, Incorporated, 1st edition.

[Elad and Aharon, 2006] Elad, M. and Aharon, M. (2006). Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745.

[Eldar and Kutyniok, 2012] Eldar, Y. C. and Kutyniok, G., editors (2012). *Compressed sensing : theory and applications*. Cambridge University Press, Cambridge, New York.

[Ender, 2010] Ender, J. H. (2010). On compressive sensing applied to radar. *Signal Processing*, 90(5):1402–1414.

[Field, 1994] Field, D. J. (1994). What is the goal of sensory coding? *Neural Computation*, 6(4):559–601.

[Flajolet and Sedgewick, 2009] Flajolet, P. and Sedgewick, R. (2009). *Analytic Combinatorics*. Cambridge University Press, New York, NY, USA, 1 edition.

[Fraikin et al., 2007] Fraikin, C., Hüper, K., and Dooren, P. V. (2007). Optimization over the stiefel manifold. *PAMM*, 7(1):1062205–1062206.

[Gamper et al., 2008] Gamper, U., Boesiger, P., and Kozerke, S. (2008). Compressed sensing in dynamic mri. *Magnetic Resonance in Medicine*, 59(2):365 – 373.

[Geisler and Perry, 2011] Geisler, W. S. and Perry, J. S. (2011). Statistics for optimal point prediction in natural images. *Journal of Vision*, 11(12).

[Gribonval and Schnass, 2008] Gribonval, R. and Schnass, K. (2008). Dictionary Identifiability from Few Training Samples. In *European Signal Processing Conference (EUSIPCO'08)*, Lausanne, Switzerland.

[Haupt et al., 2011] Haupt, J., Castro, R. M., and Nowak, R. (2011). Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 57(9):6222–6235.

[Haupt and Nowak, 2012] Haupt, J. and Nowak, R. (2012). Adaptive sensing for sparse recovery. In Eldar, Y. C. and Kutyniok, G., editors, *Compressed Sensing*, pages 269–304. Cambridge University Press. Cambridge Books Online.

[Haupt et al., 2009] Haupt, J. D., Baraniuk, R. G., Castro, R. M., and Nowak, R. D. (2009). Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements. In *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pages 1551–1555.

[Hennenfent and Herrmann, 2008] Hennenfent, G. and Herrmann, F. J. (2008). Simply denoise: wavefield reconstruction via jittered undersampling. *Geophysics*, 73(3):V19–V28.

[Herman and Strohmer, 2009] Herman, M. A. and Strohmer, T. (2009). High-resolution radar via compressed sensing. *IEEE Transactions on Signal Processing*, 57(6):2275–2284.

[Herman et al., 2015] Herman, M. A., Weston, T., McMackin, L., Li, Y., Chen, J., and Kelly, K. F. (2015). Recent results in single-pixel compressive imaging using selective measurement strategies. volume 9484, pages 94840A–94840A–18.

[Herrmann and Hennenfent, 2008] Herrmann, F. J. and Hennenfent, G. (2008). Non-parametric seismic data recovery with curvelet frames. *Geophysical Journal International*, 173:233–248.

[Heubach and Mansour, 2009] Heubach, S. and Mansour, T. (2009). *Combinatorics of Compositions and Words: Solutions Manual.* Chapman & Hall/CRC.

[Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.*, 24.

[Hromádka et al., 2008] Hromádka, T., Deweese, M. R., and Zador, A. M. (2008). Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS biology*, 6(1):e16+.

[Hwang, 1972] Hwang, F. K. (1972). A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67:605–608.

[Ito et al., 2008] Ito, I., Ong, R. C., Raman, B., and Stopfer, M. (2008). Sparse odor representation and olfactory learning. *Nature Neuroscience*, 11(10):1177–1184.

[Iwen and Tewfik, 2012] Iwen, M. A. and Tewfik, A. H. (2012). Adaptive strategies for target detection and localization in noisy environments. *IEEE Transactions on Signal Processing*, 60(5):2344–2353.

[Ji et al., 2008] Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356.

[Jung et al., 2009] Jung, H., Sung, K., Nayak, K. S., Kim, E. Y., and Ye, J. C. (2009). k-t focuss: A general compressed sensing framework for high resolution dynamic mri. *Magnetic Resonance in Medicine*, 61(1):103–116.

[Kanerva, 1988] Kanerva, P. (1988). *Sparse Distributed Memory.* MIT Press, Cambridge, MA, USA.

[Kutyniok, 2012] Kutyniok, G. (2012). Compressed sensing: Theory and applications. *CoRR*, abs/1203.3815.

[Labusch et al., 2008] Labusch, K., Barth, E., and Martinetz, T. (2008). Simple method for high-performance digit recognition based on sparse coding. *IEEE Transactions on Neural Networks*, 19(11):1985–1989.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324.

[Lee et al., 2007] Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2007). Efficient sparse coding algorithms. In Schölkopf, P. B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press.

[Lesage et al., 2005] Lesage, S., Gribonval, R., Bimbot, F., and Benaroya, L. (2005). Learning Unions of Orthonormal Bases with Thresholded Singular Value Decomposition. In *Acoustics, Speech and Signal Processing, 2005. ICASSP 2005. IEEE International Conference on*, volume V, pages V/293–V/296, Philadelphia, PA, United States. IEEE.

[Lewicki and Sejnowski, 2000] Lewicki, M. S. and Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural Comput.*, 12(2):337–365.

[Lin et al., 2014] Lin, A. C., Bygrave, A. M., de Calignon, A., Lee, T., and Miesenbock, G. (2014). Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nature Neuroscience*, 17(4):559–568.

[Lindley, 1956] Lindley, D. V. (1956). On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005.

[Lustig et al., 2008] Lustig, M., Donoho, D. L., Santos, J. M., and Pauly, J. M. (2008). Compressed sensing mri. *IEEE Signal Processing Magazine*, 25(2):72–82.

[Mach, 1886] Mach, E. (1886). *Die Analyse der Empfindungen und das Verhältnis des Physischen zum Psychischen.* Fischer.

[MacKay, 1956] MacKay, D. M. (1956). Towards an information-flow model of human behaviour. *British Journal of Psychology*, 47(1):30–43.

[Mairal et al., 2008a] Mairal, J., Elad, M., and Sapiro, G. (2008a). Sparse representation for color image restoration. *Trans. Img. Proc.*, 17(1):53–69.

[Mairal et al., 2009] Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. R. (2009). Supervised dictionary learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 1033–1040. Curran Associates, Inc.

[Mairal et al., 2008b] Mairal, J., Sapiro, G., and Elad, M. (2008b). Learning Multiscale Sparse Representations for Image and Video Restoration. *Multiscale Modeling & Simulation*, 7(1):214–241.

[Mallat, 2008] Mallat, S. (2008). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3 edition.

[Malloy and Nowak, 2014] Malloy, M. L. and Nowak, R. D. (2014). Near-optimal adaptive compressed sensing. *IEEE Transactions on Information Theory*, 60(7):4001–4012.

[Mishali and Eldar, 2009] Mishali, M. and Eldar, Y. (2009). Sparse source separation from orthogonal mixtures. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 3145–3148.

[Moler and Loan, 2003] Moler, C. and Loan, C. V. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49.

[Nesterov, 2005] Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

[Oja, 1982] Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273.

[Oja, 2002] Oja, E. (2002). Unsupervised learning in neural computation. *Theor. Comput. Sci.*, 287(1):187–207.

[Olshausen and Field, 2004] Olshausen, B. and Field, D. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487.

[Olshausen and Field, 1996a] Olshausen, B. A. and Field, D. J. (1996a). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, (381):607–609.

[Olshausen and Field, 1996b] Olshausen, B. A. and Field, D. J. (1996b). Natural image statistics and efficient coding. *Network*, 7(2):333–339.

[Olshausen and Field, 1997] Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37(23):3311–3325.

[Page, 2013] Page, D. R. (2013). Generalized algorithm for restricted weak composition generation. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 12(4):345–372.

[Palm, 2013] Palm, G. (2013). Neural associative memories and sparse coding. *Neural Netw.*, 37:165–171.

[Pant et al., 2013] Pant, J. K., Lu, W. S., and Antoniou, A. (2013). A new algorithm for compressive sensing based on total-variation norm. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 1352–1355.

[Pati et al., 2015] Pati, N., Pradhan, A., Kanoje, L. K., and Das, T. K. (2015). An approach to image compression by using sparse approximation technique. *Procedia Computer Science*, 48:769 – 775. International Conference on Computer, Communication and Convergence (ICCC 2015).

[Pati et al., 1993] Pati, Y., Rezaiifar, R., and Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers.*

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572.

[Pennebaker and Mitchell, 1992] Pennebaker, W. B. and Mitchell, J. L. (1992). *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, New York.

[Pereira et al., 2014] Pereira, M. P., Lovisolo, L., da Silva, E. A., and de Campos, M. L. (2014). On the design of maximally incoherent sensing matrices for compressed sensing using orthogonal bases and its extension for biorthogonal bases case. *Digital Signal Processing*, 27:12 – 22.

[Plumbley, 2004] Plumbley, M. D. (2004). Lie Group Methods for Optimization with Orthogonality Constraints. *Independent Component Analysis and Blind Signal Separation*, pages 1245–1252.

[Potter et al., 2010] Potter, L. C., Ertin, E., Parker, J. T., and Çetin, M. (2010). Sparsity and compressed sensing in radar imaging. *Proceedings of the IEEE*, 98(6):1006–1020.

[Qin et al., 2016] Qin, Z.-t., Yang, W.-n., Wu, X.-p., and Yang, R. (2016). *Hyperspectral Image Classification Using a New Dictionary Learning Approach with Structured Sparse Representation*, pages 719–722. Springer International Publishing, Cham.

[Rebollo-Neira and Lowe, 2002] Rebollo-Neira, L. and Lowe, D. (2002). Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 9(4):137–140.

[Roweis and Ghahramani, 1999] Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural Comput.*, 11(2):305–345.

[Rubinstein et al., 2010] Rubinstein, R., Bruckstein, A. M., and Elad, M. (2010). Dictionaries for Sparse Representation Modeling. *Proceedings of the IEEE*, 98(6):1045–1057.

[Rubinstein et al., 2008] Rubinstein, R., Zibulevsky, M., and Elad, M. (2008). Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit. Technical report.

[Rusu et al., 2016] Rusu, C., González-Prelcic, N., and Heath, R. W. (2016). Fast orthonormal sparsifying transforms based on householder reflectors. *IEEE Transactions on Signal Processing*, 64(24):6589–6599.

[Rusu and Thompson, 2017] Rusu, C. and Thompson, J. (2017). Learning fast sparsifying transforms. *IEEE Transactions on Signal Processing*, PP(99):1–1.

[Schönemann, 1966] Schönemann, P. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

[Schütze et al., 2013] Schütze, H., Barth, E., and Martinetz, T. (2013). Learning orthogonal bases for k-sparse representations. In Hammer, B., Martinetz, T., and Villmann, T., editors, *Workshop New Challenges in Neural Computation 2013*, volume 02/2013 of *Machine Learning Reports*, pages 119–120.

[Schütze et al., 2014] Schütze, H., Barth, E., and Martinetz, T. (2014). An adaptive hierarchical sensing scheme for sparse signals. In Rogowitz, B. E., Pappas, T. N., and de Ridder, H., editors, *Human Vision and Electronic Imaging XIX*, volume 9014 of *Proc. of SPIE Electronic Imaging*, pages 15:1–8.

[Schütze et al., 2016] Schütze, H., Barth, E., and Martinetz, T. (2016). Learning Efficient Data Representations with Orthogonal Sparse Coding. *IEEE Transactions on Computational Imaging*, 2(3):177–189.

[Schütze et al., 2017] Schütze, H., Barth, E., and Martinetz, T. (2017). Adaptive Hierarchical Sensing for the Efficient Sampling of Sparse and Compressible Signals. in preparation.

[Schütze et al., 2012] Schütze, H., Martinetz, T., Anders, S., and Madany Mamlouk, A. (2012). A Multivariate Approach to Estimate Complexity of FMRI Time Series. In Villa, A. E., Duch, W., Érdi, P., Masulli, F., and Palm, G., editors, *22nd International Conference on Artificial Neural Networks and Machine Learning*, volume 7553 of *Lecture Notes in Computer Science*, pages 540–547. Springer.

[Schütze et al., 2015] Schütze, H., Barth, E., and Martinetz, T. (2015). Learning orthogonal sparse representations by using geodesic flow optimization. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

[Seeger, 2008] Seeger, M. (2008). Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813.

[Seeger and Nickisch, 2008] Seeger, M. and Nickisch, H. (2008). Compressed sensing and bayesian experimental design. In *ICML 2008*, pages 912–919, New York, NY, USA. Max-Planck-Gesellschaft, ACM Press.

[Sezer et al., 2015] Sezer, O. G., Guleryuz, O. G., and Altunbasak, Y. (2015). Approximation and compression with sparse orthonormal transforms. *IEEE Transactions on Image Processing*, 24(8):2328–2343.

[Sezer et al., 2008] Sezer, O. G., Harmanci, O., and Guleryuz, O. G. (2008). Sparse orthonormal transforms for image compression. In *ICIP*, pages 149–152. IEEE.

[Skretting and Engan, 2011] Skretting, K. and Engan, K. (2011). Image compression using learned dictionaries by rls-dla and compared with k-svd. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1517–1520.

[Sun et al., 2013] Sun, B., Edgar, M. P., Bowman, R., Vittert, L. E., Welsh, S., Bowman, A., and Padgett, M. J. (2013). 3D Computational Imaging with Single-Pixel Detectors. *Science*, 340(6134):844–847.

[Sundaresan and Porikli, 2012] Sundaresan, R. and Porikli, F. (2012). Additive noise removal by sparse reconstruction on image affinity nets. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 1137–1140.

[Takhar et al., 2006] 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 *Proceedings of Computational Imaging IV at SPIE Electronic Imaging*, pages 43–52, San Jose, CA.

[Talukder and Harada, 2007] Talukder, K. H. and Harada, K. (2007). Haar wavelet based approach for image compression and quality assessment of compressed image. *IAENG International Journal of Applied Mathematics*, 36(1):49–56.

[Taubman and Marcellin, 2013] Taubman, D. and Marcellin, M. (2013). *JPEG2000 Image Compression Fundamentals, Standards and Practice.* Springer Publishing Company, Incorporated.

[Tuma and Hurley, 2009] Tuma, T. and Hurley, P. (2009). On the incoherence of noiselet and haar bases.

[Wakin et al., 2006a] Wakin, M. B., Laska, J. N., Duarte, M. F., Baron, D., Sarvotham, S., Takhar, D., Kelly, K. F., and Baraniuk, R. G. (2006a). An architecture for compressive imaging. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 1273–1276, Atlanta, GA.

[Wakin et al., 2006b] Wakin, M. B., Laska, J. N., Duarte, M. F., Baron, D., Sarvotham, S., Takhar, D., Kelly, K. F., and Baraniuk, R. G. (2006b). Compressive imaging for video representation and coding. In *Proceedings of the Picture Coding Symposium (PCS)*, Beijing, China.

[Wang et al., 2003] Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multi-scale structural similarity for image quality assessment. In *in Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers, (Asilomar*, pages 1398–1402.

[Welsh et al., 2013] Welsh, S. S., Edgar, M. P., Bowman, R., Jonathan, P., Sun, B., and Padgett, M. J. (2013). Fast full-color computational imaging with single-pixel detectors. *Opt. Express*, 21(20):23068–23074.

[Willmore and King, 2009] Willmore, B. D. and King, A. (2009). Auditory cortex: Representation through sparsification? *Current Biology*, 19(24):R1123 – R1125.

[Xiang et al., 2015] Xiang, S., Meng, G., Wang, Y., Pan, C., and Zhang, C. (2015). Image deblurring with coupled dictionary learning. *International Journal of Computer Vision*, 114(2):248–271.

[Yang et al., 2014] Yang, H., Zhu, M., Wu, X., Zhang, Z., and Huang, H. (2014). Dictionary learning approach for image deconvolution with variance estimation. *Appl. Opt.*, 53(25):5677–5684.

[Zetzsche et al., 1993] Zetzsche, C., Barth, E., and Wegmann, B. (1993). The importance of intrinsically two-dimensional image features in biological vision and picture coding. In *Digital Images and Human Vision*, pages 109–38.