Sequential Minimal Eigenvalues - An Approach to Analysis Dictionary Learning

Boaz Ophir, Michael Elad, Nancy Bertin, Mark D. Plumbley

SEQUENTIAL MINIMAL EIGENVALUES – AN APPROACH TO ANALYSIS DICTIONARY LEARNING

Boaz Ophir [a], Michael Elad [a], Nancy Bertin [b], Mark D. Plumbley [c]

[a] Technion – Israel Institute of Technology
Department of Computer Science
Haifa 32000, Israel

[b] CNRS, IRISA – UMR 6074
Campus de Beaulieu
35042 Rennes Cedex, France

[c] Queen Mary University of London
Center for Digital Music
London E1 4NS, United Kingdom

ABSTRACT

Over the past decade there has been a great interest in a synthesis-based model for signals, based on sparse and redundant representations. Such a model assumes that the signal of interest can be decomposed as a linear combination of *few* columns from a given matrix (the dictionary). An alternative, analysis-based, model can be envisioned, where an analysis operator multiplies the signal, leading to a sparse outcome. In this paper we propose a simple but effective analysis operator learning algorithm, where analysis "atoms" are learned sequentially by identifying directions that are orthogonal to a subset of the training data. We demonstrate the effectiveness of the algorithm in three experiments, treating synthetic data and real images, showing a successful and meaningful recovery of the analysis operator.

1. INTRODUCTION

In a wide range of fields, from signal processing to machine learning, sparse representations and their computation has become a well-known topic. Most of the time, the term "sparse representations" refers to a generative model where the signals of interest are described as a linear combination of a few atoms chosen from a redundant and predefined set. More specifically, the signal $\mathbf{x} \in \mathbf{R}^d$ is assumed to be composed as $\mathbf{x} = \mathbf{D}\mathbf{z}$, where $\mathbf{D} \in \mathbf{R}^{d \times n}$ is a redundant dictionary with $d \leq n$, and $\mathbf{z} \in \mathbf{R}^n$ is a sparse vector, serving as the representation of the signal. The number of non-zeros in $\mathbf{z}$, denoted $\|\mathbf{z}\|_0$, is assumed to be much smaller than d. This signal model is typically referred to as the *synthesis* model.

Why do we need such a model? In typical situations, the signal of interest $\mathbf{x}$ is only observed through a set of linear measurements $\mathbf{y} \in \mathbf{R}^m$ which are noisy and (if $m \leq d$) possibly incomplete,

$$\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{e} \quad \text{where} \quad \|\mathbf{e}\|_2 \leq \varepsilon. \tag{1}$$

The recovery of $\mathbf{x}$ from $\mathbf{y}$ in such a case becomes a much more stable process by invoking the sparsity assumption on the desired signal. Indeed, there is a wide family of methods, termed *pursuit algorithms*, that are designed to solve this problem [2].

A key problem with the model described above is the choice of $\mathbf{D}$. Naturally, when working on a specific signal source, we would like to have a dictionary that enables sparse representations for the signals in question. In recent years there has been a growing interest in learning an appropriate dictionary from a set of example signals (training set). Two well-known options are the MOD and the K-SVD algorithms [4, 1, 9].

The *synthesis* model has been intensively studied in the past decade, and it is now a well established field of research. However, there is another viewpoint to sparse representations – an *analysis* one. The analysis model relies on a linear operator $\Omega : \mathbf{R}^d \to \mathbf{R}^p$ such that the coefficient vector $\Omega\mathbf{x} \in \mathbf{R}^p$ is expected to be sparse for the signals in mind [3]. While it sounds similar to the above synthesis counterpart, it is in fact very different [7]. Interestingly, relatively little is known about the analysis model. Though this model has been used for a while in some applications [10, 11, 8, 12], the analysis model in itself has been given much less attention in recent literature, compared to the synthesis.

In this paper we focus on the analysis model, and in particular, the development of an algorithm that would learn the analysis operator Ω from a set of examples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, so that the analysis coefficients $\Omega\mathbf{X}$ are sparse. The paper is organized as follows. In Section 2, we give a precise definition of the sparse analysis model. We describe the proposed learning algorithm in Section 3, and demonstrate it in Section 4. A discussion on the limits of this method, and a perspective for future work is provided in Section 5.

2. THE ANALYSIS MODEL

As described above, the analysis model for the signal $\mathbf{x} \in \mathbf{R}^d$ uses the possibly redundant analysis operator $\Omega : \mathbf{R}^d \to \mathbf{R}^p$ (redundancy here implies $p \geq d$). This model assumes that the vector $\Omega\mathbf{x}$ should be sparse. Unless stated otherwise, we shall assume hereafter that the rows in Ω are in general position, implying that every subset of d or less rows are necessarily linearly independent. This is equivalent to the assumption that the spark of Ω^T is full [2].

In the synthesis model the representation $\mathbf{z}$ can be arbitrarily sparse, $\|\mathbf{z}\|_0 = k \ll d$. Also, the signal is characterized by the k non-zero indices in the representation vector $\mathbf{z}$, as those describe the atoms constructing it, which in turn define the subspace this signal belongs to. In contrast, the analysis model leads necessarily to a mild sparsity, $\|\Omega\mathbf{x}\|_0 \geq p - d$, as otherwise there would be d or more rows that are orthogonal to $\mathbf{x}$, which is possible only for $\mathbf{x} = 0$. Thus, for a highly redundant analysis operator, the cardinality of $\Omega\mathbf{x}$ is expected to be quite high. But this is not necessarily bad news, as we shall see.

In the analysis model we put emphasis on the zeros of the vector $\Omega\mathbf{x}$, and define the *cosparsity* ℓ of the analysis model to be the number of zeros in the vector $\Omega\mathbf{x}$,

$$\|\Omega\mathbf{x}\|_0 = p - \ell. \tag{2}$$

Naturally, $0 \leq \ell \leq d$.

In the analysis model the signal $\mathbf{x}$ is characterized by the cosparsity – the zeros of the vector $\Omega\mathbf{x}$ – as the indices of these zeros define the subspace the signal belongs to. This explanation may become clearer as we turn to describe another interpretation of these two models as special instances of *union-of-subspaces* (UoS). Indeed, both the synthesis and the analysis models form unions of subspaces in which their signals reside [6].

In a general UoS model it is assumed that there is a collection of subspaces S_γ, $\gamma \in \Gamma$, such that a signal of interest $\mathbf{x}$ coincides with an element from the union $\cup_{\gamma\in\Gamma} S_\gamma$. The synthesis model with sparsity k implies that every signal is created as a linear combination of k columns from $\mathbf{D}$. Thus, each of the $\binom{n}{k}$ possible choices of supports leads to a subspace, and those together describe the geometrical location of the signals in this model in $\mathbf{R}^d$.

The analysis model is similarly associated to a UoS model as follows. The signal that satisfies the model assumption $\|\Omega\mathbf{x}\|_0 = p-\ell$ is orthogonal to ℓ rows from Ω. Thus, it is simply in the orthogonal complement to the linear combination of these rows. This way, the analysis model corresponds to a union of all the possible $\binom{p}{\ell}$ complement subspaces for ℓ chosen rows from Ω. Note that in general, the synthesis and the analysis unions of subspaces are very different – for example, if $p = n = 2d$ and $k = d-\ell \ll d$, then the subspaces united by the two models are of the same dimension (k), but their number is entirely different, with many more subspaces included in the analysis model.

3. ANALYSIS OPERATOR LEARNING

For the analysis model to be effective, we should choose Ω such that the cosparsity ℓ is maximized for the signals we aim to operate on. This is comparable to the quest for a dictionary $\mathbf{D}$ in the synthesis model that leads to maximal sparsity (minimal k). In the same way as for the synthesis model, we have two options – either choose a pre-specified operator that seems to be in agreement with the data (e.g., Gabor filters, wavelet, etc.), or construct Ω by learning it from signal examples [9]. In this paper we target the second option, and propose a learning algorithm for Ω using a training set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ \ldots\ , \mathbf{x}_N]$.

3.1 Basic Step - Minimal Eigenvalue

The inner process of the proposed algorithm attempts to learn a single analysis vector ω, which will correspond to a single row in the analysis operator Ω. Such a vector is known to be orthogonal (or nearly so in practice) to a non-negligible subset of the training signals. By "non-negligible" we mean that if the rows of Ω are used uniformly in constructing the analysis signals, then we expect that $N\ell/p$ of the signals would be orthogonal to each row on average.

Finding a candidate row ω is done by singling out an eigenvector of the Gram matrix $\mathbf{X}_\Lambda\mathbf{X}_\Lambda^T$ of the data, where $\mathbf{X}_\Lambda$ contains a subset of the training examples, by removing columns of irrelevant examples. This problem is complex since we need to find both the group of examples to use, Λ, and the eigenvector that corresponds to it. Naturally, if Λ is known, we can target the eigenvector associated with the *smallest* eigenvalue (rather like [5]). The essence of the process is iteratively prune the training data, keeping only training vectors that have low correlation with the current estimate of the analysis vector. The analysis vector is then updated and the operation is repeated iteratively until convergence (or a predetermined stopping rule) is reached. At the final stage, the learned analysis vector is orthogonal or nearly-so to the subset of training vectors $\mathbf{x}$ that were kept all along the process, which justifies its choice as a row of Ω. Algorithm 1 describes this process, and it output is one analysis vector ω.

Algorithm 1 : Basic Algorithm – Finding one Row.

Given: Training data $\mathbf{X} \in \mathbf{R}^{d\times N}$.
Desired: A row from the analysis operator Ω that has p rows, leading to cosparsity ℓ.
Initialization: Set an initial threshold value $\theta = \infty$.
Loop: Repeat until threshold $\theta \le \theta_0$.

Initial row: Set ω to an initial random value.
Inner Loop: Repeat until convergence:

- *Inner Products:* compute the inner products between ω and all the examples $\mathbf{X}$.
- *Set Threshold:* set a threshold θ such that $c\ell N/p$ of the inner products are below this threshold ($c \le 1$).
- *Set Indicator:* set Λ to specify only inner products smaller than θ.
- *Prune:* prune irrelevant examples and build $\mathbf{X}_\Lambda$.
- *Update ω:* set ω to be the eigenvector associated with the smallest eigenvalue of $\mathbf{X}_\Lambda\mathbf{X}_\Lambda^T$.

The inner loop in the above algorithm shows convergence (which should be proven), but the final threshold found would not necessarily be small enough. The outer loop adds a random flavor to this process by choosing a different initialization, thereby enabling the overall process to find a suitable row ω that is nearly orthogonal to the set $\mathbf{X}_\Lambda$.

3.2 From a Basic Step to the Complete Algorithm

So far we have concentrated on finding a single row from Ω. Now we would like to extend this method to find all the rows. The approach we take is sequential, which means that we apply Algorithm 1 p or more times, each time producing a new candidate row to Ω. In our work we consider two ways to construct such an overall algorithm: a *deterministic* approach and a *randomized* one.

The deterministic algorithm repeats Algorithm 1 to find one row at a time, while pruning away training examples that are orthogonal to the analysis vectors already found. This practically guarantees that we will learn one distinct new analysis vector with each iteration. This algorithm is presented in Algorithm 2.

Note that the two pruning options described in Algorithm 2 have different roles. The first, corresponding to $i < \ell$ guarantees that the newly found row will be different from previous ones, as it relies on different training examples. The second pruning option for $i \ge \ell$ simply removes examples in the training set that are already well-treated by the rows found so far, thus enabling the "less-represented" examples to get a better treatment. The added step of returning to the complete training set after iteration $\ell-1$ is necessary because all those examples that were removed should be returned before entering iteration $i = \ell$.

This algorithm exhibits an obvious weakness: the pruned training set tends to shrink in size very rapidly after iteration $i = \ell$, leading to situations where the very last rows of Ω

Algorithm 2 : Overall Deterministic Algorithm.

Given: Training data $\mathbf{X} \in \mathbf{R}^{d \times N}$.
Desired: An analysis operator Ω with p rows, leading to cosparsity ℓ.
Initialization: Set $\mathbf{X}_1 = \mathbf{X}$ to contain all the training examples, and set Ω to be an empty matrix.
Loop: Repeat for $i = 1$ to p:

- *Basic Step:* run Algorithm 1 on $\mathbf{X}_i$ and find ω.
- *Accumulate:* Add the found ω as a new row to Ω.
- *Prune Training Set:*
  - For $i < \ell$ remove from $\mathbf{X}_i$ all examples that are orthogonal to all the accumulated rows in Ω thus far.
  - For $i \geq \ell$ remove from $\mathbf{X}_i$ all examples that are orthogonal to at least ℓ rows from Ω.
  - If $i = \ell - 1$, set $\mathbf{X}_i = \mathbf{X}$.

cannot be found reliably. One possible solution might be to begin the learning process with huge amounts of data, but this solution is not practical in most cases.

An alternative solution is a randomized algorithm. Instead of the pruning used in Alg. 2, at each epoch a random subset of the whole training data $\mathbf{X}$ is chosen, and used as input for the basic step. This way, we ensure that, at each iteration, we will have sufficient data to carry out a meaningful learning. Since we start each learning epoch with a different subset of the input data, we can expect to learn different atoms. However, we cannot guarantee that there will be no repetitions, so to avoid this, we compare each new atom to the previously learned ones, and accept the new atom only if it is sufficiently different.

Another situation that may take place is a detection of a row ω that is a linear combination of a pair (or a triplet) of original rows from Ω. The probability for this is small but not negligible, and in the proposed algorithm we have chosen not to prune such rows away. Note that for applications using the learned Ω, these additional rows do not pose a problem, as long as they are found in addition to the p true rows. Because of the above-described two problems, the number of iterations needed to learn the original p atoms is often larger than p. This algorithm is described in Algorithm 3.

Algorithm 3 : Overall Random Algorithm.

Given: Training data $\mathbf{X} \in \mathbf{R}^{d \times N}$.
Desired: An analysis operator Ω with p rows, leading to cosparsity ℓ.
Initialization: Set Ω to be an empty matrix.
Loop: Repeat until we have p distinct rows in Ω:

- *Choose Examples:* Set $\mathbf{X}_t$ to contain a randomly chosen subset of the training examples with ρN examples, ($\rho \ll 1$).
- *Basic Step:* run Algorithm 1 on $\mathbf{X}_t$ and find ω.
- *Admittance Criterion:* verify that ω is different from all accumulated rows in Ω, and otherwise remove it.
- *Accumulate:* Add the found ω as a new row to Ω.

Alg. 3 is expected to work well when the analysis atoms are of equal importance, i.e. all are orthogonal to about the same number of training examples. if this is not the case, we may not be able to learn the rarer atoms. One way to overcome this would be to keep track of which data samples are used when learning different analysis atoms. We may then remove from the data set, or in some way re-weight, those data samples that have already contributed in the learning of many atoms. In essence we are saying that we have already learned all we can from these samples, so we may ignore them completely in the following learning cycles. The exact formulation of this mechanism is part of our future work.

4. EXPERIMENTS

In this section we introduce three experiments that demonstrate the operation of Alg. 3. We start with purely synthetic data set that follows the analysis model, as described in Section 2. We show that Alg. 3 recovers the original Ω very accurately and with high-probability even when the data is noisy. In the second experiment we turn to a synthetic piecewise constant image, and show that the found Ω is related to finite differences, as expected. Finally, we test the learning algorithm on a true image, and suggest that the resulting rows in Ω make natural sense.

4.1 Synthetic Data

In the synthetic experiment, Ω is randomly generated with iid Gaussian entries. We have also tested a known analysis operator such as the overcomplete DCT and the results are the same. Analysis-modeled signals are synthetically created by first choosing ℓ rows from Ω uniformly at random, and then projecting a random vector onto the complement space formed by these rows. The resulting signals are then normalized. This way we guarantee that the data is exactly ℓ cosparse with respect to Ω.

We tested our algorithm on signals of dimension $d = 10$ with an analysis matrix with $p = 20$ rows, and we set the cosparsity to $\ell = 8$. We generated 10,000 input signals with the above procedure. We ran tests with and without additive Gaussian white noise, $\sigma = 0.1/\sqrt{d}$. Each iteration (in the randomized algorithm) used $\rho = 0.1$. The stopping rule for the inner-loop was set to be when the threshold θ became lower than 1.5σ, and the constant c was set to 1.

Each learned analysis vector ω has to fulfill an admittance criterion related to its similarity with the previously learned atoms. In this experiment, we accepted vectors whose maximal inner product with the rows of Ω was $\theta_s = 1 - \delta$, for different values of δ. Setting $\delta = 0.02$, all the vectors learned were true rows from Ω. When setting it to a more strict admittance criterion ($\delta = 0.01$), some learned vectors were linear combination of rows from Ω.

Results: in the noiseless experiments, *all* of the analysis vectors were recovered with a mean-squared-error of $\approx 10^{-8}$ per coefficient. The basic step was run 100 times on average until all $p = 20$ analysis vectors were found, meaning that each vector was found 5 times on average. The number of vectors found and rejected was low, even with the stricter admittance criterion.

Adding noise to the experiments affected the results mainly in two ways. Again, *all* of the analysis vectors were recovered, but with a mean-squared-error of $\approx 10^{-4}$ per coefficient, and the basic step needed to be run on average 300 times until all $p = 20$ analysis vectors were found. The number of vectors found and rejected was not affected by the noise.

4.2 Synthetic Image Data

In this experiment the randomized algorithm was run on artificially created piecewise constant images. This image was created by accumulating rotated rectangles of random size and random locations to the image. Such an image was chosen for our demonstration since we can hypothesize on the expected shape of the analysis vectors we expect to learn. For a piecewise constant data, we expect these vectors to take some form of oriented derivatives.

Our training data is composed of 8×8 patches extracted from a piecewise constant image, as in Figure 1. Thus $d = 64$. We chose to learn a 4 times redundant analysis operator, so $p = 4d = 256$. Each iteration used $\rho = 0.1$ of the data. The stopping rule for the inner loop was set to be when the threshold θ becomes lower than 10 (gray-values), and the data threshold constant c was set to 1.3.

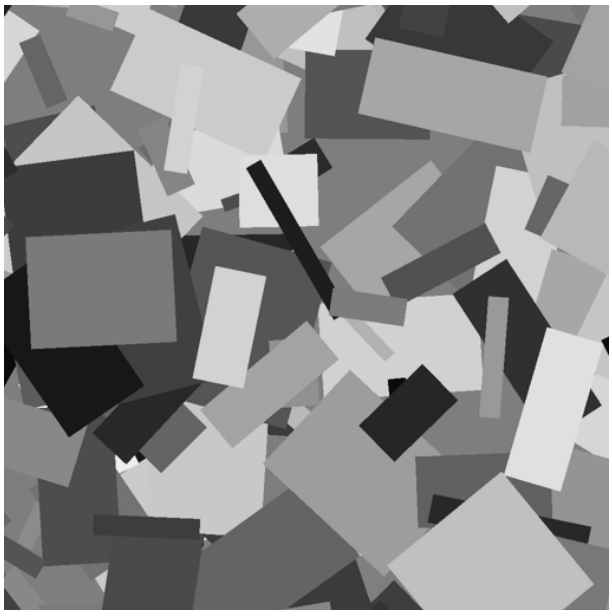

Figure 1: A sample piecewise constant image.

A major difference between this experiment and the previous one is that the analysis operator Ω we expect to learn is not necessarily of maximal spark. This implies that there could be linear dependencies between the rows. Thus, it may well be that the cosparsity in such cases exceeds the signal dimension, i.e., $\ell > d$. As a consequence, in this experiment we can try to learn a redundant analysis operator that will give us cosparsity ℓ that is much greater than $d = 64$. Indeed, the results shown in Figure 2 show sample atoms obtained for $\ell = 64$, $\ell = 128$, and $\ell = 192$. As can be seen, the atoms tend to be more localized when ℓ is larger. The atoms shown in Figure 2 represent various (first, second, third) derivatives in different locations and with different orientations, as expected.

4.3 True Image Data

The last experiment in this paper is very similar to the previous one, but applied on a natural image – "peppers" shown in Figure 3. A sample set from the analysis rows obtained for cosparsity $\ell = 3d = 192$ is given in Figure 4. As can be seen, the atoms are very similar to those obtained earlier for the synthetic image.

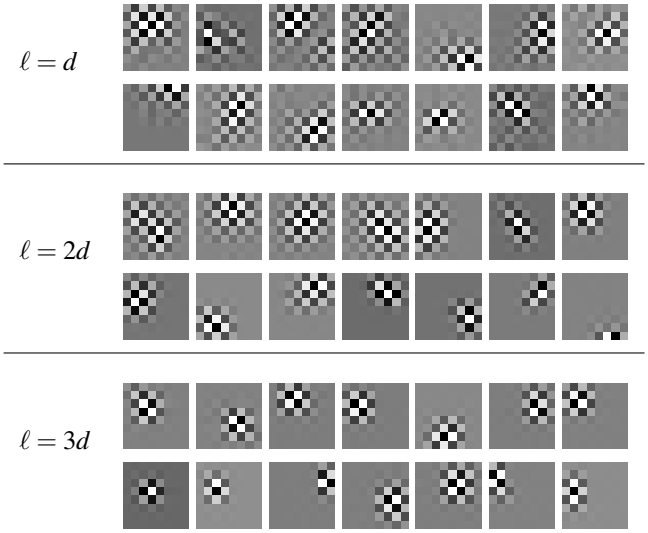

Figure 2: Sample analysis atoms found for the synthetic piecewise constant image using different ℓ values.

Figure 3: The "peppers" image.

In order to illustrate the effectiveness of the found analysis operator, Ω, Figure 5 presents the projection values $\Omega\mathbf{X}$ for 1000 randomly chosen image patches from the training set. The projection values are sorted per each column in a descending order of their absolute value. Ideally, we should get that the last $3d$ rows are exact zeros, but as we work on true data and allow for small inner products instead of exact zeros, the behavior is little-bit different. As can be seen, the projection values tend to decrease very fast towards zero (zero is white color in order to save toner). For reference, a similar projection matrix is shown for a randomly chosen Ω (with zero mean rows), and it is clear that it is much less effective in terms of the cosparsity it leads to.

5. CONCLUSIONS

Learning the *analysis* operator that gives sparse analysis coefficients is a new field of activity. In this paper we presented

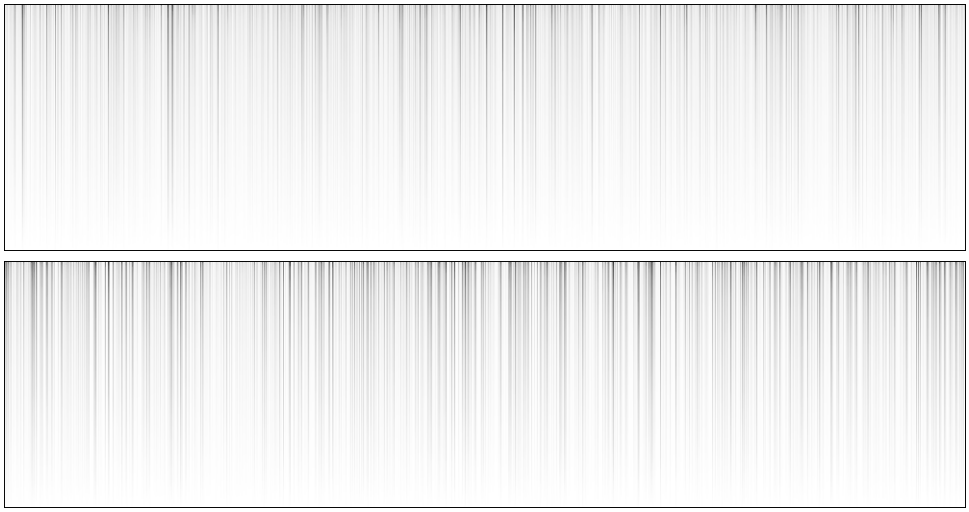

Figure 5: The projection values (after sorting their absolute values). The top refers to the learned Ω, while the bottom refers to a random matrix with Gaussian iid entries (and normalized rows).

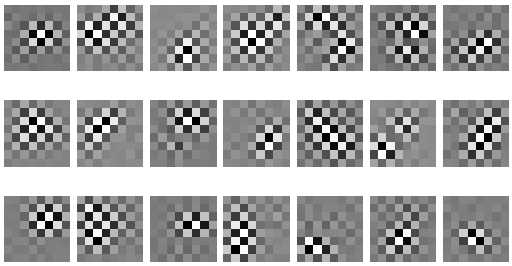

Figure 4: Sample analysis atoms found for the image "peppers", obtained for $\ell = 3d$.

a simple but effective algorithm for learning a sparsifying analysis operator. We have shown that our algorithm is able to recover the analysis operator with high precision in synthetic experiments and is able to learn derivative like operators for analysis of piecewise constant images. Future work includes studying the effects of the various parameters and understanding how they affect the learning process and the learned dictionary, handling linear dependencies within the analysis operator, proving converges for the basic set of the algorithm, and developing more elaborate analysis learning schemes.

Acknowledgement

This work was supported by the European CommissionŠs FP7-FET program, SMALL project (grant agreement no. 225913).

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[2] A. Bruckstein, D. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, February 2009.

[3] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *IOP Inverse Problems*, 23(3):947–968, June 2007.

[4] K. Engan, S. Aase, and J. Hakon-Husoy. Method of optimal directions for frame design. In *Proceedings of ICASSP*, pages 2443Ű–2446, 1999.

[5] F. Jaillet, R. Gribonval, M. Plumbley, and H. Zayyani. An l1 criterion for dictionary learning by subspace identification. In *Proceedings of ICASSP 2010*, pages 5482–5485, March 2010.

[6] Y. M. Lu and M. N. Do. A Theory for Sampling Signals From a Union of Subspaces. *IEEE Transactions on Signal Processing*, 56(6):2334–2345, June 2008.

[7] S. Nam, M. Davies, M. Elad, and R. Gribonval. Cosaprse analysis modeling - uniqueness and algorithms. In *Proceedings of ICASSP - to appear*, May 2010.

[8] J. Portilla. Image restoration through l0 analysis-based sparse optimization in tight frames. In *Proceedings of SPIE, Vol. 7446*, pages 3865–3868, August 2009.

[9] R. Rubinstein, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *IEEE Proceedings*, 98(6):1045–1057, April 2010.

[10] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, November 1992.

[11] R. Schultz and R. Stevenson. A Bayesian approach to image expansion for improved definition. *Image Processing, IEEE Transactions on*, 3(3):233 –242, May 1994.

[12] I. Selesnick and M. Figueiredo. Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors. In *Proceedings of ICIP*, August 2009.