Incoherent Unit-norm Frame Design via an Alternating Minimization Penalty Method

Mostafa Sadeghi, *Student Member, IEEE* and Massoud Babaie-Zadeh, *Senior Member, IEEE*

***Abstract*—This letter is concerned with designing unit-norm incoherent frames, i.e., a set of vectors in a finite dimensional Hilbert space with unit norms and very low absolute pair-wise correlations. Due to their widespread use in a variety of applications, including compressed sensing and coding theory, incoherent frame design has received considerable attention, and many algorithms have been proposed to this aim. In this letter, a new algorithm is presented which constructs incoherent frames by minimizing the maximum absolute pair-wise correlations (mutual coherence) of the frame vectors. Our strategy is based on an alternating minimization penalty method which admits efficient solvers using proximal algorithms. Experimental results on designing incoherent frames of various dimensions show that our algorithm outperforms some recent methods in the literature.**

***Index Terms*—Equiangular tight-frames, mutual coherence, incoherent frame design, proximal algorithms**

I. Introduction

FRAME theory [1], [2] arises in many signal processing problems, including sparse signal representation [3], [4], compressed sensing [5], source coding, robust transmission, and code division multiple access (CDMA) systems [2]. Frames are a generalization of basis vectors of an inner-product space in the sense that they allow a redundancy in the set of vectors. In fact, a frame may contain much more vectors than needed for describing and representing other vectors of the associated space. Therefore, compared to bases, frames provide a more stable and efficient way of representing signals [1].

Mutual coherence [6] is a simple yet fundamental metric for evaluating the goodness of a frame, which is defined as the maximum absolute pair-wise inner-products between the frame vectors. Performance guarantees for sparse signal recovery algorithms indicate that incoherent frames, i.e., those having mutual coherences as low as possible, are highly desirable [3], [7]–[9] (for a detailed discussion, refer to [4]). In addition to sparse signal processing, incoherent frames are also of great importance in communications and coding theory for designing robust and decodable codes [2], [10]. The need for incoherent frames and their attractive properties have led to development of practical algorithms for constructing them [11]–[21].

The authors are with the Electrical Engineering Department, Sharif University of Technology, Tehran 11155-8639, Iran (e-mail: m.saadeghii@gmail.com; mbzadeh@yahoo.com).

In this letter, a new algorithm is introduced for designing incoherent frames. This is achieved by solving a mutual coherence minimization problem through an alternating minimization penalty method. A number of numerical experiments were performed to evaluate the performance of our algorithm in designing incoherent frames of various dimensions. The results demonstrate that compared with some recent algorithms in the literature, our proposed one has a better overall performance.

The rest of the letter is structured as follows. Section II reviews some terminologies related to frames along with a summary of the previous works on designing incoherent frames. Our proposed algorithm is introduced in Section III. Finally, Section IV is devoted to numerical results.

II. Background

A. Frame theory

We briefly review the frame theory from [22]. A collection of m vectors $\{\mathbf{f}_i\}_{i=1}^m$ constitutes a frame for $\mathbb{R}^n$ if $\forall \mathbf{v} \in \mathbb{R}^n$

$$\alpha\|\mathbf{v}\|_2^2 \leq \sum_{i=1}^m |\mathbf{f}_i^T\mathbf{v}|^2 \leq \beta\|\mathbf{v}\|_2^2, \tag{1}$$

where α and β, with $0 < \alpha \leq \beta < \infty$, are called the lower and the upper frame bounds, respectively. The frame synthesis operator, $\mathbf{F}$, is defined as the matrix that has the frame vectors as its columns, i.e., $\mathbf{F} = [\mathbf{f}_1, \cdots \mathbf{f}_m]$. When referring to a frame, one usually means its synthesis operator, $\mathbf{F}$. A unit-norm frame $\mathbf{F}$ is a frame for which $\forall i: \ \|\mathbf{f}_i\|_2 = 1$. Moreover, when $\alpha = \beta$, the frame is called an α-tight frame. For a unit-norm frame $\mathbf{F}$, its mutual coherence is defined as

$$\mu(\mathbf{F}) = \max_{i,j:\ i\neq j} \ |\mathbf{f}_i^T\mathbf{f}_j|, \tag{2}$$

or equivalently

$$\mu(\mathbf{F}) = \|\mathbf{F}^T\mathbf{F} - \mathbf{I}\|_\infty, \tag{3}$$

in which, $\|\mathbf{X}\|_\infty \triangleq \max_{i,j} \ |x_{ij}|$ and $\mathbf{I}$ denotes the identity matrix. For a given frame $\mathbf{F}$, its Gram matrix is defined as $\mathbf{G} = \mathbf{F}^T\mathbf{F}$, which is a symmetric and positive semidefinite matrix of rank n. Note that the diagonal entries of $\mathbf{G}$ correspond to the squared ℓ_2 norms of the frame vectors, which are all equal to 1 for unit-norm frames, while its off-diagonal entries are inner-products between distinct frame vectors. Frames with very low mutual coherences are referred to as *incoherent frames*. While complete frames ($n = m$) can be arbitrarily incoherent ($\mu = 0$ for an orthonormal basis), the mutual coherence for a general $n \times m$ frame is lower bounded by

$$\mu(\mathbf{F}) \geq \sqrt{\frac{m-n}{n(m-1)}}, \tag{4}$$

which is known as the Welch bound (WB) [22]. This bound is achievable for a frame $\mathbf{F} \in \mathbb{R}^{n\times m}$ only when $m \leq n(n+1)/2$. Among all unit-norm frames of the same dimension, those with the minimum mutual coherence are called *Grassmannian frames* [22]. Furthermore, when this minimum value coincides with WB, they are called *optimal Grassmannian frames* or *equiangular tight frames* (ETFs) [22].

B. Incoherent frame design

From our discussion in Section I it follows that ETFs are ideal choices in applications where incoherent frames are desirable. However, their construction is difficult in practice, especially for large n and m [16], [22]. In addition, they do not exist for arbitrary dimensions; see the existence table of ETFs in [23]. Some approaches, including [11], [17]–[21], aim at constructing ETFs, using, e.g., algebraic and geometric schemes. In spite of having significant performance in some applications, these approaches usually impose certain restrictions on frame characteristics. Another family of algorithms, including [12]–[16], [24]–[26], solve convex/non-convex optimization problems to design incoherent frames. In contrary to the previously mentioned ETF design schemes, optimization-based approaches try to construct frames as close as possible to ETFs. So, they do not impose any restriction on frames and are able to construct incoherent frames of arbitrary dimensions. In this letter, we focus on optimization-based techniques.

Earlier incoherent frame design methods are based on an alternating projection approach [14], [15], in which, an initial frame $\mathbf{F}$ is refined by iteratively shrinking the large off-diagonal entries of its Gram matrix $\mathbf{G}$, reducing the rank of $\mathbf{G}$ back to n, and then factorizing $\mathbf{G}$ to get the new estimate of $\mathbf{F}$. This procedure is repeated until convergence. Subsequent works include [12], [13], [16], [24]–[26]. Specifically, a novel approach has been proposed in [12] that constructs incoherent frames by minimizing the mutual coherence as follows

$$\min_{\mathbf{F}\in\mathcal{F}} \max_{i,j:\ i\neq j} |\mathbf{f}_i^T\mathbf{f}_j|, \tag{5}$$

where $\mathcal{F} \triangleq \{\mathbf{F} \in \mathbb{R}^{n\times m} \mid \forall i:\ \|\mathbf{f}_i\|_2 = 1\}$. The strategy used in [12] to solve this non-convex problem consists of a sequential convex programming framework that approximates the solutions to (5) by solving a series of locally convex optimization problems inside a defined trust region [12]. More precisely, the i-th frame vector $\mathbf{f}_i$ is updated through

$$\min_{\mathbf{f}_i} \max_{j:\ j\neq i} |\mathbf{h}_j^T\mathbf{f}_i| \text{ s.t. } \forall i: \|\mathbf{f}_i - \mathbf{h}_i\|_2 \leq T_i, \tag{6}$$

in which, $\mathbf{h}_i$'s are the previous estimates of the frame vectors, and T_i's are positive constants characterizing the search regions. The above convex problem is then solved for each frame vector using the generic convex optimization solver CVX [27]. This approach has been further improved in [13]. The new algorithm, which is called sequential iterative decorrelation by convex optimization (SIDCO), has a better performance than the previous optimization-based algorithms [13].

III. Our work

A. Proposed algorithm

We target the same cost function (5) used by SIDCO to construct incoherent frames, but we make use of proximal algorithms [28] to solve it. To this end, let us rewrite (5) into the following equivalent problem

$$\min_{\mathbf{F}\in\mathcal{F}} \|\mathbf{F}^T\mathbf{F} - \mathbf{I}\|_\infty. \tag{7}$$

In order to solve the above problem, we define a new variable $\mathbf{Q} \triangleq \mathbf{F}^T\mathbf{F}-\mathbf{I}$, which converts (7) into the following alternative form

$$\min_{\mathbf{F}\in\mathcal{F},\mathbf{Q}} \|\mathbf{Q}\|_\infty \quad \text{s.t.} \quad \mathbf{Q} = \mathbf{F}^T\mathbf{F} - \mathbf{I}. \tag{8}$$

This new problem is then solved via penalty methods [29], leading to

$$\{\mathbf{Q}^\alpha, \mathbf{F}^\alpha\} = \operatorname*{argmin}_{\mathbf{F}\in\mathcal{F},\mathbf{Q}} \left\{\|\mathbf{Q}\|_\infty + \frac{1}{2\alpha}\|\mathbf{Q} - \mathbf{F}^T\mathbf{F} + \mathbf{I}\|_F^2\right\}, \tag{9}$$

where, $\alpha > 0$ is a penalty parameter. Note that when $\alpha \to 0$, the solutions of (9) coincide with those of (8). Let $\mathbf{F}^*$ be an optimal solution[1] of (7). Then, considering the equivalence of (8) and (7), it follows that $\mathbf{F}^\alpha \to \mathbf{F}^*$ (up to a rotation) when $\alpha \to 0$.

We solve (9) using alternating minimization, in which, the cost function is iteratively minimized over one variable while the other one is fixed. Starting with an initial $\mathbf{F}_0^\alpha$, the whole process consists of iteratively solving ($k \geq 0$)

$$\mathbf{Q}_{k+1}^\alpha = \operatorname*{argmin}_{\mathbf{Q}} \left\{\|\mathbf{Q}\|_\infty + \frac{1}{2\alpha}\|\mathbf{Q} - \mathbf{F}^{\alpha T}_k\mathbf{F}_k^\alpha + \mathbf{I}\|_F^2\right\}, \tag{10}$$

followed by

$$\mathbf{F}_{k+1}^\alpha = \operatorname*{argmin}_{\mathbf{F}\in\mathcal{F}} \frac{1}{2}\|\mathbf{Q}_{k+1}^\alpha - \mathbf{F}^T\mathbf{F} + \mathbf{I}\|_F^2. \tag{11}$$

Before proceeding further, let us introduce the notion of *proximal mapping* [28], which will be used for solving the above subproblems.

Definition 1 (proximal mapping [28])**.** *The proximal mapping of a proper and lower semicontinuous function $g:\ \mathbb{R}^m \longrightarrow (-\infty, +\infty]$ at $\mathbf{x} \in \mathbb{R}^m$ is defined as*

$$prox_g(\mathbf{x}) \triangleq \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^m} \left\{g(\mathbf{u}) + \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|_2^2\right\}.$$

Now, let us focus on the $\mathbf{Q}$-update problem given in (10). Define the vectorization operator $\text{vec}(\cdot):\ \mathbb{R}^{m\times m} \to \mathbb{R}^{m^2}$ which converts its matrix argument to a vector by stacking the columns of the matrix on top of one another, and let $\text{vec}^{-1}(\cdot)$ denote the inverse operator. It is then straightforward to show that (10) can be rewritten as

$$\mathbf{q}_{k+1}^\alpha = \operatorname*{argmin}_{\mathbf{q}} \left\{\|\mathbf{q}\|_\infty + \frac{1}{2\alpha}\|\mathbf{q} - \mathbf{p}_k^\alpha\|_2^2\right\}, \tag{12}$$

[1] Note that problem (7) has infinitely many solutions with the same cost value. In other words, if $\mathbf{F}^*$ is an optimal solution, the same is $\mathbf{UF}^*$, for any unitary matrix $\mathbf{U}$.

in which, $\mathbf{q}_{k+1}^{\alpha} \triangleq \text{vec}(\mathbf{Q}_{k+1}^{\alpha})$ and $\mathbf{p}_{k}^{\alpha} \triangleq \text{vec}(\mathbf{P}_{k}^{\alpha})$, with $\mathbf{P}_{k}^{\alpha} \triangleq \mathbf{F}^{\alpha}{}_{k}^{T}\mathbf{F}_{k}^{\alpha} - \mathbf{I}$. Note that (12) is, by definition, the proximal mapping of $\alpha\|.\|_{\infty}$ at $\mathbf{p}_{k}^{\alpha}$. Direct computation of this proximal mapping would be tricky. Alternatively, we use the following lemma to solve (12).

Lemma 1 (Moreau decomposition [28])**.** *Let $f : \mathbb{R}^m \longrightarrow (-\infty, +\infty]$ be a proper and lower semicontinuous convex function whose conjugate is denoted by f^*. Then, the following relation holds for any $\mathbf{x} \in \mathbb{R}^m$ and $\beta > 0$:*

$$\mathbf{x} = \textit{prox}_{\beta f}(\mathbf{x}) + \beta \cdot \textit{prox}_{f^*/\beta}(\mathbf{x}/\beta). \tag{13}$$

To utilize the above lemma, define $f(\mathbf{x}) = \|\mathbf{x}\|_{\infty}$. The convex conjugate of f is the indicator function of the ℓ_1 unit-norm ball [30]. So, the solution of (12) is obtained via

$$\mathbf{q}_{k+1}^{\alpha} = \text{prox}_{\alpha f}(\mathbf{p}_{k}^{\alpha}) = \mathbf{p}_{k}^{\alpha} - \alpha \cdot \text{prox}_{f^*/\alpha}(\mathbf{x}/\alpha). \tag{14}$$

Furthermore, it is straightforward to show that

$$\forall \mathbf{x} : \ \text{prox}_{f^*/\alpha}(\mathbf{x}/\alpha) = \frac{1}{\alpha}\mathcal{P}_{\mathcal{B}_1^{\alpha}}(\mathbf{x}), \tag{15}$$

where, $\mathcal{B}_1^{\alpha}$ is the ℓ_1 norm ball of radius α, and $\mathcal{P}_{\mathcal{B}_1^{\alpha}}$ denotes the projection onto $\mathcal{B}_1^{\alpha}$. Finally, $\mathbf{Q}_{k+1}^{\alpha}$ in (10) is computed as

$$\mathbf{Q}_{k+1}^{\alpha} = \mathbf{P}_{k}^{\alpha} - \text{vec}^{-1}\big(\mathcal{P}_{\mathcal{B}_1^{\alpha}}\big(\text{vec}(\mathbf{P}_{k}^{\alpha})\big)\big)\cdot \tag{16}$$

There exist a number of algorithms for computing $\mathcal{P}_{\mathcal{B}_1^{\alpha}}(\mathbf{p}_k)$; see, e.g., [31]. To accelerate the algorithm, we use an extrapolation scheme as follows

$$\begin{cases} \mathbf{P}_{k}^{\alpha} = \mathbf{F}^{\alpha}{}_{k}^{T}\mathbf{F}_{k}^{\alpha} - \mathbf{I} \\ \mathbf{R}_{k}^{\alpha} = \mathbf{P}_{k}^{\alpha} + w_1(\mathbf{P}_{k}^{\alpha} - \mathbf{P}_{k-1}^{\alpha}) \\ \mathbf{Q}_{k+1}^{\alpha} = \mathbf{R}_{k}^{\alpha} - \text{vec}^{-1}\big(\mathcal{P}_{\mathcal{B}_1^{\alpha}}\big(\text{vec}(\mathbf{R}_{k}^{\alpha})\big)\big) \end{cases}, \tag{17}$$

where, $w_1 \geq 0$ is a weighting constant.

Now, consider the $\mathbf{F}$-update in (11). We solve this problem using an accelerated proximal algorithm proposed in [32]. This amounts to

$$\begin{cases} \mathbf{S}_{(t)} = \mathbf{F}_{(t)} - 2\eta\mathbf{F}_{(t)}(\mathbf{F}_{(t)}^{T}\mathbf{F}_{(t)} - \mathbf{Q}_{k+1}^{\alpha} - \mathbf{I}) \\ \mathbf{F}_{(t+1)} = \mathcal{P}_{\mathcal{F}}\left\{\mathbf{S}_{(t)}\right\} + w_2(\mathbf{F}_{(t)} - \mathbf{F}_{(t-1)}) \end{cases}, \tag{18}$$

where, t denotes the iteration index, $\eta > 0$ is a step-size, $w_2 \geq 0$ is a weighting constant, and $\mathcal{P}_{\mathcal{F}}\{.\}$ performs the projection onto $\mathcal{F}$, which is computed by simply normalizing the columns of its argument. This iterative process is repeated for a few iterations. Under appropriate selection of η, it is guaranteed [32] that the iterative algorithm in (18) converges to a critical point of the cost function in (11). The final estimate obtained in (18) provides an approximation to $\mathbf{F}_{k+1}^{\alpha}$. The overall solver of (9) iterates between the update of $\mathbf{Q}_{k+1}^{\alpha}$ through (17), and performing a few iterations of (18) to update $\mathbf{F}_{k+1}^{\alpha}$.

Following the same approach used in a general penalty method [29], we consider a decreasing sequence of $\{\alpha_i\}$, in which, $\alpha_i \to 0$, and find approximate minimizers of (9) for each i. Moreover, the final solutions corresponding to α_i are used as the initialization points to start the minimization process corresponding to α_{i+1}. This is called *warm-starting*.

The overall procedure for solving (9) has been summarized in Algorithm 1. We call this algorithm IFD-AMPM, for incoherent frame design via alternating minimization penalty method. The parameters involved in the algorithm are discussed in the following subsection.

Algorithm 1 IFD-AMPM

```
1:  Objective: Constructing an incoherent frame F.
2:  Inputs: initial frame F_0, step-size η > 0, weighting
    constants w_1 ≥ 0 and w_2 ≥ 0, initial value for the penalty
    parameter α_0, decreasing factor of the penalty parameter
    c ∈ [0.5, 1), maximum number of outer- and inner-loop
    iterations, N_o and N_i, maximum number of F-update
    iterations T.
3:  Initialization: i = 0, α = α_0, F_0^{α_0} = F_0
4:  while i < N_o do
5:      for k = 0, ··· , N_i − 1 do
6:          Update Q_{k+1}^{α_i} through (17)
7:          for t = 0, ··· , T − 1 do
8:              Compute F_(t+1) through (18)
9:          end for
10:         F_{k+1}^{α_i} = F_(T)
11:     end for
12:     α_{i+1} = c · α_i
13:     i = i + 1
14: end while
15: Output: F
```

B. Parameter settings

We initialize our algorithm with a unit-norm tight frame (UNTF) generated using the approach proposed in [33], [34]. The step-size η should be small enough to avoid divergence. For our experiments, we found that $\eta = 0.05$ works well. Moreover, $w_1 = w_2 = 0.85$ results in a good performance. The initial value for the penalty parameter, α_0, should be sufficiently large. We set it as $\alpha_0 = 500 \times \|\mathbf{P}_0^{\alpha_0}\|_{\infty}$. The decreasing factor, c, of α should be in $[0.5, 1)$. We found that $c = 0.9$ is an appropriate choice. The parameters N_o and N_i, denoting the number of outer- and inner-loop iterations, are set according to the desired precision.

For the $\mathbf{F}$-update problem in lines 7–9 of Algorithm 1, a few iterations, say $T = 3$, suffices to reach a good solution. In fact, one advantage of our proposed algorithm is that, it benefits from warm-starting. That is, the iterations corresponding to each particular value of α (lines 5–11) and also, the iterations along the sequence of α's (lines 4–14) help reduce the works needed for updating $\mathbf{Q}$ and $\mathbf{F}$.

IV. Numerical results

This section compares the performance of our proposed algorithm with those of SIDCO [13] and another optimization-based approach, dubbed LZYCB [26], which designs incoherent UNTFs using an alternating minimization approach.

For SIDCO and LZYCB, we used the MATLAB implementations provided by their authors. To perform projection onto the ℓ_1 norm ball in IFD-AMPM, we used the algorithm[2] proposed in [31]. The total number of iterations for all the algorithms was set to 200. Moreover, for IFD-AMPM we set $N_i = 15$. Its remaining parameters were set as suggested in Subsection III-B. As a rough measure of computational complexity, runtimes of the algorithms are reported. Our

[2]MATLAB code: http://stanford.edu/~jduchi/projects/DuchiShSiCh08.html

TABLE I: Mutual coherence values reached by SIDCO [13], LZYCB [26], and IFD-AMPM for frames $\mathbf{F} \in \mathbb{R}^{n\times 120}$ with different values of n.

m	SIDCO	LZYCB	IFD-AMPM	WB
15	0.3502	0.4146	0.3225	0.2425
20	0.2775	0.3409	0.2605	0.2050
25	0.2303	0.2766	0.2183	0.1787
30	0.1977	0.2358	0.1879	0.1588
35	0.1727	0.2037	0.1649	0.1429
40	0.1529	0.1815	0.1468	0.1296
45	0.1371	0.1610	0.1319	0.1183
50	0.1236	0.1467	0.1193	0.1085
55	0.1120	0.1337	0.1086	0.0997
60	0.1019	0.1209	0.0991	0.0917
100	0.1421	0.0666	0.0449	0.0410
110	0.0840	0.0986	0.0315	0.0276
115	0.0452	0.0464	0.0219	0.0191
119	0.0084	0.0230	0.0084	0.0084

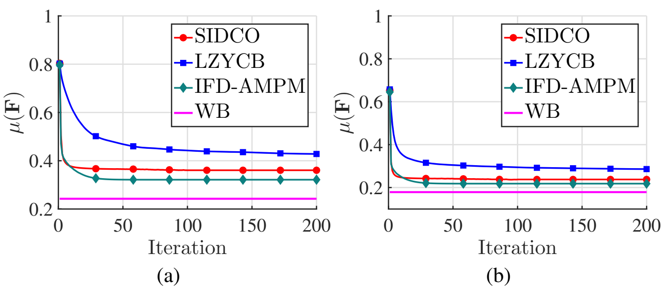

Fig. 1: Mutual coherence values along iterations of SIDCO [13], LZYCB [26], and IFD-AMPM for designing incoherent frames of dimensions: (a) 15×120, and (b) 25×120.

simulations were carried out on a 64 bit Windows 7 operating system with 8 GB RAM and an Intel core i7 CPU.

First, the algorithms are compared in designing incoherent frames of dimensions $n \times 120$ for different values of n. The minimum mutual coherences reached by the algorithms over 100 realizations are shown in Table I. As can be seen, the final mutual coherences reached by IFD-AMPM are lower than the ones corresponding to LZYCB and SIDCO. In addition, LZYCB has inferior performance to the other algorithms.

Figure 1 depicts the progress of $\mu(\mathbf{F})$ over the iterations of the algorithms (averaged over 30 different realizations) for frames of dimensions 15×120 and 25×120. It is observed that SIDCO and IFD-AMPM converge faster than LZYCB. Actually, in IFD-AMPM and SIDCO the mutual coherences decrease significantly after about 10 iterations. Furthermore, IFD-AMPM ends up with frames of lower mutual coherences than the other two algorithms. From the aspect of runtime, while each iteration of SIDCO takes approximately 18 seconds in average, for LZYCB and IFD-AMPM the runtimes per iteration are 0.023 and 0.041 seconds, respectively, which are significantly lower than that of SIDCO.

The next experiment aims to see how the performance of the algorithms compares with the numerically optimal packings found by [35]. To this end, we run the algorithms to construct frames $\mathbf{F} \in \mathbb{R}^{3\times m}$ with increasing values of m. The results are reported in Table II. As demonstrated by this table, IFD-AMPM yields frames whose mutual coherences are almost the same as those of the numerically optimal packings. Additionally, while SIDCO follows the numerically optimal results closely, it is not as good as IFD-AMPM for some values of m. LZYCB, on the other hand, fails in designing incoherent frames for $m \geq 14$.

TABLE II: Mutual coherence values reached by SIDCO [13], LZYCB [26], and IFD-AMPM together with those of the numerically optimal packings reported in [35] for frames $\mathbf{F} \in \mathbb{R}^{3\times m}$ with increasing values of m.

m	SIDCO	LZYCB	IFD-AMPM	Num. Opt.	WB
3	0	0	0	0	0
4	0.3333	0.3333	0.3333	0.3333	0.3333
5	0.4472	0.5393	0.4472	0.4472	0.4082
6	0.4472	0.4472	0.4472	0.4472	0.4472
7	0.5893	0.5777	0.5774	0.5774	0.4714
8	0.6480	0.7466	0.6478	0.6476	0.4880
9	0.6694	0.7471	0.6694	0.6694	0.5000
10	0.6870	0.6979	0.6861	0.6861	0.5092
11	0.7149	0.7565	0.7144	0.7144	0.5164
12	0.7559	0.9568	0.7445	0.7445	0.5222
13	0.7721	0.9631	0.7681	0.7681	0.5270
14	0.7828	1	0.7806	0.7806	0.5311
15	0.7888	1	0.7867	0.7866	0.5345
16	0.7969	1	0.7947	0.7947	0.5375
17	0.8204	1	0.8168	0.8168	0.5401
18	0.8283	1	0.8250	0.8250	0.5423
19	0.8388	1	0.8367	0.8367	0.5443
20	0.8447	1	0.8414	0.8414	0.5461
21	0.8488	1	0.8471	0.8460	0.5477
22	0.8609	1	0.8491	0.8490	0.5492
23	0.8632	1	0.8616	0.8616	0.5505
24	0.8724	1	0.8646	0.8646	0.5517
25	0.8779	1	0.8728	0.8725	0.5528
26	0.8818	1	0.8784	0.8770	0.5538
27	0.8851	1	0.8810	0.8809	0.5547
28	0.8869	1	0.8845	0.8842	0.5556
29	0.8944	1	0.8871	0.8868	0.5563
30	0.8956	1	0.8912	0.8910	0.5571

V. Conclusion

We addressed designing incoherent frames comprising a set of vectors with very low mutual coherences. Many algorithms have already been proposed to this aim. One family of approaches offer schemes for constructing optimal incoherent frames, known as equiangular tight-frames (ETFs), which have been successfully applied in many applications [11], [17]–[21]. However, ETFs do not exist for arbitrary dimensions, and the available algorithms for designing them usually impose certain restrictions on frame dimensions.

In this letter, we focused on another family of algorithms, including [12]–[16], [24]–[26] which do not target at designing ETFs. Instead, they try to design frames as close as possible to ETFs. In particular, we introduced a new algorithm, called IFD-AMPM, which designs incoherent frames by minimizing the mutual coherence among all unit-norm frames through an alternating minimization penalty method. Our numerical experiments on designing incoherent frames of various dimensions confirmed that IFD-AMPM outperforms some recent optimization-based approaches.

REFERENCES

[1] J. Kovacevic and A. Chebira, "Life beyond bases: The advent of frames (Part I)," *IEEE Signal Proc. Magazine*, vol. 24, no. 4, pp. 86–104, 2007.

[2] J. Kovacevic and A. Chebira, "Life beyond bases: The advent of frames (Part II)," *IEEE Signal Proc. Magazine*, vol. 24, no. 5, pp. 115–125, 2007.

[3] M. Elad, *Sparse and Redundant Representations*, Springer, 2010.

[4] W. U. Bajwa and A. Pezeshki, "Finite frames for sparse signal processing," in *Finite Frames*, P. Casazza and G. Kutyniok, Eds., chapter 10, pp. 303–335. Birkhäuser Boston, Cambridge, MA, 2012.

[5] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[6] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[7] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[8] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuit in quasi-incoherent dictionaries," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 255–261, 2006.

[9] J. A. Tropp, "On the conditioning of random subdictionaries," *Appl. Computat. Harmon. Anal.*, vol. 25, no. 1, pp. 1–24, 2008.

[10] V. K. Goyal, J. Kovacevic, and J. A. Kelner, "Quantized frame expansions with erasures," *Appl. Comput. Harmon. Anal.*, vol. 10, no. 3, pp. 203–233, 2001.

[11] W. U. Bajwa, R. Calderbank, and D. G. Mixon, "Two are better than one: Fundamental parameters of frame coherence," *Appl. Comput. Harmon. Anal.*, vol. 33, no. 1, pp. 58–78, 2012.

[12] C. Rusu, "Design of incoherent frames via convex optimization," *IEEE Signal Proc. Letters*, vol. 20, no. 7, pp. 673–676, 2013.

[13] C. Rusu and N. González-Prelcic, "Designing incoherent frames through convex techniques for optimized compressed sensing," *IEEE Trans. on Signal Proc.*, vol. 64, no. 9, pp. 2334–2344, 2016.

[14] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. on S*, vol. 55, no. 12, pp. 5695–5702, 2007.

[15] J. A. Tropp, I. S. Dhillon, R. W. Heath, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Trans. Info. Theory*, vol. 51, no. 1, pp. 188–209, 2005.

[16] E. V. Tsiligianni, L. P. Kondi, and A. K. Katsaggelos, "Construction of incoherent unit norm tight frames with application to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2319–2330, 2014.

[17] P. Xia, S. Zhou, and G. B. Giannakis, "Achieving the welch bound with difference sets," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1900–1907, 2005.

[18] S. Waldron, "On the construction of equiangular frames from graphs," *Linear Algebra Appl.*, vol. 431, no. 11, pp. 2228–2242, 2009.

[19] M. Fickus, D. G. Mixon, and J. C. Tremain, "Steiner equiangular tight frames," *Linear Algebra Appl.*, vol. 436, no. 5, pp. 1014–1027, 2012.

[20] J. Jasper, D. G. Mixon, and M. Fickus, "Kirkman equiangular tight frames and codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, 2014.

[21] M. Fickus, D. G. Mixon, and J. Jasper, "Equiangular tight frames from hyperovals," 2016, http://arxiv.org/abs/1602.05557.

[22] T. Strohmer and Heath R. W., "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[23] M. Fickus and D. G. Mixon, "Tables of the existence of equiangular tight frames," 2016, https://arxiv.org/abs/1504.00253.

[24] V. Abolghasemi, S. Ferdowsi, and S. Sanei, "A gradient-based alternating minimization approach for optimization of the measurement matrix in compressive sensing," *Signal Processing*, vol. 92, pp. 999–1009, 2012.

[25] J. Xu, Y. Pi, and Z. Cao, "Optimized projection matrix for compressive sensing," *EURASIP Journal on Advances in Signal Processing*, 2010.

[26] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai, "On projection matrix optimization for compressive sensing systems," *IEEE Trans. on Signal Proc.*, vol. 61, no. 11, pp. 2887–2898, 2013.

[27] M. Grant and S. Boyd, "Matlab software for disciplined convex programming," [Online], 2011, Available: http://cvxr.com/.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

[29] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[31] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *International Conference on Machine Learning (ICML)*, 2008.

[32] P. Ochs, Y. Chen, T. Brox, and T. Pock, "iPiano: Inertial proximal algorithm for nonconvex optimization," *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 388–1419, 2014.

[33] J. J. Benedetto and M. Fickus, "Finite normalized tight frames," *Adv. Comput. Math.*, vol. 18, pp. 357–385, 2003.

[34] P. G. Casazza, M. Fickus, and D. G. Mixon, "Auto-tuning unit norm frames," *Appl. Comput. Harmon. Anal.*, vol. 32, no. 1, pp. 1–15, 2012.

[35] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: Packings in grassmannian spaces," *Exper. Math.*, vol. 5, no. 2, pp. 139–159, 1996.