Adaptive Filter-based Reconstruction Engine Design for Compressive Sensing

Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, 106, Taiwan, R.O.C.

Abstract—The reconstruction in compressive sensing is an underdetermined question. Almost all existing reconstruction algorithms utilize pseudo inverse to solve this problem. However, the matrix inverse in pseudo inverse has high complexity. In this paper, we apply least square filter (LMS) to signal reconstruction and propose a new reconstruction algorithm for compressive sensing. The results show that proposed method has good recovery performance and low computational complexity compared with existing works. Moreover, we implemented the proposed reconstruction algorithm in 90nm CMOS which operated at 200 MHz and occupied an area of 1.36mm². Throughput of the proposed method is 70% higher than state-of-the-art under the same cost.

Keywords—compressive sensing, sparse signal reconstruction, adaptive filter, least mean square filters

I. INTRODUCTION

Compressive sensing is a novel research and attracts intensive attention recently. It can measure signals with fewer samples than Nyquist theorem, and recover signals from these few samples. The main idea of compressive sensing is to combine sampling and compressing. It lays the foundation of two features, signal sparsity and sampling incoherence [1]. Signal sparsity indicates that natural signals, such as image or video signals, are compressible on proper bases. Also, incoherent sampling expresses that the system that should sample every component of the basis with equal probability.

We can model the system of compressive sensing as

\[ y_{M \times 1} = \Phi_{M \times N} x_{N \times 1} , \] (1)

where \( x_{N \times 1} \) is input vector of sparse signal with higher dimension \( N \), \( \Phi_{M \times N} \) is sampling matrix, \( y_{M \times 1} \) is measurement vector after sampling. The \( N \) stands for the input dimension, and \( M \) is the measurement size. Sparsity, denoted as \( K \), is defined as the nonzero term in the input signal, where low sparsity means only few elements in the input signal are nonzero and vice versa. Usually, we divide the system of compressive sensing into two parts: sampling and reconstruction. In sampling, [2] proved that random Gaussian matrix had extremely high probability to satisfy incoherent sampling. The reconstruction problem in compressive sensing is an underdetermined question and can be represented as

\[ \min_{x} \| x \|_1 \text{ s.t. } \Phi x = y , \| x \|_1 := \sum_{i} |x_i| . \] (2)

This is also an \( l1 \) minimization problem.

Basis pursuit (BP) [3] and orthogonal matching pursuit (OMP) [4] are two well-known reconstruction algorithms. Both of them can achieve optimal solution in compressive sensing. BP finds the \( l1 \) solution directly, and OMP refines \( l2 \) solution iteratively to achieve \( l1 \) solution. BP is noise resilient and has the best recovery performance, but the complexity is extremely high. OMP shows good performance in software simulation, while it has problem in hardware implementation. The calculation of matrix inverse in OMP poses burden on hardware cost. Furthermore, the sparsity information is necessary in hardware implementation of OMP.

In this work, we propose a new reconstruction algorithm based on adaptive filters. Adaptive filters are widely used in digital signal processing [5]. It can adjust its transfer function with respect to an optimization algorithm driven by an error signal. Before this work, [6] used adaptive filters to solve the underdetermined question in compressive sensing. However, the algorithm proposed by [6] still includes matrix inverse and the computational complexity is high. Different from [6], proposed algorithm has no matrix inverse operation and can achieve comparable recovery performance with BP. Furthermore, we implement proposed algorithm in ASIC design. The proposed reconstruction has high throughput rate under low cost and is reconfigurable for all kinds of sparsity.

This paper is organized as follows. We review existing reconstruction algorithms in section II. Section III, a new reconstruction algorithm based on adaptive filter is proposed. The recovery performance and the complexity analysis are presented in Section IV. Moreover, we propose hardware architecture of proposed reconstruction algorithm in Section V. Finally, we make a conclusion in Section VI.

II. REVIEW OF RECONSTRUCTION ALGORITHMS

A. Basis Pursuit (BP) [3]

BP decomposes a signal into an optimal superposition of dictionary elements, where optimal solution means that this solution has the smallest \( l1 \) norm of coefficients among all possible decompositions. Simpex method and interior point method are two algorithms in linear programming. Both of them provide a foundation for BP in solving the reconstruction problem for compressive sensing.

The columns of the sampling matrix formulate a space of polytope. BP-simplex applies simplex method to the reconstruction for compressive sensing. It finds a possible solution at a vertex of the polytope and then traces along the edges of the polytope to vertices with non-decreasing values of the objective function until the optimal solution is reached. On the other hand, BP-interior derives the procedure from interior point method in linear programming. It achieves an optimal solution by walking through the interior of the feasible region.
BP guarantees to find a solution with the least \( l_1 \) norm value in compressive sensing, but the computational complexity is very high.

B. Orthogonal Matching Pursuit [4]

Orthogonal Matching Pursuit is an iterative greedy algorithm. We can consider the measurement vector to be a linear combination of different columns of the sampling matrix. In this case, the input signal is weighting coefficients, and the measurement lies in the column space of the sampling matrix. At each iteration, OMP finds a column that is most correlated to the measurement, and collects them as a set. Next, OMP finds the least square solution based on the column set and the measurement. OMP utilizes pseudo-inverse to find the least square solution as shown below:

\[
x_i = (\Phi_p^T \Phi_p)^{-1} \Phi_p^T y, \Phi_p = \Phi P,
\]

where \( P \) is a position matrix of nonzero terms in the input signal, and \( \Phi_p \) is the column set collected by correlation. [4] provided both theoretical and empirical information that OMP could approach comparable performance with BP with much less computation than BP.

Based on the idea of BP and OMP, signal reconstruction with iteratively modifying \( l_2 \) solution is better because this can achieve the optimal solution with lower complexity. However, calculation of matrix inverse in OMP increases the cost in hardware implementation. To solve the problem of matrix inverse, we would like to apply adaptive filters for finding the \( l_2 \) solution.

III. PROPOSED RECONSTRUCTION ALGORITHM

A. Reconstruction Based on Adaptive Filters

We introduce adaptive filters to reconstruction for compressive sensing. Adaptive filters have been applied to several applications such as noise cancellation, signal identification, etc. There are two well-known adaptive filter algorithms, least mean square (LMS) filters and recursive least square (RLS) filters. The goal of LMS is to minimize the mean square error between the desired signal and the output of adaptive filter. On the other hand, the objective of RLS is to minimize a weighted linear least square cost function which accumulates the error performance in different iterations. RLS has faster convergence speed; however, the computational complexity is higher than LMS at the same time.

To utilize adaptive filters in recovering sparse signal for compressive sensing, (1) can be considered in different point of view. We can regard the input signal \( x \) as a system, and the sampling matrix \( \Phi \) as input signal. Accordingly, the problem of reconstruction is transformed into a problem of system identification, which the weighting function of adaptive filters adjusts itself by the error between the measurement and the output of adaptive filters.

We propose a reconstruction algorithm based on adaptive filters, and we choose LMS filters due to its low complexity.

B. Sparse Reconstruction via LMS

LMS finds the solution with minimum mean square error (MMSE):

\[
\hat{x} = w_{\text{MMSE}} = (\Phi^T R_x \Phi + R_q)^{-1} \Phi^T R_x y.
\]

where \( R_x \) represents the autocorrelation of input signal \( x \), \( R_q \) denotes the autocorrelation of noise. LMS achieves the MMSE solution, which is also the minimum \( l_2 \) solution. However, the goal of reconstruction is to find the minimum \( l_1 \) solution, which is different from MMSE solution. Thus, we combine the location of nonzero terms and LMS to reconstruct sparse signal according to the idea in OMP. Furthermore, the autocorrelation of the input signal approaches to identity matrix for sparse signals with power, \( \varepsilon_x \). Then, for a position matrix of nonzero terms, \( P \), the approximated signal can be represented as:

\[
\hat{x} = (\Phi_p^T \Phi_p + \frac{1}{\varepsilon_x} R_q)^{-1} \Phi_p^T y, \Phi_p = \Phi P.
\]

Compare with (3), the approximated signal is identical to the result of OMP in noiseless environment. In other word, we can utilize LMS to recover signal for compressive sensing.

C. Proposed Adaptive Correlating Pursuit (ACP)

Based on previous discussion, we proposed a new reconstruction algorithm with adaptive filters, called adaptive correlating pursuit (ACP). The ACP algorithm is summarized and presented in Algorithm 1. In proposed algorithm, we utilize correlation to localize the nonzero term, and apply LMS filters to approximate the nonzero value in the input signal. With LMS filters, ACP can recover signal without matrix inverse operation. Moreover, correlation is inner product of the residual and the columns of sampling matrix. The position with larger correlation implies it has higher probability to be nonzero.

The proposed method combines the position information from correlation and the approximated value from LMS to reconstruct the input signal. The block diagram of the proposed ACP is shown in Fig. 1.

Algorithm 1: Adaptive Correlating Pursuit (ACP)

1. Estimate \( K \) from \( M \geq C \cdot K log \left( \frac{K}{M} \right) \), where \( C \) is a constant.
2. Initialize \( r_0 = y, \hat{x}_0 = 0, \text{dis} = 1, n = 1 \).
3. \( \text{corr} = |\Phi_t r_{n-1}| \).
4. \( w_0 = \hat{x}_{n-1} \).
5. for \( m = 1, 2, ..., M \)
   \[
   \phi_m = \text{the mth row of } \Phi.
   \]
   \[
   w_m = \text{LMS}(\phi_m y, w_{m-1}).
   \]
end
6. \( P = \text{position of largest} \ K \ \text{elements in corr.} \)
7. \( \hat{x}_n = 0, \text{and } \hat{x}_n(P) = w_m(P) \).
8. \( r_n = y - \Phi \hat{x}_n \).
9. \( \text{dis} = \frac{\text{abs}(r_n)}{\text{abs}(y)} \).
10. \( n = n + 1 \).
11. If \( \text{dis} > 10^{-5} \) and \( n < M \), go back step 3;
otherwise, output \( \hat{x}_n \).
IV. PERFORMANCE ANALYSIS

A. Phase Transition of Reconstruction

In first experiment, the input dimension is 1000. Fig. 2 presents the transition phase of ACP and BP. Transition phase shows the recovery performance of an algorithm for different measurement and different sparsity. If the normalized root mean square error (RMSE) is less than $10^{-3}$, this trial is called successful reconstruction. The line stands for the 50% recovery probability among total 100 trials. The recovery probability above the line is lower than 50% and below the line is higher than 50%. The average distance between the two lines in Fig. 2 is 0.0077, which means proposed ACP has comparable reconstruction performance with BP.

B. Analysis of Computational Complexity

Table I presents the computational complexity in both OMP and ACP. The difference of OMP and ACP lies on the pseudo inverse and LMS. OMP requires a matrix inversion of $K$ by $K$ matrix. As the sparsity is higher, the computational cost of OMP increases. Moreover, for specific sparsity ratio, the complexity of OMP will be higher than ACP in high input dimension, too. Thus, ACP has advantages in the cases of large input dimension or high input sparsity.

Furthermore, matrix inverse constrains the hardware design of OMP. Although the sparsity information is not essential for OMP algorithm itself, it becomes necessary in OMP hardware implementation to control the cost. If the sparsity is unknown, the matrix inverse in OMP needs to be designed for the worst case, $M$ by $M$ matrix inverse. This greatly increases the hardware cost. In contrast, ACP does not have this concern and single hardware design of ACP can be used for all kinds of sparsity.

<table>
<thead>
<tr>
<th></th>
<th>Correlation O(NM)</th>
<th>Identify O(N)</th>
<th>Pseudo Inverse O(K^3)</th>
<th>LMS O(NM)</th>
<th>Residual O(NM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMP</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>ACP</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
</tbody>
</table>

V. VLSI IMPLEMENTATION

A. Architectural Design

The major cost in reconstruction engine is occupied by the calculation of correlation, LMS, and residual. Correlation includes vector multiplication. LMS has vector multiplication, and vector addition. Residual contains vector multiplication. We optimize the architecture with remapping correlation and shared hardware to reduce hardware cost.

Correlation is defined as the inner product of residual and the columns of the sampling matrix as presented in (6).

$$ corr_{N 	imes 1} = \Phi \cdot r $$

There are total $N$ columns in the sampling matrix; therefore, we can organize the calculation of correlation into $N$ cycles. However, the calculation of both LMS and residual is dependent on the rows of the sampling matrix, and they require $M$ cycles. In order to synchronize all calculation in reconstruction, we remap the calculation in correlation. Correlation can be calculated by multiplying each element in residual with each row of the sampling matrix as shown in (7). After $M$ cycle, the results of correlation can be obtained as shown in (8).

$$ c(i) = \Phi(i,:) \cdot r \quad \text{for } 1 \leq i \leq M $$

$$ corr_{N 	imes 1} = c(M) $$

After remapping correlation, the calculation of correlation, LMS, and residual are all processed in $M$ cycles. We use parallel multipliers and adder tree for vector multiplication. Moreover, we apply parallel adders for vector addition.

To further reduce hardware cost, we want to share resources among proposed design. To begin with, we divide the computation of LMS into two stages. The first stage includes parallel multipliers and adder tree, and the second stage has another block of parallel multipliers and parallel adders. Next, we find the calculation of residual is similar to the first stage of LMS, and the second stage of LMS has same working model as correlation. Thus, we can share hardware among them and reduce cost. Before optimization, the architecture requires four blocks of parallel multipliers, two groups of parallel adders, and two adder trees. After optimization, there are only two blocks of parallel multipliers, one group of parallel adders, and one adder tree in final architecture as shown in Fig. 3. We reduce around 50% cost with shared hardware.
B. Comparison and Implementation Results

Proposed reconstruction engine is implemented in 90nm technology and can achieve 200 MHz. The chip results are shown in Table II and the final layout is shown in Fig. 4. To increase the bandwidth of memory, we divide the whole memory into eight register files. The register files are placed around the boundary of the chip because this can leave more complete space for combinational cells. Except for the memory, combinational cells including multipliers and adders occupy most of area in this design. The area of final automatic place and route is 1.36 mm$^2$ with core utilization 0.7.

![Fig. 3. Optimized architecture of proposed ACP reconstruction engine](image)

**TABLE II. CHIP RESULTS OF PROPOSED RECONSTRUCTION ENGINE**

<table>
<thead>
<tr>
<th>Input Dimension</th>
<th>Measurement Size</th>
<th>Cell Library</th>
<th>Frequency</th>
<th>Core Utilization</th>
<th>Core Area</th>
<th>Latency (K = 12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>64</td>
<td>90nm</td>
<td>200 MHz</td>
<td>0.7</td>
<td>1.36 mm$^2$</td>
<td>55.37 us</td>
</tr>
</tbody>
</table>

We compare the proposed reconstruction with [8], and the results are presented in Table III. We evaluate the hardware performance of a reconstruction engine with normalized area efficiency defined as:

$$\text{Normalized Area Efficiency (NAE)} = \frac{\text{throughput}}{\text{gate count}} \cdot \frac{\text{unit: bits/s \cdot gate}}. \quad (9)$$

[8] designed a OMP reconstruction engine. The input dimension in Table III is 256, and the measurement dimension is 64. We define a parameter, normalized area efficiency, to evaluate the hardware performance of a reconstruction engine. According to Table III, we can know that proposed reconstruction engine has better efficiency to reconstruction a sparse signal in high sparsity. With 12 nonzero terms among total 256 elements, the throughput of proposed ACP is 61% higher than [8] under the same cost. Moreover, different sparsity only affects the throughput of proposed design and does not increases hardware cost, which means proposed reconstruction is reconfigurable.

![Fig. 4. Final layout of proposed reconstruction engine](image)

**TABLE III. HARDWARE COMPARISON WITH STATE-OF-THE-ART**

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>Freq.</th>
<th>Normalized Throughput</th>
<th>Gate Count</th>
<th>NAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMP [8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>165 MHz</td>
<td>324 M 531 K</td>
<td>610</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>133 M* 797 K*</td>
<td>126 M 412 K 269</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>Proposed ACP</td>
<td>200 MHz</td>
<td>126 M 412 K 269</td>
<td>305</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>111 M 412 K 269</td>
<td>126 M 412 K 269</td>
<td>305</td>
<td></td>
</tr>
</tbody>
</table>

* the statistics is estimated according to the architecture in [8]

VI. CONCLUSION

We proposed adaptive correlating pursuit for signal reconstruction in compressive sensing. Proposed ACP has good recovery performance and lower complexity. In ASIC design, ACP achieves higher throughput with the same hardware cost compared with state-of-the-art. Furthermore, ACP is reconfigurable in different sparsity. In conclusion, proposed ACP is a low complexity and hardware friendly reconstruction algorithm for compressive sensing.

REFERENCE