SAR Imaging via Efficient Implementations of Sparse ML Approaches

Glentis, George-Othan; Zhao, Kexin; Jakobsson, Andreas; Abeida, Habti; Li, Jian

Published in:
Signal Processing

DOI:
10.1016/j.sigpro.2013.08.003

2014

Signal Processing 95 (2014) 15–26

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

SAR imaging via efficient implementations of sparse ML approaches ☆

George-Othon Glentis [a,*], Kexin Zhao [b], Andreas Jakobsson [c], Habti Abeida [d], Jian Li [b]

[a] Department of Science and Technology of Telecommunications, University of Peloponnese, Tripolis 22100 Greece
[b] Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6130, USA
[c] Department of Mathematical Statistics, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden
[d] Department of Electrical Engineering, University of Taif, Al-Haweiah 21974, Saudi Arabia

ARTICLE INFO

Article history:
Received 24 April 2013
Received in revised form
7 August 2013
Accepted 12 August 2013
Available online 22 August 2013

Keywords:
Synthetic aperture radar imaging
Non-parametric high resolution spectral analysis
Sparse estimators
Efficient algorithms

ABSTRACT

High-resolution spectral estimation techniques are of notable interest for synthetic aperture radar (SAR) imaging. Several sparse estimation techniques have been shown to provide significant performance gains as compared to conventional approaches. We consider efficient implementation of the recent iterative sparse maximum likelihood-based approaches (SMLAs). Furthermore, we present approximative fast SMLA formulation using the Quasi-Newton approach, as well as consider hybrid SMLA-MAP algorithms. The effectiveness of the discussed techniques is illustrated using numerical and experimental examples.

1. Introduction

The development of high-resolution two-dimensional (2-D) spectral estimation algorithms is of notable interest in forming accurate and reliable synthetic aperture radar (SAR) images, and the topic has, as a result, attracted much interest during recent years. Typically, SAR images are formed using periodogram-based estimators, thereby suffering from the well-known limitations in resolution and high leakage levels. Various forms of data-adaptive and often non-parametric approaches can enhance the resolution while reducing these shortcomings [1–5]. Generally, these methods require large data sets to offer reliable estimates of the second-order statistics, a requirement that is hard to satisfy in practice. To alleviate this problem, recent work has examined various forms of sparse estimation techniques, such as the sparse learning via iterative minimization (SLIM) method [6], the iterative adaptive approach (IAA) [7], and more recently a set of iterative sparse maximum likelihood-based approaches (SMLA) [8,9]. This class of methods have been found to offer significant performance improvements as compared to the traditional methods not exploiting the sparsity of the signal, generally providing reliable high-resolution estimates with excellent side lobe suppression [10–14]. Yet all these approaches suffer from being computationally cumbersome, which has resulted in a series of recent works focusing on formulating computationally efficient implementations for the SLIM and IAA estimates [15–18,14]. In this work, we continue

☆ This work was supported in part by the Swedish Research Council, Carl Trygger's foundation, National Science Foundation under Grant No. CCF-1218388, and the Department of Defense. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

* Corresponding author. Tel.: +30 2710372202; fax: +30 2710372161.
E-mail addresses: gglentis@uop.gr (G.-O. Glentis), kexinzhao@ufl.edu (K. Zhao), aj@maths.lth.se (A. Jakobsson), abeida3@yahoo.fr (H. Abeida), li@dsp.ufl.edu (J. Li).

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.08.003

this development, building on the recently developed efficient algorithms and extending them to formulate implementations for the additional matrix computations also required to form efficient SMLA implementations. As our interest is primarily in SAR imaging, we here focus on the 2-D formulations of these algorithms, noting that the 1-D case will be a special case of these, whereas higher dimensional estimates can be formed similarly, although doing so requires additional mathematical manipulations. Furthermore, we extend the recent work on forming also approximative implementations using the Quasi-Newton (QN) approach [18,19], presenting approximative SMLA formulations, as well as considering the hybrid (maximum a posterior) SMLA-MAP algorithms [8,9]. The remainder of this paper is organized as follows: in the interest of completeness and to introduce the necessary notation, we begin with briefly reviewing the SMLA algorithms in the following section. Then, in Section 3, we proceed to introduce efficient implementations of these algorithms. In Section 4, we examine the performance of the discussed estimators and the proposed implementations using both numerical and experimental 1-D spectral estimation and 2-D synthetic aperture radar imaging examples. Finally, Section 5 contains our conclusions.

2. A brief review of the SMLAs

Let $y(n_1,n_2)$, $n_1=0,1,\ldots,N_1$, $n_2=0,1,\ldots,N_2$, and denote a uniformly sampled 2-D sequence of observations for which one wishes to compute a power spectral estimate. Let

$$\mathbf{Y}_{N_1,N_2}=[\mathbf{y}_{N_1}(0)\ \ldots\ \mathbf{y}_{N_1}(N_2-1)], \tag{1}$$

$$\mathbf{y}_{N_1}(n_2)=[y(0,n_2)\ \ldots\ y(N_1-1,n_2)]^T \tag{2}$$

where $n_2=0,1,\ldots,N_2-1$. Moreover, let

$$\mathbf{y}_{N_1N_2}=\mathrm{vec}\{\mathbf{Y}_{N_1,N_2}\} \tag{3}$$

$$\mathbf{Y}_{N_1,N_2}=\mathrm{mat}\{\mathbf{y}_{N_1N_2}\} \tag{4}$$

where $\mathrm{vec}(\cdot)$ denotes column-wise vectorization, and $\mathrm{mat}(\cdot)$ the inverse operation, recreating the matrix from the vectorized matrix, and define the 2-D frequency vector

$$\mathbf{f}_{N_1N_2}(\omega_1,\omega_2)\triangleq\mathbf{f}_{N_2}(\omega_2)\otimes\mathbf{f}_{N_1}(\omega_1), \tag{5}$$

where $\otimes$ denotes the Kronecker product, and

$$\mathbf{f}_N(\omega)\triangleq[1\ \ e^{j\omega}\ \ \ldots\ \ e^{j(N-1)\omega}]^T.$$

Without loss of generality, we here restrict our attention to the case when the 2-D frequency vector in (5) is defined over a uniformly spaced grid of frequencies, such that

$$(\omega_{k_1},\omega_{k_2})\triangleq(2\pi k_1/K_1,2\pi k_2/K_2) \tag{6}$$

where $k_1=0,1,\ldots K_1-1$ and $k_2=0,1,\ldots K_2-1$, with $K_i>N_i+1$, i=1,2, and where typically $K_i\ll N_i$. To simplify notation, define $\mathbf{f}_{k_1,k_2}\triangleq\mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2})$, and denote the power of $y(n_1,n_2)$, at frequency $(\omega_{k_1},\omega_{k_2})$, with $\alpha_{k_1,k_2}\triangleq|\mathcal{Y}(\omega_{k_1},\omega_{k_2})|^2$, where $\mathcal{Y}(\omega_{k_1},\omega_{k_2})$ is the corresponding complex-valued spectral amplitude. Modeling the signal as consisting of a sparse signal corrupted by an additive Gaussian noise, an estimate of the complex valued covariance matrix of $\mathbf{y}_{N_1N_2}$ is then obtained as

$$\mathbf{R}_{N_1N_2}\triangleq\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\alpha_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H+\mathbf{\Sigma}_{N_1N_2} \tag{7}$$

where $\mathbf{\Sigma}_{N_1N_2}\triangleq\sigma^2\mathbf{I}_{N_1N_2}$ denotes the covariance matrix of the noise term with (unknown) variance σ^2. Then, for all the frequencies of interest, SMLA is formed by iteratively computing an estimate of the spectral power, α_{k_1,k_2}, for $k_1=0,1,\ldots K_1-1$, $k_2=0,1,\ldots K_2-1$, as well as the signal covariance matrix, $\mathbf{R}_{N_1N_2}$, and the noise covariance matrix, $\mathbf{\Sigma}_{N_1N_2}$, until practical convergence. As shown in [8,9], the family of SMLAs consists of four separate algorithms, termed SMLA-ℓ, for $\ell=0,\ldots,3$, formed as

SMLA-0:

$$\alpha_{k_1,k_2}=\alpha_{k_1,k_2}^2|\mathbf{f}_{k_1,k_2}^H\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2 \tag{8}$$

$$\mathbf{R}_{N_1N_2}=\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\alpha_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H+\sigma^2\mathbf{I}_{N_1N_2} \tag{9}$$

$$\sigma^2=\frac{|\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2}{\mathrm{Tr}(\mathbf{R}_{N_1N_2}^{-2})} \tag{10}$$

SMLA-1:

$$\alpha_{k_1,k_2}=\left|\frac{\mathbf{f}_{k_1,k_2}^H\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}}{\mathbf{f}_{k_1,k_2}^H\mathbf{R}_{N_1N_2}^{-1}\mathbf{f}_{k_1,k_2}}\right|^2 \tag{11}$$

$$\mathbf{R}_{N_1N_2}=\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\alpha_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H+\sigma^2\mathbf{I}_{N_1N_2} \tag{12}$$

$$\sigma^2=\frac{|\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2}{\mathrm{Tr}(\mathbf{R}_{N_1N_2}^{-2})} \tag{13}$$

SMLA-2:

$$\alpha_{k_1,k_2}=\alpha_{k_1,k_2}\frac{|\mathbf{f}_{k_1,k_2}^H\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2}{\mathbf{f}_{k_1,k_2}^H\mathbf{R}_{N_1N_2}^{-1}\mathbf{f}_{k_1,k_2}} \tag{14}$$

$$\mathbf{R}_{N_1N_2}=\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\alpha_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H+\sigma^2\mathbf{I}_{N_1N_2} \tag{15}$$

$$\sigma^2=\frac{|\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2}{\mathrm{Tr}(\mathbf{R}_{N_1N_2}^{-2})} \tag{16}$$

SMLA-3:

$$\alpha_{k_1,k_2}=\beta_{k_1,k_2}^2|\mathbf{f}_{k_1,k_2}^H\mathbf{P}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2 \tag{17}$$

$$\beta_{k_1,k_2}=\frac{1}{\mathbf{f}_{k_1,k_2}^H\mathbf{R}_{N_1N_2}^{-1}\mathbf{f}_{k_1,k_2}} \tag{18}$$

$$\mathbf{R}_{N_1N_2}=\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\alpha_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H+\sigma^2\mathbf{I}_{N_1N_2} \tag{19}$$

$$\mathbf{P}_{N_1N_2}=\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\beta_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H+\sigma^2\mathbf{I}_{N_1N_2} \tag{20}$$

$$\sigma^2=\frac{|\mathbf{R}_{N_1N_2}^{-1}\mathbf{y}_{N_1N_2}|^2}{\mathrm{Tr}(\mathbf{R}_{N_1N_2}^{-2})} \tag{21}$$

wherein, for all the four approaches, α_{k_1,k_2} is initialized using the 2-D Discrete Fourier Transform (DFT) coefficients, with, usually, only 10–15 iterations being required for convergence. As noted, direct implementations of the SMLAs are computationally intensive, requiring roughly

$$\mathcal{C}^{SMLA-0} = m(2N_1^3N_2^3 + N_1^2N_2^2K_1K_2) \tag{22}$$

$$\mathcal{C}^{SMLA-1} = m(2N_1^3N_2^3 + 2N_1^2N_2^2K_1K_2) \tag{23}$$

$$\mathcal{C}^{SMLA-2} = m(2N_1^3N_2^3 + 2N_1^2N_2^2K_1K_2) \tag{24}$$

$$\mathcal{C}^{SMLA-3} = m(3N_1^3N_2^3 + 4N_1^2N_2^2K_1K_2) \tag{25}$$

operations, where m denotes the number of SMLA iterations performed. As shown in the following, these figures can be drastically reduced by taking into account the structural properties of the covariance matrices and the pertinent trigonometric polynomials involved in each of the SMLA algorithms.

3. Fast implementations of the SMLAs

Compared to the earlier efficient implementations introduced for the SLIM and IAA algorithms [14–18,20], it can be noted that SMLAs exhibit several similarities to these, and can without modification exploit the previously formulated efficient Gohberg–Semencul (G–S) type factorizations of the estimated inverse covariance matrix, as well as for the required products of inverses. The part lacking in these aforementioned implementations is the efficient computation of the factor $\mathrm{Tr}(\mathbf{R}_{N_1N_2}^{-2})$. Fortunately, using the displacement representation of this expression, G–S factorizations of this factor may also be formulated, thereby enabling it to be efficiently computed using the Fast Fourier Transform (FFT).

Recall that, given a matrix $\mathbf{A}_{N_1N_2} \in \mathcal{C}^{N_1N_2 \times N_2N_2}$, the displacement of $\mathbf{A}_{N_1N_2}$ with respect to $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}_{N_1N_2}^T$ is defined as [21–24]

$$\nabla_{Z,Z^T}(\mathbf{A}_{N_1N_2}) \triangleq \mathbf{A}_{N_1N_2} - \mathbf{Z}_{N_1N_2}\mathbf{A}_{N_1N_2}\mathbf{Z}_{N_1N_2}^T \tag{26}$$

where $\mathbf{Z}_{N_1N_2} \triangleq \mathbf{Z}_{N_2} \otimes \mathbf{I}_{N_1}$, with

$$\mathbf{Z}_{N_2} \triangleq \begin{bmatrix} \mathbf{0}_{N_2-1}^T & 0 \\ \mathbf{I}_{N_2-1} & \mathbf{0}_{N_2-1} \end{bmatrix} \tag{27}$$

whereas $\mathbf{I}_{N_1}$ and $\mathbf{I}_{N_2-1}$ are identity matrices of appropriate dimensions. Clearly, $\mathbf{Z}_{N1N_2}^{N_2} = \mathbf{0}$. Suppose there exist integers ρ and $\gamma_i \in \{-1, 1\}$, $i = 1, 2\ldots\rho$, such that

$$\nabla_{Z,Z^T}(\mathbf{A}_{N_1N_2}) = \sum_{i=1}^{\rho} \gamma_i \mathbf{t}_{N_1N_2}^i \mathbf{s}_{N_1N_2}^{iH} \tag{28}$$

Then, $\mathbf{A}_{N_1N_2}$ can be expressed using a G–S factorization as

$$\mathbf{A}_{N_1N_2} = \sum_{i=1}^{\rho} \gamma_i \mathcal{L}(\mathbf{t}_{N_1N_2}^i)\mathcal{L}(\mathbf{s}_{N_1N_2}^i)^H \tag{29}$$

where, given $\mathbf{x} \in \mathcal{C}^{N_1N_2 \times 1}$, $\mathcal{L}(\mathbf{x})$ is the Krylov matrix of dimensions $(N_1N_2) \times N_2$, defined as

$$\mathcal{L}(\mathbf{x}) \triangleq (\mathbf{x}\mathbf{Z}_{N_1N_2}\mathbf{x}\ldots\mathbf{Z}_{N_1N_2}^{N_2-1}\mathbf{x}). \tag{30}$$

Clearly, $\mathcal{L}(\mathbf{x})$ is a block lower Toeplitz matrix of block dimensions $N_2 \times N_2$ having block entries of size $N_1 \times 1$ each. Let

$$\mathbf{T}_{N_1N_2,\rho} = [\mathbf{t}_{N_1N_2}^1 \ \cdots \ \mathbf{t}_{N_1N_2}^{\rho}] \tag{31}$$

$$\mathbf{\Gamma}_{N_1N_2,\rho} = [\mathbf{s}_{N_1N_2}^1 \ \cdots \ \mathbf{s}_{N_1N_2}^{\rho}] \tag{32}$$

$$\mathbf{\Gamma}_{\rho} = \mathrm{diag}(\gamma_1\ldots\gamma_\rho) \tag{33}$$

Then, the triplet $(\mathbf{T}_{N_1N_2,\rho}, \mathbf{S}_{N_1N_2,\rho}, \mathbf{\Gamma}_\rho)$ is called the displacement representation of $\mathbf{A}_{N_1N_2}$ with respect to $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}_{N_1N_2}^T$. Thus, given the generator vectors $\mathbf{t}_{N_1N_2}^i$, $\mathbf{s}_{N_1N_2}^i$, and scalars γ_i, $i = 1, 2\ldots\rho$, the matrix $\mathbf{A}_{N_1N_2}$ can be reconstructed using the G–S factorization in (29).

Lemma 1. *Given a Hermitian matrix $\mathbf{A}_{N_1N_2} \in \mathcal{C}^{(N_1N_2)\times(N_1N_2)}$, specified by the displacement representation $(\mathbf{T}_{N_1N_2,\rho}, \mathbf{T}_{N_1N_2,\rho}, \mathbf{\Gamma}_\rho)$, and a vector $\mathbf{a}_{N_1N_2} \in \mathcal{C}^{(N_1N_2)\times 1}$, the matrix vector product $\mathbf{A}_{N_1N_2}\mathbf{a}_{N_1N_2}$ can be computed using (29) at a cost of $(3\rho+2)N_1\phi(2N_2) + 2\rho N_1N_2$ operations, where $\phi(N)$ denotes the operations required for the computation of the 1-D FFT or inverse FFT (IFFT).*

Lemma 2. *Given $\mathbf{A}_{N_1N_2}$, the coefficients of the trigonometric polynomial defined by*

$$\varphi(\omega_1,\omega_2) \triangleq \mathbf{f}_{N_1N_2}^H(\omega_1,\omega_2)\mathbf{A}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_1,\omega_2) = \sum_{\kappa_1=-N_1+1}^{N_1-1}\sum_{\kappa_2=-N_2+1}^{N_2-1} \varphi_{\kappa_1,\kappa_2} e^{-j(\omega_1\kappa_1+\omega_2\kappa_2}$$

can be computed at a cost of $2(\rho+1)\phi(2N_1, 2N_2)$ operations, where $\phi(N, M)$ denotes the number of operations required for the computation of the2-D FFT or IFFT.

The proofs of these lemmas are given in [22,25,26].

Lemma 3. *The trace of $\mathbf{A}_{N_1N_2}$ can be estimated as*

$$\mathrm{Tr}(\mathbf{A}_{N_1N_2}) = \boldsymbol{\delta}_{N_1N_2}^T\left(\sum_{i=1}^{\rho}\gamma_i\mathbf{t}_{N_1N_2}^i \odot \mathbf{s}_{N_1N_2}^{*}\right) \tag{34}$$

where

$$\boldsymbol{\delta}_{N_1N_2} \triangleq \begin{bmatrix} N_2 \\ N_2-1 \\ \vdots \\ 1 \end{bmatrix} \otimes \mathbf{1}_{N_1}$$

with $\mathbf{1}_{N_1}$ denoting a $N_1 \times 1$ vector with all elements equal to one, and with $\odot$ denoting the Hadamard (elementwise) product of two vectors.

Proof. Using the G–S representation in (29), the diagonal of $\mathbf{A}_{N_1N_2}$ can be computed as

$$\mathrm{Diag}(\mathbf{A}_{N_1N_2}) = \sum_{i=1}^{\rho}\gamma_i\mathbf{Z}_{N_1N_2}^{i-1}(\mathbf{t}_{N_1N_2}^i \odot \mathbf{s}_{N_1N_2}^{i*}), \tag{35}$$

which, using the structure of $\mathbf{Z}_{N_1N_2}$ and the fact that the trace of matrix equals to the sum of its diagonal elements, results in (34). □

It is worth noting that [27] can be seen as a special case of Lemma 3.

Further, it can be noted that the covariance matrices involved in the SMLAs all share the following generic form:

$$\mathbf{R}_{N_1N_2} = \mathbf{R}_{N_1N_2}^o + \sigma^2\mathbf{I}_{N_1N_2} \tag{36}$$

$$\mathbf{R}_{N_1N_2}^o = \sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\alpha_{k_1,k_2}\mathbf{f}_{k_1,k_2}\mathbf{f}_{k_1,k_2}^H \tag{37}$$

implying that $\mathbf{R}^o_{N_1N_2}$ has a Toeplitz block Toeplitz (TBT) structure (see also [16,17]), and as an immediate consequence, that $\mathbf{R}_{N_1N_2}$ also has a TBT structure of the form

$$\mathbf{R}_{N_1N_2}=\begin{bmatrix}\mathbf{R}^0_{N_1} & \mathbf{R}^{1H}_{N_1} & \cdots & \mathbf{R}^{(N_2-1)H}_{N_1}\\ \mathbf{R}^1_{N_1} & \mathbf{R}^0_{N_1} & \cdots & \mathbf{R}^{(N_2-2)H}_{N_1}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{R}^{N_2-1}_{N_1} & \mathbf{R}^{N_2-2}_{N_1} & \cdots & \mathbf{R}^0_{N_1}\end{bmatrix} \tag{38}$$

where the matrix entries $\mathbf{R}^{\ell}_{N_1}$, for $\ell=0,1,\ldots N_2-1$, are Toeplitz matrices of size $N_1\times N_1$. For completeness and to introduce the further needed notation, we now proceed to form the displacement of $\mathbf{R}^{-1}_{N_1N_2}$. As shown in [15–17], this can be done by noting that $\mathbf{R}^o_{N_1N_2}$ may be extracted from a circulant block circulant (CBC) matrix of higher dimensions as (see also [25])

$$\mathbf{C}_{K_1K_2}=\mathbf{W}^H_{K_1K_2}\mathbf{D}_{K_1K_2}\mathbf{W}_{K_1K_2}=\mathbf{S}_{K_1K_2}\begin{bmatrix}\mathbf{R}^o_{N_1N_2} & \times\\ \times & \times\end{bmatrix}\mathbf{S}^T_{K_1K_2}, \tag{39}$$

where $\mathbf{D}_{K_1K_2}=\mathrm{diag}\{\alpha_{0,0},\ldots,\alpha_{K_1-1,\,K_2-1}\}$, with $\mathbf{W}_{K_1K_2}$ denoting the 2-D DFT matrix, $\mathbf{S}_{K_1K_2}$ being a suitable permutation matrix, and symbol $\times$ denoting terms of no relevance. Given the TBT structure of (38), $\mathbf{R}_{N_1N_2}$ may be partitioned as

$$\mathbf{R}_{N_1N_2}=\begin{bmatrix}\mathbf{R}_{N_1(N_2-1)} & \mathcal{R}^b_{N_2-1}\\ \mathcal{R}^{bH}_{N_2-1} & \mathbf{R}^0_{N_1}\end{bmatrix}=\begin{bmatrix}\mathbf{R}^0_{N_1} & \mathcal{R}^{fH}_{N_2-1}\\ \mathcal{R}^f_{N_2-1} & \mathbf{R}_{N_1(N_2-1)}\end{bmatrix} \tag{40}$$

where $\mathcal{R}^b_{N_2-1}$ and $\mathcal{R}^f_{N_2-1}$ denote block matrices of dimensions $N_1(N_2-1)\times N_1$. Applying the matrix inversion lemma for partitioned matrices to (40) yields (see, e.g., [28])

$$\mathbf{R}^{-1}_{N_1N_2}=\begin{bmatrix}\mathbf{R}^{-1}_{N_1(N_2-1)} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix}+\overline{\mathcal{B}}_{N_2}\overline{\mathcal{B}}^H_{N_2}=\begin{bmatrix}0 & \mathbf{0}\\ \mathbf{0}^T & \mathbf{R}^{-1}_{N_1(N_2-1)}\end{bmatrix}+\overline{\mathcal{A}}_{N_2}\overline{\mathcal{A}}^H_{N_2} \tag{41}$$

where $\overline{\mathcal{B}}_{N_2}$ and $\overline{\mathcal{A}}_{N_2}$ are block matrices of dimensions $N_1N_2\times N_1$ defined by

$$\overline{\mathcal{B}}_{N_2}=\begin{bmatrix}\mathcal{B}_{N_2-1}\\ \mathbf{I}_{N_1}\end{bmatrix}\mathbf{A}^{-b/2}_{N_1} \tag{42}$$

$$\overline{\mathcal{A}}_{N_2}=\begin{bmatrix}\mathbf{I}_{N_1}\\ \mathcal{A}_{N_2-1}\end{bmatrix}\mathbf{A}^{-f/2}_{N_1} \tag{43}$$

$$\mathcal{B}_{N_2-1}=-\mathbf{R}^{-1}_{N_1(N_2-1)}\mathcal{R}^b_{N_2-1} \tag{44}$$

$$\mathcal{A}_{N_2-1}=-\mathbf{R}^{-1}_{N_1(N_2-1)}\mathcal{R}^f_{N_2-1} \tag{45}$$

$$\mathbf{A}^b_{N_1}=\mathbf{R}^0_{N_1}+\mathcal{R}^{bH}_{N_2-1}\mathcal{B}_{N_2-1} \tag{46}$$

$$\mathbf{A}^f_{N_1}=\mathbf{R}^0_{N_1}+\mathcal{R}^{fH}_{N_2-1}\mathcal{A}_{N_2-1} \tag{47}$$

with $\mathbf{A}^{b/2}_{N_1}$ and $\mathbf{A}^{f/2}_{N_1}$ denoting the Cholesky factors of $\mathbf{A}^b_{N_1}$ and $\mathbf{A}^f_{N_1}$, respectively. Using (41), the displacement of $\mathbf{R}^{-1}_{N_1N_2}$ with respect to $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}^T_{N_1N_2}$ is estimated as

$$\nabla_{Z,Z^T}(\mathbf{R}^{-1}_{N_1N_2})=\overline{\mathcal{A}}_{N_2}\overline{\mathcal{A}}^H_{N_2}-\mathbf{Z}_{N_1N_2}\overline{\mathcal{B}}_{N_2}\overline{\mathcal{B}}^H_{N_2}\mathbf{Z}^T_{N_1N_2} \tag{48}$$

implying that a displacement representation of $\mathbf{R}^{-1}_{N_1N_2}$ has the form $(\mathbf{T}_{N_1N_2,2N_2},\mathbf{T}_{N_1N_2,2N_2},\mathbf{\Gamma}_{2N_2})$, where

$$\mathbf{T}_{N_1N_2,2N_2}\triangleq[\overline{\mathcal{A}}_{N_2}\mathbf{Z}_{N_1N_2}\overline{\mathcal{B}}_{N_2}] \tag{49}$$

$$\mathbf{\Gamma}_{2N_2}\triangleq\mathrm{diag}\{\mathbf{I}_{N_2},-\mathbf{I}_{N_2}\} \tag{50}$$

In this case the displacement rank of the representation equals $\rho(\mathbf{R}^{-1})=2N_2$. With these results, we are now ready to introduce the displacement of $\mathbf{R}^{-2}_{N_1N_2}$, noting that this particular form is a special case of more general matrix products studied recently in the context of the efficient implementation of 1-D and 2-D Magnitude Squared Coherence estimators [29–31]. Using (41),

$$\mathbf{R}^{-2}_{N_1N_2}=\begin{bmatrix}\mathbf{R}^{-2}_{N_1(N_2-1)} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix}+\overline{\mathcal{B}}_{N_2}\overline{\mathcal{C}}^H_{N_2}+\overline{\mathcal{D}}_{N_2}\overline{\mathcal{B}}^H_{N_2} \tag{51}$$

where

$$\overline{\mathcal{D}}_{N_2}=\mathbf{R}^{-1}_{N_1N_2}\overline{\mathcal{B}}_{N_2} \tag{52}$$

$$\overline{\mathcal{C}}_{N_2}=\overline{\mathcal{D}}_{N_2}-\overline{\mathcal{B}}_{N_2}\overline{\mathcal{B}}^H_{N_2}\overline{\mathcal{B}}_{N_2} \tag{53}$$

and, using (41),

$$\mathbf{R}^{-2}_{N_1N_2}=\begin{bmatrix}0 & \mathbf{0}\\ \mathbf{0}^T & \mathbf{R}^{-2}_{N_1(N_2-1)}\end{bmatrix}+\overline{\mathcal{A}}_{N_2}\overline{\mathcal{E}}^H_{N_2}+\overline{\mathcal{F}}_{N_2}\overline{\mathcal{A}}^H_{N_2} \tag{54}$$

where

$$\overline{\mathcal{F}}_{N_2}=\mathbf{R}^{-1}_{N_1N_2}\overline{\mathcal{A}}_{N_2} \tag{55}$$

$$\overline{\mathcal{E}}_{N_2}=\overline{\mathcal{F}}_{N_2}-\overline{\mathcal{A}}_{N_2}\overline{\mathcal{A}}^H_{N_2}\overline{\mathcal{A}}_{N_2} \tag{56}$$

Using (51) and (54) yields

$$\begin{aligned}\nabla_{Z,Z^T}(\mathbf{R}^{-2}_{N_1N_2})=&\,\overline{\mathcal{A}}_{N_2}\overline{\mathcal{E}}^H_{N_2}+\overline{\mathcal{F}}_{N_2}\overline{\mathcal{A}}^H_{N_2}\\ &-\mathbf{Z}_{N_1N_2}\overline{\mathcal{B}}_{N_2}\overline{\mathcal{C}}^H_{N_2}\mathbf{Z}^T_{N_1N_2}-\mathbf{Z}_{N_1N_2}\overline{\mathcal{D}}_{N_2}\overline{\mathcal{B}}^H_{N_2}\mathbf{Z}^T_{N_1N_2}\end{aligned} \tag{57}$$

which indicates that a displacement representation of $\mathbf{R}^{-2}_{N_1N_2}$ with respect to $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}^T_{N_1N_2}$ has the form

$$(\mathbf{T}_{N_1N_2,4N_2},\mathbf{S}_{N_1N_2,4N_2},\mathbf{\Gamma}_{4N_2}) \tag{58}$$

where

$$\begin{aligned}\mathbf{T}_{N_1N_2,4N_2}&\triangleq[\overline{\mathcal{A}}_{N_2}\overline{\mathcal{F}}_{N_2}\mathbf{Z}_{N_1N_2}\overline{\mathcal{B}}_{N_2}\mathbf{Z}_{N_1N_2}\overline{\mathcal{D}}_{N_2}]\\ \mathbf{S}_{N_1N_2,4N_2}&\triangleq[\overline{\mathcal{E}}_{N_2}\overline{\mathcal{A}}_{N_2}\mathbf{Z}_{N_1N_2}\overline{\mathcal{C}}_{N_2}\mathbf{Z}_{N_1N_2}\overline{\mathcal{B}}_{N_2}]\\ \mathbf{\Gamma}_{4N_2}&\triangleq\mathrm{diag}\{\mathbf{I}_{N_2},\mathbf{I}_{N_2},-\mathbf{I}_{N_2},-\mathbf{I}_{N_2}\}\end{aligned}$$

implying that the displacement rank of the representation equals $\rho(\mathbf{R}^{-2})=4N_2$.

It is worth remarking that the displacement representation of $\mathbf{R}^{-1}_{N_1N_2}$ can be computed using the celebrated Levinson–Whittle–Wiggins–Robinson (LWWR) algorithm (see, e.g., [32]), where (44) and (45) may be interpreted as multichannel backward and forward predictors, (46) and (47) as the associated prediction error power matrices, and (42) and (43) as the power normalized backward and forward predictors counterparts respectively. The LWWR algorithm requires approximately $C^{LWWR}=1.5N_1^3N_2^2$ operations. This figure may be reduced to $C^{LWWR1}=N_1^3N_2^2$, if the persymmetric property of the TBT matrix $\mathbf{R}_{N_1N_2}$ is taken into account, according to which

$$\mathbf{J}_{N_1N_2}\mathbf{R}_{N_1N_2}\mathbf{J}_{N_1N_2}=\mathbf{R}^T_{N_1N_2} \tag{59}$$

where the block exchange matrix, $\mathbf{J}_{N_1N_2}$, is defined as a block anti-diagonal matrix with the exchange matrix $\mathbf{J}_{N_1}$ along the (block) anti-diagonal, resulting in [33]

$$\mathcal{A}_{N_2-1} = \mathbf{J}_{N_1(N_2-1)}\mathcal{B}^*_{N_2-1}\mathbf{J}_{N_1} \tag{60}$$

$$\mathbf{A}^f_{N_1} = \mathbf{J}_{N_1}\mathbf{A}^{bT}_{N_1}\mathbf{J}_{N_1}. \tag{61}$$

As a further remark, we note that given the displacement representation of $\mathbf{R}^{-1}_{N_1N_2}$, the displacement representation of $\mathbf{R}^{-2}_{N_1N_2}$ can efficiently be computed at a cost of approximately $12N_1^2N_2\phi(2N_2)$ operations since (52) and (53) are TBT vector products and they can be computed using Lemma 1. Extra computational savings can be achieved, cutting the computational cost to half, by noting that using (42) and (55) may be rearranged as

$$\overline{\mathcal{F}}_{N_2} = \tilde{\mathcal{F}}_{N_2}\mathbf{A}^{-f/2}_{N_1} \tag{62}$$

$$\tilde{\mathcal{F}}_{N_2} = \mathbf{R}^{-1}_{N_1N_2}\begin{bmatrix}\mathbf{I}_{N_1}\\ \mathcal{A}_{N_2-1}\end{bmatrix} \tag{63}$$

which, in turn, allows (52) to be expressed equivalently as

$$\overline{\mathcal{D}}_{N_2} = \mathbf{J}_{N_1N_2}\tilde{\mathcal{F}}^*_{N_2}\mathbf{J}_{N_2}\mathbf{A}^{-b/2}_{N_1} \tag{64}$$

In summary, to form the SMLA-ℓ estimates, the pertinent covariance matrices $\mathbf{R}_{N_1N_2}$ given in (9), (12), (15) and (18), as well as the matrix $\mathbf{P}_{N_1N_2}$, given in (19), are computed using the block Toeplitz block to block circulant block embedding approach described by (39), where the first block column is computed using the 2-D FFT, at a cost of $\phi(K_1,K_2)$ operations. These matrices are then used by the LWWR algorithm for the computation of the displacement representations of $\mathbf{R}^{-1}_{N_1N_2}$ and $\mathbf{P}_{N_1N_2}$, which are subsequently utilized for

- the computation $\mathbf{y}^R_{N_1N_2} \triangleq \mathbf{R}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}$ and $\mathbf{y}^P_{N_1N_2} \triangleq \mathbf{P}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}$, appearing in (8), (10), (11), (13), (14), (16), (17), and (20), using Lemma 1,
- the computation of the displacement representation of $\mathbf{R}^{-2}_{N_1N_2}$ using (52), (53), (55), and (56), which is subsequently utilized for the computation of $\mathrm{Tr}(\mathbf{R}^{-2}_{N_1N_1})$, that appears in (10), (13), (16), and (21), and finally,
- the computation of the coefficients of the 2-D trigonometric polynomials $\varphi(\omega_1\omega_2) \triangleq \mathbf{f}^H_{N_1N_2}(\omega_1,\omega_2)\mathbf{R}^{-1}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_1,\omega_2)$ that appear in the denominator of (11), (14), and (18), using Lemma 2.

Then, the estimated power spectra formed by (8), (11), (14), (17), and (18) can be evaluated on the uniformly spaced grid of 2-D frequencies by applying the 2-D FFT either on the vectors $\mathbf{y}^R_{N_1N_2}$ and $\mathbf{y}^P_{N_1N_2}$, or on the coefficients of the 2-D trigonometric polynomial $\varphi(\omega_1,\omega_2)$, associated to each method, at a cost of $\phi(K_1,K_2)$ operations, where K_1 and K_2 denote the size of the grid of the 2-D frequencies of interest. Summarizing, the computational complexity of the proposed efficient implementation of the SMLA algorithms is approximately, keeping the dominant factors,

$$C^{FSMLA} \approx m(c_1N_1^3N_2^2 + c_2N_1^2\phi(2N_2) + c_3N_1\phi(2N_1,2N_2) + c_4\phi(K_1,K_2)) \tag{65}$$

where variables c_1, c_2, c_3, and c_4 depend upon the processing units required by each particular algorithm. We note that, among them, the fast SMLA-3 is the most computationally intensive and its complexity is detailed as

$$C^{FSMLA-3} \approx m(3N_1^3N_2^2 + 12N_1^2\phi(2N_2) + 4N_1\phi(2N_1,2N_2) + 4\phi(K_1,K_2)) \tag{66}$$

operations. Finally, it is worth noting that the proposed fast implementation of the SMLA algorithms has a reduced memory requirement, since the memory needed is of a size proportional to $N_1^2N_2$, which is a major improvement compared to the amount of $N_1^2N_2^2$ required by the direct brute force approach.

Further computational savings can be achieved, without more than a marginal sacrificing of performance, by using approximate estimates in place of $\mathbf{R}_{N_1N_2}$, defined by (9), (12), and (15), and $\mathbf{R}_{N_1N_2}$ and $\mathbf{P}_{N_1N_2}$, defined by (19) and (20). Reminiscent of the results recently introduced in [15], where a fast approximative CG-based 1-D IAA algorithm was presented, and later extended in [34] to a block-recursive (1-D) formulation applied to blood velocity estimation in ultrasound imaging, we here propose approximative estimates for the covariance matrices involved in all the SMLAs. The implementation is motivated by the Quasi-Newton (QN) algorithm formulated in [19], and then further developed in [35–38], wherein an efficient implementation scheme of approximate recursive least squares algorithms is formed by imposing a low order AR approximation on the input signal of the adaptive algorithm. The approximation is formed by noting that the inverse of the covariance matrix $\mathbf{R}_{N_1N_2}$, defined by (36), can be built up recursively using the LLWR algorithm, iterating (41) as

$$\mathbf{R}^{-1}_{N_1M_2} = \begin{bmatrix}\mathbf{R}^{-1}_{N_1(M_2-1)} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix} + \overline{\mathcal{B}}_{M_2}\overline{\mathcal{B}}^H_{M_2} = \begin{bmatrix}0 & \mathbf{0}\\ \mathbf{0}^T & \mathbf{R}^{-1}_{N_1(M_2-1)}\end{bmatrix} + \overline{\mathcal{A}}_{M_2}\overline{\mathcal{A}}^H_{M_2} \tag{67}$$

for $M_2 = 1,\ldots N_2-1$, where $\overline{\mathcal{B}}_{M_2}$ and $\overline{\mathcal{A}}_{M_2}$ are block matrices of dimensions $N_1M_2 \times N_1$ corresponding to lower order backward and forward predictors, propagated during the recursive procedure imposed by the LWWR algorithm. An approximative inverse covariance matrix is then constructed, denoted hereafter by $\mathbf{Q}^{-1}_{N_1N_2}$, by imposing a specific structure in the forward and backward predictors involved in (67) detailed as

$$\overline{\mathcal{B}}^Q_{M_2} = \begin{bmatrix}\mathbf{0}_{N_1}\\ \overline{\mathcal{B}}^Q_{M_2-1}\end{bmatrix}, \quad \overline{\mathcal{A}}^Q_{M_2} = \begin{bmatrix}\overline{\mathcal{A}}^Q_{M_2-1}\\ \mathbf{0}_{N_1}\end{bmatrix} \tag{68}$$

for $M_2 = M_2^Q+1, M_2^Q+2, \ldots N_2$, which is initialized by

$$\overline{\mathcal{B}}^Q_{M_2^Q} = \overline{\mathcal{B}}_{M_2^Q}, \quad \overline{\mathcal{A}}^Q_{M_2^Q} = \mathcal{A}_{M_2^Q} \tag{69}$$

and where

$$\overline{\mathcal{B}}_{M_2^Q} = \begin{bmatrix}-\mathbf{R}^{-1}_{N_1(M_2^Q-1)}\mathcal{R}^b_{M_2^Q-1}\\ \mathbf{I}_{N_1}\end{bmatrix}\mathbf{A}^{-b(M_2^Q)/2}_{N_1} \tag{70}$$

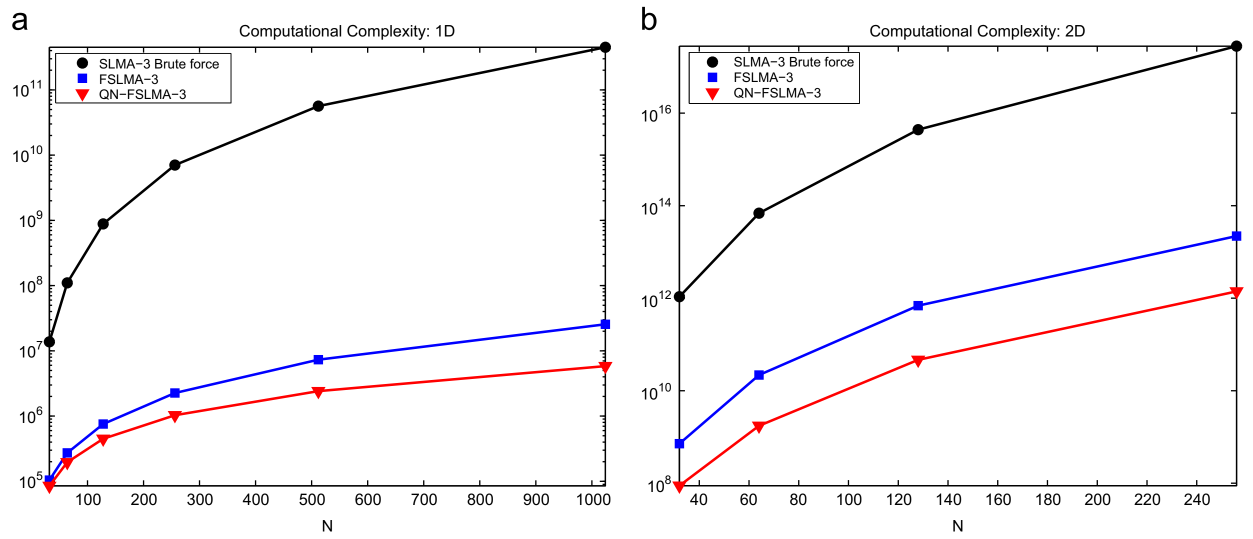

Fig. 1. Computational complexity of the SMLA-3 algorithm using brute force, the proposed fast G–S implementation and the proposed QN G–S fast implementation where the number of SMLA iterations is set equal to $m=10$, for (a) the 1-D case where the number of frequency points is $K=10N$, and the QN order is $M^Q=N/4$, and (b) for the 2-D case with $N_1=N_2=N$, $K_1=K_2=5N$, and $M^Q=N_2/4$.

$$\overline{\mathcal{A}}_{M_2^Q} = \begin{bmatrix} \mathbf{I}_{N_1} \\ -\mathbf{R}^{-1}_{N_1(M_2^Q-1)}\mathcal{R}^{f}_{M_2^Q-1} \end{bmatrix} \mathbf{A}_{N_1}^{-f(M_2^Q)/2} \tag{71}$$

$$\mathbf{A}_{N_1}^{b(M_2^Q)} = \mathbf{R}^0_{N_1} - \mathcal{R}^{bH}_{M_2^Q-1}\mathbf{R}^{-1}_{N_1(M_2^Q-1)}\mathcal{R}^{b}_{M_2^Q-1} \tag{72}$$

$$\mathbf{A}_{N_1}^{f(M_2^Q)} = \mathbf{R}^0_{N_1} - \mathcal{R}^{fH}_{M_2^Q-1}\mathbf{R}^{-1}_{N_1(M_2^Q-1)}\mathcal{R}^{f}_{M_2^Q-1} \tag{73}$$

imposing a recursive estimate of $\mathbf{Q}^{-1}_{N_1N_2}$ as

$$\begin{aligned}\mathbf{Q}^{-1}_{N_1M_2} &= \begin{bmatrix} \mathbf{Q}^{-1}_{N_1(M_2-1)} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \overline{\mathcal{B}}^{Q}_{M_2}\overline{\mathcal{B}}^{QH}_{M_2} \\ &= \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0}^T & \mathbf{Q}^{-1}_{N_1(M_2-1)} \end{bmatrix} + \overline{\mathcal{A}}^{Q}_{M_2}\overline{\mathcal{A}}^{QH}_{M_2}\end{aligned} \tag{74}$$

for $M_2 = M_2^Q+1, M_2^Q+2, \dots N_2$, which is initialized by $\mathbf{Q}^{-1}_{N1M_2^Q} = \mathbf{R}^{-1}_{N_1M_2^Q}$. Using (74), the displacement representation of $\mathbf{Q}^{-1}_{N_1N_2}$ with respect to $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}_{N_1N_2}$ takes the form

$$(\mathbf{T}_{N_1N_2,2N_2}, \mathbf{T}_{N_1N_2,2N_2}, \boldsymbol{\Gamma}_{2N_2}) \tag{75}$$

where $\mathbf{T}_{N_1N_2,2N_2} \triangleq [\overline{\mathcal{A}}^{Q}_{N_2}\mathbf{Z}_{N_1N_2}\overline{\mathcal{B}}^{Q}_{N_2}]$ and $\boldsymbol{\Gamma}_{2N_2} \triangleq \mathrm{diag}\{\mathbf{I}_{N_2}, -\mathbf{I}_{N_2}\}$, and where

$$\overline{\mathcal{A}}^{Q}_{N_2} = \begin{bmatrix} \overline{\mathcal{A}}_{M_2^Q} \\ \mathbf{0}_{N_1(N_2-M_2^Q)} \end{bmatrix}, \quad \overline{\mathcal{B}}^{Q}_{N_2} = \begin{bmatrix} \mathbf{0}_{N_1(N_2-M_2^Q)} \\ \overline{\mathcal{B}}_{M_2^Q} \end{bmatrix} \tag{76}$$

The advantage of using the approximative inverse covariance matrices in place of the exact counterparts $\mathbf{R}^{-1}_{N_1N_2}$ and $\mathbf{P}^{-1}_{N_1N_2}$ required by the SMLAs, resulting in the approximative algorithms here termed the QN-SMLA SMLAs, stems from the fact that the complexity of computing the displacement representation of the approximate inverse covariance in (75) using the LWWR algorithm requires approximately $1.5N_1^3(M_2^Q)^2$ operations as opposed to the $1.5N_1^3N_2^2$ operations required by the displacement representation of $\mathbf{R}^{-1}_{N_1N_2}$, where, typically, $M_2^Q \ll N_2$.

The computational complexity of the proposed implementations of the SMLA-3 algorithm is depicted in Fig. 1(a) for the 1-D and 1(b) for the 2-D case (the 1-D case results from the 2-D counterpart by setting $N_1=1$). Compared to the brute force implementation, the proposed implementation offers a speed up of several orders of magnitude.

4. Numerical and experimental examples

We now proceed to evaluate the discussed algorithms and proposed implementations using several 1-D spectral estimation and 2-D SAR imaging examples.

4.1. Spectral estimation

The power spectrum of a mixture of sinusoidal signals corrupted by additive zero-mean complex Gaussian noise is estimated using various techniques. The signal is composed by four sinusoids located at frequencies 0.05, 0.065, 0.27, and 0.28, with all except the fourth sinusoid having unit amplitude (the fourth sinusoid has an amplitude of 0.5). The noise variance σ^2, the data length N, and the number of the equally spaced frequency grid points K are set equal to 0.01, 100 and 1000, respectively. The performance of the spectral estimation algorithms is illustrated in Fig. 2, where superimposed spectra over 100 independent experiments are shown for each one of the spectral estimation methods used. The iterative numbers for all iterative approaches, including SLIM, IAA-R,[1] and the SMLA variants, are fixed to ten.[2] From Fig. 2(a), it is clear that the

[1] In order to estimate the noise variance, we use IAA-R (a regularized IAA algorithm which accounts for the additive noise [12]).

[2] In the examined examples, no significant further improvement was achieved after the specified number of iterations.

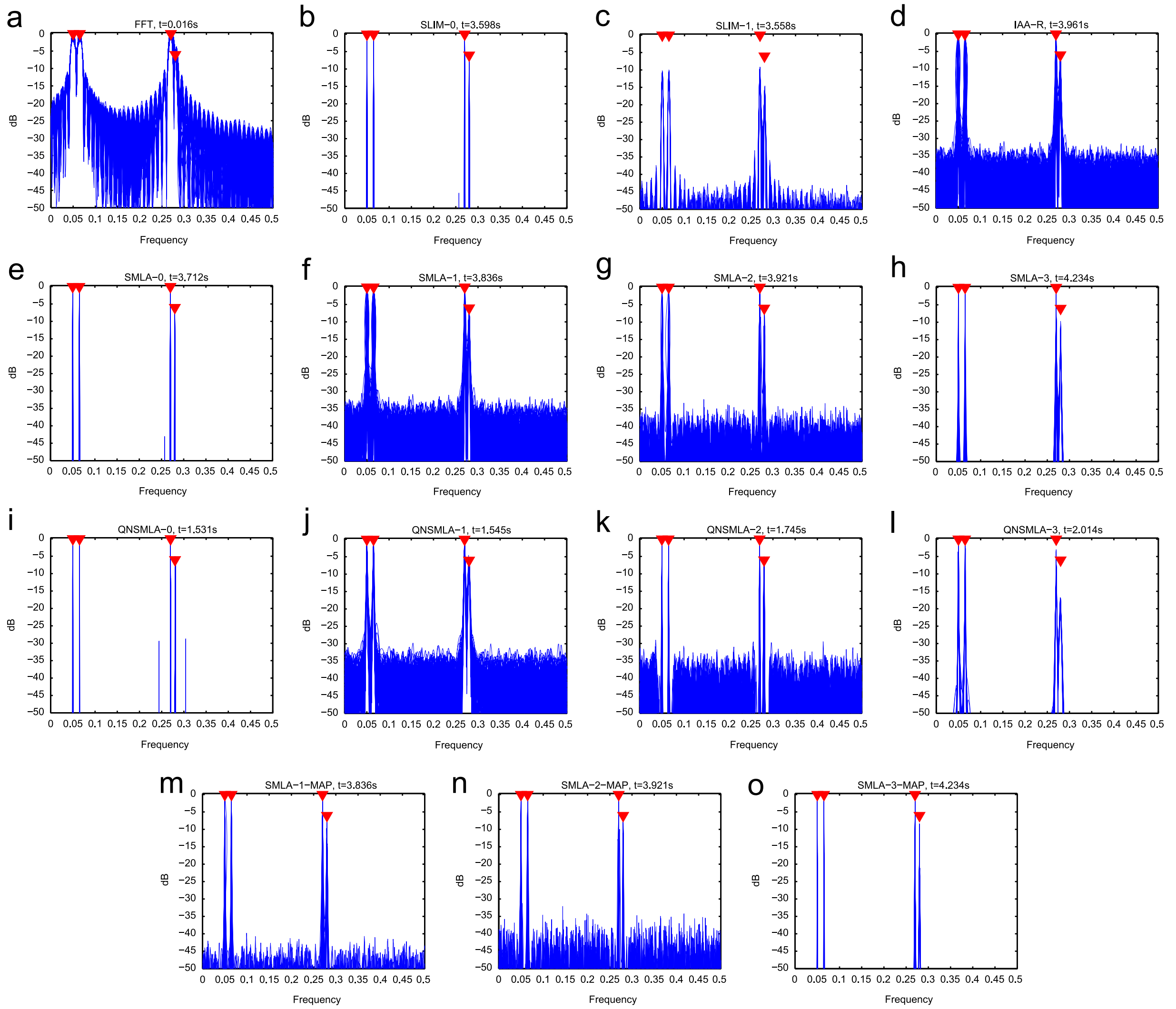

Fig. 2. Simulation results obtained via periodogram, SLIM, IAA-R, and the SMLA variants: superimposed spectra estimated using 100 independent realizations for a mixture of sinusoids using $N=100$ and $K=1000$ equally spaced frequency points. The red triangles indicate the location of the true frequencies. (a) FFT, (b) SLIM-0, (c) SLIM-1, (d) IAA-R, (e) SMLA-0, (f) SMLA-1, (g) SMLA-2, (h) SMLA-3, (i) QNSMLA-0, (j) QNSMLA-1, (k) QNSMLA-2, (l) QNSMLA-3, (m) SMLA-1-MAP, (n) SMLA-2-MAP and (o) SMLA-3-MAP. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Table 1
The noise variance estimates averaged over 100 realizations obtained by IAA-R, SLIM, the (QN-) SMLAs.

Algorithm	σ^2
SLIM-0	0.0095
SLIM-1	4.5×10^{-28}
IAA-R	0.0016
SMLA-0	0.01
SMLA-1 (-MAP)	3.3×10^{-4}
SMLA-2 (-MAP)	0.0024
SMLA-3 (-MAP)	0.0085
QN-SMLA-0	0.01
QN-SMLA-1	0.0037
QN-SMLA-2	0.0040
QN-SMLA-3	0.0093

Table 2
Computation times needed for the 1-D spectral estimation on 100 realizations.

Algorithm	Time (s)
FFT	0.016
SLIM-0	3.598
SLIM-1	3.558
IAA-R	3.961
SMLA-0	3.712
SMLA-1 (-MAP)	3.836
SMLA-2 (-MAP)	3.921
SMLA-3 (-MAP)	4.234
QN-SMLA-0	1.531
QN-SMLA-1	1.545
QN-SMLA-2	1.745
QN-SMLA-3	2.014

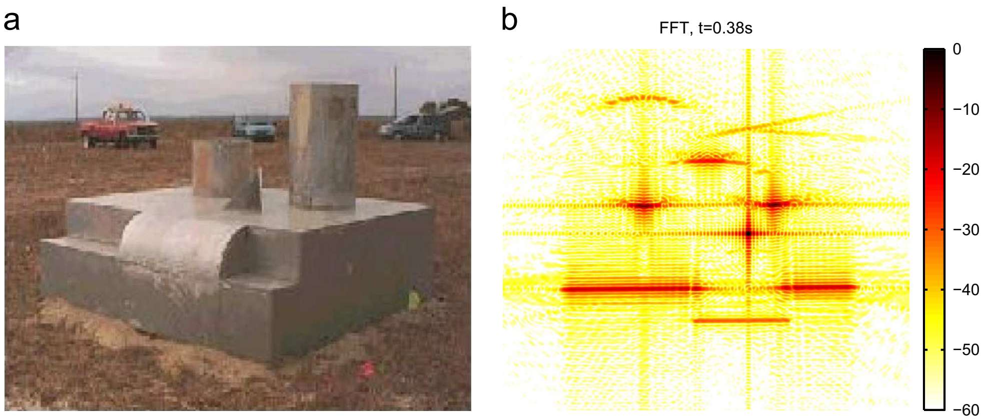

Fig. 3. Slicy object and benchmark SAR image. (a) Photograph of the object (taken at 45° azimuth angle), and (b) benchmark SAR image formed with a 288×288 (not 80×80) data matrix.

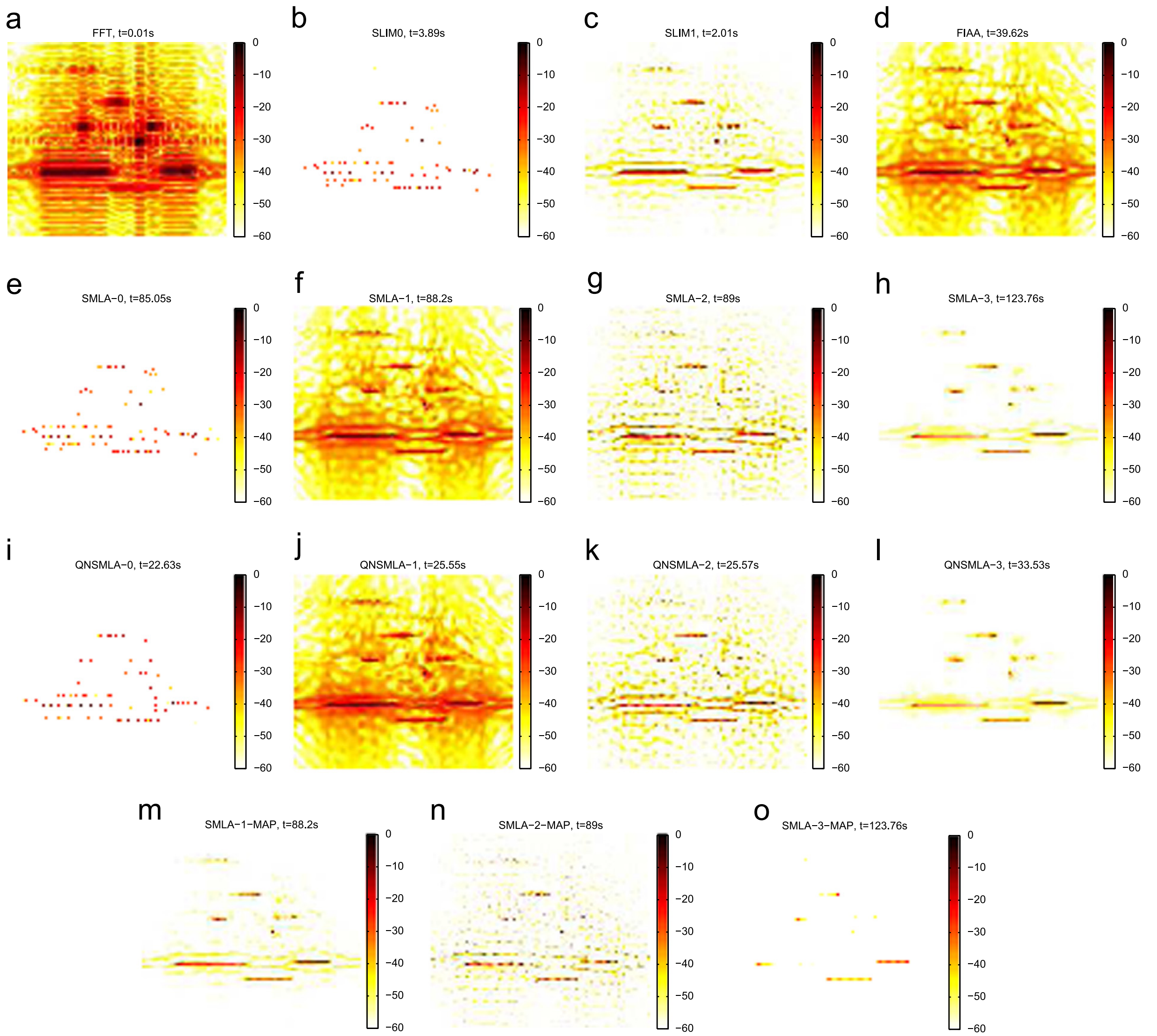

Fig. 4. Modulus of the SAR images of the Slicy object obtained from an 80×80 data matrix via FFT, SLIM, IAA, and the SMLAs. (a) FFT, (b) SLIM-0, (c) SLIM-1, (d) IAA, (e) SMLA-0, (f) SMLA-1, (g) SMLA-2, (h) SMLA-3, (i) QNSMLA-0, (j) QNSMLA-1, (k) QNSMLA-2, (l) QNSMLA-3, (m) SMLA-1-MAP, (n) SMLA-2-MAP and (o) SMLA-3-MAP.

periodogram suffers from the low resolution and high sidelobe level problems. Due to the high sidelobe levels, the fourth spectral line at frequency 0.28 fails to be detected. In comparison, SLIM, IAA-R, and the SMLA variants provide improved resolution and clearly identify all spectral lines as shown in Fig. 2(b)–(o). Specifically, IAA provides dense estimation results while both SLIM-0 and SLIM-1 achieve notably lower sidelobe levels. Also, SLIM-1 is inferior to SLIM-0 and IAA in that its spectral estimates are significantly biased downward. Among the SMLA variants, the estimation results obtained by SMLA-0 and SMLA-1 are similar to those obtained by SLIM-0 and IAA, respectively. SMLA-2 attenuates the sidelobe further compared with SMLA-1 and we conclude that the performance of SMLA-2 is between those of SMLA-0 and SMLA-1. We also remark that the estimate of the fourth spectral line given by SMLA-3 is notably biased downward. By comparing Fig. 2(i)–(l) with Fig. 2(e)–(h), we can see that the QN-SMLA variants (with $M^{Q}=32$) reduce the computational time at the cost of slight performance degradations as compared to their SMLA counterparts. In order to achieve sidelobe levels comparable to those of SLIM-0, we consider hybrid approaches that first use the SMLA variants to obtain dense spectral estimates, which are then, upon convergence, followed by a single step of SLIM-0. Since SLIM achieves sparsity based on solving a hierarchical Bayesian model through maximizing *a posteriori* probability density function [6], this single step of SLIM-0 is referred to as a MAP step, and the resulting algorithms as the SMLA-MAP algorithms. Fig. 2(m)–(o) shows the estimation results obtained via SMLA-1-MAP, SMLA-2-MAP, and SMLA-3-MAP, respectively, suggesting the effectiveness of the MAP step. Moreover, the noise variance estimates averaged over 100 realizations obtained by SLIM, IAA-R, and the SMLA variants are shown in Table 1, where SMLA-0 (as well as QN-SMLA-0) gives the most accurate estimate. The computation times needed by the aforementioned algorithms to generate the spectral estimates from 100 independent realizations on an ordinary workstation (Intel Xeon E5520 processor 2.26G Hz, 24GB RAM, Windows 7 64-bit, and MATLAB R2009b) are summarized in Table 2 showing that the proposed fast implementations make the computational complexities of the SMLA approaches comparable to those of the fast-implemented SLIM and IAA methods.

Table 3
Computation times for SAR imaging of the Slicy data (second column) and the GOTCHA data (third column).

Algorithm	Time (s)	Time (s)
FFT	0.01	0.3
SLIM-0	3.89	143.7
SLIM-1	2.01	123.0
IAA-R	39.62	4091.0
SMLA-0	85.05	7933.6
SMLA-1 (-MAP)	88.2	8151.2
SMLA-2 (-MAP)	89.0	8246.5
SMLA-3 (-MAP)	123.76	11623.4
QN-SMLA-0	22.63	2022.1
QN-SMLA-1	25.55	2312.0
QN-SMLA-2	25.57	2289.2
QN-SMLA-3	33.53	3038.6

4.2. 2-D SAR imaging

We proceed to examine the performance of the discussed estimators on the simulated Slicy data set and the experimentally measured GOTCHA data set. We begin by examining the 2-D phase-history Slicy data generated at 0° azimuth angle using XPATCH [39], a high frequency electromagnetic scattering prediction code for complex 3-D objects. A photo of the Slicy object taken at 45° azimuth angle and a SAR image benchmark obtained via the periodogram from a complete 288×288 data matrix are shown in Fig. 3(a) and (b), respectively. In the following, we examine a lower dimensional subset formed using only the $N_1 = N_2 = 80$

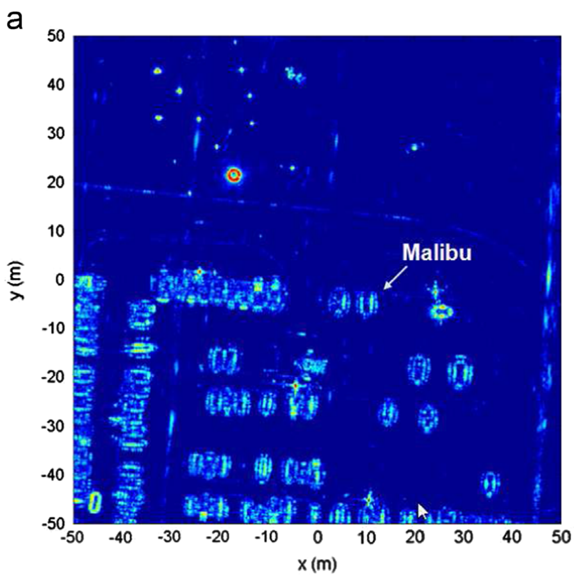

Fig. 5. 2-D SAR image (a) and photograph of Chevrolet Malibu (b) form the GOTCHA scene. Image from [40].

center block of the phase-history data, with $K_1 = K_2 = 400$ uniformly spaced 2-D frequency points. Fig. 4 shows the SAR images obtained by periodogram, SLIM, IAA, and the SMLAs using the same parameter settings as specified in Section 4.1. One again observes from Fig. 4(a) that the periodogram has low resolution and high sidelobe problems. In comparison, SLIM-0, SLIM-1 and IAA provide improved resolution and performance as shown in Fig. 4(b), (c) and (d), respectively. Specifically, SLIM-0 yields a higher resolution yet preserves less details than IAA and the performance of SLIM-1 is in between. Similar observations can be made about the performance of the SMLA variants as in Section 4.1. We also remark that the QN-SMLA variants (with $M_2^Q = 32$) generate slightly denser images with much reduced computation times compared to their SMLA counterparts. By comparing the images obtained via the SMLA variants (see Fig. 4(f)-(h)) with those of the corresponding SMLA-MAP variants (Fig. 4 (m)–(o)), we can see that the MAP step effectively converts the original dense images into much sparser ones. We remark that the images formed by the SMLA-3 and SMLA-1-MAP are sparser than that of SLIM-1 yet denser than that of SLIM-0. It appears that both SMLA-3 and SMLA-1-MAP satisfactorily balance the tradeoffs between the image resolution and detail preservation compared to SLIM-0 and SLIM-1. Table 3 summarizes the computation times needed by the aforementioned algorithms to form the $K_1 \times K_2$ SAR image from the Slicy data. As previously mentioned, the QN approach applied to the SMLA variants reduces the computation cost significantly with only slight performance degradations compared to their SMLA counterpart.

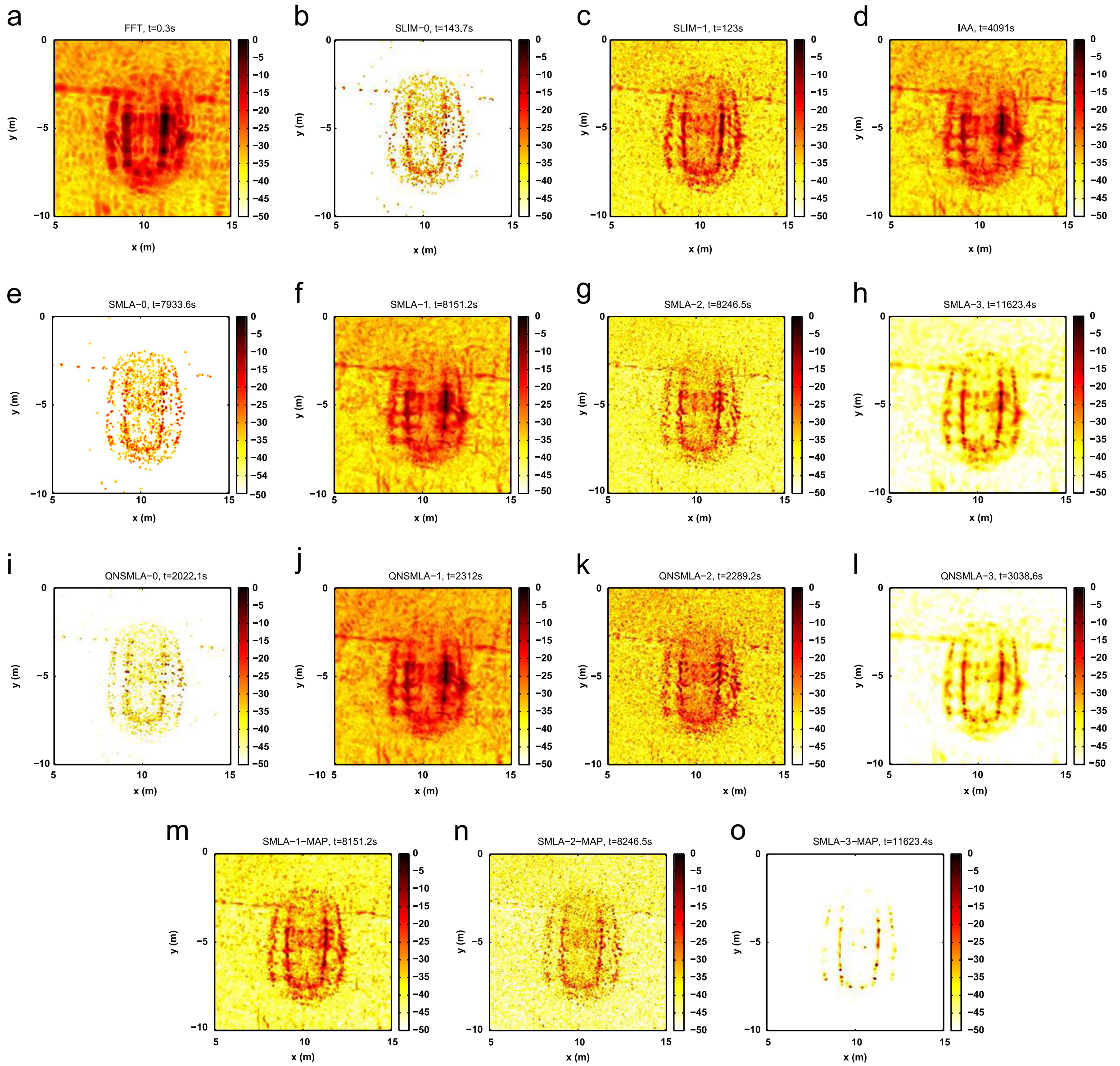

Fig. 6. Comparison of the reconstructed Malibu images obtained by FFT, SLIM, IAA, and the SMLA variants. (a) FFT, (b) SLIM-0, (c) SLIM-1, (d) IAA, (e) SMLA-0, (f) SMLA-1, (g) SMLA-2, (h) SMLA-3, (i) QNSMLA-0, (j) QNSMLA-1, (k) QNSMLA-2, (l) QNSMLA-3, (m) SMLA-1-MAP, (n)SMLA-2-MAP and (o) SMLA-3-MAP.

We proceed to examine the methods' performance for the GOTCHA Air Force Research Laboratory data set. The GOTCHA volumetric SAR data set, Version 1.0, consists of SAR phase history data collected at X-band with a 640 MHz bandwidth with full azimuth coverage at eight different elevation angles with full polarization [40]. The imaging scene consists of numerous civilian vehicles and calibration targets, as shown in Fig. 5(a). Here, we examine the performance on the phase history data with full azimuth coverage collected at the first pass for a HH polarization channel of a Chevrolet Malibu, parked in the upper corner of the parking lot as shown in Fig. 5(b). We use 4° subapertures from 0° to 360° with no overlap, which results in a total of 90 subapertures. For each subaperture, one 2-D spatial image is formed by using a 2-D FFT on the corresponding phase history (k-space) data. An 80×80 block of the spatial data centered about the Chevrolet Malibu is then chipped out and transformed back into k-space using an IFFT operation. The discussed spectral estimation techniques are then applied to the so-obtained 80×80 phase history data to get one image for each subaperture, which are then, using the auxiliary information provided by the GOTCHA data set (e.g., the antenna locations, range to scene center, azimuth and elevation angles), projected onto the ground plane and interpolated to form a 2-D ground image. The resulting 90 2-D ground images are then combined using the non-coherent max magnitude operator to yield the reconstructed Malibu image, whose dimensions are $[5, 15] \times [-10, 0]$ meters with grid size 0.05 meters in both dimensions. Fig. 6 illustrates the resulting images for the discussed methods, clearly showing the superior performance of the introduced algorithms as compared to the FFT based approach. As before, SMLA-3 and SMLA-1-MAP provide well-balanced image resolution and detail preservation and are thus preferred. The computational times of forming the 90 images are listed in Table 3, showing a comparable relative complexity as in the Slicy data example.

5. Conclusions

In this work, we have presented fast implementations of the 2-D SMLA algorithms, exploring the rich internal structure of the estimators. The proposed implementations are found to offer a significantly reduced computational complexity, with the proposed approximative implementations offering even further computational reductions, at the cost of only slight performance degradations. By including a sparsity promoting final step at the conclusion of the iterations, notable sidelobe level reductions are achieved, allowing for a satisfactory balance between the image resolution and detail preservation. The effectiveness of the algorithms has been verified using both simulated and experimentally measured data sets.

References

[1] J. Capon, High resolution frequency wave number spectrum analysis, Proceedings of the IEEE 57 (1969) 1408–1418.

[2] J. Li, P. Stoica, An adaptive filtering approach to spectral estimation and SAR imaging, IEEE Transactions on Signal Processing 44 (1996) 1469–1484.

[3] S.R. DeGraaf, SAR imaging via modern 2-D spectral estimation methods, in: Proceedings of the SPIE on Optical Engineering in Aerospace Sensing.

[4] S.R. DeGraaf, SAR imaging via modern 2-D spectral estimation methods, IEEE Transactions on Image Processing 7 (1998) 729–761.

[5] G.R. Benitz, High-definition vector imaging, MIT Lincoln Laboratory Journal 10 (1997) 147–170.

[6] X. Tan, W. Roberts, J. Li, P. Stoica, Sparse learning via iterative minimization with application to MIMO radar imaging, IEEE Transactions on Signal Processing 59 (2011) 1088–1101.

[7] T. Yardibi, J. Li, P. Stoica, M. Xue, A.B. Baggeroer, Source localization and sensing: a nonparametric iterative approach based on weighted least squares, IEEE Transactions on Aerospace and Electronic Systems 46 (2010) 425–443.

[8] H. Abeida, X. Li, J. Li, M.M. Al-Harthi, Iterative sparse maximum likelihood-based algorithms with applications to SAR imaging, International Journal of Remote Sensing Applications 2 (2012) 10–23.

[9] H. Abeida, Q. Zhang, J. Li, N. Merabtine, Iterative sparse asymptotic minimum variance based approaches for array processing, IEEE Transactions on Signal Processing 61 (2013) 933–944.

[10] P. Stoica, J. Li, J. Ling, Missing data recovery via a nonparametric iterative adaptive approach, IEEE Signal Processing Letters 16 (2009) 241–244.

[11] P. Stoica, J. Li, H. He, Spectral analysis of nonuniformly sampled data: a new approach versus the periodogram, IEEE Transactions on Signal Processing 57 (2009) 843–858.

[12] W. Roberts, P. Stoica, J. Li, T. Yardibi, F.A. Sadjadi, Iterative adaptive approaches to MIMO radar imaging, IEEE Journal of Selected Topics in Signal Processing 4 (2010) 5–20.

[13] N.R. Butt, A. Jakobsson, Coherence spectrum estimation from nonuniformly sampled sequences, IEEE Signal Processing Letters 17 (2010) 339–342.

[14] D. Vu, L. Xu, M. Xue, J. Li, Nonparametric missing sample spectral analysis and its applications to interrupted SAR, IEEE Journal of Selected Topics in Signal Processing 6 (2012) 1–14.

[15] G.-O. Glentis, A. Jakobsson, Time-recursive IAA spectral estimation, IEEE Signal Processing Letters 18 (2011) 111–114.

[16] G.-O. Glentis, A. Jakobsson, Efficient implementation of iterative adaptive approach spectral estimation techniques, IEEE Transactions on Signal Processing 59 (2011) 4154–4167.

[17] M. Xue, L. Xu, J. Li, IAA spectral estimation: fast implementation using the Gohberg–Semencul factorization, IEEE Transactions on Signal Processing 59 (2011) 3251–3261.

[18] G.-O. Glentis, A. Jakobsson, Superfast approximative implementation of the IAA spectral estimate, IEEE Transactions on Signal Processing 60 (2012) 472–478.

[19] G.V. Moustakides, S. Theodoridis, Fast Newton transversal filters – a new class of adaptive estimation algorithms, IEEE Transactions on Signal Processing 39 (1991) 2184–2193.

[20] G.-O. Glentis, K. Zhao, A. Jakobsson, J. Li, Non-parametric high-resolution SAR imaging, IEEE Transactions on Signal Processing 61 (2013) 1614–1624.

[21] T. Kailath, A.H. Sayed, Displacement structure: theory and applications, SIAM Review 37 (1995) 297–386.

[22] I. Gohberg, V. Olshevksy, Complexity of multiplication with vectors for structured matrices, Linear Algebra and its Applications 202 (1994) 163–192.

[23] D. Wood, Product rules for the displacement of near-Toeplitz matrices, Linear Algebra and its Applications 188/189 (1993) 641–663.

[24] T. Kailath, A.H. Sayed, Fast Reliable Algorithms for Matrices with Structure, SIAM, Philadelphia, USA, 1999.

[25] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[26] G.-O. Glentis, A fast algorithm for APES and capon spectral estimation, IEEE Transactions on Signal Processing 56 (2008) 4207–4220.

[27] J. Dias, J. Leitao, Efficient computation of tr(TR-1) for Toeplitz matrices, IEEE Signal Processing Letters 9 (2002) 54–56.

[28] P. Stoica, R. Moses, Spectral Analysis of Signals, Prentice Hall, Upper Saddle River, NJ, 2005.

[29] K. Angelopoulos, G.O. Glentis, A. Jakobsson, Efficient Implementation of the IAA-based magnitude squared coherence estimator, in: 17th International Conference on Digital Signal Processing, Corfu, pp. 1–6.

[30] K. Angelopoulos, G.O. Glentis, A. Jakobsson, Efficient time recursive coherence spectrum estimation, in: 20th European Signal Processing Conference, Bucharest, pp. 425–429.

[31] K. Angelopoulos, G.O. Glentis, A. Jakobsson, Computationally efficient sparsity-inducing coherence spectrum estimation of complete and non-complete data sets, Elsevier Signal Processing 93 (2013) 1221–1234.

[32] T. Söderström, P. Stoica, System Identification, Prentice Hall International, London, UK, 1989.

[33] N. Kalouptsidis, G. Carayannis, D. Manolakis, Fast algorithms for block toeplitz matrices with toeplitz entries, Signal Processing 6 (1984) 77–81.
[34] A. Jakobsson, G.O. Glentis, E. Gudmundson, Computationally efficient time-recursive IAA-based blood velocity estimation, IEEE Transactions on Signal Processing 60 (2012) 3853–3858.
[35] B. Farhang-Boroujeny, Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation, IEEE Transactions on Signal Processing 45 (1997) 1987–2000.
[36] B. Farhang-Boroujeny, H. Rao, Fast LMS/Newton algorithms for stereophonic acoustic echo cancellation, IEEE Transactions on Signal Processing 57 (2009) 2919–2930.
[37] A. Gilloire, T. Petillon, S. Theodoridis, The fast Newton transversal filter: an efficient scheme for acoustic echo cancellation in mobile radio, IEEE Transactions on Signal Processing 42 (1994) 509–518.
[38] G. Moustakides, S. Theodoridis, K. Berberidis, A fast Newton multichannel algorithm for decision feedback equalization, IEEE Transactions on Signal Processing 43 (1995) 327–331.
[39] D.J. Andersh, M. Hazlett, S.W. Lee, D.D. Reeves, D.P. Sullivan, Y. Chu, XPATCH: a high-frequency electromagnetic scattering prediction code and environment for complex three-dimensional objects, IEEE Antennas and Propagation Magazine 36 (1994) 65–69.
[40] Gotcha 2D/3D imaging challenge problem (Air Force Research Lab., Jan. 2010 [Online]. Available: ⟨https://www.sdms.afrl.af.mil/datasets/gotcha/⟩).