Learning Efficient Data Representations with Orthogonal Sparse Coding

Henry Schütze, Erhardt Barth, *Member, IEEE,* and Thomas Martinetz, *Senior Member, IEEE*

***Abstract*—We present the learning algorithm Orthogonal Sparse Coding (OSC) to find an orthogonal basis in which a given data set has a maximally sparse representation. OSC is based on stochastic descent by Hebbian-like updates and Gram-Schmidt orthogonalizations, and is motivated by an algorithm that we introduce as the Canonical Approach (CA). First, we evaluate how well OSC can recover a generating basis from synthetic data. We show that, in contrast to competing methods, OSC can recover the generating basis for quite low and, remarkably, unknown sparsity levels. Moreover, on natural image patches and on images of handwritten digits, OSC learns orthogonal bases that attain significantly sparser representations compared to alternative orthogonal transforms. Furthermore, we demonstrate an application of OSC for image compression by showing that the rate-distortion performance can be improved relative to the JPEG standard. Finally, we demonstrate state of the art image denoising performance of OSC dictionaries. Our results demonstrate the potential of OSC for feature extraction, data compression, and image denoising, which is due to two important aspects: (i) the learned bases are adapted to the signal class, and (ii) the sparse approximation problem can be solved efficiently and exactly.**

***Index Terms*—Sparse coding, sparse representation, dictionary learning, blind source separation, sparse component analysis, orthogonal mixture, transform coding, image compression, image denoising.**

I. Introduction

ACCORDING to the efficient-coding hypothesis, early work on sparse coding proposed that the goal of visual coding is to faithfully represent the visual input with minimal neural activity, an idea that goes back to Barlow [1] and is based on earlier work of Ernst Mach and Donald MacKay. This principle of efficient coding has been later extended in several ways and related to the statistics of natural images [2], [3], [4]. Natural images occupy only a small fraction of the entire signal space. As a consequence, they can indeed be encoded sparsely, meaning that they can be represented by a linear combination of rather few elementary signals out of a given collection. Sparsity can also be observed in other classes of natural signals, for instance acoustic signals [5].

The fact that natural images can be sparsely encoded has already been used for image compression. By choosing an adequate analytic transform, e.g. the Discrete Cosine Transform (DCT) or suitable wavelets, many transform coefficients are small and thus need not be encoded [6], [7]. An important progress has been made by going from such predefined transforms to dictionaries that are learned and thereby adapted to particular signal classes [4]. However, such learned dictionaries have not yet been widely used for image compression, one reason being that the resulting non-orthogonal dictionaries are more complex and computationally more demanding than orthogonal ones.

A. Overcomplete vs. Orthogonal Dictionaries

Learning a dictionary for sparse coding can be seen as the task to identify multiple linear subspaces the given training samples are contained in. A dictionary is a collection of unit length vectors, called atoms, such that any K-subset of it spans a K-dimensional subspace of the input space. Suppose a data sample can be represented by some given dictionary via a sparse coefficient vector having K non-zero entries. These coefficients then correspond to coordinates in a particular subspace that is specified by the support of the coefficient vector and the corresponding atoms.

Learning overcomplete dictionaries allows to arbitrarily increase the collection of atoms to a size larger than the dimensionality of the signal space which in turn increases the number of possible subspaces that can be used for encoding. Subspaces composed from an overcomplete dictionary are mutually non-orthogonal which, in general, enables a better adaptation to the training data set and can "represent a wider range of signal phenomena" [8]. However, not to require further conditions on the dictionary is problematic when it comes to calculating optimal sparse data representations, i.e., optimal coefficient vectors including their support. For general overcomplete dictionaries, this problem is provably NP-hard [9]. Approximative greedy algorithms like Basis Pursuit or Orthogonal Matching Pursuit can find optimal coefficients only if the dictionary obeys particular incoherence properties such as, for instance, the restricted isometry property [10]. These incoherence properties require that dictionary atoms are not too similar and can be seen as a relaxation of orthogonality. However, unlike orthogonality, it is difficult to embed such properties as constraints in dictionary learning algorithms.

Orthogonal dictionaries, on the other hand, are mathematically simple and, additionally, maximally incoherent. All possible subspaces are mutually orthogonal with the implication that optimal coefficients can be calculated simply by inner products. Moreover, an orthogonal dictionary can be easily inverted. It serves as synthesis operator and its transpose as analysis operator. Nevertheless, orthogonal bases learned for sparse coding are able to provide efficient encodings as will be shown by our numerical experiments.

The authors are with the Institute for Neuro- and Bioinformatics, University of Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany
Manuscript received XX XX, 2014; revised XX XX, 2014.

B. Learning an Orthogonal Basis for Sparse Coding

In the following, we address the task of learning an orthogonal basis that provides an optimal K-sparse representation of a given training data set. We define an orthogonal basis as a real matrix $\mathbf{U} \in \mathbb{R}^{N \times N}$ obeying $\mathbf{U}^T\mathbf{U} = \mathbf{I}_N$. Given such a basis, any signal $\mathbf{x} \in \mathbb{R}^N$ has a unique representation $\mathbf{x} = \mathbf{U}\mathbf{a}$ by the coefficient vector $\mathbf{a} \in \mathbb{R}^N$. A signal $\mathbf{x}$ is said to be sparse with respect to $\mathbf{U}$, if many coefficients, i.e. many entries of $\mathbf{a}$, are zero or close to zero. Suppose $\mathbf{U}$ is given, then the optimal K-sparse coefficient vector of a single sample $\mathbf{x}$ is found as a solution of

$$\mathbf{a}^*_{\mathbf{U},K}(\mathbf{x}) = \operatorname*{arg\,min}_{\mathbf{a}, \|\mathbf{a}\|_0 \leq K} \|\mathbf{x} - \mathbf{U}\mathbf{a}\|_2^2 \,, \tag{1}$$

where $\|\cdot\|_0 : \mathbb{R}^N \to \{0, ..., N\}$ counts the number of non-zero entries. Due to orthogonality and completeness of $\mathbf{U}$ an optimal solution $\mathbf{a}^*_{\mathbf{U},K}(\mathbf{x})$ can be easily determined by keeping the K largest entries $|a_n|$ of $\mathbf{a} = \mathbf{U}^T\mathbf{x}$ and setting the remaining entries to zero. The sparse approximation problem (1) is a subproblem of our task to find the best orthogonal sparse coding basis $\mathbf{U}^*_K(\mathbf{X})$. Let $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L)$ be a given training data set and $\mathbf{A}^*_{\mathbf{U},K}(\mathbf{X})$ a matrix that column-wise contains a solution to (1) for each $\mathbf{x}_i$. Finding $\mathbf{U}^*_K(\mathbf{X})$ is a nested optimization problem defined by

$$\mathbf{U}^*_K(\mathbf{X}) \;=\; \operatorname*{arg\,min}_{\mathbf{U} \in O(N)} \|\mathbf{X} - \mathbf{U}\mathbf{A}^*_{\mathbf{U},K}(\mathbf{X})\|_{\mathrm{F}}^2 \,, \tag{2}$$

where $O(N) = \{\mathbf{U} \in \mathbb{R}^{N \times N} \mid \mathbf{U}^T\mathbf{U} = \mathbf{I}_N\}$ denotes the orthogonal group.

C. Related Work

1) Analytic Transform Design: The problem of finding a suitable (universal) signal transform in order to efficiently encode image patches and to compress images can be traced back to the Fourier Transform and local versions thereof [11] that finally converged to the first JPEG standard [6] based on the DCT. Pioneering work in the field of wavelet analysis [12] led to a signal decomposition scheme [13], [14] that provides orthogonal multiscale transforms simply by translating and dilating an elementary function (see, e.g. ,[15], [16]). Further findings in the field of wavelet theory set the stage to more efficient image compression codecs such as JPEG2000 [7].

2) Learning Overcomplete Dictionaries: Olshausen and Field introduced the sparse coding principle and proposed a batch learning algorithm to learn a redundant dictionary that minimizes a regularized joint cost function composed of a representation error term and a term that promotes the sparsity of the representation [17]. Meanwhile, a number of algorithms have been proposed that can learn overcomplete dictionaries. Lewicki and Sejnowski proposed a probabilistic approach by gradient ascent on a log posterior with respect to the dictionary [18]. The authors also deduced that learning an overcomplete sparse coding dictionary is a generalization of Independent Component Analysis (ICA) [19], a method designed to invert linear mixtures of statistically independent source signals by maximizing the marginal non-Gaussianity of the data representation [20]. Aharon et al. proposed K-SVD [21], an algorithm that generalizes K-means clustering and iterates two alternating stages. In the first stage, a pursuit algorithm approximates the optimal K-sparse representations of the training set. In the second stage, each dictionary atom as well as associated coefficients are sequentially updated via Singular Value Decomposition (SVD) of a particular error matrix. Labusch et al. proposed Sparse Coding Neural Gas (SCNG) [22], a soft-competitive online learning algorithm based on Neural Gas [23] clustering. SCNG performs stochastic gradient descent and similarly alternates between updating the sparse coefficients and updating the dictionary atoms. Lesage et al. proposed to learn an overcomplete sparse-coding dictionary as unions of orthogonal bases [24], because it relieves the sparse approximation problem. Alternative approaches to learn overcomplete dictionaries for sparse coding can be found in [25], [26], [27], [28], [29] to name a few.

However, all the above learning algorithms do not attempt to enforce orthogonality and thus, in general, learn non-orthogonal overcomplete dictionaries which enables, for instance, to capture invariance [8].

3) Learning Orthogonal Dictionaries: Nevertheless, a few authors proposed to learn orthogonal dictionaries for sparse coding. Coifman et al. proposed the Wavelet Packet Transform [30], which is an early attempt to enhance orthogonal transforms with a certain degree of adaptivity to the represented signal. For a given signal, it allows to select a basis from a large collection of dyadic time frequency atoms derived from a specific pair of mother wavelet and scaling function.

Mishali and Eldar formulated the orthogonal sparse coding problem as a blind source separation problem [31]. Given a set of observations, the problem is to invert the linear mixture of unknown K-sparse sources by an unknown orthogonal mixing matrix. They proposed a method with two succeeding stages. The first stage estimates the entire support pattern of the coefficient matrix, i.e., all locations of non-zero coefficients. The second stage iteratively adapts (*i*) the non-zero coefficients and (*ii*) the orthogonal mixing matrix via SVD by using the estimated support pattern. However, in [31] only low-dimensional synthetic data sets were investigated, and merely two rather high sparsity levels ($K \in \{2, 3\}$) were considered. With lower sparsity levels the recovery performance of the support pattern rapidly decreases and impairs the estimation of the mixing matrix substantially.

Another issue with their first stage is the strict requirement that the given data need to have an exactly K-sparse representation (with exact zeros) in the generating basis, which does not even tolerate small amplitude noise and is therefore not applicable to real word data. For this reason, we can evaluate this approach only with noiseless synthetic data (see Subsection III-A).

Dobigeon and Tourneret proposed the Bayesian framework BOCA for a similar source separation formulation that is, however, designed for an undercomplete orthogonal mixing matrix [32]. BOCA relies on knowing specific prior distributions for the unknown model parameters. The proposed approach models the sparse sources as Bernoulli-Gaussian processes and uses a uniform prior distribution on the Stiefel manifold for the mixing matrix. A comparison of OSC or CA to BOCA is out of the scope of this paper, because we here address the complete orthogonal dictionary learning task.

Gribonval and Schnass considered the problem of learning an orthogonal sparse-coding basis by minimizing the ℓ_1-norm of the coefficient matrix with respect to dictionary and coefficient matrix such that their product synthesizes the training data [33]. Their main results are identifiability conditions that guarantee local convergence to the generating dictionary by the ℓ_1 minimization approach. They showed that a particular sparse model satisfies these conditions with high probability provided that enough samples are given. However, an explicit algorithm is not proposed and the convergence relies on a good initialization.

In [34], Bao et al. proposed a batch algorithm to learn an orthogonal sparse coding basis from patches of an image. Their method is similar to the method proposed in [35], but learns the orthogonal dictionary within the orthogonal complement of a predefined (possibly empty) set of orthogonal dictionary atoms which are not updated during learning. Their method is related to what we will introduce as Canonical Approach (CA) in Subsection II-A, as it computes closed form solutions of the two underlying subproblems. However, [34] addresses an unconstrained sparse model in Lagrangian form with a weighting factor λ that controls the trade-off between reconstruction error and sparsity. The dictionary update stage of [34] and CA is the same, and has been used previously, e.g., in [24] and [31]. The key difference between [34] and CA is the sparse coding stage, which is in one case realized by a global hard thresholding operator and in the other case realized by retaining the K most relevant coefficients for each sample. Since we are focused on the constrained K-sparse model, we have not included the threshold based method proposed in [34] in our experiments. In [34], the main learning algorithm has also been extended to learn orthogonal dictionaries on corrupted image patches with the objective to better solve image restoration problems. Furthermore, [36] has adopted the method proposed in [34] to solve a compressive MRI imaging reconstruction problem.

To the best of our knowledge there are no further methods which address problem (2) or can serve as a reference for the experiments described in Section III.

D. Structure and Contribution of the Paper

In Section II, we first introduce the batch algorithm CA (which is related to prior work) and subsequently, as our main contribution, the online algorithm OSC to learn an orthogonal sparse coding basis. For OSC, we provide a pseudo code and estimate its computational complexity. We provide a theorem which states that an OSC update reduces the cost for the processed sample. This assures stochastic descent on the cost function for the entire training data set and, hence, cost reduction at least to a local minimum. We demonstrate it in a simulation.

In Section III, we compare the performance of OSC and alternative methods in terms of how well they recover a generating orthogonal basis from sparse synthetic data in noiseless and noisy settings. We then apply OSC to real world data, (*i*) natural image patches, and (*ii*) images of hand-written digits and visualize the learned orthogonal dictionary atoms. Signal classes different from natural image patches were marginally considered by prior work on orthogonal dictionary learning. Furthermore, the optimal K-term approximation performance is analyzed on corresponding test data sets for different parameter values K and compared to alternative orthogonal transforms (analytic as well as learned).

In Section IV, we demonstrate the applicability of OSC to image compression. We compare the rate distortion for images compressed with JPEG, JPEG2000, and a modified JPEG codec for which the 8×8 DCT has been replaced by an 8×8 OSC basis. Finally, we use learned orthogonal sparse coding bases for image denoising.

II. The OSC Algorithm

A. The Canonical Approach and Motivation for OSC

First of all, we propose what we call the Canonical Approach (CA) to learn an orthogonal sparse coding basis. CA is related to orthogonal dictionary learning approaches proposed in [34] and [35]. CA can be seen as their natural modification to match the problem formulation (2). Similar to traditional sparse coding algorithms, this approach iteratively alternates between (*i*) the determination of the optimal (column-wise) K-sparse coefficient matrix $\mathbf{A}^*_{\mathbf{U},K}(\mathbf{X})$ for the current, temporary fixed basis $\mathbf{U}$, and (*ii*) the determination of the optimal orthogonal basis $\mathbf{U}^*_{\mathbf{A}}(\mathbf{X})$ for the current, temporary fixed coefficient matrix $\mathbf{A}$. For our orthogonal setting, both subproblems have closed form solutions.

For a temporary fixed $\mathbf{U}$ and a given training sample $\mathbf{x}$, the minimizer of the sparse approximation problem (1) is

$$\mathbf{a}^*_{\mathbf{U},K}(\mathbf{x}) = \mathbf{D}_K(\mathbf{x},\mathbf{U})\mathbf{U}^T\mathbf{x} \,, \tag{3}$$

where $\mathbf{D}_K(\mathbf{x},\mathbf{U})$ is a diagonal matrix having K entries equal to 1 and otherwise entries equal to 0. $\mathbf{D}_K(\mathbf{x},\mathbf{U})$ selects the K largest projections $|\mathbf{u}_n^T\mathbf{x}|$, i.e., the K largest entries $|a_n|$ of $\mathbf{a} = \mathbf{U}^T\mathbf{x}$. This step, applied to each training sample $\mathbf{x}_i$, provides an optimal coefficient matrix $\mathbf{A} = \mathbf{A}^*_{\mathbf{U},K}(\mathbf{X})$ with K-sparse columns.

Suppose such a coefficient matrix $\mathbf{A}$ (representing $\mathbf{X}$) is now temporary fixed, then the optimal orthogonal basis is given by

$$\mathbf{U}^*_{\mathbf{A}}(\mathbf{X}) = \mathbf{V}\mathbf{W}^T \,, \tag{4}$$

where $\mathbf{V}$ and $\mathbf{W}$ are the outer matrices of the SVD of $\mathbf{X}\mathbf{A}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{W}^T$ [37]. Please note that variants of (4) are also used, for instance, in [24], [31], [34], and [35]. Setting $\mathbf{U} = \mathbf{U}^*_{\mathbf{A}}(\mathbf{X})$ as given by (4) and iterating all these steps yields the CA algorithm.

We tested CA on synthetic K-sparse data for which the ground truth is known (see Subsection III-A). It turns out that, given the correct sparsity level K, CA is able to accurately recover the generating basis as far as K is rather small (high sparsity). However, In the case of larger K (lower sparsity) CA gets stuck in local minima and does not converge. This finding motivates the stochastic descent approach that will be presented in the following.

B. OSC Idea and Pseudo code

Orthogonal Sparse Coding (OSC[1]) can be seen as a generalization of Principal Component Analysis (PCA). With (3) as the solution to the sparse approximation problem (1), the cost function which has to be minimized in (2) can be reformulated as

$$E_{\mathbf{X},K}(\mathbf{U}) = -\sum_{i=1}^{L} \mathbf{x}_i^T \mathbf{U} \mathbf{D}_K(\mathbf{x}_i, \mathbf{U}) \mathbf{U}^T \mathbf{x}_i \, . \tag{5}$$

We denote the minimizer of (5) by $\mathbf{U}_K^*(\mathbf{X})$. Note that in (5) we have merged both subproblems to a certain extent, a procedure that has been proven to be also crucial for good convergence of online PCA algorithms. If $\mathbf{D}_K$ were a fixed matrix and not dependent on $\mathbf{x}_i$ and $\mathbf{U}$, e.g. the first K values in the diagonal would be constantly set to one, then PCA would provide a minimizer of (5). In contrast to PCA, however, OSC selects for each $\mathbf{x}_i$ an individual $\mathbf{D}_K$ depending on $\mathbf{U}$.

Since the minimizer of (5) cannot be obtained in closed form, we perform stochastic descent analogous to online PCA algorithms. When adapting $\mathbf{U}$, the algorithm switches between gradient descents (Hebbian-like updates) and Gram-Schmidt orthogonalization steps.

Algorithm 1 Orthogonal Sparse Coding (OSC)

```
Input: training data set $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_L) \in \mathbb{R}^{N \times L}$
       number of learning steps $t_{\max}$
       initial and final learning rate $\varepsilon_{\text{init}} \geq \varepsilon_{\text{final}}$
       sparsity level $K$ (default: $K = N$)
Output: orthogonal basis $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_N) \in \mathbb{R}^{N \times N}$
 1: Start with random orthogonal basis $\mathbf{U}$
 2: for all $t = 0, ..., t_{\max}$ do
 3:   $\varepsilon_t \leftarrow \varepsilon_{\text{init}} \left(\varepsilon_{\text{final}} / \varepsilon_{\text{init}}\right)^{t/t_{\max}}$
 4:   select sample $\mathbf{x}$ from $\mathbf{X}$ randomly and set $\mathbf{x}_{\text{res}} \leftarrow \mathbf{x}$
 5:   determine a sequence $n_1, ..., n_N$, such that $\left(\mathbf{u}_{n_1}^T \mathbf{x}\right)^2 \geq ... \geq \left(\mathbf{u}_{n_N}^T \mathbf{x}\right)^2$ holds
 6:   for all $k = 1, ..., N$ do
 7:     for all $l = 1, ..., (k-1)$ do
 8:       $\mathbf{u}_{n_k} \leftarrow \mathbf{u}_{n_k} - \left(\mathbf{u}_{n_k}^T \mathbf{u}_{n_l}\right) \mathbf{u}_{n_l}$
 9:     end for
10:     if $k \leq K$ then
11:       $y \leftarrow \mathbf{u}_{n_k}^T \mathbf{x}_{\text{res}}$
12:       $\mathbf{u}_{n_k} \leftarrow \mathbf{u}_{n_k} + \varepsilon_t \cdot y \cdot \mathbf{x}_{\text{res}}$
13:     end if
14:     normalize $\mathbf{u}_{n_k}$ to unit length
15:     $\mathbf{x}_{\text{res}} \leftarrow \mathbf{x}_{\text{res}} - \left(\mathbf{u}_{n_k}^T \mathbf{x}_{\text{res}}\right) \mathbf{u}_{n_k}$
16:   end for
17: end for
```

The OSC algorithm starts with a randomly initialized basis $\mathbf{U}$. For each learning step an entire update of $\mathbf{U}$ takes place. At the beginning of each learning step t, the current learning rate ε_t is set and a training sample $\mathbf{x}$ is randomly selected from the given training data set $\mathbf{X}$. First, OSC determines an index sequence $n_1, ..., n_N$ that sorts the squared coefficients of $\mathbf{x}$ in the current basis $\mathbf{U}$, s.t. $(\mathbf{u}_{n_1}^T \mathbf{x})^2 \geq (\mathbf{u}_{n_2}^T \mathbf{x})^2 \geq ... \geq (\mathbf{u}_{n_N}^T \mathbf{x})^2$. Note that the contribution of the selected sample $\mathbf{x}$ to the cost function (5) is given by

$$\begin{aligned} E_K(\mathbf{x}, \mathbf{U}) &= -\mathbf{x}^T \mathbf{U} \mathbf{D}_K(\mathbf{x}, \mathbf{U}) \mathbf{U}^T \mathbf{x} \\ &= -\sum_{k=1}^{K} (\mathbf{u}_{n_k}^T \mathbf{x})^2 \, . \end{aligned} \tag{6}$$

As a consequence, costs are reduced if the sum of the K largest squared coefficients is increased. Loosely speaking, this can be interpreted as concentrating the energy distribution of the sparse coefficients of $\mathbf{x}$. The determined index sequence defines the order of basis vector updates, starting with the basis vector which contributes most to (6). Before a basis vector $\mathbf{u}_{n_k}$ is updated, it is orthogonalized with respect to $\operatorname{span}(\{\mathbf{u}_{n_1}, ..., \mathbf{u}_{n_{k-1}}\})$, the span of basis vectors that were already updated during the current learning step. Then, the orthogonalized basis vector $\mathbf{u}_{n_k}$ is updated by gradient descent depending on the residual vector $\mathbf{x}_{\text{res}}$ (the original training sample $\mathbf{x}$ likewise orthogonalized with respect to $\operatorname{span}(\{\mathbf{u}_{n_1}, ..., \mathbf{u}_{n_{k-1}}\})$) such that its contribution $-(\mathbf{u}_{n_k}^T \mathbf{x}_{\text{res}})^2$ in (6) decreases. This leads to the Hebbian-like update rule

$$y \leftarrow \mathbf{u}_{n_k}^T \mathbf{x}_{\text{res}} \tag{7}$$

$$\Delta \mathbf{u}_{n_k} \leftarrow \varepsilon_t \cdot y \cdot \mathbf{x}_{\text{res}} \, . \tag{8}$$

The updated basis vector $\mathbf{u}_{n_k}$ is normalized to unit length. Subsequently, $\mathbf{x}_{\text{res}}$ is orthogonalized with respect to $\mathbf{u}_{n_k}$, thus becoming the residual vector for the update of $\mathbf{u}_{n_{k+1}}$.

How many basis vectors are updated with the learning rule depends on the sparsity parameter K. Please note, however, that due to the required orthogonalization, all basis vectors are modified even if only K were updated by (8). A learning step is completed, when the last basis vector $\mathbf{u}_{n_N}$ has been normalized. With this scheme of iterative Gram-Schmidt orthogonalization, $\mathbf{U}$ is most certainly an orthogonal basis. The learning rate ε_t cools down with the number of conducted learning steps. *Algorithm 1* lists OSC in pseudo code.

C. Computational Complexity of OSC

For a single OSC learning step, drawing a training sample $\mathbf{x}$, setting residual vector $\mathbf{x}_{\text{res}}$ (line 4) and sorting the coefficients (line 5) has complexity $\mathcal{O}(N^2) + \mathcal{O}(N \log N)$. The loop in lines 6-16 iterates $\mathcal{O}(N)$ times over all basis vectors $\mathbf{u}_{n_k}$. The Gram-Schmidt steps for each $\mathbf{u}_{n_k}$ (lines 7-9) have at most a complexity of $\mathcal{O}(N^2)$. A single Hebbian-like update (lines 11-12) of a $\mathbf{u}_{n_k}$ has a complexity of $\mathcal{O}(N)$. The length normalization of $\mathbf{u}_{n_k}$ (line 14) and the update of $\mathbf{x}_{\text{res}}$ (line 15) take likewise $\mathcal{O}(N)$. Putting all together and taking the outer loop over the learning steps into account gives a computational complexity of

$$\mathcal{O}\left(t_{\max}\left(N \log N + KN + N^2 + N^3\right)\right) \, , \tag{9}$$

which is bounded by $\mathcal{O}(t_{\max} N^3)$.

[1] A first sketch of the OSC algorithm and preliminary results have been presented at the workshop *New Challenges in Neural Computation 2013* [38].

D. Convergence Properties of OSC

OSC is an online learning algorithm that updates the sparse coding basis $\mathbf{U}$ for each presented training sample $\mathbf{x}$. In the following we provide a theorem which assures that a learning step by OSC increases the sparsity of the representation of the given $\mathbf{x}$, i.e. decreases the cost contribution of this sample to the overall cost function. Hence, OSC performs stochastic descent on the cost function which has to be minimized. For small enough step sizes ε this can be proven, but seems to be valid also for large ε according to our numerical experiments.

Theorem 1: Given an orthogonal basis $\mathbf{U}$. If $\varepsilon > 0$ is small enough, applying an OSC learning step to an arbitrary non-zero $\mathbf{x}$ yields a new orthogonal basis $\mathbf{U}'$ such that for the sequences $(\mathbf{u}^T_{n_1}\mathbf{x})^2 \ge (\mathbf{u}^T_{n_2}\mathbf{x})^2 \ge ... \ge (\mathbf{u}^T_{n_N}\mathbf{x})^2$ and $(\mathbf{u}'^T_{n_1}\mathbf{x})^2 \ge (\mathbf{u}'^T_{n_2}\mathbf{x})^2 \ge ... \ge (\mathbf{u}'^T_{n_N}\mathbf{x})^2$ the ordering

$$\frac{(\mathbf{u}'^T_{n_{k+1}}\mathbf{x})^2}{(\mathbf{u}'^T_{n_k}\mathbf{x})^2} \le \frac{(\mathbf{u}^T_{n_{k+1}}\mathbf{x})^2}{(\mathbf{u}^T_{n_k}\mathbf{x})^2} \qquad (10)$$

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

Theorem 1 states that by an OSC learning step the magnitude of a coefficient decreases relative to its predecessor in the sequence of sorted coefficients. This means, that after an OSC learning step the squared coefficients obey a stronger decay. Figure 1 illustrates, from an experiment with natural image patches, the sorted squared coefficients before and after an OSC learning step. Clearly, after the learning step more energy is distributed over less coefficients. However, the total amount of energy is preserved, because $\mathbf{U}$ is an orthogonal basis. Hence, the sparsity of the encoding of the given $\mathbf{x}$ is increased.

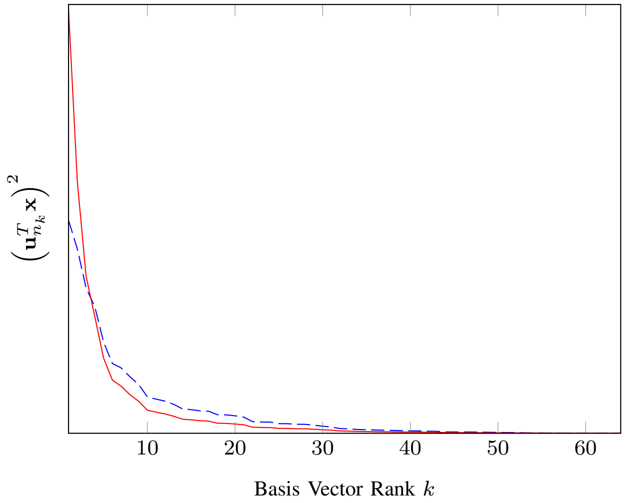


Figure 1. Sorted squared coefficients of an 8×8 natural image patch in basis $\mathbf{U}$, before (dashed, blue line) and after (solid, red line) an OSC learning step.

From Theorem 1 follows directly

Corollary 1: Given an orthogonal basis $\mathbf{U}$. Applying an OSC learning step with an arbitrary $\mathbf{x}$ leads to an $\mathbf{U}'$ such that for each $K = 1, ..., N$

$$-\sum_{k=1}^{K}(\mathbf{u}'^T_{n_k}\mathbf{x})^2 \le -\sum_{k=1}^{K}(\mathbf{u}^T_{n_k}\mathbf{x})^2 \Leftrightarrow$$
$$\|\mathbf{x}-\mathbf{U}'\mathbf{D}_K(\mathbf{x},\mathbf{U}')\mathbf{U}'^T\mathbf{x}\|_2^2 \le \|\mathbf{x}-\mathbf{U}\mathbf{D}_K(\mathbf{x},\mathbf{U})\mathbf{U}^T\mathbf{x}\|_2^2 .$$

This means that an OSC learning step reduces the costs (6) with respect to the presented training sample. It might not be obvious that OSC minimizes cost function (5) with respect to the entire training data set. However, a stochastic descent of (5) is due to the pattern-by-pattern scheme of OSC similar to a stochastic gradient descent. From the experiment with natural image patches described in Subsection III-B, Figure 2 illustrates the temporal course of the total costs (5) for the OSC basis $\mathbf{U}$ over a full learning phase. It can be seen that OSC reduces the total costs (5).

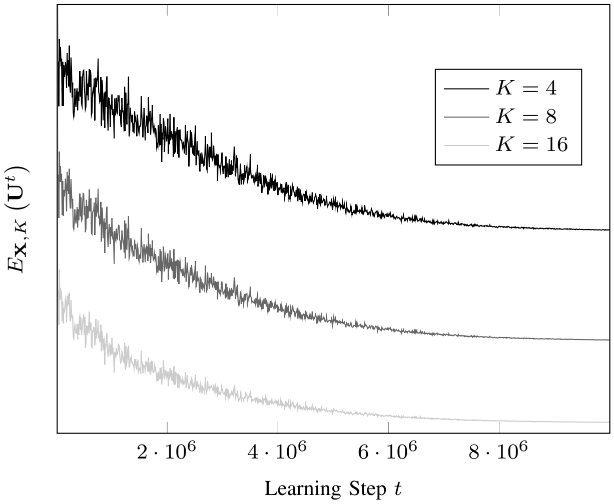


Figure 2. Training error evaluated by cost function (5) during a full learning phase of OSC on natural image patches (see experiment in Subsection III-B).

E. K-OSC and "full" OSC

In general, $E_{\mathbf{X},K}(\mathbf{U})$ has different minima for different K and OSC will provide different solutions $\mathbf{U}^*_K$. However, there are situations where it is suitable to choose user parameter K of the OSC algorithm as $K = N$ and take the output of this "full" OSC as a kind of universal solution $\mathbf{U}^*$ which minimizes (5) for many different K. In these cases it is not necessary to learn an individual basis for each individual K.

Such a situation is given, for example, if $\mathbf{U}^*_K$ does not change although K is further increased. Starting with cost function (5) and some rearrangements we can write

$$\begin{aligned} E_{\mathbf{X},K+1}(\mathbf{U}) &= -\sum_{n=1}^{N}\sum_{k=1}^{K+1}\sum_{\mathbf{x}\in S^n_k(\mathbf{U})}(\mathbf{u}^T_n\mathbf{x})^2 \\ &= E_{\mathbf{X},K}(\mathbf{U}) - \sum_{n=1}^{N}\sum_{\mathbf{x}\in S^n_{K+1}(\mathbf{U})}(\mathbf{u}^T_n\mathbf{x})^2 (11) \end{aligned}$$

$S^n_k(\mathbf{U})$ is the set of those $\mathbf{x}$ for which $(\mathbf{u}^T_n\mathbf{x})^2$ is the k-th largest term in the sequence $(\mathbf{u}^T_{n_1}\mathbf{x})^2 \ge (\mathbf{u}^T_{n_2}\mathbf{x})^2 \ge ... \ge (\mathbf{u}^T_{n_N}\mathbf{x})^2$. Since $\mathbf{U}^*_K$ is a minimum of the first term in (11), a sufficient condition for $\mathbf{U}^*_K$ to remain a minimum also of $E_{\mathbf{X},K+1}$ is $\mathbf{U}^*_K$ to be a minimum also of the second term in (11). This is the case if the gradient of the second term with respect to each $\mathbf{u}_n$ vanishes at $\mathbf{U} = \mathbf{U}^*_K$, i.e.,

$$\sum_{\mathbf{x}\in S^n_{K+1}(\mathbf{U}^*_K)} \mathbf{u}^{*T}_n\mathbf{x} = 0 \qquad \forall\, n \in \{1, ..., N\}\ , \qquad (12)$$

with $\mathbf{u}_n^*$ being the n-th column vector of $\mathbf{U}_K^*$.

Condition (12) holds, for example, if the $\mathbf{x}$ are K-sparse in $\mathbf{U}_K^*$, since each element in the sum of the gradient is zero. This is still valid if the signals are K-sparse in $\mathbf{U}_K^*$ except for additive isotropic noise, since for symmetry reasons the contributions in the sum cancel out. Hence, if we have K-sparse signals without or with isotropic noise, working with the ”full” version of OSC, i.e. with $\mathbf{U}^*$, gives us the correct solution, even if we do not know the given sparsity level K.

Another scenario appropriate for using $\mathbf{U}^*$ is given if each signal is a linear superposition of orthogonal atoms, which are drawn randomly and independently (without replacement) from the dictionary $\mathbf{V} = (\mathbf{v}_1, ..., \mathbf{v}_N)$. The coefficients are drawn independently from the probability distribution P_1 for the first chosen atom, from P_2 for the second chosen atom, etc. Because of the independencies we have $\mathbf{V} = \mathbf{U}_1^* = \mathbf{U}_2^* = ... = \mathbf{U}_{N-1}^* = \mathbf{U}^*$. Again, it is convenient to use the ”full” OSC ($K = N$), instead of having to decide how to choose the parameter K.

There are, of course, further scenarios where $\mathbf{U}^*$ is either the exact or at least a good approximative solution for many different K values. Indeed, in our experiments we have consistently observed this convenient property (see Subsections III-A, III-B, and III-C). Obviously, an algorithm that does not rely on knowing the sparsity level K is more practical.

III. Results

In the following, we label results obtained by the “full” OSC variant ($K = N$) simply as “OSC” and use otherwise the annotation “K-OSC”.

A. Results on Synthetic Data

We investigated how well K-SVD, the approach of Mishali et al. ([31]), CA, K-OSC, and OSC can recover a generating orthogonal basis from K-sparse synthetic data. Note, that K-SVD is an algorithm for finding arbitrary, non-orthogonal sparse coding dictionaries and does therefore not benefit from the orthogonality of an underlying dictionary. Nevertheless, orthogonality is a good-natured scenario for K-SVD, because the mutual coherence is minimal.

To generate a synthetic data set, we fixed signal dimensionality to $N = 256$ and sample size to $L = 1000$, whereas the sparsity level $K \in \{2, 6, ..., 58, 62\}$ was gradually varied. Each data sample was generated as 16×16 patch being K-sparse in the non-standard 2D Haar wavelet basis. The support pattern of each sample, i.e., the K locations of non-zero coefficients in the Haar wavelet domain were uniformly selected at random. Then, the non-zero coefficients were drawn randomly from a standard Gaussian distribution. In order to investigate deviations of recovery rates over multiple runs, we created 10 data sets for each sparsity level.

We applied the five algorithms to the generated data sets and provided all but OSC with the known K as user parameter. Each method conducted 100 learning epochs. To measure recovery performance for each run we followed a procedure similar to the one used in [21]. We first determined the “best matching pairs” between estimated and generating basis vectors. This was done by first sorting the overlaps[2] of all N^2 possible basis vector pairs in decreasing order. Subsequently, the “best matching pairs” were assigned according to that sequence with the requirement that estimated basis vectors and generating basis vectors obey a one-to-one assignment. We considered a generating basis vector as recovered, if it has an overlap of at least 0.8 to its matched estimated version. The recovery rate of a full basis is expressed as the ratio of recovered basis vectors.

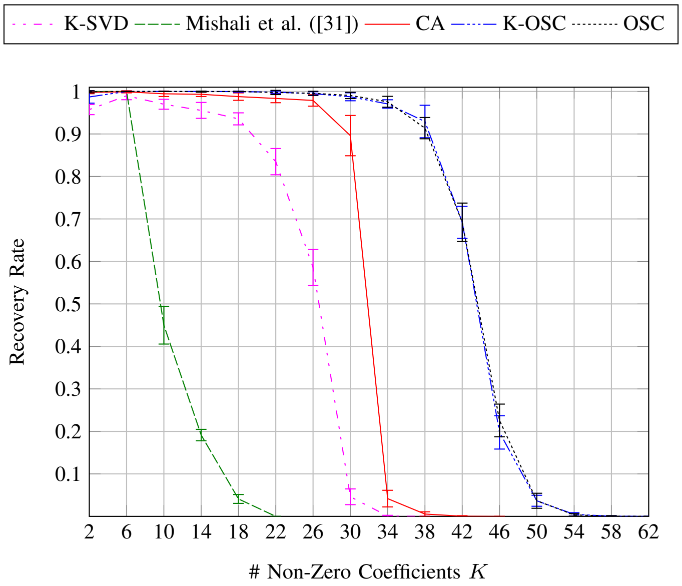
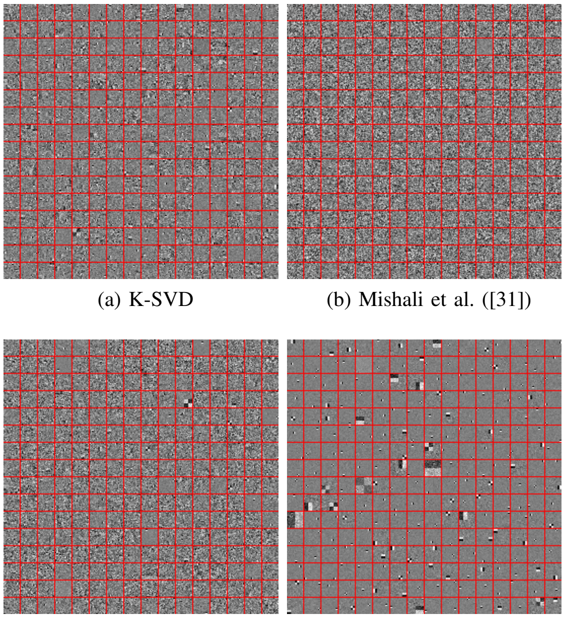


Figure 3. Mean recovery rates and standard deviations for synthetic data sets ($L = 1000$ patches of size 16×16 being K-sparse in the 2D Haar wavelet basis).

Figure 3 illustrates mean recovery rate and standard deviations of the synthetic experiment over 10 runs for each value of K. At the basis identification task, the investigated methods reveal individual limits in terms of the sparsity level. The approach of Mishali et al. ([31]) performs worst losing its perfect recovery performance at $K > 6$. The second worst performing method is K-SVD which does not obtain perfect recovery but at least achieves recovery rates > 0.9 for $K \leq 18$. CA yields recovery rates > 0.97 for $K \leq 26$ which decrease significantly at $K > 30$. K-OSC and OSC are equally best performing at the identification task. The OSC and K-OSC recovery performances are > 0.97 for $K \leq 34$ and decrease below 0.9 at $K > 38$. We would like to emphasize that in contrast to all other methods the true sparsity levels were not provided to OSC.

We repeated the synthetic experiment in the same way, but added 5 dB Gaussian noise to each data set (see Figure 5). Note that the approach of Mishali et al. ([31]) could not be used for our comparison, because due to the additive noise the support recovery stage completely fails and returns maximally non-sparse support patterns. With 5 dB additive Gaussian noise the recovery performance is degraded for all the remaining methods, i.e., recovery rates decrease faster as K increases.

[2] The overlap between two unit length vectors $\mathbf{v}$ and $\mathbf{w}$ is defined as $|\mathbf{v}^T\mathbf{w}|$.


(a) K-SVD (b) Mishali et al. ([31])

(c) CA (d) OSC

Figure 4. Bases recovered from a synthetic data set ($L = 1000$ patches of size 16×16 being $K = 34$-sparse ($\approx 13.3\%$ non-zero coefficients) in the 2D Haar wavelet basis). For this rather large K, OSC is able, in contrast to K-SVD, the approach of Mishali et al. ([31]), and CA, to extract the underlying basis (see also Figure 3). For display purposes, the entries of each basis patch (except the DC component) are shifted to have zero mean and are scaled to unit supremum norm.

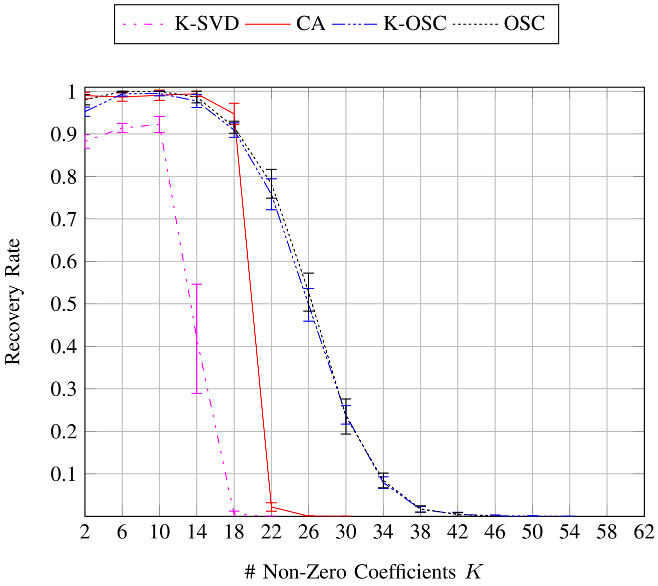


Figure 5. Mean recovery rates and standard deviations for synthetic data sets ($L = 1000$ patches of size 16×16 being K-sparse in the 2D Haar wavelet basis) with 5 dB additive Gaussian noise.

However, with increasing K the recovery performance of OSC and K-OSC decays the slowest.

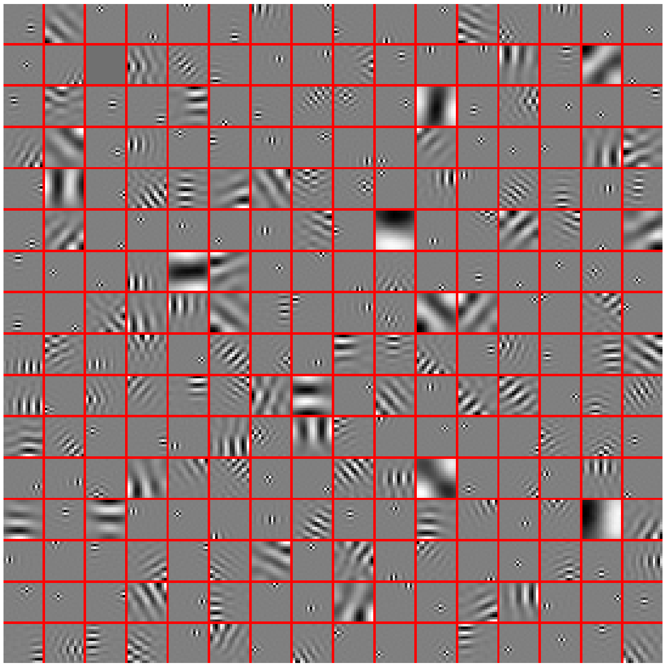


Figure 6. Orthogonal sparse coding basis for natural image patches, obtained with OSC. The set of basis functions resembles a wavelet decomposition.

B. Results on Natural Image Patches

We let OSC learn orthogonal bases to sparsely encode natural image patches. We extracted image patches from set one of the Nature Scene Collection [39], i.e., from images of nature scenes containing no man made objects or people. The uncompressed RGB images have a resolution of 2844×4284 pixels. The color channels are linearly scaled, each with a depth of 16 bits per pixel (bpp). To each color channel the logarithm to the base of 2 and a subsequent scaling into the double precision floating point range $[0, 1]$ was applied. Subsequently, the color images were converted to grayscale images. From the entire set of 308 images, we randomly selected 250 images for learning. From each image, we extracted 400 patches of size 16×16 pixels at random positions. These 10^5 image patches were used for training. In the same manner, a test data set with 23200 patches was generated from the remaining 58 images. Data preprocessing comprised the sample-wise subtraction of the DC component and of the mean vector.

We conducted $t_{\max} = 10^7$ learning steps with OSC. The initial and final learning rates were manually set to $\varepsilon_{\text{init}} = 10$ and $\varepsilon_{\text{final}} = 10^{-2}$ without an extensive parameter validation.

Figure 6 illustrates the orthogonal basis learned by OSC on natural image patches. The OSC basis resembles a wavelet decomposition. On different scales, the basis patches show selectivity for inputs with particular frequencies, orientations, and spatial localizations.

For the test data set, Figure 7 illustrates the average optimal K-term approximation performance of different bases measured by the signal-to-noise-ratio (SNR). On the one hand, we compare OSC to the predefined orthogonal transform bases of 2D DCT and non-standard 2D Haar wavelets which are known to provide decent sparse representations. On the other hand, we compare OSC to bases learned by PCA[3], CA, and K-SVD[4].

[3] With PCA, K-term approximations are derived from the K first PCs

[4] With K-SVD, K-term approximations are obtained by Batch OMP [40]

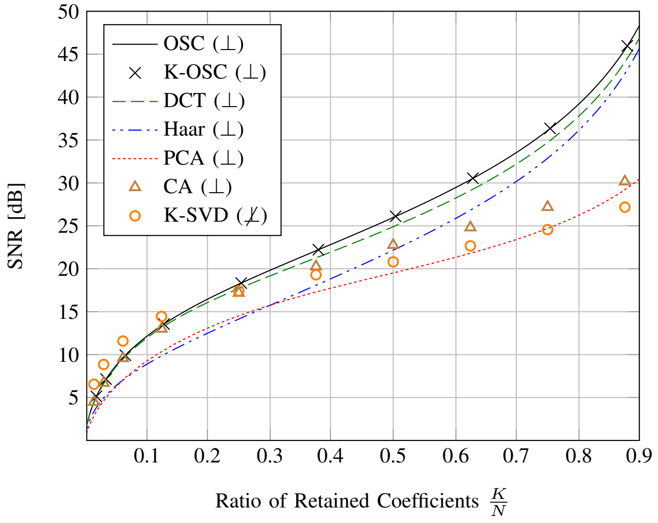


Figure 7. Average optimal K-term approximation performance on the test data set containing natural image patches.

Note that by K-SVD non-orthogonal complete dictionaries were learned. As for OSC and K-OSC we let CA and K-SVD learn for 100 epochs. Note that a comparison to the approach of Mishali et al. ([31]) was not possible, because its support recovery stage requires the existence of a strictly K-sparse representation and returned, in our attempts, maximally non-sparse support patterns for the training data.

For $K \leq 32$, K-SVD performs slightly better than OSC, K-OSC, DCT, and CA which have nearly equal approximation performances. For larger K, OSC and K-OSC equally perform best at the task to sparsely encode the test data. Both achieve a slightly better encoding performance compared to the DCT whereas the superiority compared to Haar wavelets, PCA, CA, and K-SVD is more striking. It is remarkable that the single OSC basis (see Figure 6) yields the same K-term approximation performance as the K-OSC bases which were individually learned for each tested sparsity level. It appears that this universality property of OSC is not only limited to particular artificial data settings, but also holds for natural image data.

We also investigated the stability of OSC by fixing the training set as well as the learning parameters and applied the algorithm repeatedly with different initial random bases and different random training sequences. On the fixed test data set, we computed the average optimal K-term approximation performance for each OSC basis learned in our 20 runs. We found, that the standard deviation of the K-term approximation performance was less than 0.081 dB for any $K \leq 254$, and even less than 0.017 dB for any $K \leq 230$ (i.e. $K/N \leq 0.9$).

C. Results on Images of Handwritten Digits

The same experiment was conducted with the MNIST data set [41], i.e., with images of handwritten digits. The training data set comprises $6 \cdot 10^4$ and the test data set 10^4 grayscale images of size 28×28 with a gray-level depth of 8 bit. We transformed the images to the double precision floating point range $[0, 1]$ and resized[5] them to 16×16 pixels using bicubic interpolation. From each sample its DC component and subsequently the mean vector of the training data set was subtracted. We conducted $t_{\max} = 3.6 \cdot 10^7$ learning steps with OSC. The initial and final learning rates were set to $\varepsilon_{\text{init}} = 2.8$ and $\varepsilon_{\text{final}} = 2.8 \cdot 10^{-3}$, respectively.

Figure 8 depicts the learned OSC basis patches. Note that many OSC basis patches learned on the MNIST training set show sensitivity for particular digits or digit combinations. Furthermore, some basis patches show sensitivity to localized grating patterns with different orientations. Note, that some of these grating patterns have curved shapes.

In Figure 9, we again compare the average optimal K-term approximation performance of OSC, K-OSC, 2D DCT, 2D Haar wavelets, PCA, CA, and K-SVD, this time on the MNIST test data set. K-SVD has a slightly better performance for small $K \leq 32$. For the remaining sparsity levels OSC and K-OSC have a clearly superior K-term approximation performance compared to the alternative methods. In order to achieve an average reconstruction performance of, e.g. 40 dB, OSC uses on average approximately 10% less non-zero coefficients than the second best performing Haar wavelet basis. Note that the single OSC basis yields, again, nearly the same K-term approximation performance as the K-OSC bases to which the sparsity level of the reconstruction was provided.

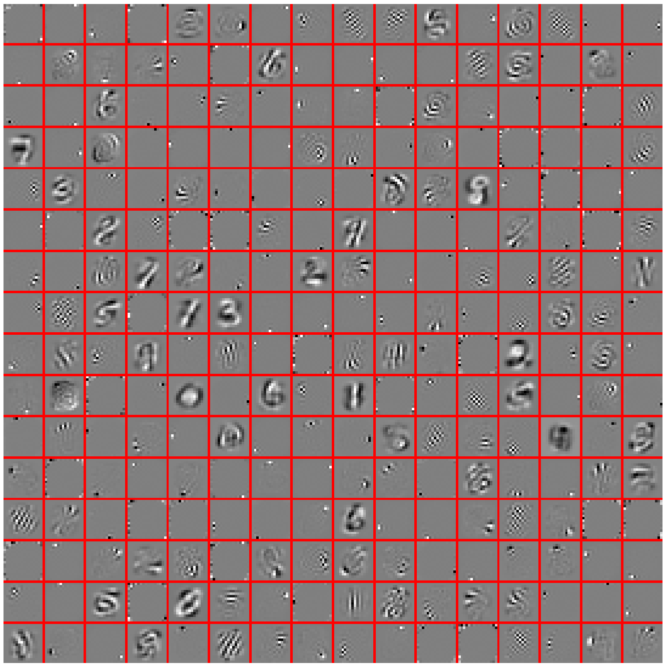


Figure 8. Orthogonal sparse coding basis for images of handwritten digits, obtained with OSC on the MNIST training set. The set of basis functions shows sensitivity for particular digits and digit combinations.

IV. Applications

A. Image Compression

To demonstrate the applicability of OSC, we conducted image compression experiments with gray level images (8 bit

[5] The images were resized such that a comparison to 2D Haar wavelets became possible.

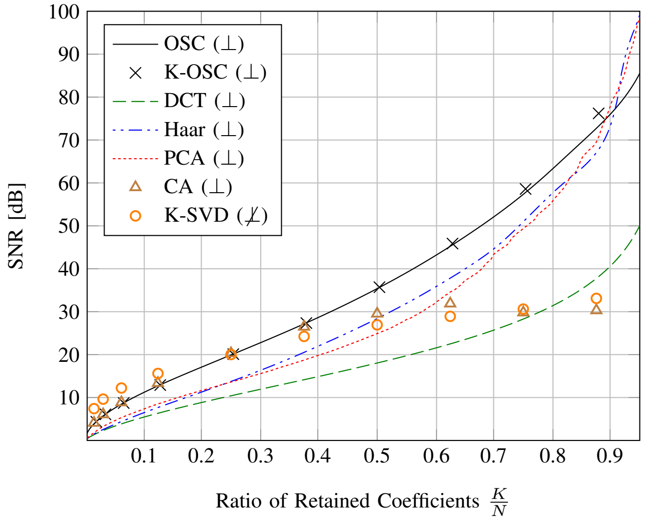


Figure 9. Average optimal K-term approximation performance on the MNIST test data set.

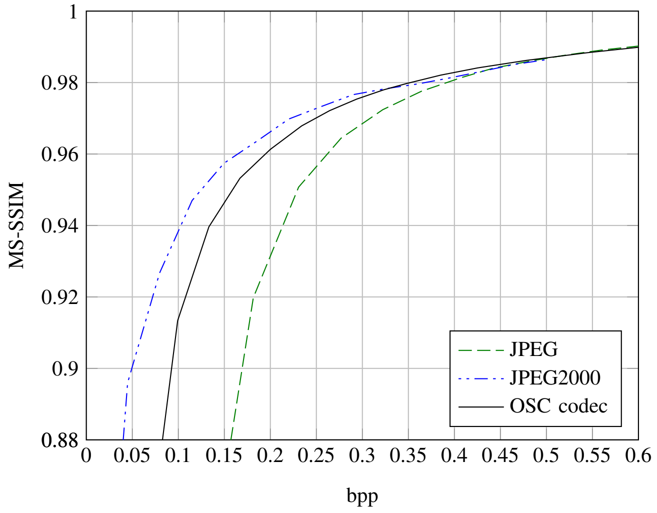


Figure 10. Rate distortion analysis for image *cathedral.pgm*

gray level depth) of a freely available benchmark data set [42] that contains 15 large uncompressed images. We divided the data set into 9 training images as well as 6 test images.

We derived an OSC codec from the JPEG baseline codec [6], where the 8×8 DCT has been replaced by an 8×8 OSC basis which was learned as described in Subsection III-B. A JPEG encoder processes the quantized AC coefficients of each image tile in a zig-zag pattern, which implicitly defines an ordering of the DCT basis functions by increasing frequency (and decreasing relevance). For the 8×8 OSC basis, we generated an analogous ordering. Therefore, it was counted how often, on average over all training samples, the coefficient of each OSC basis vector appeared among the $N/2$ largest coefficients. This yielded a ranking histogram, which was not only used to derive the ordering but also to generate the AC quantization table for the quality level 50 (on a scale from 1 to 100). The obtained ranking histogram was rescaled to have the same minimal and maximal quantization value as the JPEG luminance quantization table for the AC coefficients at that quality level. Quantization values for any other quality level are derived from this table as defined in the JPEG baseline standard. We also applied a zero run-length encoding of the AC coefficients and generated a Huffman code from patches, which were extracted from the training images.

For each test image we conducted a rate distortion analysis by varying the quality level. To measure the reconstruction performance, we evaluated the Multi-Scale Structural Similarity Index (MS-SSIM) [43] which is plotted against the bitrate (bpp), i.e., the file size of the compressed image (in bits) divided by the number of pixels. The comparison to the JPEG codec was done by applying the program `pnmtojpeg`, which uses the Independent JPEG Groups JPEG library. The used parameters were: `-grayscale`, `-dct=float`, and `-quality=n`. For the sake of a broader assessment, we also provide rate distortion curves for the JPEG2000 codec, although it is not based on an orthogonal basis but on biorthogonal Cohen-Daubechies-Feauveau wavelets. We used the Open JPEG tool `image_to_j2k` with default parameters.

Usually, we observed that the MS-SSIM for all three codecs converges to a common rate distortion curve as the bitrate exceeds an image dependent value around 0.5 bpp. For lower bitrates we consistently obtained better compression performance with the OSC codec than with the JPEG baseline standard (see Figure 10 for a prototypical rate distortion of the test image `cathedral.pgm`, see Appendix for the rate distortion analysis for the remaining test images).

Nevertheless, at low bitrates the JPEG2000 codec is still superior. Note, however, that it benefits from the multi-scale representation of images in the wavelet domain. Both, the JPEG and OSC codecs are patch based and suffer from blocking artifacts at very low bitrates. This might be one reason for their limitations compared to JPEG2000.

B. Image Denoising

We conducted experiments to assess the applicability of OSC to image denoising. The dictionaries were learned on the training data set described in Subsection III-B. Following the denoising framework proposed in [44], we distorted 9 gray value test images (512 × 512 pixels) with additive Gaussian noise with standard deviations $\sigma \in \{2, 5, 10, 15, 20, 25, 50\}$. After the noise was added, the noisy images were clipped to the range [0, 255]. From a noisy image, patches (16×16 pixels) were extracted from all locations and sparsely approximated by OMP applying a regularization with respect to the full size image reconstruction error. Note that a sparse approximation computed by OMP is optimal if the dictionary is orthogonal, i.e., the most relevant coefficients are retained. The denoised image is constructed by fusing the sparsely approximated patches. The gray value of each pixel is averaged from all its overlapping patches. For the entire image denoising procedure we used the `ompdenoise2.m` function of the KSVDBox v13 in combination with OMPBox v10 [40] with parameters as proposed in [44].

We compared image denoising performance between dictionaries learned by K-SVD, CA and OSC for 100 epochs. For

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

Table I
IMAGE DENOISING BASED ON SPARSE APPROXIMATIONS OF 16×16 IMAGE PATCHES. DENOISING PERFORMANCE IS MEASURED IN TERMS OF PSNR (DB) BETWEEN ORIGINAL IMAGE AND DENOISED ESTIMATE.

CA, we report results obtained with user parameter $K^* = 28$, because it yielded the best results of the investigated parameters $K \in \{20, 24, 28, 32\}$. For K-SVD, we report results obtained with user parameter combination $(K^*, cbsize^*) = (16, 1024)$, because it yielded the best results of the investigated parameters $(K, cbsize) \in \{4, 8, 12, 16\} \times \{512, 1024\}$. The denoising performance is listed in Table I in terms of PSNR (dB) averaged over 5 runs. The largest PSNR for each combination of image and noise level is highlighted in bold face. The experiments show that the K-SVD, CA, and OSC dictionary perform comparably well depending on the chosen images and noise levels. However, for K-SVD and CA, performance depends on the chosen parameters codebook size (only for K-SVD) and sparsity level K, while OSC does not require the optimization of such parameters.

V. Discussion and Conclusion

In this paper, we addressed the problem of learning orthogonal bases for sparse coding.

Inspired by traditional sparse coding algorithms, we first proposed the Canonical Approach (CA) which alternates between the adaptation of the basis and the update of the sparse coefficients. We showed that CA yields high performance at recovering a generating orthogonal basis from synthetic data, if the sparsity of the data is known and rather high.

As the main part of the paper we presented Orthogonal Sparse Coding (OSC), an unsupervised online learning algorithm, which is based on Hebbian learning and iterative Gram-Schmidt orthogonalization. OSC is able to identify the generating orthogonal basis from synthetic data even if the sparsity of the data is unknown or low. In contrast, K-SVD, CA, and the approach of Mishali et al. ([31]) had difficulties to converge at the basis identification task, particularly for the more challenging settings.

On natural image patches, OSC learns a basis that resembles wavelets with sensitivity to particular frequencies, orientations, and spatial localizations. In terms of optimal K-term approximation performance, OSC performs clearly better than the 2D Haar basis, PCA, CA, and K-SVD (except for very small K) and slightly better than the 2D DCT basis.

On images of handwritten digits, OSC learns a basis of patches that are sensitive to certain digits or combinations of digits, because they have adapted to the specific image class. Furthermore, the basis patches show localized gratings of various shapes. Again, the average optimal K-term reconstruction performance of the learned OSC basis is better than alternative predefined and learned bases.

For both, the synthetic and the real world data sets, we found that learning a single basis with OSC (setting user parameter $K = N$) provides a "universal" solution which gives equally good optimal K-term approximations for various sparsity levels K as compared to learning specific bases with the K-OSC algorithm for each K individually. This is very advantageous, since the "true" sparsity level K is often unknown in practice.

In contrast to PCA, encoding and dimensionality reduction are not based on a fixed linear subspace, but the best encoding subspace is selected individually for each data sample. Our results show that this additional freedom of choice is beneficial in the case of natural images and even more so for more specific signals like handwritten digits.

Moreover, we demonstrated the applicability of OSC for image compression by showing that OSC-based compression yields a better rate-distortion performance than JPEG, although it remains inferior to JPEG2000. We also demonstrated that orthogonal dictionaries learned by CA and OSC are useful for image denoising. CA and OSC dictionaries achieve denoising performance comparable to redundant dictionaries learned by K-SVD.

Regarding possible extensions, one should note that OSC is a flexible algorithm which can also learn undercomplete orthogonal dictionaries, i.e., orthogonal M-frames with $M < N$. With such an additional user-defined parameter M for the dictionary size, the orthogonal M-frame $\mathbf{U} \in \mathbb{R}^{N \times M}$ would be updated as in *Algorithm 1* except for lines 6 and 7, where N would be replaced by M. A further extension could be to learn multiple orthogonal bases from only one training set. In this case, OSC would update, for each data point, the best sparse coding basis only. Finally, one could use OSC for compressed sensing and adaptive hierarchical sensing as outlined in [45].

Acknowledgment

The research is funded by the DFG Priority Programme SPP 1527, grant number MA 2401/2-1.

References

[1] H. B. Barlow, "Possible principles underlying the transformation of sensory messages," *Sensory Communication*, pp. 217–234, 1961.

[2] D. J. Field, "What is the goal of sensory coding?" *Neural Computation*, vol. 6, no. 4, pp. 559–601, 1994.

[3] C. Zetzsche, E. Barth, and B. Wegmann, "The importance of intrinsically two-dimensional image features in biological vision and picture coding," in *Digital Images and Human Vision*, 1993, pp. 109–38.

[4] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, no. 381, pp. 607–609, 1996.

[5] M. Lewicki *et al.*, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, 2002.

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

[7] D. Taubman and M. Marcellin, Eds., *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2001.

[8] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for Sparse Representation Modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.

[9] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, Mar. 1997.

[10] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[11] J. B. Allen and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proceedings of IEEE*, vol. 65, no. 11, pp. 1558–1564, Nov. 1977.

[12] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM Journal of Mathematical Analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[13] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[14] I. Daubechies, "Orthonormal bases of compactly supported wavelets ii: Variations on a theme," *SIAM J. Math. Anal.*, vol. 24, no. 2, pp. 499–519, Mar. 1993.

[15] ——, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[16] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, Dec. 2008.

[17] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" 1997.

[18] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, vol. 12, no. 2, pp. 337–365, Feb. 2000.

[19] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[20] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, 1st ed. Wiley-Interscience, May 2001.

[21] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[22] K. Labusch, E. Barth, and T. Martinetz, "Robust and fast learning of sparse codes with stochastic gradient descent," *IEEE Transactions on Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 1048 – 1060, 2011.

[23] T. Martinetz, S. Berkovich, and K. Schulten, ""Neural-gas" network for vector quantization and its application to time-series prediction," *IEEE-Transactions on Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.

[24] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning Unions of Orthonormal Bases with Thresholded Singular Value Decomposition," in *Acoustics, Speech and Signal Processing, 2005. ICASSP 2005. IEEE International Conference on*, vol. V. Philadelphia, PA, United States: IEEE, 2005, pp. V/293–V/296.

[25] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," vol. 5, pp. 2443–2446 vol.5, 1999.

[26] M. Yaghoobi, T. Blumensath, and M. E. Davies, "Dictionary learning for sparse approximations with the majorization method." *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[27] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[28] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm." *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[29] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 438–445.

[30] R. R. Coifman, Y. Meyer, and V. Wickerhauser, "Wavelet analysis and signal processing," in *Signal Processing, Part I: Signal Processing Theory*, L. Auslander, T. Kailath, and S. K. Mitter, Eds. New York, NY: Springer-Verlag, 1990, pp. 59–68.

[31] M. Mishali and Y. Eldar, "Sparse source separation from orthogonal mixtures," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009, pp. 3145–3148.

[32] N. Dobigeon and J.-Y. Tourneret, "Bayesian orthogonal component analysis for sparse representation," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2675–2685, 2010.

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

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

[35] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 89 – 105, 2014.

[36] J. Huang, L. Guo, Q. Feng, W. Chen, and Y. Feng, "Sparsity-promoting orthogonal dictionary updating for image reconstruction from highly undersampled magnetic resonance data," *Physics in Medicine and Biology*, vol. 60, no. 14, p. 5359, 2015.

[37] P. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[38] H. Schütze, E. Barth, and T. Martinetz, "Learning orthogonal bases for k-sparse representations," in *Workshop New Challenges in Neural Computation 2013*, ser. Machine Learning Reports, B. Hammer, T. Martinetz, and T. Villmann, Eds., vol. 02/2013, 2013, pp. 119–120.

[39] W. S. Geisler and J. S. Perry, "Statistics for optimal point prediction in natural images," *Journal of Vision*, vol. 11, no. 12, Oct. 2011.

[40] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," Tech. Rep., 2008.

[41] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[42] "Image Compression | Benchmark," 2014. [Online]. Available: http://www.imagecompression.info/

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

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

[45] H. Schütze, E. Barth, and T. Martinetz, "An adaptive hierarchical sensing scheme for sparse signals," in *Human Vision and Electronic Imaging XIX*, ser. Proc. of SPIE Electronic Imaging, B. E. Rogowitz, T. N. Pappas, and H. de Ridder, Eds., vol. 9014, 2014, pp. 15:1–8.

[46] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, vol. 15, no. 3, pp. 267–273, Nov. 1982.

Appendix

Proof of Theorem 1

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

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

$$\mathbf{v}_k = \mathbf{u}_k - \sum_{l=1}^{k-1} (\mathbf{u}'^T_l \mathbf{u}_k) \mathbf{u}'_l \ , \tag{13}$$

followed by the normalized Hebbian-like main update

$$\mathbf{u}'_k = \frac{\mathbf{v}_k + \varepsilon(\mathbf{v}_k^T \mathbf{x}_k)\mathbf{x}_k}{\|\mathbf{v}_k + \varepsilon(\mathbf{v}_k^T \mathbf{x}_k)\mathbf{x}_k\|_2} \ , \tag{14}$$

where

$$\mathbf{x}_k = \mathbf{x} - \sum_{l=1}^{k-1} (\mathbf{u}'^T_l \mathbf{x}) \mathbf{u}'_l \ . \tag{15}$$

$\mathbf{u}_1$ is only updated by (14) due to (13). In that sense $\mathbf{v}_1 = \mathbf{u}_1$ and $\mathbf{x}_1 = \mathbf{x}$ due to (15).

We will show by induction that

$$\mathbf{v}_k = \mathbf{u}_k - \varepsilon(\mathbf{u}_k^T \mathbf{x}) \sum_{l=1}^{k-1} (\mathbf{u}_l^T \mathbf{x}) \mathbf{u}_l + \mathcal{O}(\varepsilon^2) \ . \tag{16}$$

Note that by (16) it holds $\|\mathbf{v}_k\|_2 \approx 1 + \mathcal{O}(\varepsilon^2)$. Hence, the Taylor expansion of update step (14) up to first order in ε is

$$\mathbf{u}'_k = \mathbf{v}_k + \varepsilon(\mathbf{v}_k^T \mathbf{x}_k)(\mathbf{x}_k - (\mathbf{v}_k^T \mathbf{x}_k)\mathbf{v}_k) + \mathcal{O}(\varepsilon^2) \ . \tag{17}$$

Note that (14) is a Oja learning rule, i.e., a Hebbian learning rule with a normalization constraint. We apply the same expansion as in Section 4 of [46].

Furthermore, since $\mathbf{v}_k^T \mathbf{x}_k = \mathbf{v}_k^T \mathbf{x}$ and with (16) we have $\mathbf{v}_k^T \mathbf{x}_k = \mathbf{u}_k^T \mathbf{x} + \mathcal{O}(\varepsilon)$ as well as $(\mathbf{v}_k^T \mathbf{x}_k)\mathbf{v}_k = (\mathbf{u}_k^T \mathbf{x})\mathbf{u}_k + \mathcal{O}(\varepsilon)$. Hence, (17) can be restated as

$$\mathbf{u}'_k = \mathbf{v}_k + \varepsilon \left(\mathbf{u}_k^T \mathbf{x}\right) \left(\mathbf{x}_k - \left(\mathbf{u}_k^T \mathbf{x}\right) \mathbf{u}_k\right) + \mathcal{O}(\varepsilon^2) \ . \tag{18}$$

We will now show (16) by induction.

Initial Step $k = 1$. According to (13), we have by definition $\mathbf{v}_1 = \mathbf{u}_1$ which satisfies (16).

Induction Step $(k-1) \to k$. According to (13), we have by definition

$$\mathbf{v}_k \quad = \quad \mathbf{u}_k - \sum_{l=1}^{k-1} (\mathbf{u}'^T_l \mathbf{u}_k) \mathbf{u}'_l \ .$$

Due to induction hypothesis (16), $\mathbf{u}'_l$ can be restated according to (18). In addition to (16) we will use $\mathbf{u}_l^T \mathbf{u}_k = 0$ and $\mathbf{v}_l^T \mathbf{u}_k = \mathcal{O}(\varepsilon^2)$ as well as $\mathbf{u}_k^T \mathbf{x}_l = \mathbf{u}_k^T \mathbf{x} + \mathcal{O}(\varepsilon)$.

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

The induction is complete.

Combining (18) and (16) gives us up to first order in ε

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

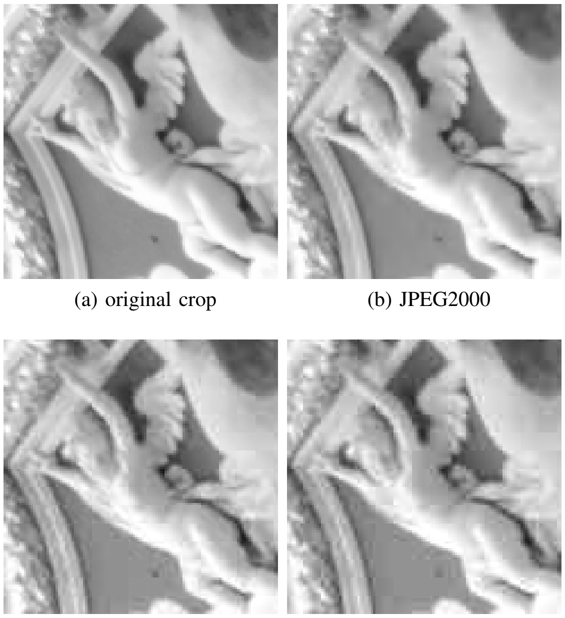
Hence, for small ε and with (15) we obtain

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

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

□

Zoom In of Compressed Image Cathedral



(a) original crop

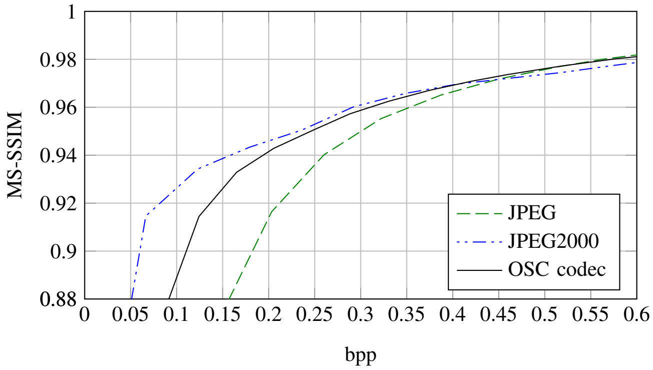
(b) JPEG2000

(c) OSC

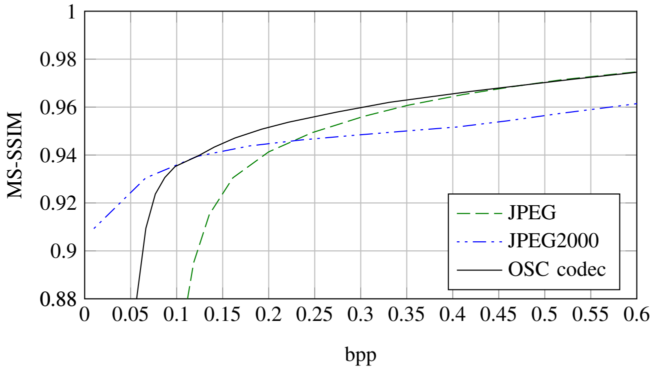
(d) JPEG

Figure 11. An image region (120 × 120 pixels) cropped from the upper left part of test image *cathedral* after compression by the JPEG2000, OSC, and JPEG codecs at compression rate 0.29 bpp.

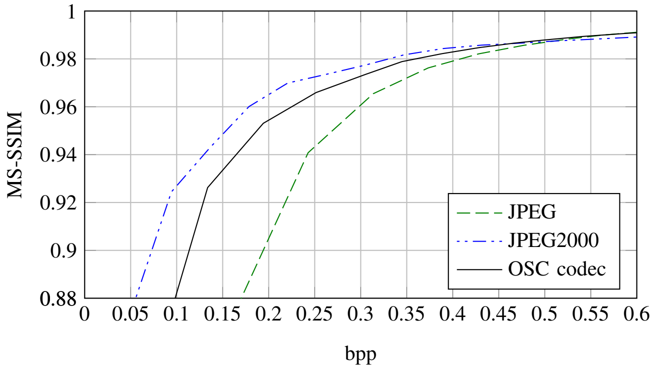
Compression Results of Remaining Test Images



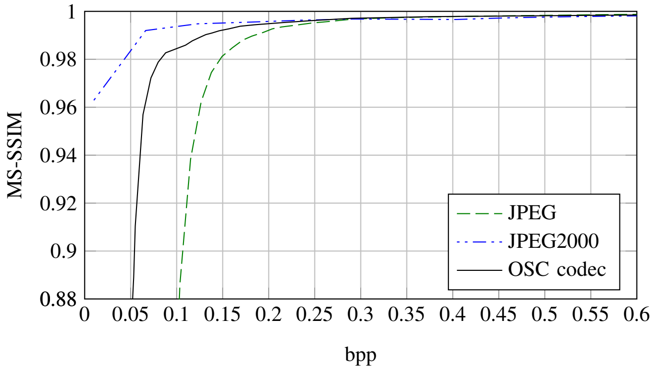
Rate distortion analysis for image *bridge.pgm*



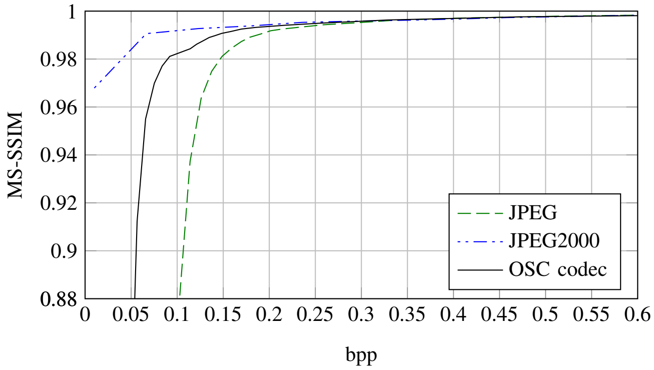
Rate distortion analysis for image *deer.pgm*



Rate distortion analysis for image *big_building.pgm*



Rate distortion analysis for image *flower_foveon.pgm*



Rate distortion analysis for image *spider_web.pgm*

Henry Schütze studied Computer Science at the Brandenburg University of Technology, Cottbus, Germany and at the University of Lübeck, Germany. He graduated in 2011 with an MSc and is now research assistant at the Institute for Neuro- and Bioinformatics of the University of Lübeck, where he pursues a PhD degree. His main research interests include Sparse Coding and Compressed Sensing.

Erhardt Barth received the PhD degree in electrical and communications engineering from the Technical University of Munich, Germany. He is a Professor at the Institute for Neuro- and Bioinformatics, University of Lübeck, Germany, where he leads the research on human and machine vision. He has conducted research at the Universities of Melbourne and Munich, the Institute for Advanced Study in Berlin, and the NASA Vision Science and Technology Group in California.

Thomas Martinetz is full professor of computer science and director of the Institute for Neuro- and Bioinformatics at the University of Lübeck, Germany. He studied Physics at the Technical University of Munich, Germany and did his PhD work in Biophysics at the Beckman Institute for Advanced Science and Technology of the University of Illinois at Urbana-Champaign, USA. From 1991 to 1996 he led the project Neural Networks for automation control at the Corporate Research Laboratories of the Siemens AG in Munich. From 1996 to 1999 he was Professor for Neural Computation at the Ruhr-University of Bochum and head of the Center for Neuroinformatics.