Speckle Removal and Change Preservation by Distance–Driven Anisotropic Diffusion of Synthetic Aperture Radar....

Article in Digital Signal Processing · December 2017
DOI: 10.1016/j.dsp.2017.11.013

3 authors, including:

Nazia Tabassum
University of Virginia
2 PUBLICATIONS 1 CITATION

Andrea Vaccari
University of Virginia
39 PUBLICATIONS 241 CITATIONS

Some of the authors of this publication are also working on these related projects:

RITARS-14-H-UVA - InSAR remote sensing for performance monitoring of transportation infrastructure at the network level View project

cooperative behavior in fruit fly larvae View project
Speckle Removal and Change Preservation by Distance-Driven Anisotropic Diffusion of Synthetic Aperture Radar Temporal Stacks

Nazia Tabassum, Andrea Vaccari, Scott Acton
University of Virginia, Charlottesville, Virginia, USA.

Abstract
We propose a method that smoothes spatially and uses, but also preserves, temporal information of synthetic aperture radar image stacks. To provide smoothing of such imagery without effacing temporal changes in the scene, we put forth an anisotropic diffusion technique, using a PDE approach. This approach smooths uniform areas and preserves and enhances edges, such as roads or other features. In order to use this anisotropic diffusion technique, a homogeneous region must first be selected in the image to calculate statistics of the speckle noise. We propose a method that uses the temporal information to automatically select a homogeneous region prior to smoothing. Our proposed smoothing method calculates the distance between either pixel time-series or pixel CDFs as a measure of similarity, or uniformity. Results demonstrate the efficacy of the approach on real and synthetic data, showing lower mean squared ratio and higher structural similarity index than leading methods, as well as temporal change preservation.

Keywords: SAR imagery, Speckle, Anisotropic diffusion, Remote sensing, 3D denoising

1. Introduction

1.1. SAR Imagery

Synthetic aperture radar (SAR) is a radar technique used for imagery [1][2][3]. These images can be used in aid of remote sensing, such as in the prevention of transportation infrastructure calamities [4]. SAR images are corrupted with a noise type called speckle [5]. Speckle is an inherent corruptive process at the

*This work was supported in part by the U.S. Department of Transportation (US DOT) under Cooperative Agreement #RITARS-14-H-UVA. The views, opinions, findings and conclusions reflected in this presentation are the responsibility of the authors only and do not represent the official policy or position of the US Department of Transportation/Office of the Assistant Secretary for Research and Technology, or any state or other entity.
heart of coherent imaging analysis such as radar, SAR, and ultrasound tech-
iques. Speckle occurs when surfaces that are being imaged are rough smaller
than the scale of the wavelength, and can be modeled as multiplicative noise
[6]. Because of this, brighter regions in the image have noise values with higher
intensity, and darker regions have noise values with lower intensity. To mini-
mize the effects of speckle, often an average of a certain number of ”looks” is
taken. One ”look” is one image of a specific location typically acquired over one
period of the satellite’s orbit. If a few looks are averaged together, the noise can
be decreased - however, any changes that occurred between looks will be lost
during this averaging process.

1.2. 2D Noise Removal Methods

Most noise removal methods rely on smoothing, which makes sharp changes
in pixel intensities more subtle. As a result, the noisy pixels begin to take on
intensities closer to those in the surrounding regions of the image. However,
blurring pixel gradients can lead to loss of edges. We will show that our method
removes noise yet retains important edges of roads and other objects.

The leading speckle removal algorithms are adaptive speckle filters, such as
Gamma MAP, Lee, Frost, Kuan, etc. [7, 8, 9, 10] These filters have a number
of drawbacks. First, these filters do not enhance edges, unlike other methods,
such as SRAD. They only prevent smoothing near or over edges. Furthermore,
since here filters become all-pass in the presence of an edge, any noise along an
edge or on top of an edge will not be removed [11].

Secondly, these filters are non-directional. Near an edge, smoothing is com-
pletely inhibited. A more sensible approach is to only inhibit smoothing per-
pendicular to the edge, so that the edge is not smoothed over.

Thirdly, there are hard thresholds used in all of the above filters. The filters
average areas of homogeneity, detected by thresholds, based on the assumption
that the original signal is equal to the average of a homogeneous region in
the noisy image [12, 13]. These thresholds may lead to artifacts produced by
this averaging (blotchy patches) and also noisy edges because of strict all-pass
filtering.

Lastly, these filters are all window-based approaches, and thus are dependent
on window size and type. Larger windows can lead to over-smoothing, which
registers as a blurred effect on the image. Smaller windows may not smooth
enough and can leave a large amount of noise behind. Choosing the correct
window size for each application may be a difficult problem to optimize. As for
window shape, most common approaches will use a square window, which leads
to rounding of features that are found in different orientations than the window.

Aside from adaptive filtering techniques, there are also simulated anneal-
ing based algorithms for image denoising. These methods promote a series of
random changes to the noisy image to approximate the maximum a posteriori
estimate of the ground truth image. Annealing methods are effective at remov-
ing multiplicative noise; however, there is often an assumption of near piecewise
constancy made that can result in over-smoothing of large areas. Speckle along
edges is also not reduced. Annealing techniques take a large number of iterations to converge. \[14, 15\]

Another popular noise removal technique is called anisotropic diffusion \[16\]. Anisotropy refers to not treating all regions of an image the same way. Isotropic diffusion, which is equivalent to convolving with a Gaussian, will smooth an entire image at the same rate throughout, regardless of object details or edges. Anisotropic diffusion will slowly diffuse the image but will treat regions in the image with many edges differently from homogeneous regions, similar to space adaptive filters \[17\]. To clarify, regions with heavy edges will be smoothed less, as with an all pass filter, while regions with few edges will be blurred akin to using a low pass filter, thus removing high frequency noise. However, as discussed earlier, noise obscures edge content, making it difficult to determine which regions should be smoothed less. We introduce a more robust version of edge-detection. The noise in regions with few edges will be removed using this low pass filter.

Our proposed approach extends speckle reducing anisotropic diffusion, or SRAD \[11\]. This method works on raw amplitude data. Our method, called Distance-Driven SRAD, or DD-SRAD, extends the SRAD equation to include the temporal coordinate \[18\]. Instead of smoothing each image in the temporal stack, or image time-series, separately using SRAD, DD-SRAD uses all of the images in the stack to estimate more accurately where edges are in the scene. DD-SRAD provides improvement over the original SRAD approach in the context of despeckling a stack of images.

1.3. 3D Noise Removal Methods

There are few methods of noise removal which operate on a stack of images and retain the integrity of each image in the stack. One method, developed by the CREATIS lab \[19\], is an anisotropic diffusion approach that smooths stacks of magnetic resonance images, using a regularization term to average noise in homogeneous regions. As this method is adapted from the original anisotropic diffusion PDE, it is only suitable for removing additive Gaussian noise, and not multiplicative speckle on raw amplitude images - the gradient magnitude used to drive anisotropic diffusion can mistakenly categorize speckle as edges.

There are few 3D speckle removal algorithms. One notable method was developed by Azzabou and Paragios in 2008 for use on ultrasound image sequences \[20\]. This is one of the first speckle removal algorithms derived for use on 3D image stacks. The algorithm uses patches to compute similarity between pixels in a neighborhood in order to preserve edge detail while smoothing, similar to non-local means \[21\]. One of the drawbacks of this method, as noted by the authors, are the assumptions made on noise independence. These assumptions make the estimation process simpler, but they may lead to inaccurate results as speckle may be spatially correlated. Also, neighborhoods used to calculate pixel similarity are not based on image structure, and may lead to over smoothing if the neighborhoods are too large. Similar 3D filters for SAR images can be found in \[22, 23\]. For comparison, DD-SRAD is a diffusion algorithm, so neighborhood size and shape are not a concern for our method.
One adaptive speckle filter called DespecKS [24] operates on SAR image stacks. DD-SRAD is derived from both SRAD and DespecKS. There are a number of drawbacks to both earlier methods, which we hope to address with DD-SRAD, as shown in Sections 2 and 3. Through discussing 3D noise removal methods, it can clearly be seen that there are no existing diffusion based methods for use on SAR temporal stacks.

1.4. Problem

The problem is smoothing a time-varying stack of SAR images using temporal information without losing the individuality of each image in the stack. DD-SRAD reduces noise, retains edges, and preserves temporal information with better performance than the state of the art smoothing algorithms, as evidenced by lower mean squared ratio and higher structural similarity index.

1.5. Goals

The goals of this paper are twofold. First, we want to show that smoothing using DD-SRAD leads to a lower mean squared ratio and higher structural similarity index than using other methods. Secondly, we will show the reader that DD-SRAD retains changes in the time-domain better than similar methods.

1.6. Overview

In this section, we have discussed the background of SAR imagery and speckle noise, and shown some of the failures of the state of the art noise removal algorithms. Section 2 will be an overview of Speckle Reducing Anisotropic Diffusion, a leading despeckle algorithm. Our methodology for denoising will be covered in Section 3 with an illustrative one-dimensional example shown in Section 4. Results from this and other methods will be displayed and assessed in Section 5. Section 6 will include discussion of Goals 1 and 2. We will conclude our work in Section 6.

2. Anisotropic Diffusion and SRAD

2.1. Anisotropic Diffusion

The anisotropic diffusion approach was first developed by Perona and Malik in 1987 [16]. The continuous form of the equation for this approach is outlined in [1].

\[
\begin{align*}
\frac{\partial I}{\partial t} &= \nabla \cdot \left[ c(||\nabla I||) \cdot \nabla I \right] \\
I(t = 0) &= I_0
\end{align*}
\]

(1)

In [1], \(\nabla I\) is the gradient of the image, \(||\) is the magnitude, \(\nabla \cdot\) is the divergence, \(I_0\) is the input image, and \(c(x)\) is a diffusion coefficient, defined as:

\[
c(x) = \frac{1}{1 + (x/k)^2}.
\]

(2)
The magnitude of the gradient is used as an edge detector within $c(\nabla I)$. As $|\nabla I| \gg k$, then $c(\nabla I) \to 0$, and filtering becomes all-pass. Similarly, as $|\nabla I| \ll k$, then $c(\nabla I) \to 1$, and filtering becomes Gaussian.

A discrete form of (1) is shown in (3).

$$I_{s}^{t+\Delta t} = I_{s}^{t} + \frac{\Delta t}{|\eta_{s}|} \sum_{p \in \eta_{s}} c(\nabla I_{s,p}^{t})\nabla I_{s,p}^{t}$$

$I_{s}^{t}$ is the image sampled discretely and $s$ is the pixel position in $(x,y)$. $\Delta t$ is the time step, and $|\eta_{s}|$ is the window around $s$. $|\eta_{s}|$ is simply the size of the window. $\nabla I_{s,p}^{t} = I_{p}^{t} - I_{s}^{t}$, $\forall p \in \eta_{s}$. $I_{s}^{t+\Delta t}$ is the image at the next time step.

Anisotropic diffusion is an effective noise removal method, particularly for additive noise types. Diffusion is an iterative noise removal process - often the input image is smoothed slowly over time, for upwards of 100 iterations. With the number of iterations inversely related to the value of $\Delta t$, the time step should be kept $\leq 0.25$ for stability [25]. For speckle noise, however, anisotropic diffusion can enhance the noise, because the gradient magnitude may treat the noise as an edge. We now move on to discuss filters specifically designed for speckle noise.

### 2.2. Speckle Reducing Anisotropic Diffusion

Speckle Reducing Anisotropic Diffusion, or SRAD, proposes a new partial differential equation (PDE) based on the original anisotropic diffusion and the adaptive speckle filters. The SRAD PDE is as follows:

$$\left\{ \begin{array}{l} \partial I(x,y;t)/\partial t = div[c(q)\nabla I(x,y;t)] \\ I(x,y;0) = I_{0}(x,y), (\partial I(x,y;t)/\partial \overrightarrow{n})|_{\partial \Omega} \end{array} \right. \tag{4}$$

where $\Omega$ is a 2D coordinate grid where the image is non-zero. $\partial \Omega$ is the border of $\Omega$, and $\overrightarrow{n}$ is the outer normal to said border.

The divergence of $c(\cdot)\nabla I$ is calculated as, where $h$ is a small spatial step size in $x$ and $y$:

$$d_{i,j}^{n} = \frac{1}{h^{2}}[c_{i+1,j}^{n}(I_{i+1,j}^{n} - I_{i,j}^{n}) + c_{i,j+1}^{n}(I_{i,j+1}^{n} - I_{i,j}^{n}) + c_{i,j}^{n}(I_{i,j+1}^{n} - I_{i,j}^{n}) + c_{i,j}^{n}(I_{i,j-1}^{n} - I_{i,j}^{n})]$$

which leads us to the SRAD update function:

$$I_{i,j}^{n+1} = I_{i,j}^{n} + \frac{\Delta t}{4} d_{i,j}^{n} \tag{5}$$

The diffusion coefficient is defined as:

$$c(q) = \frac{1}{1 + [q^{2}(x,y;t) - q_{0}^{2}(t)}/[q_{0}^{2}(t)(1 + q_{0}^{2}(t))]. \tag{7}$$

c$(q)$ is composed of two parts: the instantaneous coefficient of variation, $q(x,y;t)$, and the speckle scale function, $q_{0}(t)$. The instantaneous coefficient of variation,
or ICOV, can be thought of as the limit of the coefficient of variation as a window of interest shrinks to a single point.

The ICOV is defined as:

\[
q(x, y; t) = \sqrt{ \frac{(1/2)(|\nabla I|^2 - (1/16)(\nabla^2 I)^2)}{[1 + (1/4)(\nabla^2 I)]^2} } \tag{8}
\]

and it contains both a gradient magnitude term and a Laplacian term. The ICOV allows for edge detection in bright and dark regions because of the addition of the Laplacian term. The speckle scale function, taking the form of a generalized coefficient of variation \( \sigma_\mu \), is defined as:

\[
q_0(t) = \frac{\sqrt{\text{var}[z(t)]}}{z(t)} \tag{9}
\]

where \( z(t) \) is a homogeneous region, and \( \overline{z(t)} \) is the average over this region. This homogeneous region can be defined by the user manually, in an image region where there are few edges and noise variance is low. Often, if a user is unclear on which region to choose, they may choose to use the default setting for a homogeneous region. This default is a hard-coded set of coordinates, and so mimicks a random selection of a region from an image. This default homogeneous region may not even give a homogeneous region for every image that SRAD is used on, let alone the most homogeneous or the largest homogeneous region.

SRAD is an improvement on both the original anisotropic diffusion algorithm and on the adaptive speckle filters. At the midpoint of an edge, the Laplacian term in (8) undergoes a zero-crossing. This insures that diffusion only occurs along the contour of the edge, which is an improvement on adaptive filtering, where there is no smoothing along the edge contour. However, there is also an 'inverse' diffusion that occurs perpendicular to the edge. This is not present in simple anisotropic diffusion and is a phenomenon of SRAD. This 'inverse' diffusion leaves the dark regions of edges darker and the brighter regions brighter - essentially, SRAD enhances contrast along edges. The edges in the resulting smoothed images are more prevalent [11].

3. Distance-Driven SRAD

3.1. New Diffusion Coefficient

While the SRAD approach works well for 2D data, a novel method is desired for use with 3D data. For this reason, we propose a new coefficient of diffusion for use with 3D image stacks.

This diffusion coefficient depends on the distance between neighboring pixel time series, \( D \). Various distances can be used to calculate \( D \), which will be discussed in more detail in Section 4 and tested in Section 5. \( D \) replaces the gradient magnitude, \( |\nabla I| \), in the original SRAD coefficient of diffusion (8). For
instance, if there are two pixels on one section of a road, the intensity statistics of the two pixels should be similar through time. We expect the distance, or difference between the time series of these pixels to be minimal. However, the difference between the temporal behavior of a pixel in a field next to the road and a pixel on the road may be divergent over time. The road may be paved at different times of the year, leading to changes in intensity of the road pixel. A field pixel may change intensity based on vegetation growth, cutting of grass, etc. As the time series of these pixels behave differently through time, the distance between them will be larger than that of the two road pixels. This large distance will be equivalent to an edge: the edge delineated by the boundary of the road against the field. Just as gradient magnitude denotes edges, thus distance between neighboring pixel series can also mark edges. This edge detector is the basis for driving Distance Driven SRAD, or DD-SRAD.

The idea was inspired by a similar use of distance in [24], another noise removal algorithm that works on time-series data: the DespecKS algorithm.

The DespecKS algorithm developed in 2011 for smoothing of SAR amplitude image stacks using a space adaptive filter [24]. A test is performed to decide whether two pixels are statistically homogeneous pixels (SHP). The image stack can be represented as concatenated data vectors of intensity values. $d(P)$ is defined as:

$$d(P) = [d_1(P), d_2(P), ..., d_N(P)]^T$$

and is a vector of intensity values for arbitrary pixel $P$. $d_1(P)$ is the pixel value in the first image, etc. $N$ is the number of images in the stack. To test whether two data vectors ($d(P_1)$ and $d(P_2)$, for example) are from the same distribution, the two-sample Kolmogorov-Smirnov (KS) test can be used [26]. This tests the null hypothesis that pixels $P_1$ and $P_2$ are from the same distribution. If this null hypothesis cannot be rejected at a particular $\alpha$ significance level, then the pixels $P_1$ and $P_2$ will be considered statistically homogeneous.

To perform the KS test on pixel vectors, the empirical cumulative distribution functions (ECDF) of the pixel time series are built. If we define $x = |d|$, then an unbiased estimator for the CDF of a pixel, $S_N(X)$, can be written as:

$$S_N(X) = \begin{cases} 0, & \text{if } X < x_1 \\ \frac{k}{N}, & \text{if } x_k \leq X < x_{k+1} \\ 1, & \text{if } X \geq x_N. \end{cases}$$

$x_i$ is the $i^{th}$ element in the list of pixel intensities.

The two sample KS test measures the maximum absolute difference between the cumulative distribution functions $S_{N_1}^P$ and $S_{N_2}^P$. This difference, which is used in DD-SRAD-KS as a distance metric, is defined by

$$D_N = \sup_{x \in \mathbb{R}} |S_{N_1}^P(x) - S_{N_2}^P(x)|.$$
statistically homogeneous with another pixel $P$, it may be more likely that pixel $P_2$ is drawn from the same distribution as $P$ than the probability that $P_1$ is. For this reason, it may be dangerous to average all SHP together, disregarding the distance $D_N$ from $P$. For this reason, DD-SRAD uses the KS and other distances, instead of the KS test combined with averaging.

Another drawback is that the KS test has poor sensitivity in the tails of the probability distribution in situations where two pixels are statistically inhomogeneous [24]. Using the KS distance will solve this problem, and we also test other distances to determine which ones have better performance. There are a number of distances that one can use with DD-SRAD - we have implemented and tested three different distance metrics, to show improved performance over the state of the art algorithms, such as DespecKS.

To incorporate the KS distance into a new diffusion coefficient, the KS distance in (12) between a pixel and its four neighbors (for non-edge pixels) is calculated. This results in four distances: east, west, north, and south. These distances are summed to form an overall measure of how different the center pixel is from its neighbors. The summed distance measure, $\sum D$, is used in place of the image gradient in [8]:

$$q(x, y; t) = \sqrt{\frac{(1/2)(\sum D/I)}{[1 + (1/4)(\sum D^2/I)]^2} - \frac{(1/16)(\sum D^2/I)^2}{[1 + (1/4)(\sum D^2/I)]^2}}. \quad (13)$$

### 3.2. Homogeneous Region Detection

Homogeneous region detection automatically locates regions of lower variance in the noisy images improving on random region selection used by SRAD. In Figure [1] a detected homogeneous region is displayed on the Canny edgemap of the temporal median of the sub-stack [27]. Since there are few edge pixels within the boundary, the detected region is homogeneous.

To test our method for detecting homogeneous regions, 10 stacks of sub-images were generated from the large SAR data stacks. Each of the sub-image stacks have 67 temporal slices, and are approximately 300 pixels by 500 pixels. The sub-images were randomly sampled from the original data, ensuring the underlying terrain in each sub-image stack varies significantly from stack to stack.

To detect a homogeneous region for each sub-image stack, the method was applied to the stack to pick one region. The method can be broken into the following steps:

1. Take the temporal median of the stack. Use Canny edge detection to find the edgemap of the median image [27].
2. Filter to remove spurious, noise-derived edges. A window size of 3x3 was used for filtering for all sub-image stacks. If only one pixel in the window was marked as an edge pixel, then the pixel was marked as background.
3. Find the largest connected component in the inverted image, to determine the largest edge-free region in the median image. This is the largest homogeneous region.

This same homogeneous region was used for all temporal slices in the sub-stack, and its variance was calculated for each slice, for use in (9).

Figure 2 shows a flow-chart of the DD-SRAD algorithm, combining homogeneous region detection with the new diffusion coefficient.

4. 1-D Example of DD-SRAD Method

In this section an example of the DD-SRAD algorithm will be shown in 1-D for clarity.

4.1. Introduction of 1D Signal

Let us define a 1-D input signal using a rectangular pulse:

\[
y = \begin{cases} 
0 & \text{if } x < 1 \\
h & \text{if } 1 < x < 2 \\
0 & \text{if } x > 2 
\end{cases}
\]  

(14)

where \( h \) is the height of the signal. Consider a simple case where the height \( h \) of the signal increases as a multiple of a constant \( k \) over time. This example is illustrated in Figure 3.

If \( t \in \{1, N\} \), then the last pulse in this time-series will have height \( Nk \).
Homogeneous region detection is first performed on an input image stack, to calculate the speckle scale function. Each image in the stack is subsequently smoothed for $I$ iterations to produce the output image stack shown.

Figure 3: Pulse height increasing by factor of $k$

4.2. Using DD-SRAD-RSS

Now this 1-D signal will be used as input for a 1-D version of DD-SRAD. This case is for DD-SRAD-RSS, an unweighted approach using root sum-of-squares distance.

First, derive an expression for the RSS distance between instances in the temporal stack. For each instance, there will be one $N$-length vector of differences between that instance and all the other signals in the temporal stack. Thus, there will be $N$ of these difference vectors for the whole stack. If $t = s$ signifies the current time-instance in the stack, then subtract the signal at time $s$ from signals at time $t$, where $t$ ranges from 1 to $N$. Examples of the difference vectors are shown in Figure 4.

Note the similarity of the difference vectors to the original signal. The difference vectors are simply shifted and flipped versions of the original signal, $y(x)$.

To calculate the RSS distance for each time instance, square each distance and sum these squared distances. Then take the square root of the remaining value. In Figure 4, the height of the step ranges from $1 - s$ to $N - s$. Thus, the
Figure 4: Vector of Differences for slices in the time-series. For slice 2 ($s = 2$), there is shown the result of slice 2 from the other slices in the stack, for all $t$. This can be done for all $N$ slices.

distance metric, $d$, can be written as

$$d = \sqrt{\frac{1}{6} k^2 N^2 + N(3 - 6s) + 6s^2 - 6s + 1}. \quad (16)$$

To calculate the instantaneous coefficient of variation, substitute $d$ into (13) for $D$, and the original signal $y$ for $I$. As this is a one-dimensional case, there is only one distance used, instead of multiple directional distances, so there is no need to sum over $d$. After substituting for $D$, it can be observed that

$$q(x; t) = \sqrt{\frac{1/2}{[1 + (1/4)(d^2/y)]^2} - \frac{1/16}{d^2/y}} \quad (17)$$

which simplifies to

$$q = \frac{\sqrt{1/8} - d^2}{y + d^2/4}. \quad (18)$$

Using this $q$ and setting the value of $a_0$ to .15 (a value chosen from testing), the diffusion coefficient (7) can be written as:

$$c = \frac{1}{1 + 5.8 (q - .15)}. \quad (19)$$

The last piece needed in order to use (6) is $D$, or the coefficient of diffusion times the divergence. Since this is a one-dimensional case, the divergence is
calculated simply as the derivative in the $x$-direction. $y$ can be rewritten as a sum of heaviside step functions [23]:

$$y = (ks) \{ u[x - 1] - u[x - 2] \}.$$  (20)

The derivative of $y$ is then

$$y' = (ks) \{ \delta[x - 1] - \delta[x - 2] \}$$  (21)

where $u[x]$ is the unit step function and $\delta[x]$ is the Dirac delta function.

The update function is

$$y = y + \frac{D \Delta t}{4}.$$  (22)

Without loss of generality, set $\Delta t$ to .05. If all variables are substituted into (22), the result is:

$$y(s) = y(s) + \frac{cy'}{8}.$$  (23)

$y(s)$ is the value of the original time series at time $s$.

5. **Experiments and Results**

5.1. **Statistical Testing for Homogeneous Region Detection**

The goal of this testing is to show that homogeneous region detection chooses regions of lower variance than when using the random region detection provided with SRAD. If a high variance region is used to calculate the speckle scale statistic, the SRAD or DD-SRAD smoothing procedure will be corrupted.

The variance of our detected homogeneous region is compared to that of a randomly selected region. Comparisons are made to a randomly selected region because the original SRAD implementation uses an ad-hoc set of coordinates for choosing a homogeneous region - these region coordinates/dimensions do not change based on the image being smoothed, so the selection is close to random. To compare to randomly selected regions, we chose regions with approximately the same size as the detected homogeneous region for each sub-image stack. The randomly selected region is a square region, selected from within one image of the sub-stack. This same random region is used to crop all of the slices of the sub-stack. The variances of all of the random regions of all of the 10 sub-stacks are then calculated, 67 variances per sub-stack (corresponding to 67 slices), for both detected and randomly selected regions.

The variances of regions detected by homogeneous region detection are lower than those of the regions that are randomly selected. In Table 1 some general statistics of the two sample sets are shown. The maximum variance values are extremely different, over 10 times greater for the randomly selected region. Also, the standard deviation of variance for Random regions is much higher than that of Detected Regions - showing that the variance of Random regions has wide spread.
Table 1: Statistics for Variances of Randomly Selected vs. Detected Regions

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Random</th>
<th>Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.1015</td>
<td>0.0545</td>
</tr>
<tr>
<td>Max</td>
<td>5.1184</td>
<td>0.2603</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.7458</td>
<td>0.0333</td>
</tr>
</tbody>
</table>

Both distributions of variances were tested for normality using the Jarque-Bera test [29]. The variances for regions were not found to be normally distributed, so a simple t-test cannot be used. Instead, a two-sample Kolmogorov-Smirnov (KS) test was used to compare the two sets of variances. The two distributions of variances were tested at the 1% significance level and the null hypothesis rejected: the two variance sample sets are not from the same distribution. Thus, the variances of detected regions and randomly selected regions are different.

Moreover, the alternative hypothesis that the empirical continuous distribution function (ECDF) of the randomly selected region variances is smaller than that of the detected homogeneous region variances can be tested. In general, if the ECDF of one sample set is smaller than another ECDF, then the data values of the sample set with the smaller ECDF are larger. We reject the null hypothesis that the ECDF’s are equal in favor of the alternative hypothesis at the 1% significance level [30]. Therefore, the ECDF of random region variances is smaller than the ECDF of detected region variances, and variances of the random regions are larger than the variances of detected homogeneous regions.

5.2. Synthetic Datasets

Synthetic datasets were developed for testing. Many of the metrics used to evaluate methods, such as mean squared error (MSE), rely on comparing the smoothed image to the original ground truth image (with no noise.) As ground truth images for these satellite data are nonexistent, simulated data with similar features is used and noise added to this synthetic data. Then smooth results from our methods can be compared to the original, non-noisy images.

Noise is added to the synthetic images by generating speckle noise from a Rayleigh distribution, because our SAR dataset is observed to follow this distribution (see Figure 5) [31, 32]. Two Gaussian random variables are generated and combined to form a Rayleigh random variable. Then the base synthetic images are multiplied with this Rayleigh variable to create multiplicative speckle noise. Using Immerkaer’s fast noise estimation method on the SAR images, the estimated noise variance is between .08 and .09 for each image, so this noise variance range is used for the synthetic images as well [33].

Twenty synthetic, noisy datasets are generated, each with 11 to 17 images, resulting in 286 total images tested. The dataset size is varied because not all SAR temporal stacks will have the same number of images. To simulate
variation over time, features are introduced at different points during the time-series, and the intensity of these objects is changed over time. In the synthetic data, straight lines represent roads, and circular or ovular objects represent fields. Rectangular objects are buildings. Figure 5 shows a sample progression of two synthetic images, where the intensity of features varies over time. In particular, notice the disappearance of the thin diagonal line in (b) and the appearance of the large oval as the sequence progresses. This particular sequence shows sample transitions before noise is added.

![Figure 5: Rayleigh distributed SAR images.](a) Rayleigh distribution. (b) Histogram of our SAR dataset.](a) Image 1/17 (b) Image 9/17

Figure 6: Time varying sequence of synthetic ground truth images.

5.2.1. Methods Tested and Parameters Used

Lee Filter. For the Lee Filter, the default window size of 3 was used. This window size refers to the size of the window used for averaging. The other parameter is the coefficient of variation in a homogeneous region. As the original Lee filter does not include automatic homogeneous region detection, we select a random region (40 pixels by 40 pixels) as a homogeneous region and calculate statistics of this area.

SRAD. The SRAD algorithm was run on all datasets for 200 iterations, with a small smoothing time-step ($\Delta t$) of .05, resulting in a minimal amount of smoothing with each iteration. Using too large of a time-step in combination with fewer iterations is not preferred, as this can negatively affect the stability of diffusion algorithms. The homogeneous region chosen is the same randomly selected 40x40 region used in with the Lee Filter.
DespecKS. We implemented and tested a version of the DespecKS algorithm. The suggested 15x21 window is used to test for SHP. The alpha significance value is set to .05.

Med-SRAD. This is SRAD driven by the edges of the temporal median of the stack. Median Driven SRAD is also run for 200 iterations, with time step-size .05. The same homogeneous region is used as in SRAD and Lee filtering.

DD-SRAD. DD-SRAD is SRAD driven by distance between pixel time-series. All DD-SRAD algorithms are run for 200 iterations. The time-step used for DD-SRAD is also .05. For all DD-SRAD algorithms, homogeneous region detection is used to specify a homogeneous region, using window-size 3x3.

DD-SRAD-KS stands for Distance-Driven SRAD run using KS distance, which was described in Section 3.

DD-SRAD-B stands for Distance-Driven SRAD using Bhattacharyya distance, given by

\[
D_B(p, q) = -\ln \left( \sum_{x \in X} \sqrt{p(x)q(x)} \right) \tag{24}
\]

where \( p \) and \( q \) are probability distributions for different pixels, derived from the time series data (shown in \[3\], \[4\]). Four directional distances are computed using (24) and summed. The resulting \( \sum D \) is substituted in (13).

DD-SRAD-RSS uses average RSS distance, weighted evenly across all images regardless of position in the stack. RSS distance stands for root of sum of squares. To give an example, call one pixel time series \( u \) and a neighboring pixel time series \( v \), each of length \( N \). The average RSS distance between these pixels is

\[
D_{RSS}(u, v) = \sqrt{\frac{\sum (u - v)^2}{N}}. \tag{25}
\]

Again sum over all directional distances for use in (13).

Finally, DD-SRAD-RSS-W refers to DD-SRAD using RSS distance, but with a weighted approach. Data from slices closer to the slice of interest are weighted more heavily. The distance is calculated as

\[
D_{RSS-W}(u, v) = \sqrt{\frac{\sum (u - v)^2w(i)}{\sum_{i=1}^{N} w(i)}}. \tag{26}
\]

where \( w \) is a Gaussian weighting function centered at the current slice (image) being smoothed. The key result from using a weighted approach is that each slice in the stack will have a unique diffusion coefficient.

5.2.2. Goals

Goal 1 - Noise Removal. The first goal is to show that DD-SRAD has a lower average MSR and higher SSIM than other leading methods. MSR is mean squared ratio - it is calculated by taking the ratio of the original noise-free image, \( J \) and the speckle reduced image, \( j \). Each image has dimensions \( M \times N \).
Table 2: Average MSR and SSIM

<table>
<thead>
<tr>
<th>Method</th>
<th>MSR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee</td>
<td>39.1</td>
<td>.075</td>
</tr>
<tr>
<td>SRAD</td>
<td>16.9</td>
<td>.180</td>
</tr>
<tr>
<td>DespecKS</td>
<td>49.2</td>
<td>.069</td>
</tr>
<tr>
<td><strong>Med-SRAD</strong></td>
<td>5.87</td>
<td>.111</td>
</tr>
<tr>
<td>DD-SRAD-KS</td>
<td>6.41</td>
<td>.219</td>
</tr>
<tr>
<td>DD-SRAD-B</td>
<td>29.8</td>
<td>.080</td>
</tr>
<tr>
<td>DD-SRAD-RSS</td>
<td>8.33</td>
<td>.180</td>
</tr>
<tr>
<td>DD-SRAD-RSS-W</td>
<td>5.49</td>
<td>.236</td>
</tr>
</tbody>
</table>

The ratio is squared and averaged across the images to get an MSR for each stack, as shown in (27). $\epsilon$ is a small real number ($1 \times 10^{-6}$) added to the diffused $j$ to prevent division by zero values. If the denoised result perfectly matches the ground truth image, MSR = 1. As MSR moves away from unity, speckle filtering performance degrades. All of the MSR reported here are positive, because image intensities are greater than zero, so the methods resulting in lower MSR, or closer to 1, perform the best. The average MSR reported in Table 2 is averaged across the twenty synthetic datasets.

$$MSR = \frac{1}{MN} \sum_{k=1}^{M} \sum_{l=1}^{N} \left[ \frac{J(k,l)}{j(k,l) + \epsilon} \right]^2$$  \hspace{1cm} (27)

SSIM stands for the structural similarity index, defined by Wang et. al. A higher value of SSIM is more desirable. In Table 2, we compare the MSR and average SSIM for all methods tested.

In Figure 7, a graphical representation of the MSR results are shown.

![Figure 7: Mean squared ratio for each smoothing method.](image-url)
Table 3: Mean and Standard Deviation for Homogeneous Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee</td>
<td>.160</td>
<td>.079</td>
</tr>
<tr>
<td>SRAD</td>
<td>.183</td>
<td>.045</td>
</tr>
<tr>
<td>DespecKS</td>
<td>.157</td>
<td>.089</td>
</tr>
<tr>
<td>Med-SRAD</td>
<td>.377</td>
<td>.062</td>
</tr>
<tr>
<td>DD-SRAD-KS</td>
<td>.250</td>
<td>.058</td>
</tr>
<tr>
<td>DD-SRAD-B</td>
<td>.176</td>
<td>.078</td>
</tr>
<tr>
<td>DD-SRAD-RSS</td>
<td>.236</td>
<td>.060</td>
</tr>
<tr>
<td>DD-SRAD-RSS-W</td>
<td>.261</td>
<td>.060</td>
</tr>
</tbody>
</table>

Within DD-SRAD, DD-SRAD-B has the highest MSR and lowest SSIM. DD-SRAD-KS and DD-SRAD-RSS-W perform with similar MSR and SSIM values, with DD-SRAD-RSS-W performing the best out of all DD-SRAD approaches. DD-SRAD has lower MSR and higher SSIM than both the Lee filter and the SRAD approach, which are representative of speckle reduction techniques used by the SAR community. DD-SRAD also has lower MSR and higher SSIM than the DespecKS method, which is representative of speckle removal on temporal stacks.

Med-SRAD also performs well in terms of MSR and SSIM, but has lower performance when we analyze the standard deviation of pixel intensity in homogeneous regions. Standard deviation and mean values of homogeneous regions (found using homogeneous region detection) for the experiments have been measured, shown in Table 3. These are averaged over the stack. A lower standard deviation is preferred for homogeneous regions, because in the original, speckle-free image, a homogeneous region should have little variation, so most variation is due to noise. After smoothing, if a region has little variation, then the noise has been removed.

In Figure 8, the standard deviations are shown in graphical format.

For both experiments, the standard deviation of homogeneous regions is lower for DD-SRAD-KS, DD-SRAD-RSS, and DD-SRAD-RSS-W than for Med-SRAD. DD-SRAD-B does not have as low of a standard deviation as its counterparts. SRAD has the lowest standard deviation for homogeneous regions compared to the other methods. However, when observing images after smoothing, the flaws of SRAD become clear. In Figure 9, the SRAD algorithm has left many high intensity speckle artifacts behind. However, Figures 9c, 9f, and 9i have smoothed these bright noisy pixels.

Also, the mean intensity within the homogeneous region is much lower for the Lee, SRAD, and DespecKS algorithms than for most of the DD-SRAD algorithms. This highlights better preservation of intensity post-smoothing, as the original mean intensity of the homogeneous regions is .479. All of the smoothing algorithms lower this original mean intensity, however DD-SRAD algorithms lower this intensity to a smaller extent.

The distance metrics analyze pixel behavior through time, unlike the SRAD...
Figure 8: Standard deviation in homogeneous regions for each smoothing method. DD-SRAD smoothes homogeneous regions more effectively (i.e., lower variance) than the Lee and DespecKS filters. Although the SRAD approach promotes smoother homogeneous regions, there are other pitfalls to this approach: namely, bright speckle artifacts and degradation of overall image intensity.

Table 4: MSR of Time-Varying Structure

<table>
<thead>
<tr>
<th>Method</th>
<th>Local MSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DespecKS</td>
<td>43.496</td>
</tr>
<tr>
<td>DD-SRAD-KS</td>
<td>5.8889</td>
</tr>
<tr>
<td>DD-SRAD-B</td>
<td>26.356</td>
</tr>
<tr>
<td>DD-SRAD-RSS</td>
<td>7.5556</td>
</tr>
<tr>
<td>DD-SRAD-RSS-W</td>
<td>5.0617</td>
</tr>
</tbody>
</table>

ICOV, which only uses spatial analysis. Using temporal analysis, if bright pixels follow the same behavior as neighboring pixels, then bright speckle noise is smoothed. The SRAD algorithm cannot perform this type of temporal analysis, so bright speckle remains in the SRAD despeckled images.

Goal 2 - Temporal Change Preservation. To measure whether our algorithms preserve time varying information, DD-SRAD is compared with the DespecKS algorithm, the only other algorithm that also takes advantage of the image stack. For each synthetic dataset there is only one feature that appears or disappears over time. A rectangular bounding box is formed around this feature and the MSR then calculated for this region between smoothed images and the ground truth image. These MSRs are reported in Table 4.

All DD-SRAD methods have lower MSR than the DespecKS algorithm, showing that DD-SRAD preserves temporal information better than the DespecKS algorithm. Using a weighted RSS distance produces lower error than
Figure 9: Comparing results from Experiment 2. The resulting images when using DD-SRAD are smoother and have less harsh texture due to speckle, compared to the Lee and DespecKS filters. While the SRAD approach does provide image smoothness, high intensity speckle is not removed and results in a pattern of white dots scattered through the image. This drawback can also be observed when testing on actual SAR data, as in Figure [1].
Table 5: Mean Runtimes for Algorithms (seconds)

<table>
<thead>
<tr>
<th>Method</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee</td>
<td>16.4529</td>
</tr>
<tr>
<td>SRAD</td>
<td>11.7960</td>
</tr>
<tr>
<td>DespecKS</td>
<td>244.232</td>
</tr>
<tr>
<td>Med-SRAD</td>
<td>5.71890</td>
</tr>
<tr>
<td>DD-SRAD-KS</td>
<td>1461.80</td>
</tr>
<tr>
<td>DD-SRAD-B</td>
<td>1348.20</td>
</tr>
<tr>
<td>DD-SRAD-RSS</td>
<td>98.1444</td>
</tr>
<tr>
<td>DD-SRAD-RSS-W</td>
<td>49.3008</td>
</tr>
</tbody>
</table>

using a non-weighted RSS distance. However, at times, other distances, such as the KS distance in [1] outperforms the weighted RSS distance. If runtime of algorithms is a concern, then using weighted RSS distance is faster than using either KS distance or Bhattacharyya distance, and results are comparable, or better, in terms of MSR. This can be seen in [5]. DD-SRAD-RSS and DD-SRAD-RSS-W have runtimes that are over fifteen times smaller than when using other distances.

The difference in smoothing performance between distances may be a result of using unscaled distances. The Bhattacharyya distance values are much larger than the distances calculated using the two-sample KS test, or when using RSS distance. As a result, smoothing is inhibited to a greater extent when using Bhattacharyya distance.

Change preservation is also shown on one of the time-varying datasets in Figure 10. Images 1, 8, and 9 are displayed from the stack. DD-SRAD-RSS-W is able to preserve the intensity changes of the big square. Taking the mean of the stack just provides only one image, with no information about the intensity changes of objects in the image. For prediction and analysis in many applications, it is necessary to retain information about when features are changing.

5.3. Real SAR Datasets

The amplitude images in the real SAR dataset are acquired by Tele-Rilevamento Europa over Staunton, Virginia. These images are taken over the period of one year, about once a week - the interval between images is not always the same. There are 67 images in the dataset. As these images are large (over 600 MB each), we performed smoothing on 560 KB subsections of the images, or 509x332 pixels. Performance is shown against three additional smoothing methods: SAR-BM3D, NL-Means (non-local means), and TV-based (total variation based) [36, 37, 38]. Parameter values for the three new methods are explained below.

SAR-BM3D. For this method, the only user defined input parameter is the number of looks of the speckle noise. For these experiments, the default value of 1 was used, as the SAR dataset has not been averaged over multiple looks.
Figure 10: Showing change preservation - displaying the log of the image
There are at least fifteen other parameters embedded in the software - the default values for these parameters have been used, as these values were chosen by the authors after rigorous testing.

**NL-Means.** For this method, there are three input parameters. The search window size used is 5x5 pixels, and the similarity window size is 2x2 pixels. Both of these window sizes are the default window sizes chosen by the authors of the software used. Sigma, or the degree of filtering, is set to .09, based on the estimated noise variance.

**TV-based.** The TV-based method includes two input parameters: \(\lambda\), and the number of iterations. \(\lambda\) is a regularization parameter that controls the amount of denoising - a smaller \(\lambda\) leads to more aggressive smoothing. \(\lambda\) is set to 1. The number of iterations is set to 100. Both of these values are default values, and also produce the best result on the SAR dataset.

Smoothing results are shown on a SAR stack of 67 images in Figure 11. A reduction in noise is expected, manifested by removal of spurious edges after smoothing. Results from using other state-of-the-art methods are shown, such as SAR-BM3D in 11b and 15d [36]. While this method removes much of the noise, the edge details are completely lost for many important features, such as the lake and the secondary road. Similar loss of detail can be seen in 11d using a total variation based denoising method [38]. Another method, non-local means (NL) does not remove enough speckle [37]. In Figure 11i a result using DD-SRAD is shown - this image is the smoothest of the nine shown while also maintaining clear edges.

Zooming into a few of the results can highlight details in the quality. In Figure 12 the DD-SRAD result shows the secondary road more clearly. Also, bright specularities have been suppressed.

Some artifacts can also be shown when zooming into other regions of the results (Figure 13). Smoothing using SAR-BM3D and NL-Means does not remove all bright speckle noise and also leaves behind some texture artifacts.

The result when using a total variation based denoising method is dark, as shown in Figure 11d. This method reduces the pixel intensities from over 6 in the original image to under .2 - since all the pixel intensities are close to zero, the result is lacking contrast. This is due to the nature of the smoothing algorithm - total variation denoising produces a result that has lower total variation, or a lower integral of the absolute gradient of the image. If the noisy image has more pixels with lower intensities than pixels with higher intensities, then in order to reduce the total variation, the overall intensity of the image is lowered. To more clearly show the result when using this method, the result in Figure 11d is scaled from 0 to 1 and reproduced in Figure 14. This method has smoothed over edges and produced pixelated regions that result in an uneven texture.

The edgemaps of the smoothed SAR results are shown in Figure 15. There are more edges in Figure 15d than in Figure 15f, but this may be beneficial as a secondary road is preserved in Figure 11i that is disconnected in Figure 11f.
Figure 11: Comparing Images from Smoothed SAR Stacks

Figure 12: Secondary Road in Smoothed SAR Images
The SRAD algorithm has thus removed a true edge along with noise produced edges. This true edge is preserved by DD-SRAD.
Figure 15: Smoothed SAR Stack Edgemaps
6. Conclusions

SAR images can be used for many different applications, from farm and ecological analysis, to transportation infrastructure monitoring. The image database contains images taken over Virginia, acquired for the purpose of remotely sensing problems with Virginia roads, highways, and bridges. Being able to accurately sense changes in infrastructure topography can lead to better predictions of events, and can support proactive behavior and preventing disastrous consequences of said events (landslides, sinkholes, bridge collapse, etc.) However, SAR images inherently contain speckle noise, which degrades image structure and makes accurate analysis of amplitude data difficult. To ensure accuracy of predictions, so that false alarms are not raised, SAR images must be denoised before performing any type of analysis with them.

For denoising SAR images, we have developed a method that performs better than leading methods in speckled image denoising. This novel method, DD-SRAD, performs better than many single image denoising methods. DD-SRAD also has lower MSR and higher SSIM than the only other popular denoising method for SAR image stacks (that utilizes temporal information.) DD-SRAD not only removes more noise than this method, DespecKS, but also preserves more temporal uniqueness than DespecKS.

To continue this work, scaled distances can be implemented, as well as testing different types of distances besides the three tested here (RSS, KS, and Bhattacharyya.) Also, weighted distances can be implemented for KS and Bhattacharyya distances. More work may also be done to implement this methodology on ultrasound imagery.
References


[38] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, T. Pock, An Introduction to Total Variation for Image Analysis, De Gruyter, 2010.