Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Fachgebiet für Geometrische Optimierung und Maschinelles Lernen

Learning Sparse Data Models via Geometric Optimization with Applications to Image Processing

Simon Alois Hawe

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Wolfgang Kellerer

Prüfer der Dissertation:

1. Jun.-Prof. Dr. rer. nat. Martin Kleinsteuber
2. Prof. Michael Elad, D. Sc. Israel Institute of Technology, Haifa, Israel (schriftliche Beurteilung)
2. Univ.-Prof. Dr.-Ing. Eckehard Steinbach (mündliche Prüfung)

Die Dissertation wurde am 27.05.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 13.11.2013 angenommen.

Simon Alois Hawe. *Learning Sparse Data Models via Geometric Optimization with Applications to Image Processing.* Dissertation, Technische Universität München, Munich, Germany, 2014.

To my family and Anna

If we knew what it was we were doing,
it would not be called research,
would it?

–ALBERT EINSTEIN–

Acknowledgments

I like to take this opportunity to thank all the people who supported me during the past years, and without whom writing this thesis would not have been possible.

First and foremost I like to thank my supervisor and mentor, Prof. Martin Kleinsteuber, not only for providing me with such an interesting and challenging research topic, but also for the opportunities and the support he gave me during the past years. All those fruitful and sometimes funny academic disputations we had motivated me to keep on track and eventually allowed me setting up this thesis.

To me it is a great honor that Prof. Michael Elad, who is one of the most influential researchers in the field of sparse data modeling, agreed to be the second reviewer of my thesis. Thank you for that Michael.

Special thanks go to Prof. Klaus Diepold for providing me the opportunity working at his institute and for his supervision. Teaching computer vision together with him was a great pleasure for me and a time where I learned a lot.

Thanks to my very best friends and colleagues Johannes, Graci, Tobi, Huy, Jan, Robert, Schorsch, Max, August, Tina, Jupp, Uli, Clemens, Marko, Martin, James, Tim, Julian, Vince, and Rori for proof reading this thesis, for all those many academic discussions, and for having an awesome time together in the past and hopefully more of it in the future. Always remember: *Mia san Mia!*.

I like to thank my parents Vreni and Wolfried, my grandmother Lotte, my sisters Martina and Andrea, my brother Benno, and my brother-in-law Michael for their continued encouragement and support of all my plans and ideas.

Last, I like to thank my beloved girlfriend Anna for always supporting me, being with me, and simply taking me as I am.

Abstract

Exploiting data models lies at the core of many algorithms and techniques in neuroscience, machine learning, and signal processing. In this context, models based on sparse representations have been proven extremely valuable, and numerous algorithms for exploiting sparsity in applications and for learning sparse data models have been proposed over the last years. Two related but distinctive approaches have emerged that are known as the sparse synthesis model and the co-sparse analysis model.

The underlying assumption of the synthesis model is that a signal can be formed by linearly combining a set of building blocks, known as the atoms of a dictionary. Therein, sparsity comes into play by requiring this set to be as small as possible. In contrast to this, in the analysis model a signal is mapped to a higher dimensional space by an analysis operator and the image of this mapping is assumed to be sparse. One famous application of these models is regularizing inverse problems in image processing. Most crucial for the performance of both models is the choice of a proper dictionary/analysis operator. They should be chosen such that a maximally sparse representation of the considered signal can be achieved. Either they can be defined analytically, or be learned from representative training samples of the considered signal class. It is well-known that learned models outperform analytic ones as they are better adapted to the data of interest, which in turn leads to sparser representations. Furthermore, learning algorithms allow to find sparse representation of classes of data for which no analytic model exists.

The focus of my thesis lies on learning sparse data models considering both the synthesis and the analysis point of view with emphasis on their applications to image processing tasks. Concretely, in the first part I introduce two new algorithms called Separable Dictionary Learning (SeDiL) and Geometric Analysis Operator Learning (GOAL) that are based on geometric conjugate gradient optimization on suitable manifolds. Although these methods are general in terms of being independent of a specific signal class or application, here, my major interest in terms of applications lies on image data and solving classical linear inverse problems.

SeDiL operates on the product of unit spheres and can be used to learn both unstructured conventional dictionaries as well as dictionaries having a separable structure. Compared to the conventional approach, enforcing a separable structure allows learning dictionaries of higher dimension and reduces the computational burden for both learning the dictionary and employing it in applications. This is of special interest for image processing tasks as it allows working with larger image-patches and thus permits to capture larger image structures. Another advantage of SeDiL is the possibility to control the crucial mutual coherence of a learned dictionary.

The proposed analysis operator learning approach GOAL works on the oblique manifold, i.e. the set of full-rank matrices with normalized columns. During the learning process, the condition number and the mutual coherence of an operator are controlled, and the trivial solution is avoided inherently. Through synthetic tests, I show that GOAL outperforms all existing analysis operator learning techniques in terms of computational complexity, accuracy in finding a generating ground truth operator, and generality. Furthermore, several results for image denoising, inpainting, and superresolution reveal the state-of-the-art performance of GOAL in real world applications.

In the second part, I introduce the novel multimodal co-sparse analysis model that permits to model statistical dependencies of diverse modalities representing the same physical object. This model suggests that measurements acquired in different modalities originating from the same scene share a common co-support when a suitable set of analysis operators is used. For this set of operators, no analytic form exists and these operators must be learned from aligned example signals. To that end, I propose an extension of GOAL that uses a suitable sparsifying function to enforce the coupled co-support assumption during the learning process. The performance of the proposed model is evaluated for the task of depth map superresolution based on aligned depth- and intensity-information.

Contents

List of Figures

List of Tables

List of Symbols

$\mathcal{I}, \mathcal{J}, \mathcal{K}$	Sets, written as capital calligraphic letters.
$\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}$	Matrices, written as capital boldface letters.
$\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}$	Column vectors, written as lowercase boldface letters.
F, g, h	Scalars, written as lowercase or capital letters.
$\boldsymbol{f}_{:,j}$	j-th column of matrix $\boldsymbol{F}$.
$\boldsymbol{f}_{i,:}$	i-th row of matrix $\boldsymbol{F}$.
f_{ij}	i-th element in the j-th column of matrix $\boldsymbol{F}$.
f_i	i-th entry of vector $\boldsymbol{f}$.
$\boldsymbol{F}^{(i)}, \boldsymbol{f}^{(i)}, f^{(i)}$	Elements at the i-th iteration.
$\boldsymbol{F}^{\dagger}$	Pseudoinverse of $\boldsymbol{F}$.
$\boldsymbol{F}^{\top}, \boldsymbol{f}^{\top}$	Matrix/Vector transposed.
$\boldsymbol{I}_k$	Identity matrix of dimension $(k \times k)$.
$\boldsymbol{0}_f, \boldsymbol{0}_{h \times k}$	All zero vector and matrix of dimension f and $(h \times k)$, respectively.
$\boldsymbol{D} \in \mathbb{R}^{n \times d}$	Dictionary.
$\boldsymbol{\Omega} \in \mathbb{R}^{a \times n}$	Analysis Operator.
$\boldsymbol{A} \in \mathbb{R}^{m \times n}$	System or Measurements matrix.
$\boldsymbol{s} \in \mathbb{R}^{n}$	Signal.
$\boldsymbol{y} \in \mathbb{R}^{m}$	Vector of measurements.
$\boldsymbol{\epsilon} \in \mathbb{R}^{m}$	Noise vector.
$\boldsymbol{x} \in \mathbb{R}^{d}$	Sparse coefficient vector.
$\boldsymbol{\alpha} \in \mathbb{R}^{a}$	Analyzed vector $\boldsymbol{\Omega s}$.
$\boldsymbol{e}_i$	Unit vector with the i-th entry equal to one.
n	Signal dimension.
m	Number of measurements.
d	Number of dictionary atoms.
a	Number of analysis atoms.
M	General manifold.
S^{n-1}	Unit sphere.

$\mathrm{S}(n,d)$	Product of d unit spheres S^{n-1}.
$\mathrm{OB}(n,d)$	$(n \times d)$-dimensional Oblique Manifold.
$\\|\cdot\\|_2$	Euclidean norm.
$\\|\cdot\\|_1$	ℓ_1-norm of vectors and matrices.
$\\|\cdot\\|_p$	ℓ_p-pseudo-norm of vectors and matrices.
$\\|\cdot\\|_0$	ℓ_0-pseudo-norm of vectors and matrices.
$\\|\cdot\\|_F$	Frobenius norm.
$\\|\cdot\\|_\infty$	Infinity norm.
$\mathrm{rk}(\cdot)$	Matrix rank.
$\mathrm{tr}(\cdot)$	Matrix trace.
$\det(\cdot)$	Matrix determinant.
$\mathrm{diag}(\cdot)$	Vector containing the diagonal entries of a square matrix.
$\mathrm{ddiag}(\boldsymbol{F})$	Diagonal matrix with the same diagonal entries as $\boldsymbol{F}$.
$\lfloor\cdot\rfloor$	Round to next smaller integer.
$\lceil\cdot\rceil$	Round to next larger integer.
$\mathrm{sgn}(\cdot)$	Sign function.
$\ln(\cdot)$	Natural logarithm.

Chapter 1

Introduction

One of the core steps in numerous signal processing tasks constitutes to identifying, extracting, or recovering information that is contained in a signal that belongs to some specific class. To achieve this, one can exploit that basically every kind of data or signal carrying information has some underlying inner structure. This inner structure can be understood as a set of rules, which all representatives of a specific signal class follow. Typical examples for signal classes are natural images, medical imagery, or speech data, just to list a few out of countless examples. Now, knowing this inner structure allows composing models of the considered signal class, which in turn can be used to reveal the important information contained in a given sample of the class. Here, a model means nothing more but a set of mathematical relations that a signal is assumed to satisfy, i.e. a model mathematically represents the inner structure of a signal. The success of various applications in signal processing such as compression, interpolation, detection, recognition, or solving inverse problems heavily relies on appropriate data models. A good model should be as simple as possible while being able to represent the signal as accurately as possible [44]. The advantage of simple models has already been expressed through Occam's razor principle that can be traced back to the 14th century, which says that if two models are equally expressive the simpler one should be preferred. By a simple model, I understand one that only requires a few parameters to model a sample of a signal class. In this thesis, parsimonious signal models and their applicability for solving inverse problems in image processing are of major interest and will be covered intensively.

The goal of inverse problems in general is to determine a signal $\boldsymbol{s} \in \mathbb{R}^n$ from some given observations $\boldsymbol{y} \in \mathbb{R}^m$ that might be noisy and/or incomplete. Typical examples of inverse problems include signal denoising, signal deconvolution, or signal completion. To state this problem formally, let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be a known system matrix that models the sampling process, and let $\boldsymbol{\epsilon} \in \mathbb{R}^m$ be some additive noise term. Then the measurement process can be written as

$$y = As + \epsilon. \tag{1.1}$$

Certainly, when $\epsilon = 0_m$ and $m \geq n$, the reconstructed signal $s^\star \in \mathbb{R}^n$ simply can be computed via $s^\star = A^\dagger y$. However, in the presence of noise or when the system is underdetermined (i.e $m < n$) infinitely many solutions to Problem (1.1) exist. Hence, additional prior information about the signal at hand is required to regularize the reconstruction task. This is where signal models can help; when we know that a signal belongs to a certain class for which a model is available, one way to regularize the recovery problem is to search for the signal that is both consistent with the measurements y and at the same time is compatible with the assumed model. Commonly, these two properties can be enforced by finding the signal that minimizes two penalty functions

$$f(s) : \mathbb{R}^n \to \mathbb{R}^+, \tag{1.2}$$

which must be small when the resulting signal is consistent with the measurements and

$$g(s) : \mathbb{R}^n \to \mathbb{R}^+, \tag{1.3}$$

which must be small when the resulting signal fits the model. Function (1.1) depends on the assumed noise statistics whereas Function (1.3) depends on the postulated signal model. This general procedure is depicted in Figure 1.1.

As stated in [42] there has been an evolution of signal models during the last decades that resulted in gradually decreasing modeling errors, which in turn led to gradually increasing performance of applications that exploit those models. Important milestones that have been established throughout this evolution are the Principal Component Analysis (PCA), ℓ_2 minimization based models, weighted ℓ_2 Tikhonov regularization, Mixture of Gaussians models, or Independent Component Analysis (ICA). Currently, we are in the era of sparse and redundant data models with ubiquitous realizations and applications. These models are both theoretically interesting as well as powerful in applications. Consequently, this topic has become well-accepted and today is one of the most important fields of research in the communities of signal processing, computer vision, and machine learning [15]. The following section serves as a brief introduction on this topic, with the focus on the application of solving inverse problems.

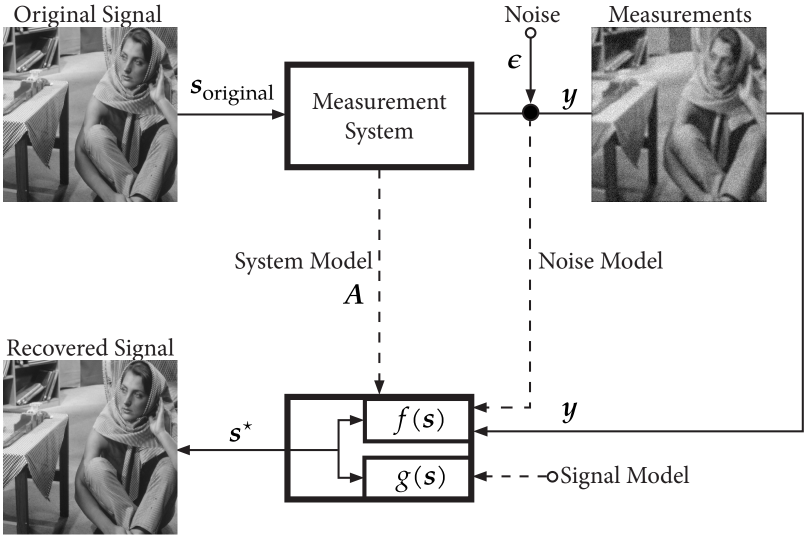
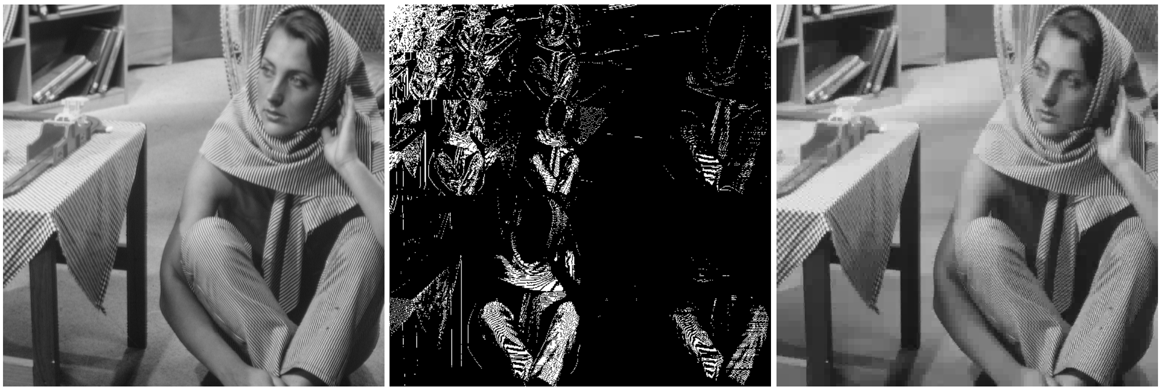


Figure 1.1: This figure depicts the general procedure of inverse problems.

1.1 Background of Sparse and Redundant Data Modeling

A commonality of many natural and informative signals is that elements of the same signal class do not cover the entire surrounding space $\mathbb{R}^n$, but actually reside in a union of much lower dimensional subspaces of dimension k, with $k \ll n$. As an example, consider the class of natural images. A natural image when transformed e.g. into the wavelet domain only contains k coefficients that are large in magnitude while all other $d-k$ coefficients are very small or exactly zero in the ideal case, see Figure 1.2(b). The k large coefficients carry all important information about the image and the corresponding wavelets span the subspace the image resides in. Knowing these coefficients is sufficient to accurately reconstruct the image in the pixel domain by taking the linear combination of the k corresponding wavelets cf. Figure 1.2(c). This fact forms the backbone of the JPEG2000 compression format [122]. Ideally, all possible $\binom{d}{k}$ combinations of wavelets form the union of k-dimensional subspaces where all natural images reside in.

The example mentioned above is just one specific instance of a more general concept, which assumes that for every class of informative signals there exists a set of d suitable building blocks $\{\boldsymbol{d}_i \in \mathbb{R}^n\}_{i=1}^d$, which are called *atomic signals*, *atoms*, or *codewords* that allows to represent each element of the class as a linear combination of only a *few* such atoms. Let the atoms form the columns of a matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \dots, \boldsymbol{d}_d] \in \mathbb{R}^{n \times d}$ called *dictionary*, and let



(a) Ground truth image. **(b)** 10% largest wavelet coefficients of (a). **(c)** Image reconstructed from the wavelet coefficients given in (b).

Figure 1.2: This figure exemplifies the capability of sparse signal representations. The most important information of an image (a) can be accurately recovered (c) using only 10% of the wavelet coefficients (b) of the ground truth image.

$\boldsymbol{x} \in \mathbb{R}^d$ be the vector of transform coefficients, then a signal can be compactly written as

$$\boldsymbol{s} = \sum_{i=1}^{d} \boldsymbol{d}_{:,i} x_i = \boldsymbol{D}\boldsymbol{x}. \tag{1.4}$$

The question now becomes how a unique and meaningful representation $\boldsymbol{x}$ of $\boldsymbol{s}$ can be found. Assuming $\boldsymbol{D}$ to be a full-rank matrix, certainly, for $d = n$ this representation is simply given by $\boldsymbol{x} = \boldsymbol{D}^\dagger \boldsymbol{s}$. However, in the general and more expressive overcomplete or redundant setting $d > n$, the dictionary has a non-trivial nullspace and consequently infinitely many realizations $\boldsymbol{x}$ exist that multiplied by $\boldsymbol{D}$ result in the same signal $\boldsymbol{s}$. Nevertheless, the representation $\boldsymbol{x}$ that reflects the prior assumption that the signal lies in a low-dimensional subspace spanned by a few columns of $\boldsymbol{D}$ is uniquely defined under some mild conditions on $\boldsymbol{D}$ and the dimension of the subspace, cf. [34]. In this case, $\boldsymbol{x}$ only contains $k \ll d$ non-zero coefficients, i.e. the vector is *sparse*, and the k columns of $\boldsymbol{D}$ spanning the subspace where the signal resides are indexed by the *support* of $\boldsymbol{x}$

$$\operatorname{supp}(\boldsymbol{x}) := \{i|\, x_i \neq 0\} \tag{1.5}$$

i.e. the positions of its non-zero entries.

Now, going back to the task of solving inverse problems, this sparsity assumption can be used to regularize the problem by seeking the sparsest vector $\boldsymbol{x}$ that most accurately explains the available measurements $\boldsymbol{y}$. To put this formally, let $g(\boldsymbol{x})$ be some penalty function that enforces $\boldsymbol{x}$ to be sparse, then the signal recovery problem can be tackled by first solving

$$\boldsymbol{x}^\star \in \arg\min_{\boldsymbol{x}} g(\boldsymbol{x}) \quad \text{subject to} \quad f(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{D}\boldsymbol{x}) \leq \epsilon, \tag{1.6}$$

and afterwards computing $\boldsymbol{s}^\star = \boldsymbol{D}\boldsymbol{x}^\star$. Therein, f depends on the assumed noise statistics of the measurements, and $\epsilon \in \mathbb{R}_0^+$ determines how closely $\boldsymbol{A}\boldsymbol{D}\boldsymbol{x}$ has to resemble the measurements. Allowing a certain discrepancy ϵ can be necessary when the signal cannot be exactly represented as a sparse decomposition of atoms but only approximately, or when the measurements are noisy. As the final signal estimate $\boldsymbol{s}^\star$ is synthesized from $\boldsymbol{x}^\star$ via the dictionary $\boldsymbol{D}$, Problem (1.6) is known as the *sparse synthesis model* [45].

The synthesis model has an interesting "fraternal twin" that also relies on a parsimonious signal representation and which is known as the *co-sparse analysis model* [45, 86]. The underlying assumption of this signal model is that there exists an *analysis operator* $\boldsymbol{\Omega} \in \mathbb{R}^{a \times n}$ with $a \geq n$ that maps $\boldsymbol{s}$ into a sparse *analyzed vector*, i.e.

$$\boldsymbol{\alpha} := \boldsymbol{\Omega}\boldsymbol{s} \in \mathbb{R}^a. \tag{1.7}$$

The analysis operator in this model can be interpreted as the counterpart of the dictionary in the synthesis model. The rows of $\boldsymbol{\Omega}$ are the *analysis atoms* and they *analyze* the signal which explains the model's name. In contrast to the synthesis model, where a signal is fully described by the non-zero elements of the sparse code $\boldsymbol{x}$, in the analysis model the zero elements of the analyzed vector $\boldsymbol{\alpha}$ describe the subspace the signal belongs to. Concretely, the signal resides in the orthogonal complement of the subspace spanned by the rows of $\boldsymbol{\Omega}$ that are indexed by the *co-support* of $\boldsymbol{\alpha}$

$$\text{co-supp}(\boldsymbol{\alpha}) := \{i | \, \alpha_i = 0\}, \tag{1.8}$$

i.e. the positions of its zero elements. To further emphasize the difference between the two models, the term *co-sparsity* has been introduced in [85, 86], which denotes the number of *zero* elements of $\boldsymbol{\alpha}$. The co-sparse analysis model can again be exploited as a regularizer for solving inverse problems via

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s}} g(\boldsymbol{\Omega}\boldsymbol{s}) \quad \text{subject to} \quad f(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}) \leq \epsilon. \tag{1.9}$$

The function f and the parameter ϵ have the same meaning as in the synthesis model. A co-sparse representation is enforced in the same way as enforcing sparsity via minimizing an appropriate function g. Note that for the interesting redundant setup, the analysis model is computationally simpler to solve as one only has to optimize over n unknowns, i.e. the dimension of the signal, as opposed to d unknowns in the synthesis model, i.e. the dimension of the sparse code.

As a short summary, in the sparse synthesis model a signal is formed by linearly combining a set of building blocks, i.e. the atoms of a dictionary. Therein the question that arises is, which smallest set of atoms can be chosen to form the signal. Conversely, in the co-sparse analysis model, the true signal is carved out by requiring it to follow a set of constraints, i.e. it must be orthogonal to many analysis atoms. For $d = a = n$, and assuming that dictionaries and analysis operators are non-singular matrices, it is known that both models are identical with $\boldsymbol{\Omega} = \boldsymbol{D}^{-1}$. However, in the interesting redundant setting $d, a \geq n$ the two models differ significantly [45] and have different strengths and weaknesses. While the synthesis model is well-studied both in theory and in applications and is more intuitive to understand due to its generative nature, the analysis model is theoretically more expressive as it exhibits a much larger set of unions of subspaces and is also simpler to optimize due to the smaller number of unknowns. It is still unclear whether one model should be preferred over the other, and both are viable options for e.g. regularizing inverse problems. Most crucial for the performances of both models in applications are:

1. An adequate measure of sparsity together with efficient algorithms for solving the arising optimization problem.

2. The choice of an expressive dictionary or analysis operator, respectively, which allow a maximally sparse and accurate representation of the considered signal class of interest.

In the following section, I provide a brief overview on the state-of-the-art concerning the two issues raised above.

1.2 State-of-the-Art

1.2.1 Sparse Synthesis Model

Finding the sparsest solution to Problem (1.6) is known as sparse coding, sparse approximation, basis selection, or variable selection. Ideally, the ℓ_0-pseudo-norm

$$\|\boldsymbol{x}\|_0 := |\{i|\; x_i \neq 0\}|, \tag{1.10}$$

which counts the non-zero entries of $\boldsymbol{x} \in \mathbb{R}^d$, should be employed as the sparsity measure, i.e. $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_0$. Unfortunately, finding the exact minimum ℓ_0-solution is NP-hard [87] as it requires a combinatorial search whose complexity grows exponentially in d. Besides that, natural signals might not be truly sparse but rather compressible. Compressible means that most of a the energy of a signal is contained in only a few large coefficients and all others are very small in magnitude but not necessarily zero. For those signals, the ℓ_0-pseudo-norm might be a suboptimal penalty function. Throughout this thesis, I will loosely use the term sparse for both truly sparse- and compressible signals. To account for these issues, other sparsifying functions that have similar properties as the ℓ_0-pseudo-norm can be used. The required behavior is that the function does not penalize large coefficients heavily, while it pushes small coefficients towards zero. Mathematically, this is accomplished by a function whose derivative is large around zero and is small or even vanishes for large values. Three commonly applied sparsifying functions having these properties and the ℓ_1-norm, which is the most prominent sparsity inducing function as it is the closest convex surrogate of the ℓ_0-pseudo-norm, are shown in Figure 1.3.

Now, one straightforward way to find a sparse solution is to use a smooth differentiable sparsity measure and utilize standard optimization solvers. However, theses methods neglect the structure that underlies the sparse coding problem and are therefore computationally inefficient. That is why a large number of efficient techniques have been proposed that are explicitly designed to solve the sparse coding task. They can be roughly categorized into (i) greedy pursuit methods like Orthogonal Matching Pursuit (OMP) [96] that find an approximate, or under some side conditions even exact minimum ℓ_0 solution,(ii) methods like Least Angle Regression (LARS) [39] based on minimizing the ℓ_1-norm which is the closest convex relaxation of the ℓ_0-pseudo-norm, and (iii) methods that employ non-convex ℓ_0-surrogates like ℓ_p-pseudo-norms with $p < 1$, e.g. the FOCal Underdetermined System Solver (FOCUSS) [61, 100]. In Section 2.1.1, I provide a more detailed overview and explanations of various sparse solvers.

The success of all sparse coding techniques most crucially depends on the choice of the dictionary $\boldsymbol{D}$. Generally, one can choose a dictionary from two major classes: (i) analytic dictionaries and (ii) learned dictionaries. Analytic dictionaries like the Discrete Cosine Transform (DCT) or diverse Wavelet Transforms [82] are built on mathematical models of the type of signal they should represent. They offer fast implementations and are applicable to a large set of signals. This generality of being able to represent many signals in a good way comes at the cost of not necessarily representing specific signals optimally. More expressive dictionaries with fine tuned atoms can be obtained by learning the structure, which

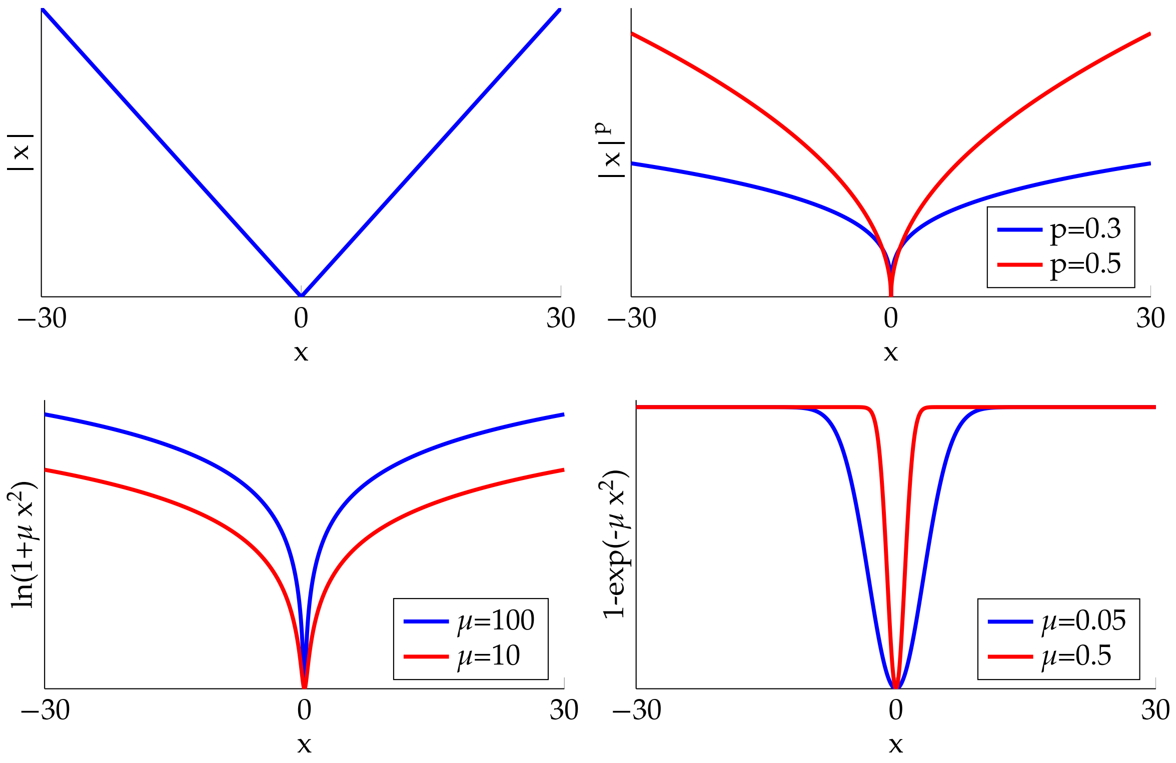
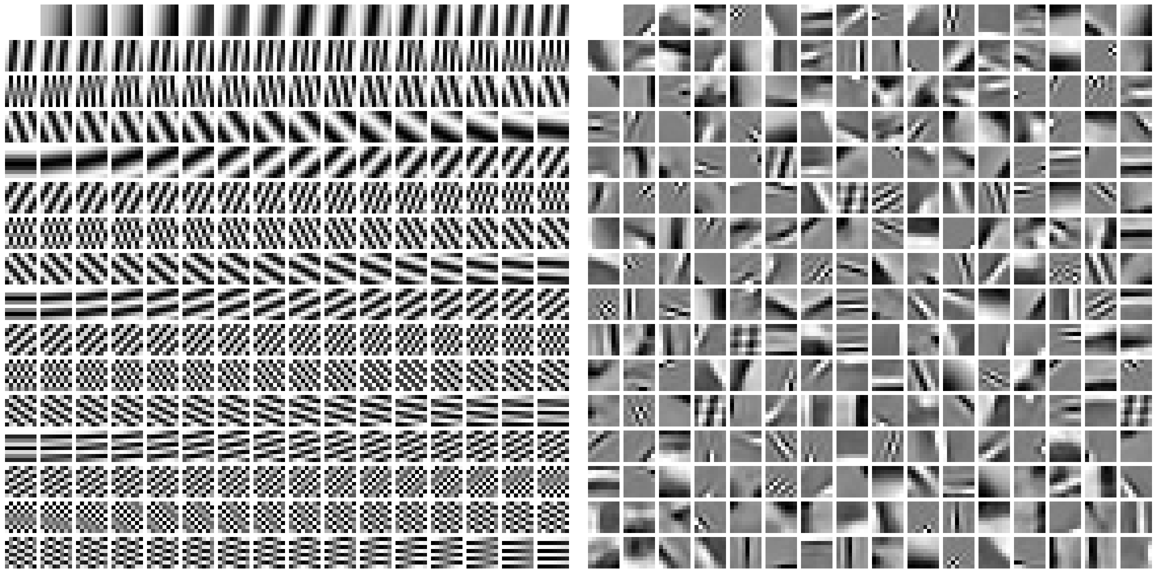


Figure 1.3: This figure shows four commonly applied sparsifying functions with different parameter settings.

underlies the considered signal class directly from a set of representative training signals. Figure 1.6 shows both the atoms of an analytically defined dictionary and the atoms of a learned dictionary. Compared to the learned atoms, the analytic ones show a more regular structure.

Dictionary learning algorithms aim at finding the dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ that allows to describe a set of M training samples $\{\boldsymbol{s}_i \in \mathbb{R}^n\}_{i=1}^M$ as closely as possible with the sparsest possible representations $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^M$. Formally, let $\boldsymbol{S} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_M] \in \mathbb{R}^{n \times M}$ be a matrix containing the M training samples arranged as its columns, and let $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_M] \in \mathbb{R}^{d \times M}$ be a matrix that contains the corresponding sparse representations. Then the dictionary learning process can be stated as

$$\{\boldsymbol{D}^\star, \boldsymbol{X}^\star\} \in \arg\min_{\boldsymbol{D},\boldsymbol{X}} G(\boldsymbol{X}) \quad \text{subject to} \quad \begin{aligned} &F(\boldsymbol{S} - \boldsymbol{D}\boldsymbol{X}) \leq \epsilon, \\ &\boldsymbol{D} \in \mathfrak{C}. \end{aligned} \tag{1.11}$$



(a) Atoms of the overcomplete discrete cosine transform.

(b) Atoms learned by SeDiL.

Figure 1.4: This figure shows the 256 atoms of dimension $n = 64$ of the analytic overcomplete discrete cosine transform (a) and of a dictionary learned by the algorithm SeDiL (b), which is introduced in Chapter 4. Each atom is represented as an (8×8) dimensional patch, where a black pixel corresponds to the smallest negative entry, a gray pixel is a zero entry, and a white pixel corresponds to the largest positive entry.

Therein, $G : \mathbb{R}^{d \times M} \to \mathbb{R}^+$ measures the overall sparsity of the training set, $F : \mathbb{R}^{n \times M} \to \mathbb{R}^+$ measures how closely each sample is represented by its sparse code, ϵ reflects the assumed noise energy, and $\mathfrak{C}$ is some predefined admissible set of solutions. Restricting possible solutions to an admissible set like matrices with normalized columns is necessary to avoid the scale ambiguity problem, i.e. obtaining entries of $\boldsymbol{D}$ that tend to infinity, while the entries of $\boldsymbol{X}$ tend to zero, as this is clearly the sparsest possible solution to Problem (1.11). Furthermore, this constraint set can be used to enforce desired properties on the dictionary such as bounded self coherence, or bounded coherence to other matrices. In simple words, the self coherence of $\boldsymbol{D}$ measures the similarity between the dictionary's atoms and is an important criterion for theoretically analysis of sparse coding techniques. I want to emphasize here that to learn a dictionary one has to optimize over both the sparse code $\boldsymbol{X}$ and the dictionary $\boldsymbol{D}$. The probably best known dictionary learning algorithms are the method of Olshausen and Field [90], the Method of Optimal Directions (MOD) [46], and K-SVD [43]. Roughly speaking, those techniques are based on alternating between fixing $\boldsymbol{D}$ and updating $\boldsymbol{X}$ with some sparse coding algorithm followed by fixing $\boldsymbol{X}$ and updating $\boldsymbol{D}$, see Figure 1.5. They

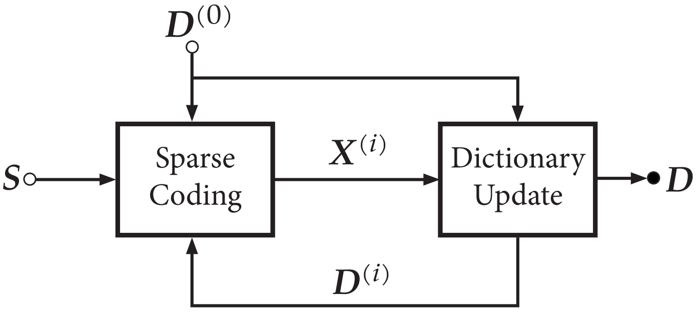


Figure 1.5: This figure depicts the two step procedure commonly followed by dictionary learning algorithms.

mainly differ in the employed constraint set and in the way the dictionary is updated. A detailed introduction on this topic is given in Section 2.1.2.

1.2.2 Co-Sparse Analysis Model

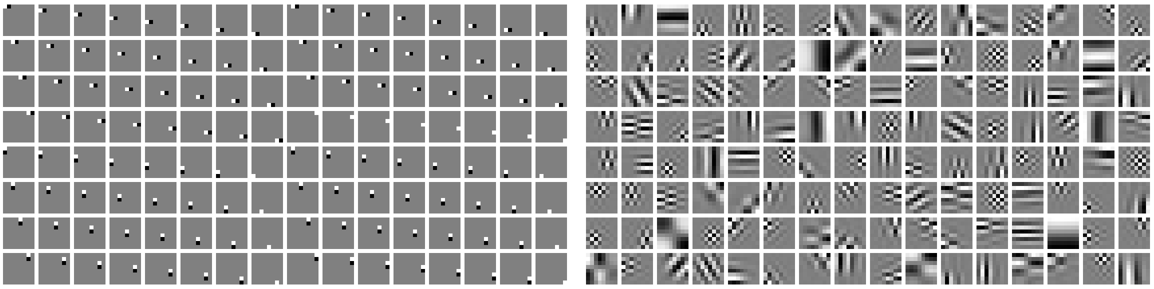
Recall that the goal of the co-sparse analysis model is to determine the signal $\boldsymbol{s}$ such that the analyzed vector $\boldsymbol{\alpha} = \boldsymbol{\Omega s}$ with respect to a given analysis operator is as sparse as possible. Certainly, when $\boldsymbol{s}$ is free of noise one of the analysis model's advantages is that a signal's sparse analyzed version is straightforwardly computed by $\boldsymbol{\alpha} = \boldsymbol{\Omega s}$, without having to solve an optimization problem. However, if noisy measurements $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{\epsilon}$ or incomplete measurements with $m < n$ are given, finding the signal that results in the sparsest vector $\boldsymbol{\alpha}$ is no longer trivial and requires to solve the co-sparse analysis optimization problem (1.9). Therefore, the same penalty functions as introduced above for the synthesis model can be used to enforce the solution to be sparse. Depending on the choice of g, the arising optimization problem can be tackled via general purpose solvers like the reweighted ℓ_1 algorithm [18] or standard first-order methods. In contrast to the synthesis sparse coding problem for which many specialized solvers are available, only a few specialized co-sparse analysis solvers such as the greedy-like methods introduced in [58] exist, which I review in more depth in Section 2.2.1.

Similar to choosing the dictionary in the synthesis model, one has the options to either select an analytically defined analysis operator or to learn an operator that is better adapted to the signal class of interest. The probably best known and most widely applied analytic analysis operator is the finite difference operator, which approximates first-order derivatives. This operator is also used for computing the famous Total Variation (TV)-norm [111] in image processing. Other analytic operators are the transposed of diverse tight frames, which are also used as dictionaries, such as the overcomplete cosine transform (ODCT).

Learning an analysis operator is conceptually similar to learning a dictionary. Concretely, let $\boldsymbol{S} \in \mathbb{R}^{n \times M}$ again denote the matrix of M training samples, then the problem is to find the solution to

$$\boldsymbol{\Omega} \in \arg\min_{\boldsymbol{\Omega}} G(\boldsymbol{\Omega}\boldsymbol{S}) \quad \text{subject to} \quad \boldsymbol{\Omega} \in \mathfrak{C}. \tag{1.12}$$

Therein, the function $G : \mathbb{R}^{a \times M} \to \mathbb{R}$ again measures the overall sparsity of the analyzed training set and $\mathfrak{C}$ is some predefined admissible set of analysis operators that is required to avoid the trivial solution $\boldsymbol{\Omega} = \boldsymbol{0}_{a \times n}$. Note that the analysis operator learning problem in the noiseless setting, i.e. when ideal training samples are considered, only requires to optimize over the operator, which is much easier compared to the dictionary learning problem where both the dictionary and the sparse code have to be determined simultaneously. In contrast to the large amount of existing dictionary learning algorithms, the number of analysis operator learning methods is rather small. Field-of-Experts [107] and Analysis K-SVD [109] are two prominent examples, and more approaches are introduced in detail in Section 2.2.2. A comparison of analytically defined analysis atoms and analysis atoms learned from example signals is presented in Figure 1.6. Again as for the synthesis case, the analytic analysis atoms show a more regular structure as compared to the learned analysis atoms.



(a) Analysis atoms of the finite difference operator.

(b) Analysis atoms learned by GOAL.

Figure 1.6: This figure shows the 128 analysis atoms of dimension $n = 64$ of the analytic finite difference operator (a) and of an analysis operator learned by the algorithm GOAL (b), which is introduced in Chapter 5. Each analysis atom is represented as an (8×8) dimensional patch, where a black pixel corresponds to the smallest negative entry, a gray pixel is a zero entry, and a white pixel corresponds to the largest positive entry.

1.3 Formulation of the Research Problem

The large number of both well-established and newly published works that deal with the problem of learning sparse signal models and their applications in diverse signal processing tasks is only one indicator for the timeliness, vitality, and importance of this topic. In this thesis, I investigate both problems regarding dictionary learning and analysis operator learning and concretely aim at solving the following issues.

1.3.1 Dictionary Learning

1. Most dictionary learning techniques produce unstructured matrices whose possible dimensions are inherently limited by available computational resources and that are rather expensive to be used in applications. In contrast, many analytic dictionaries like the ODCT can be applied very efficiently to higher dimensional signals due to having a separable matrix structure. Motivated by this, I want to find a way to learn dictionaries that have a separable structure such that they combine the advantages of being well-tuned to the signal of interest as well as offering a computationally efficient implementation.

2. The applicability of a dictionary depends on several internal properties like its condition number or its coherence to itself or to other matrices. The self coherence can be simply understood as a measure of how similar a dictionary's atoms are. For example, employing sparse coding techniques to solve inverse problems is only guaranteed to succeed when incoherent dictionaries are used. For this reason, I want to investigate how such internal properties can be controlled directly during the learning process.

3. Many dictionary learning techniques are based on a suboptimal optimization procedure that alternates between optimizing the sparse code while fixing the dictionary and updating the dictionary while fixing the sparse code, see Figure 1.1. Furthermore, the dictionary update often only aims at increasing the data fidelity and neglecting the constraint set. Consequently, a subsequent projection on the admissible set is required to obtain a feasible solution. Here, I want to jointly learn both the dictionary and the sparse codes, with the dictionary being updated such that it always remains feasible, i.e. within the set of constraints.

1.3.2 Analysis Operator Learning

1. Compared to the matured topic of dictionary learning, the problem of learning an analysis operator that is adapted to a signal class of interest is still in its infancy, and only few algorithms are available. Therefore, I want to develop a novel analysis operator learning method that is independent of a specific signal class and thus universally

applicable. The algorithm should handle large training sets efficiently and should avoid overfitting the operator to subsets of the training data. Crucial internal parameters of the operator like its mutual coherence and its condition number should be controllable during the learning process.

2. One interesting application for employing learned overcomplete analysis operators is regularizing inverse problems for image reconstruction. This problem is rather new, and it is yet unclear how the above mentioned internal properties of an analysis operator influence the reconstruction quality. In this work, I want to determine empirically an answer to this question.

3. In fields like robotics, diverse sensors are often employed that observe the same physical object but acquire measurements in different modalities. As those measurements originate from the same object, it seems likely that they are statistically dependent. How to model these dependencies between the modalities and how applications could benefit from such a modeling, are interesting and important questions. Here, I want to investigate whether the co-sparse analysis model is a valuable approach for modeling these dependencies. As an application, I consider the enhancement of corresponding intensity images and depth maps representing the identical three dimensional scenes.

1.4 Contributions

To solve the issues raised in Section 1.3, in the first part of my thesis I introduce two new algorithms called Separable Dictionary Learning (SeDiL) and Geometric Analysis Operator Learning (GOAL) that are based on geometric conjugate gradient optimization on suitable manifolds. SeDiL works on the product of unit spheres and allows learning both unstructured conventional dictionaries as well as dictionaries, which have a separable matrix structure. Due to the optimization over the product of spheres no projection onto the admissible set of solutions is required and the scale ambiguity problem is inherently avoided. It jointly optimizes over both the dictionary and the sparse code and permits to control the crucial mutual coherence of the learned dictionary. To enforce sparsity, I employ a smooth non-convex ℓ_0 surrogate. As non-convex optimization is prone to get trapped in local minima, I employ a non-monotone line search technique, which empirically showed to reduce this problem.

The proposed analysis operator learning approach GOAL works on the oblique manifold, i.e. the set of full-rank matrices with normalized columns. The algorithm controls the condition number and the mutual coherence of a learned operator and inherently avoids the trivial solution. A smooth non-convex ℓ_0 surrogate is used as the sparsifying function. To avoid overfitting the operator to specific subsets of the training data, I suggest to minimize both the empirical mean and the empirical variance of the sparsity measure of the analyzed training set. This is simply achieved by considering the square of the sparsity promoting function of each sample. In synthetic tests, I show that my method outperforms all existing analysis operator learning techniques in terms of required computation time, accuracy in finding a generating ground truth operator, and generality. To give an answer towards the question which properties an analysis operator should have to be valuable for regularizing inverse problems in imaging, I perform a large image denoising test. From this I conclude that a well suited operator should have moderate mutual coherence and be well conditioned.

In the second part, I introduce the novel multimodal co-sparse analysis model that permits to model statistical dependencies of various modalities representing the same physical object. This model suggests that measurements acquired in different modalities originating from the same scene share a common co-support, when a suitable set of analysis operators is used. For this set of operators, no analytic form exists and it must be learned from aligned example signals. Therefore, I propose an extension of GOAL that uses a suitable sparsifying function enforcing the coupled co-support assumption. The performance of the proposed model is evaluated for the task of depth map superresolution.

1.5 Thesis Outline

The main chapters of this thesis are partially based on a number of peer reviewed publications for which the references are provided right at the beginning of the respective chapter.

The remainder of this thesis is organized as follows. In Chapter 2 the current state-of-the-art regarding both the sparse synthesis model and the co-sparse analysis model is reviewed. After that, in Chapter 3 the general idea behind optimization on matrix manifolds and the specific conjugate gradient method are explained. Chapter 4 introduces the new dictionary learning method SeDiL and evaluates how dictionaries learned by SeDiL perform in image processing tasks. In Chapter 5, the novel analysis operator learning approach GOAL is introduced. Both synthetic and real world experiments are presented that compare GOAL to the state-of-the-art and show the performance of an operator learned by using GOAL for solving the three classical image processing problems of denoising, inpainting, and superresolution. In Chapter 6 the multimodal co-sparse analysis model is introduced, and its performance is evaluated experimentally for the task of depth map upsampling with the help of an aligned intensity image.

Chapter 2

State-of-the-Art on Sparse Data Modeling

This chapter reviews the current state-of-the-art on sparse data modeling considering both the sparse synthesis model as well as the co-sparse analysis model. For both models, I first review existing algorithm for enforcing the respective model assumption, and second methods that aim at learning an appropriate dictionary and analysis operator, respectively.

2.1 Sparse Synthesis Model

2.1.1 Sparse Coding

Solving the sparse coding problem (1.6) is an important and heavily researched task for which a huge number of algorithms have been proposed already. The most prominent techniques can be roughly divided into (i) greedy pursuit methods, (ii) methods based on convex relaxations of the ℓ_0-pseudo-norm, and (iii) methods that employ non-convex ℓ_0-surrogates. In the following, I shortly explain the concepts underlying those three classes of solvers each with the help of at least one representative example algorithm. I selected the respective algorithms due to their importance and common application in dictionary learning methods, which are covered in the subsequent subsection. As stated above, the amount of existing sparse coding algorithms is immense and too large to explain every single method in detail. Nevertheless, for the interested reader and to give credit to other researches, I provide pointers to the literature for further important representative algorithms of each solver class.

To fix notations, in the following a quadratic error term is assumed, i.e. $f(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, which is also in accordance to the common independent and identically distributed (i.i.d.) Gaussian noise assumption in the literature. The support of $\boldsymbol{x}$, i.e. the indices of its non-zero entries, is denoted by $\mathcal{I} := \operatorname{supp}(\boldsymbol{x})$. Furthermore, without loss of generality the columns of $\boldsymbol{D}$ are assumed to have unit Euclidean norm. With the shorthand notation $\boldsymbol{Z} := \boldsymbol{A}\boldsymbol{D} \in \mathbb{R}^{m \times d}$, I will consider three equivalent formulations for the sparse coding problem, which read as

$$x^\star \in \arg\min_{x} g(x) \quad \text{subject to} \quad \tfrac{1}{2}\|y - Zx\|_2^2 \leq \epsilon, \tag{2.1}$$

$$x^\star \in \arg\min_{x} \tfrac{1}{2}\|y - Zx\|_2^2 \quad \text{subject to} \quad g(x) \leq s, \tag{2.2}$$

$$x^\star \in \arg\min_{x} \tfrac{1}{2}\|y - Zx\|_2^2 + \lambda g(x). \tag{2.3}$$

In Equation (2.1) an upper bound $\epsilon \geq 0$ on the discrepancy between the measurements and the reconstructions is employed, in (2.2) the sparsity is upper bounded by $s > 0$, while Equation (2.3) is the unconstrained Lagrangian form of the two former problems with $\lambda > 0$ being the Lagrange multiplier that weighs between fidelity to the measurements and sparsity of the solution. Note that for an appropriate choice of ϵ, s, and λ the solutions of Problem (2.1)-(2.3) coincide.

Greedy Pursuit Methods

Greedy Pursuit Methods employ $g(x) = \|x\|_0$, which in general is NP-Hard [87], and aim at finding and approximate solution to Problem (2.1) or (2.2). Roughly speaking, these methods start with $x^{(0)} = \mathbf{0}_d$, i.e. an empty support $\mathcal{J}^{(0)} = \emptyset$, and sequentially add elements to the support of x, i.e. increasing its ℓ_0-pseudo-norm such that the error between the measurements and the current sparse approximation $\|y - Zx^{(i)}\|_2^2$ is decreased. One of the earliest and best known pursuit methods is the Orthogonal Matching Pursuit (OMP) algorithm [96], which is based on the following procedure. At the i-th iteration, it first computes the residual $r^{(i)} = y - Zx^{(i-1)}$ and then finds the column of Z that is most strongly correlated with $r^{(i)}$, i.e.

$$k^{(i)} = \arg\ \max_{k} |r^{(i)\top} z_{:,k}|. \tag{2.4}$$

This column index is then added to the support of x, i.e. $\mathcal{J}^{(i)} = \mathcal{J}^{(i-1)} \cup \{k^{(i)}\}$. In the second step, the weights of the support $x_{\mathcal{J}^{(i)}}$ are updated by projecting y orthogonally onto the columns of Z that are index by $\mathcal{J}^{(i)}$, i.e. $x_{\mathcal{J}^{(i)}} = Z^\dagger_{:,\mathcal{J}^{(i)}} y$. In this way, a new non-zero entry is added at each iteration and all weights are updated accordingly. Depending on the considered formulation the algorithm stops when the norm of the residual falls below a threshold $\|r^{(i)}\|_2 \leq \epsilon_1$, i.e. Formulation (2.1), a certain number s of non-zeros of x has been determined, i.e. Formulation (2.2), or the maximum correlation between the residual and any column lies below a threshold $\|Zr^{(i)}\|_\infty \leq \epsilon_2$. Computationally efficient implementations of OMP exist that utilize the QR-factorization [29] or the Cholesky-factorization [22] of $\boldsymbol{Z}$ to avoid having to explicitly compute its pseudoinverse.

Algorithms closely related to OMP are Matching Pursuit (MP) [83], Gradient Pursuits [13], and the Optimized OMP (OOMP) [102], which all differ in the way the support is updated. One drawback common to all these methods is that only one coefficient is added to the support at each iteration. Consequently, these methods are only computationally efficient for very sparse signals but perform rather badly when the sparsity of the signal increases due to an increased number of necessary iterations. To overcome this, greedy-like algorithms such as Stagewise Orthogonal Matching Pursuit (StOMP) [37] and Regularized Orthogonal Matching Pursuit (ROMP) [89] have been proposed that add multiple coefficients per iteration to the support of the signal. Even better performance is achieved by methods such as Compressive Sampling Matching Pursuit (CoSaMP) [88], Subspace Pursuit (SP) [25], Hard Thresholding Pursuit (HTP) [51], and Iterative Hard Thresholding (IHT) [14] that do not only update the support by adding multiple coefficients but also allow removing elements found at previous iterations.

Convex Relaxations

Convex relaxation approaches tackle the sparse coding problem by exchanging the ℓ_0-pseudo-norm with the ℓ_1-norm

$$\|\boldsymbol{x}\|_1 := \sum_i |x_i|, \tag{2.5}$$

which is its closest convex surrogate. With this and $\epsilon = 0$, Problem (2.1) in signal processing is known as Basis Pursuit (BP) while for $\epsilon > 0$ it is called Basis Pursuit Denoising (BPDN) [20]. Formulated as in Equation (2.2), from the statistical machine learning community it is known as the Least Absolute Shrinkage and Selection Operator (LASSO) [123]. One nice property about employing $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ is that the sparse coding problem becomes a Linear Program (LP), for which the globally optimal solution can be found using generic LP-solvers like interior point methods [20, 68]. However, such generic solvers are rather slow, computationally demanding, and do not always scale well to large scale problems. Due to this, several approximate ℓ_1-solvers exist that exploit the specific structure of the ℓ_1-norm to efficiently determine a solution.

Very prominent examples belonging to this class of solvers are the Least Angle Regression (LARS) [39] algorithm and its Homotopy versions LARS-LASSO [39, 94] and LARS-EN [147]. These methods solve Problem (2.3) for a sequence of decreasing values of λ and provide the full regularization path, i.e. all solutions $\boldsymbol{x}$ that correspond to all possible values of λ. Similar to OMP, these algorithms start from $\boldsymbol{x}^{(0)} = \boldsymbol{0}_d$ with the Lagrange multiplier initialized to $\lambda^{(0)} = \|\boldsymbol{Z}^\top \boldsymbol{y}\|_\infty$ and sequentially update the support of $\boldsymbol{x}$ element by element, with a new index being identified as given in Equation (2.4). The difference to OMP lies in the way the associated weights are computed. While OMP computes the weights such that the columns of $\boldsymbol{Z}_{:,\mathcal{J}^{(i)}}$ are maximally uncorrelated with the residual, LARS based methods compute them such that all columns of $\boldsymbol{Z}_{:,\mathcal{J}^{(i)}}$ are *equally* correlated with the residual. This amounts to solving a linearly-penalized least-squares problem rather than a regular least-squares problem as in OMP. In contrast to LARS, LARS-LASSO and LARS-EN do not only add elements to the support but also allow removing elements that have been added at previous iterations. As the number of required iterations is equal to the number of non-zeros of $\boldsymbol{x}$, removing elements might slow down the convergence of the algorithms; however, in practice elements are rarely removed. At each iteration, the new value $\lambda^{(i+1)}$, which either leads to adding an element to or removing it from the support of $\boldsymbol{x}$ can be computed in closed form. For an in-depth discussion of the relation between LARS, LARS-LASSO, and OMP see [36].

Another important class of approximate ℓ_1-solvers is the class of Iterative Shrinkage Thresholding (IST) algorithms as proposed for example in [28, 40, 49, 54]. They are among the most widely applied methods especially for large scale optimization problems as they only require a few matrix vector operations together with a shrinkage or soft thresholding step to solve the sparse coding problem. In the shrinkage step, all entries of a vector smaller than a threshold are set to zero, while all other entries are shrinked towards zero, i.e.

$$(\operatorname{shrink}_\lambda(\boldsymbol{x}))_i := \begin{cases} x_i - \operatorname{sgn}(x_i)\lambda & \text{if} \quad |x_i| > \lambda \\ 0 & \text{otherwise} \end{cases} . \tag{2.6}$$

In their most general formulation, IST algorithms start with $\boldsymbol{x}^{(0)} = \boldsymbol{0}_d$ and determine the solution to Problem (2.3) by iterating the update step $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + t(\operatorname{shrink}_\lambda(\boldsymbol{x}^{(i)} + \boldsymbol{Z}^\top(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{x}^{(i)})) - \boldsymbol{x}^{(i)})$. Therein, $t \in \mathbb{R}_0^+$ is the step size that can be set fixed or chosen by e.g. a line search approach. As standard IST algorithms are known to converge slowly, techniques that build upon IST have been suggested that aim at enhancing the performance while keeping its simplicity. Examples include Two Step Iterative Shrinkage Thresholding (TwIST) [12], Fast Iterative Shrinkage Thresholding (FISTA) [7], or the split Bregman method [60]. For a nice an broad overview of IST based algorithms and their theoretical properties, see [146].

Last, I want to note that many more first-order ℓ_1-solvers applicable for large scale problems exist such as Gradient Projection for Sparse Recovery (GPSR) [50], Nesterov's algorithm (NESTA) [8], and Sparse Reconstruction by Separable Approximation (SpaRSA) [132], which should be interpreted as a framework for sparse coding rather than a concrete algorithm. For a broad introduction to the topic of sparse coding by convex optimization, I refer the interested reader to [5].

Non-Convex Relaxations

Non-convex sparse coding techniques enforce sparsity by employing non-convex ℓ_0 surrogates such as the ℓ_p-pseudo-norm

$$\|\boldsymbol{x}\|_p^p := \sum_i |x_i|^p, \tag{2.7}$$

with $p < 1$. Compared to the ℓ_1-norm that penalizes larger coefficients more heavily than smaller ones, ℓ_p-pseudo-norms more closely resemble the democratic ℓ_0-pseudo-norm, as both large and small coefficients are penalized more equally. On the downside, these functions are more difficult to optimize and suffer from getting stuck in local minima that might be far away from the global optimal solution. One prominent non-convex sparse coding approach is the FOCal Underdetermined System Solver (FOCUSS) [61, 100], which is closely related to iteratively reweighted least squares. In its basic implementation, it uses $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_p^p$ and assumes $\epsilon = 0$. To derive the algorithm, Problem (2.1) is first reformulated in Lagrangian form

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) := \|\boldsymbol{x}\|_p^p + \boldsymbol{\lambda}^\top(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{x}), \tag{2.8}$$

with $\boldsymbol{\lambda} \in \mathbb{R}^m$ being a vector of Lagrange multipliers. Now, a necessary condition for $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ to be a minimum or a stationary point of Equation (2.8), is that the gradient of (2.8) with respect to $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ at $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star)$ has to vanish, i.e.

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) := p\boldsymbol{W}_{\boldsymbol{x}^\star}\boldsymbol{x}^\star - \boldsymbol{Z}^\top\boldsymbol{\lambda}^\star = \boldsymbol{0}_d, \qquad \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) := \boldsymbol{y} - \boldsymbol{Z}\boldsymbol{x}^\star = \boldsymbol{0}_m, \tag{2.9}$$

with $\boldsymbol{W}_{\boldsymbol{x}^\star} = \operatorname{diag}^{-1}([|x_1^\star|^{p-2}, \dots, |x_d^\star|^{p-2}]) \in \mathbb{R}^{d \times d}$, where $\operatorname{diag}^{-1} : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ forms a diagonal matrix with the elements of the input vector on the diagonal. This matrix simply arises from the gradient of Equation (2.7), where the i-th component is concretely given as $(\frac{\partial}{\partial x}\|\boldsymbol{x}\|_p^p)_i = p|x_i|^{p-1}\operatorname{sgn}(x_i) = p|x_i|^{p-2}x_i$. Now, eliminating $\boldsymbol{\lambda}^\star$ from (2.9) and solving for $\boldsymbol{x}^\star$ results in $\boldsymbol{x}^\star = \boldsymbol{W}_{\boldsymbol{x}^\star}^{-1}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{W}_{\boldsymbol{x}^\star}^{-1}\boldsymbol{Z}^\top)^{-1}\boldsymbol{y}$, which is the necessary condition a stationary point has to fulfill. Based on this, FOCUSS starts from some initial solution, e.g. the minimum

ℓ_2-norm solution $\boldsymbol{x}^{(0)} = \boldsymbol{Z}^{\dagger}\boldsymbol{y}$, and iteratively updates it via

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{W}_{\boldsymbol{x}^{(i)}}^{-1}\boldsymbol{Z}^{\top}(\boldsymbol{Z}\boldsymbol{W}_{\boldsymbol{x}^{(i)}}^{-1}\boldsymbol{Z}^{\top})^{-1}\boldsymbol{y}, \tag{2.10}$$

until some user defined convergence criterion is met. The algorithm can be easily extended to deal with noisy measurements, i.e. $\epsilon > 0$, by changing the update formula to

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{W}_{\boldsymbol{x}^{(i)}}^{-1}\boldsymbol{Z}^{\top}(\mu\boldsymbol{I}_m + \boldsymbol{Z}\boldsymbol{W}_{\boldsymbol{x}^{(i)}}^{-1}\boldsymbol{Z}^{\top})^{-1}\boldsymbol{y}, \tag{2.11}$$

with $\mu \in \mathbb{R}^+$ being proportional to the assumed noise power ϵ. Further extensions exist that employ other sparsifying functions like Gaussian and Shannon entropy diversity measures or the ℓ_0-pseudo-norm.

The reweighted ℓ_1-minimization technique proposed in [18] is an iterative algorithm similar to FOCUSS. Each iteration consist of solving a weighted ℓ_1-minimization problem, with the required weights being computed from the sparse coefficients determined at the previous iteration. Each weighted ℓ_1-minimization problem can be tackled by any ℓ_1-solver. Note that this general purpose technique can also be applied for solving the analysis sparse coding problem as introduced below in Section 2.2.1.

The success of finding a sparse representation of a signal most severely depends on the choice of the dictionary $\boldsymbol{D}$, regardless which sparse coding technique is applied, as the dictionary determines all possible subspaces a signal can reside in. Generally, one can choose a dictionary from two major classes: (i) analytic dictionaries and (ii) dictionaries learned from example signals. Analytic dictionaries are built on mathematical models of the type of signal they should represent. Most prominent examples include the Discrete Cosine Transform (DCT), diverse Wavelet Transforms [82], Wedgelets [33], Curvelets [142], or Contourlets [31]. Mostly, these dictionaries are either orthogonal, bi-orthogonal, or tight-frames, which in a noiseless setup allow to compute the sparse code by a simple matrix vector multiplication. They have the advantages of being theoretically sound and of offering a fast implementation that avoids explicit matrix vector multiplications in large scale settings. Due to these properties, analytic dictionaries form the backbone of modern data compression algorithms. However, their expressive capabilities and their adaptivity to more specific data is too limited to be used in mid- and high-level signal analysis and processing tasks. First approaches towards finding more expressive sparse data representations aimed at adapting the atoms of analytic dictionaries to specific signal classes. Representatives following this approach are Wavelet Packages [21], Steerable Wavelets [119], or Bandlets [72]. However, these models are still too generic to handle specific subsets of signal classes in an optimal way. Better dictionaries with finer tuned atoms can be obtained by learning the structure, which underlies the considered signal class directly from example signals belonging to the class. This task known as *dictionary learning* accounts for one of the two major topics of my thesis, and the next subsection gives an overview of state-of-the-art methods of that area.

2.1.2 Dictionary Learning

Given a set of M training samples $\{\boldsymbol{s}_i \in \mathbb{R}^n\}_{i=1}^M$ that are drawn from the signal class of interest, dictionary learning algorithms aim at finding the dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ that permits to describe all M samples as closely as possible with the sparsest possible representations $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^M$. Formally, let $\boldsymbol{S} := [\boldsymbol{s}_1, \dots, \boldsymbol{s}_M] \in \mathbb{R}^{n \times M}$ be a matrix containing the M training samples arranged as its columns, and let $\boldsymbol{X} := [\boldsymbol{x}_1, \dots, \boldsymbol{x}_M] \in \mathbb{R}^{d \times M}$ be a matrix containing the corresponding M sparse representations, then the dictionary learning process can be stated as

$$\{\boldsymbol{D}^\star, \boldsymbol{X}^\star\} \in \arg\min_{\boldsymbol{D},\boldsymbol{X}} G(\boldsymbol{X}) \quad \text{subject to} \quad \begin{aligned} &F(\boldsymbol{S} - \boldsymbol{D}\boldsymbol{X}) \leq \epsilon, \\ &\boldsymbol{D} \in \mathfrak{C}. \end{aligned} \tag{2.12}$$

Therein, the function $G : \mathbb{R}^{d \times M} \to \mathbb{R}^+$ measures the overall sparsity of the training set, $F : \mathbb{R}^{n \times M} \to \mathbb{R}^+$ measures how closely each sample is represented by its sparse code, ϵ reflects the assumed noise energy, and $\mathfrak{C}$ is some predefined admissible set of solutions. Restricting possible solutions to an admissible set is necessary to avoid the scale ambiguity problem, i.e. getting entries of $\boldsymbol{D}$ that tend to infinity, while the entries of $\boldsymbol{X}$ tend to zero, which is clearly the sparsest possible solution. Furthermore, a constraint set can be used to enforce desired internal structures or properties on the dictionary such as bounded self coherence, or coherence to other matrices. Such properties are important for both theoretical analyses of sparse coding techniques and for the success of sparsity exploiting applications. In the following, I provide explanations of the most influential achievements in the field of dictionary learning.

Maximum Likelihood-Based Dictionary Learning

The first dictionary learning approach was reported in the seminal work of Olshausen and Field in 1996 [90] and has been further extended in [91]. Their goal was to find a set of image filters, i.e. dictionary atoms that have similar properties as the receptive fields of simple cells found in the brain's primary visual cortex. Receptive fields are spatially localized i.e. their support is limited, they are oriented, and they are bandpass which means that they are selective to structure at different spatial scales. The authors claim that atoms, which permit to represent image-patches as a sparse and statistically independent linear combination, exactly account for these properties.

To infer these atoms from example data, the authors propose a probabilistic framework that aims at fitting the distribution of all possible realizations that emerge from the hypothesized model $P(\boldsymbol{s}|\boldsymbol{D})$, as closely as possible to the true distribution of all signals belonging to the considered signal class $P(\boldsymbol{s})$. The accuracy of this fitting can be assessed via the Kullback-Leibler divergence between the two distributions

$$KL\Big(P(\boldsymbol{s}) \parallel P(\boldsymbol{s}|\boldsymbol{D})\Big) = \sum_i P(\boldsymbol{s}_i) \ln\left(\frac{P(\boldsymbol{s}_i)}{P(\boldsymbol{s}_i|\boldsymbol{D})}\right), \tag{2.13}$$

which is zero when the two distributions coincide and that becomes larger the more they differ. Consequently, for a given training set, the optimal dictionary can be found by minimizing Equation (2.13). As $P(\boldsymbol{s})$ does not depend on $\boldsymbol{D}$ but is fixed, this problem is equivalent to finding the maximum likelihood estimator

$$\boldsymbol{D}^\star = \arg\max_{\boldsymbol{D}} \tfrac{1}{M} \sum_{i=1}^{M} \ln(P(\boldsymbol{s}_i|\boldsymbol{D})), \tag{2.14}$$

where the conditional probability distribution that $\boldsymbol{s}_i$ arises from the postulated sparse data model is given as

$$P(\boldsymbol{s}_i|\boldsymbol{D}) = \int P(\boldsymbol{s}_i|\boldsymbol{x},\boldsymbol{D})P(\boldsymbol{x})d\boldsymbol{x}. \tag{2.15}$$

To compute the two required distributions $P(\boldsymbol{s}_i|\boldsymbol{x},\boldsymbol{D})$ and $P(\boldsymbol{x})$, the authors made two assumptions. First, they assume that the approximation error is normally distributed with zero mean and standard deviation σ, i.e. they assume i.i.d. Gaussian noise. With this, the probability that $\boldsymbol{s}$ arises from $\boldsymbol{x}$ is

$$P(\boldsymbol{s}|\boldsymbol{x},\boldsymbol{D}) = \text{const } \exp\Big(-\tfrac{1}{2\sigma^2}\|\boldsymbol{s}-\boldsymbol{D}\boldsymbol{x}\|_2^2\Big), \tag{2.16}$$

with const being a constant scale factor. Second, the assumption about the entries of $\boldsymbol{x}$ being sparse and statistically independent is used to define the distribution $P(\boldsymbol{x})$. Statistical independence is modeled by taking a factorial distribution of the entries of $\boldsymbol{x}$, i.e. $P(\boldsymbol{x}) = \prod_i P(x_i)$. Sparsity is implemented by requiring the distribution of every coefficient x_i to be uni-modal with a peaked maximum at zero and heavy tails. With these two requirements, they end up with

$$P(\boldsymbol{x}) = \text{const } \exp\big(-\beta g(\boldsymbol{x})\big), \tag{2.17}$$

using e.g. $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ for the Laplacian distribution, or $g(\boldsymbol{x}) = \sum_i \ln(1 + x_i^2)$ for the Cauchy distribution.

Unfortunately, finding the solution to (2.14) is computationally intractable as it would require to integrate over all possible sparse codes $\boldsymbol{x}$ in Equation (2.15). Nevertheless, the integral can be approximated by evaluating $P(\boldsymbol{s}|\boldsymbol{x},\boldsymbol{D})P(\boldsymbol{x})$ only at its maximum, which is valid due to the shape of $P(\boldsymbol{x})$ having only one heavily peaked maximum. Using this, the dictionary learning problem is formulated as

$$\boldsymbol{D}^\star = \arg\max_{\boldsymbol{D}} \frac{1}{M} \sum_{i=1}^{M} \arg\max_{\boldsymbol{x}_i} \ln\big(P(\boldsymbol{s}_i|\boldsymbol{x}_i,\boldsymbol{D})P(\boldsymbol{x}_i)\big). \tag{2.18}$$

Inserting (2.16) and (2.17) into Equation (2.18) and exchanging the maximization problem by a minimization task, the final dictionary learning problem reads as

$$\boldsymbol{D}^\star = \arg\min_{\boldsymbol{D}} \frac{1}{M} \sum_{i=1}^{M} \arg\min_{\boldsymbol{x}_i} \|\boldsymbol{s}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 + \lambda g(\boldsymbol{x}_i) \tag{2.19}$$

with $\lambda = 2\sigma^2\beta$.

To solve Problem (2.19), an iterative method based on two alternating steps is suggested. In the first step, the dictionary is fixed and the sparse coefficients are updated by an iterative method such that the gradient of (2.19) with respect to each coefficient vanishes. In the second step, the sparse coefficients are fixed and the dictionary is updated via standard gradient descent with a fixed step size. In that way, an approximate ML-estimate of the dictionary is obtained. As this procedure does not avoid the trivial solution, all columns of $\boldsymbol{D}$ are normalized to have bounded Euclidean norm after the dictionary update step. Thereby, each atom is normalized independently such that the variance of the corresponding sparse coefficients is equal to a preset value. These steps are repeated until some user defined convergence criterion is met. Other closely related ML dictionary learning approaches can be found in [73, 74].

Maximum a Posteriori-Based Dictionary Learning

The dictionary learning technique proposed by Kreutz-Delgado et al. in [71] also adopts a probabilistic point of view to derive a solution to Problem (2.19). However, instead of computing a ML-estimate of $\boldsymbol{D}$ and $\boldsymbol{X}$, they compute a joint *Maximum a Posteriori* (MAP) estimate. To that end, they assume that a dictionary is an element of a compact submanifold of $\mathbb{R}^{n \times d}$ and propose two extensions to the sparse coding algorithm FOCUSS for learning such dictionaries, which they dubbed FOCUSS-FDL and FOCUSS-CNDL. FOCUSS-FDL (Frobenius-Normalized Dictionary Learning) is a method for learning dictionaries with unit Frobenius norm, while FOCUSS-CNDL (Column-Normalized Dictionary Learning) learns dictionaries whose atoms have equally fixed Euclidean norm. The two methods enforce the respective constraints on the dictionary directly during the learning procedure through appropriate update rules. The two proposed update rules are the main contribution of [71], and offer a more sophisticated update compared to simply projecting the dictionary on the admissible set of solutions after it has been updated.

The complete iterative dictionary learning procedure for both FOCUSS-FDL and FOCUSS-CNDL is as follows. The FOCUSS algorithm in its noisy version as given in Equation(2.11) with $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_p^p,\ p \leq 1$ and fixed $\boldsymbol{D}$ is used to compute the sparse code of a training sample. To account for the *joint* MAP estimation of $\boldsymbol{D}$ and $\boldsymbol{X}$, the dictionary is updated every time $M_S \ll M$ sparse coefficients vectors have been updated, i.e. at each sweep over the training set the dictionary is updated $\lceil \frac{M}{M_S} \rceil$ times. Furthermore, to update coefficient vector $\boldsymbol{x}_i$ only one iteration of the FOCUSS algorithm is executed at each sweep. Therefore, sparse code determined at the previous sweep is used for initialization. This sweep over the entire training set is repeated for a fixed number of iterations. The authors experimentally show that FOCUSS-FDL often learns useless atoms having Euclidean norm very close to zero, especially in the most relevant overcomplete dictionary learning case. This phenomenon is inherently avoided by FOCUSS-CNDL, and should therefore be preferred according to the authors.

Another MAP based algorithm that has commonalities with [71] is the majorization method introduced by Yaghoobi et al. [135]. They suggest relaxing the tight constraints of fixed Frobenius norm and fixed column norm by the constraints of upper bounded Frobenius norm and upper bounded column-norm. This relaxation results in two convex admissible sets of solutions, and the arising optimization problem is solved efficiently by a majorization method. As stated by the authors, this flexible approach further allows posing additional constraints on the dictionary such as finding a dictionary with minimum number of atoms.

Method of Optimal Directions (MOD)

In 1999, Engan et al. introduced an iterative least squares based dictionary learning algorithms coined Method of Optimal Directions (MOD) [46]. Therein, each sparse code is restricted to have at most k non-zero entries, and the goal is to find the dictionary which minimizes the Mean Squared Error (MSE) between the training set and its current sparse approximation $\boldsymbol{DX}$, which is formally stated as

$$\{\boldsymbol{D}^\star, \boldsymbol{X}^\star\} \in \arg\min_{\boldsymbol{D},\boldsymbol{X}} \tfrac{1}{Mn}\|\boldsymbol{S} - \boldsymbol{DX}\|_F^2 \quad \text{subject to} \quad \|\boldsymbol{x}_i\|_0 \leq k, \forall i = 1, \dots, M. \tag{2.20}$$

To solve Problem (2.20), the authors suggest an iterative block-coordinate descent method, which consists of first fixing $\boldsymbol{D}$ and updating the sparse code $\boldsymbol{X}$ via any sparse coding method such as OMP or FOCUSS, followed by fixing $\boldsymbol{X}$ and updating the dictionary. As $\boldsymbol{X}$ is fixed during the dictionary update step, the only function that has to be minimized is $\|\boldsymbol{S} - \boldsymbol{DX}\|_F^2$, and the minimizer can be computed in closed form via

$$\boldsymbol{D} = \boldsymbol{SX}^\top(\boldsymbol{XX}^\top)^{-1}. \tag{2.21}$$

This update step does not avoid the scale ambiguity problem and to account for this, the columns of $\boldsymbol{D}$ are normalized to unit Euclidean norm after they have been updated. These steps are repeated either until the MSE does not change significantly between two consecutive iterations, or a maximum number of iterations has been reached. The MOD algorithm only requires a few update steps until convergence, however, computing the inverse $(\boldsymbol{XX}^\top)^{-1}$ can be computationally expensive and/or unstable.

An extension of MOD has been presented in [47] for learning convolutional dictionaries. Furthermore, this modification permits to incorporate linear constraints on the dictionary entries directly into the learning process. Such linear constraints are e.g. selected entries being equal to zero or equal to certain other entries. This is motivated by structures commonly present in analytic dictionaries and reduces the computational burden of applying learned dictionaries of this kind. Another notable MOD based approach is given in [120], which is an online dictionary learning algorithm capable of handling very large training sets. This algorithm updates the dictionary continuously each time a new training sample is available using recursive least squares without having to store or process all contributing samples at each dictionary update step.

K-SVD

The probably best known and most widely used dictionary learning technique is the K-SVD algorithm developed by Aharon et al. [43]. Similar to MOD, K-SVD also follows an iterative block-coordinate descent approach to solve Problem (2.20) using any sparse coding approach like OMP in the sparse coding stage. However, it differs significantly in the dictionary update stage. While MOD updates the entire dictionary at once in a single step, K-SVD performs a sequential atom-by-atom update. Furthermore, the support of the currently considered atom, i.e. the non-zero entries of the corresponding row of $\boldsymbol{X}$, is also refined according to the updated atom. As only the non-zero entries are changed, the sparsity of $\boldsymbol{X}$ is not affected by this.

Concretely, let $\boldsymbol{d}_{:,j}$ be the atom to be updated and let $\boldsymbol{x}_{j,:}$ be the j-th row of $\boldsymbol{X}$, then the data term to be minimized can be factorized as

$$\|\boldsymbol{S} - \boldsymbol{D}\boldsymbol{X}\|_F^2 = \|\boldsymbol{S} - \sum_{i \neq j} \boldsymbol{d}_{:,i}\boldsymbol{x}_{i,:} - \boldsymbol{d}_{:,j}\boldsymbol{x}_{j,:}\|_F^2 = \|\boldsymbol{R}_j - \boldsymbol{d}_{:,j}\boldsymbol{x}_{j,:}\|_F^2. \tag{2.22}$$

Minimizing Equation (2.22) jointly with respect to $\boldsymbol{d}_{:,j}$ and $\boldsymbol{x}_{j,:}$ is simply done by a rank-1 approximation of $\boldsymbol{R}_j$ obtained through its Singular Value Decomposition (SVD). However, this update would most certainly destroy the sparsity pattern of $\boldsymbol{x}_{j,:}$. To overcome this, the authors propose to only consider the reduced matrix $\boldsymbol{R}_j^R$ that consists of the columns of $\boldsymbol{R}_j$ indexed by the support of $\boldsymbol{x}_{j,:}$. Let $\mathcal{I} = \{i \mid x_{ji} \neq 0, \ \forall i = 1, \dots, M\}$ denote this index set, then the atom and the coefficients are updated by computing the SVD $\boldsymbol{R}_j^R = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$ and setting $\boldsymbol{d}_{:,j} = \boldsymbol{u}_{:,1}$ and $\boldsymbol{x}_{j,\mathcal{I}} = \sigma_{11}\boldsymbol{v}_{:,1}^\top$. In that way, only the non-zero coefficients are updated and the sparsity is preserved or might be even increased. Furthermore, the atoms always remain normalized as $\boldsymbol{U}$ is orthogonal, thus, the scale ambiguity problem is automatically avoided.

Due to its efficiency and simplicity, several variants of the K-SVD algorithm have been proposed. These include algorithms for learning from noisy training signals [43], handling color images [80], learning discriminative dictionaries for segmentation and classification [66, 79], finding structured dictionaries [110], or learning dictionaries working on multiple scales [93].

Online Dictionary Learning

Apart from the online extension of MOD [120] all algorithms explained above are iterative learning techniques that require the complete batch of training samples together with the corresponding sparse codes at each iteration. Due to memory and computational limitations this restricts the possible number of samples to be considered. However, large training sets are valuable as they minimize the risk of overfitting the dictionary to the training data and allow to better approximate the expected cost rather than the empirical cost.

To be able to handle large training sets, in [78] Mairal et al. proposed an online dictionary learning algorithm based on stochastic approximations that only requires one sample at each iteration together with two small matrices for updating the dictionary. Concretely, they minimize a ℓ_1-regularized least squares cost function and restrict the atoms to the convex set of vectors with Euclidean norm less or equal to one, i.e.

$$\arg\min_{\boldsymbol{D}} \frac{1}{t}\sum_{i=1}^{t}\Big(\arg\min_{\boldsymbol{x}_i} \tfrac{1}{2}\|\boldsymbol{s}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 + \lambda\|\boldsymbol{x}_i\|_1\Big) \quad \text{subject to} \quad \|\boldsymbol{d}_{:,j}\|_2 \leq 1,\ \forall j = 1,\dots,d, \tag{2.23}$$

At the t-th iteration the algorithm draws one training sample $\boldsymbol{s}_t$ from the available set and computes its sparse code $\boldsymbol{x}_t$ with respect to the current dictionary using the LARS-LASSO algorithm. This sparse code is then fixed, and the dictionary is updated by minimizing the data fidelity term with respect to $\boldsymbol{D}$. To do this efficiently, the problem is reformulated as

$$\begin{aligned} \boldsymbol{D}^\star &= \arg\min_{\boldsymbol{D}} \frac{1}{t}\sum_{i=1}^{t} \tfrac{1}{2}\|\boldsymbol{s}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 && \text{subject to} \quad \|\boldsymbol{d}_{:,j}\|_2 \leq 1,\ \forall j = 1,\dots,d, \\ &= \arg\min_{\boldsymbol{D}} \frac{1}{t}\Big(\tfrac{1}{2}\operatorname{tr}(\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{A}^{(t)}) - \operatorname{tr}(\boldsymbol{D}^\top\boldsymbol{B}^{(t)})\Big) && \text{subject to} \quad \|\boldsymbol{d}_{:,j}\|_2 \leq 1,\ \forall j = 1,\dots,d. \end{aligned} \tag{2.24}$$

with $\boldsymbol{A}^{(t)} = \sum_{i=1}^{t} \boldsymbol{x}_i\boldsymbol{x}_i^T = \boldsymbol{A}^{(t-1)} + \boldsymbol{x}_t\boldsymbol{x}_t^T \in \mathbb{R}^{d\times d}$ and $\boldsymbol{B}^{(t)} = \sum_{i=1}^{t} \boldsymbol{s}_i\boldsymbol{x}_i^T = \boldsymbol{B}^{(t-1)} + \boldsymbol{s}_t\boldsymbol{x}_t^T \in \mathbb{R}^{n\times d}$. Now, the dictionary that minimizes Equation (2.24) is found in a column by column fashion, with an additional orthogonal projection of each atom onto the constraint set. For this approach, only two low-dimensional matrices $\boldsymbol{A}^{(t)}, \boldsymbol{B}^{(t)}$ have to be stored and updated at each iteration together with only solving the sparse coding problem for one single sample. Compared to always processing the complete training set as done in conventional learning methods this massively reduces the required memory and computational resources. Furthermore, the algorithm does not require any cumbersome manual step size tuning and uses the atoms determined at iteration $t-1$ to initialize the update at iteration t.

To further enhance the performance of the method, the authors suggest to simultaneously process small batches of training samples, known as mini-batches, instead of processing only one sample at each iteration. This is a heuristic commonly applied for accelerating stochastic gradient descent methods. Furthermore, to improve the rate of convergence they propose to give more weight to new data points by scaling down the old information contained in $\boldsymbol{A}^{(t-1)}$ and $\boldsymbol{B}^{(t-1)}$. The author of [99] further extended this approach in such a manner that it is possible to learn dictionaries with low coherence.

2.2 Co-Sparse Analysis Model

2.2.1 Analysis Sparse Coding

In the co-sparse analysis signal model, recall that the goal is to find the signal $\boldsymbol{s}$ such that the analyzed vector $\boldsymbol{\alpha} := \boldsymbol{\Omega s}$ is as sparse as possible. Certainly, when $\boldsymbol{s}$ is free of noise one of the advantages of the analysis model is that no optimization problem has to be solved and the sparse analyzed version of a signal is straightforwardly given as the matrix vector product $\boldsymbol{\Omega s}$. However, if the measurements are noisy, i.e. $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{\epsilon}$ or incomplete, i.e. $m < n$, finding the signal that results in the sparsest vector $\boldsymbol{\alpha}$ is no longer trivial and requires to solve the analysis sparse coding problem

$$\boldsymbol{s}^{\star} \in \arg\min_{\boldsymbol{s}} g(\boldsymbol{\Omega s}) \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{A s}\|_2^2 \leq \epsilon. \tag{2.25}$$

Here, as for the sparse coding problem introduced in 2.1.1, I employ the common quadratic error term, which corresponds to the assumption of i.i.d. Gaussian noise. Now, one straightforward way to tackle Problem (2.25), is to utilize the ℓ_1-norm [16, 45, 129] or any appropriate differentiable function as the measure of sparsity and employ any convex optimization solver or any first-order general purpose solver. Besides that, another possibility is to use a specialized solver that exploits the structure of the co-sparse analysis model. Compared to the numerous specialized synthesis sparse coding techniques, only a handful specialized analysis sparse coding approaches exist, which I review in the following.

Greedy Pursuit Methods

As for the synthesis sparse coding problem, pursuit methods exist that find an approximate solution to Problem (2.25) with $g(\boldsymbol{\Omega s}) = \|\boldsymbol{\Omega s}\|_0$ by determining the co-support of $\boldsymbol{s}$ in a greedy one-by-one fashion. Here, I want to explain the underlying concept based on the Backward-Greedy (BG) analysis sparse coding approach [109] with $\boldsymbol{A} = \boldsymbol{I}_n$, i.e. a basic denoising problem. This algorithm aims at finding the signal $\boldsymbol{s}$, which for a given analysis operator has the smallest *co-rank*. Let $\mathcal{I}$ denote the co-support of $\boldsymbol{s}$, then the co-rank of $\boldsymbol{s}$ with respect to $\boldsymbol{\Omega}$ is defined as the rank of the submatrix of $\boldsymbol{\Omega}$ whose rows are indexed by $\mathcal{I}$, i.e. $\operatorname{rk}(\boldsymbol{\Omega}_{\mathcal{I},:})$. The co-rank measure is related to the dimension c of the subspace the signal resides in via $\operatorname{rk}(\boldsymbol{\Omega}_{\mathcal{J},:}) = n - c$. Now, given the noisy measurements $\boldsymbol{y}$ the analysis based signal reconstruction problem is formulated as

$$\{\boldsymbol{s}^\star, \mathcal{J}^\star\} \in \arg\min_{\boldsymbol{s},\mathcal{J}} \|\boldsymbol{s} - \boldsymbol{y}\|_2^2 \quad \text{subject to} \quad \begin{aligned} &\boldsymbol{\Omega}_{\mathcal{J},:}\boldsymbol{s} = \boldsymbol{0}_{|\mathcal{J}|}, \\ &\operatorname{rk}(\boldsymbol{\Omega}_{\mathcal{J},:}) = n - c, \end{aligned} \tag{2.26}$$

with $|\mathcal{J}|$ being the cardinality of the set. The BG-algorithm solves Problem (2.26) by iteratively determining one row of $\boldsymbol{\Omega}$ at a time that corresponds to a *zero entry* of $\boldsymbol{\alpha}$. It starts from $\boldsymbol{s}^{(0)} = \boldsymbol{y}$ and initializes the co-support with the empty set, i.e. $\mathcal{J}^{(0)} = \emptyset$. At each iteration the index of the row $\boldsymbol{\omega}_{j,:}$, which is most uncorrelated with the current signal estimate is added to the co-support, i.e.

$$\mathcal{J}^{(i)} = \mathcal{J}^{(i-1)} \cup \arg\min_{j \notin \mathcal{J}^{(i-1)}} |\boldsymbol{\omega}_{j,:}\boldsymbol{s}^{(i-1)}|. \tag{2.27}$$

Then, the signal estimate is updated by $\boldsymbol{s}^{(i)} = (\boldsymbol{I}_n - \boldsymbol{\Omega}^\dagger_{\mathcal{J}^{(i)},:}\boldsymbol{\Omega}_{\mathcal{J}^{(i)},:})\boldsymbol{y}$, i.e. the measurements are projected on the subspace that is orthogonal to the selected rows. Note that this projection can be computed efficiently without explicitly computing the pseudoinverse $\boldsymbol{\Omega}^\dagger_{\mathcal{J}^{(i)},:}$. Finally, the current co-support is refined by adding the indices of the rows whose inner product with the current signal estimate $|\boldsymbol{\omega}_{j,:}\boldsymbol{s}^{(i)}|$ is less than a user defined threshold. The entire process is repeated until the signal estimate has the user defined co-rank $n - c$.

An improved version of BG called Optimized BG (OBG), has also been suggested in [109]. Rather than simply selecting the row whose inner product with $\boldsymbol{s}^{(i-1)}$ is the smallest, the full co-support update and projection step for all possible rows not already added to the co-support is performed. Then, the row that leads to the smallest decrease in the energy of the signal, i.e. the atom that minimizes $\|\boldsymbol{s}^{(i-1)} - \boldsymbol{s}^{(i)}\|_2$, is added to the co-support. Note that this version is computationally more demanding compared to its unoptimized counterpart.

Another greedy pursuit method is the Greedy-Analysis-Pursuit (GAP) algorithm introduced in [86]. The main conceptual difference between GAP and BG is that GAP initializes the co-support of the signal with all row indices of $\boldsymbol{\Omega}$, i.e. $\mathcal{J}^{(0)} = \{1, \ldots, a\}$, and then iteratively removes one index at a time. This means that it detects the non-zero entries of $\boldsymbol{\alpha}$, rather than its zero entries as done by BG, thus, it is computationally more efficient for signals with high level of co-sparsity.

Last, I want to mention that these methods can be used to solve more general inverse problems with $\boldsymbol{A} \neq \boldsymbol{I}_n$. Therefore, the signal update has to be changed from the standard projection step to solving the constrained least squares problem $\boldsymbol{s}^{(i)} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{\Omega}_{\mathcal{J}^{(i)},:}\boldsymbol{s}\|_2^2 + \lambda\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|_2^2$ at each iteration, see [86].

Greedy-Like Pursuit Methods

Analysis Iterative Hard Threshold (AIHT), Analysis Hard Thresholding Pursuit (AHTP) [59], Analysis CoSaMP (ACoSaMP), and Analysis Subspace Pursuit (ASP) [57] are four greedy-like algorithms that transfer the concepts of the four synthesis based sparse coding algorithms IHT, HTP, CoSaMP, and SP to the analysis sparse coding problem. All four methods are iterative algorithms that aim at finding the signal whose co-support is equal to l. Therefore, the function $\mathrm{co}_l(\boldsymbol{s})$ is required that returns the set of l indices corresponding to the l smallest values in magnitude of $\boldsymbol{\Omega s}$. The outcome of co_l can be interpreted as an approximation of the co-support of the signal, and constitutes the analysis counterpart to the hard thresholding operation performed in the synthesis case for finding the support of a signal. At each iteration, all four algorithms perform different approximate projections of the measurements on the closest co-sparse subspace to update the signal. The methods are initialized with $\boldsymbol{s}^{(0)} = \boldsymbol{0}_n$ and consequently $\mathcal{I}^{(0)} = \{1, \ldots, a\}$, and repeat the steps introduced below until a user-defined stopping criterion is met.

At the i-th iteration, AIHT and AHTP first take a step of length t in the direction of the negative gradient of the fidelity term, i.e. $\boldsymbol{s}_g = \boldsymbol{s}^{(i-1)} + t\boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}^{(i-1)})$, and update the approximate co-support via $\mathcal{I}^{(i)} = \mathrm{co}_l(\boldsymbol{s}_g)$. Then, AIHT finds the signal estimate $\boldsymbol{s}^{(i)}$ by projecting $\boldsymbol{s}_g$ onto the nullspace of $\boldsymbol{\Omega}_{\mathcal{I}^{(i)},:}$, while AHTP finds it by solving the optimization problem $\boldsymbol{s}^{(i)} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|_2^2$ subject to $\boldsymbol{\Omega}_{\mathcal{I}^{(i)},:}\boldsymbol{s} = \boldsymbol{0}_{|\mathcal{I}^{(i)}|}$. Crucial to the recovery success of both methods is the choice of the step size t, see [58] for a discussion on that issue.

ACoSaMP and ASP first compute a temporary co-support $\mathcal{C}$ that is given as the intersection of $\mathcal{I}^{(i-1)}$ and $\mathrm{co}_{cl}(\boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}^{(i-1)}))$, with $0 < c \leq 1$. The parameters c is most commonly set to one. Next, $\mathcal{C}$ is used to compute a temporary signal estimate via $\boldsymbol{s}_g = \arg\min_{\boldsymbol{s}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|_2^2$ subject to $\boldsymbol{\Omega}_{\mathcal{C},:}\boldsymbol{s} = \boldsymbol{0}_{|\mathcal{C}|}$. As the co-support of $\boldsymbol{s}_g$ can be smaller than the targeted dimension l, the co-support is reestimated by $\mathcal{I}^{(i)} = \mathrm{co}_l(\boldsymbol{s}_g)$. Using $\mathcal{I}^{(i)}$, ACoSaMP computes $\boldsymbol{s}^{(i)}$ identical to AIHT, while ASP performs the same update as AHTP.

AIHT and AHTP are computationally less demanding compared to ACoSaMP and ASP, while the latter are more efficient in finding the best co-sparse signal estimate. A theoretical analysis of all four greedy-like methods regarding recovery guarantees can be found in [58].

As for the sparse synthesis model whose performance most severely depends on the employed dictionary, the success of the co-sparse analysis model depends on the chosen analysis operator. How to chose and learn an analysis operator from example data is covered in the next section.

2.2.2 Analysis Operator Learning

Regarding the choice of the analysis operator, similar to choosing the dictionary in the sparse synthesis model, there are two possible alternatives: either select an analytically defined analysis operator or employ a learned operator that is adapted to the signal class of interest. The probably best known and most widely applied analytic analysis operator is the finite difference operator, which approximates first order derivatives. This operator is also used for computing the famous Total Variation (TV)-norm [111] in image processing. Other analytic operators are the transposed of diverse tight frames such as the overcomplete cosine transform, undecimated wavelet transforms [105], the curvelet transform [121], or a concatenation of several of these transforms [16]. As for dictionaries in the synthesis model, these general purpose approaches are not able to find an optimal co-sparse representation of more specific signals, and to account for this an analysis operator can be learned from representative examples signals. In contrast to the huge amount of existing dictionary learning methods, the number of existing analysis operator learning algorithms is rather small. In the following, I explain the most important contributions in this field.

Field-of-Experts

Field-of-Experts (FoE) introduced by Black and Roth [106, 107] is the earliest method that aims at learning an analysis operator from example signals. Even though the authors did not use the terms analysis model and analysis operator, the concepts underlying FoE are same as those underlying the analysis model.

Motivated by finding a probabilistic prior model for the spatial structure of natural images, they formulate a high-order Markov Random Field (MRF) defined over an entire image. If $\boldsymbol{s} \in \mathbb{R}^N$ denotes a vectorized image with N pixels, then each pixel of $\boldsymbol{s}$ is interpreted as one node of the MRF. For each node, a maximal clique $\boldsymbol{s}_{[j]} \in \mathbb{R}^n$ is defined, which for the j-th node is nothing more but the $(\sqrt{n} \times \sqrt{n})$-dimensional patch centered at the j-th pixel of $\boldsymbol{s}$. A local field potential is assigned to each node by analyzing the associated clique via a linear filters $\boldsymbol{\omega}_i,\ i = 1, \dots, a$, together with a expert functions $\phi(\cdot, \gamma_i), i = 1, \dots, a$ as

$$E_{FoE}(\boldsymbol{s}_{[j]}, \Theta) := -\sum_{i=1}^{a} \phi(\boldsymbol{\omega}_i^\top \boldsymbol{s}_{[j]}, \gamma_i). \tag{2.28}$$

Therein, $\Theta = \{\boldsymbol{\omega}_i, \gamma_i\}_{i=1}^{a}$ denotes the set of model parameters. Equation (2.28) connects the FoE model with the analysis model: The linear filters are the transposed of the rows of the analysis operator, and ϕ is a function that enforces the filter responses to be sparse.

One example function suggested by the authors is the logarithm of Student's-t distribution $\phi(\boldsymbol{\omega}_{i,:}^\top \boldsymbol{s}_{[j]},\ \gamma_i) = \ln(1 + \frac{1}{2}(\boldsymbol{\omega}_i^\top \boldsymbol{s}_{[j]})^2)^{-\gamma_i}$, with $\gamma_i \in \mathbb{R}^+$. With this, the Field-of-Experts model for an entire image is formulate as

$$p_{FoE}(\boldsymbol{s},\ \Theta) := \tfrac{1}{Z(\Theta)} \exp\left(\sum_{j=1}^{N} -E_{FoE}(\boldsymbol{s}_{[j]},\ \Theta)\right), \tag{2.29}$$

with $Z(\Theta)$ being a normalization factor that is also known as the partition function. Equation (2.29) is a Markov random field of experts, which explains the name of the algorithm.

Now, the required model parameters Θ are learned from a set of M training images $\mathcal{S} = \{\boldsymbol{s}_i \in \mathbb{R}^N\}_{i=1}^M$ by maximizing the log-likelihood function

$$\Theta^\star \in \arg\max_{\Theta} \sum_{i=1}^{M} \ln(p_{FoE}(\boldsymbol{s}_i,\ \Theta)). \tag{2.30}$$

In the FoE approach, this is done by standard gradient ascent, which for the i-th parameter set θ_i leads to the update formula

$$\theta_i^{(k+1)} = \theta_i^{(k)} + t\left(\left\langle \frac{\partial E_{FoE}}{\partial \theta_i} \right\rangle_{p_{FoE}} - \left\langle \frac{\partial E_{FoE}}{\partial \theta_i} \right\rangle_{\mathcal{S}}\right). \tag{2.31}$$

Therein, $t \in \mathbb{R}^+$ is a user defined step size, $\left\langle \frac{\partial E_{FoE}}{\partial \theta_i} \right\rangle_{\mathcal{S}}$ is the expectation with respect to the M training samples, and $\left\langle \frac{\partial E_{FoE}}{\partial \theta_i} \right\rangle_{p_{FoE}}$ is the expectation with respect to the model distribution p_{FoE}. While the former is simply given as the average i-th partial derivative over all training samples, the latter cannot be computed in closed form. However, it can be approximated using Markov Chain Monte Carlo (MCMC) sampling techniques. As MCMC sampling is known to be slow, the authors instead employ a hybrid Monte Carlo sampler in combination with contrastive divergence [64], i.e. initializing the sampler at the training data and just performing one MCMC iteration.

Extensions of the FoE model exist, e.g. to handle color images [84], that employ Gaussian Scale Mixtures as expert functions and an efficient Gibbs Sampler [114], or enforce the filters to be normalized and employ persistent contrastive divergence to learn them [55].

Learning Uniformly Normalized Tight Frames

A constrained ℓ_1-optimization problem for learning an analysis operator from a possibly noisy set of training samples has been introduced in [137, 138, 139]. Let $\boldsymbol{Y} \in \mathbb{R}^{n \times M}$ be the matrix whose columns constitute the noisy training samples, the analysis operator problem can be formulated as

$$\{\boldsymbol{\Omega}^\star, \boldsymbol{S}^\star\} \in \arg\min_{\boldsymbol{\Omega},\boldsymbol{S}} \|\boldsymbol{\Omega S}\|_1 + \tfrac{\lambda}{2}\|\boldsymbol{S} - \boldsymbol{Y}\|_F^2 \quad \text{subject to} \quad \boldsymbol{\Omega} \in \mathfrak{C}, \tag{2.32}$$

where an optimization over both the operator $\boldsymbol{\Omega}$, as well as the denoised training set $\boldsymbol{S} \in \mathbb{R}^{n\times M}$ has to be performed. The weighting parameter $\lambda \in \mathbb{R}^+$ is chosen depending on the assumed noise energy and $\mathfrak{C}$ denotes an admissible set of solutions. In analogy to dictionary learning, constraining the operator to an admissible set is necessary to avoid the trivial solution $\boldsymbol{\Omega} = \boldsymbol{0}_{a\times n}$ and other useless solutions. To that end, the authors discuss the properties of several constraint sets. First, they argue that the common constraint used by dictionary learning methods of solely fixing the norm of the atoms of $\boldsymbol{\Omega}$, i.e. $\|\boldsymbol{\omega}_{i,:}\|_2 = \text{const}, \forall i = 1,\dots,a$, is not sufficient to find a useful operator. Though $\boldsymbol{\Omega} = \boldsymbol{0}_{a\times n}$ is avoided, this constraint leads to a simple rank-1 operator with only one distinctive row $\boldsymbol{\omega}^\star$ given as $\boldsymbol{\omega}^\star = \arg\min_{\boldsymbol{\omega}} \|\boldsymbol{\omega S}\|_1$ repeated a times. Intuitively, this problem could be resolved by additionally requiring $\boldsymbol{\Omega}$ having full-rank. Unfortunately, this rather lose constraint leads to an ill-conditioned operator that contains rows that are all strongly correlated with $\boldsymbol{\omega}^\star$. A well-conditioned operator can be found, by demanding $\boldsymbol{\Omega}$ to be a Tight Frame (TF), i.e. $\boldsymbol{\Omega}^\top\boldsymbol{\Omega} = \boldsymbol{I}_n$. However, in the interesting overcomplete case, the authors show that this constraint alone results in an operator that is a concatenation of a $(n \times n)$-dimensional orthogonal basis, and an $(a - n \times n)$-dimensional zero matrix. Consequently, this operator does not provided any more information compared to a complete square analysis operator.

From these observations, the authors finally propose to restrict the set of solutions to matrices that are tight frames with normalized rows. This set of matrices is known as *Uniformly Normalized Tight Frames* (UNTF), and is formally defined as

$$\mathfrak{C}_{\text{UNTF}} := \left\{\boldsymbol{\Omega} \in \mathbb{R}^{a\times n} \mid \boldsymbol{\Omega}^\top\boldsymbol{\Omega} = \boldsymbol{I}_n,\ \|\boldsymbol{\omega}_{i,:}\|_2 = \text{const}, \forall i = 1,\dots,a\right\}. \tag{2.33}$$

Employing this constraint set in Problem (2.32) allows to learn a well-conditioned overcomplete analysis operator. To solve the arising optimization problem, an alternating projected subgradient method is proposed based on first fixing $\boldsymbol{S}$ and updating the operator, followed by fixing the operator and updating the signal estimates. The operator is updated by taking a step of length t in the direction of the negative subgradient $\boldsymbol{\Omega}_G = \partial_{\boldsymbol{\Omega}}\|\boldsymbol{\Omega S}\|_1$, followed by projecting $\boldsymbol{\Omega} + t\boldsymbol{\Omega}_G$ onto the UNTF-set. This projection can only be computed approximately via projecting $\boldsymbol{\Omega} + t\boldsymbol{\Omega}_G$ onto the TF-set, i.e. setting all its singular values to one and afterwards normalizing each row to const. When the projection onto the TF-set leads to a zero row, this row is replaced by a random vector normalized to const. The step-size t is either fixed or determined by a line search that ensures that the cost function decreases.

To update S a standard co-sparse coding problem has to be solved, which can be done by any co-sparse coding technique. In [139], the convexity of the ℓ_1+ℓ_2 problem is exploited and S is updated via the Douglas Rachford Splitting technique, which is also known as the Augmented Lagrangian method. The two alternating steps of the analysis operator learning algorithm are repeated either for a fixed number of iterations or until the relative change between two consecutive solutions falls below a user defined threshold. It is stated by the authors that their algorithm is not guaranteed to converge to an operator that is an UNTF, but they always observed convergence in practice.

Analysis K-SVD

Analysis K-SVD as proposed in [108, 109] is an analysis operator learning approach that takes a similar route as the widely known K-SVD dictionary learning technique. Its underlying assumption is that every ideal noise free example signal $\boldsymbol{s}_i$ lies in a low-dimensional subspace of dimension $c < n$, which is related to an analysis operator $\boldsymbol{\Omega}$ that can be learned from a set of noisy training signals $\boldsymbol{Y} \in \mathbb{R}^{n \times M}$ while simultaneously denoising the set. With $\boldsymbol{S} := [\boldsymbol{s}_1, \dots, \boldsymbol{s}_M] \in \mathbb{R}^{n \times M}$ being the denoised training set and $\mathcal{I}_i$ denoting the co-support of the i-th training signal, the analysis operator learning problem is formulated as

$$\left\{\boldsymbol{\Omega}^\star, \boldsymbol{S}^\star, \{\mathcal{I}_i^\star\}_{i=1}^M\right\} \in \arg \min_{\boldsymbol{\Omega}, \boldsymbol{S}, \{\mathcal{I}_i\}_{i=1}^M} \|\boldsymbol{S} - \boldsymbol{Y}\|_F^2 \quad \text{subject to} \quad \begin{aligned} &\boldsymbol{\Omega}_{\mathcal{I}_i,:}\boldsymbol{s}_{:,i} = \boldsymbol{0}_{|\mathcal{I}_i|}, && \forall i = 1, \dots, M \\ &\operatorname{rk}(\boldsymbol{\Omega}_{\mathcal{I}_i,:}) = n - c, && \forall i = 1, \dots, M \\ &\|\boldsymbol{\omega}_{j,:}\|_2 = 1, && \forall j = 1, \dots, a, \end{aligned} \tag{2.34}$$

where the rows of $\boldsymbol{\Omega}$ are constrained to unit Euclidean norm to avoid the trivial solution. This optimization problem is solved by an iterative two-phase block-coordinate-relaxation approach. Starting from an initial analysis operator, the algorithm first fixes $\boldsymbol{\Omega}$ and optimizes over both the signal estimates $\boldsymbol{S}$ and the co-supports $\{\mathcal{I}_i\}_{i=1}^M$ using the BG or OBG-algorithm explained in Section 2.2.1. In the second phase, the determined co-supports are used to update the operator, where similar to the synthesis K-SVD dictionary learning algorithm each analysis atom $\boldsymbol{\omega}_{j,:}$ is updated independently. Concretely, the j-th atom is updated by first identifying the set $\mathcal{C} := \{i \mid j \in \mathcal{I}_i,\ i = 1, \dots, M\}$ that contains the indices of those signal estimates whose co-supports include the index j. Then, this set is used to update the row $\boldsymbol{\omega}_{j,:}$ by solving

$$\boldsymbol{\omega}_{j,:} = \arg\min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\boldsymbol{Y}_{:,C}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\omega}\|_2 = 1, \tag{2.35}$$

with $\boldsymbol{Y}_{:,C}$ being the submatrix of $\boldsymbol{Y}$ containing the columns indexed by C. The solution to Problem (2.35) is given in closed form as the left singular vector of $\boldsymbol{Y}_{:,C}$ that corresponds to its *smallest* singular value.

The two phases of the algorithm are repeated for a fixed number of iterations. To resolve possible deadlock situations in this iterative process, the authors propose to reinitialize an atom whenever it is contained in too few co-supports of the training signals, or whenever it becomes too similar to any other atom of $\boldsymbol{\Omega}$. One proposed reinitialization is to randomly select $n - 1$ columns of $\boldsymbol{Y}$, and to use the vector that spans their nullspace as the new analysis atom. As stated by the authors, one advantage of their algorithm is that each atom is updated independently and, thus, the update can be performed in parallel.

Very closely related to the analysis K-SVD method is the approach presented in [92]. This algorithm sequentially finds the rows of the analysis operator by identifying directions that are orthogonal to a subset of the training samples. Starting from a randomly initialized vector $\boldsymbol{\omega} \in \mathbb{R}^n$, a candidate row is found by first computing the inner product of $\boldsymbol{\omega}$ with the entire training set, followed by extracting the reduced training set $\boldsymbol{S}_R$ of samples whose inner product with $\boldsymbol{\omega}$ is smaller than a threshold. Thereafter, $\boldsymbol{\omega}$ is updated to be the eigenvector corresponding to the smallest eigenvalue of the Gramian matrix $\boldsymbol{S}_R\boldsymbol{S}_R^\top$. This procedure is iterated several times until a convergence criterion is met. If the determined candidate vector is sufficiently distinctive from already found ones, it is added to $\boldsymbol{\Omega}$ as a new row, otherwise it is discarded. This process is repeated until the desired number a of rows have been determined.

Learning Sparsifying Transforms

Last, I want to mention a generalization of the analysis model called *Sparse Transform Model*, which has been introduced by Ravishankar and Bresler in [101]. The assumption that underpins this model is that a signal multiplied by a sparsifying transform $\boldsymbol{W} \in \mathbb{R}^{a \times n}$ results in an *approximately* sparse vector, i.e. $\boldsymbol{W}\boldsymbol{s} = \boldsymbol{\alpha} + \boldsymbol{\epsilon}$ where $\boldsymbol{\alpha} \in \mathbb{R}^a$ is sparse and $\boldsymbol{\epsilon} \in \mathbb{R}^a$ is called the representation error that is assumed to have small energy compared to the signal. This assumption also underlies classical transform coding techniques, which explains the name transform model. The advantage of this approach is that $\boldsymbol{\alpha}$ does not necessarily have to lie in the range space of $\boldsymbol{W}$, which means that this model is able to represent a wider class of signals compared to the analysis model. To find a signal that adheres to this model, the transform sparse coding problem

$$\boldsymbol{\alpha}^\star = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha} - \boldsymbol{W}\boldsymbol{s}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq s, \tag{2.36}$$

has to be solved. This can be done exactly by thresholding the product $\boldsymbol{W}\boldsymbol{s}$, i.e. only retaining the s largest coefficients and setting the others to zero. Once $\boldsymbol{\alpha}^\star$ is known, the corresponding signal is recovered by solving $\boldsymbol{s}^\star = \arg\min_{\boldsymbol{s}} \|\boldsymbol{\alpha}^\star - \boldsymbol{W}\boldsymbol{s}\|_2^2$, which is given analytically in closed form as $\boldsymbol{s}^\star = \boldsymbol{W}^\dagger\boldsymbol{\alpha}^\star$. Note that this approach is computationally much easier as the synthesis- or analysis sparse coding task, which both require to solve an optimization problem.

Like in the other sparse data models, successfully applying the transform model severely depends on the choice of $\boldsymbol{W}$, and learning this matrix from training samples $\{\boldsymbol{s}_i \in \mathbb{R}^n\}_{i=1}^M$ is highly valuable. Simply learning $\boldsymbol{W}$ via

$$\left\{\boldsymbol{W}^\star, \{\boldsymbol{\alpha}_i^\star\}_{i=1}^M\right\} = \arg \min_{\boldsymbol{W}, \{\boldsymbol{\alpha}_i\}_{i=1}^M} \sum_{i=1}^M \|\boldsymbol{\alpha}_i - \boldsymbol{W}\boldsymbol{s}_i\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}_i\|_0 \leq s, \ i = 1, \ldots, M. \tag{2.37}$$

suffers from the scale ambiguity problem as well as from leading to degenerated matrices that e.g. contain repeated- , zero-, or linearly dependent rows. To avoid these problems, the authors of [101] argue that $\boldsymbol{W}$ must be a full column rank matrix with bounded Frobenius norm. As their goal is to find a square matrix, i.e. $a = n$, they enforce the full-rank constraint by minimizing the negative logarithm of the determinate of $\boldsymbol{W}$. Additionally, they minimize the Frobenius-norm of $\boldsymbol{W}$. These constraints do not only avoid the aforementioned degenerated cases but also enforce $\boldsymbol{W}$ to be well-conditioned, which the authors empirically show to be an important property in applications. Combining all that, learning the sparsifying transform is accomplished by

$$\left\{\boldsymbol{W}^\star, \{\boldsymbol{\alpha}_i^\star\}_{i=1}^M\right\} = \arg \min_{\boldsymbol{W}, \{\boldsymbol{\alpha}_i\}_{i=1}^M} \sum_{i=1}^M \|\boldsymbol{\alpha}_i - \boldsymbol{W}\boldsymbol{s}_i\|_2^2 - \lambda \ln\det \boldsymbol{W} + \mu\|\boldsymbol{W}\|_F^2 \tag{2.38}$$
$$\text{subject to} \quad \|\boldsymbol{\alpha}_i\|_0 \leq s, \ i = 1, \ldots, M,$$

with $\lambda, \mu \in \mathbb{R}^+$ being two manually tuned weighting factors. To solve Problem (2.38), an alternating optimization process is suggested, which is based on fixing $\boldsymbol{W}$ and computing $\{\boldsymbol{\alpha}_i\}_{i=1}^M$ by thresholding $\{\boldsymbol{W}\boldsymbol{s}_i\}_{i=1}^M$, followed by fixing $\{\boldsymbol{\alpha}_i\}_{i=1}^M$ and solving for $\boldsymbol{W}$ by a standard conjugate-gradient scheme. An extension of Problem (2.38) is also proposed in [101] that allows to simultaneously learn $\boldsymbol{W}$ and denoise the training data.

Chapter 3

Optimization on Matrix Manifolds

When the set of solutions of an optimization problem is known to be restricted to a smooth manifold geometric optimization techniques that exploit the underlying manifold structure can be employed to efficiently solve the optimization problem. Among those techniques, geometric Conjugate Gradient (CG) methods have been proven very efficient in various applications, due to the combination of moderate computational complexity and good convergence properties, see e.g. [70] for a CG-type method on the oblique manifold.

All sparse data model learning algorithms proposed in this thesis are based on geometric CG-methods. For this reason, in this section I first shortly review the general concepts of optimization on matrix manifolds. After that, I explain how a general differentibale cost function can be optimized by means of a geometric CG-method. For a more in-depth introduction on optimization on matrix manifolds, I refer the interested reader to [1].

3.1 General Concept

Let M be a smooth Riemannian submanifold of some Euclidean space endowed with the standard Frobenius inner product

$$\langle \boldsymbol{Q}, \boldsymbol{P} \rangle := \operatorname{tr}(\boldsymbol{Q}^\top \boldsymbol{P}), \tag{3.1}$$

and let $f\colon \mathrm{M} \to \mathbb{R}$ be a differentiable cost function. The ultimate goal of this section is to introduce and understand optimization methods that are capable of solving optimization problems on manifolds, i.e. to find a solution to

$$\arg\min_{\boldsymbol{X} \in \mathrm{M}} f(\boldsymbol{X}). \tag{3.2}$$

All geometric concepts presented in this section are visualized in Figure 3.1 to alleviate the understanding.

To every point on the manifold $\boldsymbol{X} \in \mathrm{M}$ one can assign a tangent space $T_{\boldsymbol{X}}\,\mathrm{M}$. The tangent space at $\boldsymbol{X}$ is a real vector space that contains all possible directions that tangentially pass M through $\boldsymbol{X}$. Each tangent space is associated with an inner product inherited from the surrounding Euclidean space, which allows to measure distances and angles on the manifold M. An element $\boldsymbol{\Xi} \in T_{\boldsymbol{X}}\,\mathrm{M}$ is called a tangent vector at $\boldsymbol{X}$.

The Riemannian gradient of f at $\boldsymbol{X}$ is an element of the tangent space $T_{\boldsymbol{X}}\,\mathrm{M}$ that points in the direction of steepest ascent of the cost function on the manifold. For the case where f is globally defined on the entire surrounding Euclidean space, the Riemannian gradient $\boldsymbol{G}(\boldsymbol{X})$ is nothing more but the orthogonal projection of the (standard) Euclidean gradient $\nabla f(\boldsymbol{X})$ onto the tangent space $T_{\boldsymbol{X}}\,\mathrm{M}$. In formulas, this is given as

$$\boldsymbol{G}(\boldsymbol{X}) := \Pi_{T_{\boldsymbol{X}}\,\mathrm{M}}(\nabla f(\boldsymbol{X})). \tag{3.3}$$

For the optimization procedures described in the subsequent chapters, it is necessary to understand the notion of *geodesics*. A geodesic $\Gamma(\boldsymbol{X}, \boldsymbol{\Xi}, t)$ is a smooth curve on M emanating from $\boldsymbol{X}$ in the direction of $\boldsymbol{\Xi} \in T_{\boldsymbol{X}}\,\mathrm{M}$, which locally describes the shortest path between two points on M. Intuitively, it can be interpreted as the equivalent of a straight line in the manifold setting. Now, in minimization procedures conventional iterative line search methods search for the next iterate by determine how far they have to step along a given search direction such that the cost function decreases sufficiently, i.e. the search is performed along a straight line determined by the search direction. This concept is generalized to the manifold setting as follows. Given a current point $\boldsymbol{X}^{(i)}$ together with a search direction $\boldsymbol{H}^{(i)} \in T_{\boldsymbol{X}^{(i)}}\,\mathrm{M}$ at the i-th iteration, the step size $\alpha^{(i)} \in \mathbb{R}^{+}$, which leads to a sufficient decrease of the cost function can be determined by finding the minimizer of

$$\alpha^{(i)} = \arg\min_{t \geq 0} f(\Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, t)). \tag{3.4}$$

This procedure exactly corresponds to a line search along a geodesic rather than along a straight line. Once $\alpha^{(i)}$ has been determined, we obtain a new iterate that lies on M through

$$\boldsymbol{X}^{(i+1)} = \Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, \alpha^{(i)}), \tag{3.5}$$

i.e. one moves from $\boldsymbol{X}^{(i)}$ along the geodesic in the direction of $\boldsymbol{H}^{(i)}$ for length $\alpha^{(i)}$.

Returning to the problem stated at the beginning, one straightforward approach to min-

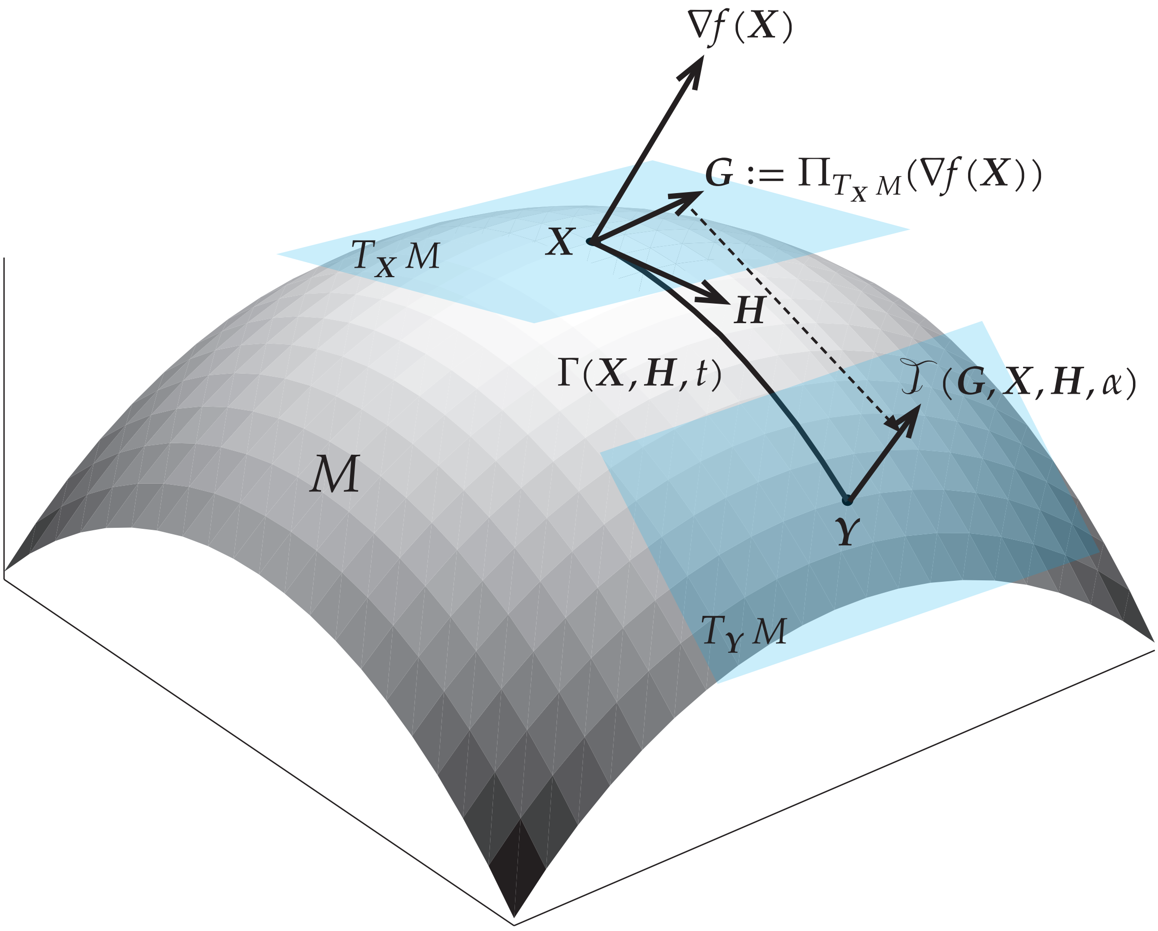

$\nabla f(\boldsymbol{X})$
$\boldsymbol{G} := \Pi_{T_{\boldsymbol{X}}M}(\nabla f(\boldsymbol{X}))$
$T_{\boldsymbol{X}}M$
$\boldsymbol{X}$
$\boldsymbol{H}$
$\Gamma(\boldsymbol{X},\boldsymbol{H},t)$
$\mathcal{T}(\boldsymbol{G},\boldsymbol{X},\boldsymbol{H},\alpha)$
M
$\boldsymbol{Y}$
$T_{\boldsymbol{Y}}M$

Figure 3.1: This figure shows two points $\boldsymbol{X}$ and $\boldsymbol{Y}$ on a manifold M together with their corresponding tangent spaces $T_{\boldsymbol{X}}$ M and $T_{\boldsymbol{Y}}$ M depicted in light blue. Furthermore, the Euclidean gradient $\nabla f(\boldsymbol{X})$ of some cost function f and its projection onto the tangent space at $\boldsymbol{X}$ $\Pi_{T_{\boldsymbol{X}}\mathrm{M}}(\nabla f(\boldsymbol{X}))$ are shown. The geodesic $\Gamma(\boldsymbol{X},\boldsymbol{H},t)$ in the direction of $\boldsymbol{H} \in T_{\boldsymbol{X}}$ M connecting the two points $\boldsymbol{X}$ and $\boldsymbol{Y}$ is shown. The dashed line typifies the role of a parallel transport of the gradient in the tangent space $T_{\boldsymbol{X}}$ M to the tangent space $T_{\boldsymbol{Y}}$ M.

imize f is to alternate Equations (3.3), (3.4), and (3.5) using $\boldsymbol{H}^{(i)} = -\boldsymbol{G}^{(i)}$, with the short hand notation $\boldsymbol{G}^{(i)} := \boldsymbol{G}(\boldsymbol{X}^{(i)})$. This approach exactly corresponds to the steepest descent on a Riemannian manifold. However, as in standard Euclidean optimization, steepest descent only has a linear rate of convergence. Therefore, here I focus on a conjugate gradient method on manifolds, as it offers a superlinear rate of convergence, while at the same time being efficiently applicable to large scale optimization problems with low computational complexity, as opposed quasi newton method or second order methods.

3.2 Geometric Conjugate Gradient

In CG-methods, the search direction for the next iteration $\boldsymbol{H}^{(i+1)} \in T_{\boldsymbol{X}^{(i+1)}}\mathrm{M}$ is a linear combination of the current gradient $\boldsymbol{G}^{(i+1)} \in T_{\boldsymbol{X}^{(i+1)}}\mathrm{M}$ and the previous search direction $\boldsymbol{H}^{(i)} \in T_{\boldsymbol{X}^{(i)}}\mathrm{M}$. Since the addition of vectors that belong to different tangent spaces is not a well-defined operation, $\boldsymbol{H}^{(i)}$ needs to be mapped from $T_{\boldsymbol{X}^{(i)}}\mathrm{M}$ to $T_{\boldsymbol{X}^{(i+1)}}\mathrm{M}$. This mapping is performed by the so-called parallel transport $\mathcal{T}(\boldsymbol{\Xi}, \boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, \alpha^{(i)})$, which transports a tangent vector $\boldsymbol{\Xi} \in T_{\boldsymbol{X}^{(i)}}\mathrm{M}$ along the geodesic $\Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, t)$ to the tangent space $T_{\boldsymbol{X}^{(i+1)}}\mathrm{M}$ while maintaining the angle of $\boldsymbol{\Xi}$ to the geodesic. With this, and using the shorthand notation

$$\mathcal{T}_{\boldsymbol{\Xi}}^{(i+1)} := \mathcal{T}(\boldsymbol{\Xi}, \boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, \alpha^{(i)}), \tag{3.6}$$

the new search direction is computed by

$$\boldsymbol{H}^{(i+1)} = -\boldsymbol{G}^{(i+1)} + \beta^{(i)}\mathcal{T}_{\boldsymbol{H}^{(i)}}^{(i+1)}, \tag{3.7}$$

where $\beta^{(i)} \in \mathbb{R}$ is calculated by some CG update formula adopted to the manifold setting. The most commonly used update formulas are those by Fletcher-Reeves (FR), Hestenes-Stiefel (HS), Polak-Ribiere (PR), and Dai-Yuan (DY). With $\boldsymbol{Y}^{(i+1)} = \boldsymbol{G}^{(i+1)} - \mathcal{T}_{\boldsymbol{G}^{(i)}}^{(i+1)}$, they read as

$$\beta_{\mathrm{FR}}^{(i)} = \frac{\langle \boldsymbol{G}^{(i+1)}, \boldsymbol{G}^{(i+1)} \rangle}{\langle \boldsymbol{G}^{(i)}, \boldsymbol{G}^{(i)} \rangle}, \tag{3.8}$$

$$\beta_{\mathrm{HS}}^{(i)} = \frac{\langle \boldsymbol{G}^{(i+1)}, \boldsymbol{Y}^{(i+1)} \rangle}{\langle \mathcal{T}_{\boldsymbol{H}^{(i)}}^{(i+1)}, \boldsymbol{Y}^{(i+1)} \rangle}, \tag{3.9}$$

$$\beta_{\mathrm{PR}}^{(i)} = \frac{\langle \boldsymbol{G}^{(i+1)}, \boldsymbol{Y}^{(i+1)} \rangle}{\langle \boldsymbol{G}^{(i)}, \boldsymbol{G}^{(i)} \rangle}, \tag{3.10}$$

$$\beta_{\mathrm{DY}}^{(i)} = \frac{\langle \boldsymbol{G}^{(i+1)}, \boldsymbol{G}^{(i+1)} \rangle}{\langle \mathcal{T}_{\boldsymbol{H}^{(i)}}^{(i+1)}, \boldsymbol{Y}^{(i+1)} \rangle}. \tag{3.11}$$

Now, a solution to Problem (3.2) is computed by alternating between computing the search direction on M via (3.7), finding an appropriate step size by solving (3.4), and updating the current optimal point through (3.5) until some user-specified convergence criterion is met, or a maximum number of iterations has been reached.

As the functions considered here are non-quadratic terms, search directions found at consecutive iterations gradually loose conjugacy. Furthermore, for optimization problems over variables of dimension n CG can only produce n conjuagte vectors. To deal with theses issues, the search direction should be reset at least every n iterations to the steepest descent direction. This reset can be performed earlier for example every fixed $j < n$ iterations, if the function value does not decrease sufficiently, or when the gradients of the cost function at two consecutive iterations are not sufficiently orthogonal, cf. [98].

Chapter 4

Geometric Dictionary Learning

This chapter is partially based on the published work:

> S. Hawe, M. Seibert, and M. Kleinsteuber. Separable Dictionary Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438--445. 2013.

As already stated in previous chapters, many techniques in computer vision, machine learning, and statistics rely on the fact that a signal of interest admits a sparse representation over some dictionary. Dictionaries can either be composed analytically, or learned from a suitable set of example signals that represent the considered class. While analytic dictionaries permit to capture the global structure of a signal and often come along with a fast implementation, learned dictionaries are known to perform better in applications as they are more adapted to the considered class of signals. When we are dealing with images, unfortunately, memory limitations and the numerical burden for (i) learning a dictionary and for (ii) employing the dictionary for image processing tasks only allows working with relatively small image-patches that only contain local image information.

To overcome these two drawbacks of learned dictionaries, in this chapter, I present an algorithm that is able to learn dictionaries that have a separable matrix structure. This structure, on the one hand, permits to work with larger patch-sizes, and on the other hand, such dictionaries can be applied more efficiently in image processing tasks compared to unstructured dictionaries. The proposed learning procedure is based on an optimization process over a product of spheres manifold, which updates the dictionary as a whole, thus controls basic dictionary properties such as mutual coherence explicitly during the learning procedure. In simple words, the mutual coherence of $\boldsymbol{D}$ measures the similarity between the atoms of the dictionary. The presented learning scheme is not limited to find separable dictionaries but is also able to learn standard unstructured ones. In that case, the presented method competes with state-of-the-art dictionary learning methods like K-SVD.

4.1 Introduction

Exploiting the fact that a signal $\boldsymbol{s} \in \mathbb{R}^n$ has a sparse representation over some dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times d}$ is the backbone of many successful signal reconstruction and data analysis algorithms. Having a sparse representation means that $\boldsymbol{s}$ is the linear combination of only a few columns of $\boldsymbol{D}$, referred to as *atoms*. Formally, this reads as

$$\boldsymbol{s} = \boldsymbol{D}\boldsymbol{x}, \tag{4.1}$$

where the transform coefficient vector $\boldsymbol{x} \in \mathbb{R}^d$ is *sparse*, i.e. most of its entries are zero or small in magnitude. For the performance of algorithms exploiting this model, it is crucial to find a dictionary that allows the signal of interest to be represented most accurately with a coefficient vector $\boldsymbol{x}$ that is as sparse as possible. Basically, dictionaries can be assigned to two classes: *analytic dictionaries* and *learned dictionaries*. Analytic dictionaries are built on mathematical models of the general type of signal they should represent. They can be used universally and allow a fast implementation. Popular examples include Wavelets [81], Bandlets [72], and Curvlets [121] among several others.

It is well-known that learned dictionaries yield a sparser representation than analytic ones. Given a set of representative training signals, dictionary learning algorithms aim at finding the dictionary over which the training set admits a maximally sparse representation. Formally, let $\boldsymbol{S} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_m] \in \mathbb{R}^{n\times M}$ be the matrix containing the M training samples arranged as its columns, and let $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_m] \in \mathbb{R}^{d\times M}$ contain the corresponding M sparse transform coefficient vectors, then learning a dictionary can be stated as the minimization problem

$$\{\boldsymbol{X}^\star, \boldsymbol{D}^\star\} = \arg\min_{\boldsymbol{X},\boldsymbol{D}} g(\boldsymbol{X}) \quad \text{subject to} \quad \begin{array}{l} \|\boldsymbol{D}\boldsymbol{X} - \boldsymbol{S}\|_F^2 \leq \epsilon, \\ \boldsymbol{D} \in \mathfrak{C}. \end{array} \tag{4.2}$$

Therein, $g : \mathbb{R}^{d\times M} \to \mathbb{R}$ is a function that promotes sparsity, ϵ reflects the noise power, and $\mathfrak{C}$ is some predefined admissible set of solutions. Common dictionary learning approaches that employ optimization problems similar to (4.2) include probabilistic ones like [46, 71, 144], and clustering based ones such as K-SVD [3], see Chapter 2 or [126] for a more comprehensive overview. Dictionaries produced by these techniques are unstructured matrices that allow determining highly sparse representations of the associated signals of interest. However, the dimensions of the corresponding signals and consequently the possible dimensions of the dictionary are inherently restricted by limited memory and limited computational resources. Furthermore, those dictionaries are computationally expensive to be applied within signal reconstruction algorithms where many matrix vector multiplications have to be performed, particularly if the processed signals are large.

To overcome these drawbacks of learned dictionaries, in this chapter, I present a method for learning dictionaries that are efficiently applicable in image reconstruction tasks and that permit to work with larger patch sizes compared to those commonly used in other dictionary learning schemes. The crucial idea is to enforce the dictionary to have a separable matrix structure. Due to this, I dubbed the algorithm *SeDiL*, short form for *Se*parable *Di*ctionary *L*earning. Here, separable means that the dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ is given by the Kronecker product of two smaller dictionaries $\boldsymbol{C} \in \mathbb{R}^{h \times c}$ and $\boldsymbol{R} \in \mathbb{R}^{w \times r}$ with $r, c \ll d$ and $h, w \ll n$, i.e.

$$\begin{aligned} \boldsymbol{D} &= \boldsymbol{R} \otimes \boldsymbol{C} \\ &= \begin{bmatrix} r_{11}\boldsymbol{C} & \cdots & r_{1r}\boldsymbol{C} \\ \vdots & \ddots & \vdots \\ r_{w1}\boldsymbol{C} & \cdots & r_{wr}\boldsymbol{C} \end{bmatrix} \in \mathbb{R}^{n \times d}. \end{aligned} \tag{4.3}$$

The relation between a signal $\boldsymbol{s} \in \mathbb{R}^{hw}$ and its sparse representation $\boldsymbol{x} \in \mathbb{R}^{cr}$ according to (4.1) is then given as

$$\boldsymbol{s} = (\boldsymbol{R} \otimes \boldsymbol{C})\boldsymbol{x} = \operatorname{vec}(\boldsymbol{C} \operatorname{vec}^{-1}(\boldsymbol{x})\boldsymbol{R}^\top), \tag{4.4}$$

where the vector space isomorphism $\operatorname{vec}\colon \mathbb{R}^{c \times r} \to \mathbb{R}^{cr}$ is defined by stacking the columns of the considered matrix on top of each other. Employing this separable structure instead of a full unstructured dictionary clearly reduces the computational cost of both the learning algorithm as well as dictionary based signal reconstruction tasks. More precisely, for a separation with $h, w \sim \sqrt{n}$, the computational burden reduces from $O(n)$ to $O(\sqrt{n})$.

Clearly, the proposed scheme is in principle applicable to any class of signals. However, here the focus lies on signals that have an inherent two-dimensional structure such as images. To fix the notation for the rest of this work, if $\boldsymbol{C}$ and $\boldsymbol{R}$ are given as above, the two dimensional signal $\boldsymbol{S} \in \mathbb{R}^{h \times w}$ is given from its sparse representation $\boldsymbol{X} \in \mathbb{R}^{r \times c}$ via

$$\boldsymbol{S} = \boldsymbol{C}\boldsymbol{X}\boldsymbol{R}^\top. \tag{4.5}$$

The proposed dictionary learning scheme SeDiL is based on an adaption of Problem (4.2) to a product of unitary spheres and incorporates a regularization term that allows to control the mutual coherence of a learned dictionary. In contrast to most learning techniques, the dictionary and the sparse code are updated simultaneously. The arising optimization problem is solved by a Riemannian conjugate gradient method combined with a non-monotone line search technique adapted to the manifold setting. For the general separable case, the method is able to learn dictionaries from high dimensional signals where conventional learning techniques fail. To show that such a dictionary is able to extract and to recover the global information contained in the training data, a separable dictionary is learned on a face database with each face image having a resolution of (64×64) pixels. This dictionary is then applied in a face inpainting experiment where large missing regions are recovered solely based on the information contained in the dictionary.

Besides that, for the special case $\boldsymbol{R} = 1$, SeDiL yields a new algorithm for learning standard unstructured dictionaries. To evaluate the applicability of SeDiL in standard image processing tasks, I present a denoising experiment that shows the performance of both a separable and a non-separable dictionary learned by SeDiL on (8×8)-dimensional image-patches. I selected this task as denoising can be seen as a standard benchmark test that allows my method to be easily compared with existing dictionary learning techniques. From the achieved results it can be seen that the separable dictionary outperforms its analytic counterpart, the overcomplete discrete cosine transform, and that the non-separable one achieves similar performance as state-of-the-art learning methods like K-SVD.

Last, I like to mention that SeDiL can be straightforwardly extended to signals of even more dimensions, such as volumetric $3D$-signals, by employing multiple Kronecker products, i.e. one for each dimension.

4.2 Structured Dictionary Learning

Instead of learning dense unstructured dictionaries as described in Chapter 2, which are costly to apply in signal reconstruction tasks and that are unable to deal with higher dimensional signals, techniques exist that aim at learning dictionaries that bypass these limitations. This line of research is still in its early stage, and only few prior works in this direction exist. In the following, I shortly review some existing techniques that focus on learning efficiently applicable and high dimensional dictionaries. After that, I introduce my novel approach.

4.2.1 Prior Art

In [110], a method is proposed for learning a dictionary such that each atom itself is sparse over some fixed analytic base dictionary. Employing a sparse dictionary in signal processing tasks requires less arithmetic operations compared to applying a dense dictionary, hence, it is computationally more efficient. The concrete learning algorithm extends the famous K-SVD approach by introducing an additional prior that enforces the sparsity constraint on the atoms of the dictionary. As this algorithm is based on the K-SVD method, however, it is not capable to handle higher dimensional signals.

An approach called compressible dictionary learning, which also follows the idea of finding a dictionary that is sparse over a given base dictionary, has been introduced in [136]. In contrast to [110], sparsity is enforced globally over the entire dictionary rather than enforcing a fixed level of sparsity over each atom. The authors state that this model is less restrictive as compared to the method of [110]. Again, this approach cannot deal with high dimensional signals.

In [2] an alternative structure for dictionaries called signature dictionary has been proposed. The structure can be interpreted as a small image, where every patch at every possible location and size is a potential dictionary atom. The advantages of this structure include near-translation-invariance, reduced overfitting to the training set, and less memory- and computational requirements compared to unstructured dictionaries. However, the small number of parameters in this model makes this dictionary more restrictive than other structures. This approach has been further extended in [9] to learn real translational-invariant atoms.

Hierarchical frameworks for tackling the problem of learning high dimensional dictionaries are presented in [67] and [134]. The latter work uses this framework in conjunction with a screening technique and random projections. I like to mention that SeDiL has the potential to be combined with such hierarchical frameworks.

Developed in parallel to the approach presented here and even published at the same conference, a method that aims at learning separable filters has been introduced [103]. By a filter, the authors understand an atom of a dictionary that has been reshaped to a matrix whose size is equal to the 2D signal it has to be applied to, e.g. the patch size in image processing. Because a separable filter can be interpreted as a rank-1 matrix, the basic idea is to enforce the separability by minimizing the rank of the filters followed by a thresholding operation on the singular values of the filters to find exact rank-1 matrices. To that end, two conceptually different learning algorithms have been suggested. The first algorithm is based on a standard convolutional ℓ_1-based dictionary learning task, augmented by an additional penalty term that enforces the nuclear norm of each filter to be minimal. The nuclear norm of a matrix corresponds to the sum of its singular values and is the closest convex relaxation of the matrix rank. Consequently, enforcing the nuclear norm of a matrix to be small amounts to finding matrices with low rank. The arising optimization problem is solved by a stochastic gradient descent scheme. The second algorithm first learns a standard dictionary and afterwards tries to approximate each of the atoms of this dictionary by a linear combination of a few separable filters, which are again found by a nuclear norm minimization. In both approaches, the final separable filters, i.e. filters with rank one, are determined in a post processing step by computing the singular values of a learned filter and setting all of them but the largest to zero. An advantage of the second approach is that it can be applied to any existing set of filters and that it can be combined with any dictionary learning technique. In contrast to this approach, here, I directly aim at learning rank-1 filters as the product of two 1D filters without requiring a thresholding operation.

Last, I want to mention two different algorithms proposed in [118] and [6] that similar to my proposed method control the mutual coherence of a dictionary during the learning processes. In [118], this is achieved via a regularization term introduced into the dictionary update step. Concretely, to update the dictionary the algorithm solves

$$\boldsymbol{D} = \arg\min_{\boldsymbol{D}} \sum_i \|\boldsymbol{D}\boldsymbol{x}_i - \boldsymbol{s}_i\|_2^2 + \lambda \|\boldsymbol{D}^\top \boldsymbol{D} - \boldsymbol{I}_d\|_F^2, \tag{4.6}$$

where the latter term influences the mutual coherence of the dictionary. The higher $\lambda \in \mathbb{R}^+$ is chosen, the lower the mutual coherence gets. The Problem (4.6) is solved by a limited BFGS algorithm. Similar to the penalty function used in SeDiL, this term reduces the average angle between all atoms, but in contrast to SeDiL it does not necessarily avoid two or more atoms to be completely identical. In [6], the mutual coherence is controlled by decorrelating the learned atoms via an iterative projection method that is complemented by a rotation of the dictionary. The rotation step, which can be computed in closed form, is done in order to reduce the approximation error $\sum_i \|\boldsymbol{D}\boldsymbol{x}_i - \boldsymbol{s}_i\|_2^2$ of the resulting decorrelated dictionary. Basically, this technique can be combined with any dictionary learning method and is performed either after the dictionary has been fully trained, or at each step of any iterative learning approach.

4.2.2 Proposed Approach

The dictionary learning technique proposed here adapts Problem (4.2) to the separable dictionary case as follows. The separable dictionary is denoted by $\boldsymbol{D} = \boldsymbol{R} \otimes \boldsymbol{C}$ and the goal is to learn $\boldsymbol{R}$ and $\boldsymbol{C}$ from a representative set of training samples $\mathcal{S} = (\boldsymbol{S}_1, \dots, \boldsymbol{S}_M) \in \mathbb{R}^{h \times w \times M}$. The set of sparse representations of all M samples is $\mathcal{X} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_M)$ and I measure the overall sparsity via

$$g(\mathcal{X}) := \sum_{j=1}^{m} \sum_{k=1}^{c} \sum_{l=1}^{r} \ln(1 + \rho |x_{klj}|^2), \tag{4.7}$$

where x_{klj} denotes the (k, l)-entry of $\boldsymbol{X}_j \in \mathbb{R}^{c \times r}$, and $\rho \in \mathbb{R}^+$ is a smoothing parameter. This non-convex sparsity measure is commonly employed in the literature and more closely resembles the ℓ_0-pseudo-norm as compared to the convex ℓ_1-norm sparsity measure.

Besides enforcing the separable matrix structure, the second important ingredient of my approach is the admissible set of solutions $\mathfrak{C}$, which imposes the following constraints on the dictionary.

(i) The *columns* of $\boldsymbol{D}$ have unit Euclidean norm, i.e. $\|\boldsymbol{d}_{:,i}\|_2 = 1$ for $i = 1, \dots, d$.

(ii) The *coherence* of $\boldsymbol{D}$ shall be moderate.

Constraint (i) is commonly employed in various dictionary learning procedures to avoid the scale ambiguity problem, i.e. getting entries of $\boldsymbol{D}$ that tend to infinity, while the entries of $\mathcal{X}$ tend to zero as this is the global minimizer of the sparsity measure $g(\mathcal{X})$. Matrices with normalized columns admit a manifold structure, known as the product of unit spheres, which is formally defined as

$$\mathrm{S}(n, d) := \{\boldsymbol{D} \in \mathbb{R}^{n \times d} | \operatorname{ddiag}(\boldsymbol{D}^\top \boldsymbol{D}) = \boldsymbol{I}_d\}. \tag{4.8}$$

Here, $\operatorname{ddiag}(\boldsymbol{Z})$ forms a diagonal matrix with the diagonal entries of the square matrix $\boldsymbol{Z}$. As the Kronecker product of two matrices with normalized columns again results in a matrix with normalized columns, and as the proposed method should be able to learn both structured and unstructured dictionaries I require that $\boldsymbol{C}$ is an element of $\mathrm{S}(h, c)$ and that $\boldsymbol{R}$ is an element of $\mathrm{S}(w, r)$.

The soft constraint (ii), which requires the mutual coherence of the dictionary to be moderate, is a well-known requirement introduced in dictionary learning methods, and is motivated by the theory of Compressive Sensing [35]. Roughly speaking, the mutual coherence of $\boldsymbol{D}$ measures the similarity between the atoms of the dictionary, or, "*a value that exposes the dictionary's vulnerability, as [...] two closely related columns may confuse any pursuit technique.*" [41]. The most common mutual coherence measure for a dictionary with normalized columns is given by

$$\mu(\boldsymbol{D}) := \max_{i<j} |\boldsymbol{d}_{:,i}^{\top}\boldsymbol{d}_{:,j}|. \tag{4.9}$$

This is a worst case measure, which has been relaxed by other measures that are better suited for practical purposes. For example, in [34, 41, 128] the coherence is measured as the average of the absolute values of all inner products between distinct atoms of $\boldsymbol{D}$ that are above a user defined threshold $t \in \mathbb{R}^+$, and [38] considers the sum of squares of all elements in $\{|\boldsymbol{d}_{:,i}^{\top}\boldsymbol{d}_{:,j}| \mid i < j\}$. Here, I introduce an alternative mutual coherence measure, which has been proven extremely useful in image processing applications and is explicitly given as

$$r(\boldsymbol{D}) := -\sum_{1\leq i<j\leq d} \ln\left(1 - (\boldsymbol{d}_{:,i}^{\top}\boldsymbol{d}_{:,j})^2\right). \tag{4.10}$$

Since this measure is differentiable, it can be integrated into smooth optimization procedures. Furthermore, when it is used within a dictionary learning scheme, the log-barrier function avoids the algorithm from producing dictionaries that contain useless mutually identical atoms.

To justify the proposed measure, note that minimizing $r(\boldsymbol{D})$ implicitly influences the well-known worst case measure $\mu(\boldsymbol{D})$. Concretely, the relation between (4.10) and the classical mutual coherence (4.9) is given by

$$\tfrac{1}{N_{\boldsymbol{D}}} r(\boldsymbol{D}) \;\leq\; -\ln(1-\mu(\boldsymbol{D})^2) \;\leq\; r(\boldsymbol{D}), \tag{4.11}$$

with $N_{\boldsymbol{D}} := \frac{d(d-1)}{2}$ denoting the number of summands in Equation (4.9). To see the validity of Equation (4.11), note that since all atoms are normalized to unit Euclidean norm, the equation $0 \leq |\boldsymbol{d}_{:,i}^{\top}\boldsymbol{d}_{:,j}|^2 \leq 1$ holds due to the Cauchy-Schwarz inequality. Consequently, all summands $-\ln(1-(\boldsymbol{d}_{:,i}^{\top}\boldsymbol{d}_{:,j})^2)$ are non-negative. Moreover,

$$\max_{i<j}(-\ln(1-(\boldsymbol{d}_{:,i}^{\top}\boldsymbol{d}_{:,j})^2)) = -\ln(1-\mu(\boldsymbol{D})^2), \tag{4.12}$$

and therefore

$$-\ln(1-\mu(\boldsymbol{D})^2) \;\leq\; r(\boldsymbol{D}) \;\leq\; -N_{\boldsymbol{D}}\ln(1-\mu(\boldsymbol{D})^2), \tag{4.13}$$

which implies Equation (4.11). Now, to exploit this relation for the desired separable dictionary case, first consider the following lemma.

Lemma 4.1. *The mutual coherence of the Kronecker product of two matrices $\boldsymbol{C}$ and $\boldsymbol{R}$ with normalized columns is equal to the maximum of the individual mutual coherences, i.e.*

$$\mu(\boldsymbol{R} \otimes \boldsymbol{C}) = \max\{\mu(\boldsymbol{C}), \mu(\boldsymbol{R})\}. \tag{4.14}$$

Proof. First, notice that since the columns of $\boldsymbol{C}$ and $\boldsymbol{R}$ all have unit Euclidean norm, the diagonal entries of both $\boldsymbol{C}^\top \boldsymbol{C}$ and $\boldsymbol{R}^\top \boldsymbol{R}$ are equal to one, thus, the mutual coherence $\mu(\boldsymbol{C})$ and $\mu(\boldsymbol{R})$ is given by the largest off-diagonal absolute value of the Gramian matrices $\boldsymbol{C}^\top \boldsymbol{C}$ and $\boldsymbol{R}^\top \boldsymbol{R}$, respectively. Analogously, $\mu(\boldsymbol{R} \otimes \boldsymbol{C})$ is simply the largest off-diagonal absolute value of the matrix $(\boldsymbol{R} \otimes \boldsymbol{C})^\top (\boldsymbol{R} \otimes \boldsymbol{C}) = (\boldsymbol{R}^\top \boldsymbol{R}) \otimes (\boldsymbol{C}^\top \boldsymbol{C})$. Due to the definition of the Kronecker product and the unit diagonal of the Gramian matrices, each entry of $\boldsymbol{R}^\top \boldsymbol{R}$ and $\boldsymbol{C}^\top \boldsymbol{C}$ reappears in the off-diagonal entries of $(\boldsymbol{R} \otimes \boldsymbol{C})^\top (\boldsymbol{R} \otimes \boldsymbol{C})$. This yields the two inequalities $\mu(\boldsymbol{R}) \leq \mu(\boldsymbol{R} \otimes \boldsymbol{C})$ and $\mu(\boldsymbol{C}) \leq \mu(\boldsymbol{R} \otimes \boldsymbol{C})$, which can be combined to

$$\max\{\mu(\boldsymbol{C}), \mu(\boldsymbol{R})\} \leq \mu(\boldsymbol{R} \otimes \boldsymbol{C}). \tag{4.15}$$

On the other hand, each entry of $(\boldsymbol{R}^\top \boldsymbol{R}) \otimes (\boldsymbol{C}^\top \boldsymbol{C})$ is a product of the entries of $\boldsymbol{R}^\top \boldsymbol{R}$ and $\boldsymbol{C}^\top \boldsymbol{C}$. This explicitly means that we can write $\mu(\boldsymbol{R} \otimes \boldsymbol{C}) = \tilde{r}\,\tilde{c}$, with $\tilde{r}$ and $\tilde{c}$ being entries of the Gramian matrices $\boldsymbol{R}^\top \boldsymbol{R}$ and $\boldsymbol{C}^\top \boldsymbol{C}$, respectively. Since we have $0 \leq \tilde{c}, \tilde{r} \leq 1$, this provides the two inequalities $\mu(\boldsymbol{R} \otimes \boldsymbol{C}) \leq \tilde{r}$ and $\mu(\boldsymbol{R} \otimes \boldsymbol{C}) \leq \tilde{c}$, and hence

$$\mu(\boldsymbol{R} \otimes \boldsymbol{C}) \leq \max\{\mu(\boldsymbol{C}), \mu(\boldsymbol{R})\}. \tag{4.16}$$

Combining Equation (4.15) and Equation (4.16) provides the desired result. ■

Now, substituting $\mu(\boldsymbol{R} \otimes \boldsymbol{C})$ into Equation (4.11) and then applying Lemma 4.1 yields

$$\max\{\tfrac{1}{N_R} r(\boldsymbol{R}),\, \tfrac{1}{N_C} r(\boldsymbol{C})\} \;\leq\; -\ln(1 - \mu(\boldsymbol{R} \otimes \boldsymbol{C})^2) \;\leq\; \max\{r(\boldsymbol{R}),\, r(\boldsymbol{C})\}, \tag{4.17}$$

which holds due to the monotone behavior of the logarithm. Therefore, if $\max\{r(\boldsymbol{R}), r(\boldsymbol{C})\}$ is small, then $\mu(\boldsymbol{R} \otimes \boldsymbol{C})$ is bounded as well. To learn a separable dictionary $\boldsymbol{D} = \boldsymbol{R} \otimes \boldsymbol{C}$ with moderate mutual coherence, I exploit the relation

$$c_1(r(\boldsymbol{R}) + r(\boldsymbol{C})) \;\leq\; \max\{r(\boldsymbol{R}),\, r(\boldsymbol{C})\} \;\leq\; c_2(r(\boldsymbol{R}) + r(\boldsymbol{C})), \tag{4.18}$$

for some positive constants $c_1, c_2 \in \mathbb{R}^+$, and minimize $r(\boldsymbol{R}) + r(\boldsymbol{C})$ instead of $\max\{r(\boldsymbol{R}),\ r(\boldsymbol{C})\}$ as this is computationally easier to handle.

Putting all the collected ingredients together, the cost function I want to minimize to learn a separable dictionary reads as

$$\begin{aligned} f\colon\ & \mathbb{R}^{c \times r \times M} \times \mathrm{S}(h,c) \times \mathrm{S}(w,r) \to \mathbb{R}, \\ & (\mathcal{X}, \boldsymbol{C}, \boldsymbol{R}) \mapsto \frac{1}{2M} \sum_{j=1}^{M} \|\boldsymbol{C}\boldsymbol{X}_j\boldsymbol{R}^\top - \boldsymbol{S}_j\|_F^2 + \frac{\lambda}{M} g(\mathcal{X}) + \kappa(r(\boldsymbol{C}) + r(\boldsymbol{R})). \end{aligned} \tag{4.19}$$

Therein, $\lambda \in \mathbb{R}^+$ weighs between the sparsity of $\mathcal{X}$ and how accurately $\{\boldsymbol{C}\boldsymbol{X}_j\boldsymbol{R}^\top\}_{j=1}^M$ reproduce the training samples. By adjusting this parameter, both perfect noise free training data as well as noisy training data can be handles by SeDiL. The second weighting factor $\kappa \in \mathbb{R}^+$ controls the mutual coherence of the learned dictionary. The higher it is chosen, the lower the dictionary's mutual coherence gets.

4.3 Separable Dictionary Learing (SeDiL)

Knowing that the feasible set of solutions to Problem (4.19) is restricted to a smooth manifold allows one to apply the concepts explained in Chapter 3 to learn the dictionary. In the following, I concretize the concepts for the situation at hand and first present the geometry of the considered problem followed by introducing the ingredients that are necessary to implement the geometric dictionary learning algorithm. The presented formulas regarding the geometry of $\mathrm{S}(n,d)$ are derived in detail in [1].

4.3.1 Geometry of the Problem

In the proposed dictionary learning scheme, the considered manifold is a product manifold given as

$$\mathrm{M} := \mathbb{R}^{c \times r \times M} \times \mathrm{S}(h,c) \times \mathrm{S}(w,r), \tag{4.20}$$

which is a Riemannian submanifold of $\mathbb{R}^{c \times r \times M} \times \mathbb{R}^{h \times c} \times \mathbb{R}^{w \times r}$. In the following, I denote an element of M by $\mathcal{Y} := (\mathcal{X}, \boldsymbol{C}, \boldsymbol{R})$.

Due to the product structure of M, the tangent space of M at a point $\mathcal{Y} \in \mathrm{M}$ is simply the product of all individual tangent spaces, i.e.

$$T_{\mathcal{Y}}\,\mathrm{M} := \mathbb{R}^{c \times r \times M} \times T_{\boldsymbol{C}}\,\mathrm{S}(h, c) \times T_{\boldsymbol{R}}\,\mathrm{S}(w, r). \tag{4.21}$$

Consequently, the orthogonal projection of some arbitrary point $\mathcal{Q} := (\mathcal{Q}_1, \boldsymbol{Q}_2, \boldsymbol{Q}_3) \in \mathbb{R}^{c \times r \times M} \times \mathbb{R}^{h \times c} \times \mathbb{R}^{w \times r}$ onto the tangent space $T_{\mathcal{Y}}\,\mathrm{M}$ is

$$\Pi_{T_{\mathcal{Y}}\,\mathrm{M}}(\mathcal{Q}) := (\mathcal{Q}_1, \Pi_{T_{\boldsymbol{C}}\,\mathrm{S}(h,c)}(\boldsymbol{Q}_2), \Pi_{T_{\boldsymbol{R}}\,\mathrm{S}(w,r)}(\boldsymbol{Q}_3)). \tag{4.22}$$

In concrete formulas, for the applied product of unit spheres manifold the tangent space at a point $\boldsymbol{D} \in \mathrm{S}(n, d)$ is given by

$$T_{\boldsymbol{D}}\,\mathrm{S}(n, d) := \{\Xi \in \mathbb{R}^{n \times d} | \operatorname{ddiag}(\boldsymbol{D}^\top \Xi) = \boldsymbol{0}_{d \times d}\}, \tag{4.23}$$

and the associated orthogonal projection of some matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$ onto the tangent space (4.23) reads as

$$\Pi_{T_{\boldsymbol{D}}\,\mathrm{S}(n,d)}(\boldsymbol{Q}) := \boldsymbol{Q} - \boldsymbol{D} \operatorname{ddiag}(\boldsymbol{D}^\top \boldsymbol{Q}). \tag{4.24}$$

Each tangent space of M is endowed with the Riemannian metric inherited from the surrounding Euclidean space, which for two points $\mathcal{Q} = (\mathcal{Q}_1, \boldsymbol{Q}_2, \boldsymbol{Q}_3) \in T_{\mathcal{Y}}\,\mathrm{M}$ and $\mathcal{P} = (\mathcal{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3) \in T_{\mathcal{Y}}\,\mathrm{M}$ is given by

$$\langle \mathcal{Q}, \mathcal{P} \rangle := \sum_{j=1}^{M} \operatorname{tr}(\boldsymbol{Q}_{1,j}^\top \boldsymbol{P}_{1,j}) + \operatorname{tr}(\boldsymbol{Q}_2^\top \boldsymbol{P}_2) + \operatorname{tr}(\boldsymbol{Q}_3^\top \boldsymbol{P}_3). \tag{4.25}$$

Next, a way to compute geodesics on the considered manifold is required. While in general there is no closed form solution to that problem, the case at hand allows for an efficient implementation.

Let $\boldsymbol{d} \in \mathrm{S}^{n-1}$ be a point on a unit sphere and let $\boldsymbol{h} \in T_{\boldsymbol{d}}\,\mathrm{S}^{n-1}$ be a tangent vector at $\boldsymbol{d}$, then the geodesic in the direction of $\boldsymbol{h}$ is a great circle given as

$$\gamma(\boldsymbol{d}, \boldsymbol{h}, t) := \begin{cases} \boldsymbol{d}, & \text{if } \|\boldsymbol{h}\|_2 = 0 \\ \boldsymbol{d} \cos(t\|\boldsymbol{h}\|_2) + \boldsymbol{h} \frac{\sin(t\|\boldsymbol{h}\|_2)}{\|\boldsymbol{h}\|_2}, & \text{otherwise.} \end{cases} \tag{4.26}$$

Using this, the geodesic through $\boldsymbol{D} \in \mathrm{S}(n, d)$ in the direction of $\boldsymbol{H} \in T_{\boldsymbol{D}}\,\mathrm{S}(n, d)$ is simply the combination of the great circles emerging from each column of $\boldsymbol{D}$ in the direction of the corresponding column of $\boldsymbol{H}$, i.e.

$$\Gamma_{\mathrm{S}(n,d)}(\boldsymbol{D},\boldsymbol{H},t) := [\gamma(\boldsymbol{d}_{:,1},\boldsymbol{h}_{:,1},t),\dots,\gamma(\boldsymbol{d}_{:,d},\boldsymbol{h}_{:,d},t)]. \tag{4.27}$$

Now, let $\mathcal{H} = (\mathcal{H}_1,\boldsymbol{H}_2,\boldsymbol{H}_3) \in T_{\mathcal{Y}}\,\mathrm{M}$ be a given search direction on the considered product manifold, due to this product structure a geodesic on M is given by

$$\Gamma_{\mathrm{M}}(\mathcal{Y},\mathcal{H},t) = (\mathcal{X} + t\mathcal{H}_1,\ \Gamma_{\mathrm{S}(h,c)}(\boldsymbol{C},\boldsymbol{H}_2,t),\ \Gamma_{\mathrm{S}(w,r)}(\boldsymbol{R},\boldsymbol{H}_3,t)). \tag{4.28}$$

To solve optimization problem (4.19), I employ a geometric conjugate gradient method as explained in Section 3.2 adapted to the problem at hand. To that end, we need to understand how the parallel transport $\mathcal{T}_{\mathrm{M}}$ between two tangent spaces is performed for the examined manifold (4.21). Therefore, again first consider the geometry of S^{n-1}, for which the parallel transport of a tangent vector $\boldsymbol{\xi} \in T_{\boldsymbol{d}}\,\mathrm{S}^{n-1}$ along the great circle $\gamma(\boldsymbol{d},\boldsymbol{h},t)$ reads as

$$\tau(\boldsymbol{\xi},\boldsymbol{d},\boldsymbol{h},t) := \boldsymbol{\xi} - \frac{\boldsymbol{\xi}^\top\boldsymbol{h}}{\|\boldsymbol{h}\|_2^2}\Big(\boldsymbol{d}\|\boldsymbol{h}\|_2\sin(t\|\boldsymbol{h}\|_2) + \boldsymbol{h}(1-\cos(t\|\boldsymbol{h}\|_2))\Big). \tag{4.29}$$

Again, due to the product structure of $\mathrm{S}(n,d)$ the parallel transport of $\boldsymbol{\Xi} \in T_{\boldsymbol{D}}\,\mathrm{S}(n,d)$ along the geodesic $\Gamma_{\mathrm{S}(n,d)}(\boldsymbol{D},\boldsymbol{H},t)$ is given by

$$\mathcal{T}_{\mathrm{S}(n,d)}(\boldsymbol{\Xi},\boldsymbol{D},\boldsymbol{H},t) := \Big[\tau(\boldsymbol{\xi}_{:,1},\boldsymbol{d}_{:,1},\boldsymbol{h}_{:,1},t),\dots,\tau(\boldsymbol{\xi}_{:,d},\boldsymbol{d}_{:,d},\boldsymbol{h}_{:,d},t)\Big]. \tag{4.30}$$

Thus, a tangent space element $\Xi := (\Xi_1,\boldsymbol{\Xi}_2,\boldsymbol{\Xi}_3) \in T_{\mathcal{Y}}\,\mathrm{M}$ is transported in the direction of $\mathcal{H} \in T_{\mathcal{Y}}\,\mathrm{M}$ via

$$\mathcal{T}_{\mathrm{M}}(\Xi,\mathcal{Y},\mathcal{H},t) :=\ (\Xi_1,\mathcal{T}_{\mathrm{S}(h,c)}(\boldsymbol{\Xi}_2,\boldsymbol{C},\boldsymbol{H}_2,t),\mathcal{T}_{\mathrm{S}(w,r)}(\boldsymbol{\Xi}_3,\boldsymbol{R},\boldsymbol{H}_3,t)). \tag{4.31}$$

4.3.2 Implementation

To employ a CG optimization method, one first requires a way to compute the CG-update parameter $\beta^{(i)}$. To that end, several update formulas exist that lead to different behaviors in applications and that have different theoretical convergence properties. Here, I employ a hybridization of the Hestenes-Stiefel (HS) formula (3.9) and the Dai Yuan (DY) formula (3.11), which is given by

$$\beta^{(i)}_{\mathrm{hyb}} = \max\left(0, \min(\beta^{(i)}_{\mathrm{DY}},\beta^{(i)}_{\mathrm{HS}})\right). \tag{4.32}$$

This update formula has been suggested in [27] and as explained therein combines the good numerical performance of HS with the desirable global convergence properties of DY.

To find an appropriate step size $\alpha^{(i)}$, I propose a Riemannian adaptation of the non-monotone line search algorithm introduced in [143]. Standard monotone line search techniques find the step size such that, at each iteration, the cost function decreases, whereas non-monotone line search methods permit some temporary increase in the function value. By that, those methods have the potential to improve the likelihood of finding a global minimum of a non-convex optimization problem as well as to increase the convergence speed, cf. [26]. This is especially useful in the present case, as the cost function is highly non-convex. Most non-monotone line search techniques determine the bound of sufficient decrease by taking the maximum value of the cost function over the last k iterations. In contrast to that, the method employed here utilizes a convex combination of the function values from *all* previous iterations. To employ this method in my setting, some straightforward modifications have to be made to the original algorithm such that it operates along the geodesic $\Gamma_{\mathrm{M}}(\mathcal{Y}, \mathcal{H}, t)$ instead of a straight line, see Algorithm 4.1 for the pseudocode. The line search

Algorithm 4.1 Non-Monotone Line Search on M at the *i*-th Iteration

```
Input: $t_0^{(i)} > 0$, $0 < c_1 < 1, 0 < c_2 < 0.5$ $0 \leq \eta^{(i)} \leq 1$, $Q^{(i)}$, $C^{(i)}$
Set: $t \leftarrow t_0^{(i)}$
  while $f(\Gamma_{\mathrm{M}}(\mathcal{Y}^{(i)}, \mathcal{H}^{(i)}, t)) > C^{(i)} + c_2 t \langle \mathcal{G}^{(i)}, \mathcal{H}^{(i)} \rangle$ do
    $t \leftarrow c_1 t$
  end while
Set: $Q^{(i+1)} \leftarrow \eta^{(i)} Q^{(i)} + 1$,
    $C^{(i+1)} \leftarrow \frac{\eta^{(i)} Q^{(i)} C^{(i)} + f\left(\Gamma_{\mathrm{M}}(\mathcal{Y}^{(i)}, \mathcal{H}^{(i)}, t)\right)}{Q^{(i+1)}}$
    $\alpha^{(i)} \leftarrow t$,
Output: $\alpha^{(i)}$, $Q^{(i+1)}$, $C^{(i+1)}$
```

is initialized with $C^{(0)} = f(\mathcal{Y}^{(0)})$ and $Q^{(0)} = 1$.

Finally, let $\boldsymbol{E}_{ij}$ denote a square matrix whose i-th entry in the j-th column is equal to one and all other entries being zero, the concrete formulas for the Euclidean gradient

$$\nabla f(\mathcal{Y}) = \left(\tfrac{\partial}{\partial \mathcal{X}} f, \tfrac{\partial}{\partial \boldsymbol{C}} f, \tfrac{\partial}{\partial \boldsymbol{R}} f \right) \tag{4.33}$$

of the considered cost function (4.19) are

$$\frac{\partial}{\partial \mathcal{X}} f = \left\{ \frac{1}{M} \boldsymbol{C}^\top (\boldsymbol{C}\boldsymbol{X}_j \boldsymbol{R}^\top - \boldsymbol{S}_j)\boldsymbol{R} + \frac{\lambda}{M} \frac{\partial}{\partial \boldsymbol{X}_j} g(\boldsymbol{X}_j) \right\}_{j=1}^{M}, \tag{4.34}$$

$$\frac{\partial}{\partial \boldsymbol{C}} f = \frac{1}{M} \sum_{j=1}^{M} (\boldsymbol{C}\boldsymbol{X}_j \boldsymbol{R}^\top - \boldsymbol{S}_j)\boldsymbol{R}\boldsymbol{X}_j^\top + \kappa \frac{\partial}{\partial \boldsymbol{C}} r(\boldsymbol{C}), \tag{4.35}$$

$$\frac{\partial}{\partial \boldsymbol{R}} f = \frac{1}{M} \sum_{j=1}^{M} (\boldsymbol{C}\boldsymbol{X}_j \boldsymbol{R}^\top - \boldsymbol{S}_j)^\top \boldsymbol{C}\boldsymbol{X}_j + \kappa \frac{\partial}{\partial \boldsymbol{R}} r(\boldsymbol{R}), \tag{4.36}$$

with

$$\frac{\partial}{\partial \boldsymbol{X}} g(\boldsymbol{X}) = 2 \sum_{k=1}^{c} \sum_{l=1}^{r} \frac{\rho x_{kl}}{1 + \rho x_{kl}^2} \boldsymbol{E}_{kl} \tag{4.37}$$

being the gradient of the sparsity promoting function (4.7) with respect to $\boldsymbol{X}$, and

$$\frac{\partial}{\partial \boldsymbol{D}} r(\boldsymbol{D}) = \boldsymbol{D} \sum_{1 \le i < j \le d} \frac{2 \boldsymbol{d}_{:,i}^\top \boldsymbol{d}_{:,j}}{1 - (\boldsymbol{d}_{:,i}^\top \boldsymbol{d}_{:,j})^2} (\boldsymbol{E}_{ij} + \boldsymbol{E}_{ji}). \tag{4.38}$$

being the gradient of the logarithmic barrier function (4.10). From this, the corresponding Riemannian gradient is simply computed as given in Equation (3.3), using the orthogonal projection (4.22). For legibility, the shorthand notation $\mathcal{G}^{(i)} := \mathcal{G}(\mathcal{Y}^{(i)})$ will be used throughout the rest of this chapter to denote the Riemannian gradient computed at the i-th iteration.

The complete SeDiL method that permits to learn a dictionary with a separable matrix structure is summarized in Algorithm 4.2. ,

4.4 Experiments

4.4.1 Patch-Based Image Denoising

To show how dictionaries learned via SeDiL perform in real applications, I present the results achieved for denoising images corrupted by additive white Gaussian noise of different standard deviation σ_{noise}. I chose this application as it can be seen as a standard benchmark for determining the performance of a dictionary and allows my method to be easily compared with existing dictionary learning techniques. The images and the noise levels chosen here are an excerpt of those commonly used in the literature. The peak signal-to-noise ratio (PSNR) between the ground truth image $\operatorname{vec}(\boldsymbol{S}) \in \mathbb{R}^N$ and the recovered image

Algorithm 4.2 Separable Dictionary Learning (SeDiL)

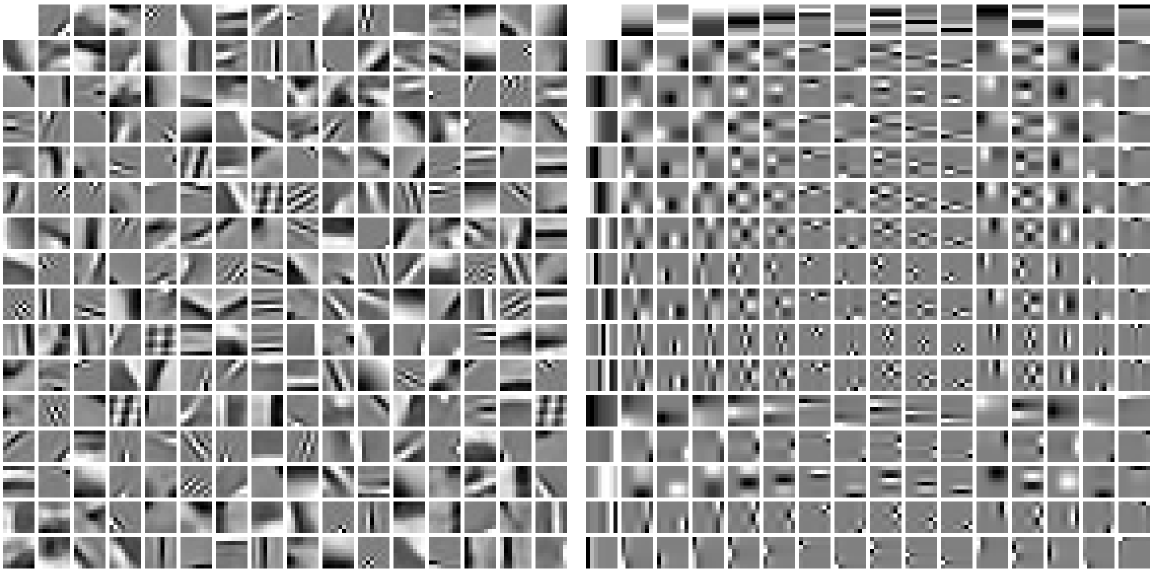
```
Input: Initial dictionaries $C^{(0)} \in \mathrm{S}(h,c), R^{(0)} \in \mathrm{S}(w,r)$, training data $\mathcal{S} \in \mathbb{R}^{h \times w \times M}$, parameters $\rho, \lambda, \kappa$, thresh
Set: $i \leftarrow 0, \mathcal{Y}^{(0)} \leftarrow (\{C^{(0)} S_j R^{(0)\top}\}_{j=1}^{M}, C^{(0)}, R^{(0)}), \mathcal{G}^{(0)} \leftarrow \Pi_{T_{\mathcal{Y}^{(0)}}\mathrm{M}}(\nabla f(\mathcal{Y}^{(0)})), \mathcal{H}^{(0)} \leftarrow -\mathcal{G}^{(0)}$
  repeat
    Compute $\alpha^{(i)}$, $Q^{(i+1)}$, $C^{(i+1)}$ according to Algorithm 4.1 in conjunction with Equation (4.19)
    $\mathcal{Y}^{(i+1)} \leftarrow \Gamma_{\mathrm{M}}(\mathcal{Y}^{(i)}, \mathcal{H}^{(i)}, \alpha^{(i)})$, cf. (4.28)
    $\mathcal{G}^{(i+1)} \leftarrow \Pi_{T_{\mathcal{Y}^{(i+1)}}\mathrm{M}}(\nabla f(\mathcal{Y}^{(i+1)}))$, cf. (4.34)-(4.36),(4.22)
    $\mathcal{H}^{(i+1)} \leftarrow -\mathcal{G}^{(i+1)} + \beta_{hyb}^{(i)} \mathcal{T}_{\mathcal{H}^{(i)}}^{(i+1)}$, cf. (4.32), (4.31)
    $i \leftarrow i + 1$
  until $\|\mathcal{G}^{(i)}\|_2 <$ thresh $\vee\ i =$ maximum # iterations
Output: $\mathcal{Y}^{\star} \leftarrow \mathcal{Y}^{(i)}$
```

$\mathrm{vec}(S^{\star}) \in \mathbb{R}^N$ computed by

$$\mathrm{PSNR} = 10 \ln \left(\frac{255^2 N}{\sum_{i=1}^{N} (s_i - s_i^{\star})^2} \right) \tag{4.39}$$

is used to quantify the reconstruction quality. As an additional quality measure, I use the Mean Structural SIMilarity Index (MSSIM) computed with the same set of parameters as originally suggested in [130]. The MSSIM ranges between zero and one, with one meaning perfect image reconstruction. Compared to the PSNR, the MSSIM better reflects the subjective visual impression of quality.

I present the denoising performance of both a universal unstructured dictionary, i.e. $D_1 = 1 \otimes C$, and a universal separable dictionary $D_2 = R \otimes C$, both learned from the same training data using SeDiL. By universal, I mean that the dictionary is not specifically learned for a certain image or image class but is universally applicable to any images showing natural scenes. Without loss of generality, I worked on square image-patches with $w = h = 8$, which is in accordance to the patch-sizes mostly used in the literature and learned four times overcomplete dictionaries. For the unstructured dictionary, this results in $c = 4wh$, and for the separable one it leads to $c = r = 2w$, i.e. C and R are of equal dimensions and $D_2 = R \otimes C \in \mathbb{R}^{4wh \times wh}$ is of the same size as its unstructured counterpart D_1. For the training phase, 40 000 image-patches were extracted from four different example images at random positions. Of course, these images are not considered further within the performance evaluations. The training patches were normalized to have zero mean and unit ℓ_2-norm. I initialized $\boldsymbol{C}$ and $\boldsymbol{R}$ with random matrices with normalized columns. Global convergence to a local minimum has always been observed, regardless of the initialization. The weighting parameters were empirically set to $\rho = 100$ and $\lambda = \kappa = \frac{0.1}{cr}$. The resulting atoms of the unstructured dictionary $\boldsymbol{D}_1$ and the separable dictionary $\boldsymbol{D}_2 = \boldsymbol{R} \otimes \boldsymbol{C}$ are shown in Figure 4.1(a) and 4.1(b), respectively.



(a) Atoms of unstructured dictionary.

(b) Atoms of separable dictionary.

Figure 4.1: This figure show atoms learned by SeDiL of (a) unstructured dictionary $\boldsymbol{D}_1 = 1 \otimes \boldsymbol{C}$ and (b) separable dictionary $\boldsymbol{D}_2 = \boldsymbol{R} \otimes \boldsymbol{C}$ for a patch size of (8×8). Each atom is shown as an (8×8) block where a black pixel corresponds to the smallest negative entry, gray is a zero entry, and white corresponds to the largest positive entry.

To denoise an image, first, the optimal sparse representations $\{\boldsymbol{X}_i^\star\}_{i=1}^N$ of all possible overlapping noisy image-patches $\{\boldsymbol{S}_i\}_{i=1}^N$ with respect to $\boldsymbol{C}, \boldsymbol{R}$ are found by solving

$$\boldsymbol{X}_i^\star = \arg\min_{\boldsymbol{X}_i} \|\boldsymbol{X}_i\|_1 + \lambda_d \|\boldsymbol{C}\boldsymbol{X}_i\boldsymbol{R}^\top - \boldsymbol{S}_i\|_F^2, \forall i = 1, \dots, N, \tag{4.40}$$

and a clean image-patch is computed from its sparse coefficients via $\boldsymbol{S}_i^\star = \boldsymbol{C}\boldsymbol{X}_i^\star\boldsymbol{R}^\top$. In the present experiments, I employed the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [7] to solve Problem (4.40). The regularization parameter λ_d depends on the noise level and I empirically set it to $\lambda_d = \frac{\sigma_{\text{noise}}}{100}$. As I am considering all overlapping image-patches, several solutions for the same pixel exist, and the final clean image is built by

Table 4.1: This table shows the PSNR in dB and the MSSIM for denoising the five test images corrupted by five different noise levels σ_{noise}. Each cell presents the results for the respective image and noise level for five different methods: top left FISTA+K-SVD dictionary, top right FISTA+unstructured SeDiL, middle left FISTA+ODCT, middle right FISTA+separable SeDiL, bottom BM3D.

	lena				**barbara**				**boat**				**peppers**				**house**			
σ_{noise} / PSNR	PSNR		MSSIM		PSNR		MSSIM		PSNR		MSSIM		PSNR		MSSIM		PSNR		MSSIM	
5 / 34.15	38.42	38.55	0.942	0.944	37.19	37.70	0.959	0.962	36.61	37.03	0.929	0.936	37.06	37.47	0.914	0.921	38.82	38.90	0.944	0.946
	38.45	38.51	0.943	0.946	37.93	37.65	0.963	0.965	37.09	37.04	0.938	0.938	37.53	37.39	0.923	0.922	39.03	38.90	0.950	0.948
	38.45		0.942		38.27		0.964		37.25		0.938		37.60		0.920		39.77		0.956	
10 / 28.13	35.41	35.49	0.907	0.909	33.08	33.71	0.922	0.928	33.54	33.67	0.879	0.882	34.75	34.83	0.875	0.877	35.66	35.63	0.896	0.897
	35.29	35.34	0.907	0.910	33.99	33.49	0.931	0.929	33.45	33.65	0.879	0.883	34.65	34.76	0.876	0.878	35.37	35.54	0.896	0.898
	35.79		0.915		34.96		0.942		33.91		0.887		35.02		0.878		36.69		0.921	
20 / 22.11	32.24	32.31	0.857	0.859	28.88	29.61	0.846	0.859	30.28	30.35	0.800	0.802	32.38	32.40	0.837	0.838	32.83	32.75	0.856	0.856
	32.00	32.11	0.856	0.858	29.95	29.28	0.865	0.854	29.94	30.25	0.792	0.800	31.98	32.23	0.832	0.838	32.11	32.45	0.848	0.854
	32.98		0.875		31.78		0.905		30.89		0.825		32.80		0.845		33.79		0.871	
30 / 18.59	30.35	30.41	0.821	0.822	26.56	27.22	0.775	0.790	28.36	28.41	0.741	0.743	30.81	30.80	0.810	0.810	30.93	30.83	0.826	0.826
	30.02	30.15	0.817	0.820	27.61	26.90	0.800	0.782	27.96	28.27	0.729	0.739	30.28	30.55	0.803	0.809	30.07	30.45	0.815	0.822
	31.22		0.843		29.82		0.868		29.13		0.779		31.32		0.820		32.13		0.847	
50 / 14.15	27.85	27.88	0.760	0.761	24.05	24.43	0.666	0.679	25.96	25.98	0.658	0.659	28.43	28.41	0.761	0.761	28.03	27.92	0.767	0.766
	27.52	27.64	0.754	0.758	24.75	24.24	0.691	0.671	25.61	25.83	0.646	0.654	27.94	28.18	0.753	0.759	27.43	27.60	0.755	0.760
	29.02		0.798		27.23		0.794		26.79		0.705		29.24		0.782		29.72		0.811	

averaging all corresponding overlapping pixels. The achieved results for the five images and the five noise level are given in Table 4.1.

To compare and rank the learned dictionaries among existing state-of-the-art techniques, I learned a universal dictionary $\boldsymbol{D}_{\text{KSVD}}$ with the K-SVD algorithm from the same training set as used for SeDiL and that is of equal dimension as the unstructured dictionary $\boldsymbol{D}_1$. Then, I used this dictionary together with FISTA to solve the same denoising Problems as described above. From Table 4.1, it can be seen that using $\boldsymbol{D}_1$ always yields similar denoising results compared to utilizing $\boldsymbol{D}_{\text{KSVD}}$. Employing the separable dictionary $\boldsymbol{D}_2$ leads to results that are slightly worse compared to employing its unstructured counterpart. This is the tribute that has to be paid for its predefined structure. However, the separability allows a fast implementation just as the popular and also separable Overcomplete Discrete Cosine Transform (ODCT). Here, it can be observed that the separable dictionary $\boldsymbol{D}_2$ learned by SeDiL outperforms the ODCT for most images, while requiring exactly the same computational cost.

4.4.2 Global Face Image Inpainting

The second advantage besides computational efficiency that comes along with the capability of learning a separable dictionary is that SeDiL permits to determine sparse representations for image-patches whose dimensions let other unstructured dictionary learning methods fail due to numerical reasons. In order to demonstrate the capability of SeDiL within this domain, a separable dictionary is learned from a training set that consists of 12 000 images each of dimension (64×64) showing frontal face views of different persons.

These training images were randomly extracted from the 13 228 faces of the "Cropped Labeled Faces in the Wild Database" [1] [65, 112]. The remaining 1228 images were used for the inpainting experiments as explained below. Note that the face positions and the facial expressions in the pictures are arbitrary and diverse; see Figure 4.3 for five exemplary chosen training face images. The dimensions of the resulting matrices $\boldsymbol{R}, \boldsymbol{C}$ were set to (64×128) and all other parameters required for the learning procedure have been chosen as above.

The ability of the separable dictionary to capture the global structure and information that underlies the training samples is illustrated by a face image inpainting experiment, where large missing regions have to be filled up solely based on the available measurements and the information contained in the dictionary. For this experiment, I assume that the position of the pixels that have to be filled up are given. The inpainting procedure is again conducted by applying FISTA on the inverse problem

$$\boldsymbol{X}^\star = \arg\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_1 + \lambda_d \| \operatorname{pr}(\boldsymbol{C}\boldsymbol{X}\boldsymbol{R}^\top) - \boldsymbol{y}\|_2^2, \tag{4.41}$$

where λ_d is again a weighting parameter, the measurements $\boldsymbol{y} \in \mathbb{R}^m$ are the available image data, and $\operatorname{pr}(\cdot) : \mathbb{R}^{w \times h} \to \mathbb{R}^m$ is a projection onto the pixel positions that correspond to the available measurements. An excerpt of the achieved results is presented in Figure 4.3. From these results, it can be seen that the learned dictionary is able to reproduce the global structure that underlies the training data, which is certainly not possible with an analytic dictionary such as the ODCT. I like to mention that this experiment should not be seen as a highly sophisticated face inpainting method, but rather should supply evidence that SeDiL is able to learn a separable dictionary that properly extracts and recovers the global information contained in the employed training set.

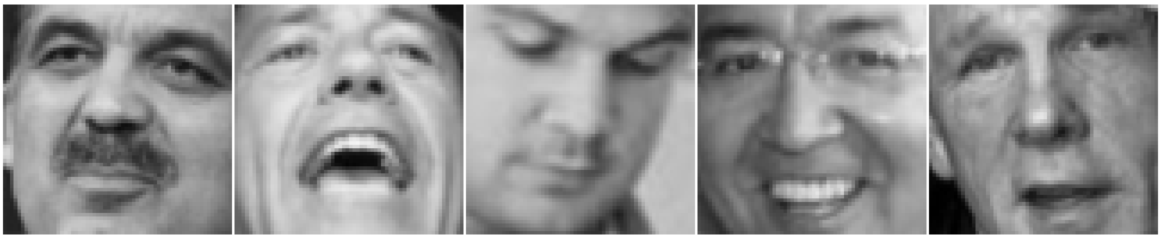


Figure 4.2: This figure shows five exemplarily chosen training images.

[1] http://itee.uq.edu.au/~conrad/lfwcrop/

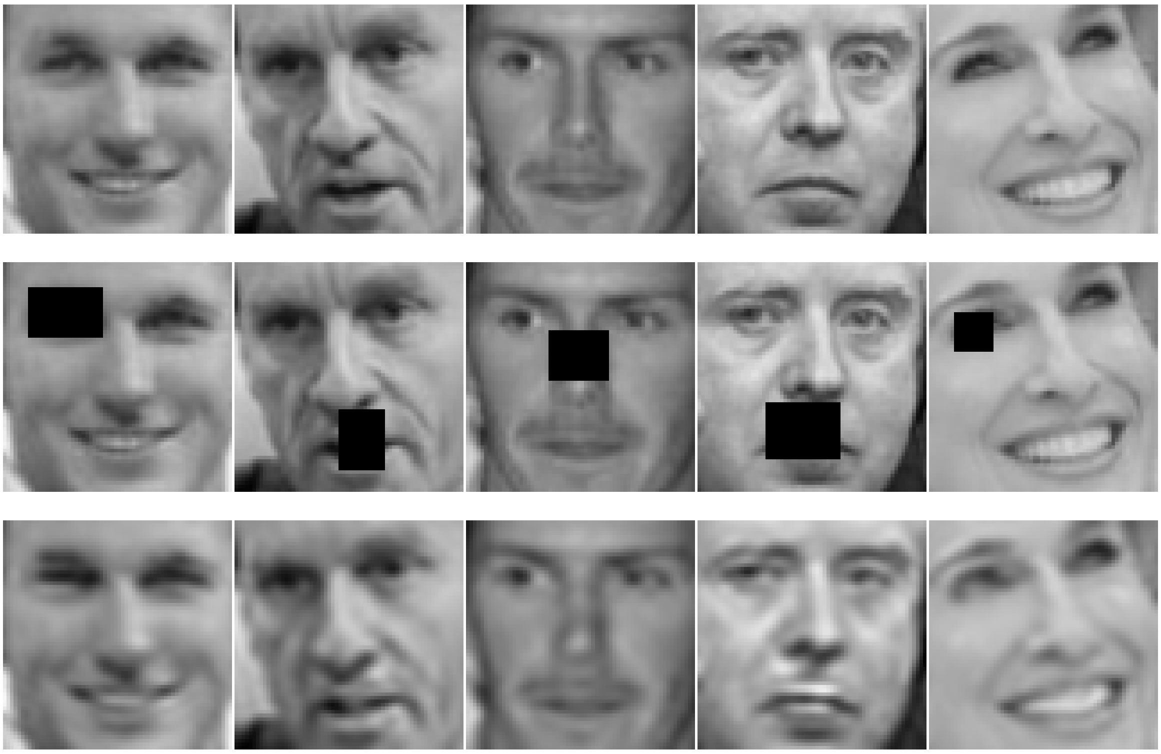


Figure 4.3: This figure presents five exemplary large scale inpainting results. The first row shows the original images from which large regions have been removed in the second row. The last row shows the inpainting results achieved by SeDiL.

4.5 Summary

In this chapter I introduced a new dictionary learning algorithm called SeDiL that is able to learn both conventional unstructured dictionaries as well as dictionaries with a separable matrix structure. Employing a separable structure on dictionaries reduces the general computational complexity of both learning and applying the dictionary from $O(n)$ to $O(\sqrt{n})$ compared to employing unstructured dictionaries, where n denotes the size of the considered signals. Due to this, separable dictionaries can be learned using far larger signal dimensions as compared to those used for learning unstructured dictionaries. Furthermore, they can be applied very efficiently in image reconstruction tasks. Another advantage of SeDiL is that it permits to control the mutual coherence of the resulting dictionary during the learning phase. To that end, I introduce a new mutual coherence measure and showed how it is related to the classical mutual coherence measure. The overall optimization procedure for learning the dictionary is solved by an efficient geometric conjugate gradient algorithm that exploits the underlying manifold structure. In this optimization framework both the dictionary and the sparse codes of the training samples are updated simultaneously. For the classical small scale image processing case, I presented numerical image denoising results that show the state-of-the-art performance of the proposed algorithm. The ability to learn sparse representations of large image-patches is demonstrated by face image inpainting experiments.

Chapter 5

Geometric Analysis Operator Learning

This chapter is partially based on the published work:

> S. Hawe, M. Kleinsteuber, and K. Diepold. Analysis Operator Learning and Its Application to Image Reconstruction. In *IEEE Transactions on Image Processing*, 22(6), pp. 2138--2150. 2013.
>
> S. Hawe, M. Kleinsteuber, and K. Diepold. Cartoon-Like Image Reconstruction via Constrained ℓ_p-Minimization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 717--720. 2012.
>
> S. Hawe, M. Kleinsteuber, and K. Diepold. Dense Disparity Maps from Sparse Disparity Measurements. In *IEEE International Conference on Computer Vision*, pp. 2126--2133. 2011.

Exploiting a priori known structural information lies at the core of many image reconstruction methods that can be stated as inverse problems. The synthesis model, which assumes that images can be decomposed into a linear combination of very few atoms of some dictionary, is now a well-established tool for designing image reconstruction algorithms. An interesting alternative is the analysis model, where the signal is multiplied by an analysis operator and the outcome is assumed to be sparse. This approach has only recently gained increasing interest. The quality of reconstruction methods based on the analysis model severely depends on the right choice of a suitable analysis operator.

In this chapter, I present an algorithm for learning an analysis operator from training images called Geometric Analysis Operator Learning (GOAL). This method is based on a non-convex ℓ_p-norm minimization on the set of full-rank matrices with normalized columns. This admissible set of solutions admits a manifold structure known as the Oblique Manifold, which is exploited to efficiently solve the arising optimization problem by a geometric conjugate gradient technique. Additionally, a penalty term is introduced that prevents the algorithm from learning redundant atoms and permits to control the mutual coherence of a learned operator.

Through a series of synthetic experiments, I show that GOAL outperforms all existing analysis operator learning techniques in terms of computational complexity, accuracy in finding a generating ground truth operator, and generality. To evaluate how an analysis operator learned by GOAL performs in real world image processing applications, I compare its performance as a regularizer for solving inverse problems with employing other operators learned by state-of-the-art analysis operator learning techniques. Concretely, the inverse problems I consider here are image denoising, image inpainting, and single image superresolution. The obtained numerical results show that GOAL outperforms all existing analysis operator learning techniques and that it achieves competitive performance in all evaluated applications compared to specialized state-of-the-art approaches.

5.1 Introduction

5.1.1 Problem Description

Linear inverse problems are ubiquitous in the field of image processing. Prominent examples are image denoising [97], image inpainting [10], image superresolution [53], or image reconstruction from few indirect measurements as in Compressive Sensing [17]. Basically, in all these problems the goal is to reconstruct an unknown image $\boldsymbol{s} \in \mathbb{R}^n$ as accurately as possible from a set of indirect, incomplete, and maybe corrupted measurements $\boldsymbol{y} \in \mathbb{R}^m$ with $n \geq m$, see [69] for a detailed introduction to inverse problems. Formally, this measurement process can be written as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\epsilon}, \tag{5.1}$$

where the vector $\boldsymbol{\epsilon} \in \mathbb{R}^m$ models sampling errors and noise, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the measurement matrix modeling the respective sampling process. In many cases, reconstructing $\boldsymbol{s}$ by simply inverting Equation (5.1) is ill-posed because either the exact measurement process and hence $\boldsymbol{A}$ is unknown, as for example in blind image deconvolution, or the number of observations is much smaller compared to the dimension of the signal, which is the case in Compressive Sensing and image inpainting. To overcome this ill-posedness and to stabilize the solution, prior knowledge or assumptions about the structure of images can be exploited.

5.1.2 Synthesis Model and Dictionary Learning

For didactical reasons, in this section I very briefly review the synthesis model as described in more detail in Chapter 1 and 2. Its underlying assumption is that natural images admit a sparse representation $\boldsymbol{x} \in \mathbb{R}^d$ over some dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ with $d \geq n$. A vector $\boldsymbol{x}$ is called sparse when most of its coefficients are equal to zero or small in magnitude. When $\boldsymbol{s}$ admits a sparse representation over $\boldsymbol{D}$, it can be expressed as a linear combination of only very few columns of the dictionary $\{\boldsymbol{d}_i\}_{i=1}^d$, called *atoms*, given as

$$\boldsymbol{s} = \boldsymbol{D}\boldsymbol{x}. \tag{5.2}$$

For $d > n$, the dictionary is said to be overcomplete or redundant.

Now, using the knowledge that (5.2) allows a sparse solution, an estimation of the original signal in Equation (5.1) can be obtained from the acquired measurements $\boldsymbol{y}$ by first solving

$$\boldsymbol{x}^\star \in \arg\min_{\boldsymbol{x}} \; g(\boldsymbol{x}) \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{D}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \epsilon, \tag{5.3}$$

and afterwards synthesizing the signal from the computed sparse coefficients via $\boldsymbol{s}^\star = \boldsymbol{D}\boldsymbol{x}^\star$. In Problem (5.3), $g : \mathbb{R}^d \to \mathbb{R}$ is a function that promotes or measures sparsity, and $\epsilon \in \mathbb{R}^+$ is an estimated upper bound on the noise power $\|\boldsymbol{e}\|_2^2$. Note that here I assumed the error to be normally distributed and, therefore employed the ℓ_2-norm for measuring how closely the reconstruction resembles the measurements. Depending on the assumed noise statistics any other appropriate error term could be employed. Regarding the sparsity promoting function g, common choices include the ℓ_p-norm of a vector $\boldsymbol{v}$

$$\|\boldsymbol{v}\|_p^p := \sum_i |v_i|^p, \tag{5.4}$$

with $0 < p \leq 1$, or differentiable approximations of (5.4). As the signal is synthesized from the sparse coefficients, the reconstruction model (5.3) is called the *synthesis* reconstruction model, cf. [45].

As explained in detail in Section 2.1.1, to find the minimizer of Problem (5.3) various algorithms based on convex or non-convex optimization, greedy pursuit methods, or Bayesian frameworks exist that may employ different choices of g. Common to all these algorithms is that their performance regarding the reconstruction quality severely depends on an appropriately chosen dictionary $\boldsymbol{D}$. Ideally, one is seeking for a dictionary where $\boldsymbol{s}$ can be represented most accurately with a coefficient vector $\boldsymbol{x}$ that is as sparse as possible. Recall that dictionaries can either be defined analytically or learned from example signals of the considered signal class of interest. While the former are universally applicable and offer fast implementations, the latter permit to find sparser representations of signals that belong to the considered signal class, which in turn leads to increased performance in applications. An in depth introduction on the topic of dictionary learning together with a review of several state-of-the-art methods is given in Section 2.1.2.

5.1.3 Analysis Model

An alternative to the synthesis model (5.3) for reconstructing a signal, is to solve

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s}} \; g(\boldsymbol{\Omega s}) \quad \text{subject to} \quad \|\boldsymbol{As} - \boldsymbol{y}\|_2^2 \leq \epsilon, \tag{5.5}$$

which is known as the *analysis model* [45]. Therein, $\boldsymbol{\Omega} \in \mathbb{R}^{a \times n}$ with $a \geq n$ is called the *analysis operator*, and the *analyzed vector* $\boldsymbol{\alpha} := \boldsymbol{\Omega s} \in \mathbb{R}^a$ is assumed to be sparse, where sparsity is again measured via an appropriate function g. As for the synthesis model given in Equation (5.3), the error is assumed to be normally distributed. In contrast to the synthesis model, where a signal is fully described by the non-zero elements of $\boldsymbol{x}$, in the analysis model the zero elements of the analyzed vector $\boldsymbol{\alpha}$ describe the subspace containing the signal. To emphasize this difference, the term *co-sparsity* has been introduced in [85], which simply counts the number of zero elements of $\boldsymbol{\alpha}$.

As the level of sparsity in the synthesis model depends on the chosen dictionary, the co-sparsity of an analyzed signal depends on the chosen analysis operator $\boldsymbol{\Omega}$. Off-the-shelf analysis operators proposed in the literature include the fused Lasso [124], the translation invariant wavelet transform [117], and probably best known the finite difference operator which is closely related to the total-variation [111]. They all have shown very good performance when used within the analysis model for solving diverse inverse problems in image processing. The question is: *Can the performance of analysis based signal reconstruction tasks be improved by applying a learned analysis operator instead of an analytic one, as it is the case for the synthesis model where learned dictionaries outperform analytic dictionaries?* In [45], it has been discussed that the two models differ significantly, and the naïve way of learning a dictionary and simply employing its transposed or its pseudoinverse as the learned analysis operator fails. Hence, different algorithms are required to learn an analysis operator from example data.

5.2 Analysis Operator Learning

5.2.1 General Scheme

The topic of analysis operator learning has only recently started being investigated, and only a small number of prior work exists. Basically, given a set of M training samples $\{\boldsymbol{s}_i \in \mathbb{R}^n\}_{i=1}^M$, the general goal of analysis operator learning techniques is to find a matrix $\boldsymbol{\Omega} \in \mathbb{R}^{a \times n}$ with $a \geq n$, which leads to a maximally sparse representation $\boldsymbol{\Omega s}_i$ of each of the M training samples. For the case of image processing, the training samples are distinctive vectorized image-patches extracted from a set of training images. Let $\boldsymbol{S} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_M] \in \mathbb{R}^{n \times M}$ be a matrix where its columns constitute the training samples, then the problem can be formally written as

$$\boldsymbol{\Omega}^\star \in \arg\min_{\boldsymbol{\Omega}} \ G(\boldsymbol{\Omega S}) \quad \text{subject to} \quad \boldsymbol{\Omega} \in \mathfrak{C}, \tag{5.6}$$

where $\boldsymbol{\Omega}$ is required to be an element of some constraint set $\mathfrak{C}$, and G is an appropriate function that measures the sparsity of the matrix $\boldsymbol{\Omega S}$. In Section 2.2.2, I explain the relevant analysis operator learning methods that aim at tackling Problem (5.6). These methods mainly differ in the used sparsity measure and the employed constraint set, which is necessary to avoid the trivial solution $\boldsymbol{\Omega} = \boldsymbol{0}_{a \times n}$ and that furthermore permits to enforce certain properties on the operator. In the following I provide a motivation for the sparsity measure and the constraint set employed here to tackle the analysis operator learning problem.

5.2.2 Motivation of the Proposed Approach

Sparsity Measure

In the quest for designing an analysis operator learning algorithm, the natural question arises: *What properties should an analysis operator possess to call it a good operator depending on ones needs?* Clearly, given a signal $\boldsymbol{s}$ that belongs to a certain signal class, the aim is to find an operator $\boldsymbol{\Omega}$ such that $\boldsymbol{\Omega s}$ is *as sparse as possible*, which clearly motivates to minimize the *expected sparsity* $\mathbb{E}[g(\boldsymbol{\Omega s})]$. All state-of-the-art learning methods presented in Section 2.2.2 can be explained in this way, i.e. for their employed measure of sparsity g they aim at learning an analysis operator $\boldsymbol{\Omega}$ that minimizes the empirical arithmetic mean of the sparsity over all M randomly drawn training samples. This, however, does not necessarily mean to learn *the optimal* operator if the purpose is to sparsely represent a large set of signals that all belong to the same class whose intraclass diversity is large, e.g. the class of natural image-patches. The reason for this is that even if the *expected* sparsity with respect to the

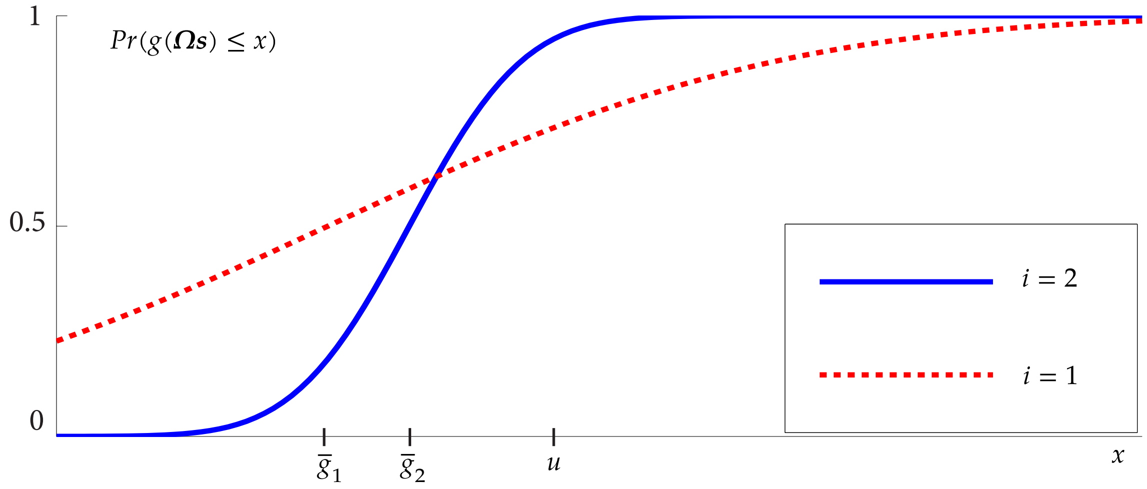


Figure 5.1: This figure illustrates two possible distributions $Pr(g(\boldsymbol{\Omega}_i\mathbf{s}) \leq x)$ for two analysis operators. $\boldsymbol{\Omega}_1$: low expectation $\bar{g}_1$, high variance (dashed line); $\boldsymbol{\Omega}_2$: moderate expectation $\bar{g}_2$, moderate variance (solid line). Although $\boldsymbol{\Omega}_1$ yields a smaller expectation, there are more signals compared to $\boldsymbol{\Omega}_2$ where the sparsity model fails, i.e. $Pr(g(\boldsymbol{\Omega}_1\mathbf{s}) \geq u) > Pr(g(\boldsymbol{\Omega}_2\mathbf{s}) \geq u)$ for some suitable upper bound u.

learned operator is low, there is a high probability that some realizations of this signal class cannot be represented in a sparse way, i.e. that for a given upper bound u, the probability $Pr(g(\boldsymbol{\Omega}\mathbf{s}) \geq u)$ exceeds a tolerable value. This phenomenon is illustrated in Figure 5.1.

Because of this, the algorithm presented here aims at minimizing the empirical expectation of a sparsifying function $g(\boldsymbol{\Omega}\mathbf{s}_i)$ for all training samples $\mathbf{s}_i$, while additionally keeping its empirical variance moderate. In other words, I try to avoid that the analyzed vectors of many similar training samples become "*very sparse*" and consequently prevent $\boldsymbol{\Omega}$ from being adapted to the remaining ones that show more diverse and more interesting structure. For image processing, this is of particular interest if the training patches are chosen randomly from natural images, because there is a high probability of collecting a large subset of very similar patches, e.g. very smooth homogeneous regions, that bias the learning process.

Concretely, the goal is to find an $\boldsymbol{\Omega}$ that minimizes both the squared empirical mean

$$\bar{g}^2 = \left(\frac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i)\right)^2, \tag{5.7}$$

as well as the empirical variance

$$\sigma_g^2 = \tfrac{1}{M}\sum_{i=1}^{M}\left(g(\boldsymbol{\Omega}\mathbf{s}_i) - \overline{g}\right)^2, \tag{5.8}$$

of the sparsity of all M analyzed vectors. Here, I achieve this by minimizing the sum of (5.7) and (5.8), which results in

$$\begin{aligned}\overline{g}^2 + \sigma_g^2 &= \overline{g}^2 + \tfrac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i)^2 + \overline{g}^2 - 2g(\boldsymbol{\Omega}\mathbf{s}_i)\overline{g}\\ &= \overline{g}^2 + \tfrac{1}{M}\sum_{i=1}^{M}\overline{g}^2 - \tfrac{2}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i)\overline{g} + \tfrac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i)^2\\ &= 2\overline{g}^2 - 2\overline{g}\tfrac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i) + \tfrac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i)^2\\ &= \tfrac{1}{M}\sum_{i=1}^{M} g(\boldsymbol{\Omega}\mathbf{s}_i)^2. \end{aligned}\tag{5.9}$$

Using $g(\cdot) = \frac{1}{p}\|\cdot\|_p^p$, and introducing the factor $\frac{1}{2}$ together with the shorthand notation $\boldsymbol{V} = \boldsymbol{\Omega S} \in \mathbb{R}^{a\times M}$, the final sparsity measure I suggest here reads as

$$J_p(\boldsymbol{V}) := \tfrac{1}{2M}\sum_{i=1}^{M}\left(\tfrac{1}{p}\sum_{j=1}^{a}|v_{ji}|^p\right)^2 = \tfrac{1}{2M}\sum_{i=1}^{M}\left(\tfrac{1}{p}\|\boldsymbol{v}_{:,i}\|_p^p\right)^2, \tag{5.10}$$

with $0 \leq p \leq 1$.

Constraint Set and Penalty Functions

Now, going back to the problem of learning an analysis operator, certainly, without any constraints on $\boldsymbol{\Omega}$, the useless solution $\boldsymbol{\Omega} = \boldsymbol{0}_{a\times n}$ is the global minimizer of Problem (5.6). To avoid the trivial solution and to enforce certain properties on the operator as explained later in this section, I suggest to regularize the problem by imposing the following three constraints on $\boldsymbol{\Omega}$.

(i) The *rows* $\boldsymbol{\omega}_{i,:}$ of the analysis operator $\boldsymbol{\Omega}$ have unit Euclidean norm, i.e. $\|\boldsymbol{\omega}_{i,:}\|_2 = 1$ for $i = 1, \dots, a$.

(ii) The analysis operator $\boldsymbol{\Omega}$ has full-rank, i.e. $\mathrm{rk}(\boldsymbol{\Omega}) = n$.

(iii) The analysis operator $\boldsymbol{\Omega}$ does not have linear dependent rows, i.e. $\boldsymbol{\omega}_{i,:} \neq \pm\boldsymbol{\omega}_{j,:}$ for $i \neq j$.

Constraint (i) is a common regularization employed in dictionary/analysis operator learning algorithms and avoids ambiguities due to scaling, i.e. when the entries of a row-vector $\boldsymbol{\omega}_{i,:}$ are all small then $\boldsymbol{\omega}_{i,:}$ will be small as well and no information is gained by that. The rank condition (ii) on $\boldsymbol{\Omega}$ is motivated by the fact that different input samples $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathbb{R}^n$ with $\boldsymbol{s}_1 \neq \boldsymbol{s}_2$ should be mapped to different analyzed vectors $\boldsymbol{\Omega s}_1 \neq \boldsymbol{\Omega s}_2$. With Condition (iii) useless redundant transform coefficients in an analyzed vector are avoided.

The two constraints (i) and (ii) motivate to consider the set of full-rank matrices with normalized *columns*, which has the nice property of admitting a manifold structure known as the *Oblique Manifold* [127]

$$\mathrm{OB}(n,a) := \{\boldsymbol{X} \in \mathbb{R}^{n\times a} |\ \mathrm{rk}(\boldsymbol{X}) = n,\ \mathrm{ddiag}(\boldsymbol{X}^\top \boldsymbol{X}) = \boldsymbol{I}_a\}. \tag{5.11}$$

This definition only yields a non-empty set if $a \geq n$, which is the case examined here. Thus, from now on I consider $a \geq n$. Remember that by constraint (i) I require the *rows* of $\boldsymbol{\Omega}$ to have unit Euclidean norm. Hence, I restrict the *transposed* of the learned analysis operator to be an element of the oblique manifold, i.e. $\boldsymbol{\Omega}^\top \in \mathrm{OB}(n,a)$.

Since $\mathrm{OB}(n,a)$ is open and dense in the set of matrices with normalized columns, an additional penalty term is necessary that ensures the result to adhere to the rank constraint (ii) and that prevents iterates from approaching the boundary of $\mathrm{OB}(n,a)$. Due to this, first consider the following lemma.

Lemma 5.1. *For all elements $\boldsymbol{X} \in OB(n,a)$ with $1 < n \leq a$ the inequality*

$$0 < \det\left(\tfrac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top\right) \leq \left(\tfrac{1}{n}\right)^n \tag{5.12}$$

holds true.

Proof. Due to the full-rank condition on $\boldsymbol{X}$, the Gramian matrix $\boldsymbol{X}\boldsymbol{X}^\top$ is positive definite, consequently the strict inequality $0 < \det(\frac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top)$ applies. To see the second inequality of Lemma 5.1, due to the unit norm columns of $\boldsymbol{X}$ observe that

$$\|\boldsymbol{X}\|_F^2 = \mathrm{tr}(\boldsymbol{X}\boldsymbol{X}^\top) = a, \tag{5.13}$$

which implies $\mathrm{tr}(\frac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top) = 1$. Since the trace of a matrix is equal to the sum of its eigenvalues, which are strictly positive in the present case, it follows that the strict inequality $0 < \lambda_i < 1$ holds true for all eigenvalues $\lambda_i, i = 1, \dots, n$ of $\frac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top$. From the well-known relation between the arithmetic- and the geometric-mean it can be seen that

$$\sqrt[n]{\prod \lambda_i} \leq \tfrac{1}{n} \sum \lambda_i. \tag{5.14}$$

Since the determinant of a matrix is equal to the product of its eigenvalues, together with $\sum \lambda_i = \operatorname{tr}(\frac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top) = 1$, it follows that

$$\det\left(\tfrac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top\right) = \prod \lambda_i \leq \left(\tfrac{1}{n}\right)^n, \tag{5.15}$$

which completes the proof. ■

Now, considering Lemma 5.1 and recall that $\boldsymbol{\Omega}^\top \in \mathrm{OB}(n, a)$, I eventually propose to enforce the full-rank constraint through minimizing the penalty function

$$h(\boldsymbol{\Omega}) := -\tfrac{1}{n\ln(n)} \ln\det\left(\tfrac{1}{a}\boldsymbol{\Omega}^\top\boldsymbol{\Omega}\right), \tag{5.16}$$

with the normalization factor $\frac{1}{n\ln(n)}$ arising from Equation (5.15).

Next, to enforce Condition (iii), the following result proves useful.

Lemma 5.2. *For a matrix $\boldsymbol{X} \in OB(n, a)$ with $1 < n \leq a$, the inequality $|\boldsymbol{x}_{:,i}^\top \boldsymbol{x}_{:,j}| \leq 1$ applies, where equality holds true if and only if $\boldsymbol{x}_{:,i} = \pm\boldsymbol{x}_{:,j}$.*

Proof. By the definition of $\mathrm{OB}(n, a)$ the columns of $\boldsymbol{X}$ are normalized, consequently Lemma 5.2 directly follows from the well-known Cauchy-Schwarz inequality. ■

Using Lemma 5.2, Condition (iii) can be enforced by minimizing the logarithmic barrier function of the scalar products of all distinctive rows of $\boldsymbol{\Omega}$, i.e.

$$r(\boldsymbol{\Omega}) := -\tfrac{2}{a(a-1)} \sum_{1 \leq i < j \leq a} \ln(1 - (\boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top)^2), \tag{5.17}$$

with $\frac{a(a-1)}{2}$ being the number of summands.

Finally, combining all introduced constraints, the optimization problem I propose here for learning the transposed analysis operator reads as

$$\boldsymbol{\Omega}^\top \in \arg \min_{\boldsymbol{X} \in \mathrm{OB}(n,a)} J_p(\boldsymbol{X}^\top \boldsymbol{S}) + \kappa\, h(\boldsymbol{X}^\top) + \mu\, r(\boldsymbol{X}^\top). \tag{5.18}$$

Therein, the two weighting factors $\kappa, \mu \in \mathbb{R}^+$ control the influence of the two constraints on the final analysis operator.

Influence of the Weighting Parameters

In the following, I briefly discuss how the two penalty functions r and h and the weighting parameters κ and μ influence an operator learned by the proposed algorithm. First, the following lemma clarifies the role of κ.

Lemma 5.3. *Let $\boldsymbol{\Omega}$ be a minimum of h in the set of transposed oblique matrices, i.e.*

$$\boldsymbol{\Omega}^\top \in \arg \min_{\boldsymbol{X} \in OB(n,a)} h(\boldsymbol{X}^\top), \tag{5.19}$$

then the condition number of $\boldsymbol{\Omega}$ is equal to one.

Proof. It is well-known that equality of the arithmetic and the geometric mean in Equation (5.14) holds true, if and only if all eigenvalues λ_i of $\frac{1}{a}\boldsymbol{X}\boldsymbol{X}^\top$ are identical, i.e. $\lambda_1 = \ldots = \lambda_n$. Hence, if $\boldsymbol{\Omega}^\top \in \arg \min_{\boldsymbol{X} \in \mathrm{OB}(n,a)} h(\boldsymbol{X}^\top)$ holds, then it follows that $\det(\frac{1}{a}\boldsymbol{\Omega}^\top\boldsymbol{\Omega}) = (\frac{1}{n})^n$, and consequently all singular values of $\boldsymbol{\Omega}$ coincide. This implies that the condition number of $\boldsymbol{\Omega}$, which is defined as the quotient of the largest to the smallest singular value, is equal to one. ■

With other words, the minima of Problem (5.19) are uniformly normalized tight frames (UNTF), which have been used in [139] to regularize the analysis operator learning problem. For the algorithm proposed here, it can be concluded from Lemma 5.3 that the larger κ is chosen, the smaller the condition number of $\boldsymbol{\Omega}$ gets, approaching one at the limit. Thus, learning analysis operators that are UNTFs is a special case of my method. To further understand how the condition of an analysis operator influences its applicability in signal processing tasks, remember the well-known inequality

$$\sigma_{\min}\|\boldsymbol{s}_1 - \boldsymbol{s}_2\|_2 \leq \|\boldsymbol{\Omega}\boldsymbol{s}_1 - \boldsymbol{\Omega}\boldsymbol{s}_2\|_2 \leq \sigma_{\max}\|\boldsymbol{s}_1 - \boldsymbol{s}_2\|_2, \tag{5.20}$$

with $\sigma_{\min}$ being the smallest singular value of $\boldsymbol{\Omega}$ and $\sigma_{\max}$ being the largest one, respectively. From this inequality it follows that an analysis operator found with a large κ, i.e. obeying $\sigma_{\min} \approx \sigma_{\max}$, carries over distinctness of different signals to their analyzed versions. In other words, different signals are mapped to different analyzed vectors, which in turn helps to to find a unique solution of inverse problems that are regularized by the analysis model.

The second weighting parameter μ controls the influence of the penalty term (5.17) on the learning process, and consequently regulates the redundancy between the rows of the analysis operator. This in turn avoids useless redundant coefficients in an analyzed vector $\boldsymbol{\Omega s}$. To show further implications, consider the following lemma.

Lemma 5.4. *The difference between the i-th and the j-th entry of an analyzed vector $\boldsymbol{\Omega s}$ is bounded by*

$$|\boldsymbol{\omega}_{i,:}\boldsymbol{s} - \boldsymbol{\omega}_{j,:}\boldsymbol{s}| \leq \sqrt{2(1 - \boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top)}\, \|\boldsymbol{s}\|_2, \tag{5.21}$$

with $\boldsymbol{\omega}_{i,:}, \boldsymbol{\omega}_{j,:}$ again being the i-th and j-th row of $\boldsymbol{\Omega}$.

Proof. From the Cauchy-Schwarz inequality one gets

$$|\boldsymbol{\omega}_{i,:}\boldsymbol{s} - \boldsymbol{\omega}_{j,:}\boldsymbol{s}| = |(\boldsymbol{\omega}_{i,:} - \boldsymbol{\omega}_{j,:})\boldsymbol{s}| \leq \|\boldsymbol{\omega}_{i,:} - \boldsymbol{\omega}_{j,:}\|_2 \|\boldsymbol{s}\|_2. \tag{5.22}$$

Since by definition $\|\boldsymbol{\omega}_{i,:}\|_2 = \|\boldsymbol{\omega}_{j,:}\|_2 = 1$, it follows that $\|\boldsymbol{\omega}_{i,:} - \boldsymbol{\omega}_{j,:}\|_2 = \sqrt{2(1 - \boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top)}$. ■

Lemma 5.4 implies, that if the i-th entry of the analyzed vector is significantly larger than zero, then a large absolute value of $\boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top$ prevents the j-th entry from being small. To achieve a high level of co-sparsity, this is an unwanted effect that GOAL avoids via the function r given in Equation (5.17). The larger μ is chosen, the more weight is assigned to r and the more diverse the rows of $\boldsymbol{\Omega}$ become. I want to mention here, that the same effect is achieved by minimizing the mutual coherence of the analysis operator, which is given as $\max_{i \neq j} |\boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top|$. My experiments suggest that enlarging μ leads to minimizing the mutual coherence of an analysis operator.

Now, having introduced all necessary ingredients for the analysis operator learning method GOAL, in the following section, I explain how the manifold structure of $\mathrm{OB}(n,a)$ together with the concepts introduced in Chapter 3 can be exploited to efficiently solve the associated optimization problem.

5.3 Geometric Analysis Operator Learning Algorithm (GOAL)

Knowing that the feasible set of solutions to Problem (5.18) is restricted to a smooth manifold permits to formulate a geometric conjugate gradient (CG)-method to learn the analysis operator. In this subsection, I present all necessary ingredients to implement this approach. Results regarding the geometry of $\mathrm{OB}(n,a)$ are derived e.g. in [127]. To enhance legibility, and since the dimensions n and a are fixed throughout the rest of this chapter, the oblique manifold is further on denoted by OB.

First, as OB is a submanifold of the product of unit spheres manifold, all formulas regrading the tangent space, the orthogonal projection onto the tangent space, the geodesics, and the parallel transport are equivalent to those given in Section 4.3. Next, to employ a geometric CG-method, a differentiable cost function f is required. However, the original cost function presented in Problem (5.18) is not differentiable due to the non-smoothness of the ℓ_p-pseudo-norm (5.10). To overcome this problem, I exchange Function (5.10) with a smooth approximation that is concretely given as

$$J_{p,\nu}(\boldsymbol{V}) := \tfrac{1}{2M}\sum_{j=1}^{M}\left(\tfrac{1}{p}\sum_{i=1}^{a}(v_{ij}^2+\nu)^{\frac{p}{2}}\right)^2, \tag{5.23}$$

with $\nu \in \mathbb{R}^+$ being a smoothing parameter. The smaller ν is chosen, the more closely the approximation resembles the original function (5.10). Now, to implement the operator learning algorithm the gradient of the cost function is required. Using $\boldsymbol{V} = \boldsymbol{\Omega S}$ together with the shorthand notation $z_{ij} := (\boldsymbol{\Omega S})_{ij}$, the gradient of the sparsity inducing function (5.23) reads as

$$\tfrac{\partial}{\partial \boldsymbol{\Omega}} J_{p,\nu}(\boldsymbol{\Omega S}) = \left[\frac{1}{M}\sum_{j=1}^{M}\frac{1}{p}\sum_{i=1}^{a}(z_{ij}^2+\nu)^{\frac{p}{2}}\sum_{i=1}^{a}\left(z_{ij}(z_{ij}^2+\nu)^{\frac{p}{2}-1}\boldsymbol{E}_{ij}\right)\right]\boldsymbol{S}^\top. \tag{5.24}$$

Next, I reformulate the rank penalty term (5.16) as

$$\begin{aligned} h(\boldsymbol{\Omega}) &= -\tfrac{1}{n\ln(n)}\ln\det\left(\tfrac{1}{a}\boldsymbol{\Omega}^\top\boldsymbol{\Omega}\right) \\ &= -\tfrac{1}{n\ln(n)}\ln\left(\prod_i \tfrac{1}{a}\lambda_i\right) \\ &= -\tfrac{1}{n\ln(n)}\ln\left(\tfrac{1}{a^n}\prod_i \lambda_i\right) \\ &= \tfrac{\ln(a)}{\ln(n)} - \tfrac{1}{n\ln(n)}\ln\det(\boldsymbol{\Omega}^\top\boldsymbol{\Omega}), \end{aligned} \tag{5.25}$$

with λ_i denoting the eigenvalues of $\boldsymbol{\Omega}^\top\boldsymbol{\Omega}$. This reformulation is necessary to avoid numerical instabilities in real implementations when n and a become so large that the factor $\frac{1}{a^n}$ dominates the penalty term and leads to $h(\boldsymbol{\Omega}) = -\ln(0) = \infty$, independent of the true rank of the operator. The gradient of the rank penalty term is not affected by this reformulation and is given as

$$\tfrac{\partial}{\partial \boldsymbol{\Omega}} h(\boldsymbol{\Omega}) = -\tfrac{2}{n\ln(n)}\boldsymbol{\Omega}(\boldsymbol{\Omega}^\top\boldsymbol{\Omega})^{-1}. \tag{5.26}$$

Last, with $\boldsymbol{E}_{ji}$ denoting a square matrix whose i-th entry in the j-th column is equal to one and all other entries being zero the gradient of the logarithmic barrier function (5.17) reads as

$$\tfrac{\partial}{\partial \boldsymbol{\Omega}} r(\boldsymbol{\Omega}) = \tfrac{2}{a(a-1)} \left[\sum_{1 \leq i < j \leq a} \frac{2\boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top}{1 - (\boldsymbol{\omega}_{i,:}\boldsymbol{\omega}_{j,:}^\top)^2} (\boldsymbol{E}_{ij} + \boldsymbol{E}_{ji}) \right] \boldsymbol{\Omega}. \tag{5.27}$$

Combining Equations (5.24), (5.26), and (5.27), the gradient of the cost function

$$f(\boldsymbol{X}) := J_{p,\nu}(\boldsymbol{X}^\top \boldsymbol{S}) + \kappa\, h(\boldsymbol{X}^\top) + \mu\, r(\boldsymbol{X}^\top) \tag{5.28}$$

that I suggest here for learning an analysis operator is given by

$$\nabla f(\boldsymbol{X}) = \tfrac{\partial}{\partial \boldsymbol{X}} J_{p,\nu}(\boldsymbol{X}^\top \boldsymbol{S}) + \kappa \tfrac{\partial}{\partial \boldsymbol{X}} h(\boldsymbol{X}^\top) + \mu \tfrac{\partial}{\partial \boldsymbol{X}} r(\boldsymbol{X}^\top). \tag{5.29}$$

As in Chapter 3, in following I use the shorthand notation $\boldsymbol{G}^{(i)} := \boldsymbol{G}(\boldsymbol{X}^{(i)})$ to denote the Riemannian gradient determined at the i-th iteration.

Finally, for the CG-update parameter $\beta^{(i)}$, I employ the same hybrid formula (4.32) as used for the separable dictionary learning algorithm. To compute the step size $\alpha^{(i)}$, I use an adaption of the well-known backtracking line search to the geodesic $\Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, t)$. Roughly speaking, this algorithm determines the step size by iteratively decreasing an initial step size $t_0^{(i)}$ by a constant factor $c_1 < 1$ until the Armijo condition that ensures a sufficient decrease of f is met. The entire procedure is given in Algorithm 5.1. Such simple line search techniques are very efficient, while being almost as accurate as computing the exact minimizer of Problem (3.4) which, however, is computationally more demanding. In my implementation, I empirically set $c_1 = 0.9$ and $c_2 = 10^{-2}$ and as also proposed in [56] used

Algorithm 5.1 Backtracking Line Search on Oblique Manifold

```
Input: $t_0^{(i)} > 0,\ 0 < c_1 < 1,\ 0 < c_2 < 0.5,\ \boldsymbol{X}^{(i)}, \boldsymbol{G}^{(i)}, \boldsymbol{H}^{(i)}$
Set: $t \leftarrow t_0^{(i)}$
  while $f(\Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, t)) > f(\boldsymbol{X}^{(i)}) + t c_2 \langle \boldsymbol{G}^{(i)}, \boldsymbol{H}^{(i)} \rangle$ do
    $t \leftarrow c_1 t$
  end while
Output: $\alpha^{(i)} \leftarrow t$
```

$$t_0^{(0)} = \|\boldsymbol{G}^{(0)}\|_F^{-1}, \tag{5.30}$$

as an initial guess for the step size at the first CG-iteration $i = 0$. In the subsequent iterations, the backtracking line search is initialized by the step size found at the previous iteration divided by the line search parameter c_1, i.e.

$$t_0^{(i)} = \frac{\alpha^{(i-1)}}{c_1}. \tag{5.31}$$

The complete approach GOAL for learning the analysis operator is summarized in Algorithm 5.2. Note that under the condition that the Fletcher-Reeves update formula is used together with some mild conditions on the step-size selection, the convergence of Algorithm 5.2 to a critical point, i.e. $\liminf_{i\to\infty}\|\boldsymbol{G}^{(i)}\| = 0$, is guaranteed by a result provided in [104].

Algorithm 5.2 Geometric Analysis Operator Learning (GOAL)

```
Input: Initial analysis operator $\boldsymbol{\Omega}_{\text{init}}^\top \in \mathrm{OB}$, training data $\boldsymbol{S}$, parameters $p, \nu, \kappa, \mu$
Set: $i \leftarrow 0, \boldsymbol{X}^{(0)} \leftarrow \boldsymbol{\Omega}_{\text{init}}^\top, \boldsymbol{G}^{(0)} \leftarrow \Pi_{T_{\boldsymbol{X}^{(0)}}\mathrm{M}}(\nabla f(\boldsymbol{X}^{(0)})), \boldsymbol{H}^{(0)} \leftarrow -\boldsymbol{G}^{(0)}$
  repeat
    $\alpha^{(i)} \leftarrow \arg\min_{t\geq 0} f(\Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, t))$, cf. Algorithm 5.1 in conjunction with Equation (5.28)
    $\boldsymbol{X}^{(i+1)} \leftarrow \Gamma(\boldsymbol{X}^{(i)}, \boldsymbol{H}^{(i)}, \alpha^{(i)})$, cf. Equation (4.28)
    $\boldsymbol{G}^{(i+1)} \leftarrow \Pi_{T_{\boldsymbol{X}^{(i+1)}}\mathrm{M}}(\nabla f(\boldsymbol{X}^{(i+1)}))$, cf. Equations (4.22) and (5.29)
    $\beta^{(i)} \leftarrow \max\left(0, \min(\beta_{\mathrm{DY}}^{(i)}, \beta_{\mathrm{HS}}^{(i)})\right)$, cf. Equations (3.9), (3.11)
    $\boldsymbol{H}^{(i+1)} \leftarrow -\boldsymbol{G}^{(i+1)} + \beta^{(i)} \mathcal{T}_{\boldsymbol{H}^{(i)}}^{(i+1)}$, cf. Equations (3.6), (4.30)
    $i \leftarrow i + 1$
  until $\|\boldsymbol{X}^{(i)} - \boldsymbol{X}^{(i-1)}\|_F < 10^{-4} \vee i =$ maximum # iterations
Output: $\boldsymbol{\Omega}^\star \leftarrow \boldsymbol{X}^{(i)\top}$
```

5.4 Synthetic Experiments

In this section, I evaluate the performance of GOAL to recover a known ground truth analysis operator $\boldsymbol{\Omega} \in \mathbb{R}^{a\times n}$ from synthetically created training samples. The training samples were created such that they reside in a r-dimensional subspace with $r < n$ and admit a co-sparse representation with $n - r$ zeros over the ground truth operator. To concretely generate such a training sample $\boldsymbol{s} \in \mathbb{R}^n$, I followed the procedure presented in [92], which consists of first selecting a random $n - r$ dimensional set of row indices $\mathcal{I} \subset \{1, \dots, a\}$, and then projecting a random vector $\boldsymbol{u} \in \mathbb{R}^n$ onto the orthogonal complement of the selected rows, i.e.

$$\boldsymbol{s} = (\boldsymbol{I}_n - \boldsymbol{\Omega}^{\dagger}_{\mathcal{J},:}\boldsymbol{\Omega}_{\mathcal{J},:})\boldsymbol{u}. \tag{5.32}$$

In that way, the level of co-sparsity of $\boldsymbol{s}$ with respect to $\boldsymbol{\Omega}$ is either $n - r$ if the operator is in general position, or higher if the rows of the operator exhibit linear dependencies, as it is the case for the finite difference operator.

In the following experiments, I employed several test sets, each consisting of $M = 25\,000$ signals of dimension $n = 25$ created as described above using a two times overcomplete ground truth analysis operator, i.e. $a = 2n = 50$. All elements $\boldsymbol{s}_i$ of the same set have the same level of co-sparsity. To test the influence of the generating operator on the ability of the learning algorithm to recover it, I used three different ground truth operators that are (i) an operator with random Gaussian entries $\boldsymbol{\Omega}_{\mathrm{RAND}} \in \mathbb{R}^{50\times 25}$, (ii) the 2D-finite difference operator $\boldsymbol{\Omega}_{\mathrm{DIFF}} \in \mathbb{R}^{50\times 25}$, and (iii) a randomly generated uniformly normalized tight frame $\boldsymbol{\Omega}_{\mathrm{UNTF}} \in \mathbb{R}^{50\times 25}$. For every generating operator I created $n - 1 = 24$ training sets with different level of co-sparsity varying between one and $n - 1 = 24$. From each of these 72 training sets, I learned an analysis operator $\boldsymbol{\Omega}^{\star} \in \mathbb{R}^{50\times 25}$ using GOAL, and measured how closely $\boldsymbol{\Omega}^{\star}$ fits the respective ground truth operator. To that end, I used two standard measures from the literature that I called C_1 and C_2, which are described below.

C_1 denotes the percentage of exactly recovered analysis atoms. The i-th atom $\boldsymbol{\omega}^{\star}_{i,:}$ is assumed to be recovered exactly whenever

$$\min_j (1 - |\boldsymbol{\omega}^{\star}_{i,:}\boldsymbol{\omega}^{\top}_{j,:}|) < 0.01, \tag{5.33}$$

holds true.

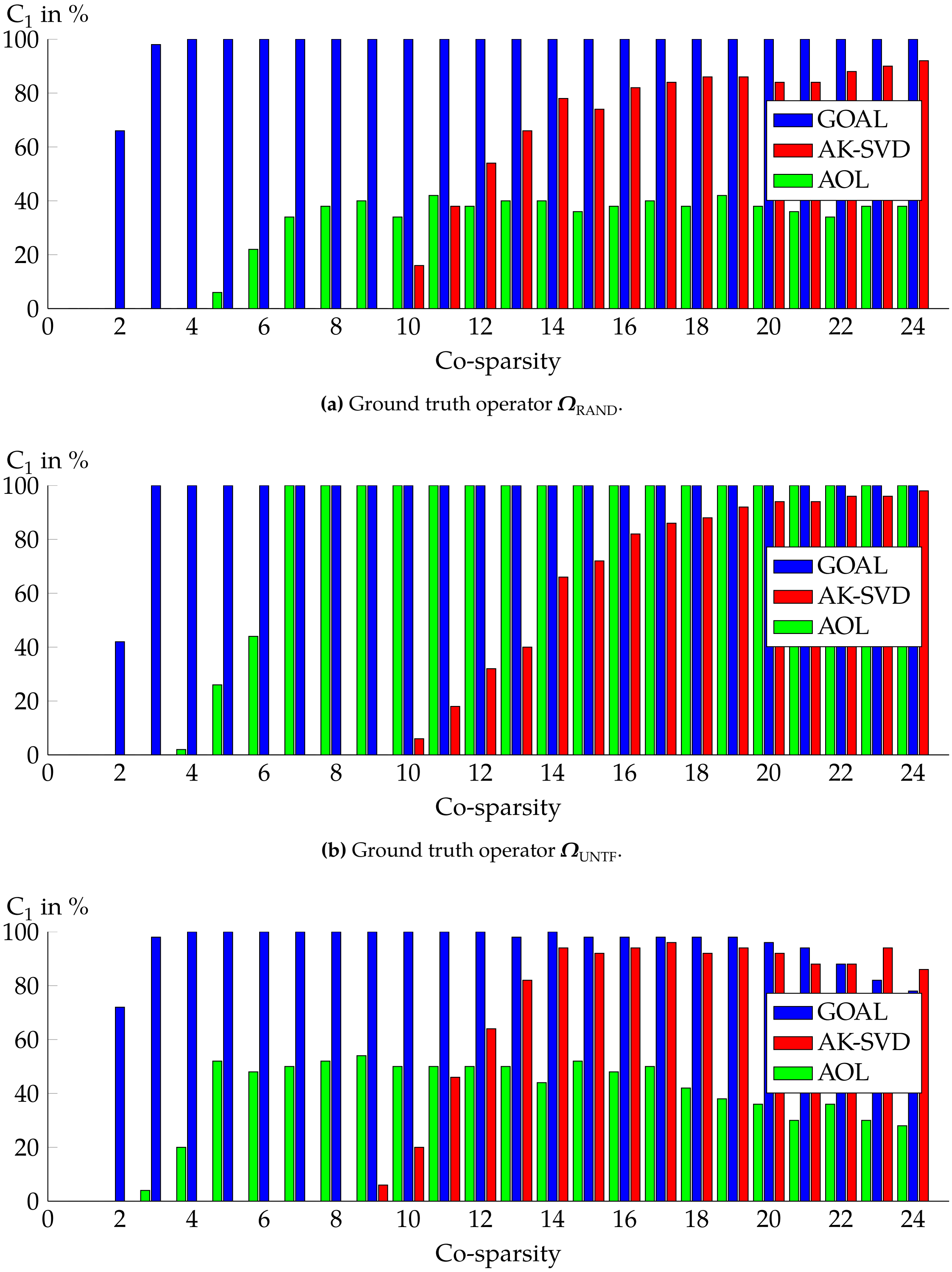
C_2 is the Euclidean distance between the recovered operator $\boldsymbol{\Omega}^{\star}$ and the generating operator, i.e. $C_2 = \|\boldsymbol{\Omega} - \boldsymbol{\Omega}^{\star}\|_F$. As correctly recovered atoms are likely to have different row indices in the ground truth operator $\boldsymbol{\Omega}$ and the recovered operator $\boldsymbol{\Omega}^{\star}$, C_2 is computed after reordering the rows in the $\boldsymbol{\Omega}^{\star}$ such that the most similar rows of $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}^{\star}$ have the same row index.

To rank the operator recovery performance of GOAL among the performance of other analysis operator learning techniques, I additionally made all experiments with $\boldsymbol{\Omega}^{\star}_{\mathrm{AKSVD}} \in \mathbb{R}^{50\times 25}$ learned by AK-SVD [109], and $\boldsymbol{\Omega}^{\star}_{AOL} \in \mathbb{R}^{50\times 25}$ learned by the UNTF constraint based algorithm AOL [137]. All three algorithms employed the same training set and used the same initial operator. The parameters of all three algorithms have been tuned manually to achieve the best possible operator recovery performance. I set the number of iterations to 300 for both GOAL and AK-SVD, and to 8 000 for AOL, without employing any further convergence criterion.

Figures 5.2(a)-(c) show the measure C_1 with respect to the level of co-sparsity obtained from all 72 training sets for the three compared algorithms. The corresponding results regarding C_2 are given in Table 5.1. For the generating operators $\boldsymbol{\Omega}_{\text{RAND}}$, GOAL always achieves the best recovery performance regarding both C_1 and C_2, independent of the level of co-sparsity. when the generating operator was an UNTF, AOL has better recovering performance then AK-SVD and approximately the same performance as GOAL for high-levels of co-sparsity while being worse than GOAL for moderate to low levels of co-sparsity. Regarding $\boldsymbol{\Omega}_{\text{DIFF}}$, AK-SVD performs best for high levels of co-sparsity, while GOAL achieves the best performance for the majority of the training sets. It can be further seen that the performance of all algorithms decreases the lower the level of co-sparsity of the respective training set is. Note that GOAL only fails completely for the lowest possible level of co-sparsity $n - r = 1$.

Another important performance criterion, especially for real world applications, is the computational complexity of an analysis operator learning algorithm. To quantify that, I measured the computation time required by each algorithm to learn an analysis operator using the test setup as above. Unoptimized Matlab implementations were used and executed on a standard desktop PC with a 3.2 GHz Intel i7 six core CPU and 16 Gb RAM. In this test, GOAL required around 0.04 seconds per iteration independent of the level of co-sparsity of the underlying training set. AK-SVD required between 2 and 60 seconds per iteration depending on the level of co-sparsity with the higher the level of co-sparsity is, the more computation time it required. This is due to the computationally demanding Backward Greedy (BG) algorithm used for solving the analysis sparse coding problem, whose number of iterations is equal to the targeted level of co-sparsity, see 2.2.1 for a more detailed description of the BG algorithm. AOL only needed around 0.02 seconds per iteration; however, it has to perform far more iterations compared to the two other methods. Note that in all my experiments after around 75 iterations the operator found by GOAL did not change anymore, which shows the good convergence property of GOAL. This behavior is visualized in Figures 5.3(a)-(c), which for the three compared methods present the respective evolution of C_2 with respect to the iteration number for recovering an UNTF analysis operator from a training set with level of co-sparsity $n - r = 21$.

From these experiments, I conclude that the UNTF based method AOL is too restrictive to reliably find an optimal analysis operator, and that AK-SVD only performs well for signals

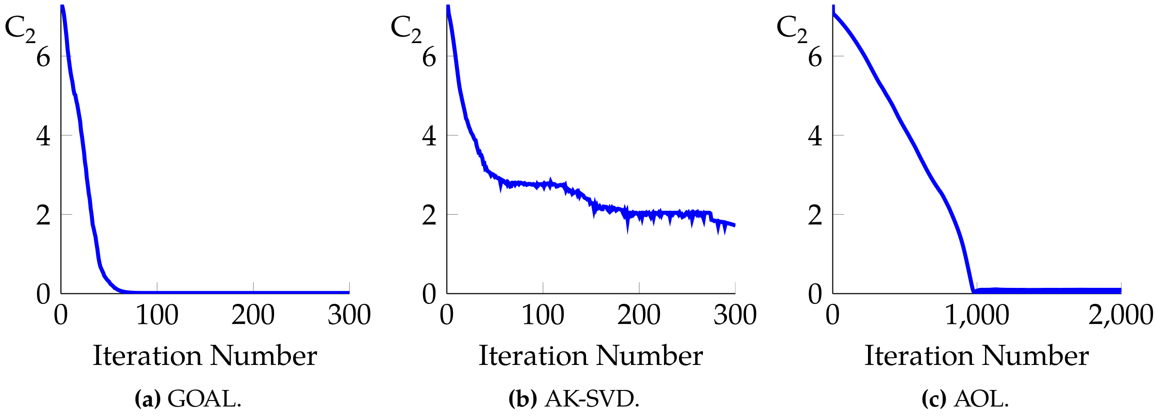


(a) Ground truth operator $\boldsymbol{\Omega}_{\text{RAND}}$.

(b) Ground truth operator $\boldsymbol{\Omega}_{\text{UNTF}}$.

(c) Ground truth operator $\boldsymbol{\Omega}_{\text{DIFF}}$.

Figure 5.2: This figure shows the measure C_1 for all three methods with respect to the level of co-sparsity of the respective training set.



(a) GOAL. **(b)** AK-SVD. **(c)** AOL.

Figure 5.3: This figure shows the evolution of C_2 over the iteration number for the three compared learning techniques GOAL, AK-SVD, and AOL to recover a UNTF from a training set with signal co-sparsity of $n-r = 21$.

with high level of co-sparsity. In contrast, GOAL always achieves decent results almost independent of the level of co-sparsity of the training set and the underlying generating operator. Furthermore, GOAL only requires a few iterations until it converges and only has to perform some basic algebraic operations to update the operator. Interestingly, choosing κ and μ such that GOAL achieves the best recovery performance only depends on the level of co-sparsity but not on the generating ground truth analysis operator.

In the remainder of this chapter, I evaluate how GOAL performs in real image processing applications. To that end, in the next section I explain how the patch based operator can be applied to achieve global image reconstruction results.

5.5 Analysis Operator Based Image Reconstruction

In this section I explain how the patch based analysis operator $\boldsymbol{\Omega}^\star \in \mathbb{R}^{a \times n}$ is utilized to reconstruct an unknown image $\boldsymbol{s} \in \mathbb{R}^N$ from some given measurements $\boldsymbol{y} \in \mathbb{R}^m$ following the analysis approach (5.5). Here, the vector $\boldsymbol{s} \in \mathbb{R}^N$ denotes a vectorized image of dimension $N = wh$, with w being the width and h being the height of the image, respectively, obtained by stacking the columns of the image above each other. In the following, I will loosely speak of $\boldsymbol{s}$ as the image. Remember, that the size of $\boldsymbol{\Omega}^\star$ is very small compared to the size of the image, and it has to be applied locally to small image-patches rather than globally to the entire image.

To globally reconstruct the image, the simplest way is to partition the image into non-overlapping patches, reconstruct each patch individually, and stick the reconstructed

Table 5.1: This table presents the quality measure C_2, i.e. the Euclidean distance between a learned operator and the corresponding ground truth operator for the three generating operators $\boldsymbol{\Omega}_{\mathrm{RAND}}, \boldsymbol{\Omega}_{\mathrm{UNTF}}, \boldsymbol{\Omega}_{\mathrm{DIFF}}$ with respect to the co-sparsities of the training sets for the three compared learning methods GOAL, AK-SVD, and AOL. For each set, bold-faced digits highlight the best result.

	Ω_{RAND}			Ω_{UNTF}			Ω_{DIFF}		
Co-sparsity	**GOAL**	**AK-SVD**	**AOL**	**GOAL**	**AK-SVD**	**AOL**	**GOAL**	**AK-SVD**	**AOL**
1	**6.38**	7.32	7.08	**6.81**	7.28	6.91	**6.32**	7.20	6.99
2	**3.68**	7.34	6.85	**4.98**	7.29	6.78	**3.31**	7.22	6.79
3	**0.76**	7.30	6.70	**0.21**	7.29	6.54	**1.03**	7.18	6.10
4	**0.15**	7.29	6.59	**0.15**	7.29	6.52	**0.28**	7.19	5.09
5	**0.14**	7.22	6.16	**0.13**	7.25	5.40	**0.20**	7.11	3.72
6	**0.12**	7.03	5.18	**0.15**	7.16	4.55	**0.14**	7.02	3.58
7	**0.12**	6.93	4.87	0.13	7.18	**0.01**	**0.14**	6.92	3.65
8	**0.12**	6.55	4.44	0.12	7.20	**0.02**	**0.14**	6.70	3.22
9	**0.13**	6.34	4.44	0.08	7.11	**0.02**	**0.14**	5.98	3.34
10	**0.15**	5.76	4.16	0.09	6.83	**0.02**	**0.18**	5.56	3.31
11	**0.15**	4.86	4.27	0.09	6.06	**0.02**	**0.16**	4.46	3.31
12	**0.17**	4.29	4.16	0.09	5.51	**0.02**	**0.16**	3.86	3.52
13	**0.17**	3.85	4.12	0.08	4.95	**0.02**	**1.00**	2.63	3.60
14	**0.19**	3.31	4.24	0.09	3.99	**0.03**	**0.18**	1.73	3.80
15	**0.21**	3.52	3.95	0.08	3.72	**0.03**	**1.00**	1.99	3.77
16	**0.23**	3.02	4.22	0.09	3.12	**0.03**	**1.00**	1.73	3.60
17	**0.25**	3.00	4.13	0.10	2.86	**0.03**	**1.01**	1.40	3.63
18	**0.28**	2.80	3.99	0.07	2.48	**0.03**	**1.00**	1.96	3.84
19	**0.30**	2.68	3.97	0.07	2.22	**0.03**	**1.01**	1.72	4.09
20	**0.33**	3.03	3.94	0.08	1.95	**0.04**	**1.39**	1.96	4.23
21	**0.40**	2.86	3.87	0.07	1.72	**0.04**	**1.69**	2.32	4.36
22	**0.42**	2.66	3.92	0.07	1.57	**0.04**	2.41	**2.19**	4.32
23	**0.46**	2.45	3.93	0.08	1.58	**0.04**	2.86	**1.70**	4.41
24	**0.49**	2.22	4.01	0.08	1.06	**0.04**	3.21	**1.93**	4.55

patches together to form the final image. However, the quality of this approach highly depends on the chosen partitioning of the image. Furthermore, this naïve approach leads to spurious artifacts at patch boundaries, and fails for example in inpainting tasks where large holes compared to the patch size have to be filled up. To reduce such artifacts, a common approach is to work with overlapping patches, where each patch is reconstructed individually and the entire image is formed by averaging over the reconstructed overlapping regions in a final step. However, this method still misses global support during the reconstruction process, and consequently leads to poor inpainting results and is not applicable for e.g. Compressive Sensing tasks. To overcome these drawbacks, I use a method related to the patch based synthesis approach from [43] and the FoE algorithm [107], which provides global support from local information. In words, instead of optimizing over each patch individually and combining them in a final step, I optimize over the *entire* image demanding that a pixel is reconstructed such that the average sparsity of all patches it belongs to is minimized. When all possible patch positions are taken into account, this procedure is entirely partitioning-invariant. For legibility and without loss of generality, in the following I assume square patches of size $(\sqrt{n} \times \sqrt{n})$, with $\sqrt{n}$ being a positive integer.

Formally, let $\boldsymbol{r} \subseteq \{1, \dots, h\}$ and $\boldsymbol{c} \subseteq \{1, \dots, w\}$ denote sets of indices with $r_{i+1} - r_i = d_v$, $c_{i+1} - c_i = d_h$ and $1 \leq d_v, d_h \leq \sqrt{n}$. Therein, d_v and d_h denote the degree of overlap between two adjacent patches in vertical, and horizontal direction, respectively. In my image reconstruction task, I consider all image-patches whose center positions are an element of the Cartesian product set $\boldsymbol{r} \times \boldsymbol{c}$. Hence, with $|\cdot|$ denoting the cardinality of a set, the total number of patches being considered is equal to $|\boldsymbol{r} \times \boldsymbol{c}| = |\boldsymbol{r}||\boldsymbol{c}|$. Now, let $\boldsymbol{P}_{rc}$ be a binary $(n \times N)$ matrix that extracts the patch centered at position (r, c). With this notation, the (global) sparsity promoting function is formulated as

$$\sum_{r \in \boldsymbol{r}} \sum_{c \in \boldsymbol{c}} \sum_{i=1}^{a} ((\boldsymbol{\Omega}^{\star} \boldsymbol{P}_{rc} \boldsymbol{s})_i^2 + \nu)^{\frac{p}{2}}, \tag{5.34}$$

which measures the overall approximated ℓ_p-pseudo-norm of all considered analyzed image-patches. To use the same notation as in the standard analysis model formulation (5.5), I compactly rewrite Equation (5.34) as

$$g(\boldsymbol{\Omega}^F \boldsymbol{s}) := \sum_{i=1}^{K} \left((\boldsymbol{\Omega}^F \boldsymbol{s})_i^2 + \nu \right)^{\frac{p}{2}}, \tag{5.35}$$

with $K = a|\boldsymbol{r}||\boldsymbol{c}|$ and

$$\boldsymbol{\Omega}^F := \begin{bmatrix} \boldsymbol{\Omega}^{\star} \boldsymbol{P}_{r_1 c_1} \\ \boldsymbol{\Omega}^{\star} \boldsymbol{P}_{r_1 c_2} \\ \vdots \\ \boldsymbol{\Omega}^{\star} \boldsymbol{P}_{r_{|\boldsymbol{r}|} c_{|\boldsymbol{c}|}} \end{bmatrix} \in \mathbb{R}^{K \times N} \tag{5.36}$$

being the *global* analysis operator that expands the patch based one to the entire image. Problems that arise at image boundaries are treated by employing constant padding, i.e. replicating the values at the image boundaries for $\lfloor \frac{\sqrt{n}}{2} \rfloor$ times. Certainly, for image processing applications $\boldsymbol{\Omega}^F$ is too large to be stored explicitly and applied in terms of a matrix vector multiplication. Fortunately, applying $\boldsymbol{\Omega}^F$ and its transposed can be implemented efficiently using sliding window techniques, and the matrix vector notation is solely used for legibility.

In addition to enforcing the reconstructed image to follow the co-sparse analysis model, according to [63], one can exploit the fact that the range of pixel intensities is limited by a lower bound b_l and an upper bound b_u. A simple way to enforce this bounding constraint is to minimize the differentiable function $B(\boldsymbol{s}) := \sum_{i=1}^{N} b(s_i)$, where b is a penalty term given as

$$b(s) = \begin{cases} |s - b_u|^2 & \text{if } s \geq b_u \\ |s - b_l|^2 & \text{if } s \leq b_l \\ 0 & \text{otherwise} \end{cases} . \tag{5.37}$$

Finally, combining the two penalty terms (5.35) and (5.37) with the data fidelity term, the analysis based image reconstruction problem employed here reads as

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s}} \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{s} - \boldsymbol{y}\|_2^2 + B(\boldsymbol{s}) + \lambda g(\boldsymbol{\Omega}^F \boldsymbol{s}). \tag{5.38}$$

Therein, $\lambda \in \mathbb{R}^+$ balances between the sparsity of the solution's analysis coefficients and the solution's fidelity to the measurements. The measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times N}$ and the measurements $\boldsymbol{y} \in \mathbb{R}^m$ depend on the application. In the next section, I evaluate how an operator learned by GOAL performs in image reconstruction applications.

5.6 Evaluation and Experiments on Real Image Data

The first part of this section aims at finding an answer towards the question of what is a good analysis operator for solving image reconstruction problems. Here, I try to answer this question by relating the image reconstruction quality of an analysis operator with its mutual coherence and its condition number. This in turn permits to select the optimal weighting parameters κ and μ for GOAL. Using the determined parameters, I learn one general analysis operator $\boldsymbol{\Omega}^\star$ by GOAL, and compare its image denoising performance with other analysis based approaches. In the second part, I utilize this operator $\boldsymbol{\Omega}^\star$ as a regularizer for solving the two classical image reconstruction problems of image inpainting and single image superresolution. For these two tasks, I compare the achieved results with employing the currently best performing analysis approach FoE [107], the best sparse synthesis based approach for the respective task, and some state-of-the-art methods specifically designed for each application. Note that here, I limit the evaluation to gray scale images, but the approach can be straightforwardly extended to color images.

5.6.1 Global Parameters Selection and Image Reconstruction

To quantify the reconstruction quality, as commonly done in the literature, I use the peak signal-to-noise ratio (PSNR) between the ground truth image and the recovered image, which is computed as given in Equation (4.39). Moreover, I measure the reconstruction quality using the Mean Structural SIMilarity Index (MSSIM) [130], with the same set of parameters as originally suggested in [130]. Compared to PSNR, the MSSIM better reflects a human observer's visual impression of quality. It ranges between zero and one, with one meaning perfect image reconstruction.

Throughout all experiments, I fixed the size of the image-patches to (8×8), i.e. $n = 64$. This is in accordance to the patch-sizes commonly used in the literature and yields a good trade-off between reconstruction quality and numerical burden. For all applications, an image is reconstructed by solving the minimization problem (5.38) via the conjugate gradient method proposed in [63]. Considering the pixel intensity bounds, I used $b_l = 0$ and $b_u = 255$, which is the common intensity range in 8-bit gray-scale image formats. The sparsity promoting function (5.35) with $p = 0.4$ and $\nu = 10^{-6}$ is used for both learning the analysis operator by GOAL, and for reconstructing the images. The patch based reconstruction algorithm as explained in Section 5.5 achieves the best results for the maximum possible overlap $d_h = d_v = 1$, which consequently is employed here. The Lagrange multiplier λ and the measurements matrix $\boldsymbol{A}$ depend on the respective application, and are briefly discussed in the corresponding subsections.

5.6.2 Analysis Operator Evaluation and Learning Parameter Selection

To compare the quality of distinctively learned analysis operators and to select appropriate parameters for GOAL, I chose the task of image denoising as a baseline experiment. The images to be denoised have artificially been corrupted by additive white Gaussian noise (AWGN) of varying standard deviation σ_{noise}. Besides helping to select the learning parameters, this baseline experiment is further used to compare GOAL with other analysis operator learning methods. I would like to emphasize that the choice of image denoising as a baseline experiment is not crucial, neither for selecting the learning parameters, nor for ranking the learning approaches. In fact, any other reconstruction task as discussed below leads to the same parameters and the same ranking of the different learning algorithms.

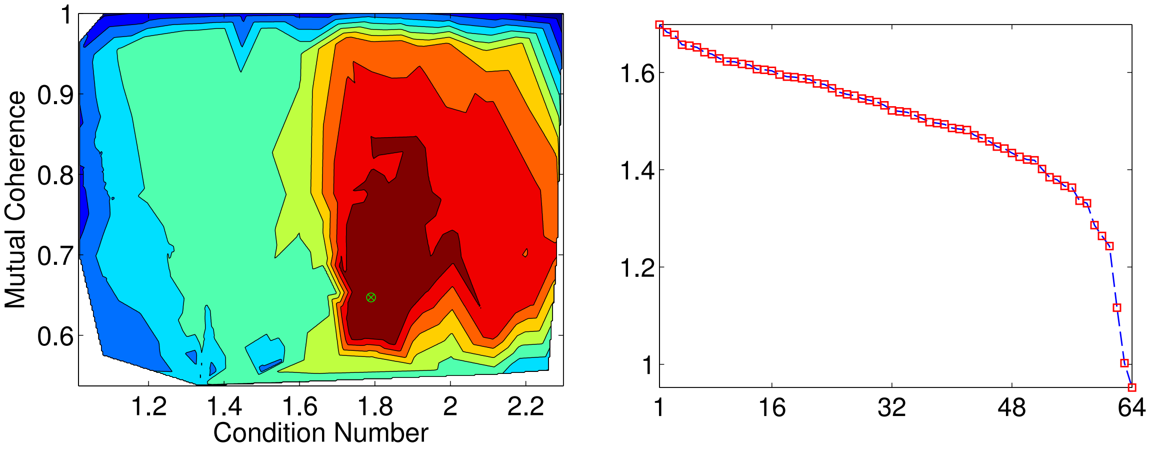
Considering the problem of image denoising, the measurement matrix $\boldsymbol{A}$ in Equation (5.38) is simply the identity matrix $\boldsymbol{I}_N$. As it is common in the denoising literature, the noise level σ_{noise} is assumed to be known and λ is adjusted accordingly; the larger it is chosen, the more noise is expected to be present. From my experiments, I found that $\lambda = \frac{\sigma_{\text{noise}}}{16}$ is a good choice. The reconstruction algorithm is terminated after $6-30$ CG-iterations depending on the noise level, i.e. the higher the noise level is the more iterations are performed. To find the best performing analysis operator, I learned several operators with varying values for μ, κ, and a with all other parameters being set according to Subsection 5.6.1. Then, I evaluated the performance of each resulting operator for the baseline experiment, which consists of denoising the five test images, each corrupted with the five noise levels as given in Table 5.2. This in total leads to 25 reconstruction results per operator. As the final performance measure, I use the average PSNR of the 25 results.

To train the operator, a set of $M = 200\,000$ image-patches was employed, with each patch normalized to have zero mean and unit Euclidean norm. These patches have randomly been extracted from the five training images shown in Figure 5.4. Certainly, these images are not considered within any of the following performance evaluations. All operators have been learned from the same training set. Each time, GOAL was initialized with a random matrix with normalized rows. Tests with other initializations like the overcomplete discrete cosine transform did not show remarkable influence on the final operator's performance.

Figure 5.4: This figure shows the five training images used for learning an analysis operator with GOAL.

From the achieved results I can conclude that image reconstruction tasks based on the analysis model clearly benefit from overcompleteness of the employed operator. The larger a is chosen, the better the operator performs with saturation starting at $a = 2n$. Therefore, I fixed the number of analysis atoms for all further experiments to $a = 2n$.

Regarding κ and μ, note that by Lemma 5.3 and Lemma 5.4 these parameters influence the condition number and the mutual coherence of a learned operator. Towards answering the question of what is a good and appropriate condition number and mutual coherence for an analysis operator, Figure 5.5(a) shows the relative denoising performance of 400 operators learned by GOAL in relation to their respective mutual coherence and condition number. It can be seen that operators with low condition number ~ 1.8 and moderate mutual coherence ~ 0.65 achieve the best performance. I would like to mention that according to my experiments, this relation is mostly independent from the degree of overcompleteness. As already expected from the synthetic experiments conducted in Section 5.4, the best learned analysis operator is not a uniformly normalized tight frame, as this constraint is too restrictive and prevents the operator from sufficiently sparsifying the training data. The concrete values, which led to the best performing analysis operator $\boldsymbol{\Omega}^\star \in \mathbb{R}^{128\times 64}$ are $\kappa = 9000$ and $\mu = 0.01$. The singular values of this operator are shown in Figure 5.5(b) and its atoms are visualized in Figure 5.6. This operator $\boldsymbol{\Omega}^\star$ remains unaltered throughout all following image processing experiments in Subsections 5.6.3 - 5.6.5.



(a) Performance Heatmap.

(b) Singular Values of $\boldsymbol{\Omega}^\star$.

Figure 5.5: This figure presents (a) the relative average denoising performance of 400 analysis operators learned by GOAL in relation to their mutual coherence and their condition number. Color ranges from dark blue (worst) to dark red (best), indicating the denoising performance. The green dot corresponds to the best performing operator $\boldsymbol{\Omega}^\star$. (b) Singular values of $\boldsymbol{\Omega}^\star$.

5.6.3 Image Denoising and Comparison with Related Analysis Operator Learning Methods

The purpose of this subsection is to rank GOAL among other analysis operator learning methods, and to compare its performance with state-of-the-art denoising algorithms. Concretely, I compare the denoising performance using $\boldsymbol{\Omega}^\star$ learned by GOAL with the finite difference operator for computing the total-variation (TV) [24] which is the currently best known analysis operator, with an operator learned by the recently proposed method AOL [139], and with the currently best performing analysis approach FoE [107]. Note that I used the same training set and the same level of overcompleteness for learning the operator by

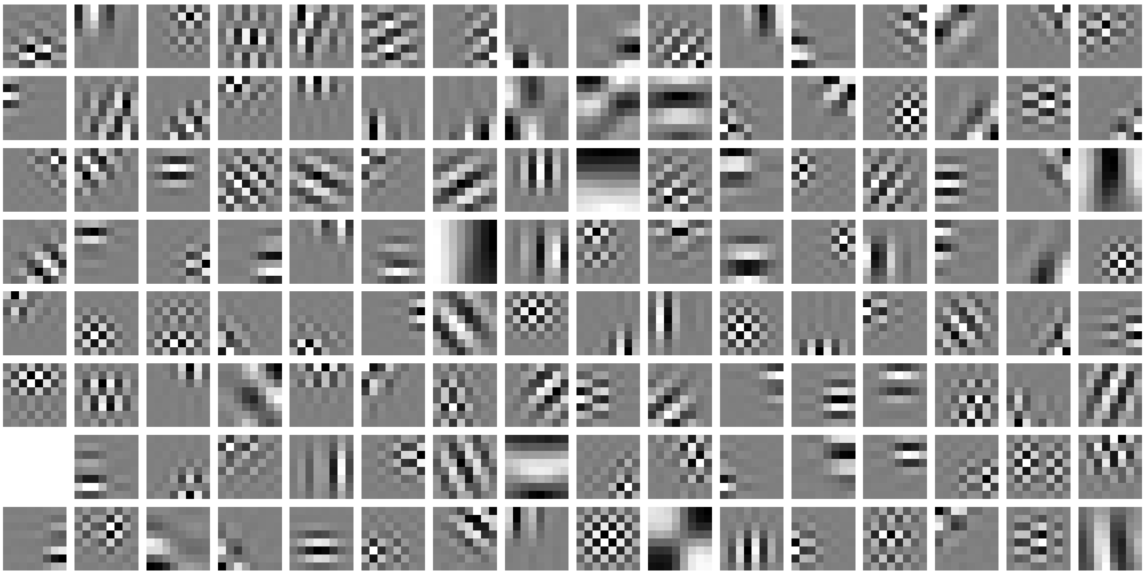


Figure 5.6: This figure shows analysis atoms of the learned analysis operator $\boldsymbol{\Omega}^{\star} \in \mathbb{R}^{128 \times 64}$. Each of the 128 analysis atoms is represented as an 8×8 square, where black corresponds to the smallest negative entry, gray is a zero entry, and white corresponds to the largest positive entry.

AOL as for GOAL. For FoE, I employed the setup as originally suggested by the authors. Concerning the required computation time for learning an analysis operator, for this setting GOAL needs about 4-minutes on an Intel i7 3.2 GHz six core CPU and 16 GB RAM. In contrast, AOL is approximately 40 times slower, and FoE is the computationally most expensive method requiring several hours. All three learning techniques were tested using unoptimized Matlab code.

The achieved results for the five test images and the five noise levels are given in Table 5.2. Employing the operator learned by GOAL achieves the best results among the analysis based methods both regarding PSNR and MSSIM. For a visual assessment, Figure 5.7 exemplarily shows four denoising results achieved by employing the four compared analysis operators. Visually, the operator learned by GOAL creates the most natural looking result.

To judge the denoising performances of the compared analysis methods globally, I additionally give the results achieved by the current state-of-the-art denoising method BM3D [23], and the synthesis model based K-SVD Denoising algorithm [43], which are specifically designed for the purpose of image denoising. In most of the cases my method performs slightly better than the K-SVD approach, especially for higher noise levels, and besides of

Table 5.2: This table presents the achieved PSNR in decibels (dB) and MSSIM for denoising five test images, each corrupted with five noise levels. Each cell contains the achieved results for the respective image with six different algorithms, which are: Top left GOAL, top right AOL [138], middle left TV [24], middle right FoE [107], bottom left K-SVD denoising [43], and bottom right BM3D [23].

	lena				barbara				man				boat				couple			
σ_{noise} / PSNR	PSNR		MSSIM		PSNR		MSSIM		PSNR		MSSIM		PSNR		MSSIM		PSNR		MSSIM	
5 / 34.15	38.65	36.51	0.945	0.924	37.96	35.95	0.962	0.944	37.77	35.91	0.954	0.932	37.09	35.77	0.938	0.926	37.43	35.55	0.951	0.932
	37.65	38.19	0.936	0.938	35.56	37.25	0.948	0.958	36.79	37.45	0.944	0.949	36.17	36.33	0.925	0.917	36.26	37.06	0.940	0.944
	38.48	38.45	0.944	0.942	38.12	38.27	0.964	0.964	37.51	37.79	0.952	0.954	37.14	37.25	0.939	0.938	37.24	37.14	0.950	0.951
10 / 28.13	35.58	32.20	0.910	0.856	33.98	31.27	0.930	0.883	33.88	31.33	0.907	0.851	33.72	31.24	0.883	0.842	33.75	30.87	0.903	0.844
	34.24	35.12	0.890	0.901	30.84	32.91	0.886	0.923	32.90	33.44	0.884	0.893	32.54	33.23	0.863	0.868	32.32	33.37	0.878	0.889
	35.52	35.79	0.910	0.915	34.56	34.96	0.936	0.942	33.64	33.97	0.901	0.907	33.68	33.91	0.883	0.887	33.62	33.86	0.901	0.909
20 / 22.11	32.63	28.50	0.869	0.772	30.17	27.26	0.880	0.791	30.44	27.33	0.831	0.720	30.62	27.16	0.819	0.711	30.39	26.95	0.833	0.727
	31.09	31.97	0.827	0.856	26.79	28.39	0.773	0.849	29.63	29.75	0.795	0.801	29.30	29.96	0.778	0.793	28.87	29.77	0.783	0.807
	32.39	32.98	0.861	0.875	30.87	31.78	0.881	0.905	30.17	30.59	0.814	0.833	30.44	30.89	0.805	0.825	30.08	30.68	0.817	0.847
25 / 20.17	31.65	27.47	0.854	0.742	29.05	26.08	0.856	0.750	29.43	26.28	0.801	0.677	29.61	26.08	0.792	0.671	29.32	25.81	0.802	0.679
	30.05	30.87	0.796	0.836	25.73	27.05	0.724	0.813	28.66	28.62	0.759	0.761	28.32	28.87	0.744	0.758	27.87	28.57	0.746	0.767
	31.33	32.02	0.842	0.859	29.59	30.72	0.850	0.887	29.14	29.62	0.780	0.804	29.36	29.92	0.772	0.801	28.92	29.65	0.780	0.820
30 / 18.59	30.86	26.50	0.839	0.717	27.93	24.95	0.818	0.706	28.64	25.30	0.774	0.638	28.80	25.07	0.769	0.630	28.46	24.79	0.780	0.633
	29.40	30.00	0.786	0.823	24.91	25.97	0.690	0.787	27.95	27.85	0.736	0.740	27.56	28.01	0.720	0.737	27.09	27.70	0.715	0.743
	30.44	31.22	0.823	0.843	28.56	29.82	0.821	0.868	28.30	28.87	0.750	0.780	28.48	29.13	0.744	0.779	27.95	28.81	0.746	0.795

the "barabara" image it is at most $\sim$ 0.5dB worse than BM3D. This relatively bad performance on the "barbara" image can be explained by the very special structure of this image that rarely occurs in natural images, and that is smoothed by the learned operator. To overcome this drawback, a more sophisticated training set selection could help.

5.6.4 Image Inpainting

In image inpainting as originally proposed in [10], the goal is to fill up a set of damaged or disturbing pixels such that the resulting image is visually appealing. This is necessary for example to restore damaged photographs, for removing disturbances caused by e.g. defective hardware, or for deleting unwanted objects. Typically, the positions of the pixels to be filled up are given a priori. In the present formulation, when $N - m$ pixels must be inpainted, this leads to a binary $(m \times N)$ dimensional measurements matrix $\boldsymbol{A}$, where each row contains exactly one entry equal to one and all other are zero. The position of the non-zero entry corresponds to the position of a pixel with known intensity. Hence, $\boldsymbol{A}$ reflects the available image information. Regarding λ, it can be utilized in a way that my method simultaneously inpaints missing pixels and denoises the remaining ones.

As an image inpainting example, I disturbed some ground truth images artificially by removing $N - m$ pixels randomly distributed over the entire image as exemplary shown in Figure 5.8(a). In that way, the reconstruction quality can be judged both visually and quantitatively. The data is assumed to be free of noise, and I empirically selected $\lambda = 10^{-2}$. Figures 5.8(b)-(d) show three exemplary results for reconstructing the "lena" image, given 10% of all pixels using the operator learned by GOAL, FoE, and the recently proposed syn-

Table 5.3: This table shows the achieved results for inpainting five test images with varying number of missing pixels using three different methods. In each cell, the PSNR in dB and the MSSIM are given for GOAL (top), FoE [107](middle), and method [144] (bottom).

% of missing pixels	lena		barbara		boat		man		house	
	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
90%	28.57	0.840	22.61	0.696	25.61	0.743	26.35	0.755	28.35	0.828
	28.06	0.822	22.45	0.682	25.14	0.719	26.23	0.747	28.18	0.828
	27.63	0.804	22.49	0.658	24.80	0.683	25.56	0.715	26.62	0.784
80%	31.82	0.895	24.90	0.814	28.55	0.833	28.93	0.847	32.00	0.887
	31.09	0.880	23.48	0.762	27.76	0.804	28.51	0.836	31.36	0.880
	30.95	0.878	24.72	0.780	27.80	0.804	28.24	0.821	30.20	0.874
50%	37.75	0.956	34.51	0.965	34.47	0.936	34.12	0.947	38.89	0.961
	36.70	0.947	28.64	0.919	33.17	0.907	33.49	0.940	37.72	0.957
	36.75	0.943	33.21	0.953	33.77	0.918	33.27	0.934	38.11	0.960
20%	43.53	0.985	42.12	0.991	41.04	0.982	40.15	0.985	45.43	0.990
	42.29	0.981	36.03	0.981	38.45	0.963	39.15	0.982	44.21	0.989
	40.77	0.965	40.63	0.983	39.45	0.966	39.06	0.977	42.95	0.978

thesis model based method [144]. In table 5.3 the results for inpainting further images from varying numbers of missing pixels is given. It can be seen that the proposed methods performs best among the compared approaches, independent of the respective configuration.

5.6.5 Single Image Superresolution

In single image superresolution (SR), the goal is to reconstruct a high-resolution image $\boldsymbol{s} \in \mathbb{R}^N$ from an observed low-resolution image $\boldsymbol{y} \in \mathbb{R}^m$, where $N > m$. The low-resolution image $\boldsymbol{y}$ is assumed to be a low-pass filtered, i.e. blured, and downsampled version of $\boldsymbol{s}$. Mathematically, this process can be formulated as $\boldsymbol{y} = \boldsymbol{QBs} + \boldsymbol{\epsilon}$, where $\boldsymbol{Q} \in \mathbb{R}^{m \times N}$ is a decimation operator and $\boldsymbol{B} \in \mathbb{R}^{N \times N}$ is a low-pass or blur operator. Hence, the associated measurement matrix is given by $\boldsymbol{A} = \boldsymbol{QB}$. In the ideal case, the exact blur kernel is known or an estimate is given. Here, I consider the more realistic case of an unknown kernel and employ a general blur model. Concretely, to apply my approach for magnifying an image by a factor of s in both vertical and horizontal dimension, I model the blur via a Gaussian kernel of dimension $((2s-1) \times (2s-1))$ with standard deviation $\sigma_{\text{blur}} = \frac{s}{3}$.

For the conducted experiments, I first artificially created a low-resolution image by downsampling a ground truth image by a factor of s using bicubic interpolation. Then, I employed the five different methods Bicubic interpolation, FoE [107], the method from [140], and the analysis model with $\boldsymbol{\Omega}^\star$ to magnify each artificially created low-resolution image by the same factor s. This upsampled version is then again compared with the original

Table 5.4: This table presents the results in terms of PSNR and MSSIM for upsampling the seven test images by a factor of $s = 3$ using four different algorithms GOAL, FoE [107], method [140], and Bicubic interpolation.

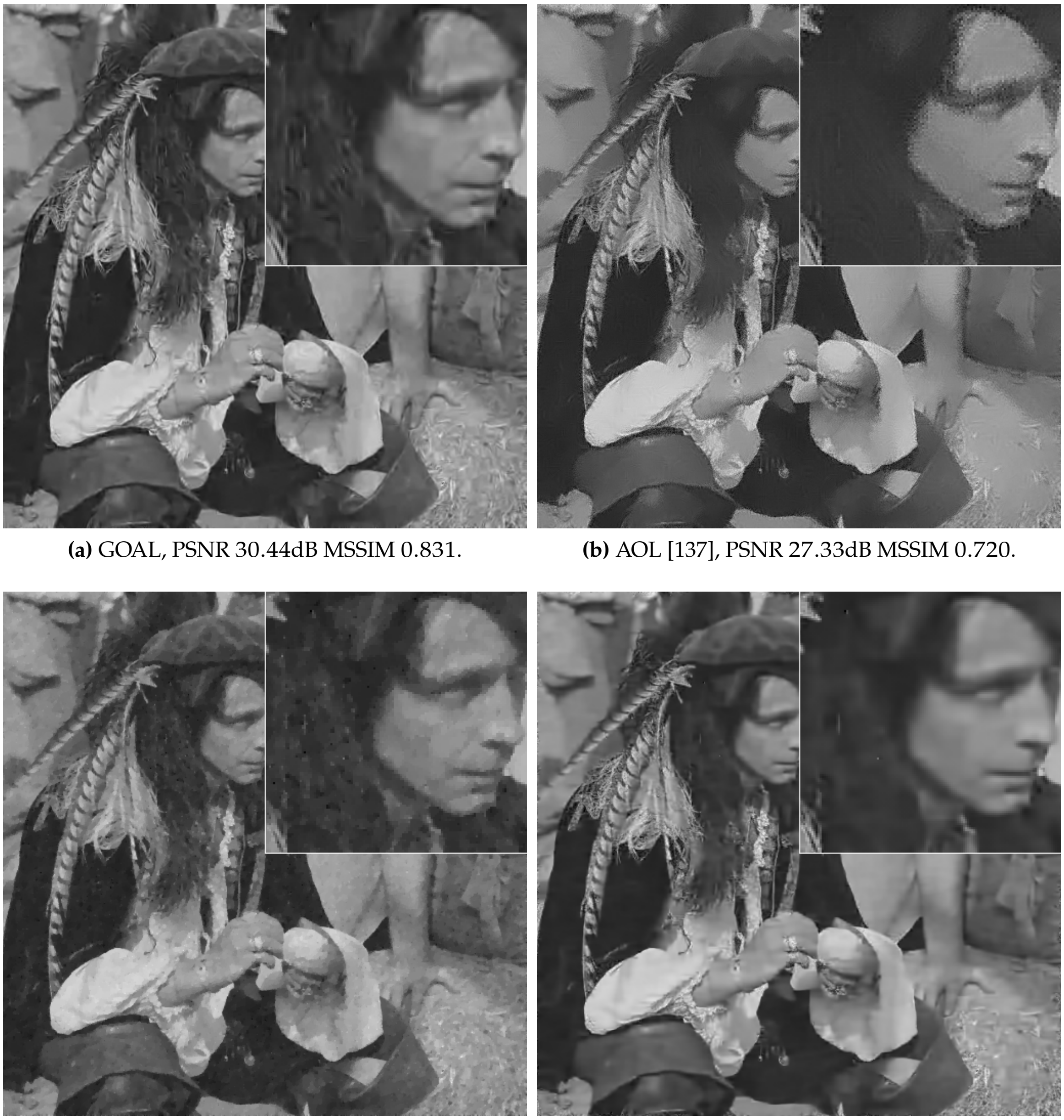
Method	face		august		barbara		lena		man		boat		couple	
	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
GOAL	32.37	0.801	23.28	0.791	24.42	0.731	32.36	0.889	29.48	0.837	28.25	0.800	27.79	0.786
FoE	32.19	0.797	22.95	0.782	24.30	0.727	31.82	0.885	29.17	0.832	28.00	0.797	27.64	0.782
Method [140]	32.16	0.795	22.90	0.771	24.25	0.719	32.00	0.881	29.29	0.829	28.04	0.793	27.56	0.778
Bicubic	31.57	0.771	22.07	0.724	24.13	0.703	30.81	0.863	28.39	0.796	27.18	0.759	26.92	0.743

image in terms of PSNR and MSSIM in order to judge the reconstruction quality. Table 5.4 presents the achieved results for upsampling the respective images by a factor of $s = 3$. The presented results show that the analysis based image reconstruction employing an operator learned by GOAL outperforms the current state-of-the-art. I want to emphasize again that the blur kernel used for downsampling is different from the blur kernel used within the upsampling procedure.

Note that many single image superresolution algorithms rely on clean noise free input data, whereas the general analysis approach as formulated in Problem (5.38) naturally handles noisy data and is able to simultaneously upsample and denoise an image. In Figure 5.9, I present the result for simultaneously denoising and upsampling a low-resolution version of the image "august" by a factor of $s = 3$, which has been corrupted by AWGN with $\sigma_{\text{noise}} = 8$. As it can be seen, employing the analysis operator learned by GOAL produces the best results both visually and quantitatively, especially regarding the MSSIM. Due to high texture, this image is challenging to upscale even when no noise is present, which can be seen from the second column of Table 5.4.

5.7 Summary

This chapter dealt with the topic of learning an analysis operator from example signals and introduced how to apply a patch based operator for solving inverse problems in image processing. To learn the operator, I developed the novel algorithm GOAL, which is based on a ℓ_p-minimization on the set of full-rank matrices with normalized columns. A geometric conjugate gradient method on the oblique manifold was suggested to efficiently solve the associated optimization problem. Furthermore, I proposed a partitioning invariant method for employing a local patch based analysis operator such that globally consistent image reconstruction results are achieved. A series of synthetic experiments revealed that GOAL outperforms all existing analysis operator learning techniques in terms of computational complexity, ability to find a generating ground truth operator, and generality. To answer the question of what characterizes a well performing analysis operator in image processing applications, I related the mutual coherence and the condition number of an operator with its performance when employed for the task of image denoising. From this I concluded that a good operator should be well-conditioned and have moderate mutual coherence. Besides that, I compared an operator learned by the proposed method with several operators learned by state-of-the-art analysis operator learning algorithms for the task of image denoising. In these experiments, the operator learned by GOAL consistently outperforms all others. For the classical image processing tasks of image inpainting and single image superresolution, I provided promising results that are competitive with, and even outperform current state-of-the-art techniques. Similar as for the synthesis signal reconstruction model with dictionaries, I expect that the performance of the analysis approach can be further increased by learning the particular operator with regards to the specific problem at hand, or by employing a specialized training set. This assumption is already supported by a publication of my colleagues and me that investigates the topic of Analysis Based Blind Compressive Sensing [131].



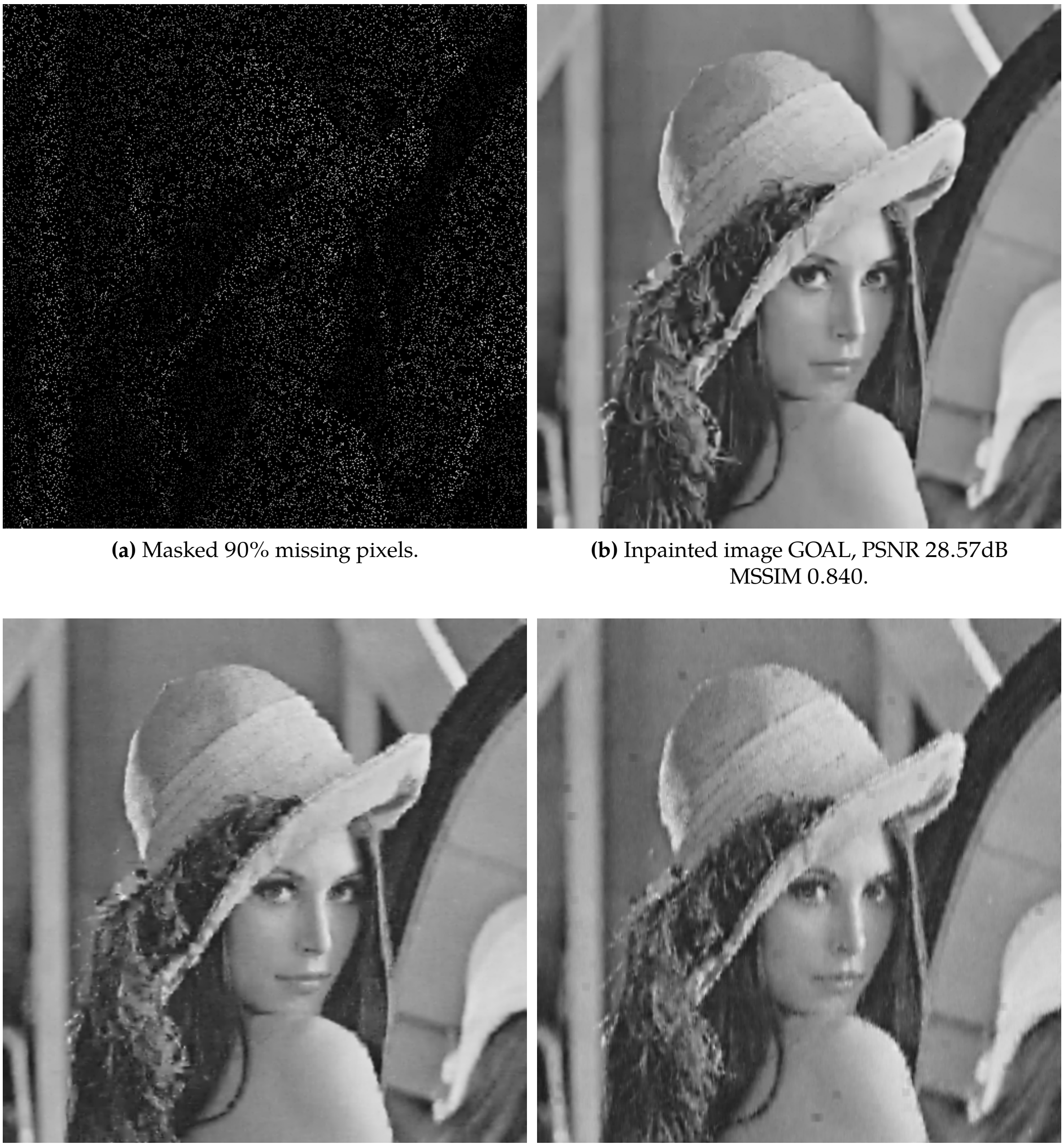
(a) GOAL, PSNR 30.44dB MSSIM 0.831.

(b) AOL [137], PSNR 27.33dB MSSIM 0.720.

(c) TV [24], PSNR 29.63dB MSSIM 0.795.

(d) FoE [107], PSNR 29.75dB MSSIM 0.801.

Figure 5.7: This figure presents four images that exemplarily show the artifacts typically created by denoising an image using the four compared analysis operators. The image shown here is the "man" image, which has been degraded by $\sigma_{\text{noise}} = 20$. A close up is provided for each image for a better visualization.



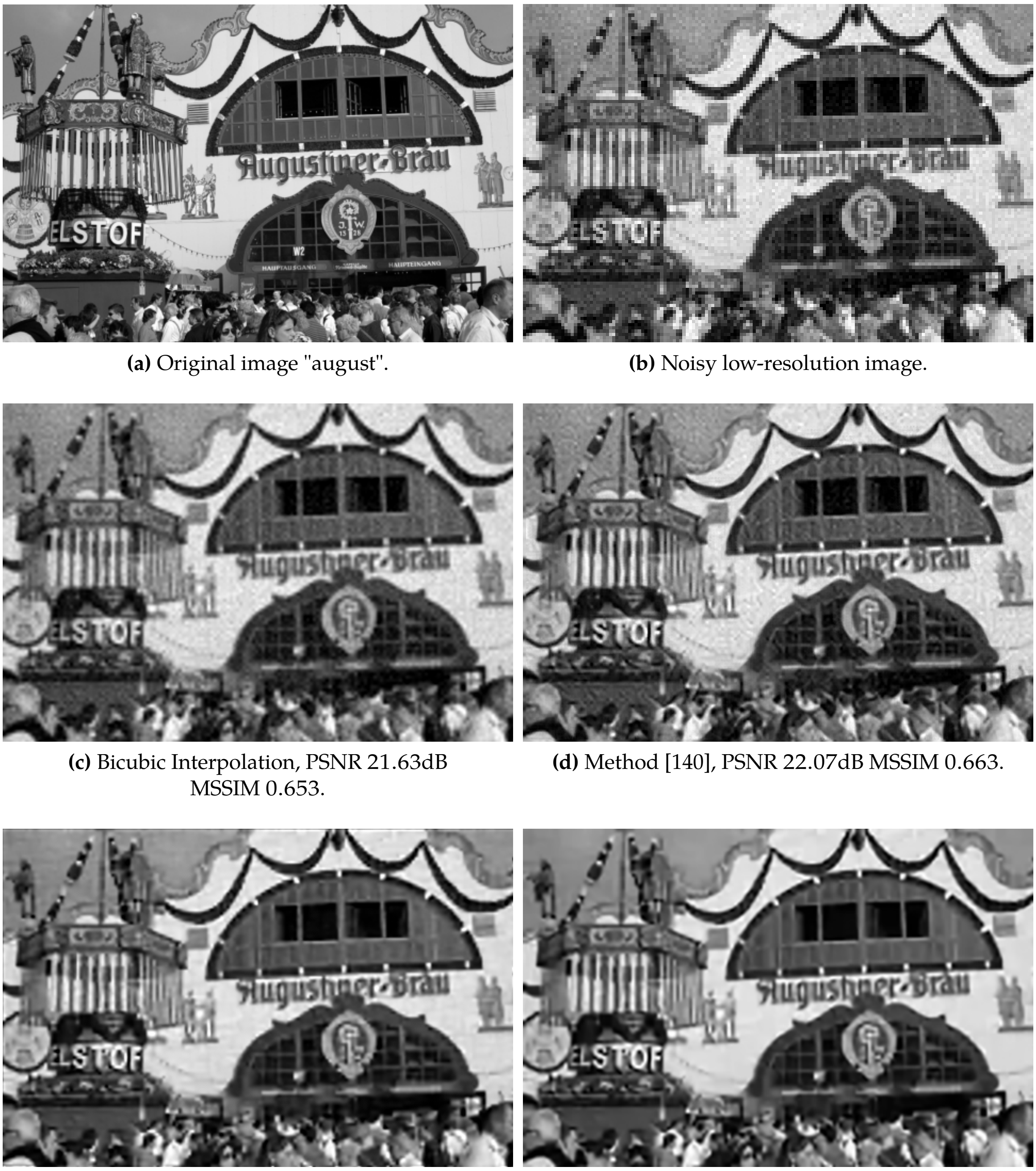
(a) Masked 90% missing pixels.

(b) Inpainted image GOAL, PSNR 28.57dB MSSIM 0.840.

(c) Inpainted image FoE, PSNR 28.06dB MSSIM 0.822.

(d) Inpainted image [144], PSNR 27.63dB MSSIM 0.804.

Figure 5.8: This figure shows the achieved results for filling up missing pixels of the "lena" image from 10% of all pixels using $\boldsymbol{\Omega}^*$ learned by GOAL (b), FoE (c), and [144] (d).



(a) Original image "august".

(b) Noisy low-resolution image.

(c) Bicubic Interpolation, PSNR 21.63dB MSSIM 0.653.

(d) Method [140], PSNR 22.07dB MSSIM 0.663.

(e) FoE [107], PSNR 22.17dB MSSIM 0.711.

(f) GOAL, PSNR 22.45dB MSSIM 0.726.

Figure 5.9: This figure shows single image superresolution results on noisy data of four algorithms that are (c) Bicubic interpolation (d) method [140] (e) FoE [107], and (f) GOAL, for magnifying a low-resolution version of the "august" image by a factor of three. In the captions of each image, the corresponding PSNR and MSSIM are given. The low-resolution image has been corrupted by AWGN with $\sigma_{\text{noise}} = 8$.

Chapter 6

Multimodal Co-Sparse Analysis Model

This chapter is partially based on the submitted work:

> Joint Intensity and Depth Analysis Operator Learning for Depth Map Superresolution. *Submitted to IEEE International Conference on Computer Vision*, 2013.
>
> S. Hawe, M. Kleinsteuber, and K. Diepold. Analysis Operator Learning and Its Application to Image Reconstruction. In *IEEE Transactions on Image Processing*, 22(6), pp. 2138--2150. 2013.

This chapter introduces the *Multimodal Co-Sparse Analysis Model*, which is able to capture and describe the interdependencies between diverse measurements from the same scene or object that have been acquired in different modalities. The underlying assumption of this model is that such related measurements share a common co-support with respect to a suitable set of analysis operators. This set of operators is called the multimodal analysis operator, which forms the core of the proposed data model. For this operator, no analytic form exists; consequently, it must be learned from aligned multimodal example signals. To that end, I propose an efficient conjugate gradient method for minimizing a smooth cost function on the oblique manifold, which is basically an adaptation of GOAL to the mutlimodal signal setting. This learning process can be done off-line, and returns an application independent multimodal analysis operator. This operator allows exploiting the multimodal co-sparse analysis model as a prior for solving diverse linear inverse problems.

As a driving application example, I use the two modalities image intensity and scene depth and explain how the proposed model can be exploited to infer a high-resolution depth map from low-resolution depth measurements given a corresponding and registered high-resolution intensity image. By a set of numerical examples I show that the arising algorithm achieves state-of-the-art performance for the task of depth map superresolution.

Finally, note that the proposed data model is not limited to this special bi-modal application, but is generally applicable to any combination of signal modalities as for example in the field of medical image processing, where a patient's treatment is often based on information gained from several measurements acquired with different imaging technologies like computer tomography, X-ray, or ultrasound.

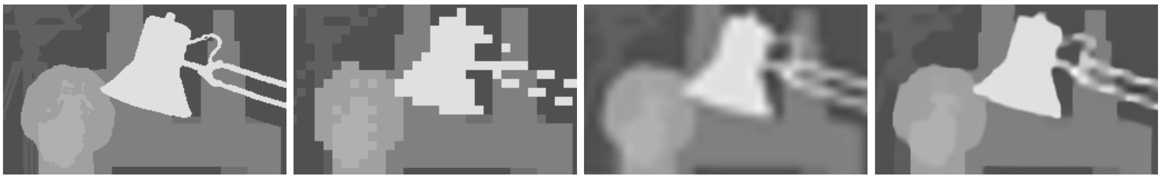
6.1 Introduction

Measurements that have been acquired simultaneously from diverse modalities such as sound, temperature, or light all carry different information about the same scene or object. Even though the information itself contained in the measurements is different, the measurements might share statistical dependencies. Knowing these dependencies can help to combine the measurements in order to gain a more complete and thorough understanding and description of the underlying scene.

One very prominent example of such related modalities that I cover intensively here is the pair of intensity images and depth measurements. Combing these measurements permits to create texturized 3D scene models. The need for having such 3D models with high quality can be seen from the vast amount of technical applications in fields like robotics, 3D video rendering, or human computer interaction that are built upon them. While the visual information is gathered in high quality by standard cameras, the 3D scene information is typically acquired in rather low or moderate quality either via passive or active range sensors, which both have different strengths an weaknesses.

Passive range sensing, i.e. 3D from stereo intensity images, is based on essentially three steps. First, ambient light that is reflected from the same object surface is captured at multiple displaced views. Second, the disparities of corresponding light intensity samples between the different views are determined. Third, the distance to the sensor is obtained using the computed disparities together with the knowledge of the relative positions between all views. Despite very active research in this area and significant improvements regarding the quality of the depth maps over the past years, stereo methods still have difficulties to cope with noise, texture-less regions, repetitive texture, and occluded areas. For an overview of stereo methods, I refer the reader to [116].

Active range sensors, on the other hand, emit light and either measure the time-of-flight of a modulated ray, e.g. LIDAR or PMD, or capture the reflection pattern of a structured light source to infer the distance to objects, as it is done for example by the well-known Microsoft Kinect. Because active sensors acquire reliable depth measurements independent of the



(a) Ground truth. **(b)** Nearest-neighbor interpolation. **(c)** Bicubic interpolation. **(d)** Proposed method.

Figure 6.1: This figure presents a visual comparison of different upscaling methods on a close up of the test image Tsukuba from [113], which has been downsampled and then remagnified by a factor of eight in both vertical and horizontal direction.

occurring texture, and due to their real-time capability, they are becoming more and more popular in both industrial and academical applications. However, the main drawbacks are that the resulting depth maps are of low-resolution (LR) and that they are relatively noisy. To overcome these limitations, different methods for upsampling, inpainting, and denoising LR depth maps from range sensors have been proposed through the last years, see Section 6.3.3.

The fact that both ambient and artificially emitted light is reflected by the same object surface naturally suggests a co-occurrence of signal patterns in both the depth map obtained by an active range sensor as well as in a corresponding registered camera intensity image. Indeed, some of the most successful methods for reconstructing and refining depth maps aim at exploiting this statistical dependency.

Inspired by the success of signal reconstruction based on sparse data representations as detailed in the previous chapters, I introduce the multimodal co-sparse analysis model that is able to reveal dependencies between different but related signal modalities. The assumption that underlies the proposed model is that measurements, which originate from the same scene, share a common co-support with respect to a suitable set of analysis operators. For this set or operators, no analytic form exists and it must be determined from aligned training data. Therefore, I propose a Riemannian conjugate gradient algorithm on the oblique manifold, which basically is an extension of the algorithm GOAL proposed in Chapter 5 to the mutimodal signal scenario. The introduced data model can be exploited in several signal processing tasks in a way that information obtained from different modalities can support each other. To show its performance in applications, I present a depth map reconstruction task. To that end, an operator is learned once off-line and is then used in conjunction with a high-resolution (HR) intensity image to reconstruct a corresponding HR depth map from low resolution depth measurements. The numerical experiments show that the proposed approach outperforms state-of-the-art methods both visually and quantitatively, and they underpin the validity of this novel data model.

6.2 Multimodal Co-Sparse Analysis Model

6.2.1 Model Assumption

The problem I want to tackle here is the following. Given c measurements $\{\boldsymbol{y}_i \in \mathbb{R}^{m_i}\}_{i=1}^{c}$ of an object, which have been acquired in different modalities and that might by corrupted or incomplete, how can the corresponding high quality signals $\{\boldsymbol{s}_i \in \mathbb{R}^{n_i}\}_{i=1}^{c}$ of probably different dimensions n_i be accurately reconstructed using the available measurements? By incomplete, I mean that the measurements are of smaller dimensions than corresponding signals, i.e. $m_i < n_i$. Formally, the relation between $\boldsymbol{s}_i$ and $\boldsymbol{y}_i$ is given by

$$\boldsymbol{y}_i = \boldsymbol{A}_i \boldsymbol{s}_i + \boldsymbol{\epsilon}_i, \tag{6.1}$$

with $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times n_i}$ being the measurement matrix modeling the sampling process for the i-th signal modality, and $\boldsymbol{\epsilon}_i \in \mathbb{R}^{m_i}$ being the corresponding error vector modeling noise and potential sampling errors. Since the measurements are noisy and/or incomplete, inverting Equation (6.1) to reconstruct $\boldsymbol{s}_i$ is a highly ill-posed problem. Again, using additional information about the structure of the signal helps to regularize this linear inverse problem and to determine a feasible solution. Here, my goal is to exploit the geometry of the previously introduced co-sparse analysis model [85] to solve this signal reconstruction task.

Recall that the idea which underlies the co-sparse analysis model is that a signal $\boldsymbol{s}$ multiplied by an analysis operator $\boldsymbol{\Omega} \in \mathbb{R}^{a \times n}$ with $a \geq n$ results in a sparse vector $\boldsymbol{\alpha} = \boldsymbol{\Omega s} \in \mathbb{R}^{a}$. If $g \colon \mathbb{R}^a \to \mathbb{R}$ again denotes a sparsity-inducing function, the analysis model assumption can be utilized to recover a single signal via

$$\boldsymbol{s}^\star \in \arg\min_{\boldsymbol{s}} g(\boldsymbol{\Omega s}) \quad \text{subject to} \quad f(\boldsymbol{A s} - \boldsymbol{y}) \leq \epsilon, \tag{6.2}$$

with $f : \mathbb{R}^m \to \mathbb{R}$ being a function that reflects the assumed noise characteristics and $\epsilon \in \mathbb{R}_0^+$ being an estimate of how strongly the measurements are corrupted.

Now, remember that the original problem stated at the beginning of the section was not to recover a single signal, but c different signals in different modalities. A straightforward approach for achieving this is to solve Problem (6.2) for each signal modality independently. However, this approach completely discards the fact that the signal modalities represent the same object, and thus, might be statistically dependent. To that end, I want to find an answer to the question, how these statistical dependencies can be revealed and consequently how they can be exploited to achieve better signal reconstruction results when related modalities are processed jointly rather than individually.

In order to establish a notion of statistical dependency, remember that in the co-sparse analysis model the *zero entries* of the analyzed vector $\boldsymbol{\alpha}$ determine the structure of a signal [85]. Geometrically, $\boldsymbol{s}$ lies in the intersection of all hyperplanes whose normal vectors are given by the rows of $\boldsymbol{\Omega}$ indexed by the zero entries of $\boldsymbol{\alpha}$. This index set is called the *co-support* of $\boldsymbol{s}$, and it is formally given by

$$\text{co-supp}(\boldsymbol{\Omega s}) := \{j \mid (\boldsymbol{\Omega s})_j = 0\}. \tag{6.3}$$

Next, assume that all c signal modalities $\{\boldsymbol{s}_i \in \mathbb{R}^{n_i}\}_{i=1}^{c}$ allow a co-sparse representation with an appropriate set of corresponding analysis operators $\{\boldsymbol{\Omega}_i \in \mathbb{R}^{a \times n_i}\}_{i=1}^{c}$. Based on the knowledge that the structure of a signal is encoded in its co-support (6.3), I postulate that *a suitable set of analysis operators can be found such that the co-supports of* $\{\boldsymbol{\Omega}_i\boldsymbol{s}_i \in \mathbb{R}^{a}\}_{i=1}^{c}$ *are statistically dependent, if all analyzed signal modalities originate from the same object*. Formally, the multimodal co-sparse analysis model assumes that the conditional probability that index j belongs to the co-supports of $\boldsymbol{s}_i$, $\forall i \neq r$ given that j belongs to the co-support of $\boldsymbol{s}_r$ is significantly higher than the unconditional probabilities, i.e.

$$Pr\left(j \in \text{co-supp}(\boldsymbol{\Omega}_i\boldsymbol{s}_i) \mid j \in \text{co-supp}(\boldsymbol{\Omega}_r\boldsymbol{s}_r)\right) \gg Pr\left(j \in \text{co-supp}(\boldsymbol{\Omega}_r\boldsymbol{s}_r)\right), \quad \forall i \neq r. \tag{6.4}$$

Figure 6.2 visualizes the described multimodal signal model assumption. Note that even if the dimensions of the signals are distinctive, the operators *must* map them into analysis spaces of equal dimension a, otherwise my model assumption would not be applicable.

Clearly, this model is idealized since in practice the entries of the analyzed vectors are not exactly equal to zero but rather small in magnitude. Taking all that into account, condition (6.4) can be implemented through minimizing a function that enforces both all c analyzed signals $\boldsymbol{\Omega}_i\boldsymbol{s}_i$, $\forall i = 1, \dots, c$ to be sparse as well as *that the zeros or small entries of all corresponding analyzed signals occur at the same positions*. With the general sparsity promoting function

$$\boldsymbol{x} \mapsto \sum_{k=1}^{a} \ln(1 + \nu x_k^2), \tag{6.5}$$

where $\nu \in \mathbb{R}^+$ is a positive smoothing parameter, the function I suggest here for measuring

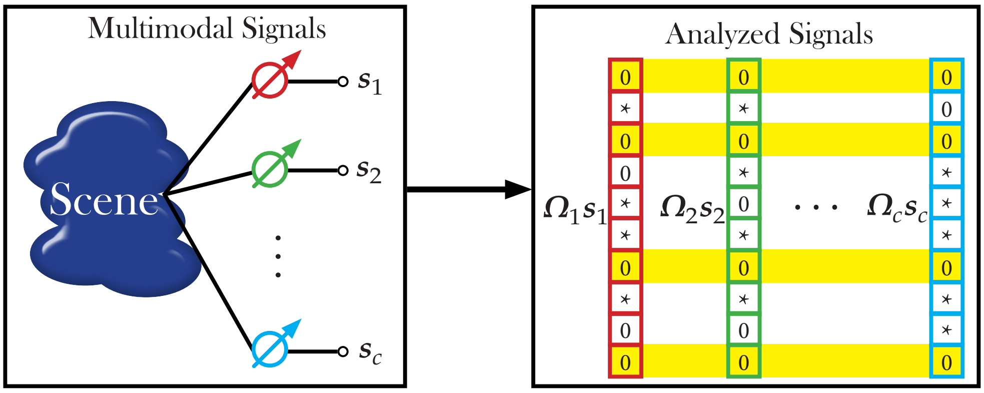


Figure 6.2: This figure depicts the coupled co-support assumption that underlies the multimodal co-sparse analysis model introduced in this chapter. The coupled co-support of the analyzed signals is highlighted in yellow.

and enforcing the co-support coupling reads as

$$g_{Co}(\{\boldsymbol{\Omega}_i \boldsymbol{s}_i\}_{i=1}^{c}) := \sum_{k=1}^{a} \ln\left(1 + \nu \sum_{i=1}^{c} (\boldsymbol{\Omega}_i \boldsymbol{s}_i)_k^2\right). \tag{6.6}$$

Basically, Equation (6.6) simply measures the sparsity of a vector whose i-th entry is equal to the sum of the squares of the i-th entries of all c considered analyzed vectors. Note that any smooth sparsity measure other than (6.5) could also be employed here. Finally, using Equation (6.6) together with the standard single modal analysis model (6.2), the multimodal analysis signal model is given as

$$\{\boldsymbol{s}_i^{\star}\}_{i=1}^{c} \in \arg\min_{\{\boldsymbol{s}_i\}_{i=1}^{c}} g_{Co}(\{\boldsymbol{\Omega}_i \boldsymbol{s}_i\}_{i=1}^{c}) \quad \text{subject to} \quad f(\boldsymbol{A}_i \boldsymbol{s}_i - \boldsymbol{y}_i) \leq \epsilon_i, \forall i = 1, \ldots, c, \tag{6.7}$$

with one pair of measurement matrix and measurements $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times n_i}, \boldsymbol{y}_i \in \mathbb{R}^{m_i}$ associated with each signal modality $\boldsymbol{s}_i$.

Most important for the multimodal co-sparse analysis model is the set of multimodal analysis operators that yields analyzed vectors with a coupled co-support. In the next section, I explain how this set can be jointly learned from training data, such that aligned signals analyzed by these operators adhere to the introduced model.

6.2.2 Multimodal Analysis Operator Learning

Regarding the choice of the multimodal analysis operator, it is not possible to simply choose a set of analytic operators or operators that have been learned independently on the basis of example signals of each modality, as those operators will most certainly not result in analyzed vectors whose co-supports are correlated. Due to this, it is necessary to learn the operators jointly such that the correlated co-support assumption is fulfilled. Among existing analysis operator learning algorithms such as [109, 139], only the method GOAL proposed here in Chapter 5 can straightforwardly be extended to the consider setting, since it is based on a standard conjugate gradient method on manifolds that directly allows to integrate any further smooth regularization term. For this reason, I again follow this approach here, and extend it such that a jointly co-sparse representation of related signals can be found according to the suggested multimodal co-sparse analysis model.

To learn the multimodal analysis operator $\{\boldsymbol{\Omega}_i \in \mathbb{R}^{a \times n_i}\}_{i=1}^c$, I employ a set of M training sets $\left\{\{\mathbf{s}_{i,j} \in \mathbb{R}^{n_i}\}_{i=1}^c\right\}_{j=1}^M$, where the c elements of the j-th set correspond to measurements of the same object but acquired in c different modalities. For the subsequently considered application of intensity and depth processing, these measurements are couples of HR intensity and HR depth patches representing the same excerpt of a scene. Now, my goal is as follows: Given the M training sets, find the c operators such that $g_{Co}(\{\boldsymbol{\Omega}_i \mathbf{s}_{i,j}\}_{i=1}^c)$ is maximally sparse for all M sets. Naturally, this can be achieved by minimizing the expectation of Function (6.6) over the entire training set. However, based on the arguments provided Section 5.2.2, I instead employ the sum of squares of Equation (6.6), which can be interpreted as a balanced optimization over both the expectation *and* the variance. This avoids the operator from being overfitted to a possibly dominant subset of training samples that shows similar structure, and also results in increased performance when the operator is employed in real world applications. With this, I end up with

$$G(\{\boldsymbol{\Omega}_i\}_{i=1}^c) := \frac{1}{M} \sum_{j=1}^{M} g_{Co}(\{\boldsymbol{\Omega}_i \mathbf{s}_{i,j}\}_{i=1}^c)^2, \tag{6.8}$$

as the coupled sparsifying function to be minimized for the complete training set.

To regularize the training process as in the algorithm GOAL introduced in Chapter 5, I restrict the set of possible solutions of the *transposed* of a single analysis operator to the oblique manifold $\mathrm{OB}(n, a)$, which allows an efficient formulation of the multimodal analysis operator learning task as a constrained optimization problem that directly exploits the underlying matrix manifold's geometry. To adhere to the rank constraint of OB, I again employ the penalty function $h : \mathbb{R}^{a \times n} \to \mathbb{R}^+$ as given in Equation (5.25). Furthermore, I utilize the penalty function $r : \mathbb{R}^{a \times n} \to \mathbb{R}^+$ as given in Equation (5.17) to enforce the resulting operators to have distinctive rows and to control their mutual coherence. Now putting all collected ingredients together, the problem of jointly learning the set of multimodal analysis operators is given by

$$\{\boldsymbol{\Omega}_i^\top\}_{i=1}^c \in \arg \min_{\{X_i \in \mathrm{OB}(n_i,a)\}_{i=1}^c} G(\{\boldsymbol{X}_i^\top\}_{i=1}^c) + \sum_{i=1}^c \kappa h(\boldsymbol{X}_i^\top) + \mu r(\boldsymbol{X}_i^\top), \tag{6.9}$$

with $\kappa, \mu \in \mathbb{R}^+$ being positive weighting factors. Regarding the influence of the employed penalty terms and the weighting factors on the resulting analysis operators, refer to Section 5.2.2. The arising optimization problem is solved using the geometric conjugate gradient method with the hybrid CG-update parameter formula (4.32) and standard backtracking line search as explained in Section 5.3.

6.3 Joint Intensity and Depth Processing

In this section, I explain how a pair of patch based multimodal analysis operators $\boldsymbol{\Omega}_I \in \mathbb{R}^{a \times n_I}, \boldsymbol{\Omega}_D \in \mathbb{R}^{a \times n_D}$ learned via the procedure introduced in the previous section can be used to jointly reconstruct an aligned pair of intensity and depth signals $\boldsymbol{s}_I \in \mathbb{R}^{N_I}, \boldsymbol{s}_D \in \mathbb{R}^{N_D}$ from corresponding measurements $\boldsymbol{y}_I \in \mathbb{R}^{m_I}, \boldsymbol{y}_D \in \mathbb{R}^{m_D}$. Furthermore, I present a method for selecting an informative training set for learning the pair of operators. This sophisticated selection scheme is necessary due to the special nature of depth maps where a simple training set selection based on randomly drawing patches from ground truth depth maps results in a suboptimal optimal set of operators. Through a large number of experiments, I found that the proposed selection scheme improves the overall reconstruction performance of the operators. For legibility reasons, in the remainder of this chapter I assume that both operators are applied on signals of equal dimension and that both HR signals have the same dimensions, i.e. $n_D = n_I := n$ and $N_D = N_I := N$.

6.3.1 Reconstruction Algorithm

First, note that $\boldsymbol{s}_I$ and $\boldsymbol{s}_D$ are the vectorized versions of a full HR intensity image and a full HR depth map, respectively, obtained by ordering their entries lexicographically, with $N = wh$ where w and h denote the height and width of both HR signals. Now, recall that an analysis operator has to be applied locally to n-dimensional patches rather than globally to the complete N-dimensional signal. Instead of reconstructing each patch individually and combining the patches in a final step to form the signal, the complete N-dimensional signal is reconstructed such that neighboring patches support each other during the optimization process. Concretely, I require that each entry of a signal is reconstructed such that the average sparsity of all analyzed patches it belongs to is minimal. To that end, I employ the same procedure as introduced in Section 5.5 based on a global analysis operator $\boldsymbol{\Omega}^F$ created from a patch based one as given in Equation (5.36). To deal with signal boundary problems, I follow the well-known reflective boundary conditions. It should be mentioned that applying $\boldsymbol{\Omega}^F$ can be efficiently implemented using image filtering techniques.

With two global operators $\boldsymbol{\Omega}_I^F$ and $\boldsymbol{\Omega}_D^F$ constructed from the corresponding patch based operators $\boldsymbol{\Omega}_I$ and $\boldsymbol{\Omega}_D$ as in Equation (5.36), I reformulate the multimodal co-sparse analysis model (6.7) for the special case of joint intensity and depth reconstruction in unconstrained Lagrangian form as

$$\{\boldsymbol{s}_I^\star, \boldsymbol{s}_D^\star\} \in \arg\min_{\boldsymbol{s}_I, \boldsymbol{s}_D} \tfrac{1}{2}\|\boldsymbol{A}_I\boldsymbol{s}_I - \boldsymbol{y}_I\|_2^2 + \tfrac{1}{2}\|\boldsymbol{A}_D\boldsymbol{s}_D - \boldsymbol{y}_D\|_2^2 + \lambda g_{Co}(\boldsymbol{\Omega}_I^F\boldsymbol{s}_I, \boldsymbol{\Omega}_D^F\boldsymbol{s}_D). \tag{6.10}$$

Therein, I employed a quadratic error term $f(\cdot) = \|\cdot\|_2^2$, i.e. the error is assumed to be additive and normally distributed. The sparsifying function g_{Co} is the same as the one used for learning the operators, see Equation (6.6). Consequently, it enforces the analyzed versions of both modalities to have a correlated co-support and hence couples the reconstruction results of the two signals. The measurement matrices $\boldsymbol{A}_I \in \mathbb{R}^{m_I \times N}$ and $\boldsymbol{A}_D \in \mathbb{R}^{m_D \times N}$ model the sampling process of each modality. The Lagrangian multiplier $\lambda \in \mathbb{R}^+$ balances the impact of the sparsity prior and the impact of the data fidelity terms. Therefore, it is used to control how closely the determined signal estimates approximate the corresponding measurements.

Depending on the measurement matrices, several inverse problems such as denoising, inpainting, or upsampling can be tackled via solving Problem (6.10). This can be accomplished either jointly for both signals, or by fixing one and optimizing over the other. Here, the focus lies on enhancing the quality of depth measurements $\boldsymbol{y}_D$, given a fixed high quality intensity signal, i.e. $\boldsymbol{y}_I = \boldsymbol{s}_I$. In this case, $\boldsymbol{A}_I$ is the identity operator $\boldsymbol{I}_N \in \mathbb{R}^{N \times N}$ and $\|\boldsymbol{A}_I\boldsymbol{s}_I - \boldsymbol{y}_I\|_2^2 = 0$. Furthermore, the analyzed intensity signal is constant during the optimization process, i.e. $\boldsymbol{\Omega}_I^F\boldsymbol{s}_I = \boldsymbol{c} =$ constant and consequently Problem (6.10) simplifies to

$$\boldsymbol{s}_D^\star \in \arg\min_{\boldsymbol{s}_D} \tfrac{1}{2}\|\boldsymbol{A}_D\boldsymbol{s}_D - \boldsymbol{y}_D\|_2^2 + \lambda g_{Co}(\boldsymbol{c}, \boldsymbol{\Omega}_D^F\boldsymbol{s}_D), \tag{6.11}$$

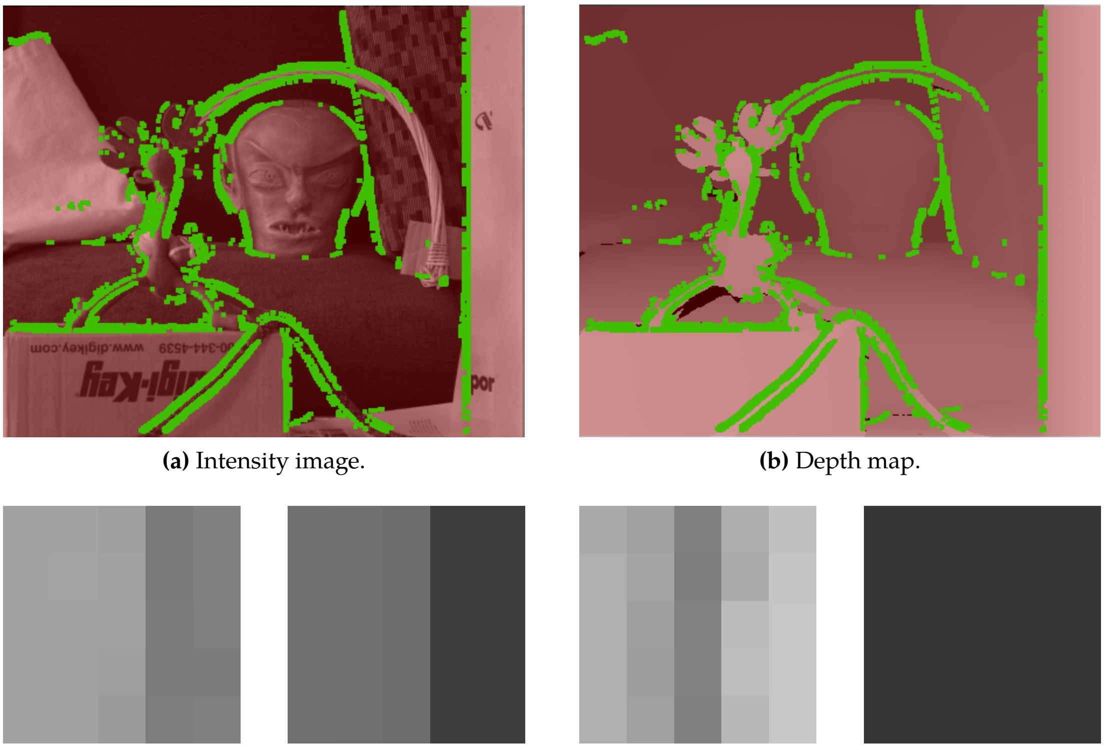
for the considered depth map enhancement task. Through this formulation, information about the structure of an observed scene that has been extracted from an intensity image and its co-support helps to determine the corresponding depth map with aligned co-support. The above optimization problem (6.11) is solved with a standard Euclidean conjugate gradient method using the Hestenes-Stiefel update formula and employing an Armijo step size selection.

6.3.2 Training Set Selection

Like all example based learning methods, the way the training samples are selected impacts the properties of the resulting operators and consequently affects their overall performance when applied in signal processing tasks. Due to the special characteristics of depth maps, selecting corresponding intensity and depth pairs at random positions, as for example done for GOAL, is a suboptimal procedure for two reasons. First, the majority of small depth patches are smooth homogeneous regions that do not carry important structural information and thus are useless for the training process, see Figure 6.4. Second, many depth patches simply show horizontal and vertical edges. Though the penalty term (6.8) used for learning the operators helps to prevent the result from overfitting to a specific subset of the training data such patches bias the learning process if no other structures are present in the training set. Therefore, one needs to ensure the training set to be sufficiently diverse in order to find a generally applicable multimodal analysis operator.

To overcome these two drawbacks, viable training pairs are selected according to two simple criteria. First, the gradient of the depth map is computed and all corresponding patch pairs where the energy of the depth gradient is low are discarded. In this way, useless smooth regions are removed. Second, to find a diverse training set the final patches are selected according to their dominant gradient orientation. For a single patch, its dominant orientation is found by composing a histogram of gradient orientations with b equally spaced bins and selecting the orientation that corresponds to the largest bin as the dominant orientation. In that way, one out of b possible orientations is assigned to each patch. Here, I only consider orientations in the range of $[0, \pi]$[1] and utilized $b = 18$. Next, the patches are grouped according to their dominant orientation, which in total results in $c \leq b$ groups, with c being the number of actually occurring orientations. Finally, from each of these c groups all, or maximally $\lceil \frac{M}{c} \rceil$ patches of highest depth gradient energy are added to the training set. An example of accepted and rejected sample patch pairs is illustrated in Figure 6.3.

[1] All orientations o in the range of $[-\pi, 0[$ are set to $o = o + \pi$.



(a) Intensity image.

(b) Depth map.

(c) Accepted patch pair (left intensity patch, right depth patch).

(d) Discarded patch pair (left intensity patch, right depth patch).

Figure 6.3: This figure exemplifies the training set selection process. A pair of corresponding intensity image and depth map are given in (a) and (b), respectively, where 2000 used patches are marked green and discarded regions are marked lite red. In (c) a close up of an accepted patch pair is given, and a close up of an unwanted and discarded pair is shown in (d).

6.3.3 Prior Art on Depth Map Superresolution

Increasing the resolution of depth maps obtained from range sensors has become an important research topic, and diverse approaches treating this problem have been proposed throughout the past years. Many of these methods originate from the closely related problem of intensity image superresolution. However, those methods mostly aim at producing pleasantly looking results, which is different from the goal of achieving geometrically sound scene depth maps. Straightforward upsamling methods like nearest-neighbor, bilinear, or bicubic interpolation produce undesirable staircasing or blurring artifacts, see Figure 6.1. Here, I shortly review more sophisticated methods for depth map SR that aim at reducing these artifacts.

In a first attempt, methods have been proposed that use smoothing priors from edge statistics [48] or local self-similarities [52]. These methods only require the LR depth map without a corresponding intensity image, but, either have difficulties in textured areas, or only work well for small upscaling factors.

A different approach, which also solely requires depth information, is based on fusing multiple LR depth maps of a scene obtained from slightly displaced range sensors into a single HR depth map. To that end, Schuon et al. [115] developed a global energy optimization framework employing data fidelity and geometry priors. This idea has been extended for better edge-preservation by Bhavsar et al. in [11].

A number of recently introduced methods aim at exploiting co-aligned discontinuities in corresponding intensity and depth images of the same scene. They fuse the HR and LR data utilizing Markov Random Fields (MRF). Depth map refinement based on MRFs has been first explored in [30], extended in [76] with a depth specific data term, and combined with depth from passive stereo in [145]. In order to better preserve local structures and to remove outliers, Park et al. [95] add a non-local means term to their MRF formulation. Aodha et al. [4] treat depth SR as a MRF labeling problem that matches LR depth patches to HR depth patches from a predefined database.

Inspired by successful stereo matching algorithms, Yang et al. [141] proposed an algorithm that iteratively employs a bilateral filter to improve depth map superresolution using an additional HR intensity image. Chan et al. [19] extended this approach by incorporating a noise model specific to depth data. Xiang et al. [133] included sub-pixel accuracy, and Dolson et al. [32] addressed temporal coherence across a depth data stream from LIDAR scanners by combining a bilateral filter with a Gaussian framework.

Finally, methods exist that exploit dependencies between sparse representations of intensity and depth signals over appropriate dictionaries. In [62], the complex wavelet transform is used as the dictionary and both the HR intensity image and the LR depth map are transformed into this domain. After that, the resulting detail and approximation coefficients are fused using a dual tree to obtain the final HR depth map. Instead of employing predefined bases or dictionaries, approaches that utilize learned dictionaries are known to lead to state-of-the-art performance in diverse classical image reconstruction tasks, cf. [44, 80]. Surprisingly, specific depth map enhancement techniques based on sparse representations learned from depth data are rare and this topic has only recently started to be explored. Mahmoudi et al. [77] proposed a method based on first learning a depth dictionary from noisy samples, then refining and denoising these samples, and finally learning a new dictionary from the denoised samples to inpaint, denoise, and super-resolve projected depth maps from 3D models. Closest to the presented approach are the recent efforts of [75] and

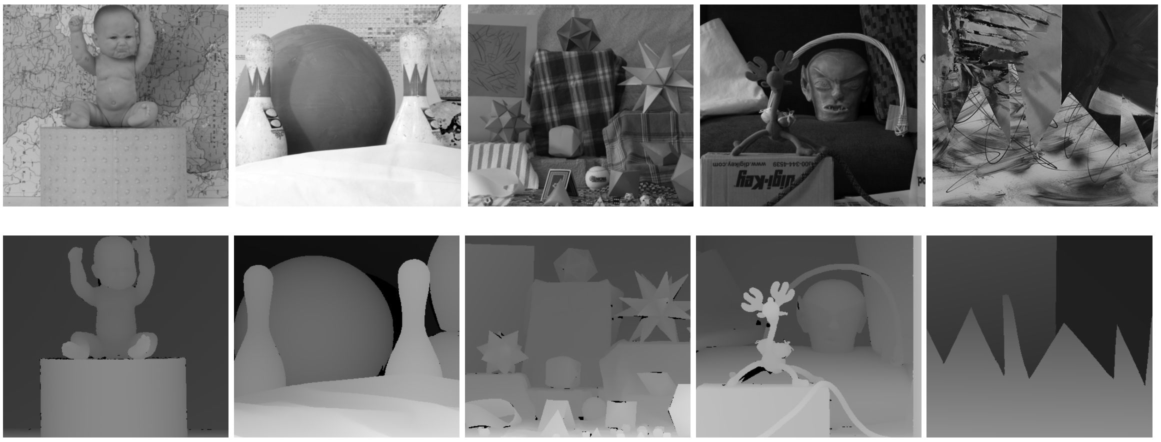


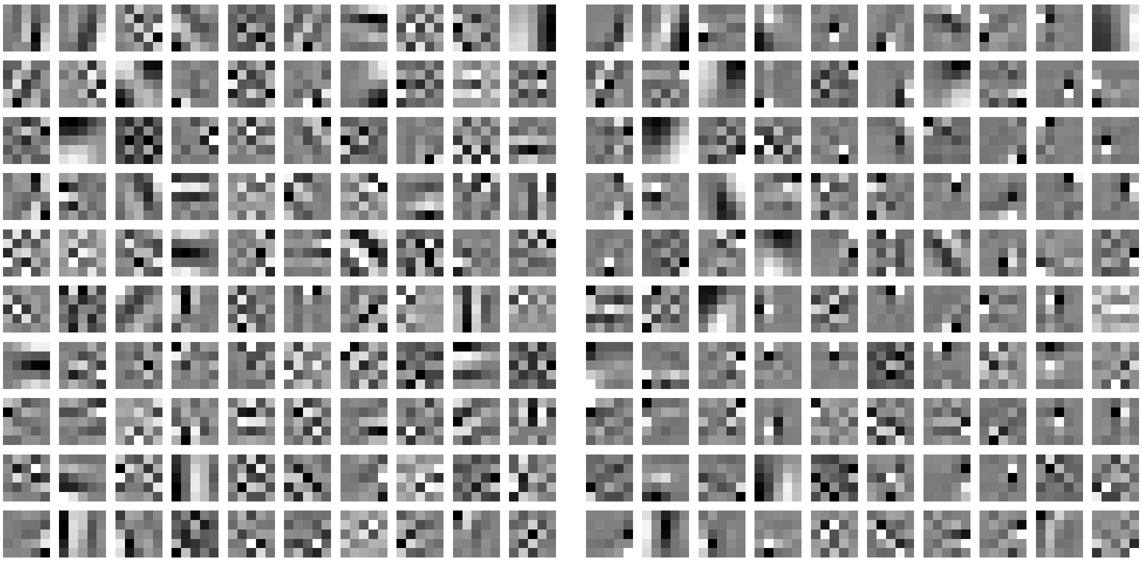
Figure 6.4: This figure shows the five pairs of aligned intensity images and depth maps from [113] used for learning the multimodal co-sparse analysis operator. Top row: intensity images, bottom row: registered depth maps.

[125]. In both methods, dictionaries are learned independently from depth and intensity samples, and the two modalities are coupled based on their sparse representations during the depth map reconstruction phase. In [75], three dictionaries are composed from LR depth, HR depth, and HR color samples to learn a respective mapping function based on edge features. In contrast to this, in [125] only two dictionaries one for intensity patches and one for depth patches are learned, and the similarity of the *support* of corresponding sparse representations with respect to the learned dictionaries is used to model the coupling.

6.4 Experimental Evaluation

This section presents a set of experiments to evaluate the performance of my multimodal analysis operator applied for the task of depth map enhancement. To that end, I use the well-known Middlebury stereo dataset [113], which provides aligned intensity images and depth maps for a number of different test scenes.

The five intensity and depth image pairs presented in Figure 6.4 have been chosen for learning the multimodal analysis operator. From these images, a total number of $M = 10000$ sample patches out of all possible patches of square size with $\sqrt{n} = 5$ have been gathered according to the criterion described in Section 6.3.2. As it is common in dictionary and conventional analysis operator learning methods, all training patches have been normalized to have zero-mean and unit ℓ_2-norm. To account for the zero-mean learning samples during the reconstruction process, the two learned analysis operators $\boldsymbol{\Omega}_I^\star$ and $\boldsymbol{\Omega}_D^\star$ were both mul-



(a) Intensity analysis atoms.

(b) Depth analysis atoms.

Figure 6.5: This figure presents the analysis atoms of a multimodal analysis operator learned from aligned intensity images and depth maps. The intensity analysis atoms are shown in (a) and the depth analysis atomes are given in (b). Black pixels correspond to the smallest negative values, gray pixels are zeros and white correspond to the largest positive values.

tiplied from the left by $\boldsymbol{N} = \left(\boldsymbol{I}_n - \frac{1}{n}\boldsymbol{J}\right)$, where $\boldsymbol{J} \in \mathbb{R}^{n \times n}$ denotes the identity matrix, i.e the final operators are given as $\boldsymbol{\Omega}_I = \boldsymbol{\Omega}_I^{\star}\boldsymbol{N}$ and $\boldsymbol{\Omega}_D = \boldsymbol{\Omega}_D^{\star}\boldsymbol{N}$. Regarding the number of analysis atoms a, I chose four times overcompleteness, i.e. $a = 4n$, which for the concrete dimensions used here results in two (100×25)-dimensional operators. The smoothing parameter in the sparsifying function (6.6) was set to $\nu = 10^3$ and the remaining parameters were set to $\kappa = 9 \cdot 10^4$ and $\mu = 10^2$. I determined these numbers empirically regarding the criteria explained in 5.6.2, i.e. the two learned analysis operators should have low condition number and moderate mutual coherence. Figure 6.5 shows the analysis atoms of the learned operators.

Using the learned operators, a depth map is reconstructed by solving Problem (6.11). Concerning the required Lagrangian multiplier λ, larger values lead to faster convergence of the optimization process but may large differences between the measurements and the reconstructed depth map. As explained below, the LR depth maps used in the experiments here are quasi noise-free, thus, λ could be set to a small value in order to maintain high data fidelity. However, small values for λ lead to a higher number of necessary iterations until the algorithm stops. To achieve descend results with few iterations, a common approach is to run the optimization process I times, with continuously reduced values of λ, cf. [63]. At each restart the algorithm is initialized with the reconstruction result determined at the previous iteration. Here, I repeated the optimization process ten times, i.e. $I = 10$, starting with $\lambda^{(1)} = 50$ and reducing it to a final value of $\lambda^{(I)} = 1$. Concretely, at the i-th iteration the Lagrangian multiplier for the next iteration $i + 1$ is computed by $\lambda^{(i+1)} = \lambda^{(i)}(\frac{\lambda^{(I)}}{\lambda^{(1)}})^{I-1}$.

To compare my method with the current state-of-the-art, I performed artificial tests on the four standard test images 'Tsukuba', 'Venus', 'Teddy', and 'Cones' that are part of the well-known stereo Middlebury dataset. From these depth maps, I generated synthetic LR depth maps by downscaling the ground truth depth maps by a factor of s in both vertical and horizontal dimension. For that purpose, the available HR depth maps were first blurred with a Gaussian kernel of size $((2s - 1) \times (2s - 1))$ and standard deviation $\sigma = \frac{s}{3}$ before downsampling. I used the resulting LR depth map and the corresponding HR intensity image as the input for the proposed depth map SR algorithm. These artificial tests permit to quantitatively compare the considered algorithms.

Following the methodology described in the work of comparable depth map SR approaches, I used the Middlebury stereo matching online evaluation tool[2] to assess the accuracy of the achieved results with respect to the ground truth data. The measure used by this tool is the percentage of badly recovered depth values, where a depth values is declared bad whenever the difference between the ground truth pixel and the recovered pixel is greater than one. Additionally, I measured the root-mean-squared error (RMSE) based on 8-bit result depth maps. To show the advantage of the multimodal co-sparse analysis model compared to its single modal counterpart, I performed all superresolution depth map experiments based on the standard analysis model formulation. To that end, I employed an operator learned by the algorithm GOAL from the depth samples of the training set used for learning the multimodal operator. All parameters were empirically found such that the resulting operator achieves the best possible results.

Table 6.1 and Table 6.2 present the numerical results regarding the two employed measures for upsampling the four test images by factors of $s = 2$, $s = 4$, and $s = 8$. The quantitative comparison with other depth map SR methods demonstrates the superior performance of the multimodal analysis operator across all test sets. It reaches near perfect results for small upscaling factors, and numerically the improvements over state-of-the-art methods is of particular significance for larger magnification factors.

Visually, as it can be seen from Figure 6.1, the suggested approach improves depth map SR considerably over simple interpolation approaches. Particularly in the most important

[2] http://vision.middlebury.edu/stereo/eval/

s	Method	Tsukuba	Venus	Teddy	Cones
2×	Nearest-Neighbor	1.24	0.37	4.97	2.51
	Yang et al. [141]	1.16	0.25	2.43	2.39
	GOAL	1.03	0.22	2.95	3.56
	Proposed method	**0.47**	**0.09**	**1.41**	**1.81**
4×	Nearest-Neighbor	3.53	0.81	6.71	5.44
	Yang et al.	2.56	0.42	5.95	**4.76**
	GOAL	2.95	0.65	4.80	6.54
	Proposed method	**1.73**	**0.25**	**3.54**	5.16
8×	Nearest-Neighbor	3.56	1.90	10.9	10.4
	Yang et al.	6.95	1.19	11.50	11.00
	Lu et al. [76]	5.09	1.00	9.87	11.30
	GOAL	5.59	1.24	11.40	12.30
	Proposed method	**3.53**	**0.33**	**6.49**	**9.22**

Table 6.1: This table gives a numerical comparison of my achieved experimental results to other depth map SR approaches for different upscaling factors s. The numbers represent the percentage of bad pixels with respect to all pixels of the ground truth data and an error threshold of $\delta = 1$.

s	Method	Tsukuba	Venus	Teddy	Cones
2×	Nearest-Neighbor	0.612	0.288	1.543	1.531
	Chan et al. [19]	n/a	0.216	1.023	1.353
	Aodha et al. [4]	0.601	0.296	0.977	1.227
	GOAL	0.278	0.105	0.996	0.939
	Proposed method	**0.255**	**0.075**	**0.702**	**0.680**
4×	Nearest-Neighbor	1.189	0.408	1.943	2.470
	Chan et al.	n/a	0.273	**1.125**	1.450
	Aodha et al.	0.833	0.395	1.184	1.779
	GOAL	0.450	0.179	1.389	1.398
	Proposed method	**0.346**	**0.129**	1.347	**1.383**
8×	Nearest-Neighbor	1.135	0.546	2.614	3.260
	Chan et al.	n/a	0.369	**1.410**	**1.635**
	GOAL	0.713	0.249	1.743	1.883
	Proposed method	**0.675**	**0.156**	1.662	1.871

Table 6.2: This table presents a numerical comparison of my achieved experimental results to other depth map SR approaches. The numbers represent the RMSE in comparison with the ground truth depth map.

areas that show discontinuities, neither staircasing nor substantial blurring artifacts occur. Even if SR is conducted using large upscaling factors, edges can be preserved with great detail due to the additional knowledge provided by the intensity image. For a further visual assessment, the 12 depth maps created by the proposed algorithm and that correspond to the results given in the tables are shown in Figure 6.7.

Considering the computational complexity of the algorithm, in my current unoptimized Matlab implementation, reconstructing a HR depth map with 500 CG-iterations takes up to three minutes on a standard desktop PC with a 3.2 GHz Intel i7 six core CPU and 16 Gb RAM. Since most of the processing time is dedicated to parallelizable filtering operations, I expect a considerably lower computation time with a better software implementation and processing on a GPU. Furthermore, the number of CG-iterations in the reconstruction may be reduced significantly. As shown in Figure 6.6, the last 400 iterations only reduce the relative RMSE by about 0.2% and descent recovery results are already achieved with only 50 CG steps.

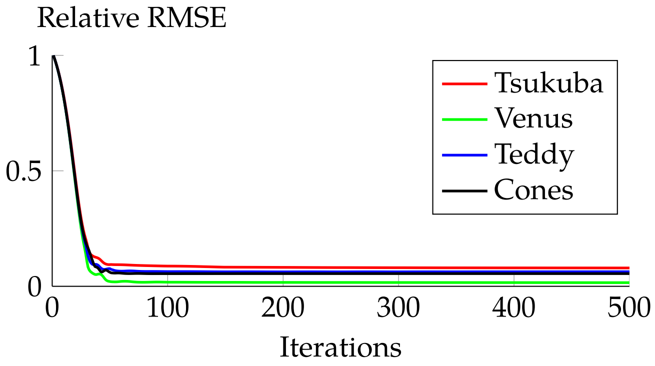
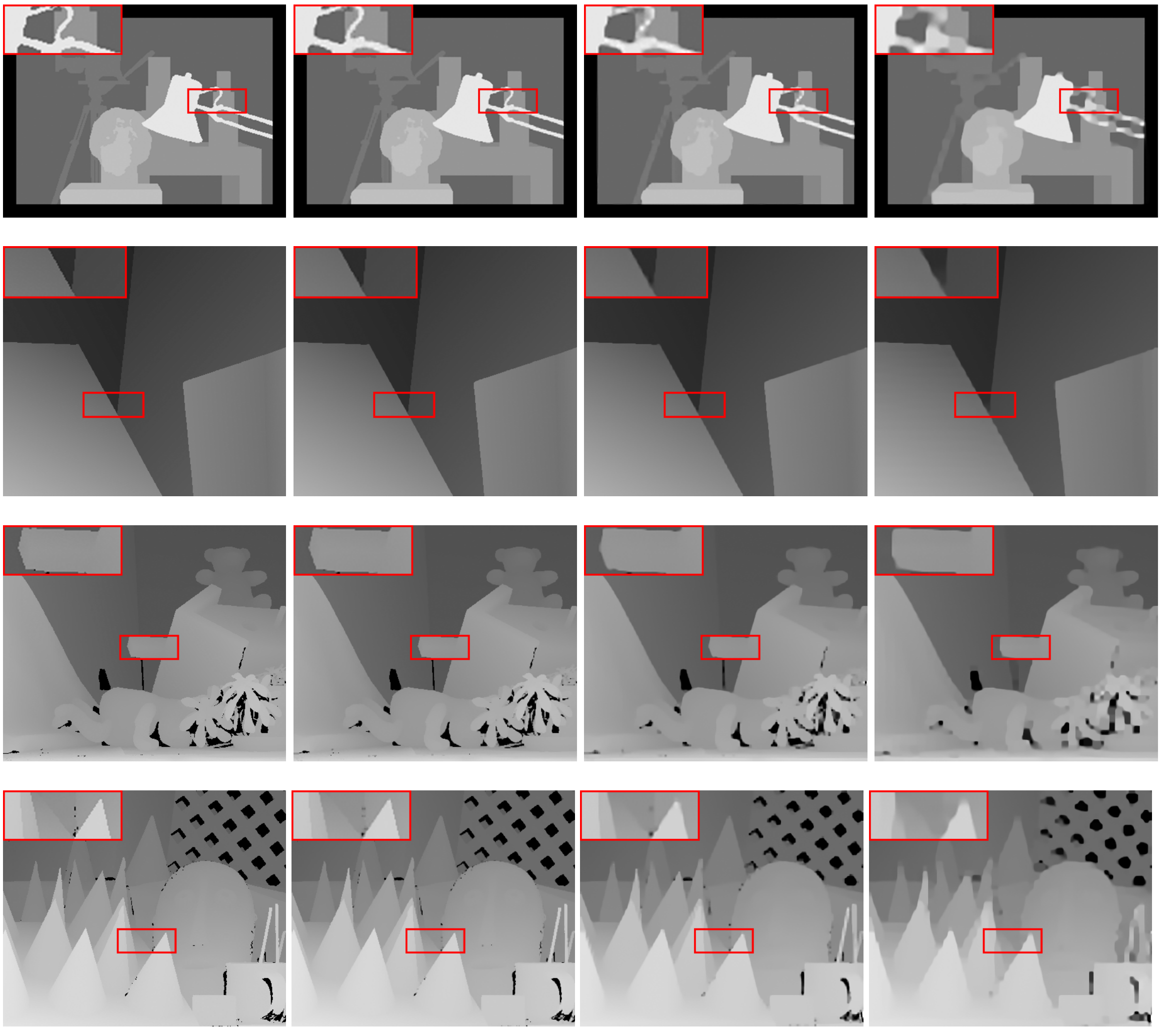


Figure 6.6: This figure shows the relative RMSE over the CG-iterations for upscaling the synthetic test images by a factor of $s = 8$.

6.5 Summary

In this chapter I proposed the new concept of multimodal co-sparse analysis modeling, and how this model can be exploited in signal reconstructing tasks. One necessity for this model is to have an appropriate set of multimodal analysis operators, which allow enforcing the proposed assumption of related signal modalities having a statistically dependent co-support. To that end, I introduced an efficient learning scheme based on conjugate gradient descent on the oblique manifold, that allows to jointly infer these operators from a



(a) Ground truth. **(b)** $s = 2$. **(c)** $s = 4$. **(d)** $s = 8$.

Figure 6.7: This figure presents the depth map superresolution results achieved with my proposed method. From top to bottom: Tsukuba, Venus, Teddy, and Cones. Columns (a) through (d) depict: ground truth (a), upscaling factor $s = 2$ (b), $s = 4$ (c), and $s = 8$ (d). For a better visual assessment, for each depth map a close up of an interesting region is provided. Notice how fine details and clear edges can be preserved even with large upscaling factors.

set of aligned training signals. As an example application, I showed how the model can be used to infer high-resolution depth maps from low-resolution depth samples given an additional high-resolution intensity image of the same scene. The presented numerical results show its excellent performance with improvements over the current state-of-the-art, and underpin the validity of the proposed model.

Other applications that can benefit from the proposed model can be found for example in the field of medical image processing. Therein, often various registered measurements are acquired from one patient in diverse modalities such as X-ray, magnetic resonance tomography, photon emission computed tomography, or ultrasound. Jointly extracting information out of all these modalities, or enhancing the respective signals is highly valuable to simplify and improve medical diagnostics. Furthermore, the model could be incorporated as a regularizer into variational methods for estimating depth maps or optical flow fields from stereo images. Currently, these methods are based on total-variation minimization which employs the finite difference operator, but, as also shown in this thesis, learned analysis operators outperform this analytic operator. Besides using a learned operator, additionally exploiting the dependencies between the two modalities as presented here is a promising path to follow.

Chapter 7

Conclusion

In this thesis, I investigated the problem of learning sparse data models from example data, regarding both the sparse synthesis and the co-sparse analysis point of view with emphasis on their applications to image processing tasks. In the first part, I introduced two new algorithms called Separable Dictionary Learning (SeDiL) and Geometric Analysis Operator Learning (GOAL) that are based on geometric conjugate gradient optimization on suitable matrix manifolds. Although the two approaches are designed to be signal independent in general, here, I mainly focused on applying the models to regularize classical inverse problems arising in the field of image processing. In the second part of this thesis, I presented the new multimodal co-sparse analysis model, which permits to model statistical dependencies of different modalities representing the same physical object.

Though a large number of dictionary learning approaches have been already introduced, the topic of finding structured dictionaries that enable fast implementations as well as enforcing specific internal properties on the dictionary during the learning process is still an open research issue. For those reasons, I introduced SeDiL, a novel learning approach that enforces a dictionary to have a separable matrix structure. This structure permits to learn dictionaries of high dimensions and reduces the computational complexity for both learning the dictionary and employing it in applications. Additionally, SeDiL is a new algorithm for learning unstructured conventional dictionaries, and enables controlling the mutual coherence of a learned dictionary. Note that this approach is straightforwardly extendable to data of more than two dimensions like volumetric medical images using the notion of multilinear algebra. In that realm, efficiently applicable dictionaries are even more relevant due to the exponentially growing dimensions of the signals and the associated dictionaries.

For the co-sparse analysis model, learning comprehensive signal representations from training data is a comparatively unexplored topic and only few methods are available. Here, I introduced the new analysis operator learning approach GOAL, which finds an operator by a non-convex optimization procedure with the feasible set of solutions restricted to the oblique manifold. With the imposed constraint set, the trivial solution is avoided and internal properties of the resulting analysis operator are controlled directly during the learning process. By a series of synthetic experiments, I showed that GOAL outperforms all existing analysis operator learning techniques in terms of computational complexity, ability to find a generating ground truth operator, and generality. Additionally, I employed an operator learned by GOAL for solving the classical inverse problems of image denoising, inpainting, and superresolution. The obtained results reveal the state-of-the-art performance of GOAL for these image processing tasks. As for dictionaries in the synthesis case, analysis operators that come along with an efficient implementation are important for a successful application in real world problems. To that end, adapting the separable dictionary learning technique such that a separable analysis operator can be learned is a valuable task for future work.

Based on the standard co-sparse analysis model, I introduced a novel concept called multimodal co-sparse analysis model that permits to model statistical dependencies of diverse measurements from the same object that have been acquired in different modalities. Because those measurements originate from the same object, the assumption that underlies the proposed model is that they share a common co-support with respect to a suitable set of multimodal analysis operators. For this set of operators, no analytic form exists and it must be learned from aligned and corresponding training signals. To that end, I proposed an extension of GOAL that uses a suitable sparsifying function to enforce the coupled co-support assumption. I evaluated the performance of the model for the task of depth map superresolution. Here, my method achieves state-of-the-art performance. Other applications that could benefit from the proposed model can be found for example in the field of medical image processing, where a patient's treatment is often based on information gained from several measurements acquired with diverse imaging technologies such as computer tomography, x-ray, or ultrasound. The computational bottleneck of processing such high dimensional data could be handled by combining the multimodal co-sparse analysis model with the separability constraint in future work.

Bibliography

1. P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

2. M. Aharon and M. Elad. Sparse and Redundant Modeling of Image Content Using an Image-Signature-Dictionary. In *SIAM Journal on Imaging Sciences*, 1(3), pp. 228--247, 2008.

3. M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. In *IEEE Transactions on Signal Processing*, 54(11), pp. 4311--4322, 2006.

4. O.M. Aodha, N.D.F. Campbell, A. Nair, and G. Brostow. Patch Based Synthesis for Single Depth Image Super-Resolution. In *European Conference on Computer Vision*, pp. 71--84. 2012.

5. F. Bach, R. Jenatton, J. Mairal, and J. Obozinski. Convex Optimization with Sparsity-Inducing Norms. In *Optimization for Machine Learning*. MIT Press, 2011.

6. D. Barchiesi and M.D. Plumbley. Learning Incoherent Dictionaries for Sparse Approximation Using Iterative Projections and Rotations. In *IEEE Transactions on Signal Processing*, 61(8), pp. 2055--2065, 2013.

7. A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. In *SIAM Journal on Imaging Sciences*, 2(1), pp. 183--202, 2009.

8. S. Becker, J. Bobin, and E.J. Candès. NESTA: A Fast and Accurate First-Order Method for Sparse Recovery. In *SIAM Journal on Imaging Sciences*, 4(1), pp. 1--39, 2009.

9. L. Benoit, J. Mairal, F. Bach, and J. Ponce. Sparse Image Representation with Epitomes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2913--2920. 2011.

10. M. Bertalmìo, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *ACM SIGGRAPH*, pp. 417--424. 2000.

11. A.V. Bhavsar and A.N. Rajagopalan. Range Map Superresolution-Inpainting, and Reconstruction from Sparse Data. In *Computer Vision and Image Understanding*, 116(4), pp. 572--591, 2012.

12. J.M. Bioucas-Dias and M.A.T. Figueiredo. A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration. In *IEEE Transactions on Image Processing*, 16(12), pp. 2992--3004, 2007.

13. T. Blumensath and M.E. Davies. Gradient Pursuits. In *IEEE Transactions on Signal Processing*, 56(6), pp. 2370--2382, 2008.

14. T. Blumensath and M.E. Davies. Iterative Thresholding for Sparse Approximations. In *The Journal of Fourier Analysis and Applications*, 14(5), pp. 629--654, 2008.

15. A.M. Bruckstein, D.L. Donoho, and M. Elad. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. In *SIAM Review*, 51(1), pp. 34--81, 2009.

16. E.J. Candès, Y.C. Eldar, D. Needell, and P. Randall. Compressed Sensing with Coherent and Redundant Dictionaries. In *Applied and Computational Harmonic Analysis*, 31(1), pp. 59--73, 2011.

17. E.J. Candès, J. Romberg, and T. Tao. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. In *IEEE Transactions on Information Theory*, 52(2), pp. 489--509, 2006.

18. E.J. Candès, M.B. Wakin, and S. Boyd. Enhancing Sparsity by Reweighted l1 Minimization. In *Journal of Fourier Analysis and Applications*, 14(5), pp. 877--905, 2008.

19. D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A Noise-Aware Filter for Real-Time Depth Upsampling. In *ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, pp. 1--12. 2008.

20. S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic Decomposition by Basis Pursuit. In *SIAM Journal on Scientific Computing*, 20(1), pp. 33--61, 1999.

21. R.R. Coifman and M.V. Wickerhauser. Entropy-Based Algorithms for Best Basis Selection. In *IEEE Transactions on Information Theory*, 38(2), pp. 713--718, 1992.

22. S.F. Cotter, R. Adler, R.D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. In *IEE Proceedings - Vision, Image and Signal Processing*, 146(5), pp. 235--244, 1999.

23. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. In *IEEE Transactions on Image Processing*, 16(8), pp. 2080--2095, 2007.

24. J. Dahl, P.C. Hansen, S. Jensen, and T.L. Jensen. Algorithms and Software for Total Variation Image Reconstruction via First-Order Methods. In *Numerical Algorithms*, 53(1), pp. 67--92, 2010.

25. W. Dai and O. Milenkovic. Subspace Pursuit for Compressive Sensing Signal Reconstruction. In *IEEE Transactions on Information Theory*, 55(5), pp. 2230--2249, 2009.

26. Y.H. Dai. On the Nonmonotone Line Search. In *Journal of Optimization Theory and Applications*, 112(2), pp. 315--330, 2002.

27. Y.H. Dai and Y. Yuan. An Efficient Hybrid Conjugate Gradient Method for Unconstrained Optimization. In *Annals of Operations Research*, 103(1), pp. 33--47, 2001.

28. I. Daubechies, M. Defrise, and C. De Mol. An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint. In *Communications on Pure and Applied Mathematics*, 57(11), pp. 1413--1457, 2004.

29. G. Davis, S. Mallat, and Z. Zhang. Adaptive Time-Frequency Decompositions with Matching Pursuits. In *Optical Engineering*, 33(7), pp. 2183--2191, 1994.

30. J. Diebel and S. Thrun. An Application of Markov Random Fields to Range Sensing. In *Advances in Neural Information Processing Systems*, pp. 291--298. 2005.

31. M.N. Do and M. Vetterli. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. In *IEEE Transactions on Image Processing*, 14(12), pp. 2091--2106, 2005.

32. J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling Range Data in Dynamic Environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1141--1148. 2010.

33. D.L. Donoho. Wedgelets: Nearly-Minimax Estimation of Edges. In *Annals of Statistics*, 27(3), pp. 859--897, 1999.

34. D.L. Donoho and M. Elad. Optimally Sparse Representation in General (Nonorthogonal) Dictionaries via l1-Minimization. In *Proceedings of the National Academy of Sciences of the United States of America*, 100(5), pp. 2197--2202, 2003.

35. D.L. Donoho and X. Huo. Uncertainty Principles and Ideal Atomic Decomposition. In *IEEE Transactions on Information Theory*, 47(7), pp. 2845 --2862, 2001.

36. D.L. Donoho and Y. Tsaig. Fast Solution of l1-Norm Minimization Problems When the Solution May Be Sparse. In *IEEE Transactions on Information Theory*, 54(11), pp. 4789--4812, 2008.

37. D.L. Donoho, Y. Tsaig, I. Drori, and J.L. Starck. Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. In *IEEE Transactions on Information Theory*, 58(2), pp. 1094--1121, 2012.

38. J.M. Duarte-Carvajalino and G. Sapiro. Learning to Sense Sparse Signals: Simultaneous Sensing Matrix and Sparsifying Dictionary Optimization. In *IEEE Transactions on Image Processing*, 18(7), pp. 1395--1408, 2009.

39. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. In *Annals of Statistics*, 32(4), pp. 407--499, 2004.

40. M. Elad. Why Simple Shrinkage Is Still Relevant for Redundant Representations? In *IEEE Transactions on Information Theory*, 52(12), pp. 5559--5569, 2006.

41. M. Elad. Optimized Projections for Compressed Sensing. In *IEEE Transactions on Signal Processing*, 55(12), pp. 5695--5702, 2007.

42. M. Elad. Sparse and Redundant Representation Modeling ? What Next? In *IEEE Signal Processing Letters*, 19(12), pp. 922--928, 2012.

43. M. Elad and M. Aharon. Image Denoising via Sparse and Redundant Representations Over Learned Dictionaries. In *IEEE Transactions on Image Processing*, 15(12), pp. 3736--3745, 2006.

44. M. Elad, M.A.T. Figueiredo, and Y. M. On the Role of Sparse and Redundant Representations in Image Processing. In *Proceedings of the IEEE*, 98(6), pp. 972--982, 2010.

45. M. Elad, P. Milanfar, and R. Rubinstein. Analysis Versus Synthesis in Signal Priors. In *Inverse Problems*, 3(3), pp. 947--968, 2007.

46. K. Engan, S. Aase, and J. Hakon Husoy. Method of Optimal Directions for Frame Design. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2443--2446. 1999.

47. K. Engan, K. Skretting, and J.H. Husoy. Family of Iterative LS-Based Dictionary Learning Algorithms, ILS-DLA, for Sparse Signal Representation. In *Digital Signal Processing*, 17(1), pp. 32--49, 2007.

48. R. Fattal. Image Upsampling via Imposed Edge Statistics. In *ACM Transactions on Graphics*, 26(3), pp. 95:1--95:8, 2007.

49. M.A.T. Figueiredo and R.D. Nowak. An EM Algorithm for Wavelet-Based Image Restoration. In *IEEE Transactions on Image Processing*, 12(8), pp. 906--916, 2003.

50. M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. In *IEEE Journal of Selected Topics in Signal Processing*, 1(4), pp. 586--597, 2007.

51. S. Foucart. Hard Thresholding Pursuit: An Algorithm for Compressive Sensing. In *SIAM Journal on Numerical Analysis*, 49(6), pp. 2543--2563, 2011.

52. G. Freedman and R. Fattal. Image and Video Upscaling from Local Self-Examples. In *ACM Transactions on Graphics*, 30(2), pp. 1--11, 2011.

53. W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-Based Super-Resolution. In *IEEE Computer Graphics and Applications*, 22(2), pp. 56--65, 2002.

54. W.J. Fu. Penalized Regressions: The Bridge Versus the Lasso. In *Journal of Computational and Graphical Statistics*, 7(3), pp. 397--416, 1998.

55. Q. Gao and S. Roth. How Well Do Filter-Based MRFs Model Natural Images? In *DAGM/OAGM Symposium*, pp. 62--72. 2012.

56. J.C. Gilbert and J. Nocedal. Global Convergence Properties of Conjugate Gradient Methods for Optimization. In *SIAM Journal on Optimization*, 2(1), pp. 21--42, 1992.

57. R. Giryes and M. Elad. CoSaMP and SP for the Cosparse Analysis Model. In *European Signal Processing Conference*, pp. 964--968. 2012.

58. R. Giryes, S. Nam, M. Elad, R. Gribonval, and M. Davies. Greedy-Like Algorithms for the Cosparse Analysis Model. In *Special Issue on Sparse Approximate Solution of Linear Systems in Linear Algebra and its Applications*, 2013.

59. R. Giryes, S. Nam, R. Gribonval, and M. Davies. Iterative Cosparse Projection Algorithms for the Recovery of Cosparse Vectors. In *European Signal Processing Conference*, pp. 1460--1464. 2011.

60. T. Goldstein and S. Osher. The Split Bregman Method for l1-Regularized Problems. In *SIAM Journal on Imgaging Science*, 2(2), pp. 323--343, 2009.

61. I.F. Gorodnitsky and B.D. Rao. Sparse Signal Reconstruction from Limited Data Using FOCUSS: A Re-Weighted Minimum Norm Algorithm. In *IEEE Transactions on Signal Processing*, 45(3), pp. 600--616, 1997.

62. S.A. Gudmundsson and J.R. Sveinsson. ToF-CCD Image Fusion using Complex Wavelets. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1557--1560. 2011.

63. S. Hawe, M. Kleinsteuber, and K. Diepold. Cartoon-Like Image Reconstruction via Constrained lp-Minimization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 717--720. 2012.

64. G.E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. In *Neural Computation*, 14(8), pp. 1771--1800, 2002.

65. G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

66. Z. J., L. Zhe, and L.S. Davis. Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1697--1704. 2011.

67. R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal Methods for Sparse Hierarchical Dictionary Learning. In *International Conference on Machine Learning*, pp. 487--494. 2010.

68. S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An Interior-Point Method for Large-Scale l1-Regularized Least Squares. In *IEEE Journal of Selected Topics in Signal Processing*, 1(4), pp. 606--617, 2007.

69. A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer, New York, NY, USA, 1996.

70. M. Kleinsteuber and H. Shen. Blind Source Separation with Compressively Sensed Linear Mixtures. In *IEEE Signal Processing Letters*, 19(2), pp. 107--110, 2012.

71. K. Kreutz-Delgado, J.F. Murray, B. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski. Dictionary Learning Algorithms for Sparse Representation. In *Neural Computation*, 15(2), pp. 349--396, 2003.

72. E. Le Pennec and S. Mallat. Sparse Geometric Image Representations with Bandelets. In *IEEE Transactions on Image Processing*, 14(4), pp. 423--438, 2005.

73. M.S. Lewicki and B.A. Olshausen. A Probabilistic Framework for the Adaptation and Comparison of Image Codes. In *Journal of the Optical Society of America*, 7(16), pp. 1587--1601, 1999.

74. M.S. Lewicki and T.J. Sejnowski. Learning Overcomplete Representations. In *Neural Computation*, 12(2), pp. 337--365, 2000.

75. Y. Li, T. Xue, L. Sun, and J. Liu. Joint Example-Based Depth Map Super-Resolution. In *IEEE International Conference on Multimedia and Expo*, pp. 152--157. 2012.

76. J. Lu, D. Min, R.S. Pahwa, and M.N. Do. A Revisit to MRF-Based Depth Map Super-Resolution and Enhancement. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 985--988. 2011.

77. M. Mahmoudi and G. Sapiro. Sparse Representations for Range Data Restoration. In *IEEE Transactions on Image Processing*, 21(5), pp. 2909--2915, 2012.

78. J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. In *Journal of Machine Learning Research*, 11(1), pp. 19--60, 2010.

79. J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative Learned Dictionaries for Local Image Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1--8. 2008.

80. J. Mairal, M. Elad, and G. Sapiro. Sparse Representation for Color Image Restoration. In *IEEE Transactions on Image Processing*, 17(1), pp. 53--69, 2008.

81. S.G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), pp. 674--693, 1989.

82. S.G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, USA, 1999.

83. S.G. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. In *IEEE Transactions on Signal Processing*, 41(12), pp. 3397--3415, 1993.

84. J.J. Mcauley, T.S. Caetano, A.J. Smola, and M.O. Franz. Learning High-Order MRF Priors of Color Images. In *International Conference on Machine Learning*, pp. 617--624. 2006.

85. S. Nam, M.E. Davies, M. Elad, and R. Gribonval. Cosparse Analysis Modeling - Uniqueness and Algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5804--5807. 2011.

86. S. Nam, M.E. Davies, M. Elad, and R. Gribonval. The Cosparse Analysis Model and Algorithms. In *Applied and Computational Harmonic Analysis*, 34(1), pp. 30--56, 2013.

87. B.K. Natarajan. Sparse Approximate Solutions to Linear Systems. In *SIAM Journal on Computing*, 24(2), pp. 227--234, 1995.

88. D. Needell and J.A. Tropp. CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples. In *Applied and Computational Harmonic Analysis*, 26(3), pp. 301--321, 2009.

89. D. Needell and R. Vershynin. Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit. In *Foundations of Computational Mathematics*, 9(3), pp. 317--334, 2009.

90. B.A. Olshausen and D.J. Field. Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. In *Nature*, 381(6583), pp. 607--609, 1996.

91. B.A. Olshausen and D.J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1. In *Vision Research*, 37(23), pp. 3311--3325, 1997.

92. B. Ophir, M. Elad, N. Bertin, and M.D. Plumbley. Sequential Minimal Eigenvalues - An Approach to Analysis Dictionary Learning. In *European Signal Processing Conference*, pp. 1465--1469. 2011.

93. B. Ophir, M. Lustig, and M. Elad. Multi-Scale Dictionary Learning Using Wavelets. In *IEEE Journal of Selected Topics in Signal Processing*, 5(5), pp. 1014--1024, 2011.

94. M.R. Osborne, B. Presnell, and B.A. Turlach. A New Approach to Variable Selection in Least Squares Problems. In *IMA Journal of Numerical Analysis*, 20(3), pp. 389--403, 2000.

95. J. Park, H. Kim, M.S. Brown, and I. Kweon. High Quality Depth Map Upsampling for 3D-TOF Cameras. In *IEEE International Conference on Computer Vision*, pp. 1623--1630. 2011.

96. Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *Conference on Signals, Systems and Computers*, pp. 40--44. 1993.

97. J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain. In *IEEE Transactions on Image Processing*, 12(11), pp. 1338--1351, 2003.

98. M.J.D. Powell. Restart Procedures for the Conjugate Gradient Method. In *Mathematical Programming*, 12(1), pp. 241--254, 1977.

99. I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3501--3508. 2010.

100. B.D. Rao and K. Kreutz-Delgado. An Affine Scaling Methodology for Best Basis Selection. In *IEEE Transactions on Signal Processing*, 47(1), pp. 187--200, 1999.

101. S. Ravishankar and Y. Bresler. Learning Sparsifying Transforms. In *IEEE Transactions on Signal Processing*, 61(5), pp. 1072--1086, 2013.

102. L. Rebollo-Neira and D. Lowe. Optimized Orthogonal Matching Pursuit Approach. In *IEEE Signal Processing Letters*, 9(4), pp. 137--140, 2002.

103. R. Rigamonti, V. Lepetit, and P. Fua. Learning Separable Filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1--8. 2013.

104. W. Ring and B. Wirth. Optimization Methods on Riemannian Manifolds and Their Application to Shape Space. In *SIAM Journal on Optimization*, 22(2), pp. 596--627, 2012.

105. A. Ron and Z. Shen. Affine Systems in L2(Rd): The Analysis of the Analysis Operator. In *Journal of Functional Analysis*, 148(2), pp. 408--447, 1996.

106. S. Roth and M.J. Black. Fields of Experts: A Framework for Learning Image Priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 860--867. 2005.

107. S. Roth and M.J. Black. Fields of Experts. In *International Journal of Computer Vision*, 82(2), pp. 205--229, 2009.

108. R. Rubinstein, T. Faktor, and M. Elad. K-SVD Dictionary-Learning for the Analysis Sparse Model. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5405--5408. 2012.

109. R. Rubinstein, T. Peleg, and M. Elad. Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model. In *IEEE Transactions on Signal Processing*, 61(3), pp. 661--677, 2012.

110. R. Rubinstein, M. Zibulevsky, and M. Elad. Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation. In *IEEE Transactions on Signal Processing*, 58(3), pp. 1553--1564, 2010.

111. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. In *Physica D*, 60(1-4), pp. 259--268, 1992.

112. C. Sanderson and B.C. Lovell. Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference. In *International Conference on Advances in Biometrics*, pp. 199--208. 2009.

113. D. Scharstein and R. Szeliski. High-Accuracy Stereo Depth Maps using Structured Light. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 195--202. 2003.

114. U. Schmidt, Q. Gao, and S. Roth. A Generative Perspective on MRFs in Low-Level Vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1751--1758. 2010.

115. S. Schuon, C. Theobalt, J. Davis, and S. Thrun. LidarBoost: Depth Superresolution for ToF 3D Shape Scanning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 343--350. 2009.

116. S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 519--528. 2006.

117. I.W. Selesnick and M.A.T. Figueiredo. Signal Restoration with Overcomplete Wavelet Transforms: Comparison of Analysis and Synthesis Priors. In *In Proceedings of SPIE Wavelets XIII*. 2009.

118. C.D. Sigg, T. Dikk, and J.M. Buhmann. Learning Dictionaries with Bounded Self-Coherence. In *IEEE Signal Processing Letters*, 19(12), pp. 861--864, 2012.

119. E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable Multiscale Transforms. In *IEEE Transactions on Information Theory*, 38(2), pp. 587--607, 1992.

120. K. Skretting and K. Engan. Recursive Least Squares Dictionary Learning Algorithm. In *IEEE Transactions on Signal Processing*, 58(4), pp. 2121--2130, 2010.

121. J.L. Starck, E.J. Candès, and D.L. Donoho. The Curvelet Transform for Image Denoising. In *IEEE Transactions on Image Processing*, 11(6), pp. 670--684, 2002.

122. D.S. Taubman and M.W. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Kluwer Academic Publishers, SECS 642 Boston, 2002.

123. R. Tibshirani. Regression Shrinkage and Selection via the Lasso. In *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1), pp. 267--288, 1996.

124. R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and Smoothness via the Fused Lasso. In *Journal of the Royal Statistical Society Series B*, pp. 91--108, 2005.

125. I. Tošić and S. Drewes. Learning Joint Intensity-Depth Sparse Representations. In *arXiv preprint*, 2012. ArXiv:1201.0566v1.

126. I. Tošić and P. Frossard. Dictionary Learning. In *IEEE Signal Processing Magazine*, 28(2), pp. 27--38, 2011.

127. N.T. Trendafilov. A Continuous-Time Approach to the Oblique Procrustes Problem. In *Behaviormetrika*, 26(2), pp. 167--181, 1999.

128. J.A. Tropp. Greed is Good: Algorithmic Results for Sparse Approximation. In *IEEE Transactions on Information Theory*, 50(10), pp. 2231--2242, 2004.

129. S. Vaiter, G. Peyre, C. Dossal, and J. Fadili. Robust Sparse Analysis Regularization. In *IEEE Transactions on Information Theory*, 59(4), pp. 2001--2016, 2013.

130. Z. W., A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. In *IEEE Transactions on Image Processing*, 13(4), pp. 600--612, 2004.

131. J. Wörmann, S. Hawe, and M. Kleinsteuber. Analysis Based Blind Compressive Sensing. In *IEEE Signal Processing Letters*, 20(5), pp. 491--494, 2013.

132. S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo. Sparse Reconstruction by Separable Approximation. In *IEEE Transactions on Signal Processing*, 57(7), pp. 2479--2493, 2009.

133. X. Xiang, G. Li, J. Tong, and Z. Pan. Fast and Simple Super Resolution for Range Data. In *International Conference on Cyberworlds*, pp. 319--324. 2010.

134. Z.J. Xiang, H. Xu, and P.J. Ramadge. Learning Sparse Representations of High Dimensional Data on Large Scale Dictionaries. In *Advances in Neural Information Processing Systems*, pp. 900--908. 2011.

135. M. Yaghoobi, T. Blumensath, and M.E. Davies. Dictionary Learning for Sparse Approximations with the Majorization Method. In *IEEE Transactions on Signal Processing*, 57(6), pp. 2178--2191, 2009.

136. M. Yaghoobi and M.E. Davies. Compressible Dictionary Learning for Fast Sparse Approximations. In *IEEE Workshop on Statistical Signal Processing*, pp. 662--665. 2009.

137. M. Yaghoobi, S. Nam, R. Gribonval, and M.E. Davies. Analysis Operator Learning for Overcomplete Cosparse Representations. In *European Signal Processing Conference*, pp. 1470--1474. 2011.

138. M. Yaghoobi, S. Nam, R. Gribonval, and M.E. Davies. Noise Aware Analysis Operator Learning for Approximately Cosparse Signals. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5409--5412. 2012.

139. M. Yaghoobi, S. Nam, R. Gribonval, and M. Davies. Constrained Overcomplete Analysis Operator Learning for Cosparse Signal Modelling. In *IEEE Transactions on Signal Processing*, 61(9), pp. 2341--2355, 2013.

140. J. Yang, J. Wright, T.S. Huang, and Y. Ma. Image Super-Resolution via Sparse Representation. In *IEEE Transactions on Image Processing*, 19(11), pp. 2861--2873, 2010.

141. Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-Depth Super Resolution for Range Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1--8. 2007.

142. L. Ying, L. Demanet, and E. Candès. Fast Discrete Curvelet Transforms. In *SIAM Multiscale Modeling and Simulation*, 5(3), pp. 861--899, 2006.

143. H. Zhang and W.W. Hager. A Nonmonotone Line Search Technique and Its Application to Unconstrained Optimization. In *SIAM Journal on Optimization*, 14(4), pp. 1043--1056, 2004.

144. M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin. Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images. In *IEEE Transactions on Image Processing*, 21(1), pp. 130--144, 2012.

145. J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1--8. 2008.

146. M. Zibulevsky and M. Elad. L1-L2 Optimization in Signal and Image Processing. In *IEEE Signal Processing Magazine*, 27(3), pp. 76--88, 2010.

147. H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. In *Journal of the Royal Statistical Society, Series B*, 67(2), pp. 301--320, 2005.