Research Article

Experiment Design Regularization-Based Hardware/Software Codesign for Real-Time Enhanced Imaging in Uncertain Remote Sensing Environment

A. Castillo Atoche,[1, 2] D. Torres Roman,[1] and Y. Shkvarko[1]

[1] *Cinvestav del IPN, Unidad Guadalajara, Guadalajara, Avenida Científica # 1145, Colonia El Bajío, Zapopan, Jalisco, C.P. 45015, Mexico*
[2] *Universidad Autonoma de Yucatan, Av. Industrias No Contaminantes s/n, Col. Cordemex. Merida, Mexico*

Correspondence should be addressed to A. Castillo Atoche, acastill@uady.mx

A new aggregated Hardware/Software (HW/SW) codesign approach to optimization of the digital signal processing techniques for enhanced imaging with real-world uncertain remote sensing (RS) data based on the concept of descriptive experiment design regularization (DEDR) is addressed. We consider the applications of the developed approach to typical single-look synthetic aperture radar (SAR) imaging systems operating in the real-world uncertain RS scenarios. The software design is aimed at the algorithmic-level decrease of the computational load of the large-scale SAR image enhancement tasks. The innovative algorithmic idea is to incorporate into the DEDR-optimized fixed-point iterative reconstruction/enhancement procedure the convex convergence enforcement regularization via constructing the proper multilevel projections onto convex sets (POCS) in the solution domain. The hardware design is performed via systolic array computing based on a Xilinx Field Programmable Gate Array (FPGA) XC4VSX35-10ff668 and is aimed at implementing the unified DEDR-POCS image enhancement/reconstruction procedures in a computationally efficient multi-level parallel fashion that meets the (near) real-time image processing requirements. Finally, we comment on the simulation results indicative of the significantly increased performance efficiency both in resolution enhancement and in computational complexity reduction metrics gained with the proposed aggregated HW/SW co-design approach.

1. Introduction

In this paper, we address a new aggregated Hardware/Software (HW/SW) codesign approach to optimization of the digital signal and image processing techniques as required for enhanced remote sensing (RS) of the environment with the use of high-resolution array radar and synthetic aperture radar (SAR) systems. At the algorithm-design level, the RS imaging problem is treated as an ill-posed nonlinear inverse problem of reconstruction of the spatial spectrum pattern (SSP) of the backscattered field distributed over the remotely sensed scene via processing the SAR data signals distorted in the uncertain stochastic measurement channel [1–6]. The operational scenario uncertainties are attributed to inevitable random signal perturbations in inhomogeneous propagation medium [1, 2, 4], possible imperfect radar/SAR system calibration [1, 3], and SAR carrier trajectory deviations [3, 5, 6]. The unified approach that we address to solve such a problem is based on the recently proposed concept of descriptive experiment design regularization (DEDR) [7, 8]. The general DEDR method constructed in [7, 8] incorporates into the minimum risk (MR) nonparametric estimation strategy [4] the experiment design-motivated constraints of the image identifiably for the discrete-form signal formation operator (SFO) specified by the employed signal modulation format [4–6]. On one hand, a considerable advantage of such DEDR paradigm relates to its flexibility in designing the desirable error metrics in the corresponding image representation space via defining different descriptive cost functions [7, 9]. On the other hand, the crucial limitations of the DEDR method relate to the necessity of performing simultaneously the

solution-dependent SFO inversion operations and adaptive adjustments of the degrees of freedom of the overall DEDR image enhancement techniques ruled by the employed fixed-point iterative process [8]. For the real-world large-scale RS scenes, such adaptive full-format DEDR-optimal method turns out to be computationally extremely consuming, therefore cannot be recommended as a practical technique realizable in (near) real time [10]. The innovative idea of this paper is to aggregate the DEDR-optimal fixed-point iterative reconstruction/enhancement procedures developed in the previous studies [7, 8, 10] with the multi-level robustness and convergence enforcing regularization via constructing the proper projections onto convex sets (POCS) in the solution domain. The established POCS-regularized iterative DEDR technique is performed separately along the range and azimuth directions over the scene frame making an optimal use of the range-azimuth sparseness properties of the employed radar/SAR modulation format. Thus, at the SW codesign stage, we address two conceptually innovative propositions that distinguish our approach from the previous studies [7, 10]. First, two possible observation scenarios (instead of one) are unified now under the DEDR paradigm for the HW/SW codesign, namely, (i) regular case without model uncertainties and (ii) uncertain scenario with random perturbations in the SFO. Second, the POCS regularization is proposed to be performed in an aggregated multi-level fashion to make the optimal use of the non-trivial RS system model information for constructing the corresponding robustness and convergence enforcing POCS operators. In particular, the positivity and range-azimuth orthogonalization projectors of [10] are aggregated with the point spread function (PSF) sparseness enforcing sliding window projectors acting in parallel over both range and azimuth image frames that set the corresponding PSF pixel values to zeroes outside their specified support regions. Such aggregated POCS regularization drastically speeds up the resulting fixed-point iterative DEDR techniques making them exactly well fitted for the systolic computational implementation form; that is, provides the SW algorithmic base for the further HW codesign level of the problem treatment.

At the HW codesign stage, we propose to pursue the System-on-Chip (SoC) single Field Programmable (FP) unit integration approach [9–14], which allows efficient coupling/integration of a number of predetermined complex components. Such a programmable unit is a viable solution for rapid prototyping and digital implementation of the radar/SAR image enhancement techniques developed at the SW codesign stage, in spite of designing the process in a common personal computer (PC) [11–14]. The main advantage of the proposed FPSoC platform is that all required component designs, including the embedded processor unit, memory, and peripherals are algorithmically "adapted" for the particular developed POCS-regularized iterative fixed-point DEDR image enhancement techniques. Therefore, at the HW design stage, the novel contribution of this study is twofold: first, the addressed HW/SW codesign methodology is aimed at an HW implementation of the developed software using *systolic arrays* as coprocessors units;

second, the proposed systolic-based processing architecture is particularly adapted for computational implementation of the unified DEDR-POCS techniques in a computationally efficient fashion that meets the (near) real-time overall RS imaging system requirements. We resume this study with the analysis of the simulation results related to enhancement of the real-world degraded large-scale SAR imagery (i.e., acquired in uncertain operational scenarios) indicative of the significantly increased reconstruction efficiency gained with the proposed HW/SW codesign approach.

2. Background

2.1. Continuous-Form Problem Model. The general formalism of the RS imaging problem presented in this paper is a structural extension of the problem considered in [4, 7, 8], hence some crucial model elements are repeated for convenience to the reader. Consider a coherent RS experiment in a random medium and the narrowband SAR assumption [1] that enables us to model the extended object backscattered wavefield in the baseband format [3] by imposing its time invariant complex scattering (backscattering) function $e(\mathbf{x})$ in the object image domain (scattering surface) $X \ni \mathbf{x}$. The measurement radar/SAR data field $u(\mathbf{y}) = s(\mathbf{y}) + n(\mathbf{y})$ consists of the echo signals s and additive noise n and is available for observations and recordings within the prescribed time-space observation domain $Y = T \times P$, where $\mathbf{y} = (t, \boldsymbol{\rho})^{\mathrm{T}}$ defines the time(t)-space$(\boldsymbol{\rho})$ points in Y; $t \in T, \boldsymbol{\rho} \in P$; $\mathbf{y} \in Y$. The model of the data field u is defined by specifying the stochastic equation of observation (EO) that in the conventional integral form may be written as [4, 8]

$$
\begin{aligned}
u(\mathbf{y}) &= \left(\widetilde{S}e(\mathbf{x})\right)(\mathbf{y}) + n(\mathbf{y}) \\
&= \int_X \widetilde{S}(\mathbf{y}, \mathbf{x})e(\mathbf{x})d\mathbf{x} + n(\mathbf{y}) \\
&= \int_X S(\mathbf{y}, \mathbf{x})e(\mathbf{x})d\mathbf{x} + \int_X \Delta S(\mathbf{y}, \mathbf{x})e(\mathbf{x})d\mathbf{x} + n(\mathbf{y}).
\end{aligned} \tag{1}
$$

The random kernel $\widetilde{S}(\mathbf{y}, \mathbf{x})$ of the perturbed random signal formation operator (SFO) $\widetilde{S}$ given by (1) defines the signal field formation model. Its mean $S(\mathbf{y}, \mathbf{x})$ is referred to as the nominal modulation law in the data formation channel defined by the time-space modulation of signals employed in a particular imaging radar/SAR system [3], and the variation about the mean $\Delta S(\mathbf{y}, \mathbf{x}) = \mu(\mathbf{y}, \mathbf{x})S(\mathbf{y}, \mathbf{x})$ models the stochastic perturbations of the wavefield at different propagation paths, where $\mu(\mathbf{y}, \mathbf{x})$ represents the zero-mean multiplicative noise that models random propagation perturbations in the medium (the so-called general Rytov model [3, 5, 6]). The fields e, n, u in (1) are assumed to be zero-mean complex valued Gaussian random fields [3]. Next, we assume an incoherent nature of the backscattered field $e(\mathbf{x})$. This is naturally inherent to the RS experiments [1, 3, 5, 6] and leads to the δ-form of the object field correlation function, $R_e(\mathbf{x}_1, \mathbf{x}_2) = b(\mathbf{x}_1)\delta(\mathbf{x}_1 - \mathbf{x}_2)$, where $e(\mathbf{x})$ and $b(\mathbf{x}) = \langle |e(\mathbf{x})|^2 \rangle$ are referred to as the object random complex scattering function and its average power scattering function or spatial spectrum pattern (SSP), respectively.

The problem of enhanced RS imaging is to develop a signal processing method for performing the efficient estimation of the SSP $b(\mathbf{x})$ by processing the available radar/SAR measurements of the data wavefield $u(\mathbf{y})$. Such estimate $\hat{b}(\mathbf{x})$ of the SSP $b(\mathbf{x})$ is referred to as the desired reconstructed RS image of the remotely sensed scene.

2.2. Discrete-Form Problem Model. Now we proceed from the stochastic integral-form EO (1) to its finite-dimensional discrete (vector) form approximation [8]

$$\mathbf{u} = \widetilde{\mathbf{S}}\mathbf{e} + \mathbf{n} = \mathbf{S}\mathbf{e} + \Delta\mathbf{e} + \mathbf{n}, \qquad (2)$$

in which the disturbed SFO matrix

$$\widetilde{\mathbf{S}} = \mathbf{S} + \Delta \qquad (3)$$

is the discrete-form approximation of the integral SFO $\widetilde{S}$ defined by the EO (1), and $\mathbf{e}$, $\mathbf{n}$, $\mathbf{u}$ represent zero-mean vectors composed of the decomposition (sampling) coefficients $\{e_k; k = 1,\ldots,K\}, \{n_m; m = 1,\ldots,M\}$, and $\{u_m; m = 1,\ldots,M\}$, respectively [7]. These vectors are characterized by the correlation matrices: $\mathbf{R_e} = \mathbf{D} = \mathbf{D}(\mathbf{b}) = \mathrm{diag}(\mathbf{b})$ (a diagonal matrix with vector $\mathbf{b}$ at its principal diagonal), $\mathbf{R_n}$, and $\mathbf{R_u} = \langle\widetilde{\mathbf{S}}\mathbf{R_e}\widetilde{\mathbf{S}}^+\rangle_{p(\Delta)} + \mathbf{R_n}$, respectively, where $\langle\cdot\rangle_{p(\Delta)}$ defines the averaging performed over the randomness of Δ characterized by the probability density function $p(\Delta)$ *unknown* to the observer, and superscript $^+$ stands for Hermitian conjugate (conjugate transpose). Vector $\mathbf{b}$ is composed of the elements $b_k = \langle e_k e_k^*\rangle = \langle|e_k|^2\rangle$; $k = 1,\ldots,K$, and is referred to as a K-D vector-form representation of the SSP. The SSP vector $\mathbf{b}$ is associated with the so-called lexicographically ordered image pixels [7, 9]. The corresponding conventional $K_y \times K_x$ rectangular frame-ordered scene image $\mathbf{B} = \{b(k_x, k_y); k_x = 1,\ldots,K_x; k_y = 1,\ldots,K_y\}$ relates to its lexicographically ordered vector-form representation $\mathbf{b} = \{b_k = b(k); k = 1,\ldots,K = K_y \times K_x\}$ via the standard row by row expansion (so-called lexicographical reordering) procedure, $\mathbf{B} = L\{\mathbf{b}\}$ [9]. Note that in the simple case of a certain operational scenario [1, 3, 7], the discrete-form (i.e., matrix-form) SFO $\mathbf{S}$ is assumed to be deterministic, in which case the random perturbation term in (3) is irrelevant, $\Delta = 0$.

The digital enhanced RS imaging problem is formally stated as follows: to reconstruct the scene pixel frame image $\hat{\mathbf{B}}$ via lexicographical reordering $\hat{\mathbf{B}} = L\{\hat{\mathbf{b}}\}$ of the SSP vector estimate $\hat{\mathbf{b}}$ estimated from whatever available discrete measurements of the recorded radar/SAR data $\mathbf{u}$. The reconstructed SSP vector $\hat{\mathbf{b}}$ is an estimate of the second-order statistics of the scattering vector $\mathbf{e}$ observed through the perturbed SFO (3) contaminated with additive noise $\mathbf{n}$ and corrupted also with the signal-dependent multiplicative noise $\Delta\mathbf{e}$, hence, the enhanced RS imaging problem at hand must be qualified and treated as a statistical nonlinear inverse problem *with model uncertainties*. The high-resolution sensing implies formation of the RS image $\hat{\mathbf{B}}$ based on some statistically optimal solution of such an inverse problem *robust* against the problem model uncertainties. In this paper we propose to unify the POCS regularization with the DEDR method originally developed in [7, 8].

3. Unified DEDR Method

3.1. DEDR Strategy for Certain Operational Scenario. In the descriptive statistical formalism, the desired SSP vector $\hat{\mathbf{b}}$ is recognized to be the vector of a principal diagonal of the estimate of the correlation matrix $\mathbf{R_e}(\mathbf{b})$; that is, $\hat{\mathbf{b}} = \{\hat{\mathbf{R}}_e\}_{\mathrm{diag}}$. Thus one can seek to estimate $\hat{\mathbf{b}} = \{\hat{\mathbf{R}}_e\}_{\mathrm{diag}}$ given the data correlation matrix $\mathbf{R_u}$ preestimated empirically via averaging $J \geq 1$ recorded data vector snapshots $\{\mathbf{u}_{(j)}\}$; for example, [7]

$$\mathbf{Y} = \hat{\mathbf{R}}_\mathbf{u} = \underset{j\in J}{\mathrm{aver}}\{\mathbf{u}_{(j)}\mathbf{u}_{(j)}^+\},$$

$$= \frac{1}{J}\sum_{j=1}^{J}\mathbf{u}_{(j)}\mathbf{u}_{(j)}^+, \qquad (4)$$

by determining the solution operator (SO) $\mathbf{F}$ such that

$$\hat{\mathbf{b}} = \{\hat{\mathbf{R}}_e\}_{\mathrm{diag}} = \{\mathbf{F}\mathbf{Y}\mathbf{F}^+\}_{\mathrm{diag}} \qquad (5)$$

where $\{\cdot\}_{\mathrm{diag}}$ defines the vector composed of the principal diagonal of the embraced matrix.

To optimize the search for $\mathbf{F}$ in the *certain* operational scenario, the DEDR strategy was proposed in [7] as

$$\mathbf{F} \longrightarrow \min_{\mathbf{F}}\{\mathfrak{R}(\mathbf{F})\}, \qquad (6)$$

$$\mathfrak{R}(\mathbf{F}) = \mathrm{trace}\{(\mathbf{F}\mathbf{S} - \mathbf{I})\mathbf{A}(\mathbf{F}\mathbf{S} - \mathbf{I})^+\}$$
$$+ \alpha\, \mathrm{trace}\{\mathbf{F}\mathbf{R_n}\mathbf{F}^+\} \qquad (7)$$

that implies the minimization of the weighted sum of the systematic and fluctuation errors in the desired estimate $\hat{\mathbf{b}}$ where the selection (adjustment) of the regularization parameter α and the weight matrix $\mathbf{A}$ provide the additional experiment design degrees of freedom incorporating any descriptive properties of a solution if those are known a priori [3, 7]. It is easy to recognize that the strategy (6) is a structural extension of the statistical minimum risk estimation strategy [4] for the nonlinear spectral estimation problem at hand because in both cases the balance between the gained spatial resolution and the noise suppression in the resulting estimate is to be optimized.

3.2. Extended DEDR Strategy for Uncertain Scenario. To optimize the search for the desired SO $\mathbf{F}$ in the *uncertain* operational scenario with the randomly perturbed SFO (3), the *extended DEDR* strategy was proposed in [8] as

$$\mathbf{F} = \arg\min_{\mathbf{F}}\ \max_{(\|\Delta\|^2)_{p(\Delta)}\leq\delta}\{\mathfrak{R}_{\mathrm{ext}}(\mathbf{F})\}, \qquad (8)$$

$$\text{subject to}\quad \left\langle\|\Delta\|^2\right\rangle_{p(\Delta)} \leq \delta \qquad (9)$$

where the conditioning term (9) represents the worst-case statistical performance (WCSP) regularizing constraint imposed on the unknown second-order statistics $\langle \|\Delta\|^2 \rangle_{p(\Delta)}$ of the random distortion component Δ of the SFO matrix (3), and the DEDR "extended risk" is defined by

$$\mathfrak{R}_{\text{ext}}(\mathbf{F}) = \text{tr}\left\{ \left\langle \left(\mathbf{F}\tilde{\mathbf{S}} - \mathbf{I}\right)\mathbf{A}\left(\mathbf{F}\tilde{\mathbf{S}} - \mathbf{I}\right)^+ \right\rangle_{p(\Delta)} \right\} \\ + \alpha\,\text{tr}\{\mathbf{F}\mathbf{R_n}\mathbf{F}^+\}. \tag{10}$$

where the regularization parameter α and the metrics inducing weight matrix $\mathbf{A}$ compose the processing level "degrees of freedom" of the DEDR method.

To proceed with the derivation of the robust SO (8), in [8], the risk function (10) was next decomposed and evaluated for its the maximum value applying the Cauchy-Schwarz inequality and Loewner ordering [9] of the weight matrix $\mathbf{A} \leq \gamma\mathbf{I}$ with the scaled Loewner ordering factor $\gamma = \min\{\hat{\gamma} : \mathbf{A} \leq \hat{\gamma}\mathbf{I}\}$. With these robustifications [8], the extended DEDR strategy (8) is transformed into the following optimization problem:

$$\mathbf{F} = \longrightarrow \min_{\mathbf{F}}\{\mathfrak{R}_{\Sigma}(\mathbf{F})\} \tag{11}$$

with the *aggregated* DEDR risk function [8]

$$\mathfrak{R}_{\Sigma}(\mathbf{F}) = \text{tr}\left\{(\mathbf{F}\mathbf{S} - \mathbf{I})\mathbf{A}(\mathbf{F}\mathbf{S} - \mathbf{I})^+\right\} \\ + \alpha\,\text{tr}\{\mathbf{F}\mathbf{R}_{\Sigma}\mathbf{F}^+\} \tag{12}$$

where

$$\mathbf{R}_{\Sigma} = \mathbf{R}_{\Sigma}(\beta) \\ = (\mathbf{R_n} + \beta\mathbf{I}), \quad \beta = \frac{\delta}{\alpha} \geq 0. \tag{13}$$

3.3. DEDR-Optimal Solution Operators.
Examining the DEDR strategies (6) and (11) one can deduce that those both are structurally similar and differ only by the definition of the second (i.e., noise) risk component terms in (7) and (12). In the certain operational scenario [5–7], the trace $\{\mathbf{F}\mathbf{R_n}\mathbf{F}^+\}$ for the noise error measure is used, while in the uncertain scenario [8] the augmented measure $\text{tr}\{\mathbf{F}\mathbf{R}_{\Sigma}\mathbf{F}^+\}$ is employed with the diagonal loaded extension (13) of the composite noise correlation matrix $\mathbf{R}_{\Sigma}$. The established structural similarity (the so-called problem model homomorphism [5, 6]) of two DEDR problems (6) and (11) makes it possible to unify the solutions for both scenarios. Doing so, we specify the SOs for both considered operational scenarios, namely:

(1) *SO for certain operational scenario* follows directly from the solution to the optimization problem (6) found in the previous study [7] that results in

$$\mathbf{F} = \mathbf{K}\mathbf{S}^+\mathbf{R_n}^{-1}, \tag{14}$$

where

$$\mathbf{K} = (\mathbf{S}^+\mathbf{R_n}^{-1}\mathbf{S} + \alpha\mathbf{A}^{-1})^{-1} \tag{15}$$

represents the so-called regularized reconstruction operator [7]; $\mathbf{R_n}^{-1}$ is the noise whitening filter, and the adjoint (i.e., Hermitian transpose) SFO $\mathbf{S}^+$ defines the matched spatial filter in the conventional signal processing terminology [1, 3];

(2) *SO for uncertain operational scenario* follows as structural extension of (14) for the augmented (diagonal loaded) $\mathbf{R}_{\Sigma}$ that yields [8]

$$\mathbf{F} = \mathbf{K}_{\Sigma}\mathbf{S}^+\mathbf{R}_{\Sigma}^{-1}, \tag{16}$$

where

$$\mathbf{K}_{\Sigma} = (\mathbf{S}^+\mathbf{R}_{\Sigma}^{-1}\mathbf{S} + \alpha\mathbf{A}^{-1})^{-1} \tag{17}$$

represents the robustified reconstruction operator for the *uncertain* scenario.

3.4. DEDR-Related Imaging Techniques.
Here we exemplify three practically motivated DEDR-related imaging techniques [7, 8], that will be used at the HW codesign stage, namely, the conventional matched spatial filtering (MSF) method, and two high-resolution reconstructive imaging techniques: (i) the robust spatial filtering (RSF), and (ii) the robust adaptive spatial filtering (RASF) methods.

(1) *MSF.* The MSF algorithm is a member of the DEDR-related family [7] specified for $\alpha \gg \|\mathbf{S}^+\mathbf{S}\|$, that is, the case of a dominating priority of suppression of noise over the systematic error in the optimization problem (6). In this case, the SO (14) is approximated by the matched spatial filter (MSF) [7]:

$$\mathbf{F}_{\text{MSF}} = \mathbf{F}^{(1)} \approx \mathbf{S}^+. \tag{18}$$

(2) *RSF.* The RSF method implies no preference to any prior model information (i.e., $\mathbf{A}^{-1} = \mathbf{I}$) and balanced minimization of the systematic and noise error measures in (9), (11) by adjusting the regularization parameter α to the inverse of the signal-to-noise ratio (SNR). In that case the SO becomes the Tikhonov-type robust spatial filter (RSF) [7]:

$$\mathbf{F}_{\text{RSF}} = \mathbf{F}^{(2)} \\ = (\mathbf{S}^+\mathbf{S} + \alpha_{\text{RSF}}\mathbf{I})^{-1}\mathbf{S}^+, \tag{19}$$

in which the RSF regularization parameter α_{RSF} is adjusted to a particular operational scenario model, namely, $\alpha_{\text{RSF}} = (N_0/b_0)$ for the case of a certain operational scenario [7], and $\alpha_{\text{RSF}} = (N_{\Sigma}/b_0)$ in the uncertain operational scenario case [8], respectively, where N_0 represents the white observation noise power density, b_0 is the average a priori SSP value, and $N_{\Sigma} = N_0 + \beta$ corresponds to the augmented noise power density in the correlation matrix specified by (13).

(3) *RASF*. In the Bayesian statistically optimal problem treatment, α and $\mathbf{A}$ are adjusted in an adaptive fashion following the Bayesian minimum risk strategy [8], that is, $\alpha \mathbf{A}^{-1} = \hat{\mathbf{D}} = \mathrm{diag}(\hat{\mathbf{b}})$, the diagonal matrix with the estimate $\hat{\mathbf{b}}$ at its principal diagonal, in which case the SOs (14), (16) become itself solution-dependent operators that result in the following robust adaptive spatial filters (RASFs):

$$\mathbf{F}_{\mathrm{RASF}} = \mathbf{F}^{(3)}$$
$$= \left(\mathbf{S}^+ \mathbf{R_n}^{-1} \mathbf{S} + \hat{\mathbf{D}}^{-1}\right)^{-1} \mathbf{S}^+ \mathbf{R_n}^{-1} \quad (20)$$

for the certain operational scenario [7], and

$$\mathbf{F}_{\mathrm{RASF\Sigma}} = \mathbf{F}^{(4)}$$
$$= \left(\mathbf{S}^+ \mathbf{R_\Sigma}^{-1} \mathbf{S} + \hat{\mathbf{D}}^{-1}\right)^{-1} \mathbf{S}^+ \mathbf{R_\Sigma}^{-1} \quad (21)$$

for the uncertain operational scenario [8], respectively. Next, in all practical RS scenarios [1–3] (and specifically, in SAR uncertain imaging applications [2, 7, 8]), it is a common practice to accept the robust white additive noise model, that is $\mathbf{R_n} = N_0 \mathbf{I}$, attributing the unknown correlated noise component as well as multiplicative speckle noise to the composite uncertain noise term $\Delta \mathbf{e}$ in (2), in which case $\mathbf{R_\Sigma} = N_\Sigma \mathbf{I}$ with the composite noise power density $N_\Sigma = N_0 + \beta$, the initial observation noise variance N_0 augmented by the loading factor β specified by (13).

Using the defined above SOs, the DEDR-related data processing techniques in the conventional pixel-frame format can be unified now as follows:

$$\hat{\mathbf{B}} = L\{\hat{\mathbf{b}}\}$$
$$= L\left\{\left\{\mathbf{F}^{(p)} \mathbf{Y} \mathbf{F}^{(p)+}\right\}_{\mathrm{diag}}\right\}; \quad p = 1, 2, 3, 4 \quad (22)$$

with $\mathbf{F}^{(1)} = \mathbf{F}_{\mathrm{MSF}}$; $\mathbf{F}^{(2)} = \mathbf{F}_{\mathrm{RSF}}$, and $\mathbf{F}^{(3)} = \mathbf{F}_{\mathrm{RASF}}$, $\mathbf{F}^{(4)} = \mathbf{F}_{\mathrm{RASF\Sigma}}$, respectively. Any other feasible adjustments of the DEDR degrees of freedom (the regularization parameters α, β, and the weight matrix $\mathbf{A}$) provide other possible DEDR-related SSP reconstruction techniques, that we do not consider in this paper.

4. POCS Regularized DEDR Method

Because of the extremely high dimension $(K \times K) = (K_y \times K_x) \times (K_y \times K_x) \sim 10^{12}$ of the operator inversions required to form the corresponding SOs specified by (20), (21), it is questionable to recommend the general-form DEDR-optimal method (22) as a practical enhanced RS imaging technique realizable in (near) real computational time. Hence, one has to proceed from the conventional-form $(K_y \times K_x) \times (K_y \times K_x)$-dimensional RSF and RASF algorithms (that require cumbersome operator inversions (20)–(22) to more computationally efficient *iterative* techniques that do *not* involve the large-scale operator inversions and incorporate

the convergence enforcement regularization into the DEDR procedure via constructing the proper projections onto convex sets (POCS) in the solution domain. In the considered here RS imaging applications, such POCS is aimed at performing the factorization of the overall procedures over the orthogonal range (y)-azimuth (x) coordinates in the scene frame making also an optimal use of the sparseness properties of the employed radar/SAR modulation format. Thus, the innovative idea is to perform the POCS regularization in an aggregated multi-level fashion. In particular, we propose to aggregate the positivity and range-azimuth orthogonalization projectors constructed previously in [10] with the point spread function (PSF) sparseness enforcing sliding window projectors acting in parallel over both range and azimuth image frames that set the corresponding PSF pixel values to zeroes outside their specified support regions. In this section, we address such a unified multi-level POCS-regularized iterative DEDR method as an extension of the previously proposed single-level DEDR-POCS [10] that we develop here in two stages.

4.1. First Stage: Fixed-Point Iterative DEDR Algorithm. The first stage is a structural extension of the fixed-point method considered in [10], the extension being done for the case of the unified SOs specified now by (14) and (16). Thus, following the fixed-point algorithm design scheme of [10, Section 3], we first, specify a sequence of the iterative DEDR-POCS estimates

$$\hat{\mathbf{b}}_{[n+1]} = \boldsymbol{\mathcal{T}}_{[n]}\left(\hat{\mathbf{b}}_{[n]}; \mathbf{Y}\right)$$
$$= \boldsymbol{\mathcal{P}}\left\{\mathbf{K}_{[n]} \mathbf{S}^+ \mathbf{Y} \mathbf{S} \mathbf{K}_{[n]}\right\}_{\mathrm{diag}}, \quad (23)$$

$n = 0, 1, \ldots$, where $\boldsymbol{\mathcal{P}}$ is a convergence enforcing projector (i.e., the POCS-regularizing operator) that will be construct at the second design stage (in the next subsection). In (23),

$$\mathbf{K}_{[n]} = \mathbf{K}\left(\hat{\mathbf{b}}_{[\mathbf{n}]}\right)$$
$$= \left(\boldsymbol{\Psi} + N_\Sigma \mathbf{D}^{-1}\left(\hat{\mathbf{b}}_{[\mathbf{n}]}\right)\right)^{-1} \quad (24)$$

represents the self-adjoint reconstruction operator at the *i*th iteration step, $n = 0, 1, \ldots$, and

$$\boldsymbol{\Psi} = \mathbf{S}^+ \mathbf{S} \quad (25)$$

is the nominal system point spread function (PSF) operator (a $K \times K$ matrix). Applying routinely the fixed-point technique [9, 10] to the equation (23), we derive the desired extended POCS-regularized iterative SSP estimation algorithm

$$\hat{\mathbf{b}}_{[n+1]} = \boldsymbol{\mathcal{P}}\hat{\mathbf{b}}_{[0]} + \boldsymbol{\mathcal{P}}\mathbf{T}_{[n]}\hat{\mathbf{b}}_{[\mathbf{n}]}; \quad n = 0, 1, \ldots. \quad (26)$$

Here,

$$\mathbf{T}_{[n]} = 2\,\mathrm{diag}\left(\left\{\mathbf{W}_{[n]}(\hat{\mathbf{b}}_{[n]})\right\}_{\mathrm{diag}}\right)$$
$$- \mathbf{H}_{[n]}\left(\hat{\mathbf{b}}_{[n]}\right); \quad n = 0, 1, \ldots \quad (27)$$
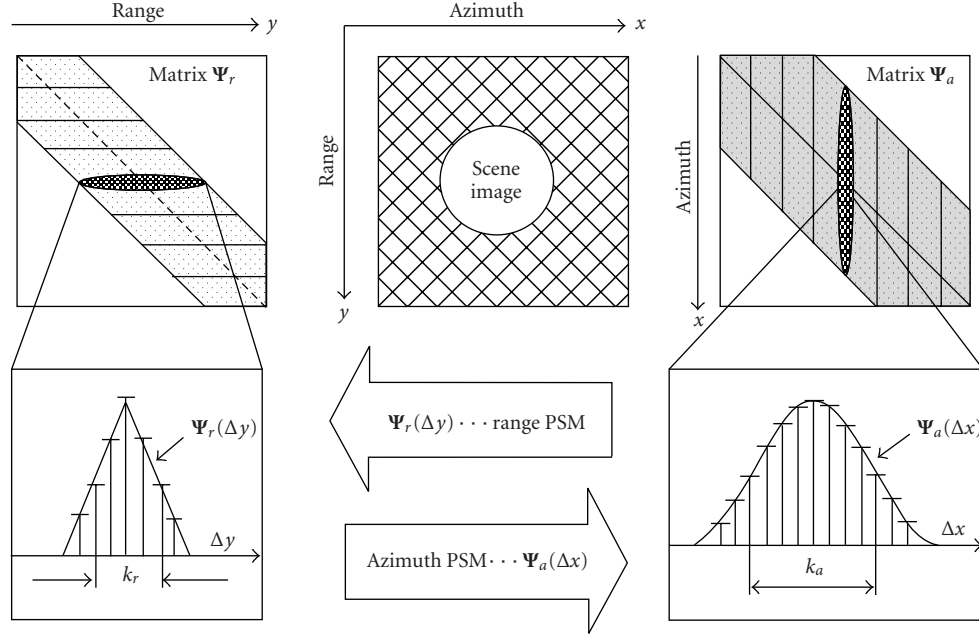
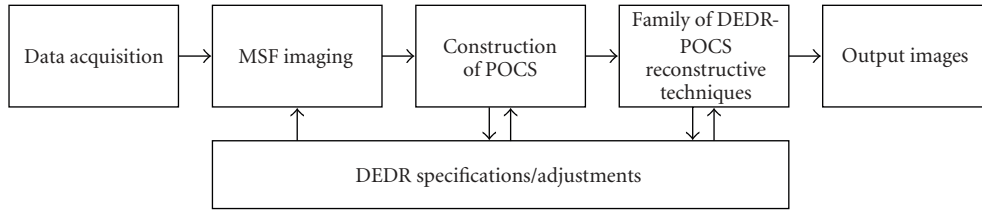FIGURE 1: Illustration of the scene image degradations over the range and azimuth directions with factorized PSMs.

FIGURE 2: General framework of the DEDR method.

represents the solution-dependent matrix-form iteration operator, in which

$$\mathbf{W}_{[n]}\left(\hat{\mathbf{b}}_{[n]}\right) = \mathbf{I} - \mathbf{\Psi} - N_\Sigma \mathbf{D}^{-1}\left(\hat{\mathbf{b}}_{[n]}\right), \qquad (28)$$

$$\mathbf{H}_{[n]}\left(\hat{\mathbf{b}}_{[n]}\right) = \mathbf{W}_{[n]}\left(\hat{\mathbf{b}}_{[n]}\right) \circ \mathbf{W}_{[n]}\left(\hat{\mathbf{b}}_{[n]}\right); \qquad (29)$$

∘ denotes the Shur-Hadamar (element-by-element) matrix product, and the zero-step iteration

$$\hat{\mathbf{b}}_{[0]} = \hat{\mathbf{b}}^{\mathrm{MSF}} = \{\mathbf{S}^+\mathbf{Y}\mathbf{S}\}\,\mathrm{diag} \qquad (30)$$

is formed as an outcome of the MSF algorithm from the DEDR family (22) specified for the adjoint SFO solution operator $\mathbf{S}^+$.

4.2. B. Second Stage: Multilevel POCS Regularization. Next, to specify the regularizing POCS projector operator $\mathscr{P}$ in the fixed-point algorithm (26) let us make the use of factorization of the PSM (25) over the azimuth (x) and range (y) coordinates valid for all existing imaging radar/SAR systems [2, 3, 9]. Such factorization is illustrated in Figure 1. We formalize this stage by introducing the range-azimuth

factorization operator $\mathscr{P}_{a\perp r}$, the same one as in the previous POCS regularization considered in [10]. Next, to make a use of the intrinsic sparseness properties of the SAR point spread functions over the range and azimuth frames, we propose to incorporate the new POCS regularization stage via constructing the x-y factorized projection operator (algorithm) $\mathscr{P}_{\kappa_a\perp\kappa_r}$ that acts as a composition of the orthogonal sliding windows [9] with the window apertures adjusted to the PSM widths: (i) $2\kappa_a$ specifies the azimuth window frame adjusted to the effective pixel width of the non-zero strip $\Psi_a(x)$ of the azimuth PSM $\mathbf{\Psi}_a$ along the x axis; (ii) $2\kappa_r$ specifies the range window frame adjusted to the effective pixel width of the non-zero strip $\Psi_r(y)$ of the range **PSM** $\mathbf{\Psi}_r$ along the y axis, respectively, as illustrated in Figure 1. Such the sliding window projector $\mathscr{P}_{\kappa_a\perp\kappa_r}$ is an easy-to-implement numerical algorithm [9] that simply sets the pixels values to zero outside the support regions $2\kappa_a \subset K_x$ and $2\kappa_r \subset K_y$ around every particular pixel $\hat{\mathbf{B}}_{[i](k_x,k_y)}$; $k_x = 1,\ldots,K_x$; $k_v = 1,\ldots,K_y$ in the rectangular image frame $\hat{\mathbf{B}}_{[n]} = L\{\hat{\mathbf{b}}_{[n]}\}$ separately reconstructed via (26) along the corresponding x and y axes, respectively. Last, following [10], to enforce prior knowledge on the intrinsic positivity of the SSP we impose, in addition to $\mathscr{P}_{a\perp r}$ and $\mathscr{P}_{\kappa_a\perp\kappa_r}$, the positivity operator (algorithm) $\mathscr{P}_+$ that has the

effect of clipping off all negative values [8]. The defined above orthogonal projecting window $\mathscr{P}_{\kappa_a \perp \kappa_r}$ and positivity operator $\mathscr{P}_+$ are projectors onto convex sets, that is POCS operators [9], thus a composition

$$\mathscr{P} = \mathscr{P}_+ \mathscr{P}_{\kappa_a \perp \kappa_r} \mathscr{P}_{a \perp r} \qquad (31)$$

is a POCS operator as well. While this definition in the terms of the proposed aggregated projections sounds complicated, the algorithmic meaning of (31) is very simple and is easily established in the algorithmic form familiar to the signal processing and RS communities. Acting on a $\hat{\mathbf{b}}_{[n]}$ (that may be not a member of the convex set at a particular iteration i), the $\mathscr{P}$ applied to $\hat{\mathbf{b}}_{[n]}$ produces the member of the convex cone set composed of non-negative elements that is nearest to $\hat{\mathbf{b}}_{[n]}$ in the sense of minimization of the L_2 norm $\|\mathscr{P}\hat{\mathbf{b}}_{[n]} - \hat{\mathbf{b}}_{[n]}\|$ [9, Section 15.4].

Now, the application of the $\mathscr{P}$ constructed by (31) to the iteration process (26) with the corresponding lexicographical reordering $\hat{\mathbf{B}} = L\{\hat{\mathbf{b}}\}$ yields the desired resulting POCS-regularized fixed point update rule

$$\hat{\mathbf{B}}_{[n+1]} = \mathscr{P}\left\{L\left\{\hat{\mathbf{b}}_{[0]}\right\}\right\} + \mathscr{P}\left\{\mathbf{T}_{[n]} L\left\{\hat{\mathbf{b}}_{[n]}\right\}\right\}; \quad n = 0, 1, \ldots, \qquad (32)$$

in which the zero-step iteration $\hat{\mathbf{B}}_{[0]} = L\{\hat{\mathbf{b}}_{[0]}\}$ is formed using the conventional (i.e., low-resolution) MSF imaging algorithm (30), the aggregated convergence enforcing POCS regularizing operator is constructed by (31), and the matrix-form fixed-point iteration operator $\mathbf{T}_{[n]}$ is specified by (27).

We address such POCS-regularized DEDR technique (32) as the unified DEDR-POCS method. Its general framework is presented in Figure 2. Note that the fixed-point process (32) does *not* involve the cumbersome operator inversions (in contrast to the initial DEDR techniques defined by (5), (22) and, moreover, it is performed separately along the range (y) and azimuth (x) directions making an optimal use of the PSM sparseness properties ($\kappa_a \ll K_x, \kappa_r \ll K_y$). These features of the POCS-regularized RSF and RASF algorithms generalized by (32) result in the drastically decreased algorithmic computational complexity (e.g, $(K_x/\kappa_a) \times (K_x/\kappa_a) \sim 10^3 \cdots 10^4$ times at each iteration for the typical large-scale SAR image formats [1, 2]) that we will verify and analyze in more details further on in Section 6.

4.3. DEDR-POCS Convergence. We accomplish our algorithmic developments at the SW codesign stage with the analytical analysis of the convergence issues related to the developed unified DEDR-POCS method. Following the POCS regularization formalism [9], the convergence enforcing projectors in the iterated procedure (32) are to be constructed formally as

$$\mathscr{P}_\iota^\lambda = \mathscr{I} - \lambda_\iota(\mathscr{P}_\iota - \mathscr{I}); \quad \iota = 1, 2, \ldots;$$
$$\mathscr{P}_1 = \mathscr{P}_{a \perp r}; \quad \mathscr{P}_2 = \mathscr{P}_{\kappa_a \perp \kappa_r}; \quad \mathscr{P}_3 = \mathscr{P}_+, \qquad (33)$$

where $\lambda_\iota; \iota = 1, 2, 3$ represent the relaxation (speeding-up) regularization parameters and $\mathscr{I}$ is the identity operator. The

iteration rule (32) for the composed regularizing projectors (33) becomes

$$\hat{\mathbf{B}}_{[n+1]} = \mathscr{P}_3^\lambda \mathscr{P}_2^\lambda \mathscr{P}_1^\lambda \hat{\mathbf{B}}_{[0]}$$
$$+ \mathscr{P}_3^\lambda \mathscr{P}_2^\lambda \mathscr{P}_1^\lambda \left\{\mathbf{T}_{[n]} \hat{\mathbf{B}}_{[n]}\right\}; \quad i = 0, 1, \ldots \qquad (34)$$

and is *guaranteed* to converge to the point in the intersection of the convex sets specified by $\mathscr{P}_\iota^\lambda$ provided $0 < \lambda_\iota < 2$ for all $\iota = 1, 2, 3$ *regardless* of the initialization $\hat{\mathbf{B}}_{[0]}$ that is a direct sequence of the fundamental theorem of POCS [9, page 1066]. Note that the employed specifications of the projectors in (33), that is, $\mathscr{P}_1 = \mathscr{P}_{a \perp r}$; $\mathscr{P}_2 = \mathscr{P}_{\kappa_a \perp \kappa_r}$; $\mathscr{P}_3 = \mathscr{P}_+$; with $\lambda_\iota = 1$ for all $\iota = 1, 2, 3$, and $\hat{\mathbf{B}}_{[0]} = L\{\hat{\mathbf{b}}_{MSF}\}$, satisfy these POCS convergence conditions, in which case the formal convergent POCS procedure (34) becomes the developed above fixed-point DEDR-POCS algorithm given by (32).

Now we are ready to proceed with the hardware codesign implementation stage of our development.

5. Hardware/Software Codesign Methodology

The all-software execution of the prescribed RS image formation and reconstruction operations in modern high-speed personal computers (PC) or any existing digital signal processors (DSP) may be intensively time consuming [15]. These high computational complexities of the general-form DEDR-POCS algorithms make them definitely unacceptable for real time PC-aided implementation.

When a coprocessor-based solution is employed in the HW/SW codesign architecture, the computational time can be drastically reduced [16]. As an introductive example, consider computation of the matrix product $\mathbf{AB}$, where $\mathbf{A}$ and $\mathbf{B}$ define matrices of sizes $k \times m$ and $m \times p$, respectively. Then to execute this product in a conventional sequential way, $k \times m \times p$ multiply accumulation (MAC) operations are required. Therefore, the computational time required by a sequential processor or a high-speed PC for the all-software execution of the matrix product is of the order $O(k \times m \times p)$. With the incorporation of a parallel and/or pipelined coprocessor alongside an embedded processor the required computational time is immediately reduced to $O(k \times m \times p/n)$, where n defines the employed parallelism level.

In this section, we present a concise description of the proposed HW/SW codesign approach particularly adapted to the DEDR-POCS type algorithms, and demonstrate its flexibility in performing an efficient HW implementation of the SW processing tasks with systolic arrays as coprocessors units. In [10], we presented an initial version of the HW/SW-architecture for implementing the digital processing of a large-scale RS imagery in other operational context. The architecture developed in [10] did not involve systolic arrays and is considered here simply as a reference for the new pursued HW/SW codesign paradigm presented in Figure 3, where the corresponding blocks are to be designed to speed-up the digital signal processing operations of the DEDR-POCS-related algorithms developed at the previous SW stage of the overall HW/SW to meet the real time imaging system
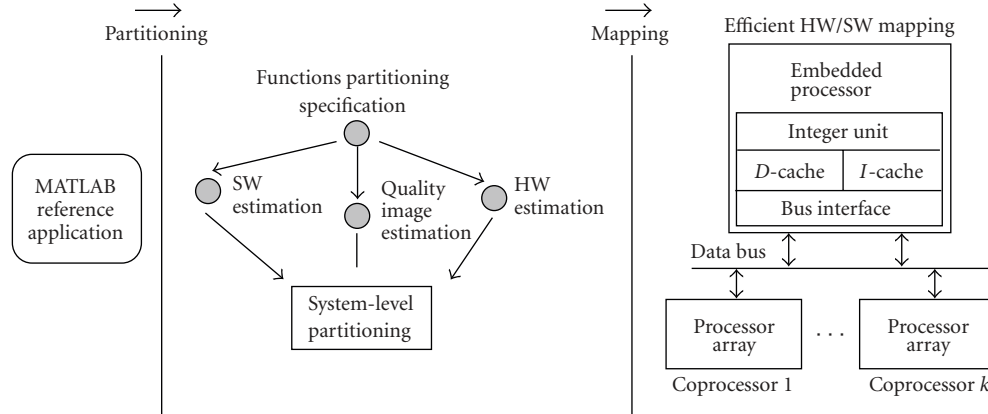
FIGURE 3: Illustration of the HW/SW codesign paradigm.

requirements. Our codesign methodology encompasses the following general stages: (i) algorithmic implementation (reference simulation in the MATLAB platform); (ii) computational tasks partitioning process (definition of the number of coprocessors), and (iii) operational mapping process employed to map the computation execution tasks onto the HW blocks (reconfigurable arrays).

In the HW design, we use the precision of 32 bits for performing all fixed-point operations, in particular, 9-bit integer and 23-bits decimal for the implementation of each coprocessor. Such the precision guarantees numerical computational errors less than 10^5 referring to the MATLAB Fixed Point Toolbox [17]. Using such the MATLAB fixed-point toolbox we generated all the numerical test sequences required to verify computationally the proposed HW/SW codesign methodology (i.e., test sequences for performing the SW simulation and for the HW verifications). The results of such SW simulation and HW performance analysis will be presented and discussed further on in Sections 6.3 and 6.4. Finally, the host processor (the standard MicroBlaze embedded processor [18] in this study) performs the following functions: loading and storing of images, data transfer to the HW coprocessors, and data formatting for performing the correspondent mathematical operations.

5.1. Algorithmic Implementation. In this section, we develop the procedures for computational implementation of the DEDR-POCS-related RSF and RASF algorithms in the MATLAB platform. This reference implementation scheme will be next compared with the proposed HW/SW codesign architecture based on the use of the single Field Programmable Gate Array chip.

To implement the iterative fixed-point DEDR-POCS-related RSF and RASF algorithms (32), we first, specify the corresponding computational procedures in the rectangular scene frame $r = (x, y) \in R$ over the azimuth (horizontal axis, x) and range direction (vertical axis, y), respectively. Such multi-stage procedures are formalized via the unified algorithmic scheme presented in Table 1.

From the analysis of the algorithmic implementation scheme of Table 1, we outline the following important

remarks regarding the possible HW/SW partitioning of the computational tasks required for implementing both RSF and RASF algorithms.

(i) First, the PSMs (25), $\mathbf{\Psi}_a$ and $\mathbf{\Psi}_r$ factorized over the azimuth and range axes can be calculated concurrently that we refer to as $\mathbf{\Psi}_a || \mathbf{\Psi}_r$, where symbol $||$ specifies now the concurrent execution of the corresponding computational operations.

(ii) Second, the zero step iteration (MSF image) $\hat{\mathbf{B}}_{[0]}$ can be computed using the same factorized structure analogues to $\mathbf{\Psi}_a || \mathbf{\Psi}_r$.

(iii) Third, the reconstructed image $\hat{\mathbf{B}}_{[i+1]}$, at the current $(i + 1)$st iteration step is an iteratively updated function of $\{\hat{\mathbf{B}}_{[i]}\}$ computed at the previous ith iteration that also admits the factorized computing.

5.2. Partitioning Phase. One of the challenging problems of the HW/SW codesign is to perform an efficient HW/SW partitioning of the computational tasks. The goal of the partitioning stage is to find which computational tasks can be implemented in an efficient parallelized HW/SW architecture seeking for balanced *area-time* trade-offs between different admissible design solutions [18–20]. In this study, the iterative fixed-point POCS-DEDR regularized algorithm has been partitioned at the algorithmic level to minimize the overall signal processing (SP) *time* via transferring some required reconstructive SP functions from the SW to the HW. The solution to this problem requires, first, the definition of a partitioning model that meets all the specification requirements (functionality, goals and constraints).

The system partitioning is clearly influenced by the target architecture onto which the HW and the SW will be mapped. The target architecture proposed in this study consists of one 32 bits RISC instruction set embedded processor (MicroBlaze) running the software and three dedicated coprocessors implemented by systolic processor arrays.

We begin with the specifications of the system-level partitioning functions and detailing the selected design quality attributes for the HW/SW codesign aimed at the

TABLE 1: Computational scheme for implementing the POCS-regularized RSF and RASF algorithms.

(i)	Data acquisition $\mathbf{u}_{(j)}$; $j = 1, \ldots, J$
(ii)	Formation of the current RS correlation data matrix $\mathbf{Y}$ (4)
(iii)	Specification of the observation noise correlation model $\{\mathbf{R_n}; \mathbf{R_n}^{-1}\}$
(iv)	Specification of the POCS model parameters $\{\kappa_a; \kappa_r\}$
(v)	Specification of the POCS operator components $\{\boldsymbol{\mathscr{P}}_+; \boldsymbol{\mathscr{P}}_{\kappa_a \perp \kappa_r}; \boldsymbol{\mathscr{P}}_{a \perp r}\}$
(vi)	Specification of the azimuth-range SFOs $\{\mathbf{S}_a; \mathbf{S}_r\}$
(vii)	Computations of the azimuth-range PSMs $\{\boldsymbol{\Psi}_a; \boldsymbol{\Psi}_r\}$ (25)
(viii)	Formation of the azimuth-range POCS operators see (31)
(ix)	Formation of the MSF image $\widehat{\mathbf{B}}_{[0]} = L\{\widehat{\mathbf{b}}_{[0]}\}$ (30)
(x)	Iterative POCS-RSF image enhancement $\widehat{\mathbf{B}}_{\text{RSF}}$ (using (32)) with the robust updating $\mathbf{A}^{-1} = \text{diag}^{-1}(\widehat{\mathbf{b}}_{[0]})$ of the iterative reconstruction operator (27)
(xi)	Iterative POCS-RASF image reconstruction $\widehat{\mathbf{B}}_{\text{RASF}}$ (using (32)) employing the adaptive updating $\mathbf{A}^{-1} = \text{diag}^{-1}(\widehat{\mathbf{b}}_{[i]})$ of the iterative reconstruction operator (27)
(xii)	Termination of the iteration procedures
(xiii)	Image enhancement/reconstruction performance analysis using the adopted quality metrics.

definition of the computational tasks that can be implemented in a systolic computing form, namely: hardware area (*ha*), hardware execution time (*ht*), software execution time (*St*), and the selected system resolution (*n*); where max *ha*, max *ht* and max *St* represent the upper bounds of these constraints. In particular, for implementing the iterative fixed-point POCS-regularized RSF and RASF algorithms, the partitioning process must satisfy the following performance requirements.

(i) In order to ensure a viable solution, the system must always satisfy the constraints: $0 \le ha < \max ha$, $0 \le ht < \max ht$, for each *i*th hardware coprocessor $\{\Gamma_i; i = 1, 2, 3\}$ and $0 \le St < \max St$, for the embedded processor Ω. These three hardware coprocessors $\{\Gamma_i\}$ and the embedded processor compose the target architecture $\boldsymbol{\mathcal{D}} = \{\Omega, \Gamma_1, \Gamma_2, \Gamma_3, \Upsilon\}$, for the pre-selected FPGA Υ with the corresponding predetermined architecture constraints $\boldsymbol{\mathcal{C}} : \{0 \le ha < \max ha; 0 \le ht < \max ht; 0 \le St < \max St\}$ [18].

(ii) Each block implementation $\{\boldsymbol{\mathcal{G}}(\Gamma_i)\}, \boldsymbol{\mathcal{G}}(\Omega)$ must satisfy the predefined execution time performance requirements [18]: $\tau\{\boldsymbol{\mathcal{G}}(\Gamma_i \mid \boldsymbol{\mathcal{C}}_i); i = 1, 2, 3\}$ and $\tau\{\boldsymbol{\mathcal{G}}(\Omega \mid \boldsymbol{\mathcal{C}}_4)\}$ conditioned by the specified above architecture constraints $\{\boldsymbol{\mathcal{C}}_i : \{0 \le ht_i < \max ht_i; 0 \le ha_i < \max ha_i\} \forall i = 1, 2, 3\}$, and $\boldsymbol{\mathcal{C}}_4 : 0 \le St < \max St$, correspondingly.

Next, the system architecture $\boldsymbol{\mathcal{D}}$ is to be specified to meet the desirable time consuming performances via bounding the total expected system processing time $\tau = \tau\{\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{D}} \mid \boldsymbol{\mathcal{C}})\}$ evaluated by

$$\tau\{\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{D}} \mid \boldsymbol{\mathcal{C}})\} = \left(\max_i \{\tau\{\boldsymbol{\mathcal{G}}(\Gamma_i \mid \boldsymbol{\mathcal{C}}_i)\}\} \right.$$
$$\left. + \tau\{\boldsymbol{\mathcal{G}}(\Omega \mid \boldsymbol{\mathcal{C}}_4)\} \right) < T_{\text{MATLAB}}, \quad (35)$$

where T_{MATLAB} represents the execution time required for implementing the corresponding DEDR-POCS-related RSF and RASF algorithms in the standard MATLAB computational environment.

Following such partitioning paradigm, we decompose now the fixed-point POCS-regularized RSF and RASF algorithms developed at the SW-design into the standard MicroBlaze embedded processor Ω with three coprocessors $\{\Gamma_i; i = 1, 2, 3\}$ as illustrated in Figure 4. The first coprocessor Γ_1 (referred to as MSF coprocessor) implements the zero-step iteration to form the MSF image $\widehat{\mathbf{B}}_{[0]} = L\{\widehat{\mathbf{b}}_{[0]}\}$ specified by (30). The second coprocessor Γ_2 (referred to as PSM coprocessor) implements the computations of the PSM $\boldsymbol{\Psi}_a || \boldsymbol{\Psi}_r$ given by (25) concurrently over the azimuth and the range directions. The third coprocessor Γ_3 (referred to as Iterative POCS coprocessor) performs the required robust updating $\mathbf{A}^{-1} = \text{diag}^{-1}(\widehat{\mathbf{b}}_{[0]})$ for implementing the RSF algorithm and the adaptive updating $\mathbf{A}^{-1} = \text{diag}^{-1}(\widehat{\mathbf{b}}_{[n]})$ for implementing the POCS-regularized RASF image reconstruction algorithm, respectively. All three coprocessors $\{\Gamma_i; i = 1, 2, 3\}$ are next implemented as systolic processor arrays while the embedded processor Ω executes all the required operational and control functions: loading and storing of the images, data transfer to the HW coprocessors, and data formatting for execution of all required numerical operations.

5.3. Mapping Phase. In this section, we proceed with the development of the procedure for mapping the corresponding algorithms onto array processors. A systolic array consists of a number of processor elements (PEs) with the corresponding interconnection links among the PEs, and the mapping technique transforms a space representation into a space-time representation [21]. Systolic arrays are being used for matrix operations and required specific processing algorithms, such as, transform techniques, matrix multiplication, convolution, and so forth, [21, 22]. The methodology
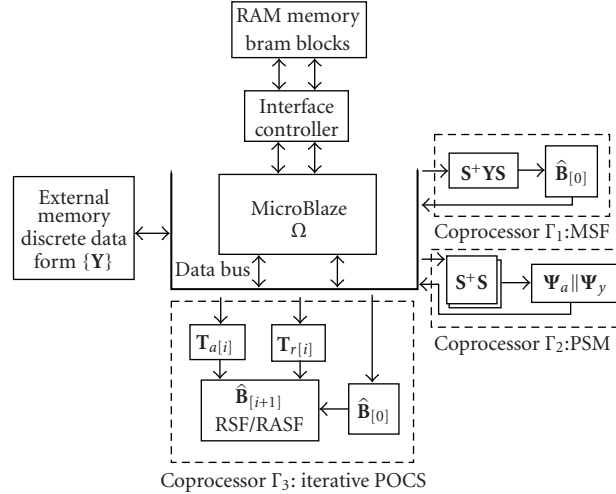
FIGURE 4: Illustration of the partitioning phase of the HW/SW codesign.

of mapping the algorithms onto array structures is depicted in Figure 5.

First, to achieve the desired maximal possible parallelism in an algorithm, we perform the analysis of the data dependencies in the corresponding computations. Then, the algorithm is transformed into a single assignment algorithm without global communication. A dependence graph (DG) is used to analyze these data dependencies of the corresponding algorithms [21]. Following [21], DG is defined as $\mathbf{G} = [\mathbf{P}, \mathbf{E}]$, where $\mathbf{P}$ represents a set of nodes and $\mathbf{E}$ is a set of arcs (or edges), that is, each edge $e \in \mathbf{E}$ connects the corresponding pair of nodes $p_1, p_2 \in \mathbf{P}$ and the connection is formalized by $e = p_1 \xrightarrow{e} p_2$.

Second, we employ the systolic design paradigm to map a high dimensional (N-dimensional) DG to a lower dimensional Signal to Flow Graph (SFG) [21]. Recall that the systolic array is a space-time representation, in which the function description defines the behavior within a node, whereas the structural description specifies the interconnections (edges and delays) between the nodes [21, 22]. In order to derive a regular systolic array architecture with minimum number of nodes, we employ the linear projection approach for processor assignment following the methodology developed in [21, 22], that is, the nodes of the DG in a certain straight line are projected onto the corresponding PEs in the processor array represented by the corresponding projection vector $\mathbf{d}$. Thus, we seek for a linear order reduction transformation [22]:

$$\mathbf{\Phi} : \mathbf{G}^N \longrightarrow \hat{\mathbf{G}}^{N-1} \qquad (36)$$

that maps the N-dimensional DG ($\mathbf{G}^N$) onto the ($N - 1$)-dimensional SFG ($\hat{\mathbf{G}}^{N-1}$).

Such the desired linear transformation matrix $\mathbf{\Phi}$ admits the partitioning in two functions [22]

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Sigma} \end{bmatrix}. \qquad (37)$$

Here, $\mathbf{\Pi}$ defines a $(1 \times p)$ vector composed of the first row of $\mathbf{\Phi}$ that determines the time scheduling. This vector indicates the normal direction of the equi-temporal hyper-planes in the DG, "equi-temporal" being understood in the sense that all the nodes on the same hyper-plane must be processed at the same time [22]. The submatrix $\mathbf{\Sigma}$ of $(p-1) \times p$ dimension (the rest rows of $\mathbf{\Phi}$), determine the space processor. With this mapping, we are now ready to proceed with the construction of the required regular ($N - 1$)-dimensional systolic arrays.

5.4. HW Implementation. Once the HW/SW codesign has been defined, the three coprocessors employed in the architecture exemplified in Figure 4 can be implemented using the HW systolic arrays. In this study, we are oriented at the use of the Xilinx MicroBlaze soft processor that employs the On Chip Peripheral Bus (OPB) for transferring the data from/to the memory to/from the coprocessor [23]. Such the OPB is a fully synchronous bus that connects other separate 32 bit data buses. This system architecture (based on the FPGA XC4VSX35-10ff668 with the embedded processor and the OPB buses) restricts the corresponding processing frequency to 100 MHz. The typical rate of the OPB bus is 133 MByte/s, providing that each data transfer of 32-bits is accomplished at 30.05 ns [23]. Next, to avoid multiple data transfer from the embedded processor data memory to the coprocessors, a register file is to be implemented inside each coprocessor.

The first systolic array (referred to as the MSF coprocessor) implements the zero-step iteration of the unified fixed-point DEDR-POCS procedure (32) to form the MSF image $\hat{\mathbf{B}}_{[0]} = L\{\{\mathbf{S}^+\mathbf{Y}\mathbf{S}\}_{\mathrm{diag}}\}$ as specified by (30). The function of this systolic array is to perform the triple matrix multiplication, where matrix $\mathbf{S}$ has the band-Toeplitz structure [5, 6] with the width of the non-zero strip over the azimuth frame equal to $2\kappa_a$. Following the methodology addressed in the previous section, the triple matrix multiplication corresponding to the MSF function can be implemented using a cascade systolic array. First, the multiplication of a band-Toeplitz matrix and a rectangular matrix is performed

and then, the result is multiplied with another band-Toeplitz matrix. Each slide of the DG in the multiplication of the band-Toeplitz matrix and the rectangular matrix is employed using the following specifications in the transformations defined by (37): $\boldsymbol{\Pi} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ for the vector schedule, $\mathbf{d} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ for the projection vector and $\boldsymbol{\Sigma} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ for the space processor that determine the resulting transformation matrix $\boldsymbol{\Phi}$ (37). In Figure 6(a), we illustrate the triple matrix multiplication mapped into a cascade systolic array with the relevant MSF systolic array architecture exemplified in Figure 6(b). The corresponding computations require only $O(2\kappa_a \times K_x \times 2\kappa_a)$ fixed-point operations, with $2\kappa_a \ll K_x$, where, as previously, $2\kappa_a$ defines the width of the non-zero strip in the factorized band-Toeplitz PSM (25) and K_x is the original image dimension over the azimuth frame.

The MSF coprocessor systolic architecture of Figure 6(b) consists of identical linearly-connected processing elements (PEs). In our case, the internal structure of each PE contains a multiplier and an adder. Each PE receives 32-bits operands and generates 64-bits product. Then, the product is truncated to 32-bits with a fixed-point adopted representation of 9 integers and 24 decimals. Next, since the band-Toeplitz type matrix $\mathbf{S}_a$ is preloaded, the incoming data $\mathbf{Y}$ are transmitted in parallel to the corresponding PEs. After $2\kappa_a$ cycles of clock, the data outputs are produced and transferred to the registers (gray blocks in Figure 6(a)). Once the first of the triple matricial product is completed, the data transfer to the second array begins. The control unit block guarantees the correct synchronization between the arrival of the input data and the computations for each PE. The result buffer of Figure 6(b) consists of a shift buffer used to store the $2k_a \times K_x \times 2k_a$ elements generated in parallel by the boundary PEs. Finally, the bus interface unit realizes the communication between the systolic array and the embedded processor.

The second systolic array (referred to as the PSM coprocessor) implements the computation of the Point Spread Matrix (PSM) function $\boldsymbol{\Psi} = \mathbf{S}^+\mathbf{S}$ concurrently over the azimuth and range axes, that is $\boldsymbol{\Psi}_a || \boldsymbol{\Psi}_r$, where, as previously, symbol $||$ specifies the concurrent execution of the corresponding computational operations. In the PSM function, both matrices $\boldsymbol{\Psi}_a$ and $\boldsymbol{\Psi}_r$ are band-Toeplitz type matrices $(\dim\{\boldsymbol{\Psi}_a\} = K_x \times K_x; \dim\{\boldsymbol{\Psi}_r\} = K_y \times K_y)$ with the widths of the non-zero strips equal to $2\kappa_a$ and $2\kappa_r$, correspondingly, where due to the PSM sparseness, $2\kappa_a \ll K_x$ and $2\kappa_r \ll K_y$. Thus, to perform the required reconstruction over the azimuth direction, it is possible to achieve full parallelism with only an $2\kappa_a \times 2\kappa_a$ rectangular array (as opposed to an original full-dimensional $K_x \times K_x$ array in the general case [10, 21]). Due to the range-azimuth factorization, the same parallelism is achievable in the range direction as well with the corresponding $2\kappa_r \times 2\kappa_r$ rectangular systolic array. The matrix multiplication of two band-Toeplitz type matrices employs now the following specifications in the transformations defined by (37): $\boldsymbol{\Pi} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ for the vector schedule, $\mathbf{d} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ for the projection vector, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ for the space processor that determine the resulting transformation matrix $\boldsymbol{\Phi}$ (37). The topological

distribution of the processing elements (PEs) in such systolic structure is shown in Figure 7(a). The corresponding PSM systolic coprocessor architecture is presented in Figure 7(b) with three independent directions of data flow.

The third coprocessor (referred to as the Iterative POCS coprocessor) performs the adaptive updating of the iterative reconstruction operator $\mathbf{T}_{[n]}$ in the corresponding fixed-point DEDR-POCS procedure (32). The key operations of this coprocessor are to perform the standard 1-D convolution and the vector-matrix multiplication. The systolic array for performing the 1-D convolution employs now the following specifications in the transformations defined by (37): $\boldsymbol{\Pi} = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ for the vector schedule, $\mathbf{d} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ for the projection vector, and $\boldsymbol{\Sigma} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ for the space processor that determine the resulting transformation matrix $\boldsymbol{\Phi}$ (37). Figure 8(a) illustrates the 1-D convolution systolic array and Figure 8(b) presents the relevant systolic architecture.

In summary, the developed systolic architectures perform the parallel and pipelined schemes which exploit the proposed above mapping methodology. These architectures provide the necessary HW-level implementation of the SW-optimized complex multi-purpose RS imaging algorithms.

6. Simulations and Performance Analysis

6.1. Simulation Experiment Specifications. In the verification simulation experiments, we considered a conventional single-look SAR with the fractionally synthesized aperture as an RS imaging system [1, 2]. Recall, that signal formation operator (SFO) of such a SAR is factored along two axes in the image plane [3]: the azimuth or cross-range coordinate (horizontal axis, x) and the slant range (vertical axis, y), respectively. We considered the conventional triangular SAR range ambiguity function (AF) [3] $\Psi_r(y)$ and Gaussian approximation [5, 6], $\Psi_a(x) = \exp(-(x)^2/a^2)$, of the SAR azimuth AF with the adjustable fractional parameter, a. Note that in the imaging radar applications [3, 4], an AF is referred to as the continuous-form approximation of the PSM $\boldsymbol{\Psi}$ defined by (25) and serves as an equivalent to the point spread function in the conventional image processing terminology [9]. The image degradation and noising effects were incorporated to simulate the process of formation of the degraded speckle-corrupted MSF images. First, following [1, 3] the degradation in the spatial resolution due to the fractional aperture synthesis mode were simulated via blurring the original image with the range AF $\Psi_r(\Delta y)$ along the y axis and with the azimuth AF $\Psi_a(\Delta x)$ along the x axis, respectively. Next, the degradations at the image-formation level due to the propagation and calibration uncertainties were simulated using the statistical model of a SAR image defocusing [2, 3]. For a considered single-look SAR, the conventional MSF image formation algorithm (30) implies, first, application of the regular adjoint SFO $\mathbf{S}^+$ to the zero-mean Gaussian data realization $\mathbf{u}$, and second, performing the element-by-element (i.e., pixel-by-pixel) squared detection of $\mathbf{S}^+\mathbf{u}$ to compose the corresponding SSP pixel estimates $\{\hat{b}_{\mathrm{MSF}\,k} = |\{\mathbf{S}^+\mathbf{u}\}_{kk}|^2; k = 1,\ldots,K\}$. Consequently, the MSF pixel estimates $\{\hat{B}_{[i](k_x,k_y)} = L\{\hat{b}_{\mathrm{MSF}\,k}\}\}$ are chi-squared
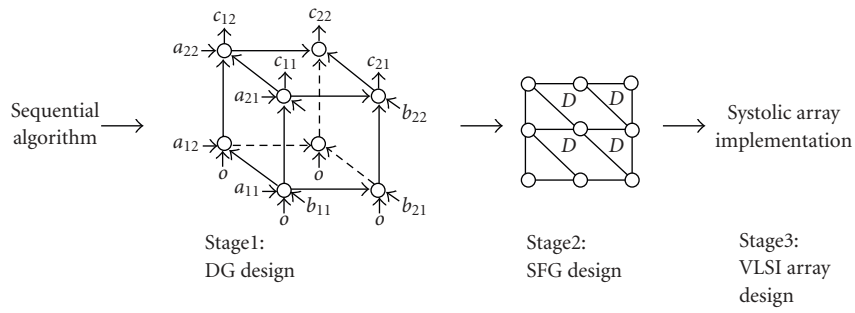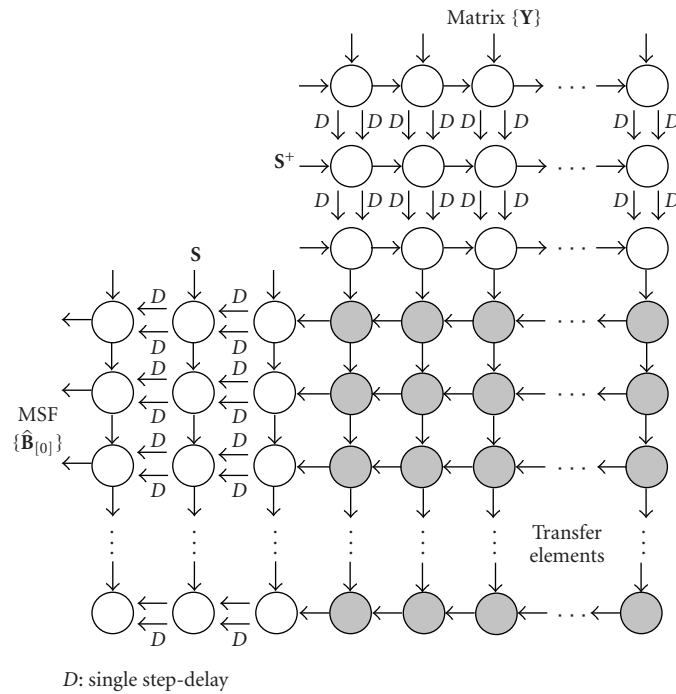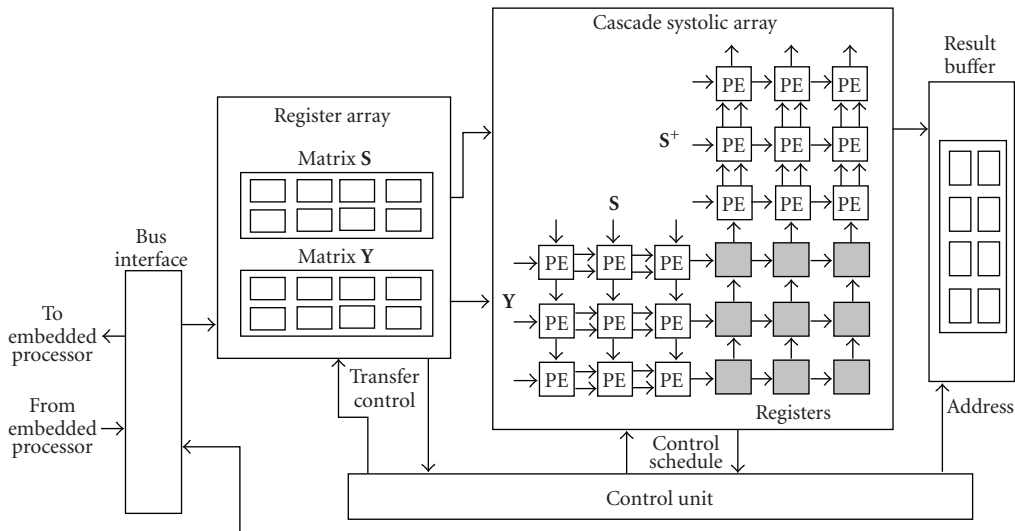
Figure 5: Mapping Design Methodology.

(a)

(b)

Figure 6: MSF implementation: (a) cascade systolic array for performing the MSF function; (b) MSF systolic architecture.
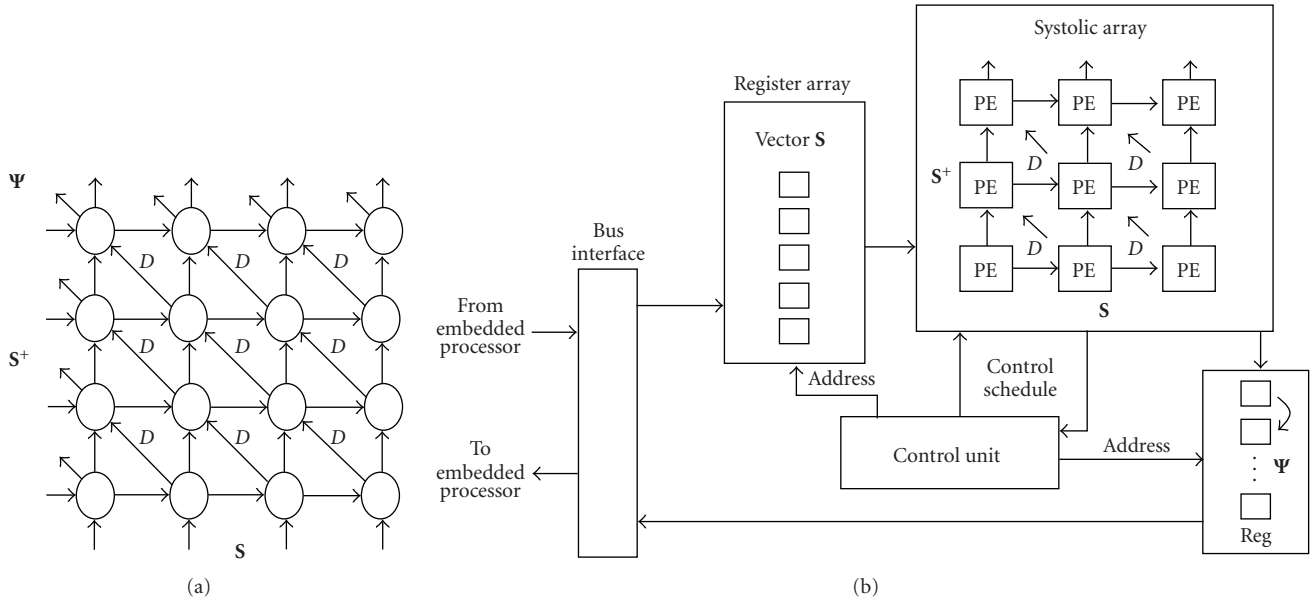
FIGURE 7: PSM implementation: (a) systolic array for performing the PSM function; (b) PSM coprocessor systolic architecture.
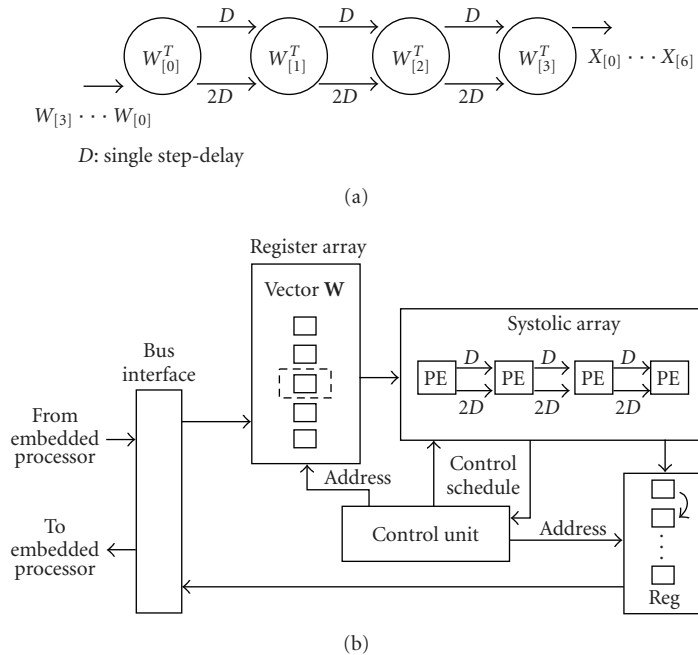

FIGURE 8: Convolution function: (a) systolic array for performing the 1D convolution; (b) systolic array architecture.

distributed χ_2^2 with two degrees of freedom, and such a distribution is a negative exponential Rayleigh distribution [2, 9]. Thus, to comply with the technically-motivated MSF image formation scheme, the composite multiplicative noise was simulated as a realization of the χ_2^2-distributed random variables with the pixel mean value assigned to the actual degraded scene image pixel that directly obeys the statistical speckle model [2, 5, 6]. Such signal-dependent multiplicative image noise dominates the additive noise component in the data in the sense that $N_\Sigma \gg N_0$, hence the estimate

$\hat{N}_\Sigma$ performed empirically via the application of the local statistics method [2] was used to adjust the regularization degrees of freedom (regularization factors) in all simulated DEDR-related SSP reconstruction procedures.

We have run the simulation experiments for both certain and uncertain operational scenarios. In the both scenarios, we considered the MSF, RSF and RASF algorithms from the DEDR-POCS family (22). Also, to compare the developed algorithms with the conventional SAR image enhancement techniques [1–3], the celebrated Lee adaptive de-speckling

FIGURE 9: Original test scene borrowed from the high-resolution RS imagery.
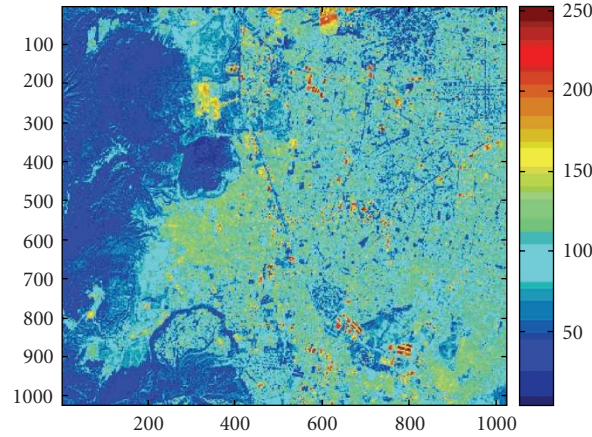

FIGURE 10: Original test scene represented in MATLAB pseudocolor scale.

filter based on the local statistics method [2] was simulated. The family of four simulated techniques were renumbered as $p = 1, \ldots, 4$. The first one ($p = 1$) relates to the conventional MSF estimator (30) that employs the adjoint SO $\mathbf{F}^{(1)} = \mathbf{S}^+$. This degraded MSF image $\hat{\mathbf{B}}^{(1)} = L\{\hat{\mathbf{b}}_{MSF}\}$ was then post-processed applying the Lee adaptive de-speckling filter [2] that we refer to as the adaptively de-speckled MSF image $\hat{\mathbf{B}}^{(2)}$, that is, $p = 2$. Next, the non-adaptive RSF algorithm with the solution operator $\mathbf{F}^{(3)} = \mathbf{F}_{RSF}$ defined by (19) was applied to enhance the original MSF image $\hat{\mathbf{B}}^{(1)}$ employing the iterative DEDR-RSF version of the unified fixed-point iterative procedure (32); the resulting DEDR-RSF enhanced image was specified as $\hat{\mathbf{B}}^{(3)} = L\{\hat{\mathbf{b}}_{RSF}\}$ and numbered as $p = 3$. Last, the fourth simulated technique corresponds to the adaptive DEDR-RASF method (32) with the optimal solution operator $\mathbf{F}^{(4)} = \mathbf{F}_{RASF}$ given by (21); the resulting adaptively enhanced DEDR-RASF image was specified as $\hat{\mathbf{B}}^{(4)} = L\{\hat{\mathbf{b}}_{RASF}\}$ and numbered correspondingly as $p = 4$. In the second (uncertain) simulated scenario, the system AF

was additionally distorted over the azimuth frame within the realistic interval of $\Delta \kappa a = 0.07 \kappa a$ that corresponds to the partially uncompensated carrier trajectory deviations interval [2, 10]. For both scenarios, the simulations were run for different composite signal-to-noise ratios (SNR) μ defined as the ratio of the average signal component in the rough image formed using the MSF algorithm (30) to the relevant noise component in the same image, $\mu = (b_0/N_\Sigma)(\mathrm{tr}\{\boldsymbol{\Psi}_a\}\,\mathrm{tr}\{\boldsymbol{\Psi}_r\})^{-1} \times \mathrm{tr}\{(\boldsymbol{\Psi}_a)^2\}\,\mathrm{tr}\{(\boldsymbol{\Psi}_r)^2\}$, where b_0 represents the average gray level of the original scene image.

6.2. Performance Metrics. The first adopted quality metric was borrowed from the classical image reconstruction applications [9] defined as an improvement in the output signal-to-noise ratio (IOSNR):

$$\mathrm{IOSNR} = 10 \log_{10} \frac{\sum_{k=1}^{K} \left(\hat{b}_k^{(MSF)} - b_k \right)^2}{\sum_{k=1}^{K} \left(\hat{b}_k^{(p)} - b_k \right)^2}; \quad p = 2, 3, 4, \quad (38)$$

TABLE 2: IOSNR values provided with three simulated DEDR-related methods, $p = 2, 3, 4$ (2) adaptive despeckling filter; (3) DEDR-RSF; (4) DEDR-RASF; results are reported for the certain and uncertain simulated scenarios.

| SNR (dB) | IOSNR$^{(p)}$; $p = 2, 3, 4$ | | | | | |
| | FIRST (CERTAIN) SCENARIO: $\kappa_r = 6; \kappa_a = 15$ | | | SECOND (UNCERTAIN) SCENARIO: $\kappa_r = 6; \kappa_a = 18$ | | |
	IOSNR$^{(2)}$	IOSNR$^{(3)}$	IOSNR$^{(4)}$	IOSNR$^{(2)}$	IOSNR$^{(3)}$	IOSNR$^{(4)}$
5	1.83	3.73	9.12	1.31	3.45	6.36
10	2.47	4.80	10.11	1.96	4.14	7.93
15	3.25	7.87	11.12	3.21	6.67	8.52
20	4.64	9.05	13.42	4.02	8.32	10.28

where b_k represents the value of the kth element (pixel) of the original image $\mathbf{B}$, $\widehat{b}_k^{(MSF)}$ represents the value of the kth element (pixel) of the degraded image formed applying the MSF technique (37), and $\widehat{b}_k^{(p)}$ represents a value of the kth pixel of the image reconstructed with three simulated enhancement methods, $p = 2, 3, 4$ where $p = 2$ corresponds to the adaptive de-speckling algorithm based on the local statistics method [2], $p = 3$ corresponds to the POCS-RSF algorithm and $p = 4$ corresponds to the POCS-RASF algorithm, that is, the best one from the developed DEDR-POCS family, respectively. The second adopted metric, the so-called mean absolute error (*MAE*), was employed as a metric suitable for quantification of edges and fine detail preservation in the reconstructed image defined as [15]

$$\text{MAE} = 10 \log_{10} \left(\left(\frac{1}{K} \right) \sum_{k=1}^{K} \left| \widehat{b}_k^{(p)} - b_k \right| \right); \quad p = 2, 3, 4. \quad (39)$$

According to these quality metrics, the higher is the IOSNR, and the lower is the MAE, the better is the improvement of the image enhanced/reconstructed with the particular employed algorithm.

6.3. Simulations. In this study, the simulations were performed with a large scale (1K-by-1K) pixel-format image borrowed from the real-world high-resolution terrain SAR imagery (south-west Guadalajara region, Mexico [24]). The quantitative measures of the image enhancement/reconstruction performance gains achieved with the particular employed POCS-RSF and POCS-RASF techniques for different SNRs evaluated with two different quality metrics (38), (39) are reported in Table 2. Figure 9 shows the original scene image (not observable with the simulated SAR systems). Figure 10 illustrates the same original test scene represented in MATLAB pseudocolor scale.

The images of Figures 11(a) through 11(h) present the results of image formation and enhancement applying different DEDR-related estimators without model uncertainties as specified in the figure captions. In the second simulated scenario, the fractional SAR system suffered from more severe degradations because of the additional system defocusing and multiplicative speckle noising due to the operational scenario uncertainties. Figures 12(a) thru 12(h) present the results of image formation and enhancement applying different DEDR-related estimators in the simulated

uncertain operational scenario as specified in the figure captions. From the analysis of the reported simulation results, it is evident that the RASF method overperformed the robust nonadaptive RSF in both simulated scenarios. This demonstrates that employing the adaptive RASF technique from the DEDR-POCS family one could substantially improve the quality of the RS images (reconstructed from both certain and uncertain RS measurement data) approaching in the same time (near) real-time computational performances.

Next in Figure 13, we present the convergence curves related to the iterative-form implementation of the POCS-RSF/RASF techniques for the test case 15 dB SNR (i.e., $\mu = 15$ dB). From the analysis of these curves one can deduce that after 40 iterations both POCS-RSF and POCS-RASF algorithms begin to suffer from some numerical instabilities. This type of numerical instability is a subtle issue in constructing the regularized iterative techniques for different ill-conditioned problems, for example [1, 3, 9], and so forth. Moreover, the relationship between the resulting *IOSNR* quality metric and the visual reconstructed image quality is not fully understood, although, of course, one would expect a high degree of correlation between the two [9]. These observations are in concord with the similar observations from other studies of the inverse imaging problems in other ill-posed contexts, for example, [1, 2, 9, 10]. In our case, due to the POCS regularization, the appearance of the DEDR reconstructed images demonstrated substantial improvement up to 15 iterations from the MSF starting point. Next, the appearance of the reconstructed images changed very little from that of the 15 to 25 iterations. The changes became perceptually undistinguishable after $25 \cdots 30$ iterations. This behavior can be used as a motivation for the *empirical* stopping rule at $25 \cdots 30$ iterations of the POCS-regularized iterative POCS-RSF and POCS-RASF algorithms.

6.4. HW/SW Codesign Performance Analysis. In this section, we complete our study with the comparative analysis of the computational complexities of the simulated iterative DEDR-POCS algorithms implemented using the systolic coprocessors constructed following the addressed HW/SW codesign approach. The synthesis metrics related to the implementation of the systolic arrays architectures as coprocessors are summarized in Table 3. First, we exemplify the MSF, PSM and iterative POCS coprocessor architectures for the following simplified specifications: data matrices of size 12×12 and two Band-Toeplitz PSF matrices of the same

TABLE 3: Synthesis metrics. Specifications for data matrices of size 12×12 and two Band-Toeplitz PSF matrices of the same 12×12 pixel size with equal bandwidths of $2\kappa_a = 3$ and $2\kappa_r = 3$.

Synthesis metrics	Systolic array coprocessors		
	MSF	*PSM*	Iterative POCS
Number of slices	905 (5.89%)[c]	242 (1.57%)	216 (1.41%)
Number of [a]DSP'48	192 (100%)	9 (4.68%)	12 (6.25%)
Number of [b]LUTs	932 (3.03%)	—	—
Number of flip-flops	1845 (6.01%)	480 (1.56%)	432 (1.40%)
Maximum frequency	115.30 MHz	152.20 MHz	148.69 MHz
Maximum pin delay	8.67 ns	6.57 ns	6.72 ns

[a]DSP'48 *are dedicated DSP blocks in high-end FPGAs–such as the Xilinx XtremeDSP slice in Virtex-4* [18].
[b]LUTs *is an acronym to Look Up Tables structures.*
[c]*Data in parenthesis (.) report the percentages of the occupied HW resources related to the particular synthesis metrics in terms of the total available on the relevant device.*

TABLE 4: Processing times required for implementing the conventional DEDR and the developed POCS-regularized (SW/HW codesign-based) unified DEDR-POCS techniques (RSF and RASF).

Implementation method	Processing time [seconds]	
	RSF (per iteration)	RASF (per iteration)
Hypothetical Full-Format Implementation(Evaluated PC-Oriented Implementation)	5171.6	5655
Factorized Fixed-Point POCS-Regularized Implementation (PC-Oriented)	19.70	20.05
Previous HW/SW codesign-based implementation [10] (without systolic arrays)	7.82	7.985
Proposed HW/SW codesign-based implementation (with systolic arrays)	2.51	2.56

Note–Processing times may vary depending on the processor type, CPU memory and the software used.

12×12 pixel size with equal bandwidths of $2\kappa_a = 3$ and $2\kappa_r = 3$ pixels. The relevant SP performance analysis results are resumed in Table 3. Next, in Figures 14(a) through 14(c), we summarize the relevant HW synthesis performances for the realistic case of large-scale processed RS scenes (e.g., to $1K \times 1K$ pixel size) and report the overall resource utilization performances attained with the proposed HW coprocessors architectures for different number of processing elements (PEs).

Next, the reported metrics of Table 3 specify the area and time behaviors of the corresponding hardware systolic arrays, that is the corresponding MSF, the PSM, and the iterative POCS architectures specified above in Section 5. From the analysis of the data reported in Table 3 and Figures 14(a) through 14s(c), one can deduce the following: With the proposed HW/SW codesign architecture (in which the embedded processors iterate properly the corresponding SP procedures) the DEDR-POCS-related algorithms can be efficiently implemented in an iterative fixed-point fashion also for the realistic large-scale scenes (e.g., $1K \times 1K$ pixel size). Pursuing the proposed systolic computing architecture concept, the increased scene dimensionality requires the proper segmentation of the scene frame with the parallelized computing performed over the partitioned segments followed by the relevant integration of the overall partial processed data. Such partitioned systolic HW/SW codesign computing-oriented processing can be performed directly following the architecture design concept proposed and specified in the previous Section 5.4. Additionally, the scalability in terms

of Flip-Flops, Slices and LUTs (i.e., the HW resources of the FPGA) for the proposed MSF, PSM and iterative POCS coprocessors are reported in Figures 14(a) through 14(c). In fact, the corresponding DEDR-related SP algorithms can be efficiently implemented in a Field Programmable Systems on Chip (FPSoC) mode in spite of employing conventional systems based on multi-FPGAs or PC-Clusters [12–14, 16]. The latter is practically inspired and desirable for a wide range of RS and general SP applications due to the large range density of the existing FPGAs that incorporate huge resources of logical gates, block RAM memory modules and soft or hard-embedded processors integrated on the same chip with the relevant custom co-processing HW blocks, and so forth. For example, an alternative approach for high-speed computational implementation of the reconstructive RS image processing based on the use of clusters of PCs was presented in [12–14]. In [12], the cluster NSPO Parallel TestBed for performing parallel radiometric and geometrical corrections of the large-scale 3600×2944-pixel RS images was implemented. The reconstructive image processing was conducted using a PC-Cluster composed by three PCs each one with a Pentium-III 550 MHz with 128 MB of RAM connected with 100 Mbps Fast-Ethernet LAN. The processing time achieved with such three-PCs cluster was only 33.3 seconds (near-real time for conventional RS users), while the corresponding processing performed with one single processor required 84.65 seconds. In [13, 14], another kind of parallel architecture was implemented for morphological classification of hyperspectral RS imagery at the NASA's
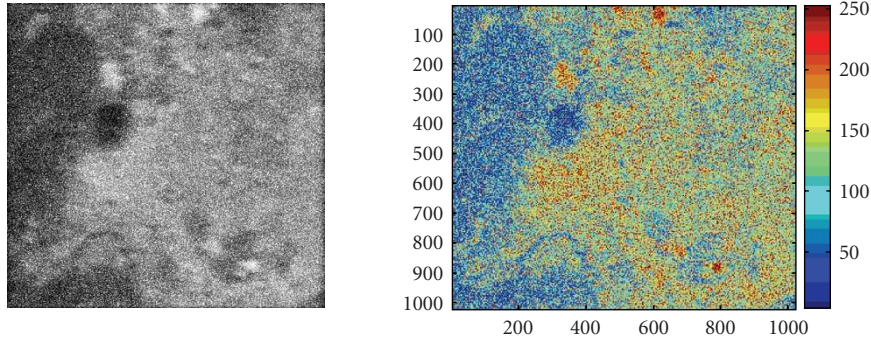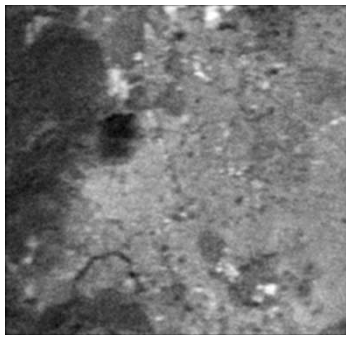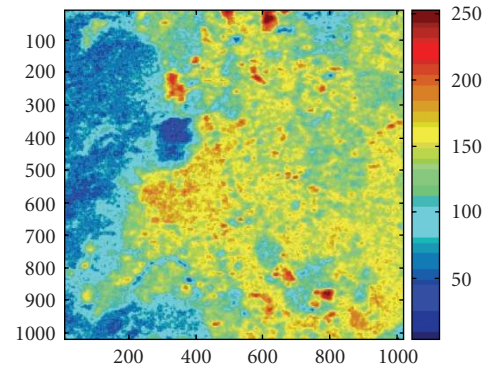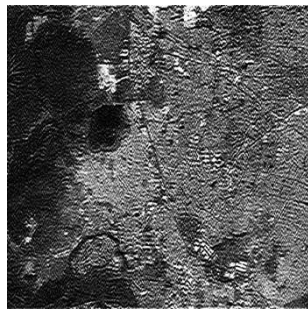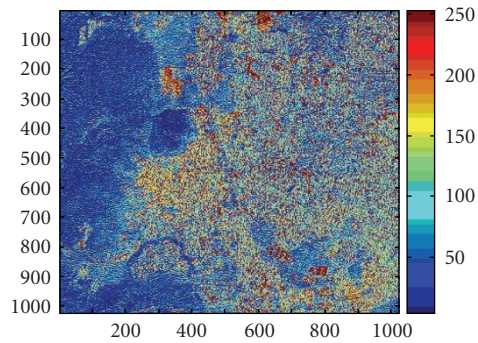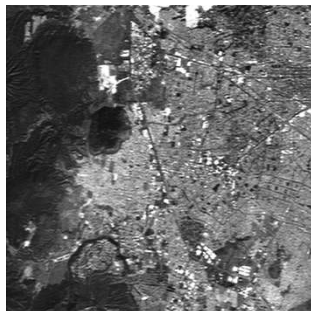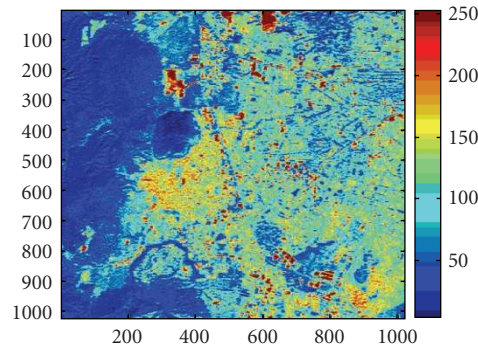
FIGURE 11: Simulation results for certain observation scenario: (SNR $\mu = 10$ dB): (a) degraded scene image formed applying the MSF method; (b) the same degraded scene represented in the MATLAB pseudo-color scale; (c) image reconstructed applying the Lee adaptive de-speckling algorithm; (d) the same adaptively de-speckled scene represented in the MATLAB pseudo-color scale; (e) image reconstructed applying the POCS-RSF algorithm; (f) image reconstructed applying the POCS-RSF algorithm represented in the MATLAB pseudo-color scale; (g) image reconstructed applying the POCS-RASF algorithm; (h) image reconstructed applying the POCS-RASF algorithm represented in the MATLAB pseudo-color scale.
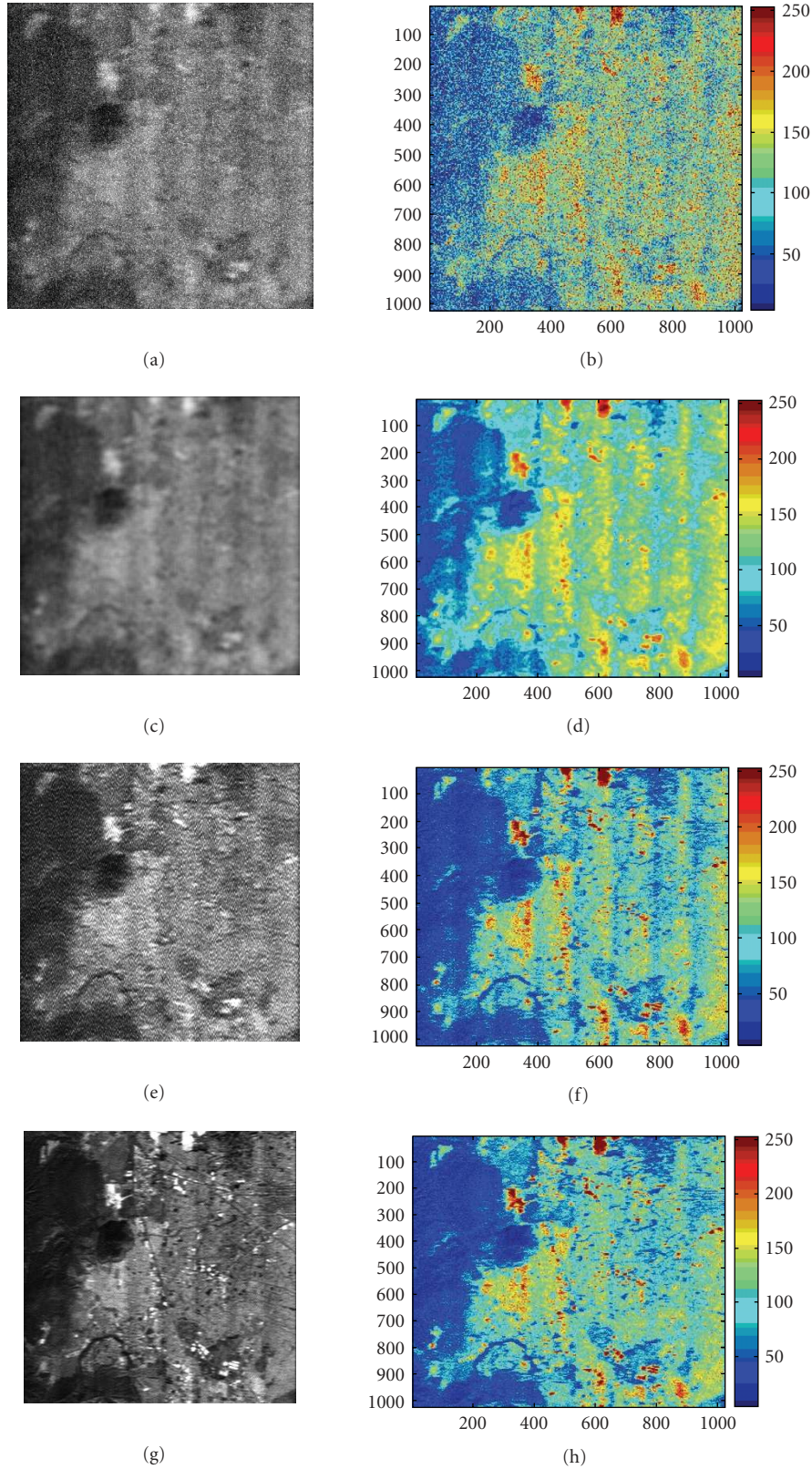
(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

FIGURE 12: Simulation results for uncertain observation scenario: (SNR $\mu = 10$ dB): (a) degraded scene image formed applying the MSF method; (b) the same degraded scene represented in the MATLAB pseudo-color scale; (c) image reconstructed applying the Lee adaptive de-speckling algorithm; (d) the same adaptively de-speckled scene represented in the MATLAB pseudo-color scale; (e) image reconstructed applying the POCS-RSF algorithm; (f) image reconstructed applying the POCS-RSF algorithm represented in the MATLAB pseudo-color scale; (g) image reconstructed applying the POCS-RASF algorithm; (h) image reconstructed applying the POCS-RASF algorithm represented in the MATLAB pseudo-color scale.
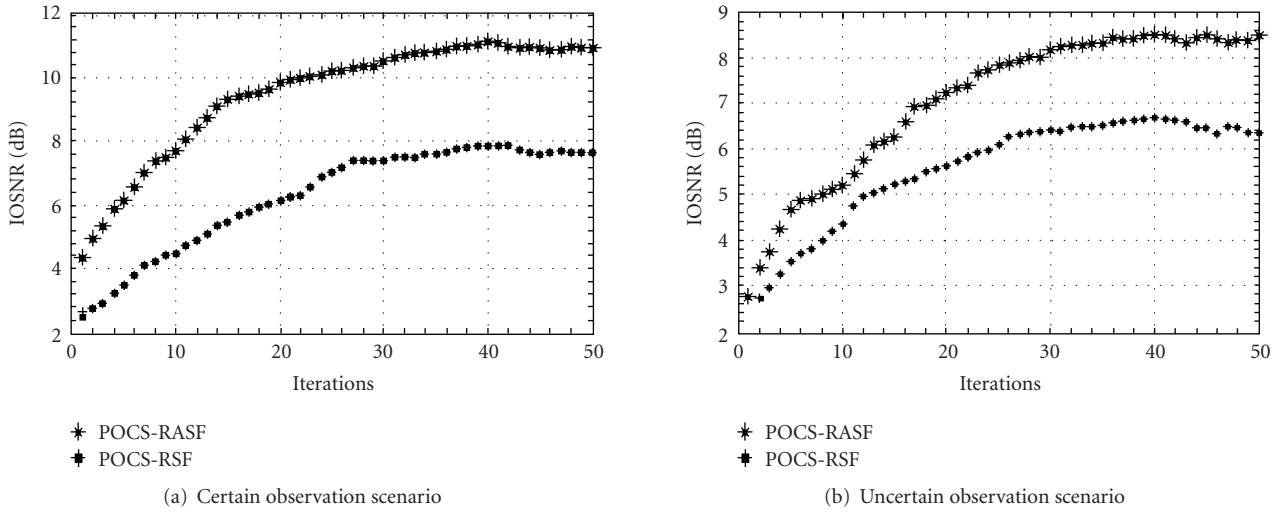
(a) Certain observation scenario

(b) Uncertain observation scenario

FIGURE 13: Convergence curves of the iterative POCS-RSF and POCS-RASF algorithms.

(a) MSF coprocessor

(b) PSM coprocessor
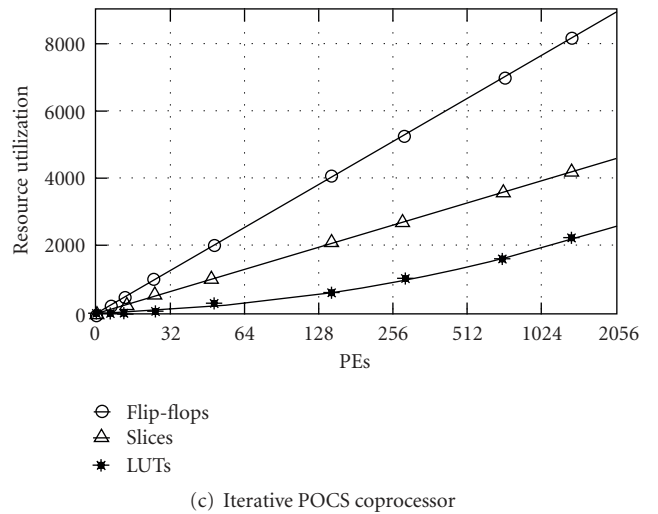
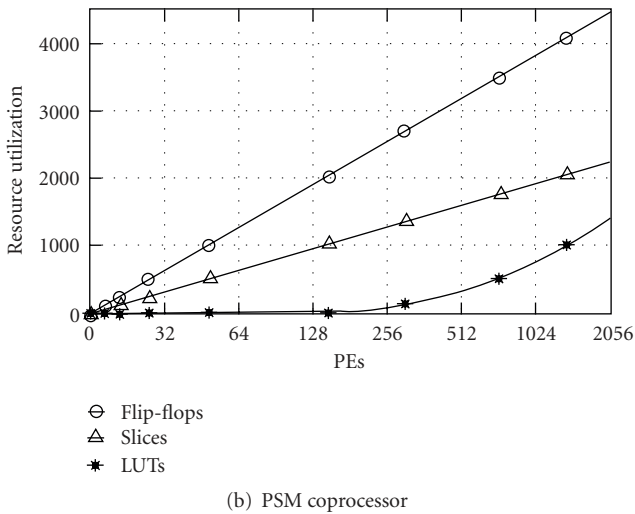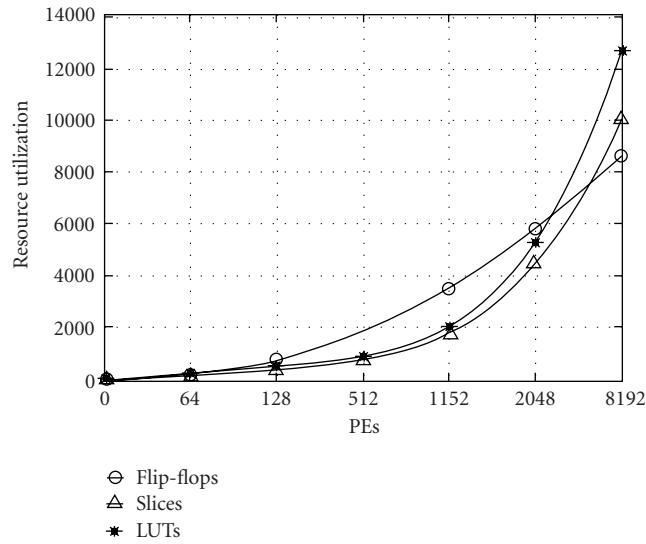(c) Iterative POCS coprocessor

FIGURE 14: Resource utilization varying the number of PEs.

Goddard Space Flight Center. The parallel classifier of [14] uses 256-processor Beowulf cluster (Thunderhead cluster) with hybrid neural parallelism that enables such a system to perform an accurate classification of the hyperspectral RS scenes in only 17 seconds.

As a result, advances on high performance computing as well as on specialized high performance hardware modules are necessarily required to achieve the near-real processing time performances for complex RS algorithms.

Last, we compared the required processing time of the general-form RSF/RASF DEDR-related procedures (22) and the iterative fixed-point DEDR-POCS-regularized algorithm (32), both implemented using the conventional MATLAB software in a personal computer (PC) running at 3 GHz with a AMD Athlon (tm) 64 dual-core processor and 2 GB of RAM memory. Also, the same DEDR-related algorithms were implemented using the proposed HW/SW codesign architecture (soft- and hardware) without systolic and with systolic arrays employing the Xilinx FPGA XC4VSX35-10ff668. The corresponding comparative results are reported in Table 4. Analyzing these reported results, one may deduce the following. The iterative fixed-point DEDR-POCS-regularized algorithm (32) manifests the (near) real time high-resolution enhancement/reconstruction of the RS imagery. The implementation of the proposed HW/SW codesign architecture helps to reduce drastically the overall processing time. Particularly, the proposed implementation of the iterative POCS-regularized RASF algorithm with systolic arrays takes only 2.56 seconds for each iteration of the image reconstruction. In total, it takes 64 sec for 25 iterations. This new computation time is approximately 3 times less than the previous implementation without systolic arrays [10], 8 times less than the corresponding processing time achievable with the MATLAB POCS-based implementation, and it is $\sim 10^3$ times less than the hypothetical processing time required for implementing the full-format conventional general-form DEDR-RASF algorithm (22) without POCS regularization and without systolic computing.

7. Concluding Remarks

The principal result of the undertaken study relates to the digital signal processing-oriented solution of the RS image enhancement/reconstruction problems in a (near) real time computing mode (the "near real time" being understood in context of conventional RS users) via exploiting the aggregated hardware/software (HW/SW) codesign paradigm that results in an efficient hardware implementation architecture based on the use of systolic array processors. We have approached the goal of the (near) real time computational implementation of the enhancement/reconstruction of the RS imagery from two directions. First, we have analytically established that to alleviate the problem ill-posedness and reduce the overall computational load of the large-scale image enhancement/reconstruction tasks at the algorithmic processing level, some special form of descriptive experiment design projection-type numerical regularization must be employed. This stage was developed and addressed here as the unified DEDR method, and the efficient fixed-point numerical iterative technique that incorporates the proper construction of the relevant orthogonally factorized regularizing projector onto convex sets (POCS) in the solution domain was designed and specified for the particular employed RS sensor system, namely, the side-looking imaging synthetic aperture radar (SAR) operating in both certain and uncertain scenarios. We have also examined how such SAR-adapted POCS-regularized fixed-point iterative technique can be executed concurrently over the orthogonal range-azimuth coordinates with optimal use of the sparseness properties of the overall SAR system point spread function characteristics. The algorithmic-level advantages of such unified DEDR-POCS-regularized RS image enhancement/reconstruction techniques relate to the theoretically guaranteed convergence of the corresponding fixed-point iterative process with the proper factorization of the numerical reconstructive procedures over the orthogonal range-azimuth directions in the representation image frame.

Second, we have examined that pursuing the proposed HW/SW codesign paradigm and employing the systolic arrays as co-processing units, the (near) real time image processing requirements can be achieved due to performing the corresponding computations in an efficient systolic architecture mode. The unified algorithmic (software-level, SW) and systematic (hardware-level, HW) codesign approach was verified via computer simulation experiments indicative of its efficiency for performing the RS image enhancement and reconstruction tasks in (near) real computational time. The tested DEDR-POCS-related techniques implemented numerically using the proposed HW/SW codesigned computational architecture overperform the previously developed methods both in the attained reconstruction quality and the achievable computational complexity, that is manifest the substantially reduced overall computational time (e.g., up to three orders with respect to the schemes that do not aggregate the POCS regularization with the systolic computing). We do believe that pursuing the DEDR-POCS-related HW/SW codesign paradigm with systolic array hardware accelerators one could approach definitely the real time computational requirements while performing the reconstructive processing of the large-scale RS imagery attaining the enhancement/reconstruction performance gains close to the limiting bounds.

References

[1] D. R. Wehner, *High-Resolution Radar*, Artech House, Boston, Mass, USA, 2nd edition, 1994.

[2] J. S. Lee, "Speckle suppression and analysis for synthetic aperture radar images," *Optical Engineering*, vol. 25, no. 5, pp. 636–643, 1986.

[3] F. M. Henderson and A. V. Lewis, "Principles and applications of imaging radar," in *Manual of Remote Sensing*, John Wiley & Sons, New York, NY, USA, 3rd edition, 1998.

[4] Y. V. Shkvarko, "Estimation of wavefield power distribution in the remotely sensed environment: Bayesian maximum entropy approach," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2333–2346, 2002.

[5] Y. V. Shkvarko, "Unifying regularization and Bayesian estimation methods for enhanced imaging with remotely sensed

data—part I: theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 923–931, 2004.

[6] Y. V. Shkvarko, "Unifying regularization and Bayesian estimation methods for enhanced imaging with remotely sensed data—part II: implementation and performance issues," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 932–940, 2004.

[7] Y. V. Shkvarko, "From matched spatial filtering towards the fused statistical descriptive regularization method for enhanced radar imaging," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 39657, 9 pages, 2006.

[8] Y. V. Shkvarko, H. Perez-Meana, and A. Castillo-Atoche, "Enhanced radar imaging in uncertain environment: a descriptive experiment design regularization paradigm," *International Journal of Navigation and Observation*, vol. 2008, Article ID 810816, 11 pages, 2008.

[9] H. H. Barrett and K. J. Myers, *Foundations of Image Science*, John Willey & Sons, New York, NY, USA, 2004.

[10] A. Castillo, Y. V. Shkvarko, D. Torres, and H. M. Perez, "Convex regularization-based hardware/software co-design for real-time enhancement of remote sensing imagery," *International Journal of Real Time Image Processing*, vol. 4, no. 3, pp. 261–272, 2008.

[11] V. I. Ponomaryov, "Real-time 2D-3D filtering using order statistics based algorithms," *Journal of Real-Time Image Processing*, vol. 1, no. 3, pp. 173–194, 2007.

[12] C. T. Yang, C. L. Chang, C. C. Hung, and F. Wu, "Using a Beowulf cluster for a remote sensing application," in *Proceedings of the 22nd Asian Conference on Remote Sensing*, vol. 1, Singapore, November 2001.

[13] "Thunderhead System," http://newton.gsfc.nasa.gov/thunderhead/.

[14] A. Plaza and J. Plaza, "Parallel morphological classification of hyperspectral imagery using extended opening and closing by reconstruction operations," in *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS '08)*, pp. I.58–I.61, Boston, Mass, USA, July 2008.

[15] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Array*, Springer, Berlin, Germany, 2001.

[16] J. Greco, G. Cieslewski, A. Jacobs, I. A. Troxel, and A. D. George, "Hardware/software interface for high-performance space computing with FPGA coprocessors," in *Proceedings of the IEEE Aerospace Conference*, pp. 10–25, Big Sky, Mont, USA, March 2006.

[17] "Fixed-Point Toolbox™ User's Guide, MATLAB," http://www.mathworks.com/.

[18] "EDK 9.1 MicroBlaze tutorial in Virtex-4," Xilinx, http://www.xilinx.com/.

[19] A. Marquardt, V. Betz, and J. Rose, "Speed and area tradeoffs in cluster-based FPGA architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 1, pp. 84–93, 2000.

[20] M. López-Vallejo and J. C. López, "On the hardware-software partitioning problem: system modeling and partitioning techniques," *ACM Transactions on Design Automation of Electronic Systems*, vol. 8, no. 3, pp. 269–297, 2003.

[21] S. C. Lo and S. N. Jean, "Mapping algorithms to VLSI array processors," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '88)*, pp. 2033–2036, New York, NY, USA, 1988.

[22] S. Y. Kung, *VLSI Array Processors*, Prentice Hall, Upper Saddle River, NJ, USA, 1988.

[23] "Xilinx Application Note XAPP967: creating an OPB IPIF-based IP and using it in EDK," 2007.

[24] "Space Imaging," GeoEye, 2008, http://www.euspaceimaging.com/.