ICIC Express Letters
Volume **10**, Number **9**, September **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 2211–2217

A RANDOM REGULARIZATION MATCHING PURSUIT ALGORITHM FOR TOMOSAR IMAGING

Yongjie Luo, Xunchao Cong and Qun Wan

School of Electronic Engineering
University of Electronic Science and Technology of China
No. 2006, Xiyuan Ave., West Hi-Tech Zone, Chengdu 611731, P. R. China
yongjie.luo@163.com; 201311020326@std.uestc.edu.cn; wanqun@uestc.edu.cn

Received January 2016; accepted April 2016

Abstract. *In this paper we propose a new matching pursuit algorithm, termed Random Regularized Matching Pursuit (RrMP), to solve compressed sampling tomoSAR imaging problem. This method pursues multiple columns of a sensing matrix at each iteration, randomly bisects these columns, chooses the subset with smaller residual energy, makes a dropout/replacement operation to the entries of the subset by a rule, and estimates the amplitudes supporting on the candidate entries. The advantages of our proposed algorithm are wider sparse reconstruction range and less CPU time, and it has ability to save more CPU time by parallel computing.*

Keywords: TomoSAR imaging, Compressed sensing, Matching pursuit, Random regularization

1. **Introduction.** SAR imaging is an important application in radar signal processing. TomoSAR imaging is a spatial scatterer distribution reconstruction problem which decodes the reflector position from the phase shift hiding in successively received signals [10]. The tomoSAR imaging algorithms can be categorised to two classes: finding dense solution and finding sparse solution. The former is based on the Nyquist sampling theory and discrete Fourier transform, e.g., Polar Format Algorithm (PFA) [11, 12]; the latter can be regarded as finding the sparse solution of an under-determined linear system, so it is naturally induced to the Compressed Sensing (CS) problem [1, 2, 3, 4]. Even a small size of two-dimensional SAR image, e.g., 100 × 100, becomes large if the image is reshaped from a matrix form to a vector form. It makes the well-known sparse-induced methods, e.g., ℓ_0 norm optimization algorithms Orthogonal Matching Pursuit (OMP) [5], Compressive Sampling Matching Pursuit (CoSaMP) [6], run slowly. In response to this issue, we propose a new matching pursuit based algorithm, termed Random Regularized Matching Pursuit (RrMP). This algorithm utilizes some kind of bisecting and dropout/replacement operation to regularize the indexes of sparse representation vector. In this way RrMP algorithm achieves less CPU time and wider sparsity range while keeping the same performance, compared with OMP, CoSaMP, Basis Pursuit (BP) [7] and Approximate Message Passing (AMP) [8] algorithms.

The rest of this paper is organized as follows. We describe the tomoSAR imaging physical model and the equivalent compressed sensing model in Section 2, and then explain the RrMP algorithm in Section 3, and then make experiments in Section 4. We draw a conclusion in the last section.

2. **Problem Formulation.** In this work, we give some assumptions of free space propagation, narrow bandwidths, far-field plane waves, and a linear single bounce (Born) scattering approximation. This treatment is standard and follows from a theoretical analysis of the scalar wave equation [9]. Figure 1 shows the diagram of tomoSAR physical model. Suppose the origin of cartesian system (u, v) is the space rotation center (o), and

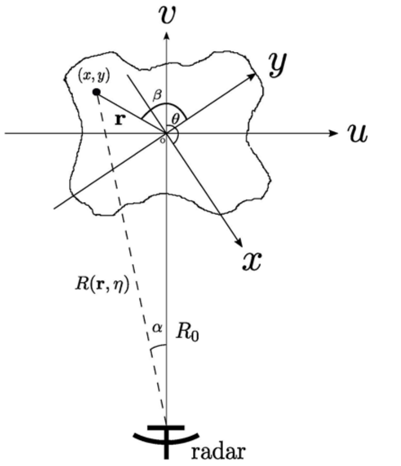


Figure 1. The original physical model of tomoSAR imaging

the target is fixed at (o) when a radar measures it. A radar is located at $(0, R_0)$, where R_0 is a constant. Another cartesian system (x, y) overlaps on the previous cartesian system (u, v) with a bias angle θ between y axis and v axis. The coordinate transform is given by

$$u = x\cos\theta + y\sin\theta = r\sin(\beta - \theta), \tag{1}$$

$$v = y\cos\theta - x\sin\theta = r\cos(\beta - \theta). \tag{2}$$

The shape of target is irregular, and it can be partitioned to grids. Scatterers are deployed in this area. $\boldsymbol{r}$ denotes a vector from the origin to a scatterer located at (x, y), and β denotes the rotation angle between $\boldsymbol{r}$ and y axis. The distance from the radar to the scatterer is $R(\boldsymbol{r}, \eta)$, where η is a slow time parameter. Assume Doppler shift of target is zero for tomoSAR imaging. Define the Linear Frequency Modulation (LFM) signal in radar application as follows

$$p(t) = w(t)\exp\left\{j\left(2\pi f_c t + K_r \pi t^2\right)\right\}, \tag{3}$$

where f_c is carrier frequency, K_r is frequency modulation rate, $w(t)$ is a rectangular window function, $w(t) = 1$ for $0 \le t \le \tau_w$, and $w(t) = 0$ for otherwise. The received echo signal is given by

$$y(t, \eta) = \int_r f(r)p\left(t - \frac{2R(\boldsymbol{r}, \eta)}{c}\right) + \epsilon(t), \tag{4}$$

where $f(r)$ is a scattering function, $p\left(t - \frac{2R(\boldsymbol{r},\eta)}{c}\right)$ is the delay function of (3). Assume the incident ray is a plane wave. After de-chirp operation, the two dimensional spatial spectra of tomoSAR echo are given by

$$Y(f, \theta) = \int_x \int_y g(x, y)\exp\{-2jf(x\cos\theta + y\sin\theta)\}, \tag{5}$$

where $f = 2\pi/\lambda$ is spatial frequency, λ is wavelength, (x, y) is the position of a scatterer in the target coordinate, θ is the rotation angle between radar coordinate and target coordinate, and $g(x, y)$ is the scattering coefficient. Using summation to replace integration, the discrete variables (in vector form) are: spatial frequency $\mathbf{f} \in \mathbb{C}^P$, where P is frequency sampling number; rotation angle $\boldsymbol{\theta} \in \mathbb{C}^Q$, where Q is angle sampling number; scattering coefficient vector $\mathbf{x} \in \mathbb{C}^N$ which is composed of $g(x, y)$, where $N \ge P \times Q$ is the pixels number of a tomoSAR image; received echo signal $\mathbf{y} \in \mathbb{C}^M$; and Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{C}^M$, where $M = P \times Q$. The linear system form of (5) is

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \tag{6}$$

where $F(p,q,n) \triangleq \exp\{-2jf_p(x_n\cos\theta_q + y_n\sin\theta_q)\}$, and then

$$\mathbf{A} = \begin{pmatrix} F(1,1,1) & F(1,1,2) & \cdots & F(1,1,N) \\ \cdots & \cdots & \cdots & \cdots \\ F(P,1,1) & F(P,1,2) & \cdots & F(P,1,N) \\ \cdots & \cdots & \cdots & \cdots \\ F(1,Q,1) & F(1,Q,2) & \cdots & F(1,Q,N) \\ \cdots & \cdots & \cdots & \cdots \\ F(P,Q,1) & F(P,Q,2) & \cdots & F(P,Q,N) \end{pmatrix}.$$

Notice that the scatterers are very sparse in a tomoSAR image. According to the compressed sampling theory, it is feasible to randomly draw some rows from $\mathbf{A}$ to decrease P and Q which are corresponding to the sampling rate. In this way people can use cheaper commercial device to design the tomoSAR imaging instrument.

To solve the linear system (6), we consider an optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_p \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \le \epsilon. \tag{7}$$

Thanks to the very sparse property of the tomoSAR image, we set $p = 0$ to induce an ℓ_0 norm optimization for (7) since the ℓ_0 norm optimization is more applicable to very sparse case than ℓ_1 norm optimization.

3. **RrMP Algorithm.**

In this section we propose the RrMP algorithm. Our original idea is simple: firstly, to accelerate pursuit speed the algorithm should find multiple columns of the sensing matrix $\mathbf{A}$ at each iteration, like the improvement of CoSaMP over OMP; secondly, since CoSaMP does not work well for strong column correlation sensing matrix, the algorithm should not pursue too many columns at each iteration, and it must do some kind of regularization for the indexes of the columns. Following this idea, we explain the steps of RrMP algorithm.

We use $\mathbf{A}'$ to denote transpose, the subscript l to denote left, r to denote right, and $|U|$ to designate the cardinality of a temporary support U. We define two functions. $R_{2s\setminus s}(\mathbf{v})$ finds the indexes of $2s < K$ absolutely largest entries of $\mathbf{v}$, where s is the probe length, and then shuffles these indexes and bisects them. Outputs of this function are the left and right indexes subset and the amplitudes supported on them, respectively. Support updating function $H(\mathbf{u}, \mathbf{c}, \Lambda, S)$ generates a new indexes set by a group of rules according to the input parameters, and we will explain it later.

3.1. **Algorithm description.**

The steps of RrMP are listed in Algorithm 1. Line 2 obtains two index candidates, Λ_l and Λ_r, and the correlation coefficients, $\mathbf{c}_l$ and $\mathbf{c}_r$, between the residual and columns of $\mathbf{A}$. It is a key step, since the shuffle and bisecting operation are equivalent to locally random search in an N dimensional $\{0,1\}$ space. Line 3 merges old support set S^n with the two candidates, respectively. Lines 4 and 5 roughly estimate $\mathbf{x}$ by the Least Squares (LS) operation. Lines 6 and 7 choose the minimal residual branch as the new candidate. This prune operation can be regarded as a regularization. Line 8 updates the support set. Line 9 computes a refined estimation of $\mathbf{x}$ based on the so-obtained support in line 8. It is worth noting that lines 3-5 can be done by parallel computing, on account of the independent left subset and right subset. In this way about 1/3 CPU time can be saved, since the CPU time of matching pursuit based methods mainly depends on the LS computation.

Now we define the support updating function $H\left(\mathbf{u}^{n+1}, \mathbf{c}^{n+1}, \Lambda^{n+1}, S^n\right)$ as follows. Firstly, define two threshold functions corresponding to the sparse representation coefficients $\mathbf{u}$ which are evolved with iterations

$$a \triangleq \min_{i\in S^n}\left\{\left|\left(\mathbf{u}^{n+1}\right)\right|_i\right\}, \quad b \triangleq \max_{j\in\Lambda^{n+1}}\left\{\left|\left(\mathbf{u}^{n+1}\right)\right|_j\right\}. \tag{8}$$

Algorithm 1 Random regularized matching pursuit algorithm

```
Input: $\mathbf{y}$, $\mathbf{A}$, $K$, $s$
Output: $\mathbf{x}^{n+1}$
$S^0 \leftarrow \emptyset$, $\mathbf{x}^0 \leftarrow \mathbf{0}$
$(\Lambda_l, \mathbf{c}_l, \Lambda_r, \mathbf{c}_r) \leftarrow R_{2s\backslash s}(\mathbf{A}'(\mathbf{y} - \mathbf{A}\mathbf{x}^n))$
$U_l \leftarrow S^n \cup \Lambda_l$, $U_r \leftarrow S^n \cup \Lambda_r$
$\mathbf{u}_l \leftarrow \arg\min_{\mathbf{z}}\{\|\mathbf{y} - \mathbf{A}\mathbf{z}\|_2, \mathbf{z} \subset \mathit{supp}(U_l)\}$
$\mathbf{u}_r \leftarrow \arg\min_{\mathbf{z}}\{\|\mathbf{y} - \mathbf{A}\mathbf{z}\|_2, \mathbf{z} \subset \mathit{supp}(U_r)\}$
$\theta = \arg\min_{\{l,r\}} \{\|\mathbf{y} - \mathbf{A}\mathbf{u}_l\|_2, \|\mathbf{y} - \mathbf{A}\mathbf{u}_r\|_2\}$
$\Lambda^{n+1} \leftarrow \Lambda_\theta$, $\mathbf{c}^{n+1} \leftarrow \mathbf{c}_\theta$, $\mathbf{u}^{n+1} \leftarrow \mathbf{u}_\theta$, $U^{n+1} \leftarrow U_\theta$
$S^{n+1} \leftarrow H\left(\mathbf{u}^{n+1}, \mathbf{c}^{n+1}, \Lambda^{n+1}, S^n\right)$
$\mathbf{x}^{n+1} \leftarrow \arg\min_{\mathbf{z}} \{\|\mathbf{y} - \mathbf{A}\mathbf{z}\|_2, \mathbf{z} \subset \mathit{supp}(S^{n+1})\}$
```

Secondly, define the indexes updating procedure just for $\mathbf{u}$

$$P \triangleq \begin{cases} S^n \cup \left\{ j : \arg\max_{j \in \Lambda^{n+1}} \left\{ \left|\left(\mathbf{u}^{n+1}\right)_j\right| \right\} \right\}, & \frac{a}{2} > b \\ S^n \cup \left\{ j : j \in \Lambda^{n+1}, \left|\left(\mathbf{u}^{n+1}\right)_j\right| \geq \frac{a}{2} \right\}, & a > b \geq \frac{a}{2} \\ S^n \cup \left\{ j : j \in \Lambda^{n+1}, \left|\left(\mathbf{u}^{n+1}\right)_j\right| \geq \frac{b}{2} \right\}, & b > a \geq \frac{b}{2} \\ \left\{ i : i \in S^n, \left|\left(\mathbf{u}^{n+1}\right)_i\right| \geq \frac{b}{2} \right\} \cup \left\{ j \in \Lambda^{n+1}, \left|\left(\mathbf{u}^{n+1}\right)_j\right| \geq \frac{b}{2} \right\}, & \frac{b}{2} \geq a. \end{cases} \tag{9}$$

Thirdly, define the the indexes updating procedure just for the correlation coefficients

$$Q \triangleq \left\{ j : j \in \Lambda^{n+1}, \left|\left(\mathbf{c}^{n+1}\right)_j\right| \geq 0.5 \left\|\mathbf{c}^{n+1}\right\|_\infty \right\}. \tag{10}$$

Lastly, we merge the two updated indexes to obtain the output of $H\left(\mathbf{u}^{n+1}, \mathbf{c}^{n+1}, \Lambda^{n+1}, S^n\right)$

$$S^{n+1} = P \cup Q. \tag{11}$$

The idea of this support updating function comes from the following two points: firstly, not all the indexes in the candidate set are correct, the algorithm should drop or replace some indexes, as shown in (9) and (10), and these dropout/replacement rules can be considered as a kind of regularization; secondly, the candidate indexes corresponding to correlation coefficients and to sparse vector do not necessarily overlap entirely, and some correct indexes maybe belong to the former and some belong to the latter; therefore, it is necessary to merge them, as shown in (11).

When $|P|$ is equivalent to the sparsity K, if the precision does not reach the error bound, set $S^{n+1} = P$, we keep running RrMP at most $2s$ iterations and then stop the algorithm.

3.2. **Computational complexity.** For matching pursuit based algorithms, the main CPU time exhausts in LS steps. It is well known that *Householder LS* method needs $O\left(2N^2(M - N/3)\right)$ flop. We cannot accurately determine how many candidates in the s new column indexes can be picked according to $H\left(\mathbf{u}^{n+1}, \mathbf{c}^{n+1}, \Lambda^{n+1}, S^n\right)$ function, but if we roughly set the number of winners in these candidates to $s/2$, the computational complexity of RrMP can be estimated as $O([(4M - K - 2)K^3 + (6M - 1)K^2 + 2MK]/s)$.

4. **Experiments.** In this section, we investigate the advantages of RrMP algorithm via experiments, in comparison with five well-known algorithms: OMP and CoSaMP, which come from Needell's work[1]; AMP and GAMP, which come from Kamilov's work[2] and

[1] http://www.cmc.edu/pages/faculty/DNeedell

[2] http://people.epfl.ch/ulugbek.kamilov

Rangan's work *gamplab*[3]; we use Matlab's **linprog** function to solve BP. The Relative Mean Square Error (RMSE) is regarded as the performance metric, $RMSE = 1/Q \times \sum_{l=1}^{Q} \left\|\hat{\mathbf{x}}^l - \mathbf{x}^l\right\|_2 / \left\|\mathbf{x}^l\right\|_2$, where $\hat{\mathbf{x}}^l$ is the estimate of $\mathbf{x}$ at the l-th trial. To avoid symbol confusion, notation L is in place of notation s in Algorithm 1, e.g., RrMP-L4 means probe length 4. We use Matlab's **pinv** function to implement LS in RrMP, OMP and CoSaMP algorithms. We do not use parallel computing to lines 3-5 in Algorithm 1 for fair play.

The first experiment investigates exact reconstruction frequency variation with the growing of sparsity K. We set the exact reconstruction threshold $\xi = 0.015$, since the energy of noise $\boldsymbol{\epsilon}$ is set to 0.0015, i.e., one-tenth of ξ. One trial is regarded as successful if the RMSE is less than ξ, and that means RMSE does not exceed 10 times of the energy of noise. The exact reconstruction frequency can be obtained by 200 trials. Figure 2 shows that each of RrMP-L2/L4/L6/L8 achieves obviously higher exact reconstruction frequency than OMP, CoSaMP, BP and AMP when the sparsity K grows up. Define the sparsity ratio as $\delta = K/N$. It is worth noting that the performance of CoSaMP decreases drastically when K is greater than 40, or equivalently $\delta > 15.6\%$, and the performance of BP and AMP also decline fast with greater K. In contrast, RrMP works well in wider sparsity range about 30% more than CoSaMP, and about 20% more than BP and AMP.

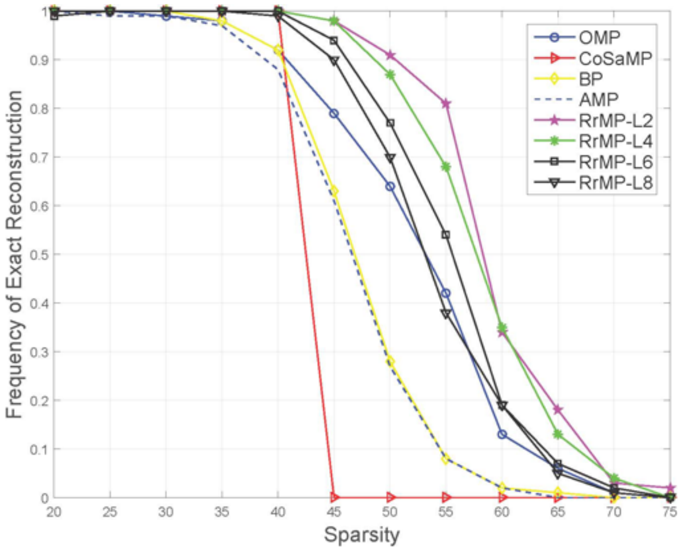
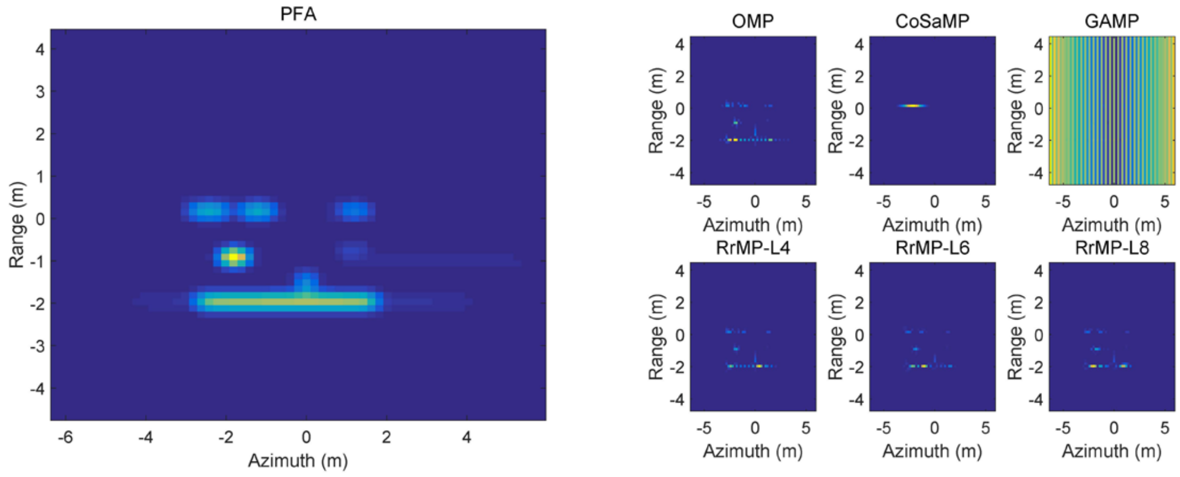


FIGURE 2. Exact reconstruction frequency comparison, $N = 256$, $M = 128$

The second experiment shows tomoSAR image reconstruction result. The target is a real world crawler crane. TomoSAR carrier frequency is $f_c = 9$(GHz). There are $P = 101$ frequency samples drawn over 1(GHz) bandwidth centered at f_c, and $Q = 101$ angle samples drawn over $87.5° \sim 92.5°$ centered at 90° azimuth, uniformly spaced respectively. The total pixels number is $N = P \times Q = 10201$. We create $\mathbf{A}$ from (6), and then draw $M = \lfloor 0.5N \rfloor$ rows randomly. Since the measurements number M is usually chosen to be $O(K \log N)$, K can be estimated roughly to $\lambda M / \log N$, and we set $K = 60$. Using PFA reconstruction result as a reference, we compare OMP, CoSaMP, GAMP with RrMP-L4/L6/L8. Note that AMP algorithm is not suitable for highly correlated sensing matrix; therefore, we use GAMP in place of AMP to see whether it can work in the tomoSAR imaging application. Figure 3 shows that CoSaMP and GAMP cannot recover the tomoSAR image, but OMP and RrMP all work well. On the one hand, from the relative error comparison in Table 1, we can see that the performance of RrMP-L4/L6/L8 is almost the same as that of OMP; on the other hand, RrMP runs much faster than OMP, CoSaMP and GAMP, and the CPU time of RrMP-L4/L6/L8 is only 48.14%, 41.26%

[3] http://eeweb.poly.edu/ srangan/index.html



PFA reconstructed image CS reconstructed images

FIGURE 3. TomoSAR imaging comparison

TABLE 1. Relative error and CPU time

	OMP	CoSaMP	GAMP	RrMP-L4	RrMP-L6	RrMP-L8
Relative error	0.0188	$4.4101e+04$	125.9500	0.0230	0.0326	0.0350
CPU time	31.8303	24.6734	50.5852	15.3234	13.1325	14.7225

and 46.25% of OMP. Taken together, we find that our proposed RrMP algorithm has a remarkable speed advantage and comparable performance with OMP algorithm.

5. **Conclusion.** In this paper we propose RrMP algorithm to solve the compressed sampling tomoSAR imaging problem. This algorithm can reduce much CPU time by pursuing multiple columns of the sensing matrix at each iteration, and improve the reconstruction performance by randomly bisecting and pruning/merging these columns. It has ability to do parallel computing in three steps which is helpful to accelerate speed further. Simulation and real world tomoSAR imaging experiments confirm the effectiveness of RrMP algorithm. In future work we plan to explore the convergence proof of this algorithm.

Acknowledgment. This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant U1533125.

REFERENCES

[1] L. Potter, E. Ertin, J. Parker and M. Cetin, Sparsity and compressed sensing in radar imaging, *Proc. of the IEEE*, vol.98, no.6, pp.1006-1020, 2010.

[2] D. Xu, L. Du, H. Liu, P. Wang, J. Yan, Y. Cong and X. Han, Compressive sensing of stepped-frequency radar based on transfer learning, *IEEE Trans. Signal Processing*, vol.63, no.12, pp.3076-3087, 2015.

[3] X. Zhang, T. Bai, H. Meng and J. Chen, Compressive sensing-based ISAR imaging via the combination of the sparsity and nonlocal total variation, *IEEE Geoscience and Remote Sensing Letters*, vol.11, no.5, pp.990-994, 2014.

[4] M. Cetin, I. Stojanovic, N. Onhon, K. Varshney, S. Samadi, W. Karl and A. Willsky, Sparsity-driven synthetic aperture radar imaging: Reconstruction, auto focusing, moving targets, and compressed sensing, *IEEE Signal Processing Magazine*, vol.31, no.4, pp.27-40, 2014.

[5] S. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries, *IEEE Trans. Signal Processing*, vol.41, no.12, pp.3397-3415, 1993.

[6] D. Needell and J. A. Tropp, Cosamp: Iterative signal recovery from incomplete and inaccurate samples, *Appl. Comput. Harmonic Anal.*, vol.26, no.3, pp.301-321, 2009.

[7] E. Candes, J. Romberg and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Inf. Theory*, vol.52, no.2, pp.489-509, 2006.

[8] D. L. Donoho, A. Maleki and A. Motanari, Message passing algorithms for compressed sensing, *Proc. of Natl. Acad. Sci.*, vol.106, pp.18914-18919, 2009.

[9] M. Cheney and B. Borden, Imaging moving targets from scattered waves, *Inverse Prob.*, vol.24, no.3, pp.35005-35026, 2008.

[10] X. X. Zhu and R. Bamler, Superresolving SAR tomography for multidimensional imaging of urban areas: Compressive sensing-based TomoSAR inversion, *IEEE Signal Processing Magazine*, vol.31, no.4, pp.51-58, 2014.

[11] W. G. Carrara, R. S. Goodman and R. M. Majewski, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*, Artech House, 1995.

[12] X. C. Cong and J. B. Liu, Millimeter-wave spotlight circular synthetic aperture radar (SCSAR) imaging for foreign object debris on airport runway, *IEEE the 12th International Conference on ICSP*, pp.1968-1972, 2014.