Sparse Denoising with Learned Composite Structured Dictionaries

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

***Abstract*—In the sparse representation field recent studies using composite dictionaries have shown encouraging results in performing noise removal. In this paper we look at dictionary composition in the particular case of dictionaries structured as a union of orthonormal bases. Our study focuses on denoising performance, providing new algorithms that outperform existing solutions, and also speed, resulting in different algorithms that execute a lot faster with a negligible denoising penalty.**

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

I. Introduction

The sparse representation field has seen a lot of interest in the last decade with numerous signal processing applications in various domains such as compression, detection, recovery and denoising. In this article we focus on the algorithms dedicated to noise removal through dictionary learning (DL).

Most dictionary methods attack the learning problem [1], [2] through an iterative shrinkage process that alternatively improves the representations and the dictionary by fixing one while the other is refined. Given a training signal set $Y \in \mathbb{R}^{p\times m}$ this dual optimization problem can be formally expressed as:

$$\begin{aligned} \underset{D,X}{\text{minimize}} \quad & \|Y - DX\|_F^2 \\ \text{subject to} \quad & \|X(i)\|_0 \leq s,\ \forall i \end{aligned} \tag{1}$$

where $D \in \mathbb{R}^{p\times n}$ is the dictionary whose columns are also called atoms, $X \in \mathbb{R}^{n\times m}$ are the sparse representations and $X(i)$ are the columns from X, each bound to have no more than s non-zero elements. We denote with $\|.\|_F$ the Frobenius norm and with $\|.\|_0$ the total number of non-zero entries from a given vector (also known as the l_0 norm). Solving this leads to an approximation of the form:

$$Y = DX + R \tag{2}$$

where we denote the residual as R.

Given a set of signals Y that has been perturbed by a standard white gaussian noise Z, with noise level σ, and denoting with Y_c the unknown clean signals set that we want to recover, we have:

$$Y = Y_c + \sigma Z \tag{3}$$

And so, the goal of a DL denoising algorithm is to model Y_c such that the residual R from (2) matches the added noise:

$$Y_c \approx DX, R \approx \sigma Z \tag{4}$$

There are two approaches when choosing the training set for denoising with DL. The first one uses an external signals set [3] that results in a versatile set of representation atoms making the dictionary suited for a larger class of signals with the risk of running in to cases where none of the atoms are able to properly fit a specific model. The second approach focusses on training the dictionary with the same internal set [4] (or similar sets) of signals as the ones we want to denoise. This has the advantage of giving sharper results as the atoms are more specialized with the downside of a rather poor performance at high noise levels where the noise can become a participating part of the modeling atoms [5].

A recent trend in the DL community [6] [7] is taking advantage of both worlds by creating a dictionary following each training approach and merging the two into a larger dictionary with which the actual denoising is performed. First a dictionary is obtained from an external training set, using an algorithm such as K-SVD [8], and then for each new signal set an extra dedicated dictionary is trained and composed with the former [7]. Then the new signal set is sparsely represented by selecting atoms from both dictionaries through algorithms such as orthogonal matching pursuit (OMP) [9]. These composite dictionaries have been shown to outperform the vanilla approach [7]. The disadvantage with this way of doing things is the loss of generality of the DL algorithms due to the fact that the composite representations are now tightly coupled to the second specialized dictionary which is not a problem for applications such as image denoising but would be, for example, suboptimal for compression. The algorithmic and implementation details of K-SVD and its approximate version AK-SVD can be found in [10].

In this paper we study the composite approach when applied to the class of dictionaries structured as a union of orthonormal basis (UONB) [11]:

$$\begin{aligned} \underset{D,X}{\text{minimize}} \quad & \|Y - [Q_1 Q_2 \dots Q_K]X\|_F^2 \\ \text{subject to} \quad & \|X(i)\|_0 \leq s,\ \forall i \\ & Q_j^T Q_j = I_p,\ 1 \leq j \leq K \end{aligned} \tag{5}$$

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
4 **Approximation:** apply Procrustes orthogonalization (7) on Y and X to approximate Q

where we denote with Q a single orthonormal base and with K the total number of bases in the dictionary; and provide new algorithms based on the single orthonormal block algorithm (SBO) [12] that improve the quality of the denoised signals while also providing smaller execution times.

The manuscript is structured as follows: in section II we present the strategy of applying dictionary composition to structured dictionaries and explain the resulting algorithms, while in section III we provide numeric simulations to support our results, with conclusions and future research in section IV.

II. Composite structured dictionaries

SBO performs dictionary refinement independently on each block through 1ONB [11] depicted in algorithm 1.

The dictionary is initialized in step 1 with an orthonormal matrix created from the singular value decomposition (SVD) of the training set Y. Then, keeping the dictionary fixed, the representations are computed in step 3 though hard-thresholding the top s absolute values of the entries of each column from X. This can be implemented through a partial selection algorithm:

$$x = SELECT(Q^T y, s) \tag{6}$$

where x is the s sparse representation of signal y through base Q. Fixing the representations, the dictionary is refined in step 4 by performing the following orthogonal approximation of X and Y:

$$\begin{aligned} P &= Y X^T \\ U\Sigma V^T &= \text{SVD}(P) \\ Q &= U V^T \end{aligned} \tag{7}$$

which is known as Procrustes orthogonalization. This process is repeated R times, where 4–5 iterations have been empirically shown to be enough [12].

SBO signal representation is performed by choosing a different single best orthonormal base for representing each signal from the training set. In order to find the best orthobase we need to compute the sparse representations $x^{(i)}$ of a signal y with every orthobase Q_i (with $i = 1 \dots K$) from the dictionary and pick the one where the representation energy is highest. As shown in [12], computing the energy of the representations suffices because the norm is preserved by the orthonormal dictionary blocks. And so, we formalize finding the best representation base j with:

$$j = \underset{i=1\dots K}{\text{argmax}} \sum_{t=1}^{s} |x^{(i)}(t)|^2 \tag{8}$$

where $x^{(i)} = \text{SELECT}(Q_i^T y, s)$ and t iterates the s non-zero elements of $x^{(i)}$. We then proceed to represent y solely with the j-th base. This leads to a splitting of the signal set in K parts corresponding to each orthobase.

Starting with a few initial bases, SBO expands its dictionary at each iteration by adding a new orthorblock trained through 1ONB on a given percentage of the worst represented signals. After a new base is created, the signals set is repartitioned as described earlier around (8) and the entire union is refined by applying 1ONB on each dictionary block. This expansion continues until a stopping criterion is met.

Even though SBO has larger memory requirements due to an increased dictionary size, its advantage over AK-SVD is the training and represntation speed while maintaining a competitive approximation quality. To give a general impression of the differences between the two we compared their efficient parallel GPU implementations termed Parallel SBO (P-SBO) and Parallel AK-SVD (PAK-SVD) [13]. We measured the time it takes to perform dictionary learning (t_{learn}), the sparse representation time (t_{rep}) and the quality of the representations measured as the root mean squared error (RMSE). Dictionary sizes for both methods were picked in accordance with the comparison tests performed by the authors of SBO [12]. The results for PAK-SVD running with full atom parallelism on a training set of m signals of size p with a sparsity of s for K iterations and varied dictionary sizes n can be seen in table I. We performed training on the same data set with P-SBO starting with K_0 initial bases with R 1ONB rounds and varied the dictionary size K. The results are shown in table II. We can see that there is a big difference between their execution time while, indeed, the representation quality is about the same.

TABLE I
PAK-SVD performance for $m = 32768$, $p = 64$, $s = 8$ and $K = 100$.

n	64	96	128	160	256
$t_{learn}(s)$	366.8	396.7	416.5	438.4	642.4
$t_{rep}(s)$	0.3467	0.3753	0.8207	0.5889	2.2436
RMSE	0.0271	0.0246	0.0242	0.0230	0.0216

TABLE II
Parallel SBO performance for $m = 32768$, $p = 64$, $s = 8$ with $K_0 = 5$ and $R = 6$

K	8	16	24	32	64
$t_{learn}(s)$	1.8	6.7	12.3	20.9	85.4
$t_{rep}(s)$	0.0020	0.0021	0.0022	0.0021	0.0021
RMSE	0.0268	0.0245	0.0240	0.0238	0.0235

Algorithm 2: SBO-C1

Data: training signals set Y_t, noisy signals Y
Result: denoised signals Y_c

1 Train generic dictionary $E = \text{SBO}(Y_t)$
2 Train orthoblock with noisy set: $F = \text{1ONB}(Y)$
3 Assign each column k from Y to one orthobase $Q_E^{(k)}$ from E following (8)
4 **foreach** *signal k from Y* **do**
5   **Compose:** $D_c = [Q_E^{(k)} F]$
6   **Represent:** $X(k) = \text{OMP}(D_c, Y(k))$
7   **Denoise:** $Y_c(k) = D_c X(k)$

Algorithm 3: SBO-C

Data: training signals set Y_t, noisy signals Y
Result: denoised signals Y_c

1 Train external dictionary $E = \text{SBO}(Y_t)$
2 Train internal dictionary $F = \text{SBO}(Y)$
3 Assign each column k from Y to one orthobase $Q_E^{(k)}$ from E and one orthobase $Q_F^{(k)}$ from F following (8)
4 **foreach** *signal k from Y* **do**
5   **Compose:** $D_c = [Q_E^{(k)} Q_F^{(k)}]$
6   **Represent:** $X(k) = \text{OMP}(D_c, Y(k))$
7   **Denoise:** $Y_c(k) = D_c X(k)$

A. Composing SBO with 1ONB

Our first proposal for denoising with composite structured dictionaries is to mix an SBO trained external dictionary with a specific orthoblock trained with 1ONB on the noisy set. We term this method SBO-C1 and describe it in algorithm 2.

We first use SBO to train the external dictionary E on the training set Y_t and 1ONB to train the internal dictionary F on the noisy set Y (steps 1 and 2). Next we allocate each signal from Y to a block from E (step 3). We denote a column k from matrix Y with $Y(k)$ and the SBO base it has been assigned to with $Q_E^{(k)}$. The focal point in the algorithm is composing a dedicated dictionary D_c for each signal made out of its assigned block $Q_E^{(k)}$ and the internal orthoblock F (step 5). With the composite dictionary we proceed to compute the sparse representations $X(k)$ using the OMP algorithm in step 6. The factorization of the composite dictionary and the representations from step 7 provides us with the denoised signal $Y_c(k)$ as described around equation (4).

The simplistic 1ONB training of the internal dictionary leads to mediocre denoising results (as described in section III), but it is worth mentioning that it manages to perform better than the composite AK-SVD algorithm when the sparsity constrained is loosened such that $s > \sqrt{p}$. In terms of speed, it offers the fastest dictionary training phase (steps 1 and 2), but representation and denoising (steps 6 and 7) take just as long as it would with an identically sized dictionary trained with plain or composite AK-SVD.

B. Composite SBO

A natural step towards improving the representation quality of SBO-C1 is to expand the internal dictionary to more than one orthoblock. This can be achieved by performing another SBO session, this time on the noisy set, in order to create an extended internal dictionary that is still smaller than the external one. We keep the internal dictionary small in order to avoid modelling the noise as described in the introduction. Our experiments have shown good results with keeping a size of half the number of bases from the external dictionary. This algorithm is a direct corespondent of the composite AK-SVD method. We call it SBO-C and describe it in algorithm 3.

In the new algorithm, step 2 is modified in order to train an internal SBO dictionary instead of a single 1ONB base. This also affects step 3 where an extra assignment operation needs to be performed for the internal dictionary. We denote with $Q_F^{(k)}$ the base from the internal dictionary assigned to signal k from the noisy set Y. The composed dictionary (step 5) maintains its size but its SBO internal dictionary component leads to a more specialized orthobase $Q_F^{(k)}$ and provides sharper results. Representation and denoising (steps 6–7) do not take advantage of the composite structure of the dictionary and so they perform the same as they would with plain or composite AK-SVD.

Even though the training stage (steps 1–3) has an increased complexity due to the second SBO training round, it is still a lot faster than plain and composite AK-SVD which suffer from large execution times as shown in table I.

C. Hybrids

In our pursuit of improving denoising performance, we also studied the case of hybrid dictionary compositions between AK-SVD and SBO.

One option is to use SBO as the external dictionary and train an AK-SVD block instead of the 1ONB base in algorithm 2. The rest of SBO-C1 would remain the same with the observation that the composite dictionary is now done with F learned from AK-SVD in step 5. We found that preserving the block size for the AK-SVD dictionary is enough to outperform SBO-C1. This also helps prevent noise modelling and keeps a minimal execution time. The effects of increasing the dictionary size can be observed in the first line of table I.

Reversing the roles, we can train AK-SVD on the external data set and then use SBO on the internal noisy sets as described in algorithm 4.

This way the expensive AK-SVD training is performed on the large training set (step 1) and SBO on the small noisy set (step 2). The orthobase assignment needs to be performed only on the internal dictionary in step 3 resulting in a specialized orthobase $Q_F^{(k)}$ as described in subsection II-B. The composed dictionary would then be made of the AK-SVD external dictionary and the assigned internal orthobase (step 5). Representation (step 6) and denoising (step 7) perform

Algorithm 4: Composite AK-SVD with SBO

Data: training signals set Y_t, noisy signals Y
Result: denoised signals Y_c

1 Train external dictionary $E = \text{AK-SVD}(Y_t)$
2 Train internal dictionary $F = \text{SBO}(Y)$
3 Assign each column k from Y to one orthobase $Q_F^{(k)}$ from F following (8)
4 **foreach** *signal k from Y* **do**
5   **Compose:** $D_c = [E\ Q_F^{(k)}]$
6   **Represent:** $X(k) = \text{OMP}(D_c, Y(k))$
7   **Denoise:** $Y_c(k) = D_c X(k)$

the same steps as former algorithms and have an identical execution cost as plain or composite AK-SVD.

Algorithm 4 would outperform a composite AK-SVD solution in an online denoising scenario where external training, representation and denoising would be identical but the internal training for the hybrid would perform much faster in production with an efficient SBO implementation handling incoming noisy data instead of another AK-SVD instance.

III. Results and performance

Our denoising experiments were performed on images from Volume 3 of the USC-SIPI [14] database. For the training set we picked ten arbitrary grayscale images normalized and organized into 8×8 random patches. The noisy set was built in the same way from a different image to which we added white gaussian noise in order to obtain the desired singnal to noise ratio (SNR). Denoising performance is expressed as:

$$\text{RMSE} = \frac{\|Y_c - DX\|_F}{\sqrt{pm}} \tag{9}$$

where Y_c is the clean image and DX is the factorization result of steps 7 from the algorithms in section II (see description around equation (4)).

We shorten AK-SVD with AK, composite AK-SVD with AK-C and the hybrid variants with S-AK where we used SBO as the external dictionary and AK-SVD as the internal one and vice-versa as AK-S.

A. Composite dictionaries with $n = 64 + 64$

The first series of experiments constraints each composite dictionary to a fixed size of $n = 128$ with 64 atoms from the external dictionary and another 64 atoms from the internal dictionary.

In tables III–V we present the results of denoising at $s = \{4, 8, 12\}$ sparsity constraints. We used a set of $m = 8192$ signals randomly picked from 9 images for training the external dictionaries. The internal set was built from a single noisy image at different $N = \{10, 20, 30, 40, 50\}$ dB SNR levels. SBO was used with $K = 16$ bases when training external dictionaries and $K = 8$ when training internal dictionaries. 1ONB was always ran for $R = 5$ rounds as that is sufficient according to [12]. AK-SVD trained external and internal dictionaries with $n = 64$ atoms and $n = 128$ for the classic non-composite version. We followed the dictionary sizes used to compare the two methods in [12].

TABLE III
Denoising with $n = 64 + 64$ and $s = 4$

N	SBO	SBO-C1	SBO-C	AK	AK-C	S-AK	AK-S
10	0.0514	0.0498	0.0481	0.0473	**0.0462**	0.0465	0.0467
20	0.0490	0.0473	0.0439	0.0452	0.0441	0.0435	**0.0427**
30	0.0488	0.0468	0.0434	0.0449	0.0439	0.0434	**0.0421**
40	0.0487	0.0468	0.0434	0.0450	0.0439	0.0433	**0.0423**
50	0.0491	0.0468	0.0435	0.0450	0.0438	0.0433	**0.0422**

TABLE IV
Denoising with $n = 64 + 64$ and $s = 8$

N	SBO	SBO-C1	SBO-C	AK	AK-C	S-AK	AK-S
10	0.0441	0.0421	0.0412	0.0401	**0.0395**	0.0403	0.0402
20	0.0392	0.0363	0.0339	0.0357	0.0355	0.0341	**0.0329**
30	0.0389	0.0358	0.0333	0.0352	0.0350	0.0335	**0.0322**
40	0.0388	0.0356	0.0330	0.0350	0.0350	0.0334	**0.0322**
50	0.0386	0.0355	0.0330	0.0353	0.0351	0.0335	**0.0321**

As expected, when the sparsity constraint is loosened the denoising performance improves. The tables also show that the methods have a consistent performance across sparsity constraints and noise levels. The hybrid version AK-S is the winner in most cases except for the very noisy one where composite AK-SVD takes the lead. Even then we can see that the difference is not significant and both hybrid versions come in close in second place. If, on the other hand, speed requirements might justify a small performance compromise SBO-C is the best choice due to its small execution time at less than 3% performance loss when compared to AK-S. SBO-C is also more than 60x faster than AK-S as can be seen from tables I and II. At higher sparsity targets, such as the ones from tables IV and V, even SBO-C1 might provide a good option being the fastest method with a 10% performance penalty. Another interesting observation is that, except for the 10dB case, SBO-C is outperforming composite AK-SVD.

In figure 1 we show the denoising performance evolution of each method as the SNR drops. For this experiment we used the same data as that from the tables above and changed the sparsity to be $s = 10$. AK-S is a clear winner while plain SBO is the poorest denoiser. AK-SVD, AK-C and SBO-C1 are somewhere close in the middle presenting similar performances. Naturally, SBO-C outperforms the hybrid S-AK due to a larger internal dictionary.

B. Composite dictionaries with $n = 128 + 64$

We present here a second experiment where we increased the size of the external dictionaries to $n = 128$ and maintained the internals at $n = 64$ leading to a fixed composite dictionary size of $n = 192$. This experiment was aimed at comparing the plain and composite AK-SVD variants with our hybrid

TABLE V

DENOISING WITH $n = 64 + 64$ AND $s = 12$

N	SBO	SBO-C1	SBO-C	AK	AK-C	S-AK	AK-S
10	0.0400	0.0380	0.0377	0.0364	**0.0361**	0.0372	0.0371
20	0.0328	0.0291	0.0275	0.0303	0.0303	0.0284	**0.0272**
30	0.0319	0.0281	0.0262	0.0293	0.0297	0.0275	**0.0258**
40	0.0318	0.0279	0.0261	0.0293	0.0295	0.0273	**0.0258**
50	0.0317	0.0280	0.0260	0.0294	0.0297	0.0271	**0.0259**

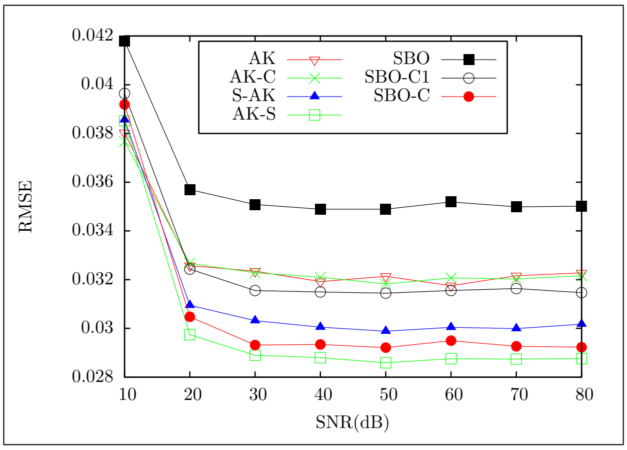

Fig. 1: Denoising with $n = 64 + 64$ and $s = 10$.

TABLE VI

DENOISING WITH $n = 128 + 64$ DICTIONARIES

s	N	SBO-C	AK	AK-C	S-AK	AK-S
4	10	0.0480	0.0462	**0.0454**	0.0466	0.0455
	20	0.0440	0.0439	0.0431	0.0436	**0.0413**
	30	0.0435	0.0437	0.0429	0.0433	**0.0410**
8	10	0.0412	0.0388	**0.0385**	0.0403	0.0391
	20	0.0339	0.0343	0.0341	0.0342	**0.0316**
	30	0.0331	0.0337	0.0336	0.0335	**0.0309**
12	10	0.0378	0.0355	**0.0354**	0.0371	0.0360
	20	0.0273	0.0282	0.0286	0.0282	**0.0257**
	30	0.0262	0.0273	0.0279	0.0273	**0.0244**

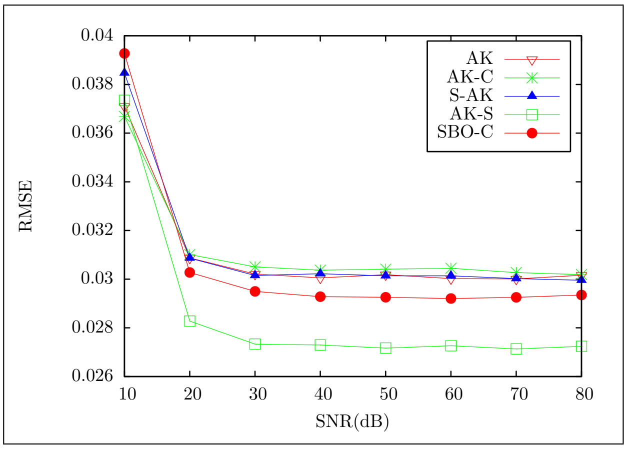

Fig. 2: Denoising with $n = 128 + 64$ and $s = 10$.

proposals. We used the same data as in our first experiment, the only change here is the dictionary size. For reference we also ran SBO-C on the same input data even though its composite dictionary size is limited to $n = 128$.

Table VI is structured the same way as tables III–V with a reduced number of noise levels $N = \{10, 20, 30\}dB$ because higher SNRs showed identical results to $N = 30dB$. As we can see AK-C is still the best performer in the worst case scenario, but AK-S is coming close in 2nd place. The rest of the cases present AK-S as the clear winner at all sparsity levels. It is interesting to see that SBO-C is the 2nd runner up for larger sparsity values ($s = \{8, 12\}$) with AK-C taking the 2nd place at $s = 4$.

Figure 2 shows the denoising performance as the SNR improves. We can clearly see here that AK-S is maintaining its first place position with SBO-C coming in second and AK, AK-C and AK-S fighting for third place. We can also see the plateau past the 30dB mark that allowed us to resume table VI to the three SNR levels.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we studied and proposed 4 new algorithms for denoising using composite dictionaries structured as a union of orthonormal bases. SBO-C1 is the fastest algorithm providing a composition between SBO dictionaries and 1ONB blocks. SBO-C is an extended version that instead composes two SBO dictionaries offering improved denoising results while being more than 60x faster than the best performing algorithm with a penalty on the denoising quality of less than 3%. We also investigated the hybrid case where we mix a generic dictionary built with AK-SVD with a structured SBO dictionary. The hybrid case provided the best results in our experiments performed on grayscale images with a significant impact on the execution time due to the use of AK-SVD.

In the future we plan on researching ways in which we can adapt the representation stage to take advantage of the composite structure of the dictionary.

V. ACKNOWLEDGEMENT

This work was supported by the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, project number PN-II-ID-PCE-2011-3-0400 and 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] I. Tosic and P. Frossard, "Dictionary Learning," *IEEE Signal Proc. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[3] 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, 2006.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.

[5] C.Y. Yang, J.B. Huang, and M.H. Yang, "Exploiting self-similarities for single frame super-resolution," in *Computer Vision–ACCV 2010*, pp. 497–510. Springer, 2011.

[6] L. Le Magoarou and R. Gribonval, "Chasing butterflies: In search of efficient dictionaries," in *Acoustics, Speech, and Signal Processing, 2015. Proceedings. (ICASSP '15). IEEE International Conference on*, 2015, pp. 1–5.

[7] Z. Wang, Y. Yang, J. Yang, and T.S. Huang, "Designing A composite dictionary adaptively from joint examples," *CoRR*, vol. abs/1503.03621, 2015.

[8] M. Aharon, M. Elad, and A.M. 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.

[9] 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.

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

[11] 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.

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

[13] P. Irofti and B. Dumitrescu, "GPU parallel implementation of the approximate K-SVD algorithm using OpenCL," in *22nd European Signal Processing Conference*, 2014, pp. 1–5.

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