SNR enhancement for Brillouin distributed optical fiber sensors based on asynchronous control

PEIXUAN ZHANG,[1,2,3] BIWEI WANG,[4] YANAN YANG,[1,2] ABUL KALAM AZAD,[5] KUO LUO,[1,2] KUANGLU YU,[1,2,*] CHANGYUAN YU,[4] AND CHAO LU[4]

[1]*Institute of Information Science, School of Computer and Information Technology, Beijing Jiaotong University, 100044, Beijing, China*
[2]*Beijing Key Laboratory of Advanced Information Science & Network Technology, Beijing 100044, China*
[3]*Weihai Campus, Beijing Jiaotong University, Weihai 264401, China*
[4]*Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
[5]*Department of Electrical and Electronic Engineering, University of Dhaka, Dhaka-1000, Bangladesh*
[*]*klyu@bjtu.edu.cn*

Abstract: We propose the asynchronous control of anisotropic diffusion (AD) algorithm, and such asynchronous anisotropic diffusion (AAD) algorithm is demonstrated experimentally to reduce noise from the sensing signals obtained from Brillouin distributed optical fiber sensors. The performance of the proposed AAD algorithm is analyzed in detail for different experimental conditions and compared with that of block-matching and 3D filtering, two-dimensional wavelet denoising, AD, and non-local means algorithms. Some key factors of the proposed algorithm, such as the impact of convolution kernel size on the performance of AD algorithms, the influence of low sampling point number (SPN) on the quality of Brillouin frequency shift and the selection of diffusion thresholds are analyzed and discussed with experimental results. The experimental results validate that the AAD algorithm can provide better root-mean-square error (RMSE) and spatial resolution (SR) than the other four algorithms, especially for higher signal-to-noise ratio (SNR) improvement and higher SPNs. For lower SPNs, the performance of AAD is also not inferior to the RMSE performance of NLM and AD. The runtime of the AAD algorithm is also quite low. Moreover, the proposed algorithm offers the best SR performance as compared to other noise reduction algorithms investigated in this study. Thus, the proposed AAD algorithm can be an effective candidate to improve the measurement accuracy of Brillouin distributed optical fiber sensors.

1. Introduction

Brillouin distributed optical fiber sensing technology has become a hot spot in the research of optical fiber sensors in recent years [1–5]. Such technology has found many applications in long-distance measurements of distributed temperature and strain with good accuracy [6–10]. The technology is based on the principle that strain and temperature changes are linearly related to Brillouin frequency shift (BFS) of the local Brillouin gain spectrum (BGS) along the fiber. In order to obtain BGSs with sufficiently high signal-to-noise ratio (SNR), the Brillouin distributed optical fiber sensing technology demands the sampling of sensing signals multiple times and averaging of such sensing signals [11]. This process takes a long time to complete the measurement of BGSs along the fiber, which is not suitable for dynamic measurement [12]. Consequently, the use of suitable signal denoising algorithms to enhance the SNR of the BGSs along the fiber has

become one of the key issues in improving the performance and broadening the applications of Brillouin distributed optical fiber sensors.

The BGSs along the fiber as a whole can be regarded as an image. Thus, noise reduction algorithms commonly applied in image and video processing have been used recently to reduce noise from the BGSs. Farahani et al. [13] applied one-dimensional wavelet transform for reducing the noise of the BGSs obtained from Brillouin optical time domain analysis (BOTDA) sensor. The combination of ensemble averaging and two different wavelet shrinkage methods in such study can shorten the sensor's measurement time by up to 90%. Soto et al. [14] claimed to use two-dimensional wavelet denoising (2D-WD) and non-local means (NLM) based 2D image restoration algorithms to enhance the response of distributed optical fiber sensors, and to significantly improve the SNR performance up to 20 dB without the assist of additional hardware. Wu et al. [15] proposed and successfully demonstrated the use of advanced block-matching and 3D filtering (BM3D) based denoising techniques for BOTDA sensor to achieve high SNR improvement even in the case of small sampling point number (SPN) with least degradation of measurement accuracy and experimental spatial resolution. Wang et al. [16] first employed video block-matching and 3D filtering (VBM3D) based denoising algorithm in a 100.8 km long-distance BOTDA sensing system with a spatial resolution of 2 m, and achieved a temperature uncertainty of 0.43 °C. However, such algorithm suffers from high computational complexity. Luo et al. [17] also proposed the use of the classic Perona-Malik (AD algorithm in this paper) model [18], which is one of the anisotropic diffusion (AD) algorithms, to reduce the noise of BGS image, and demonstrated the superiority of this time efficient algorithm in preserving the experimental spatial resolution. However, such traditional AD algorithm has shown relatively poor root-mean-square error (RMSE) performance in BOTDA noise reduction.

In order to demonstrate the performance of asynchronous control (AC) algorithms, we propose an algorithm that combines AC algorithm with AD algorithm as an example, which is termed as asynchronous anisotropic diffusion (AAD) in this paper. The proposed algorithm of AAD provides better performance through AC to improve the SNR of the BGSs, while suppressing the over-smooth situation and maintaining the frequency shift edge(s).

The contents of this paper are divided into three parts. The 'Algorithms' part includes the principle and the execution process of AAD. In the 'Experiments' part, the performances of five algorithms namely 2D-WD, NLM, BM3D, AD, and AAD are compared in terms of RMSE and spatial resolution (SR) with the SNR increased by 12 dB to 20 dB (step of 2 dB) under the SPNs of 2 to 10 (step of 2 points). Experimental results show that under high SNR improvement and high SPNs, the RMSE performance and SR of AAD are both the best among the five algorithms, while under low SPNs, RMSE of AAD is better than or similar to NLM and AD and the best spatial resolution is still guaranteed. In the 'Discussion' part, we first discuss the impacts of different sizes of convolution kernels on the performance of AD algorithms. We then study the effect of low SPN on the acquisition of BFS. We also find that under low SNR BGS samples, the anisotropic diffusion threshold of AD algorithms shows the robustness. Zaslawski et al. conducted preliminary work comparing 1-D and 2-D algorithms using Gaussian filter and NLM [19], and we have further investigated their performance on SR. In summary, AAD algorithm is a competitive candidate for noise reduction in Brillouin and other distributed optical fiber sensing systems.

2. Algorithms

In this part, we first introduce the principles of anisotropic diffusion algorithm [17], Catté_PM [20], and asynchronous control. Then, we illustrate AAD algorithm considering the requirements for BGS images to suppress local over-smooth and to enhance edge retention.

2.1.   *Anisotropic diffusion algorithms*

2.1.1.   Perona_Malik model

The anisotropic diffusion (AD) methods based on partial differential equations has been applied to many applications including image denoising [21], edge detection [22], image segmentation [23], image enhancement [24] and other fields. The advantage of using AD is that it can suppress noises, while maintaining or even enhancing the edges of the image. The most classic of these is the model proposed by Perona and Malik [18]:

$$
\begin{cases}
\frac{\partial I}{\partial t} = div[g(||\nabla I||) \cdot \nabla I] \\
\qquad I(t = 0) = I_0
\end{cases}, \tag{1}
$$

where I is the input image, ∇ is the gradient operator, $g(||\nabla I||)$ is the diffuse function [18], which can be expressed as:

$$
g(||\nabla I||) = \frac{1}{1 + \left(\frac{||\nabla I||}{\kappa}\right)^2}, \tag{2}
$$

where κ is the gradient threshold. In our application,

Equation (1) needs to be discretized with brightness values associated with the Brillouin gain spectrum. For this, 8-direction-neighbors discretization of Eq. (1) is employed in our case, which can be given as:

$$
I^{t+1} = I^t + \lambda
\begin{bmatrix}
g_N \cdot \nabla_N I + g_S \cdot \nabla_S I \\
+ g_E \cdot \nabla_E I + g_W \cdot \nabla_W I \\
+ g_{NW} \cdot \nabla_{NW} I + g_{NE} \cdot \nabla_{NE} I \\
+ g_{SE} \cdot \nabla_{SE} I + g_{SW} \cdot \nabla_{SW} I
\end{bmatrix}. \tag{3}
$$

2.1.2.   Catté_PM model

Aiming at the shortcomings of the PM model to strong noise and the ill-posed problem, Catté et al. proposed an improved PM model, referred to as the Catté_PM model [20], in which the gradient modulus of image is replaced by the gradient modulus after Gaussian smoothing of the original image in order to obtain the diffusion weight, due to the fact that Gaussian smoothing can effectively suppress Gaussian noise.

Given that additive noises, such as Gaussian white noise σ, are dominant in BGS [25], the image degradation model of BGS can be expressed as Eq.(4), where BGS_0 is the Brillouin gain spectrum without noise in ideal condition and BGS is the noisy Brillouin gain spectrum in experimental conditions.

$$
BGS = BGS_0 + \sigma. \tag{4}
$$

The gradient modulus calculated by Gaussian smoothed image can better reflect the change of the edge of BGS_0. Therefore, we use Catté_PM to suppress Gaussian noise in BGS. Catté_PM algorithm is expressed as

$$
\begin{cases}
\frac{\partial I}{\partial t} = div\left[g\left(\left|\left|\nabla\left(G_\sigma * I\right)\right|\right|\right) \cdot \nabla I\right] \\
\qquad I(t = 0) = I_0
\end{cases}, \tag{5}
$$

where G_σ is the two-dimensional Gaussian function with variance σ. In these algorithms, the flux function is used to measure the intensity of diffusion [18, 26], which can be described as:

$$
\phi = \frac{||\nabla I||}{g(||\nabla I||)}. \tag{6}
$$

Figure 1 shows the schematic diagram of the flux function. It can be known from Fig. 1(b) that the flat area (I) will be significantly smoothed, and the part with edges (II) will be preserved. When the noise level is equal to the diffusion threshold κ, the smoothing intensity is the largest and the noise is under the best suppression. Therefore, κ should theoretically be equal to the noise standard deviation σ. However, we found in actual experiments that when $\kappa \geq \sigma$ with the other parameters unchanged, the noise reduction effect does not change significantly. The threshold will be further described in the discussion section.
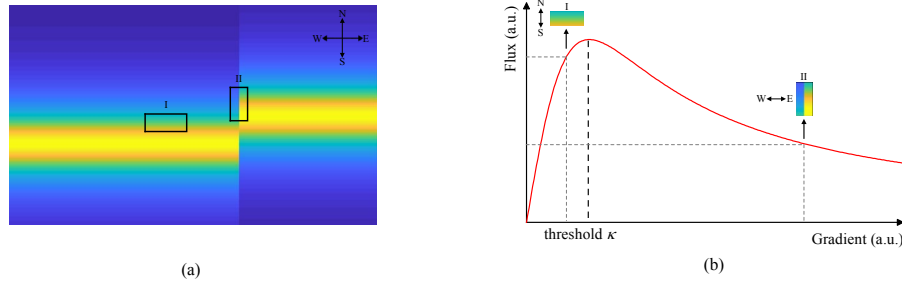

Fig. 1. Demonstration of the schematic diagram of the flux function. (a) BGSs data with frequency shift; (b) the flux function for different areas of a BGS image.

2.2. Asynchronous control

In the study, we found that the algorithm can be iterated and after each iteration, the data that meets certain conditions are specially processed. This kind of operation will interfere with the natural evolution of data. As a result, the final result is often very different from the result obtained by allowing its natural evolution. Because the data that only meets the conditions are processed, we call it asynchronous control (AC).

In this study, the AC algorithm is used in 'Local Over-smoothing Suppression' and 'Edge Retention' of the AAD algorithm. And different from Catté_PM, the size of the convolution kernel for gradient calculation in AAD can be adjusted by parameters. Different convolution kernels have an impact on the runtime and noise reduction effect, which is briefly discussed later.

2.2.1. Local over-smoothing suppression

From Eq. (5), all points in Catté_PM algorithm participate in the calculation at each iteration. This means that the diffusion time of each point is completely consistent. From the flux function in Fig. 1(b), the smoothing intensity before threshold κ rises fast. So even if the neighboring point has small fluctuations, it will participate in the anisotropic diffusion operation to be smoothed. This will lead to over-smoothing. To suppress this, the AAD algorithm controls the diffusion of each point in each direction through the AC algorithm.

As shown in Fig. 2, the control method first calculates the gradient modulus absolute value of a point within a certain range, and the sum of absolute values called cumulative gradient modulus is used to represent the flatness (fluctuation) of the point in each direction. Then the accumulated gradient modulus of the entire BGS is normalized to form a normalized cumulative gradient modulus matrix in each direction in the same size of BGS image. In other words, scale its value range to between 0 and 1. The threshold θ_{N0} (in Fig. 2) is used to filter the normalized cumulative gradient modulus to find out the points whose surrounding points are relatively flat, i.e., the points of normalized cumulative gradient modulus in $[0, \theta_{N0}]$ do not participate in the anisotropic diffusion operation. This process is equivalent to a low pass filter. Finally, the control matrix M_N (a Boolean matrix, the points that in $[0, \theta_{N0}]$ is 0, the points that in $(\theta_{N0}, 1]$ is 1) is

formed. Through the AC algorithm, the number of iterations to achieve the same SNR increases. However, the advantage is obvious as the overall noise level of the BGS can be uniformed, and the RMSE under the same SNR can be significantly reduced.
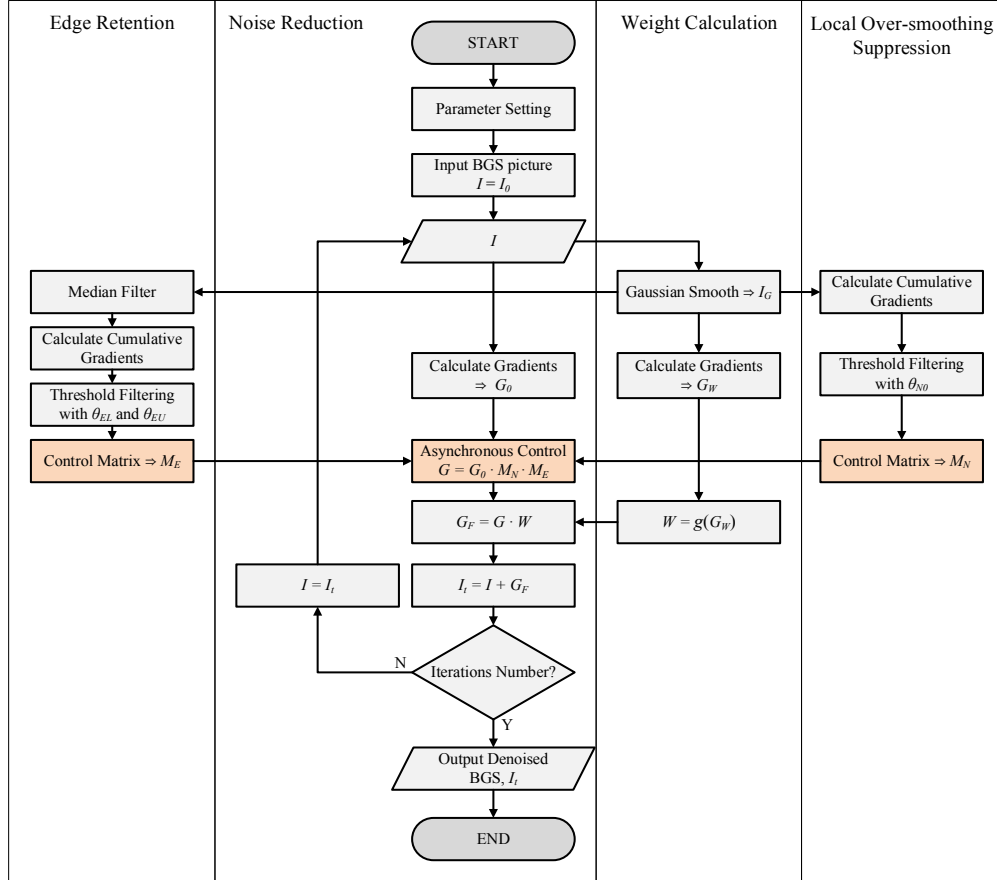

Fig. 2. The flow chart of AAD algorithm.

2.2.2.   Edge retention

Since the edge of BGS image is not a single-pixel edge, traditional edge detection algorithms such as Laplace operator [27], Sobel operator [28] are not effective in detecting BGS edges. The traditional AD algorithm cannot effectively distinguish between the edge and the noise if the sudden change of the edge gradient is submerged in the noise of BGS. Therefore, for a low SNR data, the result of diffusion may not effectively remove the noise, and it might enhance the noise. Considering the edge feature of BGS is step-like, the change of gradient before and after the edge is uniform along the frequency direction. For the case where temperature varies slowly in space, the section can be regarded as a flat area, which is processed by the local over-smooth suppression mechanism. We also use the AC algorithm in order to keep the edges. In general, the detection range of the accumulated gradient modulus of edge retention is similar to the detection range of the accumulated gradient modulus of local over-smoothing suppression. However, the range can also be set manually. In this work, by setting the lower and upper thresholds θ_{EL} and θ_{EU} in, the

direction of the point where the normalized cumulative gradient modulus falls in $[\theta_{EL}, \theta_{EU}]$ does not participate in current anisotropic diffusion operation. So, it is equivalent to a band pass filter.

2.3. Asynchronous anisotropic diffusion (AAD) algorithm

The operation of AAD algorithm is described with the help of the flowchart shown in Fig. 2. The algorithm was designed to automatically adjust the parameters based on the data to be denoised I with few manually set parameters, such as the number of iterations, the convolution kernel size, the over-smoothing threshold θ_{N0}, the edge retention thresholds θ_{EL} and θ_{EU}. The Gaussian smoothing is first performed by applying median filter on BGS image to generate I_G, which is used to calculate the diffusion weight W and the cumulative gradient modulus for local over-smooth suppression. After median filtering, I_G is used to calculate the cumulative gradient modulus of edge retention. Two cumulative gradient modulus matrices are filtered by the threshold to form two control matrices M_N and M_E. At the same time, the gradient modulus G_0 is obtained through convolution with I. The step involved in AC ($M_N.M_E.G_0 = G$) is highlighted in the flowchart shown in Fig. 2. Then, the dot product of the gradient modulus G_F could be obtained after AC and the diffusion weight ($G \cdot W$) can be calculated. Finally, G_F is added on I to obtain the diffused image I_t for next iteration. If the number of iterations is not reached, the AAD operations (all operations above) will be performed on I_t, whilst if the number of iterations is reached, I_t will be the output. The formulae of AAD are as follows:

$$
\begin{cases}
\frac{\partial I}{\partial t} = div[g(||\nabla I_G||) \cdot \nabla I] \\
I = I_0 \qquad\qquad\qquad\qquad\qquad\quad t = 0 \\
||\nabla I|| = 0 \qquad\qquad (\{\theta_N < \theta_{N0}\} \cup \{\theta_{EL} < \theta_E < \theta_{EU}\})
\end{cases}
, \qquad (7)
$$

where t is the number of iterations, θ_N and θ_E represents the normalized cumulative gradient modulus of local over-smoothing suppression and edge retention, respectively. θ_{N0} is the over-smoothing suppression threshold, θ_{EL} and θ_{EU} are respectively the lower and upper edge retention thresholds. The result of discretization of Eq. (7) is the same as in Eq. (3). The physical meaning and optimization principles of key parameters of AAD are listed in Table 1.

Table 1. The meaning and optimization principles of key parameters of AAD

Parameter	Notation	Range	Physical meaning	Optimization principle
Number of iterations	t	$t > 0$ (integer)	The total number of algorithm iterations ($\geq$ the actual number of iterations at each position).	Adjust it until the desired SNR improvement is obtained.
Over-smoothing suppression threshold	θ_{N0}	$0 < \theta_{N0} < 1$	For removing the part of normalized cumulative gradient modulus $< \theta_{N0}$ from diffusion operation.	Around 0.1, depends on the noise reduction effect.
Upper and lower Edge retention thresholds	θ_{EL}, θ_{EU}	$0 < \theta_{EL} < \theta_{EU} < 1$	For removing the part of $\theta_{EL} <$ normalized cumulative gradient modulus $< \theta_{EU}$ from diffusion operation.	Around 0.6 and 0.9, respectively, depends on the noise reduction effect.
Convolution kernel size	S	$S \geq 3$ (integer)	The size of the convolution kernel for anisotropic diffusion.	Typically, 3 or 5. See the discussion part.

3.   Experiments

3.1.   Experimental setup and parameters

The setup of BOTDA system used to collect data in this experiment is the same as used by Luo et al. [17]. The total length of the optical fiber used in the setup is 99 km. In the experiment, the width of pump-pulse is adopted to be 20 ns. We set the starting point at 500 m from the end of the optical fiber and the BGSs along the last 500 m (i.e., 98.5 km to 99 km) have been used as raw experimental data. Because this section would be the lowest SNR part (due to attenuation and accumulated noises) which is most difficult to be processed. And the data along the first 98 km doesn't contain any hot spots or sections, while our main target was also to examine the denoising performance of various algorithms at the end of a ~ 100 km fiber.

The improvement of SNR and SPN are two important parameters which affect the performance of noise reduction algorithms [15]. The SPN of the experimental raw data is 10 points/2-meters. We resample such raw data using piecewise linear method to generate four other SPNs of 2, 4, 6, 8 points/2-meters. Consequently, a total of 5 different SPNs are formed with the original data. The SNR of raw data is < 0 dB, and the SNR improvement varies from 12 dB to 20 dB at step of 2 dB. In this work, the SNR is calculated using the method in Ref. [29]. We applied the controlled variable method to compare the noise reduction effects of different SPNs under the same SNR improvement and the noise reduction effects of different SNR improvements under the same SPN, respectively.

The RMSE, SR and average runtime are calculated to indicate the denoising performance of different algorithms. Here, we calculate the RMSE by comparing the actual temperature (60.3°C) of the last 192 m (from the part of the last 500 m) heating section with the measured temperature extracted by Lorentzian curve fitting (LCF) [30]. The spatial resolution is computed as the length from 10% to 90% of the rising edge [15], [31]. Lastly, the average runtime of an algorithm, which is used to quantify the algorithm's computational complexity, is calculated to be the average time taken by the algorithm from the beginning to the end of data processing under the same conditions on a same machine.

We use and compare the performances of different classic and most recent noise reduction algorithms in processing BGSs that includes Gaussian filtering, 2D-WD (referred to as WD in this paper), NLM, BM3D and AD algorithms. Among these algorithms, the first two have been widely used for quite a long time but the excellent performances of NLM, BM3D and AD in BGS noise reduction have been reported in recent years.

3.2.   Experimental results

In this section, we process data from two different BOTDA setups to testify the robust and the generalized performance of the AAD algorithm. Firstly, AAD is used to denoise the raw BGS data for the whole 99 km fiber, the raw BGS, denoised BGS with AAD and the BFS obtained from them are demonstrated in Fig. 3. Then, each algorithm is used to denoise the raw BGS data for the fiber section from 98.75 km to 98.85 km of the fiber. And the raw and AAD denoised BGS are demonstrated in Fig. 4.

Then the noise reduction effects of each algorithm are calculated, analyzed and compared. The BFSs of the original or denoised BGSs are determined by employing LCF. The BFSs fitted after applying noise reduction algorithms of each of WD, BM3D, NLM, and AAD are shown in Fig. 5 along with the BFS fitted without applying noise reduction algorithm as well as the BFS fitted after applying AD noise reduction algorithm.

In the peak (I) region, the curves for using WD and BM3D are smoother than AD in Fig. 5 (a), (b). The BFS curve obtained after applying NLM noise reduction is not as smooth as that of AD, which means that the RMSE for using NLM is larger than that of AD in Fig. 5 (c). The peak of the BFS for using AAD in Fig. 5(d) at peak I is slightly lower than that of AD and NLM.
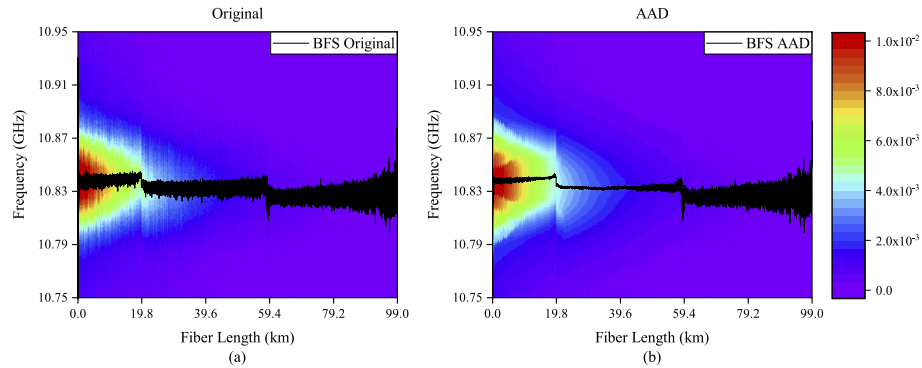
Fig. 3. Measured raw BGS (a) and the AAD denoised BGS (b) with 20 dB SNR improvement and their corresponding BFS along the FUT.
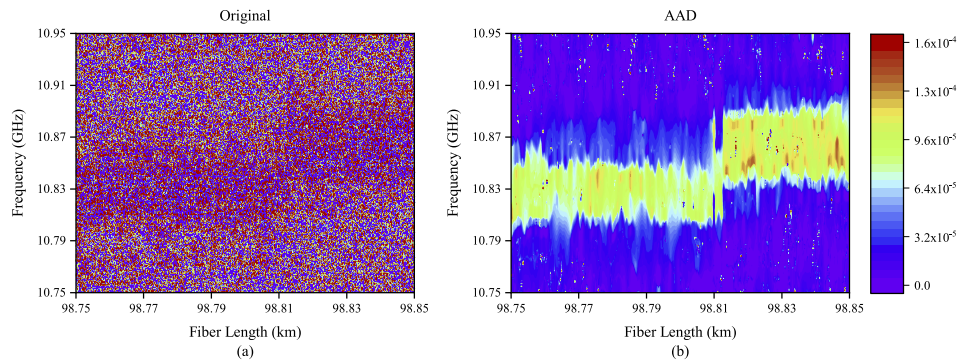

Fig. 4. Measured raw BGS (a) and the AAD Denoised BGS (b) with 20 dB SNR improvement from 98.75 km to 98.85 km of a 99 km section along FUT.

However, the leading edge of its peak I rises later than the others. In the frequency shift (II) region, the lower edge of the frequency step shift for using WD and BM3D rises too early in Fig. 5(a), (b), which will make its spatial resolution far inferior to AD and make the obtained frequency shift position inaccurate. The curve for using NLM in Fig. 5(c) is similar to that of AD at peak I and the frequency shift at section 2 is also similar. However, this unsmooth BFS curve affects the spatial resolution performance and judgment of frequency shift position. At the frequency shift of AAD in Fig. 5(d), the lower edge rises lately among the four algorithms. In addition, the height of the frequency shift (II) is slightly higher than AD, which makes its spatial resolution higher than those of the other algorithms.

Secondly, another BGS data along a 40.05 km-long fiber acquired with 40 ns pump pulse is then processed, where the last 50 meters of the fiber was water-bathed at 40°C, 50°C, and 60°C, respectively. The BFSs along the last 250 m are plotted in Fig. 6. The SNR of the three raw BGS images are 1.5 dB, and the SNR improvement by these five algorithms are 15 dB. As shown in the previous experiment, AAD has attained the best spatial resolution in all cases in these algorithm comparison experiments, which will be discussed later in the 'Spatial Resolution' section. In addition, the AAD curve has the highest rise height on the upper edge of the frequency shift in all cases. Moreover, the BFS curve obtained with AAD is relatively smooth among the five algorithms and there is no dropping in the middle of the frequency shift as in BM3D.
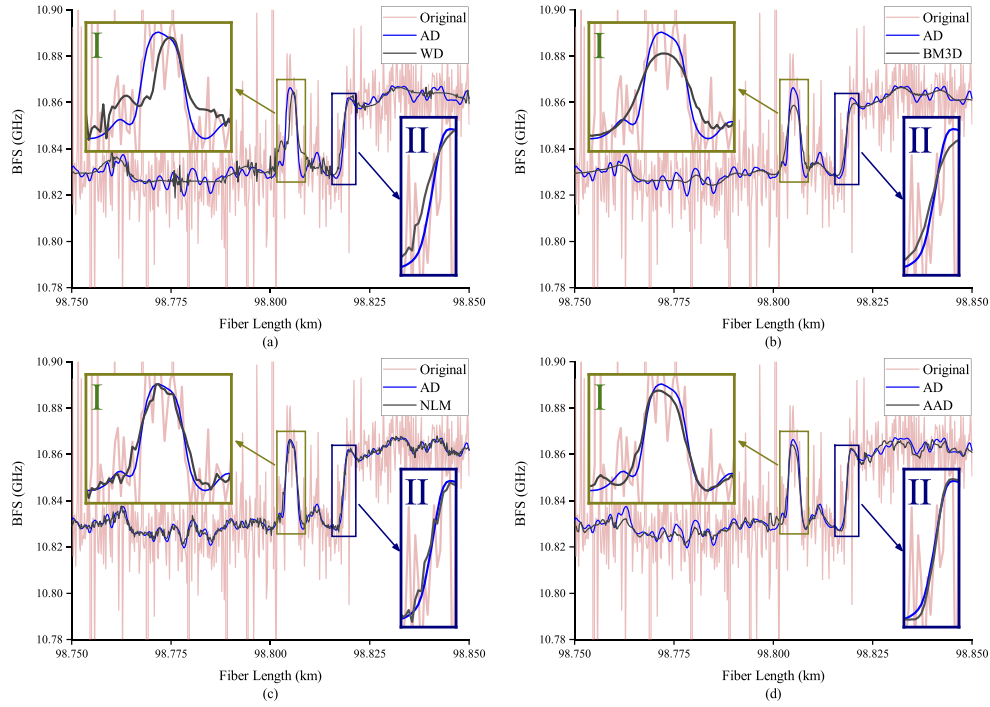
Fig. 5. Comparison of the origin BFS, AD denoised BFS and the BFSs after (a) WD, (b) BM3D, (c) NLM, and (d) AAD de-noising algorithms along the fiber section from 98.75 km to 98.85 km.
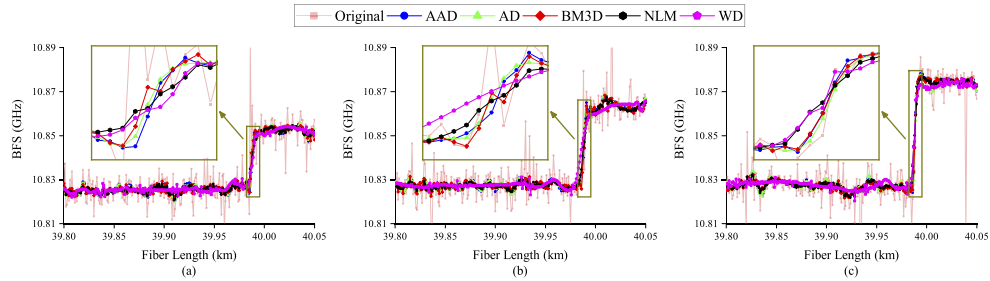

Fig. 6. Denoised BFSs along the section starting from 39.80 km to 40.05 km of a second 40.05 km fiber after applying different noise reduction algorithms. The last 50 m section of the fiber is heated at (a) 40℃, (b) 50℃, and (c) 60℃. The inset shows the frequency shift section.

3.3. Performance comparison

The performances of different noise reduction algorithms are compared in terms of RMSE, SR and average runtime. The RMSEs of BFSs along the 98.83 km to 98.85 km hot section after applying noise reduction algorithms are shown in Fig. 7 for five different SPNs. The results in Fig. 7 show that the RMSE performance of WD under different SPNs is not stable and the RMSEs are higher for SPNs of 4 and 8 but lower for SPNs of 2, 6 and 10. However, the trend of RMSEs for using WD under the same SPN is relatively smooth. The BM3D algorithm performs almost the best among the five algorithms. However, the non-monotonicity occurs for such algorithm as can be observed easily in Fig. 7(d) and (e). The instability of these two algorithms

(i.e., WD and BM3D) is caused by the resampling process used to obtain different SPNs. The NLM algorithm has a mediocre performance in RMSEs which is the most stable among these five algorithms even if the improvement of SNR is decreased. Although the trend of RMSE of AD is stable, the RMSEs are the largest for each of the five different SNR improvement used in this study. The RMSEs for using AAD under all SPNs and SNR improvements are much smaller than that of AD and the change in RMSE for using AAD increases a little if higher SPNs are used. It is also observed in Fig. 7 that the RMSEs for using AAD are comparable to that for using NLM at lower SPNs. However, the AAD outperforms NLM for higher SPNs irrespective to the SNR improvement. For instance, the RMSEs for using AAD at SPNs of 10 and 8 are significantly smaller than that for using NLM. Similarly, the RMSEs for using AAD at higher SNR improvement are also superior to that for using NLM at lower SPNs. It is further observed in Fig. 7 that the RMSE performance of AAD is slightly worse than that of BM3D, especially at lower SPNs as well as lower SNR improvement. However, at higher SPNs (8 and 10) and higher SNR improvement (> 16 dB) the performance of AAD is comparable to that of BM3D. The overall RMSE performance of AAD under a particular SPN is relatively smooth and this algorithm performs better for higher SNR improvements.
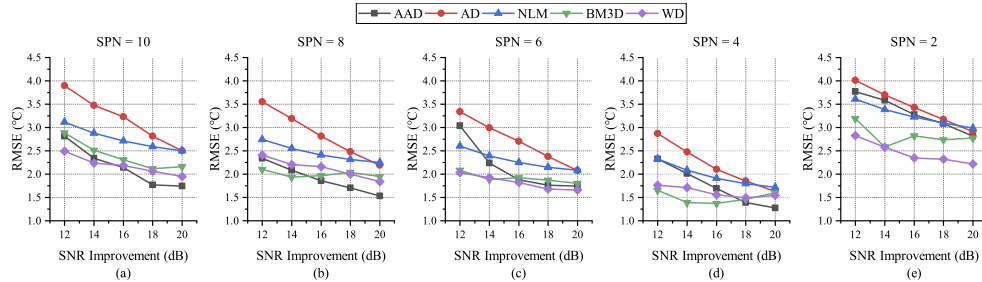

Fig. 7. RMSE of BFSs after applying AAD, AD, NLM, BM3D and WD under different SPNs.

We then looked into the performances of different algorithms in terms of SR. The SR employing all algorithms at five different SPNs and five different SNR improvements are shown in Fig. 8. In almost all the cases, the SR performance of WD is the worst among all five while the results of NLM and BM3D are much worse than those of AD and AAD, especially for higher SPNs. The SR performances of WD and BM3D in Fig. 8 are also not stable compared to that of NLM, AD and AAD for all tested cases. As illustrated in Fig. 8 (e) that the performance of all algorithms is seriously affected by lower SPNs. The SRs for AD and AAD at all SPN and SNR improvement are very stable and almost unaffected, and they are significantly smaller than other methods. For instance, the SRs for using WD, NLM, BM3D, AD and AAD in Fig. 8(b) are 2.51 m, 2.00 m, 2.03 m, 1.27 m and 1.24 m respectively for a SPN of 8 and SNR improvement of 16 dB. Consequently, the SR performance of WD, NLM, BM3D and AD are 2.02, 1.61, 1.64 and 1.02 times worse than that of AAD. These ratios also increase if higher SPNs are applied. It is worth mentioning that the selected lengths of SRs presented here are slightly shorter than 10%-90% of the rising edge as a result of keeping the length of the straight-line fitting under different SPNs consistent. However, such results can well demonstrate the trend of SR with different SPN.

In addition, we also computed SRs at different temperatures for denoised BFSs shown in Fig. 6, where the SPN is 4 and SNR improvement is 15 dB. The results are shown in Table 2. The large spatial resolution at 50°C is due to the slight deviation of the rising edge position of the frequency shift from the other two temperatures. As shown in Table 2, the SR of AAD is not affected by the temperature difference both before and after the frequency shift as that of other algorithms. These results again confirm that the SR performance of AAD is better.
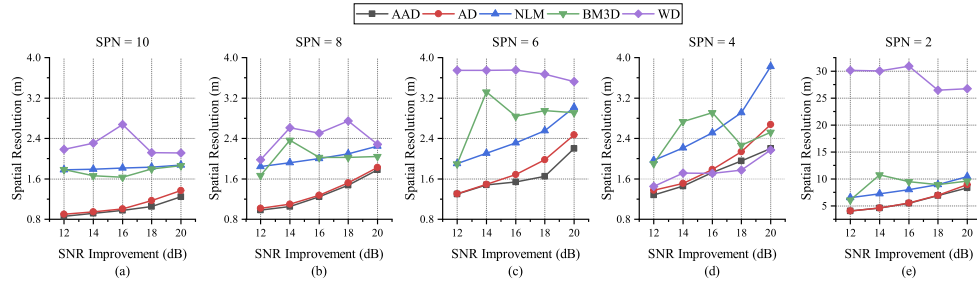
Fig. 8. The spatial resolution of different noise reduction algorithms.

Table 2. Spatial resolution (SR) of different noise reduction algorithms for the 40.05 km fiber

SR (m)		Algorithms				
		AAD	AD	BM3D	NLM	WD
Temp (°C)	40	2.23	2.49	4.02	8.03	12.67
	50	3.25	4.14	7.66	8.64	13.19
	60	2.71	2.84	3.08	5.24	3.82

Table 3. Values of parameters used in the denoising algorithms

Algorithms	Parameters	Setting
AAD	Diffusion constant	1/8
	Diffusion equation	Eq. (7)
	Kernel size	5 × 5
BM3D	Sliding step	2
	Block size	12 × 12
	Search window size	13 × 13
	Max matched blocks	16
	Threshold strategy	Hard & Wiener
NLM	Similarity window size	3 × 3
	Search window size	13 × 13
AD	Diffusion constant	1/8
	Diffusion equation	Eq. (1)
WD	Wavelet	Sym5
	Decomposition level	5
	Threshold strategy	Hard

Next, we investigate the complexity of these algorithms in terms of the data processing time. To achieve this, the runtimes of each algorithm at all SPNs are recorded and then averaged at different SNR improvements with the parameters listed in Table 3. The hardware platform for implementing different noise reduction algorithms is composed of an Intel Core i7-7700HQ CPU @ 2.80 GHz and a 16 GB RAM, and the Python interpreter is Python 3.7. The runtimes for using five different algorithms are presented in Table 4. The results show a longer runtime for a larger SPN for all algorithms, i.e., larger sample numbers of the BOTDA traces. For all five different SPNs, BM3D takes the longest runtime to obtain the results. In descending order, the average time consumption are then followed by AAD, WD, NLM, and AD, where AD takes the shortest

runtime. The runtime of AAD is reasonably good as compared to other algorithms, especially for lower SPNs, and it requires only 3% to 24% of that of BM3D and takes slightly longer than other algorithms.

Table 4. Runtime for applying different noise reduction algorithms

Runtime (s)		SPNs				
		2	4	6	8	10
Algorithms	AAD	0.85	4.45	16.66	30.30	40.60
	BM3D	31.31	64.10	98.73	132.24	165.66
	NLM	0.30	0.86	1.58	2.09	2.57
	AD	0.037	0.08	0.38	0.57	0.58
	WD	1.16	2.23	3.28	4.63	5.81

We would like to point out that the runtime of AD algorithm is proportional to the iteration number needed to process the data. So, if we consider only that number, the time taken by AAD should be 2 to 4 times longer than that by AD. However, AAD suppresses over-smooth points and edge points from participating in the smoothing process, which results in a smaller SNR increase for each iteration than AD. Hence, it requires more iteration to obtain the same SNR improvement as the latter. In addition, AAD also performs AC operation at each iteration, and this additional calculation also prolongs the time consumption.

4. Discussion

In this part, we will further discuss some important issues in the experiment, which includes the impact of the convolution kernel size on algorithm performance, the impact of a low SPN on BFS, the robustness of anisotropic diffusion threshold, and the performance comparison of 1D and 2D algorithms on noise reduction.

4.1. Convolution kernel size

In our study, the size of the convolution kernel has a large impact on AAD. The kernels, which have eight different directions as shown in Fig. 9, can be a square with an odd size greater than or equal to 3 as long, i.e., 3×3, or 7×7.
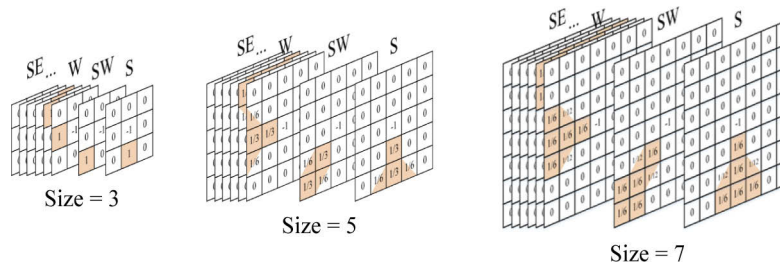

Size = 3   Size = 5   Size = 7

Fig. 9. Convolution Kernels of Different Sizes.

In AD algorithms, which include both AD and AAD, the convolution kernel determines the degree of data relevance at each iteration. In terms of RMSE, a larger kernel has a wider convolution range with more neighboring points when smoothing, and a wider convolution range will improve the RMSE performance. However, when it comes to spatial resolution, a wider convolution range will cause a decrease. As a compromise, a 5×5 convolution kernel was applied

in paper. On the other hand, a larger kernel achieves the same SNR improvement employing a smaller number of iterations as compared to a smaller kernel, which can save runtime.

4.2. RMSE and SR relationship with SPN

The overall trend of RMSE for different SPNs in Fig. 7 manifests that the RMSE drops if SPN decreases except for very low SPN (e.g., 2). The reason is that fewer sampling points along the same length on the optical fiber for RMSE are engaged in RMSE calculation. In such case, the corresponding RMSE is likely to become smaller for lower SPN. However, when SPN = 2 (the lowest SPN used in this study), there is a counter-trend situation and the RMSE increases significantly. This is because all of the BM3D, NLM, AD, and AAD algorithms rely on the neighboring points to judge and reduce the noise of the center point (by matching or convolution). It is assumed that any point in the matrix (BGS image) has continuity with all of or part of its neighboring points (such as edge points). However, too few SPNs may undermine this continuity and cause the deterioration of RMSE. A similar situation also appears in the SR performance.

As shown in Fig. 8, the SR increases slowly if SPN is dropped from 10 to 4. However, the SRs of all algorithms increase abruptly at SPN of 2. This is because when the SPN is too small (e.g., ≤ 2 data points on the system's spatial resolution length, which is 2 m for the 99 km system), the BGS information is greatly lost, leading to a sharp deterioration in the spatial resolution. The abrupt degradation of SR at a low SPN (SPN = 2) is illustrated in Fig. 10 as an example by using WD as this algorithm can obtain the most obvious outcome of denoising a BGS image in lower SPNs. In the BFS curve, since the 2.3 m peak section is quite close to the frequency shift section (only 11.8 m), it becomes over-smoothed and flattened to one single ramp after filtering. Therefore, low SPNs should be avoided in denoising BGSs as it is likely to cause severe SR deterioration.
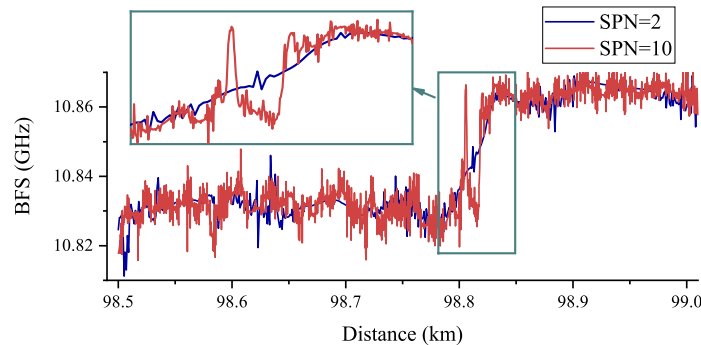

Fig. 10. Denoised BFSs for using WD with 12 dB SNR Improvement at SPNs of 2 and 10.

4.3. Robustness of diffusion threshold

As mentioned in section 2.1, the noise level equal to the diffusion threshold κ would result in the largest smoothing effect. In other words, the noise is under maximum suppression. Generally, the theoretical value of the threshold κ should be equal to the noise standard deviation σ, which is confirmed by the flux function visualized in Fig. 1. However, it is found in our study that for $\kappa \geq \sigma$ with other parameters fixed, the noise reduction effect does not change significantly. In particular, for $\kappa \geq 3\sigma$, the setting of κ has almost no influence on the noise reduction effect.

To study the effect of diffusion threshold κ, the SNR improvement, RMSE, and SR with κ are illustrated in Fig. 11, where SPN is 10 and the iteration number is 22. The AD algorithm is used as an example as it has fewer parameters which prevent other parameters from interfering with

the κ robustness result. The results obtained for using AAD would be expected to be similar. As in Fig. 11, that the SNR improvement increases rapidly for slight increase of κ in the beginning until κ reaches to σ, and such SNR improvement tends to be stabilized for $\kappa \geq 3\sigma$. In addition, the RMSE drops rapidly for the increase of κ values within the range from 0.02σ to σ and becomes stabilized for $\kappa \geq 3\sigma$. It is also shown in Fig. 11 that the spatial resolution varies in a different way which fluctuates a bit until $\kappa = \sigma$ and stabilizes for $\kappa \geq 3\sigma$.
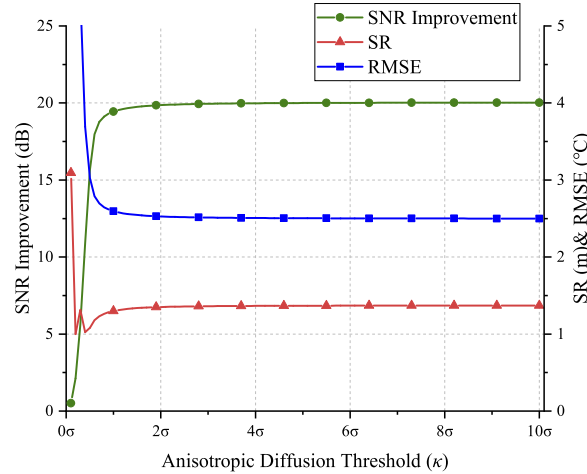

Fig. 11. The SNR Improvement, RMSE and SR using AD with SPN of 10 and iterations of 22.

In fact, the diffusion threshold is designed to distinguish edges from noise. However, the gradient modulus of noise of our experimental data (SNR < 0) has not only the components whose amplitudes are larger than the edge but also those with smaller amplitudes than the edge. So, the edges cannot be distinguished from the noise with only one diffusion threshold. Hence, without considering the edge, the threshold κ should be set so that all the noises fall to the left side of the peak of the flux function to perform best noise reduction which can smooth the points with larger gradients. This means that the use of larger threshold κ performs better noise reduction. Considering different characteristics for edge and noise, the former has spatial continuity in BGS while the latter is mostly scattered on BGS. The combination of the feature of diffusion with AD algorithms helps to check the continuity between a point and its neighbors. Even under the same diffusion threshold, the retention effect is much stronger for the edge than for the noise. In conclusion, in order to be robust, $\kappa = 3\sigma$ is suitable for the AD algorithm family, especially for BGS data having strong noises.

4.4. Performance of 1D and 2D noise reduction algorithms

The performances of 1D and 2D Gaussian noise reduction algorithms with that of the AAD algorithm are also analyzed and compared. For simplicity, the spatial resolution of NLM, WD and other algorithms are not repeated here. Zaslawski et al. has carried out preliminary theoretical and experimental research on noise reduction performance of the 1D and 2D algorithms [19]. But the work does not include the influence of noise reduction algorithms on the sensor's spatial resolution, which is, however, greatly affected by the noise reduction algorithm. Hence, we analyze and compare the effect of 1D and 2D Gaussian and AAD algorithms on SR, full width at half-maximum (FWHM) of the hot spot peak (thus in meter), height of the peak (Fig. 12 inset I), and RMSE. The parameters used in 1D and 2D Gaussian approaches are the same as in Ref. [19]. For the AADs, we set the SNR improvement consistent with the 2D Gaussian noise reduction

algorithm (about 20 dB). The SPN of BGS image is set to be 4, to match that in Ref. [19]. The results are listed in Table 5.
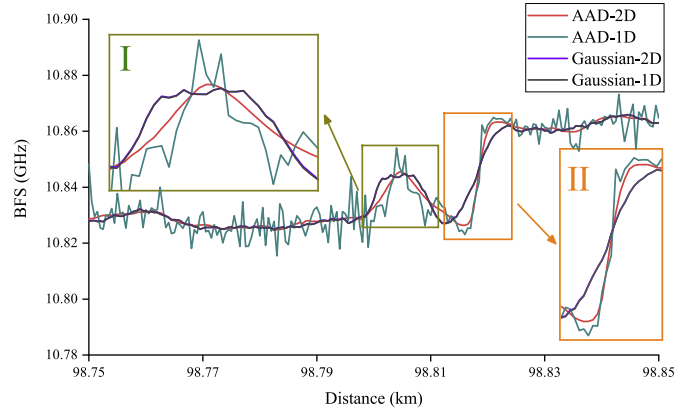

Fig. 12. BFSs obtained by 1D and 2D AAD, and by 1D and 2D Gaussian Filter denoising.

Table 5. SR, FWHM, peak height and RMSE for using different algorithms

Parameters	Algorithm			
	2D-AAD	1D-AAD	2D-Gaussian	1D-Gaussian
SR (m)	2.2	1.58	4.30	4.27
Peak FWHM (m)	7.5	2.25	9	9
Peak Frequency (GHz)	10.8454	10.8540	10.8447	10.8447
RMSE (℃)	1.28	3.02	1.44	1.49

The 2D-AAD performs better than both Gaussian algorithms in determining all the four parameters. In particular, the widths of the peak after applying 1D and 2D Gaussian algorithms are much larger than that of 2D-AAD, indicating that it has suffered greater distortion. The SNR improvement of the two algorithms in the 2D form is 20 dB, and then the algorithms are transformed into a 1D form without changing other parameters. The results are shown in Fig. 12. The BFS curves of the Gaussian methods basically overlap with each other. These results confirm the claim that the noise reduction effects of 1D (along fast axis) and 2D Gaussian noise reduction algorithms are basically the same as stated in Ref. [19]. However, this is not true for AAD. Although the SR, FWHM and peak height of 1D-AAD in Table 5 appear to be better than that of 2D-AAD, its RMSE lags far behind. This is due to the poor noise reduction effect of 1D-AAD, which makes peak I and frequency shift in inset II in Fig. 12 more affected by noise. This effect is also responsible for providing the inaccurate results for SR, FWHM and peak height. Although Lorentzian curve fitting has a certain overlap with the effect of 2D denoising, for AAD which operates in a non-linear way, its 2D form still has a better performance than the 1D counterpart with the same parameters.

Finally, we compare the SNR improvement capabilities of 1D and 2D Gaussian filtering algorithms with that of AAD as shown in Fig. 13. Only the dimension of the algorithm and the parameters that control SNR improvement (the variance for Gaussian filter, and the number of iterations for AAD) are changed. The SNR improvements offered by the 1D algorithms are approximately 1/3 of that by the 2D counterparts. This is due to the fact that the 2D Gaussian filtering requires only a small variance to achieve the same SNR improvement as compared to the 1D Gaussian filter. And 2D-AAD needs a small number of iterations to achieve the same SNR

improvement as compared to 1D-AAD. A smaller variance or a smaller number of iterations is more beneficial for maintaining spatial resolution performance, which is another reason why 2D methods should be recommended.
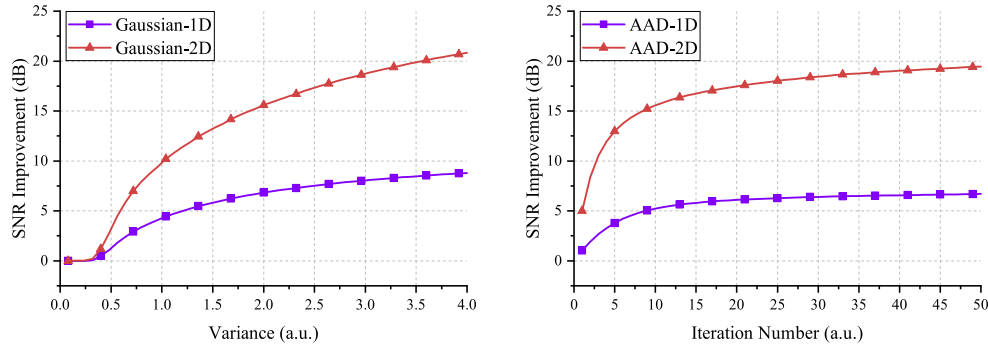

Fig. 13. SNR improvement of BGS for using Gaussian Filters and AADs.

5. Conclusions

In this paper, we have proposed a noise reduction algorithm based on asynchronous control (AC) of anisotropic diffusion (AD), and the proposed asynchronous anisotropic diffusion (AAD) algorithm is applied to improve the SNR of the BGS images of the Brillouin distributed optical fiber sensors. The performances of the proposed AAD algorithm have been analyzed theoretically and validated experimentally in detail for various experimental conditions. The experimental results show that AAD algorithm improves the RMSE performance of Brillouin distributed sensors significantly as compared to AD. The RMSE of the AAD is also comparable to other noise reduction algorithms (e.g., WD, NLM and BM3D) currently used in optical fiber sensing technology. Moreover, the proposed algorithm has a distinguished ability to maintain the spatial resolution of the fiber sensors better than AD, WD, NLM and BM3D algorithms.

We have also discussed some critical issues in denoising sensing signals obtained from Brillouin sensors. Firstly, the effect of noise reduction using 1D and 2D algorithms on the spatial resolution deterioration has been investigated and analyzed in detail, which shows that the 2D denoising algorithm performs better than 1D algorithm in terms of spatial resolution, SNR improvement and processing speed. Secondly, we also found that the use of very low SPN in BGS raw data seriously affects the SR after applying denoising algorithms, which suggests the DAQ's sample rate should be larger than certain values. Finally, although the diffusion threshold κ of the AD algorithm was supposed to be equal to the noise standard deviation σ, our robust analysis indicates that κ should be around 3σ especially when dealing with the BGSs with low SNR.

In summary, the proposed AAD method shows its advantages in terms of SR and RMSE performance, robustness, and controllable effects in BGS image denoising. The proposed asynchronous control method could also be possibly extended and combined with all cellular automata algorithms to find much wider applications. The use of AC algorithms with AD could be competitive in denoising sensing signals obtained from Brillouin as well as other distributed optical fiber sensors to provide highly accurate distributed measurements with a well-preserved spatial resolution.

References

1. A. Minardo, R. Bernini, and L. Zeni, "A simple technique for reducing pump depletion in long-range distributed Brillouin fiber sensors," IEEE Sens. J. **9**(6), 633–634 (2009).
2. A. Motil, A. Bergman, and M. Tur, "State of the art of Brillouin fiber-optic distributed sensing," Opt. Laser Technol. **78**, 81–103 (2016).
3. F. Bastianini, F. Matta, A. Rizzo, N. Galati, and A. Nanni, "Overview of recent bridge monitoring applications using distributed Brillouin fiber optic sensors," J. Nondestruct. Test **12**(9), 269–276 (2007).
4. F. Rodríguez-Barrios, S. Martín-López, A. Carrasco-Sanz, P. Corredera, J. D. Ania-Castañón, L. Thévenaz, and M. González-Herráez, "Distributed Brillouin fiber sensor assisted by first-order Raman amplification," J. Lightwave Technol. **28**(15), 2162–2172 (2010).
5. X. Gao, Z. Yang, S. Wang, X. Hong, X. Sun, M. A. Soto, J. Wu, and L. Thevenaz, "Impact of optical noises on unipolar-coded Brillouin optical time-domain analyzers," Opt. Express **29**(14), 22146–22158 (2021).
6. H. Murayama, K. Kageyama, H. Naruse, A. Shimada, and K. Uzawa, "Application of fiber-optic distributed sensors to health monitoring for full-scale composite structures," J. Intel. Mat. Syst. Str. **14**(1), 3–13 (2003).
7. A. Minardo, R. Bernini, L. Amato, and L. Zeni, "Bridge monitoring using Brillouin fiber-optic sensors," IEEE Sens. J. **12**(1), 145–150 (2012).
8. C. A. Galindez-Jamioy and J. M. Lopez-Higuera, "Brillouin distributed fiber sensors: an overview and applications," J. Sensors **2012**, 1–17 (2012).
9. S. Uchida, E. Levenberg, and A. Klar, "On-specimen strain measurement with fiber optic distributed sensing," Measurement **60**, 104–113 (2015).
10. T. Horiguchi, K. Shimizu, T. Kurashima, M. Tateda, and Y. Koyamada, "Development of a distributed sensing technique using Brillouin scattering," J. Lightwave Technol. **13**(7), 1296–1302 (1995).
11. Y. Xiang, M. R. Foreman, and P. Török, "SNR enhancement in brillouin microspectroscopy using spectrum reconstruction," Biomed. Opt. Express **11**(2), 1020–1031 (2020).
12. R. Bernini, A. Minardo, and L. Zeni, "Dynamic strain measurement in optical fibers by stimulated Brillouin scattering," Opt. Lett. **34**(17), 2613–2615 (2009).
13. M. A. Farahani, M. T. Wylie, E. Castillo-Guerra, and B. G. Colpitts, "Reduction in the number of averages required in BOTDA sensors using wavelet denoising techniques," J. Lightwave Technol. **30**(8), 1134–1142 (2012).
14. M. A. Soto, J. A. Ramirez, and L. Thevenaz, "Intensifying the response of distributed optical fibre sensors using 2D and 3D image restoration," Nat. Commun. **7**(1), 1–11 (2016).
15. H. Wu, L. Wang, Z. Zhao, N. Guo, C. Shu, and C. Lu, "Brillouin optical time domain analyzer sensors assisted by advanced image denoising techniques," Opt. Express **26**(5), 5126–5139 (2018).
16. B. Wang, L. Wang, C. Yu, and C. Lu, "Long-distance BOTDA sensing systems using video-BM3D denoising for both static and slowly varying environment," Opt. Express **27**(25), 36100–36113 (2019).
17. K. Luo, B. Wang, N. Guo, K. Yu, C. Yu, and C. Lu, "Enhancing SNR by Anisotropic Diffusion for Brillouin Distributed Optical Fiber Sensors," J. Lightwave Technol. **38**(20), 5844–5852 (2020).
18. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Machine Intell. **12**(7), 629–639 (1990).
19. S. Zaslawski, Z. Yang, and L. Thévenaz, "On the 2D post-processing of Brillouin optical time-domain analysis," J. Lightwave Technol. **38**(14), 3723–3736 (2020).
20. F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," SIAM J. Numer. Anal. **29**(1), 182–193 (1992).
21. Y. Yu and S. T. Acton, "Speckle reducing anisotropic diffusion," IEEE Trans. on Image Process. **11**(11), 1260–1270 (2002).
22. C. Lopez-Molina, M. Galar, H. Bustince, and B. De Baets, "On the impact of anisotropic diffusion on edge detection," Pattern Recogn. **47**(1), 270–281 (2014).
23. W. Wang, L. Zhu, J. Qin, Y.-P. Chui, B. N. Li, and P.-A. Heng, "Multiscale geodesic active contours for ultrasound image segmentation using speckle reducing anisotropic diffusion," Opt. Laser Eng. **54**, 105–116 (2014).
24. J. Monteil and A. Beghdadi, "A new interpretation and improvement of the nonlinear anisotropic diffusion for image enhancement," IEEE Trans. Pattern Anal. Machine Intell. **21**(9), 940–946 (1999).
25. X. Qian, X. Jia, Z. Wang, B. Zhang, N. Xue, W. Sun, Q. He, and H. Wu, "Noise level estimation of BOTDA for optimal non-local means denoising," Appl. Opt. **56**(16), 4727–4734 (2017).
26. M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," IEEE Trans. on Image Process. **7**(3), 421–432 (1998).
27. L. J. Van Vliet, I. T. Young, and G. L. Beckers, "A nonlinear Laplace operator as edge detector in noisy images," Comput. Vis. Graph. Image Processing **45**(2), 167–195 (1989).

28. N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," IEEE J. Solid-st. Circ. **23**(2), 358–367 (1988).

29. A. K. Azad, F. N. Khan, W. H. Alarashi, N. Guo, A. P. T. Lau, and C. Lu, "Temperature extraction in Brillouin optical time-domain analysis sensors using principal component analysis based pattern recognition," Opt. Express **25**(14), 16534–16549 (2017).

30. A. K. Azad, L. Wang, N. Guo, H.-Y. Tam, and C. Lu, "Signal processing using artificial neural network for BOTDA sensor system," Opt. Express **24**(6), 6769–6782 (2016).

31. M. A. Soto, G. Bolognini, and F. Di Pasquale, "Analysis of optical pulse coding in spontaneous Brillouin-based distributed temperature sensors," Opt. Express **16**(23), 19097–19111 (2008).

32. L. Zhang, "A demo for AAD algorithm, which is modified from BJTUSenosr's AD Algorithm,," GitHub (2022) [accessed 14 January 2022], https://github.com/Lux-Zhang/Asynchronous-Anisotropic-Diffusion-Algorithm.