Abstract—Compressed sensing (CS) offers a joint compression and sensing processes, based on the existence of a sparse representation of the treated signal and a set of projected measurements. Work on CS thus far typically assumes that the projections are drawn at random. In this paper, we consider the optimization of these projections. Since such a direct optimization is prohibitive, we target an average measure of the mutual coherence of the effective dictionary, and demonstrate that this leads to better CS reconstruction performance. Both the basis pursuit (BP) and the orthogonal matching pursuit (OMP) are shown to benefit from the newly designed projections, with a reduction of the error rate by a factor of 10 and beyond.

Index Terms—Basis pursuit (BP), compressed sensing (CS), mutual coherence, optimized projections, orthogonal matching pursuit (OMP), sparse and redundant representations.

I. INTRODUCTION

Consider a family of signals \( \{x_j\}_j \in \mathbb{R}^n \) known to have sparse representations over a fixed dictionary \( D \in \mathbb{R}^{m \times k} \). We typically (but not necessarily) assume that the dictionary is redundant, meaning that \( k > n \). Thus, we have that these signals can be described by

\[
\forall j, \quad x_j = D\alpha_j
\]

with \( \|\alpha_j\|_0 \leq T \ll n \) for all \( j \). The \( \ell^0 \)-norm used here simply counts the number of nonzeros in \( \alpha_j \).

Compressed sensing (CS) is an innovative and revolutionary idea that offers a joint sensing and compression processes for such signals [1]–[7]. Using a projection matrix \( P \in \mathbb{R}^{p \times n} \) with \( T < p \ll n \), CS suggests to represent \( x_j \) by \( p \) scalars given by

\[
y_j = Px_j.
\]

The original signal \( x_j \) can be reconstructed from \( y_j \) by exploiting the sparsity of its representation—i.e., among all possible \( \alpha \) satisfying \( y_j = P\alpha \), we seek the sparsest. If this representation coincides with \( \alpha_j \), we get a perfect reconstruction of the signal using (1). This reconstruction thus requires the solution of

\[
\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad y_j = P\alpha
\]

which is known to be NP-hard even for moderate sizes of the linear system in the constraint [8], [9]. Approximation techniques, known as pursuit algorithms, are deployed and proven to lead to the true result for very sparse solutions [10]–[12].

Work on CS thus far assumes that \( P \) is drawn at random, which simplifies its theoretical analysis, and also facilitates a simple implementation [1]–[7]. In this paper, we show that by optimizing the choice of \( P \) such that it leads to better coherence of the effective dictionary, a substantially better CS reconstruction performance is obtained, with both the basis pursuit (BP) [10] and the orthogonal matching pursuit (OMP) algorithms [11], [12].

In Section II, we provide the intuition behind CS, along with a statement of the main results in the literature regarding its expected performance, which are related to this work. Section III concentrates on a proposed iterative method for improving the projections based on the mutual coherence (as will be defined shortly) of the overall new dictionary. We present experimental results in Section IV and show the performance gain obtained with the optimized projections. As this work is the first to consider the design of the projections and as it approaches this problem indirectly by improving the mutual coherence, there is clearly a room for future work and improvements. Ideas on how to further extend this work are brought in Section V.

II. COMPRESSED SENSING—THE BASICS

We have described previously the core idea behind CS. The first question one must ask is why will it work at all? In order to answer this question, we need to recall the definition of the mutual coherence [13], [14].

Definition 1: For a dictionary \( D \), its mutual coherence is defined as the largest absolute and normalized inner product between different columns in \( D \). Formally, this reads

\[
\mu(D) = \max_{1 \leq i,j \leq k \text{ and } i \neq j} \frac{|d_i^T d_j|}{\|d_i\| \cdot \|d_j\|}.
\]

The mutual coherence provides a measure of the worst similarity between the dictionary columns, a value that exposes the dictionary’s vulnerability, as such two closely related columns may confuse any pursuit technique.

A different way to understand the mutual coherence is by considering the Gram matrix \( G = D^T D \), computed using the dictionary after normalizing each of its columns. The off-diagonal entries in \( G \) are the inner products that appear in (4). The mutual coherence is the off-diagonal entry \( g_{ij} \) with the largest magnitude.

Suppose that the signal \( x_0 \) has been constructed by \( x_0 = D\alpha_0 \) with a sparse representation. Suppose further that the following inequality is satisfied:

\[
\|\alpha_0\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(D)} \right).
\]
A fundamental set of results state the following[13]–[15].

1. The vector $\alpha_0$ is necessarily the sparsest one to describe $x_0$, i.e., it is the solution of

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad x_0 = D\alpha.$$  

(6)

2. The BP algorithm for approximating $\alpha_0$, which is solving the linear program

$$\min_{\alpha} ||\alpha||_1 \quad \text{s.t.} \quad x_0 = D\alpha.$$  

(7)

is guaranteed to find $\alpha_0$ exactly.

3. The OMP for approximating $\alpha_0$ is also guaranteed to succeed. The OMP is a greedy and sequential method that accumulates the nonzeros in $\alpha_0$ one at a time, while attempting to obtain the fastest decrease of the residual error $||x_0 - D\alpha||_2$.

Based on the previous points, suppose that the projection matrix $P$ has been chosen and we are to solve

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad y_0 = Px_0 = P\alpha.$$  

(8)

If the original representation satisfies the requirement

$$||\alpha_0||_0 \leq \frac{1}{2} \left(1 + \frac{1}{\mu(PD)} \right)$$  

(9)

then, necessarily, the original $\alpha_0$ is the solution of the problem posed in (8), both pursuit methods will manage to recover it perfectly, and thus reconstruct $x_0$ well. The requirement posed in (9) is typically more strict compared to the one posed in (5), and especially for $p \ll n$, because $\mu(PD)$ is typically larger than $\mu(D)$.

All the aforementioned implies that if $P$ is designed such that $\mu(PD)$ is as small as possible, this allows a wider set of candidate signals to reside under the umbrella of successful CS behavior. While this conclusion is true from a worst-case standpoint, it turns out that the mutual coherence as defined previously does not do justice to the actual behavior of sparse representations and pursuit algorithms’ performance. Thus, if we relax our expectations and allow a small fraction of signals with the same representation’s cardinality to fail, then values of $||\alpha_0||_0$ substantially beyond the above bound are still leading to successful CS. Considering the average performance of CS as a function of this cardinality, an “average” measure of coherence is more likely to describe its true behavior.

Another way to observe the over pessimism of the mutual coherence in (4) is the following: The most fundamental question in CS is “How many measurements are required for successful reconstruction”? Assuming that the cardinality of the representation $||\alpha_0||_0 = T$ is known, one needs at least $2T$ measurements to form a nonlinear set of $2T$ equations with $2T$ unknowns (the indices and values of the nonzero entries).

If we address this question from the point of view of the value of $\mu(PD)$, we are likely to find that $O(T^2)$ measurements are needed at least, losing much of the compressibility potential in CS.1 Replacing the measure $\mu(PD)$ with a parallel one that considers some sort of an average measure of coherence may do more justice to the conclusion about the required number of measurements.

Indeed, recent work has established that for a high success rate of CS, it is enough to use $O(T)$ measurements, with an appropriate coefficient (e.g., $C_{\text{rest}} \cdot \log(n) \cdot T$, as found in [4]). These results are typically accompanied by an assumption about the specific dictionary structure, the use of random projections, and considering an asymptotic case where the relative sizes grow to infinity. Needless to say, these works avoid the coherence measure or bypass it elegantly.

One might ask why consider the mutual coherence or its average descendants when optimizing $P$? If our goal is the optimization of CS performance, why not address it directly? While we could expect better results when addressing the CS performance directly, optimizing $P$ such that BP/OMP perform better turns out to be a very hard problem. In choosing to work with the coherence, we simplify the optimization, while accepting the unavoidable sacrifice in performance.

A similar and related question could be: why not use other simplified measures that hint to the CS performance? In that respect, as the CS theoretical work in [1]–[3] leans so strongly on the uniform uncertainly principle (UUP) in deriving their performance bound, why not use this tighter measure? Again, while true in principle, an attempt to work with UUP is expected to be much more complex. Optimizing with respect to (w.r.t.) the UUP implies considering all submatrices of $PD$ with $T$ columns and minimizing their condition number. Such a combinatorial measure is much harder to address, compared to the simple coherence. Still, future work may lead to an efficient harnessing of the UUP to the design of $P$.

III. OPTIMIZING THE PROJECTION MATRIX

In this section, we will consider a different mutual coherence, which reflects average behavior. We define it as follows.

**Definition 2:** For a dictionary $D$, its $t$-averaged mutual coherence is defined as the average of all absolute and normalized inner products between different columns in $D$ (denoted as $g_{ij}$) that are above $t$. Formally

$$\mu_t(D) = \frac{1}{1 \leq i,j \leq k \text{ and } i \neq j} \sum_{1 \leq i,j \leq k \text{ and } i \neq j} \frac{|g_{ij}|}{1 \leq i,j \leq k \text{ and } i \neq j} \cdot \left( |g_{ij}| \geq t \right).$$  

(10)

For $t = 0$, we obtain a simple average of the absolute entries of $G$. As the value of $t$ grows, we obtain that $\mu_t(D)$ grows and approaches $\mu(D)$ from below. Also, it is obvious from the definition that $\mu_t(D) \geq t$. In the optimization procedure we are about to describe, we will target this value and minimize it iteratively.

We mention once again that a different and more direct approach towards the design of the projection matrix would be its learning based on signal examples and tests involving the pursuit algorithm deployed. We believe that such a method is likely to lead to better performance compared to the method described here. Nevertheless, such a direct scheme is also expected to be far more complex and involved, and thus, its replacement with the optimization of $\mu_t(PD)$ is an appealing alternative.

Put very simply, our goal is to minimize $\mu_t(PD)$ with respect to $P$, assuming that the dictionary $D$ and the parameter $t$.
Objective: Minimize $\mu_t(\mathbf{PD})$ with respect to $\mathbf{P}$.

Input: Use the following parameters:
- $t$ or $t\%$ - coherence (fixed or relative) threshold,
- $\mathbf{D}$ - the dictionary,
- $p$ - number of measurements,
- $\gamma$ - down-scaling factor, and
- $\text{Iter}$ - number of iterations.

Initialization: Set $\mathbf{P}_0 \in \mathbb{R}^{p \times n}$ to be an arbitrary random matrix.

Loop: Set $q = 0$ (iteration counter) and repeat $\text{Iter}$ times:
1. Normalize: Normalize the columns in the matrix $\mathbf{P}_q \mathbf{D}$ and obtain the effective dictionary $\hat{\mathbf{D}}_q$.
2. Compute Gram Matrix: $\mathbf{G}_q = \hat{\mathbf{D}}_q \hat{\mathbf{D}}_q^T$.
3. Set Threshold: If mode of operation is fixed, use $t$ as threshold. Otherwise, choose $t$ such that $t\%$ of the off-diagonal entries in $\mathbf{G}_q$ are above it.
4. Shrink: Update the Gram matrix and obtain $\hat{\mathbf{G}}_q$ by
   $$
   \hat{y}_{ij} = \begin{cases} 
   \gamma g_{ij} & |g_{ij}| \geq t \\
   \gamma t \cdot \text{sign}(g_{ij}) & t > |g_{ij}| \geq \gamma t \\
   g_{ij} & \gamma t > |g_{ij}|
   \end{cases}.
   $$
5. Reduce Rank: Apply SVD and force the rank of $\hat{\mathbf{G}}_q$ to be equal to $p$.
6. Squared-Root: Build the squared-root of $\hat{\mathbf{G}}_q$, $\mathbf{S}_q^T \mathbf{S}_q = \hat{\mathbf{G}}_q$, where $\mathbf{S}_q$ is of size $p \times k$.
7. Update $\mathbf{P}$: Find $\mathbf{P}_{q+1}$ that minimizes the error $\|\mathbf{S}_q - \mathbf{PD}\|^2$.
8. Advance: Set $q = q + 1$.

Result: The output of the above algorithm is $\mathbf{P}_{\text{iter}}$.

Fig. 1. Numerical algorithm for optimizing the projection matrix $\mathbf{P}$.

are known and fixed. Since $\mu_t(\mathbf{PD})$ is defined via the entries of the Gram matrix, we propose an iterative scheme that includes transformations from and to the Gram matrix in every iteration. This algorithm is inspired by a similar approach adopted in [16] for the design of Grassmannian frames that minimize the mutual coherence of a desired dictionary. While the work in [16] considers $\mathbf{D}$ as an unknown to be built, we target $\mathbf{P}$ in the expression $\mu_t(\mathbf{PD})$, which adds more complications.

A slightly different mode of operation of the previous algorithm can be proposed, where $t$ varies from one iteration to another, by addressing at all times a constant fraction of the entries in the Gram matrix. For example, the value $t$ can be updated at each iteration such that it targets the top $20\%$ of the inner products. We will denote the average mutual coherence of the top $t\%$ by $\mu_{t\%}(\mathbf{PD})$, and, as will be shown in Section IV, it is this measure that we will work with. The algorithm for optimizing $\mathbf{P}$ with the previous two options is described in Fig. 1.

In this algorithm, we start with a random set of $p$ projections stored in the matrix $\mathbf{P}$. As our main objective is the reduction of the inner products that are above $t$ in absolute value (assuming the first mode of operation), the Gram matrix of the normalized effective dictionary is computed and these values are “shrunk” multiplying their values by $0 < \gamma < 1$. In order to preserve the ordering of the absolute entries in the Gram matrix, entries in $\mathbf{G}$ with magnitude below $t$ but above $\gamma t$ are “shrunk” by a smaller amount, using the function

$$
\hat{g}_{ij} = \begin{cases} 
\gamma |g_{ij}| & |g_{ij}| \geq t \\
\gamma t \cdot \text{sign}(g_{ij}) & t > |g_{ij}| \geq \gamma t \\
|g_{ij}| & \gamma t > |g_{ij}|
\end{cases}.
$$

This function is described graphically for $t = 0.5$ and $\gamma = 0.6$ in Fig. 2. For convenience, the functions $\hat{g}_{ij} = g_{ij}$ and $\hat{g}_{ij} = \gamma |g_{ij}|$ are also shown. As can be seen, the proposed function is embraced between these two, switching to the slower one at $t$.

The aforementioned shrinking operation causes the resulting Gram matrix to become full rank in the general case. Thus, the next steps are mending this by forcing a rank $p$ and finding the matrix $\mathbf{P}$ that could best describe the squared root of the obtained Gram matrix. Thus, Steps 1–4 in the algorithm are addressing the objective of the process—the reduction of $\mu_t(\mathbf{PD})$—and Steps 5–7 are responsible for the feasibility of the proposed new Gram matrix and the identity of the emerged projection matrix.

Regarding convergence properties, not much can be said in general. The overall problem is far from being convex and convergence is guaranteed only if $\gamma$ is chosen very close to $1$. However, as we show next, in practice, one can choose $\gamma \approx 0.5$ and still get convergence, and in fact, an accelerated one, compared to the use of higher values. Since the objective function can be evaluated after every iteration with almost no additional cost, this could be used for an automatic stopping of the algorithm (in case an ascent is started), and even for tuning the parameters dynamically from one iteration to another.

To illustrate the behavior of the proposed algorithm, we provide a demonstration of its results in Fig. 3. Considering a random dictionary (every entry is drawn from i.i.d. zero mean and unit variance Gaussian distribution) of size $200 \times 400$, we seek the best projection matrix containing 30 projections, such
that $\mu_t\{PD\}$ is minimized for $t = 0.2$. The initialization is a random matrix $P_0$ of size $30 \times 200$, built the same way as the dictionary.

We use several values of $\gamma$, from 0.55 to 0.95. In all cases, we obtain convergence and it is faster as $\gamma$ is smaller. The value of $\mu_t\{PD\}$ is by definition above $t$, but as can be seen, it gets smaller quite effectively.

Fig. 4 presents the histogram of the absolute off-diagonal entries of $G = D^T P^T PD$ (assuming normalized columns) before the optimization and after 50 iterations (using $\gamma = 0.5$ and $t = 0.2$). As can be seen, there is a marked shift towards the origin of the histogram after optimization, with an emphasis on the tail which represents the higher values. A similar effect is seen also in Fig. 5, which presents similar histograms, this time working with $\ell_0 = 40\%$. Thus, in this run, we target at every iteration the minimization of the average of the top 40% of the off-diagonal entries in the Gram matrix.

IV. COMPRESSED SENSING: EXPERIMENTAL RESULTS

It is now time to assess how the optimized projections perform in the CS setting. We should remind the reader that, in this paper, we assume that by optimizing $\mu_t\{PD\}$ w.r.t. $P$, one leads to more informative projections, which in turn leads to better CS performance. This link between $\mu_t$ and CS is yet to be theoretically analyzed and proven, and here, we limit our study to an empirical one. The proposed test includes the following steps.

Stage 1) Generate data: Choose a dictionary $D \in \mathcal{R}^{n \times k}$ and synthesize $N$ test signals $\{x_j\}_{j=1}^N$ by generating $N$ sparse vectors $\{\alpha_{j}\}_{j=1}^N$ of length $k$ each, and computing $\forall j, x_j = D\alpha_j$. All representations are to be built using the same low cardinality $||\alpha||_0 = T$.

Stage 2) Initial projection: For a chosen number of measurements $p$, create a random projection matrix $P$ and apply it to the signals, obtaining $\forall j, y_j = Px_j$. Compute the effective dictionary $\hat{D} = PD$.

Stage 3) Performance tests: Apply the BP and the OMP to reconstruct the signals by approximating the solution of

$$\hat{\alpha}_j = \arg \min_{\alpha} ||\alpha||_0 \ s.t. \ y_j = \hat{D}\alpha$$

and testing the error $||x_j - D\hat{\alpha}_j||_2$. Measure the average error rate—a reconstruction with a mean squared error above some threshold is considered as a reconstruction failure.

Stage 4) Optimize projections: Use the algorithm as described in Section III to optimize the projection matrix $P$.

Stage 5) Reevaluation of CS performance: Reapply the performance tests of the BP and the OMP as described previously, and see how the newly designed projections influence the CS behavior.

We have followed the previously described stages in the following experiments. The first experiment studies the performance of CS before and after the optimization of the projections, with BP and OMP, and for varying amounts of measurements. The second one studies the effect of the cardinality of the representations. The third studies the effect of the dimensionality...
on the CS performance, before and after optimization of projections. The fourth explores the role of the parameter $\ell\%$ on the CS performance. The fifth and last experiment refers to the structure (or, in fact, the lack of it) obtained in the optimized projections.

In the first experiment, we used a random dictionary of size $80 \times 120$, i.e., $n = 80$ and $k = 120$. Other options such as a redundant discrete cosine transform (DCT) dictionary, were tested as well, and found to lead to qualitatively the same results, and thus omitted. This size was chosen as it enables the CS performance evaluation in reasonable time. We generated $N = 100000$ sparse vectors of length $k = 120$ with $T = 4$ nonzeros in each. The nonzeros’ locations were chosen at random and populated with i.i.d. zero-mean and unit variance Gaussian values. These sparse vectors were used to create the example signals with which the CS performance was evaluated. CS performance was tested with varying values of $p$ in the range $[16, 40]$. The relative error rate was evaluated as a function of $p$ for both the BP and the OMP, before and after the projection optimization. The projection optimization (per every value of $p$) was done using up to 1000 iterations\footnote{The algorithm is stopped in case of an increase in the value $\mu E_k$.} using $\gamma = 0.95$ and varying $\ell\% = 20\%$. The results are shown in Fig. 6.

Each point in the shown graph represents an average performance, accumulated over a possibly varying number of experiments. While every point is supposed to present an average performance over $N = 100000$ examples, in cases where more than 300 errors were accumulated, the test was stopped and the average so far was used instead. This was done in order to reduce the overall test runtime. Another substantial speedup was obtained by replacing the BP direct test (which requires a linear programming solver) with a much faster alternative, as described in Appendix I.

As can be seen and as expected, the results of both pursuit techniques improve as $p$ increases. In this test, the BP performs much better than the OMP. The optimized projections are indeed leading to the improved performance for both algorithms.

For some values of $p$, there is nearly a $10 : 1$ improvement factor for the BP and more than $100 : 1$ improvement for the OMP. Indeed, the OMP with the optimized projections lead to the better performance compared to the original BP, for low and mid-values of $p$.

The second experiment is similar to the first one, this time fixing $p = 25$ and varying $T$ in the range $[1, 7]$. The results are shown in Fig. 7. As expected, as $T$ grows, performance deteriorates. However, the optimized projections are consistent in their improved performance.

We should emphasize that the presented results do not include a thorough optimization of the parameters $\gamma$ and $t$ and the relation between $\mu_k$ and the CS performance remains still obscure at this stage. Also, our experiments concentrated on one specific choice of dictionary size that enables reasonable runtime simulation and this has an impact on the relatively weak performance CS shows. Other experiments we have done with much larger dictionaries show the same improvement as aforementioned, but require too long runtime for gathering fair statistics, and thus, are avoided. Still, the point this paper makes about the potential to get better projections and thereby improving CS performance, is clearly demonstrated.

The next experiment explores the effect of the dimensionality on the CS performance, before and after the optimization of the projection directions. We repeat the previous experiment but vary the dimension of the signals $n$ in the range $[40, 160]$, while updating $k$, $p$, and $T$ to be linearly proportional to it, using $k = 1.5 \cdot n$, $p = n/4$, and $T = n/20$. The idea behind such an experiment is that with the proportional growth of all these parameters together, we get a better indication to the asymptotic performance, as studied theoretically in [1]–[7].

The obtained results, in Fig. 8, show a consistent improvement in CS performance for the OMP, before and after the optimization of the projections. A similar but much slower improvement is also observed for the BP. Since in this experiment the ratio $p/T$ is fixed and equal to 5, these results suggest that this “oversampling” becomes more effective as $n \to \infty$. 

![Fig. 6. CS relative errors as a function of the number of measurements $p$, with random projections and optimized projections. Note: A vanishing graph implies a zero error rate.](image1)

![Fig. 7. CS relative errors as a function of the signals’ cardinality $T$, with random projections and optimized projections.](image2)
Fig. 8. CS relative errors as a function of the signals’ dimension $n$. This experiment assumes that $k$, $p$, and $T$ are growing linearly with $n$.

Fig. 9. CS relative errors as a function of the parameter $\ell\%$ in the optimization process of the projections.

We now proceed with the fourth experiment, considering different values of the parameter $\ell\%$ and see its influence of the CS performance. We used again a random dictionary of size $80 \times 120$, and assumed we have $p = 30$ measurements. We consider both a randomly initialized matrix $\mathbf{P}$ and its optimized version. In the optimization, we vary $\ell\%$ in the range 5%-95% and evaluate the CS performance with these projections for the OMP and the BP. Fig. 9 displays the obtained results.

As can be seen, choosing too small values (i.e., tending to address the regular mutual coherence) or too high values (i.e., considering the average of the absolute off-diagonal entries in $\mathbf{G}$) of $\ell\%$ both lead to inferior results. Best results in this case are obtained for $\ell\% = 10\%-20\%$, leading to a substantial improvement over wrong choices of this parameter. These results suggest that when studying the entries of the Gram matrix, the very small values (in magnitude) should be left untouched, as they serve well the CS performance.

The last experiment reported here is motivated by the question whether the obtained optimized projections exhibit an intuitive structure. For this experiment, we considered $\mathbf{D} = \mathbf{I}$, an identity matrix of size $80 \times 80$. At first, initializing the projections randomly, we found out that the optimized projections show no structure at all, appearing as pseudonoise. Thus, we initialized instead with a randomly chosen subset of 25 rows from the DCT unitary matrix, and use this as initialization to the optimization procedure. As can be seen in Fig. 10, the resulting projection shows no structure and, in fact, loses all relation to the original projection set. As a side note, we add that, in this experiment, the error rate for both the BP and the OMP improved by a factor of $\approx 10$ with the optimized projections.

V. CONCLUSION

CS is an emerging field of activity with beautiful theoretical results that state that signals can be compressed and sensed at the same time. This is based on the structural assumption such signals are satisfying—having a sparse and redundant representation over a specific dictionary. A crucial ingredient in the deployment of the CS idea is the use of linear projections that mix the signal. This paper aims to show that better choices of such mixtures are within reach. The projections can be designed such that the average mutual coherence of the effective dictionary becomes favorable. We have defined this property, shown how to design a projection operator based on it, and demonstrated how it indeed leads to a better CS performance.

The idea of optimizing the projections is appealing and should be further studied. In the following, there are several intriguing questions that future work could consider.

- How can the proposed optimization algorithm be performed or approximated for very high dimensions? This is important in cases where the CS is deployed on images or other signals of high dimensions.
- Optimizing the projections can be done alternatively using a direct method that considers CS performance, rather than...
addressing an indirect measure as done in this paper. Similarly, one might consider the optimization with respect to the UUP. Further work is required to explore these options and show how effective they are compared to the one discussed in this paper.

- We should develop a theoretical link between the average mutual coherence as presented here, to the CS performance, so as to give better justification for the proposed work. Perhaps there is yet another simple measure of the effective dictionary $\mathbf{PD}$ that could replace $\mu_{\text{UUP}}(\mathbf{PD})$ and lead to better results.

**APPENDIX**

**EVALUATING BP’S PERFORMANCE**

The problem we face is the following: We generate a sparse vector $\mathbf{a}_0$ and compute from it the measurement vector $\mathbf{y}_0 = \mathbf{PDa}_0$. In order to determine whether BP succeeds in the recovery of the signal $\mathbf{x} = \mathbf{D}a_0$, we should solve

$$\hat{a} = \arg\min_{a} |a|_1 \quad \text{s.t.} \quad \mathbf{y}_0 = \mathbf{PDa}$$

(A1)

and check whether $\hat{a} = a_0$. The problem in such a direct approach is the need to deploy a linear programming (LP) solver per each test, and as we are interested in many thousands of such tests, this approach becomes prohibitive.

Since we are dealing here with a synthetic test, where the desired solution is a priori known, we can replace the direct solution of (A1) with a much more moderate test of considering $a_0$ and checking whether it is indeed its minimum. In order to do so, we consider the necessary first-order Karush–Kuhn–Tucker (KKT) conditions, as emerging from the Lagrangian of (A1). The Lagrangian is given by

$$\mathcal{L}(\alpha, \lambda) = |a|_1 + \lambda^T(y_0 - \mathbf{PDa})$$

(A2)

with $\lambda$ serving as the Lagrange multipliers. Taking its derivative with respect to $\alpha$ and using the fact that the derivative of the absolute value at zero leads to the feasible interval $[-1, 1]$ (considering the subgradients), we obtain

$$\mathbf{D}^T\mathbf{P}^T\lambda = \begin{cases} +1, & \alpha_0(j) > 0 \\ -1, & \alpha_0(j) < 0 \\ 0, & \alpha_0(j) = 0 \end{cases}$$

(A3)

where one must require $\forall j: -1 \leq \alpha_0(j) < 1$.

Thus, if we find a feasible solution $\lambda$ to this system, we can guarantee that $a_0$ is the solution of (A1) and thus the BP is expected to succeed. If we cannot find a solution, we suspect that BP fails. Declaration of failure in such a case is definitely possible, but leads to an upper bound on the true number of errors, as our numerical scheme for solving (A3) may fail in spite of the BP success. Assuming that the expected number of such suspected failures is substantially smaller compared to $N$ (as is indeed the case in our simulations), we can directly try to solve (A1) for these few cases, and see whether failure takes place.

As for the solution of (A3), this can be achieved in various ways. We separate the equation set to two parts—the equality and the inequality constraints, denoted by $A_1\lambda = b$ and $-1 \leq A_2\lambda \leq 1$, respectively. We use the penalty method, minimizing the function

$$f(\lambda) = ||A_1\lambda - b||_2^2 + \beta||W\alpha_2\lambda||_2^2$$

(A4)

with respect to $\lambda$. The matrix $W$ is a diagonal weight matrix, initialized as $W = I$. Starting with a very small $\beta$, the first constraint is satisfied while the second might be violated. Iterating and increasing the value of $\beta$, the first term remains zero while the second one gets closer to the satisfaction of $-1 \leq A_2\lambda \leq 1$. A more delicate update step can be proposed, where the extreme entries in the vector $|A_2\lambda|$ that are above 1 are treated by increasing their weight in $W$. A finite number of such an iterative algorithm (50 iterations) was used and shown to be 1 and 2 orders of magnitude faster than the full LP solver.

**ACKNOWLEDGMENT**

The author would like to thank Dr. M. Zibulevsky for helpful discussions and his fruitful ideas on how to speed up the tests carried out in this work. He would also like to thank the two referees who studied carefully this work and helped in improving the quality of this paper.

**REFERENCES**


Michael Elad received the B.Sc., M.Sc., and D.Sc. degrees from the Department of Electrical Engineering, The Technion—Israel Institute of Technology, Haifa, Israel, in 1986, 1988, and 1997, respectively.

From 1988 to 1993, he served in the Israeli Air Force. From 1997 to 2000, he was with Hewlett-Packard Laboratories as an R&D Engineer. From 2000 to 2001, he headed the Research Division at Jigami Corporation, Israel. From 2001 to 2003, he was a Research Associate with the Computer Science Department, Stanford University, Stanford, CA (SCCM program). Since September 2003, he has been with the Department of Computer Science, The Technion, as an Assistant Professor. He works in the field of signal and image processing, specializing particularly in inverse problems, sparse representations, and overcomplete transforms.

Dr. Elad received The Technion’s Best Lecturer Award four times (1999, 2000, 2004, and 2005). He is also the recipient of the Guttwirth and the Wolf fellowships.