Research Article

Wavelet Compressive Sampling Signal Reconstruction Using Upside-Down Tree Structure

Yijiu Zhao, Xiaoyan Zhuang, Zhijian Dai, and Houjun Wang

School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Yijiu Zhao, yijiuzhao@gmail.com

This paper suggests an upside-down tree-based orthogonal matching pursuit (UDT-OMP) compressive sampling signal reconstruction method in wavelet domain. An upside-down tree for the wavelet coefficients of signal is constructed, and an improved version of orthogonal matching pursuit is presented. The proposed algorithm reconstructs compressive sampling signal by exploiting the upside-down tree structure of the wavelet coefficients of signal besides its sparsity in wavelet basis. Compared with conventional greedy pursuit algorithms: orthogonal matching pursuit (OMP) and tree-based orthogonal matching pursuit (TOMP), signal-to-noise ratio (SNR) using UDT-OMP is significantly improved.

1. Introduction

Compressive sampling (CS) [1, 2], as a new emerging signal processing theory, which has received considerable attention in signal processing. CS exploits the sparse structure in signal, and it enables signal reconstruction from a small number of random samples. A variety of signal recovery algorithms have been proposed to reconstruct the sparse signal. Generally, there are two classes of methods: convex optimization and greedy pursuit algorithms. Although the convex optimization method, such as basis pursuit (BP) [3], is powerful for sparse signal reconstruction, it may be computationally burdensome. Considering the computational complexity and difficulty of realization, greedy pursuit algorithms, especially matching pursuit (MP) [4] and orthogonal matching pursuit (OMP) [5], are attractive for engineering problems.

MP algorithm is computationally efficient and often features good performance; however, when the basis Ψ is not given by an orthogonal basis, MP algorithm cannot find the best approximation for original signal [6]. As an alternative, OMP orthogonalizes each

selected column vector associated with maximum projection in basis $\mathbf{\Psi}$, and OMP does not suffer the aforementioned flaw. In CS field, although there are many works about theory analysis and practice implementation of OMP, these recovery algorithms are generic in the sense that they do not exploit any particular structure in the signal besides its sparsity in some basis. However, for some signals there is additional a priori information that we can exploit for improving recovery performance. For example, the piecewise smooth signals, which are widely used in practice, are not only sparse in wavelet domain, but also form a connected subtree.

In this paper, we present an improved OMP signal recovery algorithm by employing an upside-down tree structure of signal in wavelet domain (we refer to this tree-based algorithm as UDT-OMP). Proposed algorithm is evaluated by signal-to-noise ratio (SNR) as a measure of quality of reconstructed signal. We have compared the performance of UDT-OMP with OMP and tree-based orthogonal matching pursuit (TOMP) with SNR as a function of number of measurements.

2. Compressive Sampling (CS) Background

CS is a novel sampling paradigm that goes against the common wisdom in data acquisition. CS states that a sparse or compressible signal can be recovered from a small salient set of random projections. To make it possible, there are two fundamental premises [7]: sparsity, which pertains to the signals of interest, and incoherence, which pertains to the sensing modality. Sparsity expresses the idea that the "information rate [8]" of a high-dimensional signal may be much lower than the maximum frequency presented in the signal. Consider an N-dimensional signal x, which can be sparsely represented in an appropriate basis $\mathbf{\Psi}$, and the transfer coefficients of x are given by $\alpha_i = < x, \psi_i > (i = 1, 2, \ldots, N)$. There are no more than K nonzero entries in $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T (K \ll N)$, and the signal x is called K-sparse signal. Just the opposite, incoherence means that the measurement matrix $\mathbf{\Phi}$ has dense representation in the basis $\mathbf{\Psi}$, and $\mathbf{\Phi}$ is independent of $\mathbf{\Psi}$.

CS also extends to that so-called compressible signals that are not exactly sparse but can be closely approximated as such (i.e., wavelet coefficients of signal and image). Sparse signals have coefficients $\boldsymbol{\alpha}$ that, when sorted, decay according to a power law: $|\alpha_i| < C \cdot i^{-1/p}$ for some positive constant C and $p \leqslant 1$; the smaller the decay exponent p, the faster the decay and the better the recovery performance we can expect from CS. In practice, most of man-made and natural signals are sparse or compressible in the sense that they have concise representations when expressed in an appropriate basis, such as Fourier basis and wavelet basis. For a K-sparse signal $\mathbf{x}$, we can find its M ($M = O(K \log(N/K))$ [9]) linear measurements $\mathbf{y} = \mathbf{\Phi x}(\mathbf{x} = \mathbf{\Psi \alpha})$, and $\mathbf{\Phi}$ is the measurement matrix of size $M \times N$ ($M < N$), and signal $\mathbf{x}$ can be reconstructed by solving the following inverse problem:

$$\min \|\boldsymbol{\alpha}\|_0 \quad \text{s.t. } \mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi \alpha}, \tag{2.1}$$

where the l_0 norm used here simply counts the number of nonzero entries in $\boldsymbol{\alpha}$.

Since $M < N$, (2.1) is an ill-posed problem, and there are many possible solutions of (2.1). The original signal $\mathbf{x}$ can be reconstructed from $\mathbf{y}$ by exploring its sparse expression, that is, among all possible $\widetilde{\boldsymbol{\alpha}}$ that satisfy $\mathbf{y} = \mathbf{\Phi \Psi \widetilde{\alpha}}$, seek the sparsest. It is known to be NP-hard to solve problem (2.1) and different suboptimal strategies are adopted in practice such as BP and OMP.

3. Orthogonal Matching Pursuit (OMP) Algorithm

MP approximation is improved by orthogonalizing the directions of projection with a Gram-Schmidt procedure; it is known as orthogonal matching pursuit (OMP). This orthogonalization was introduced in [6], and it was intensively studied in [5]. The improved precision of this orthogonalization is penalized by the higher computation of the Gram-Schmidt. In CS, OMP selects column vectors from dictionary $\mathbf{D}$ (= $\boldsymbol{\Phi}\boldsymbol{\Psi}$) that minimizes the difference, also called the residual $\mathbf{r}_k$, between measurement y and approximation. Specially, starting with $\mathbf{r}_0 = \mathbf{y}$, the OMP algorithm searches for the kth atom with maximum projection as

$$\lambda_k = \arg \max_i |\langle \mathbf{r}_{k-1}, \mathbf{d}_i \rangle|, \tag{3.1}$$

and it updates the residual as

$$\mathbf{r}_k = \mathbf{y} - P_{\mathrm{span}\{\mathbf{d}_{\lambda_1}, \mathbf{d}_{\lambda_2}, \dots, \mathbf{d}_{\lambda_k}\}} \mathbf{y}. \tag{3.2}$$

Here, $\mathbf{d}_i$ is the ith column vector of $\mathbf{D}$, and P_V denotes the orthogonal projection onto the subspace V, and $P_V = V(V^T * V) - 1 * V^T$. OMP algorithm in parallel applies the Gram-Schmidt orthogonalization upon chosen atoms for efficient computation of projections. It can be formulated as a *subset selection problem* where a minimum subset of columns of $\mathbf{D}$ matrix is chosen to approximate the observation vector $\mathbf{y}$ in the least square sense. The OMP algorithm successively chooses an additional column of $\mathbf{D}$ matrix to reduce the approximation error. Equivalent, the OMP method begins with a tentative solution of $\boldsymbol{\alpha}$ with a single nonzero entry, and gradually adds nonzero entries one by one until the approximation error of $\mathbf{y}$ meets a pre-determined criterion. The accuracy of approximation increases with the number of iterations; however, the iterative number must not be bigger than the number of random measurements and we cannot expect to recover signal from higher dimensions than that of measurements.

4. UDT-OMP Algorithm

4.1. Tree Structure in Wavelet Domain

In CS, both BP and OMP reconstruct signal based on sparsity without considering any other particular structure that may exist in the signal. The wavelet transform of a piecewise smooth signal (many punctured real-world phenomena give rise to such signal), an important subclass of sparse signals, yields a sparse, structured representation of signals in this class: the significant coefficients tend to form a connected subtree of the wavelet coefficient tree. Figure 1 shows an example of wavelet representation of such signal. Only a few wavelet coefficients are significant, which form sparse subtrees when represented in multiscale wavelet transform. The number of such subtrees equals to that of the discontinuities in signals.

In this work, we only focus on 1D signals, and similar arguments apply for 2D and multidimensional signals. Consider a signal $\mathbf{x}$ of length $N = 2^J$, given a bandpass wavelet function $\psi(t)$ and a lowpass scaling function $\phi(t)$, the discrete wavelet transform (DWT)

coefficients $\boldsymbol{\alpha}$ of size N in terms of shifted versions of $\phi(t)$ and shifted and dilated versions of $\psi(t)$. The wavelet representation of $\mathbf{x}$ is given [10]

$$\mathbf{x} = \sum_{i=0}^{N_J-1} v_{J,i}\phi_{J,i} + \sum_{j=1}^{J} \sum_{i=0}^{N_J-1} w_{j,i}\psi_{j,i}, \tag{4.1}$$

where J denotes the scale of analysis and scale J indicates the coarsest scale or lowest resolution of analysis. $N_j = N/2^j$ is the number of coefficients at scale $j \in \{1, \ldots, J\}$, and i is the position, $0 \leq i \leq 2^j - 1$. In terms of matrix notation, $\mathbf{x}$ has the representation $\mathbf{x} = \boldsymbol{\Psi\alpha}$, where $\boldsymbol{\Psi}$ is a matrix containing the scaling and wavelet functions as columns and the scaling and wavelet coefficients are as follows:

$$\boldsymbol{\alpha} = \begin{bmatrix} v_{J,i} & w_{0,0} & w_{1,0} & w_{2,0} & \ldots \end{bmatrix}^T. \tag{4.2}$$

According to the statistical analysis in wavelet domain, wavelet coefficients have the following two properties [11].

> *Compression*. The wavelet transforms of real-world signals decay exponentially as the scale become finer, and they tend to be sparse (as depicted in Figure 1(b)).
>
> *Tree structure*. Those significant wavelet coefficients propagate across scale in wavelet tree, and they are well organized in a tree structure (as depicted in Figure 1(c)).

This tree structure was exploited by previous CS reconstruction algorithms known as iterative reweighted l_1-norm minimization with the wavelet Hidden Markov tree model (HMT-IRWL1) [12] and TOMP [13]. HMT-IRWL1 integrates the HMT model to enforce the wavelet coefficient structure during IRWL1, which updates the weight values with state probability of HMT model and highly depends upon accuracy of model. If accurate Markov model is available, then the HMT-IRWL1 could be powerful to recover the sparse signal. Although, in practice CS applications, only a small set of random measurements are available, one cannot get an accurate Markov model from these measurements. TOMP evaluates the sums of projections along each wavelet coefficients connected by a subtree, which is an improved OMP recovery algorithm. Both HMT-IRWL1 and TOMP are based on upright subtree, and one father node connects with two children nodes, as depicted in Figure 1(d).

4.2. UDT-OMP Algorithm

In previous tree-based CS reconstruction algorithms, trees are assumed to be upright, which means that those significant coefficients propagate from coarser scale to finer scale. Examining the tree structure in Figure 1(c), we noticed that not only do they organize in subtrees, but also those subtrees are upside down. From finer scale to coarser scale, wavelet coefficients become larger, and there are more significant coefficients in coarser scales. In this paper, we motivate an improved version OMP by employing weighting an upside-down tree (UDT-OMP).

The input of UDT-OMP is a dictionary $\mathbf{D}$ with size of $M \times N$ $(M \ll N)$, a measurement vector $\mathbf{y}$ with length of M, upward extending coefficient l, weighting value c, and iterative number K. UDT-OMP returns a reconstruction sparse vector $\tilde{\boldsymbol{\alpha}}$ of length N that is subject to $\mathbf{y} = \boldsymbol{\Phi\Psi\alpha}$.
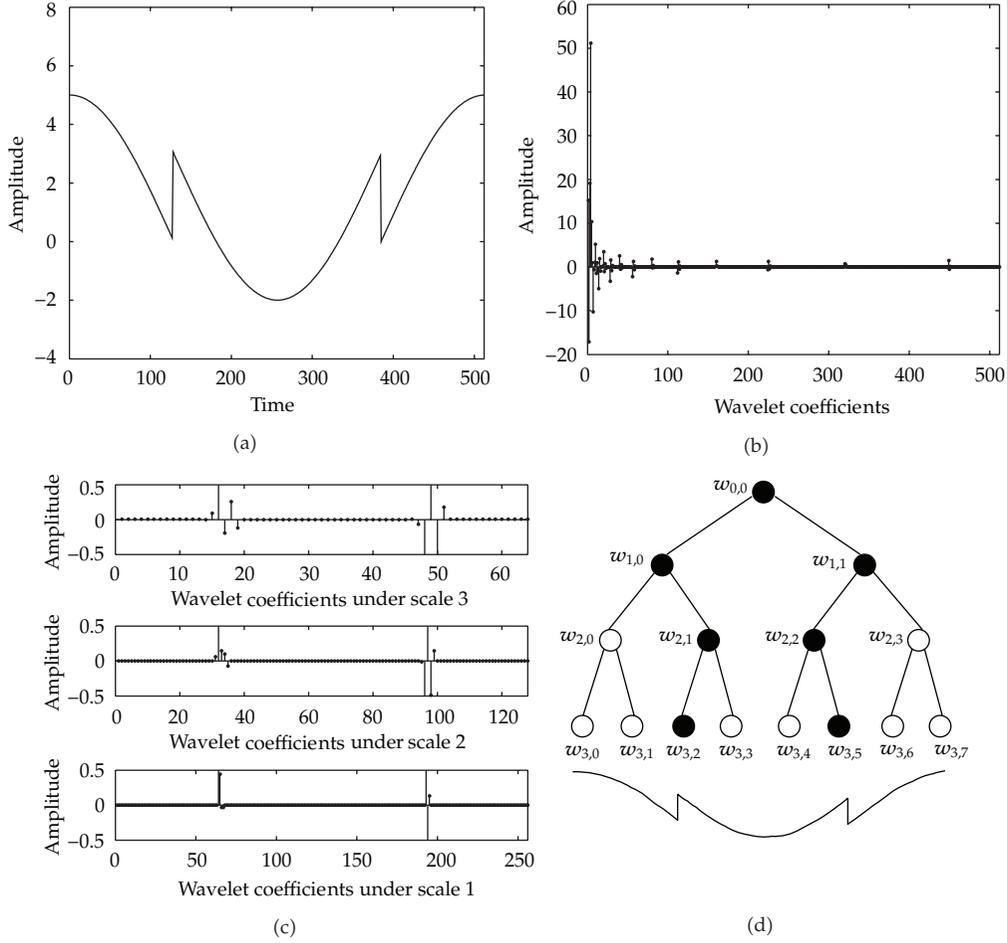
Figure 1: Example of sparse representations in the wavelet domain: (a) piecewise smooth signal; (b) wavelet transform of piecewise smooth signal, only few coefficients are significant and they are compressible or near sparse; (c) the significant coefficients are well organized in tree structure across the scales; (d) binary tree for piecewise smooth signal, the significant wavelet coefficients arise from the discontinuities in the signal (the black circles denote the large wavelet coefficients).

UDT-OMP evaluates the projection of each single column vector and searches for the maximum projection. According to the columns associated with the maximum projections, UDT-OMP constructs an upside-down subtree upward coarser scale with depth of l, which will be weighted in the next iteration of searching for the maximum projection.

In (4.2), we noticed that the wavelet transform of signal α can be divided into two parts: scaling coefficients and wavelet coefficients [13]

$$\alpha = \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix}, \tag{4.3}$$

where $\mathbf{v}$ contains all the scaling coefficients and $\mathbf{w}$ contains all the wavelet coefficients of signal $\mathbf{x}$. Since the scaling coefficients are significant, we aim to recover all the scaling

coefficients $\mathbf{v}$ and those significant wavelet coefficients in $\mathbf{w}$. In particular, we recover $\mathbf{v}$ and $\mathbf{w}$ separately and revise (2.1) as follows:

$$\min \|\mathbf{w}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi} \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix}. \tag{4.4}$$

UDT-OMP algorithm consists of two steps. We first limit our search space in the columns associated with the scaling coefficients and then in the columns associated with the wavelet coefficients. Let $\mathbf{\Lambda}_k$ be the selected set of columns in dictionary $\mathbf{D}$ from the beginning k iterations. Let $\mathbf{U}_k$ be the Gram-Schmidt version of $\mathbf{\Lambda}_k$. According to (3.1) and (3.2), OMP aims to search for the columns with the maximum projections, which associate with the significant coefficients. Firstly, we set the selected sets $\mathbf{\Lambda}_0$ and $\mathbf{U}_0$ to be empty. In step 1, we select all columns of $\mathbf{D}$ corresponding to the scaling coefficients since these coefficients are significant. According to the construction of wavelet basis $\mathbf{\Psi}$, the wavelet transform level J (also called the scale of analysis) is deterministic and so as the positions of scaling coefficients (as depicted in (4.2)):

$$\mathbf{\Lambda}_0 = \mathbf{D}\{\text{all columns associated with the scaling coefficients}\}. \tag{4.5}$$

All vectors in $\mathbf{\Lambda}_0$ are sequentially orthogonalized using Gram-Schmidt and stored in $\mathbf{U}_0$. At end of step 1, the residual is updated by

$$\mathbf{r}_0 = \mathbf{y} - P_{\text{span}\{\mathbf{\Lambda}_0\}}\mathbf{y}. \tag{4.6}$$

In step 2, we focus on recovering the wavelet coefficients which have sparse tree structure. Step 2 is the repetition of iterations. Let $\{\mathbf{S}_k\}_{k=0}^K$ be the candidate set of vectors, which will be weighted in the next searching of maximum projection. We initialize all the elements of $\mathbf{S}_0$ as 1.

In the first iteration, we initialize the counter $k = 1$. Then UDT-OMP investigates the weighted projections of the current residual on all the column vectors of $\mathbf{D}$

$$\theta_i^k = \mathbf{S}_{k-1}(i) \cdot |\langle \mathbf{r}_{k-1}, \mathbf{d}_i \rangle|,$$
$$\lambda^k = \arg \max_{1 \le i \le N} \left\{ \theta_i^k \right\}. \tag{4.7}$$

According to the maximum projection position i, we construct the kth upside-down subtree T_k with depth of l (l is an integer, l should be greater than 1 and not greater than the level of wavelet transform). At each finer scale, every four consecutive nodes connect two nodes of nearest coarser scale, which forms an upside-down subtree. For example, if $w_{3,2}$ (as depicted in Figure 1(d)) is current maximum projection, $w_{2,0}$ and $w_{2,1}$ (both connect with four consecutive nodes $w_{3,0}$, $w_{3,1}$, $w_{3,2}$, and $w_{3,3}$) would be children nodes in the nearest coarser scale, and $w_{1,0}$ and $w_{1,1}$ are children nodes in the next coarser scale:

$$\mathbf{S}_k(i) = \{c : i \in T_k\}. \tag{4.8}$$

Inputs: CS matrix $\mathbf{D}$, measurement $\mathbf{y}$, upward extending coefficient l, weighting value c, and iterative number K, wavelet transform level J, $\mathbf{S}_0 = \{1\}$
Outputs: approximated signal $\widetilde{x}$
Initialize: $\boldsymbol{\alpha}_0 = \{0\}$, $\mathbf{r}_0 = \{0\}$, $k = 0$, $\boldsymbol{\Lambda}_0 = \{1,2,\ldots, N_J \, (= N/2^J)\}$
$r_0 = y - P_{\text{span}\{\Lambda_0\}} y$
While halting criterion false **do**
     (1) $k = k + 1$
     (2) $\boldsymbol{\theta}_i^k = \mathbf{S}_{k-1}(i) \cdot |\langle \mathbf{r}_{k-1}, \mathbf{d}_i \rangle|$
     (3) $\lambda^k = \arg \max_{1 \leq i \leq N} \{\boldsymbol{\theta}_i^k\}$
     (4) construct upside-down tree T_k
       $\mathbf{S}_k(i) = \{c : i \in T_k\}$
     (5) $\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda}_{k-1} \cup \lambda^k$
     (6) $\mathbf{U}_k = P_{\text{GS}}(\boldsymbol{\Lambda}_k)$
     (7) $\mathbf{r}_k = \mathbf{y} - P_{\text{span}\{\Lambda_k\}} \mathbf{y}$
end while
$\widetilde{\boldsymbol{\alpha}} = \mathbf{U}_k \mathbf{y}$
return $\widetilde{\mathbf{x}} = \boldsymbol{\Phi\Psi}\widetilde{\boldsymbol{\alpha}}$

Algorithm 1: Upside-down tree orthogonal matching pursuit.

Since we want to enforce the selected columns, the weighting value c should be greater than 1. We add the newly chosen node into our selected set $\boldsymbol{\Lambda}_k$:

$$\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda}_{k-1} \cup \lambda^k,$$
$$\mathbf{U}_k = P_{\text{GS}}(\boldsymbol{\Lambda}_k), \tag{4.9}$$

where $P_{\text{GS}}(\cdot)$ denotes orthogonalization using Gram-Schmidt. The residual is updated as follows:

$$\mathbf{r}_k = \mathbf{y} - P_{\text{span}\{\Lambda_k\}} \mathbf{y}. \tag{4.10}$$

After K iterations of (4.7)–(4.10), the significant coefficients are determined by the columns in $\boldsymbol{\Lambda}_k$, and they are represented as follows:

$$\widetilde{\boldsymbol{\alpha}} = \mathbf{U}_k \mathbf{y}, \tag{4.11}$$

and the approximation of the original signal is

$$\widetilde{\mathbf{x}} = \boldsymbol{\Psi}\widetilde{\boldsymbol{\alpha}}. \tag{4.12}$$

The pseudocodes of UDT-OMP are described as in Algorithm 1.
For simplicity, sparsity level K (assumed known) can be used as halting criterion here. If K is unknown, we can modify the iteration in the above by letting k run from 1 to M but adding a threshold ε for $|\mathbf{r}_k|_1$ below which the iteration is terminated.

(a)

(b)

(c)

(d)
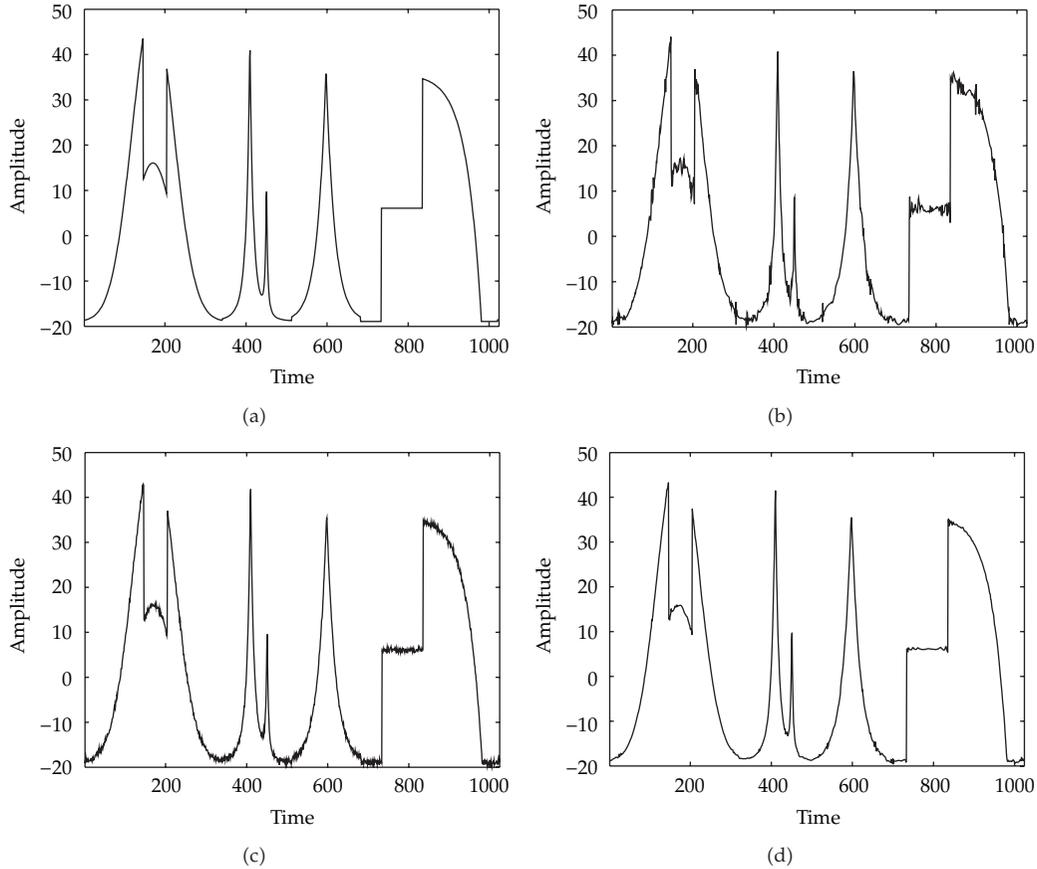
Figure 2: An example piecewise smooth signal of length 1024 and its reconstructions from 300 random measurements using OMP, TOMP, UDT-OMP. (a) Original signal. (b) OMP reconstructed signal, SNR = 27.63 dB. (c) TOMP reconstructed signal, SNR = 34.57 dB. (d) UDT-OMP reconstructed signal, SNR = 39.62 dB.

5. Experiment Result

To demonstrate the advantage of upside-down tree structure, we evaluated UDT-OMP algorithm by comparing the performance of OMP, TOMP, and UDT-OMP. We used piecewise smooth signal of length $N = 1024$. 4-level Daubechies 8 wavelets were applied to sparsify the test signal, upward extending coefficient was set $l = 2$, and weighting value $c = e$ (natural number). Samples were obtained using a measurement matrix with i.i.d. Gaussian entries.

In the first experiment, we reconstructed the signal from 300 random measurements by OMP, TOMP, and UDT-OMP, respectively. Figure 2 depicts the reconstructions, the SNR of the OMP reconstructed signal is 27.63 dB, the SNR of the TOMP reconstructed signal is 34.57 dB, and the UDT-OMP reconstruction achieves SNR of 39.62 dB. In this experiment, UDT-OMP gains more than about 12 dB and 5 dB over OMP and TOMP, respectively. From experimental result, the advantage of the UDT-OMP method is clearly demonstrated.

In the second experiment, we reconstructed the test signal from different numbers of measurements using OMP, TOMP, and UDT-OMP. Numbers of measurements over the range of 100 to 400 in increment of 30 are tested. 100 trials are repeated for each specific
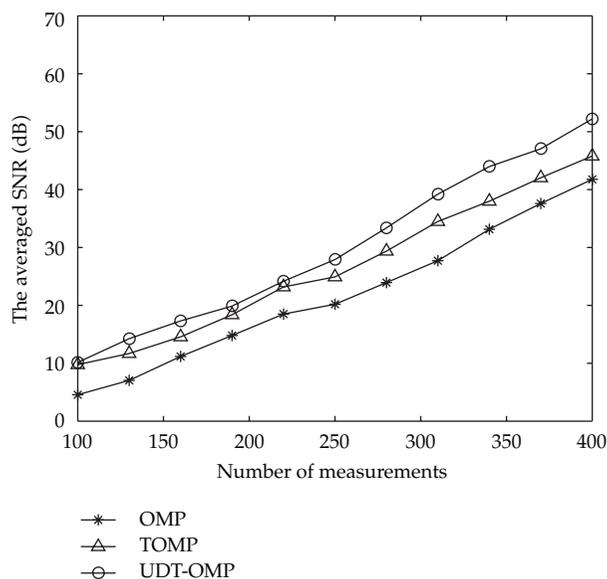
Figure 3: Comparison between the performance of OMP, TOMP, and UDT-OMP.

number, and the averaged reconstruction SNR is plotted in Figure 3. Through exploring the upside-down sparse tree structure, the UDT-OMP method outperforms OMP and TOMP in piecewise smooth signal reconstruction.

6. Conclusions

This paper introduced an upside-down tree structure weighting scheme for OMP algorithm in wavelet domain CS signal reconstruction. The UDT-OMP weights the nodes that connect in the subtree with significant values. Different from tree structure presented in the previous CS recovery algorithms, UDT-OMP constructs the tree using upside-down structure rather than upright structure. UDT-OMP weights the projections that should have larger coefficients. The experimental results show that our method outperforms OMP and TOMP, and it can achieve more accurate approximation in piecewise smooth signal reconstruction. In this paper, we only considered the constant weight value. We can also adopt different weight values at different scales.
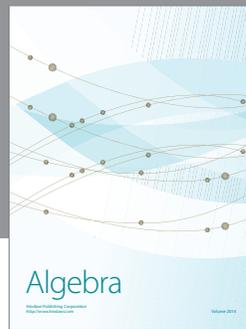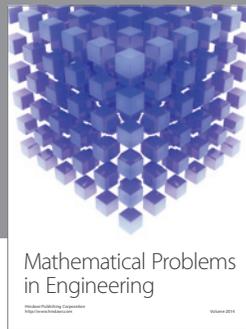
Acknowledgments

References

[1] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[4] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[5] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[6] R. A. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Advances in Computational Mathematics*, vol. 5, no. 2-3, pp. 173–187, 1996.

[7] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling: a sensing/sampling paradigm that goes against the common knowledge in data acquisition," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[8] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations: theory, algorithms, and performance bounds," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 31–40, 2008.

[9] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[10] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Elsevier/Academic Press, Amsterdam, The Netherlands, 3rd edition, 2009.

[11] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.

[12] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a Hidden Markov Tree Model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 5137–5140, Las Vegas, Nev, USA, April 2008.

[13] C. La and M. N. Do, "Signal reconstruction using sparse tree representations," in *Wavelets XI*, vol. 5914 of *Proceedings of SPIE*, pp. 1–11, San Diego, Calif, USA, August 2005.