sensors

Article

A Two-Stage Reconstruction Processor for Human Detection in Compressive Sensing CMOS Radar

Kuei-Chi Tsao [1], Ling Lee [2], Ta-Shun Chu [1] and Yuan-Hao Huang [2,*]

[1] Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan; andrew000129@gmail.com (K.-C.T.); tschu@ee.nthu.edu.tw (T.-S.C.)

[2] Institute of Communications Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan; leelinpp@gmail.com

* Correspondence: yhhuang@ee.nthu.edu.tw; Tel.: +886-3-516-2213

Received: 28 February 2018; Accepted: 3 April 2018; Published: 5 April 2018

Abstract: Complementary metal-oxide-semiconductor (CMOS) radar has recently gained much research attraction because small and low-power CMOS devices are very suitable for deploying sensing nodes in a low-power wireless sensing system. This study focuses on the signal processing of a wireless CMOS impulse radar system that can detect humans and objects in the home-care internet-of-things sensing system. The challenges of low-power CMOS radar systems are the weakness of human signals and the high computational complexity of the target detection algorithm. The compressive sensing-based detection algorithm can relax the computational costs by avoiding the utilization of matched filters and reducing the analog-to-digital converter bandwidth requirement. The orthogonal matching pursuit (OMP) is one of the popular signal reconstruction algorithms for compressive sensing radar; however, the complexity is still very high because the high resolution of human respiration leads to high-dimension signal reconstruction. Thus, this paper proposes a two-stage reconstruction algorithm for compressive sensing radar. The proposed algorithm not only has lower complexity than the OMP algorithm by 75% but also achieves better positioning performance than the OMP algorithm especially in noisy environments. This study also designed and implemented the algorithm by using Vertex-7 FPGA chip (Xilinx, San Jose, CA, USA). The proposed reconstruction processor can support the 256 × 13 real-time radar image display with a throughput of 28.2 frames per second.

Keywords: compressive sensing; CMOS radar; ranging

1. Introduction

The complementary metal-oxide-semiconductor (CMOS) radar system has recently drawn much research attention because of the great demand of low-power actively sensing devices for the internet-of-thing (IOT) system [1–3]. The human-centric home-care system is one of the most potential applications of the CMOS radar system, in which an impulse-radio radar transceiver can transmit high-frequency impulses to scan the weak respiration signals and identify the human body in the noisy environments [4–6]. However, the high-resolution signal increases the dimension of grid space, leading to the increased computational complexity of the target detection. Thus, the target detection and weak human feature extraction are two important signal processing issues for the human-centric CMOS radar system.

Traditional human detection algorithms detected respiration frequency in either time domain or frequency domain [7–9]. However, most of them were developed by using simple sinusoidal respiration model. Our previous work [10] designed a CMOS impulse radar system and developed a respiration feature extraction algorithm for detecting fine human features. The proposed single-input single-output (SISO) impulse radar system can detect single human body in a tunable fixed distance (1D distance) and extract the respiration features based on a four-segment linear waveform model.

The target detection and localization is another signal processing issue for the impulse radar system because of its great computational complexity. Traditional radar systems usually achieve high-accuracy target detection regardless of the computational cost because the radar station is usually equipped with high-performance digital signal processors. For an IOT system, the computational power of the target detection algorithm becomes a critical design issue because of the low-power requirement of wireless sensing nodes. Therefore, many researchers [11,12] applied compressive sensing (CS) reconstruction algorithms to the object detection to eliminate high-power matched filters and high-bandwidth analog-to-digital converters. One the other hand, some researchers designed the shape of the transmitted impulse to satisfy the incoherent constraint of CS. Thus, the reconstruction algorithm can rebuild the target scene with better performance and lower complexity than a classical radar in the literature [13,14].

The reconstruction algorithm dominates the signal processing complexity of a CS radar system [15]. l_1-minimization (basis pursuit) and greedy reconstruction are two typical algorithms. l_1-minimization [12,13] has robust reconstruction performance but involves very high complexity. Greedy reconstruction algorithms, such as orthogonal matching pursuit (OMP), has worse performance than l_1-minimization. Thus, the greedy reconstruction algorithms [16–21] are more suitable for low-power hardware implementation. For a certain CS-radar-scanned area, higher resolution leads to higher grid density and higher reconstruction complexity. This design issue is especially stringent for a human-centric wireless sensing system because the respiratory vibration is extremely fine and requires a high resolution radar system. This work proposes a low-complexity two-stage OMP reconstruction algorithm for a single-input multiple-output (SIMO) CS radar. The contributions of the proposed algorithm are twofold. Firstly, the computational complexity of the proposed algorithm is significantly lower than the OMP algorithm, especially when the resolution of the CS radar system increases. Secondly, the detection performance of the proposed algorithm is better than that of the OMP algorithm especially in noisy environments. Finally, a reconstruction processor was designed and implemented for a real-time radar image display for the single-input and multiple-output (SIMO) CS radar system.

The rest of this paper is organized as follows. Section 2 introduces the signal model of the CS radar system and the signal reconstruction algorithms for target detection. Section 3 presents the proposed low-complexity reconstruction algorithm. Section 4 analyses the computational complexity and performance of the proposed algorithm. Section 5 presents the FPGA design and implementation of the proposed reconstruction processor. Finally, conclusions are given in Section 6.

2. Compressive Sensing Radar System

2.1. Compressive Sensing

The measurement process of compressive sensing system can be expressed by

$$\mathbf{y} = \mathbf{\Phi r} \tag{1}$$

where $\mathbf{y}$ is an M × 1 sensing vector; $\mathbf{\Phi}$ is an M × N measurement matrix; and $\mathbf{r}$ is an N × 1 vector of the original signal with a higher dimension than $\mathbf{y}$. Signal sparsity and incoherence of sensing matrix are requisites of efficient signal reconstruction [22,23]. $\mathbf{r}$ can be described by

$$\mathbf{r} = \mathbf{\Psi x} \tag{2}$$

where $\mathbf{\Psi}$ is an N × N basis matrix of the sparse domain and $\mathbf{x}$ is an N × 1 coefficient vector in the sparse domain. There are only a few non-zero elements in $\mathbf{x}$. Using $\mathbf{\Psi}$ as basis, signal $\mathbf{r}$ can be transformed to the sparse domain as $\mathbf{x}$ with only K non-zero elements. K is much smaller than N. Then, the overall compressive sensing signal model can be expressed by

$$\mathbf{y} = \mathbf{\Phi r} = \mathbf{\Phi \Psi x} = \mathbf{Ax} \tag{3}$$

where $\mathbf{A} = \mathbf{\Phi\Psi}$ is the sensing matrix of the CS system.

2.2. SISO Compressive Sensing Radar System

In a single-input single-output compressive sensing impulse radar system, the received signal can be expressed by

$$y(t) = \sum_i p_i s(t - \tau_i) \tag{4}$$

where $s(t)$ is transmitted impulse signal. The i-th target reflects the transmitted impulse signal with signal gain p_i and propagation delay τ_i. d_i is the distance between the radar and the i-th target and can be calculated by $d_i = \frac{\tau_i c}{2}$, where c is the light velocity. The received signal can be also rewritten as follows:

$$y(t) = \sum_i \alpha_i z(t; d_i) \tag{5}$$

where $z(t; d_i)$ is the echo signal of the i-th target and α_i is the signal gain coefficient determined by the path loss and other properties of the target, such as energy reflection/absorption ratio of the target.

2.3. MIMO Compressive Sensing Radar System

A multiple-input multiple-output (MIMO) radar system has many advantages over a SISO radar system [24]. The MIMO radar can scan targets in a 2-D space by using phase array antennas [25]. On the other hand, multiple transmitted impulse signals can be properly designed to reduce the degree of coherence of the compressive sensing system so as to improve the signal reconstruction accuracy [13,14]. The received signal of the MIMO compressive sensing radar can be expressed by

$$y(t) = \sum_k \alpha_k z(t; \phi_k, d_k) \tag{6}$$

where α_k is amplitude coefficient of the k-th target, and ϕ_k and d_k are the direction (angle) and distance of the k-th target, respectively. The echo signal $z(t; \phi, d)$ is extended from $z(t; d)$ in Equation (5) with an additional dimension ϕ .

Assume that a MIMO radar system has $\mathrm{N_R}$ receive antennas and $\mathrm{N_T}$ transmit antennas, and each receive antenna acquires $\mathrm{N_S}$ samples. The echo signal matrix of a target with distance d and angle ϕ is denoted as $\mathbf{Z}(t; \phi, d)$. Echo signal $z(t; \phi, d)$ in Equation (6) is a vectorized version of the echo signal matrix $\mathbf{Z}(t; \phi, d)$. Echo signal matrix $\mathbf{Z}(t; \phi, d)$ is an $\mathrm{N_R} \times \mathrm{N_S}$ matrix, and echo signal $z(t; \phi, d)$ is an $\mathrm{N_R N_S} \times 1$ vector. The echo signal matrix $\mathbf{Z}(t; \phi, d)$ can be constructed by using uniformly-spaced linear array (ULA) [12] and can be expressed as follows:

$$\mathbf{Z}(t; \phi, d) = \mathbf{a_R}(\phi) \mathbf{a_T}^{\mathrm{T}}(\phi) \mathbf{S}(t - \tau) \tag{7}$$

where $\mathbf{a_R}(\phi)$ is an $\mathrm{N_R} \times 1$ receive matrix, $\mathbf{a_T}(\phi)$ is an $\mathrm{N_T} \times 1$ transmit matrix, and $\mathbf{S}(t - \tau)$ is an $\mathrm{N_T} \times \mathrm{N_S}$ transmitted signal matrix. The delay τ can be calculated by using $\tau = \frac{2\mathrm{d}}{\mathrm{c}}$. In the transmitted signal matrix $\mathbf{S}(t - \tau)$, each row of the matrix are the delayed transmitted signal $\mathbf{s}_i(t - \tau)$ of the i-th transmit antenna. Transmit matrix $\mathbf{a_T}(\phi)$ and receive matrix $\mathbf{a_R}(\phi)$ are given by Equations (9) and (10), respectively, in which d_T and d_R are the normalized spacings (distance divided by wavelength) of the transmit antennas and receive antennas, respectively [12]. According to Equations (8)–(10), echo signal matrix $\mathbf{Z}(t; \phi, d)$ in Equation (7) can be established. Then, echo signal vector $z(t; \phi, d)$ in Equation (6) can be generated by vectorizing the echo signal matrix $\mathbf{Z}(t; \phi, d)$.

$$\mathbf{S}(t-\tau) = \begin{bmatrix} \cdots & \mathrm{N_s} \text{ samples of } \mathbf{s_1}(t-\tau) & \cdots \\ \cdots & \mathrm{N_s} \text{ samples of } \mathbf{s_2}(t-\tau) & \cdots \\ & \vdots & \\ \cdots & \mathrm{N_s} \text{ samples of } \mathbf{s}_{\mathrm{N_T}}(t-\tau) & \cdots \end{bmatrix} \tag{8}$$

$$\mathbf{a_T}(\phi) = \begin{bmatrix} 1 & e^{j2\pi d_T \sin(\phi)} & \ldots & e^{j2\pi d_T \sin(\phi)(\mathrm{N_T}-1)} \end{bmatrix}^{\mathrm{T}} \tag{9}$$

$$\mathbf{a_R}(\phi) = \begin{bmatrix} 1 & e^{j2\pi d_R \sin(\phi)} & \ldots & e^{j2\pi d_R \sin(\phi)(\mathrm{N_R}-1)} \end{bmatrix}^{\mathrm{T}} \tag{10}$$

The target scene of an MIMO radar system is a discrete $\mathrm{N_r} \times \mathrm{N}_\theta$ range-azimuth grid.

When a target is at the range-azimuth grid point (θ_i, r_j) within the coverage of the MIMO radar system, that is, $1 \le i \le \mathrm{N}_\theta$ and $1 \le j \le \mathrm{N_r}$, the received signal can be expressed by

$$y(t) = \sum_{i=1}^{\mathrm{N}_\theta} \sum_{j=1}^{\mathrm{N_r}} x_{ij} z(t;\theta_i, r_j), \tag{11}$$

where $z(t;\theta_i, r_j)$ is the echo signal of the target at (θ_i, r_j) and x_{ij} is the signal gain coefficient. If there is no target at (θ_i, r_j), then $x_{ij} = 0$. The object detection and localization is to find a set of non-zero x_{ij}s in Equation (11). Equation (11) can be expressed in matrix form in Figure 1. The number of targets, denoted as K, is much smaller than the number of grid points ($\mathrm{K} \ll \mathrm{N}_\theta \mathrm{N_r}$). Thus, the matrix form in Figure 1 can be regarded as a CS signal model and sensing matrix $\mathbf{A}$ is composed of $z(t;\theta_i, r_j)$. The object detection searches a set of echo signals in the received signal, and indices of echo signals represent their locations.

$$\underset{\mathrm{N_R N_S \times 1}}{\begin{bmatrix} y \end{bmatrix}} = \underset{\substack{\mathrm{N_R N_S \times N_r N_\theta} \\ \text{Sensing Matrix}}}{\begin{bmatrix} z(t;\theta_1, r_1) & z(t;\theta_1, r_2) & \cdots & z(t;\theta_i, r_j) & \cdots & z(t;\theta_{N_\theta}, r_{N_r}) \end{bmatrix}} \underset{\mathrm{N_r N_\theta \times 1}\ \text{Range-Azimuth Grid}}{\begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{ij} \\ \vdots \\ x_{N_\theta N_r} \end{bmatrix}}$$

where

$$\mathrm{y} = \Big[\underbrace{\mathrm{y_{Rx1}}(1) \ \ \mathrm{y_{Rx2}}(1) \ \cdots \ \mathrm{y_{RxN_R}}(1)}_{\text{Sample 1}} \ \ \underbrace{\mathrm{y_{Rx1}}(2) \ \ \mathrm{y_{Rx2}}(2) \ \cdots \ \mathrm{y_{RxN_R}}(2)}_{\text{Sample 2}} \ \cdots \ \underbrace{\mathrm{y_{Rx1}}(\mathrm{N_s}) \ \cdots \ \mathrm{y_{RxN_R}}(\mathrm{N_s})}_{\text{Sample Ns}} \Big]^{\mathrm{T}}$$

Figure 1. Multiple-input multiple-output (MIMO) compressive sensing radar model.

2.4. Path Loss and Human Respiration Signal Model

Path loss of the travelling electromagnetic wave determines the signal gain coefficient x_{ij}. This work utilizes a radar system [10] integrated with a CMOS impulse radar chip [1] to measure the path loss parameters of metallic objects and human bodies, as shown in Figure 2a,b. The CMOS radar system transmits 25 dBm 1 GHz sinusoidal impulse with a repetitive 10 MHz frequency, resulting in 0.94 mm range resolution. Figure 3a shows the measured results of the path loss versus logthmic radius distance. The power attenuation of human is more than 20 dB than that of a mental object due to the high energy absorption ratio of human body. The 0.94 mm resolution is fine enough to capture tiny respiratory information. This study adds four-segment linear waveform (FSLW) [10] respiration signals into the echo signals of human target to simulate the practical human respiration.

Based on the measurement results, this study constructs a SIMO impulse radar system with 4 receive antennas and 1 transmit antenna, which are configured as shown in Figure 3b. Each receive antenna receives 128 samples for each iteration. The maximum detection range is 5 m, and direction (angle) is between +45 and −45 degrees. The range resolution is $N_r = 256$ and angle resolution is $N_\theta = 13$, that is, target scene forms 256 × 13 range-azimuth grid. The sensing matrix of the CS radar system is a 512 × 3328 matrix. $\mathbf{a_T}(\phi)$ and $\mathbf{a_R}(\phi)$ are determined by (9) and (10) with $N_t = 1$ and $N_r = 4$, respectively. The receive antenna spacing d_R is $\frac{\lambda}{2}$. There are three targets in the scene. One is a non-human object and the others are human bodies, which are modelled by the FSLW respiration model [10] with intensity $A = 0.01$ m, inspiration speed $\beta_1 = 0.5$, expiration speed $\beta_2 = 0.5$, and respiration holding ratio $X = 0.5$. The respiration rate of two persons are 0.5 Hz and 0.8 Hz, respectively. The non-human target is located at the distance 4.5 m and angle 22.5°. One person is located at the distance 2 m and angle −7.5°, and the other is located at distance 3.5 m and angle 30°.

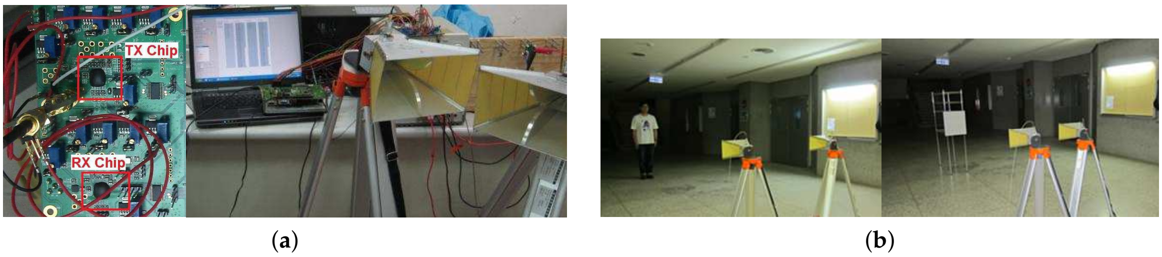


Figure 2. (**a**) The complementary metal-oxide-semiconductor (CMOS) radar system [10] and (**b**) measurement environments.

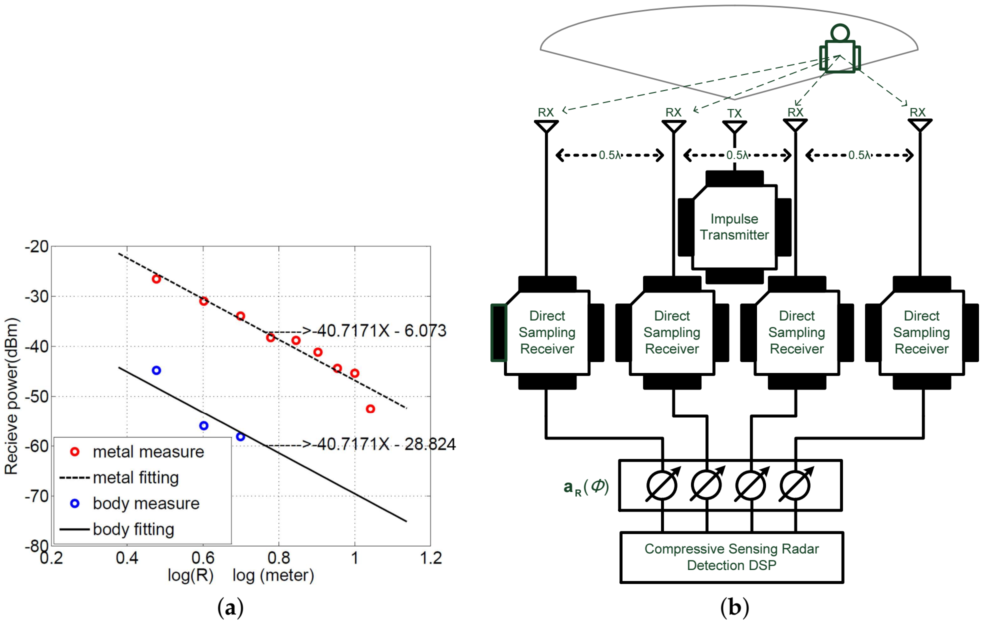


Figure 3. (**a**) Single-input multiple-output (SIMO) radar configuration and (**b**) the measured path loss models of the human body and metal object.

2.5. Reconstruction Algorithms for Compressive Sensing

Orthogonal matching pursuit (OMP) [26] is one of the popular greedy reconstruction algorithms for compressive sensing. For a compressive sensing framework $\mathbf{y} = \mathbf{A}\mathbf{x}$, $\mathbf{y}$ is M × 1 measurement vector, $\mathbf{A}$ is M × N sensing matrix, and $\mathbf{x}$ is N × 1 signal vector. First, the OMP algorithm searches the support set iteratively according to the matching results by

$$\alpha = \mathbf{A}^T\mathbf{y} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \end{bmatrix}^T \mathbf{y} \tag{12}$$

The determined index with the maximum matched gain is added into the support set $\mathcal{I}_k$ of the k-th iteration. Then, the estimated signal vector of the k-th iteration, denoted as $\hat{\mathbf{x}}_k$, can be reconstructed by $\hat{\mathbf{x}}_k = \mathbf{A}^{\dagger}_{\mathcal{I}_k}\mathbf{y}$, where $\mathbf{A}^{\dagger}_{\mathcal{I}_k}$ is the pseudo-inverse of $\mathbf{A}_{\mathcal{I}_k}$. Then, the residual of the k-th iteration can be calculated by $\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{\mathcal{I}_k}\hat{\mathbf{x}}_k$. The residual $\mathbf{r}_k$ is then used to match the index of the next iteration by $\boldsymbol{\alpha}_k = \mathbf{A}^{\mathrm{T}}_{\tilde{\mathcal{I}}_k}\mathbf{r}_k$, where $\tilde{\mathcal{I}}_k$ is complementary set of support set $\mathcal{I}_k$. Algorithm 1 shows the pseudo code of the OMP algorithm.

Algorithm 1 Orthogonal Matching Pursuit Algorithm

```
Input: Sensing matrix $\mathbf{A}$, measurement $\mathbf{y}$, target sparsity K
Output: Support set $\mathcal{I}_k$, estimate $\hat{\mathbf{x}}_k$
1: Initialization:
      • $k = 1; \mathcal{I}_0 = \phi;$
      • $\mathbf{r}_0 = \mathbf{y}; \boldsymbol{\alpha}_0 = \mathbf{A}^T\mathbf{r}_{k-1}$
2: repeat
3:    $i_k = \arg\max(|\boldsymbol{\alpha}_{k-1}|)$
4:    $\mathcal{I}_k = \mathcal{I}_{k-1} \cup \{i_k\};$
5:    $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}^{\dagger}_{\mathcal{I}_k}\mathbf{y};$
6:    $\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{\mathcal{I}_k}\hat{\mathbf{x}}_{\mathcal{I}_k};$
7:    $\boldsymbol{\alpha}_k = \mathbf{A}^T_{\tilde{\mathcal{I}}_k}\mathbf{r}_k$
8:    k=k+1;
9: until $(k \geq K)$
```

2.6. Orthogonal Matching Pursuit via Matrix Inversion Bypass

The computational bottleneck of the OMP algorithm is the matrix inversion in Step 5 of Algorithm 1. Thus, this study uses the Schur-Banachiewicz block-wise inversion [27,28] to reduce the complexity. The derived OMP algorithm is called the orthogonal matching pursuit via matrix inversion bypass (OMP-MIB) [29]. We denote $\mathrm{N} \times \mathrm{N}$ matrix $\mathbf{G} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{G}_{\mathcal{I},\mathcal{J}} = \mathbf{A}^{\mathrm{T}}_{\mathcal{I}}\mathbf{A}_{\mathcal{J}}$, where $\mathcal{I}$ and $\mathcal{J}$ are any two support sets. Thus, the estimated signal vector can be rewritten as

$$\hat{\mathbf{x}}_k = \mathbf{A}^{\dagger}_{\mathcal{I}_k}\mathbf{y} = \mathbf{G}^{-1}_{\mathcal{I}_k,\mathcal{I}_k}\mathbf{A}^{\mathrm{T}}_{\mathcal{I}_k}\mathbf{y}, \tag{13}$$

where $\mathbf{G}_{\mathcal{I}_k,\mathcal{I}_k}$ is a $\mathrm{K} \times \mathrm{K}$ matrix. Schur-Banachiewicz block-wise inversion is then applied to Equation (13). Hence, matrix inversion in Equation (13) can be expressed by

$$\begin{aligned}\mathbf{G}^{-1}_{\mathcal{I}_k,\mathcal{I}_k} &= \begin{bmatrix} \mathbf{G}_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} & \mathbf{G}_{\mathcal{I}_{k-1},i} \\ \mathbf{G}_{i,\mathcal{I}_{k-1}} & \mathbf{G}_{i,i} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{G}^{-1}_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} + \mathbf{W}^{\mathrm{T}}\mathbf{V}\mathbf{W} & -\mathbf{W}^{\mathrm{T}}\mathbf{V} \\ -\mathbf{V}\mathbf{W} & \mathbf{V} \end{bmatrix}\end{aligned} \tag{14}$$

where $\mathbf{W}$ is $1 \times (\mathrm{K}-1)$ matrix is

$$\mathbf{W} = \mathbf{G}_{i,\mathcal{I}_{k-1}}\mathbf{G}^{-1}_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} \tag{15}$$

and $\mathbf{V}$ is a scalar:

$$\mathbf{V} = \left(\mathbf{G}_{i,i} - \mathbf{G}_{i,\mathcal{I}_{k-1}}\mathbf{G}^{-1}_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}\mathbf{G}_{\mathcal{I}_{k-1},i}\right)^{-1} \tag{16}$$

Equation (14) shows that the matrix inversion of $\mathbf{G}_{\mathcal{I}_k,\mathcal{I}_k}$ can be replaced by several matrix operations. Since $\mathbf{V}$ is a simple reciprocal operation, matrix inversion operation is not required in Equation (14). The pseudo code of the OMP-MIB algorithm is shown in Algorithm 2.

Algorithm 2 Orthogonal Matching Pursuit via Matrix Inversion Bypass

```
Input: Sensing matrix $\mathbf{A}$, measurement $\mathbf{y}$, target sparsity $K$
Output: Reconstructed signal $\hat{\mathbf{x}}$, support set $\mathcal{I}_k$
Initialization:
    1. $\mathcal{I}_0 = \phi$; $\mathbf{r}_0 = \mathbf{y}$;
    2. $\boldsymbol{\alpha}_0 = \mathbf{A}^T\mathbf{r}_0$; $\mathbf{G} = \mathbf{A}^T\mathbf{A}$;
    3. $i_1 = \arg\max(|\boldsymbol{\alpha}_0|), \mathcal{I}_1 = \mathcal{I}_0 \cup \{i_1\}$;
    4. $\mathbf{G}^{-1}_{\mathcal{I}_1,\mathcal{I}_1} = 1/G_{i_1,i_1}$
    5. $\hat{\mathbf{x}}_{\mathcal{I}_1} = \mathbf{A}^{\dagger}_{\mathcal{I}_1}\mathbf{y}$;
    6. $\boldsymbol{\alpha}_1 = \mathbf{A}^T_{\tilde{\mathcal{I}}_1}\mathbf{y} - \mathbf{G}_{\tilde{\mathcal{I}}_1,\mathcal{I}_1}\hat{\mathbf{x}}_{\mathcal{I}_1}$;
    7. Iteration counter $k = 2$;
repeat
    $i_k = \arg\max(|\boldsymbol{\alpha}_{k-1}|)$
    $\mathcal{I}_k = \mathcal{I}_{k-1} \cup \{i_k\}$
    $\mathbf{W} = \mathbf{G}_{i_k,I_{k-1}}\mathbf{G}^{-1}_{I_{k-1},I_{k-1}}$
    $\mathbf{V} = \left(\mathbf{G}_{i_k,i_k} - \mathbf{W}\mathbf{G}_{\mathcal{I}_{k-1},i_k}\right)^{-1}$
    $\mathbf{U} = \mathbf{W}\mathbf{A}^{\mathrm{T}}_{\mathcal{I}_{k-1}}\mathbf{y} - \mathbf{A}^{\mathrm{T}}_{i}\mathbf{y}$
    $\mathbf{G}^{-1}_{\mathcal{I}_k,\mathcal{I}_k} = \begin{bmatrix} \mathbf{G}^{-1}_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} + \mathbf{V}\mathbf{W}^{\mathrm{T}}\mathbf{W} & -\mathbf{V}\mathbf{W}^{\mathrm{T}} \\ -\mathbf{V}\mathbf{W} & \mathbf{V} \end{bmatrix}$
    $\boldsymbol{\alpha}_k = \boldsymbol{\alpha}_{k-1} - \mathbf{G}_{\tilde{\mathcal{I}}_k,\mathcal{I}_k}\begin{bmatrix} \mathbf{V}\mathbf{U}\mathbf{W}^{\mathrm{T}} \\ -\mathbf{V}\mathbf{U} \end{bmatrix}$
    $\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{W}^{\mathrm{T}}\mathbf{V}\mathbf{U} \\ -\mathbf{V}\mathbf{U} \end{bmatrix}$
until $k \geq K$
```

3. Two-Stage Reconstruction Algorithm

The complexity of the OMP reconstruction algorithm increases along with the increased sensing resolution and dimension of the sensing matrix. In order to reduce high reconstruction complexity for the CMOS impulse radar system, this study proposes a two-stage OMP reconstruction algorithm including block-wise OMP estimation, weight updating and decision mechanism, and fine estimation. Figure 4a shows the processing flow chart of the proposed algorithm.

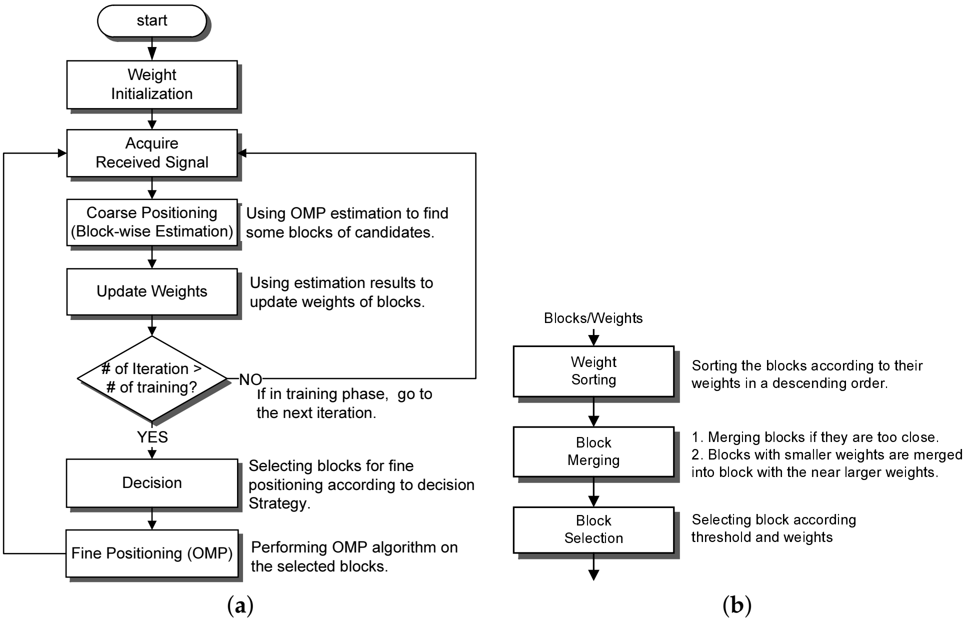

start
Weight Initialization
Acquire Received Signal
Coarse Positioning (Block-wise Estimation)
Using OMP estimation to find some blocks of candidates.
Update Weights
Using estimation results to update weights of blocks.
# of Iteration > # of training?
NO
If in training phase, go to the next iteration.
YES
Decision
Selecting blocks for fine positioning according to decision Strategy.
Fine Positioning (OMP)
Performing OMP algorithm on the selected blocks.
(a)
Blocks/Weights
Weight Sorting
Sorting the blocks according to their weights in a descending order.
Block Merging
1. Merging blocks if they are too close.
2. Blocks with smaller weights are merged into block with the near larger weights.
Block Selection
Selecting block according threshold and weights
(b)

Figure 4. (**a**) Flow chart of the proposed low-complexity two-stage OMP reconstruction algorithm and (**b**) the decision strategy.

3.1. Block-Wise OMP Estimation

The proposed algorithm first performs block-wise OMP, which reduces the sensing matrix size to estimate the rough target regions. In the CS radar, each column of a sensing matrix $\mathbf{A}$ represents the echo signal of a specific grid point. Because the neighbouring grid points have similar echo signals, $\mathbf{A}$ can be separated into several blocks and each block is the concatenation of neighbouring columns. Thus, the target locations can be coarsely estimated by the block-wise OMP estimation. The block-wise OMP algorithm first downsizes $\mathbf{A}$ into a new sensing matrix $\mathbf{D}$ by $\mathbf{D}_j = \mathbf{A}_{(j-1)\times(B+1)} + \cdots + \mathbf{A}_{j\times B}$, where B is the block size. The j-th column of a downsized sensing matrix $\mathbf{D}$ is the sum of columns in the j-th block of $\mathbf{A}$. The block-wise OMP estimation can be regarded as the OMP using $\mathbf{D}$ as the sensing matrix. The detailed block-wise OMP algorithm is shown in Algorithm 3. The block-wise OMP estimation selects several block candidates and stores their indices in a candidate set $\mathcal{I}_{block}$ for the following weight updating.

Algorithm 3 Block-Wise Estimation Algorithm

```
Input: Sensing matrix $\mathbf{A}$, measurement $\mathbf{y}$, target sparsity K, block size B
Output: Support set $\mathcal{I}_k$
Initialization:
    • $k$=1; $\mathcal{I}_0 = \phi$;
    • New sensing matrix $\mathbf{D}$ where $\mathbf{D}_j = \mathbf{A}_{(j-1)\mathrm{B}+1} + \cdots + \mathbf{A}_{j\mathrm{B}}$
    • $\mathbf{r}_0 = \mathbf{y}$; $\boldsymbol{\alpha}_0 = \mathbf{D}^{\mathrm{T}}\mathbf{r}_{k-1}$
repeat
    $i_{\mathrm{block}} = \arg\max(|\boldsymbol{\alpha}_{k-1}|)$
    $\mathcal{I}_k = \mathcal{I}_{k-1} \cup \{i_{\mathrm{block}}\}$;
    $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{D}^{\dagger}_{\mathcal{I}_k}\mathbf{y}$;
    $\mathbf{r}_k = \mathbf{y} - \mathbf{D}_{\mathcal{I}_k}\hat{\mathbf{x}}_{\mathcal{I}_k}$;
    $\boldsymbol{\alpha}_k = \mathbf{D}^{\mathrm{T}}_{\mathcal{I}_k}\mathbf{r}_k$
    $k$=$k$+1;
until ($k \geq K$)
```

3.2. Weight Updating

Each block has a weight representing the possibility of having targets within the block. The proposed algorithm calculates weights by using historical coarse estimation results. The proposed weighting mechanism utilizes the fact that the blocks near a known object have higher opportunities of having objects in, even in a noisy environment. Thus, the proposed algorithm collects coarse estimation results of the previous iteration for evaluating the block weights. The weight updating is expressed by $\mathbf{Z}\left(\mathcal{I}_{block}\right) = \mathbf{Z}\left(\mathcal{I}_{block}\right) + 1$, where is the candidate set selected by the coarse estimation. The size of $\mathcal{I}_{block}$ is determined by the number of block candidates K_b of the compressive sensing system. The algorithm adopts several sensing matrix $\mathbf{A}$ from the UWB radar system and skips the fine estimation in the first a few T_{num} iterations of the training mode. This weight computation helps to increase the reliability of the weight distribution. The algorithm usually needs only one or two training iterations in low-noise conditions.

3.3. Decision Strategy for Fine Estimation

After updating weights, the algorithm selects the blocks to be performed by the following fine estimation according to the decision strategy shown in Figure 4b. The first step performs weight sorting. The blocks are sorted in the descending order of their weights. The second step is block merging. Since the size of human body is much larger than the respiratory spatial variation, the neighbouring selected blocks could represent the same target. Thus, this step merges the block candidates within a merging distance M_g into the block with the largest weight. Then, the merged blocks are discarded and their weights are added to the weight of the winner block. The final step selects the remaining candidate blocks by comparing the normalized weights with a specified threshold T_h. After the threshold decision, the fine estimation combines the columns of the selected blocks as sensing matrix to perform the fine object positioning. Notice that the block merging in the decision process assumes that the size of human body is much larger than the respiratory spatial variation because the radar ranging resolution is much higher than the body resolution. Thus, sparsity K in the fine estimation OMP is no larger than K_b to locate the detailed positions of the human bodies or objects. Following simulation and hardware design assume $K = K_b$ without the loss of generality.

After the location of targets are estimated by the OMP estimation, the algorithm returns back to the next iteration. Notice that the block weights are not reset and remain the same at the beginning of the next iteration. The detailed two-stage reconstruction algorithm is shown in Algorithm 4.

Algorithm 4 Proposed Two-Stage Reconstruction Algorithm

```
Input: sensing matrix $\mathbf{A}$; received signal $\mathbf{y}$; number of targets K; block size B; training number $\mathrm{T}_{\mathrm{num}}$; threshold $\mathrm{T}_{\mathrm{h}}$;
    merging distance $\mathrm{M}_{\mathrm{g}}$; number of block candidates $\mathrm{K}_{\mathrm{b}}$;
Output: Support set $\mathcal{I}_{out}$
1:  Initialization: Let iteration counter $k = 1$, and initialize weights $\mathbf{Z}\left(1 : \frac{\mathrm{N}}{\mathrm{B}}\right) = 0$;
2:  Acquire received signal $\mathbf{y}_k$ of $k$-th iteration;
3:  Coarse positioning(Block-Wise Estimation Algorithm):
      (A) Calculate downsized sensing matrix $\mathbf{D}$ where $\mathbf{D}_j = \mathbf{A}_{(j-1)\mathrm{B}+1} + \cdots + \mathbf{A}_{j\mathrm{B}}$
      (B) $\mathcal{I}_{block} = OMP\left(\mathbf{D}, \mathbf{y}_k, \mathrm{K}_{\mathrm{b}}\right)$;
4:  Update weights: $\mathbf{Z}\left(\mathcal{I}_{block}\right) = \mathbf{Z}\left(\mathcal{I}_{block}\right) + 1$;
5:  if $k \leq \mathrm{T}_{num}$ then
6:      Let $k = k + 1$ and goto Step 2 for next iteration;
7:  end if
8:  Decision:
      (A) Sorting: $[\mathbf{Z}_{sort}, \mathcal{I}_{sort}] = Sort\left(\mathbf{Z}\left(\mathcal{I}_{block}\right), \mathcal{I}_{block}\right)$
      (B) Merging:
        1. for all $i$ and $j$ such that $|\mathcal{I}_{sort}(i) - \mathcal{I}_{sort}(j)| < \mathrm{M}_g$ and $i \neq j$ do
        2.     if $\mathbf{Z}_{sort}(i) > \mathbf{Z}_{sort}(j)$ then
        3.         $\mathbf{Z}_{sort}(i) = \mathbf{Z}_{sort}(i) + \mathbf{Z}_{sort}(j)$
        4.       and remove $\mathcal{I}_{sort}(j)$ from $\mathcal{I}_{sort}$
        5.     else
        6.         $\mathbf{Z}_{sort}(j) = \mathbf{Z}_{sort}(j) + \mathbf{Z}_{sort}(i)$
        7.       and remove $\mathcal{I}_{sort}(i)$ from $\mathcal{I}_{sort}$
        8.     end if
        9. end for
      (C) Selection:
            1. Threshold calculation: $t = \mathrm{T}_h \times sum(\mathbf{Z})$
            2. $\mathcal{I}_{select} = \mathcal{I}_{sort}(1 : \mathrm{K})$
            3. Removing $\mathcal{I}_{select}(i)$ from $\mathcal{I}_{select}$ for every $i$ that satisfies $\mathbf{Z}_{sort}(i) < t$
9:      Fine positioning: $\mathcal{I}_{out} = OMP\left(\mathbf{A}_{\mathcal{I}_{select}}, \mathbf{y}_k, \mathrm{K}\right)$;
10:     Output $\mathcal{I}_{out}$, and let $k = k + 1$;
11:     Goto Step 2 in order to acquire new received signal;
```

4. Complexity and Performance Analysis

This section analyses the computational complexity of the conventional OMP algorithm, OMP via MIB algorithm, and the proposed two-stage OMP algorithms with and without MIB. The complexity analysis is based on the number of multiplications of several matrix operations. The complexity of the matrix inversion for an $n \times n$ matrix is $\mathcal{O}(n^3)$. The complexity of the matrix multiplication $\mathbf{A} \times \mathbf{B}$ is $\mathcal{O}(mnk)$, where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $n \times k$ matrix. The complexity of the matrix pseudo-inverse $\mathbf{A}^{\dagger}$ is $\mathcal{O}(n^3 + 2mn^2)$.

4.1. Orthogonal Matching Pursuit Algorithm

Algorithm 1 shows the OMP algorithm. The computational complexity of the pseudo matrix inversion and multiplication in Step 5 is $\mathcal{O}(k^3 + 2mk^2 + mk)$. The complexity of Step 6 and Step 7 are $\mathcal{O}(mk)$ and $\mathcal{O}((n-k)m)$. Thus, total complexity of the OMP algorithm is $\mathcal{O}(k^3 + 2mk^2 + nm + mk)$. For the fair comparison with the OMP-MIB algorithm, the computation of matrix pseudo-inverse $\mathbf{A}^{\dagger}_{\mathcal{I}_k} = (\mathbf{A}^T_{\mathcal{I}_k}\mathbf{A}_{\mathcal{I}_k})^{-1}\mathbf{A}^T_{\mathcal{I}_k}$ can be reduced to $\mathbf{A}^{\dagger}_{\mathcal{I}_k} = (\mathbf{G}_{\mathcal{I}_k,\mathcal{I}_k})^{-1}\mathbf{A}^T_{\mathcal{I}_k}$. Therefore, computational complexity of matrix pseudo-inverse $\mathbf{A}^{\dagger}_{\mathcal{I}_k}$ is $\mathcal{O}(k^3 + mk^2)$. Thus, total complexity of the OMP algorithm is $\mathcal{O}(k^3 + mk^2 + nm + mk)$.

4.2. OMP-MIB Algorithm

Algorithm 2 shows the OMP-MIB algorithm. The complexity of calculating $\mathbf{W}, \mathbf{V}$, and $\mathbf{U}$ are $\mathcal{O}((k-1)^2)$, $\mathcal{O}(k)$, and $\mathcal{O}(mk + k - 1)$, respectively. Steps 8, 9, and 10 calculate $\mathbf{G}^{-1}_{\mathcal{I}_k,\mathcal{I}_k}$, $\boldsymbol{\alpha}_k$, and $\hat{\mathbf{x}}_k$, respectively. The computational complexity of calculating $\mathbf{V}\mathbf{W}^{\mathrm{T}}\mathbf{W}$ in Step 8 is $\mathcal{O}((k-1)^2 + (k-1))$. In Step 9, the complexity of $\begin{bmatrix}\mathbf{VUW}^{\mathrm{T}} & -\mathbf{VU}\end{bmatrix}^{\mathrm{T}}$ is $\mathcal{O}(k)$. Thus, computational complexity of calculating $\mathbf{G}_{\tilde{\mathcal{I}}_k,\mathcal{I}_k}\begin{bmatrix}\mathbf{VUW}^{\mathrm{T}} \\ -\mathbf{VU}\end{bmatrix}$ is $\mathcal{O}(nk - k^2)$. Thus, the total complexity of the OMP via MIB algorithm is $\mathcal{O}(k^2 + nk + mk - 1) \approx \mathcal{O}(k^2 + nk + mk)$.

4.3. Proposed Two-Stage OMP Reconstruction Algorithm

The major complexity of the proposed two-stage reconstruction algorithm are the OMP calculations for both coarse and fine positioning. The block-wise estimation is an OMP algorithm for a downsized $m \times n_c$ sensing matrix $\mathbf{D}$. Thus, the complexity of the coarse positioning is $\mathcal{O}\left(k^3 + mk^2 + n_c m + mk\right)$. For block size b, n_c can be expressed as $n_c = \frac{n}{b}$. Thus, the complexity of the coarse positioning becomes $\mathcal{O}\left(k^3 + mk^2 + \frac{n}{b}m + mk\right)$. The fine positioning uses a $m \times n_f$ sensing matrix to perform the OMP algorithm for the selected blocks. Thus, the complexity is $\mathcal{O}\left(k^3 + mk^2 + n_f m + mk\right)$, and n_f can be expressed as $n_f = kb$. Hence, the complexity of the fine positioning becomes $\mathcal{O}\left(k^3 + mk^2 + kbm + mk\right)$. The total complexity of the proposed two-stage reconstruction algorithm is $\mathcal{O}\left(k^3 + mk^2 + \frac{n}{b}m + mk\right) + \mathcal{O}\left(k^3 + mk^2 + kbm + mk\right)$.

In addition, both the coarse and fine positioning can be realized by using OMP-MIB algorithm. Thus, the complexity of coarse and fine positioning are $\mathcal{O}\left(k^2 + n_c k + mk\right)$ and $\mathcal{O}\left(k^2 + n_f k + mk\right)$, respectively. By substituting $n_c = \frac{n}{b}$ and $n_f = kb$ into $\mathcal{O}\left(k^2 + n_c k + mk\right)$ and $\mathcal{O}\left(k^2 + n_f k + mk\right)$, respectively, Thus, the complexity of the coarse and fine positioning by using OMP-MIB algorithm are $\mathcal{O}\left(k^2 + \frac{n}{b}k + mk\right)$ and $\mathcal{O}\left((b+1)\,k^2 + mk\right)$.

Figure 5a shows the complexity analysis results with the sparsity factor $k = 3$ and block size $b = 4$. The complexity of two-stage OMP algorithm is much lower than the OMP algorithm. The OMP via MIB algorithm can further reduce the complexity of the OMP and two-stage OMP algorithms, as shown in Figure 5b. In this case, the proposed two-stage algorithm still has lower complexity.

Figure 6 shows the complexity of various block sizes. Figure 6b shows the complexity of the OMP-MIB and the proposed two-stage OMP-MIB algorithm since they are too close in Figure 6a. The complexity of the proposed algorithm decreases along with the increased block size because of the decreased dimension of the sensing matrix of the block-wise estimation. The complexity of the proposed two-stage OMP reconstruction with block size $b = 4$ reduces approximately 75% complexity of the conventional OMP algorithm. Figure 6b shows that proposed two-stage OMP-MIB algorithm with block size $b = 4$ reduces approximately 50% complexity comparing to original OMP via MIB algorithm. The complexity reduction percentage would further increases when the radar resolution increases, that is, the dimension of the original sensing matrix $\mathbf{A}$ increases. This is a great benefit to human detection because the impulse radar usually requires very high radio frequency and very high spatial resolution in order to acquire the tiny movement of the human respiration.

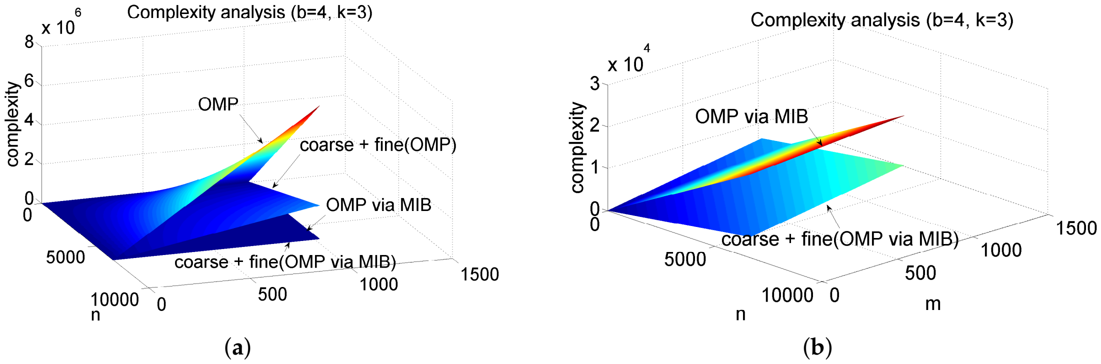


Figure 5. Complexity analysis results of (**a**) the orthogonal matching pursuit (OMP) and two-stage OMP algorithms w/o OMP-Matrix Inversion Bypass (MIB) scheme and (**b**) the OMP and two-stage OMP algorithms with OMP-MIB scheme.

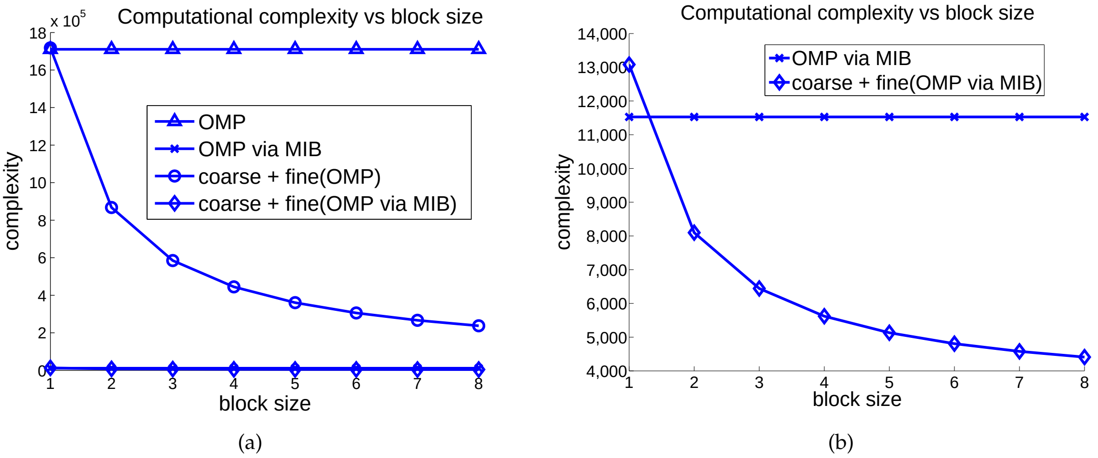


Figure 6. Computational complexity versus block size where $m = 512$, and $n = 3328$.

4.4. Simulation Result

Figure 7 shows the positioning results of the OMP and two-stage OMP algorithms. 500 impulses are used to simulate the CS radar processing under different SNR conditions. The sensing matrix **A** is derived from the MIMO compressive sensing model in Sections 2.2–2.4. The "estimate" points are the detected targets, which are sent to the respiration feature extraction algorithm [10] to identify the human and object. When SNR is 24 dB (Figure 7a,d), both the OMP and two-stage OMP algorithm can perfectly locate the object and humans. When SNR is 6 dB (Figure 7b,e), the OMP algorithm generates a few incorrect estimation results of the object and human locations, but the proposed algorithm still perfectly locates the object and humans. When SNR is −6 dB (Figure 7c,f). However, the proposed algorithm can still detect one human target with some incorrect results and detect the non-human target perfectly. The proposed two-stage reconstruction algorithm has better positioning performance than the conventional OMP algorithm especially in noisy environments.

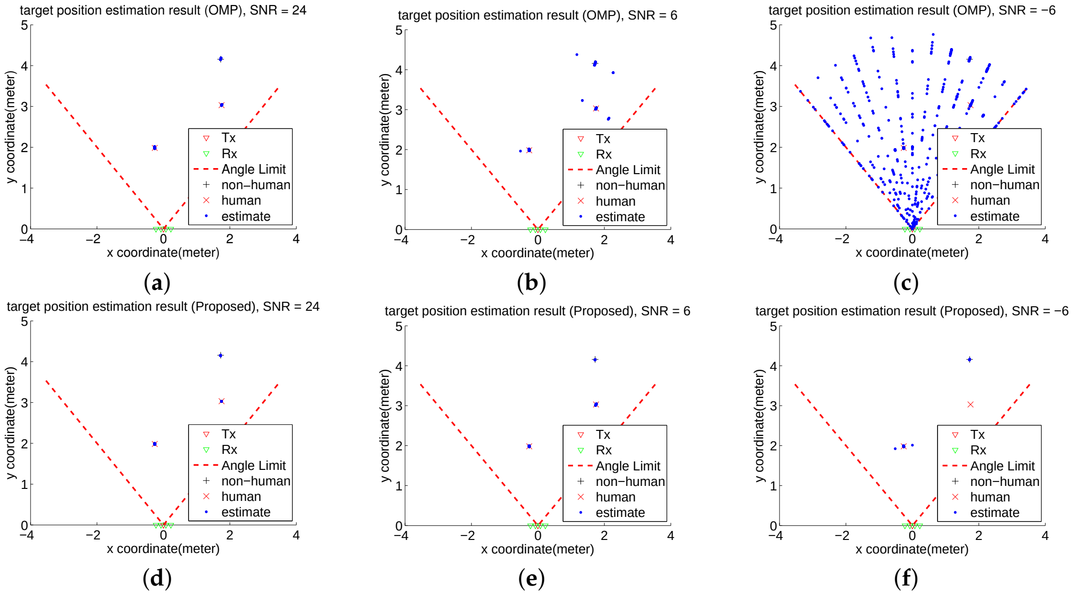


Figure 7. The compressive sensing (CS) radar scenes using the OMP reconstruction in (**a**) 24 dB, (**b**) 6 dB, and (**c**) −6 dB SNR environments. The CS radar scenes using the two-stage OMP reconstruction in (**d**) 24 dB, (**e**) 6 dB, and (**f**) −6 dB SNR environments.

4.5. Performance Analysis

This study uses three metrics to perform detailed analysis of the reconstruction and detection results including normalized mean square error (NMSE), number of hits, and hit ratio. These metrics show some properties that are not revealed in Figure 7.

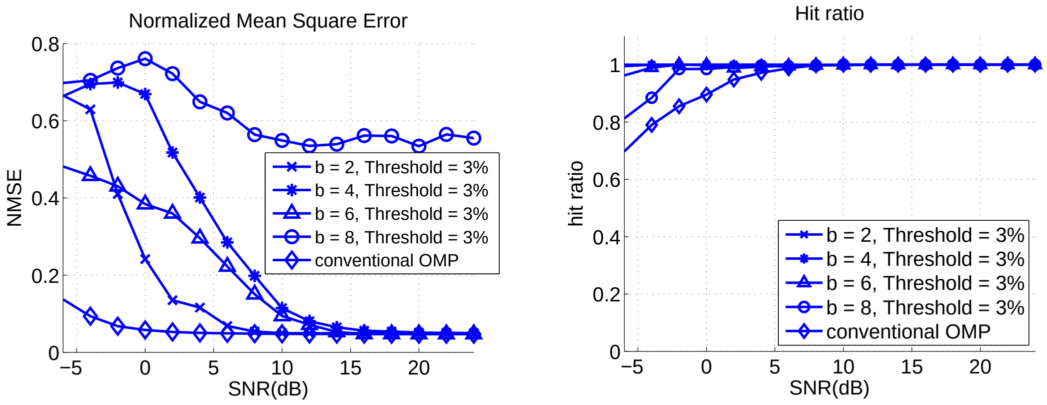
The normalized mean squared error of the reconstructed signals by CS radar is defined by

$$\textbf{NMSE} = \frac{\|\mathbf{y}_{estimate} - \mathbf{y}_{true}\|_2^2}{\|\mathbf{y}_{true}\|_2^2}, \tag{17}$$

where $\mathbf{y}_{true}$ is the received radar signal and $\mathbf{y}_{estimate}$ is the reconstructed signal by the OMP or the proposed algorithm. Number of hits is the number of correct estimates in the simulation. The hit ratio is the detection probability defined by

$$\textbf{Hit Ratio} = \frac{\text{Number of Correct Estimates}}{\text{Number of Total Estimates}}. \tag{18}$$

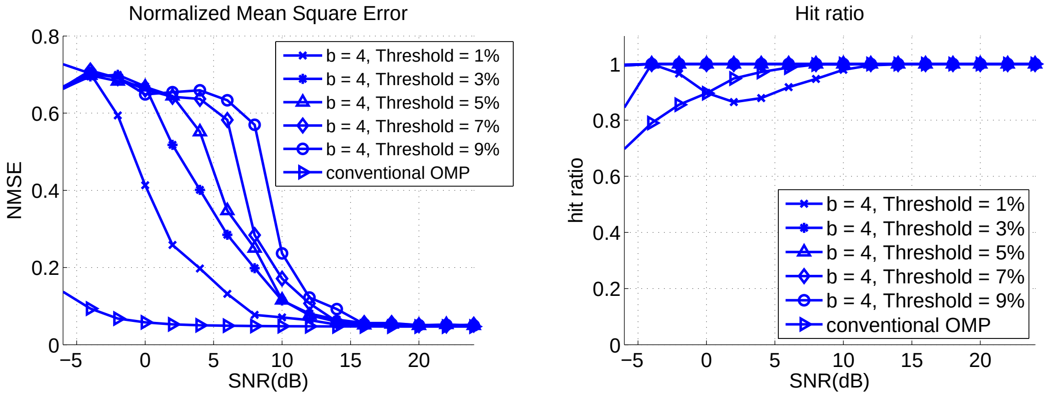
Figure 8a shows the NMSE verus SNR performances of the OMP and two-stage OMP algorithms with different block sizes. The proposed two-stage OMP algorithm has worse NMSE performance because the proposed algorithm utilizes threshold mechanism to discard bad estimates and distort the reconstruction signal. However, the NMSE does not necessarily reflect the detection and positioning performance, as shown in Figure 7, because a lower NMSE value only means that the signal is better reconstructed by the OMP process but the reconstructed signal has an incorrect combination of indices, that is, the support set $\mathcal{I}_K$. The proposed algorithm with $b = 6$ has better NMSE than that with $b = 4$ because the threshold mechanism with $b = 4$ is more strict than that with $b = 6$ because the block weights is not normalized as we mentioned in Section 3. Hence, the proposed algorithm with $b = 6$ makes estimated results more easily to reach the threshold, leading to lower NMSE. This is similar to the reason that causes the better RMSE of the OMP algorithm. Figure 8b shows that the hit ratio of the two-stage OMP is better than that of the OMP. This implies that the matching pursuit quality of the two-stage OMP algorithm is better than that of the OMP algorithm.



(**a**) (**b**)

Figure 8. Analysis results of (**a**) NMSE and (**b**) hit ratio for different block sizes.

Figure 9a shows the NMSE versus SNR with different thresholds. The OMP has lower NMSE than the proposed algorithm because of the threshold mechanism. A higher threshold of the proposed algorithm causes a higher NMSE value because more matched results are discarded. Figure 9b shows the hit ratio of the algorithms for different thresholds. A lower threshold leads to a larger number of hits. Threshold $\mathrm{T}_h = 1\%$ has the highest hit number in the analysis of the proposed algorithm, but the hit ratio is unstable when SNR lower than 10 dB.



(**a**) (**b**)

Figure 9. Analysis results of (**a**) normalized mean square error (NMSE) and (**b**) hit ratio for different thresholds.

5. Architecture Design and Implementation

This section introduces the hardware design and implementation of the proposed two-stage reconstruction algorithm. The OMP processing for coarse and fine positioning are the most complex and time consuming steps. Thus, an OMP processor was designed to accelerate the reconstruction processing.

5.1. Architecture of OMP via MIB Processor

Figure 10 shows the overall architecture, which can be divided into two parts. One is Matching Result Update and Index Selection Unit, and the other is Parameter Update Unit. The gray blocks are data memories including ROMs for G and sensing matrix, a single-port SRAM for $\boldsymbol{\alpha}_{init}$, and a dual-port SRAM for $\boldsymbol{\alpha}$. G_inv, GIK_IK, $\boldsymbol{\alpha}_{init}\mathcal{I}_k$, and support set $\mathcal{I}_k$ are register file. GIK_IK and $\boldsymbol{\alpha}_{init}\mathcal{I}_k$ are buffers for parallel data streams. $\mathcal{I}_k$ is the storage of support set. Initially, $\boldsymbol{\alpha}$ serves as register and loads the received signal firstly for calculating matching result, and then initialization steps are executed by matching result update circuit and index selection circuit. After initialization, updating GIK_IK and $\boldsymbol{\alpha}_{init}\mathcal{I}_k$ that store specific value from G ROM and $\boldsymbol{\alpha}_{init}$ RAM for computing parameter $\mathbf{W}$, $\mathbf{V}$, $\mathbf{U}$, and $\mathbf{W}$, $\mathbf{V}$, $\mathbf{U}$ are calculated by parameter update unit sequentially. Then $\mathbf{G}^{-1}_{\mathcal{I}_k,\mathcal{I}_k}$ and $\boldsymbol{\alpha}_k$ are updated. Support set is also updated during the process of updating $\boldsymbol{\alpha}_k$. Then the algorithm goes to next iteration.

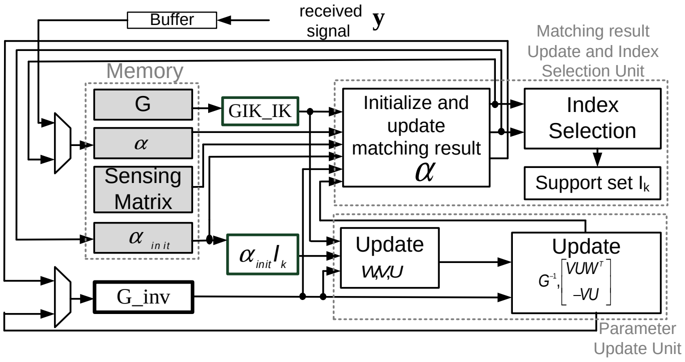


Figure 10. Block diagram of the proposed architecture.

5.2. *Matching Result Update and Index Selection Unit*

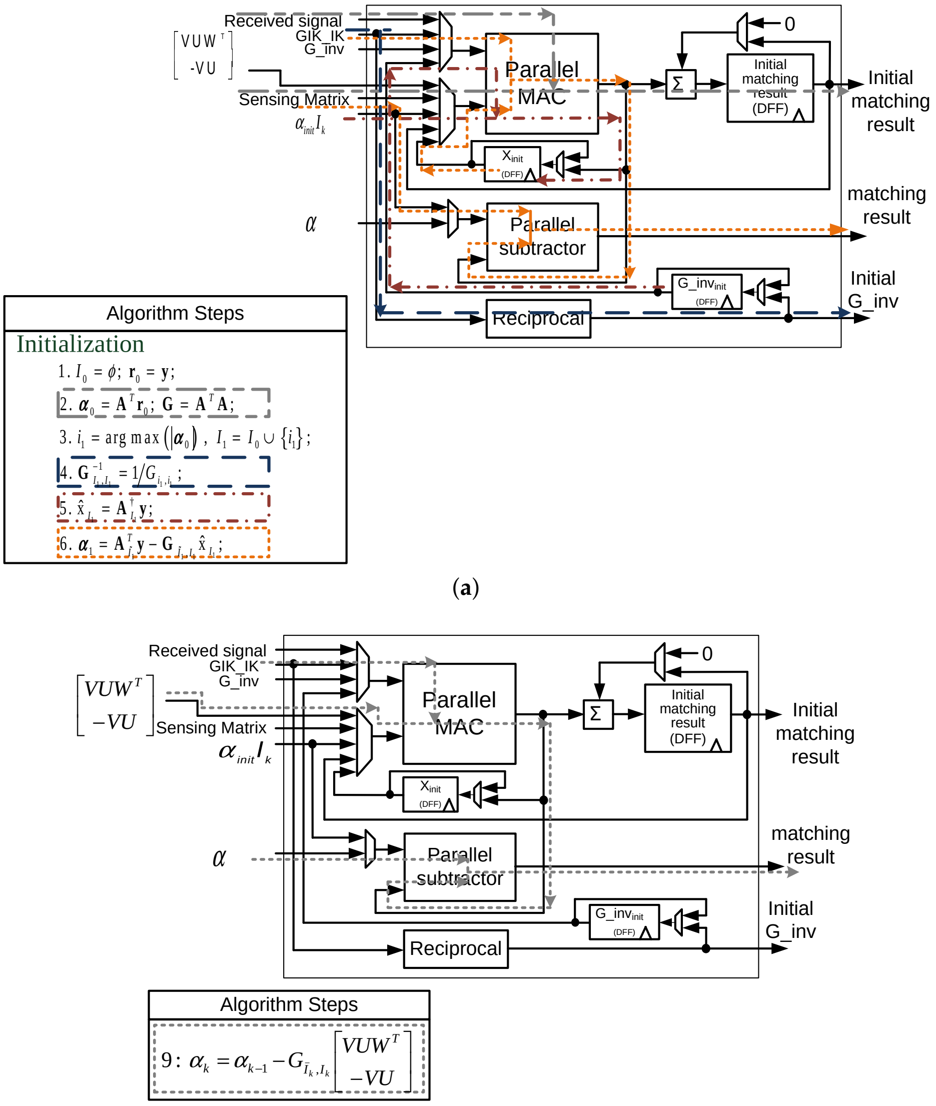
5.2.1. Initialization and Matching Result Update Circuit

Figure 11 shows the matching result update circuit. The Parallel MAC consists of $\mathrm{N}_{\mathrm{mul}}$ multiply-and-accumulate circuits (MACs) that determine the speed of matrix multiplication. The number of subtractor circuits in the Parallel Subtractor is also $\mathrm{N}_{\mathrm{mul}}$.

The matching result update circuit not only calculates the matching result but also performs the initialization of the OMP-MIB algorithm. In initialization phase, parallel MAC calculates the initial matching result first. Because of the extremely high dimension of sensing matrix and received signals, it is impossible to execute the parallel MAC for vector-vector multiplication of each column of sensing matrix in a clock cycle. Thus, the partial results are accumulated by summation circuit and registers. For an M × N sensing matrix, the matching result update circuit produces one element of $\boldsymbol{\alpha}_0$ in every $\frac{\mathrm{M}}{\mathrm{N}_{\mathrm{mul}}}$ clock cycles. Since the elements of initial matching result vector $\boldsymbol{\alpha}_0$ are computed sequentially, index selection circuit compares the maximum value of initial matching result vector one by one, therefore, reducing the complexity of index selection circuit. The resultant $\boldsymbol{\alpha}_0$ is then stored in memory $\boldsymbol{\alpha}_{init}$. The calculations of initial matching result can be finished in $\mathrm{N}\frac{\mathrm{M}}{\mathrm{N}_{\mathrm{mul}}}$ clock cycles.

In the initialization phase, index i_1 is computed by index selection circuit, and $\mathbf{G}^{-1}_{\mathcal{I}_1,\mathcal{I}_1}$ is computed by reciprocal circuit. After $\mathbf{G}^{-1}_{\mathcal{I}_1,\mathcal{I}_1}$ is calculated, $\hat{\mathbf{x}}_{\mathcal{I}_1}$ is produced by parallel MAC and stored in $\mathbf{x}_{init}$, which will be used for the calculation of new matching result $\boldsymbol{\alpha}_1$ by parallel MAC and parallel subtractor circuits. Figure 11a shows the processing flow of the initialization in the data path circuit.

The matching result update circuit updates $\boldsymbol{\alpha}_k$ based on $\boldsymbol{\alpha}_{k-1}$ and matrices $\mathbf{W}$, $\mathbf{V}$, $\mathbf{U}$. Dual-port SRAM is used for simultaneous reading and writing of $\boldsymbol{\alpha}$s. $\boldsymbol{\alpha}_k$ is obtained by subtracting $\boldsymbol{\alpha}_{k-1}$ by the parallel MAC output of the matrix multiplication $\mathbf{G}_{\tilde{\mathcal{I}}_k,\mathcal{I}_k} \times \begin{bmatrix} \mathbf{VUW}^{\mathrm{T}} \\ -\mathbf{VU} \end{bmatrix}$. Parallel MAC calculates $\mathrm{N}_{\mathrm{mul}}$ elements of matrix multiplication operation in every k clock cycles, where k is the iteration counter and $\mathrm{N}_{\mathrm{mul}}$ is the number of MAC circuits. The $\mathrm{N}_{\mathrm{mul}}$ partial results from parallel MAC subtract the matching result vector $\boldsymbol{\alpha}_{k-1}$ concurrently by parallel subtractor circuits to obtain $\mathrm{N}_{\mathrm{mul}}$ elements of $\boldsymbol{\alpha}_k$. There are $\mathrm{N} - k$ elements in the vector $\boldsymbol{\alpha}_k$ for a M × N sensing matrix and $\mathrm{N} - k >> \mathrm{N}_{\mathrm{mul}}$. Therefore, the matching result updating is the most time consuming step in the algorithm. Figure 11b shows the processing flow of matching result updating process.


Received signal
GIK_IK
G_inv
$\begin{bmatrix} VUW^T \\ -VU \end{bmatrix}$
Sensing Matrix
$\alpha_{init} I_k$
α
Parallel MAC
Σ
0
Initial matching result (DFF)
Initial matching result
X_{init} (DFF)
matching result
Parallel subtractor
G_inv_{init} (DFF)
Initial G_inv
Reciprocal
Algorithm Steps
Initialization
1. $I_0 = \phi$; $\mathbf{r}_0 = \mathbf{y}$;
2. $\boldsymbol{\alpha}_0 = \mathbf{A}^T\mathbf{r}_0$; $\mathbf{G} = \mathbf{A}^T\mathbf{A}$;
3. $i_1 = \arg\max(|\boldsymbol{\alpha}_0|)$, $I_1 = I_0 \cup \{i_1\}$;
4. $\mathbf{G}^{-1}_{I_1,I_1} = 1/G_{i_1,i_1}$;
5. $\hat{\mathbf{x}}_{I_1} = \mathbf{A}^{\dagger}_{I_1}\mathbf{y}$;
6. $\boldsymbol{\alpha}_1 = \mathbf{A}^T_{\bar{I}_1}\mathbf{y} - \mathbf{G}_{\bar{I}_1,I_1}\hat{\mathbf{x}}_{I_1}$;
9: $\alpha_k = \alpha_{k-1} - G_{\bar{I}_k,I_k}\begin{bmatrix} VUW^T \\ -VU \end{bmatrix}$

(**a**)

(**b**)

Figure 11. Processing flows of matching result update circuit for (**a**) initialization and (**b**) common matching result update in the data path circuits.

5.2.2. Index Selection Circuit

Figure 12 shows the index selection circuit. At initialization stage, the matching result update circuit calculates the elements of initial matching result vector $\boldsymbol{\alpha}_0$ sequentially, and then index selection circuit compares and selects the index of the elements with maximum matching result. At the iteration stage, the matching result update circuit outputs $\mathrm{N_{mul}}$ matching results concurrently. Therefore, index selection circuit compares these $\mathrm{N_{mul}}$ matching results by using maximum circuit. Maximum matching result of the maximum circuit is further compared with the previous maximum matching result. The larger result is preserved for the next comparison. Figure 12 shows processing flows of index selection circuit for the initialization and common matching result update processes.

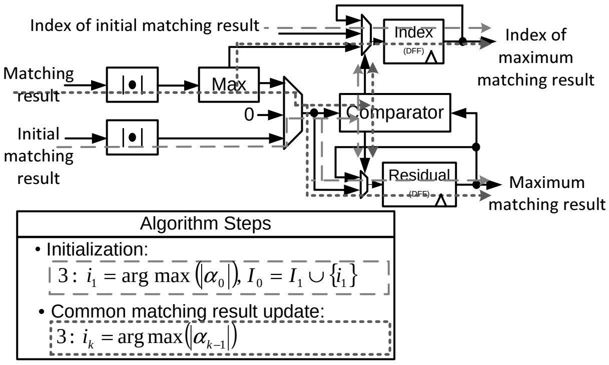
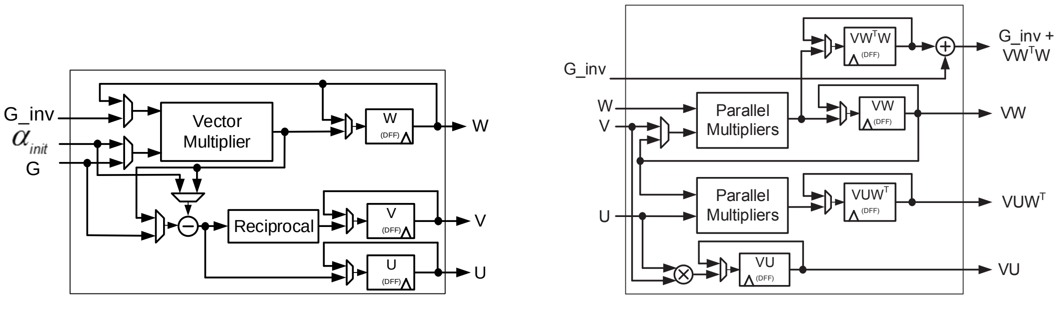

Index of initial matching result
Index (DFF)
Index of maximum matching result
Matching result
|•|
Max
0
Comparator
Initial matching result
|•|
Residual (DFF)
Maximum matching result
Algorithm Steps
• Initialization:
$3: i_1 = \arg\max(|\alpha_0|), I_0 = I_1 \cup \{i_1\}$
• Common matching result update:
$3: i_k = \arg\max(|\alpha_{k-1}|)$

Figure 12. Data path of index selection circuit.

5.3. Parameter Update Unit

Figure 13 shows the architecture of parameter update unit, which consists two circuits to update the matrices **W**, **V**, **U**. The vector multiplier and parallel multiplier both use $K - 1$ multipliers to execute vector multiplication and parallel multiplication. Figure 13a shows the parameter update circuit for **W**, **V**, and **U**. **W** is first calculated in $k - 1$ clock cycles in the k-th iteration. Then, **V** and **U** are computed in one clock cycle. After **W**, **V** and **U** are updated, the circuit shown in Figure 13b computes the matrix multiplications of **VW**, $\mathbf{VUW}^{\mathrm{T}}$, and $\mathbf{VW}^{\mathrm{T}}\mathbf{W}$, which are used for updating new matching result $\boldsymbol{\alpha}_k$ and $\mathbf{G}^{-1}_{\tilde{\mathcal{I}}_k,\mathcal{I}_k}$ stored in memory G_inv.

Figure 13b shows the several matrix multiplication circuits. First, **VW** is calculated in one clock cycle. Then, **VU**, $\mathbf{VUW}^{\mathrm{T}}$, and $\mathbf{VW}^{\mathrm{T}}\mathbf{W}$ are calculated concurrently. **VU** and $\mathbf{VUW}^{\mathrm{T}}$ require one clock cycle, and $\mathbf{VW}^{\mathrm{T}}\mathbf{W}$ requires $k - 1$ clock cycles. After these matrices are updated, new matching result $\boldsymbol{\alpha}_k$ and index i_k can be determined by the matching result update circuit and index selection circuit.



(a) (b)

Figure 13. Architecture of parameter update unit where (**a**) updates **W**, **V**, **U** and (**b**) updates G_inv, $\mathbf{VUW}^{\mathrm{T}}$, **VU**.

In the block diagrams of the afore-mentioned circuits, some processing modules, such as parallel MACs, parallel substractors, parallel multipliers, and vector multiplier, can be referred to our previous works [30,31].

5.4. Implementation Results and Comparison

The proposed two-stage OMP processor was designed and implemented by using Software and Xilinx Virtex-7 FPGA (Xilinx, San Jose, CA, USA). The proposed two-stage reconstruction algorithm was realized by Matlab software (MathWorks, Natick, MA, USA) except that the pure OMP processing functions (Line 3.(B) and 9 in Algorithm 4) are accelerated by FPGA hardware. The original CS radar system has a 512 × 3328 sensing matrix derived from the MIMO compressive sensing model in Section III B, C, and D, with $b = 4$, $k = 8$, and SNR = −20 dB, that is, the coarse estimation performs OMP reconstruction for a 512 × 832 sensing matrix, and the fine estimation performs OMP reconstruction for a 512 × 32 sensing matrix. The OMP processor was implemented for both 512 × 832 and 512 × 32 sensing matrices. In the fixed-point simulation, the number of correctly selected indices in $\boldsymbol{I}_k$ are used to determine the number of fractional bits with 13 integer bits of the signals in memory.

Table 1 shows the hardware resources utilized in the proposed processor. The processing time for reconstruction of a 512 × 3328 CS matrix is 35 ms. The corresponding radar image frame rate is about 28.2 frames per second, which approaches the typical video frame rate. Table 2 compares the proposed OMP processor with other reconstruction processors in the literature. The major differences of the proposed processor are the large compressing sensing matrix size and targeted application. The CS radar system has an extremely large CS matrix dimension to provide very high radar image resolution for the detection of very fine human respiration signal. Thus, it is difficult and almost impossible to implement such a high-dimension signal reconstruction processor for the CS radar system. The proposed two-stage reconstruction algorithm successfully reduces the complexity by reducing the CS matrix dimension in the coarse positioning and still maintains the high-precision reconstruction by fine-positioning. Although traditional reconstruction processors have small processing latencies, they only support much lower dimensions than the proposed two-stage OMP processor. If these traditional algorithms are applied to the compressive sensing radar system with such a high dimension, it is impossible to realize a real-time processor for the CS radar system. In the application aspects, the proposed processor supports 28.2 frames per second for the radar image real-time display. This image refreshing throughput is good enough for the human vision response.

Table 1. FPGA Implementation Results.

Plate-Form	**Xlinx FPGA + Software**
Model	**Vertex-7**
Slice Registers	6535
Slice LUTs	207,967
Block RAMs	1092
DSPs	560
Clock	318 MHz
Latency	35.4 ms
Radar Image Resolution	256 × 13
Radar Image Rate	28.2 frames/s

Table 2. VLSI implementations of OMP processors.

	Proposed	[19]	[21]	[29]	[32]	[30]	[33]
Algorithm	Two-Stage OMB-MIB	OMP	OMP	OMP-MIB	OMP	PIS-MIB-SOMP	SGP
Technology	Virtex-7	Virtex-5	65 nm	65 nm	Vertex-6	90 nm	90 nm
(N,M)	(3328,512)	(128,32)	(256,64)	(256,150)	(1024,256)	(1024,256)	(256,64)
Sparsity	8	5	8	variable	36	12	8
Clock (MHz)	318	39	165	500	120	141	150
Latency Time (µs)	35,400	24	13.7	NONE	340	72.2	61.97
Function	Radar Object Detection	Signal Reconstruct	Signal Reconstruct	Signal Reconstruct	Signal Reconstruct	Signal Reconstruct	Signal Reconstruct

6. Conclusions

This study proposes a two-stage reconstruction algorithm for the compressive sensing radar. The proposed algorithm has better positioning performance and lower complexity than the traditional OMP algorithm. The OMP can be replaced by any reduced-complexity algorithm, such as OMP via MIB, but the proposed two-stage reconstruction is still effective in reducing cost and improving performance. This work applied a practically-measured path-loss model and a human respiratory signal model from a CMOS impulse radar system to configure a 4 × 1 SIMO radar system to analyse the proposed algorithm. The proposed algorithm utilizes block-wise OMP estimation, weight threshold mechanism, and fine estimation to reduce the complexity and improve the performance. The simulation results show that the proposed algorithm needs only 25% computational complexity of the conventional OMP algorithm and has a better positioning performance than the OMP algorithm especially in noisy environments.

Acknowledgments: This work was supported by Ministry of Science and Technology, Taiwan (MOST 103-2220-E-007-005).

Author Contributions: Ta-Shun Chu and Yuan-Hao Huang conceived and designed the experiments; Kuei-Chi Tsao and Ling-Lee performed the experiments and analyzed the data; Kuei-Chi Tsao and Yuan-Hao Huang wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lai, C.-M.; Wu, J.-M.; Huang, P.-C.; Chu, T.-S. A scalable direct-sampling broadband radar receiver supporting simultaneous digital multibeam array in 65 nm cmos. In Proceedings of the IEEE ISSCC Digest, San Francisco, CA, USA, 17–21 February 2013; pp. 242–243.
2. Andersen, N.; Granhaug, K.; Michaelsen, J.A.; Bagga, S.; Hjortland, H.A.; Knutsen, M.R.; Lande, T.S.; Wisland, D.T. A 118-mW Pulse-Based Radar SoC in 55-nm CMOS for Non-Contact Human Vital Signs Detection. *IEEE J. Solid-State Circuits* **2017**, *52*, 3421–3433.
3. Hung, W.-P.; Chang, C.-H.; Lee, T.-H. Real-Time and Noncontact Impulse Radio Radar System for um Movement Accuracy and Vital-Sign Monitoring Applications. *IEEE Sens. J.* **2017**, *17*, 2349–2358.
4. Lazaro, A.; Girbau, D.; Villarino, R.; Ramos, A. Vital signs monitoring using impulse based UWB signal. In Proceedings of the 41th European Mircowave Conference, Manchester, UK, 10–13 October 2011; pp. 135–138.
5. Nahar, S.; Tran, N.; Ren, L.; Kilic, O.; Fathy, A.E. Through-wall Detection of Human Breathing Rate Using Compressive Sensing Technique. In Proceedings of the IEEE Radio and Wireless Symposium, Phoenix, AZ, USA, 15–18 January 2017; pp. 104–107.
6. Zhong, Y.; Yang, Y.; Zhu, X.; Dutkiewicz, E.; Zhou, Z.; Jiang, T. Device-Free Sensing for Personnel Detection in a Foliage Environment. *IEEE Geosci. Remote Sens. Lett.* **2016**, *14*, 921–925.
7. Zhou, Y.-S.; Kong, L.; Cui, G.-L.; Yang, J.-Y. Remote sensing of human body by stepped-frequency continuous-wave. In Proceedings of the 3rd International Conference Bioinformatics Biomedical Engineering, Beijing, China, 11–13 June 2009; pp. 1–4.
8. Leib, M.; Menzel, W.; Schleicher, B.; Hermann, S. Vital signs monitoring with a UWB radar based on a correlation receiver. In Proceedings of the 4th European Conference on Antennas Propagation, Barcelona, Spain, 12–16 April 2010; pp. 1–5.
9. Amir, M.; Khausru, H.; Ghavami, M. Study of Human Being Detection in an Indoor Environment Using Ultra Wideband Radar. In Proceedings of the IEEE International Conference on Ultra-WideBand (ICUWB), Paris, France, 1–3 Septemer 2014; pp. 113–118.
10. Hsieh, C.-H.; Chiu, Y.-F.; Shen, Y.-H.; Chu, T.-S.; Huang, Y.-H. A UWB radar signal processing platform for real-time human respiratory feature extraction based on four-segment linear waveform model. *IEEE Trans. Biomed. Circuits Syst.* **2016**, *10*, 219–230.

11. Baraniuk, R.; Steeghs, P. Compressive radar imaging. In Proceedings of the IEEE Radar Conference, Boston, MA, USA, 17–20 April 2007; pp. 128–133.
12. Strohmer, T.; Friedlander, B. Compressed sensing for MIMO radar—Algorithms and performance. In Proceedings of the Conference of the Forty-Third Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–4 November 2009; pp. 464–468.
13. Herman, M.; Strohmer, T. Compressed sensing radar. In Proceedings of the IEEE Radar Conference, Rome, Italy, 26–30 May 2008; pp. 1–6.
14. Herman, M.A.; Strohmer, T. High-resolution radar via compressed sensing. *IEEE Trans. Signal Process.* **2009**, *57*, 2275–2284.
15. Qaisar, S.; Bilal, R.; Iqbal, W.; Naureen, M.; Lee, S. Compressive sensing: From theory to applications, a survey. *J. Commun. Netw.* **2013**, *15*, 443–456.
16. Donoho, D.; Tsaig, Y.; Drori, I.; Starck, J.-L. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **2012**, *58*, 1094–1121.
17. Dai, W.; Milenkovic, O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inf. Theory* **2009**, *55*, 2230–2249.
18. Chatterjee, S.; Hari, K.V.S.; Handel, P.; Skoglund, M. Projection-based atom selection in orthogonal matching pursuit for compressive sensing. In Proceedings of the National Conference on Communications (NCC), Kharagpur, India, 3–5 February 2012; pp. 1–5.
19. Septimus A.; Steinberg, R. Compressive sampling hardware reconstruction. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France, 30 May–2 June 2010; pp. 3316–3319.
20. Chatterjee, S.; Sundman, D.; Skoglund, M. Look ahead orthogonal matching pursuit. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, 22–27 May 2011; pp. 4024–4027.
21. Stanislaus, J.; Mohsenin, T. High performance compressive sensing reconstruction hardware with QRD process. In Proceedings of the IEEE International Symposium on Circuits and Systems, Seoul, South Korea, 20–23 May 2012; pp. 29–32.
22. Candes, E.; Wakin, M. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30.
23. Donoho, D. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306.
24. Zeng, J.; Dong, Z. Some MIMO radar advantages over phased array radar. In Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, 30–31 May 2010; pp. 211–213.
25. Fishler, E.; Haimovich, A.; Blum, R.; Cimini, R.; Chizhik, D.; Valenzuela, R. Performance of MIMO radar systems: Advantages of angular diversity. In Proceedings of the Conference of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 7–10 November 2004; Volume 1, pp. 305–309 .
26. Tropp J.; Gilbert, A. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **2007**, *53*, 4655–4666.
27. Bjorck, A. *Numerical Methods for Least Squatres Problems*; Society of Industrial and Applied Mathematics: Philadelphia, PA, USA, 1996.
28. Bailey, D.; Ferguson, H. A strassen-newton algorithm for high-speed parallelizable matrix inversion. In Proceedings of the 1988 Supercomputing, Orlando, FL, USA, 12–17 November 1988; Volume 1, pp. 419–424.
29. Huang, G.; Wang, L. High-speed signal reconstruction with orthogonal matching pursuit via matrix inversion bypass. In Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS), Quebec City, QC, Canada, 17–19 October 2012; pp. 191–196.
30. Jhang, J.-W.; Huang, Y.-H. A high-SNR projection-based atom selection OMP processor for compressive sensing. *IEEE Trans. Very Large Scale Integr. Syst.* **2015**, *23*, 2209–2220.
31. Lee, Y.-Y.; Wang, C.-H.; Huang, Y.-H. A Hybrid RF/Baseband Precoding Processor Based on Parallel-Index-Selection Matrix-Inversion-Bypass Simultaneous Orthogonal Matching Pursuit for Millimeter Wave MIMO Systems. *IEEE Trans. Signal Process.* **2015**, *63*, 305–317.

32. Rabah, H.; Amira, A.; Mohanty, B.K.; Almaadeed, S.; Meher, P.K. FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction. *IEEE Trans. Very Large Scale Integr. Syst.* **2015**, *23*, 2209–2220.
33. Lin, Y.-M.; Chen, Y.; Huang, N.-S.; Wu, A.-Y. Low-complexity stochastic gradient pursuit algorithm and architecture for robust compressive sensing reconstruction. *IEEE Trans. Signal Process.* **2017**, *65*, 638–650.