Analysis Operator Learning and Its Application to Image Reconstruction

Simon Hawe, Martin Kleinsteuber, and Klaus Diepold

Abstract

http:// 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 the design of 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 an analysis model severely depends on the right choice of the suitable operator.

In this work, we present an algorithm for learning an analysis operator from training images. Our method is based on ℓ_p-norm minimization on the set of full rank matrices with normalized columns. We carefully introduce the employed conjugate gradient method on manifolds, and explain the underlying geometry of the constraints. Moreover, we compare our approach to state-of-the-art methods for image denoising, inpainting, and single image super-resolution. Our numerical results show competitive performance of our general approach in all presented applications compared to the specialized state-of-the-art techniques.

Index Terms

Analysis Operator Learning, Inverse Problems, Image Reconstruction, Geometric Conjugate Gradient, Oblique Manifold

I. Introduction

A. Problem Description

LINEAR inverse problems are ubiquitous in the field of image processing. Prominent examples are image denoising [1], inpainting [2], super-resolution [3], or image reconstruction from few indirect measurements as in Compressive Sensing [4]. Basically, in all these problems the goal is to reconstruct an unknown image $\mathbf{s} \in \mathbb{R}^n$ as accurately as possible from a set of indirect and maybe corrupted measurements $\mathbf{y} \in \mathbb{R}^m$ with $n \geq m$, see [5] for a detailed introduction to inverse problems. Formally, this measurement process can be written as

$$\mathbf{y} = \mathcal{A}\mathbf{s} + \mathbf{e}, \tag{1}$$

where the vector $\mathbf{e} \in \mathbb{R}^m$ models sampling errors and noise, and $\mathcal{A} \in \mathbb{R}^{m \times n}$ is the measurement matrix modeling the sampling process. In many cases, reconstructing $\mathbf{s}$ by simply inverting Equation (1) is ill-posed because either the exact measurement process and hence $\mathcal{A}$ is unknown as 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 or image inpainting. To overcome the ill-posedness and to stabilize the solution, prior knowledge or assumptions about the general statistics of images can be exploited.

 The paper has been published in IEEE Transaction on Image Processing 2013.

IEEE Xplore: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6459595&contentType=Early+Access+Articles&queryText%3DSImon+Hav

DOI: 10.1109/TIP.2013.2246175

The authors are with the Department of Electrical Engineering, Technische Universität München, Arcisstraße 21, Munich 80290, Germany (e-mail: {simon.hawe,kleinsteuber,kldi}@tum.de, web: www.gol.ei.tum.de)

This work has been supported by the Cluster of Excellence *CoTeSys* - Cognition for Technical Systems, funded by the German Research Foundation (DFG).

B. Synthesis Model and Dictionary Learning

One assumption that has proven to be successful in image reconstruction, cf. [6], is that natural images admit a sparse representation $\mathbf{x} \in \mathbb{R}^d$ over some dictionary $\mathcal{D} \in \mathbb{R}^{n \times d}$ with $d \geq n$. A vector $\mathbf{x}$ is called sparse when most of its coefficients are equal to zero or small in magnitude. When $\mathbf{s}$ admits a sparse representation over $\mathcal{D}$, it can be expressed as a linear combination of only very few columns of the dictionary $\{\mathbf{d}_i\}_{i=1}^d$, called *atoms*, which reads as

$$\mathbf{s} = \mathcal{D}\mathbf{x}. \tag{2}$$

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

Now, using the knowledge that (2) allows a sparse solution, an estimation of the original signal in (1) can be obtained from the measurements $\mathbf{y}$ by first solving

$$\mathbf{x}^\star = \underset{\mathbf{x} \in \mathbb{R}^d}{\arg\min}\ g(\mathbf{x})\ \text{ subject to }\ \|\mathcal{A}\mathcal{D}\mathbf{x} - \mathbf{y}\|_2^2 \leq \epsilon, \tag{3}$$

and afterwards synthesizing the signal from the computed sparse coefficients via $\mathbf{s}^\star = \mathcal{D}\mathbf{x}^\star$. Therein, $g : \mathbb{R}^d \to \mathbb{R}$ is a function that promotes or measures sparsity, and $\epsilon \in \mathbb{R}^+$ is an estimated upper bound on the noise power $\|\mathbf{e}\|_2^2$. Common choices for g include the ℓ_p-norm

$$\|\mathbf{v}\|_p^p := \sum_i |v_i|^p, \tag{4}$$

with $0 < p \leq 1$ and differentiable approximations of (4). As the signal is synthesized from the sparse coefficients, the reconstruction model (3) is called the *synthesis* reconstruction model [7].

To find the minimizer of Problem (3), various algorithms based on convex or non-convex optimization, greedy pursuit methods, or Bayesian frameworks exist that may employ different choices of g. For a broad overview of such algorithms, we refer the interested reader to [8]. What all these algorithms have in common, is that their performance regarding the reconstruction quality severely depends on an appropriately chosen dictionary $\mathcal{D}$. Ideally, one is seeking for a dictionary where $\mathbf{s}$ can be represented most accurately with a coefficient vector $\mathbf{x}$ that is as sparse as possible. Basically, dictionaries can be assigned to two major classes: *analytic dictionaries* and *learned dictionaries*.

Analytic dictionaries are built on mathematical models of a general type of signal, e.g. natural images, they should represent. Popular examples include Wavelets [9], Bandlets [10], and Curvlets [11] among several others, or a concatenation of various such bases/dictionaries. They offer the advantages of low computational complexity and of being universally applicable to a wide set of signals. However, this universality comes at the cost of not giving the optimally sparse representation for more specific classes of signals, e.g. face images.

It is now well known that signals belonging to a specific class can be represented with fewer coefficients over a dictionary that has been learned using a representative training set, than over analytic dictionaries. This is desirable for various image reconstruction applications as it readily improves their performance and accuracy [12]–[14]. Basically, the goal is to find a dictionary over which a training set admits a maximally sparse representation. In contrast to analytic dictionaries, which can be applied globally to an entire image, learned dictionaries are small dense matrices that have to be applied locally to small image patches. Hence, the training set consists of small patches extracted from some example images. This restriction to patches mainly arises from limited memory, and limited computational resources.

Roughly speaking, starting from some initial dictionary the learning algorithms iteratively update the atoms of the dictionary, such that the sparsity of the training set is increased. This procedure is often performed via block-coordinate relaxation, which alternates between finding the sparsest representation of the training set while fixing the atoms, and optimizing the atoms that most accurately reproduce the training set using the previously determined sparse representation. Three conceptually different approaches for learning a dictionary became well established, which are probabilistic ones like [15], clustering based ones such as K-SVD [16], and recent approaches which aim at learning dictionaries with specific matrix structures that allow fast computations like [17]. For a comprehensive overview of dictionary learning techniques see [18].

C. Analysis Model

An alternative to the synthesis model (3) for reconstructing a signal, is to solve

$$\mathbf{s}^\star = \underset{\mathbf{s}\in\mathbb{R}^n}{\arg\min}\; g(\mathbf{\Omega s}) \;\text{ subject to }\; \|\mathcal{A}\mathbf{s}-\mathbf{y}\|_2^2 \leq \epsilon, \tag{5}$$

which is known as the *analysis model* [7]. Therein, $\mathbf{\Omega} \in \mathbb{R}^{k\times n}$ with $k \geq n$ is called the *analysis operator*, and the *analyzed vector* $\mathbf{\Omega s} \in \mathbb{R}^k$ is assumed to be sparse, where sparsity is again measured via an appropriate function g. In contrast to the synthesis model, where a signal is fully described by the non-zero elements of $\mathbf{x}$, in the analysis model the zero elements of the analyzed vector $\mathbf{\Omega s}$ described the subspace containing the signal. To emphasize this difference, the term *cosparsity* has been introduced in [19], which simply counts the number of zero elements of $\mathbf{\Omega s}$. As the sparsity in the synthesis model depends on the chosen dictionary, the cosparsity of an analyzed signal depends on the choice of the analysis operator $\mathbf{\Omega}$.

Different analysis operators proposed in the literature include the fused Lasso [20], the translation invariant wavelet transform [21], and probably best known the finite difference operator which is closely related to the total-variation [22]. They all have shown very good performance when used within the analysis model for solving diverse inverse problems in imaging. The question is: *Can the performance of analysis based signal reconstruction be improved when a learned analysis operator is applied instead of a predefined one, as it is the case for the synthesis model where learned dictionaries outperform analytic dictionaries?* In [7], 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 pseudo-inverse as the learned analysis operator fails. Hence, different algorithms are required for analysis operator learning.

D. Contributions

In this work, we introduce a new algorithm based on geometric optimization for learning a patch based analysis operator from a set of training samples, which we name GOAL (GeOmetric Analysis operator Learning). The method relies on a minimization problem, which is carefully motivated in Section II-B. Therein, we also discuss the question of what is a suitable analysis operator for image reconstruction, and how to antagonize overfitting the operator to a subset of the training samples. An efficient geometric conjugate gradient method on the so-called oblique manifold is proposed in Section III for learning the analysis operator. Furthermore, in Section IV we explain how to apply the local patch based analysis operator to achieve global reconstruction results. Section V sheds some light on the influence of the parameters required by GOAL and how to select them, and compares our method to other analysis operator learning techniques. The quality of the operator learned by GOAL on natural image patches is further investigated in terms of image denoising, inpainting, and single image super-resolution. The numerical results show the broad and effective applicability of our general approach.

E. Notations

Matrices are written as capital calligraphic letters like $\mathcal{X}$, column vectors are denoted by boldfaced small letters e.g. $\mathbf{x}$, whereas scalars are either capital or small letters like n, N. By v_i we denote the i^{th} element of the vector $\mathbf{v}$, v_{ij} denotes the i^{th} element in the j^{th} column of a matrix $\mathcal{V}$. The vector $\mathbf{v}_{:,i}$ denotes the i^{th} column of $\mathcal{V}$ whereas $\mathbf{v}_{i,:}$ denotes the transposed of the i^{th} row of $\mathcal{V}$. By $\mathcal{E}_{ij}$, we denote a matrix whose i^{th} entry in the j^{th} column is equal to one, and all others are zero. $\mathcal{I}_k$ denotes the identity matrix of dimension $(k\times k)$, $\mathbf{0}$ denotes the zero-matrix of appropriate dimension, and $\operatorname{ddiag}(\mathcal{V})$ is the diagonal matrix whose entries on the diagonal are those of $\mathcal{V}$. By $\|\mathcal{V}\|_F^2 = \sum_{i,j} v_{ij}^2$ we denote the squared Frobenius norm of a matrix $\mathcal{V}$, $\operatorname{tr}(\mathcal{V})$ is the trace of $\mathcal{V}$, and $\operatorname{rk}(\mathcal{V})$ denotes the rank.

II. Analysis Operator Learning

A. Prior Art

The topic of analysis operator learning has only recently started to be investigated, and only few prior work exists. In the sequel, we shortly review analysis operator learning methods that are applicable for image processing tasks.

Given a set of M training samples $\left\{\mathbf{s}_i \in \mathbb{R}^n\right\}_{i=1}^M$, the goal of analysis operator learning is to find a matrix $\mathbf{\Omega} \in \mathbb{R}^{k \times n}$ with $k \geq n$, which leads to a maximally cosparse representation $\mathbf{\Omega}\mathbf{s}_i$ of each training sample. As mentioned in Subsection I-B, the training samples are distinctive vectorized image patches extracted from a set of example images. Let $\mathcal{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_M] \in \mathbb{R}^{n \times M}$ be a matrix where the training samples constitute its columns, then the problem is to find

$$\mathbf{\Omega}^\star = \arg\min_{\mathbf{\Omega}} \; G(\mathbf{\Omega}\mathcal{S}), \tag{6}$$

where $\mathbf{\Omega}$ is subject to some constraints, and G is some function that measures the sparsity of the matrix $\mathbf{\Omega}\mathcal{S}$. In [23], an algorithm is proposed in which the rows of the analysis operator are found sequentially 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 $\mathcal{S}_R$ 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 $\mathcal{S}_R\mathcal{S}_R^\top$. 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 $\mathbf{\Omega}$ as a new row, otherwise it is discarded. This process is repeated until the desired number k of rows have been found.

An adaption of the widely known K-SVD dictionary learning algorithm to the problem of analysis operator learning is presented in [24]. As in the original K-SVD algorithm, $G(\mathbf{\Omega}\mathcal{S}) = \sum_i \|\mathbf{\Omega}\mathbf{s}_i\|_0$ is employed as the sparsifying function and the target cosparsity is required as an input to the algorithm. The arising optimization problem is solved by alternating between a sparse coding stage over each training sample while fixing $\mathbf{\Omega}$ using an ordinary analysis pursuit method, and updating the analysis operator using the optimized training set. Then, each row of $\mathbf{\Omega}$ is updated in a similar way as described in the previous paragraph for the method of [23]. Interestingly, the operator learned on piecewise constant image patches by [23] and [24] closely mimics the finite difference operator.

In [25], the authors use $G(\mathbf{\Omega}\mathcal{S}) = \sum_i \|\mathbf{\Omega}\mathbf{s}_i\|_1$ as the sparsity promoting function and suggest a constrained optimization technique that utilizes a projected subgradient method for iteratively solving (6). To exclude the trivial solution, the set of possible analysis operators is restricted to the set of Uniform Normalized Tight Frames, i.e. matrices with uniform row norm and orthonormal columns. The authors state that this algorithm has the limitation of requiring noiseless training samples whose analyzed vectors $\{\mathbf{\Omega}\mathbf{s}_i\}_{i=1}^M$ are exactly cosparse.

To overcome this restriction, the same authors propose an extension of this algorithm that simultaneously learns the analysis operator and denoises the training samples, cf. [26]. This is achieved by alternating between updating the analysis operator via the projected subgradient algorithm and denoising the samples using an Augmented Lagrangian method. Therein, the authors state that their results for image denoising using the learned operator are only slightly worse compared to employing the commonly used finite difference operator.

An interesting idea related to the analysis model, called Fields-of-Experts (FoE) has been proposed in [27]. The method relies on learning high-order Markov Random Field image priors with potential functions extending over large pixel neighborhoods, i.e. overlapping image patches. Motivated by a probabilistic model, they use the student-t distribution of several linear filter responses as the potential function, where the filters, which correspond to atoms from an analysis operator point of view, have been learned from training patches. Compared to our work and the methods explained above, their learned operator used in the experiments is underdetermined, i.e. $k < n$, the algorithms only works for small patches due to computational reasons, and the atoms are learned independently, while in contrast GOAL updates the analysis operator as a whole.

B. Motivation of Our Approach

In the quest for designing an analysis operator learning algorithm, the natural question arises: *What is a good analysis operator for our needs?* Clearly, given a signal $\mathbf{s}$ that belongs to a certain signal class, the aim is to find an $\mathbf{\Omega}$ such that $\mathbf{\Omega}\mathbf{s}$ is *as sparse as possible*. This motivates to minimize the *expected sparsity* $\mathbb{E}[g(\mathbf{\Omega}\mathbf{s})]$. All approaches presented in Subsection II-A can be explained in this way, i.e. for their sparsity measure g they aim at learning an $\mathbf{\Omega}$ that minimizes the empirical mean of the sparsity over all randomly drawn training samples. This, however, does not necessarily mean to learn *the optimal* $\mathbf{\Omega}$ if the purpose is to reconstruct several signals belonging to a diverse class, e.g. natural image patches. The reason for this is that even if the *expected* sparsity is low, it may

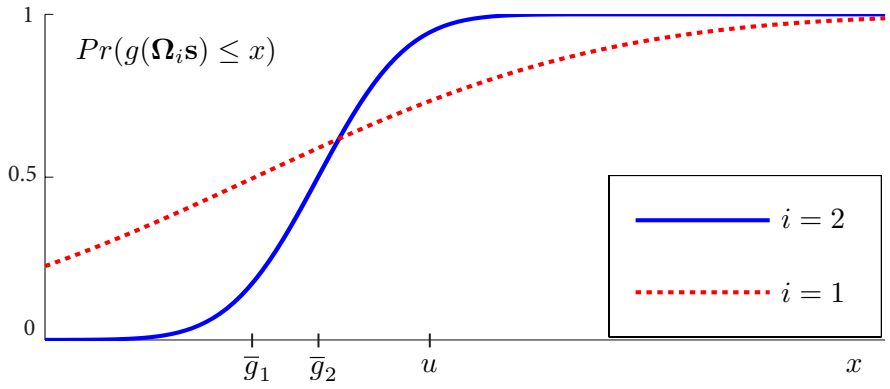


Fig. 1. Illustration of two possible distributions $Pr(g(\mathbf{\Omega}_i\mathbf{s}) \leq x)$ for two analysis operators. $\mathbf{\Omega}_1$: low expectation $\overline{g}_1$, high variance (dashed line); $\mathbf{\Omega}_2$: moderate expectation $\overline{g}_2$, moderate variance. Although $\mathbf{\Omega}_1$ yields a smaller expectation, there are more signals compared to $\mathbf{\Omega}_2$ where the sparsity model fails, i.e. $Pr(g(\mathbf{\Omega}_1\mathbf{s}) \geq u) > Pr(g(\mathbf{\Omega}_2\mathbf{s}) \geq u)$ for a suitable upper bound u.

happen with 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 $Pr(g(\mathbf{\Omega}\mathbf{s}) \geq u)$ exceeds a tolerable value, cf. Figure 1.

The algorithm presented here aims at minimizing the empirical expectation of a sparsifying function $g(\mathbf{\Omega}\mathbf{s}_i)$ for all training samples $\mathbf{s}_i$, while additionally keeping the empirical variance moderate. In other words, we try to avoid that the analyzed vectors of many similar training samples become *very sparse* and consequently prevent $\mathbf{\Omega}$ from being adapted to the remaining ones. 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. homogeneous regions, that bias the learning process.

Concretely, we want to find an $\mathbf{\Omega}$ that minimizes both the squared empirical mean

$$\overline{g}^2 = \left(\tfrac{1}{M}\sum_i g(\mathbf{\Omega}\mathbf{s}_i)\right)^2 \tag{7}$$

and the empirical variance

$$s^2 = \tfrac{1}{M}\sum_i \left(g(\mathbf{\Omega}\mathbf{s}_i) - \overline{g}\right)^2 \tag{8}$$

of the sparsity of the analyzed vectors. We achieve this by minimizing the sum of both, which is readily given by

$$\overline{g}^2 + s^2 = \tfrac{1}{M}\sum_i g(\mathbf{\Omega}\mathbf{s}_i)^2. \tag{9}$$

Using $g(\cdot) = \|\cdot\|_p^p$, and introducing the factor $\frac{1}{2}$ the function we employ reads as

$$J_p(\mathcal{V}) := \tfrac{1}{2M}\sum_{j=1}^{M}\left(\tfrac{1}{p}\sum_{i=1}^{k}|v_{ij}|^p\right)^2 = \tfrac{1}{2M}\sum_{j=1}^{M}\left(\tfrac{1}{p}\|\mathbf{v}_{:,j}\|_p^p\right)^2, \tag{10}$$

with $0 \leq p \leq 1$ and $\mathcal{V} = \mathbf{\Omega}\mathcal{S}$.

Certainly, without additional prior assumptions on $\mathbf{\Omega}$, the useless solution $\mathbf{\Omega} = \mathbf{0}$ is the global minimizer of Problem (6). To avoid the trivial solution and for other reasons explained later in this section, we regularize the problem by imposing the following three constraints on $\mathbf{\Omega}$.

(i) The *rows* of $\mathbf{\Omega}$ have unit Euclidean norm, i.e. $\|\boldsymbol{\omega}_{i,:}\|_2 = 1$ for $i = 1, \ldots, k$.
(ii) The analysis operator $\mathbf{\Omega}$ has full rank, i.e. $\mathrm{rk}(\mathbf{\Omega}) = n$.
(iii) The analysis operator $\mathbf{\Omega}$ does not have linear dependent rows, i.e. $\boldsymbol{\omega}_{i,:} \neq \pm\boldsymbol{\omega}_{j,:}$ for $i \neq j$.

The rank condition (ii) on $\mathbf{\Omega}$ is motivated by the fact that different input samples $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{R}^n$ with $\mathbf{s}_1 \neq \mathbf{s}_2$ should be mapped to different analyzed vectors $\mathbf{\Omega}\mathbf{s}_1 \neq \mathbf{\Omega}\mathbf{s}_2$. With Condition (iii) redundant transform coefficients in an analyzed vector are avoided.

These constraints motivate the consideration of the set of full rank matrices with normalized *columns*, which admits a manifold structure known as the *oblique manifold* [28]

$$\mathrm{OB}(n,k) := \{\mathcal{X} \in \mathbb{R}^{n\times k} | \, \mathrm{rk}(\mathcal{X}) = n, \mathrm{ddiag}(\mathcal{X}^\top \mathcal{X}) = \mathcal{I}_k\}. \tag{11}$$

Note, that this definition only yields a non-empty set if $k \geq n$, which is the interesting case in this work. Thus, from now on, we assume $k \geq n$. Remember that we require the *rows* of $\mathbf{\Omega}$ to have unit Euclidean norm. Hence, we restrict the *transposed* of the learned analysis operator to be an element of $\mathrm{OB}(n,k)$.

Since $\mathrm{OB}(n,k)$ is open and dense in the set of matrices with normalized columns, we need a penalty function that ensures the rank constraint (ii) and prevents iterates to approach the boundary of $\mathrm{OB}(n,k)$.

Lemma 1: The inequality $0 < \det(\frac{1}{k}\mathcal{X}\mathcal{X}^\top) \leq (\frac{1}{n})^n$ holds true for all $\mathcal{X} \in \mathrm{OB}(n,k)$, where $1 < n \leq k$.

Proof: Due to the full rank condition on $\mathcal{X}$, the product $\mathcal{X}\mathcal{X}^\top$ is positive definite, consequently the strict inequality $0 < \det(\frac{1}{k}\mathcal{X}\mathcal{X}^\top)$ applies. To see the second inequality of Lemma 1, observe that

$$\|\mathcal{X}\|_F^2 = \mathrm{tr}(\mathcal{X}\mathcal{X}^\top) = k, \tag{12}$$

which implies $\mathrm{tr}(\frac{1}{k}\mathcal{X}\mathcal{X}^\top) = 1$. Since the trace of a matrix is equal to the sum of its eigenvalues, which are strictly positive in our case, it follows that the strict inequality $0 < \lambda_i < 1$ holds true for all eigenvalues λ_i of $\frac{1}{k}\mathcal{X}\mathcal{X}^\top$. From the well known relation between the arithmetic and the geometric mean we see

$$\sqrt[n]{\Pi\lambda_i} \leq \tfrac{1}{n}\sum \lambda_i. \tag{13}$$

Now, since the determinant of a matrix is equal to the product of its eigenvalues, and with $\sum \lambda_i = \mathrm{tr}(\frac{1}{k}\mathcal{X}\mathcal{X}^\top) = 1$, we have

$$\det(\tfrac{1}{k}\mathcal{X}\mathcal{X}^\top) = \Pi\lambda_i \leq (\tfrac{1}{n})^n, \tag{14}$$

which completes the proof. ■

Recalling that $\mathbf{\Omega}^\top \in \mathrm{OB}(n,k)$ and considering Lemma 1, we can enforce the full rank constraint with the penalty function

$$h(\mathbf{\Omega}) := -\tfrac{1}{n\log(n)} \log\det(\tfrac{1}{k}\mathbf{\Omega}^\top\mathbf{\Omega}). \tag{15}$$

Regarding Condition (iii), the following result proves useful.

Lemma 2: For a matrix $\mathcal{X} \in \mathrm{OB}(n,k)$ with $1 < n \leq k$, the inequality $|\mathbf{x}_{:,i}^\top \mathbf{x}_{:,j}| \leq 1$ applies, where equality holds true if and only if $\mathbf{x}_{:,i} = \pm\mathbf{x}_{:,j}$.

Proof: By the definition of $\mathrm{OB}(n,k)$ the columns of $\mathcal{X}$ are normalized, consequently Lemma 2 follows directly from the Cauchy-Schwarz inequality. ■

Thus, Condition (iii) can be enforced via the logarithmic barrier function of the scalar products between all distinctive rows of $\mathbf{\Omega}$, i.e.

$$r(\mathbf{\Omega}) := -\sum_{1\leq i<j\leq k} \log(1 - (\boldsymbol{\omega}_{i,:}^\top \boldsymbol{\omega}_{j,:})^2). \tag{16}$$

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

$$\mathbf{\Omega}^\top = \underset{\mathcal{X}\in\mathrm{OB}(n,k)}{\arg\min}\ J_p(\mathcal{X}^\top\mathcal{S}) + \kappa\ h(\mathcal{X}^\top) + \mu\ r(\mathcal{X}^\top). \tag{17}$$

Therein, the two weighting factors $\kappa, \mu \in \mathbb{R}^+$ control the influence of the two constraints on the final solution. The following lemma clarifies the role of κ.

Lemma 3: Let $\mathbf{\Omega}$ be a minimum of h in the set of transposed oblique matrices, i.e.

$$\mathbf{\Omega}^\top \in \underset{\mathcal{X}\in\mathrm{OB}(n,k)}{\arg\min}\ h(\mathcal{X}^\top), \tag{18}$$

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

Proof: It is well known that equality of the arithmetic and the geometric mean in Equation (13) holds true, if and only if all eigenvalues λ_i of $\frac{1}{k}\mathcal{X}\mathcal{X}^\top$ are equal, i.e. $\lambda_1 = \ldots = \lambda_n$. Hence, if $\mathbf{\Omega}^\top \in \underset{\mathcal{X}\in\mathrm{OB}(n,k)}{\arg\min}\ h(\mathcal{X}^\top)$, then

$\det(\frac{1}{k}\mathbf{\Omega}^\top\mathbf{\Omega}) = (\frac{1}{n})^n$, and consequently all singular values of $\mathbf{\Omega}$ coincide. This implies that the condition number of $\mathbf{\Omega}$, which is defined as the quotient of the largest to the smallest singular value, is equal to one. ■
With other words, the minima of h are uniformly normalized tight frames, cf. [25], [26]. From Lemma 3 we can conclude that with larger κ the condition number of $\mathbf{\Omega}$ approaches one. Now, recall the inequality

$$\sigma_{\min}\|\mathbf{s}_1 - \mathbf{s}_2\|_2 \leq \|\mathbf{\Omega}(\mathbf{s}_1 - \mathbf{s}_2)\|_2 \leq \sigma_{\max}\|\mathbf{s}_1 - \mathbf{s}_2\|_2, \tag{19}$$

with $\sigma_{\min}$ being the smallest and $\sigma_{\max}$ being the largest singular value of $\mathbf{\Omega}$. From this 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. The parameter μ regulates the redundancy between the rows of the analysis operator and consequently avoids redundant coefficients in the analyzed vector $\mathbf{\Omega}\mathbf{s}$.

Lemma 4: The difference between any two entries of the analyzed vector $\mathbf{\Omega}\mathbf{s}$ is bounded by

$$|\boldsymbol{\omega}_{i,:}^\top\mathbf{s} - \boldsymbol{\omega}_{j,:}^\top\mathbf{s}| \leq \sqrt{2(1 - \boldsymbol{\omega}_{i,:}^\top\boldsymbol{\omega}_{j,:})}\ \|\mathbf{s}\|_2. \tag{20}$$

Proof: From the Cauchy-Schwarz inequality we get

$$|\boldsymbol{\omega}_{i,:}^\top\mathbf{s} - \boldsymbol{\omega}_{j,:}^\top\mathbf{s}| = |(\boldsymbol{\omega}_{i,:} - \boldsymbol{\omega}_{j,:})^\top\mathbf{s}| \leq \|\boldsymbol{\omega}_{i,:} - \boldsymbol{\omega}_{j,:}\|_2\|\mathbf{s}\|_2. \tag{21}$$

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

The above lemma implies, that if the i^{th} entry of the analyzed vector is significantly larger than 0 then a large absolute value of $\boldsymbol{\omega}_{i,:}^\top\boldsymbol{\omega}_{j,:}$ prevents the j^{th} entry to be small. To achieve large cosparsity, this is an unwanted effect that our approach avoids via the log-barrier function r in (16). It is worth mentioning that the same effect is achieved by minimizing the analysis operator's mutual coherence $\max_{i\neq j} |\boldsymbol{\omega}_{i,:}^\top\boldsymbol{\omega}_{j,:}|$ and that our experiments suggest that enlarging μ leads to minimizing the mutual coherence.

In the next section, we explain how the manifold structure of $\mathrm{OB}(n, k)$ can be exploited to efficiently learn the analysis operator.

III. Analysis Operator Learning Algorithm

Knowing that the feasible set of solutions to Problem (17) is restricted to a smooth manifold allows us to formulate a geometric conjugate gradient (CG-) method to learn the analysis operator. Geometric CG-methods have been proven efficient in various applications, due to the combination of moderate computational complexity and good convergence properties, see e.g. [29] for a CG-type method on the oblique manifold.

To make this work self contained, we start by shortly reviewing the general concepts of optimization on matrix manifolds. After that we present the concrete formulas and implementation details for our optimization problem on the oblique manifold. For an in-depth introduction on optimization on matrix manifolds, we refer the interested reader to [30].

A. Optimization on Matrix Manifolds

Let M be a smooth Riemannian submanifold of $\mathbb{R}^{n\times k}$ with the standard Frobenius inner product $\langle\mathcal{Q}, \mathcal{P}\rangle := \mathrm{tr}(\mathcal{Q}^\top\mathcal{P})$, and let $f\colon \mathbb{R}^{n\times k} \to \mathbb{R}$ be a differentiable cost function. We consider the problem of finding

$$\arg\min_{\mathcal{X}\in\mathrm{M}} f(\mathcal{X}). \tag{22}$$

The concepts presented in this subsection are visualized in Figure 2 to alleviate the understanding.

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

The Riemannian gradient of f at $\mathcal{X}$ is an element of the tangent space $T_\mathcal{X}\mathrm{M}$ that points in the direction of steepest ascent of the cost function on the manifold. As we require M to be a submanifold of $\mathbb{R}^{n\times k}$ and since by

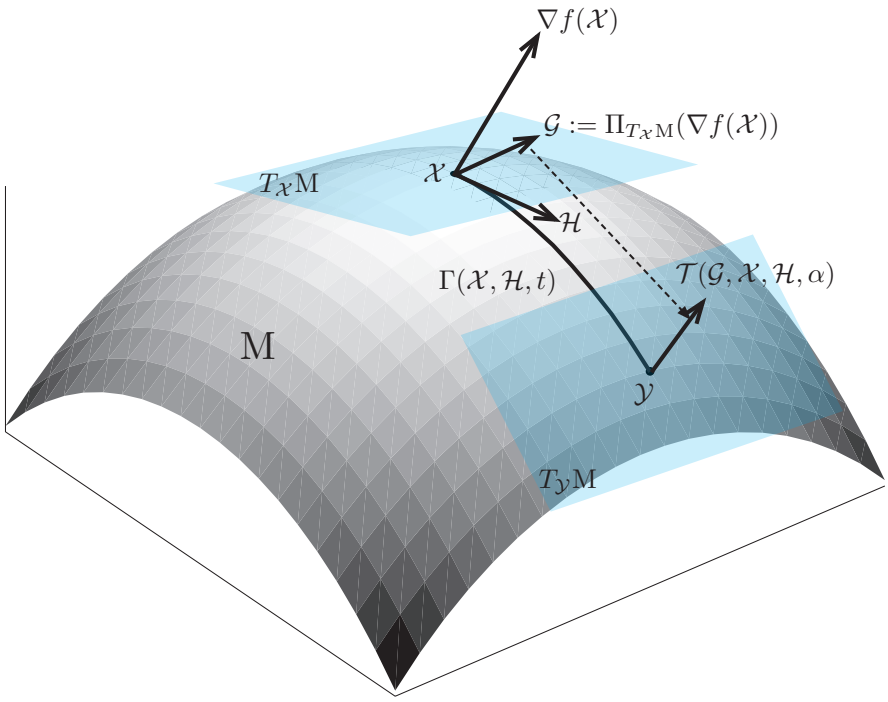


Fig. 2. This figure shows two points $\mathcal{X}$ and $\mathcal{Y}$ on a manifold M together with their tangent spaces $T_\mathcal{X}\mathrm{M}$ and $T_\mathcal{Y}\mathrm{M}$. Furthermore, the Euclidean gradient $\nabla f(\mathcal{X})$ and its projection onto the tangent space $\Pi_{T_\mathcal{X}\mathrm{M}}(\nabla f(\mathcal{X}))$ are depicted. The geodesic $\Gamma(\mathcal{X}, \mathcal{H}, t)$ in the direction of $\mathcal{H} \in T_\mathcal{X}\mathrm{M}$ connecting the two points is shown. The dashed line typifies the role of a parallel transport of the gradient in $T_\mathcal{X}\mathrm{M}$ to $T_\mathcal{Y}\mathrm{M}$.

assumption f is defined on the whole $\mathbb{R}^{n\times k}$, the Riemannian gradient $\mathcal{G}(\mathcal{X})$ is simply the orthogonal projection of the (standard) gradient $\nabla f(\mathcal{X})$ onto the tangent space $T_\mathcal{X}\mathrm{M}$. In formulas, this reads as

$$\mathcal{G}(\mathcal{X}) := \Pi_{T_\mathcal{X}\mathrm{M}}(\nabla f(\mathcal{X})). \tag{23}$$

A *geodesic* is a smooth curve $\Gamma(\mathcal{X}, \Xi, t)$ emanating from $\mathcal{X}$ in the direction of $\Xi \in T_\mathcal{X}\mathrm{M}$, 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.

Conventional line search methods search for the next iterate along a straight line. This is generalized to the manifold setting as follows. Given a current optimal point $\mathcal{X}^{(i)}$ and a search direction $\mathcal{H}^{(i)} \in T_{\mathcal{X}^{(i)}}\mathrm{M}$ at the i^{th} iteration, the step size $\alpha^{(i)}$ which leads to sufficient decrease of f can be determined by finding the minimizer of

$$\alpha^{(i)} = \arg\min_{t\geq 0} f(\Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, t)). \tag{24}$$

Once $\alpha^{(i)}$ has been determined, the new iterate is computed by

$$\mathcal{X}^{(i+1)} = \Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, \alpha^{(i)}). \tag{25}$$

Now, one straightforward approach to minimize f is to alternate Equations (23), (24), and (25) using $\mathcal{H}^{(i)} = -\mathcal{G}^{(i)}$, with the short hand notation $\mathcal{G}^{(i)} := \mathcal{G}(\mathcal{X}^{(i)})$, which corresponds to the steepest descent on a Riemannian manifold. However, as in standard optimization, steepest descent only has a linear rate of convergence. Therefore, we employ a conjugate gradient method on a manifold, as it offers a superlinear rate of convergence, while still being applicable to large scale optimization problems with low computational complexity.

In CG-methods, the updated search direction $\mathcal{H}^{(i+1)} \in T_{\mathcal{X}^{(i+1)}}\mathrm{M}$ is a linear combination of the gradient $\mathcal{G}^{(i+1)} \in T_{\mathcal{X}^{(i+1)}}\mathrm{M}$ and the previous search direction $\mathcal{H}^{(i)} \in T_{\mathcal{X}^{(i)}}\mathrm{M}$. Since adding vectors that belong to different tangent spaces is not defined, we need to map $\mathcal{H}^{(i)}$ from $T_{\mathcal{X}^{(i)}}\mathrm{M}$ to $T_{\mathcal{X}^{(i+1)}}\mathrm{M}$. This is done by the so-called parallel transport $\mathcal{T}(\Xi, \mathcal{X}^{(i)}, \mathcal{H}^{(i)}, \alpha^{(i)})$, which transports a tangent vector $\Xi \in T_{\mathcal{X}^{(i)}}\mathrm{M}$ along the geodesic $\Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, t)$ to the tangent space $T_{\mathcal{X}^{(i+1)}}\mathrm{M}$. Now, using the shorthand notation

$$\mathcal{T}_\Xi^{(i+1)} := \mathcal{T}(\Xi, \mathcal{X}^{(i)}, \mathcal{H}^{(i)}, \alpha^{(i)}), \tag{26}$$

the new search direction is computed by

$$\mathcal{H}^{(i+1)} = -\mathcal{G}^{(i+1)} + \beta^{(i)} \mathcal{T}_{\mathcal{H}^{(i)}}^{(i+1)}, \tag{27}$$

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

$$\beta_{FR}^{(i)} = \frac{\langle \mathcal{G}^{(i+1)}, \mathcal{G}^{(i+1)} \rangle}{\langle \mathcal{G}^{(i)}, \mathcal{G}^{(i)} \rangle}, \tag{28}$$

$$\beta_{HS}^{(i)} = \frac{\langle \mathcal{G}^{(i+1)}, \mathcal{Y}^{(i+1)} \rangle}{\langle \mathcal{T}_{\mathcal{H}^{(i)}}^{(i+1)}, \mathcal{Y}^{(i+1)} \rangle}, \tag{29}$$

$$\beta_{DY}^{(i)} = \frac{\langle \mathcal{G}^{(i+1)}, \mathcal{G}^{(i+1)} \rangle}{\langle \mathcal{T}_{\mathcal{H}^{(i)}}^{(i+1)}, \mathcal{Y}^{(i+1)} \rangle}. \tag{30}$$

Now, a solution to Problem (22) is computed by alternating between finding the search direction on M and updating the current optimal point until some user-specified convergence criterion is met, or a maximum number of iterations has been reached.

B. Geometric Conjugate Gradient for Analysis Operator Learning

In this subsection we derive all ingredients to implement the geometric conjugate gradient method as described in the previous subsection for the task of learning the analysis operator. Results regarding the geometry of $\mathrm{OB}(n, k)$ are derived e.g. in [30]. To enhance legibility, and since the dimensions n and k are fixed throughout the rest of the paper, the oblique manifold is further on denoted by OB.

The tangent space at $\mathcal{X} \in \mathrm{OB}$ is given by

$$T_{\mathcal{X}}\mathrm{OB} = \{\Xi \in \mathbb{R}^{n \times k} | \operatorname{ddiag}(\mathcal{X}^\top \Xi) = \mathbf{0}\}. \tag{31}$$

The orthogonal projection of a matrix $\mathcal{Q} \in \mathbb{R}^{n \times k}$ onto the tangent space $T_{\mathcal{X}}\mathrm{OB}$ is

$$\Pi_{T_{\mathcal{X}}\mathrm{OB}}(\mathcal{Q}) = \mathcal{Q} - \mathcal{X} \operatorname{ddiag}(\mathcal{X}^\top \mathcal{Q}). \tag{32}$$

Regarding geodesics, note that in general a geodesic is the solution of a second order ordinary differential equation, meaning that for arbitrary manifolds, its computation as well as computing the parallel transport is not feasible. Fortunately, as the oblique manifold is a Riemannian submanifold of a product of k unit spheres S^{n-1}, the formulas for parallel transport and the exponential mapping allow an efficient implementation.

Let $\mathbf{x} \in S^{n-1}$ be a point on a sphere and $\mathbf{h} \in T_{\mathbf{x}} S^{n-1}$ be a tangent vector at $\mathbf{x}$, then the geodesic in the direction of $\mathbf{h}$ is a great circle

$$\gamma(\mathbf{x}, \mathbf{h}, t) = \begin{cases} \mathbf{x}, & \text{if } \|\mathbf{h}\|_2 = 0 \\ \mathbf{x}\cos(t\|\mathbf{h}\|_2) + \mathbf{h}\frac{\sin(t\|\mathbf{h}\|_2)}{\|\mathbf{h}\|_2}, & \text{otherwise.} \end{cases} \tag{33}$$

The associated parallel transport of a tangent vector $\boldsymbol{\xi} \in T_{\mathbf{x}} S^{n-1}$ along the great circle $\gamma(\mathbf{x}, \mathbf{h}, t)$ reads as

$$\tau(\boldsymbol{\xi}, \mathbf{x}, \mathbf{h}, t) = \boldsymbol{\xi} - \frac{\boldsymbol{\xi}^\top \mathbf{h}}{\|\mathbf{h}\|_2^2} \Big(\mathbf{x}\|\mathbf{h}\|_2 \sin(t\|\mathbf{h}\|_2) + \mathbf{h}(1 - \cos(t\|\mathbf{h}\|_2)) \Big). \tag{34}$$

As OB is a submanifold of the product of unit spheres, the geodesic through $\mathcal{X} \in \mathrm{OB}$ in the direction of $\mathcal{H} \in T_{\mathcal{X}}\mathrm{OB}$ is simply the combination of the great circles emerging by concatenating each column of $\mathcal{X}$ with the corresponding column of $\mathcal{H}$, i.e.

$$\Gamma(\mathcal{X}, \mathcal{H}, t) = \Big[\gamma(\mathbf{x}_{:,1}, \mathbf{h}_{:,1}, t), \ldots, \gamma(\mathbf{x}_{:,k}, \mathbf{h}_{:,k}, t) \Big]. \tag{35}$$

Accordingly, the parallel transport of $\Xi \in T_{\mathcal{X}}\mathrm{OB}$ along the geodesic $\Gamma(\mathcal{X}, \mathcal{H}, t)$ is given by

$$\mathcal{T}(\Xi, \mathcal{X}, \mathcal{H}, t) = \left[\tau(\boldsymbol{\xi}_{:,1}, \mathbf{x}_{:,1}, \mathbf{h}_{:,1}, t), \ldots, \tau(\boldsymbol{\xi}_{:,k}, \mathbf{x}_{:,k}, \mathbf{h}_{:,k}, t)\right]. \tag{36}$$

Now, to use the geometric CG-method for learning the analysis operator, we require a differentiable cost function f. Since, the cost function presented in Problem (17) is not differentiable due to the non-smoothness of the (p, q)-pseudo-norm (10), we exchange Function (10) with a smooth approximation, which is given by

$$J_{p,\nu}(\mathcal{V}) := \tfrac{1}{2M} \sum_{j=1}^{M} \left(\tfrac{1}{p} \sum_{i=1}^{k} (v_{ij}^2 + \nu)^{\frac{p}{2}} \right)^2, \tag{37}$$

with $\nu \in \mathbb{R}^+$ being the smoothing parameter. The smaller ν is, the more closely the approximation resembles the original function. Again, taking $\mathcal{V} = \Omega\mathcal{S}$ and with the shorthand notation $z_{ij} := (\Omega\mathcal{S})_{ij}$, the gradient of the applied sparsity promoting function (37) reads as

$$\tfrac{\partial}{\partial \boldsymbol{\Omega}} J_{p,\nu}(\boldsymbol{\Omega}\mathcal{S}) = \left[\tfrac{1}{M} \sum_{j=1}^{M} \tfrac{1}{p} \sum_{i=1}^{k} (z_{ij}^2 + \nu)^{\frac{p}{2}} \sum_{i=1}^{k} \left\{ z_{ij} (z_{ij}^2 + \nu)^{\frac{p}{2}-1} \mathcal{E}_{ij} \right\} \right] \mathcal{S}^\top. \tag{38}$$

The gradient of the rank penalty term (15) is

$$\tfrac{\partial}{\partial \boldsymbol{\Omega}} h(\boldsymbol{\Omega}) = -\tfrac{2}{kn\log(n)} \boldsymbol{\Omega} (\tfrac{1}{k} \boldsymbol{\Omega}^\top \boldsymbol{\Omega})^{-1} \tag{39}$$

and the gradient of the logarithmic barrier function (16) is

$$\tfrac{\partial}{\partial \boldsymbol{\Omega}} r(\boldsymbol{\Omega}) = \left[\sum_{1 \le i < j \le k} \frac{2\boldsymbol{\omega}_{i,:}^\top \boldsymbol{\omega}_{j,:}}{1 - (\boldsymbol{\omega}_{i,:}^\top \boldsymbol{\omega}_{j,:})^2} (\mathcal{E}_{ij} + \mathcal{E}_{ji}) \right] \boldsymbol{\Omega}. \tag{40}$$

Combining Equations (38), (39), and (40), the gradient of the cost function

$$f(\mathcal{X}) := J_{p,\nu}(\mathcal{X}^\top \mathcal{S}) + \kappa\ h(\mathcal{X}^\top) + \mu\ r(\mathcal{X}^\top) \tag{41}$$

which is used for learning the analysis operator reads as

$$\nabla f(\mathcal{X}) = \tfrac{\partial}{\partial \mathcal{X}} J_{p,\nu}(\mathcal{X}^\top \mathcal{S}) + \kappa \tfrac{\partial}{\partial \mathcal{X}} h(\mathcal{X}^\top) + \mu \tfrac{\partial}{\partial \mathcal{X}} r(\mathcal{X}^\top). \tag{42}$$

Regarding the CG-update parameter $\beta^{(i)}$, we employ a hybridization of the Hestenes-Stiefel Formula (29) and the Dai Yuan formula (30)

$$\beta_{hyb}^{(i)} = \max\left(0, \min(\beta_{DY}^{(i)}, \beta_{HS}^{(i)})\right), \tag{43}$$

which has been suggested in [31]. As explained therein, formula (43) combines the good numerical performance of HS with the desirable global convergence properties of DY.

Finally, to compute the step size $\alpha^{(i)}$, we use an adaption of the well-known backtracking line search to the geodesic $\Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, t)$. In that, an initial step size $t_0^{(i)}$ is iteratively decreased by a constant factor $c_1 < 1$ until the Armijo condition is met, see Algorithm 1 for the entire procedure. In our implementation we empirically chose $c_1 = 0.9$ and $c_2 = 10^{-2}$. As an initial guess for the step size at the first CG-iteration $i = 0$, we choose

$$t_0^{(0)} = \|\mathcal{G}^{(0)}\|_F^{-1}, \tag{44}$$

as proposed in [32]. In the subsequent iterations, the backtracking line search is initialized by the previous step size divided by the line search parameter, i.e. $t_0^{(i)} = \frac{\alpha^{(i-1)}}{c_1}$. Our complete approach for learning the analysis operator is summarized in Algorithm 2. Note, that under the conditions that the Fletcher-Reeves update formula is used and some mild conditions on the step-size selection, the convergence of Algorithm 2 to a critical point, i.e. $\liminf_{i\to\infty} \|\mathcal{G}^{(i)}\| = 0$, is guaranteed by a result provided in [33].

Algorithm 1 Backtracking Line Search on Oblique Manifold

Input: $t_0^{(i)} > 0,\ 0 < c_1 < 1,\ 0 < c_2 < 0.5,\ \mathcal{X}^{(i)}, \mathcal{G}^{(i)}, \mathcal{H}^{(i)}$
Set: $t \leftarrow t_0^{(i)}$
  while $f(\Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, t)) > f(\mathcal{X}^{(i)}) + tc_2\langle \mathcal{G}^{(i)}, \mathcal{H}^{(i)} \rangle$ **do**
    $t \leftarrow c_1 t$
  end while
Output: $\alpha^{(i)} \leftarrow t$

Algorithm 2 Geometric Analysis Operator Learning (GOAL)

Input: Initial analysis operator $\mathbf{\Omega}_{init}$, training data $\mathcal{S}$, parameters p, ν, κ, μ
Set: $i \leftarrow 0$, $\mathcal{X}^{(0)} \leftarrow \mathbf{\Omega}_{init}^{\top}$, $\mathcal{H}^{(0)} \leftarrow -\mathcal{G}^{(0)}$
  repeat
    $\alpha^{(i)} \leftarrow \arg\min_{t \geq 0} f(\Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, t))$, cf. Algorithm 1 in conjunction with Equation (41)
    $\mathcal{X}^{(i+1)} \leftarrow \Gamma(\mathcal{X}^{(i)}, \mathcal{H}^{(i)}, \alpha^{(i)})$, cf. Equation (35)
    $\mathcal{G}^{(i+1)} \leftarrow \Pi_{T_{\mathcal{X}^{(i+1)}}\mathrm{M}}(\nabla f(\mathcal{X}^{(i+1)}))$, cf. Equations (32) and (42)
    $\beta^{(i)} \leftarrow \max\left(0, \min(\beta_{DY}^{(i)}, \beta_{HS}^{(i)})\right)$, cf. Equations (29), (30)
    $\mathcal{H}^{(i+1)} \leftarrow -\mathcal{G}^{(i+1)} + \beta^{(i)} \mathcal{T}_{\mathcal{H}^{(i)}}^{(i+1)}$, cf. Equations (26), (36)
    $i \leftarrow i + 1$
  until $\|\mathcal{X}^{(i)} - \mathcal{X}^{(i-1)}\|_F < 10^{-4} \vee i =$ maximum # iterations
Output: $\mathbf{\Omega}^{\star} \leftarrow \mathcal{X}^{(i)\top}$

IV. Analysis Operator based Image Reconstruction

In this section we explain how the analysis operator $\mathbf{\Omega}^{\star} \in \mathbb{R}^{k \times n}$ is utilized for reconstructing an unknown image $\mathbf{s} \in \mathbb{R}^N$ from some measurements $\mathbf{y} \in \mathbb{R}^m$ following the analysis approach (5). Here, the vector $\mathbf{s} \in \mathbb{R}^N$ 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, we will loosely speak of $\mathbf{s}$ as the image.

Remember, that the size of $\mathbf{\Omega}^{\star}$ 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. Artifacts that arise from naïve patch-wise reconstruction are commonly reduced by considering overlapping patches. Thereby, each patch is reconstructed individually and the entire image is formed by averaging over the overlapping regions in a final step. However, this method misses global support during the reconstruction process, hence, it leads to poor inpainting results and is not applicable for e.g. Compressive Sensing tasks. To overcome these drawbacks, we use a method related to the patch based synthesis approach from [12] and the method used in [27], which provides global support from local information. Instead of optimizing over each patch individually and combining them in a final step, we 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, we assume square patches i.e. of size $(\sqrt{n} \times \sqrt{n})$ with $\sqrt{n}$ being a positive integer.

Formally, let $\mathbf{r} \subseteq \{1, \ldots, h\}$ and $\mathbf{c} \subseteq \{1, \ldots, w\}$ 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, d_h determine the degree of overlap between two adjacent patches in vertical, and horizontal direction, respectively. We consider all image patches whose center is an element of the cartesian product set $\mathbf{r} \times \mathbf{c}$. Hence, with $|\cdot|$ denoting the cardinality of a set, the total number of patches being considered is equal to $|\mathbf{r}||\mathbf{c}|$. Now, let $\mathcal{P}_{rc}$ be a binary $(n \times N)$ matrix that extracts the patch centered at position (r, c). With this notation, we formulate the (global) sparsity promoting function as

$$\sum_{r \in \mathbf{r}} \sum_{c \in \mathbf{c}} \sum_{i=1}^{k} ((\mathbf{\Omega}^{\star} \mathcal{P}_{rc} \mathbf{s})_i^2 + \nu)^{\frac{p}{2}}, \tag{45}$$

which measures the overall approximated ℓ_p-pseudo-norm of the considered analyzed image patches. We compactly rewrite Equation (45) as

$$g(\mathbf{\Omega}^F\mathbf{s}) := \sum_{i=1}^{K} \left((\mathbf{\Omega}^F\mathbf{s})_i^2 + \nu\right)^{\frac{p}{2}}, \tag{46}$$

with $K = k|\mathbf{r}||\mathbf{c}|$ and

$$\mathbf{\Omega}^F := \begin{bmatrix} \mathbf{\Omega}^\star\mathcal{P}_{r_1c_1} \\ \mathbf{\Omega}^\star\mathcal{P}_{r_1c_2} \\ \vdots \\ \mathbf{\Omega}^\star\mathcal{P}_{r_{|\mathbf{r}|}c_{|\mathbf{c}|}} \end{bmatrix} \in \mathbb{R}^{K\times N} \tag{47}$$

being the *global* analysis operator that expands the patch based one to the entire image. We treat image boundary effects by employing constant padding, i.e. replicating the values at the image boundaries $\lfloor\frac{\sqrt{n}}{2}\rfloor$ times, where $\lfloor\cdot\rfloor$ denotes rounding to the smaller integer. Certainly, for image processing applications $\mathbf{\Omega}^F$ is too large for being applied in terms of matrix vector multiplication. Fortunately, applying $\mathbf{\Omega}^F$ and its transposed can be implemented efficiently using sliding window techniques, and the matrix vector notation is solely used for legibility.

According to [34], we exploit the fact that the range of pixel intensities is limited by a lower bound b_l and an upper bound b_u. We enforce this bounding constraint by minimizing the differentiable function $\boldsymbol{b}(\mathbf{s}) := \sum_{i=1}^{N} b(s_i)$, where b is a penalty term given as

$$b(s) = \begin{cases} |s - b_u|^2 & \text{if } s \geq b_u \\ |s - b_l|^2 & \text{if } s \leq b_l \\ 0 & \text{otherwise} \end{cases}. \tag{48}$$

Finally, combining the two constraints (46) and (48) with the data fidelity term, the analysis based image reconstruction problem is to solve

$$\mathbf{s}^\star = \underset{\mathbf{s}\in\mathbb{R}^N}{\arg\min}\, \tfrac{1}{2}\|\mathcal{A}\mathbf{s} - \mathbf{y}\|_2^2 + \boldsymbol{b}(\mathbf{s}) + \lambda g(\mathbf{\Omega}^F\mathbf{s}). \tag{49}$$

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 $\mathcal{A} \in \mathbb{R}^{m\times N}$ and the measurements $\mathbf{y} \in \mathbb{R}^m$ are application dependent.

V. Evaluation and Experiments

The first part of this section aims at answering the question of what is a good analysis operator for solving image reconstruction problems and relates the quality of an analysis operator with its mutual coherence and its condition number. This, in turn allows to select the optimal weighting parameters κ and μ for GOAL. Using this parameters, we learn one general analysis operator $\mathbf{\Omega}^\star$ by GOAL, and compare its image denoising performance with other analysis approaches. In the second part, we employ this $\mathbf{\Omega}^\star$ unaltered for solving two classical image reconstruction tasks of image inpainting and single image super-resolution, and compare our results with the currently best analysis approach FoE [27], and state-of-the-art methods specifically designed for each respective application.

A. Global Parameters and Image Reconstruction

To quantify the reconstruction quality, as usual, we use the peak signal-to-noise ratio $\mathit{PSNR} = 10\log(255^2 N/\sum_{i=1}^{N}(s_i - s_i^\star)^2)$. Moreover, we measure the quality using the Mean Structural SIMilarity Index (*MSSIM*) [35], with the same set of parameters as originally suggested in [35]. 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, we fixed the size of the image patches to (8×8), i.e. $n = 64$. This is in accordance to the patch-sizes mostly used in the literature, and yields a good trade-off between reconstruction quality and numerical burden. Images are reconstructed by solving the minimization problem (49) via the conjugate gradient

method proposed in [34]. Considering the pixel intensity bounds, we used $b_l = 0$ and $b_u = 255$, which is the common intensity range in 8-bit grayscale image formats. The sparsity promoting function (46) with $p = 0.4$ and $\nu = 10^{-6}$ is used for both learning the analysis operator by GOAL, and reconstructing the images. Our patch based reconstruction algorithm as explained in Section IV achieves the best results for the maximum possible overlap $d_h = d_v = 1$. The Lagrange multiplier λ and the measurements matrix $\mathcal{A}$ depend on the application, and are briefly discussed in the respective subsections.

B. Analysis Operator Evaluation and Parameter Selection

For evaluating the quality of an analysis operator and for selecting appropriate parameters for GOAL, we choose image denoising as the baseline experiment. The images to be reconstructed have artificially been corrupted by additive white Gaussian noise (AWGN) of varying standard deviation σ_{noise}. This baseline experiment is further used to compare GOAL with other analysis operator learning methods. We 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.

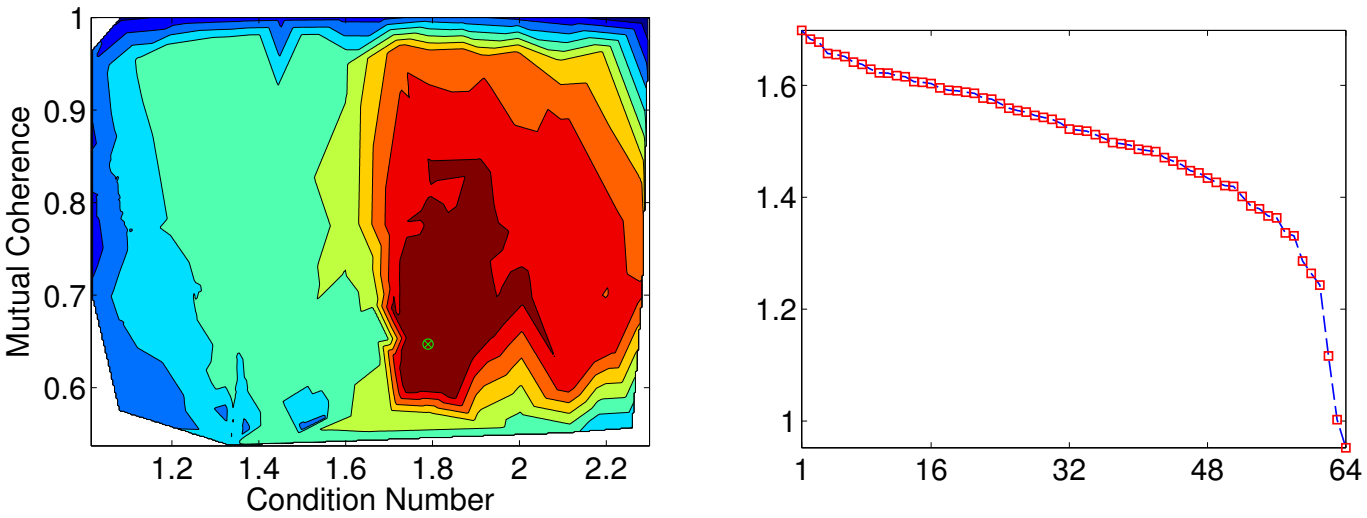
For image denoising, the measurement matrix $\mathcal{A}$ in Equation (49) is the identity matrix. As it is common in the denoising literature, we assume the noise level σ_{noise} to be known and adjust λ accordingly. From our experiments, we found that $\lambda = \frac{\sigma_{noise}}{16}$ is a good choice. We terminate our algorithm after $6 - 30$ iterations depending on the noise level, i.e. the higher the noise level is the more iterations are required. To find an optimal analysis operator, we learned several operators with varying values for μ, κ, and k and fixed all other parameters according to Subsection V-A. Then, we evaluated their performance for the baseline task, which consists of denoising the five test images, each corrupted with the five noise levels as given in Table I. As the final performance measure we use the average *PSNR* of the 25 achieved results. The training set consisted of $M = 200\ 000$ image patches, each normalized to unit Euclidean norm, that have randomly been extracted from the five training images shown in Figure 3. Certainly, these images are not considered within any of the performance evaluations. Each time, we initialized GOAL with a random matrix having normalized rows. Tests with other initializations like an overcomplete DCT did not influence the final operator.

Fig. 3. Five training images used for learning the analysis operator.

The results showed, that our approach clearly benefits from over-completeness. The larger we choose k, the better the operator performs with saturation starting at $k = 2n$. Therefore, we fixed the number of atoms for all further experiments to $k = 2n$. Regarding κ and μ, note that by Lemma 3 and 4 these parameters influence the condition number and the mutual coherence of the learned operator. Towards answering the question of what is a good condition number and mutual coherence for an analysis operator, Figure 4(a) shows the relative denoising performance of 400 operators learned by GOAL in relation to their mutual coherence and condition. We like to mention that according to our experiments, this relation is mostly independent from the degree of over-completeness. It is also interesting to notice that the best learned analysis operator is not a uniformly tight frame. The concrete values, which led to the best performing analysis operator $\mathbf{\Omega}^\star \in \mathbb{R}^{128\times 64}$ in our experiments are $\kappa = 9000$ and $\mu = 0.01$. Its singular values are shown in Figure 4(b) and its atoms are visualized in Figure 5. This operator $\mathbf{\Omega}^\star$ remains unaltered throughout all following experiments in Subsections V-C – V-E.

C. Comparison with Related Approaches

The purpose of this subsection is to rank our approach among other analysis operator learning methods, and to compare its performance with state-of-the-art denoising algorithms. Concretely, we compare the denoising



(a) (b)

Fig. 4. (a) 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). The green dot corresponds to the best performing operator $\mathbf{\Omega}^\star$. (b) Singular values of $\mathbf{\Omega}^\star$

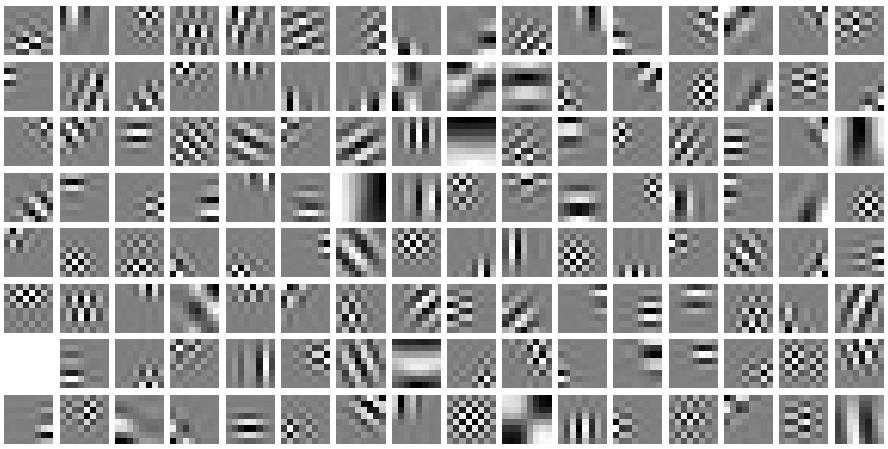


Fig. 5. Learned atoms of the analysis operator $\mathbf{\Omega}^\star$. Each of the 128 atoms is represented as a 8×8 square, where black corresponds to the smallest negative entry, gray is a zero entry, and white corresponds to the largest positive entry.

performance using $\mathbf{\Omega}^\star$ learned by GOAL with total-variation (TV) [36] which is the currently best known analysis operator, with the recently proposed method AOL [25], and with the currently best performing analysis operator FoE [27]. Note that we used the same training set and dimensions for learning the operator by AOL as for GOAL. For FoE we used the same setup as originally suggested by the authors. Concerning the required computation time for learning an analysis operator, for this setting GOAL needs about 10-minutes on an Intel Core i7 3.2 GHz quad-core with 8GB RAM. In contrast, AOL is approximately ten times slower, and FoE is the computationally most expensive method requiring several hours. All three methods are implemented in unoptimized Matlab code.

The achieved results for the five test images and the five noise levels are given in Table I. Our approach achieves the best results among the analysis methods both regarding *PSNR*, and *MSSIM*. For a visual assessment, Figure 6 exemplarily shows some denoising results achieved by the four analysis operators.

To judge the analysis methods' denoising performance globally, we additionally give the results achieved by current state-of-the-art methods BM3D [37] and K-SVD Denoising [12], which are specifically designed for the purpose of image denoising. In most of the cases our method performs slightly better than the K-SVD approach, especially for higher noise levels, and besides of the "barabara" image it is at most ≈ 0.5dB worse than BM3D. This effect is due to the very special structure of the "barbara" image that rarely occurs in natural images, which are smoothed by the learned operator.

(a) GOAL, *PSNR* 30.44dB *MSSIM* 0.831

(b) AOL [25], *PSNR* 27.33dB *MSSIM* 0.720

(c) TV [36], *PSNR* 29.63dB *MSSIM* 0.795

(d) FoE [27], *PSNR* 29.75dB *MSSIM* 0.801

Fig. 6. Images exemplarily showing the typical artifacts created by the four compared analysis operators for image denoising ("man" image, $\sigma_{noise} = 20$). For a better visualization a close up is provided for each image.

D. Image Inpainting

In image inpainting as originally proposed in [2], 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 the restoration of 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 our formulation, when $N - m$ pixels must be inpainted, this leads to a binary $m \times N$ dimensional measurements matrix $\mathcal{A}$, where each row contains exactly one entry equal to

TABLE I

ACHIEVED *PSNR* IN DECIBELS (DB) AND *MSSIM* FOR DENOISING FIVE TEST IMAGES CORRUPTED BY 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 [26], MIDDLE LEFT TV [36], MIDDLE RIGHT FOE [27], BOTTOM LEFT K-SVD DENOISING [12] BOTTOM RIGHT BM3D [37]

	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

TABLE II

RESULTS ACHIEVED FOR INPAINTING THREE 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 [27](MIDDLE), AND METHOD [38] (BOTTOM).

% of missing pixels	**lena**		**boat**		**man**	
	PSNR	*MSSIM*	*PSNR*	*MSSIM*	*PSNR*	*MSSIM*
0.90%	28.57	0.840	25.61	0.743	26.35	0.755
	28.06	0.822	25.14	0.719	26.23	0.747
	27.63	0.804	24.80	0.683	25.56	0.715
0.80%	31.82	0.895	28.55	0.833	28.93	0.847
	31.09	0.880	27.76	0.804	28.51	0.836
	30.95	0.878	27.80	0.804	28.24	0.821
0.50%	37.75	0.956	34.47	0.936	34.12	0.947
	36.70	0.947	33.17	0.907	33.49	0.940
	36.75	0.943	33.77	0.918	33.27	0.934
0.20%	43.53	0.985	41.04	0.982	40.15	0.985
	42.29	0.981	38.45	0.963	39.15	0.982
	40.77	0.965	39.45	0.966	39.06	0.977

one. Its position corresponds to a pixel with known intensity. Hence, $\mathcal{A}$ reflects the available image information. Regarding λ, it can be used in a way that our method simultaneously inpaints missing pixels and denoises the remaining ones.

As an example for image inpainting, we disturbed some ground-truth images artificially by removing $N - m$ pixels randomly distributed over the entire image as exemplary shown in Figure 7(a). In that way, the reconstruction quality can be judged both visually and quantitatively. We assumed the data to be free of noise, and empirically selected $\lambda = 10^{-2}$. In Figure 7, we show exemplary results for reconstructing the "lena" image from 10% of all pixels using GOAL, FoE, and the recently proposed synthesis based method [38]. Table II gives a comparison of further images and further number of missing pixels. It can be seen that our methods performs best independent of the configuration.

E. Single Image Super-Resolution

In single image super-resolution (SR), the goal is to reconstruct a high resolution image $\mathbf{s} \in \mathbb{R}^N$ from an observed low resolution image $\mathbf{y} \in \mathbb{R}^m$. In that, $\mathbf{y}$ is assumed to be a blurred and downsampled version of $\mathbf{s}$. Mathematically, this process can be formulated as $\mathbf{y} = \mathcal{D}\mathcal{B}\mathbf{s} + \mathbf{e}$ where $\mathcal{D} \in \mathbb{R}^{m \times N}$ is a decimation operator and $\mathcal{B} \in \mathbb{R}^{N \times N}$ is a blur operator. Hence, the measurement matrix is given by $\mathcal{A} = \mathcal{D}\mathcal{B}$. In the ideal case, the exact blur kernel is known or an estimate is given. Here, we consider the more realistic case of an unknown blur kernel. Therefore, to

(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 [38], *PSNR* 27.63dB *MSSIM* 0.804.

Fig. 7. Results for reconstructing the "lena" image from 10% of all pixels using $\mathbf{\Omega}^\star$ learny by GOAL, FoE, and [38].

apply our approach for magnifying an image by a factor of d in both vertical and horizontal dimension, we model the blur via a Gaussian kernel of dimension $(2d-1)\times(2d-1)$ and with standard deviation $\sigma_{blur} = \frac{d}{3}$.

For our experiments, we artificially created a low resolution image by downsampling a ground-truth image by a factor of d using bicubic interpolation. Then, we employed bicubic interpolation, FoE, the method from [39], and GOAL to magnify this low resolution image by the same factor d. This upsampled version is then compared with the original image in terms of *PSNR* and *MSSIM*. In Table III, we present the results for upsampling the respective images by $d = 3$. The presented results show that our method outperforms the current state-of-the-art. We want

TABLE III
THE RESULTS IN TERMS OF *PSNR* AND *MSSIM* FOR UPSAMPLING THE SEVEN TEST IMAGES BY A FACTOR OF $d = 3$ USING FIVE DIFFERENT ALGORITHMS GOAL, FOE [27], METHOD [39], 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 [39]	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

to emphasize that the blur kernel used for downsampling is different from the blur kernel used in our upsampling procedure.

Note that many single image super-resolution algorithms rely on clean noise free input data, whereas the general analysis approach as formulated in Equation (49) naturally handles noisy data, and is able to perform simultaneous upsampling and denoising. In Figure 8 we present the result for simultaneously denoising and upsampling a low resolution version of the image "august" by a factor of $d = 3$, which has been corrupted by AWGN with $\sigma_{noise} = 8$. As it can be seen, our method produces the best results both visually and quantitatively, especially regarding the *MSSIM*. Due to high texture this image is hard to upscale even when no noise is present, see the second column of Table III. Results obtained for other images confirm this good performance of GOAL but are not presented here due to space limitation.

VI. CONCLUSION

This paper deals with the topic of learning an analysis operator from example image patches, and how to apply it for solving inverse problems in imaging. To learn the operator, we motivate an ℓ_p-minimization on the set of full-rank matrices with normalized columns. A geometric conjugate gradient method on the oblique manifold is suggested to solve the arising optimization task. Furthermore, we give a partitioning invariant method for employing the local patch based analysis operator such that globally consistent reconstruction results are achieved. For the famous tasks of image denoising, image inpainting, and single image super-resolution, we provide 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, we expect that depending on the application at hand, the performance of the analysis approach can be further increased by learning the particular operator with regard to the specific problem, or employing a specialized training set.

REFERENCES

[1] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[2] M. Bertalmìo, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *ACM SIGGRAPH*, 2000, pp. 417–424.

[3] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[4] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[5] A. Kirsch, *An introduction to the mathematical theory of inverse problems*. Springer, 1991.

[6] M. Elad, M. A. T. Figueiredo, and Y. M., "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[7] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 3, no. 3, pp. 947–968, 2007.

[8] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[9] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[10] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.

[11] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.

[12] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

(a) Original image "august".

(b) Noisy low resolution image.

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

(d) Method [39], *PSNR* 22.07dB *MSSIM* 0.663.

(e) FoE [27], *PSNR* 22.17dB *MSSIM* 0.711.

(f) GOAL, *PSNR* 22.45dB *MSSIM* 0.726.

Fig. 8. Single image super-resolution results of four algorithms on noisy data, for magnifying a low resolution image by a factor of three together with the corresponding *PSNR* and *MSSIM*. The low resolution image has been corrupted by AWGN with $\sigma_{noise} = 8$.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. 1, pp. 19–60, 2010.

[14] 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," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[15] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[17] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.
[18] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.
[19] S. Nam, M. Davies, M. Elad, and R. Gribonval, "Cosparse analysis modeling - uniqueness and algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 5804–5807.
[20] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society Series B*, pp. 91–108, 2005.
[21] 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.
[22] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, 1992.
[23] B. Ophir, M. Elad, N. Bertin, and M. D. Plumbley, "Sequential minimal eigenvalues - an approach to analysis dictionary learning," in *European Signal Processing Conference*, 2011, pp. 1465–1469.
[24] 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*, 2012, pp. 5405–5408.
[25] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations," in *European Signal Processing Conference*, 2011, pp. 1470–1474.
[26] ——, "Noise aware analysis operator learning for approximately cosparse signals," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012, pp. 5409–5412.
[27] S. Roth and M. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.
[28] N. T. Trendafilov, "A continuous-time approach to the oblique procrustes problem," *Behaviormetrika*, vol. 26, no. 2, pp. 167–181, 1999.
[29] M. Kleinsteuber and H. Shen, "Blind source separation with compressively sensed linear mixtures," *IEEE Signal Processing Letters*, vol. 19, no. 2, pp. 107–110, 2012.
[30] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.
[31] Y. Dai and Y. Yuan, "An efficient hybrid conjugate gradient method for unconstrained optimization," *Annals of Operations Research*, vol. 103, no. 1-4, pp. 33–47, 2001.
[32] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.
[33] W. Ring and B. Wirth, "Optimization methods on riemannian manifolds and their application to shape space," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 596–627, 2012.
[34] 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*, 2012, pp. 717–720.
[35] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[36] J. Dahl, P. C. Hansen, S. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numerical Algorithms*, vol. 53, no. 1, pp. 67–92, 2010.
[37] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
[38] M. Zhou, H. Chen, J. Paisley, R. L., L. L., Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.
[39] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.