Cosparse Dictionary Learning for the Orthogonal Case

Paul Irofti
Department of Automatic Control and Computers
University Politehnica of Bucharest
313 Spl. Independenţei, 060042 Bucharest, Romania
Email: paul@irofti.net

Bogdan Dumitrescu
Department of Signal Processing
Tampere University of Technology
PO BOX 553, 33101, Tampere, Finland
Email: bogdan.dumitrescu@acse.pub.ro

***Abstract*—Dictionary learning is usually approached by looking at the support of the sparse representations. Recent years have shown results in dictionary improvement by investigating the cosupport via the analysis-based cosparse model. In this paper we present a new cosparse learning algorithm for orthogonal dictionary blocks that provides significant dictionary recovery improvements and representation error shrinkage. Furthermore, we show the beneficial effects of using this algorithm inside existing methods based on building the dictionary as a structured union of orthonormal bases.**

***Index Terms*—sparse representation, orthogonal blocks, cosparse, dictionary design**

I. Introduction

Dictionary learning through sparse representations has shown promising results in signal processing with efficient applications in compression, detection, denoising and compressed sensing. One of the central problem in this field is modeling observed signals by using just a few columns, also called atoms, from a learned dictionary. This process is denoted in the literature as the synthesis-based sparse representation model [1], where we are interested in the few non-zero entries of the approximations. Recent years have shown approximation improvements when instead we analyze the set of atoms that do not participate in representing a signal. This process is also called the cosparse analysis model and is described in detail in [2]. In both models the focal point is the representation of a signal $y \in \mathbb{R}^p$ using a set of atoms from the dictionary.

In the sparse or synthesis model, the dictionary is $D \in \mathbb{R}^{p \times n}$ and the representation is found by attempting to solve

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|y - Dx\|_2^2 \\ \text{subject to} \quad & \|x\|_0 \leq s \end{aligned} \tag{1}$$

The representation vector $x \in \mathbb{R}^n$ is sparse, having at most s nonzero elements; its so called ℓ_0-norm (which is not properly a norm) is denoted $\|.\|_0$. Thus, the signal y is approximated in an s-dimensional subspace whose basis is made of s atoms of the dictionary (also called support). The dictionary is typically overcomplete, i.e. $p < n$, such that the number of s-dimensional subspaces is large and hence the approximation is better. The most popular method for approximately solving (1) in the dictionary learning context is Orthogonal Matching Pursuit (OMP) [3].

In the cosparse or analysis model, the overcomplete dictionary is $\Omega \in \mathbb{R}^{n \times p}$, with $n > p$, and the atoms are the rows of the dictionary. The representation, now denoted $z \in \mathbb{R}^p$, is orthogonal on a set $\mathcal{I}$ of $n - s$ atoms (named cosupport), so again it lies in an s-dimensional subspace, and the representation problem is

$$\begin{aligned} \underset{z,\mathcal{I}}{\text{minimize}} \quad & \|y - z\|_2^2 \\ \text{subject to} \quad & \Omega_{\mathcal{I}} z = 0 \\ & \text{rank}(\Omega_{\mathcal{I}}) = n - s \end{aligned} \tag{2}$$

where $\Omega_{\mathcal{I}}$ contains the rows of Ω with indices in $\mathcal{I}$.

We focus on dictionary learning (DL) in the sparse model. DL methods start with a given set of training signals $Y \in \mathbb{R}^{p \times m}$ and a sparsity level s and attempt to find a dictionary D by solving

$$\begin{aligned} \underset{D,X}{\text{minimize}} \quad & \|Y - DX\|_F^2 \\ \text{subject to} \quad & \|x_i\|_0 \leq s, \ \forall i \end{aligned} \tag{3}$$

where $X \in \mathbb{R}^{n \times m}$ is the associated s-sparse representations matrix, with at most s nonzero elements on each column and x_i is column i of matrix X. The atoms are constrained to unit norm. An overview of the field along with existing solutions and applications are presented in [1], [4]. The minimization from (3) is usually [5]–[7] solved with an alternation process in which one of X or D is optimized while keeping the other fixed.

In the general case the dictionary D is not structured in a specific way, but some methods [8], [9] approach the problem by building the redundant atoms set as a union of smaller orthogonal dictionaries also termed as orthonormal bases (ONBs) that transform the optimization problem (3) into:

$$\begin{aligned} \underset{Q_j,X}{\text{minimize}} \quad & \|Y - [Q_1 \ Q_2 \ \dots \ Q_M]X\|_F^2 \\ \text{subject to} \quad & \|x_i\|_0 \leq s, \ \forall i \\ & Q_j^T Q_j = I_p, \ 1 \leq j \leq M \end{aligned} \tag{4}$$

where the union of M ONBs denoted $Q_j \in \mathbb{R}^{p \times p}$, with $j = 1 \dots M$, represents the dictionary D.

Representative for this approach are the union of orthonormal bases algorithm (UONB) [8] and the single block orthogonal algorithm (SBO) [9], which use different methods for representation. The former makes use of atoms from all bases by OMP selection, while the later constrains the atoms pool to a single block. Even though SBO uses a single base for representation, the more bases it has in its union the greater the chance of finding a better fit that minimizes the error even further. SBO has to use more dictionary blocks than UONB in order to achieve the same representation error but nevertheless it is able to provide faster execution times.

The basic operation for both methods is the training of an orthogonal block, using an alternation process adapted to orthogonality. In particular, the atoms optimization is done through Procrustes orthogonalization [10]. In this paper we propose a new dictionary training algorithm for the orthogonal case, inspired from the cosparse DL method from [11]. The combination of techniques from both the sparse and cosparse approaches is the key to better representations and is possible due to the special characteristics induced by orthogonality.

The manuscript is structured as follows: section II describes our new cosparse orthonormal block training algorithm and its relation to the synthesis version, followed by numerical results supporting its dictionary recovery and representation error improvements in section III.

II. Cosparse orthonormal block training

We start with the simple remark that the sparse (1) and cosparse (2) representation problems have the same optimal error if the dictionary is orthogonal. Indeed, given D orthogonal, the problem (1) is solved by computing $x = D^T y$ and keeping only the largest (in absolute value) s elements, the others being forced to zero. This holds because the objective of (1) is equal to $\|D^T y - x\|_2^2$. For the cosparse problem (2), the atoms are now rows instead of columns, so the dictionary is $\Omega = D^T$. The two problems are connected via the relation $D^T z = x$. The $n - s$ atoms that are orthogonal on z are those corresponding to the positions of zeros in x. Otherwise said, the problem (2) is solved by computing $D^T y$ and setting to zero the $n - s$ smallest elements (in absolute value). We work on complementary subspaces, but the final result is the same.

A. Building one orthonormal block

UONB [8] and especially SBO [9] use 1ONB (algorithm 1) to build one orthonormal block. 1ONB starts with an orthogonal matrix based on the SVD decomposition of the training signal set Y (step 1 in algorithm 1). It then proceeds to represent Y with the new block Q by performing a hard-thresholding on the largest s entries in absolute value of the resulting matrix X (step 3). The dictionary block is then refined through an orthogonal approximation of the training signals Y and the representations X (step 4). This approximation process is also called Procrustes orthogonalization and its computation

Algorithm 1: 1ONB

Data: signals set Y,
target sparsity s,
number of rounds R
Result: dictionary Q and sparse representations X

1 **Initialization:** Let $Q = U$ where $U\Sigma V^T = \text{SVD}(Y)$
2 **for** $r \leftarrow 1$ **to** R **do**
3 **Update:** $X = Q^T Y$ and select the largest s entries of each column setting the others to zero
4 **Approximation:** apply Procrustes orthogonalization (5) on Y and X to approximate Q

steps, given X and Y, are as follows:

$$\begin{aligned} P &= YX^T \\ U\Sigma V^T &= \text{SVD}(P) \\ Q &= UV^T \end{aligned} \tag{5}$$

In order to further improve the dictionary block, steps 3 and 4 are repeated several times as dictated by the number of rounds R in step 2.

B. Building cosparse orthonormal blocks

Using the idea behind 1ONB and concepts inspired from cosparse DL, we propose a new method for training an orthogonal block, described in algorithm 2. Since the sparse and cosparse models are interchangeable in the orthogonal case, as explained above, we adopt an idea used for atom update in the cosparse K-SVD algorithm [11]. Denoting Q the orthogonal dictionary, an atom q_i is optimized by solving the problem

$$\begin{aligned} \underset{q_i}{\text{minimize}} \quad & \|q_i^T Y_{\mathcal{I}}\|_2^2 \\ \text{subject to} \quad & \|q_i\|_2 = 1 \end{aligned} \tag{6}$$

where $\mathcal{I}$ is the set of signals that do not use the atom q_i in their representation (or, taking the cosparse view, on which q_i should ideally be orthogonal). The objective function is the total orthogonality error between the current atom and the vectors it should be orthogonal on. The solution of (6) is the singular vector corresponding to the smallest singular value of $Y_{\mathcal{I}}$. (Note the duality with sparse K-SVD, where the singular vector of the largest singular value was involved.)

We give now a step-by-step description of algorithm 2, named 1ONB-COSP in the sequel. The initialization of the orthoblock Q in step 1 and the computation of the sparse representations X in step 2 are done the same way as described in algorithm 1. Following the general approach for atom optimization in DL, we sequentially update each atom q_i from Q in the loop from step 3, using the atom refinement solution described in (6): first we extract the signals that are not using the current atom i in their representation (step 4) and then we proceed to refine q_i in step 5 by solving (6). We found that updating the representations immediately after the change in atom q_i significantly improves the final representation error.

Algorithm 2: 1ONB-COSP

Data: signals set Y and target sparsity s
Result: dictionary Q and sparse representations X

1 **Initialization:** Let $Q = U$ where $U\Sigma V^T = \text{SVD}(Y)$
2 Compute $X = Q^T Y$ and select the largest s entries of each column
3 **foreach** *atom i in dictionary Q* **do**
4 **Extract:** $\mathcal{I} = \{j | X_{i,j} = 0\}$
5 **Refine:** solve (6) to get q_i
6 **Update:** $X = Q^T Y$ and select the largest s entries of each column
7 **Restructure:** apply Procrustes approximation (5) on Y and X to orthogonalize Q

Algorithm 3: 1ONB-COSP+

Data: signals set Y, sparsity s, rounds R
Result: dictionary Q and sparse representations X

1 $\{Q, X\} = \text{1ONB-COSP}(Y, s)$
2 **for** $r \leftarrow 1$ **to** R **do**
3 **Update:** $X = Q^T Y$ and select the largest s entries of each column
4 **Approximation:** apply Procrustes orthogonalization (5) on Y and X to approximate Q

So, in step 6 we create new representations with the updated dictionary Q the same way we did in step 2. Note that, with the replacement of q_i, the dictionary Q is no longer orthogonal and it is important to use it unstructured when updating the representations in step 6. We restructure Q as an orthogonal matrix right before proceeding to the next atom update by applying (5) in step 7. Numerical simulations showed that it is better to use the old representations built in step 6 in the next atom update iteration instead of computing new ones with the restructured dictionary from step 7, although this may be counterintuitive.

Updating each individual atom at a time (step 3) shows an increased complexity of 1ONB-COSP when compared to 1ONB that updates the entire dictionary at once. This might partly explain why 1ONB needs a few refinement rounds (e.g. $R = 5$ or $R = 6$ in step 2) until error improvement stalls [9], whereas for 1ONB-COSP our simulations showed that a single refinement of each atom is enough and repeating the dictionary training steps 3–7 does not improve the final quality of the orthoblock Q.

We also found that further error improvement appears if, as described in algorithm 3, 1ONB-COSP (step 1) is followed by R 1ONB training rounds (steps 2–4). We denote 1ONB-COSP+ this succession of algorithms. On the contrary, running 1ONB first and then performing 1ONB-COSP did not show any improvement in the end result.

TABLE I
PERCENTAGE OF RECOVERED ATOMS

s	Method	SNR			
		10	20	30	∞
3	1ONB	46.7	53.5	57.4	53.8
	1ONB-COSP	99.9	100.0	100.0	91.9
	1ONB-COSP+	100.0	100.0	100.0	99.4
4	1ONB	15.5	30.9	28.9	28.8
	1ONB-COSP	96.7	99.1	98.8	89.8
	1ONB-COSP+	98.2	99.8	99.4	97.8
5	1ONB	2.3	9.1	12.6	11.1
	1ONB-COSP	85.4	91.5	95.8	90.2
	1ONB-COSP+	91.5	95.2	98.0	95.8

III. RESULTS AND PERFORMANCE

We present numerical results indicating the quality improvements when using the cosparse approach. First we show the benefits of 1ONB-COSP and 1ONB-COSP+ for the dictionary recovery and the sparse image representation problems when using a single orthogonal block. Then, we present its impact on the algorithms that make use of it for DL, when the dictionary is a union of orthogonal blocks. We always used identical input and parametrization (where applicable) when comparing methods.

A. Dictionary recovery

We started with a random square matrix of dimension $p = 20$ on which we ran the SVD decomposition and used the left orthogonal transformation matrix as the original dictionary. We then generated a data set Y of $m = 600$ columns, each obtained as a linear combination of $s \in \{3, 4, 5\}$ randomly chosen atoms. We perturbed Y by adding white gaussian noise of 10, 20, 30 and ∞ dB SNR levels. We ran 1ONB and 1ONB-COSP+ with $R = 5$ rounds (enough to converge as described in [9]) on the new signal set and compared the original dictionary with the learned dictionary; two atoms, one from the former, the other from the later dictionary, are considered (nearly) identical if their scalar product is larger than 0.99 in absolute value. The algorithms were provided with the original sparsity level s that was used in generating the clean data set Y. Table I shows a big improvement in the percentages of recovered atoms, averaged over 50 runs. 1ONB-COSP is vastly superior to 1ONB. 1ONB-COSP+ improves the results where there is room for improvement, especially for larger s.

B. Dictionary learning

We generated the training data set with images from USC-SIPI [12]. Each image was normalized and organized into 8×8 random patches.

In figure 1 we present the average representation error over 100 runs for varying signals set sizes when using orthogonal blocks of dimension $p = 64$ ($n = 64$ atoms) with a sparsity constraint of $s = 8$. We used $R = 5$ rounds for 1ONB and 1ONB-COSP+. Both cosparse methods are consistent in providing a better dictionary than plain 1ONB. More so, at the

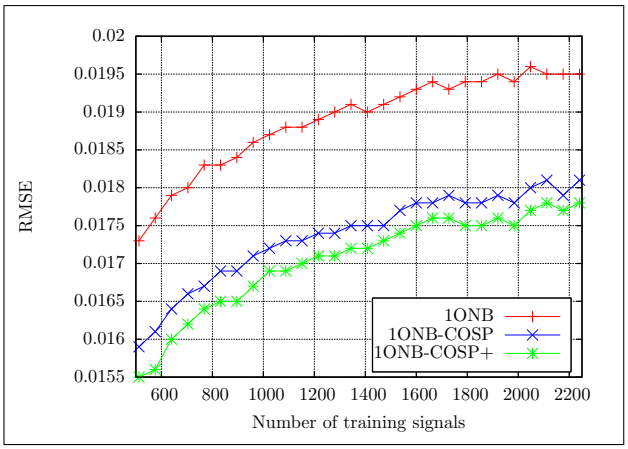

Fig. 1: Error evolution for sparse and cosparse algorithms.

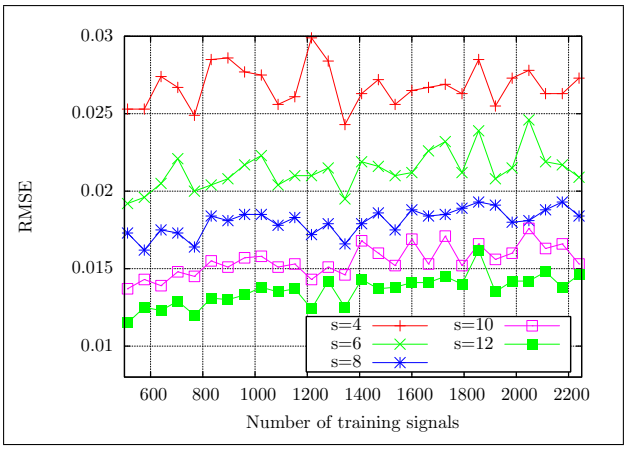

Fig. 2: Error evolution for different sparsity constraints.

TABLE II
FINAL ERRORS FOR SPARSE AND COSPARSE ALGORITHMS

s	m	1ONB	COSP	COSP+
4	512	0.0288	0.0280	0.0268
	1024	0.0292	0.0283	0.0275
	1536	0.0293	0.0278	0.0276
	2048	0.0304	0.0295	0.0293
6	512	0.0239	0.0236	0.0217
	1024	0.0243	0.0231	0.0230
	1536	0.0245	0.0243	0.0230
	2048	0.0257	0.0244	0.0242
8	512	0.0204	0.0185	0.0183
	1024	0.0211	0.0198	0.0189
	1536	0.0209	0.0196	0.0192
	2048	0.0224	0.0223	0.0205
10	512	0.0176	0.0158	0.0155
	1024	0.0183	0.0171	0.0167
	1536	0.0183	0.0171	0.0168
	2048	0.0195	0.0191	0.0184
12	512	0.0152	0.0151	0.0139
	1024	0.0159	0.0173	0.0154
	1536	0.0160	0.0163	0.0153
	2048	0.0170	0.0165	0.0160

cost of an increase in execution time, 1ONB-COSP+ performs better than 1ONB-COSP.

Table II shows the final errors after running a single round of tests on $p = 64$ sized dictionary blocks with varied sparsity constraints $s \in \{4, 6, 8, 10, 12\}$ and different training set sizes ($m \in \{512, 1024, 1536, 2048\}$) for each sparsity level. Except for two results ($s = 12$), 1ONB-COSP presents an improvement in approximation error over 1ONB, while 1ONB-COSP+ always outperforms both methods.

Figure 2 shows the sparsity impact on the representations obtained with algorithm 2. Using the same dimensions for the input data as the ones used in figure 1 we changed the sparsity constraint from $s = 4$ up to $s = 12$ for 1ONB-COSP while keeping the same training signal set. We can see a natural increase in error as the number of signals in the data set grows. Also visible is the clear difference in representation quality as the sparsity constraint is loosened.

C. Unions of orthonormal bases with cosparse training

Algorithms that train overcomplete dictionaries as a union of orthonormal bases (such as UONB [8] and SBO [9]) use 1ONB to build one orthonormal block. UONB uses OMP for representation, without taking any special advantage from the structure of the dictionary. In the dictionary initialization stage, we train the orthobases with 1ONB using distinct subsets of the training signals. However, 1ONB cannot be fully used in the iterative training of a block, due to the fact that atoms from all blocks can appear in representation. On the other hand, since it uses a single block for representation, SBO can appeal to the simple representation algorithm described above for orthogonal dictionary. SBO starts with a few 1ONB trained orthobases and then adds new bases to the existing union, that are also 1ONB trained with the set of the worst represented signals. Finally the extended union is refined by applying 1ONB on each dictionary block. The last two steps are repeated until a stopping criterion is met.

To show how our cosparse approach behaves we substitute 1ONB with 1ONB-COSP or 1ONB-COSP+ in the dictionary initialization and update stage of SBO and UONB, without any other algorithmic modifications. Our goal here is to improve the performance of SBO and UONB, for a comparison with generic DL methods such as AK-SVD we direct the reader to the numeric simulations from [9].

Following the comparison tests from [9] we used $M = 3$ orthobases for UONB and $M = 16$ for SBO (this makes representation speed similar for the two methods). The signals have size $p = 64$, being generated from images as in the previous subsection. We impose a sparsity constraint $s = 10$ and a number of $R = 5$ training rounds for 1ONB and 1ONB-COSP+. We ran multiple tests on varying signals set sizes from $m = 4096$ to $m = 8192$. We plot the results in figures 3 and 4.

Figure 3 shows the average representation error over 10 runs when performing SBO with 1ONB, 1ONB-COSP and 1ONB-COSP+. Because SBO makes use of 1ONB training during initialization and also during the main iterations, the approximation improvement is consistent with the results seen

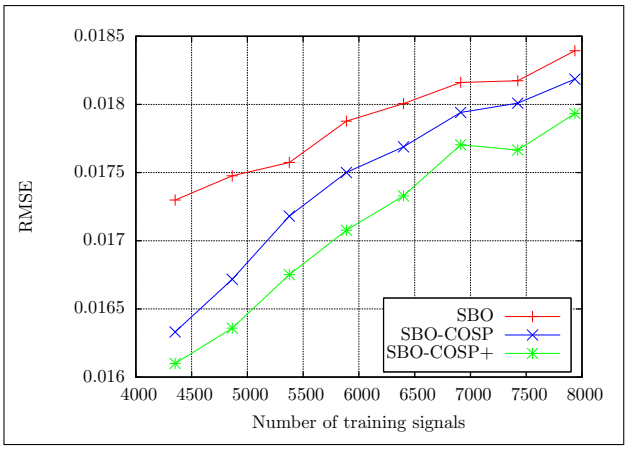

Fig. 3: Representation error comparison of SBO variants.

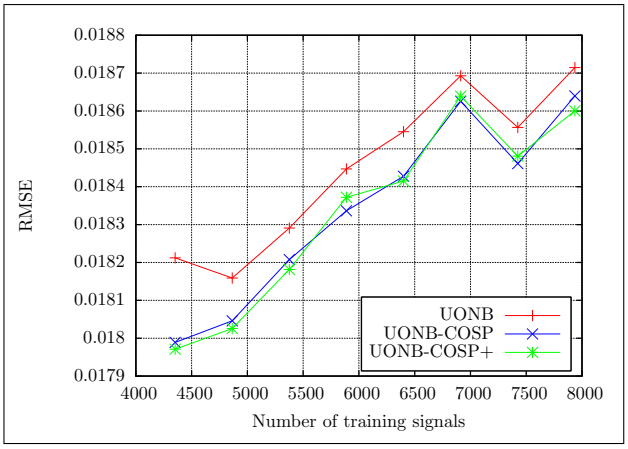

Fig. 4: Representation error comparison of UONB variants.

in figure 1.

Using the same average as described in figure 3, we show the performance of UONB with all three 1ONB variants in figure 4. Because 1ONB is only used at initialization the cosparse variants have less impact on the overall performance of UONB.

IV. Conclusions and Future Work

In this paper we have presented a new algorithm for learning orthogonal dictionary blocks in a cosparse fashion. The new algorithm shows significant improvements at recovering dictionary atoms and provides a smaller representation error when tested on synthetic and empirical data. We also show that the improvement in representation holds when applying the cosparse algorithms within existing methods that create the dictionary as a union orthonormal bases.

In the future we plan on finding a method that takes advantage of the orthogonal block cosparse training when learning multiple orthoblock dictionaries.

V. Acknowledgement

The authors thank Cristian Rusu for interesting discussions and suggestions on the subject.

This work was supported by the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, project number PN-II-ID-PCE-2011-3-0400. P. Irofti was also supported by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132395.

References

[1] R. Rubinstein, A.M. Bruckstein, and M. Elad, "Dictionaries for Sparse Representations Modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, June 2010.

[2] S. Nam, M.E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[3] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 1–3.

[4] I. Tosic and P. Frossard, "Dictionary Learning," *IEEE Signal Proc. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[5] K. Engan, S.O. Aase, and J.H. Husoy, "Method of optimal directions for frame design," in *IEEE Int. Conf. Acoustics Speech Signal Proc.*, 1999, vol. 5, pp. 2443–2446.

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

[7] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," *Technical Report - CS Technion*, 2008.

[8] 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. Proceedings. (ICASSP '05). IEEE International Conference on*, March 2005, vol. 5, pp. v/293–v/296 Vol. 5.

[9] C. Rusu and B. Dumitrescu, "Block orthonormal overcomplete dictionary learning," in *21st European Signal Processing Conference*, 2013, pp. 1–5.

[10] J.C. Gower and G.B. Dijksterhuis, *Procrustes problems*, vol. 3, Oxford University Press Oxford, 2004.

[11] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *Signal Processing, IEEE Transactions on*, vol. 61, no. 3, pp. 661–677, 2013.

[12] A.G. Weber, "The USC-SIPI Image Database," 1997.