geoconvention
Virtual Event
September 13 - 15
2021

Adapting natural-image machine-learned denoiser for noise suppression in seismic and drilling datasets

Nasser Kazemi [1*], *Roman Shor* [1], *Jean Auriol* [2], *and Ian Gates* [1]
[1] *University of Calgary*
[2] *University of Paris- Saclay, CNRS, L2S, Inria*
[*] *nasser.kazeminojadeh@ucalgary.ca*

Abstract

In the past decade or so, machine learning algorithms gained remarkable attraction in the signal processing community. The availability of big datasets and high-performance computing resulted in developing a library of machine-learned algorithms. However, such algorithms are biased towards the training dataset. Moreover, compared to natural images, the comprehensively labeled seismic and drilling datasets are not available. Thus, developing efficient and generalizable machine-learned denoisers for such datasets is challenging. Accordingly, the lack of generalizability of learned operators hinders the applicability of such operators in processing seismic and drilling datasets. To address this issue, we introduce a workflow that transfers the feed-forward denoising convolutional neural networks (DnCNN) operator from the camera images to the seismic/drilling domain. To do so, we consider the statistical differences between the camera images and seismic/drilling data and develop a simple yet efficient workflow for adapting the DnCNN operator to these new domains. The simulation and field datasets show that our proposed workflow outperforms the classical Fx-deconvolution and original DnCNN methods in suppressing noise in seismic and drilling datasets.

Introduction

Suppressing noise in seismic and measurement while drilling data is an active area of research (Canales, 1984; Abma and Claerbout, 1995; Brandon et al., 1999; Sinanovic et al., 2004; Trickett, 2008; Oropeza and Sacchi, 2011; Naghizadeh and Sacchi, 2012; Chen and Fomel, 2015; Yu et al., 2019). In multi-sensor recordings, the signal shows spatial coherency, and the noise component can be either spatially coherent or random. The denoising methods can be local or non-local (Bonar and Sacchi, 2012; Chen and Fomel, 2015). Several algorithms take advantage of the transformation techniques to separate the noise from the signal in the transformed domain and apply denoising by filtering or inversion (Canales, 1984; Abma and Claerbout, 1995; Trickett, 2008; Oropeza and Sacchi, 2011; Naghizadeh and Sacchi, 2012). These algorithms are used on datasets with different natures.

To give an idea about the similarity of the algorithms across domains, it suffices to mention that non-local means algorithm used for denoising camera images, MRI images, radar data, microscopy data, and seismic data (Buades et al., 2005; Coupé et al., 2008; Buades et al., 2010; Wei and Yin, 2010; Deledalle et al., 2010; Huang et al., 2011; Bonar and Sacchi, 2012). However, despite the tremendous progress in developing such algorithms, their performance depends on several assumptions that may not be satisfied from one dataset to another. Moreover, to achieve optimal performance, one should tune several parameters. Recently, authors explored the benefits of using deep learning-based denoising algorithms (Yu et al., 2019; Zhu et al., 2019; Richardson and Feller, 2019). These algorithms, once trained, are fast and efficient; however, their performance is biased towards the training dataset and lacks generalizability. To address this issue, we introduce a workflow that adapts the natural-image learned feed-forward denoising convolutional neural networks (DnCNN) to the seismic/drilling domain efficiently. The proposed workflow alleviates building comprehensive training/labeled data for seismic and drilling, hence it is generalizable after domain adaptation.

Machine-learned Denoiser and Domain Adaptation Strategy

DnCNN is a residual learning method, which is combined with batch normalization to increase performance and learning speed (He et al., 2016; Zhang et al., 2017). In other words, the network learns to remove the latent clean image, which is hidden in the layers of the network, and predict the noise component as a residual output. The network has a simple structure and can be efficiently parallelized to take advantage of advanced computational resources. Hence, we adopt the DnCNN method as a noise estimating operator. DnCNN works as follows. Consider a noisy gray-colored image, i.e., it has only one image channel,

$$\mathbf{d} = \mathbf{s} + \mathbf{n}, \tag{1}$$

where $\mathbf{d}$ is a noisy image, $\mathbf{s}$ is a clean image, and $\mathbf{n}$ is an additive white Gaussian noise. In this model, DnCNN acts as a noise estimating operator

$$\mathscr{L}(\mathbf{d}) \approx \mathbf{n}, \quad \text{and} \quad \mathbf{s} \approx \mathbf{d} - \mathscr{L}(\mathbf{d}), \tag{2}$$

where $\mathscr{L}(\cdot)$ is the DnCNN-learned mapping operator. To learn the model, DnCNN minimizes the averaged mean-squared-error between the estimated and ground-truth noise components by solving

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \quad \frac{1}{2N} \sum_{j=1}^{N} \|\mathscr{L}(\mathbf{d}_j;\Theta) - (\mathbf{d}_j - \mathbf{s}_j)\|_F^2, \tag{3}$$

where Θ are the trainable parameters in DnCNN, $\mathbf{d}_j$ and $\mathbf{s}_j$ are the j^{th} noisy-clean training image (patch) pairs, N is the total number of images in the training library, and F is Frobenius norm. Figure 1 shows the schematics of the DnCNN architecture for learning the mapping function.

The DnCNN has a deep architecture with three types of layers. The layers are built by combining a convolutional layer (Conv) with a Rectifier Linear Unit (ReLU) (Krizhevsky et al., 2012), and batch normalization (BN) (Ioffe and Szegedy, 2015). The first layer is Conv+ReLU with 64 filters of size $3 \times 3 \times 1$, which generates 64 feature maps, and ReLU is used to promote non-linearity. ReLU acts as a function that outputs the positive values of the input and zeros out the negative part, i.e., ReLU($input$)=$max(0, input)$. From the second layer to the $D-1$ layer, we have Conv+ BN+ReLU, where D is the depth of DnCNN architecture. In these layers, we have 64 filters of size $3 \times 3 \times 64$, and then batch normalization and ReLU functions are applied to the filters. Finally, the last layer is the Conv layer with a single filter of size $3 \times 3 \times 64$ which reconstructs the output. Note that the gray images are used for training purposes. In a nutshell, DnCNN is a combination of residual learning formulation and batch normalization. To learn the training parameters, the optimization problem represented in Equation (3) is solved by applying the Adam algorithm (Kingma and Ba, 2014). For more information about DnCNN, readers are referred to (Zhang et al., 2017; Kazemi, 2020).

Now, considering that camera images are $\mathbf{s} \in \mathbb{R}^+$, while seismic/drilling data are $\mathbf{s} \in \mathbb{R}$, direct application of DnCNN operator $\mathscr{L}$ on seismic/drilling data will not result in satisfactory performances. Hence, we propose to apply the operator as

$$\mathscr{L}_{da}(\mathbf{d}) = \mathscr{L}(\mathbf{d}^+) - \mathscr{L}(\{-\mathbf{d}\}^+), \tag{4}$$

where $\mathscr{L}_{da}$ is domain-adapted DnCNN operator, and $\mathbf{d}^+ = max\{\mathbf{d}, 0\}$.

Numerical Examples

In this section, we apply the domain-adapted DnCNN denoiser on seismic and drilling datasets. In the case of sismic data, we also compare the results with original DnCNN and Fx-deconvolution denoiser. Note that we use the pre-trained DnCNN operators which are trained on gray-colored camera images. The package uses the DnCNN method developed by Zhang et al. (2017), and trains the DnCNN operators on data with selective ranges of standard deviations of noise.

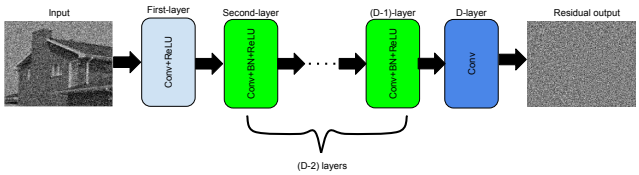

Figure 1 *The schematic representation of DnCNN architecture.*

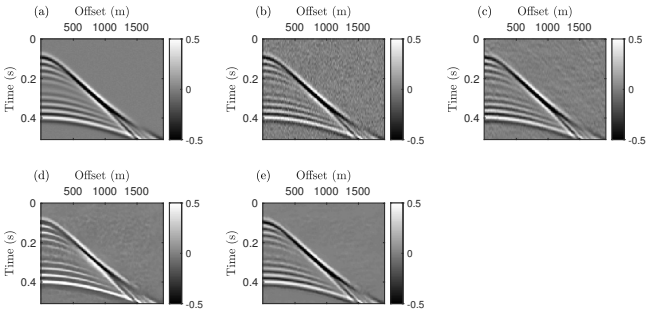

Figure 2 *Performances of denoisers. (a) Noise-free data. (b) Noisy data with SNR=2. (c)-(e) denoising with F-x deconvolution, DnCNN, and domain-adapted DnCNN, respectively.*

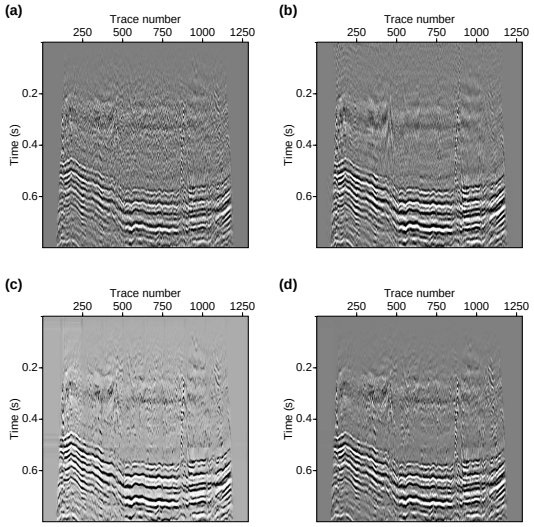

Figure 3 *Performances of denoisers on real data. (a) Recorded data. (b)-(d) denoising with Fx-deconvolution, DnCNN, and domain-adapted DnCNN, respectively.*

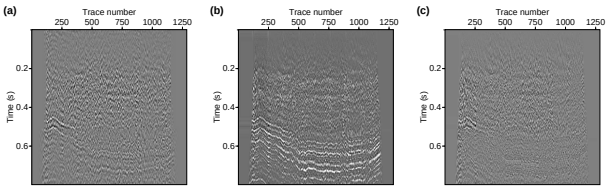

Figure 4 *Estimated noise panels for real data in Figure 3. (a)-(c) Estimated noise by using Fx-deconvolution, DnCNN, and domain-adapted DnCNN, respectively.*

Seismic noise suppression

To evaluate the performances of the DnCNN, Fx-deconvolution, and domain-adapted DnCNN, we generate a synthetic shot gather with strong and week amplitude events (Figure 2a). Then, we add a band-limited white Gaussian noise, with SNR=2, on the data (Figure 2b). The denoising results after applications of Fx-deconvolution, DnCNN, and domain-adapted DnCNN denoisers are depicted in Figures 2c- 2e, respectively. The quality of reconstruction computed by $Q = 10\log \frac{\|\mathbf{s}^{true}\|_2^2}{\|\mathbf{s}^{est}-\mathbf{s}^{true}\|_2^2}$, where $\mathbf{s}^{est}$ is estimated signal, $\mathbf{s}^{true}$ is ground-truth signal, and Q is quality of reconstruction. In Figure 2, the quality of reconstructions are $Q = 13.71,\ 5.99,\ 17.41$ for Fx-deconvolution, DnCNN, and domain-adapted DnCNN denoiser, respectively. The results show that the domain-adapted DnCNN outperforms the other two denoisers. The method is also successfully applied to real data. The real data shown in Figure 3a is the stack section of processed land data provided by Geofizyka Torun Sp. Z.o.o, Poland. We used Madagascar open-source software package (Fomel et al., 2013), to process the data. The denoising results after applications of Fx-deconvolution, DnCNN, and domain-adapted

DnCNN denoisers are depicted in Figures 3c- 3e, respectively. In real data, we do not have access to the ground-truth result, so we only show the estimated noise panels (Figure 4). The noise panels show the signal leakage for the Fx-deconvolution and DnCNN denoiser, however, those leakages are significantly reduced after using the domain-adapted DnCNN method.

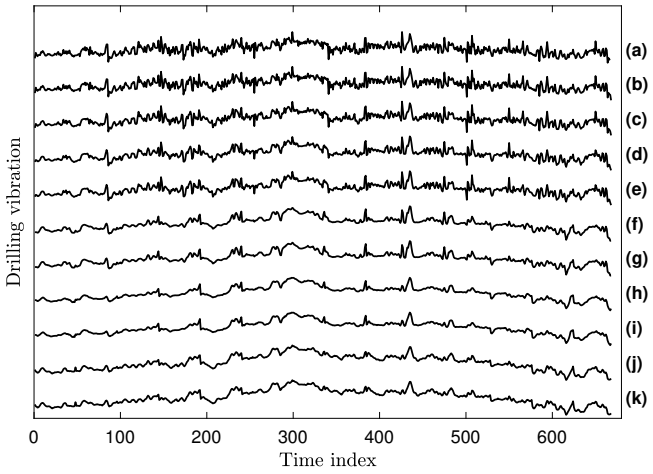

Figure 5 *Performances of domain-adapted DnCNN on the first drilling dataset. (a) Recorded accelerometer maximum time series. (b)-(k) Noise-suppressed traces by using domain-adapted DnCNN on (a) with standard deviations of 10, 20, 30,..., 90, and 100, respectively.*

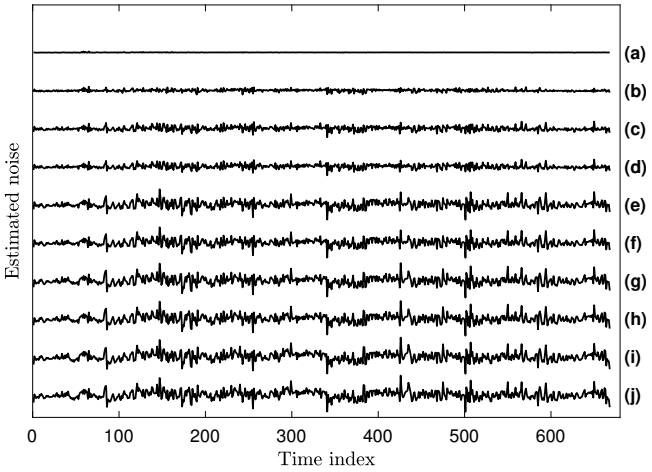

Figure 6 *Noise estimated traces of first dataset. (a)- (j) are the noise estimated by subtracting Figures 5b- 5k from Figure 5a, respectively.*

Drilling noise suppression

To evaluate the performances of the denoising operators on the drilling dataset, we use data from three-axis accelerometers from near the bit recording during drilling (we report the results on two different field trails). The accelerometers are recording the vibrations of the bottom-hole assembly (BHA) in axial and two transverse directions. Drill bit-rock interactions crush the rocks while drilling and this process generate significant elastic energy that propagates along the drill string. Since the drilling is continuously happening, recording data with a high sampling rate is not a trivial task. So, the receivers only report the maximum, minimum, mean, and standard deviation of data for a given period. The drilling data are corrupted by dynamic effects of the drillstring and by surface noise as well as degradation as it travels along the drill string. To reduce the noise on accelerometer data we only use time series corresponding to maximum and minimum values in each receiver. Since there are three accelerometers, we end up having six traces. Before we apply our domain-adapted DnCNN to the data, we remove the mean from each trace and normalize the traces so that they show similar statistics. After doing so, we further divide the time series into several windows and reshape the data to have several traces. For example, in the first dataset, each trace has 672 time samples and there are six traces in total. Hence, after removing the mean and normalizing the traces, we divide traces into eight time windows each with 84 time samples. Then, we end up having 48 traces each with 84 time samples. Then, the domain-adapted DnCNN is applied on this dataset with a 32×32 batch size. Figure 5a shows the recorded maximum trace of the first accelerometer. After applying the domain-adapted DnCNN operators with DnCNN operators that are trained on different ranges of standard deviation on noise, we depict the noise suppressed results in Figures 5b- 5k. The standard

deviations are 10, 20, 30,..., 90, and 100 for Figures 5b- 5k, respectively. Since we do not have access to the ground-truth result, we only show the estimated noise panels (Figure 6). From noise panels, we conclude that the result of denoising with a standard deviation of 40 is optimal. The noise estimation with a standard deviation smaller than 40 results in under-estimation of the noise and with standard deviation larger than 40 results in over-estimation of the noise.

Figure 7 shows the denoising results with a standard deviation of 40 on the maximum traces of three accelerometers. The method successfully suppresses noise on all three traces. Each panel from bottom to top shows original, noise suppressed and estimated noise traces. Later, we apply the domain-adapted DnCNN denoiser with a standard deviation of 40 on the second dataset. In this dataset, we have 4000 time samples and six traces corresponding to the minimum and maximum values on three accelerometers. After removing mean, normalizing, and windowing traces, we end up having 48 traces each with 500 time samples. Then, the domain-adapted DnCNN is applied on this dataset with a 32×32 batch size. The noise suppressed results similar to that of Figure 7 are shown in Figure 8. The results indicate that the domain-adapted DnCNN can be applied on a different dataset in a specific domain, after adaptation, without modifying its training parameters or training from scratch. Through domain adaptation, the new operator now is generalized.

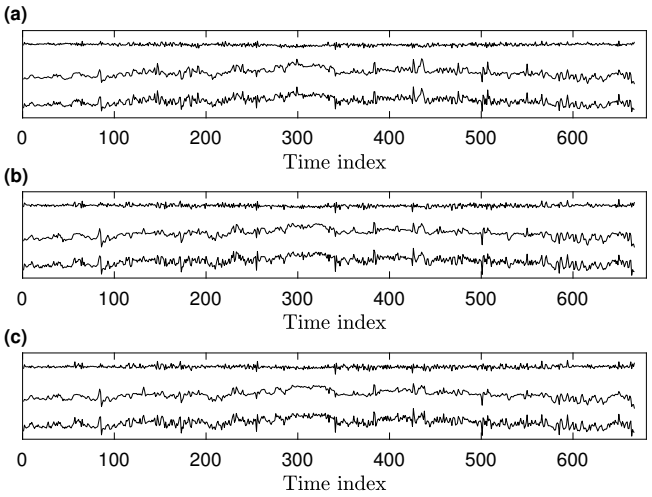

Figure 7 *The performances of the domain-adapted DnCNN denoiser with a standard deviation of 40 on the first dataset. (a)-(c) From bottom to top are the recorded, noise-suppressed and estimated noise traces for the first, second and third accelerometer maximum rcordings, respectively.*

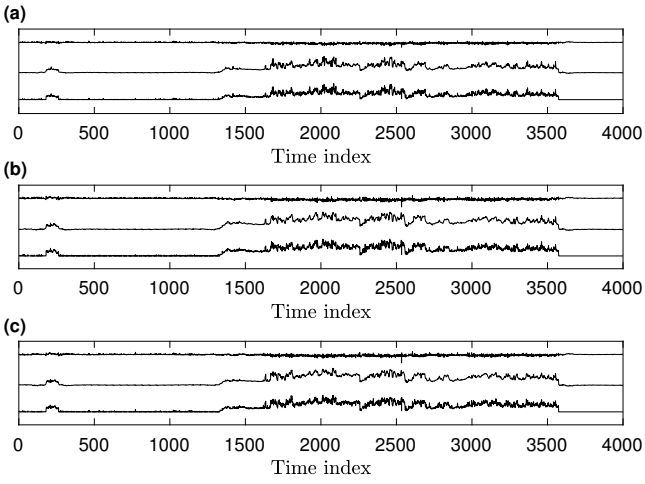

Figure 8 *The performances of the domain-adapted DnCNN denoiser with a standard deviation of 40 on the second dataset. (a)-(c) From bottom to top are the recorded, noise-suppressed and estimated noise traces for the first, second and third accelerometer maximum rcordings, respectively.*

Conclusions

We developed a workflow to adapt the natural-image learned DnCNN denoiser for suppressing noise in seismic and drilling datasets. The workflow is validated on synthetic data and then applied on field seismic and drilling datasets. To adapt the original DnCNN denoiser operator to a new domain, we looked into the statistical differences between the original and target domains. The numerical examples show that the workflow is efficient and generalizable. However, to automate the process, developing a approximate noise level estimation workflow is necessary.

Acknowledgment

This work is partially supported by the University of Calgary's Canada First Research Excellence Fund Program, the Global Research Initiative in Sustainable Low Carbon Unconventional Resources. Seismic data is publicly available from http://www.freeusp.org/RaceCarWebsite/TechTransfer/Tutorials/Processing_2D.

References

Abma, R. and J. Claerbout, 1995, Lateral prediction for noise attenuation by tx and fx techniques: Geophysics, **60**, 1887–1896.
Bonar, D. and M. Sacchi, 2012, Denoising seismic data using the nonlocal means algorithm: Geophysics, **77**, A5–A8.
Brandon, T., M. Mintchev, H. Tabler, et al., 1999, Adaptive compensation of the mud pump noise in a measurement-while-drilling system: SPE Journal, **4**, 128–133.
Buades, A., B. Coll, and J.-M. Morel, 2005, A review of image denoising algorithms, with a new one: Multiscale Modeling & Simulation, **4**, 490–530.
——— 2010, Image denoising methods. a new nonlocal principle: SIAM review, **52**, 113–147.
Canales, L. L., 1984, Random noise reduction, *in* SEG Technical Program Expanded Abstracts 1984, 525–527.
Chen, Y. and S. Fomel, 2015, Random noise attenuation using local signal-and-noise orthogonalization: Geophysics, **80**, WD1–WD9.
Coupé, P., P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, 2008, An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images: IEEE transactions on medical imaging, **27**, 425–441.
Deledalle, C.-A., L. Denis, and F. Tupin, 2010, Nl-insar: Nonlocal interferogram estimation: IEEE Transactions on Geoscience and Remote Sensing, **49**, 1441–1452.
Fomel, S., P. Sava, I. Vlad, Y. Liu, and V. Bashkardin, 2013, Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments: Journal of Open Research Software, **1**.
He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.
Huang, J., J. Ma, N. Liu, H. Zhang, Z. Bian, Y. Feng, Q. Feng, and W. Chen, 2011, Sparse angular ct reconstruction using non-local means based iterative-correction pocs: Computers in Biology and Medicine, **41**, 195–205.
Ioffe, S. and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: arXiv preprint arXiv:1502.03167.
Kazemi, N., 2020, Across-domains transferability of deep-red in de-noising and compressive sensing recovery of seismic data: arXiv preprint arXiv:2007.10250.
Kingma, D. P. and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv preprint arXiv:1412.6980.
Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, 1097–1105.
Naghizadeh, M. and M. Sacchi, 2012, Multicomponent f-x seismic random noise attenuation via vector autoregressive operators: Geophysics, **77**, V91–V99.
Oropeza, V. and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, V25–V32.
Richardson, A. and C. Feller, 2019, Seismic data denoising and deblending using deep learning: arXiv preprint arXiv:1907.01497.
Sinanovic, S., D. H. Johnson, V. V. Shah, and W. R. Gardner, 2004, Data communication along the drill string using acoustic waves: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, iv–909.
Trickett, S., 2008, F-xy cadzow noise suppression, *in* SEG Technical Program Expanded Abstracts 2008, 2586–2590. Society of Exploration Geophysicists.
Wei, D.-Y. and C.-C. Yin, 2010, An optimized locally adaptive non-local means denoising filter for cryo-electron microscopy data: Journal of structural biology, **172**, 211–218.
Yu, S., J. Ma, and W. Wang, 2019, Deep learning for denoising: Geophysics, **84**, V333–V350.
Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang, 2017, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising: IEEE Transactions on Image Processing, **26**, 3142–3155.
Zhu, W., S. M. Mousavi, and G. C. Beroza, 2019, Seismic signal denoising and decomposition using deep neural networks: IEEE Transactions on Geoscience and Remote Sensing, **57**, 9476–9488.