Plug-and-Play Priors Enabled SAR Image Inpainting in the Presence of Speckle Noise

Satyakam Baraha
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
Odisha, India
satyakambaraha93@gmail.com

Ajit Kumar Sahoo
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
Odisha, India
ajitsahoo@nitrkl.ac.in

***Abstract*—Synthetic aperture radar (SAR), being a coherent imaging system, usually produces images that are affected by granular deformities known as speckle. Image restoration from such noisy observation is an ill-posed problem. Model-based optimization is the framework that effectively tackle such inverse problems by building the degradation model and utilizing the prior information. The modular structure of alternating direction method of multipliers (ADMM) converges to solution by iteratively minimizing the cost function, which is the sum of the above two models. Recently, Plug-and-Play (PnP) ADMM is developed, which provides the flexibility to use image denoisers in place of regularizers. Image estimation from partial/incomplete observation is quite challenging and open topic of research in the literature. In this paper, SAR image reconstruction under multiplicative noise is discussed for image inpainting using PnP ADMM. Simulation results show that denoisers can be used to restore the images affected by large fractions of missing pixels.**

***Index Terms*—ADMM, inpainting, model based reconstruction, plug-and-play, speckle.**

I. Introduction

The primary objective of image restoration is to process the degraded image in such a way that it results in more suitable than the original observation for a particular application. The interpretation of such deteriorated visual data is quite challenging, hence leads to different image reconstruction tasks. For example, image inpainting is a common problem in image restoration, where a set of pixels from the image are lost due to the bit transmission error and limitation of the imaging sensor. Further, the probability distribution of noise and the image formation model entirely depend on the physics of the image acquisition sensor. In the coherent imaging systems such as SAR, the clean image $\mathbf{f}$ is corrupted by multiplicative speckle noise. The general model of the degraded image is given by

$$\mathbf{g} = (\mathbf{H}\mathbf{f} \odot \boldsymbol{\xi}_m). \tag{1}$$

The notations $\mathbf{g}, \mathbf{f}, \boldsymbol{\xi}_m \in \mathbb{R}^{N \times 1}$ correspond to degraded, clean images and noise respectively and these vectors are of length N arranged in lexicographic ordering. The matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$ act as a binary mask with values 0 and 1, when dealt with image inpainting and $\odot$ denotes element-wise multiplication. For SAR imagery, the multiplicative noise $\boldsymbol{\xi}_m$ is assumed to follow Rayleigh and Poisson distributions.

Regularized inversion for image restoration is an active trend followed in most of the literature for image restoration. The unknown image is estimated by calculating the maximum a posteriori (MAP), that minimizes the sum involving negative log of data fidelity and prior terms.

$$\begin{aligned}\hat{\mathbf{f}} &= \arg\min_{\mathbf{f}}\{-\log p(\mathbf{g}|\mathbf{f}) - \log p(\mathbf{f})\} \\ &= \arg\min_{\mathbf{f}}\{l(\mathbf{g};\mathbf{f}) + \lambda r(\mathbf{f})\},\end{aligned} \tag{2}$$

where $l(\mathbf{g};\mathbf{f})$ is the forward model and $r(\mathbf{f})$ represents the prior. The optimization tasks of the form as shown in Eq. (2) are solved using ADMM and reported in a large number of works in literature.

A. Related Works

Total variation (TV) based regularized inversion is widely used to reconstruct the images. The minimization of TV norm can preserve the edge information in the restored images. Seabra *et al.* [1] proposed Log-Euclidean priors for Rayleigh corrupted ultrasound images. TV norm is linked with the data model and the convex cost function is minimized using Newton's method. Neri and Zara [2] formulated a TV-based model to solve the denoising and inpainting problem for Gaussian noise and solved it using the primal-dual method. Afonso and Sanches [3], [4] recovered the images from partial noisy observations using TV regularizer and solved it using the ADMM approach. Though the above methods provide good reconstruction accuracy, but TV norm is non-smooth, which requires an additional algorithm such as Chambolle's method [5] to solve it, which makes it quite slow. Further, use of TV-based regularizers often results in stair-casing artifacts [6] in the flat regions of the image. Later, the authors of [4] have extended their work in [7] towards blind image inpainting where they used TV regularizer on the image to make it piecewise-smooth and ℓ_0 norm on the mask to make it sparse and solved it using ADMM. Though the method is computationally faster and more accurate, but the presence of ℓ_0 norm in the cost function makes it non-convex.

Plug-and-Play priors a variant of classical ADMM demonstrated by Venkatakrishnan *et al.* [8], where state-of-the-art denoisers can be used as priors. It works effectively even when the prior information regarding the image is not available. Chan *et al.* [9] extended PnP ADMM for the continuation scheme to restore the images corrupted by additive white Gaussian noise. Primal-dual splitting based PnP (PDSPnP) for removal

978-1-7281-6916-3/20/$31.00 ©2020 IEEE

of Gaussian noise was reported by the authors of [10]. It avoids the requirement of inner loops, hence computationally efficient compared to classical PnP ADMM. Significant contributions have been made by Brifman *et al.* [11], where they exploited the denoiser to perform both denoising as well as super-resolution operation in the presence of Gaussian noise.

B. Motivation and Contributions

ADMM is the most versatile method used in the literature to solve the optimization problems. The selection of appropriate regularizer is crucial and depends upon the specific application and the clean image $\mathbf{f}$. Common priors such as l_0 and l_1 norms introduce sparsity into the solution of the optimization problem, whereas minimization of TV regularizer smooths the solution. But, prior information regarding the latent image does not need to be known beforehand. PnP ADMM uses standard denoisers to tackle the above problem efficiently, but are widely studied for linear Gaussian noise model. Coherent models such as SAR is an exception to it as it deals with multiplicative noise that is non-linear. Further, most of the image restoration algorithms are focused on denoising only. This paper tries to bridge the gap found in the above discussions by proposing the following contributions:

1) Image inpainting under multiplicative noises like Rayleigh and Poisson is proposed using PnP ADMM.
2) M-sample intensity averaging is performed in the presence of Rayleigh corrupted speckle and the resultant amplitude follows Nakagami distributions which improves the performance.
3) The inpainting approach can restore the images up to 95 percent loss of pixel values from the image.

C. Organization

The remaining of the manuscript is structured as follows. Section II deals with the formulation of the optimization problem. The proposed method for image inpainting involving the PnP ADMM continuation scheme is addressed in Section III. Simulation results are demonstrated in Section IV, which compares the proposed method with state-of-the-art methods. Section V concludes the paper along with the path for future research.

II. Preliminaries and Formulation

Let $\mathbf{g} = \{g_i\}_{i=1}^{N}$ be the pixel amplitude values observed in a given region. Presence of speckle makes the distribution of pixels random in nature. In this work, the joint distributions of the amplitude values are considered, which are independent and identically distributed and is given by

$$p(\mathbf{g}|\mathbf{f}) = \prod_{i=1}^{N} p(g_i|f_i). \tag{3}$$

Probability density function (pdf) of fully developed Rayleigh corrupted speckle is given by [12]

$$p(g_i|f_i) = \frac{g_i}{(\mathbf{Hf})_i}\exp\left(-\frac{g_i^2}{2(\mathbf{Hf})_i}\right). \tag{4}$$

The forward model is defined in Eq. (2) as negative log-likelihood function and can be given by

$$l(\mathbf{g};\mathbf{f}) = -\log p(\mathbf{g}|\mathbf{f}) = \sum_{i=1}^{N}\{\log((\mathbf{Hf})_i) + \frac{g_i^2}{2(\mathbf{Hf})_i}\}. \tag{5}$$

Similarly, the pdf for Poisson based speckle is

$$p(g_i|f_i) = \frac{e^{-(\mathbf{Hf})_i}((\mathbf{Hf})_i)^{g_i}}{g_i!}. \tag{6}$$

and the corresponding data model is given by

$$l(\mathbf{g};\mathbf{f}) = \sum_{i=1}^{N}\{(\mathbf{Hf})_i - g_i\log((\mathbf{Hf})_i)\}. \tag{7}$$

When $\mathbf{H} = \mathbf{I}$, Eq. (1) is a standard despeckling problem. The work regarding SAR image despeckling can be found in our preliminary work [13]. $\mathbf{H}$ is a diagonal matrix. Hence i^{th} term of $(\mathbf{Hf})$ is defined as $(\mathbf{Hf})_i = h_i f_i$. For image inpainting, the diagonal element h_i is 1 and 0, when the pixel values in clean image $\mathbf{f}$ are observed and missed respectively. Presence of logarithmic term in Eq. (5) and Eq. (7) makes the data term non convex. Hence take the transformation $z_i = \log(h_i)$ and $y_i = \log(f_i)$. To avoid logarithm null, a small positive scalar $\epsilon \approx 10^{-3}$ is taken in place of 0 for h_i. Now Eq. (5) and Eq. (7) can be reformulated as convex optimization problem as follows.

$$\begin{aligned}\ell(\mathbf{g};\mathbf{y}) &= \sum_{i=1}^{N}\{(y_i+z_i) + \frac{g_i^2}{2}e^{-(y_i+z_i)}\}\\ &= \sum_{i=1}^{N}\{y_i + \frac{g_i^2}{2}e^{-(y_i+z_i)}\},\end{aligned} \tag{8}$$

and

$$\begin{aligned}\ell(\mathbf{g};\mathbf{y}) &= \sum_{i=1}^{N}\{e^{(y_i+z_i)} - g_i(y_i+z_i)\}\\ &= \sum_{i=1}^{N}\{e^{(y_i+z_i)} - g_i y_i\}.\end{aligned} \tag{9}$$

The cost function in Eq. (2) is minimized with respect to $\mathbf{f}$. After the transformation $y_i = \log(f_i)$, the minimization is with respect to $\mathbf{y}$. Hence, the terms independent of $\mathbf{y}$ are eliminated in Eq. (8) and Eq. (9). Hence, the unconstrained optimization problem in Eq. (2) is given by

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}}\{l(\mathbf{g};\mathbf{y}) + \lambda r(\mathbf{y})\}, \tag{10}$$

which defines a wide range of problems such as image denoising, and inpainting in the presence of multiplicative noise.

III. Proposed Method

M-sample intensity averaging is performed on the amplitudes of Rayleigh distributed speckle noise using a mask of size $P \times P$, i.e. $\bar{\xi}_{m_i} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\xi_{m_i}^2}$. The resultant amplitude of the speckle follows the Nakagami distributions [14]. The pdf of Nakagami distributed speckle is given by

$$p(g_i|f_i) = \frac{2M^M}{\Gamma(M)(2(\mathbf{Hf})_i)^M}(g_i)^{2M-1}\exp(-\frac{Mg_i^2}{2(\mathbf{Hf})_i}), \quad (11)$$

where $M = P^2$. The forward model for Nakagami distributions can be derived in the same way following the steps of Eq. (5) and Eq. (8) as follows:

$$l(\mathbf{g};\mathbf{y}) = M\sum_{i=1}^{N}\{y_i + \frac{g_i^2}{2}e^{-(y_i+z_i)}\}. \quad (12)$$

A. PnP framework for image restoration

Variable splitting is performed on Eq. (10), which introduces a decision variable as the argument of r to solve the constrained optimization problem as shown below [15]:

$$\{\hat{\mathbf{y}}, \hat{\mathbf{m}}\} = \arg\min_{\mathbf{y},\mathbf{m}}\{l(\mathbf{g};\mathbf{y}) + \lambda r(\mathbf{m})\}, \text{ s.t. } \mathbf{y} = \mathbf{m}. \quad (13)$$

The augmented Lagrangian turns the constraint into a penalty term with β as its penalty parameter and $\mathbf{w}$ as the dual variable.

$$\mathscr{L}_\beta(\mathbf{y},\mathbf{m},\mathbf{w}) = l(\mathbf{g};\mathbf{y}) + \lambda r(\mathbf{m}) + \mathbf{w}^T(\mathbf{y}-\mathbf{m}) + \frac{\beta}{2}\|\mathbf{y}-\mathbf{m}\|_2^2. \quad (14)$$

PnP ADMM with continuation scheme tries to solve Eq. (14) by iteratively updating $\mathbf{y}, \mathbf{m}$, and the scaled Lagrangian multiplier $\boldsymbol{\omega} = (\frac{1}{\beta})\mathbf{w}$.

1) Update of $\mathbf{y}$*:* The decision variable $\mathbf{m}$ and scaled Lagrangian multiplier $\boldsymbol{\omega}$ are fixed at this stage. Hence the regularizer part can de discarded. The optimization has the following form

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}}\{l(\mathbf{g};\mathbf{y}) + \frac{\beta}{2}\|\mathbf{y} - \mathbf{m}^k + \boldsymbol{\omega}^k\|_2^2\}, \quad (15)$$

which can be solved using Newton's method as given in Eq. (16) and Eq. (17), respectively, for Nakagami and Poisson distributed speckle.

$$\begin{aligned} y_i^{k+1} =& y_i^k - [M(\frac{g_i^2 e^{-z_i}}{2}e^{-y_i^k}) + \beta^k]^{-1} \\ & [M(1 - \frac{g_i^2 e^{-z_i}}{2}e^{-y_i^k}) + \beta^k(y_i^k - m_i^k + \omega_i^k)]. \end{aligned} \quad (16)$$

$$y_i^{k+1} = y_i^k - [e^{z_i}e^{y_i^k} + \beta^k]^{-1}[(e^{z_i}e^{y_i^k} - g_i) + \beta^k(y_i^k - m_i^k + \omega_i^k)]. \quad (17)$$

2) Update of $\mathbf{m}$*:* The primal variable $\mathbf{y}$ and scaled Lagrangian multiplier $\boldsymbol{\omega}$ are frozen at this stage. Hence the forward model can be dropped. The optimization problem leads to

$$\mathbf{m}^{k+1} = \arg\min_{\mathbf{m}}\{\lambda r(\mathbf{m}) + \frac{\beta}{2}\|\mathbf{y}^{k+1} - \mathbf{m} + \boldsymbol{\omega}^k\|_2^2\}. \quad (18)$$

PnP ADMM employs state-of-the-art bounded denoisers [9] to solve the above problems, hence doesn't necessitate any regularizers.

$$\mathbf{m}^{k+1} = \mathcal{D}_{\sigma_k}(\mathbf{y}^{k+1} + \boldsymbol{\omega}^k), \quad (19)$$

where $\sigma_k^2 = \frac{\lambda}{\beta^k}$ is the variance of the denoiser. In this work, standard denoisers such as TV and BM3D are used to implement the above task.

3) Update of $\boldsymbol{\omega}$*:* The scaled multiplier $\boldsymbol{\omega}$ controls the minimizations given by Eq. (17) and Eq. (19). Hence $\boldsymbol{\omega}$ gets updated using

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k + (\mathbf{y}^{k+1} - \mathbf{m}^{k+1}), \quad (20)$$

The proposed technique is briefly discussed in Algorithm 1.

Algorithm 1 Proposed Method for Image Inpainting.

```
Inputs: Provide the following parameters
    – Penalty parameter : $\beta$,
    – Variance controlling parameter : $\lambda$,
    – Variable to update $\beta$ : $\gamma \geq 1$,
    – Binary mask : $\mathbf{H}$,
    – Standard denoiser : $\mathcal{D}_{\sigma_k}(.)$ with $\sigma_k = \sqrt{\frac{\lambda}{\beta^k}}$,
    – Limit of tolerance : tol= $10^{-3}$ and
    – $M = 9$ corresponds to $3 \times 3$ intensity averaging.
Initial Guess:
    – $\beta^0 = \beta$;
    – $\mathbf{y}^0 = \mathbf{m}^0 = $ 2D_shepard_interpolation($\mathbf{g}$);
    – $\boldsymbol{\omega}^0 = \mathbf{0}$;
repeat
    Update $y_i^{k+1}$ using Eq. (16) and Eq. (17) for Nakagami and Poisson distributions respectively.
    $\mathbf{m}^{k+1} = \mathcal{D}_{\sigma_k}(\mathbf{y}^{k+1} + \boldsymbol{\omega}^k)$;
    $\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k + (\mathbf{y}^{k+1} - \mathbf{m}^{k+1})$;
    Calculate the residual $\psi^{k+1} = \frac{1}{\sqrt{N}}\big(\|\mathbf{y}^{k+1} - \mathbf{y}^k\|_2 + \|\mathbf{m}^{k+1} - \mathbf{m}^k\|_2 + \|\boldsymbol{\omega}^{k+1} - \boldsymbol{\omega}^k\|_2\big)$;
if $\psi^{k+1} \geq$ tol then
        $\beta^{k+1} = \gamma\beta^k$;
else
        $\beta^{k+1} = \beta^k$;
end if
    $k = k + 1$;
until stopping criteria is satisfied
Output: $\hat{\mathbf{f}} = e^{\mathbf{y}}$.
```

IV. Experimental Results

This section demonstrates the efficacy of the proposed method by comparing it with state-of-the-art methods. Simulations are performed aiming to achieve image inpainting for SAR images under the effect of speckle noises. The images are normalized in the range between 0 and 1. Noise variance of Rayleigh corrupted speckle is $2v_n^2(1 - \frac{\pi}{4})$, where v_n^2 is the variance of Normal distributions [13]. In the simulation, $v_n^2 = 1$ is considered, hence the noise variance is $2(1 - \frac{\pi}{4})$. Similarly, for Poisson distributed speckle the effect of noise increases by increasing the peak amplitude value in the image. Experiments are performed considering the peak value (P.V.) as 4. All the simulations are conducted on the computer having R2017b version of the MATLAB, with an Intel(R) Core(TM) i5-7500 CPU 3.40GHz, 4 GB RAM and 64-bit operating system installed. Standard denoisers such as TV and BM3D are incorporated as priors in the proposed method. Simulations are

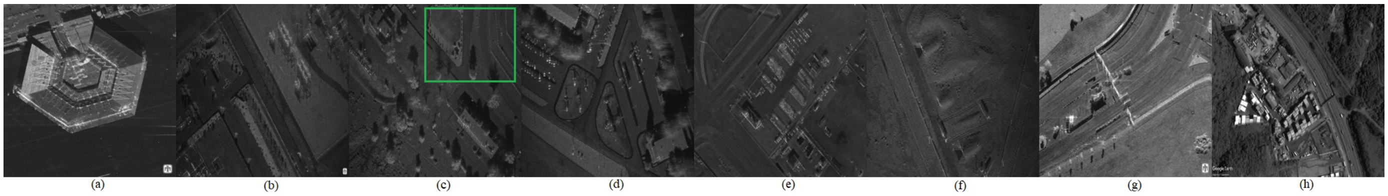


Fig. 1: SAR test images used in simulations.

carried out on 8 grayscale SAR images of different contrasts collected from public database [16] as shown in Fig. 1. The parameter set up is the same as in the case of [13]. The ‘Proposed-PnPTV’ and ‘Proposed-PnPBM3D’ in this section indicate the PnP ADMM for multiplicative noise linked with TV and BM3D denoisers respectively.

A. Performance Metrics

The image assessment metrics such as structural similarity index (SSIM), mean absolute error (MAE) and CPU time are computed and tabulated in Table I to compare the proposed method with other methods. The bold letters marked represents the best value achieved for the specific parameter. SSIM is calculated between the reference image and restored image, and defined as follows:

$$SSIM = \frac{(2\mu_{\mathbf{f}}\mu_{\hat{\mathbf{f}}} + c_1)(2\sigma_{\mathbf{f}\hat{\mathbf{f}}} + c_2)}{(\mu_{\mathbf{f}}^2 + \mu_{\hat{\mathbf{f}}}^2 + c_1)(\sigma_{\mathbf{f}}^2 + \sigma_{\hat{\mathbf{f}}}^2 + c_2)}, \tag{21}$$

where $\mu_{\mathbf{f}}$ and $\mu_{\hat{\mathbf{f}}}$ are the mean values of the reference and the restored images and $\sigma_{\mathbf{f}}^2$ and $\sigma_{\hat{\mathbf{f}}}^2$ correspond to variance in them. c_1 and c_2 are treated as constants. The MAE value is given by

$$MAE = \frac{1}{N}|\mathbf{f} - \hat{\mathbf{f}}|, \tag{22}$$

where N is the number of pixels present in the clean image.

B. Image Inpainting

Synthetically a binary mask is generated, where a random fraction of elements of the mask are zero and multiplied with the clean image. Later, it is corrupted with speckle noise models. This work considers three cases of missed pixels observations, i.e. for $0.5, 0.7$ and 0.95. TV based despeckling methods such as [1] works for no loss of pixels. Hence pixel nearest neighbor (PNN) based interpolation [17] technique is linked with TV regularizer in literature to deal with loss of pixels. The performance metrics are computed to compare the proposed method with PNN-TV [1], [17], image inpainting [4] and blind inpainting [7]. Table I shows a detailed comparison of the proposed work with the above state-of-the-art for image inpainting in the presence of Rayleigh and Poisson noises. The Proposed-PnPBM3D framework gives better result for both the cases. It is faster compared to other methods. The Proposed-PnPTV framework also gives comparable values for SSIM as that of the Proposed-PnPBM3D. For Rayleigh based speckle, PNN-TV [1], [17] gives good results, but it does not perform well when 95% of the pixels are missing. Similarly, in the case of [4], SSIM value is relatively low compared to that of the proposed method in all the cases and it is slowest among all. While for Poisson based speckle, the performance of PNN-TV [1], [17] deteriorates and the blind image inpainting [7] restores the images effectively, but results in loss of important details. A portion of the image of Fig. 1 (c) is magnified to visualize the effect of image reconstruction for all the methods. Fig. 2 (a) displays the Rayleigh noise corrupted image with 70% loss of pixels. Though PNN-TV [1], [17] effectively restores the missed observations, but the effect of noise still exists, which is visible from Fig. 2 (b). The quality of the restored image by [4] as shown in Fig. 2 (c) is low. The proposed PnP ADMM scheme for multiplicative noises when linked with TV and BM3D denoisers effectively suppresses the noise and restore the missing pixels as shown in Fig. 2 (d) and (e) respectively. The horizontal profiles of all the methods is shown in Fig. 2 (f), which clearly states that the proposed method performs better compared to that of state-of-the-art methods.

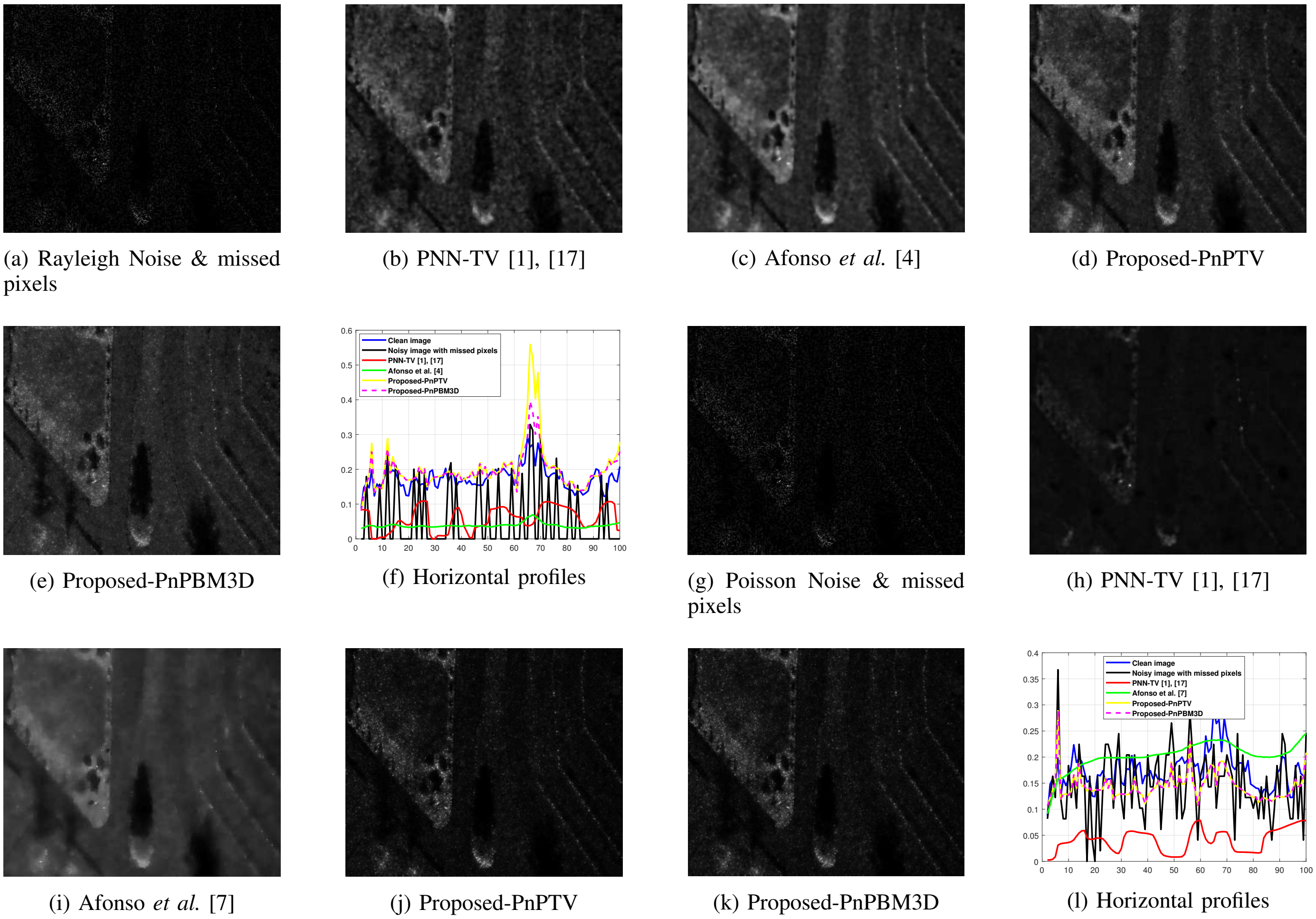
Similarly, for Poisson speckle, the noise contaminated image along with 70% loss of pixels is shown in Fig. 2 (g). The ability to restore the image for PNN-TV [1], [17] is lowest compared to other methods as shown in 2 (h). Reconstructed image by [7] as shown in Fig. 2 (i) is highly blurred, resulting in loss of important details. The restored images using the proposed method with TV and BM3D denoisers are shown in Fig. 2 (j) and (k) respectively. Fig. 2 (l) displays the horizontal profiles of all the methods, and the proposed PnP ADMM scheme with BM3D denoiser outperforms the other methods. The noise effect is suppressed as well as the fractions of missing pixels are properly restored in both the cases. It is evident that the accuracy of the proposed method is high compared to [7].

V. Conclusion and Future work

A plug-and-play prior based approach is proposed in this work to address the problem of image inpainting in the presence of multiplicative noise models. Intensity averaging is incorporated for Rayleigh corrupted speckle, which changes it into a Nakagami based model to improve the performance. MAP estimation is performed to effectively tackle both Rayleigh and Poisson distributions. PnP ADMM’s continuation scheme, when linked with standard denoisers, gives suitable image inpainting results. The advantage of the proposed framework is that it is faster and can reconstruct up to 95% of missing pixels. Future research includes the implementation of RED based prior in the presence of multiplicative noise models. Variance stabilization methods may be applied to convert the multiplicative nature of noise into additive one and further state-of-the-art techniques can be applied to restore it.

TABLE I: Image inpainting for Rayleigh and Poisson distributed speckle.

Noise Models	Missing Fractions	Parameters	Methods	SAR Test Images								Average Value
				(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	
Rayleigh with intensity averaging of 3 × 3 mask	0.5	SSIM	PNN-TV [1], [17]	0.675	0.682	0.658	0.676	0.654	0.597	0.652	0.617	0.651
			Afonso *et al.* [4]	0.350	0.324	0.312	0.323	0.325	0.298	0.340	0.286	0.319
			Proposed-PnPTV	0.822	0.714	0.680	0.708	0.774	0.758	0.798	0.878	0.766
			Proposed-PnPBM3D	0.828	0.847	0.840	0.847	0.801	0.788	0.808	0.889	**0.831**
		MAE	PNN-TV [1], [17]	0.077	0.058	0.062	0.059	0.073	0.065	0.068	0.087	0.068
			Afonso *et al.* [4]	0.178	0.141	0.148	0.140	0.180	0.163	0.150	0.178	0.159
			Proposed-PnPTV	0.058	0.053	0.057	0.051	0.052	0.082	0.051	0.064	0.058
			Proposed-PnPBM3D	0.052	0.028	0.032	0.031	0.041	0.038	0.047	0.059	**0.041**
		CPU Time (in sec.)	PNN-TV [1], [17]	5.563	2.288	5.494	5.491	2.954	0.483	2.619	11.079	4.496
			Afonso *et al.* [4]	9.331	6.112	14.156	16.277	3.387	7.137	7.232	17.406	10.129
			Proposed-PnPTV	6.336	2.991	6.952	6.803	5.108	6.912	4.560	11.257	6.364
			Proposed-PnPBM3D	3.251	1.494	3.422	3.389	1.876	3.424	2.396	8.301	**3.444**
	0.7	SSIM	PNN-TV [1], [17]	0.351	0.353	0.337	0.352	0.307	0.300	0.320	0.314	0.329
			Afonso *et al.* [4]	0.335	0.317	0.305	0.315	0.315	0.289	0.330	0.271	0.309
			Proposed-PnPTV	0.826	0.727	0.691	0.717	0.598	0.584	0.762	0.878	0.722
			Proposed-PnPBM3D	0.829	0.842	0.826	0.836	0.804	0.758	0.799	0.887	**0.822**
		MAE	PNN-TV [1], [17]	0.135	0.105	0.111	0.106	0.129	0.121	0.116	0.143	0.120
			Afonso *et al.* [4]	0.178	0.141	0.148	0.140	0.180	0.163	0.151	0.178	0.159
			Proposed-PnPTV	0.050	0.049	0.052	0.056	0.046	0.068	0.055	0.058	0.054
			Proposed-PnPBM3D	0.046	0.025	0.030	0.028	0.037	0.035	0.043	0.053	**0.037**
		CPU Time (in sec.)	PNN-TV [1], [17]	5.480	2.290	5.494	5.484	2.937	5.531	2.629	11.031	5.109
			Afonso *et al.* [4]	9.047	3.554	8.189	8.281	4.052	7.540	3.872	21.931	8.308
			Proposed-PnPTV	6.665	3.287	9.015	6.564	4.959	6.955	4.820	11.279	6.693
			Proposed-PnPBM3D	3.549	1.777	3.747	3.758	2.112	3.711	2.419	8.351	**3.678**
	0.95	SSIM	PNN-TV [1], [17]	0.211	0.211	0.210	0.211	0.211	0.210	0.213	0.207	0.210
			Afonso *et al.* [4]	0.295	0.294	0.281	0.292	0.285	0.266	0.300	0.209	0.277
			Proposed-PnPTV	0.707	0.607	0.552	0.594	0.512	0.539	0.661	0.707	0.609
			Proposed-PnPBM3D	0.711	0.712	0.683	0.700	0.686	0.570	0.679	0.719	**0.682**
		MAE	PNN-TV [1], [17]	0.240	0.182	0.191	0.182	0.233	0.210	0.198	0.237	0.209
			Afonso *et al.* [4]	0.179	0.141	0.148	0.141	0.181	0.163	0.152	0.180	0.160
			Proposed-PnPTV	0.049	0.050	0.053	0.055	0.089	0.070	0.054	0.064	0.060
			Proposed-PnPBM3D	0.043	0.027	0.031	0.030	0.035	0.035	0.045	0.057	**0.037**
		CPU Time (in sec.)	PNN-TV [1], [17]	5.562	2.281	5.479	5.475	2.944	5.504	2.632	11.177	5.131
			Afonso *et al.* [4]	26.750	5.062	12.445	12.097	7.031	12.641	7.283	44.061	15.921
			Proposed-PnPTV	5.856	3.271	7.807	7.842	4.613	7.211	4.964	11.168	6.591
			Proposed-PnPBM3D	3.664	1.434	3.797	3.884	2.020	3.804	2.482	8.549	**3.704**
Poisson with P.V.=4	0.5	SSIM	PNN-TV [1], [17]	0.467	0.516	0.527	0.516	0.471	0.467	0.500	0.369	0.479
			Afonso *et al.* [7]	0.774	0.712	0.687	0.700	0.735	0.641	0.704	0.616	0.696
			Proposed-PnPTV	0.860	0.810	0.801	0.804	0.796	0.752	0.775	0.818	0.802
			Proposed-PnPBM3D	0.862	0.818	0.813	0.815	0.807	0.770	0.778	0.818	**0.810**
		MAE	PNN-TV [1], [17]	0.146	0.103	0.100	0.101	0.139	0.114	0.111	0.149	0.120
			Afonso *et al.* [7]	0.030	0.026	0.056	0.032	0.058	0.043	0.057	0.045	**0.043**
			Proposed-PnPTV	0.050	0.039	0.041	0.040	0.062	0.047	0.044	0.059	0.047
			Proposed-PnPBM3D	0.049	0.038	0.039	0.039	0.061	0.046	0.044	0.058	0.046
		CPU Time (in sec.)	PNN-TV [1], [17]	5.510	2.268	5.535	5.532	2.954	5.552	2.629	11.008	5.123
			Afonso *et al.* [7]	34.681	14.980	33.796	35.731	14.081	31.495	17.518	59.635	30.239
			Proposed-PnPTV	7.636	4.529	7.968	7.656	7.349	8.705	4.658	9.825	7.290
			Proposed-PnPBM3D	5.213	2.201	5.363	5.278	2.739	5.295	2.407	8.454	**4.618**
	0.7	SSIM	PNN-TV [1], [17]	0.223	0.241	0.247	0.236	0.212	0.225	0.227	0.160	0.221
			Afonso *et al.* [7]	0.737	0.697	0.663	0.686	0.710	0.626	0.683	0.558	0.670
			Proposed-PnPTV	0.840	0.784	0.772	0.776	0.773	0.709	0.731	0.782	0.770
			Proposed-PnPBM3D	0.841	0.790	0.781	0.785	0.780	0.723	0.757	0.784	**0.780**
		MAE	PNN-TV [1], [17]	0.190	0.142	0.145	0.141	0.184	0.160	0.153	0.191	0.163
			Afonso *et al.* [7]	0.031	0.033	0.046	0.035	0.053	0.035	0.052	0.048	**0.041**
			Proposed-PnPTV	0.048	0.039	0.041	0.041	0.063	0.048	0.049	0.059	0.048
			Proposed-PnPBM3D	0.048	0.039	0.040	0.040	0.062	0.047	0.043	0.055	0.046
		CPU Time (in sec.)	PNN-TV [1], [17]	5.510	2.288	5.623	5.477	2.964	5.517	2.747	10.954	5.135
			Afonso *et al.* [7]	36.020	19.869	36.871	45.303	19.003	33.324	23.400	63.024	34.601
			Proposed-PnPTV	8.274	4.390	8.316	7.752	7.308	8.731	4.419	10.665	7.481
			Proposed-PnPBM3D	5.096	2.222	5.306	5.366	2.780	5.309	2.447	8.514	**4.630**
	0.95	SSIM	PNN-TV [1], [17]	0.205	0.206	0.207	0.207	0.205	0.206	0.209	0.203	0.206
			Afonso *et al.* [7]	0.583	0.489	0.585	0.595	0.625	0.578	0.594	0.336	0.548
			Proposed-PnPTV	0.719	0.678	0.648	0.662	0.641	0.558	0.642	0.624	0.646
			Proposed-PnPBM3D	0.720	0.680	0.654	0.668	0.647	0.564	0.645	0.626	**0.650**
		MAE	PNN-TV [1], [17]	0.242	0.183	0.192	0.183	0.234	0.211	0.199	0.239	0.210
			Afonso *et al.* [7]	0.070	0.212	0.059	0.043	0.063	0.058	0.053	0.091	0.081
			Proposed-PnPTV	0.054	0.043	0.046	0.045	0.069	0.053	0.049	0.067	0.053
			Proposed-PnPBM3D	0.053	0.042	0.044	0.044	0.068	0.052	0.049	0.065	**0.052**
		CPU Time (in sec.)	PNN-TV [1], [17]	5.467	2.321	5.531	5.491	2.937	5.454	2.674	10.931	5.100
			Afonso *et al.* [7]	136.471	61.199	140.422	138.861	77.060	135.621	68.734	231.171	123.692
			Proposed-PnPTV	8.942	4.398	6.881	7.439	7.478	9.429	4.819	10.806	7.524
			Proposed-PnPBM3D	5.241	2.205	5.312	5.329	2.810	5.321	2.491	8.750	**4.682**



(a) Rayleigh Noise & missed pixels

(b) PNN-TV [1], [17]

(c) Afonso *et al.* [4]

(d) Proposed-PnPTV

(e) Proposed-PnPBM3D

(f) Horizontal profiles

(g) Poisson Noise & missed pixels

(h) PNN-TV [1], [17]

(i) Afonso *et al.* [7]

(j) Proposed-PnPTV

(k) Proposed-PnPBM3D

(l) Horizontal profiles

Fig. 2: SAR image inpainting (70% loss of pixels) with Rayleigh and Poisson contaminated speckle.

References

[1] J. Seabra, J. Xavier, and J. Sanches, “Convex ultrasound image reconstruction with log-euclidean priors,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 435–438, Aug 2008.

[2] M. Neri and E. Zara, “Total variation-based image inpainting and denoising using a primal-dual active set method,” *Philippine Science Letters*, vol. 7, no. 1, pp. 97–103, 2014.

[3] M. V. Afonso and J. M. Sanches, “A total variation based reconstruction algorithm for 3d ultrasound,” in *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 149–156, Springer, 2013.

[4] M. Afonso and J. M. Sanches, “Image reconstruction under multiplicative speckle noise using total variation,” *Neurocomputing*, vol. 150, pp. 200–213, 2015.

[5] A. Chambolle, “An algorithm for total variation minimization and applications,” *Journal of Mathematical imaging and vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[6] X. Zeng and S. Li, “An efficient adaptive total variation regularization for image denoising,” in *2013 Seventh International Conference on Image and Graphics*, pp. 55–59, July 2013.

[7] M. V. Afonso and J. M. R. Sanches, “Blind inpainting using ℓ_0 and total variation regularization,” *IEEE Transactions on Image Processing*, vol. 24, pp. 2239–2253, July 2015.

[8] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-play priors for model based reconstruction,” in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, Dec 2013.

[9] S. H. Chan, X. Wang, and O. A. Elgendy, “Plug-and-play admm for image restoration: Fixed-point convergence and applications,” *IEEE Transactions on Computational Imaging*, vol. 3, pp. 84–98, March 2017.

[10] S. Ono, “Primal-dual plug-and-play image restoration,” *IEEE Signal Processing Letters*, vol. 24, pp. 1108–1112, Aug 2017.

[11] A. Brifman, Y. Romano, and M. Elad, “Turning a denoiser into a super-resolver using plug and play priors,” in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1404–1408, Sep. 2016.

[12] J. C. Seabra, F. Ciompi, O. Pujol, J. Mauri, P. Radeva, and J. Sanches, “Rayleigh mixture model for plaque characterization in intravascular ultrasound,” *IEEE Transactions on Biomedical Engineering*, vol. 58, pp. 1314–1324, May 2011.

[13] S. Baraha and A. K. Sahoo, “Sar image despeckling using plug-and-play admm,” *IET Radar, Sonar Navigation*, vol. 14, no. 9, pp. 1297–1309, 2020.

[14] C. Yu, C. Zhang, and L. Xie, “A multiplicative nakagami speckle reduction algorithm for ultrasound images,” *Multidimensional Systems and Signal Processing*, vol. 23, no. 4, pp. 499–513, 2012.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] “Sandia National Laboratories.” https://www.sandia.gov/radar/imagery/index.html/. Accessed: 2020-04-01.

[17] R. Rohling, A. Gee, and L. Berman, “A comparison of freehand three-dimensional ultrasound reconstruction techniques,” *Medical Image Analysis*, vol. 3, no. 4, pp. 339 – 359, 1999.