Fast Correlation Method for Partial Fourier and Hadamard Sensing Matrices in Matching Pursuit Algorithms

Kee-Hoon KIM, Hosung PARK, Seokbeom HONG, Nonmembers, and Jong-Seon NO, Member

SUMMARY There have been many matching pursuit algorithms (MPAs) which handle the sparse signal recovery problem, called compressed sensing (CS). In the MPAs, the correlation step makes a dominant computational complexity. In this paper, we propose a new fast correlation method for the MPA when we use partial Fourier sensing matrices and partial Hadamard sensing matrices which are widely used as the sensing matrix in CS. The proposed correlation method can be applied to almost all MPAs without causing any degradation of their recovery performance. Also, the proposed correlation method can reduce the computational complexity of the MPAs well even though there are restrictions depending on a used MPA and parameters.

key words: compressed sensing, Fourier, Hadamard, low-complexity, matching pursuit, sensing matrix

1. Introduction

Compressed sensing (CS) is a novel sampling technique, where one can recover sparse signals from the undersampled measurements [1]. In a typical CS problem, the goal is to exactly reconstruct the \( N \times 1 \) \( K \)-sparse signal vector \( x \) based on the \( M \times 1 \) measurement vector \( y \). By \( K \)-sparse, we mean that there are at most \( K \) nonzero elements in \( x \). The complex vectors \( x \) and \( y \) are linearly related to each other as

\[
y = \Phi x
\]

where \( \Phi \) is the \( M \times N \) sensing matrix. Here, the relation of \( K \), \( M \), and \( N \) is generally given as \( K < M \ll N \).

For the sensing matrix \( \Phi \), partial Fourier matrices and partial Hadamard matrices are widely used, where we mean that a partial matrix is constructed by selecting \( M \) rows of \( N \times N \) original matrix \( A \). In other words, \( \Phi = S_\Omega A \), where \( S_\Omega \) is the \( M \times N \) selection matrix consisting of \( M \) rows (selected row indices from an index set \( \Omega \)) of \( N \times N \) identity matrix \( I \).

Firstly, the partial Fourier sensing matrix is frequently used in CS because of its good recovery performance, fast implementation using the fast Fourier transform (FFT), and applicability to the signals in the various practical systems [2],[3]. The examples include channel estimation in communication systems [4] and magnetic resonance imaging [5]. For the partial Fourier sensing matrix, the index set \( \Omega \) can be constructed randomly or based on the cyclic difference set [6].

Secondly, some recent researches showed that well-designed deterministic sensing matrices based on linear block codes have better performance and less complexity for signal recovery compared to random sensing matrices [7],[8]. It is well known that a sensing matrix whose columns are bipolar-presented codewords of a binary linear block code can be viewed as a partial Hadamard matrix. Also, we can utilize the computational efficiency of the fast Hadamard transform (FHT).

To recover the sparse signal \( x \) in (1), some matching pursuit algorithms (MPAs) are proposed, and they received significant attention recently because of their low computational complexity and simple geometric interpretation. The MPAs find a sparse estimation of the signal \( x \) from \( y \) in a greedy fashion. The MPAs work iteratively by computing the nonzero components of \( x \), where the corresponding columns in \( \Phi \) have the high correlation with the current residual. Examples include the orthogonal matching pursuit (OMP) [9] and its modified versions such as the compressive sampling matching pursuit (CoSaMP) [10], the regularized OMP (ROMP) [11], the subspace pursuit (SP) [12], and the backtracking-based adaptive OMP (BAOMP) [13]. In Algorithm 1, we summarize the OMP which is the most basic algorithm among the MPAs. The steps marked by \( \diamond \) are the common steps to the MPAs.

Algorithm 1; Conventional OMP recovery algorithm

1. Initialize : \( r_0 = y, \Lambda_0 = \emptyset, t = 1, \diamond \)
2. Correlation : \( h_{t-1} = \Phi^H r_{t-1}, \diamond \)
3. Identification : \( \Lambda_t = \text{arg max}_{j=1, \ldots, N} | h_{t-1}(j) |. \)
4. Augment the index set : \( \Lambda_t = \Lambda_{t-1} \cup \{ \Lambda_t \} \).
5. Construct \( \Phi_{t} : \Phi_{t} = \Phi S_{\Lambda_t}^T \), where \( S_{\Lambda_t} \) is the selection matrix consisting of rows (indices from \( \Lambda_t \)) of \( I \).
6. Least squares : \( x_t = (\Phi_{t}^H \Phi_{t})^{-1} \Phi_{t}^H y. \)
7. Update current residual : \( a_t = \Phi_{t} x_t, r_t = y - a_t, \diamond \)
8. \( t = t + 1 \), return to 2) if the halting criterion is not triggered. \( \diamond \)

In Algorithm 1, performing \( h_{t-1} = \Phi^H r_{t-1} \) in 2) is computing the correlations between the current residual \( r_{t-1} \) and the columns of \( \Phi \). Here, we denote \( h_{t-1} \) as the \( N \times 1 \) correlation vector at the \( t \)-th iteration. Generally, the components of...
are complex numbers. The whole computational complexity of the OMP algorithm is dominated by the correlation step, and so are the other MPAs’ [9].

In this paper, we propose a new fast correlation method which can be applied to almost all MPAs including OMP, CoSaMP, ROMP, SP, and BAOMP. The recovery performances of the MPAs by the proposed method are exactly the same as those of the original MPAs. Also, the proposed correlation method can reduce the computational complexity of the MPAs well even though there are restrictions depending on a used MPA and parameters.

The proposed correlation method can be adopted when the sensing matrix is the partial Fourier matrix or the partial Hadamard matrix which are widely used in CS. In this paper, we utilize the following two common characteristics of N × N Fourier matrices and Hadamard matrices.

- (c1) Unitary matrix U whose every element has the magnitude 1/√N.
- (c2) The set {√N u1, √N u2, …, √N uN}, where u_n is the n-th column of U, is closed under element-wise multiplication o.

2. Notations and the Conventional Correlation Method for MPAs

2.1 Notations

Upper and lower case letters denote matrices and vectors, respectively. Also, we use (·)T and (·)H for transpose and complex conjugate transpose, respectively.

2.2 The Conventional Correlation Method for MPAs

Generally, the correlation step (h_t−1 = ΦH f_t−1) in MPAs can be performed by matrix-vector multiplication, which requires the large computational complexity. However, one can utilize FFT and FHT for the partial Fourier sensing matrix and the partial Hadamard sensing matrix, respectively, as

\[ h_{t-1} = \Phi^H f_{t-1} = U^H S_{Ω}^T f_{t-1} = \text{FFT}/\text{FHT}[S_{Ω}^T f_{t-1}] \]

where the sensing matrix is given as \( \Phi = S_{Ω} U \). The computational complexity of FFT or FHT is smaller than the matrix-vector multiplication, but it is still high.

3. A New Fast Correlation Method for MPAs

3.1 A New Fast Correlation Method

In this subsection, we describe the proposed fast correlation method for general MPAs. The MPAs have the common steps marked by φ in Algorithm 1, and we derive the fast correlation method for those steps. In the following derivation, U is an unitary matrix satisfying (c1) and (c2). The sensing matrix is given as \( \Phi = S_{Ω} U \).

Using the steps 5) and 7) in Algorithm 1, the correlation step 2) at the t-th iteration can be rewritten as

\[ h_{t-1} = \Phi^H f_{t-1} = \Phi^H y - \Phi^H a_{t-1} = \Phi^H y - \Phi^H \Phi S_{Ω}^T x_{t-1} = h_0 - U^H S_{Ω}^T S_{Ω} U S_{Ω}^T x_{t-1} \]  (3)

where \( h_0 = \Phi^H y \).

In (3), \( S_{Ω}^T x_{t-1} \) can be represented as

\[ S_{Ω}^T x_{t-1} = \sum_{τ=1}^{|Ω|} x_{t-1}(τ)e_{Ω−1}(τ) \]  (4)

where \( Ω−1(τ) \) is the τ-th element of the index set \( Ω−1 \), \( e_{Ω−1}(τ) \) is the \( Ω−1(τ) \)-th column of I, \( |Ω−1| \) is the cardinality of \( Ω−1 \), and \( x_{t-1}(τ) \) is the τ-th element of \( x_{t-1} \). Using (3) and (4), we obtain

\[ h_{t-1} = h_0 - \sum_{τ=1}^{|Ω|} x_{t-1}(τ) U^H S_{Ω}^T S_{Ω} U e_{Ω−1}(τ) \]

\[ = h_0 - \sum_{τ=1}^{|Ω|} x_{t-1}(τ) U^H S_{Ω}^T S_{Ω} D_{Ω−1}(τ) \frac{1}{\sqrt{N}} \]  (5)

where \( D_{Ω−1}(τ) = \sqrt{N} \cdot \text{diag}(u_{Ω−1}(τ)) \), and \( 1^T \) is the all-one column vector. Since \( S_{Ω}^T S_{Ω} \) is the \( N \times N \) diagonal matrix, (5) can be rewritten as

\[ h_{t-1} = h_0 - \sum_{τ=1}^{|Ω|} x_{t-1}(τ) U^H D_{Ω−1}(τ) S_{Ω}^T S_{Ω} D_{Ω−1}(τ) \frac{1}{\sqrt{N}} \]  (6)

where \( P_{Ω−1}(τ) = U^H D_{Ω−1}(τ) U \), and \( c = U^H S_{Ω}^T S_{Ω} \frac{1}{\sqrt{N}} \cdot 1^T \) is called the correlation kernel vector. Note that the correlation kernel vector \( c \) is independent to the sparse signal vector \( x \), so it can be computed and stored in advance. Also, \( S_{Ω}^T S_{Ω} \frac{1}{\sqrt{N}} \cdot 1^T \) of \( c \) is real and \( c \) becomes the conjugate symmetric signal vector. The matrix \( P_{Ω−1}(τ) \) in (6) is a permutation matrix, which will be proved in the following theorem. The multiplication of the permutation matrix, simple reordering the components of \( c \), can be performed with negligible computational complexity.

Theorem 1: \( P_{Ω−1}(τ) = U^H D_{Ω−1}(τ) U \) is a permutation matrix (i.e., a binary square matrix that has exactly one element 1 in each row and each column and 0s, elsewhere) if
the unitary matrix $U$ satisfies (c1) and (c2).

Proof: $U^H D_{\lambda_{-1}(\tau)} U$ can be expressed as

$$
U^H D_{\lambda_{-1}(\tau)} U = U^H \cdot \sqrt{N} \cdot \text{diag}(u_{\lambda_{-1}(\tau)}) \cdot (u_1 \cdots u_N)
$$

$$
= \frac{1}{\sqrt{N}} \cdot U^H \left( \sqrt{N} u_{\lambda_{-1}(\tau)} \circ \sqrt{N} u_t \cdots \sqrt{N} u_{\lambda_{-1}(\tau)} \circ \sqrt{N} u_N \right)
$$

(7)

where $\sqrt{N} u_{\lambda_{-1}(\tau)} \circ \sqrt{N} u_t$, $n = 1, \cdots, N$, are distinct column vectors because their elements are nonzero by (c1), and they belong to the set $\{ \sqrt{N} u_1, \sqrt{N} u_2, \cdots, \sqrt{N} u_N \}$ due to (c2). Therefore, (7) can be rewritten as

$$
U^H D_{\lambda_{-1}(\tau)} U = \frac{1}{\sqrt{N}} \cdot U^H \left( \sqrt{N} u_1 \cdots \sqrt{N} u_N \right) P_{\lambda_{-1}(\tau)}
$$

$$
= P_{\lambda_{-1}(\tau)}
$$

(8)

where $P_{\lambda_{-1}(\tau)}$ is the permutation matrix which is determined by $\lambda_{-1}(\tau)$ and the structure of $U$. \qed

To sum it up, the correlation step at the $t$-th iteration (i.e., computing $h_{t-1}$) can be performed by $|\Lambda_{-1}|$ subtractions of properly scaled and permutated versions of the correlation kernel vector $c$ to the initial correlation vector $h_0$ as in (6). The correlation kernel vector can be stored in advance, and the permutation matrices can be directly computed by $\lambda_{-1}(\tau)$ as explained in the following subsection.

Figure 1 shows the magnitude of elements in the correlation kernel vector $c$ and the initial correlation vector $h_0$ for 50 $\times$ 100 well-designed partial Fourier sensing matrix and $K = 5$. The partial Fourier sensing matrix in Fig. 1 is constructed by 50 rows from the 100 $\times$ 100 Fourier matrix based on a truncated (107, 53, 26) cyclic difference set suitable for the size of 50 $\times$ 100 [6]. The sparse signal vector $x$ has five nonzero elements, so $h_0$ has roughly five peaks. Also, the correlation kernel vector $c$ has one peak at the index one, which is already known for the well-designed partial Fourier sensing matrices. Moreover, the correlation kernel vector $c$ has the conjugate symmetric property due to the discrete Fourier transform property.

3.2 Permutation Matrices $P_{\lambda_{-1}(\tau)}$

In order to implement the proposed correlation method in (6), $N$ permutation matrices should be computed and stored in advance, but it is impractical. However, the permutation matrices can be easily determined by using the properties of the Fourier matrix and the Hadamard matrix to solve this storage problem.

3.2.1 $P_{\lambda_{-1}(\tau)}$ for the Partial Fourier Sensing Matrix

In this case, $P_{\lambda_{-1}(\tau)}$ can be directly determined by $\lambda_{-1}(\tau)$. Considering (7), (8), and the structure of the Fourier matrix whose columns consist of linear phases, it is easy to check that performing $P_{\lambda_{-1}(\tau)} c$ is the same as cyclically shifting $c$ by $\lambda_{-1}(\tau) - 1$. Thus, there is no need to store the permutation matrices.

3.2.2 $P_{\lambda_{-1}(\tau)}$ for the Partial Hadamard Sensing Matrix

Similarly, there is no need to store the permutation matrices. In this case, considering (7), (8), and the Sylvester’s Hadamard matrix structure, it is not difficult to check that $P_{\lambda_{-1}(\tau)}$ can be easily determined by $\lambda_{-1}(\tau)$ with simple bitwise operation as

$$
P_{\lambda_{-1}(\tau)} = (e(\lambda_{-1}(\tau) - 1) \oplus 0) + 1 \cdots (e(\lambda_{-1}(\tau) - 1) \oplus (N-1) + 1)
$$

(9)

where $(j)_2$ is the binary representation of $j$, and $\oplus$ denotes the bitwise exclusive OR.

4. A Fast OMP Recovery Algorithm

The proposed fast correlation method can be applied to many MPAs, among which the OMP algorithm is the most basic algorithm. In this section, we apply the proposed fast correlation method to the conventional OMP algorithm. Also, we discuss the computational complexity of the proposed OMP algorithm with the fast correlation method. Applying the proposed correlation method to other MPAs is straightforward and entirely analogous with this section.

4.1 The Fast Correlation Method for the OMP

The proposed correlation method (6) for a general MPA can be easily converted for the OMP as

$$
h_{t-1} = \begin{cases} 
\Phi^H r_0, & t = 1 \\
\Phi^H r_0 - \sum_{\tau=1}^{t-1} u_{\lambda_{-1}(\tau)} P_{\lambda_{-1}(\tau)} c, & t > 1
\end{cases}
$$

(10)

Also, the proposed OMP recovery algorithm can be given by simply replacing the correlation step 2) in Algorithm 1 with (10).

The only difference between the proposed and the conventional OMP algorithms is the simplified computation of correlation, so the proposed OMP algorithm guarantees the
same recovery performance as the conventional OMP algorithm with reduction of computational complexity.

4.2 Computational Complexity Analysis

In this subsection, we investigate the computational complexity of the proposed OMP algorithm in the cases of using the partial Fourier and the partial Hadamard sensing matrices. We remark that the proposed OMP algorithm performs the \( (t - 1) \) subtractions of properly scaled and permuted versions of the correlation kernel vector \( c \) at the \( t \)-th iteration to compute the correlation vector \( h_{t-1} \) in the second equation in (10).

We consider the case when \( N \) is a power of two, which is used very often in signal processing. Of course, the proposed method can also be used for any \( N \). For the exact comparison, we consider the number of flops of each algorithm. Here, we regard one complex multiplication as six flops and one complex addition as two flops.

4.2.1 Computational Complexity Using the Partial Fourier Sensing Matrix

The correlation step at each iteration in the conventional OMP algorithm can be implemented by the \( N \)-point FFT which requires \( (N/2) \log_2 N \) complex multiplications and \( N \log_2 N \) complex additions.

For the proposed OMP algorithm in (10), the first iteration \( (t = 1) \) can be implemented by one FFT. Meanwhile, when \( t > 1 \), the correlation vector is computed by using the correlation kernel vector. Exploiting the conjugate symmetric property of the correlation kernel vector using the partial Fourier sensing matrix, performing the second equation in (10) has the amount of \( (N/2) (t - 1) \) complex multiplications, \( 2(N/2) (t - 1) \) real additions, and \( N(t - 1) \) complex additions at the \( t \)-th iteration. Aggregately, the proposed OMP algorithm requires \( 6N(t - 1) \) flops at the \( t \)-th iteration.

4.2.2 Computational Complexity Using the Partial Hadamard Sensing Matrix

The correlation step at each iteration in the conventional OMP algorithm can be implemented by the \( N \)-point FHT which requires \( N \log_2 N \) complex additions.

For the proposed OMP algorithm in (10), the first iteration \( (t = 1) \) can be implemented by one FHT. Meanwhile, when \( t > 1 \), the correlation vector is computed by using the correlation kernel vector. Since the correlation kernel vector \( c \) consists of only a small number of integer coefficients compared to \( N \) using the partial Hadamard sensing matrix, the computational complexity for scaling by \( x_{t-1}(r) \) can be negligible. Therefore, performing the second equation in (10) has the amount of \( N(t - 1) \) complex additions. Table 1 summarizes this subsection.

4.2.3 A Restriction of the Proposed OMP Algorithm

Table 1 shows that the computational complexity of the proposed OMP algorithm at the \( t \)-th iteration is proportional to the value of \( t - 1 \), and thus there would be an excessive point. That is, for the value of \( t \) larger than the point, the proposed OMP algorithm has no computational benefit compared to the conventional OMP algorithm. When we use the partial Fourier sensing matrix, the proposed OMP algorithm has the computational benefit when the value of \( t \) is within the range as

\[
1 \leq t \leq \left\lfloor \frac{5}{6} \log_2 N + 1 \right\rfloor.
\]

Then, for \( t \) larger than the upper bound in (11), the conventional OMP algorithm has to be used.

Likewise, for the partial Hadamard sensing matrix, the proposed OMP algorithm has the computational benefit when the value of \( t \) is within the range as

\[
1 \leq t \leq \left\lfloor \log_2 N + 1 \right\rfloor.
\]

Then, for \( t \) larger than the upper bound in (12), the conventional OMP algorithm has to be used. To sum it up, the switch strategy is needed for some situations. An example of the switch strategy will be shown in Sect. 5.

5. Numerical Analysis

Here we present some numerical results characterizing the computational benefit of the proposed OMP algorithm compared to the conventional OMP algorithm. The results are given using the partial Fourier and the partial Hadamard sensing matrices with \( M = 64 \) and \( N \geq 1024 \), which are practical sizes for CS applications. For given \( M \) and \( N \), the cases when \( K \) is smaller than or equal to 10 are generally considered, because this range guarantees reliable recovery. Also, \( t \) is smaller than or equal to \( K \) in OMP algorithms. Therefore, we plot the computational complexities for \( 1 \leq t \leq 10 \) in Fig. 2 and Fig. 3.

Figure 2 shows the relative computational complexity of the proposed OMP algorithm compared to the conventional OMP algorithm when the partial Fourier sensing matrices are used. The proposed OMP algorithm has a benefit to reduce the computational complexity substantially for almost all the practical parameters. Especially, for large \( N \), the proposed OMP algorithm has more reduction of computational complexity. For instance, when \( M = 64 \), \( N = 8192 \), and \( K = 6 \), one can perform the OMP algorithm for six iterations (from \( t = 1 \) to \( t = 6 \)). In this case, the proposed OMP algorithm requires only 39.6% computational complexity in total compared to the conventional OMP algorithm while they run for six iterations.
Since the computational complexity of the proposed OMP algorithm at the $t$-th iteration is proportional to $t - 1$, there is an excessive point, so the switch strategy is needed for some situations as we mentioned. For instance, for $M = 64$ and $N = 1024$, there is the excessive point $t = 10$. In this case, the proposed OMP algorithm can be used from $t = 1$ to $t = 9$, and the conventional OMP algorithm can be used for the 10-th iteration. Similarly, we can choose a different recovery algorithm as occasion demands. This adaptability of the proposed method is one of its advantages.

Figure 3 shows the relative computational complexity of the proposed OMP algorithm compared to the conventional OMP algorithm when the partial Hadamard sensing matrices are used. This description of the proposed CoSaMP. The CoSaMP repetitively updates $K$ locations and coefficients of possible nonzero components at each iteration regardless of the iteration number $t$. Thus, different from the proposed OMP algorithm, the computational complexity of the proposed CoSaMP algorithm at each iteration is fixed for $t > 1$ and proportional to the value of $K$ as in Table 2. Therefore, the proposed CoSaMP algorithm has the computational benefit compared to the conventional CoSaMP algorithm when $N > 2^{t-2}K$ and when $N > 2^t$ for the partial Fourier sensing matrix and the partial Hadamard sensing matrix, respectively, and the conditions can be satisfied for an extremely sparse signal $x$. This is a restriction of the proposed CoSaMP algorithm.

Figure 4 shows the relative computational complexity of the proposed CoSaMP compared to the conventional CoSaMP at the $t$-th iteration, when the partial Fourier matrices are used. Especially, the proposed CoSaMP algorithm has a good computational benefit for small $K$ and large $N$ as we expected from Table 2. For instance, when $M = 64$, $N = 8192$, and $K = 4$, the proposed CoSaMP algorithm requires only the 37% computational complexity compared to the conventional CoSaMP algorithm except the first iteration.

6. Conclusions

In this paper, the fast correlation method for the MPAs with partial Fourier and partial Hadamard sensing matrices is presented. The proposed correlation method can be applied alternatively to almost all MPAs which are widely adopted in
CS. The proposed correlation method can reduce the computational complexity of the MPAs because the correlation step has a huge computational complexity in MPAs. The numerical results are given for the OMP and the CoSaMP algorithms with the proposed correlation method. The proposed correlation method has the computational benefit for the OMP and CoSaMP algorithms even though the proposed correlation method works effectively for an extremely sparse signal when the CoSaMP algorithm is used. Most importantly, this application does not cause any degradation of their recovery performance.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2012-0000186).

References


Kee-Hoon Kim received the B.S. and M.S. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science. His area of research interests includes compressed sensing, OFDM systems, and communications theory.

Hosung Park received the B.S., M.S., and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2007, 2009, and 2013, respectively. He was a postdoctoral researcher working with Prof. Jong-Seon No and Prof. Young-Han Kim at Institute of New Media and Communications in Seoul National University, Seoul, Korea, from March 2013 to August 2013. Since September 2013, he has worked with Prof. Young-Han Kim as a postdoctoral scholar in Department of Electrical and Computer Engineering, University of California, San Diego, USA. His research interests include low-density parity-check codes, coding theory, coding for memory, compressed sensing, network information theory, and network coding.

Seokbeom Hong received the B.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2007 and 2013, respectively. He is currently a senior engineer at Samsung Electronics, Gyeonggi-do, Korea. His area of research interests includes compressed sensing, error-correcting codes, and communications theory.

Jong-Seon No (S’80, M’88, SM’10, F’12) received the B.S. and M.S.E.E. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1988. He was a Senior MTS at Hughes Network Systems from February 1988 to July 1990. He was also an Associate Professor in the Department of Electronic Engineering, Konkuk University, Seoul, from September 1990 to July 1999. He joined the Faculty of the Department of Electrical and Computer Engineering, Seoul National University, in August 1999, where he is currently a Professor. His area of research interests includes error-correcting codes, sequences, cryptography, space-time codes, LDPC codes, and wireless communication systems.