

Article

# Feature Preserving Autofocus Algorithm for Phase Error Correction of SAR Images

Haemin Lee \* , Chang-Sik Jung and Ki-Wan Kim

Agency for Defense Development, Yuseong P.O. Box 35, Daejeon 34186, Korea; csjung@add.re.kr (C.-S.J.); kiwankim@add.re.kr (K.-W.K.)

\* Correspondence: hmlee@add.re.kr; Tel.: +82-42-821-4412

**Abstract:** Autofocus is an essential technique for airborne synthetic aperture radar (SAR) imaging to correct phase errors mainly due to unexpected motion error. There are several well-known conventional autofocus methods such as phase gradient autofocus (PGA) and minimum entropy (ME). Although these methods are still widely used for various SAR applications, each method has drawbacks such as limited bandwidth of estimation, low convergence rate, huge computation burden, etc. In this paper, feature preserving autofocus (FPA) algorithm is newly proposed. The algorithm is based on the minimization of the cost function containing a regularization term. The algorithm is designed for postprocessing purpose, which is different from the existing regularization-based algorithms such as sparsity-driven autofocus (SDA). This difference makes the proposed method far more straightforward and efficient than those existing algorithms. The experimental results show that the proposed algorithm achieves better performance, convergence, and robustness than the existing postprocessing autofocus algorithms.

**Keywords:** autofocus; feature preserving regularization; phase errors; synthetic aperture radar (SAR)



**Citation:** Lee, H.; Jung, C.-S.; Kim, K.-W. Feature Preserving Autofocus Algorithm for Phase Error Correction of SAR Images. *Sensors* **2021**, *21*, 2370. <https://doi.org/10.3390/s21072370>

Academic Editor: Antonio Iodice

Received: 1 March 2021

Accepted: 23 March 2021

Published: 29 March 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Synthetic aperture radar (SAR) can create high cross-range resolution images through coherent processing of the returned pulses received at different antenna position due to a moving platform, which enables provision for the effect of a large virtual aperture size [1–3]. To achieve full performance of SAR, the exact flight trajectory of the platform should be provided to the signal processor for proper motion compensation. Especially in the case of airborne SAR, the effect of the unexpected platform motion due to the factors such as wind gusts and aircraft vibrations should be compensated before SAR processing. These compensations for airborne SAR are usually performed by using the measured navigation data such as the output of global positioning system (GPS), inertial navigation system (INS), or embedded GPS/INS (EGI) [4]. However, these data also contain the measurement errors due to the inaccuracy of the navigation sensors, and these residual errors cause the phase errors of SAR data, which degrades the quality of the SAR image. There are several methods to solve this kind of image quality degradation, which are usually called autofocus.

Phase gradient autofocus (PGA) is one of the most widely used algorithms to estimate the phase error [5]. The method assumes all the complex reflectivity in the image windowed with appropriate window size, except the center-shifted point target at each range bin, are distributed as zero-mean Gaussian random noises. These assumptions limit the performance of the algorithm in spite of its robustness and fast convergence. The point target and random noise assumption are inappropriate to the scene that contains dominant targets close to each other in the azimuth direction. Quality phase gradient autofocus (QPGA) and generalized phase gradient autofocus (GPGA) are proposed to alleviate this problem [6,7]. However, both QPGA and GPGA use the limited window size, which also limits the bandwidth of the estimated phase error. Therefore, these PGA-based methods

are not suitable for estimating the phase error containing the high frequency components such as the error generated by the GPS measurement update in EGI.

Optimization of sharpness metrics is also a well-known autofocus technique [8,9]. These kinds of methods are used to optimize the cost function, which enables the scene to be well focused. One of the most commonly used cost functions is entropy, and such methods are often called the minimum entropy (ME) method [10–15]. ME-based methods are not restricted by the assumptions in PGA, so there is no limitation on the bandwidth of estimation. There are various ways to achieve the optimization because these methods have no closed-form solutions [8]. One way is to model the estimated error as a polynomial and adjust the coefficients to minimize the entropy of the image [10]. Although this method achieves appropriate performance, the higher order components of the phase errors are difficult to estimate because of the limited order of the estimates, which is similar to the bandwidth issue in PGA. Even though the order of the polynomial can be adaptively increased, it would be computationally intensive. The optimization method based on the trial and error [11] also experiences similar or a more extensive computational problem.

The most widely used ME methods can be divided into two categories. The first is based on gradient searching. One of the well-known methods in this category is the monotonic iterative algorithm (MIA) [12], which minimizes well-defined surrogate functions instead of the entropy. An algorithm based on Newton's method to obtain the phase estimate, which makes the gradient of entropy zero, also provides reasonable performance [13]. These methods can be carried out through the fast Fourier transform (FFT), which enables the fast phase estimation. Another category is based on the fixed-point iteration [14,15]. In these methods, the fixed-point iteration is used to solve the implicit equation of the phase, which makes the derivative of the entropy zero. Every iteration can also be conducted through FFT. The algorithms in these two categories show similar estimation accuracy and convergence.

Regularization-based autofocus techniques are also proposed by some authors [16–22], which are methods that are quite different from postprocessing autofocus such as PGA and ME. These methods are based on a regularized reconstruction of the SAR images, which are often called compressive sensing [22], sparsity-driven imaging [23], or feature-enhanced SAR imaging [24–26]. Unlike the conventional SAR image formation such as the polar format algorithm (PFA) [1], regularization-based imaging methods have advantages such as high resolution, which is not limited by the SAR system bandwidth, and suppression of the artifact owing to speckle and side-lobe, even if the datasets are nonuniform and undersampled. Sparsity-driven autofocus (SDA), one of the well-known autofocus methods in this area, is to minimize the cost function, composed of a fidelity and regularization term, jointly with the estimate of image and the phase error [16]. SDA achieves quite accurate phase estimation with preservation of the advantages of regularized reconstruction. Similar approaches have been proposed such as optimization of cost function, including total variation [17] and modified Tikhonov regularization-based autofocus [18,19] to improve the performance and reduce the computational burden. Although the regularization-based methods achieve high quality SAR images, these are hard to utilize for the SAR mission, which requires on-board processing and a large scene size. The reason is that these methods perform not only the estimation of phase error but also the reconstruction of the image simultaneously, which requires huge computation power. The large computation burden of regularized reconstruction is one of the reasons why PFA is still widely utilized for on-board SAR image formation. If the SAR system is well designed to meet the performance requirements, the only concern for image degradation is the phase error, not the sparse sampling.

In this paper, we propose a new postprocessing autofocus algorithm for phase-corrupted images. The algorithm is designed to minimize the cost function containing regularization term based on  $l_1$ -norm, which is similar to regularization-based autofocus such as SDA. However, the proposed algorithm deals with the processed complex image corrupted by the phase error, whereas SDA deals with returned pulse data before pro-

cessing. This difference makes the proposed algorithm straightforward and requires only simple calculation for FFT and soft-threshold. The equation for minimization of the cost function is carried out through an indirect optimization manner, and fixed-point iteration is used to obtain the optimal solution.

The rest of this paper is outlined as follows. In Section 2, the fundamental background for the proposed algorithm, such as iterative shrinkage thresholding algorithm (ISTA) and denoising [27], are explained. In Section 3, we define the cost function for the proposed method and present the iterative algorithm to achieve the minimization. Then, we demonstrate the performance, convergence, and robustness of the proposed algorithm and compare them with those of the existing autofocus methods such as PGA, GPGA, and ME through experimental results in Section 4. Finally, the conclusions are presented in Section 5.

## 2. Fundamental Background

### 2.1. Phase Error Correction in SAR Images

The typical SAR image formations are achieved through the two-dimensional inverse discrete Fourier transform (DFT) of the motion compensated echo signal formatted to Cartesian coordinates. If we define the range-compressed, motion compensated echo signal as  $G(n, m)$ , where  $n$  and  $m$  are the indices of range cells of the scene and received pulses, respectively, the complex image  $g(n, k)$  is formulated as

$$g(n, k) = \sum_{m=0}^{M-1} G(n, m) \exp\left(j \frac{2\pi}{M} km\right) \quad (1)$$

where  $k$  is the indices of azimuth cells of the scene,  $M$  is the number of received pulses, and  $j^2 = -1$ . The residual phase errors after the motion compensation in the SAR image are mainly due to the errors in the navigation data, and these errors corrupt each returned pulse signal phase. Therefore, most of the phase distortions are in azimuth direction, and the phase-corrected complex image can be formulated as

$$\hat{g}(n, k) = \sum_{m=0}^{M-1} G(n, m) \exp[j\varphi(m)] \exp\left(j \frac{2\pi}{M} km\right) \quad (2)$$

where  $\varphi(m)$  is the estimated residual phase error. If we can obtain an accurate estimation of the phase error, the corrected image is also obtained through (2), which can be carried out through FFT.

### 2.2. Phase Adjustment for Desired Image

If we have the desired complex image and want to obtain the phase adjustment for minimizing the error between the desired and phase adjusted image, we can formulate the following cost function:

$$\begin{aligned} J &= \frac{1}{2} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} |\hat{g}(n, k) - h(n, k)|^2 \\ &= \operatorname{Re} \left[ \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \hat{g}(n, k) h^*(n, k) \right] + \frac{1}{2} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} |\hat{g}(n, k)|^2 + \frac{1}{2} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} |h(n, k)|^2 \end{aligned} \quad (3)$$

where  $h(n, k)$  is the desired complex image and  $N$  is the number of range cells of the scene;  $(\cdot)^*$  and  $\operatorname{Re}(\cdot)$  denotes complex conjugate and the real part of the complex number,

respectively. We assumed that the number of azimuth cells of the scene is equal to  $M$ . Let the derivative with respect to  $\varphi(m)$  be zero to minimize the cost function in (3):

$$\begin{aligned}\frac{\partial J}{\partial \varphi(m)} &= \frac{\partial}{\partial \varphi(m)} \left\{ \operatorname{Re} \left[ \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \hat{g}(n, k) h^*(n, k) \right] \right\} \\ &= \operatorname{Re} \left[ \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial \hat{g}(n, k)}{\partial \varphi(m)} h^*(n, k) \right] = 0\end{aligned}\quad (4)$$

The second and third additive terms in (3) are ignored because the phase does not have an effect to the variation of the total intensity. The derivative of  $\hat{g}(n, k)$  with respect to  $\varphi(m)$  is derived from (2) as given in the following equation:

$$\frac{\partial \hat{g}(n, k)}{\partial \varphi(m)} = jG(n, m) \exp[j\varphi(m)] \exp\left(j\frac{2\pi}{M}km\right)\quad (5)$$

Then, (4) becomes

$$\frac{\partial J}{\partial \varphi(m)} = \operatorname{Im} \left\{ \exp[j\varphi(m)] \sum_{n=0}^{N-1} G(n, m) H^*(n, m) \right\} = 0\quad (6)$$

where  $\operatorname{Im}(\cdot)$  denotes the imaginary part of the complex number and  $H(n, m)$  is defined as

$$H(n, m) = \sum_{k=0}^{M-1} h(n, k) \exp\left(-j\frac{2\pi}{M}km\right)\quad (7)$$

$H(n, m)$  in (7) can be computed through FFT. From (6), the sufficient solution for  $\varphi(m)$  is obtained from the following equation:

$$\varphi(m) = \angle \left[ \sum_{n=0}^{N-1} G^*(n, m) H(n, m) \right]\quad (8)$$

where  $\angle(\cdot)$  denotes the phase of the complex number. The result in (8) will be used in Section 3 to explain the proposed method.

### 2.3. Brief Review of Regularized Reconstruction

SAR image formation is basically solving an inverse problem to obtain the image from the returned signal. If we know the exact SAR observation model, SAR imaging is identical to solving the following equation with respect to  $\mathbf{x}$  [25]:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{v}\quad (9)$$

where  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{v}$  comprise an  $MN \times 1$  vector, which is the column stacked version of the complex image, returned echo, and measurement noise, respectively, and  $\mathbf{C}$  is the discretized observation kernel. The following cost function is usually used to solve the inverse problem in (9) [25]:

$$J = \frac{1}{2} \|\mathbf{y} - \mathbf{C}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p\quad (10)$$

where  $\|\cdot\|_p$  denotes the  $l_p$ -norm and  $\lambda$  is a weighting parameter for regularization term. The first term represents the fidelity term for preserving the dependence on the observation model and the second term is the regularization term for enhancement of the feature of the image. The typical choice for  $p$  is 1 [26]. The optimization can be achieved through two ways, one is the gradient descent algorithm [26] and the other is the iterative shrinkage thresholding algorithm (ISTA) [28]. For  $p = 1$ , ISTA is represented as

$$\mathbf{x}_{i+1} = S_\lambda \left( \mathbf{x}_i - \alpha \mathbf{C}^H (\mathbf{C}\mathbf{x}_i - \mathbf{y}) \right)\quad (11)$$

where  $\alpha$  is a sufficiently small positive number,  $\mathbf{x}_i$  is estimate of  $\mathbf{x}$  in  $i$ -th iteration, and  $S_\lambda(\cdot)$  is a soft-thresholding function defined as

$$S_\lambda(x) = \exp[j(\angle x)](|x| - \lambda)_+ \quad (12)$$

$$\left( (x)_+ \right) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Iteration in (11) achieves the optimal image for (10) if  $\alpha$  is smaller than the largest eigenvalue of  $\mathbf{C}^H \mathbf{C}$ . Unlike the inverse problem of (9) and (10), if we already have the complex image formatted by (1) and want to carry out the feature enhancement in postprocessing,  $\mathbf{x}$  and  $\mathbf{y}$  become the column stacked version of the complex image before and after the postprocessing, respectively, and kernel  $\mathbf{C}$  becomes the identity matrix. Then, (10) is rewritten as

$$J = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (13)$$

$$= \frac{1}{2} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \left( |g(n, k) - \bar{g}(n, k)|^2 + \lambda |\bar{g}(n, k)| \right)$$

where  $\bar{g}(n, k)$  is the postprocessed complex image. Unlike the iterative solution in (11), the optimization of (13) can be achieved by the following the closed-form solution [27]:

$$\bar{g}(n, k) = S_\lambda(g(n, k)) \quad (14)$$

If we approximate  $|\bar{g}(n, k)|$  to be  $[\bar{g}^*(n, k)\bar{g}(n, k) + \varepsilon]^{\frac{1}{2}}$  for a sufficiently small positive scalar  $\varepsilon$ , then it satisfies

$$\left. \frac{\partial J}{\partial \bar{g}(n, k)} \right|_{\bar{g}(n, k) = S_\lambda(g(n, k))} \approx 0 \quad (15)$$

From (14), each element of  $\bar{g}$  whose absolute value is smaller than  $\lambda$  is removed, and the absolute value of the other elements are decrease by  $\lambda$ . This so-called “denoising” [28] is an important procedure for the proposed algorithm described in the next section.

### 3. Proposed Method

In this section, we define a cost function and derive the equation for its optimality in an indirect optimization manner. We then propose an algorithm to obtain the solution for the equation.

#### 3.1. Cost Function and Its Minimization

The phase-corrupted image, owing to the residual phase error, shows distorted features such as degraded resolution and unexpected side-lobes, and hence the goal for phase estimation can be interpreted as to remove these distortions. Therefore, the objective of our method is a phase adjustment to achieve the feature enhancement, which can be realized by concurrent optimization of (3) and (13), i.e., the following cost function:

$$J(\bar{g}, \varphi) = \frac{1}{2} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \left( |\hat{g}(n, k) - \bar{g}(n, k)|^2 + \lambda |\bar{g}(n, k)| \right) \quad (16)$$

where  $\hat{g}$  is the function of  $\varphi$  according to (2). The minimization of (16) enables obtaining the feature-enhanced image  $\bar{g}$  and the phase adjustment  $\varphi$  to make the phase-corrected image close to  $\bar{g}$ . We can achieve the minimization through the following indirect optimization.

Let  $\bar{g}$  be the solution of the necessary condition for the minimum of (16), i.e.,  $\partial J / \partial \bar{g} = 0$ ; then, the solution is obtained from (14) as

$$\bar{g}(n, k) = S_\lambda(\hat{g}(n, k)) \quad (17)$$

If (17) holds for every  $\varphi$ , then we can replace the cost function in (16) by the following indirect cost function:

$$V(\varphi) = \frac{1}{2} \sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \left( |\hat{g}(n, k) - S_\lambda(\hat{g}(n, k))|^2 + \lambda |S_\lambda(\hat{g}(n, k))| \right) \quad (18)$$

The derivative of (18) with respect to  $\varphi$  is

$$\frac{\partial V(\varphi)}{\partial \varphi} = \frac{\partial J(\bar{g}, \varphi)}{\partial \bar{g}} \frac{\partial \bar{g}}{\partial \varphi} + \frac{\partial J(\bar{g}, \varphi)}{\partial \varphi} \approx \frac{\partial J(\bar{g}, \varphi)}{\partial \varphi} \quad (19)$$

because  $\partial J / \partial \bar{g} \approx 0$  for  $\bar{g}$  in (17). Let (19) to be zero for minimization of  $V(\varphi)$ , then the condition for optimality can be derived by using the result in (6) as given in the following equation:

$$\frac{\partial J}{\partial \varphi(m)} = \text{Im} \left\{ \exp[j\varphi(m)] \sum_{n=0}^{N-1} G(n, m) \bar{G}^*(n, m) \right\} = 0 \quad (20)$$

where  $\bar{G}(n, m)$  is defined as

$$\begin{aligned} \bar{G}(n, m) &= \sum_{k=0}^{M-1} \bar{g}(n, k) \exp(-j\frac{2\pi}{M}km) \\ &= \sum_{k=0}^{M-1} S_\lambda(\hat{g}(n, k)) \exp(-j\frac{2\pi}{M}km) \end{aligned} \quad (21)$$

The solution of (20) is the final phase error estimate of our proposed method.

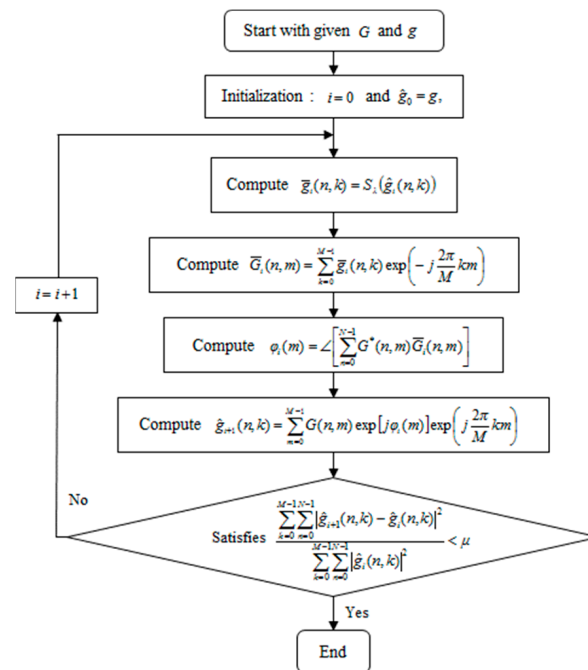
### 3.2. Algorithm

The sufficient condition for (20) can be written in a manner similar to that in (8):

$$\varphi(m) = \angle \left[ \sum_{n=0}^{N-1} G^*(n, m) \bar{G}(n, m) \right] \quad (22)$$

Equation (22) is not a closed-form solution of  $\varphi(m)$  unlike (8), because  $\bar{G}(n, m)$  is still the function of  $\varphi(m)$ . This implicit equation can be solved by the fixed-point iteration method, which has been applied to solve the similar problems for ME [13,14]. The flowchart of the proposed algorithm is represented in Figure 1. The computation of (2) and (21) in the algorithm can be carried out through FFT, which enables a fast iteration.

In every iteration,  $\bar{g}_i$  takes the same role of the desired complex image, i.e.,  $h$  in (7), which is the denoised image of  $\hat{g}_i$ . Because of the soft-thresholding function in (17), a few elements of  $\hat{g}_i$  whose absolute value is larger than  $\lambda$  are preserved in  $\bar{g}_i$  and the others are set to be zero. We call these preserved elements “features” in this paper. These remained features are the reference for the phase estimation in every iteration. In other words, the phase adjustment is achieved to fit the given image to the feature-preserved image. Therefore, we refer to the proposed algorithm as “feature preserving autofocus (FPA)” in this paper. Since the existing regularization-based autofocus algorithms such as SDA deal with the returned pulse data before processing, image formation algorithms such as the gradient descent algorithm and ISTA described in Section 2.3 are required for every iteration. Therefore, these algorithms require huge computation for a large scene size. Meanwhile, the proposed FPA requires only one soft-thresholding to obtain the reference image for phase adjustment. This simple iteration sequence makes the proposed algorithm more suitable for on-board SAR processing.



**Figure 1.** Flowchart of the proposed algorithm.

### 3.3. Selection of the Threshold

It is worth noting that a careful choice of the threshold, i.e.,  $\lambda$ , is recommended for the proposed FPA. A choice of small  $\lambda$  enables the features to contain a sufficiently large number of scatterers, which achieves accurate estimation. However, if  $\lambda$  is too small, the remaining features would also contain the artifacts of the SAR image such as side-lobes even though their magnitude are reduced. Thus, it may cause low speed of convergence, and insufficient estimation accuracy if the artifacts still remain in the features at the final iteration. On the other hand, if  $\lambda$  is too large, some dominant scatterers for phase estimation would be removed from the features. In this case, the estimation is too concentrated in a few remained strong scatterers, which leads to local optimal estimation even if it enables to achieve faster convergence rate.

To solve these tradeoffs, we propose varying the threshold to satisfy both fast convergence and optimal performance. The fast convergence can be achieved by setting a large  $\lambda$  at the initial iteration loop, and the accurate estimation can be carried out through gradually decreasing the value of  $\lambda$  for the rest of the iterations. For example, we suggest a simple asymptotically decreasing model for  $\lambda$ :

$$\lambda_i = \alpha \lambda_{i-1} = \alpha^i \lambda_0 \quad (23)$$

where  $\alpha$  and  $\lambda_i$  represent the positive forgetting factor less than 1 and the threshold for  $i$ -th iteration, respectively. The threshold for the initial loop, i.e.,  $\lambda_0$ , would be a user-defined sufficiently large positive value, but small enough to achieve an appropriate number of features containing the main scatterers. There are various ways to select the main scatterers, such as the methods in [6,7,13]. Any of these methods would work to select the appropriate initial threshold. Alternatively, we observed that the proposed algorithm with an initial threshold slightly smaller than the maximum amplitude of the image, which enables the features to contain at least one scatterer, performs sufficient convergence and performance for every case in this paper. The performance comparison for constant and varying threshold is described in the next section.

#### 4. Experimental Results

In this section, some experimental results are demonstrated to verify the benefits of our proposed algorithm. In Section 4.1, we verify the performance and characteristics of the proposed method with a different threshold  $\lambda$ , and explain the tradeoff between the convergence and accuracy. Then, we compare the results for the constant threshold with those of the varying threshold to compromise the tradeoff, which is introduced in the previous section. In Section 4.2, we demonstrate the convergence and the performance of the proposed method with various types of the phase error, and compare them with those of the existing autofocus algorithms. PFA is utilized for SAR imaging to all the experimental results. The quantitative measures to verify the performance of the autofocus algorithm are defined as

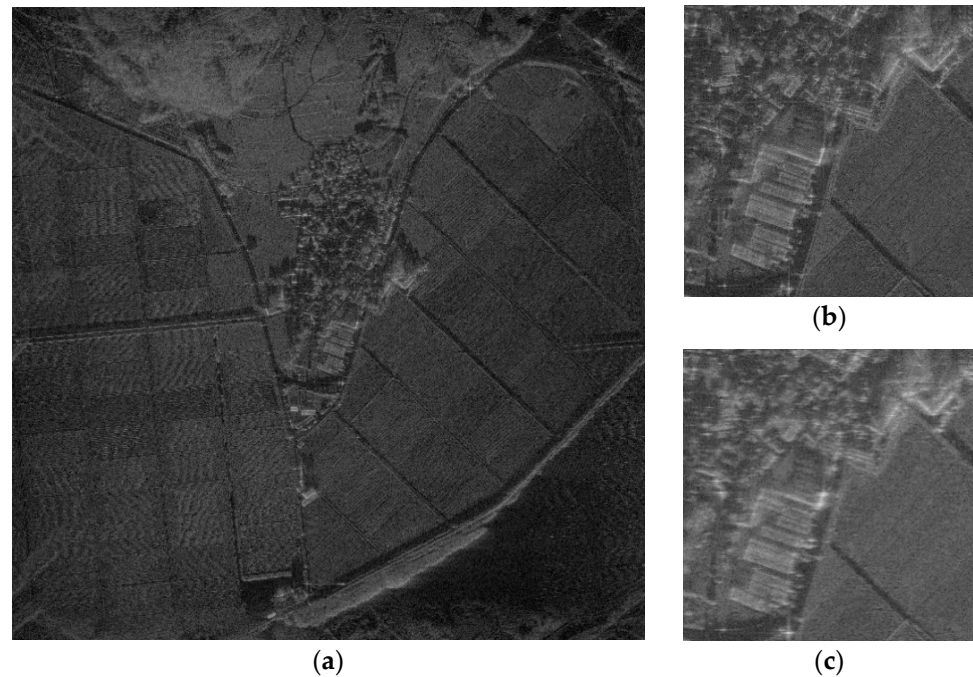
$$\begin{aligned}
 IC(\hat{g}) &= \frac{\sqrt{E\left(\left[|\hat{g}(n,k)| - E\{|\hat{g}(n,k)|\}\right]^2\right)}}{E[|\hat{g}(n,k)|]} \\
 IE(\hat{g}) &= -\sum_{k=0}^{M-1} \sum_{n=0}^{N-1} \frac{|\hat{g}(n,k)|^2}{E[|\hat{g}(n,k)|^2]} \ln\left(\frac{|\hat{g}(n,k)|^2}{E[|\hat{g}(n,k)|^2]}\right)
 \end{aligned} \tag{24}$$

where  $E(\cdot)$  represents the spatial mean operator, and  $IC(\hat{g})$  and  $IE(\hat{g})$  represent the contrast and entropy of the image  $\hat{g}$ , respectively. The initial image  $g$  is scaled to have a maximum magnitude of 1 for all the cases of the experiment, which makes the threshold  $\lambda$  for the soft-threshold function within 0 to 1. The constant  $\mu$  for stop criteria is set to be  $10^{-4}$  for all cases.

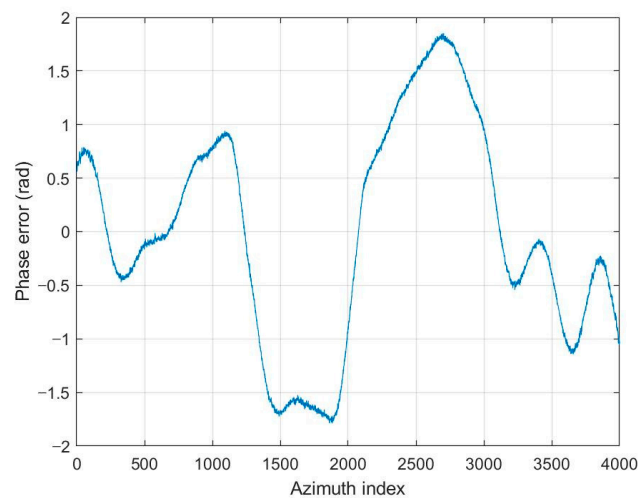
##### 4.1. Performance and Convergence of the Proposed Method

The SAR image used for the proposed FPA is shown in Figure 2a. The size of the image is  $4000 \times 4000$  pixels. The vertical and horizontal coordinates of the image represent the range and azimuth, respectively. The definitions of coordinates are the same for all of the other SAR images in this paper. Figure 2b,c shows the image enlarged at the point near the center of the image both without and with the phase error of Figure 3, respectively. The images corrected by FPA are shown in Figure 4, and the variation of contrast and entropy at each iteration are represented in Figure 5. The images in Figure 4a–d are the results for a constant threshold  $\lambda$  fixed by 0.01, 0.1, 0.3, and 0.9, respectively. Although the images look similar to each other, their contrast and entropy are slightly different, as shown in Figure 5 and Table 1. For a small value of  $\lambda$ , the convergence rate is low, even though the measures tend to converge to its optimal value. On the contrary, a large value of  $\lambda$  enables fast convergence, whereas the final performances are degraded. The reason for these characteristics can be explained by the number of features in Figure 5c. It shows that the smaller value of the threshold applied, the larger the number of features the algorithm uses for each iteration. As mentioned in the previous section, the features may contain the artifacts if the number of features is too large, which causes an adverse effect on the convergence rate. In the case of a small number of features, on the other hand, some dominant scatterers are omitted and it may interfere with the global optimality of the estimation, even if it achieves fast convergence.





**Figure 2.** Synthetic aperture radar (SAR) image constructed by polar format algorithm (PFA). (a) Full size image. (b) Enlarged image without the phase error. (c) Enlarged image with the phase error.



**Figure 3.** Phase error applied to the SAR image in Figure 2c.

As explained in the previous section, we applied a varying threshold to the proposed FPA to satisfy both fast convergence and optimal performance. We set the threshold  $\lambda$  for initial iteration to a relatively large value, and gradually decrease the value at each iteration as in (23). The initial threshold  $\lambda_0$  and the forgetting factor  $\alpha$  are set to 0.9 and 0.5, respectively, and the corrected image from this method is shown in Figure 4e. As shown in Figure 5a,b, the quantitative measures reach the optimal value with an appropriate iteration number. Unlike the fixed threshold cases, the number of features increases for each iteration, as shown in Figure 5c, which enables the features of the current iteration to contain the dominant scatterers omitted at the previous iteration.

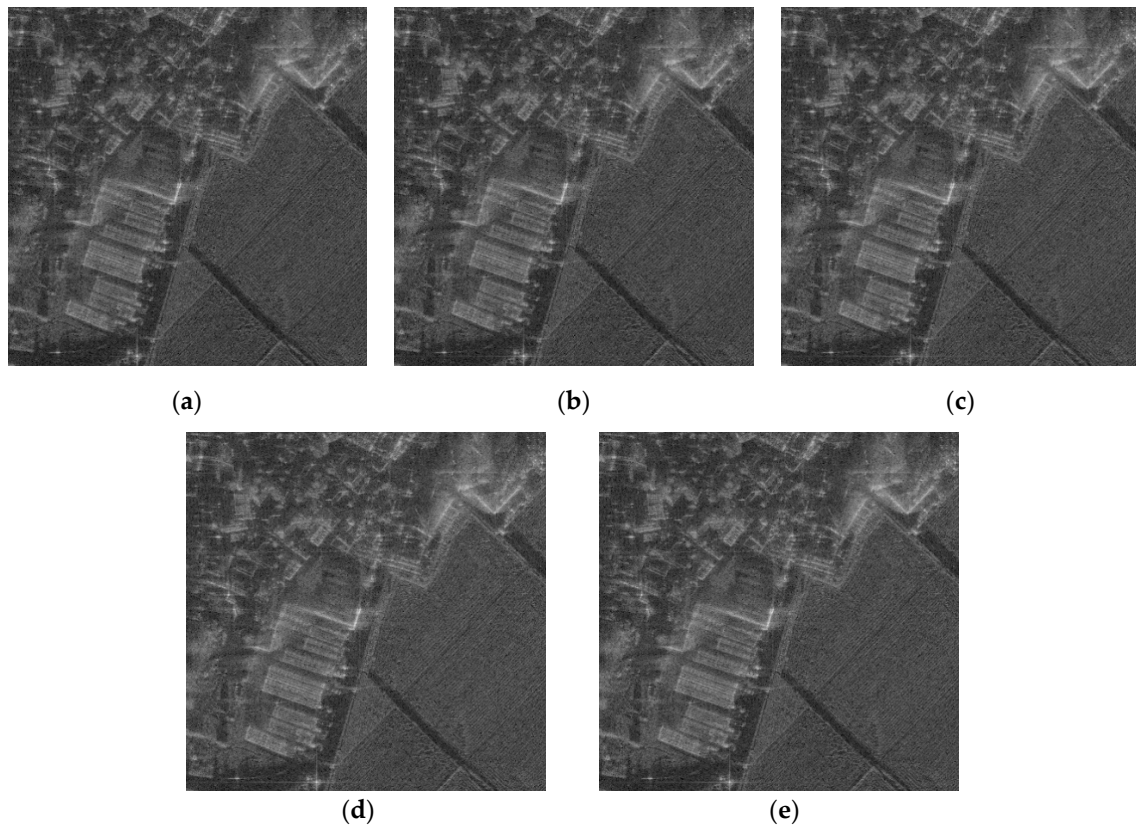


Figure 4. SAR image corrected by proposed FPA with the threshold set at (a) 0.01, (b) 0.1, (c) 0.3, (d) 0.9, (e) varying.

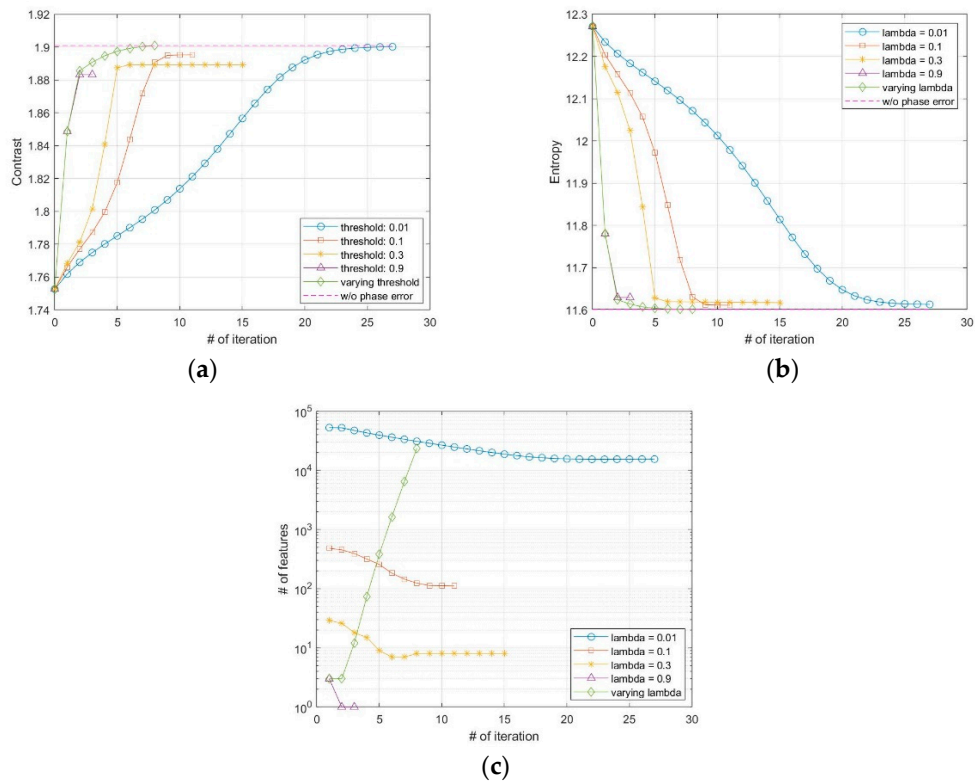


Figure 5. Variation of the quantitative measures of (a) contrast, (b) entropy, and (c) the number of features at each iteration.

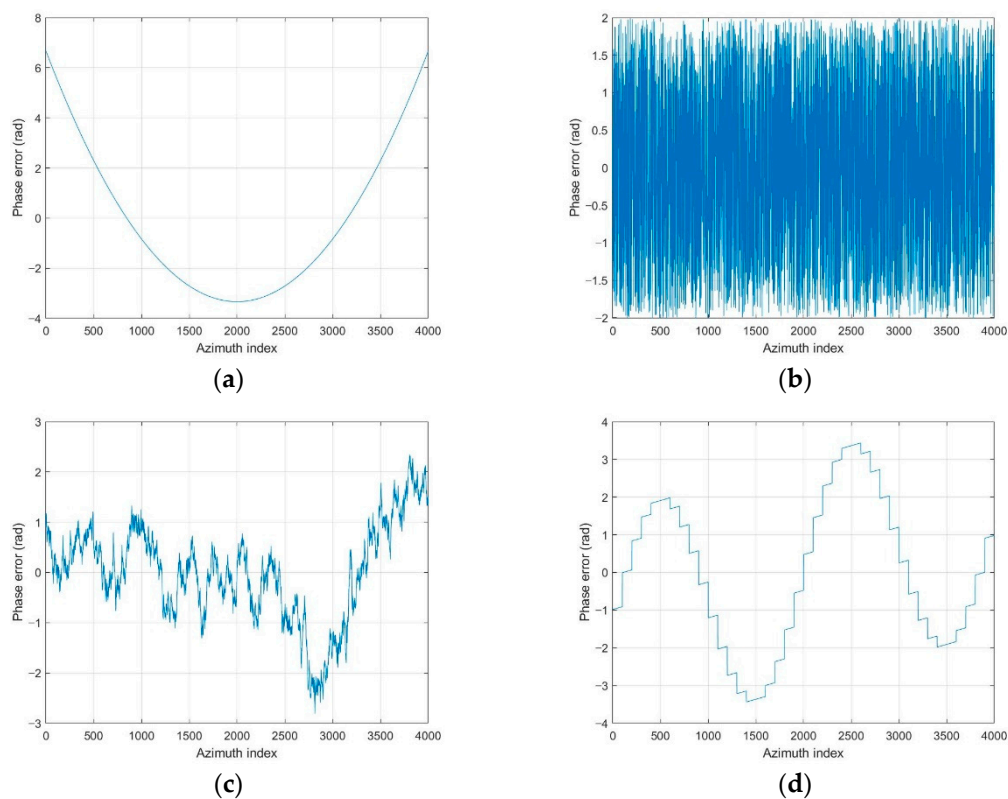
**Table 1.** Quality of the images with different conditions.

Conditions for Image Formation	Contrast	Entropy
Original image reconstructed by PFA	1.901	11.601
Phase-corrupted image	1.753	12.271
Image corrected by FPA with $\lambda = 0.01$	1.900	11.613
Image corrected by FPA with $\lambda = 0.1$	1.895	11.611
Image corrected by FPA with $\lambda = 0.3$	1.889	11.617
Image corrected by FPA with $\lambda = 0.9$	1.883	11.630
Image corrected by FPA with varying $\lambda$	1.901	11.601

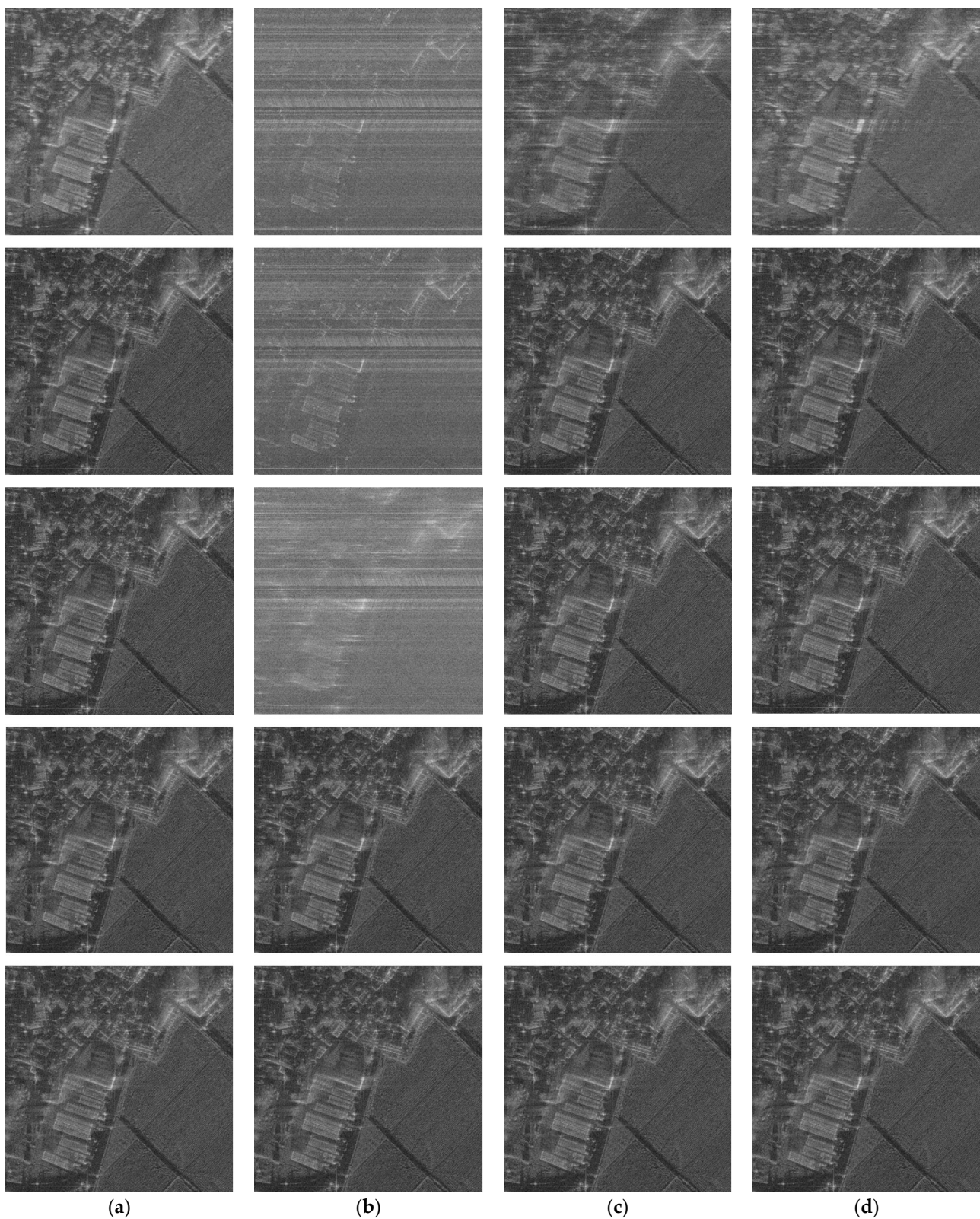
#### 4.2. Comparison with Existing Autofocus Algorithms

We have compared the proposed method with existing postprocessing autofocus methods of PGA [5], GPGA [7], and ME [12]. Constant false alarm rate (CFAR) detection was used to select the strongest scatterers for GPGA, which is a modified algorithm of PGA. The stop conditions for PGA, GPGA, and ME were the same as that of proposed FPA, and we used the varying threshold for FPA as described in Section 3.3. The value of  $\lambda_0$  and  $\alpha$  for FPA were the same as the previous experiment.

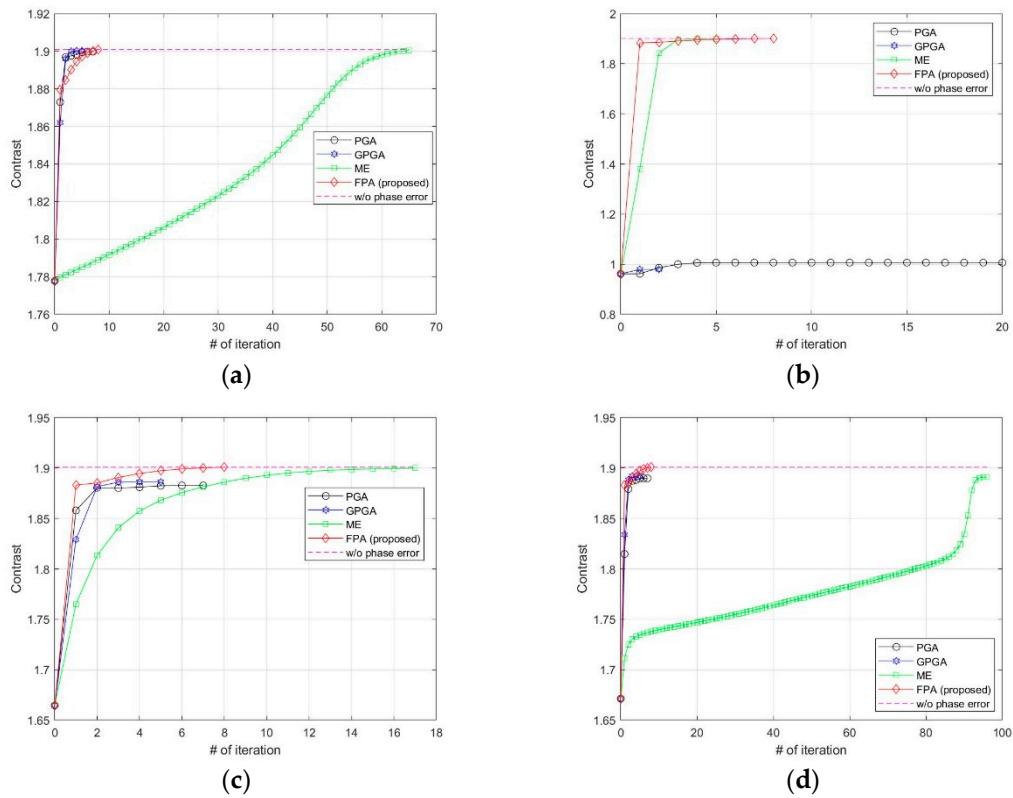
We added various types of phase errors as shown in Figure 6 to the scene in Figure 2a. The images corrected by PGA, GPGA, ME, and the proposed FPA are shown in Figure 7. The contrast and entropy variation for each iteration are represented in Figures 8 and 9, respectively. The performance measures of the images for each method and phase error are represented in Table 2, and the number of iteration and total computing time are shown in Table 3. The computation time was measured with a workstation equipped with Intel<sup>®</sup> Xeon<sup>®</sup> Gold 6140 CPU. MATLAB was used as the programming language.



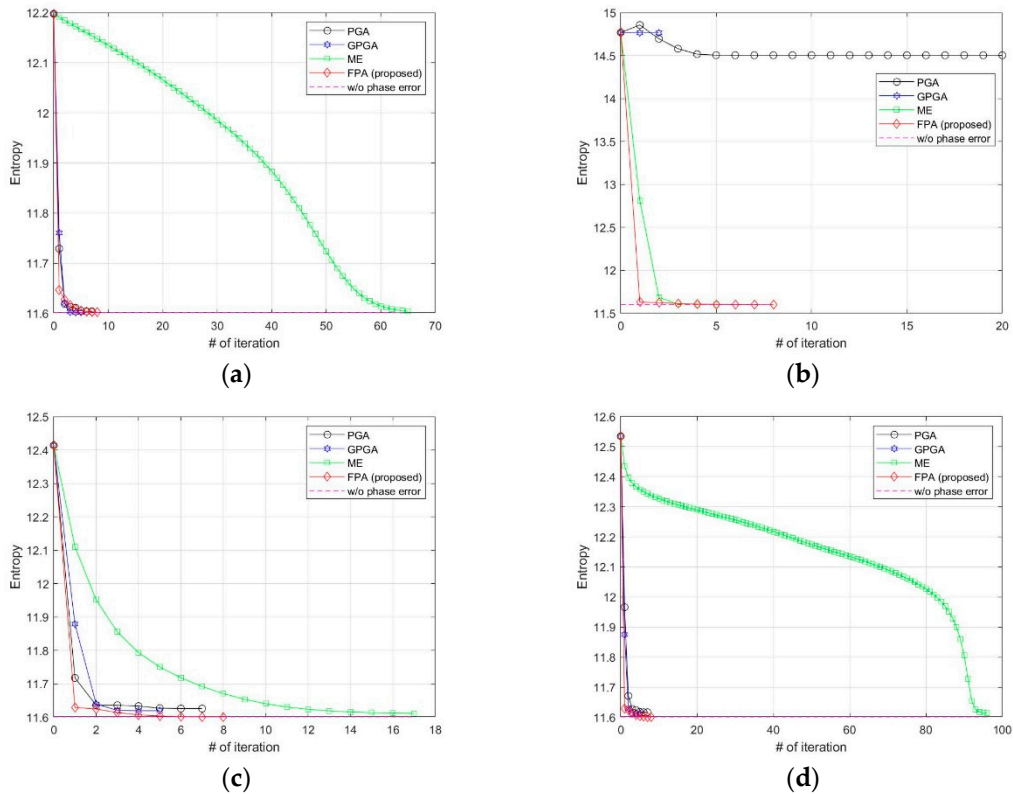
**Figure 6.** Phase errors to verify the performance of the autofocus algorithms. (a) Quadratic error. (b) Uniformly distributed random error. (c) Wiener process error. (d) Sinusoidal error with discontinuity.



**Figure 7.** Phase-corrected images for different types of errors and autofocus algorithms: first row—without autofocus; second row—phase gradient autofocus (PGA); third row—generalized phase gradient autofocus (GPGA); fourth row—minimum entropy (ME); fifth row—proposed FPA. (a) Results for quadratic error. (b) Results for uniformly distributed random error. (c) Results for Wiener process noise. (d) Results for sinusoidal error with discontinuity.



**Figure 8.** Contrast variations at each iteration for (a) quadratic error, (b) uniformly distributed random error, (c) Wiener process error, (d) sinusoidal error with discontinuity.



**Figure 9.** Entropy variations at each iteration for (a) quadratic error, (b) uniformly distributed random error, (c) Wiener process error, (d) sinusoidal error with discontinuity.

**Table 2.** Comparison of performance for different autofocus algorithms.

Autofocus Algorithm	Contrast				Entropy			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
No autofocus	1.778	0.961	1.665	1.671	12.198	14.768	12.414	12.535
PGA	1.900	1.006	1.883	1.890	11.604	14.504	11.626	11.617
GPGA	1.900	0.979	1.886	1.892	11.602	14.764	11.620	11.612
ME	1.900	1.900	1.900	1.891	11.604	11.604	11.612	11.614
Proposed FPA	1.901	1.901	1.901	1.901	11.603	11.601	11.601	11.601

(a)–(d) denote the types of phase errors, same as Figure 6.

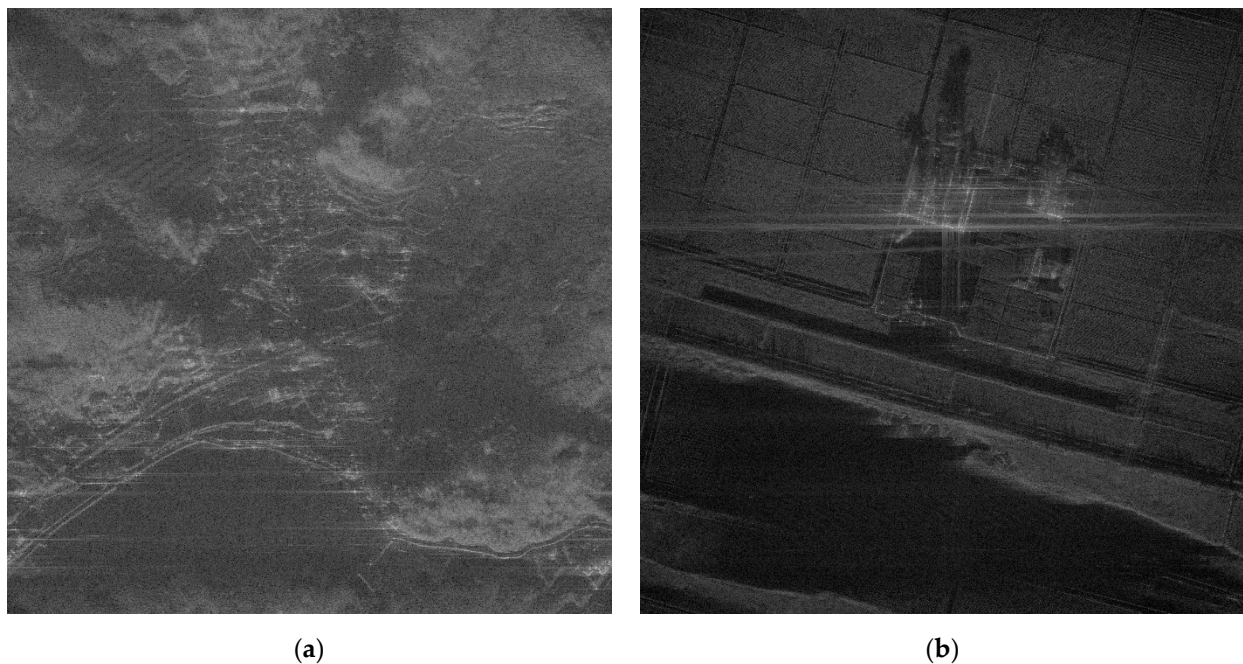
**Table 3.** Comparison of the number of iterations and computation time for different autofocus algorithms.

Autofocus Algorithm	Number of Iterations				Computation Time (s)			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
No autofocus	-	-	-	-	-	-	-	-
PGA	7	20	7	7	51.7	146.8	58.6	55.9
GPGA	5	2	5	5	77.0	37.0	89.9	84.5
ME	65	5	17	96	252.6	22.3	65.5	344.7
Proposed FPA	8	8	8	8	33.9	31.9	33.4	26.8

(a)–(d) denote the types of phase errors, same as Figure 6.

It is well-known fact that PGA shows fast convergence and sufficient performance for low-frequency errors, such as the quadratic error shown in Figure 6a, which can be observed from the results in Figures 7a, 8a and 9a, and Tables 2 and 3. GPGA shows almost the same result with PGA with less iteration, but requires more computation time due to the additional process such as the strongest peak selection through CFAR for every iteration. The proposed method shows a similar trend with less computation time, because both PGA and GPGA require additional procedure for center-shifting, windowing, elimination of linear phase, etc. Meanwhile, ME requires more iterations due to the large phase error, even though it slowly converges to the optimal performance as shown in those figures. Unlike the quadratic error case, the image with random phase error cannot be corrected by PGA and GPGA, as shown in Figures 7b, 8b and 9b, and Tables 2 and 3. It is a natural result because of the assumptions and limited window size of PGA, described in Section 1. Meanwhile, ME and FPA show nearly optimal performance with few iterations. Therefore, it can be inferred from these results that the performance and convergence rate of the proposed FPA are not limited by the bandwidth of the phase error, unlike PGA and ME.

In a practical SAR system, Wiener process and discontinuous phase error can occur because of the navigation systems for motion compensation. If we use an INS system, the Wiener process errors are generated due to the integration of IMU measurements that contain the Gaussian white noises. Furthermore, the navigation data experience some discontinuities if the system uses GPS measurement updates. The phase errors in Figure 6c,d are this kind of error, and the autofocus results for these errors are represented in (c) and (d) of Figures 7–9, and Tables 2 and 3. These results verify that FPA also shows the best performance with appropriate iteration number and computation time. Therefore, we verified the performance and convergence of the proposed FPA as well as its robustness for most types of phase errors through these results. We performed the same procedure for two more scenes in Figure 10 to verify the reliability of the algorithm. Scenes A and B are selected to have a higher and lower value of entropy, i.e., lower and higher contrast, than those of the image in Figure 2a. The results for these two scenes are represented in Tables 4–7, and similar trends of performance and convergence are observed when compared to that of the results in Tables 2 and 3. FPA shows best performance with sufficiently small iterations and computation time for all cases, and it verifies that FPA produces reliable performance.



**Figure 10.** Additional scene to verify the reliability of FPA: (a) scene A and (b) scene B.

**Table 4.** Comparison of performance for different autofocus algorithms for scene A.

Autofocus Algorithm	Contrast				Entropy			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
No autofocus	1.590	0.873	1.519	1.544	12.965	15.209	13.164	13.194
PGA	1.616	0.893	1.614	1.619	12.715	15.269	12.710	12.702
GPGA	1.616	0.930	1.616	1.620	12.713	14.908	12.704	12.698
ME	1.615	1.626	1.624	1.618	12.726	12.690	12.699	12.821
Proposed FPA	1.624	1.626	1.626	1.626	12.707	12.687	12.687	12.687

(a)–(d) denote the types of phase errors, same as Figure 6.

**Table 5.** Comparison of the number of iterations and computation time for different autofocus algorithms for scene A.

Autofocus Algorithm	Number of Iterations				Computation Time (s)			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
No autofocus	-	-	-	-	-	-	-	-
PGA	6	3	7	7	49.8	24.9	59.3	61.0
GPGA	6	6	6	6	101.4	101.5	111.8	114.5
ME	83	12	23	99	325.9	52.4	109.9	465.0
Proposed FPA	10	9	9	9	35.7	39.4	40.6	46.2

(a)–(d) denote the types of phase errors, same as Figure 6.

**Table 6.** Comparison of performance for different autofocus algorithms for scene B.

Autofocus Algorithm	Contrast				Entropy			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
No autofocus	9.087	4.053	8.046	8.306	6.540	10.951	6.934	7.139
PGA	9.624	4.261	9.314	9.466	5.611	10.371	5.652	5.634
GPGA	9.622	4.333	9.349	9.485	5.612	10.175	5.646	5.631
ME	9.634	7.812	9.623	9.459	5.617	6.058	5.627	5.663
Proposed FPA	9.647	9.647	9.647	9.640	5.614	5.612	5.612	5.649

(a)–(d) denote the types of phase errors, same as Figure 6.

**Table 7.** Comparison of the number of iterations and computation time for different autofocus algorithms for scene B.

Autofocus Algorithm	Number of Iterations				Computation Time (s)			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
No autofocus	-	-	-	-	-	-	-	-
PGA	7	10	7	7	49.9	65.3	53.6	55.8
GPGA	6	5	7	7	108.3	97.1	110.2	118.8
ME	70	5	16	94	201.2	11.5	67.6	420.5
Proposed FPA	9	9	9	8	25.6	27.6	25.4	35.7

(a)–(d) denote the types of phase errors, same as Figure 6.

## 5. Conclusions

In this paper, we proposed and demonstrated a new autofocus method for post-processing of a phase-corrupted SAR image based on minimization of the cost function that consists of fidelity and regularization terms. The equation to achieve the optimality is derived by indirect optimization, and the algorithm to solve the equation is proposed. Each iteration in the proposed algorithm requires only one soft-thresholding for the reference image formation, and it enables more efficient processing than the existing regularization-based autofocus such as SDA. The tradeoff between the performance and convergence for the proposed FPA can be compromised by selecting proper constant threshold or using an asymptotically decreasing threshold with appropriate initial value and forgetting factor. The experimental results verified its better performance, convergence, and robustness when compared to the existing methods of PGA and ME. We verified the reliability of the proposed method by performing additional two experiments with a different scene.

Although the proposed FPA shows sufficient performance and convergence for these experiments, there are still remaining factors to improve, such as selection of features and determining the threshold for each iteration. These factors would depend on the scene. Hence, the modifications through adaptive methods would improve the performance and convergence of the proposed algorithm, which are the future works for this study.

**Author Contributions:** Conceptualization, H.L.; Data curation, H.L. and C.-S.J.; Formal analysis, H.L. and K.-W.K.; Investigation, H.L. and C.-S.J.; Methodology, H.L. and K.-W.K.; Supervision, K.-W.K.; Validation, C.-S.J.; Writing-original draft, H.L.; Writing-review & editing, C.-S.J. and K.-W.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Carrara, W.G.; Goodman, R.S.; Majewski, R.M. *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*; Artech House: Norwood, MA, USA, 1995.
2. Jakowatz, C.V.; Wahl, D.E.; Eichel, P.H.; Ghiglia, D.C.; Thompson, P.A. *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*; Springer: New York, NY, USA, 2012.
3. Cumming, I.G.; Wong, F.H. *Digital Processing of Synthetic Aperture Radar Data: Algorithms and Implementation*; Artech House: Norwood, MA, USA, 2005.
4. Kim, T.J. Motion measurement for high-accuracy real-time airborne SAR. *Proc. SPIE* **2004**, *5410*, 36–44.
5. Wahl, D.E.; Eichel, P.H.; Ghiglia, D.C.; Jakowatz, C.V. Phase gradient autofocus-A robust tool for high resolution SAR phase correction. *IEEE Trans. Aerosp. Electron. Syst.* **1994**, *30*, 827–835. [[CrossRef](#)]
6. Chan, H.L.; Yeo, T.S. Noniterative quality phase-gradient autofocus (QPGA) algorithm for spotlight SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **1998**, *36*, 1531–1539. [[CrossRef](#)]
7. Evers, A.; Jackson, J. A Generalized Phase Gradient Autofocus Algorithm. *IEEE Trans. Comput. Imaging* **2019**, *5*, 606–619. [[CrossRef](#)]



8. Fienup, J.R.; Miller, J.J. Abberation correction by maximizing generalized sharpness metrics. *J. Opt. Soc. Am. A Opt. Image Sci.* **2003**, *20*, 609–620. [[CrossRef](#)] [[PubMed](#)]
9. Fienup, J.R. Synthetic-aperture radar autofocus by maximizing sharpness. *Opt. Lett.* **2000**, *25*, 221–223. [[CrossRef](#)] [[PubMed](#)]
10. Wang, J.; Liu, X. SAR minimum-entropy autofocus using an adaptive-order polynomial model. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 512–516. [[CrossRef](#)]
11. Li, X.; Liu, G.; Ni, J. Autofocusing of ISAR images based on entropy minimization. *IEEE Trans. Aerosp. Electron. Syst.* **1999**, *35*, 1240–1252. [[CrossRef](#)]
12. Kragh, T.J.; Kharbouch, A.A. Monotonic iterative algorithm for minimum-entropy autofocus. In Proceedings of the Adaptive Sensor Array Processing (ASAP) Workshop, Lexington, MA, USA, 6–7 June 2006.
13. Zeng, T.; Wang, R.; Li, F. SAR Image Autofocus Utilizing Minimum-Entropy Criterion. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1552–1556. [[CrossRef](#)]
14. Wang, J.; Liu, X.; Zhou, Z. Minimum-entropy phase adjustment for ISAR. *IEEE Proc. Radar Sonar Navig.* **2004**, *151*, 203–209. [[CrossRef](#)]
15. Huang, X.; Ji, K.; Leng, X.; Dong, G.; Xing, X. Refocusing Moving Ship Targets in SAR Images Based on Fast Minimum Entropy Phase Compensation. *Sensors* **2019**, *19*, 1154. [[CrossRef](#)]
16. Önhon, N.O.; Çetin, M. A sparsity-driven approach for joint SAR imaging and phase error correction. *IEEE Trans. Image Process.* **2012**, *21*, 2075–2088. [[CrossRef](#)]
17. Ugur, S.; Arıkan, O. SAR image reconstruction and autofocus by compressed sensing. *Digit. Signal Process.* **2012**, *22*, 923–932. [[CrossRef](#)]
18. Hao, J.; Chen, G.; Cao, Z.; Yin, W.; Zhao, Q. Image reconstruction algorithm for EMT based on modified Tikhonov regularization method. In Proceedings of the 2012 IEEE International Instrumentation and Measurement Technology Conference, Graz, Austria, 13–16 May 2012; pp. 2507–2510.
19. Kang, M.S.; Kim, K.T. Compressive sensing based SAR imaging and autofocus using improved Tikhonov regularization. *IEEE Sens. J.* **2019**, *19*, 5529–5540. [[CrossRef](#)]
20. Kantor, J.M. Polar Format-Based Compressive SAR Image Reconstruction with Integrated Autofocus. *IEEE Geosci. Remote Sens.* **2019**, *58*, 3458–3468. [[CrossRef](#)]
21. Zhao, L.; Wang, L.; Bi, G.; Li, S.; Yang, L.; Zhang, H. Structured sparsity-driven autofocus algorithm for high-resolution radar imagery. *Signal Proc.* **2016**, *125*, 376–388. [[CrossRef](#)]
22. Wei, S.J.; Zhang, X.L.; Shi, J. An autofocus approach for model error correction in compressed sensing SAR imaging. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012; pp. 3987–3990.
23. Donoho, D.L. Compressed sensing. *IEEE Trans. Inform. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
24. Çetin, M.; Stojanovic, I.; Önhon, O.; Varshney, K.; Samadi, S.; Karl, W.C.; Willsky, A.S. Sparsity-driven synthetic aperture radar imaging: Reconstruction, autofocusing, moving targets, and compressed sensing. *IEEE Signal Process. Mag.* **2014**, *31*, 27–40. [[CrossRef](#)]
25. Çetin, M.; Karl, W.C. Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization. *IEEE Trans. Image Process.* **2001**, *10*, 623–631. [[CrossRef](#)]
26. Çetin, M.; Karl, W.C.; Willsky, A.S. Feature-preserving regularization method for complex-valued inverse problems with application to coherent imaging. *Opt. Eng.* **2006**, *45*. [[CrossRef](#)]
27. Daubechies, I.; Defrise, M.; De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **2004**, *57*, 1413–1457. [[CrossRef](#)]
28. Donoho, D.L. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory* **1995**, *41*, 613–627. [[CrossRef](#)]