SAR Despeckle Filtering by Sparse Coding on Affinity Nets (SCAN)

Porikli, F.; Sundaresan, R.; Suwa, K.

TR2012-029 April 2012

Abstract

This paper presents a new approach for multiplicative noise removal in SAR images based on sparse coding by dictionary learning and collaborative filtering. First, an affinity net is formed by clustering log-similar image patches where a cluster is represented as a node in the net. For each cluster, an under-complete dictionary is computed using the alternative decision method that iteratively updates the dictionary and the sparse coefficients. The nodes belonging to the same cluster are then reconstructed by a sparse combination of the corresponding dictionary atoms. The reconstructed patches are finally collaboratively aggregated to build the denoised image. Experimental results demonstrate superior despeckle filtering performance.

Synthetic Aperture Radar, 2012. EUSAR. 9th European Conference on

SAR Despeckling by Sparse Reconstruction on Affinity Nets (SRAN)

Fatih Porikli*, Rajagopalan Sundaresan*, Kei Suwa†
*Mitsubishi Electric Research Labs, USA, †Mitsubishi Electric Corporation, Japan

Abstract

This paper presents a new approach for multiplicative noise removal in SAR images based on sparse coding by dictionary learning and collaborative filtering. First, an affinity net is formed by clustering log-similar image patches where a cluster is represented as a node in the net. For each cluster, an under-complete dictionary is computed using the alternative decision method that iteratively updates the dictionary and the sparse coefficients. The nodes belonging to the same cluster are then reconstructed by a sparse combination of the corresponding dictionary atoms. The reconstructed patches are finally collaboratively aggregated to build the denoised image. Experimental results demonstrate superior despeckle filtering performance.

1 Introduction

Synthetic Aperture Radar (SAR) images are produced by transmitting electromagnetic waves and coherently integrating the received pulses. Because of the nature of data acquisition, SAR images are inherently affected by speckle noise and artifacts. Image despeckling problem has been well studied over the years [1, 2, 3, 4, 5, 6, 11] and it is still a popular research area thanks to the recently introduced concepts of nonlocal means [7] and dictionary learning [8]. Speckle is often modeled as multiplicative noise given as follows

$$y_k = x_k * z_k \tag{1}$$

where y_k, x_k and z_k correspond to the contaminated intensity, the original intensity, and the noise level of the k-th pixel, respectively. Several methods have suggested first using a log transform to convert the multiplicative noise into an additive representation and then to apply additive noise filtering strategies.

The wavelet shrinkage techniques make use of a set of predefined complete wavelet bases (decimated and undecimated) on the log-intensity image where the wavelet transform coefficients are thresholded (sometimes called as coring) in an adaptive fashion to remove the higher frequency variation, which is typically assumed to be the noise. Instead of using complete bases, Foucher [2] extracts a global over-complete dictionary from the given data and imposes sparsity on the representation in order to suppress the incoherent noise. Even though it may generate slightly better results than the wavelet shrinkage techniques for Gaussian noise, the use of a single over-complete dictionary causes over-smoothing especially when the input vs reconstruction blending parameter is not preset accurately.

Recently, Delledale et.al [3] introduced a probabilistic patch based (PPB) filter based on the nonlocal means approach, posing the denoising task as a weighted maximum likelihood estimation problem. The nonlocal means approach estimates a pixel by computing the weighted average of a set other patches in the image. These weights are directly proportional to the similarity of the patch surrounding the reference pixel and the patches surrounding the candidate pixels in a preset search window. The idea is to refine iteratively these weights by including patches from the estimate of the image parameters. PPB can cope with non-Gaussian noise; however, it also removes fine and low-intensity details.

To prevent from over-smoothing and removal of important details, here we propose an alternative strategy to suppress multiplicative noise. We take advantage of locally adaptive dictionary learning on an affinity net of log-similar image patches in a collaborative filtering framework. Our intuition is that the use of under-complete dictionaries for patch clusters would establish flexible representations of the local appearance variations while efficiently rejecting the outliers, and the aggregation of similar patch reconstructions would provide texture consistency not only over the local patch but also over the entire image. In other words, the fine and statistically meaningful details are preserved through the cluster dictionaries and projected back on the denoised image, and the incoherent appearances are restrained through the collaborative aggregation process.

2 SRAN Algorithm

Since the input image is contaminated with the multiplicative noise, we first apply a log-transform to use additive filtering in the log-intensity domain instead of multiplicative filtering in the intensity domain. We construct an appearance similarity graph, called as affinity net, for the log-transformed image. In this affinity net, each node corresponds to an image patch and each vertex to the similarity between two patches.

For each node, we find the cluster of similar nodes. We form a cluster matrix and decompose it into a linear function (matrix multiplication) of a codebook (dictionary) and its sparse coding coefficients. The dictionary is designed to best represent the coherent variations along same pixel locations in the cluster. To achieve this we set the size of the dictionary smaller than the rank of the cluster matrix. As a result, we represent and filter each cluster by its own unique small under-complete dictionary.

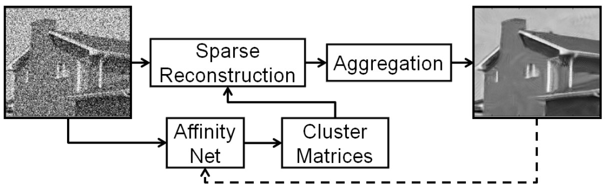

Figure 1: Schematic of our SRAN algorithm.

After learning the dictionary for the current cluster, we reconstruct the nodes in the cluster by the orthogonal matching pursuit to obtain the sparse representations. Since a single node could participate in different clusters, there will be overlaps among the clusters, thus, we obtain multiple estimates for a pixel. We compute the final denoised image by aggregating these multiple estimates; this aggregation imposes a strong collaborative constraint and efficiently limits the amount of the inconsistent variations due to the noise without excessive smoothing. In order to further improve the initial affinity net construction, we use the filtered image to determine the clusters and apply sparse reconstruction and aggregation on the non-filtered image. Figure 1 illustrates these filtering stages.

2.1 Affinity Net

The affinity net is a weighted undirected graph where the weights correspond to the similarity between the patches. We arrange the pixels of a patch into a vector form hence our feature space is the intensity values where a point in the feature space represents a particular patch appearance. We define the similarity between two node vectors p_i and p_j as

$$\omega(p_i, p_j) = < p_i, p_j > = \sum_{k=1}^{K} p_{i,k} p_{j,k} \tag{2}$$

where k is the pixel index with respect to the patch grid and K is the patch size, i.e. the number of pixels contained in a patch. Although it is possible to use other metrics (e.g. ℓ_1 weights, frequency features, etc.) to measure the similarity between patches, we preferred the dot product to not bias for any particular intensity and pixel position.

We use the affinity net to learn a locally adapting dictionary for each node. However, using the entire affinity net to learn that many dictionaries would be computationally prohibitive since the affinity net contains as many clusters as the number of image pixels and it is fully connected. In order to decrease the computational load drastically and to concentrate on more relevant samples, we cluster nodes by keeping a maximum of m vertices that have sufficiently high weights $w(p_i, p_j) > \tau$ within a search window of radius r. A cluster c_i is denoted as

$$c_i = \{ \ p_j : \omega(p_i, p_j) > \tau, i = 1, .., n \ \} \tag{3}$$

for n pixels in image Y. Clustering process puts the nodes having close coordinates together in the feature space. We empirically observed that the noise removal performance is not sensitive to the value of τ as long as most clusters contain 10 to 100 patches. If the affinity between two nodes is high, a node is attached to the cluster of the other node as illustrated in Fig. 2. Depending on the self-similarity of the image, clusters can be in different sizes. A reference node might not contain any similar nodes and hence the cluster size will be one. Also, a single node could be present in one or more clusters and participate in filtering of each cluster they are present. In this case, we will obtain multiple estimates of a node.

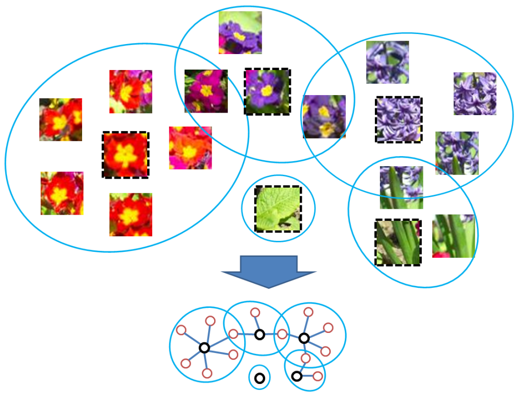

Figure 2: Illustration of an affinity net.

To obtain a more robust similarity measurement and clustering, we construct the affinity net on the filtered result of our method. In other words, we iterate our algorithm to refine the affinity net weights: the first time to get an initial estimate, and the second time to find the final denoised image using the initial estimate. Alternatively, it is also possible to apply a hard thresholding scheme (for instance, the first stage of [10]) to get the initial estimate.

2.2 Sparse Reconstruction

For each the affinity net cluster c_i, we form a cluster matrix $C_i^{M \times K}$ and decompose it into a linear function of a dictionary $D_i^{M \times d}$ and its sparse coefficients $X_i^{d \times K}$, where M is the number of nodes in the cluster, d is the number of atoms in the dictionary, and K is the number of pixels in the patch. We arrange the column vectors p_i of the cluster nodes into a cluster matrix as $C_i = [.., p_i, ..]^T$. In C_i, each row corresponds to cluster node including the reference node. The number of columns in this matrix is the dimensionality (size) of the patch, while the number of rows is the number of patches in the cluster.

As opposed to the traditional column-wise ordering, we use the row-wise ordering before learning the dictionary.

Our intuition here is that the variation along the columns, that is, the intensity variation of the same pixel locations across the similar patches, should be very small since the patches are all similar. A pixel intensity that does not comply with its references in the other patches in the cluster should not be considered as a representative in the dictionary.

Note that we are not aiming for a perfect reconstruction but rather elimination of the inconsistent pixel intensities by a linear combination of a few atoms in the dictionary. Because of this special row-wise structure of the cluster matrix, a small dictionary can successfully capture the coherent pixel intensity changes. Our cluster dictionary is therefore significantly under-complete; the number of columns d (and the maximum rank of the dictionary) is less than the patch size $d << K$.

We jointly learn the dictionary and the sparse coefficients by alternating between a sparse coding stage and a dictionary update stage to solve the optimization problem

$$D_i^*, X_i^* = \arg\min_{D_i, X_i} ||x_j||_0 \,, \forall j \;\; ||C_i - D_i X_i||_2^2 \leq \epsilon \quad (4)$$

where x_j is the coefficient column vector corresponding to pixel j in the patch grid and ϵ is the maximum allowed sparsity. We set the initial dictionary matrix with d random and ℓ_2 normalized columns of C_i. Alternatively, the k-means clustering (with d centers) can be applied to cluster nodes to obtain the initial dictionary.

In the sparse coding stage, we use the orthogonal matching pursuit to compute the representation vectors x_j. We update the dictionary one column at a time by first defining the group of rows (pixels) that use this atom then computing the overall representation error matrix for this group without using the current atom, and then applying singular value decomposition to assign the first column of the left decomposition matrix as the new updated atom.

For the clusters that the number of nodes is very small (e.g. $M < 4$), we apply the Wiener in the second time application of our algorithm. We estimate the noise variance from the difference between the first time application result and the original noisy image.

After the dictionary learning and sparse coefficient computation, we have the new cluster matrix C_i^*. The rows of C_i^* are the reconstructed patches. We assign these reconstructed patches to the respective locations from where they are extracted. Because patches are overlapping and a patch may belong to multiple clusters, there are multiple estimates for a given pixel. The final estimate at a pixel is obtained by computing the average of all computed estimates. This collaborative aggregation enables preserving commonly shared patterns and rejecting speckle noise at the same time.

Since the noisy image is log-transformed in the first step, we subtract the nonzero mean [9] of the noise and take the exponent of the final estimate to produce the filtered result [4]. The above procedure is summarized in the following algorithm.

```
**Input**: noisy image $\log Y$, initial estimate $Y_0 = Y$
**Output**: denoised image $I$
cluster patches $p_i \in Y_0 \rightarrow c_i$
**for** each $c_i$ **do**
  arrange $p_i \in Y$ (not $Y_0$) into cluster matrix $C_i$ using $\omega(p_i, p_j)$ from $Y_0$
  **if** $M < 4$ and first run **then**
    $C_i^* \leftarrow$ Wiener($C_i$)
  **else**
    $C_i^* = D_i^*, X_i^* \leftarrow$ Eq. 4
  **end if**
**end for**
$Y_0 \leftarrow$ back project and aggregate $C_i^*$.
second run: subtract nonzero mean, $I = \exp Y_0$ .
```

3 Experimental Results

For quantitative assessment we tested the SRAN on natural images and for qualitative evaluation we used real SAR data, which lack a ground truth and hence the quality of filtering is generally observed by visual inspection. For natural images, we have access to the original clean image. We compare our method against the iterative and the non-iterative PPB [3], Foucher's K-SVD based speckle removal [2], and the log-BM3D filter [10] (also reported in [11]). These filters report the best multiplicative noise removal performance in the recent literature.

For multiplicative noise simulation, we considered one-look $L = 1$ SAR scenario where we modeled the noise as a Nakagami-Rayleigh distribution [9]. We set the patch size $K = 64 = 8 \times 8$ and the search range $r = 39$. We limited the minimum similarity between nodes to $\tau = 0.002$ and the maximum number of similar nodes in a cluster to $m = 400$. We used a simple hard thresholding scheme suggested in [10] to obtain the initial estimate.

To investigate the performance on the natural images we used the Structural Similarity Index Metric (SSIM), which is highly popular due to its consistency with the human eye perception. SSIM concerns luminance, contrast and structure and aims to evaluate the perceptual quality of an approximate image comparing to the original one.

Figure 4 show sample results, and Table 1 summarizes the SSIM and PSNR scores where the best scores are indicated in bold. From the table it is evident that the SRAN filter performs better than the Foucher [2] and the non-iterative PPB filters. When we compare the PSNR values, our method is +2 dB better than Foucher and gives comparable scores to the iterative PPB on average. To our advantage, SRAN is qualitatively more accurate than all other filters (including iterative PPB) at preserving fine details.

We also show results for a sample SAR image obtained from SANDIA website [12]. Since there is no ground truth for SAR data, we rely on visual inspection to compare different techniques. As we can see, SRAN does an excellent job in preserving the details particularly around the stadium and roads.

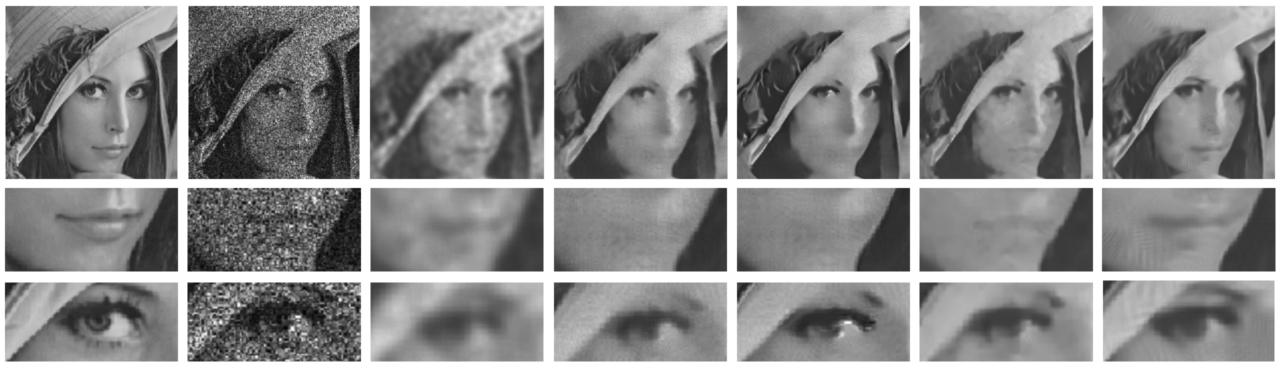

Figure 4: Foucher's method fails to remove despeckle noise effectively: it causes too much blur and uneven response causing artificial texture in smooth regions. PPB and PPB with 25 iteration both over-smoothen the fine texture and remove important details as can be seen in the mouth region. PPB with 25 iterations on the other hand, produces undesirable hallucinated textures as visible in the over emphasized white pixels around the pupil. Log-BM3D causes blocky blobs and tends to blur low intensity texture. Qualitatively, SRAN results are preferable to all others.

4 Conclusion

We presented a SAR speckle filter based on the adaptive dictionary learning on the affinity net. Instead of a single over-complete dictionary, our filter uses multiple under-complete dictionaries for patch clusters. SRAN provides quantitative scores on par with or higher than the state-of-the-art, yet qualitatively superior performance.

Table 1: PSNR and SSIM comparisons of various filters

PSNR / SSIM	Barbara	Boat	House	Lena
Noisy	-1.05 0.19	-2.97 0.15	-3.57 0.09	-2.43 0.12
Foucher [2]	8.26 0.55	7.78 0.54	8.13 0.64	9.74 0.67
PPB [3]	9.80 0.60	8.64 0.53	9.25 0.59	11.06 0.63
PPB (iter.) [3]	10.67 0.65	**9.52** 0.57	10.59 0.64	**12.28** 0.68
log-BM3D [10]*	9.46 0.63	9.02 0.59	9.90 **0.73**	11.08 **0.73**
SRAN	**10.80** **0.68**	9.46 **0.59**	**10.96** 0.67	11.84 0.70

* Similar version is presented in [11].

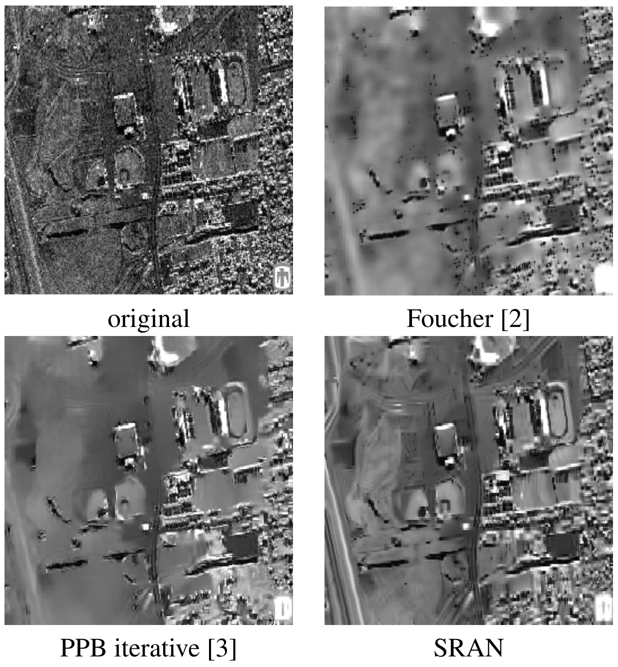

Figure 3: Only SRAN preserves important details.

References

[1] A. Achim, P. Tsakalides, A. Bezarianos, *SAR image denoising via Bayesian wavelet Shrinkage based on heavy tiled modeling*, IEEE Transactions on Geoscience and Remote Sensing, vol.41, no.8, 2003.

[2] S. Foucher, *SAR image filtering via learned dictionaries and sparse representations*, IEEE International Remote Sensing and Geoscience Symposium, 2008.

[3] C. Delledale, L. Denis, F. Tupin, *Iterative Weighted Maximum Likelihood Denoising with Probabilistic Patch-Based Weights*, IEEE Transactions on Image Processing, vol.18, no.12, 2009.

[4] M. Bhuiyan, M. Ahmad, M. Swamy, *Spatially adaptive wavelet method using the Cauchy prior for denoising the SAR images*, IEEE Transactions on Circuits and Systems for Video Technology, vol.17, no.4, 2007.

[5] F. Argenti, A. Alparone, *Speckle removal from SAR images in the undecimated wavelet domain*, IEEE Transactions on Geoscience and Remote Sensing, vol.40, 2002.

[6] F. Argenti, T. Bianchi, A. Alparone, *Multiresolution MAP despeckling of SAR images based on locally adaptive generalized Gaussian pdf modeling*, IEEE Transactions on Image processing, vol.15, no. 11, 2006.

[7] A. Buades, B. Coll, J. Morel, *A non local algorithm for image denoising*, IEEE Conf. Computer Vision and Pattern Recognition, 2005.

[8] M. Elad, M. Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, IEEE Transactions on Image Processing, vol.15, no.12, 2006.

[9] H. Xie, L. Pierce, F. Ulaby, *Statistical properties of logarithmically transformed speckle*, IEEE Transactions on Geoscience and Remote Sensing, vol.40, no.3, 2002.

[10] K. Dabov, A. Foi, V. Katvonik, K. Egiazarian, *Image denoising by sparse 3D domain collaborative filtering*, IEEE Transactions on Image Processing, vol.16, no.8, 2007.

[11] S. Parrilli, M. Poderico, C. Angelino, G. Scarpa, L. Verdoliva, *A nonlocal approach for SAR image denoising*, IEEE International Geoscience and Remote Sensing Symposium, 2010.

[12] http://www.sandia.gov/RADAR/imagery.html