Despeckling of Synthetic Aperture Radar Images using Monte Carlo Texture Likelihood Sampling

Jeffrey Glaister, Alexander Wong, David A. Clausi

***Abstract*—Speckle noise is found in synthetic aperture radar (SAR) images and can affect visualization and analysis. A novel stochastic texture-based algorithm is proposed to suppress speckle noise while preserving the underlying structural and texture detail. Based on a sorted local texture model and a Fisher-Tippett logarithmic-space speckle distribution model, a Monte Carlo texture likelihood sampling strategy is proposed to estimate the true signal. The algorithm is compared to six other classic and state-of-the-art despeckling techniques. The comparison is performed both on synthetic noisy images added and on actual SAR images. Using peak signal-to-noise ratio, contrast-to-noise ratio and structural similarity index as image quality metrics, the proposed algorithm shows strong despeckling performance when compared to existing despeckling algorithms.**

***Keywords*-speckle noise, synthetic aperture radar, noise reduction, Fisher-Tippett noise**

I. Introduction

Synthetic aperture radar (SAR) is an active remote sensing system that generates and transmits microwave electromagnetic (EM) radiation to the surface of a target region [1]. Different amounts of the EM signal are absorbed and reflected by the target. The reflected signal is detected by the SAR satellite and used to construct images of large areas. Unlike real aperture radar systems, SAR systems take advantage of the movement of the satellite with respect to the target, so the image can have fine resolution with a small antenna [1]. Another advantage with SAR systems is that they do not rely on energy from the sun to image a target because the satellite generates the EM signal itself. Also, the microwaves are not significantly affected by cloud cover or fog [1]. However, due to constructive and destructive EM wave interference during image acquisition, SAR images are heavily corrupted by speckle noise [2]. Speckle noise is modelled as a multiplicative noise and gives the images a grainy appearance. It can affect automated image processing algorithms, such as region classification or segmentation. Therefore, there is a need to correct for speckle noise as a pre-processing step.

Many techniques exist to reduce speckle noise in SAR images. One technique is to account for noise at source by utilizing the capabilities of the SAR system itself. Multiple images of the same scene are obtained and averaged together to form a single image [3]. This is called a multiple-look image and this process reduces the speckle noise. However, it requires additional processing at source. Due to properties of SAR spatial resolution, it will also result in coarser resolution [4].

Other noise reduction techniques process the SAR image after it has been obtained. Classic filters include the averaging and Gaussian filter [5]. These filters use nearby pixels to estimate a pixel's true value. However, these filters do not adapt to scene statistics, so they are unable to preserve details such as edges causing blurry images. Classic adaptive spatial filters have been used for many years for SAR speckle noise reduction and try to preserve details. These include the Lee filter [6], Frost filter [7], and Gamma MAP filter [8]. These filters use local pixel intensity statistics to adjust the amount of smoothing and noise removal in certain areas [9]. In areas with large variance of pixel intensities, such as areas with high levels of detail or edges, the filter applies less smoothing in order to preserve those details. In homogeneous areas with little detail, a smoothing kernel is applied to remove the noise.

More recent speckle reduction approaches include anisotropic diffusion methods and wavelet domain despeckling. Anisotropic diffusion methods smooth images to reduce noise, but the diffusion process avoids affecting edges and structures [10][11]. Unlike adaptive filters, anisotropic diffusion methods still smooth areas with high levels of detail to remove noise, but do so in a way that improves retention of the underlying structures in the image. Wavelet domain despeckling techniques apply a discrete wavelet transform, which separates the image into high and low frequency components [4][12][13]. Since noise is assumed to be contained in the high frequency components, either a thresholding technique or a filter is applied to those components.

Another approach to speckle noise reduction is to transform the SAR image into logarithmic space, where the speckle noise is additive and can reduced using state-of-the-art despeckling methods such as BM3D [14]. While such state-of-the-art despeckling techniques have been shown to efficiently remove noise without blurring details, they also assume that the noise follows a Gaussian distribution, which is invalid in the case of SAR speckle noise in logarithmic space [15]. As such, these methods do not take into account the underlying speckle statistics in SAR images.

The denoised SAR images are used as inputs for a classification or segmentation algorithm. Texture characteristics are useful features to characterize and distinguish different regions [16]. Texture extraction methods include co-occurrence probabilities [17] or Gabor filters [18]. Despeckling algorithms should minimize any degradation caused to the underlying textural structures in the image. The classic Gaussian and

Jeffrey Glaister, Alexander Wong and David A. Clausi are part of the Vision and Image Processing Lab with the Department of Systems Design Engineering, University of Waterloo, Ontario, Canada, N2L 3G1. {jlglaist,a28wong,dclausi}@uwaterloo.ca

averaging filters result in blurring of details and the other aforementioned despeckling algorithms do not explicitly try to preserve textural characteristics. Existing texture-preserving speckle reduction algorithms [19][20] do not take into account the actual noise distribution. This motivates the incorporation of a texture model to determine texture likelihood and preserve the textural details while despeckling.

The main contribution of this paper is a novel despeckling method based on the concept of Monte Carlo texture likelihood sampling, that we refer to as the MCTLS algorithm. A rotation-invariant sorted local texture model is employed to characterize local texture properties within the SAR image. By combining this sorted local texture model with a Fisher-Tippett logarithmic-space speckle distribution model, a Monte Carlo texture likelihood sampling strategy is introduced to estimate the true signal in a stochastic manner.

The rest of the paper is organized as follows. In Section II, the speckle noise model is introduced and the methodology behind the algorithm is presented. In Section III, the testing methodology and image quality metrics are introduced. Results from qualitative and quantitative comparisons of the MCTLS algorithm to existing techniques are presented in Section IV. Finally, in Section V, conclusions are given.

II. Methodology

In this section, before describing the MCTLS algorithm, assumptions and models used in this paper are first defined. The speckle noise statistics assumed for the MCTLS algorithm and the sorted local texture model for characterizing textural characteristics are defined. Then, the problem formulation behind the MCTLS algorithm is presented. Finally, an explanation of the Monte Carlo texture likelihood sampling approach to solve the formulated problem is given, which incorporates both the speckle noise statistics and the sorted local texture model. An summary of the sequence of steps in the MCTLS algorithm is provided.

A. Speckle Noise Statistics

In acquired SAR images, speckle noise is defined to have a multiplicative relationship. For completion, the noise model also includes an additive component. This is shown in Equation 1, where $M(x,y)$ is the measured signal at pixel location (x,y), $S(x,y)$ is the actual signal, $N_m(x,y)$ is the speckle noise and $N_a(x,y)$ is additive noise [13].

$$M(x,y) = S(x,y) \cdot N_m(x,y) + N_a(x,y) \tag{1}$$

Normally, $N_m(x,y)$ is assumed to take on the Rayleigh distribution [21]. Furthermore, $N_a(x,y)$ component is less significant and is ignored [13]. In log-compressed SAR images, this multiplicative relationship becomes an additive relationship, as seen in Equation 2 where $m(x,y)$ and $s(x,y)$ are the log-compressed actual and measured signals. The multiplicative noise $N_m(x,y)$ is transformed into an additive component $\eta(x,y)$.

$$m(x,y) = s(x,y) + \eta(x,y) \tag{2}$$

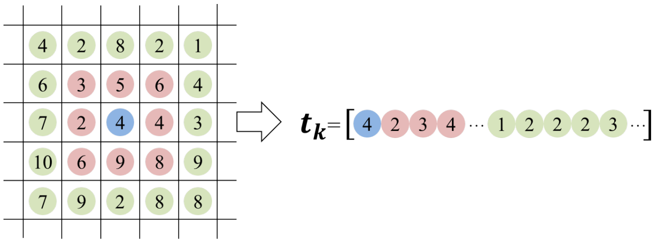


Fig. 2. Illustration of the rotationally-invariant sorted local texture model, modified from [23]. The numbers represent sample pixel intensities. Pixels are grouped by distance from the center pixel m_k and each group of pixels is sorted to get vector $\boldsymbol{t_k}$.

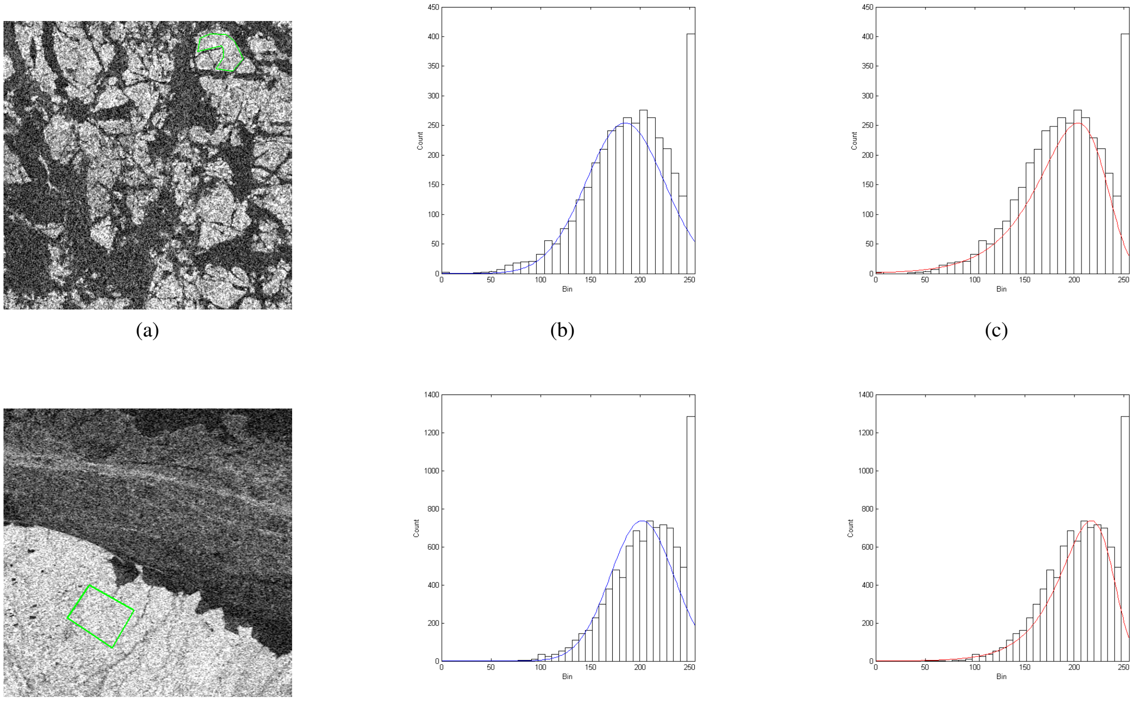
Due to the log-transform, the noise model can no longer be assumed to follow a Rayleigh distribution. The noise $\eta(x,y)$ is determined to take on the Fisher-Tippett distribution [15][22]. This noise distribution has been used to develop other despeckling algorithms and has been shown to be valid for ultrasound images, which use the same noise model as SAR images [22].

To demonstrate that this assumption is valid in SAR images, the noise characteristics of sample SAR images were analyzed. A homogeneous region of sea ice in two SAR images provided by MacDonald Dettwiler and Associates (MDA) Corporation from RADARSAT-2 were used for the analysis. The HV polarized scenes used are shown in Figures 1(a) and 1(d). The histogram of the pixel intensities in the regions highlighted in green are plotted. Furthermore, the Gaussian distribution was fit to the intensities in Figures 1(b) and 1(e), and Fisher-Tippett distribution was fit to the intensities in Figures 1(c) and 1(f). The fitted distributions are overlayed on top of the histogram and are scaled by the highest intensity value for comparison purposes. The noise appears to fit the Fisher-Tippett distribution better than the Gaussian distribution.

B. Sorted Local Texture Model

To capture the local texture characteristics within SAR images, a sorted local texture model is utilized. Sorted texture models are advantageous because they are rotationally invariant and take into account the spatial relationships within a texture [23]. The sorted local texture model can be described as follows. First, pixels in a specified local neighbourhood around the pixel of interest are extracted. The pixels are then grouped based on spatial distance from the pixel of interest. Next, pixels in the same group are sorted based on pixel intensity. This process is shown in Figure 2. The sorted texture descriptor for pixel m_k is represented by the vector $\boldsymbol{t_k}$.

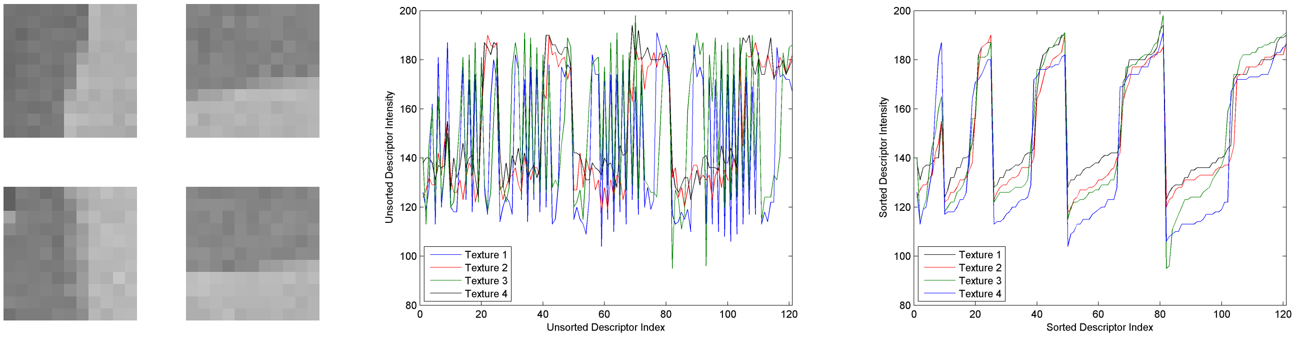
To demonstrate the rotational invariance of the sorted local texture model, Figure 3(a) shows a collection of 11×11 neighbourhoods with highly similar textural characteristics with different orientations. Figure 3(b) shows the corresponding pixel intensities when an unsorted local texture model is used and Figure 3(c) shows the pixel intensities when the sorted local texture model is used. The horizontal axes in those figures are the indices of the texture descriptor vector. While an unsorted local texture model produces texture representations for the neighbourhoods that are very different from each other,



(a) (b) (c)

(d) (e) (f)

Fig. 1. Using the Fisher-Tippett distribution to model noise in SAR images. The histogram of pixel intensities in the highlighted region in (a) are plotted in (b) and (c). In (b), a Gaussian distribution is fit to the histogram and in (c), a Fisher-Tippett distribution is fit. ©MacDonald, Dettwiler and Associates Ltd. 2010. All rights reserved.



(a) (b) (c)

Fig. 3. Comparison of the sorted local texture model versus unsorted local texture model. Four highly similar texture 11×11 neighbourhoods with different orientations, shown in (a), are taken from the Pentagon scene and used to construct texture representations. The texture representations produced using the unsorted local texture model (b) appear highly dissimilar from each other, while texture representations using the sorted local texture model (c) are highly similar.

the sorted local texture model produces highly similar texture representations.

C. Problem Formulation

The following is the derivation of the problem formulation used to estimate the actual signal s from the measured signal m [24]. In order to estimate the actual signal from Equation 2, the inverse problem can be formulated as Bayesian least squares using the posterior distribution $p(s|m)$:

$$\begin{aligned}\hat{s} &= \arg\min_{s}\{E((s-\hat{s})^2|m)\}\\ &= \arg\min_{s}\Big(\int (s-\hat{s})^2\, p(s|m)ds\Big)\end{aligned} \tag{3}$$

To minimize the estimate of s, the derivative of the $\arg\min$ argument from (3) is taken and set to zero:

$$\frac{\partial}{\partial \hat{s}} \int (s-\hat{s})^2 \, p(s|m) ds = \int 2(s-\hat{s}) \, p(s|m) ds = 0$$

$$\int s \, p(s|m) ds = \int \hat{s} \, p(s|m) ds \quad (4)$$

The right hand side of (4) can be simplified:

$$\begin{aligned} \int \hat{s} \, p(s|m) ds &= \hat{s} \int p(s|m) ds \\ &= \hat{s}, \end{aligned} \quad (5)$$

Finally, using the result of (5), the estimate $\hat{s}$ can be expressed as the conditional mean of s given m:

$$\begin{aligned} \hat{s} &= \int s \, p(s|m) ds \\ &= E(s|m). \end{aligned} \quad (6)$$

However, estimating the conditional mean directly is very challenging, as the posterior $p(s|m)$ is generally not known. Therefore, a Monte Carlo texture-likelihood sampling approach is used instead to estimate the posterior [25]. Using a sampling-based approach avoids assuming a parametric model for the posterior distribution $p(s|m)$. The proposed sampling algorithm allows speckle statistics and texture characteristics to be incorporated under a common framework for estimating the true signal.

D. Posterior Estimation using Monte Carlo Texture Likelihood Sampling

This section outlines the methodology for the Monte Carlo texture likelihood sampling. First, k is defined to be the location of a pixel in the original log-compressed SAR scene and m_k is the pixel intensity in the measure scene at location k. Therefore,

$$k \triangleq (x, y). \quad (7)$$

In this implementation of Monte Carlo sampling, a set of representative samples Ω and associated importance weights are determined from the other pixels m_k in a search space surrounding the pixel of interest m_0. The pixels m_k in the search space are selected based on a uniform instrumental distribution $Q(m_k|m_0)$. A uniform distribution is used to give all pixels in the search space around m_0 an equal probability of being selected.

Once the subset of pixels has been selected, an acceptance probability $\alpha(m_k|m_0)$ is calculated for each selected pixel. The acceptance probability function $\alpha(m_k|m_0)$ compares the textural characteristics of the neighbourhood around a selected pixel m_k with that of the neighbourhood around m_0 to determine if m_k is a realization of $p(s|m)$. To achieve this goal, the concept of texture likelihood is introduced into the Monte Carlo sampling framework. Let the sorted texture representation of m_k ($\boldsymbol{t_k}$) be a realization of the sorted texture representation of m_0 ($\boldsymbol{t_0}$), contaminated by a noise process η_{m_k,m_0}:

$$\boldsymbol{t_k} = \boldsymbol{t_0} + \eta_{m_k,m_0}. \quad (8)$$

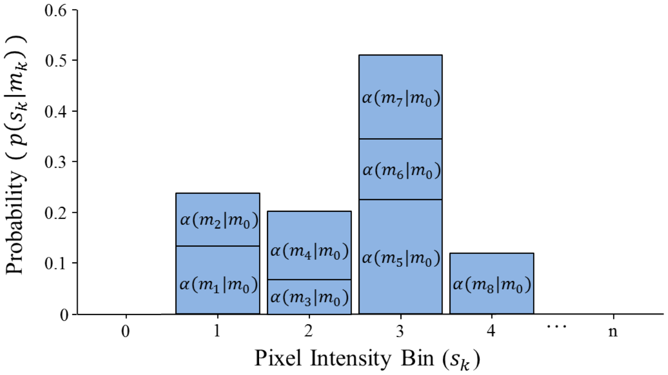


Fig. 4. Sample posterior distribution $\hat{p}\,(s|m)$, built from pixels accepted in the set Ω. Each stacked element corresponds to a pixel m_k in Ω, where the height is $\alpha(m_k|m_0)$ and bin location is m_k. The histogram is normalized so that $\sum_k \hat{p}\,(s_k|m_k) = 1$.

Based on the assumption that the noise process η_{m_k,m_0} follows a Fisher-Tippett distribution [26], the texture likelihood function $\ell(m_k|m_0)$ can be defined as:

$$\ell(m_k|m_0) = \prod_j \frac{1}{\beta} e^{[\frac{t_k[j]-t_0[j]}{\beta} + e^{[\frac{t_k[j]-t_0[j]}{\beta}]}]} \quad (9)$$

where $t_k[j]$ and $t_0[j]$ are the j^{th} elements from the sorted local texture representations corresponding to m_k and m_0. The scaling parameter β comes directly from the scene to be despeckled. Using a homogeneous region in the scene, the histogram is constructed and the Fisher-Tippett distribution is fit to that histogram. The β parameter found using maximum likelihood estimation is used directly in despeckling.

Given the texture likelihood defined in Equation 9, the acceptance probability $\alpha(m_k|m_0)$ can be defined as

$$\alpha(m_k|m_0) = \frac{\ell(m_k|m_0)}{\prod_j exp(\lambda)} \quad (10)$$

The denominator contains the term $\exp(\lambda)$, which is a normalization term so that $\alpha(m_k|m_0)$ is 1 if the sorted local texture representations $\boldsymbol{t_0}$ and $\boldsymbol{t_k}$ are identical. Elements in the local texture representations are assumed to be independent, which is why the acceptance probability is the product of the probabilities from each site in the neighbourhood.

The pixel m_k is accepted into the set Ω with a probability of $\alpha(m_k|m_0)$ and is used to estimate the posterior distribution $p(s|m)$. Furthermore, $\alpha(m_k|m_0)$ is also used as the associated importance weight for the accepted pixel. To perform the acceptance, a random value u is drawn from a uniform distribution $U(0, 1)$. If $u \leq \alpha(m_k|m_0)$, then m_k is accepted into Ω.

This process is repeated until the desired number of samples N are selected from the search space and the posterior distribution is calculated as a weighted histogram. This process is shown as an example histogram in Figure 4. Each element of the stacked bar chart corresponds to a pixel m_k in the set Ω. The height of the element being added to the histogram is the importance weight $\alpha(m_k|m_0)$. Each element is put into the histogram bin equal to the value of the pixel intensity m_k.

In Figure 4, the set Ω is comprised of eight pixels (m_1 to m_8) and the pixel intensities can range from 0 to n. Finally, the histogram is normalized, so that $\sum_k \hat{p}\left(s_k|m_k\right) = 1$. Based on the estimated $\hat{p}\left(s|m\right)$, the conditional mean from (6) can be solved to obtain $\hat{s}$. For testing, samples are drawn from 11×11 search spaces, while texture descriptors were computed from 7×7 neighborhoods.

E. Summary of MCTLS Algorithm

To provide a clearer understanding of the algorithm methodology, the following summarizes the steps involved in the MCTLS algorithm:

1) Iterating through each pixel of interest m_0 in the image, randomly draw a sample m_k from a search space around m_0 using a uniform instrumental distribution.
2) Determine the sorted texture descriptors $\boldsymbol{t_0}$ and $\boldsymbol{t_k}$ (Section II-B).
3) Compute acceptance probability $\alpha(m_k|m_0)$ using Equation 10.
4) Generate a random value u from a uniform distribution $U(0,1)$. If $u \leq \alpha(m_k|m_0)$, include k in the set of accepted samples Ω with an importance weight equal to $\alpha(m_k|m_0)$. Otherwise, m_k is discarded.
5) Repeat steps 1-4 until the desired number of samples N are selected from the search space.
6) Estimate the posterior distribution $\hat{p}(s|m)$ based on the samples in Ω and their importance weights.
7) Compute the noise-free estimate $\hat{s}$ for pixel m_0 as the conditional mean in Equation 6.

III. Testing

Testing the MCTLS algorithm involves comparing to other existing algorithms. The state-of-the-art algorithms used for comparison are the following: Lee filter [6], Frost filter [7], Gamma MAP [8], speckle reducing anisotropic diffusion (SRAD) [10], wavelet-based multiresolution bilateral filtering [12], and block matching and 3D filtering (BM3D) [14]. The log-compressed image was used when despeckling using the MCTLS, wavelet-based and BM3D algorithms.

The first comparison was performed using an image where synthetic speckle noise has been added manually. This allows the despeckled images to be compared to the original noise-free image. An image of the Pentagon from the USC-SIPI image database [27] is used and is shown in Figure 5(a). The image with Fisher-Tippett speckle noise added ($\beta = 30$) is shown in Figure 5(b). The speckle noise was added using the *evrnd* function in MATLAB to generate the random noise with specified β and then, the pixel intensities were clamped to [0,255]. In addition to a visual comparison, the algorithms were compared using peak signal-to-noise ratio ($PSNR$), contrast-to-noise ratio (CNR), and structural similarity index ($SSIM$). These metrics are commonly used to assess image quality after despeckling [28] [29]. The comparison metrics are defined in (11) to (13).

$$PSNR = \left[\frac{max(\hat{S}^2)}{\frac{1}{rc}\sum_{i=1}^{r}\sum_{j=1}^{c}[\hat{S}(i,j) - S(i,j)]^2}\right] \tag{11}$$

$$CNR = \frac{1}{H}\sum_{h=1}^{H}\frac{\mu_h - \mu_b}{\sigma_n} \tag{12}$$

$$SSIM = \frac{(2\mu_S\mu_{\hat{S}} + c_1)(2\sigma_{S\hat{S}} + c_2)}{(\mu_S^2 + \mu_{\hat{S}}^2 + c_1)(\sigma_S^2 + \sigma_{\hat{S}}^2 + c_2)} \tag{13}$$

PSNR (dB) is the ratio between the maximum signal intensity and noise. Equation 11 shows how to calculate PSNR for the despeckled image $\hat{S}$ of size r by c using the noise-free reference image S. CNR is the ratio between the contrast of a background region versus a foreground region and the noise deviation. Equation 12 calculates the CNR, where μ_h is the mean intensity of a homogeneous foreground region and μ_b is the mean intensity for a background region. The blue region in Figure 5(c) is chosen as the background region, and the red regions are the foreground regions. The parameter σ_n is the standard deviation of noise, estimated by comparing the noisy and original noise-free images. The green highlighted area is used only as a region for visual comparison.

Finally, SSIM is an image quality metric that quantifies the change in structural information compared to a reference image [29]. Equation 13 calculates SSIM, where μ_S and $\mu_{\hat{S}}$ are average intensities in despeckled image S and noise-free image $\hat{S}$, σ_S^2 and $\sigma_{\hat{S}}^2$ are variances in those images, and c_1 and c_2 are constants. In addition to the image quality metrics, algorithm runtime is measured to compare computational complexity.

The second comparison involves despeckled real SAR images using the algorithms. The three SAR scenes used for comparison are shown in Figure 6. Due to the size of a real SAR scene and time constraints, the entire scene was not compared, but instead regions from three images were used and are outlined in red in Figure 6. The C-band SAR images were provided by MDA Corporation and were acquired by RADARSAT-2 in ScanSAR Wide beam mode. The HV polarized scene is of Arctic sea ice with an approximate resolution of 100 m × 100 m. These results were compared visually and quantitatively by measuring the equivalent number of looks (ENL). The equation for ENL is given in (14), where μ_h and σ_h^2 are the mean and variance associated with a set of H homogeneous regions in the scene. The regions used to calculate ENL for each scene are highlighted in Figures 11(a), 12(a) and 13(a).

$$ENL = \frac{1}{H}\sum_{h=1}^{H}\frac{\mu_h^2}{\sigma_h^2} \tag{14}$$

IV. Results

A. Synthetic noise image results

Figure 7 shows the original image, noisy image and the resulting images after despeckling enlarged to a region of interest. This region of interest corresponds to the green region in Figure 5c and provides a better visualization of texture detail preservation. The image quality metrics for each algorithm and noise-level are used to generate PSNR, CNR and SSIM curves in Figures 8, 9 and 10. The runtime for each algorithm

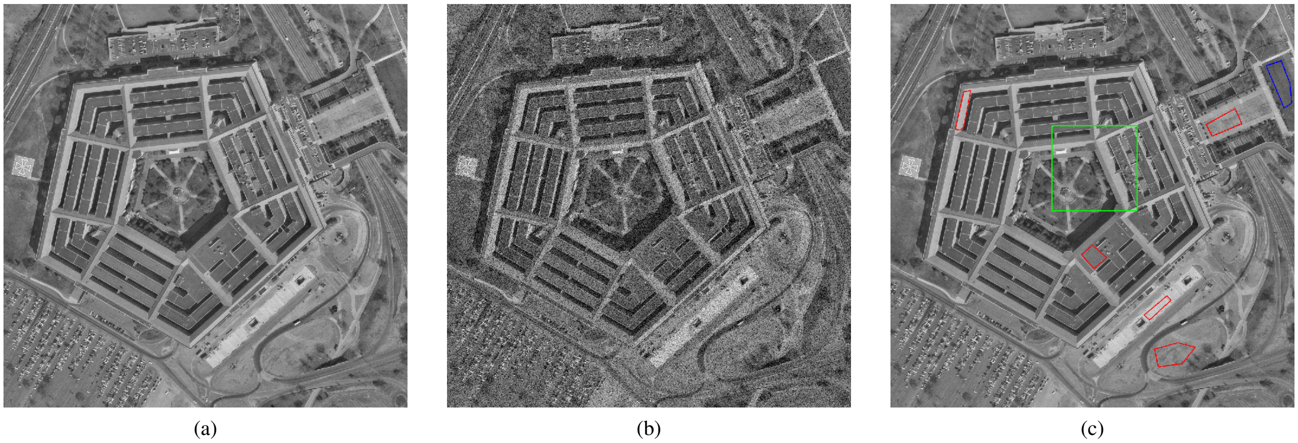

(a) (b) (c)

Fig. 5. Image with synthetic noise used for testing: (a) original noise-free image; (b) image with speckle noise ($\beta = 30$) added; and (c) regions used for image quality metrics. In (c), the blue outline indicates the homogeneous background region, the red outlines indicate the homogeneous foreground regions, and the green outline indicates a region enlarged for visual comparison.

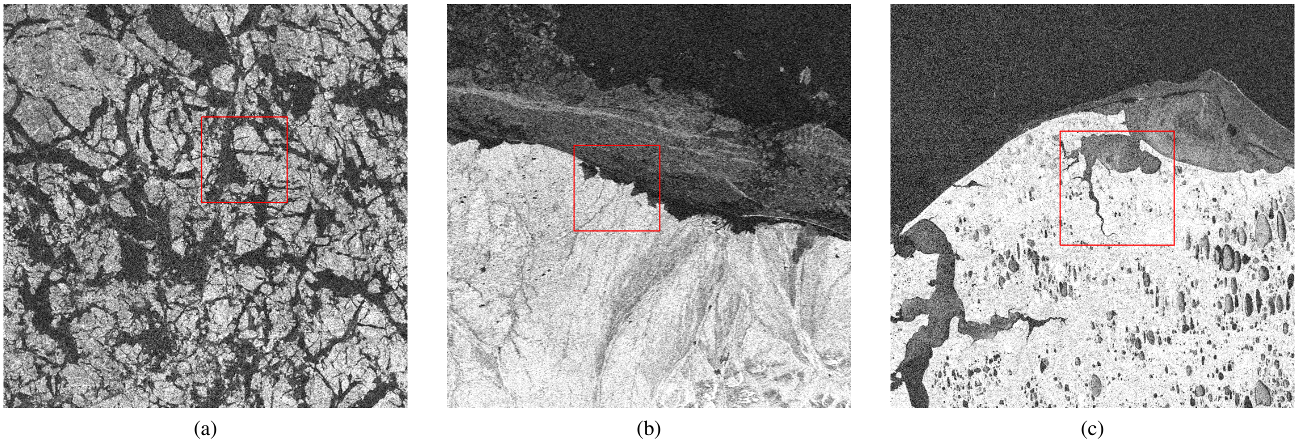

(a) (b) (c)

Fig. 6. Three SAR scenes used for testing. The red outlines indicate the region enlarged for visual comparison in Figures 11, 12 and 13. ©MacDonald, Dettwiler and Associates Ltd. 2010. All rights reserved.

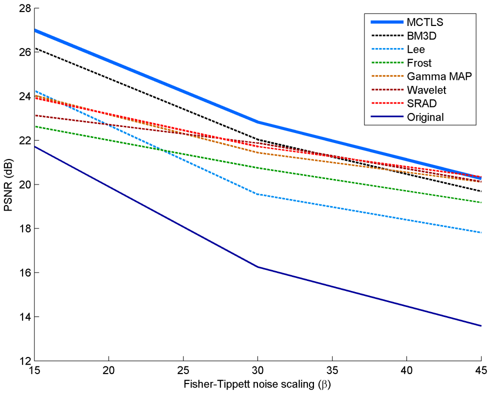


Fig. 8. Peak signal-to-noise ratio curves for each despeckling algorithm at increasing levels of the noise scaling parameter in Fisher-Tippett speckle noise.

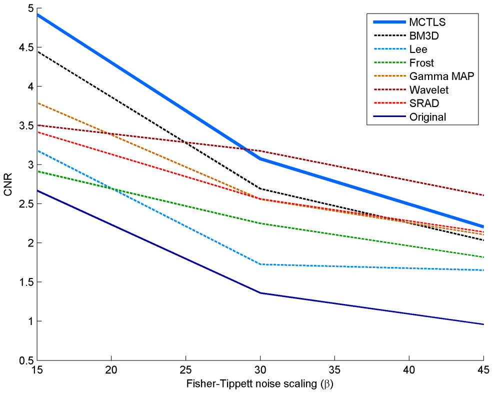
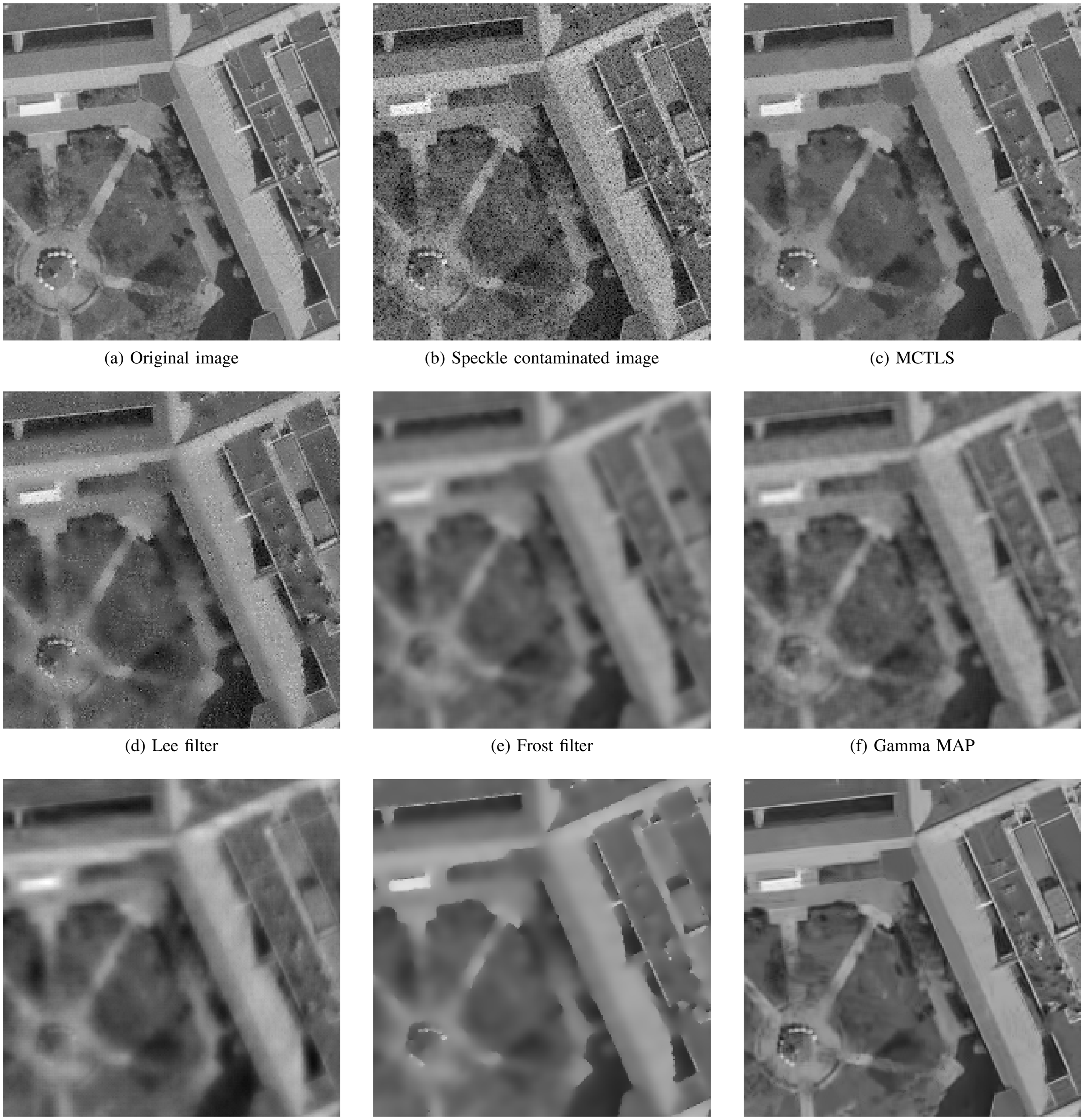


Fig. 9. Contrast-to-noise ratio curves for each despeckling algorithm at increasing levels of the noise scaling parameter in Fisher-Tippett speckle noise.

to despeckle a 512×512 synthetic SAR image is provided in Table I.

From Figure 8, the MCTLS algorithm produces despeckled images with the highest PSNR for all tested images with synthetic speckle noise. This is because it takes the Fisher-Tippett distribution of speckle noise directly into account.



(a) Original image

(b) Speckle contaminated image

(c) MCTLS

(d) Lee filter

(e) Frost filter

(f) Gamma MAP

(g) Wavelet-based bilateral filtering

(h) SRAD

(i) BM3D

Fig. 7. Despeckling results, enlarged to show the green highlighted region in Figure 5c: (a) original image; (b) speckle contaminated image ($\beta = 30$); (c) MCTLS; (d) Lee filter [6]; (e) Frost filter [7]; (f) Gamma MAP [8]; (g) wavelet-based bilateral filtering [12]; (h) SRAD [10]; and (i) BM3D [14].

TABLE I
RUNTIME FOR THE DESPECKLING ALGORITHMS ON A SYNTHETIC 512×512 SAR SCENE

Methods	Runtime (s)
MCTLS	58.93
Lee	15.86
Frost	24.71
Gamma MAP	15.39
Wavelet	12.78
SRAD	2.32
BM3D	3.67

From Figure 9, the MCTLS algorithm has the highest CNR at low noise levels, but the wavelet-based algorithm performs better at higher noise levels. Finally, from Figure 10, the MCTLS algorithm and BM3D algorithm have the highest SSIM at all noise levels. SSIM is a measure of how the algorithms preserve structure compared to the original noise-free image. This is an indication that the MCTLS algorithm is able to preserve edges and structures when despeckling.

Visually, from Figure 7, the classic despeckling algorithms (Lee filter, Frost filter, and Gamma MAP filter) result in

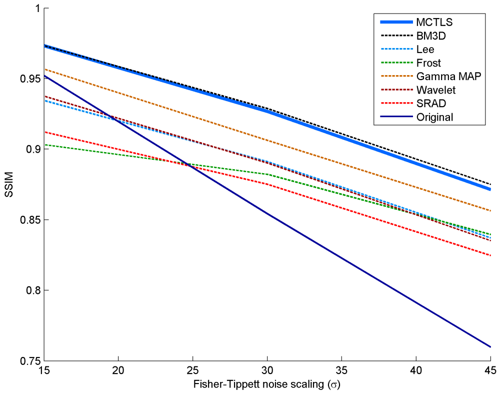


Fig. 10. Structural similarity index curves for each despeckling algorithm at increasing levels of the noise scaling parameter in Fisher-Tippett speckle noise.

blurred images with little detail. This is reflected in the low SSIM values for those algorithms. The wavelet algorithm performs better than the classic algorithms, but still results in a noticeable loss in detail. The SRAD algorithm is able to preserve large-scale edges well, but blurs details by causing many areas of the image to become homogeneous. Finally, the BM3D algorithm and MCTLS algorithm perform comparably as they are both able to remove noise while preserving detail, with the MCTLS algorithm is better at preserving textural characteristics. This can be seen in Figures 7(c) and 7(i) with the roof and grass textures. While the proposed algorithm retains some of the original grass texture, with trees and shadows, the BM3D algorithm blurs the grass texture. Similarly, the details in the roof are preserved with the proposed algorithm, but blurred with the BM3D algorithm.

To measure computational complexity, the runtime for each despeckling algorithm applied to a 512×512 image is given in Table I. The algorithms were implemented in MATLAB and run on a computer with an Intel Core i5-2400S CPU (2.5 GHz, 6 GB RAM). While the MCTLS algorithm has the highest runtime, optimizing and utilising parallel processing can greatly increase the efficiency of the algorithm. Furthermore, the MCTLS algorithm scales linearly with the number of pixels.

Finally, bias in the denoised image is already accounted for by the MCTLS algorithm. This is because the algorithm uses the noise distribution directly. However, this noise bias may be present in the corrected images produced using other despeckling algorithms. This can partially explain the difference in PSNR between BM3D and the proposed algorithm, even though the SSIM metrics are close. SSIM accounts for similarity in visual detail between the corrected and original image but does not account for signal fidelity and restoration quality. Overall, MCTLS is an algorithm with a straightforward implementation that shows strong despeckling results.

B. SAR image results

TABLE II
EQUIVALENT NUMBER OF LOOKS FOR THE DESPECKLED SAR SCENES

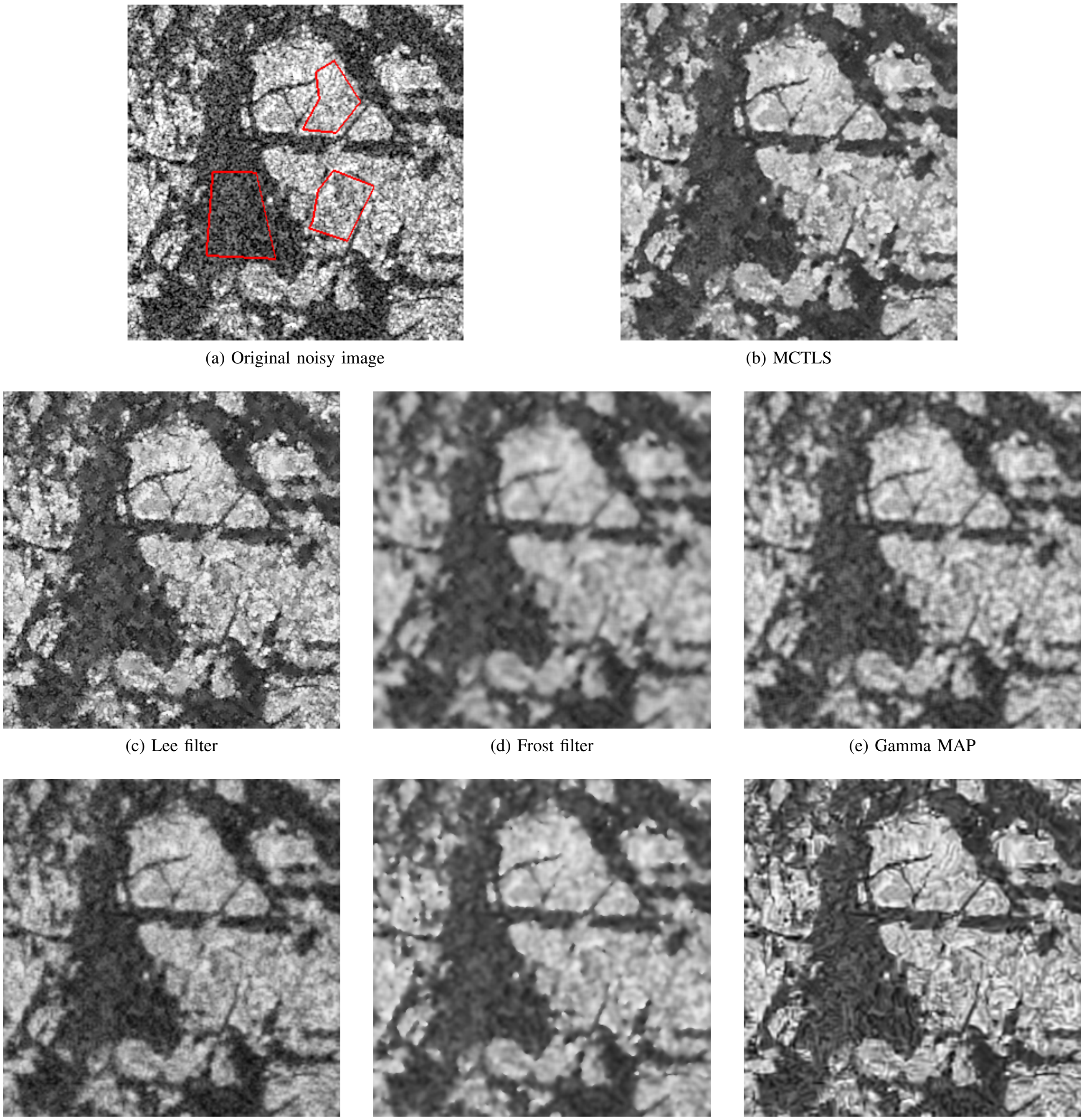
Methods	Equivalent Number of Looks		
	Scene 1	Scene 2	Scene 3
Original Image	12.08	32.01	57.94
MCTLS	38.93	203.34	595.32
Lee	23.86	123.89	706.11
Frost	84.66	444.55	793.87
Gamma MAP	50.50	186.57	447.11
Wavelet	41.50	284.67	706.73
SRAD	65.36	119.82	532.42
BM3D	29.85	127.53	438.37

Figures 11, 12 and 13 show despeckled SAR images using the proposed and comparison algorithms. The equivalent number of looks metric for each SAR scene and denoising algorithm are shown in Table II. As seen in the synthetic images, the classic filters (Lee filter, Frost filter, and Gamma MAP filter) result in images that appear smooth, with many details removed or blurred. The SRAD algorithm can preserve edges, but the homogeneous regions in the scene appear to be too smooth. The Wavelet-based bilateral filtering removes noise but it results in a noticeable loss in detail. These algorithms also had the highest ENL, but this can be because they produced images with less detail and very low variance. As a result, the ENL values tend to be higher than the proposed MCTLS algorithm.

Again, the BM3D and MCTLS algorithms perform comparably in terms of speckle noise reduction. However, prominent ripple-like artifacts are introduced by the despeckled images produced using BM3D, which are not present in that proposed using the proposed method. Furthermore, the proposed method does a better job preserving texture detail. This is important for texture classifiers, since introducing artificial textures can cause the classifiers to misclassify areas. In terms of ENL, the MCTLS algorithm has a higher ENL than BM3D for all three SAR scenes.

V. CONCLUSION

A novel algorithm for despeckling SAR images using Monte Carlo texture likelihood sampling has been proposed. The noise statistics were examined and found that speckle noise in a log-compressed SAR image follows a Fisher-Tippett distribution. Based on a sorted local texture model and a logarithmic-space Fisher-Tippett speckle distribution model, a Monte Carlo texture likelihood sampling posterior estimation algorithm is proposed to estimate the true signal. Testing on synthetic noisy images and real SAR images found that the MCTLS algorithm shows strong despeckling performance when compared to state-of-the-art despeckling algorithms. Future work involves combining the proposed methodology directly with segmentation and classification algorithms to investigate the effectiveness of such combined systems for improving segmentation and classification of SAR images. Furthermore, adapting the MCTLS algorithm to denoise other types of images while preserving underlying structural and texture detail will be investigated.



(a) Original noisy image

(b) MCTLS

(c) Lee filter

(d) Frost filter

(e) Gamma MAP

(f) Wavelet-based bilateral filtering

(g) SRAD

(h) BM3D

Fig. 11. Results from despeckling the SAR image in Figure 6(a), enlarged to show the red highlighted region: (a) original noisy image; (b) MCTLS; (c) Lee filter [6]; (d) Frost filter [7]; (e) Gamma MAP [8]; (f) wavelet-based bilateral filtering [12]; (g) SRAD [10]; and (h) BM3D [14]. The regions highlighted in red are the regions used to calculate ENL. ©MacDonald, Dettwiler and Associates Ltd. 2010. All rights reserved.

VI. Acknowledgements

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Geomatics for Informed Decisions (GEOIDE) Network for financially supporting this project.

References

[1] J. C. Curlander and R. N. McDonough, *Synthetic Aperture Radar - Systems and Signal Processing*. Toronto, Canada: John Wiler and Sons, 1991.

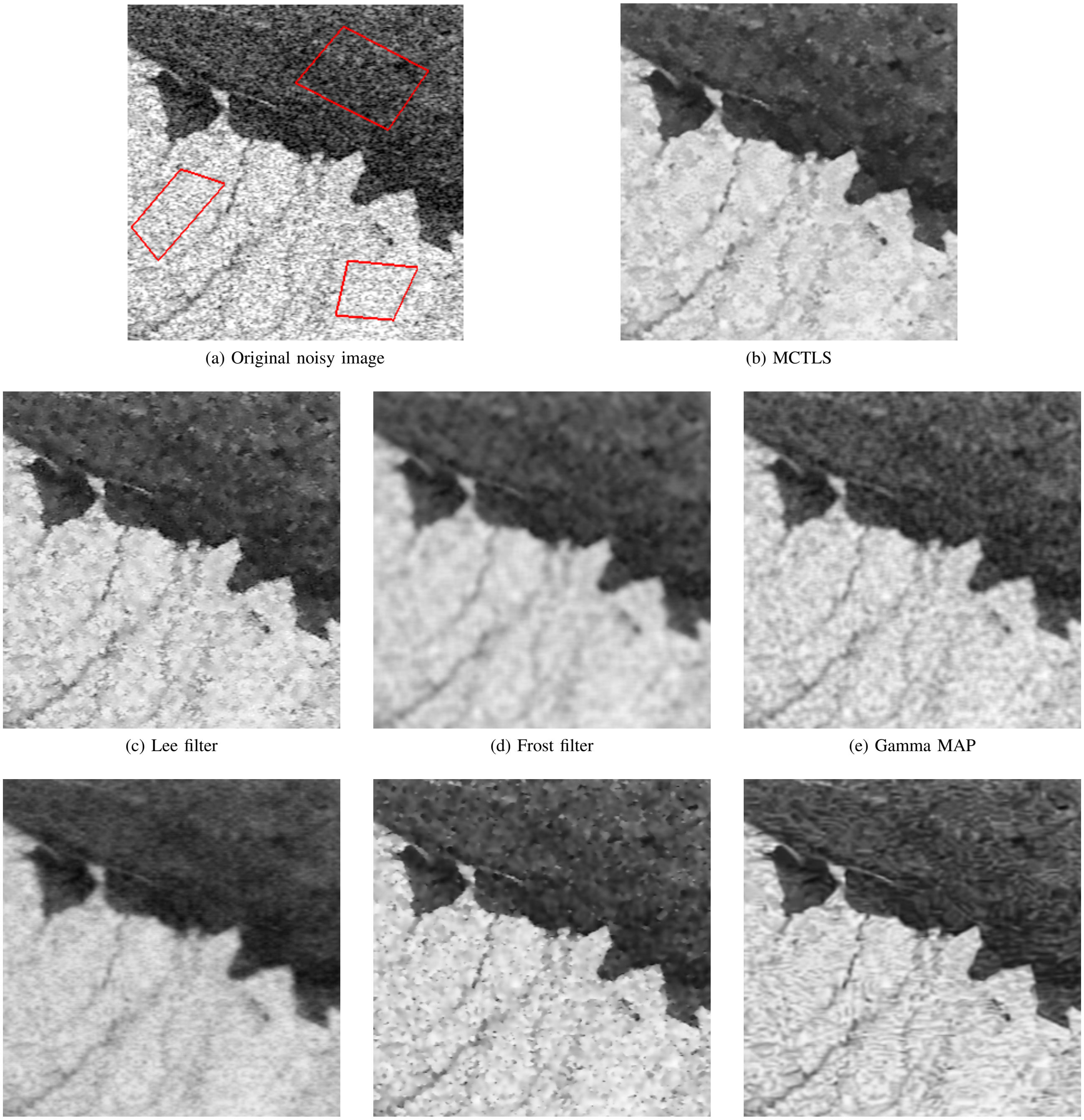
[2] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*. Norwood, MA: Artech House, 1998.

[3] F. Li, C. Croft and D. N. Held, “Comparison of several techniques to obtain multiple-look SAR imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. Ge-21, no. 3, July 1983.

[4] L. Gagnon and A. Jouan, “Speckle filtering of SAR images - A comparative study between complex-wavelet-based and standard filters,” *Proceedings of SPIE*, vol. 3169, pp. 80-91, 1997.

[5] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Upper Saddle River, NJ: Pearson Prentice Hall, 2008.

[6] J. S. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, Mar. 1980.



(a) Original noisy image

(b) MCTLS

(c) Lee filter

(d) Frost filter

(e) Gamma MAP

(f) Wavelet-based bilateral filtering

(g) SRAD

(h) BM3D

Fig. 12. Results from despeckling the SAR image in Figure 6(b), enlarged to show the red highlighted region: (a) original noisy image; (b) MCTLS; (c) Lee filter [6]; (d) Frost filter [7]; (e) Gamma MAP [8]; (f) wavelet-based bilateral filtering [12]; (g) SRAD [10]; and (h) BM3D [14]. The regions highlighted in red are the regions used to calculate ENL. ©MacDonald, Dettwiler and Associates Ltd. 2010. All rights reserved.

[7] V. S. Frost, J. A. Stiles, K. S. Shanmugan, and J. C. Holtzman, "A model for radar images and its application to adaptive digital filtering of multiplicative noise," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, Mar. 1982.

[8] A. Lopes, E. Nezry, R. Touzi, and H. Laur, "Structure detection and statistical adaptive speckle filtering in SAR images," *International Journal of Remote Sensing*, vol. 14, no. 9, pp. 1735-1758, 1993.

[9] Z. Shi and K. B. Fung, "A comparison of digital speckle filters," *Proceedings of 1994 International Geoscience And Remote Sensing Symposium*, pp. 2129-2133, August 1994.

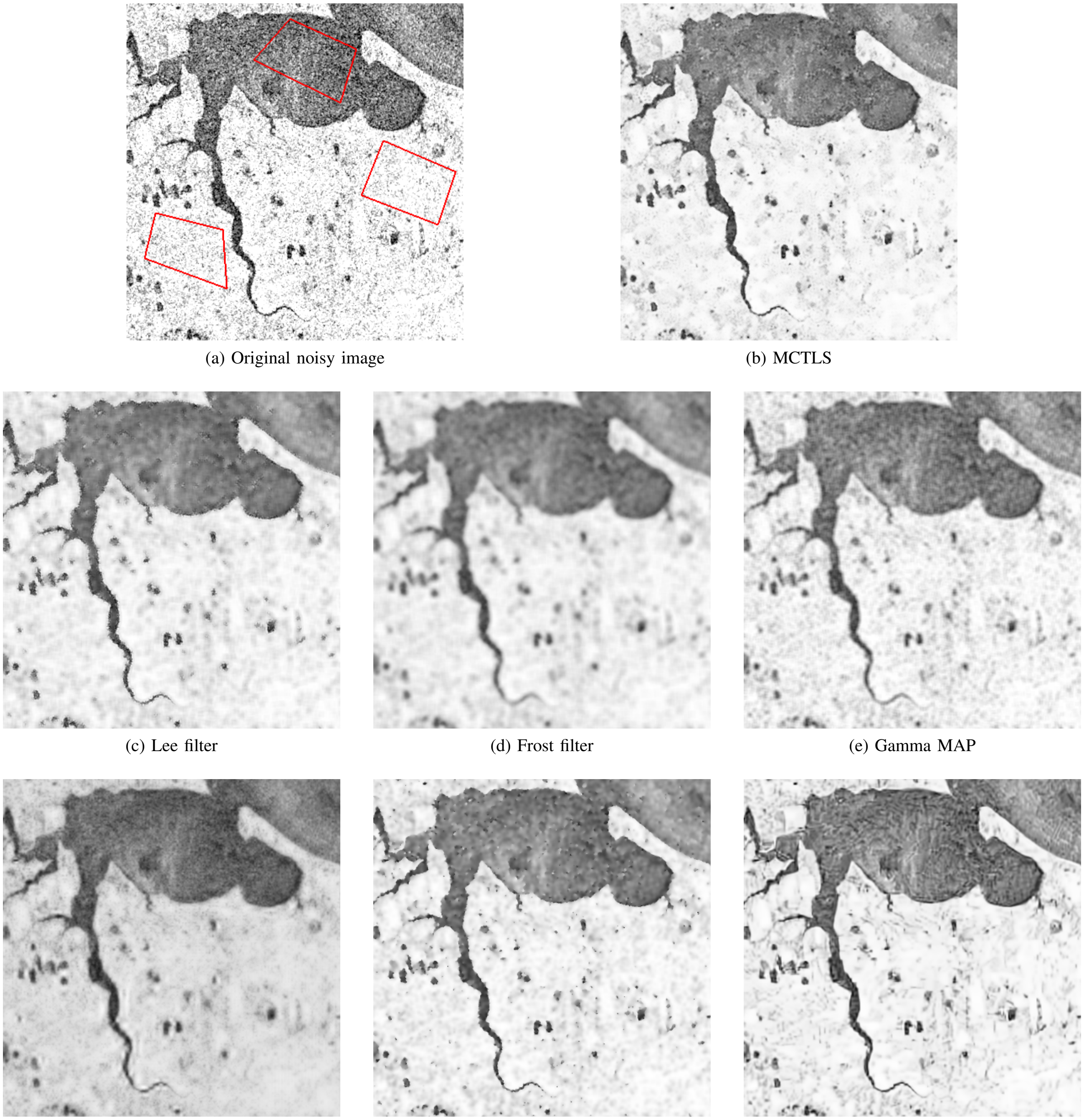
[10] Y. Yu, and S. T. Acton, "Speckle reducing anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 11, no. 11, November 2002.

[11] S. Easanuruk, S. Mitatha, S. Intajag, S. Chitwong, "Speckle reduction using fuzzy morphological anisotropic diffusion," *Proceedings of Fifth International Conference on Information, Communications and Signal Processing*, pp. 748-751, 2005.

[12] B. Lu, and Y. Ku, "Speckle reduction with multiresolution bilateral filtering for SAR image," *Proceedings of 2010 International Conference on Machine Vision and Human-machine Interface*, pp. 700-703, 2010.

[13] A. Achim, P. Tsakalides, and A. Bezerianos, "SAR image denoising via Bayesian wavelet shrinkage based on heavy-tailed modeling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 8, pp. 1773-1783, August 2003.

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, August, 2007.



(a) Original noisy image

(b) MCTLS

(c) Lee filter

(d) Frost filter

(e) Gamma MAP

(f) Wavelet-based bilateral filtering

(g) SRAD

(h) BM3D

Fig. 13. Results from despeckling the SAR image in Figure 6(c), enlarged to show the red highlighted region: (a) original noisy image; (b) MCTLS; (c) Lee filter [6]; (d) Frost filter [7]; (e) Gamma MAP [8]; (f) wavelet-based bilateral filtering [12]; (g) SRAD [10]; and (h) BM3D [14]. The regions highlighted in red are the regions used to calculate ENL. ©MacDonald, Dettwiler and Associates Ltd. 2010. All rights reserved.

[15] H. Xie, L. E. Pierce, and F. T. Ulaby, “Statistical properties of logarithmically transformed speckle,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 3, pp. 721-727, March 2002.

[16] Q. A. Holmes, D. R. Nuesch, and R. A. Shuchman, “Textural analysis and real-time classification of sea-ice types using digital SAR data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-22, no. 2, pp. 113-120, March 1984.

[17] L. Soh and C. Tsatsoulis, “Texture analysis of SAR sea ice imagery using gray-level co-occurence matrices,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 780-795, March 1999.

[18] A. K. Jain and F. Farrokhnia, “Unsupervised texture segmentation using Gabor filters,” *Pattern Recognition*, vol. 24, no. 12, pp. 11671186, 1991.

[19] O. D’Hondt, L. Ferro-Famil, and E. Pottier, “Nonstationary spatial texture estimation applied to adaptive speckle reduction of SAR data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 476-480, October 2006.

[20] H. Li, W. Hong, Y. Wu, and H. Tai, “Texture-preserving despeckling of SAR images using evidence framework,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 537-541, October 2007.

[21] E. Kuruoglu and J. Zerubia, “Modeling SAR images with a generalization of the Rayleigh distribution,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, April 2004.

[22] G. Slabaugh, G. Unal, T. Fang and M. Wels, “Ultrasound-specific segmentation via decorrelation and statistical region-based active contours,” *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 45-53, 2006.

[23] L. Liu, P. Fieguth, D. Clausi, and G. Kuang, "Sorted random projections for robust rotation-invariant texture classification," *Pattern Recognition*, vol. 45, issue 6, pp. 2405-2418, 2012.

[24] P. Fieguth, *Statistical Image Processing and Multidimensional Modeling*. New York, NY: Springer, 2011.

[25] M. Chen, "Importance-weighted marginal bayesian posterior density estimation," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 818-824, Sepember 1994.

[26] D. Kaplan and Q. Ma, "On the statistical characteristics of log-compressed Rayleigh signals," *Proceedings of IEEE Ultrasonics Symposium*, vol. 2, pp. 961964, 1993.

[27] USC-SIPI image database, University of South California, http://sipi.usc.edu/database/.

[28] D. C. Adler, T. H. Ko, and J. G. Fujimoto, "Speckle reduction in optical coherence tomography images by use of a spatially adaptive wavelet filter," *Optics Letters*, vol. 29, no. 24, pp. 2878-2880, 2004.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, April 2004.