LUND UNIVERSITY

Non-Parametric High-Resolution SAR Imaging

G. O. GLENTIS, K. ZHAO, A. JAKOBSSON, AND J. LI

Published in: IEEE Transactions of Signal Processing
doi:10.1109/TSP.2012.2232662

Lund 2013

Mathematical Statistics
Centre for Mathematical Sciences
Lund University

Non-Parametric High-Resolution SAR Imaging

G. O. Glentis*, *Member, IEEE* , K. Zhao†, A. Jakobsson**, *Senior Member, IEEE* , and J. Li†, *Fellow, IEEE*

***Abstract*— The development of high-resolution two-dimensional spectral estimation techniques is of notable interest in synthetic aperture radar (SAR) imaging. Typically, data-independent techniques are exploited to form the SAR images, although such approaches will suffer from limited resolution and high sidelobe levels. Recent work on data-adaptive approaches have shown that both the iterative adaptive approach (IAA) and the sparse learning via iterative minimization (SLIM) algorithm offer excellent performance with high-resolution and low side lobe levels for both complete and incomplete data sets. Regrettably, both algorithms are computationally intensive if applied directly to the phase history data to form the SAR images. To help alleviate this, efficient implementations have also been proposed. In this paper, we further this work, proposing yet further improved implementation strategies, including approaches using the segmented IAA approach and the approximative quasi-Newton technique. Furthermore, we introduce a combined IAA-MAP algorithm as well as a hybrid IAA- and SLIM-based estimation scheme for SAR imaging. The effectiveness of the SAR imaging algorithms and the computational complexities of their fast implementations are demonstrated using the simulated Slicy data set and the experimentally measured GOTCHA data set.**

***Index Terms*— Spectral estimation, synthetic aperture radar imaging, data adaptive techniques, efficient algorithms.**

I. Introduction

SYNTHETIC aperture radar (SAR) systems find applicability in a wide variety of commercial and governmental applications, including monitoring, mapping, and reconnaissance systems, and offers notable benefits due to such systems' ability to image in all weather conditions and times of the day. The measured SAR images are generally processed via various pre- and post-processing techniques, such as data-independent Fourier transforms and back-projections, estimating the scene reflectivity intensity to form an intensity image. The Fourier methods exploit the relationship between the signal phase history measurements and the scene reflectivity, but generally suffers from limited resolution and/or sidelobe artifacts, as well as the speckle phenomena [1]–[5]. To reduce these effects, one is typically forced to include various forms of smoothing and filtering, although this will result in further reduced image resolution. These drawbacks have led to an interest in finding improved processing techniques to form the two-dimensional (2-D) spectral estimate required to form the SAR image [5]–[10]. Of the presented approaches, the data-dependent Capon and APES algorithms [7], [10]–[12] seem particularly promising, although both methods generally require multiple snapshots (which is hard to satisfy due to the platform motion) or the use of sub-apertures (which would lead to lower resolution) to form the required sample covariance matrix estimate. The use of sparse signal recovery methods have also been investigated (see, e.g., [13], [14]) as such systems are well able to recover sparse radar images with high resolution, without requiring multiple snapshots to do so. Regrettably, this form of approaches is sensitive to the choice of the various user parameters, which are typically difficult to select in practice, as well as lack robustness to varying noise levels [15]. Recently, the iterative adaptive approach (IAA) [16] and the sparse learning via iterative minimization (SLIM) algorithm [17] have also been investigated for high-resolution spectral estimation (see, e.g. [18]–[22]), with both algorithms showing excellent performance for both complete or incomplete data sets. Regrettably, both algorithms are computationally intensive, and there has as a result been several works on how to form computationally efficient 1-D and 2-D implementations of these estimates for uniformly and non-uniformly sampled data sequences [22]–[26]. These implementations are formed exploiting the methods' inherent low displacement ranks, together with the development of suitable Gohberg-Semencul (GS) representations, as well as making use of Levinson-style and/or (possibly preconditioned) conjugate gradient (CG) solvers of the resulting linear systems of equations. In this paper, we further this work by combining and improving on the earlier presented implementations, including extending the segmented IAA (SIAA) algorithm introduced in [21] and the approximative quasi-Newton preconditioning CG algorithm developed for 1-D data sequences in [26] to 2-D data sets. Furthermore, we introduce a combined IAA-MAP algorithm as well as a hybrid IAA- and SLIM-based estimation scheme.

The remainder of this paper is organized as follows. In the following section, we briefly review the 2-D IAA and SLIM algorithms, respectively. Then, in Section III, we present improved efficient implementations of the IAA algorithm, followed in Section IV with similar improved techniques for the SLIM algorithm. In Section V, we then introduce several hybrid estimation schemes, whereas Section VI presents the performance of the proposed implementations on both simulated and measured SAR data sets. Finally, the paper is concluded in Section VII.

This work was supported in part by the Swedish Research Council, Carl Trygger's foundation, as well as ARO under grant No. W911NF-11-2-0039, DoD under grant No. HM1582-10-1-0017, and NSF under Grant No. ECCS-0729727. 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.

*G. O. Glentis is with the Department of Science and Technology of Telecommunications, University of Peloponnese, Tripolis, 22100 Greece, email: gglentis@uop.gr.

†K. Zhao and J. Li are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6130, USA, email: kexinzhao@ufl.edu, li@dsp.ufl.edu.

**A. Jakobsson is with the Dept. of Mathematical Statistics, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden, email: aj@maths.lth.se.

II. The 2-D IAA and SLIM algorithms

Let $y(n_1, n_2)$ denote the 2-D phase history data of interest, and organize the data in a column-wise form, introducing

$$\mathbf{Y}_{N_1,N_2} = \begin{bmatrix}\mathbf{y}_{N_1}(0) & \dots & \mathbf{y}_{N_1}(N_2-1)\end{bmatrix} \quad (1)$$

$$\mathbf{y}_{N_1}(n_2) = \begin{bmatrix} y(0,n_2) & \dots & y(N_1-1,n_2)\end{bmatrix}^T \quad (2)$$

where $n_1 = 0, 1, \dots, N_1 - 1$ and $n_2 = 0, 1, \dots, N_2 - 1$. Moreover, let $\mathbf{y}_{N_1N_2} = \text{vec}\{\mathbf{Y}_{N_1,N_2}\}$ and $\mathbf{Y}_{N_1,N_2} = \text{mat}\{\mathbf{y}_{N_1N_2}\}$, where $\text{vec}\{\cdot\}$ denotes column-wise vectorization, and $\text{mat}\{\cdot\}$ the inverse operation, recreating the matrix from the vectorized matrix. Furthermore, define the 2-D frequency vector

$$\mathbf{f}_{N_1N_2}(\omega_1, \omega_2) \triangleq \mathbf{f}_{N_2}(\omega_2) \otimes \mathbf{f}_{N_1}(\omega_1), \quad (3)$$

where $\otimes$ denotes the Kronecker product, and $\mathbf{f}_N(\omega) \triangleq \begin{bmatrix} 1 & e^{\jmath\omega} & \dots & e^{\jmath(N-1)\omega}\end{bmatrix}^T$. The 2-D data model can then be written as

$$\mathbf{y}_{N_1N_2} = \mathbf{F}_{N_1N_2,K_1K_2}\boldsymbol{\alpha}_{K_1K_2} + \mathbf{e}_{N_1N_2} \quad (4)$$

where

$$\mathbf{F}_{N_1N_2,K_1K_2} \triangleq \begin{bmatrix}\mathbf{f}_{N_1N_2}(\omega_0,\omega_0) & \dots & \mathbf{f}_{N_1N_2}(\omega_{K_1-1},\omega_{K_2-1}\end{bmatrix}$$

is composed by the 2-D frequency vectors of interest, and

$$\boldsymbol{\alpha}_{K_1K_2} \triangleq \begin{bmatrix} \alpha(\omega_0,\omega_0) & \dots & \alpha(\omega_{K_1-1},\omega_{K_2-1})\end{bmatrix}^T \quad (5)$$

contains the complex amplitudes associated with each 2-D frequency pair $(\omega_{k_1}, \omega_{k_2})$, whereas $\mathbf{e}_{N_1N_2}$ denotes an additive noise. As detailed in [16], the IAA estimate of $\boldsymbol{\alpha}_{K_1K_2}$ is formed as the estimate minimizing

$$\min_{\boldsymbol{\alpha}_{K_1K_2}} \left|\mathbf{y}_{N_1N_2} - \hat{\mathbf{y}}_{N_1N_2}(\omega_{k_1},\omega_{k_2})\right|^2_{\left[\mathbf{R}^e_{N_1N_2}(\omega_{k_1},\omega_{k_2})\right]^{-1}} \quad (6)$$

over $\alpha(\omega_{k_1}, \omega_{k_2})$, with $(\omega_{k_1}, \omega_{k_2})$, for $k_1 = 0, 1, \dots, K_1 - 1$ and $k_2 = 0, 1, \dots, K_2 - 1$, denoting the 2-D frequency grid of interest, typically with $K_1 > N_1$ and $K_2 > N_2$, where

$$\hat{\mathbf{y}}_{N_1N_2}(\omega_{k_1},\omega_{k_2}) \triangleq \alpha(\omega_{k_1},\omega_{k_2})\mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2}) \quad (7)$$

and

$$\mathbf{R}^e_{N_1N_2}(\omega_{k_1},\omega_{k_2}) \triangleq \mathbf{R}_{N_1N_2} - \hat{\mathbf{y}}_{N_1N_2}(\omega_{k_1},\omega_{k_2})\hat{\mathbf{y}}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})$$

is the noise plus interference covariance matrix, whereas the data covariance matrix is estimated as

$$\begin{aligned}\mathbf{R}_{N_1N_2} &\triangleq \mathbf{F}_{N_1N_2,K_1K_2}\mathbf{D}_{K_1K_2}\mathbf{F}^H_{N_1N_2,K_1K_2} \\ &= \sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\hat{\mathbf{y}}_{N_1N_2}(\omega_{k_1},\omega_{k_2})\hat{\mathbf{y}}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\end{aligned}$$

with $\mathbf{D}_{K_1K_2} = \text{diag}\left\{|\alpha(\omega_0,\omega_0)|^2, \dots, |\alpha(\omega_{K_1-1},\omega_{K_2-1})|^2\right\}$, $|\mathbf{x}|^2_{\mathbf{A}} \triangleq \mathbf{x}^H\mathbf{A}\mathbf{x}$, and $(\cdot)^H$ denote the conjugate transpose. Minimizing (6) with respect to (wrt) $\alpha(\omega_{k_1}, \omega_{k_2})$ yields

$$\alpha(\omega_{k_1},\omega_{k_2}) = \frac{\mathbf{f}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\left[\mathbf{R}^e_{N_1N_2}(\omega_{k_1},\omega_{k_2})\right]^{-1}\mathbf{y}_{N_1N_2}}{\mathbf{f}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\left[\mathbf{R}^e_{N_1N_2}(\omega_{k_1},\omega_{k_2})\right]^{-1}\mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2})} \quad (8)$$

The 2-D IAA algorithm is then formed by iterating

$$\begin{aligned}\alpha(\omega_{k_1},\omega_{k_2}) &= \frac{\mathbf{f}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\mathbf{R}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}}{\mathbf{f}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\mathbf{R}^{-1}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2})} \\ &\triangleq \frac{\psi(\omega_{k_1},\omega_{k_2}}{\varphi(\omega_{k_1},\omega_{k_2})} \quad (9)\\ \mathbf{R}_{N_1N_2} &= \mathbf{F}_{N_1N_2,K_1K_2}\mathbf{D}_{K_1K_2}\mathbf{F}^H_{N_1N_2,K_1K_2} \quad (10)\end{aligned}$$

until convergence, where (9) has been derived from (8) using the matrix inversion lemma, bypassing the need of computing $\left[\mathbf{R}^e_{N_1N_2}(\omega_{k_1},\omega_{k_2})\right]^{-1}$ for each 2-D frequency pair $(\omega_{k_1}, \omega_{k_2})$. Typically, $\mathbf{R}_{N_1N_2}$ is initialized to the identity matrix $\mathbf{I}_{N_1N_2}$. The computational cost of the resulting 2-D IAA algorithm using brute force is approximately $N_1^3N_2^3 + (2N_1^2N_2^2 + N_1N_2)K_1K_2$ operations.

The SLIM algorithm introduced in [17] is instead formed by minimizing the regularized cost function

$$N_1N_2\log(\eta) + \frac{1}{\eta}|\mathbf{y}_{N_1N_2} - \tilde{\mathbf{y}}_{N_1N_2}|^2 + \frac{2}{q}\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}(|\beta(\omega_{k_1},\omega_{k_2})|^q - 1) \quad (11)$$

wrt the system parameters $\boldsymbol{\beta}_{N_1N_2}$, formed as

$$\boldsymbol{\beta}_{K_1K_2} \triangleq \left[\beta(\omega_0,\omega_0) \ \dots \beta(\omega_{K_1-1}, K_2-1)\right]^T, \quad (12)$$

and the covariance noise variable η of the underline data model

$$\begin{aligned}\mathbf{y}_{N_1N_2} &= \tilde{\mathbf{y}}_{N_1N_2} + \mathbf{e}_{N_1N_2} \quad (13)\\ \tilde{\mathbf{y}}_{N_1N_2} &\triangleq \mathbf{F}_{N_1N_2K_1K_2}\boldsymbol{\beta}_{K_1K_2} \quad (14)\end{aligned}$$

and with $0 < q \leq 1$. Summarizing, the 2-D SLIM algorithm is formed by iterating [17]

$$\begin{aligned}\boldsymbol{\beta}_{K_1K_2} &= \mathbf{P}_{K_1K_2}\mathbf{F}^H_{N_1N_2,K_1K_2}\boldsymbol{\Sigma}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2} \quad (15)\\ \boldsymbol{\Sigma}_{N_1N_2} &= \mathbf{F}_{N_1N_2,K_1K_2}\mathbf{P}_{K_1K_2}\mathbf{F}^H_{N_1N_2,K_1K_2} + \eta\mathbf{I}_{N_1N_2} \quad (16)\\ \eta &= \frac{1}{N_1N_2}\left|\mathbf{y}_{N_1N_2} - \mathbf{F}^H_{N_1N_2,K_1K_2}\boldsymbol{\beta}_{K_1K_2}\right|^2 \quad (17)\end{aligned}$$

until practical convergence, with

$$\mathbf{P}_{K_1K_2} \triangleq \text{diag}\left\{|\beta(\omega_0,\omega_0)|^{q-2} \ \dots \ |\beta(\omega_{K_1-1},\omega_{K_2-1})|^{q-2}\right\}$$

where typically, $\boldsymbol{\beta}_{K_1K_2}$ is initialized by the 2-D DFT of $\mathbf{y}_{N_1N_2}$, i.e.,

$$\boldsymbol{\beta}^{ini}_{K_1K_2} = \mathbf{F}^H_{N_1N_2,K_1K_2}\mathbf{y}_{N_1N_2} \quad (18)$$

whereas η is initialized as

$$\eta^{ini} = \lambda\frac{1}{N_1N_2}\left|\mathbf{y}_{N_1N_2} - \mathbf{F}^H_{N_1N_2,K_1K_2}\boldsymbol{\beta}^{ini}_{K_1K_2}\right|^2 \quad (19)$$

where λ is a positive scaling factor.

III. Fast computation of the 2-D IAA algorithm

Given the high complexity required to form the 2-D IAA estimate, we proceed to examine ways to form computationally efficient implementations of the algorithm. We first briefly review the efficient 2-D IAA implementation recently developed independently in [24], [25], since this implementation will be the backbone of the schemes then presented herein. In this implementation, the computational reduction is achieved by making use of the inherently low displacement rank of the underlying data covariance matrix, whose Toeplitz block Toeplitz (TBT) structure allows for fast matrix inversion and matrix vector multiplication, which together with the fast computation of the relevant data dependent trigonometric polynomials (see also [27]) result in computational schemes several orders of magnitude faster than that of the direct (brute force) implementation. We then proceed to note that further computational savings can be achieved by applying a segmented data procedure, wherein the original image is first segmented into several possibly overlapped parts, each of which is subsequently processed by the IAA algorithm. Finally, reminiscent to the fast approximative implementation of the 1-D IAA algorithm presented in [26] , where the data covariance matrix is approximated by a lower order representation formed using a lower order autoregressive (AR) model, an approximative 2-D IAA algorithm and its efficient implementation are also presented. As the 2-D frequency vector (3) is defined over a uniformly spaced grid of frequencies $(\omega_{k_1},\omega_{k_2}) \triangleq (2\pi k_1/K_1,\ 2\pi k_2/K_2)$, where $k_1 = 0,1,\dots K_1-1$ and $k_2 = 0,1,\dots K_2-1$, the 2-D covariance matrix $\mathbf{R}_{N_1N_2}$, defined by (10), is a TBT matrix of the form

$$\mathbf{R}_{N_1N_2} = \begin{bmatrix} \mathbf{R}^0_{N_1} & \mathbf{R}^{1H}_{N_1} & \dots & \mathbf{R}^{(N_2-1)H}_{N_1} \\ \mathbf{R}^1_{N_1} & \mathbf{R}^0_{N_1} & \dots & \mathbf{R}^{(N_2-2)H}_{N_1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}^{N_2-1}_{N_1} & \mathbf{R}^{N_2-2}_{N_1} & \dots & \mathbf{R}^0_{N_1} \end{bmatrix} \tag{20}$$

where the matrix entries $\mathbf{R}^{\ell}_{N_1}$, for $\ell = 0,1,\dots N_2-1$, are Toeplitz matrices of size $N_1 \times N_1$. As shown in [23]–[25], $\mathbf{R}_{N_1N_2}$ may be extracted from a circulant block circulant (CBC) matrix of higher dimensions as (see also [28])

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

where $\mathbf{W}_{K_1K_2}$ denotes the 2-D Discrete Fourier Transform (DFT) matrix and $\mathbf{S}_{K_1K_2}$ is a suitable permutation matrix. The TBT structure of $\mathbf{R}_{N_1N_2}$ allows for a low displacement rank representation which results in efficient matrix inversion and fast matrix vector multiplication used in the sequel for solving the linear system of equations that appears in the numerator of (9) and defined for further use as $\mathbf{d}_{N_1N_2} \triangleq \mathbf{R}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}$, as well as for the efficient computation of the coefficients of the 2-D polynomial $\varphi(\omega_{k_1,k_2})$ that appears in the denominator of (9). Since (10) is a TBT matrix, it may be partitioned as

$$\mathbf{R}_{N_1N_2} = \begin{bmatrix} \mathbf{R}_{N_1(N_2-1)} & \boldsymbol{\mathcal{R}}^b_{N_2-1} \\ \boldsymbol{\mathcal{R}}^{bH}_{N_2-1} & \mathbf{R}^0_{N_1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^0_{N_1} & \boldsymbol{\mathcal{R}}^{fH}_{N_2-1} \\ \boldsymbol{\mathcal{R}}^f_{N_2-1} & \mathbf{R}_{N_1(N_2-1)} \end{bmatrix} \tag{21}$$

where $\boldsymbol{\mathcal{R}}^b_{N_2-1}$ and $\boldsymbol{\mathcal{R}}^f_{N_2-1}$ denote block matrices of dimensions $N_1(N_2-1)\times N_1$. Applying the matrix inversion lemma for partitioned matrices to (21) yields (see, e.g. [29])

$$\mathbf{R}^{-1}_{N_1N_2} = \begin{bmatrix} \mathbf{R}^{-1}_{N_1(N_2-1)} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{bmatrix} + \bar{\boldsymbol{\mathcal{B}}}_{N_2}\bar{\boldsymbol{\mathcal{B}}}^H_{N_2} \tag{22}$$

$$= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{R}^{-1}_{N_1(N_2-1)} \end{bmatrix} + \bar{\boldsymbol{\mathcal{A}}}_{N_2}\bar{\boldsymbol{\mathcal{A}}}^H_{N_2} \tag{23}$$

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

$$\bar{\boldsymbol{\mathcal{B}}}_{N_2} = \begin{bmatrix} \boldsymbol{\mathcal{B}}_{N_2-1} \\ \mathbf{I}_{N_1} \end{bmatrix}\mathbf{A}^{-b/2}_{N_1} \tag{24}$$

$$\bar{\boldsymbol{\mathcal{A}}}_{N_2} = \begin{bmatrix} \mathbf{I}_{N_1} \\ \boldsymbol{\mathcal{A}}_{N_2-1} \end{bmatrix}\mathbf{A}^{-f/2}_{N_1} \tag{25}$$

$$\boldsymbol{\mathcal{B}}_{N_2-1} = -\mathbf{R}^{-1}_{N_1(N_2-1)}\boldsymbol{\mathcal{R}}^b_{N_2-1} \tag{26}$$

$$\boldsymbol{\mathcal{A}}_{N_2-1} = -\mathbf{R}^{-1}_{N_1(N_2-1)}\boldsymbol{\mathcal{R}}^f_{N_2-1} \tag{27}$$

$$\mathbf{A}^b_{N_1} = \mathbf{R}^0_{N_1} + \boldsymbol{\mathcal{R}}^{bH}_{N_2-1}\boldsymbol{\mathcal{B}}_{N_2-1} \tag{28}$$

$$\mathbf{A}^f_{N_1} = \mathbf{R}^0_{N_1} + \boldsymbol{\mathcal{R}}^{fH}_{N_2-1}\boldsymbol{\mathcal{A}}_{N_2-1} \tag{29}$$

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. Define a block shifting matrix, $\mathbf{Z}_{N_1N_2}$, as $\mathbf{Z}_{N_1N_2} = \mathbf{Z}_{N_2} \otimes \mathbf{I}_{N_1}$, where

$$\mathbf{Z}_{N_2} = \begin{bmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{N_2-1} & \mathbf{0} \end{bmatrix}. \tag{30}$$

Clearly, $[\mathbf{Z}_{N_1N_2}]^{N_2} = \mathbf{0}_{N_1N_2}$. Using the above partitioning of $\mathbf{R}^{-1}_{N_1N_2}$, its displacement with respect to the lower shifting block matrices $\mathbf{Z}_{N_1N_2}$ and $\mathbf{Z}^T_{N_1N_2}$ takes the form [30]

$$\nabla_{\mathbf{Z},\mathbf{Z}^T}\left(\mathbf{R}^{-1}_{N_1N_2}\right) \triangleq \bar{\boldsymbol{\mathcal{A}}}_{N_2}\bar{\boldsymbol{\mathcal{A}}}^H_{N_2} - \mathbf{Z}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}_{N_2}\bar{\boldsymbol{\mathcal{B}}}^H_{N_2}\mathbf{Z}^T_{N_1N_2} \tag{31}$$

resulting a suitable GS factorization of $\mathbf{R}^{-1}_{N_1N_2}$ of the form

$$\mathbf{R}^{-1}_{N_1N_2} = \sum_{i=1}^{2}\sigma_i\boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{T}}^i_{N_2})\boldsymbol{\mathcal{L}}^H(\boldsymbol{\mathcal{T}}^i_{N_2}) \tag{32}$$

where $\sigma_1 = 1$, $\sigma_2 = -1$, $\boldsymbol{\mathcal{T}}^1_{N_2} \triangleq \bar{\boldsymbol{\mathcal{A}}}_{N_2}$, and $\boldsymbol{\mathcal{T}}^2_{N_2} \triangleq \mathbf{Z}_{N_1N_2}\bar{\boldsymbol{\mathcal{B}}}_{N_2}$, with $\boldsymbol{\mathcal{L}}(\cdot)$ denoting a block lower Toeplitz matrix of block dimensions $N_2\times N_2$, having block entries of size $N_1\times N_1$ each. Thus, the matrix-vector or matrix-matrix products involving (32) can be organized using the block DFT, and its fast implementation computed using the FFT [31], provided that the generator block matrices $\boldsymbol{\mathcal{T}}^1_{N_2}$ and $\boldsymbol{\mathcal{T}}^2_{N_2}$ are available. The latter can be efficiently estimated using the celebrated Levinson-Whittle-Wiggins-Robinson (LWWR) algorithm (see, e.g., [32]). The computational complexity of the LWWR algorithm is approximately $C^{LWWR} = 1.5N_1^3N_2^2 + 4N_1^3N_2$. Moreover, due to the persymmetric property that each Toeplitz matrix entry possesses, i.e., as $\mathbf{J}_{N_1N_2}\mathbf{R}_{N_1N_2}\mathbf{J}_{N_1N_2} = \mathbf{R}^T_{N_1N_2}$, 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, it holds that [33] $\boldsymbol{\mathcal{A}}_{N_2-1} = \mathbf{J}_{N_1(N_2-1)}\boldsymbol{\mathcal{B}}^*_{N_2-1}\mathbf{J}_{N_1}$ and $\mathbf{A}^f_{N_1} = \mathbf{J}_{N_1}\mathbf{A}^{bT}_{N_1}\mathbf{J}_{N_1}$, which may be used for further reduction of the computational cost (although the resulting scheme may then be more sensitive to the numerical implementation). Using the LWWR algorithm

TABLE I
THE FAST 2-D IAA ALGORITHM

$$
\begin{aligned}
\mathbf{R}_{N_1(N_2-1)}\boldsymbol{\mathcal{A}}_{N_2-1} &= -\boldsymbol{\mathcal{R}}^{f}_{N_2-1} \quad \text{solve using LWWR} \\
\mathbf{R}^{-1}_{N_1N_2} &= \sum_{i=1}^{2}\sigma_i\boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{T}}^{i}_{N_2})\boldsymbol{\mathcal{L}}^{H}(\boldsymbol{\mathcal{T}}^{i}_{N_2}) \\
\mathbf{d}_{N_1N_2} &= \mathbf{R}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2} \\
\psi(\omega_{k_1},\omega_{k_2}) &= \mathbf{f}^{H}_{N_1N_2}(\omega_{k_1},\omega_{k_2})\mathbf{d}_{N_1N_2} \\
\varphi(\omega_{k_1},\omega_{k_2}) &= \mathbf{f}^{H}_{N_1N_2}(\omega_{k_1},\omega_{k_2})\mathbf{R}^{-1}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2}) \\
\alpha(\omega_{k_1},\omega_{k_2}) &= \frac{\psi(\omega_{k_1},\omega_{k_2})}{\varphi(\omega_{k_1},\omega_{k_2})} \\
\mathbf{D}_{K_1K_2} &= \operatorname{diag}\left\{|\alpha(\omega_0,\omega_0)|^2,\ldots,|\alpha(\omega_{K_1-1},\omega_{K_2-1})|^2\right\} \\
\mathbf{C}_{K_1K_2} &= \mathbf{W}^{H}_{K_1K_2}\mathbf{D}_{K_1K_2}\mathbf{W}_{K_1K_2} \\
\mathbf{C}_{K_1K_2} &= \mathbf{S}_{K_1K_2}\begin{bmatrix}\mathbf{R}_{N_1N_2} & \times \\ \times & \times\end{bmatrix}\mathbf{S}^{T}_{K_1K_2}
\end{aligned}
$$

for the computation of the displacement representation of (32), $\mathbf{d}_{N_1N_2}$ may be computed at a cost of approximately $2N_1^2\phi(2N_2)+8N_1^2N_2$ operations, where $\phi(2N_2)$ denotes the cost of performing a 1-D FFT (IFFT) of size $2N_2\times 1$, since the GS factorization (32) involves products of block lower Toeplitz matrices, thus allowing for a fast matrix vector multiplication using the FFT. Finally, the coefficients of the 2-D trigonometric polynomial $\varphi(\omega_{k_1},\omega_{k_2})$ that appears in the denominator of (9) are computed using the GS representation in (32) and the fast scheme developed in [27]. The resulting fast IAA (FIAA) implementation, presented independently in [24], [25], and for the readers convenience summarized in Table I, requires roughly

$$
\begin{aligned}
C^{FIAA} \approx m_I\big[&1.5N_2^2N_1^3+4N_1^2\phi(2N_2)+ \\
&8N_1^2N_2+5N_1\phi(2N_1,2N_2)+3\phi(K_1,K_2)\big], \qquad (33)
\end{aligned}
$$

operations, where m_I is the number of 2-D IAA iterations and $\phi(K_1,K_2)$ denotes the cost of performing a 2-D FFT (IFFT) of size $K_1\times K_2$.

Motivated by the segmented IAA (SIAA) algorithm introduced in [21], which allows for a trade-off between variance and bias of the estimate, we proceed to develop a fast 2-D SIAA algorithm, wherein the 2-D data set is divided into L possible overlapping segments of size $S_1\times S_2$ each, with $S_1<N_1$ and $S_2<N_2$. The 2-D SIAA algorithm is then formed by iterating

$$
\begin{aligned}
\alpha^{(\ell)}(\omega_{k_1},\omega_{k_2}) &= \frac{\mathbf{f}^{H}_{S_1S_2}(\omega_{k_1},\omega_{k_2})\mathbf{R}^{-1}_{S_1S_2}\mathbf{y}^{(\ell)}_{S_1S_2}}{\mathbf{f}^{H}_{S_1S_2}(\omega_{k_1},\omega_{k_2})\mathbf{R}^{-1}_{S_1S_2}\mathbf{f}_{S_1S_2}(\omega_{k_1},\omega_{k_2})} \\
&\triangleq \frac{\psi^{(\ell)}(\omega_{k_1},\omega_{k_2})}{\varphi(\omega_{k_1},\omega_{k_2})}, \ell=1,2\ldots L \qquad (34) \\
\mathbf{R}_{S_1S_2} &= \mathbf{F}_{S_1S_2,K_1K_2}\mathbf{D}^{s}_{K_1K_2}\mathbf{F}^{H}_{S_1S_2,K_1K_2} \qquad (35)
\end{aligned}
$$

until practical convergence, with

$$
\mathbf{D}^{s}_{K_1K_2} \triangleq \operatorname{diag}\left\{\Phi^{s}(\omega_0,\omega_0)\ldots\Phi^{s}(\omega_{K_1-1},\omega_{K_2-1})\right\},
$$

where

$$
\Phi^{s}(\omega_{k_1},\omega_{k_2}) = \frac{1}{L}\sum_{\ell=1}^{L}|\alpha^{(\ell)}(\omega_{k_1},\omega_{k_2})|^2 \qquad (36)
$$

is the averaged spectra at $(\omega_{k_1},\omega_{k_2})$ over all segments, with $\mathbf{y}^{(\ell)}_{S_1S_2} \triangleq \operatorname{vec}\left\{\mathbf{Y}^{(\ell)}_{S_1S_2}\right\}$ denoting the vectorized data corresponding to the ℓ-th image segment $\mathbf{Y}^{(\ell)}_{S_1S_2}$ of size $S_1\times S_2$. The 2-D SIAA can be efficiently implemented using a similar approach as in the case of the 2-D FIAA algorithm, although in this case, the vectors $\mathbf{d}^{\ell}_{S_1S_2} \triangleq \mathbf{R}^{-1}_{S_1S_2}\mathbf{y}^{\ell}_{S_1S_2}$, for $\ell=1,2\ldots L$, that appear in the numerator of (34) are computed using the GS factorization of $\mathbf{R}^{-1}_{S_1S_2}$. Thus, the overall complexity of the 2-D fast SIAA (FSIAA) algorithm, including the dominant factors only, is given by

$$
\begin{aligned}
C^{FSIAA} = m_{SI}\big[&1.5S_2^2S_1^3+5S_1\phi(2S_1,2S_2)+4S_1^2\phi(2S_2) \\
&6LS_1\phi(2S_2)+(2L+1)\phi(K_1,K_2)\big],
\end{aligned}
$$

where m_{SI} is the number of 2-D SIAA iterations.

The above discussed 2-D FIAA and FSIAA implementations form exact implementation of the corresponding brute force algorithms, and although substantially faster than the brute force implementations, these can still be computationally prohibitive for some applications. For this reason, we recently introduced a fast approximative CG-based 1-D IAA algorithm in [23]. This algorithm has also been extended to a block-recursive (1-D) formulation applied to blood velocity estimation in ultrasound imaging [34]. Here, we further extend on this work, allowing the algorithm to also handle 2-D data sets. The resulting implementation is substantially more efficient than even the above fast implementations, without having more than a marginal effect in the accuracy of the estimated parameters. The implementation is motivated by the Quasi-Newton (QN) algorithm formulated in [35], and then further developed in [36]–[39], 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. In the spectral estimation case, this concept may be exploited by constructing a 2-D QN approximation of the covariance matrix by extrapolating a lower order, incomplete, solution of the 2-D linear system for the full size data matrix under consideration. The thus proposed 2-D QN-IAA algorithm implicitly estimates the approximative covariance matrix $\mathbf{Q}_{N_1N_2}$ in place of $\mathbf{R}_{N_1N_2}$, such that

$$
\mathbf{Q}^{-1}_{N_1N_2} = \begin{bmatrix}\mathbf{0} & \mathbf{0}^{T} \\ \mathbf{0} & \mathbf{R}^{-1}_{N_1M_2}\end{bmatrix} + \mathbf{A}_{2D}\mathbf{A}^{H}_{2D} \qquad (37)
$$

where $\mathbf{A}_{2D} \triangleq [\boldsymbol{\mathcal{A}}^{Q}_{N_2}\ \mathbf{Z}_{N_2}\boldsymbol{\mathcal{A}}^{Q}_{N_2}\ \ldots\mathbf{Z}^{N_2-M_2}_{N_2}\boldsymbol{\mathcal{A}}^{Q}_{N_2}]$ with $\boldsymbol{\mathcal{A}}^{Q}_{N_2} = [\bar{\boldsymbol{\mathcal{A}}}^{T}_{M_2}\ \mathbf{0}^{T}_{(N_2-M_2)N_1}]^{T}$, where $\mathbf{A}_{2D}$ is a block Toeplitz matrix of block size $N_2\times(N_2-M_2)$ with matrix entries of size $N_1\times N_1$. Here, (37) results from an incomplete 2-D LWWR algorithm where, by construction, the 2-D forward and backward matrix valued reflection coefficient are set equal to zero, $\mathbf{K}^{(\ell)f}_{N_1}=\mathbf{0}$ and $\mathbf{K}^{(\ell)b}_{N_1}=\mathbf{0}$, for $\ell=M_2+1,M_2+2,\ldots,N_2$. Thus, an approximate 2-D IAA algorithm can be derived by the direct use of the matrix $\mathbf{Q}_{N_1N_2}$ in place of $\mathbf{R}_{N_1N_2}$ that appears in (9). Since the inverse $\mathbf{Q}^{-1}_{N_1N_2}$ is already available, no further computations are required for this purpose. Thus, the resulting

approximate 2-D IAA algorithm is formed by iterating

$$\begin{aligned}\tilde{\alpha}(\omega_{k_1},\omega_{k_2}) &= \frac{\mathbf{f}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\mathbf{Q}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}}{\mathbf{f}^H_{N_1N_2}(\omega_{k_1},\omega_{k_2})\mathbf{Q}^{-1}_{N_1N_2}\mathbf{f}_{N_1N_2}(\omega_{k_1},\omega_{k_2})} \\ &\triangleq \frac{\tilde{\psi}(\omega_{k_1},\omega_{k_2})}{\tilde{\varphi}(\omega_{k_1},\omega_{k_2})} \qquad (38)\\ \mathbf{R}_{N_1M_2} &= \mathbf{F}_{N_1M_2,K_1K_2}\tilde{\mathbf{D}}_{K_1K_2}\mathbf{F}^H_{N_1M_2,K_1K_2} \qquad (39)\end{aligned}$$

until practical convergence, where

$$\tilde{\mathbf{D}}_{K_1K_2} \triangleq \mathrm{diag}\left\{|\hat{\alpha}(\omega_0,\omega_0)|^2 \ldots |\hat{\alpha}(\omega_{K_1-1},\omega_{K_2-1})|^2\right\}.$$

The resulting 2D QN-IAA algorithm can be implemented efficiently using the techniques developed above, although the variable $\tilde{\mathbf{d}}_{N_1N_2} \triangleq \mathbf{Q}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}$ that appears in the numerator of (38) is now computed using (37) as

$$\tilde{\mathbf{d}}_{N_1N_2} = \begin{bmatrix}\mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R}^{-1}_{N_1M_2}\end{bmatrix}\mathbf{y}_{N_1N_2} + \mathbf{A}_{2D}\mathbf{A}^H_{2D}\mathbf{y}_{N_1N_2} \qquad (40)$$

which can efficiently be implemented using the GS factorization of $\mathbf{R}^{-1}_{N_1M_2}$ and the fact that the matrix $\mathbf{A}_{2D}$ is block Toeplitz, at a cost of $2N_1^2\phi(2M_2) + 8N_1^2M_2 + N_1^2\phi(2M_2) + 2N_1^2N_2$ operations. Moreover, $\tilde{\varphi}(\omega_{k_1},\omega_{k_2})$ that appears in the denominator of (38) can be expressed as

$$\begin{aligned}\tilde{\varphi}(\omega_{k_1},\omega_{k_2}) &= \mathbf{f}^H_{N_1M_2}(\omega_{k_1},\omega_{k_2})\mathbf{Q}^{-1}_{N_1M_2}\mathbf{f}_{N_1M_2}(\omega_{k_1},\omega_{k_2}) + \\ &\quad (N_2-M_2)\left|\mathbf{f}^H_{N_1M_2}(\omega_{k_1},\omega_{k_2})\bar{\mathcal{A}}_{M_2}\right|^2 \qquad (41)\end{aligned}$$

allowing for a reduction in the computational cost for the estimation of the coefficients of the 2D trigonometric polynomial. The overall computational complexity of the 2-D QN-FIAA spectral estimation algorithm is given by

$$\begin{aligned}C^{QN-FIAA} \approx m\big[&1.5N_1^3M_2^2 + 2N_1^2\phi(2M_2) + 8N_1^2M_2 + \\ &N_1^2\phi(2M_2) + 2N_1^2N_2 + 2N_1^2\phi(2M_2) + 2N_1^2\phi(N_1) + \\ &5N_1\phi(2N_1,2M_2) + 3\phi(K_1,K_2)\big], \qquad (42)\end{aligned}$$

where m is the number of 2-D QN-IAA iterations. Further computational savings can be achieved by noting that, in most situations $M_2 \ll N_1$, implying that the application of suitable permutations allows us to construct an equivalent matrix defined as $\tilde{\mathbf{R}}_{M_2N_1} = \boldsymbol{\mathcal{S}}\mathbf{R}_{N_1M_2}\boldsymbol{\mathcal{S}}^T$, where $\boldsymbol{\mathcal{S}}$ is a permutation matrix and $\mathbf{R}_{M_2N_1}$ is a block Toeplitz matrix with block size $N_1 \times N_1$, having Toeplitz matrix entries of size $M_2 \times M_2$. Thus, $\mathbf{R}^{-1}_{N_1M_2} = \boldsymbol{\mathcal{S}}^T\tilde{\mathbf{R}}^{-1}_{M_2N_1}\boldsymbol{\mathcal{S}}$, implying that the generators of the inverse matrix may be obtained using the LWWR algorithm applied to block matrices with entries of size $M_2 \times M_2$. The new generators are thus "thinner" than the previous ones, having width M_2 instead of N_2, and thus allowing for a more efficient implementation, requiring only $1.5M_2^3N_1^2$ operations, which is notably less than the earlier required $1.5N_1^3M_2^2$ operations, especially when $M_2 \ll N_1$. The block GS representation in (32) is then restructured accordingly to accommodate thinner block matrices, implying that the first product in (40) can be organized as

$$\begin{bmatrix}\mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{\mathcal{S}}^T\tilde{\mathbf{R}}^{-1}_{M_2N_1}\boldsymbol{\mathcal{S}}\end{bmatrix}\mathbf{y}_{N_1M_2}$$

resulting in a cost of $2M_2^2\phi(2N_1) + 8M_2^2N_1$ operations. Finally, we examine how the forward predictor required in the construction of $\bar{\boldsymbol{\mathcal{A}}}_{M_2}$ can be computed. Recall that

$$\mathbf{R}_{N_1M_2}\hat{\boldsymbol{\mathcal{A}}}_{M_2} = \begin{bmatrix}\mathbf{I}_{N_1} \\ 0\end{bmatrix} \qquad (43)$$

or

$$\tilde{\mathbf{R}}_{M_2N_1}\boldsymbol{\mathcal{S}}\hat{\boldsymbol{\mathcal{A}}}_{M_2} = \boldsymbol{\mathcal{S}}\begin{bmatrix}\mathbf{I}_{N_1} \\ 0\end{bmatrix} \qquad (44)$$

which implies that

$$\hat{\boldsymbol{\mathcal{A}}}_{M_2} = \boldsymbol{\mathcal{S}}^T\tilde{\mathbf{R}}^{-1}_{M_2N_1}\boldsymbol{\mathcal{S}}\begin{bmatrix}\mathbf{I}_{N_1} \\ 0\end{bmatrix} \qquad (45)$$

Given the GS representation of $\tilde{\mathbf{R}}_{M_2N_1}$, this step can be accomplished in $4N_1^2M_2^2 + 3M_2N_1\phi(2N_1)$ operations, and finally $\bar{\boldsymbol{\mathcal{A}}}_{M_2} = \hat{\boldsymbol{\mathcal{A}}}_{M_2}[\mathbf{A}^f_{N_1}]^{1/2H}$, where $[\mathbf{A}^f_{N_1}]^{-1} = \hat{\boldsymbol{\mathcal{A}}}_{M_2}[1]$, with $[\mathbf{A}^f_{N_1}]^{1/2}$ denoting the Cholesky factor of $\mathbf{A}^f_{N_1}$. Then, in total, the computational complexity of this alternative 2-D QN IAA implementation is

$$\begin{aligned}C^{QN-FIAA} \approx m\big[&1.5M_2^3N_1^2 + N_1^3M_2 + 4N_1^2M_2^2 + \\ &3M_2N_1\phi(2N_1) + 2N_1^2\phi(N_1) + 5N_1\phi(2N_1,2M_2) + 3\phi(K_1,K_2)\big].\end{aligned}$$

Summarizing our analysis, the fast implementation of the 2-D IAA algorithms discussed so far may be computed at a cost of approximately

$$\begin{aligned}\mathcal{C}_1^{FIAA} &\approx 1.5N_1^3N_2^2 + 1.5K_1K_2\log_2(K_1K_2) \\ \mathcal{C}_1^{QNFIAA-I} &\approx 1.5N_1^3M_2^2 + 1.5K_1K_2\log_2(K_1K_2) \\ \mathcal{C}_1^{QNFIAA-II} &\approx 1.5N_1^2M_2^3 + 1.5K_1K_2\log_2(K_1K_2)\end{aligned}$$

operations per IAA iteration. To get some insight into the above expressions, consider the special, yet common, case where $N \triangleq N_1 = N_2$, $K \triangleq K_1 = K_2$, and let $M_2 = N_2/3$, which result in $\mathcal{C}_1^{FIAA} \approx 1.5N^5 + 3K^2\log_2(K)$, $\mathcal{C}_1^{QNFIAA-I} \approx 1/6N^5 + 3K^2\log_2(K)$ and $\mathcal{C}_1^{QNFIAA-II} \approx 1/18N^5 + 3K^2\log_2(K)$, respectively. On the other hand, the computational complexity of the 2-D SFIAA algorithm with $L = 5$ overlapped segments, each of size equal to the one fourth of original image, i.e., $S_1 = S_2 = N/2$, reduces to $\mathcal{C}_1^{SFIAA} \approx 3N^5/64 + 11K^2\log_2(K)$. It is worth noting that the steps of the algorithm can to a large extent be parallelized. The algorithm essentially requires two basic processing components that solves the 2-D TBT linear system and computes the 2-D FFT. The latter can be implemented in parallel using a bank of 1-D FFT units, while the former, which is by far the most computational demanding unit of the algorithm, allows for a parallel implementation using a Schur-type implementation [40]–[44]. Such an implementation avoids the inner product computations inherently involved in the LWWR recursions, thereby allowing for a parallel implementation with locally recursive algorithms via a transformation using a canonical mapping methodology [45]–[47], allowing for an efficient array implementation via a systolic or a wavefront architecture. The reader is referred to [40]–[49] for a further discussion on these aspects.

TABLE II
SUMMARY OF THE FAST 2-D SLIM ALGORITHM

$$
\begin{aligned}
\mathbf{P}_{K_1K_2} &= \text{diag}\left\{|\beta(\omega_0,\omega_0)|^{q-2} \ \ldots \ |\beta(\omega_{K_1-1},\omega_{K_2-1})|^{q-2}\right\} \\
\mathbf{C}^o_{K_1K_2} &= \mathbf{W}^H_{K_1K_2}\mathbf{P}_{K_1K_2}\mathbf{W}_{K_1K_2} \\
\mathbf{C}_{K_1K_2} &= \mathbf{S}_{K_1K_2}\begin{bmatrix}\mathbf{\Sigma}^o_{N_1N_2} & \times \\ \times & \times\end{bmatrix}\mathbf{S}^T_{K_1K_2} \\
\mathbf{\Sigma}_{N_1N_2} &= \mathbf{\Sigma}^o_{N_1N_2} + \eta\mathbf{I}_{N_1N_2} \\
\mathbf{r}_{N_1N_2} &= \mathbf{y}_{N_1N_2} - \mathbf{\Sigma}_{N_1N_2}\mathbf{d}_{N_1N_2} \\
\rho_0 &= |\mathbf{r}_{N_1N_2}|^2, \quad \kappa = 1
\end{aligned}
$$

while $\sqrt{\rho_k} > \epsilon|\mathbf{y}_{N_1N_2}|^2$ and $\kappa < \kappa_{max}$

$$
\begin{aligned}
\gamma &= \rho_{\kappa-1}/\rho_{\kappa-2} \\
\mathbf{p}_{N_1N_2} &= \mathbf{r}_{N_1N_2} + \gamma\mathbf{p}_{N_1N_2} \\
\mathbf{w}_{N_1N_2} &= \mathbf{\Sigma}_{N_1N_2}\mathbf{p}_{N_1N_2} \\
\delta &= \rho_{\kappa-1}/(\mathbf{p}^H_{N_1N_2}\mathbf{w}_{N_1N_2} \\
\mathbf{d}_{N_1N_2} &= \mathbf{d}_{N_1N_2} + \delta\mathbf{p}_{N_1N_2} \\
\mathbf{r}_{N_1N_2} &= \mathbf{r}_{N_1N_2} - \delta\mathbf{w}_{N_1N_2} \\
\rho_\kappa &= |\mathbf{r}_{N_1N_2}|^2, \quad \kappa = \kappa + 1
\end{aligned}
$$

endwhile

$$
\begin{aligned}
\boldsymbol{\beta}_{K_1K_2} &= \mathbf{P}_{K_1K_2}\mathbf{F}^H_{N_1N_2,K_1K_2}\mathbf{d}_{N_1N_2} \\
\eta &= \frac{1}{N_1N_2}\left|\mathbf{y}_{N_1N_2} - \mathbf{F}^H_{N_1N_2,K_1K_2}\boldsymbol{\beta}_{K_1K_2}\right|^2
\end{aligned}
$$

IV. FAST COMPUTATION OF THE 2-D SLIM ALGORITHM

Brute force implementation of the SLIM algorithm described by (15)-(17) requires approximately $N_1^3N_2^3 + N_1^2N_2^2K_1K_2$ operations per iteration, with usually no more than 10-15 iterations necessary for convergence. Fortunately, this often prohibitive computational burden may be substantially reduced when the parameters sought correspond to a uniformly spaced grid of 2-D frequencies. Indeed, the calculations in (15)-(17) are very similar to those of the IAA algorithm described in the previous section, although the SLIM iterations are actually simpler due to the absence of the trigonometric polynomial that appears in the denominator of (9) for the IAA algorithm. As (16) is a TBT matrix whose elements can be efficiently computed using TBT to CBC embedding and the 2-D FFT, and $\boldsymbol{\sigma}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2}$ that appears in (15) can be computed by means of the LWWR algorithm, the resulting computational cost per iteration can be reduced to approximately

$$\mathcal{C}^{FSLIM} \approx 1.5N_1^3N_2^2 + 1.5K_1K_2\log_2(K_1K_2)$$

operations. As it has been pointed out in [17], [22], this figure can be further reduced by using an iterative CG-based linear solver instead of the LWWR algorithm for the solution on the TBT linear system involved in (15). This is particularly suitable as the SLIM algorithm only requires the solution of the TBT linear system (in contrast to the IAA case where in addition the displacement representation of the TBT matrix is also needed) and as the CG algorithm may be expected to converge much faster than it is anticipated by the dimensionality of the TBT matrix, since, due to the assumptions of the line spectral model under consideration, the rank of (16) can be expected to be relatively small. The resulting 2-D CG-SLIM algorithm is summarized in Table II, having a computational complexity of approximately

$$\mathcal{C}^{FSLIM-CG} \approx \kappa_{CG}\left(5N_1N_2 + 2\phi(2N_1,2N_2)\right) +$$

$$3\phi(K_1,K_2) + \phi(2N_1,2N_2) \approx \kappa_{CG}4N_1N_2\log_2(4N_1N_2)$$

operations per SLIM iteration, provided that the TBT vector multiplications are computed using circular embedding and the 2-D FFT, with κ_{CG} denoting the number of CG iterations required for convergence. Compared to the fast CG 2-D SLIM implementation presented in [17], [22] and where fast TBT computations are organized using circular embedding of size equal to the size of the 2-D frequency grid, resulting in a complexity approximately estimated as $\kappa_{CG}K_1K_2\log_2(K_1K_2)$, the proposed approach is faster, yet mathematically equivalent. For a typical 2-D spectral estimation scenario, where $N \triangleq N_1 = N_2$ and $K \triangleq K_1 = K_2$, and where $K = 5N$, the proposed implementation requires about 6 times less computations than that of the previously presented approach; the larger K is compared to N, the larger the gain is, with the overall complexity of the fast 2-D SLIM implementations being about

$$
\begin{aligned}
\mathcal{C}_1^{FSLIM-GS} &\approx 1.5N^5 + 1.5K^2\log_2(K) \\
\mathcal{C}_1^{FSLIM-CG} &\approx \kappa_{CG}8N^2\log_2(4N) + 3K^2\log_2(K)
\end{aligned}
$$

per SLIM iteration.

V. HYBRID SPECTRAL ESTIMATION SCHEMES

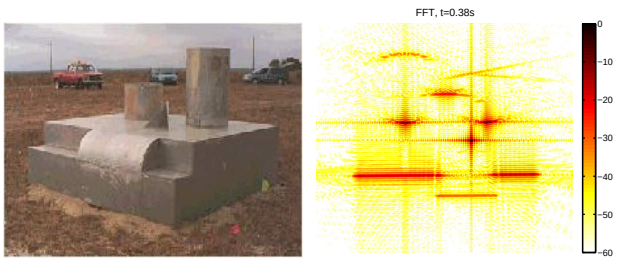
The aforementioned spectral estimation methods possess various merits and limitations. Below, we consider several hybrid methods that take advantages of these merits while overcoming the limitations of the separate methods. First, we note that the 2-D IAA algorithms will provide a non-parametric, robust, and user parameter free algorithm, which has also been found to be more accurate than the corresponding SLIM estimates, although with a notably higher sidelobe level (see also [22]). In order to achieve sidelobe levels comparable to those of SLIM, one may instead form a combined approach that first apply the 2-D IAA estimate to compute a dense spectral estimate, which is then, upon convergence, followed by a refinement stage formed as

$$
\begin{aligned}
\hat{\alpha}(\omega_{k_1,k_2}) &\triangleq |\alpha(\omega_{k_1},\omega_{k_2})|^2\mathbf{f}^H_{K_1K_2}(\omega_{k_1},\omega_{k_2})\mathbf{R}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2} \\
&= |\alpha(\omega_{k_1},\omega_{k_2})|^2\psi(\omega_{k_1},\omega_{k_2}) \qquad (46)
\end{aligned}
$$

Since SLIM achieves sparsity based on solving a hierarchical Bayesian model through maximizing the a posteriori probability density function (MAP), this ad hoc step is referred to as a MAP step, and the resulting algorithm as the 2-D IAA-MAP algorithm. Using similar arguments, one may similarly form the 2-D QN-IAA-MAP algorithm by instead using (38)-(39) followed by the MAP step

$$
\begin{aligned}
\hat{\tilde{\alpha}}(\omega_{k_1,k_2}) &\triangleq |\tilde{\alpha}(\omega_{k_1},\omega_{k_2})|^2\mathbf{f}^H_{K_1K_2}(\omega_{k_1},\omega_{k_2})\mathbf{Q}^{-1}_{N_1N_2}\mathbf{y}_{N_1N_2} \\
&= |\tilde{\alpha}(\omega_{k_1},\omega_{k_2})|^2\tilde{\psi}(\omega_{k_1},\omega_{k_2}) \qquad (47)
\end{aligned}
$$

The computational effort of performing the MAP step in both cases is negligible, and the resulting algorithms can thus be implemented at cost given by (33) and (42), respectively.



(a) (b)

Fig. 1. 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 data matrix.

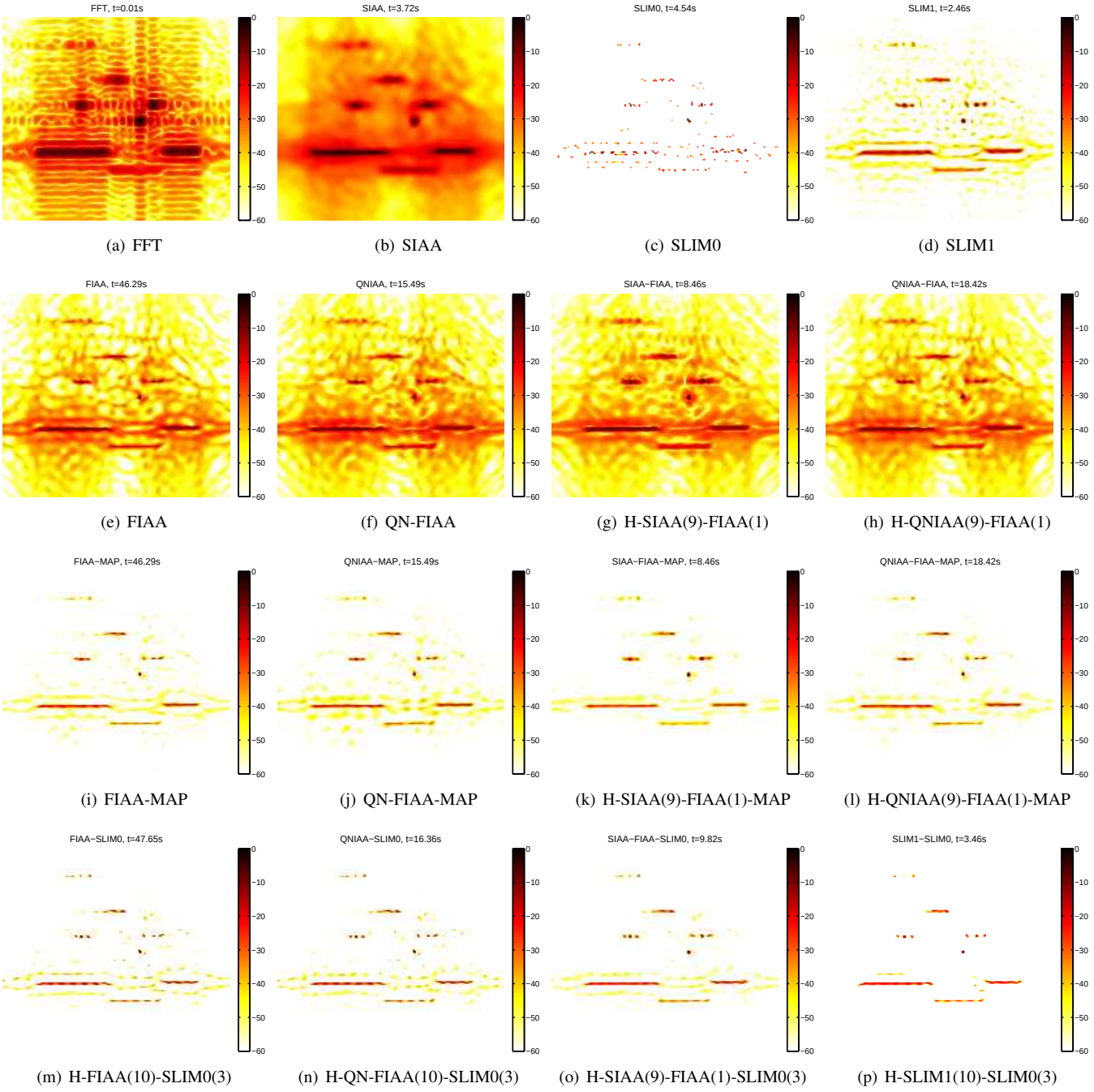
Alternatively, one may note that the 2-D IAA algorithms, being initialized by setting the data covariance matrix $\mathbf{R}_{N_1N_2} = \mathbf{I}_{N_1N_2}$ will result in an initial spectrum which is identical to the (scaled and zero padded) 2-D DFT of the data vector $\mathbf{y}_{N_1N_2}$. The resulting low resolution and high sidelobe levels may slow down the 2-D IAA algorithm, necessitating several iterations to achieve convergence, while at the same time requiring unnecessary high level of computations in the earlier iterations. This drawback can be circumvented if a more accurate and relatively cheap spectral estimate is used in place of the 2-D DFT during the initialization of the IAA. Since both the 2-D SFIAA and the 2-D QN-FIAA are less expensive than the 2-D FIAA algorithm, while at the same time are capable of producing spectra of higher quality than that of the 2-D DFT method, a reasonable hybrid method may be formed by exploiting these cheaper and somewhat less accurate estimators, followed by more expensive and accurate estimators at the latter iterations. We term the method combing m_{si} iterations of the 2-D SFIAA algorithm followed by m_i iterations of the 2-D FIAA algorithm the 2-D H-SFIAA(m_{si})FIAA(m_i) scheme. Similarly, the 2-D H-QNFIAA(m_{qi})FIAA(m_i) scheme may be formed by instead using the 2-D QN-FIAA during the earlier m_{qi} iterations. Finally, we note that the 2-D FIAA, the 2-D SFIAA or the 2-D QN-FIAA can be used for the initialization of the 2-D SLIM recursions in (15)-(17), in place of (18), resulting in similar hybrid IAA/SLIM spectral estimation schemes.

VI. Numerical and Experimental Examples

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 [50], 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 FFT from a complete 288×288 data matrix are shown in Figures 1(a) and 1(b), respectively. In the following, we examine a lower dimensional subset formed using only the $N_1 = N_2 = 80$ center block of the phase-history data, with $K_1 = K_2 = 400$ uniformly spaced 2-D frequency points. Figure 2 shows the SAR images obtained by the aforementioned spectral estimation techniques, including the FFT-based estimate, using 10 SLIM or IAA iterations for the respective methods, or 9 SFIAA (QN-FIAA) iterations followed by one FIAA iteration for the hybrid IAA schemes, as well as 3 additional SLIM-0 iterations at the conclusion of the 10 iterations of the various algorithms for the hybrid SLIM variants[1]. Furthermore, $L = 5$, $M_2 = 32$, and $\epsilon = 10^{-6}$ for the SFIAA, QN-FIAA and CG-SLIM algorithms, and their corresponding hybrid schemes, respectively. As shown in Figure 2, the FFT is, as expected, found to yield low resolution and high sidelobes, whereas the IAA and SLIM based estimates can be found to result in significantly higher resolution and lower sidelobe levels. As is clear from the figure, the hybrid methods allows for notably sparser estimates, with both the (hybrid) IAA-MAP and hybrid SLIM variants satisfactorily balancing the tradeoff between the image resolution and detail preservation as compared to SLIM-0 and SLIM-1. Table III summarizes the computation times needed by the aforementioned algorithms to form the $K_1 \times K_2$ SAR image from the Slicy data on an ordinary workstation (Intel Xeon E5506 processor 2.13G Hz, 12GB RAM, Windows 7 64-bit, and MATLAB R2010b). As previously mentioned, the hybrid IAA schemes reduce the computation cost significantly with only a slight performance degradation as compared to their FIAA counterpart. It is worth noting that by performing several iterations of SLIM-0 at the conclusion of various algorithms including the (hybrid) IAA algorithms and SLIM-1, the hybrid SLIM schemes can greatly suppress the sidelobe levels without drastically increasing the computation complexities.

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 [51]. The imaging scene consists of numerous civilian vehicles and calibration targets, as shown in Figure 3. 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 Figure 3. 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 $N_1 \times N_2 = 80 \times 80$ block of the spatial data centered about the Chevrolet Malibu is then chipped out and transformed back into k-space using an inverse FFT (IFFT) operation. The discussed spectral estimation techniques are then applied to the so-obtained $N_1 \times N_2$ phase history data to get one $K_1 \times K_2 = 400 \times 400$ image for each subaperture. By using the auxiliary information provided by the GOTCHA data set (e.g., the antenna locations, range to scene center, azimuth and elevation angles), the image is then 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 recon-

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



(a) FFT (b) SIAA (c) SLIM0 (d) SLIM1

(e) FIAA (f) QN-FIAA (g) H-SIAA(9)-FIAA(1) (h) H-QNIAA(9)-FIAA(1)

(i) FIAA-MAP (j) QN-FIAA-MAP (k) H-SIAA(9)-FIAA(1)-MAP (l) H-QNIAA(9)-FIAA(1)-MAP

(m) H-FIAA(10)-SLIM0(3) (n) H-QN-FIAA(10)-SLIM0(3) (o) H-SIAA(9)-FIAA(1)-SLIM0(3) (p) H-SLIM1(10)-SLIM0(3)

Fig. 2. Modulus of the SAR images of the Slicy object obtained from an 80×80 data matrix via FFT, (hybrid) IAA (-MAP), and (hybrid) SLIM variants.

structed Malibu image, whose dimensions are $[5, 15] \times [-10, 0]$ meters with grid size 0.05 meters in both dimensions (i.e., forming a 201×201 image). Figure 4 illustrates the resulting images for the discussed methods, using the above introduced parameter settings, clearly indicating the superior performance of the introduced algorithms as compared to the FFT based approach. As before, the SLIM-0 estimate can be seen to be too sparse to preserve certain vehicle features, whereas the hybrid methods are found to again provide high resolution images with low sidelobe levels. Table III summarizes the computation times needed by the discussed methods to form the reconstructed Malibu images (the running times start from applying the various algorithms to the 80×80 phase history data for each subaperture until the so-obtained 90 subimages are fused to form the final images as shown in Figure 4). Table III illustrates that the computational time ratios of the Gotcha to Slicy data sets for the various algorithms are mostly near or slightly above 90, which is reasonable since, for the Slicy data, one process one 80×80 data matrix whereas for the Gotcha data, one process 90 80×80 data matrices. The exceptions are the ratios for FFT, SLIM-0, SLIM-1 and H-SLIM1-SLIM0. For FFT, the reason for a ratio much higher than 90 is that the time consumed in the image fusion process dominates the overall running time for Gotcha. For the CG-based SLIM variants, the reason for ratios lower than 90 is that the running time depends largely on the number of total

TABLE III

COMPUTATION TIMES FOR SAR IMAGING OF THE SLICY DATA (SECOND COLUMN) AND THE GOTCHA DATA (THIRD COLUMN) AND THE COMPUTATIONAL TIME RATIOS OF GOTCHA TO SLICY (FOURTH COLUMN).

Algorithm	Time (s)	Time (s)	Ratio
FFT	0.01	4.1	412.6
SIAA	3.72	345.3	92.8
SLIM-0	4.54	183.6	40.4
SLIM-1	2.46	155.7	63.3
FIAA (-MAP)	46.29	5111.1	110.4
QN-FIAA (-MAP)	15.49	1507.4	97.3
H-SIAA-FIAA (-MAP)	8.46	802.2	94.8
H-QNIAA-FIAA (-MAP)	18.42	1947.7	105.7
H-FIAA-SLIM0	47.65	5127.0	107.6
H-QN-FIAA-SLIM0	16.36	1539.6	94.1
H-SIAA-FIAA-SLIM0	9.82	902.8	91.9
H-SLIM1-SLIM0	3.46	201.7	58.3

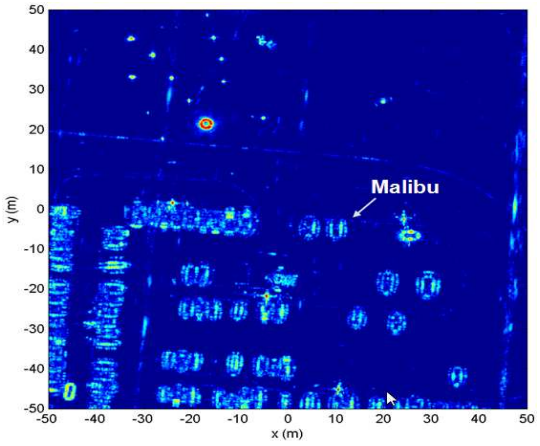


Fig. 3. 2-D SAR image of the GOTCHA scene (from [51]).

CG iterations required for convergence for a given accuracy controlled by the error threshold. Take SLIM-0 for example. The ratio of the running time is around 40 and the total number of CG iterations required by SLIM-0 for Gotcha and Slicy is 24774 and 1171, respectively. This approximate ratio of 21 is significantly lower than 90, which explains the low ratio of running times for SLIM-0.

VII. CONCLUSIONS

In this work, we have presented fast implementations of the 2-D IAA and SLIM 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 degradation. By including a sparsity promoting final step at the conclusion of the iterations, notable sidelobe level reductions are achieved, allowing for a satisfactorily balance between the image resolution and detail preservation. The effectiveness of the algorithms have been verified using both simulated and experimentally measured data sets.

REFERENCES

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

[2] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thomson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*, Kluwer, Norwell, M.A., 1996.

[3] G. R. Benitz, "High-definition vector imaging," *MIT Lincoln Laboratory Journal*, vol. 10, no. 2, pp. 147–170, 1997.

[4] P. . Rosen, S. Hensley, I. R. Joughin, F. K. Li, S. N. Madsen, E. Rodriguez, and R. M. Goldstein, "Synthetic Aperture Radar Interferometry," *Proceedings of the IEEE*, vol. 88, no. 3, pp. 333–382, March 2000.

[5] S. R. DeGraaf, "SAR Imaging via Modern 2-D Spectral Estimation Methods," *IEEE Transactions on Image Processing*, vol. 7, no. 5, pp. 729–761, May 1998.

[6] J. Homer, I. D. Longstaff, and G. Callaghan, "High resolution 3-D SAR via multi-baseline interferometry," in *Proc. IEEE Int. Geosci. Remote Sensing Symp.*, Lincoln, NB, 1996, pp. 796–798.

[7] J. Li and P. Stoica, "An Adaptive Filtering Approach to Spectral Estimation and SAR Imaging," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1469–1484, June 1996.

[8] Z.-S. Liu, R. Wu, and J. Li, "Complex ISAR Imaging of Maneuvering Targets via the Capon Estimator," *IEEE Transactions on Signal Processing*, vol. 47, no. 5, pp. 1262–1271, May 1999.

[9] A. Jakobsson, S. L. Marple, Jr., and P. Stoica, "Two-Dimensional Capon Spectrum Analysis," *IEEE Transactions on Signal Processing*, vol. 48, no. 9, pp. 2651–2661, September 2000.

[10] M. R. Palsetia and J. Li, "Using APES for Interferometric SAR Imaging," *IEEE Transactions on Image Processing*, vol. 7, no. 9, pp. 1340–1353, September 1998.

[11] F. Gini and F. Lombardini, "Multilook APES for Multibaseline SAR Interferometry," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1800–1803, July 2002.

[12] A. Jakobsson, F. Gini, and F. Lombardini, "Robust Estimation of Radar Reflectivities in Multibaseline InSAR," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 2, pp. 751–758, 2005.

[13] M. Cetin and W. C. Karl, "Feature-enhanced synthetic aperture radar image formation based on non-quadratic regularization," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 623–631, 2001.

[14] K. R. Varshney, M. Cetin, J. W. Fisher, and A. S. Willsky, "Sparse representation in structured disctionaries with application to synthetic aperture radar," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3548–3561, 2008.

[15] D. Vu, X. Tan, M. Xue, and J. Li, "A Bayesian Approach to SAR Imaging," *Digit. Signal Process.*, submitted.

[16] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source Localization and Sensing: A Nonparametric Iterative Approach Based on Weighted Least Squares," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 425–443, January 2010.

[17] X. Tan, W. Roberts, J. Li, and P. Stoica, "Sparse Learning via Iterative Minimization With Application to MIMO Radar Imaging," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1088–1101, March 2011.

[18] P. Stoica, J. Li, and J. Ling, "Missing Data Recovery via a Nonparametric Iterative Adaptive Approach," *IEEE Signal Processing Letters*, vol. 16, no. 4, pp. 241–244, April 2009.

[19] P. Stoica, Jian Li, and Hao He, "Spectral Analysis of Nonuniformly Sampled Data: A New Approach Versus the Periodogram," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 843–858, March 2009.

[20] W. Roberts, P. Stoica, J. Li, T. Yardibi, and F. A. Sadjadi, "Iterative Adaptive Approaches to MIMO Radar Imaging," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 5–20, Feb 2010.

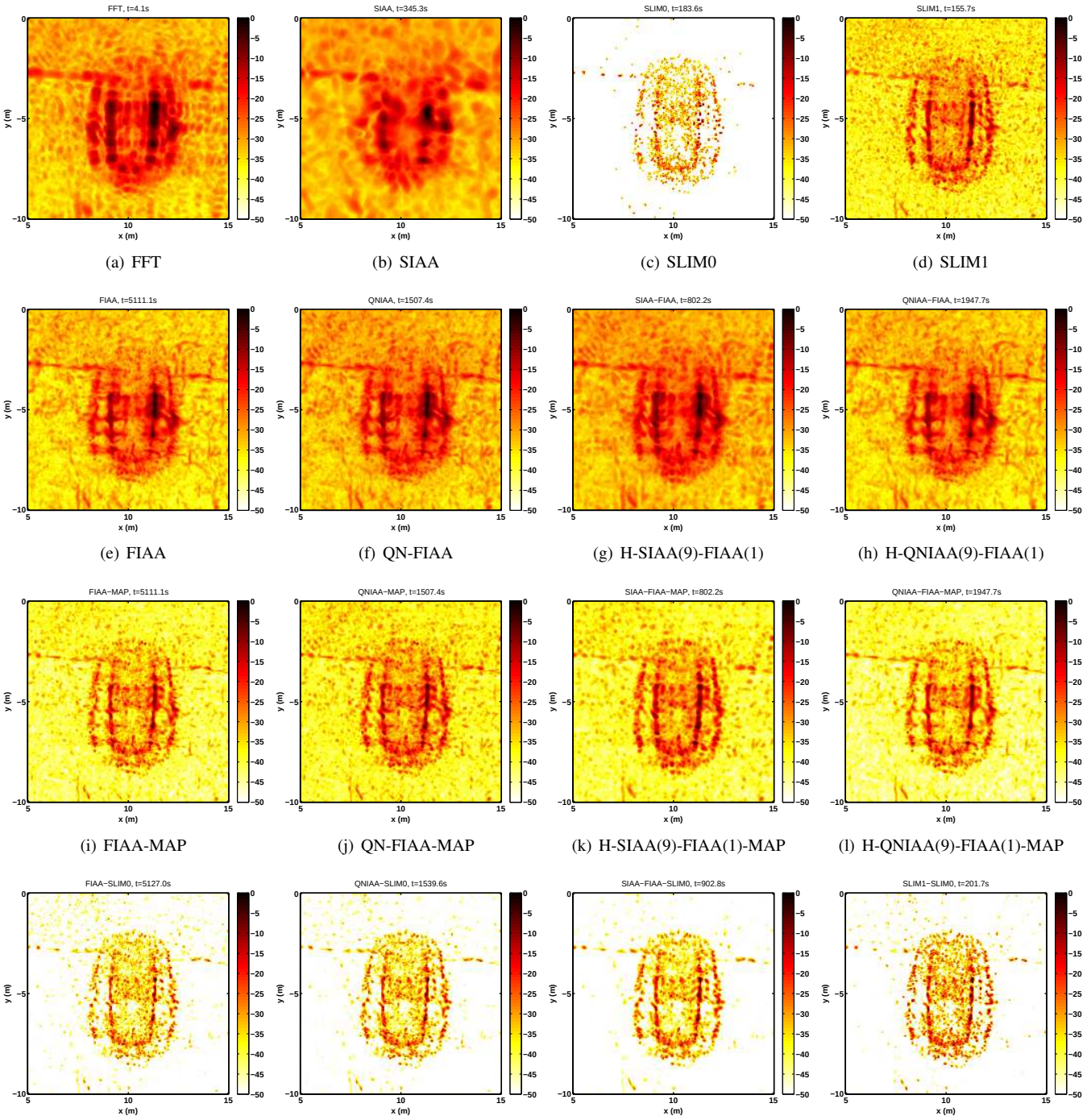
[21] N. R. Butt and A. Jakobsson, "Coherence Spectrum Estimation From Nonuniformly Sampled Sequences," *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 339–342, April 2010.

[22] D. Vu, L. Xu, M. Xue, and J. Li, "Nonparametric Missing Sample Spectral Analysis and Its Applications to Interrupted SAR," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 1, pp. 1–14, Feb. 2012.

[23] G.-O. Glentis and A. Jakobsson, "Time-Recursive IAA Spectral Estimation," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 111–114, Feb. 2011.

[24] G.-O. Glentis and A. Jakobsson, "Efficient Implementation of Iterative Adaptive Approach Spectral Estimation Techniques," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4154–4167, Sept. 2011.

[25] M. Xue, L. Xu, and J. Li, "IAA Spectral Estimation: Fast Implementation using the Gohberg-Semencul Factorization," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3251 – 3261, July 2011.



(a) FFT (b) SIAA (c) SLIM0 (d) SLIM1

(e) FIAA (f) QN-FIAA (g) H-SIAA(9)-FIAA(1) (h) H-QNIAA(9)-FIAA(1)

(i) FIAA-MAP (j) QN-FIAA-MAP (k) H-SIAA(9)-FIAA(1)-MAP (l) H-QNIAA(9)-FIAA(1)-MAP

(m) H-FIAA(10)-SLIM0(3) (n) H-QN-FIAA(10)-SLIM0(3) (o) H-SIAA(9)-FIAA(1)-SLIM0(3) (p) H-SLIM1(10)-SLIM0(3)

Fig. 4. Comparison of the reconstructed Malibu images obtained by FFT, (hybrid) IAA (-MAP), and (hybrid) SLIM variants.

[26] G.-O. Glentis and A. Jakobsson, "Superfast Approximative Implementation of the IAA Spectral Estimate," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 472–478, Jan. 2012.

[27] G.-O. Glentis, "A Fast Algorithm for APES and Capon Spectral Estimation," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4207–4220, Sept. 2008.

[28] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1989.

[29] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, Upper Saddle River, N.J., 2005.

[30] T. Kailath and A. H. Sayed, "Displacement Structure: Theory and Applications," *SIAM Review*, vol. 37, no. 3, pp. 297–386, September 1995.

[31] N. Kalouptsidis, D. Manolakis, and G. Carayannis, "A family of computationally efficient algorithms for multichannel signal processing - a tutorial review," *Signal Process.*, vol. 5, pp. 5–19, 1983.

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

[33] N. Kalouptsidis, G. Carayannis, and D. Manolakis, "Fast Algorithms for Block Toeplitz Matrices with Toeplitz Entries," *Signal Process.*, vol. 6, pp. 77–81, 1984.

[34] A. Jakobsson, G. O. Glentis, and E. Gudmundson, "Computationally Efficient Time-Recursive IAA-Based Blood Velocity Estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, July 2012.

[35] G. V. Moustakides and S. Theodoridis, "Fast Newton Transversal Filters - A New Class of Adaptive Estimation Algorithms," *IEEE Transactions*

on Signal Processing, vol. 39, no. 10, pp. 2184–2193, Oct. 1991.
[36] B. Farhang-Boroujeny, "Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation," *IEEE Transactions on Signal Processing*, vol. 45, no. 8, pp. 1987–2000, August 1997.
[37] B. Farhang-Boroujeny and H. Rao, "Fast LMS/Newton algorithms for stereophonic acoustic echo cancellation," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 2919–2930, Aug. 2009.
[38] A. Gilloire, T. Petillon, and S. Theodoridis, "The fast Newton transversal filter: an efficient scheme for acoustic echo cancellation in mobile radio," *IEEE Transactions on Signal Processing*, vol. 42, no. 3, pp. 509–518, March 1994.
[39] G. Moustakides, S. Theodoridis, and K. Berberidis, "A fast Newton multichannel algorithm for decision feedback equalization," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 327–331, Jan. 1995.
[40] G. O. Glentis and N. Kalouptsidis, "Efficient Algorithms for the Solution of Block Linear Systems With Toeplitz Entries," *Linear Algebra Appl.*, vol. 179, pp. 85–104, 1993.
[41] G. O. Glentis and N. Kalouptsidis, "Efficient solution of block linear systems with Toeplitz entries using a channel decomposition technique," *Signal Processing*, vol. 37, pp. 15–60, 1994.
[42] A. Liavas and S. Theodoridis, "Efficient Levinson- and Schur-type algorithms for block near-to-Toeplitz systems of equations ," *Signal Processing*, vol. 35, pp. 241255, 1994.
[43] I. Bouras, G. O. Glentis, and N. Kalouptsides, "Architectures for block Toeplitz systems," *Signal Processin*, vol. 51, pp. 167–190, 1996.
[44] T. Kailath and A. H. Sayed, *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, USA, 1999.
[45] S. Y. Kung, *VLSI Array Processors*, Prentice Hall, Englewood Cliffs, N. J., 1988.
[46] P. Pirsch, *Architectures for Digital Signal Processing*, Wiley, Chichester, 1998.
[47] K Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, Wiley, New York, 1999.
[48] J. Badia, P. Alonso, and A. Vidal, "An efficient parallel algorithm to solve block-Toeplitz systems," *The Journal of Supercomputing*, vol. 32, pp. 251278, 2005.
[49] J. Badia, P. Alonso, and A. Vidal, "Solving the block-Toeplitz least-squares problem in parallel," *Concurrency and computation: Practice and Experience*, vol. 17, pp. 4967, 2005.
[50] D. J. Andersh, M. Hazlett, S. W. Lee, D. D. Reeves, D. P. Sullivan, and Y. Chu, "XPATCH: a High-Frequency Electromagnetic Scattering Prediction Code and Environment for Complex Three-Dimensional Objects," *IEEE Antennas and Propagation Magazine*, vol. 36, no. 1, pp. 65–69, February 1994.
[51] "Gotcha 2D/3D imaging challenge problem," Air Force Research Lab., Jan. 2010 [Online]. Available: https://www.sdms.afrl.af.mil/datasets/gotcha/.

PLACE PHOTO HERE

George-Othon Glentis (ST'88, M'91) was born in Athens, Greece, on July 24, 1965. He received the B.Sc. degree in Physics in 1987 and the Ph.D degree in Informatics in 1991, both from the University of Athens. From 1988-1991, he held a four years research fellowship from the Institute of Informatics of the National Center for Physical Science, DEMOCRITOS. From 1993-1995, he was with the Electrical Engineering Department of the University of Twente, The Netherlands, and with the Faculté des Sciencés Appliquées, Université Catholique de Louvain, Belgium, as a EU HCM research fellow. From 1996-1997 he was with the Department of Informatics, University of Athens, as a EU TMR research fellow. From 1998- 2005 he was an associate professor at the Department of Electronics, Technological Education Institute of Crete, Chania, Greece. During the summer 2011 he has been with the Spectral Analysis Laboratory, University of Florida, as a visiting Associate Professor. He is currently an associate professor at the Department of Science and Technology of Telecommunications, University of Peloponnese, Tripolis, Greece. His research interests include signal and image processing, and telecommunication applications. He has published over 60 papers in major journals and conferences.

PLACE PHOTO HERE

Kexin Zhao received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, in 2009, and the M.Sc. degree from the University of Florida, Gainesville, in 2011, both in electrical engineering. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Florida, Gainesville. His research interests are in the areas of underwater acoustic communications and active sonar system design.

PLACE PHOTO HERE

Andreas Jakobsson (S'95-M'00-SM'06) received his M.Sc. from Lund Institute of Technology and his Ph.D. in Signal Processing from Uppsala University in 1993 and 2000, respectively. Since, he has held positions with Global IP Sound AB, the Swedish Royal Institute of Technology, King's College London, and Karlstad University, as well as held an Honorary Research Fellowship at Cardiff University. He has been a visiting researcher at King's College London, Brigham Young University, Stanford University, Katholieke Universiteit Leuven, and University of California, San Diego, as well as acted as an expert for the IAEA. He is currently Professor of Mathematical Statistics at Lund University, Sweden. He has published his research findings in over 100 refereed journal and conference papers, and has filed five patents. He has also co-authored (together with M. G. Christensen) a book on multi-pitch estimation (Morgan & Claypool, 2009). He is a member of The Royal Swedish Physiographic Society as well as a Senior Member of IEEE, a member of the IEEE Sensor Array and Multichannel (SAM) Signal Processing Technical Committee, and an Associate Editor for Elsevier Signal Processing and the Journal of Electrical and Computer Engineering. He has previously also been an Associate Editor for the IEEE Transactions on Signal Processing (2006-2010), the IEEE Signal Processing Letters (2007-2011), and the Research Letters in Signal Processing (2007-2009). His research interests include statistical and array signal processing, detection and estimation theory, and related application in remote sensing, telecommunication and biomedicine.

PLACE
PHOTO
HERE

Jian Li (S'87-M'91-SM'97-F'05) received the M.Sc. and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1987 and 1991, respectively.

From April 1991 to June 1991, she was an Adjunct Assistant Professor with the Department of Electrical Engineering, The Ohio State University, Columbus. From July 1991 to June 1993, she was an Assistant Professor with the Department of Electrical Engineering, University of Kentucky, Lexington. Since August 1993, she has been with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, where she is currently a Professor. In Fall 2007, she was on sabbatical leave at MIT, Cambridge, Massachusetts. Her current research interests include spectral estimation, statistical and array signal processing, and their applications.

Dr. Li is a Fellow of IEEE and a Fellow of IET. She is a member of Sigma Xi and Phi Kappa Phi. She received the 1994 National Science Foundation Young Investigator Award and the 1996 Office of Naval Research Young Investigator Award. She was an Executive Committee Member of the 2002 International Conference on Acoustics, Speech, and Signal Processing, Orlando, Florida, May 2002. She was an Associate Editor of the IEEE Transactions on Signal Processing from 1999 to 2005, an Associate Editor of the IEEE Signal Processing Magazine from 2003 to 2005, and a member of the Editorial Board of Signal Processing, a publication of the European Association for Signal Processing (EURASIP), from 2005 to 2007. She has been a member of the Editorial Board of the IEEE Signal Processing Magazine since 2010 and a member of the Editorial Board of Digital Signal Processing – A Review Journal, a publication of Elsevier, since 2006. She is a co-author of the papers that have received the First and Second Place Best Student Paper Awards, respectively, at the 2005 and 2007 Annual Asilomar Conferences on Signals, Systems, and Computers in Pacific Grove, California. She is a co-author of the paper that has received the M. Barry Carlton Award for the best paper published in IEEE Transactions on Aerospace and Electronic Systems in 2005. She is also a co-author of the paper that has received the Lockheed Martin Best Student Paper Award at the 2009 SPIE Defense, Security, and Sensing Conference in Orlando, Florida.